UNIVERSIDADE TÉCNICA DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

Decentralized Low Communication 6DoF Full State Formation Navigation

Sónia Maria Martinho Marques

(Mestre)

Dissertação para obtenção do Grau de Doutoramento em

Engenharia Electrotécnica e de Computadores

Orientador: Doutor Pedro Manuel Urbano de Almeida Lima

Júri

Presidente:	Reitor da Universidade Técnica de Lisboa
Vogais:	Doutora Maria Isabel Lobato de Faria Ribeiro
	Doutor Pedro Manuel Urbano de Almeida Lima
	Doutor Bogdan Udrea
	Doutora Anna Guerman
	Doutor Fernando José Parracho Lau
	Doutor Paulo Jorge Coelho Ramalho Oliveira.

Abril de 2009

Resumo

Uma formação de satélites em órbita tornou-se a tecnologia chave para missões espaciais presentes e futuras. Este conceito permite ir muito além dos objectivos científicos alcançados pelas estruturas monolíticas predominates até aos nossos dias. Contudo, este tipo de sistemas multi-satélite não se obtem simplesmente transpondo as técnicas já existentes em satélites únicos. Outros factores têm de ser considerados, tais como a coordenação dos elementos da formação.

Este trabalho foi motivado pela necessidade de desenvolver um método que usa uma arquitectura descentralizada para estimação dos componentes de translação e de rotação (6 graus de liberdade) do vector de estado completo de uma formação de N-satélites. Assim, tem-se a vantagem de eliminar a dependência de um único satélite central, sendo o fluxo de informação apenas entre pares de veículos, enquanto proporciona a estimativa do estado completo da formação em cada veículo. Este trabalho apresenta uma formulação mais geral, que se aplica à estimação do estado total de uma formação de N-veículos. Com esse propósito, a teoria de grafos é usada para modelar o fluxo de medidas e informação entre os veículos da formação, estudar o impacto que a quebra de uma ligação (por exemplo devido a falhas de um ou mais veículos) tem no desempenho do algoritmo, e estabelecer requisitos necessários para a robustez do algoritmo a alterações da topologia do fluxo de informação e de medidas relativas. O método de estimação é baseado no Filtro de Kalman Estendido (EKF), quando são usadas medidas locais obtidas num veículo, e o algoritmo Intersecção de Covariância (CI), quando é usada a informação obtida a partir de outro veículo da formação. Os algoritmos desenvolvidos são aplicados a uma simulação realista de três satélites numa órbita de transferência geoestacionária (GTO), cenário actual de referência para a missão Darwin da Agência Espacial Europeia (ESA). São apresentados resultados de experiências realizadas com o estimador de estado inserido dentro e fora da cadeia de controlo da formação. Os principais resultados demonstram a implementação com sucesso do estimador proposto, a importância de usar estimativas de outros veículos, de modo a obter-se as componentes do vector de estado que não são estimadas localmente, bem como o uso do método de CI para combinar variáveis correlacionadas. Outros resultados relevantes incluem a qualidade alcançada pelos algoritmos de navegação, por comparação o limite inferior de Cramer-Rao (PCRLB), a robustez dos algoritmos a incertezas de paramêtros, a importância de afinar a matrizes de covariância do ruído do processo e das observações usando os testes de consistência, testes da estimativa normalizada do erro quadrático (NEES) e da inovação quadrática normalizada (NIS), e confirmado pela comparação com o PCRLB.

Palavras-chave: Filtro de navegação descentralizada, estimação do vector de estado completo de uma formação de veículos, filtro de Kalman estendido, método de intersecção de covariância, teoria algébrica dos grafos, navegação de uma formação de satélites.

Abstract

Formation Flying Spacecraft have become the key enabling technology for today and future space missions. This concept has enabled scientific objectives to move beyond those accomplished by the restricted single monolithic structures predominant until now. However, design and implementation of this type of multi-spacecraft systems is not simply scalable from that of single spacecraft platforms, as extra factors such as coordination within the formation need to be considered.

This work was motivated by the need to develop a method that uses a decentralized architecture to estimate the 6 Degrees of Freedom (DoF) translational and rotational components of a full-order state vector of a formation of N-spacecraft. This would have the advantage of eliminating the dependence on a single master spacecraft while still enabling the full state to be determined.

The thesis presents a general formulation of the problem of estimating the full state of N-vehicle formation. Graph theory was applied with three main objectives in mind: to model the measurements and the information flow within the formation; to analyse the impact of a connection breakdown (a vehicle failure, for example) on the performance of the navigation algorithm; and to establish requirements for the algorithm to be robust to changes in the topology of the *measurement graph* and the *information flow graph*. The estimation method is based on an Extended Kalman Filter (EKF) for measurements obtained locally by each vehicle's sensor and a Covariance Intersection (CI) algorithm for information communicated by other vehicles in the formation.

The developed algorithms are applied to the realistic simulation of a three-spacecraft system that emulates the European Space Agency (ESA) Darwin mission. Results of experiments with and without the estimator in the Guidance and Control loop are presented. Realistic simulations show that the implementation of the proposed 6DoF navigation filter was successful. The use of information from other vehicle's estimates to compute the non-observable state vector components not estimated locally was shown to be both important and feasible, as was the use of the CI method to fuse correlated entities. Other relevant results include the accuracy of the achieved full state vector estimates and the navigation quality by comparison with the PCRLB (Posterior Cramer-Rao Lower Bound); the capability of the navigation algorithms to overcome the initial error, to overcome environmental disturbances and to track control changes; the navigation's robustness to uncertainty of the parameters; and the importance of tuning the process and measurement noise covariance matrices using the Normalized Estimation Error Squared (NEES) and the Normalized Innovation Squared (NIS) consistency tests, as confirmed by the comparision with the PCRLB.

Keywords: Decentralized navigation, formation state estimation, extended Kalman filter, covariance intersection, algebraic graph theory, formation flying spacecraft.

Preface and Acknowledgments

I greatly acknowledge my thesis to my supervisor Professor Pedro Lima for his "navigation", support, friendship and for believing in my work capacity, during this work. He was a supervisor that lead me in the direction of high quality work. I also thank him for giving me the opportunity to participate in the ESA project, "Formation Estimation Methodologies for Distributed Spacecraft".

I would like to thank my long time friends, who always believed in me, Elisabete do Carmo, Joaquim Gorjão, Barbara Lemos, Sónia Oliveira, Scott Williams, Malcolm Shuster and others that are more recent friends but not less important, Filipa Viola, Isabel Rodrigues, Adriana Schmocker, Marcel Stöckli, and to Jeremy Thorne, who gave me a precious help about the English written on this manuscript.

I would like to express my gratitude to my dear colleagues and friends Carlos Banha, Dejan Milutinovic and Rodrigo Ventura for their friendship and support. Also I would like to thank to my colleagues at EST, Prof. João Catarino, Prof. João Lourenço and also to Professors Gustavo Silva and a special thanks to Professor Dias Pereira that made possible for me to obtain a PRODEP III scholarship, as well as to the president of the ESTEC school, Prof. Doutor João Francisco.

A especial thanks to European Space Agency (ESA), which gave me the opportunity to work in the project "Formation Estimation Methodologies for Distributed Spacecraft", ESA contract N^o 17529 /03/NL/LvH/bj, and to work in innovative research within an international environment, where I had the pleasure to work and learn with Dr. Bodgan Udrea, and the people from DEIMOS SA, Augusto Caramagno, Luis F. Peñin, Andreas Mafficini and João Araújo. Moreover, to the Institute for Systems and Robotics (ISR) and Fundação para a Ciencia e a Tecnologia (ISR/IST plurianual funding) for providing me with excellent work conditions and that financed my presentation in the American Institute of Aeronautics and Astronautics (AIAA) conference in Tampa, EUA. To PRODEP III project 05.03/LVT/1069.012/03 for the scholarship, making this work to become possible. Also, I would like to thank to Augusto Caramagno and Luis F. Peñin, from DEIMOS Space, Luis F. Peñin, João Araújo and Andreas Mafficini from DEIMOS Engenharia Lda, for useful insights on formation flying mission analysis and specifications, as well as for making available DEIMOS' FF-FES simulator.

Last but not least, I would like to thank my family, for their continued support, to my special friend Elisabeth do Carmo who is like family to me and to Tonia Ahanotu who makes me feel appreciated for who I am. Also to Teresa Salazar Neto who told me that I have an option of life and to my dearest Michael Schmocker, who always gave me the calm feeling needed to still enjoy life.

Contents

1 INTRODUCTION **1**

1.1 Motivation 1

1.1.1 Proposed approach for a vehicle 3

1.1.2 Centralized *vs* decentralized architectures 4

1.1.3 Decentralized architectures and communication network 8

1.2 Goals and Original Contributions 9

1.3 Thesis Outline 10

2 LITERATURE SURVEY **13**

2.1 Introduction 13

2.2 Formation Flying Spacecraft State Estimation 13

2.3 Robot Formations State Estimation 15

2.4 Types of Formation Translational State Estimators 17

2.5 Attitude Estimation in Single Vehicles 19

2.6 Range Sensors 19

3 PROBLEM FORMULATION **21**

3.1 Introduction 21

3.2 *Formation Estimation* Graph 23

3.3 *Measurement* Graph 24

3.3.1 Sensor failure scenario 26

3.3.2 Observability 28

3.4 *Information Flow* Graph 30

3.4.1 Peer-to-peer topology 31

3.5 Formation Navigation Problem 32

3.6 Summary 32

4 SIX DEGREES OF FREEDOM DECENTRALIZED FORMATION STATE ESTIMATION 33

4.1 Preliminaries . . . 35
4.1.1 Six degrees of freedom formation state vector . . . 35
4.1.2 Formation dynamics . . . 40
4.2 Formation Navigation Analysis . . . 42
4.2.1 *Formation estimation* graph . . . 43
4.2.2 *Measurement* graph . . . 43
4.2.3 *Information flow* graph . . . 48
4.3 Formation Translational Navigation Design . . . 51
4.3.1 *Sensor failure* scenario . . . 52
4.3.2 Algorithm . . . 54
4.3.3 Example of three formation flying spacecraft . . . 56
4.4 Recursive Nonlinear Estimation . . . 57
4.4.1 Extended Kalman filter . . . 58
4.5 Fusion of Multiple Vehicle State Estimates . . . 61
4.5.1 Correlation between formation state estimates of different vehicles . . . 62
4.5.2 Covariance intersection . . . 66
4.6 Translational State Vector Estimation Algorithm . . . 70
4.6.1 Relative position and translational velocity . . . 70
4.7 Rotational State Vector Estimation Algorithm . . . 72
4.7.1 Absolute attitude and angular velocity . . . 73
4.8 Summary . . . 77

5 EXTENDED KALMAN FILTER ANALYSIS ISSUES 79

5.1 Introduction . . . 79
5.2 Stability . . . 79
5.2.1 Riccati difference equation . . . 80
5.2.2 Linear time-invariant example . . . 81
5.2.3 Observability of nonlinear systems . . . 84
5.3 Posterior Cramer-Rao Lower Bound . . . 88
5.4 Filter Consistency . . . 91
5.4.1 Normalized estimation error squared test . . . 91
5.4.2 Normalized innovation squared test . . . 93
5.5 Summary . . . 94

6 SIMULATION SETUP **95**

6.1 Formation Flying Functional Engineering Simulator . . . 97

6.2 Geostationary Transfer Orbit Mission . . . 97

6.2.1 Modes definition . . . 99

6.2.2 Requirements . . . 101

6.2.3 Orbital parameters simulation setup . . . 101

6.3 Six Degrees of Freedom Integrated Navigation Implementation . . . 103

6.3.1 Coordinate systems and rotations matrix . . . 104

6.3.2 Translational state vector . . . 110

6.3.3 Rotational state vector . . . 112

6.3.4 Translational linearized equations of motion . . . 112

6.3.5 Rotational equations of motion . . . 115

6.3.6 Implementation . . . 115

6.3.7 Gramian observability . . . 130

6.3.8 Filtering and prediction . . . 131

6.3.9 Modularity . . . 131

6.4 Tests Description . . . 132

6.4.1 Guidance and control . . . 136

6.4.2 Experiments description . . . 137

6.5 Error Analysis . . . 141

6.5.1 Relative position and translational velocity error . . . 142

6.5.2 Angular velocity error . . . 142

6.5.3 Quaternion error . . . 142

6.5.4 Euler angles error . . . 142

6.5.5 Standard deviation . . . 143

6.5.6 Root mean square error . . . 144

6.5.7 Monte-Carlo . . . 146

6.5.8 Normalized estimation error squared test . . . 147

6.5.9 Normalized innovation squared test . . . 149

6.6 Summary . . . 149

7 RESULTS **151**

7.1 Introduction . . . 151

7.2 System Conditions . . . 152

7.2.1 Initial conditions . . . 152

7.2.2 Process noise covariance matrix . . . 153
7.2.3 Measurement noise covariance matrix . . . 154
7.2.4 Sensor measurements analysis . . . 156
7.3 Navigation Algorithms Conditions . . . 159
7.3.1 EKF initial conditions . . . 159
7.3.2 Tuning the process noise covariance matrix . . . 162
7.3.3 Tuning the measurement noise covariance matrix . . . 163
7.4 Results for Formation Acquisition Mode . . . 165
7.4.1 Experiment 1 - local extended Kalman filter only with nominal measurement noise 165
7.4.2 Experiment 2 - local extended Kalman filter only with tuned measurement and process noises . . . 167
7.4.3 Experiment 3 - local extended Kalman filter and fusion with covariance intersection 170
7.4.4 Experiment 4 - local extended Kalman filter and fusion with covariance intersection and simulator with perturbations . . . 177
7.4.5 Experiment 5 - implementation of guidance and control . . . 183
7.4.6 Experiment 6 - implementation of guidance, navigation and control . . . 188
7.5 Results for Baseline Control Mode . . . 188
7.5.1 Experiment 7 - guidance implementation . . . 189
7.5.2 Experiment 8 - integrated guidance and navigation as observer . . . 189
7.5.3 Experiment 9 - Monte-Carlo results for guidance and control with navigation as observer . . . 196
7.6 Summary . . . 196

8 CONCLUSIONS AND FUTURE WORK 199
8.1 Future Work . . . 201

A GRAPHS 211
A.1 Graph Theory . . . 211
A.2 Algebraic Graph Theory . . . 215

B EXTENDED KALMAN FILTER 221
B.1 Continuous-Discrete Extended Kalman Filter Equations . . . 221
B.2 Algorithm for Relative Position and Translational Velocity Estimation . . . 225
B.2.1 Sensor observation . . . 225
B.2.2 Fusion of state vectors estimates . . . 226

B.2.3 Algorithm in pseudo-code form . . . 227
B.3 Algorithm for Attitude and Angular Velocity Estimation . . . 228
B.3.1 Sensor observation . . . 229
B.3.2 Fusion of state vectors estimates . . . 230
B.3.3 Algorithm in pseudo-code form . . . 231

C RELATIVE DYNAMICS **235**
C.1 Elliptic Orbit Parameterization . . . 235
C.2 Two-Body Problem . . . 238
C.3 Relative Motion . . . 240
C.3.1 Geometric equation . . . 240
C.3.2 Clohessy-Wiltshire (Hill's) equations, time dependent . . . 242
C.3.3 Clohessy-Wiltshire (Hill's) equations, true anomaly dependent . . . 244

Glossary

DARWIN A ESA space-based interferometer that will operate in the infrared part of the spectrum. Its principal aims will be to search for Earth-like planets around nearby stars and then look for signatures of life, such as the presence of ozone in the atmosphere.

Deep Space originally it was used to refer to any region of space beyond the Moon's orbit, but nowadays it is used to refer to any region beyond the orbit of Mars. In this work, it means a region that is outside the field of the NAVSTAR constellation.

Formation center is the formation reference orbit frame, or formation center, which from all spacecraft in the formation position themselves to. It is described by its orbital elements and not necessarly placed in the center of the formation. In this work it is identified by the Local Vertical Local Horizontal (LVLH) frame.

Geostationary Orbit a orbit with a period of one sideral day (1436.6 minutes) with zero inclination and eccentricity, *i.e.,* an equatorial and circular orbit.

Geostationary Transfer Orbit (GTO) An elliptical orbit, with an apogee (high point) of 35,784 km, a perigee (low point) of a few hundred km, and an inclination roughly equal to the latitude of the launch site, into which a spacecraft is initially placed before being transferred to a geosynchronous or geostationary orbit.

Guidance, Navigation and Control

Guidance The process of directing the movements of an aeronautical or space vehicle, with particular reference to the selection of a flight path or trajectory;

Navigation the process of estimating the formation state;

Control the process of tracking the planned path, rejecting perturbations in given frequency range(s).

Interferometer An instrument, consisting of linked telescopes, which collects the electromagnetic radiation arriving from an astronomical object along two or more different paths, and combines the results to form an interference pattern. The resulting resolution is much higher than that achievable by the component telescopes working independently.

J2 disturbance includes the point mass effect as well as the dominant effect of the asymmetry in the gravitational field, representing the North/South hemisphere Earth oblateness.

Michelson Interferometer The Michelson interferometer is an optical instrument that combines two separate light beams by splitting a beam of monochromatic light so that one beam strikes a fixed mirror and the other a moving mirror. When the reflected beams are brought back together, an interference pattern results.

NAVSTAR A global positioning system (GPS), often referred to (particularly in the U.S.) as the GPS or NAVSTAR-GPS, which is based on a global system of 24 US Department of Defense navigational satellites (21 operational plus 3 spares), completed in 1993 and designed to provide time, position, and velocity data for ships, planes, and land-based vehicles, and for many other purposes. The NAVSTAR satellites are arranged in six planes, each in a 12-hour, 20,000-km-high orbit. They transmit signals that allow the determination, to great accuracy, of the locations of special receivers. These receivers can be fixed on Earth, or in moving vehicles, aircraft, or in satellites in low Earth orbit.

Orbital parameters standard orbital parameters or Kepler parameters, semi-major axis, eccentricity, inclination, right ascension of the ascending node, argument of perigee, true anomaly, (a_{smx}, e_{cc}, i_{cl}, R_{aan}, ϖ, θ_t) parameterize an eccentric orbit.

Science Experiment is when the instruments or devices onboard the spacecraft are taking pictures or collecting data for later analysis, *i.e.*, the science mission is being performed, which usually occurs only during certain periods of the orbit.

Secular Drift is a gradual drift, taking place over a long period. Secular change is a continuous, nonperiodic change in one of the attributes of the states of a system. Secular perturbation is a slow, continuous change in one of the elements of an orbit. Secular stability is a condition in which the equilibrium configuration of a system is stable over a long period of time. A secular variable is a star whose brightness changes over centuries.

Notation

Vectors and Matrices

Variables are denoted by upper and lower case letters in italic type, as for example, ω for quaternion.

Vectors are defined as column vectors denoted by lower case letters in boldface type, as the vector $\mathbf{x}$ with dimension $n_x \times 1$:

$$\mathbf{x} = \begin{bmatrix} x_1 & \cdots & x_{n_x} \end{bmatrix}^T$$

or

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_{n_x} \end{bmatrix}$$

Throughout this work the state vector will be defined as $\mathbf{x}$ and we will use the subscript ij in the state vector, $\mathbf{x}_{ij}$, to denote the vehicle's index and not to define the position of state vector components in the state vector.

In case the vector stands for physical quantities, the vector components are x, y, z, instead 1, 2, 3, as for example the angular velocity, $\boldsymbol{\omega} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$. When two vectors, $\mathbf{b} = \begin{bmatrix} b_1 & \cdots & b_{n_b} \end{bmatrix}^T$ and $\mathbf{c} = \begin{bmatrix} c_1 & \cdots & c_{n_c} \end{bmatrix}^T$ are embedded components of the vector $\mathbf{a}$, then vector $\mathbf{a}$ is written as follows,

$$\mathbf{a} = \begin{bmatrix} \mathbf{b}^T & \mathbf{c}^T \end{bmatrix}^T$$

or

$$\mathbf{a} = \begin{bmatrix} b_1 & \cdots & b_{n_b} & c_1 & \cdots & c_{n_c} \end{bmatrix}^T$$

Matrices are denoted by upper case letters in boldface type, as the matrix $\mathbf{C}$ made up of elements $[c_{n,m}]$, of dimension $(n_c \times m_c)$, as follows,

$$\mathbf{C} = [c_{n,m}] = \begin{bmatrix} c_{1,1} & \cdots & c_{1,m_c} \\ \vdots & \ddots & \vdots \\ c_{n_c,1} & \cdots & c_{n_c m_c} \end{bmatrix}$$

Also the columns of the matrix can denoted by lower case letters as vectors where in subscript is indicated the x, y, or z corresponding to the first, second or third columns, as follows, $\mathbf{C} = \begin{bmatrix} \vdots & \vdots & \vdots \\ \mathbf{c}_x & \mathbf{c}_y & \mathbf{c}_z \\ \vdots & \vdots & \vdots \end{bmatrix}$.

A *graph* is denoted by the symbols, Ð, ℵ, ¥, ₣, and ℘.

List of Symbols

a_{smx}	Orbit's semi-major axis.	p. 235
$\mathbf{A}$	Dynamic matrix of time-invariant system.	p. 29
$\mathbf{A}(q)$	Atitude matrix (3×3) obtained from quaternions, q.	p. 38
$\mathbf{A}_{LVLH}^{IPQ}$	Rotation matrix (3×3) obtained from orbital parameters instead quaternions, q.	p. 108
$\mathbf{A}(q_{b_j}^{b_i}) = \mathbf{A}_{b_j}^{b_i}(q)$	Atitude matrix which describes the rotation between body frame b_i and body frame b_j.	p. 38
$\mathbf{A}(k)$	Dynamic matrix of time-variant system.	p. 86
b_i	Body frame of vehicle i.	p. 106
$\overrightarrow{\mathbf{b}} = (\overrightarrow{\mathbf{b}}_x,\ \overrightarrow{\mathbf{b}}_y,\ \overrightarrow{\mathbf{b}}_z)$	Versors of the body frame.	p. 106
$\mathbf{B}(\theta_t)$	Continuous-time input gain and noise gain matrix $(n_x \times n_u)$.	p. 124
$\mathbf{B}_g(Đ)$	Incidence matrix $(N \times N_e)$, of directed graph Đ, definition 47.	p. 215
c	The speed of light $(3 \times 10^8\ [m/s])$.	p. 118
$\deg_{in}$,$\deg_{out}$	Indegree (outdegree) is the number of edges that come in (out). a vertice of a graph repectively, definition 48.	p. 215
$\mathbf{C}$	Measurement matrix $(n_y \times n_u)$.	p. 85
$\mathbf{D}_g(Đ)$	Degree matrix $(N \times N)$ for matrix Đ, defintion 50.	p. 216
Đ	Directed graph or digraph, definition 37.	p. 213
$\overrightarrow{\mathbf{e}} = (\overrightarrow{\mathbf{e}}_x,\ \overrightarrow{\mathbf{e}}_y,\ \overrightarrow{\mathbf{e}}_z)$	Versors defined as a unit vectors indicating the orientation of the LVLH frame.	p. 105
$e(ℵ)$	Edge of the graph ℵ, definition 5.	p. 211
e_{cc}	Eccentricity of the orbit.	p. 235
$eig\{\mathbf{L}(Đ)\}$	Laplacian eigenvalues of the graph Đ.	p. 216
$E_{multipath\rho}^{i,j}$	Multipath error for RF measurement in case of a pseudo-range signal.	p. 117
$E(ℵ)$	Edge set of the graph ℵ, defintion 5.	p. 211
$E\{\cdot\}$	Expected value, mean, average or first moment.	p. 41
$f(\cdot)$	Continuous time process model.	p. 40
f_c	Formation center.	p. 105
$\mathbf{F}(t)$	Jacobian matrix of $f(t)$, $\mathbf{F}(t) \triangleq \left[\nabla f(\mathbf{x},t)^T\right]^T\Big\vert_{\mathbf{x}=\hat{\mathbf{x}}(t)}$.	p. 60
G_{rt}	Universal gravitation constant.	p. 101
$h(\cdot)$	Observation model.	p. 41
$\mathbf{H}$	Observation matrix.	p. 42

$\mathbf{H}(k)$	Jacobian matrix of $h(k)$, $\mathbf{H}(k) = \left[\nabla h(\mathbf{x},k)^T\right]^T\Big\vert_{\mathbf{x}=\hat{\mathbf{x}}(k)}$	p. 58
i_{cl}	Inclination of the orbit	p. 236
$\overrightarrow{\mathbf{I}}$, $\overrightarrow{\mathbf{J}}$, $\overrightarrow{\mathbf{K}}$	Versors of the Inertial Planet Frame (IPQ)	p. 104
$\mathbf{I}$	Moments of inertia, $\mathbf{I} = \begin{bmatrix} I_x & I_y & I_z \end{bmatrix}^T$	p. 115
$\mathbf{J}$	Fisher Information matrix (FIM)	p. 88
k	Time step.	p. 49
$\overrightarrow{\mathbf{k}} = (\overrightarrow{\mathbf{k}}_x,\ \overrightarrow{\mathbf{k}}_y,\ \overrightarrow{\mathbf{k}}_z)$	Unit vectors (versors) indicating the orientation of the Tillerson's frame.	p. 107
$\mathbf{K}$	Kalman Gain.	p. 50
$\mathbf{L}(\aleph)$	Laplacian ($N \times N$) of graph $\aleph$, definition 50.	p. 216
$\mathbf{m}_{ext}$	External moments vector applied to the vehicle.	p. 115
M	Number of Monte-Carlo simulations.	p. 136
$n \times m$	Matrix dimension, n is the number of rows and m is the number of columns.	p. 62
n_t	Natural frequency of the reference orbit.	p. 101
n_x	Dimension of vector $\mathbf{x}$, $n_x \times 1$, or state dimension.	p. 41
n_y	Dimension of vector $\mathbf{y}$, $n_y \times 1$, or measurements dimension.	p. 41
n_u	Dimension of vector $\mathbf{u}$, $n_u \times 1$, or input control dimension.	p. 40
N	Number of vehicles in a formation, or number of vertices of a graph.	p. 2
N_e	Size of a graph, or number of edges in a graph, definition 6.	p. 211
$N_v(\aleph)$	Adjacent vertices of a graph $\aleph$, definition 12.	p. 212
$_o$	Subscript o stands for observable.	p. 49
$_{\bar{o}}$	Subscript $\bar{o}$ stands for non-observable.	p. 49
$p(\mathbf{y},\mathbf{x})$	Joint probability density function of $(\mathbf{y},\mathbf{x})$.	p. 88
$\mathbf{P}(0\vert 0)$	Initial error covariance matrix.	p. 57
$\mathbf{P}$	Error covariance matrix.	p. 50
$[\mathbf{p}\times]$	Skew symmetric matrix.	p. 38
q_j^i	Quaternion which represents the rotation between body frame b_i. and body frame b_j.	p. 38
$\mathbf{q}$	Vector part of the quaternion (3×1).	p. 37
q	Quaternion (4×1).	p. 37
$\mathbf{Q}(k)$	Process noise obtained from the power spectrum of the continuous noise.	p. 223
$\breve{\mathbf{Q}}(t)$	Process noise covariance matrix	p. 41
$\mathbf{R}$	Measurement noise covariance matrix.	p. 58
R_{aan}	Right Ascencion of the Ascending Node (RAAN).	p. 236

R_1, R_2, R_3	Antenna number 1, antenna number 2 and antenna number 3.	p. 118
$\mathbf{S}(k)$	Innovation covariance matrix as a funtion of the discrete time k.	p. 59
$\mathbf{T}_g(Ð)$	Adjacent matrix $(N \times N)$, associated to the graph Ð, definition 46.	p. 215
T^i, T^j	Time bias of the receiving s/c i and transmitting s/c j clock, respectively for RF measurements in case of a pseudo-range signal.	p. 118
T_s	Sampling period.	p. 60
^{T}	Superscript T stands for transpose.	p. 36
t	Continuous time.	p. 40
t_p	Passage time at perigee.	p. 112
v_r	Source or root vertex of a directed graph Ð, definition 41.	p. 214
v_s	Sink vertex of a directed graph Ð, definition 42.	p. 214
$\mathbf{v}(k)$	Observation noise vector of a linear discrete-time output equation.	p. 81
$\mathbf{u}\,(t)$	Control input vector.	p. 40
$\overrightarrow{\mathbf{u}} = (\overrightarrow{\mathbf{u}}_1, \overrightarrow{\mathbf{u}}_2, \overrightarrow{\mathbf{u}}_3)$	Versor of a rotation frame.	p. 106
^{U}	Superscript U stands for underlying graph, definition 40.	p. 214
$V(ℵ)$	Vertex set of graph ℵ, defintion 5.	p. 211
$v_i(ℵ)$	Vertex i of a graph ℵ, or vertices of the a formation, with $i = 1, 2, .., N$ vehicle, definition 5.	p. 211
w_c	Weight variable from Covariance Intersection (CI) method.	p. 49
$\breve{\mathbf{w}}\,(t)$	Process noise in continuous time.	p. 40
$\mathbf{w}\,(k)$	Process noise in discrete time.	p. 85
$\mathbf{w}_i$	Perturbations affecting the i^{th} vehicle of a formation.	p. 114
$\mathbf{x}(t)$	Full-order state vector in continuous time of a N-vehicle formation.	p. 35
$\hat{\mathbf{x}}$	$\hat{\mathbf{x}}$ is the estimate of vector $\mathbf{x}$, definition 1.	p. 62
$\widetilde{\mathbf{x}}$	$\widetilde{\mathbf{x}}$ is the estimation error of vector $\mathbf{x}$, $\widetilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$.	p. 62
$\mathbf{x}_{trans}$	Translational relative part of the state vectorbof the state vector. The translational relative vector has relative position, $\boldsymbol{\chi}_{ij}$ and translational velocity, $\dot{\boldsymbol{\chi}}_{ij}$, components. Notice that the components sub-index ij stands for relative relation between vehicle i and vehicle j and not the position of the vector components in the vector.	p. 36
$\mathbf{x}_{rot}$	Rotational absolute part of the state vector. The rotational vector have attitude, q, and angular velocity, $\boldsymbol{\omega}$, components.	p. 37
$\mathbf{x}_{ij}$	Full-order relative state vector. Notice that the sub-index ij stands for relative relation between vehicle i and vehicle j and not the position of the vector components in the vector. Definition 1.	p. 23

$\mathbf{x}(0)$	Initial state vector.	p. 57
x, y, z	Axis of a Cartesian coordinate system.	p. 36
x^{IPQ}, y^{IPQ}, z^{IPQ}	Axis of the Inertial Planet Frame (IPQ).	p. 104
x^{LVLH}, y^{LVLH}, z^{LVLH}	Axis of the Local Vertical Local Horizon (LVLH) frame.	p. 105
$\mathbf{y}^i(k)$	Measurement (from sensors at vehicle i) vector in discrete time.	p. 41
$\mathbf{z}(k)$	Observation vector in discrete time.	p. 63
σ	Standard deviation.	p. 62
σ_R	Standard deviations of the measurement noise.	p. 82
σ_Q	Standard deviations of the process noise.	p. 82
α	Rotation angle.	p. 106
ϵ_x	Chi-square distribution with state dimension n_x.	p. 91
$\epsilon_{\boldsymbol{\nu}}$	Chi-square distribution with measurements dimension n_y.	p. 93
$\delta(t-\tau)$	Kronecker delta function.	p. 41
$\delta\hat{\mathbf{x}}$	State estimate perturbation.	p. 58
$\delta\mathbf{x}$	Small state perturbation.	p. 121
$\delta\hat{\mathbf{x}}_q \equiv \mathbf{K}\delta\hat{q}$	Quaternion estimate perturbation (4×1).	p. 76
$\delta\hat{\mathbf{x}}_{\mathbf{q}} \equiv \mathbf{K}\delta\hat{\mathbf{q}}$	Quaternion vector part estimate perturbation (3×1).	p. 74
$\delta\hat{\mathbf{x}}_\omega \equiv \mathbf{K}\delta\hat{\boldsymbol{\omega}}$	Angular velocity estimate perturbation vector (3×1).	p. 74
$\delta\hat{\mathbf{q}}$	Quaternion measurement residual or innovation.	p. 74
$\delta\hat{q}$	Quaternion vector part measurement residual.	p. 74
δq	Small quaternion perturbation.	p. 74
$\delta\mathbf{q}$	Small perturbation of the quaternion vector part.	p. 74
$\delta\boldsymbol{\omega}$	Small angular velocity perturbation.	p. 128
$\delta\mathbf{y}$	Small measurement perturbation.	p. 223
θ_t	True anomaly.	p. 236
ρ_{ij}	Distance (scalar), which is the relative distance between vehicle i and j, $\rho_{ij} = \left\|\boldsymbol{\chi}_i - \boldsymbol{\chi}_j\right\|$, $i = 1, ...N-1, j = 2, ..., N, j > i$.	p. 36
ρ_j^i	Relative distance between vehicle i (receiver) and j (transmitter), $\rho_j^i = \rho_{ji} + noise$.	p. 117
ς	Correlation coefficient.	p. 62
$\boldsymbol{\chi}$	Position vector.	p. 36
$\boldsymbol{\chi}_{ij}$	Relative position vector between vehicles i and j. This is the vector with coordinates of the vehicle to which the distance (scalar) is measured.	p. 36
$\dot{\boldsymbol{\chi}}$	Translational velocity vector.	p. 36

$\dot{\boldsymbol{\chi}}_{ij}$	Relative translational velocity vector between vehicles i and j.	p. 36
$\boldsymbol{\omega}$	Angular velocity vector.	p. 37
$\boldsymbol{\iota}_i$	External forces vector acting on the vehicle i, which are the sum of the control inputs $\mathbf{u}_i$ and perturbations $\mathbf{w}_i$.	p. 114
$\boldsymbol{\nu}(k)$	Observation noise vector of a nonlinear discrete-time output equation.	p. 41
ϖ	Argument of perigee.	p. 236
ψ, θ, ϕ	Euler angles, roll , pitch and yaw.	p. 142
$\boldsymbol{\Phi}$	State transition matrix.	p. 60
$\boldsymbol{\Xi}(q)$	Right operator, matrix representation of quaternion product.	p. 38
μ_t	Gravitational parameter, $\mu_t = G_{rt}(M_{Earth} + m_{s/c})$.	p. 101
$\boldsymbol{\Psi}$	Observability Gramian.	p. 86
$\boldsymbol{\Lambda}(q)$	Left operator, matrix representation of quaternion product.	p. 38
$\mathbb{F}$	*Formation estimation* graph, definition 1.	p. 23
$¥$	*Measurement* graph, definition 2.	p. 24
$¥^i$	*Measurement* subgraph, $¥^i$ is a subgraph of $¥, ¥^i \subseteq ¥$, definition 3.	p. 25
$\wp$	*Information flow* graph, definition 4.	p. 30
$\|\cdot\|$	Module.	p. 36
$\aleph$	Generic graph, definition 5.	p. 211
$\aleph'(\aleph)$	Spanning graph of $\aleph$, definition 30.	p. 213
$\otimes$	Operation denoting the quaternion multiplication.	p. 38
$\in$	Is an element of.	p. 40
$\mathbb{R}$	Real numbers.	p. 40
$\cap$	Intersection of graphs, definition 27.	p. 213
$\cup$	Union of graphs, definition 27.	p. 213
$\mathbb{N}$	Natural numbers.	p. 49
var	Variance.	p. 62
cov	Covariance.	p. 62
∇_x	Gradient operator.	p. 59
$\times$	Cross product.	p. 38
$\aleph + \beth$	Sum graphs of graphs $\aleph$ and $\beth$, definition 28.	p. 213
$\aleph' \subseteq \aleph$	$\aleph'$ is a subset of $\aleph$, definition 26.	p. 213
$^{-1}$	Superscript -1 stands for Inverse.	p. 50
Δt	Runge-Kutta step size.	p. 60
$' \equiv \frac{d}{d\theta_t}$	Derivatives *w.r.t.* the true anomaly, θ_t.	p. 113
$\mathbf{1}$	Identity matrix.	p. 38

Acronyms and Abbreviations

a priori from the Latin, meaning *from the former* or less literally *before experience*

AFFS Autonomous Formation Flying Sensor

AFRL Air Force Research Laboratory

BCM Baseline Control Mode

C/A Coarse Acquisition

CDGPS Carrier-phase Differential GPS

CI Covariance Intersection

CLAM Cooperative Localization and Mapping

DIF Decentralized Information Filter

DKF Decentralized Kalman Filter

DoF Degree of freedom

DS-3 Deep Space mission 3

EKF Extended Kalman Filter

Eq. Equation

et al. from the Latin *Et alii* meaning *and others*

etc. from the Latin *et cetera* meaning *and the rest*

e.g. from the Latin *exempli gratia* meaning *for example*

EO Earth Observing satellites

ESA European Space Agency

FAC Formation Acquisition Mode

FAM Fine Acquisition Mode

FEMDS Formation Estimation Methodologies for Distributed Spacecraft

FF Formation flying

FF-FES Formation Flying - Functional Engineering Simulator

FIM Fisher Information Matrix

FOF Full Order Filter

glob Global frame

GC Guidance and Control

GEO Geostationary Orbit

GMTI Ground Moving Target Indication

GNC Guidance, Navigation and Control

GPS Global Positioning System

GTO Geostationary Transfer Orbit

ICEKF Iterative Cascade Extended Kalman Filter

ICM Interferometer Construction Mode

i.e. from the Latin *id est* meaning *that is*

ION-F Ionospheric Observation Nanosatellite Formation

IPQ Inertial Planet Frame

KF Kalman Filter

loc Local frame

LEO Low Earth Orbit

LFM Loose Formation Mode

LISA Laser Interferometer Space Antenna

LOS Line of sight

LTI Linear Time-Invariant

LVLH Local Vertical Local Horizontal

MC Monte Carlo

MEMS Micro-ElectroMechanical Systems

MLE Maximum Likehood Estimator

MSE Mean Square Error

N/A Not Applicable

NEES Normalized Estimation Error Squared

NIS Normalized Innovation Squared

ORF Observation Reference Frame

PCRLB Posterior Cramer-Rao Lower Bound

PID A Proportional-Integral-Derivative controller

RMS Root Mean Square

RF Radio Frequency

ROF Reduced Order Filter

rot Rotational part of the state vector

SA Selected Availability

SAR Synthetic Aperture Radar

s/c Spacecraft

SIRA Solar Imaging Radio Array

SKF Schimdt Kalman Filter

SLAM Simultaneous Localization And Mapping

ST Star Tracker

$SO(3)$ Special orthogonal Lie group of order 3

TechSat 21 Technology Satellite of 21st Century

trans Translational part of the state vector

TPF Terrestrial Planet Finder

UKF Unscented Kalman Filter

UT Unscented Transform

vs from the Latin *vertere*, *versum* meaning *against*

w.r.t. meaning with respect to

3D Three-dimensional space

List of Figures

1-1 Centralized architecture showing a) the measurement network and b) the communication network. The arrow direction in a) represents the flow of sensor information, *i.e.,* in a) the master vehicle estimator receives sensor measurements from s/c 1,2,3,..., N, and in b) broadcasts the state estimates of the formation back to each vehicle. 4

1-2 Decentralized architecture with a) the measurement network and b) the communication network. The networks are not centralized but instead, the measurements in a) are collected by each spacecraft and, the communications in b) are processed among spacecraft, the amount of communications being minimal in this particular example. 5

1-3 Distributed or Hierarchical architecture measurement network. The central group estimator 1...M represent the sub-masters of the sub-formation 1..N- vehicle, that collect the measurements to the master vehicle of the overall formation. 6

1-4 Distributed or Hierarchical architecture communication network. The central estimator 1...M represent the sub-masters of the sub-formation 1..N, that communicate information to the master vehicle of the overall formation. 7

3-1 The *formation estimation* graph $\mathbb{F}$, which corresponds to the representation of the components of the full relative formation state vector, which are to be estimated. 24

3-2 Example of a *measurement* graph $\mathbb{Y}$, defined as the relative measurements links between vehicles in a formation. According to the figure, the edges set is $E(\mathbb{Y}) = \{(1,2),(2,1),(2,3),(2,4),(3,2),(3,4),(4,2),(4,3)\}$, and e.g., the edge $(2,3)$ is a directed edge from vehicle 2 to vehicle 3 indicates that vertex v_3 measures its relative distance to vertex v_2. 25

3-3 Measurement graph of 6 vehicles for a sensor failure scenario. 27

3-4 Fully connected *information flow* graph, where all vehicles share information with all the other vehicles in the formation. 30

3-5 *Peer-to-peer* topology $\wp^P$, which is a directed cycle graph, with minimum number of edges connecting all vertices of the graph once. 31

3-6 Rooted spanning tree. 32

4-1 Representation of a relative vector where $\boldsymbol{\chi}_{ij} = \boldsymbol{\chi}_j - \boldsymbol{\chi}_i = -\boldsymbol{\chi}_{ji}$ and its absolute value is $\boldsymbol{\rho}_{ij} = \|\boldsymbol{\chi}_i - \boldsymbol{\chi}_j\| = \|\boldsymbol{\chi}_j - \boldsymbol{\chi}_i\|$. 36

4-2 Translational variable relationship between relative state vector and absolute state vector. 39

4-3 Intersection of a) the *formation estimation* graph $\mathbb{F}$, and b) the underlying *measurement* subgraph in vehicle i, $\left(\mathbb{Y}^i\right)^U$, results in c) the observable formation estimated components of the *formation estimation* graph on vehicle i. 44

4-4 For a formation of 4 vehicles, the *measurement* graph $\mathbb{Y}$ is decomposed into the four subsets $\mathbb{Y}^i = \left(V, E^i\right)$, $\mathbb{Y} = \mathbb{Y}^1 \cup \mathbb{Y}^2 \cup \mathbb{Y}^3 \cup \mathbb{Y}^4$, where a) $\mathbb{Y}^1$ b) $\mathbb{Y}^2$c) $\mathbb{Y}^3$ d) $\mathbb{Y}^4$, each 3-regular. 45

4-5 The *measurement* subgraphs needed to determine the full state vector in vehicle 1, for a formation of $N = 4$ vehicles, with $i \in \{1, ..., N\}$. The union of the *measurement* subgraphs a)$\mathbb{Y}^i, i = 1$, b) $\mathbb{Y}^{i+1}, i+1 = 2$, and c) $\mathbb{Y}^{i+2}$, $i+2 = 3$ in vehicle filter 1 is d) $\mathbb{Y}^i \cup \mathbb{Y}^{i+1} \cup \mathbb{Y}^{i+2}$. Notice, that the union of *measurement* subgraphs is coincident with the *formation estimation* graph $\mathbb{F}$. 46

4-6 The graphs envolved in the formation navigation problem: a) *peer-to-peer* graph b) i^{th} *sensed measurement* graph, $\mathbb{Y}^i$ c) underlying *formation estimation* graph. 52

4-7 Decentralized filter, conjugating the CI algorithm with Kalman filter. 53

4-8 a) *sensed measurement* graph, $\mathbb{Y}^1$, $\deg_{in}(v_1) = 4$, dominated vertex, with maximum number of received measurements opposite to b) *sensed measurement* graph $\mathbb{Y}^3$, $\deg_{in}(v_3) = 0$, undominated vertex, with zero number of received measurements. 54

4-9 Peer-to-peer *Information flow* graph $\wp^P$ and *measurement* graph $\mathbb{Y}$ for a 3 s/c formation flying example. 56

4-10 *formation estimation* graph for the 3 s/c formation flying example. 57

4-11 The error ellipse (for a given threshold) of the cross-covariance $\mathbf{P}_{xy}$ (dashed) for different cross-correlations of the linear combination of two randoms variables, x and y, always lies within the intersection of the error ellipses (for a given threshold) of its covariances $\mathbf{P}_{xx}$ (solid) and $\mathbf{P}_{yy}$ (doted) respectively, for a Maximum likehood estimator. Reprinted from [49]. 71

4-12 Block diagram of the EKF algorithm with CI used in the state vector estimation of relative position and relative translational velocity. 72

4-13 Block diagram of the EKF algorithm with EKF Fusion used in the state vector estimation of inertial attitude and angular velocity. 77

5-1 The plot shows the results (RMS) obtained by the KF for a scalar system. The top plot shows the evolution of RMS obtained by the KF when the process error is constant, $\sigma_Q^2 = (10)^2$, and the variance of the measurements ranges through the values presented in the table. The bottom figure shows the evolution of RMS obtained from KF when the error in the measurements is constant, $\sigma_R^2 = (10)^2$, and the variance of the process noise ranges through the values in the table 5.1. The mean RMS values are computed according to equation 5.12. 84

5-2 The plots show the results (error variance) obtained by the KF for a scalar system. The top plot shows the evolution of error variance obtained by the KF when the process error is constant, $\sigma_Q^2 = (10)^2$, and the variance of the measurements ranges through the values presented in the table. The bottom figure shows the evolution of error variance obtained from KF when the error in the measurements is constant, $\sigma_R^2 = (10)^2$, and the variance of the process noise ranges through the values in the table 5.1. 85

5-3 **Error variance analysis,** scalar case where the interval $[-0.05, 0.05]$ represents the region indicated by the chi-square test for two-sided 95% probability region. a) the filter is consistent, the error covariance matrix is commensurate with the error. The bias is greatly reduced or zero. b) the filter is inconsistent, its precision is high but its accuracy is very low, corresponding to a NEES plot above the upper limit of the 95% probability region of the chi-square distribution. In practice the error covariance is too small for the corresponding state vector error. c) the filter is pessismistic, the precision is low due for example to bias. The NEES curve is below the upper limit of the 95% probability region of the chi-square distribution. In practice the error covariance is too big compared to a small state vector error. 92

6-1 Block diagram of the main simulator blocks. 98

6-2 Block diagram of the main subsystems inside each s/c in the simulator. 98

6-3 GNC block diagram inside each s/c in the simulator. 99

6-4 Proposed mode sequence. Reprinted from [68]. 100

6-5 Considered modes upon the baseline GTO (not at scale). Reprinted from [68]. 101

6-6 Inertial Planet Frame (IPQ) is the reference inertial coordinate system also known as Earth Centered Inertial (ECI) frame. 107

6-7 Angle/axis rotation about z axis. x, y, z are the original axis and X, Y, Z are the rotating x, y, z axis around z axis. 108

6-8 LVLH frame, view from above the orbital plane. 109

6-9 Body frame versors of a spacecraft. 110

6-10 Generic position of two spacecraft and the position representation of each spacecraft *w.r.t.* the LVLH frame. The relative distance between spacecraft is also represented. 113
6-11 Representation of a relative vector where $\boldsymbol{\chi}_{ij} = \boldsymbol{\chi}_j - \boldsymbol{\chi}_i = -\boldsymbol{\chi}_{ji}$ and its absolute value is $\boldsymbol{\rho}_{ij} = \|\boldsymbol{\chi}_i - \boldsymbol{\chi}_j\| = \|\boldsymbol{\chi}_j - \boldsymbol{\chi}_i\|$. 114
6-12 Diagramatic view of the communication between s/c and between navigation and GC, filtering and prediction times. The estimates of attitude and angular velocity are used online by the relative position and translational velocity navigation algorithm. 116
6-13 Three measurements, ρ_j^{i,R_1}, ρ_j^{i,R_2}, ρ_j^{i,R_3}, transmitted from spacecraft i are received by each antenna, R_1, R_2, R_3, on spacecraft i. 118
6-14 Geometric representation of the measurement received by antenna 3 in the s/c i and transmitted by the s/c j. 119
6-15 Translational and rotational block diagrams inside each Navigation block in the simulator. 132
6-16 View from above the orbital plane of the 3 spacecraft absolute positions *w.r.t.* Earth, in IPQ. Reprinted from [68]. 134
6-17 Representation of the Formation Acquisition mode (FAC) in the GTO orbit. The initial conditions coincide with the deployment of the s/c, so the s/c can be within a sphere of 8km diameter and the final conditions corresponds to building the formation to perform the science experiment, i.e., the initial conditions of the ICM where the s/c should be 25 meter away from each other. 134
6-18 Formation Flying geometry to accomplish Mission Science. 137
6-19 **Normal probability density function** defined by the mean, m, and by the standard deviation, σ. The probability of a random value being inside the $\pm 2\sigma$ bounds (σ is the standard deviation) centered around the mean is 95%. Figure reprinted from [56]. 144

7-1 24 hour trajectory for GTO orbit, where the counting of time starts with true anomaly θ_0 and the perigee occurs at 9 and 21 hour. 156
7-2 **RF measurements real noise at s/c 1**. The real error noise obtained between the receiving antennas in s/c 1 and s/c 2 (top row) and s/c 1 and s/c 3 (bottom row) for RF subsystem. The standard deviation envelope is also plotted and was computed from the measurement noise covariance matrix diagonal elements. 157
7-3 **2 orbits run open-loop.** The evolution of the distances between the s/c 1 and s/c 2 (green solid line), and between s/c 1 and s/c 3 (blue solid line) without any GC applied to the formation. 158

7-4 **2 orbits run open-loop.** Projection in the x-y plane of the 3 s/c relative trajectories in IPQ without GC applied to the formation. s/c 1 (red solid line), s/c 2 (green solid lie) and s/c 3 (blue solid lie). 158

7-5 **2-orbits run with GC.** The evolution of the distances between the s/c 1 and s/c 2 (green solid lie), and between s/c 1 and s/c 3 (blue solid line) without any GC applied to the formation. 159

7-6 **2-orbits run with GC.** Projection in the x-y plane of the 3 s/c relative trajectories in IPQ with the GC applied to the formation. s/c 1 (red solid line), s/c 2 (green solid line) and s/c 3 (blue solid lie). 160

7-7 **ST measurements real noise at s/c 1.** The real error noise obtained for the star tracker measurements in s/c 1, 2 or 3 enclosed by the standard deviation envelope. 161

7-8 **Experiment 1** - The innovation process obtained in local EKF which is computed as the difference between the RF measurements at the receiving antennas in s/c 1 and s/c 2 (top row) or s/c 1 and s/c 3 (bottom row) and a sensor model. Standard deviation envelope obtained from the Innovation matrix ($\mathbf{S} = \mathbf{HPH}^T + \mathbf{R}$) diagonal elements. 167

7-9 **Experiment 1** - Results for the $\mathbf{P}_{EKF}$ (solid line) of relative position obtained in s/c 1 for a 6 hours orbit segment and the PCRLB (dotted line). 168

7-10 **Experiment 1** - NEES-NIS plots for translational state vector. Top plot: NEES test, Bottom plot: NIS test.The plots are computed for all state vector components, $n_x = 12$ corresponding to limits $[5.23; 21]$, and $n_y = 6$ corresponding to limits $[1.64; 12.6]$, on each s/c. 168

7-11 **Experiment 1** - NEES-NIS plots for rotational state vector. Top plot: NEES test, Bottom plot: NIS test.The plots are computed for all state vector components, $n_x = 6$ corresponding to limits $[1.64; 12.6]$, and $n_y = 3$ corresponding to limits $[0.352; 7.81]$, on each s/c. 169

7-12 **Experiment 2**. The innovation process obtained in local EKF enclosed by the standard deviation envelope. The innovation process is computed as the difference between the RF measuments at the receiving antennas in s/c 1 and s/c 2 (top row) or s/c 1 and s/c 3 (bottom row) and a sensor model. 171

7-13 **Experiment 2** - Results for the $\mathbf{P}_{EKF}$ (solid line) of relative position obtained in s/c 1 for a 6 hours orbit segment and the PCRLB (dotted line). 171

7-14 **Experiment 2** - Results for the estimation error of relative position obtained in s/c 1 for a 6 hours orbit segment. The xyz_{12} and xyz_{13} components are updated locally with the RF measurements while the components xyz_{32} are just propagated through the equations of motion. 172
7-15 **Experiment 2** - Results for the RMS of relative position obtained in s/c 1 for a 6 hours orbit segment. The xyz_{12} and xyz_{13} components are updated locally with the RF measurements while the components xyz_{32} are just propagated through the equations of motion. 172
7-16 **Experiment 2** - NEES-NIS plots for trsnalational state vector. Top plot: NEES test, Bottom plot: NIS test.The plots are computed for all state vector components, $n_x = 12$ corresponding to limits $[5.23; 21]$, and $n_y = 6$ corresponding to limits $[1.64; 12.6]$, on each s/c. 173
7-17 **Experiment 2**. The innovation process obtained in local EKF, enclosed by the standard deviation envelope . This is computed as the difference between the ST measuments at the receiving antennas in s/c 1 and s/c 2 (top row) or s/c 1 and s/c 3 (bottom row) and a sensor model. 173
7-18 **Experiment 2** - Results for the estimation error of the Euler angles obtained in s/c 1 for a 6 hours orbit segment. The components of s/c 1 are updated locally with the ST measurements while the components of s/c 2 and 3 are just propagated through the equations of motion. 174
7-19 **Experiment 2** - Results for the estimation error of the angular velocity for s/c 1 for a 6 hours segment. The components of s/c 1 are updated locally with the ST measurements while the components of s/c 2 and 3 are just propagated through the equations of motion. 174
7-20 **Experiment 2** - Results for the RMS of Euler angles obtained in s/c 1 for a 6 hours orbit segment. The ψ(r1), θ(p1), ϕ(y1) for s/c 1 are updated locally with the ST measurements while for s/c 2 and 3, the angular velocity components are just propagated through the equations of motion. 175
7-21 **Experiment 2** - Results for the RMS of angular velocity $[rad/s]$ obtained in s/c 1 for a 6 hours orbit segment. The ω_{xyz} for s/c 1 are updated locally with the ST measurements while for s/c 2 and 3, the angular velocity components are just propagated through the equations of motion. 175

7-22 **Experiment 2** - NEES-NIS plots for rotational state vector. Top plot: NEES test, Bottom plot: NIS test.The plots are computed for all state vector components, $n_x = 6$ corresponding to limits $[1.64; 12.6]$, and $n_y = 3$ corresponding to limits $[0.352; 7.81]$, on each s/c. 176

7-23 **Experiment 3** - Results for the estimation error of relative position obtained in s/c 1 for a 6 hours orbit segment. The components of xyz_{12} and xyz_{13} are updated locally with the RF measurements while the components xyz_{32} are updated due to the CI algorithm. . 177

7-24 **Experiment 3** - Results for the estimation error of relative translational velocity obtained in s/c 1 for a 6 hours orbit segment. The components of xyz_{12} and xyz_{13} are updated locally with the RF measurements while the components xyz_{32} are updated due to the CI algorithm. 178

7-25 **Experiment 3** - Results for the estimation error of relative position enclosed by the standard deviation envelope obtained in s/c 1 for a 6 hours orbit segment. The components of xyz_{12} and xyz_{13} are updated locally with the RF measurements while the components xyz_{32} are updated due to the CI algorithm. 178

7-26 **Experiment 3** - Results for the weighted parameter w for s/c 1. 179

7-27 **Experiment 3** - Results for the RMS of relative position obtained in s/c 1 for a 6 hours orbit segment. The components of xyz_{12} and xyz_{13} are updated locally with the RF measurements while the components xyz_{32} are updated due to the CI algorithm. 179

7-28 **Experiment 3** - Results for the $\mathbf{P}_{EKF}$ (solid line) of relative position obtained in s/c 1 for a 6 hours orbit segment and the PCRLB (dotted line). The components of xyz_{12} and xyz_{13} are updated locally with the RF measurements while the components xyz_{32} are updated due to the CI algorithm. 180

7-29 **Experiment 3** - NEES-NIS plots for the translational state vector. Top plot: NEES test, Bottom plot: NIS test. The plots are computed for all state vector components, $n_x = 18$ of s/c 1 and $n_y = 18$ corresponding to limits $[9.40; 28.9]$, on each s/c. 180

7-30 **Experiment 3** - Results for the estimation of Euler angles obtained in s/c 1 for a 6 hours orbit segment. The components of s/c 1 are updated locally with the ST measurements while the componentsof s/c 2 and 3 are updated due to the CI algorithm. 181

7-31 **Experiment 3** - Results for the RMS of Euler angles obtained in s/c 1 for a 6 hours orbit segment. The ψ(r1), θ(p1), ϕ(y1) for s/c 1 are updated locally with the ST measurements while for s/c 2 and 3, the angular velocity components are updated due to the CI algorithm.181

7-32 **Experiment 3** - Results for the estimation error of the angular velocity for s/c 1 for a 6 hours segment. The components of xyz_{12} and xyz_{13} are updated locally with the RF measurements while the components xyz_{32} are updated due to the CI algorithm. 182
7-33 **Experiment 3** - Results for the RMS of angular velocity $[rad/s]$ obtained in s/c 1 for a 6 hours orbit segment. The components of xyz_{12} and xyz_{13} are updated locally with the RF measurements while the components xyz_{32} are updated due to the CI algorithm. . . . 182
7-34 **Experiment 3** - NEES-NIS plots for the rotational state vector. Top plot: NEES test, Bottom plot: NIS test.The plots are computed for all state vector components, $n_x = 18$ and $n_y = 18$ corresponding to limits $[9.40; 28.9]$, on each s/c. 183
7-35 **Experiment 4** - Results for the estimation error of relative position obtained in s/c 1 for 6 hours orbit segment. The xyz_{12} and xyz_{13}components are updated locally with the RF measurements while the xyz_{32} components are updated due to the CI algorithm. Perturbations are added to the simulator. 184
7-36 **Experiment 4** - Results for the RMS of relative position obtained in s/c 1 for a 6 hours orbit segment. The xyz_{12} and xyz_{13}components are updated locally with the RF measurements while the xyz_{32} components are updated due to the CI algorithm which combines the local state vector estimates with the predecessor state vector estimates. Perturbations are added to the simulator. 184
7-37 **Experiment 4** - Results for the estimation error of Euler angles obtained in s/c 1 for a 6 hours orbit segment. The components of s/c 1 are updated locally with the ST measurements while the components of s/c 2 and 3 are updated with CI algorithm which combines the local state vector estimates with the predecessor state vector estimates. Perturbations are added to the simulator. 185
7-38 **Experiment 4** - Results for the RMS of Euler angles obtained in s/c 1 for a 6 hours orbit segment. The ψ(r1), θ(p1), ϕ(y1) for s/c 1 are updated locally with the ST measurements while the components of s/c 2 and 3 are updated with CI. Perturbations are added to the simulator. 185
7-39 **Experiment 4** - Results for the estimation error of angular velocity for s/c 1 for a 6 hours segment. The components of s/c 1 are updated locally with the ST measurements while the components of s/c 2 and 3 are updated due to the CI algorithm. Perturbations are added to the simulator. 186

7-40 **Experiment 4** - Results for the RMS of angular velocity obtained in s/c 1 for a 6 hours orbit segment. The ω_{xyz} for s/c 1 are updated locally with the ST measurements while for s/c 2 and 3, the angular velocity components are updated due to the CI algorithm. Perturbations are added to the simulator. 186
7-41 **Experiment 5** - The evolution of the distances between the s/c 1 and s/c 2 (green solid lie), and between s/c 1 and s/c 3 (blue solid line). 187
7-42 **Experiment 5** - The evolution of the aperture angle of the triangle formation. 187
7-43 **Experiment 6** - Results for the estimation error of relative position obtained in s/c 1 for GNC 6 hours orbit segment. The xyz_{12} and xyz_{13}components are updated locally with the RF measurements while the xyz_{32} components are updated due to the CI algorithm. Perturbations are added to the simulator. 188
7-44 **Experiment 7 -** Evolution of the s/c position, which remains a triangle, in the local reference frame, for 2 hour BCM orbit. Reprinted from [9]. 190
7-45 **Experiment 7** - The evolution of the distances between the s/c 1 and s/c 2 (green solid line), and between s/c 1 and s/c 3 (blue solid line) for BCM 2 hour orbit. Reprinted from [9]. 190
7-46 **Experiment 7** - The evolution of the aperture angle of the triangle formation in BCM mode. Reprinted from [9]. 191
7-47 **Experiment 7** - View of the BCM 2 hours orbit of the 3 s/c absolute positions *w.r.t.* inertial reference frame. Reprinted from [9]. 191
7-48 **Experiment 7** - Projection in the $z-x$ plane of the 3 s/c relative trajectories in IPQ with the GC applied to the formation. s/c 1 (red solid line) , s/c 2 (green solid line), s/c 3 (blue solid line). Reprinted from [9]. 192
7-49 **Experiment 7** - Projection in the $x-y$ plane of the 3 s/c relative trajectories in IPQ with the GC applied to the formation. s/c 1 (red solid line) , s/c 2 (green solid line), s/c 3 (blue solid line). Reprinted from [9]. 192
7-50 **Experiment 7** - The evolution of the distances between the s/c 1 and the other between the s/c 1 and s/c 2 (green solid line), and between s/c 1 and s/c 3 (blue solid line) for the GC applied to the formation in BCM and the rest of the orbit in LFM. Reprinted from [9].193
7-51 **Experiment 7** - The evolution of the aperture angle of the triangle formation in BCM and in the rest of the orbit in LFM. Reprinted from [9]. 193

7-52 **Experiment 8 -** Results (detail plot) for the estimation error of relative position obtained in s/c 1 for a 2 hours orbit segment. The components of xyz_{12} and xyz_{13} are updated locally with the RF measurements while the components xyz_{32} are updated due to the CI algorithm. Reprinted from [9]. 194
7-53 **Experiment 8 -** Results (detail plot) for the estimation error of relative translational velocity obtained in s/c 1 for a 2 hours orbit segment. The components of xyz_{12} and xyz_{13} are updated locally with the RF measurements while the components of s/c 2 and 3 are just propagated through the equations of motion. Reprinted from [9]. 194
7-54 **Experiment 8** - Results (detail plot) for the estimation of Euler angles obtained in s/c 1 for a 2 hours orbit segment. The components of s/c 1 are updated locally with the ST measurements while the components of s/c 2 and 3 are just propagated through the equations of motion. 195
7-55 **Experiment 8** - Results (detail plot) for the estimation of angular velocity obtained in s/c 1 for a 2 hours orbit segment. The components of s/c 1 are updated locally with the ST measurements while the components of s/c 2 and 3 are just propagated through the equations of motion. 195
7-56 **Experiment 9** - Results for the RMS of Euler angles (left plot) and angular velocity (right plot) obtained in s/c 1 for a 6 hours orbit segment. The ψ(r1), θ(p1), ϕ(y1) and ω_{xyz} for s/c 1 are updated locally with the ST measurements while for s/c 2 and 3, the quaternions and angular velocity components are just propagated through the equations of motion. 196

A-1 Example of a graph. 219

C-1 Geometry representation of the elliptical orbit parameters inclination, i_{cl}, right ascension of the ascending node, R_{aan}, and argument of perigee, ϖ. Reprinted from [86]. 236
C-2 Geometry representation of the elliptical orbit parameters, θ_t, true anomaly, and semi-major axis, a_{smx}, semi-minor axis, b_{smx}, in the orbital plane. Reprinted from [86]. 237
C-3 The ecliptic or plane of the Earth's orbit and the celestial equator cross at two points, the vernal equinox and the autumnal equinox. The celestial equator is tilted 23.5 degrees in relation to the ecliptic. Reprinted from [87]. 237
C-4 The celestial equator represented on Earth. The celestial sphere is divided in the celestial equator. Reprinted from [88]. 238
C-5 Position of the s/c f_c placed in reference frame, *w.r.t.* Earth ant its IPQ frame. 239
C-6 relative motion of a s/c j *w.r.t.* the formation center which describes a reference orbit. . . 239

C-7 Relation between LVLH frame versors $\overrightarrow{\mathbf{e}}_x$, $\overrightarrow{\mathbf{e}}_y$, $\overrightarrow{\mathbf{e}}_z$, and Tillerson's frame versors $\overrightarrow{\mathbf{k}}_x$, $\overrightarrow{\mathbf{k}}_y$, $\overrightarrow{\mathbf{k}}_z$. 247

List of Tables

3.1 The *indegree* of the vertices for the sensor failure scenario of Fig. 3-3. 28

4.1 For a formation of N-vehicle, there are $N-1$ state vector components determined by the local (vehicle filter i) *measurement* subgraph $E(¥^i)$ ($n^o(\boldsymbol{\chi}_o)$ from $E(¥^i)$) and $N-2$ components that are determined by the first external *measurement* subgraph added to the vehicle filter i (number of observable state vector components, $n^o(\boldsymbol{\chi}_o)$ from $E\left(¥^{i+1}\right)$) , $N-3$ components by the second external measurement subgraph added to the vehicle filter i ($n^o(\boldsymbol{\chi}_o)$ from $E\left(¥^{i+2}\right)$), ... etc. 47

4.2 Summary of the *measurement* subgraph analysis, as a function of the number of vehicles in the formation, namely: the number of state vector components that are determined through the local *measurement* subgraph $E\left(¥^i\right)$; the dimension of the state vector represented by the links in its *formation estimation* graph; the number non-observable state vector components, $n^o(\boldsymbol{\chi}_{\bar{o}})$ (subscript $\bar{o}$ stands for non-observable); the number of external *measurement* subgraphs required to determine completely the state vector. 48

5.1 1^{st} case: Range value for the variance of the measurement noise, while the variance of the process noise is constant. 2^{nd} case: Range value for the variance of the process noise, while the variance of the measurement noise is constant. 83

6.1 Specifications for all the modes of the GTO mission: Formation Acquisition Mode (FAC), Loose Formation Mode (LFM), Interferometer Construction Mode (ICM) and Baseline Control Mode (BCM). $1deg = 1^o$, $1' = 1/60th$ of 1^o, $1ȷ = 1/60th$ of $1'$. 102

6.2 Specifications of the orbital parameters defined in the dynamic equations for GTO orbit. . 102

6.3 Specifications of the additonal GTO orbital parameters for FAC mode. 103

6.4 Specifications of the additional GTO orbital parameters for BCM mode. 103

6.5 Specifications of each s/c in the 3-FF Darwin mission. 104

6.6 FAC mode relative position and translational velocity initial conditions for the FF. s/c1≡s/c2≡s/c3. 104
6.7 FAC mode and BCM attitude and angular velocity initial conditions for the FF. s/c1≡s/c2≡s/c3. 105
6.8 BCM mode relative position and translational velocity initial conditions for the FF. s/c1≡s/c2≡s/c3. 105
6.9 RF subsystem features simulated in the FF-FES. 106
6.10 ST system features simulated in the FF-FES. 106
6.11 Formation Acquisition Mode (FAC) specifications. 133
6.12 Baseline Control Mode (BCM) specifications. 135
6.13 Experiments setup with *a)* $\begin{cases} \hat{\chi}(\theta_0) = \chi(\theta_0) - 1[m] \\ \hat{\chi}'(\theta_0) = \chi'(\theta_0) - 1[m/rad] \end{cases}$ *b)* $\begin{cases} \hat{\psi}, \hat{\theta}, \hat{\phi}(t_0) = \psi, \theta, \phi(t_0) - 10[^o] \\ \omega(t_0) = \bar{\omega}(t_0) - 10[\text{arcsec}/s] \end{cases}$ *c)* with nominal measurement noise covariance matrix *d)* with perturbations included in the simulator *e)* $\begin{cases} \hat{\chi}(\theta_0) = \chi(\theta_0) \\ \hat{\chi}'(\theta_0) = \chi'(\theta_0) \end{cases}$ *f)* $\begin{cases} \hat{\psi}, \hat{\theta}, \hat{\phi}(t_0) = \psi, \theta, \phi(t_0) \\ \omega(t_0) = \bar{\omega}(t_0) \end{cases}$. Experiments 1-4 and 8, 9 are with the estimator as observer, the navigation estimates are not fedback into the loop. Experiments 6 is closed-loop with navigation included, GNC with the navigation algorithm, where the true values of the variables are fed to the GC algorithms. 138
6.14 Description of the Monte-Carlo uncertainty parameters used on the simulator. 141

7.1 Initial Navigation conditions for attitude and angular velocity. s/c 1 is equal to s/c 2 and s/c 3. 162

Chapter 1

INTRODUCTION

1.1 Motivation

A current trend in present and/or future space and Earth missions is the use of multi-spacecraft systems operating in formation. The multi-spacecraft concept involves replacing a single spacecraft (s/c) with many small, inexpensive spacecraft moving in a number of different configurations (formation, cluster, constellation). The deployment of several s/c in large-scale formations enables scientific objectives to be accomplished that require higher resolution Earth observation, higher accuracy in extra solar planetary observation, or greater region coverage than would be possible using monolithic platforms. For example, ground observation with a single mirror telescope at very high spatial resolution of about one metre in the visible spectral band would require an aperture of about 30 metres. Such a monolithic mirror telescope would have much larger mass and would require a larger volume to be accommodated inside the launcher than would a multi-spacecraft solution (*e.g.*, based on interferometry).

Attaining such goals requires flying the s/c in a formation with impressive control and estimation accuracy, detecting the deviation of millionths of a metre from the nominal position of one s/c by another s/c located hundreds to some kilometers away. Therefore, the role of navigation, *i.e.,* the estimation of the formation state vector at regular intervals, becomes of increased relevance, since high accuracy, sometimes to millimeter level, may be required to accomplish particular scientific objectives. The multi-spacecraft concept leads to more technology demands *w.r.t.* Guidance, Navigation and Control (GNC) algorithms, but provides greater autonomy and robustness. The ability to remotely add, remove or replace individuals elements after launch provides flexibility, in that the system can be reconfigured for different missions, and robustness, in that not only can the system continue to function in the event of individual s/c failure, but the failed s/c can also be replaced. Furthermore, economies of scale enable a

multi-spacecraft system to be less expensive to manufacture, and to be better able to take advantage of the latest technologies, than a single s/c system.

However, multiple-vehicle systems present a significant challenge to the current methods usually applied to single-vehicle platforms. The cooperation framework brings new problems, such as undesirable bottlenecks within inter-vehicle communication and the requirement for large volumes of data processing, in addition to the need to avoid collisions between elements particularly in tight formations. Neither are control and navigation algorithms trivially transferable. If we consider the Formation Flying (FF) configuration, for example, an extra challenge arises, as greater levels of autonomy, real-time coordination and continuous relative navigation and control (while taking into account the actions of the other elements) are required than for any other type of configuration. The main feature of the FF vehicle configuration is that the geometry of the formation is designed for a purpose, and operational steps are taken to maintain that geometry during the mission lifetime. Usual constellation and cluster configurations, on the other hand, requires less accuracy/precision in the relative navigation and control, since they require cooperation among vehicles only on a kilometre scale. Moreover, the cluster and constellation configurations do not require a fixed geometry but as the vehicles are located in different (though similar) orbits and include redundant vehicles so a single failure would not jeopardise the mission.

Most of the time, relative measurements among vehicles (*i.e.,* inter-vehicle measurement) are critical, especially in deep-space s/c missions where absolute positions cannot be measured or inferred (*e.g.*, by Global Positioning System (GPS)). Thus, inaccuracies in localization of the elements in the formation during the execution of the control commands may prevent the group reaching its goal or even jeopardize the whole mission. Therefore, concerning the navigation system, the navigation sensors spreading all over the team elements and computational effort distributed across the elements of the formation require new design definitions not existing in a single vehicle framework.

The purpose of this work is to devise and investigate a novel solution for the navigation of N-vehicle formation. This work was motivated by the need to develop a method to estimate the six degrees of freedom (6DoF) translational and rotational components of a full-order state vector of a N-vehicle formation, using a decentralized architecture, and avoiding fully-connected communication and measurement networks. The advantageous of modularity, flexibility, scalability and robustness to single vehicle failures make this type of design attractive. The target missions are those with vehicles that do not have absolute positioning sensors available on board. An example are deep space s/c missions, where the access to GPS signals is not possible or is very limited. An example of these are Geostationary Transfer Orbit (GTO) missions, where the translational state vector is usually related to the orbital frame or the body frame of a reference s/c, expressing the relationships between the s/c in terms of relative position and translational velocity. However, the full state is only complete with the rotational state estimates

vector, *i.e.,* attitude and angular velocity. These variables are usually defined as absolute because their components are usually based on inertial measurements, *e.g.*, Star Tracker (ST) measurements, which are defined *w.r.t.* Earth linked frames.

1.1.1 Proposed approach for a vehicle

The recursive filter used to estimate the full state of a N-vehicle formation is the EKF (Extended Kalman Filter) which, despite being suboptimal, is convenient for its implementation simplicity, due to its direct derivation from the state space model, and which also enjoys wide popularity for its accuracy and speed. However, due to the decentralized architecture, the estimates have to flow through the entire formation, either in full order filters or through reduced order filters, in order to update the state vector for the entire formation, as stressed by [1]. This can be critical for the estimator stability when this information is combined twice, which could happen due to a number of different factors. Since none of the individual elements knows the overall system, the state variables can not be assumed to be independent and even the noise can be correlated. This jeopardizes the advantage inherent in the decentralized architecture, such as flexibility, robustness, modularity or extensibility [2]. An algorithm, called Covariance Intersection (CI) Algorithm, assumes that the true statistics are unknown and allows filtering and estimation to be performed in the presence of unknown correlation between estimates that capture the relative relations between vehicles within the formation. In contrast to the EKF, the measurements and the state estimates are assumed to be independent denoting mutually uncorrelated variables. Therefore, the CI algorithm is an interesting approach, offering the possibility to deal with variables with unknown cross-covariances, while maintaining the consistency of their estimates, as suggested in [3].

This work addresses the N-vehicle formation whose motion can be approximated by Clohessy-Wiltshire equations when parametrized by the true anomaly, as described in detail in appendix C (subsection C.3.3), and assuming no perturbations. For this case the translational and rotational motions have decoupled dynamics, making it possible to handle separately their state vectors. Therefore, the proposed estimation filter is divided in two parts, to determine the translational and the rotational state vector estimates. The filtering part of the estimation algorithm is organized as follows: the calculation of the local state estimates through the local sensor measurements performed by the EKF, and the update of the remaining state vector variables by using the previous vehicle state navigation knowledge. In the last step, a correlation problem arises leading the EKF to diverge for the translational relative state vector part, while for the rotational part, which is absolute, such a correlation problem does not exist. Thus, for the translational part of the state vector, a Covariance Intersection (CI) algorithm is used to overcome the correlation problem, though at the expense of reduced accuracy since the error of the combined estimates is lower bounded by the error of the EKF. For the rotational part of the state

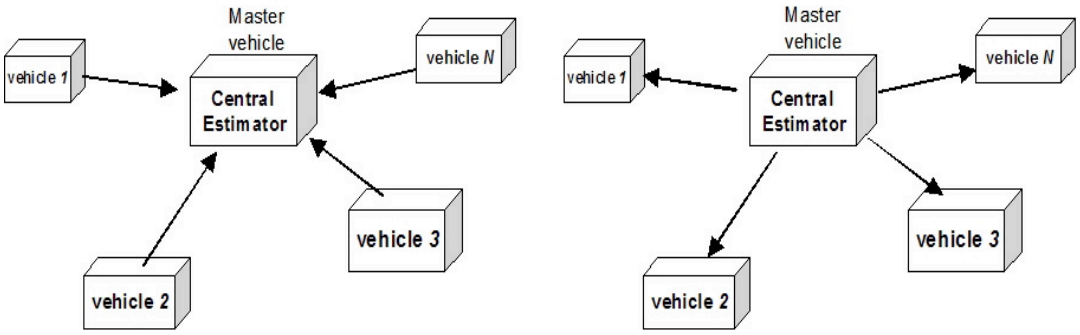

Figure 1-1: Centralized architecture showing a) the measurement network and b) the communication network. The arrow direction in a) represents the flow of sensor information, *i.e.,* in a) the master vehicle estimator receives sensor measurements from s/c 1,2,3,..., N, and in b) broadcasts the state estimates of the formation back to each vehicle.

vector, the EKF can be applied, since the state vector is absolute.

1.1.2 Centralized *vs* decentralized architectures

The major issue that arises in the literature on GNC, Guidance and Control (GC) strategies for cooperative FF vehicles is the development of estimation and control algorithms under a suitable general architecture that scales well to a large number of formation elements. The formation state estimation architectures are usually categorized in the literature according to three classes: Centralized, Decentralized and Distributed Architectures [4], [5], [1], [6].

In the *Centralized* approach there is a single estimator (master vehicle) that collects the sensor measurements, control inputs and configurations of all vehicles, and estimates the formation state, broadcasting the results to the formation elements, as shown in Fig. 1-1. Conceptually, the estimation algorithms used are similar to those for a single vehicle where the decision-making is concentrated in the master and where there is no possibility of task, or sensor fusion duplication. The most common estimation filters in this case are the Kalman filter and the Information filter. The Kalman filter is a well know filter which for linear systems minimizes the mean square estimation error of the state vector at a single time, based on the measurements at that time and gradually reducing importance on previous measurements. However, high computational load with bottlenecks and large communication bandwidth are some of the main disadvantages of a central filter estimator. Also, the need to have global knowledge of the complete formation state makes it difficult to add or remove formation vehicles or to change dynamically the formation structure. Above all, the reliance on a single vehicle for all computations is a risky solution, vulnerable as it is to vehicle failure.

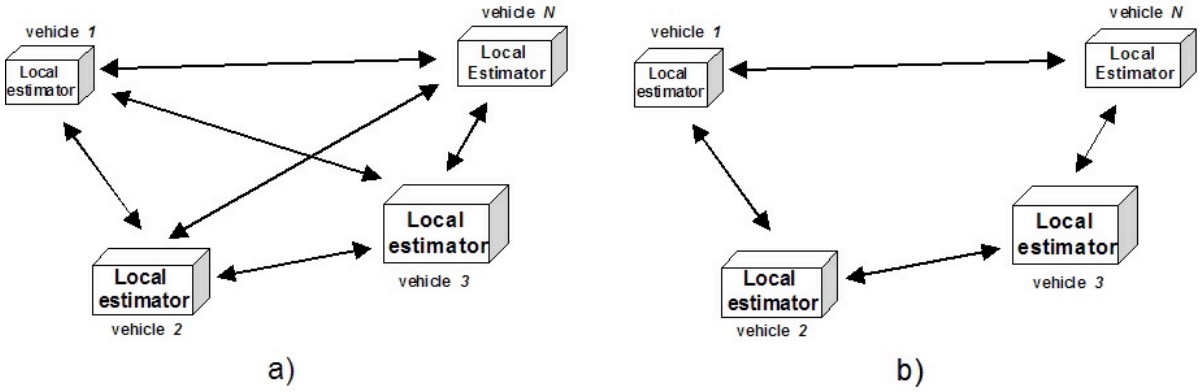

Figure 1-2: Decentralized architecture with a) the measurement network and b) the communication network. The networks are not centralized but instead, the measurements in a) are collected by each spacecraft and, the communications in b) are processed among spacecraft, the amount of communications being minimal in this particular example.

In the *Decentralized* estimation approach there is no master estimator but instead all vehicles have identical estimation algorithms, as shown in Fig. 1-2, and each vehicle estimates the subset of the formation state variables required for control. The data itself is collected in a decentralized manner, which spreads the computational effort through all the vehicles. There is no master where the global estimates are computed since each vehicle processes its own estimates with local system knowledge. Only peer-to-peer (*i.e.,* between vehicles) communication is allowed. This also has the advantage of the decentralized architectures by having the computations already distributed across the formation elements.

One intermediate solution is the *Distributed* [5] or *Hierarchical* [4] approach, where each vehicle has its own local estimator, but the information must be integrated by a central estimator, possibly a virtual one, that combines the local estimates into a formation state estimate, as shown in Fig. 1-3. Compared with the centralized architecture, this is a useful approach to reduce the communication within the formation and to spread the computational effort through the overall formation at the same level, as well as to overcome the synchronization problem from the iteration techniques in the centralized architecture or due to the GPS system for s/c missions. There are variations of this kind of architectures [7], but most of them still retain some of the disadvantages associated with the centralized architecture. For example the hierarchical clustering architecture considered in [1] describes a topology which is a mix of centralized and decentralized architecture applicable to formations with large numbers of satellites. This kind of architecture splits the whole formation into sub-formations in a decentralized manner, where navigation, control, estimation and navigation are performed independently. The sub-formations

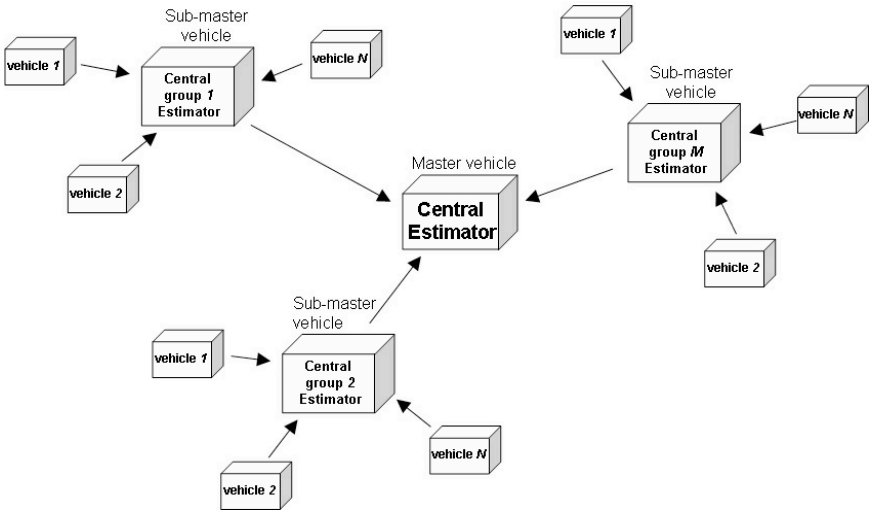

Figure 1-3: Distributed or Hierarchical architecture measurement network. The central group estimator $1...M$ represent the sub-masters of the sub-formation $1..N$- vehicle, that collect the measurements to the master vehicle of the overall formation.

are then organized in a centralized topology, where the master element is the only one linked to all the other sub-formations, as shown in Fig. 1-4. The main advantage of this kind of topology is that the several levels of sub-formations do not need to be tightly synchronized or to run at the same update rate. Also, any of the algorithms can be applied at any level of the structure. However it is still, to some degree, vulnerable to communication bottlenecks.

Most pioneer work is based on *centralized* solutions, where data is gathered and processed at a central node. Modern work has concentrated on *decentralized* solutions, where the information is processed locally at each vehicle, taking advantage of the local data availability and of the resulting distributed computation. It has been shown [4] that fully connected decentralized solutions can obtain the same results as a centralized data fusion system. However, this is achieved under fully connected sensor topology and maintenance of a complete global system model at each element of the formation. This means limited scalability and usage of communication and computation resources [4], leading to very limited use for large systems.

A summary of the main approaches to formation navigation found in the literature, as well as their pros and cons, follows.

Centralized: There is a single estimator (master vehicle) that collects the sensor measurements, control inputs and configurations of all vehicle, and estimates the formation state, broadcasting the results to the formation elements. Conceptually the estimation algorithms used are similar to those for a

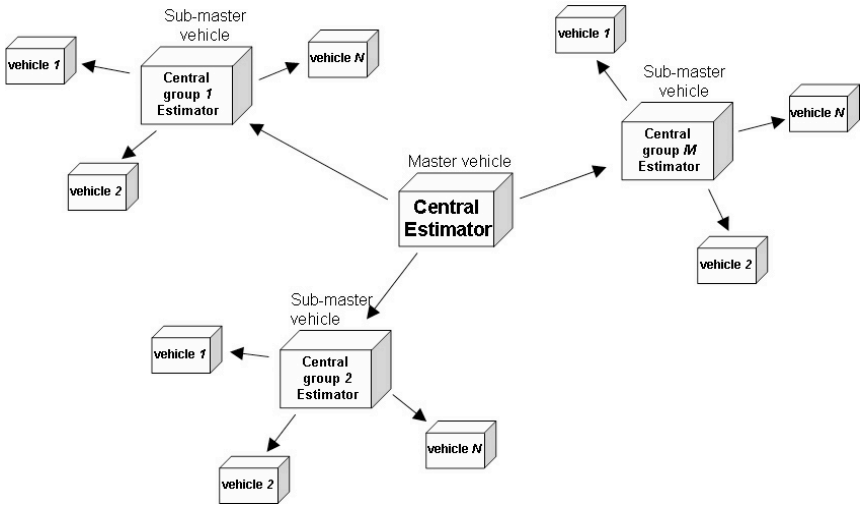

Figure 1-4: Distributed or Hierarchical architecture communication network. The central estimator $1...M$ represent the sub-masters of the sub-formation $1..N$, that communicate information to the master vehicle of the overall formation.

single vehicle.

- **Solutions**: Kalman filter, Information filter.
- **Pros**: full state knowledge, optimal solution for linear systems.
- **Cons**: high computational load with bottlenecks, high communication bandwidth, low robustness to failures, difficult to add or remove formation vehicle to change dynamically the formation structure.

Decentralized: There is no master estimator and each vehicle estimates the subset of the formation state variables required for control. The data itself is collected in a decentralized manner, which spreads the computational effort throughout all vehicle. Only peer-to-peer (*i.e.,* between vehicle) communication is allowed.

- **Solutions**: Full Order Filter (FOF), where each vehicle estimates the entire formation state and Reduced Order Filter (ROF), where each vehicle estimates only its local state with some information (either the state itself or some elements of the state covariance matrix) of the entire formation in order to linearize its measurement equations.
- **Pros**: computations are already distributed across the formation elements.
- **Cons**: some of the solutions still require considerable communication because state estimates must be passed on to the teammates.

Distributed or Hierarchical: each vehicle has its own local estimator, but the information must be integrated by a central estimator, possibly a virtual one, that combines the local estimates into a formation state estimate.

- **Solutions**: An interesting variation is to split the formation in several independent sub-formations, whose information is then centralized by a master.
- **Pros**: no tight synchronization or same update rate required, reduced communication bandwidth *w.r.t.* centralized approach.
- **Cons**: still vulnerable to communication bottlenecks.

Due to the decentralized advantages/disadvantages compared to the other architectures, the decentralized architecture was chosen for the formation state estimation architecture in this work. There are several issues with the flow of information using decentralized architectures, and these will be described in more detail in the following subsection.

1.1.3 Decentralized architectures and communication network

Depending on the communication effort required to spread each vehicle measurements, we can consider two different choices for the state vector dimension: the Full Order decentralized Filters (FOF's) and the Reduced Order decentralized Filters (ROF's) [1]. A FOF is algebraically equivalent to the centralized form of the filter, where the entire fleet measurements are available to each vehicle to compute the **entire formation state estimates**.

The ROF's are distinguished by computing only **local state estimates** through a local filter with some information (either the state itself or some elements of the state covariance matrix) of the other formation vehicles in order to minimize the communication links among the formation vehicles and to linearize the measurement equations.

In [4], the communication among the vehicles is made selectively, according to when the vehicles have relevant information to share. For the reduced order case the inter-vehicle communication is independent of the number of members, as opposed to the full-order case [1]. Thus, the full-order solution has a higher computational and inter-vehicle communication load, making the reduced order the best solution with respect to communications. According to [4], the limitations of the fully-connected decentralized estimators architecture are related to redundant communication, limited scalability, excessive computation and vulnerability to communications link loss. Thus, these kind of topologies have very limited practical use for large scale systems.

Since the knowledge of the full state vector of the entire FF at each vehicle is usually required from the control and guidance point of view, **it is desirable to design decentralized estimators that**

estimate the full state vector without having the fully-connected communication network. In this way, the disadvantages associated with the fully-connected case, discussed before, are avoided but still the full state vector is estimated at each vehicle. This is the approach followed in this thesis.

1.2 Goals and Original Contributions

The main goals and contributions of this work are:

- To address the problem of navigation of a N-vehicle formation, *i.e.*, to estimate in an integrated way, the full formation state vector (full 6DoF, translational and rotational state vector) of the full N-vehicle formation in each vehicle of the formation. The formation navigation setup targets the challenging decentralized architecture (no master estimator), with information flow only allowed between peers (vehicle i and $i-1$, with $i = 1, .., N$) and measurements available on each vehicle are the ones sensed by each vehicle and not the measurements from all vehicles in the formation. This work, also contributes to the problem of estimating the local (vehicle i) non-observable relative state vector components, $\hat{\mathbf{x}}_{\bar{o}} = \hat{\mathbf{x}}_{jl}$, with $j, l = 1, .., N, j \neq l \neq i$, *i.e.*, all relative state vector components between vehicles that do not include vehicle i. This problem arises from estimating the 6DoF full relative formation state vector on each vehicle. The proposed algorithm combines an EKF and a CI estimator in an integrated way to estimate locally the full 6DoF state vector using a graph framework. Using two graphs, (*measurement* graph and *information flow* graph) and the EKF+CI, a decentralized formation navigation architecture is developed.

- A method to analyse systematically the robustness of the formation to the loss of communications and/or distance sensor in one or more of its vehicles based on graph theory. Each vehicle represents a vertex on the graph. The measurements available at each vehicle are represented as edges in the *measurement* graph and the state vector estimates flowing between vehicle peers are represented as edges in the *information flow* graph. The algorithm computes the number of state vector components possible to estimate at each vehicle of the formation; the vehicles that do not contribute for the formation navigation determination; and also those where it is not possible to determine the full 6DoF locally. This contribution can be used as a N-vehicle formation management tool, enabling the classification of defective vehicles and the possibility to put them aside from the core formation without affecting the mission.

- Demonstration of the estimator results in a very realistic simulation of a FF s/c mission. The only way to validate a multiple spacecraft space mission on Earth is through software platforms. The decentralized formation navigation architecture developed was validated in such a platform where

3-FF s/c orbit the Earth in Geostationary Transfer Orbit (GTO), aiming to do science in the apogee. The navigation algorithm efficiency by comparing its performance bounds to the PCRLB was performed in the given platform.

1.3 Thesis Outline

The present document is organized as follows:

Chapter 2, **Literature Survey** - The relevant literature on Formation navigation is reviewed, concentrating mostly on the contributions from robotic and space communities.

Chapter 3, **Problem Formulation** - The navigation problem of locally (at each vehicle of a N-vehicle formation) estimating the 6DoF translational and rotational components of a full-order relative state vector of a N-vehicle formation is solved. This is done in a decentralized way, with local filters sharing information with each other, within a minimum communication framework. The work developed presents a general formulation that can be applied to the problem of estimating the full state of a N-vehicle formation. Graph theory is used to model the measurements and the information flow within the formation. The work contributes with a graph based algorithm to analyse the impact of a connection breakdown (due to partial or total failure of a vehicle's measurement and/or communication links) in the navigation algorithm performance.

Chapter 4, **6DoF Decentralized Formation State Estimation -** Considering the formation navigation problem, defined in the preceding chapter, the design of a formation navigation filter is studied using algebraic graph theory, and an integrated algorithm is proposed to estimate the translational relative state vector from relative measurements. Using two graphs, *measurement* graph and *information flow* graph, and the EKF+CI, a decentralized formation navigation architecture is developed. Moreover, the algorithm is divided in two parts, depending if the state vector components are estimated through the local measurements represented by the *measurement* subgraph or if the locally non-observable state vector components are estimated through the state vector estimates transmitted from the previous vehicle, represented by the communication links in the *information flow* subgraph. Also, the 6DoF full-order state vector estimation algorithm is divided in two parts: one estimates the translational relative state vector of a N-vehicle formation, consisting of the relative position and translational velocity, and the other estimates the rotational absolute state vector of a N-vehicle formation, which is defined in terms of quaternions and angular velocity. Moreover, the correlation problems arising from estimating the translational relative state vector is as well addressed in this chapter, and also is presented the navigation algorithm adaptation to handle quaternions.

Chapter 5, **EKF Analysis Issues** - The filter stability for linear and nonlinear systems is discussed,

in particular using the observability concept. Also the Posterior Cramer-Rao Lower Bound (PCRLB), which evaluates whether the suboptimal EKF is close to the optimal estimate or not, is discussed. This study includes the consistency test, which is very important for nonlinear systems due to the approximations made, tests for bias, and also tests whether the actual MSE matrix matches the covariance matrix assumptions. These concepts are applied to 3-FF s/c, in the situation where the observation and process noises matrices are tuned according to the Normalized Estimation Error Squared (NEES) and Normalized Innovation Squared (NIS) tests, based on a GTO mission simulation.

Chapter 6, **Simulation Setup** - The case study for the application of the algorithms is introduced[1]. A description of a realistic software platform, the FF-FES simulator, is performed, as well as of the Darwin mission and FF s/c orbit is presented. Moreover, the navigation algorithm is applied to the particular case of 3-FF s/c, including a brief introduction to the relative dynamic equations for an eccentric orbit, subsequently particularized for the relative navigation problem and also for the dynamics of the rotational part of the vector. Some aspects of the implementation of the algorithm in the simulator are also addressed in order to compute the 6DoF state vector according to the proposed filter design. The simulation experiments carried out are described in detail.

Chapter 7, **Results -** This chapter collects some results for 3-s/c flying in formation[1], obtained using the FF-FES[2] simulator, as an illustrative example of the concepts introduced in Chapter 4. The algorithm is successful validated for the most relevant phases of the GTO, such as:

- the science experiment Baseline Control Mode (BCM) where the accuracy requirements are higher;
- the initial deployment Formation Acquisition (FAC) mode where the formation starts from an initial random disposition (with the s/c far away from each other) and moves to a final geometric topology where the specified distance among s/c require the estimation algorithm to track the formation state.

Monte-Carlo tests are also presented, where the formation state estimate is accomplished while starting from an initial value different from the true state value, under the effect of several unmodeled perturbations.

Results also include the estimator in the loop with the GC algorithm[3] [8] for the FAC mode and with a PID control for the BCM mode, using an optimized low thrust profile, [9]. The results presented are obtained with RF sensor and with pseudo-code signals for the translational state vector part and

[1] developed by ISR/IST in scope of the ESA project "Formation Flying Methodologies for Distributed Spacecraft" - ESA RFQ/3-10624/03/NL/LvH/bj in March 2005.
[2] developed by DEIMOS Engenharia, Lda.
[3] developed by ISR/IST in scope of the ESA project "Formation Flying Methodologies for Distributed Spacecraft" - ESA RFQ/3-10624/03/NL/LvH/bj in March 2005.

ST sensor for the rotational state vector part. Monte-Carlo simulations are also performed for the assessment of the performances of the closed-loop GNC design *w.r.t.* model parameter uncertainties and unmodeled nonlinearities, as well as to errors in the initial conditions.

Chapter 8, **Conclusions and Future Work** - In the last chapter we review the contributions of the thesis and present the overall conclusions, as well as suggesting several important topics for future research.

Appendix A, **Graphs** - In this appendix relevant concepts of graph theory used in this work are revised, as well as the concepts used for the matrix representation of the graphs.

Appendix B, **Extended Kalman Filter** - In this appendix the EKF is derived from a general continuous-discrete time nonlinear system. Also the EKF equations applied to the case study are presented.

Appendix C, **Relative Dynamics** - In this appendix the equations of s/c relative dynamics for an elliptic orbit are derived.

Chapter 2

LITERATURE SURVEY

2.1 Introduction

This chapter attempts to summarize the major contributions from the space and robotic communities towards the design and development of multi-vehicle systems capable of autonomous formation navigation. Since a s/c formation aims to be more efficient going beyond the restricted capabilities of a single s/c its accuracy requirements, especially concerning relative variables, are high. Therefore, the first section addresses the impact of the relative distance between s/c on state estimation accuracy, reported in the literature, and provides some examples of future planned missions where these interests are reflected. The second section covers the contribution from the Robotics literature, followed by two sections, covering the formation state vector estimation topics for translational and rotational components from the space literature. Since the sensors have a direct impact on the accuracy of the state vector estimates, Section 2.6 focuses on sensor technology relevant for formation navigation and a new approach based on optical sensor.

2.2 Formation Flying Spacecraft State Estimation

An adequate filter should operate within a real-time autonomous FF control architecture. The focus of the state is in the accurate relative state between the elements of the formation, since it is the most important requirement if aimed at accomplishing a mission as a group. In certain missions, the accuracy of the relative state formation is within the order of centimetres when the distance between s/c is in the hundreds of metres. High accuracy is more difficult to obtain for long baselines. Also the accuracy of the relative state estimation has a strong impact for any control method that takes into account the

design of fuel-optimal trajectories [10], [11]. Tillerson [10] has shown that, in particular, relative velocity error plays an important role in the FF performance where an error of $\pm 0.002[m/s]$ results in $\sim 30[m]$ in track position offset after only one orbit. In fact, the velocity error has an effect ~ 450 times larger than the relative position errors between two s/c, in terms of the secular intrack drift, which is comparable to the drift expected from differential $J2$ disturbances.

Examples of FF space missions under study and specification, showing the importance of accurate estimation of the state, are:

- The Deep Space mission 3 (**DS-3)** will control multiple s/c in a heliocentric orbit, to within a fraction of the wavelength of light in order to perform optical stellar interferometry [12]. Two of the three s/c will collect the starlight that will be combined by the third s/c that will perform the interferometer detection. Coarse and relative stationkeeping are performed by the Autonomous Formation Flying (AFF) sensor and for more precise relative position, laser metrology is used. The AFF is a GPS-like sensor that internally generates GPS models (instead of using signals from GPS constellation) that are received and processed by receivers to enable accurate positioning and orientation of the multiple s/c. The AFF sensor is believed to accomplish $\sim 1[cm]$ accuracy for relative position and $\sim 1[mm/s]$ for velocity through the differentiation of the carrier phase signal between the receiving AFF sensor signals [13]. This type of sensor will be very important to deep space missions, where the formation is well beyond the coverage of the NAVSTAR GPS constellation.

- In the **TPF** mission, observations will study all aspects of planets outside our solar system: from their formation and development in disks of dust and gas around newly forming stars to the presence and features of those planets orbiting the nearest stars; from the numbers at various sizes and places to their suitability as a support for life. By combining the high sensitivity of space telescopes with revolutionary imaging technologies, the TPF observatories will measure the size, temperature, and placement of planets as small as the Earth in the habitable zones of distant solar systems. The related program in ESA, **Darwin** - three space telescopes plus one to serve as communication hub (launch foreseen for 2015) will scan the nearby universe, looking for signs of life on Earth-like planets, and will possibly be placed in a highly eccentric orbit, 1.5 million kilometres from Earth.

- The TechSat-21 program [14] seeks to develop fleets of low-cost, highly capable microsatellites to replace large, expensive communications and remote sensing satellites. A space-based radar mission for Ground Moving Target Indication (GMTI), combines micro-electromechanical systems (MEMS), sparse aperture design, and formation control strategies. In order to accomplish the

desired resolution, the requirement regarding real-time relative position estimation accuracy is about 3[cm], with distances ranging from 500 to 5000 metres between s/c. An example of the initiatives under the TechSat-21 program is the University Nanosatellite Program that involves the development of up to ten low-cost university s/c. The ORION and Emerald missions are examples of this program. The **ORION** is a student-built mission [15], designed to be the first to use carrier-phase differential GPS (CDGPS) for precise relative navigation of microsatellites and control of the FF. The Orion s/c operates in a semi-autonomous fashion which reduces the need for frequent earth communications, and has a cold-gas propulsion system for maneuvering the vehicles, allowing a real-time demonstration of the FF control using various control modes and formation configurations. In order to accomplish the accuracy of the relative position and velocity, a GPS receiver instrument is provided which is believed to accomplish $\sim 1[cm]$ accuracy for relative position and $\sim 1[mm/s]$ for velocity, through the differentiation of the carrier phase signal between the receiver GPS signals [15].

- Another example of using interferometer techniques to achieve higher order resolution in Earth observation is the project EUCLID CEPA 9 RTP 9.9 "*High Resolution Optical Satellite Sensor*". This was a project under Alenia Spazio leadership, that aimed at studying the design and implementation of a laser metrology breadboard to be used as an internal sensor in the co-phasing demonstrator, to be implemented in single s/c with several arms, each of them supporting a set of smaller aperture sub-telescopes equivalent, for resolution purposes, to a single monolithic mirror telescope. Using the synthetic aperture technique, such a device reconstructs the original image of an object, starting from the image formed on the common focal plane of the multi-aperture telescope. To this purpose, the set of sub-telescopes must observe simultaneously the object while keeping constant (within a fraction of the wavelengths) the phase difference between the various wavefronts, which are combined together so as to fulfill the coherence and co-phasing conditions, and to operate as an interferometer. The main task of the control system is to cancel the optical path difference between the interferometer arm pairs, by interferometric measurements, to obtain a polychromatic fringe coherence wave.

2.3 Robot Formations State Estimation

Multi-robot teams are normally required to know, either with high or coarse accuracy, the state vector, *i.e.*, the posture (position and orientation) of all elements of the team. Partial or total knowledge of other robot coordinates in a reference frame is crucial to accomplish team coordination. For example, in a football robot team, knowing the localization of all the members, or at least of those closest to

the ball in the playing field, is essential to fulfill their ultimate goal, *i.e.*, to score [16]. The problem of estimating all members of a team of robots in a global frame is part of the problem of mobile robot localization. Cooperative localization is a problem in a multi-robot scenario, where the elements of the group cooperate among themselves to achieve the same goal [17], [18].

In a unit-centered reference or leader reference type of control architecture, each robot must estimate the other robot's state and its own state in a global system. In this case, the centralized estimation architecture appears as the natural candidate. Despite this dependence on the control strategy, other requirements are much more decisive, such as the amount of communications, failures, accuracy, computational load and uncertainty in the localization. Usually in mobile multi-robot systems, the main concern centres around the localization and mapping - either simultaneously (SLAM) [19] or cooperatively (CLAM) [20] - with the control problem being the dominating issue and localization being relatively minor: indeed the control actions in the group directly relate to the success of the mission. However, in a multi-robotic framework, the localization problem has a much higher order of complexity and may directly enable the group to accomplish their goal or mission because it provides the information for the multi-system elements to perform closed-loop, real-time Guidance and Control (GC). Also, it demands higher reliability than for single-robot systems since the uncertainty in the localization of the members may propagate through the system making the group much more sensitive to such uncertainty [20]. Often, the localization uncertainty accumulates and is corrected against landmarks or based on sensor-based motions planned to reduce the uncertainty. Thus, for team cooperation, reliable and accurate state estimation must be continually delivered. To achieve these requirements an estimation filter computes the state vector of the individual members of the multi-robot using *cooperative (collaborative) localization* approaches where the robots combine sensor measurements of other members to improve localization performance in several ways [18].

Roumeliotis *et al.* [17] emphasized the importance of localization and explored the localization accuracy of a team of mobile robots (where the localization problem involves a much higher order of complexity than for a single-robot) by proposing a Kalman Filter (KF) estimator for a group of robots moving simultaneously. The authors proposed an Extended Kalman Filter (EKF) for reduced order position state estimation, distributed across the robot team elements, but the heterogeneous sensor data is collected and fused in a centralized KF algorithm. Moreover, the absolute positioning is assumed to be always available and does not take advantage of the relative sensor information but of physical landmarks. Fox *et al.* [18] claim that exchanging robot's relative estimations will improve individual robot localization, thus reducing the uncertainty in the estimation location in a global localization context and in an indoor environment. In order to do this, they propose a particle filter, an extension of the single robot Monte Carlo localization algorithm to the multi-robot frame. However, the interdependency

of the state estimates due to the relative measurements is ignored, leading to overly optimistic estimates. A Maximum Likehood Estimator (MLE) is used by [21] considering a centralized schema, and in [22] a Distributed MLE approach is proposed where each robot is used as a landmark, allowing the team of robots to move in an outdoor environment without considering physical landmarks. A set membership approach to the SLAM problem for a single robot is used in [23], where the disturbances coming from the dynamic model and from the measurement model are assumed to be unknown but bounded, and no statistical assumptions are made regarding the nature of the errors.

2.4 Types of Formation Translational State Estimators

When surveying the literature on Navigation strategies for formation vehicles, many relevant issues for the formation navigation problem are available in the literature on robotic vehicles and spacecraft control, where the control problem is the dominating issue, but none propose or study a full-order decentralized estimate filter to estimate the formation state. Moreover, most of the work performed for formation navigation only consider the translational components of the state formation vector and not the 6DoF, the translational and the rotational components of a full-order state vector of a N-spacecraft formation.

The most typical estimator filters are divided in two main categories: the **Full Order Filter (FOF)**, where each satellite estimates the entire formation state using a fully-connected communication network, and the **Reduced Order Filter (ROF)**, where each satellite estimates only its local state with a peer-to-peer communication network.

The FOFs commonly used are the Decentralized Information Filter (DIF), also reported in [4], which is a decentralized version of the Information filter, the Decentralized Kalman Filter (DKF) also reported in [4], which is a state space version of the DIF and the Full Order Iterative Cascade Filter. Each of the filters first locally estimates the entire formation state using only the local measurements in each satellite, and then each satellite transmits its state estimate to every other element in the formation [24]. However, FOFs require a **fully-connected communication system** in order for each vehicle to update the information from every other vehicle. Also, to prevent instability, in addition to the information vector, the information matrix must also be transmitted at every time step, leading to intensive communication.

In ROFs the main idea is to have communication only between vehicles close to each other, instead of a fully-connected system, thus minimizing the communication in the formation. Some ROFs recently reported (*e.g.*, in NASA GRC's Decentralized Relative Navigation Algorithms for Adaptive Inter-Spacecraft Communications Networks project) are the Iterative Cascade Extended Kalman Filter (ICEKF) [24] and the Schmidt Kalman Filter (SKF) [1]. The ICEKF is identical to the full-order version

with the difference that each vehicle only estimates their local state instead of the entire formation state, and then passes the results on to one of its teammates. However, it presents communication problems because it takes several iterations through the entire formation before the algorithm converges. The SKF is a reduced version of the KF where not only the local state vector is transmitted to the next vehicles but also some important parts of the covariance matrix in order to prevent instability. A possible solution would be an algorithm that estimates the full order state vector (as ICEKF but avoiding the iteration problem), transmitting the covariance matrix besides the local state vector to prevent divergence problems in the KF (as SKF does), but considering the predecessor state vector estimates as a "measurement". However, the new "measurement" and the local state vector estimate are correlated and the degree of correlation is unknown. This will lead to convergence problems.

A new method, Covariance Intersection (CI) algorithm [2], [25] combines the information of random variables and computes the state variables without knowing the degree of correlation among them. CI is a solution to be used for the decentralized estimation to overcome the problems resulting from the use of the Kalman Filter, discussed above.

Most of the algorithms concerning estimation of FF are designed for LEO satellites. Recently, GPS techniques were developed to perform not only orbit determination and absolute positioning but also precise relative navigation, where GPS measurements are used in the estimation system to determine the relative position and attitude estimates through differential carrier phase measurements [26], [27], [28], [29], [30], [31], [32]. In comparison with the conventional vehicle sensors such as star trackers, horizon and Sun sensors, and inertial measurements systems, the GPS-based estimation methods offer much lower cost with higher accuracy, approximately $0.01[m]$ for relative position and $0.5[mm/s]$ in relative velocity [30]. However, most GPS techniques require data from the NAVSTAR GPS constellation, a solution which is not applicable for deep-space orbits. Nevertheless, GPS techniques can still be used in high orbits such as GEO or with limited availability of GPS signals to solve relative position and attitude among all s/c, by using pseudolites transceivers, which are signal generators that transmit and receive GPS like signals either by augmenting the GPS signal or by replacing them completely [33], [31]. The relative distance can be computed by differentiating measurements from antennas placed on separate satellites and the attitude by differentiating measurements from antennas placed on the same satellite. Unfortunately, the augmented measurements taken by each satellite are correlated, making it difficult to use the decentralized topology for the estimation system.

2.5 Attitude Estimation in Single Vehicles

The rotational state vector estimation is reported in the literature separately from the translational state vector estimate and is dedicated to the absolute state vector estimation obtained in a single s/c [34]. Most recent work [27], reports attitude estimates by taking advantage of differential measurements from GPS sensors and is therefore defined within an absolute frame. Moreover, most of the attitude algorithms are applied to a single s/c.

Most attitude estimation algorithms used currently are based on Extended Kalman Filter (EKF) or point-by-point (deterministic) approaches. Crassidis *et al.* [27] describe an algorithm to determine s/c attitude using GPS differential carrier phase measurements but only for one satellite. Also in [26] is described an algorithm for attitude determination, derived from the nonlinear predictive filter [35].

The deterministic approaches have several advantages over the EKF [36], [37], mainly because they do not require initialization at any time in the estimator algorithm and do not need initial knowledge of or any assumptions on the attitude. However, they require a minimum of two sensor measurements to work properly. In [38] a Moon-Sun attitude sensor is presented which takes advantage of the night-sky observations and thus computes the complete three-axis attitude estimation of a satellite using a deterministic algorithm: ESOQ [39]. However this has never been applied or analyzed in a FF scenario with an arbitrary number of satellites. Also in [1], a trade-off study of several estimators is proposed to assess which estimators are best suited to architectures according to the number of elements of the formation, as well as the impact of the computational effort required to perform the estimation calculations, the communication flow across the formation elements, and the performance. However, no single estimator algorithm is suggested for a general number of vehicles. Also, the robustness and stability of changing the estimator algorithms when the number of vehicles in the formation changes is not discussed, which suggests that changing the number of vehicles implies uploading a more suitable estimation system.

2.6 Range Sensors

Due to the demanding requirements of precise relative navigation in FF, sensor technologies are being developed to guarantee such specifications. A discussion about the navigation sensors for the determination of the relative position and orientation is made in [40] for s/c formations. Some of those sensors are: spaceborne GPS Receivers, laser tracking with inter-satellite links (*e.g.*, the Gemini mission, launched in 2004, is meant to demonstrate that the FF based on laser radar sensor and laser interferometer, augmented with GPS techniques, will give highly precise relative inter-satellite distances in a scenario of a two micro-satellite formation [41] in LEO orbit), and radar tracking with inter-satellite links (*e.g.*,

as described in [42]). Its accuracy and some implementation problems are also referred to. Also, [42] describes the Radio Frequency (RF) sensor, the Automonous FF Sensor (AFFS) [13], [43] based on GPS techniques, for precise relative position and attitude determination in real-time for the StarLight two-spacecraft stellar optical interferometer mission. The accuracy of the AFFS brings the two satellite's relative position within the range of the optical metrology system. However, as stressed in [42], robustness concerning recovery from temporary failure or time synchronization has not yet been addressed. Also, an overview of the sensor design and considerations about its limitations and performance over s/c manoeuvers and formation control design is made in this paper.

For deep space applications, where no observations of NAVSTAR GPS satellites are available at all, pseudolites can be placed onboard each element of the entire formation, acting as GPS-like signal transmitters and receivers. As a result, there are enough signals to compute the relative position and attitude of the formation elements. Therefore, the algorithms previously applied for LEO using differential carrier phase measurements can also be applied in deep-space missions [33], [31].

A new approach, the VISion-based NAVigation system (VISNAV) was proposed by [44] that comprises an optical sensor which provides a line of sight vector from the master s/c to the secondary satellite. This sensor is made up of a Position Sensing Diode (PSD) placed in the focal plane of a wide angle lens, which is used to see only specific light sources emitted by LEDs (beacons). The currents induced by the light emitted by the beacons in the PSD, after some mathematical manipulation, provide the line-of-sight vector. The VISNAV system is claimed to operate with reduced cost, size and weight compared with the existing autonomous FF sensor technology. Using this new technology, in [45] an approach to obtaining estimates of relative navigation and attitude of the flying formation through an observer system is described. This is an alternative to the EKF, which has the disadvantage of being sensitive to divergence, an effect usually reduced by redundant sensors. However, since the measurements are based on the line-of-sight projection onto the focal plane, the attitude and relative navigation cannot be decoupled. Although this problem is considered in the observer design, by designing two independent observers for attitude and position, the error statistics are assumed to be known Gaussian noise. Therefore, particle filters are an attractive method for solving this problem, since they do not assume a Gaussian probabilistic distribution for the uncertainties but instead deal with mullti-modal probabilistic distribution, more appropriate to the error statistics [46].

Chapter 3

PROBLEM FORMULATION

3.1 Introduction

In this chapter, we formulate the problem of local estimation of the 6DoF full-order state vector of a N-vehicle formation. 6DoF refers to the translational and rotational components. Full-order means that all components of the formation state are estimated. The architecture is decentralized because measurements are made locally at each vehicle and the full state is estimated at each vehicle, not in a single master vehicle.

Knowing that the relative components can be built up from the absolute components of the state vector, and that determining relative components based on relative measurements has wide applicability and is a more demanding problem than computing the absolute components, we denote in this chapter, the definition of the full state vector component to its relative part.

Typically, in a vehicle formation, each vehicle measures distances and angles *w.r.t.* other vehicles in the formation, and communication among vehicles is used to exchange relevant information for full relative state vector estimation.

Two graphs are used to model the formation state estimator:

- *measurement* graph, representing the relative measurements available at each vehicle of the formation
- *information flow* graph, representing the information flowing through the N-vehicle formation.

The advantages of using only local filters at each vehicle and local measurements are:

- reduced computational load when computing the filter updates;

- reduced dependency on communication of measurements to which the filter has no direct access.

However, in this case, the state vector is only locally observable, because each vehicle's local filter only uses local measurements, and thus the related state vector components are updated, leaving the remaining components without the means of being updated by sensor measurements. Therefore, the measurements sensed at each vehicle must be combined with information flowing through all the vehicles, to estimate the full relative state vector in a local filter. From this information fusion, different problems arise:

- the state vector components, which represent the relative quantities among vehicles in the formation, have coupled dynamics;
- there is correlation when local full relative state estimate is combined with the predecessor local full relative state estimate, causing problems in EKF information fusion. For example, if state estimates are derived from information flowing around the formation, and this same information is shared by different vehicle filters, a correlation problem arises between state elements, because the relative position between vehicles not measured locally depend on local distance measurements, making them correlated;
- the impact of a connection breakdown (due to partial or total failure of the vehicles sensor(s) and/or communication links) on the performance of the navigation algorithm must be analysed.

Also, it is in the formation's best interest to have the minimum number of communication links between vehicles, to avoid the large bandwidth required by a fully connected formation, and to reduce the filter algorithm sensitivity to communication failures. Thus, it is advantageous to devise an algorithm that estimates the full relative state vector without a fully connected communication topology.

In this chapter, we propose a framework to represent and analyse decentralized 6DoF full relative state estimation methods for a vehicle formation, using graphs. The formation navigation problem can be significantly simplified when the relationships between vehicles are expressed by means of the graph Laplacian, which captures the graph structure, and at the same time defines requirements necessary for formation state estimation. These requirements ensure that cooperation among the vehicles is observable, flexible and robust to topology changes of the *measurement* and the *information flow* graphs. *Cooperative observability* is used here in the sense that the information shared among vehicles within a formation leads to observability at each vehicle, since each vehicle has in its state vector the state vector components of all the other vehicles.

Inspired by the work of Zhipu and Murray [47] on multi-vehicle formation control strategy, we use three different graphs with the same vertices, distinguished by their edges. The vertices capture the

individual vehicle dynamics and the local estimator (*i.e.,* vehicle navigation filter), while the edges describe the measurements flowing through the formation for one of the graphs, the communications among vehicles for the second graph, and the relationships between vehicles for a third graph. The relationships between vehicles are represented by the components of the full relative formation state vector, which are to be estimated, are links in the *formation estimation* graph. This graph is meant to form a reference graph for the other two graphs (the *measurement* and the *information flow* graphs) such that the resulting solution will lead to the estimation of the formation state.

This chapter is organized as follows: in Section 3.2, we describe the *formation estimation* graph, which will represent the components of the full relative state vector to be determined by the estimator filter in each vehicle. The two graphs representing the measurements and the information flowing through the formation, are presented in Sections 3.3 and 3.4, respectively. With respect to inter-vehicle communication, Section 3.4 focuses on building an information graph that is as simple as possible, ensuring that any two vertices are always connected in a decentralized way. A framework to deal with sensor failure is also considered.

The chapter concludes in Section 3.5 with the formulation of the formation navigation problem based upon the elements described in the preceding sections. Some relevant concepts of graph theory, and their relationship with the formulation of the formation navigation problem, are described in Appendix A.

3.2 *Formation Estimation* Graph

Definition 1 (Formation Estimation Graph) *Let us consider the graph* $\mathbb{F}$*, such that* $\mathbb{F} = (V(\mathbb{F}), E(\mathbb{F}))$*, where the vertices* V *of* $\mathbb{F}$ *represent a finite set of* N *vehicles and* $V(\mathbb{F}) = \{v_i | i = 1, 2, ..., N\}$*. To each vertex* v_i *is assigned a filter that computes the estimates of the full relative formation state vector,* $\hat{\mathbf{x}}_{ij}(t)$*, where* $\mathbf{x}_{ij}(t) \in \mathbb{R}^{n_x}$ *is the full translational relative state vector and relative rotational state vector of a* N*-vehicle formation; the translational vector have position and translational velocity components and the rotational vector have attitude and angular velocity components (details in subsetion 4.1.1); the subscript* ij *is the vehicle's index and components of the relative state vector stand for relative relation between vehicle* i *and* $j \in \{1, ..., N\} \setminus \{i\}$*. Therefore, the edges set* $E(\mathbb{F}) \subset V(\mathbb{F}) \times V(\mathbb{F})$ *represents the inter-vehicle relative position and relative translational velocity, relative orientation (attitude) and relative angular velocity, quantified in terms of the state vector components,* $\mathbf{x}_{ij} = \begin{bmatrix} \boldsymbol{\chi}_{ij}^T & \dot{\boldsymbol{\chi}}_{ij}^T & q_{ij}^T & \boldsymbol{\omega}_{ij}^T \end{bmatrix}^T$*. Since* $\mathbf{x}_{ij} = -\mathbf{x}_{ij}$*, the directions of the edges are not relevant and thus the full formation state vector components* $\mathbf{x}_{ij}$ *are defined for* $i = 1, .., N-1, j = 2, .., N$*,* $j > i$*, and its dimension is reduced to half, i.e., there exist* $N(N-1)/2$ *edges with associated* $\boldsymbol{\chi}_{ij}$*,* $\dot{\boldsymbol{\chi}}_{ij}$*,* q_{ij}*,* $\boldsymbol{\omega}_{ij}$ *each. Thus, the formation estimation graph* $\mathbb{F}$ *is as shown in Fig. 3-1, with* $|V(\mathbb{F})| = N$ *and* $|E(\mathbb{F})| = N(N-1)/2$*.*

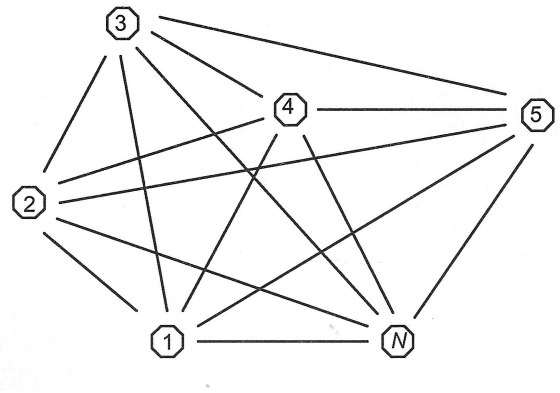

Figure 3-1: The *formation estimation* graph ₣, which corresponds to the representation of the components of the full relative formation state vector, which are to be estimated.

The Laplacian matrix for graph ₣, defined in (A.2), as,

$$\mathbf{L}(₣) = \mathbf{D}_g(₣) - \mathbf{T}_g(₣) \tag{3.1}$$

where $\mathbf{T}_g(₣)$ is the adjacency matrix, and $\mathbf{D}_g(¥)$ is the degree matrix, with the diagonal being the degree of each vertex.

The eigenvalues of the Laplacian, defined in (A.2), of a fully-connected *formation estimation graph* ₣ are,

$$eig\{\mathbf{L}(₣)\} = \left\{0^{(1)}, N^{(N-1)}\right\} \tag{3.2}$$

where the superscript stands for the eigenvalue multiplicity, the dimension of the Laplacian matrix being equal to $N \times N$. Notice that the *formation estimation* graph is a complete graph with N vertices and $N(N-1)/2$ edges, and therefore it has degree $N-1$, meaning that it is a $(N-1)$-regular graph. Throughout this work, an eigenvalue will be referred to either through its multiplicity, as $eig^{(1)}\{\cdot\} = 0^{(1)}$, or through the eigenvalue value, as $eig^{0}\{\cdot\} = 0^{(1)}$.

3.3 *Measurement* Graph

Definition 2 (Measurement Graph) *Let us consider the digraph ¥, such that ¥ $= (V(¥), E(¥))$. The vertices of ¥ are the same vertex set as V for graph ₣, while the edges, $E(¥) \subset V(¥) \times V(¥)$, represent the measurement links, where each order pair $(v_j, v_i) \in E(¥)$ if vertex v_i measures its relative distance to vertex v_j, ρ_j^i, i.e., there is a directed edge from vehicle j to vehicle i.*

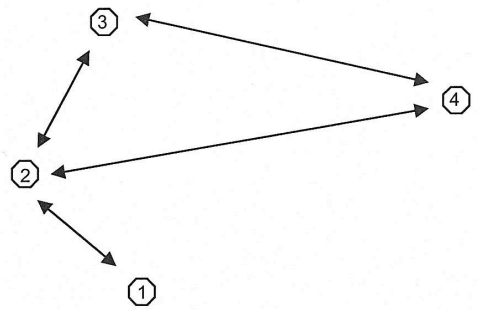

Figure 3-2: Example of a *measurement* graph ¥, defined as the relative measurements links between vehicles in a formation. According to the figure, the edges set is $E(¥) = \{(1,2),(2,1),(2,3),(2,4),(3,2),(3,4),(4,2),(4,3)\}$, and e.g., the edge $(2,3)$ is a directed edge from vehicle 2 to vehicle 3 indicates that vertex v_3 measures its relative distance to vertex v_2.

In this graph the role of the edges concerns whether relative measurements are available or not at each vehicle filter, *i.e.,* it takes into account that the sensor information can be restricted in the sense that there exists sensor information to which not all vehicles have access, as shown in Fig. 3-2. The key value of the *measurement* graph is the *indegree* (defined in Appendix A as the number of edges that come into a vertex) of each vertex, along with its meaning in terms of linear independent measurements.

The *measurement* graph, models the relative measurements between vehicles in the formation, *i.e.,* for each vehicle i, only the relative measurements between vehicle i and vehicle j are available for estimator i. Thus, the elements of adjacency matrix are, $t_{ij} = 1$ if the relative distance between vehicle filter j and i is sensed by vehicle filter i, the degree matrix diagonal being (for the directed graph) the indegree of each vertex, corresponding to the number of sensed measurements. The eigenvalues of $\mathbf{L}(¥)$ considering a fully-connected graph are,

$$eig\{\mathbf{L}(¥)\} = \left\{0^{(1)}, N^{(N-1)}\right\} \tag{3.3}$$

and the rank of the Laplacian matrix $\mathbf{L}(¥)$ will be $N-1$, indicating the maximal number of linearly independent columns, also indicated by the multiplicity of the non null eigenvalue. Graph ¥ is therefore $(N-1)$-regular, with indegree/outdegree $N-1$. Notice that the *measurement graph* does not have to be fully-connected. However the fully connected case represents the topology with maximum measurement links.

Definition 3 (*Measurement* subgraph) *From the measurement graph definition it is implicit that the arrows on each vehicle are the measurements obtained from each local sensor, and in this way are the only measurements available at each vehicle filter. In order to illustrate clearly this implicit property*

of the measurement graph we can decompose the measurement graph of a formation of N-vehicle into N measurement subgraphs, $¥^i$*, such that*

$$\sum_{i=1}^{N} ¥^i = ¥$$

This measurement subgraph corresponds to a spanning subgraph $¥^i = (V, E^i)$ *defined as* $E^i \subset E(¥)$*, with* $E^i = \{(i,j), i \neq j\}$ *for each vehicle* i*. Also, the sum of all the Laplacians for each vehicle* L_i *is the Laplacian of the measurement graph of the formation,*

$$\mathbf{L}(¥) = \sum_{i=1}^{N} \mathbf{L}_i$$

The eigenvalues of $\mathbf{L}\left(¥^i\right)$ *considering a fully-connected graph are*

$$eig\left\{\mathbf{L}\left(¥^i\right)\right\} = \left\{0^{(N-1)}, (N-1)^{(1)}\right\} \tag{3.4}$$

where the non-null eigenvalue indicates the number of independent state variables in the measurement subgraph $¥^i$*,* $eig^{(1)}\left\{\mathbf{L}\left(¥^i\right)\right\} = (N-1)^{(1)}$*.*

3.3.1 Sensor failure scenario

When sensor information is more restricted, such as in a sensor failure scenario (where the malfunction of a relative sensor is represented as an isolated vehicle), or a vehicle with partial malfunction in the sensors, or just vehicle occlusion during one of the measurements transmission, the vehicle can not measure the relative distance to other vehicles. For example, let us consider the formation shown in Fig. 3-3. Vehicle 6 is occluded and thus neither able to take any relative distance from the others nor able to be detected by the other vehicles; vehicle 3 cannot measure any relative distances to the others but its relative distance can be measured by the others (except vehicle 6); vehicle 1 can measure the relative distance to the other vehicles but cannot be detected by the others; vehicles 2, 4 and 5 have a partial malfunction in the measurement links, represented by occlusion of directions $5 \rightarrow 2$, $5 \rightarrow 3$ and $2 \rightarrow 4$. In the example, the *sensor failure* scenario is modeled by the graph ¥, such that $¥ = (V(¥), E(¥))$ describing the sensor restriction among vehicles. The set of edges of the *formation estimation* graph is $E(\mathrm{F}), = \{(2,1),(2,5),(3,1),(3,2),(3,4),(3,5),(4,1),(4,5),(4,2),(5,1),(5,4)\}$. The *measurement* graph Laplacian is,

$$\mathbf{L}(¥) = \mathbf{D}_g(¥) - \mathbf{T}_g(¥)$$

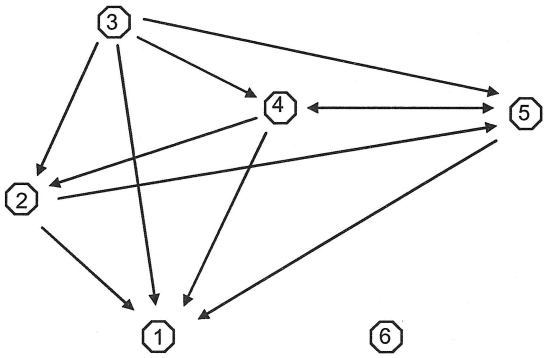

Figure 3-3: Measurement graph of 6 vehicles for a sensor failure scenario.

$$\mathbf{L}(¥) = \begin{bmatrix} 4 & -1 & -1 & -1 & -1 & 0 \\ 0 & 2 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & -1 & -1 & -1 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

where the eigenvalues of $\mathbf{L}(¥)$ are $0^{(2)}$, $1^{(1)}$ and $3^{(2)}$. The multiplicity of the zero eigenvalue means that there exist 2 connected components in the graph, the isolated vertex and the rest of the graph. Thus, the *measurement* graph is decomposed into two subgraphs, $¥' = ¥^6$ and $¥' = ¥ : ¥^1 \cup ¥^2 \cup ¥^3 \cup ¥^4 \cup ¥^5$.

The indegree of a vertex v_i, $\deg_{in}(v_i)$, defined in (A.2), is the number of edges that come in vertex v_i. Therefore, the number of measurements sensed by each vertex can be computed by the indegree of each vertex, shown in Table 3.1. Also, it can be inferred from the Laplacian eigenvalues of each of the *measurement* subgraphs,

$$\begin{aligned} eig\left\{\mathbf{L}\left(¥^1\right)\right\} &= \left\{0^{(4)}, 4^{(1)}\right\} \\ eig\left\{\mathbf{L}\left(¥^2\right)\right\} &= \left\{0^{(4)}, 2^{(1)}\right\} \\ eig\left\{\mathbf{L}\left(¥^3\right)\right\} &= \left\{0^{(4)}, 0^{(1)}\right\} \\ eig\left\{\mathbf{L}\left(¥^4\right)\right\} &= \left\{0^{(4)}, 2^{(1)}\right\} \\ eig\left\{\mathbf{L}\left(¥^5\right)\right\} &= \left\{0^{(4)}, 3^{(1)}\right\} \end{aligned}$$

Notice that vehicle 3, with zero indegree, is the one undominated vertex: it is a source vertex in

vehicle v_i	1	2	3	4	5
$deg_{in}(v_i)$	4	2	0	2	3

Table 3.1: The *indegree* of the vertices for the sensor failure scenario of Fig. 3-3.

the graph. This has a direct impact on the connectivity of the graph, which is not strongly connected, since the zero eigenvalue of its Laplacian is not simple. The path to vertex 3 is not possible, therefore vertex 3 does not contribute to the updating of the state vector estimates since it does not sense any measurements. On the other hand, vertex 1 is a sink vertex with a maximum number of measurements. Its indegree is 4, $\deg_{in}(v_1) = 4$, therefore it gives the highest contribution to the state vector estimation.

3.3.2 Observability

From the *measurement* graph defined previously, we want to provide an intuitive perspective of the observability of the full relative state vector at each vehicle i. Considering the example of N-FF s/c, with $N = 4$, where for sake of simplicity the relative state vector only includes the translational components, position in y, $\boldsymbol{\chi} = y$. Therefore, $\mathbf{x}^i_{trans} = \begin{bmatrix} (\boldsymbol{\chi}_{ij})^T & (\boldsymbol{\chi}_{jl})^T \end{bmatrix}^T$, and it is a vector of dimension n_x. $\boldsymbol{\chi}_{ij}$ represents the translational relative position state vector components between vehicle i and all the other $N-1$ vehicles in the formation, and $\boldsymbol{\chi}_{jl}$ the components between vehicles that do not include vehicle i. Also consider $\mathbf{z}^i(t)$ as the observation vector of dimension n_z, which contains all relative measurements sensed by vehicle i. The measurements sensed by all vehicles $\neq i$ are set to zero. Notice that the letter $\mathbf{z}$ was used, instead the letter $\mathbf{y}$, to define measurements vector, in order to have the measurements vector distinguished from the state vector component, y.

Taking into account $\mathbf{x}_{ij} = -\mathbf{x}_{ij}$, as stated in definition 1, with $i, j = 1, 2, 3, 4$, $i \neq j$ and applying the same to the components between vehicles that do not include vehicle i, *i.e.*, $\mathbf{x}_{jl} = -\mathbf{x}_{jl}$, with $j, l = 1, 2, 3, 4$, $j \neq l$. The state vector for the 4-FF s/c is $\mathbf{x}^1_{trans} = \begin{bmatrix} y_{12} & y_{13} & y_{14} & y_{23} & y_{24} & y_{34} \end{bmatrix}^T$, where $\begin{bmatrix} y_{12} & y_{13} & y_{14} \end{bmatrix}^T$ is the translational relative position state vector components between vehicle i, and all the other $N-1$ vehicles in the formation $\boldsymbol{\chi}_{ij}$, and $\begin{bmatrix} y_{23} & y_{24} & y_{34} \end{bmatrix}^T$ is the translational relative position state vector components between vehicles that do not include vehicle i, $\boldsymbol{\chi}_{jl}$. We use the Clohessy-Wiltshire equations (6.15a) to describe the relative dynamics among vehicles. Considering a circular orbit, the eccentricity is zero, $e_{cc} = 0$ and the Clohessy-Wiltshire equations become in state space representation,

$$\frac{d}{d\theta_t} \begin{bmatrix} y \\ y' \end{bmatrix}^i = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y \\ y' \end{bmatrix}^i$$

which applying to our state space dynamics, it follows,

$$\frac{d}{d\theta_t}\left[\begin{array}{c}\boldsymbol{\chi}_{ij}\\ \hline \boldsymbol{\chi}_{jl}\\ \hline \boldsymbol{\chi}'_{ij}\\ \boldsymbol{\chi}'_{jl}\end{array}\right]^i = \left[\begin{array}{cc|cc} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{1}_{3\times3} & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{1}_{3\times3}\\ \hline \mathbf{1}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times3} & \mathbf{1}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3}\end{array}\right]\left[\begin{array}{c}\boldsymbol{\chi}_{ij}\\ \hline \boldsymbol{\chi}_{jl}\\ \hline \boldsymbol{\chi}'_{ij}\\ \boldsymbol{\chi}'_{jl}\end{array}\right]^i$$

In this system the relative measurements received in vehicle i are those represented by the links in the *measurement* subgraph Υ^i, $E\left(\Upsilon^i\right) = z_{ij}$. So the measurement equation in this situation is[1],

$$\mathbf{z}^i = \left[\begin{array}{cc|cc} \mathbf{1}_{n_y\times3} & \mathbf{0}_{n_y\times3} & \mathbf{0}_{n_y\times3} & \mathbf{0}_{n_y\times3}\end{array}\right]\left(\left[\begin{array}{cc|cc} \boldsymbol{\chi}_{ij} & \boldsymbol{\chi}_{jl} & \boldsymbol{\chi}'_{ij} & \boldsymbol{\chi}'_{jl}\end{array}\right]^i\right)^T$$

where $\mathbf{z}$ is a vector of dimension n_z.

Assuming that the formation dynamic equations of the relative state representation for vehicle i, can be modeled by a linear time-invariant system as follows,

$$\begin{aligned}\frac{d}{d\theta_t}\mathbf{x}^i_{trans}\left(\theta_t\right) &= \mathbf{A}\mathbf{x}^i_{trans}\left(\theta_t\right)\\ \mathbf{z}^i(\theta_t) &= \mathbf{C}\mathbf{x}^i_{trans}(\theta_t)\end{aligned} \tag{3.5}$$

with no process or observation noise for sake of simplicity.

The observability matrix for the time-invariant system will be,

$$O = \left[\begin{array}{c}\mathbf{C}\\ \mathbf{CA}\\ \mathbf{CA}^2\\ \vdots\\ \mathbf{CA}^{n_x-1}\end{array}\right] = \left[\begin{array}{cc|cc} \mathbf{1}_{n_y\times3} & \mathbf{0}_{n_y\times3} & \mathbf{0}_{n_y\times3} & \mathbf{0}_{n_y\times3}\\ \mathbf{0}_{n_y\times3} & \mathbf{0}_{n_y\times3} & \mathbf{1}_{n_y\times3} & \mathbf{0}_{n_y\times3}\\ \mathbf{1}_{n_y\times3} & \mathbf{0}_{n_y\times3} & \mathbf{0}_{n_y\times3} & \mathbf{0}_{n_y\times3}\end{array}\right]$$

where $n_x \times n_x$ is the dynamic matrix dimension. Assuming that the dimension of the relative state vector components between vehicles that do not include vehicle i ($\boldsymbol{\chi}_{jl}$ and $\dot{\boldsymbol{\chi}}_{jl}$) is m, the observability matrix has rank $n_x - m$, meaning that the state components x_{ij} and $\dot{x}_{ij}$ are observable while the state components x_{jl} and $\dot{x}_{jl}$ are not observable. Therefore, the filter needs more information about the state to compute its full relative state vector estimates. This might be obtained by transmitting the measurements through the links represented in the *information flow* graph. However, if there is a sensor

[1] For notational simplicity the time dependence is suppressed.

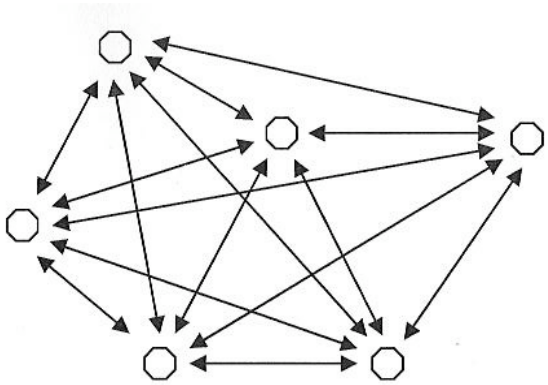

Figure 3-4: Fully connected *information flow* graph, where all vehicles share information with all the other vehicles in the formation.

failure or there is a *measurement* graph with fewer connected edges in each vertex, there will always be states that are not observable, as in the graph shown in Fig. 3-3. Therefore, the alternative is to obtain the needed information from other vehicle estimates. However this also raises the issue of communication failures which leads to the question of what is the minimum topology required in order for the full relative state to be observable. In the next section we will model the communication among vehicles of a formation by the *information flow* graph.

3.4 *Information Flow* Graph

Definition 4 (Information Flow Graph) *Let us consider the digraph* $\wp$, $\wp = (V(\wp), E(\wp))$, *known as the information flow graph, where the vertices of* $\wp$ *are a finite set of* N *vehicles, enumerated according to* $V(\wp) = \{v_i | i = 1, 2, ..., N\}$ *and the edge set* $E(\wp) \subset V(\wp) \times V(\wp)$ *represents the information flow among vehicles or vertices of the formation. Each ordered pair* $(v_i, v_j) \in E(\wp)$ *if the estimator in vehicle* i *transmits information to the estimator in vehicle* j.

For the *information flow* graph, two different topologies can be considered, one leading to the minimum number of communication links, the other to the maximum number of communication links. If all possible directed edges are allowed we have a $(N-1)$-regular directed graph, as shown in Fig. 3-4, with fully connected estimators. The digraph is strongly connected, the number of edges is maximum, and this is not a convenient solution for a formation architecture. In this work, we are seeking a decentralized architecture, with low communication and equal connectivity between vehicles, thus avoiding any of the vertices being masters.

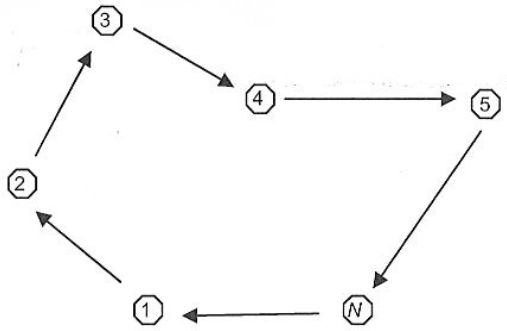

Figure 3-5: *Peer-to-peer* topology $\wp^P$, which is a directed cycle graph, with minimum number of edges connecting all vertices of the graph once.

3.4.1 Peer-to-peer topology

Due to the advantages of low communication for a N-vehicle formation, we are interested to ensure the smallest possible number of edges between the vehicles in the *information flow* graph while keeping it strongly connected. If we remove from the previously defined strongly connected digraph the maximum number of edges possible such that the vertices remain connected, the information goes through every vertex just once, and the same number of connections exist in each vehicle, we come up with a directed cycle graph, where all the edges are oriented in the same direction. In this case every vertex in the directed cycle graph obtained has indegree and outdegree 1, $deg_{in}(\nu_i) = deg_{out}(\nu_i) = 1$, therefore it is still regular and strongly connected. In fact, this can be confirmed by the Laplacian eigenvalues, which are $0^{(1)}$ and $1^{(N)}$ for a N-dimensional Laplacian matrix.

Because the ring graph is the underlying graph from the directed cycle graph, we will denote the latter as the peer-to-peer *information flow* graph, $\wp^p$, as shown in Fig. 3-5, where the edge set will be $E(\wp^p) = \{(i, i+1), ..., (N-1, N), (N, 1)\}$. If another vertex is removed, the *information flow* graph goes from decentralized to centralized topology. A centralized vertex means vulnerability to vertex failure, as for example in the rooted spanning tree shown in Fig. 3-6. Here the subgraph obtained is not cyclic and is not strongly connected, since not all vertices can reach all the other vertices; the indegree of vehicle 1 is zero, despite the fact that there is a zero eigenvalue in its Laplacian (eigenvalues are $1^{(N-1)}$and $0^{(1)}$).

We assume throughout the work that the defined graphs are all synchronized, meaning that each vertex represents the vehicle state at a given time, synchronized among vehicles[2], with measurements and information flow also synchronized.

[2]This is guaranteed by hardware devices, such as the Radiofrequency system [48].

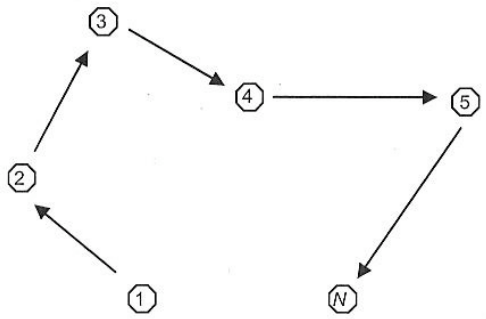

Figure 3-6: Rooted spanning tree.

3.5 Formation Navigation Problem

The formation navigation problem, tackled in this thesis, is to design a local estimator using the *formation estimation* graph Ŧ with *a measurement* graph ¥, and coupled vehicle nonlinear dynamics, $\dot{\mathbf{x}}_{ij}(t) = f(\mathbf{x}_{ij}(t), \mathbf{u}(t), t) + \breve{\mathbf{w}}(t)$ such that it computes the full relative formation state, $\mathbf{x}_{ij}(t)$ at any vehicle of a N-vehicle formation, minimizing the error estimation variance of the full relative state vector, based on an *information flow* graph, which uses minimal inter-vehicle communication, while overcoming the observability problem. The *formation estimation* graph is as defined in Section 3.2; the *measurement* graph is as described in Section 3.3; the *peer-to-peer* information flow graph is as discussed in Section 3.4 and the observability problems described in subsection 3.3.2. Then, within this framework, the requirements for estimation of the full 6DoF relative state vector of a N-vehicle formation need to be established, and tested in the sensor failure scenario, as described in subsection 3.3.

3.6 Summary

In this chapter we described the navigation problem of locally in each vehicle of a N-vehicle formation, estimating the full relative state vector based on the idea that the relative measurements usually available on each vehicle are restricted. Two directed graphs were used to model the local measurements and the information flowing through the formation: the *measurement* and the *information flow* graphs, respectively. In order to obtain a decentralized architecture for the navigation, an estimator is equally defined for each vehicle, bearing in mind the problem of having components of the state vector that are non-observable. The *information flow* graph is then used to supply to the vehicle filter the information needed to estimate the rest of the full relative state vector.

Chapter 4

SIX DEGREES OF FREEDOM DECENTRALIZED FORMATION STATE ESTIMATION

The graph-based formation navigation framework, introduced in the previous chapter, aims at estimating the 6DoF full relative formation state, in each vehicle, from noisy relative distance measurements, using filters which share information with each other. The filters should be

- decentralized, in the sense that there is not a single master vehicle in the formation,
- based on the measurements obtained locally at each vehicle (according to the *measurement* graph), and not on all the measurements,
- able to estimate the full relative state vector of the formation using minimal communication, *i.e.,* a peer-to-peer communication topology.

In this chapter we represent the algorithm for 6DoF decentralized formation state estimation, that solves the formation navigation problem established in the previous chapter for a full state vector. Thus, the algorithm is divided in two parts: one estimates the state vector components through the local measurements represented by the *measurement* subgraph and the other part estimates the locally non-observable state vector components through the state vector estimates transmitted from the previous vehicle, represented by the communication links in the *information flow* subgraph. Moreover, the 6DoF full state vector estimation is divided in two parts: one estimates the translational relative state vector of a N-vehicle formation and the other estimates the rotational absolute state vector of a N-vehicle

formation. Therefore, the full translational relative state vector defined in definition 1, is reduced to the relative position and translational relative velocity, while the attitude and angular velocity components will be defined as absolute components.

The full translational relative state vector of a N-vehicle formation, based on [3], uses:

a) EKF filtering, for updating the state vector estimates using local measurements;

b) the Covariance Intersection (CI) algorithm [49], [3], to avoid the correlation problem (see subsection 4.5.1), which arises when the local full relative state estimate is combined with the predecessor local full relative state estimate;

c) EKF prediction, for all prediction steps.

The EKF has been a standard method for nonlinear estimation problems for years, which (though being a suboptimal filter) is simple to implement, since it is derived directly from the state space model, and which also enjoys widespread popularity for its accuracy and speed. Since its applications are usually in nonlinear problems at discrete points in time, the nonlinear equations that describe the system need to be both linearized, such that the linear Kalman filter equations can be used at each operating point, and to be discretized in time.

The full rotational absolute state vector of a N-vehicle formation uses a) and c) as well, but replaces b) by:

b) the EKF algorithm (EKF Fusion), for the fusion of absolute state vector estimates, since the operations performed by the CI algorithm violate the quaternion operations, and the correlation problem is not an issue with absolute measurements. The EKF fusion is described in the context of the rotational part of the state vector, in subsection 4.7.1.

The absolute (rotational) part of the state is defined as such, because its components are usually based on absolute measurements, such as ST measurements in s/c, which are defined *w.r.t.* Earth. Nevertheless, when rotational relative state vector components are required due to formation functions, the relative state vector components can be built based on the absolute components of the state vector. Likewise, it is shown that the translational relative state vector components are straightforward to obtain from absolute translational state vector components, when they are available.

Obtaining relative state vector estimates is more demanding because relative state vector components are coupled due to the measurements, the dynamics and also the performance goals:

- coupled through measurements, because they express inter-vehicles measurements (*e.g.*, relative distance, relative orientation) and thus depend on state vector components from different vehicles;

- coupled through dynamics;
- coupled through performance goals, because the formation mission is specified in terms of the estimated relative quantities.

This chapter starts, in Section 4.1, with the definition of notation used to describe the 6DoF formation state vector which will be used throughout this work, as well as the formation equations of motion. The feasibility of determining the full state vector estimates in each vehicle is studied and analysis is performed according to each of the graphs defined, the *measurement* graph in subsection 4.2.2 and the *information flow* graph in subsection 4.2.3. This discussion leads to the design of the filter, in Section 4.3. It is proposed that the filter algorithm determines the translational relative state vector given the *measurement* graph and a peer-to-peer topology for the *information flow* graph. Analysis of the sensor failure scenario, also defined in the preceding chapter, is performed, which covers several possibilities for a measurement formation failure scenario. Section 4.4 outlines the Extended Kalman filter (filtering and prediction). The fusion of information from several vehicles is addressed in Section 4.5. The CI method, introduced in [50], is described, highlighting its application to correlated measurement problems. The algorithm to determine the full translational relative part of the state vector is introduced in Section 4.6. Also, the algorithm for the rotational absolute part is described in Section 4.7. The chapter closes with a summary that outlines the full state estimation algorithm and the achievements of the resulting architecture.

4.1 Preliminaries

This section concerns the 6DoF state vector notation with the description of its nonlinear translational and rotational dynamics, which are decoupled and thus handled separately.

4.1.1 Six degrees of freedom formation state vector

The full-order state vector is divided in two parts, as follows

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_{trans}^T & \mathbf{x}_{rot}^T \end{bmatrix}^T \tag{4.1}$$

where $\mathbf{x}_{trans}$ denotes the translational relative vector with relative position and translational relative velocity components and $\mathbf{x}_{rot}$ denotes the rotational absolute vector with absolute attitude and absolute angular velocity components.

Translational relative state vector

The translational relative vector has relative position and translational relative velocity components, as follows,

$$\mathbf{x}_{trans} = \begin{bmatrix} \boldsymbol{\chi}_{ij}^T & \dot{\boldsymbol{\chi}}_{ij}^T \end{bmatrix}^T \tag{4.2}$$

where the components express the relative relations among the vehicle, *e.g.*, $\boldsymbol{\chi}_{ij}$ is a three dimensional vector of relative positions, as shown in Fig. (4-1),

$$\boldsymbol{\chi}_{ij} = \boldsymbol{\chi}_j - \boldsymbol{\chi}_i = -\boldsymbol{\chi}_{ji}$$

where the subscript ij stands for relative relation between vehicle i and $j \in \{1, ..., N\} \setminus \{i\}$, with,

$$\boldsymbol{\chi}_{ij} = \begin{bmatrix} x_j - x_i & y_j - y_i & z_j - z_i \end{bmatrix}^T$$

with $\boldsymbol{\chi} = \begin{bmatrix} x & y & z \end{bmatrix}^T$ where x, y, z, are the axis of a Cartesian coordinate system, and $\dot{\boldsymbol{\chi}}_{ij}$ is a three dimensional vector of translational relative velocity,

$$\dot{\boldsymbol{\chi}}_{ij} = \dot{\boldsymbol{\chi}}_j - \dot{\boldsymbol{\chi}}_i = -\dot{\boldsymbol{\chi}}_{ji}$$

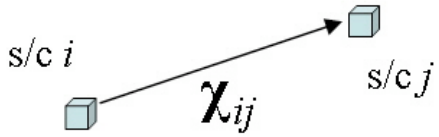

Figure 4-1: Representation of a relative vector where $\boldsymbol{\chi}_{ij} = \boldsymbol{\chi}_j - \boldsymbol{\chi}_i = -\boldsymbol{\chi}_{ji}$ and its absolute value is $\boldsymbol{\rho}_{ij} = \|\boldsymbol{\chi}_i - \boldsymbol{\chi}_j\| = \|\boldsymbol{\chi}_j - \boldsymbol{\chi}_i\|$.

Moreover, the relative distance is defined as follows,

$$\rho_{ij} = \|\boldsymbol{\chi}_i - \boldsymbol{\chi}_j\| = \|\boldsymbol{\chi}_j - \boldsymbol{\chi}_i\| \tag{4.3}$$

Considering the frames, the full translational relative part of the state vector for vehicle i *w.r.t.* vehicle j is defined as follows,

$$\mathbf{x}_{trans}^{loc} = \begin{bmatrix} \left(\boldsymbol{\chi}_{ij}^{loc}\right)^T & \left(\dot{\boldsymbol{\chi}}_{ij}^{loc}\right)^T \end{bmatrix}^T \tag{4.4}$$

where the state vector relates body frame b_i (abbreviated to i) *w.r.t.* body frame b_j (abbreviated to

j), written in a non-inertial frame denoted local frame, *loc*, used as the formation orbit reference frame (reference frames defined in detail in subsection 6.3.1), $\forall i, j$ with $j \neq i$. For example, if the local frame is assumed to be the body frame b_i of vehicle i, abbreviated as i, then the state vector is,

$$\mathbf{x}^i_{trans} = \begin{bmatrix} \left(\boldsymbol{\chi}^i_{ij}\right)^T & \left(\dot{\boldsymbol{\chi}}^i_{ij}\right)^T \end{bmatrix}^T$$

with $i, j = \{1, ..., N\}, i \neq j$, for each vehicle $i = 1, ..., N$.

Rotational absolute state vector

The rotational absolutepart of the state vector have its components are represented in the global frame for vehicle i *w.r.t.* a global world frame is defined as follows,

$$\mathbf{x}^{glob}_{rot} = \begin{bmatrix} \left(q^i_{glob}\right)^T & \left(\boldsymbol{\omega}^{glob}_i\right)^T \end{bmatrix}^T \tag{4.5}$$

where q is the attitude, expressed in quaternions, and represents the rotation between body frame *glob* and the body frame i, and ω is the three dimensional angular velocity vector. The angular velocity of body frame b_i (abbreviated to i), written in the glob frame, *glob*. Dropping the frames for simplicity, when it is clear the local frame under discussion, the rotational absolute part of the state vector can also be denoted as $\mathbf{x}_{rot}$.

The quaternion belongs to the special orthogonal Lie group of dimension $SO(3)$, which can be identified with rotations in space, defined as,

$$q = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^T = \begin{bmatrix} \mathbf{q} & q_4 \end{bmatrix} \tag{4.6}$$

or in terms of the Euler axis/angle detailed in subsection 6.3.1, by,

$$\begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix} = \begin{bmatrix} \overrightarrow{\mathbf{u}} \sin\left(\frac{\alpha}{2}\right) \\ \cos\left(\frac{\alpha}{2}\right) \end{bmatrix} \tag{4.7}$$

where α is the rotation angle and $\overrightarrow{\mathbf{u}} = (\overrightarrow{\mathbf{u}}_1, \overrightarrow{\mathbf{u}}_2, \overrightarrow{\mathbf{u}}_3)$ is the versor of the axis of rotation.

In relation to the quaternion state, the *unit constraint*, $q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1$ has to be ensured because it represents the orthogonality in the rotation. The attitude matrix $\mathbf{A}(q)$ or Direction Cosine

Matrix (DCM), is expressed in terms of the quaternion as follows,

$$\begin{aligned}\mathbf{A}(q) &= \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2\left(q_1 q_2 + q_3 q_4\right) & 2\left(q_1 q_3 - q_2 q_4\right) \\ 2\left(q_1 q_2 - q_3 q_4\right) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2\left(q_2 q_3 + q_1 q_4\right) \\ 2\left(q_1 q_3 + q_2 q_4\right) & 2\left(q_2 q_3 - q_1 q_4\right) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \\ \mathbf{A}(q) &= \left(q_4^2 - \|q\|^2\right)\mathbf{1}_{3\times 3} + 2qq^T - 2q_4\left[q\times\right]\end{aligned} \tag{4.8}$$

where $\mathbf{1}_{3\times 3}$ is the identity matrix with dimension 3×3. The operation $\otimes$, denotes the quaternion multiplication,

$$q \otimes p = \mathbf{\Xi}(p)q = \mathbf{\Lambda}(q)p \tag{4.9}$$

with p and q being quaternions,

$$\mathbf{\Xi}(p) = \begin{bmatrix} p_4\mathbf{1}_{3\times 3} + [\mathbf{p}\times] & \mathbf{p} \\ -\mathbf{p}^T & p_4 \end{bmatrix} \tag{4.10}$$

$$\mathbf{\Lambda}(q) = \begin{bmatrix} q_4\mathbf{1}_{3\times 3} - [\mathbf{q}\times] & \mathbf{q} \\ -\mathbf{q}^T & q_4 \end{bmatrix} \tag{4.11}$$

where the skew-symmetric matrix $[\mathbf{p}\times]$, defined by

$$[\mathbf{p}\times] = \begin{bmatrix} 0 & -p_z & p_y \\ p_z & 0 & -p_x \\ -p_y & p_x & 0 \end{bmatrix} \tag{4.12}$$

represents the cross product of two vectors, $\mathbf{a}$ and $\mathbf{c}$,

$$[\mathbf{a}\times]\,\mathbf{c} = -\,[\mathbf{c}\times]\,\mathbf{a} \tag{4.13}$$

Notice that the attitude matrix $\mathbf{A}(q_j^i)$ describes the rotation between body frame j and body frame i, while the quaternion represents the rotation between body frame i and the body frame j, and can also be written as,

$$\mathbf{A}(q_{b_j}^{b_i}) = \mathbf{A}_{b_j}^{b_i}(q) \tag{4.14}$$

Also, an attitude matrix that it is expressed in terms of other quantities rather then quaternions is denoted by rotation matrix, $\mathbf{A}_j^i$. Moreover, using (4.8) the quaternion can be obtained from a rotation matrix.

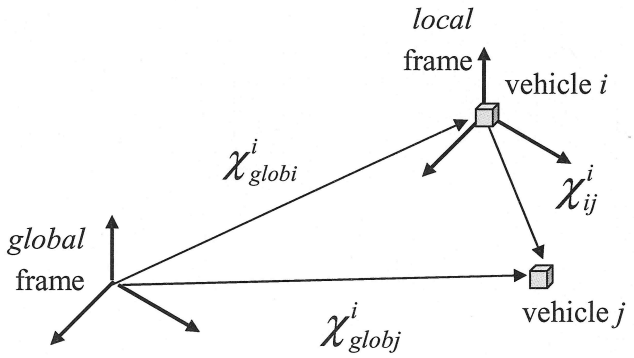

Figure 4-2: Translational variable relationship between relative state vector and absolute state vector.

Global versus local frame

The global frame is related to the local frame through the attitude matrix (4.8) that relates both frames $\mathbf{A}\left(q^{loc}_{glob}\right)$, as follows

$$\mathbf{x}^{loc} = \mathbf{A}\left(q^{loc}_{glob}\right)\mathbf{x}^{glob} \tag{4.15}$$

Absolute versus relative

The absolute state vector translational components, now written in a local frame, the body frame i, $\boldsymbol{\chi}^i_{globj}$ and $\boldsymbol{\chi}^i_{globi}$, can be related to the relative components by,

$$\boldsymbol{\chi}^i_{ij} = \boldsymbol{\chi}^i_{globj} - \boldsymbol{\chi}^i_{globi}$$

Since $-\boldsymbol{\chi}^i_{globi} = \boldsymbol{\chi}^i_{iglob}$, then,

$$\boldsymbol{\chi}^i_{ij} = \boldsymbol{\chi}^i_{globj} + \boldsymbol{\chi}^i_{iglob}$$

as shown in Fig 4-2. The same can be applied to the translational velocity. The local frame is related to the global frame through the rotation (4.15).

The quaternion variables, which represent the orientation of vehicle i axis in the inertial frame, can be used to parametrize the attitude matrix which transforms the vector representation from absolute to the body frame. Thus, it is possible to represent the relative attitude matrix between vehicle body frames as,

$$\mathbf{A}\left(q^i_j\right) = \mathbf{A}\left(q^i_{glob}\right)\mathbf{A}\left(q^{glob}_j\right)$$

$$\mathbf{A}\left(q^i_j\right) = \mathbf{A}\left(q^i_{glob}\right)\mathbf{A}^T\left(q^j_{glob}\right) \tag{4.16}$$

with $i, j = 1, ..., N$, $i \neq j$. The related quaternion can be obtained through Eq. (4.8). Thus, a set of four solutions can be determined, so that if for the first set q_4 is the first to be found, q_1 is the second to be found, q_2 is the third to be found and q_3 is the last. Numerical inaccuracies may arise when q_4 is very small. One may overcome this by computing the maximum of $1 + a_{11} + a_{22} + a_{33}$, $1 + a_{11} + a_{22} + a_{33}$ and $1 + a_{11} + a_{22} + a_{33}$, and based on this, switch among solutions, as suggested in [51]. Another possible path consists of computing directly the quaternion that represents the orientation between vehicles,

$$\begin{aligned} q_j^i &= q_{glob}^i \otimes q_j^{glob} \\ &= q_{abs}^i \otimes \left(q_{glob}^j\right)^T \\ &= q_{glob}^i \otimes \begin{bmatrix} -\mathbf{q}_{glob}^j \\ q_{glob}^j \end{bmatrix} \end{aligned}$$

where $\otimes$ is the quaternion multiplication. The attitude matrix can then be computed, if needed, using the equation (4.8). This way, the process of obtaining the quaternion from the attitude matrix is overcome, avoiding the problem of having four possible solutions, described before.

4.1.2 Formation dynamics

The formation equations of motion, are presented as the dynamics of any vehicle of a N-vehicle formation, and can be written in a compact form,

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t), t) + \breve{\mathbf{w}}(t) \tag{4.17}$$

where $f(\cdot)$ is the continuous time process model, a nonlinear function of the state $\mathbf{x}(t)$, $\mathbf{x} \in \mathbb{R}^{n_x}$, where n_x is the dimension of state vector, *i.e.* the number of state variables. $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ is the control input vector, where n_u is the dimension of the control input vector, *i.e.* the number of vehicle actuators.

Since it was assumed that the translational and rotational motions are decoupled we can split the dynamics (4.17) into translational relative dynamics and rotational absolute dynamics, dropping the frames for the sake of simplicity, follows,

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{\mathbf{x}}_{trans}(t) \\ \dot{\mathbf{x}}_{rot}(t) \end{bmatrix} = \begin{bmatrix} f(\mathbf{x}_{trans}(t), \mathbf{u}(t), t) \\ f(\mathbf{x}_{rot}(t), \mathbf{u}(t), t) \end{bmatrix} + \begin{bmatrix} \breve{\mathbf{w}}_{trans}(t) \\ \breve{\mathbf{w}}_{rot}(t) \end{bmatrix}$$

and described in the next two subsections.

Formation relative dynamics

The state dynamics of any vehicle as described before, includes the translational relative dynamics (*i.e.*, relative position and relative velocity) of the i^{th} relative j^{th} vehicle in the formation, $i \neq j$,

$$\dot{\mathbf{x}}_{trans}\left(t\right) = f(\mathbf{x}_{trans}\left(t\right), \mathbf{u}\left(t\right), t) + \breve{\mathbf{w}}_{trans}(t)$$

where n_x is number of state variables, which in this case is the relative positions and velocities of all vehicles *w.r.t.* another vehicle in the formation.

The output is given by,

$$\mathbf{y}^i_{trans}(k) = h^i(\mathbf{x}_{trans}(k), k) + \boldsymbol{\nu}^i(k) \tag{4.18}$$

where $\mathbf{y}^i_{trans}(t)$ is the $n_y \times 1-$dimensional observation vector of vehicle i at time t, $\mathbf{y}^i_{trans}(t)$ contains all relative measurements sensed by vehicle i, and where $\boldsymbol{\nu}^i(k)$ is observation noise, assumed to be white Gaussian with,

$$\begin{aligned} E\left\{\boldsymbol{\nu}(k)\boldsymbol{\nu}^T(k)\right\} &= \mathbf{R}_{trans}(k) \\ E\left\{\boldsymbol{\nu}(k)\right\} &= 0 \end{aligned}$$

Moreover, we assume that the process noise, $\breve{\mathbf{w}}(t)$, is independent of the observation noise, $\boldsymbol{\nu}(k)$, and is also assumed, for all t and k, to be white Gaussian with,

$$\begin{aligned} E\left\{\breve{\mathbf{w}}_{trans}(t)\breve{\mathbf{w}}^T_{trans}(\tau)\right\} &= \breve{\mathbf{Q}}_{trans}(t)\delta(t-\tau) \\ E\left\{\breve{\mathbf{w}}_{trans}(t)\right\} &= 0 \end{aligned}$$

Throughout this work, for sake of notation simplicity, when it is clear the part of the full state vector under discussion, we will suppress the subscript *trans* from the variables.

Formation absolute dynamics

The state dynamics of any vehicle as described before, includes the rotational absolute dynamics of any vehicle in the formation *w.r.t.* a global frame,

$$\dot{\mathbf{x}}_{rot}\left(t\right) = f(\mathbf{x}_{rot}\left(t\right), \mathbf{u}\left(t\right), t) + \breve{\mathbf{w}}_{rot}(t)$$

where n_x is number of state variables, which in this case is the rotational absolute and angular velocity coordinates of all vehicles in the formation in a given global frame.

The output is given by,

$$\mathbf{y}^i_{rot}(k) = \mathbf{H}^i(\mathbf{x}_{rot}(k), k) + \mathbf{v}^i(k) \tag{4.19}$$

where $\mathbf{y}^i_{rot}(t)$ is the $n_y \times 1-$dimensional observation vector of vehicle i at time t, $\mathbf{y}^i_{rot}(t)$ contains all absolute measurements sensed by vehicle i, and where $\mathbf{v}^i(k)$ is observation noise, assumed to be white Gaussian with,

$$\begin{aligned} E\left\{\mathbf{v}(k)\mathbf{v}^T(k)\right\} &= \mathbf{R}_{rot}(k) \\ E\left\{\mathbf{v}(k)\right\} &= 0 \end{aligned}$$

Moreover, we assume that the process noise, $\breve{\mathbf{w}}_{rot}(t)$, is independent of the observation noise, $\mathbf{v}(k)$, and is also assumed, for all t and k, to be white Gaussian with,

$$\begin{aligned} E\left\{\breve{\mathbf{w}}_{rot}(t)\breve{\mathbf{w}}^T_{rot}(\tau)\right\} &= \breve{\mathbf{Q}}_{rot}(t)\delta(t-\tau) \\ E\left\{\breve{\mathbf{w}}_{rot}(t)\right\} &= 0 \end{aligned}$$

Throughout this work, for sake of notation simplicity, when it is clear the part of the full state vector under discussion, we will suppress the subscript *rot* from the variables.

4.2 Formation Navigation Analysis

After defining, in the preceding chapter, the formation navigation problem based on graph theory, several questions arise at this stage:

- What kind of information must be communicated through the *information flow* graph in order to have the 6DoF full state vector of a N-vehicle formation locally estimated?
- Is it achievable a *peer-to-peer* communication topology where just peer-to-peer information flow is allowed?
- How many *measurement* subgraphs are needed to fully estimate the state vector?
- What are the conditions that the *measurement* graph and the *information flow* graph have to satisfy in order to guarantee that the 6DoF full state vector of a N-vehicle formation is locally estimated?

4.2.1 *Formation estimation* graph

As we discussed in the last chapter, a N-vehicle formation, each with a local estimator, aims to estimate the full relative state vector components of the formation, represented by the *formation estimation* graph $\mathfrak{F}$, $\mathfrak{F} = (V(\mathfrak{F}), E(\mathfrak{F}))$. The estimation filter at each vehicle is associated with the vertices of the graph, enumerated according to $V(\mathfrak{F}) = \{v_i | i = 1, 2, ..., N\}$. The full formation relative state vector dimension is the size of the *formation estimation* graph, $|E(\mathfrak{F})|$: $N(N-1)/2$, for each component subset: relative position, translational velocity, attitude and angular velocity.

Formation estimation graph example for 4-vehicle formation

Let us consider a formation of 4 vehicles, each with an estimator, where $|V(\mathfrak{F})| = 4$, and the edge set represents the components of the full relative state vector of the formation to be determined. To facilitate the analysis, lets us consider only the relative positions of the translational part of the relative state vector of the formation for s/c 1 (superscript 1), $\hat{\mathbf{x}}^1_{trans} \in \mathbb{R}^{|E(\mathfrak{F})|}$, with $\mathbf{x}^1_{trans}$ as defined in subsection 4.1.1,

$$\hat{\mathbf{x}}^1_{trans} = \begin{bmatrix} (\hat{\boldsymbol{\chi}}_{12})^T & (\hat{\boldsymbol{\chi}}_{13})^T & (\hat{\boldsymbol{\chi}}_{14})^T & (\hat{\boldsymbol{\chi}}_{23})^T & (\hat{\boldsymbol{\chi}}_{24})^T & (\hat{\boldsymbol{\chi}}_{34})^T \end{bmatrix}^T \tag{4.20}$$

The size of the graph is $|E(\mathfrak{F})| = 6$, but each component has coordinates in 3D, *i.e.,* each $\hat{\boldsymbol{\chi}}_{ij} \in \mathbb{R}^3$. Thus the set of edges is defined as $E(\mathfrak{F}) = \{\hat{\boldsymbol{\chi}}_{12}, \hat{\boldsymbol{\chi}}_{13}, \hat{\boldsymbol{\chi}}_{14}, \hat{\boldsymbol{\chi}}_{23}, \hat{\boldsymbol{\chi}}_{24}, \hat{\boldsymbol{\chi}}_{34}\}$.

4.2.2 *Measurement* graph

Notice, that the *measurement* graph does not have to be fully-connected, but even for a fully connected *measurement* graph, there are state vector components that are not measured on each s/c. A study of which relative quantities are not measured locally, and the impact of this in the required information flow, follows.

Let us consider the fully connected *measurement* subgraph $\mathfrak{Y}^i$ with Laplacian eigenvalues given by Eq. (3.4), $eig\left\{\mathbf{L}\left(\mathfrak{Y}^i\right)\right\} = \left\{0^{(N-1)}, (N-1)^1\right\}$. The multiplicity one Laplacian eigenvalue indicates the number of independent variables from the *measurement* subgraph that can be determined by the vehicle filter whose state vector components are modeled by the *formation estimation* graph. That is, by intersecting the *formation estimation* graph $\mathfrak{F}$ and the underlying *measurement* subgraph $\left(\mathfrak{Y}^i\right)^U$, as shown in Fig. 4-3, we obtain the spanning graph (subgraph) of the *formation estimation* graph on each vehicle,

$$\left(\mathfrak{Y}^i\right)^U \cap \mathfrak{F} = \left(\mathfrak{F}^i\right)', i = 1, ..., N$$

which represents the number of independent state variables that can be determined by the *measurement*

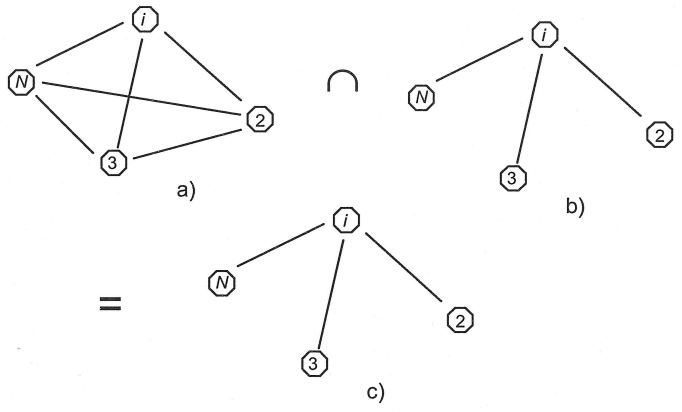

Figure 4-3: Intersection of a) the *formation estimation* graph $\mathbb{F}$, and b) the underlying *measurement* subgraph in vehicle i, $\left(¥^i\right)^U$, results in c) the observable formation estimated components of the *formation estimation* graph on vehicle i.

subgraph on each vehicle (the observable components, $\hat{\mathbf{x}}_o$ where subscript o stands for observable).

***Measurement* graph example for 4-vehicle formation**

Let us assume for now that the *measurement* graph ¥ for a 4–vehicle formation is fully-connected and it can be decomposed into four 3– regular subgraphs $¥^i = \left(V\left(¥\right), E^i\left(¥^i\right)\right)$, $¥ = ¥^1 \cup ¥^2 \cup ¥^3 \cup ¥^4$, where each edge subset $E^i\left(¥^i\right) = \{\rho^i_j, i \neq j, i = 1, ..., 4\}$, as shown in Fig. 4-4. An example of determining ρ^i_j is given in (6.20) by using Radio Frequency (RF) measurements. Thus, $E^1\left(¥^1\right) = \{\rho^1_2, \rho^1_3, \rho^1_4\}$, $E^2\left(¥^2\right) = \{\rho^2_1, \rho^2_3, \rho^2_4\}$, $E^3\left(¥^3\right) = \{\rho^3_1, \rho^3_2, \rho^3_4\}$, $E^4\left(¥^4\right) = \{\rho^4_1, \rho^4_2, \rho^4_3\}$. Knowing that the measurements are a function of the state vector components,

$$\rho^i_j = f(\hat{\boldsymbol{\chi}}_{ij}) \text{ or } \rho^j_i = f(-\hat{\boldsymbol{\chi}}_{ij}) \tag{4.21}$$

it is convenient for the edges in each subgraph $¥^i$ to have their elements expressed in terms of the measurements that directly represent the components of the state vector. Thus,

$$y^1 = \{f(\hat{\boldsymbol{\chi}}_{12}), f(\hat{\boldsymbol{\chi}}_{13}), f(\hat{\boldsymbol{\chi}}_{14})\} \tag{4.22}$$

$$y^2 = \{f(-\hat{\boldsymbol{\chi}}_{12}), f(\hat{\boldsymbol{\chi}}_{23}), f(\hat{\boldsymbol{\chi}}_{24})\}$$

$$y^3 = \{f(-\hat{\boldsymbol{\chi}}_{13}), f(-\hat{\boldsymbol{\chi}}_{23}), f(\hat{\boldsymbol{\chi}}_{34})\}$$

$$y^4 = \{f(-\hat{\boldsymbol{\chi}}_{14}), f(-\hat{\boldsymbol{\chi}}_{24}), f(-\hat{\boldsymbol{\chi}}_{34})\} \tag{4.23}$$

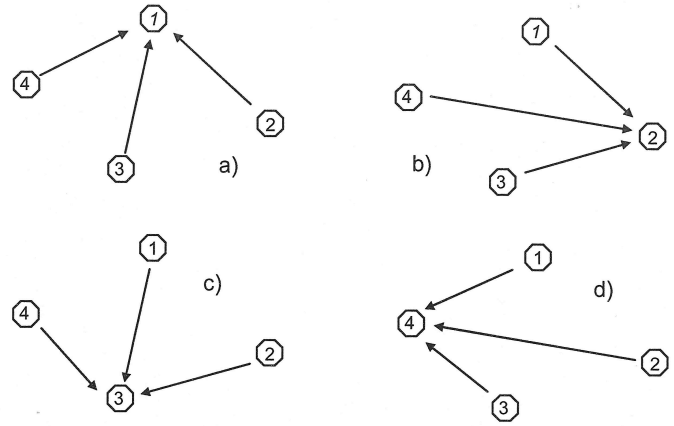

Figure 4-4: For a formation of 4 vehicles, the *measurement* graph ¥ is decomposed into the four subsets $¥^i = \left(V, E^i\right)$, $¥ = ¥^1 \cup ¥^2 \cup ¥^3 \cup ¥^4$, where a) $¥^1$ b) $¥^2$c) $¥^3$ d) $¥^4$, each 3- regular.

It is clear that for each *measurement* subgraph, $¥^i$ the number of independent measurements, *i.e.,* 3, is equal to the number of the state vector components estimated locally (observable components, $\hat{\mathbf{x}}_o$ where subscript o stands for observable). For vehicle filter 1, the observable components are,

$$\hat{\mathbf{x}}^1_{trans,o} = \begin{bmatrix} \hat{\boldsymbol{\chi}}_{12} & \hat{\boldsymbol{\chi}}_{13} & \hat{\boldsymbol{\chi}}_{14} \end{bmatrix}^T \tag{4.24}$$

and the remainder, *i.e.,* $\hat{\boldsymbol{\chi}}_{23}$, $\hat{\boldsymbol{\chi}}_{24}$, $\hat{\boldsymbol{\chi}}_{34}$, are non-observable. This can be inferred from the multiplicity one Laplacian eigenvalue (3.4) for the formation of 4 vehicles,

$$eig\left\{\mathbf{L}\left(¥^i\right)\right\} = \left\{0^{(3)}, 3^{(1)}\right\}$$

From this example, one might think that if the number of independent measurements per *measurement* subgraph is equal to its size, $|E(¥^i)| = N-1$, then we would expect that by adding N *measurement* subgraphs, there would be $N\left(N-1\right)$ components of the state vector that could be determined. In fact this is not so, because there is one repeated measurement every time a *measurement* subgraph is added. This will decrease by one the number of components determined in the state vector.

Measurement subgraph example for 4-vehicle formation

The state vector to be determined by the estimator from vehicle 1 is given by Eq. (4.20) and the edges in *measurement* subgraph 1 are given by Eq. (4.22). Considering now vehicle 4, in the same formation, the edges in *measurement* subgraph 4 are given by Eq. (4.23). Comparing both *measurement* subgraphs, it

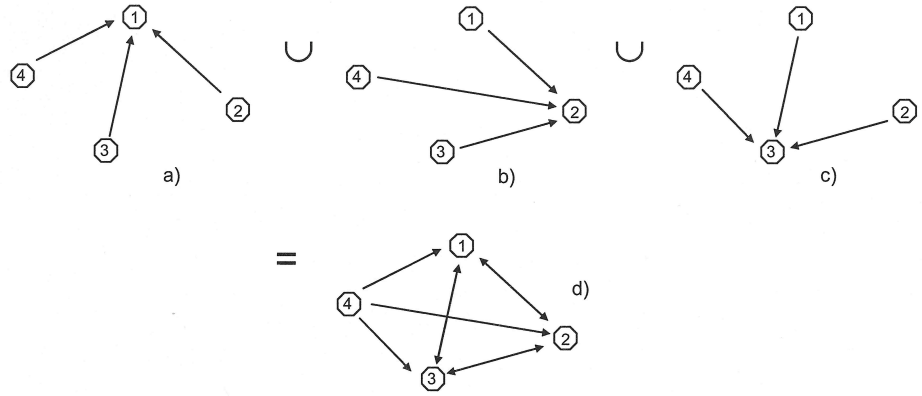

Figure 4-5: The *measurement* subgraphs needed to determine the full state vector in vehicle 1, for a formation of $N = 4$ vehicles, with $i \in \{1, ..., N\}$. The union of the *measurement* subgraphs a)$\mathbb{Y}^i, i = 1$, b) $\mathbb{Y}^{i+1}, i+1 = 2$, and c) $\mathbb{Y}^{i+2}$, $i+2 = 3$ in vehicle filter 1 is d) $\mathbb{Y}^i \cup \mathbb{Y}^{i+1} \cup \mathbb{Y}^{i+2}$. Notice, that the union of *measurement* subgraphs is coincident with the *formation estimation* graph $\mathbb{F}$.

is clear that $\hat{\boldsymbol{\chi}}_{14}$ is repeated in both, thus if vehicle 1 can use both *measurement* subgraphs ($E\left(\mathbb{Y}^1\right)$ and $E\left(\mathbb{Y}^4\right)$) to determine its state vector components, the *measurement* subgraph 1 will determine $N-1 = 3$ components of the state vector from vehicle 1, and the *measurement* subgraph 4 will determine $N-2 = 2$ of the components, leaving the remaining state vector component $\hat{\boldsymbol{\chi}}_{23}$ to be determined. This component remains non-observable unless the *measurement* subgraph 3 or 2 is added to the estimator on vehicle 1. Thus, $N-1 = 3$ state vector components are determined by the local *measurement* subgraph $\mathbb{Y}^i$, then $N-2 = 2$ components are determined by the first external *measurement* subgraph $\mathbb{Y}^{i+1}$, and $N-3 = 1$ by the second $\mathbb{Y}^{i+2}$, as shown in Fig. 4-5. The multiplicity one Laplacian eigenvalue of the *measurement* subgraphs is $eig^{(1)}\left\{\mathbf{L}\left(\mathbb{Y}^i\right)\right\} = 3^{(1)}$. The multiplicity one eigenvalue of the union of three *measurement* subgraphs, $eig^{(1)}\left\{\mathbf{L}\left(\mathbb{Y}^i \cup \mathbb{Y}^{i+1} \cup \mathbb{Y}^{i+2}\right)\right\} = 1^{(1)}$. This which also has a null eigenvalue with multiplicity one, $eig^{0}\left\{\mathbf{L}\left(\mathbb{Y}^i \cup \mathbb{Y}^{i+1} \cup \mathbb{Y}^{i+2}\right)\right\} = 0^{(1)}$.

Notice that the sum (or union) of the underlying *measurement* subgraphs $\left(\mathbb{Y}^i\right)^U$ up to $N-1$, is equivalent to the *formation estimation* digraph $\mathbb{F}$,

$$\sum_{i \in V | l=1}^{l=N-1} \left(\mathbb{Y}^{i+l}\right)^U = \mathbb{F}$$

That is, there are $N-1$ components of the state vector that can be determined by the local *measurement* subgraph, $E\left(\mathbb{Y}^i\right)$, as expressed by the multiplicity one Laplacian eigenvalue. The remaining non-observable components have to be determined adding $N-2$ *measurement* subgraphs. This is

confirmed by the multiplicity one Laplacian eigenvalue ($eig^{(1)}\{\mathbf{L}(¥^i)\}$) and the null eigenvalue with multiplicity one ($eig^0\{\mathbf{L}(¥^i)\}$), of the union of $N-1$ *measurement* subgraphs,

$$\begin{aligned}
eig^{(1)}\left(\mathbf{L}\left(¥^i\right)\right) &= (N-1)^{(1)}, eig^0\left(¥^i\right) = 0^{(N-1)} \\
eig^{(1)}\left(\mathbf{L}\left(¥^i \cup ¥^{i+1}\right)\right) &= (N-2)^{(1)}, eig^0\left(¥^i\right) = 0^{(N-2)} \\
eig^{(1)}\left(\mathbf{L}\left(¥^i \cup ¥^{i+1} \cup ¥^{i+2}\right)\right) &= (N-3)^{(1)}, eig^0\left(¥^i\right) = 0^{(N-3)} \\
eig^{(1)}\left(\mathbf{L}\left(¥^i \cup ¥^{i+1} \cup ¥^{i+2} \cup ¥^{i+3}\right)\right) &= (N-4)^{(1)}, eig^0\left(¥^i\right) = 0^{(N-4)} \\
&\vdots \\
eig^{(1)}\mathbf{L}\left(\left(\sum_{i\in V|l=1}^{l=N-1} ¥^{i+l}\right)\right) &= 1^{(1)}, eig^0\left(\mathbf{L}\left(¥^i\right)\right) = 0^{(1)}
\end{aligned} \tag{4.25}$$

A generalization for an N-vehicle formation is summarized in Table 4.1. The *measurement* subgraph analysis is summarized in Table 4.2.

Formation of N-vehicle	$N=3$	$N=4$	$N=5$	$N=6$	$N=7$	$\cdots$
$E\left(¥^i\right)$	$n^o\left(\boldsymbol{\chi}_o\right)=2$	$n^o\left(\boldsymbol{\chi}_o\right)=3$	$n^o\left(\boldsymbol{\chi}_o\right)=4$	$n^o\left(\boldsymbol{\chi}_o\right)=5$	$n^o\left(\boldsymbol{\chi}_o\right)=6$	$\cdots$
$E\left(¥^i\right)+E\left(¥^{i+1}\right)$	$n^o\left(\boldsymbol{\chi}_o\right)=1$	$n^o\left(\boldsymbol{\chi}_o\right)=2$	$n^o\left(\boldsymbol{\chi}_o\right)=3$	$n^o\left(\boldsymbol{\chi}_o\right)=4$	$n^o\left(\boldsymbol{\chi}_o\right)=5$	$\cdots$
$E\left(¥^i\right)+...+E\left(¥^{i+2}\right)$	-	$n^o\left(\boldsymbol{\chi}_o\right)=1$	$n^o\left(\boldsymbol{\chi}_o\right)=2$	$n^o\left(\boldsymbol{\chi}_o\right)=3$	$n^o\left(\boldsymbol{\chi}_o\right)=4$	$\cdots$
$E\left(¥^i\right)+...+E\left(¥^{i+3}\right)$	-	-	$n^o\left(\boldsymbol{\chi}_o\right)=1$	$n^o\left(\boldsymbol{\chi}_o\right)=2$	$n^o\left(\boldsymbol{\chi}_o\right)=3$	$\cdots$
$E\left(¥^i\right)+...+E\left(¥^{i+4}\right)$	-	-	-	$n^o\left(\boldsymbol{\chi}_o\right)=1$	$n^o\left(\boldsymbol{\chi}_o\right)=2$	$\cdots$
$E\left(¥^i\right)+...+E\left(¥^{i+N-2}\right)$	-	-	-	-	$n^o\left(\boldsymbol{\chi}_o\right)=1$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Table 4.1: For a formation of N-vehicle, there are $N-1$ state vector components determined by the local (vehicle filter i) *measurement* subgraph $E(¥^i)$ ($n^o\left(\boldsymbol{\chi}_o\right)$ from $E(¥^i)$) and $N-2$ components that are determined by the first external *measurement* subgraph added to the vehicle filter i (number of observable state vector components, $n^o\left(\boldsymbol{\chi}_o\right)$ from $E\left(¥^{i+1}\right)$) , $N-3$ components by the second external measurement subgraph added to the vehicle filter i ($n^o\left(\boldsymbol{\chi}_o\right)$ from $E\left(¥^{i+2}\right)$), ... etc.

The vehicle filter i determines $N-1$ observable state vector components through the local *measurement* subgraph $¥^i$, leaving $\frac{N^2-3N+2}{2}$ non-observable components to be determined. Adding an external (to vehicle i) *measurement* subgraph, $¥^{i+1}$, determines $N-2$ observable state vector components; if another $¥^{i+2}$, is added this determines $N-3$ observable state vector components, and so on until just one non-observable component is determined by the last external *measurement* subgraph. This one non-observable component indicates that all the remain components are already observed and it is determined by adding $N-2$ *measurement* subgraph to the local *measurement* subgraph, as shown in Table 4.1.

vertices	N	3	4	5	6	7	$\cdots$
$n^o(\boldsymbol{\chi}_o)$ from $E(¥^i)$	$N-1$	2	3	4	5	6	$\cdots$
$\mathbf{x}$ dimension, $\lvert E(₣)\rvert$	$\frac{N(N-1)}{2}$	3	6	10	15	21	$\cdots$
$n^o(\boldsymbol{\chi}_{\bar{o}})$ on vehicle i	$\frac{N^2-3N+2}{2}$	1	3	6	10	15	$\cdots$
$n^o\left(E(¥^{i+1}), E(¥^{i+2}), ...\right)$	$N-2$	1	2	3	4	5	$\cdots$

Table 4.2: Summary of the *measurement* subgraph analysis, as a function of the number of vehicles in the formation, namely: the number of state vector components that are determined through the local *measurement* subgraph $E(¥^i)$; the dimension of the state vector represented by the links in its *formation estimation* graph; the number non-observable state vector components, $n^o(\boldsymbol{\chi}_{\bar{o}})$ (subscript $\bar{o}$ stands for non-observable); the number of external *measurement* subgraphs required to determine completely the state vector.

Therefore, the number of communication links needed to determine the full relative state estimate in each vehicle of a N-vehicle formation is $N-2$. So, we will analyze how to achieve this in the peer-to-peer *information flow* graph with only one communication link per vehicle, as well as which information needs to be transmitted.

4.2.3 *Information flow* graph

As discussed before, in order to obtain the full state vector estimates we need to have $N-2$ external *measurement* subgraphs in each vehicle, leading to two possible scenarios: either each vehicle receives the *measurement* subgraph from the other $N-2$ vehicles, or it uses the peer-to-peer *information flow* graph (as defined in subsection 3.4.1) and analyzes whether it is feasible to estimate the full state vector with just one communication link per vehicle.

Let us consider the most restricted scenario (with communication reduced to a minimum), the peer-to-peer communication topology for the *information flow* graph. In the alternative fully connected *information flow* graph, it would clearly be feasible to determine the full state vector estimates when all measurements are available for each vehicle filter, but not of interest from the reduction of communication perspective.

In this work, the state vector estimates transmitted by other vehicles are viewed by the receiving vehicle as "measurements" and fused with the local state vector estimates. Both observable and non-observable components are used, and combined in such a way that the estimates computed by the local *measurement* graph (observable components) have a larger weight than the ones that are just propagated through the equations of dynamics (non-observable components). However, a correlation problem arises when the local full state estimate is combined with the predecessor local full state estimates, as will be explained in detail in Section 4.5. To overcome the correlation problem, the CI method, introduced in [50], is used. Its description is given in subsection 4.5.2.

For now, in order to simplify the analysis of the fusion, we will assume that the effect of transmitting the state vector estimates to its neighbour vehicle is equivalent to having the *measurement* subgraphs transmitted. If we recall the example described in subsection 4.2.1, the measurements are functions of the state vector components, thus this assumption is valid if the system is observable (*i.e.,* the state can be inferred from measurements and output). Therefore, the *measurement* subgraphs are embedded in the communicated state vector observable estimates through the observable components.

Let us consider that, in vehicle i, the non-observable components (subscript $\bar{o}$ stands for non-observable), $\hat{\mathbf{x}}^i_{\bar{o}}$ can not be determined from the local *measurement* subgraph $\Yup^i$ but can be determined from the predecessor estimated state vector, $\hat{\mathbf{x}}^{i-1}_{o}$, computed locally by its *measurement* subgraph $\Yup^{i-1}$ and therefore observable. However, each vehicle filter requires more than one external *measurement* subgraph, if the formation is larger than or equal to 4 vehicles, as shown in Table 4.1. The question is how the local state vector estimates are combined with the state vector estimates from the previous vehicle, since the latter are viewed as measurements.

For now let us assume here the following equations to update the state vector, where the fusion of two state vector estimates is the weighted sum of both, as follows,

$$\hat{\mathbf{x}}^i_{trans,\bar{o}}(k|k) = w_c \hat{\mathbf{x}}^i_{trans,\bar{o}}(k|k-1) + (1-w_c)\,\hat{\mathbf{x}}^{i-1}_{trans,o}(k|k-1) \tag{4.26}$$

where $k \in \mathbb{N}$ is the time step, $\hat{\mathbf{x}}^i_{trans,\bar{o}}(k|k)$ is the estimate of the non-observable components (subscript $\bar{o}$) of the relative vector $\mathbf{x}_{trans}$ for vehicle i, at time k given observations up to, and including time k, $\hat{\mathbf{x}}^i_{trans,\bar{o}}(k|k-1)$ is the estimate of the non-observable components of the relative vector $\mathbf{x}_{trans}$ for vehicle i, at time k given observations up to, and including time $k-1$, $\hat{\mathbf{x}}^{i-1}_{trans,o}(k|k-1)$ is the estimate of the observable components (subscript o) of the relative vector $\mathbf{x}_{trans}$ for vehicle $i-1$, at time k given observations up to, and including time $k-1$, and the weight variable, w_c, takes real values between 0 and 1, $w_c \in [0,1]$ (later in subsection 4.5.2 we will see that they are a simplified version of the CI method),

$$\begin{aligned} w_c &= 0 \Longleftrightarrow \hat{\mathbf{x}}^i_{trans,\bar{o}}(k|k) = \hat{\mathbf{x}}^{i-1}_{trans,o}(k|k-1) \\ w_c &= 1 \Longleftrightarrow \hat{\mathbf{x}}^i_{trans,\bar{o}}(k|k) = \hat{\mathbf{x}}^{i}_{trans,\bar{o}}(k|k-1) \end{aligned}$$

- For $w_c = 0$, the non-observable components from vehicle i are updated only based on the predecessor vehicle filter $i-1$ observable component.
- For $w_c = 1$, the predecessor vehicle component does not contribute at all to the update of the non-observable component of vehicle filter i.

If $\hat{\mathbf{x}}^{i-1}_{o}$ is computed locally by its *measurement* graph, $\Yup^{i-1}$, assuming that the measurement error

is smaller then the process error, then the vehicle $i-1$ observable components are updated based on the measurements sensor, which has smaller error variance

$$\hat{\mathbf{x}}^{i-1}_{trans,o}(k|k) = \hat{\mathbf{x}}^{i-1}_{trans,o}(k|k-1) + \mathbf{K}\left(\mathbf{y}^{i-1}(k) - \mathbf{H}\hat{\mathbf{x}}^{i-1}_{trans,o}(k|k-1)\right) \tag{4.27}$$

where $\mathbf{K}$ is the Kalman gain and $\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}$ is the innovation or measurement residual in Kalman filter algorithm.

For vehicle filter i, its non-observable components, which are know are assumed to match the observable components from vehicle filter $i-1$, have the following error,

$$\widetilde{\mathbf{x}}^{i}_{trans,\bar{o}}(k) = \mathbf{x}_{trans} - \hat{\mathbf{x}}^{i}_{trans,\bar{o}}(k|k)$$

and using (4.26)

$$\widetilde{\mathbf{x}}^{i}_{trans,\bar{o}}(k) = \mathbf{x}_{trans} - w_c\hat{\mathbf{x}}^{i}_{trans,\bar{o}}(k|k-1) - (1-w_c)\,\hat{\mathbf{x}}^{i-1}_{trans,o}(k|k-1)$$

At vehicle i, the state vector estimation error for the non-observable components can be $\widetilde{\mathbf{x}}^{i}_{trans,\bar{o}}(k) = \mathbf{x}_{trans} - \hat{\mathbf{x}}^{i-1}_{trans,o}(k|k-1)$ if $w_c = 0$ or $\widetilde{\mathbf{x}}^{i}_{trans,\bar{o}}(k) = \mathbf{x}_{trans} - \hat{\mathbf{x}}^{i}_{trans,\bar{o}}(k|k-1)$ if $w_c = 1$, meaning that the non-observable components of the vehicle i will have the contribution of the vehicle $i-1$ observable components, depending on the weight value. The weight value can be computed considering the contribution of error covariance matrices from both state vector estimates, $\hat{\mathbf{x}}^{i}_{trans,\bar{o}}(k|k-1)$ and $\hat{\mathbf{x}}^{i-1}_{trans,o}(k|k-1)$, *i.e.*, $\mathbf{P}^{i}(k|k-1)$ and $\mathbf{P}^{i-1}(k|k-1)$. Notice that the non-observable components of vehicle i are now observable. If we consider as before that the effect of transmitting the full relative state vector estimate is, in a simplified version, equivalent to transmitting the *measurement* subgraph then, intuitively, the preceding result is also valid if we have more *measurement* subgraphs going through one communication link as is the case in the *peer-to-peer* information flow graph. Moreover, there is no need to complete a cycle to accomplish this, but only $N-2$ vehicles linked in a ring, (see Table 4.2).

However, this is only the simplified version of the CI method to give insight of the CI equations used. In the covariance intersection method, reported in subsection 4.5.2, the update of the state vector is performed through the equations,

$$\begin{aligned}
\hat{\mathbf{x}}^{i}(k|k) &= \mathbf{P}^{i}(k|k)\left\{w_c(\mathbf{P}^{i}(k|k-1))^{-1}\hat{\mathbf{x}}^{i}(k|k-1) + \right. \\
&\quad \left. (1-w_c)\left(\mathbf{P}^{i-1}(k|k-1)\right)^{-1}\hat{\mathbf{x}}^{i-1}(k|k-1)\right\} \\
(\mathbf{P}^{i}(k|k))^{-1} &= w_c(\mathbf{P}^{i}(k|k-1))^{-1} + (1-w_c)(\mathbf{P}^{i-1}(k|k-1))^{-1}
\end{aligned} \tag{4.28}$$

which verifies the simplified equations adopted before, with the state vector estimates not only weighted by the parameter w_c but also by the corresponding error covariance matrix. Therefore, the state vector estimates and its error covariance matrix (since $\mathbf{P}$ is symmetric, only the lower or upper triangular part, plus the diagonal have to be transmitted) have to be transmitted between vehicles, according to the peer-to-peer formation estimation graph.

***Information flow* graph example for 4-vehicle formation**

From (4.20) and using *measurement* subgraph Υ^1, we have $N-1=3$ observable components (from Eq. (4.24) $\hat{\mathbf{x}}^1_{trans,o} = \begin{bmatrix} \hat{\boldsymbol{\chi}}_{12} & \hat{\boldsymbol{\chi}}_{13} & \hat{\boldsymbol{\chi}}_{14} \end{bmatrix}$) and three non-observable components ($\hat{\mathbf{x}}^1_{trans,\bar{o}} = \begin{bmatrix} \hat{\boldsymbol{\chi}}_{23} & \hat{\boldsymbol{\chi}}_{24} & \hat{\boldsymbol{\chi}}_{34} \end{bmatrix}$). Using the peer-to-peer *information flow* graph, defined in subsection 3.4.1, information flows between pairs of vehicles, thus for vehicle filter 1, the observable state estimates from vehicle filter 4 (updated through *measurement* subgraph 4) are used to update the $N-2=2$ state vector components, $\hat{\boldsymbol{\chi}}_{24}$ and $\hat{\boldsymbol{\chi}}_{34}$. On the other hand, the non-observable component $\hat{\boldsymbol{\chi}}_{23}$ in vehicle filter 1 and 4, is updated benefitting of vehicle filter estimates from vehicle filter 3. As stated before in subsection 4.2.2, the $N-1=3$ external *measurement* subgraph is going to contribute with one state vector estimate. The result is the updating of all state vector components in vehicle filter 1.

4.3 Formation Translational Navigation Design

To compute the formation estimation graph, two graphs are considered on each vehicle i: the *measurement* graph and the peer-to-peer *information flow* graph, as shown in Fig. 4-6. The *measurement* graph concerns the measurements that are made locally at each vehicle filter, using the relative sensors (*e.g.*, to measure the distance with respect to another vehicle), and which are used in the filtering step of the Kalman filter (which is the same for all vehicles) to update some components of the local relative state vector. Afterwards, these full state vector estimates are then sent to the next vehicle filter $i+1$, in the peer-to-peer *information flow* graph. The peer-to-peer *information flow* graph allows communication of information between pairs of vehicles. This information is the full relative state vector and not the *measurement* subgraph as assumed before. Yet, the *measurement* subgraph, Υ^i, is embedded in the observable components of the vehicle filter i full relative state vector estimates, the same way that the *measurement* subgraph, Υ^{i-1}, is embedded in the full relative state vector estimates transmitted by vehicle filter $i-1$, and so on. Therefore, when the full state vector estimates are transmitted to the next vehicle filter $i+1$ through the channels in the peer-to-peer *information* flow graph, they are considered as observations in the receiving vehicle filter $i+1$. These observations are correlated with the state vector estimates to be computed there, and the degree of correlation is unknown. Moreover, the availability

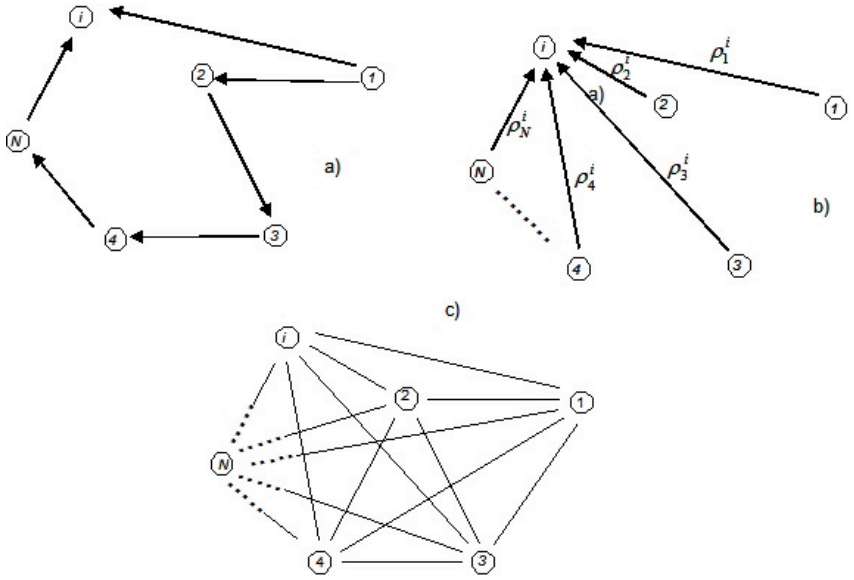

Figure 4-6: The graphs envolved in the formation navigation problem: a) *peer-to-peer* graph b) i^{th} *sensed measurement* graph, $¥^i$ c) underlying *formation estimation* graph.

of a limited number of sensors intensifies the need to use other vehicle entire state estimates, therefore the problem of correlation that occurs when a vehicle filter combines the state estimate sent by another vehicle filter with its own state grows stronger.

In summary, in the filtering step, the EKF is used when observations are measurements from the relative sensors, and the CI algorithm is applied whenever the observations are the state vector estimates from the predecessor vehicle, as shown in Fig. 4-7.

4.3.1 *Sensor failure* scenario

Recall the *sensor failure* scenario defined in the last chapter (reference to Fig. 3-3), where a formation of 6 vehicles is considered with several cases of measurement failures: total; partial malfunction in the relative sensor system; or merely occlusion during one of the transmissions. The Laplacian eigenvalues are $0^{(2)}$, $1^{(1)}$ and $3^{(2)}$, where the multiplicity of the null eigenvalue detects the isolated vehicle. Because of the isolated vehicle and from the analysis performed before, it would not be possible to obtain the estimate of the full state vector dimension, $\frac{N(N-1)}{2}$, since there is no way to combine the *measurement* graph and the *peer-to-peer* information flow graph to obtain the *formation estimation* graph. This means that if a vertex from the *measurement* graph has indegree and outdegree zero then it is not possible to compute the full state vector estimates. Therefore, vehicle 6 must be considered out of the formation, and the state vector components, or edges, related to it eliminated. Nevertheless, the state vector estimates

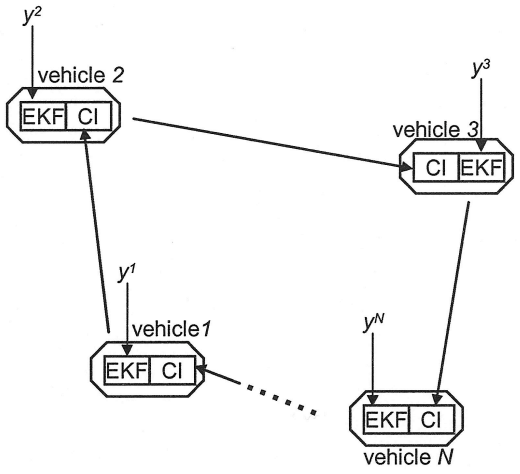

Figure 4-7: Decentralized filter, conjugating the CI algorithm with Kalman filter.

of the other vehicles can always be communicated, to update their own state vector estimates.

With the formation reduced to 5 vehicles the issue now is that the *measurement* graph is not fully connected, due to partial malfunction. So, how can this affect the state vector estimates in each vehicle? Notice that vehicle 3 is an undominated vertex, since $\deg_{in}(v_3) = 0$, and with a *measurement* graph, $¥^3$, having a null edge set, the Laplacian eigenvalues are zero. This means that it has no local *measurement* subgraph, as opposed to the *measurement* subgraph from vehicle 1, which has the maximum number of measurements (edges) on its local *measurement* subgraph. In fact, $¥^1$ is a dominated vertex, with $\deg_{in}(v_1) = 4$, and with Laplacian eigenvalues $eig\left\{\mathbf{L}\left(¥^4\right)\right\} = \left\{0^{(4)}, 4^{(1)}\right\}$. Therefore, the contribution of $¥^3$ to the *peer-to-peer* information flow graph is null, and the *measurement* graph $¥^1$ gives the maximum contribution, as shown in Fig. 4-8. Thus, we could establish a relation for all vertices as follows,

$$\deg_{in}(v_1) > \deg_{in}(v_5) > \deg_{in}(v_2) = \deg_{in}(v_4) > \deg_{in}(v_3) = 0$$

Since the *measurement* graph from vehicle 3, $¥^3$, does not contribute to the updating of any of its own state vector components, its role in the *peer-to-peer* information flow graph will aim at updating its own state vector at the cost of information from the other vehicles. Consequently, the communication link from the other vehicles is essential. For the other vehicles it is clear that at least one communication link must exist, so the full relative state vector will be updated at all vehicles.

To conclude, the full relative state vector update is feasible when the peer-to-peer *information flow* graph is used (indegree/outdegree equal to one), whenever the intersection with the *formation estimation*

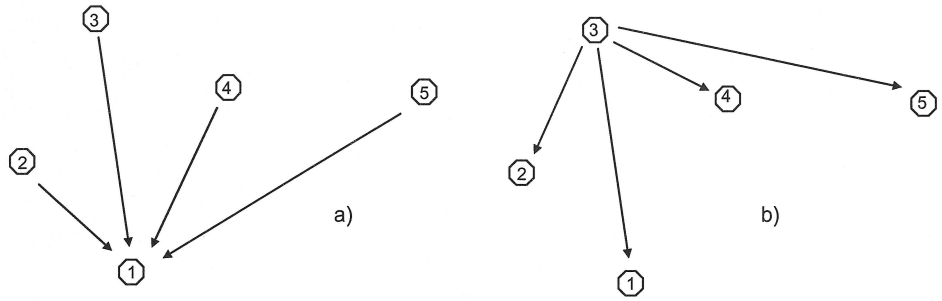

Figure 4-8: a) *sensed measurement* graph, $\Yup^1$, $\deg_{in}(v_1) = 4$, dominated vertex, with maximum number of received measurements opposite to b) *sensed measurement* graph $\Yup^3$, $\deg_{in}(v_3) = 0$, undominated vertex, with zero number of received measurements.

graph of the sum of the underlying *measurement* graphs without undominated vertices, results in the *formation estimation* graph,

$$\sum_{i\in V,\deg_{in}(v_i)>0,l=1}^{l=N-1} \left(\Yup^{i+l}\right) \cap \Fup = \Fup \tag{4.29}$$

which can also be inferred from the Laplacian eigenvalues of the union. The multiplicity one Laplacian eigenvalue is one and the null eigenvalue has multiplicity one, see equation (4.25).

Equation (4.29) is a necessary pre-condition, but not sufficient. We also must guarantee that there always exists at least one communication link at each vertex, *i.e.,* $\deg_{in}(\cdot) \geq 1$ on the *information flow* graph.

An algorithm to compute if a full state vector estimates is possible on each vehicle of N-vehicle formation, follows which also quantifies the number of state vector components that can be estimated. For a N-vehicle formation consider the *formation estimation* graph, the *measurement* graph, the *measurement* subgraphs and the *information flow* graph (*i.e.,* the peer-to-peer topology). At the starting point is assumed that the full state vector estimates is attained and its number is the *formation estimation* graph size.

4.3.2 Algorithm

Consider a formation of N vehicles:

1. If the *measurement* graph of a formation is κ-regular, all vertices are dominant with value κ, then

   1.1 decompose the *measurement* graph into *measurement* subgraphs (as defined in Section 3.3);

1.2 combine the underlying *measurement* subgraphs, as discussed in subsection 4.2.2, in order to match the *formation estimation* graph

$$\sum_{i\in V, l=1}^{l=N-1} \left(¥^{i+l}\right)^{U} = ₣$$

This can be accomplished by adding *measurement* subgraphs until its multiplicity one eigenvalue is one and the null eigenvalue has multiplicity one;

1.3 verify that the *information flow* graph has indegree and outdegree equal to one ($\deg_{in}(v_i) = \deg_{out}(v_i) = 1$), *i.e.,* peer-to-peer topology, otherwise jump to 3.

2. If the *measurement* graph of a formation is not κ-regular, there are dominant and undominant vertices, then

2.1 decompose the *measurement* graph into *measurement* subgraphs (as defined in Section 3.3);

2.2 take the vertices that are undominated out of the algorithm, *i.e.,* the state vector components that are related to the vertex. Add one per undominated vertex to the number of vertices that are not able to determine the full state vector estimates from the *information flow* graph;

2.3 combine the remaining *measurement* subgraphs, such that the combination matches the *formation estimation* graph,

$$\sum_{i\in V, l=1, \deg_{in}(v_i)>0}^{l=N-1} \left(¥^{i+l}\right)^{U} = ₣$$

This can be accomplished by adding *measurement* subgraphs until its multiplicity one eigenvalue is one and the null eigenvalue has multiplicity one, otherwise jump to 3;

2.4 verify that the *information flow* graph has indegree/outdegree equal to one, *i.e.,* peer-to-peer topology.

3. For a vertex not in the *information flow* graph,

3.1 if $\deg_{in}(v_i) = 1$ and $\deg_{out}(v_i) = 0$ (sink vertex);

3.1.1 the correspondent vertex is not in the previous algorithm, but the vehicle filter is able to estimate the full state vector since it is able to receive the state vector estimate from the previous vehicle;

3.2 if $\deg_{in}(v_i) = 0$ (source vertex) and $\deg_{out}(v_i) = 1$;

3.2.1 the correspondent vertex is in the algorithm if it is not undominated in its *measurement* subgraph. However, the vehicle filter is not able to estimate the full state vector, but

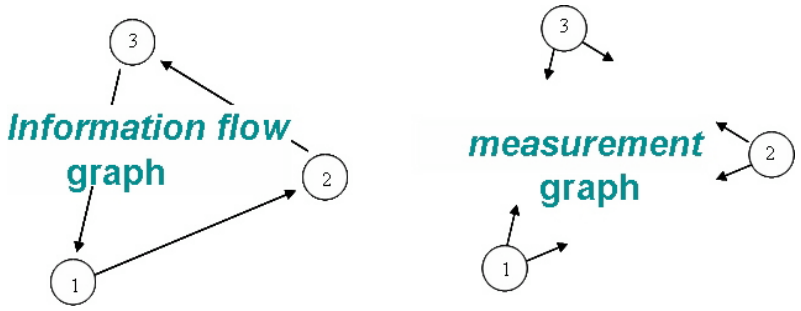

Figure 4-9: Peer-to-peer *Information flow* graph $\wp^P$ and *measurement* graph ¥ for a 3 s/c formation flying example.

only the state vector components that are locally observable. Add one to the number of vertices that are not able to determine the full state vector estimates, to account for the state vector components not estimated.

4.3.3 Example of three formation flying spacecraft

For the example of $N = 3$ formation flying s/c, where the *measurement* graph and *information flow* graph are as shown in Fig. 4-9, the *formation estimation* graph order is $\frac{N(N-1)}{2} = 3$ for each of the component subsets (relative position, translational relative velocity, relative attitude, relative angular velocity) represented by the formation estimation graph, shown in Fig. 4-10. Knowing that each component subset has x, y, z coordinates, then in total there are 18 components of the translational relative state vector (order of *formation estimation* graph $\times$ 2 translation components subset $\times$ 3 coordinates),

$$\mathbf{x}_{trans} = \begin{bmatrix} \boldsymbol{\chi}_{12}^T & \boldsymbol{\chi}_{13}^T & \boldsymbol{\chi}_{23}^T & \dot{\boldsymbol{\chi}}_{12}^T & \dot{\boldsymbol{\chi}}_{13}^T & \dot{\boldsymbol{\chi}}_{23}^T \end{bmatrix}^T$$
$$\begin{bmatrix} \begin{bmatrix} x_{12} \\ y_{12} \\ z_{12} \end{bmatrix}^T & \begin{bmatrix} x_{13} \\ y_{13} \\ z_{13} \end{bmatrix}^T & \begin{bmatrix} x_{23} \\ y_{23} \\ z_{23} \end{bmatrix}^T & \begin{bmatrix} \dot{x}_{12} \\ \dot{y}_{12} \\ \dot{z}_{12} \end{bmatrix}^T & \begin{bmatrix} \dot{x}_{13} \\ \dot{y}_{13} \\ \dot{z}_{13} \end{bmatrix}^T & \begin{bmatrix} \dot{x}_{23} \\ \dot{y}_{23} \\ \dot{z}_{23} \end{bmatrix}^T \end{bmatrix}^T$$

Notice that the components of vector $\begin{bmatrix} x_{23} & y_{23} & z_{23} \end{bmatrix}^T$ and $\begin{bmatrix} \dot{x}_{23} & \dot{y}_{23} & \dot{z}_{23} \end{bmatrix}^T$ are not observable in s/c 1 and so forward.

As can be verified, the *measurement* graph is 2-regular, thus all vertices are dominated with value 2. This way, all vertices have the maximum contribution to the local state vector estimation, as discussed previously. The union of any two *measurement* subgraphs gives the *formation estimation* graph, whose eigenvalue with multiplicity one is one, and the null eigenvalue has multiplicity one. Thus, all the state vector components can be estimated locally at each s/c.

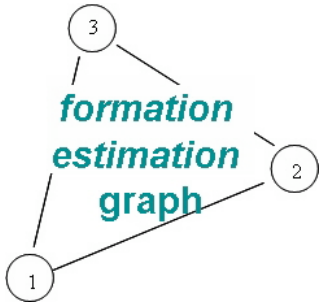

Figure 4-10: *formation estimation* graph for the 3 s/c formation flying example.

4.4 Recursive Nonlinear Estimation

The Kalman filter is the optimal recursive state estimator, in the sense that it minimizes the mean square estimation error, when applied to the state estimation of linear dynamic systems subject to zero-mean Gaussian process and observation noise. Nevertheless, it can still be used for nonlinear systems by linearizing the trajectory at each point and applying the Kalman filter algorithm. However, this is a sub-optimal solution and the Kalman filter stability property is not guaranteed. Some of the problems that may arise are:

- the measurement and process noise may not be Gaussian white noise sequences as assumed;
- variables assumed to be uncorrelated may be correlated;
- the linearization of the process and the measurement models may introduce significant errors;
- the initial state, $\mathbf{x}(0)$, and initial state error covariance matrix, $\mathbf{P}\,(0|0)$, are required.

Despite the described problems, it is well-known that the EKF is the simplest and most widely used approach, and has been used to solve successfully countless nonlinear practical problems [52]. Besides, its great popularity arises from a number of important and well-known properties valid for linear systems, which include:

- Optimality: The filter is the optimal linear Mean Square Error (MSE) estimator. Given a linear update rule, no other estimator yields an estimate that has a smaller MSE.
- Recursiveness: In each iteration only the current sensor observations are used and it is not necessary to maintain and re-process previous observations. This minimizes the storage necessary and allows estimates to be made on-line with modest computational resources.

- Linearity: The state estimate is obtained through a weighted linear sum of information from the predictions and the observations. This provides a simple and efficient means of combining information.

- Covariance Modelling: The uncertainties associated with the system behavior and observations are explicitly modeled using random variables with specified means and covariances. This provides a confidence measure in the estimates provided by the algorithm.

Because the system is nonlinear, the local 6DoF EKF algorithm, used in each vehicle to update the components of the full relative state vector, uses a linearized model of the nonlinear system. In this section, we review the equations of the Extended Kalman filter, emphasizing their differences towards the KF algorithm.

4.4.1 Extended Kalman filter

Since the state differential equations, as well as those relating the state and the measurements, are in general nonlinear, as defined in Section 4.1.2, the Kalman filter is not applicable unless this nonlinear system is linearized.

Consider the nonlinear equation being defined in subsection 4.1.2. The linearized and discretized model to be used by the EKF is derived in appendix B, and summarized as follows:

Filtering

After acquiring measurements, the full state estimate is corrected by the small perturbation state estimates

$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \delta\hat{\mathbf{x}}\,(k|k) \tag{4.30}$$

where the state estimate perturbation is,

$$\delta\hat{\mathbf{x}}\,(k|k) = \mathbf{K}(k)\left(\mathbf{y}(k) - h(k, \hat{\mathbf{x}}(k|k-1))\right) \tag{4.31}$$

with Kalman Gain matrix,

$$\mathbf{K}(k) = \mathbf{P}(k|k-1)\mathbf{H}^T(k)\left[\mathbf{H}(k)\mathbf{P}(k|k-1)\mathbf{H}^T(k) + \mathbf{R}(k)\right]^{-1} \tag{4.32}$$

where $\mathbf{P}(k|k-1)$ is the error covariance matrix, $\mathbf{R}$ is the measurement noise matrix, and $\mathbf{H}(k)$ is the Jacobian of function h evaluated at the current filter's estimates,

$$\mathbf{H}(k) \triangleq \left[\nabla h(k, \mathbf{x})^T\right]^T\Big|_{\mathbf{x}=\hat{\mathbf{x}}(k|k)} \tag{4.33}$$

where ∇_x is the gradient operator, $\nabla_x = \begin{bmatrix} \frac{\partial}{\partial x_1} \\ \vdots \\ \frac{\partial}{\partial x_{n_x}} \end{bmatrix}$. The innovation covariance matrix

$$\mathbf{S}(k) = \mathbf{H}(k)\mathbf{P}(k|k-1)\mathbf{H}^T(k) + \mathbf{R}(k)$$

and the update state covariance as a function of $\mathbf{S}$ is,

$$\mathbf{P}(k|k) = \mathbf{P}(k|k-1) - \mathbf{K}(k)\mathbf{S}^{-1}(k)\mathbf{K}^T(k) \tag{4.34}$$

or

$$\mathbf{P}(k|k) = [\mathbf{1} - \mathbf{K}(k)\mathbf{H}(k)]\,\mathbf{P}(k|k-1) \tag{4.35}$$

The state covariance matrix update equation (4.34) is very sensitive to round-off errors and may lose its positive definiteness and symmetry. For large uncertainty in the initial estimate, the prior error covariance matrix has very large variances (values along the major diagonal) and if the measurements are initially very accurate, the state covariance update equation (4.35) will have an undetermined solution

$$\begin{aligned} \mathbf{P}(k|k) &= [\mathbf{1} - \mathbf{K}(k)\mathbf{H}(k)]\,\mathbf{P}(k|k-1) \\ &\simeq 0 \times \infty \end{aligned}$$

Thus, an alternative update formula for the update state covariance is the Joseph form, to guarantee positive semi-definiteness,

$$\mathbf{P}(k|k) = [\mathbf{1} - \mathbf{K}(k)\mathbf{H}(k)]\,\mathbf{P}(k|k-1)\,[\mathbf{1} - \mathbf{K}(k)\mathbf{H}(k)]^T + \mathbf{K}(k)\mathbf{R}(k)\mathbf{K}^T(k) \tag{4.36}$$

which has natural symmetry. While Eq. (4.35) has a subtraction that may lead its eigenvalues to be negative, the term $[\mathbf{1} - \mathbf{K}(k)\mathbf{H}(k)]$ in (4.36) is multiplied by its transpose, which preserves the positive definiteness and it is thus less sensitive to round-off errors. The drawback is that more computational effort is required to compute (4.36) compared to (4.35).

Prediction

The state is predicted through the following equation,

$$\frac{d}{dt}\hat{\mathbf{x}}(t) = f(\hat{\mathbf{x}}(t), t) \tag{4.37}$$

which is approximated by a fourth-order four-stage Runge-Kutta method, [53], as follows,

$$\hat{\mathbf{x}}_{\ell+1} = \hat{\mathbf{x}}_\ell + \frac{\Delta t}{6}\left(\mathbf{\mathfrak{P}}_1 + 2\mathbf{\mathfrak{P}}_2 + 2\mathbf{\mathfrak{P}}_3 + \mathbf{\mathfrak{P}}_4\right)$$

with,

$$\begin{aligned}
\mathbf{\mathfrak{P}}_1 &= f(\hat{\mathbf{x}}_\ell, t_\ell) \\
\mathbf{\mathfrak{P}}_2 &= f(\hat{\mathbf{x}}_\ell + \frac{\Delta t}{2}\mathbf{\mathfrak{P}}_1, t_\ell + \frac{\Delta t}{2}) \\
\mathbf{\mathfrak{P}}_3 &= f(\hat{\mathbf{x}}_\ell + \frac{\Delta t}{2}\mathbf{\mathfrak{P}}_2, t_\ell + \frac{\Delta t}{2}) \\
\mathbf{\mathfrak{P}}_4 &= f(\hat{\mathbf{x}}_\ell + \Delta t\mathbf{\mathfrak{P}}_3, t_\ell + \Delta t)
\end{aligned}$$

where Δt is the Runge-Kutta step size and $\ell \in \mathbb{N}^0$. The error covariance matrix is updated by

$$\mathbf{P}(k+1|k) = \mathbf{\Phi}(k)\mathbf{P}(k|k)(\mathbf{\Phi}(k))^T + \mathbf{Q}(k) \tag{4.38}$$

From the previous equation, $\mathbf{Q}$ is the covariance matrix of the process noise, mainly due to unmodeled dynamics, which (as we will discuss in Chapter 5) is required to be small, and $\mathbf{\Phi}(k)$ is the state transition matrix $(n_x \times n_x)$ which is approximated by $\mathbf{\Phi}(k) \approx \mathbf{1} + \mathbf{F}(\hat{\mathbf{x}}(t), t)T_s + \frac{(\mathbf{F}(\hat{\mathbf{x}}(t),t)T_s)^2}{2!}$, where T_s is the sampling period used to propagate the estimate. The matrix $\mathbf{F}(t)$ is the Jacobian of function f evaluated at the current filter's estimates,

$$\mathbf{F}(t) \triangleq \left[\nabla f(\mathbf{x},t)^T\right]^T\Big|_{\mathbf{x}=\hat{\mathbf{x}}(t)} \tag{4.39}$$

The nonlinearities of the system dynamics are approximated by a linearized version of the nonlinear system model around the previous step state estimate. For this approximation to be valid, and not to diverge, this linearization should be a good approximation of the nonlinear model for the uncertainty region associated with the state estimate. Rather than propagating the non-Gaussian pdf, the EKF considers, at each step, a linearization of the nonlinear dynamics around the last consecutive predicted and filtered estimates of the state and, for the linearized dynamics, applies the Kalman filter equations. In the Extended Kalman Filter, contrary to the optimal linear KF, the estimate at time k is only approximately the state mean

$$\hat{\mathbf{x}}(k|k) \approx E\left\{\mathbf{x}(k)\right\} \tag{4.40}$$

and its associated covariance matrix is the MSE matrix, rather than the error covariance matrix, as a consequence of the fact that the estimate is not in general equal but instead approximated to the mean:

$$MSE[\hat{\mathbf{x}}(k|k)] = E\left\{\tilde{\mathbf{x}}(k|k)\tilde{\mathbf{x}}^T(k|k)\right\} = E\left\{(\mathbf{x}(k) - E\left\{\mathbf{x}(k)\right\})(\mathbf{x}(k) - E\left\{\mathbf{x}(k)\right\})^T\right\}$$

$$MSE[\hat{\mathbf{x}}(k|k)] \approx E\left\{(\mathbf{x}(k) - \hat{\mathbf{x}}(k|k))(\mathbf{x}(k) - \hat{\mathbf{x}}^T(k|k))\right\} = \mathbf{P}(k|k)$$

The approximation of the EKF is not optimal, since a global property of a function, *i.e.,* its average, is replaced by a local property, its derivative. However, the performance of the filter can be significantly improved by extending the EKF to its second order terms or through the Iterated Extended Kalman Filter (IKF), which is useful in reducing the effect of measurement function nonlinearity, both presented in the next chapter.

4.5 Fusion of Multiple Vehicle State Estimates

In an environment of a decentralized architecture where information from vehicle filter estimates is combined, it is mandatory to provide a solution that guarantees a consistent result, from the estimate fusion process. The Kalman filter provides optimal results as long as the given estimates are consistent and their degree of correlation is known, yet when there is no knowledge of the cross-covariance, the estimates produced are not guaranteed to be consistent, resulting in divergence of the filter. Consistency is covered in Section 5.4, and is especially concerned with the strength of the error estimate relative to its uncertainty. Covariance Intersection (CI) and Covariance Union (CU) methods, proposed by [54], are related solutions that provide consistency in the results without knowing the degree of correlation. The main difference between them is that CU is guaranteed to preserve consistency as long as one of the given estimates is consistent whereas for the CI solution the given estimates that are to be fused have to be consistent. The CU method is used to restore the consistency of the given estimates and is used as a complement to the CI method, as stressed by [54]. Since we are assuming that the estimates, determined locally in each vehicle by the EKF, are consistent, the use of CU is dismissed, but can always be included to support the CI method before the data fusion.

The CI algorithm provides a way to compute an upper bound on the covariance matrix for the problem of fusing variables that are suspected or known to be correlated, without previous knowledge of the degree of correlation of the given estimates, thus avoiding the instability problems that affect the performance of the filters. Furthermore, it enables a decentralized solution to be obtained for our formation full state estimation problem, as we saw previous in Section 4.2.

The main issues addressed in [3] are:

- the state estimates received from other vehicle filters are affected by the estimation error,
- the CI filter is used to fuse the estimates of the previous vehicle, and
- minimum communication is used in the formation, as required by the *peer-to-peer* information flow graph.

In this section, we will start by showing that when using Kalman filter state update equations there is a correlation between state vector estimates, whose degree of correlation is unknown, affecting the performance of the filter.

4.5.1 Correlation between formation state estimates of different vehicles

In this subsection we show that indeed the formation relative state estimates of the different vehicles are correlated. Considering a general state vector, $\mathbf{x}$ with dimension $n_x \times 1$, it is known that the off-diagonal terms of the covariance matrix are indicators of the cross-correlation between the elements of the state vector,

$$\begin{aligned} cov(\mathbf{x}) &\triangleq E\{(\mathbf{x} - E\{\mathbf{x}\})(\mathbf{x} - E\{\mathbf{x}\})^T\} \\ &= \begin{bmatrix} var(x_1) & cov(x_1, x_2) & \cdots & cov(x_1, x_{n_x}) \\ cov(x_2, x_1) & var(x_2) & \cdots & cov(x_2, x_{n_x}) \\ \vdots & \vdots & \ddots & \vdots \\ cov(x_{n_x}, x_1) & cov(x_{n_x}, x_2) & \cdots & var(x_{n_x}) \end{bmatrix} \\ &= \begin{bmatrix} \sigma_{x_1}^2 & \varsigma_{x_1,x_2}\sigma_{x_1}\sigma_{x_2} & \cdots & \varsigma_{x_1,x_{n_x}}\sigma_{x_1}\sigma_{x_{n_x}} \\ \varsigma_{x_2,x_1}\sigma_{x_2}\sigma_{x_1} & \sigma_{x_2}^2 & \cdots & \varsigma_{x_2,x_{n_x}}\sigma_{x_2}\sigma_{x_{n_x}} \\ \vdots & \vdots & \ddots & \vdots \\ \varsigma_{x_{n_x},x_1}\sigma_{x_{n_x}}\sigma_{x_1} & \varsigma_{x_{n_x},x_2}\sigma_{x_{n_x}}\sigma_{x_2} & \cdots & \sigma_{x_{n_x}}^2 \end{bmatrix} \end{aligned}$$

with dimension $n_x \times n_x$, the number of rows $n = 1...n_x$, the number of columns $m = 1...n_x$, $\varsigma_{\mathbf{x}_n,\mathbf{x}_m}$ is the correlation coefficient of $\mathbf{x}_n$ and $\mathbf{x}_m$, $\varsigma_{\mathbf{x}_n,\mathbf{x}_m} = \frac{E\{\mathbf{x}_n\mathbf{x}_m\} - E\{\mathbf{x}_n\}E\{\mathbf{x}_m\}}{\sigma_{\mathbf{x}_n}\sigma_{\mathbf{x}_m}}$, and σ_x is the standard deviation of $\mathbf{x}$. For the uncorrelated case, $\varsigma_{\mathbf{x}_n,\mathbf{x}_m} = 0$ and the covariance matrix becomes diagonal. Using the state estimate error vector instead, the idea is to compute the error covariance matrix, $\mathbf{P}^i = E\{(\widetilde{\mathbf{x}}^i)(\widetilde{\mathbf{x}}^i)^T\}$ for each vehicle i in a N-vehicle formation.

Denoting the entire formation state vector estimated in the $i^{th} - 1$ vehicle as $\widetilde{\mathbf{x}}$, the i^{th} vehicle will get observations as follows,

$$\mathbf{z}^i(k) = \mathbf{x}(k) + \widetilde{\mathbf{x}}^{i-1}(k-1), \;\; E\left\{\widetilde{\mathbf{x}}^{i-1}(k-1)(\widetilde{\mathbf{x}}^{i-1}(k-1))^T\right\} = \mathbf{P}^{i-1}(k) \text{ for vehicle } i \tag{4.41}$$

where $\widetilde{\mathbf{x}}^i$ is the estimation error of the Kalman filter for vehicle $i = 1, ..., N$. Knowing that the estimation error is as follows,

$$\widetilde{\mathbf{x}}^i = \mathbf{x} - \hat{\mathbf{x}}^i \text{ for vehicle } i \tag{4.42}$$

the error covariance matrix associated with the updated estimate is

$$\mathbf{P}^i = E\left\{\widetilde{\mathbf{x}}^i(\widetilde{\mathbf{x}}^i)^T\right\} = E\left\{(\mathbf{x}-\hat{\mathbf{x}}^i)(\mathbf{x}-\hat{\mathbf{x}}^i)^T\right\} \text{ for vehicle } i \tag{4.43}$$

Comparing (4.41) with a generic model observation model,

$$\mathbf{z}(k) = \mathbf{H}\mathbf{x}(k) + \mathbf{v}(k) \tag{4.44}$$

the measurement noise covariance matrix $\mathbf{R} = E\left\{\mathbf{v}\mathbf{v}^T\right\}$ is the error covariance matrix of the $i^{th}-1$ vehicle and $\mathbf{H} = \mathbf{1}$.

Knowing that the general estimate updating equation is

$$\hat{\mathbf{x}}^i(k|k) = \hat{\mathbf{x}}^i(k|k-1) + \mathbf{K}(k)(\mathbf{z}^i(k) - \mathbf{H}\hat{\mathbf{x}}^i(k|k-1)) \text{ for vehicle } i$$

where $\mathbf{K}$ is the Kalman Gain matrix, and using (4.41) and $\mathbf{H} = \mathbf{1}$ it follows,

$$\hat{\mathbf{x}}^i(k|k) = \hat{\mathbf{x}}^i(k|k-1) + \mathbf{K}(k)(\mathbf{x}(k) + \widetilde{\mathbf{x}}^{i-1}(k-1) - \hat{\mathbf{x}}^i(k|k-1)) \text{ for vehicle } i$$

and using (4.42),

$$\hat{\mathbf{x}}^i(k|k) = \hat{\mathbf{x}}^i(k|k-1) + \mathbf{K}(k)(\widetilde{\mathbf{x}}^i(k-1) + \widetilde{\mathbf{x}}^{i-1}(k-1))$$

The error covariance matrix (4.43) using the previous estimate updating equation becomes, for vehicle i

$$\begin{aligned}
\mathbf{P}^i(k|k) &= E\{\{(\mathbf{x}-\hat{\mathbf{x}}^i(k|k-1)) - \mathbf{K}(k)\left(\widetilde{\mathbf{x}}^i(k-1) + \widetilde{\mathbf{x}}^{i-1}(k-1)\right)\} \\
&\quad \left\{(\mathbf{x}-\hat{\mathbf{x}}^i(k|k-1)) - \mathbf{K}(k)(\widetilde{\mathbf{x}}^i(k-1) + \widetilde{\mathbf{x}}^{i-1}(k-1)\right\}^T\} \\
&= E\{\left\{(\widetilde{\mathbf{x}}^i(k-1)) - \mathbf{K}(k)(\widetilde{\mathbf{x}}^i(k-1) + \widetilde{\mathbf{x}}^{i-1}(k-1))\right\} \\
&\quad \left\{(\widetilde{\mathbf{x}}^i(k-1)) - \mathbf{K}(k)(\widetilde{\mathbf{x}}^i(k-1) + \widetilde{\mathbf{x}}^{i-1}(k-1))\right\}^T\} \\
&= E\left\{(\widetilde{\mathbf{x}}^i(k-1)\left(\widetilde{\mathbf{x}}^i(k-1)\right)^T\right\} - E\left\{\widetilde{\mathbf{x}}^i(k-1)\left(\widetilde{\mathbf{x}}^i(k-1) + \widetilde{\mathbf{x}}^{i-1}(k-1)\right)^T\right\}\mathbf{K}^T(k) \\
&\quad -\mathbf{K}(k)E\left\{\left(\widetilde{\mathbf{x}}^i(k-1) + \widetilde{\mathbf{x}}^{i-1}(k-1)\right)(\widetilde{\mathbf{x}}^i(k-1))^T\right\} \\
&\quad +\mathbf{K}(k)E\left\{\left(\widetilde{\mathbf{x}}^i(k-1) + \widetilde{\mathbf{x}}^{i-1}(k-1)\right)\left(\widetilde{\mathbf{x}}^i(k-1) + \widetilde{\mathbf{x}}^{i-1}(k-1)\right)^T\right\}\mathbf{K}^T(k)
\end{aligned}$$

then,

$$\begin{aligned}\mathbf{P}^i(k|k) &= \mathbf{P}^i(k|k-1) - E\left\{\widetilde{\mathbf{x}}^i(k-1)\left(\widetilde{\mathbf{x}}^i(k-1)\right)^T\right\}\mathbf{K}^T(k) - E\left\{\widetilde{\mathbf{x}}^i(k-1)\left(\widetilde{\mathbf{x}}^{i-1}(k-1)\right)^T\right\}\mathbf{K}^T(k)\\ &\quad -\mathbf{K}(k)E\left\{\widetilde{\mathbf{x}}^i(k-1)(\widetilde{\mathbf{x}}^i(k-1))^T\right\} - \mathbf{K}(k)E\left\{\widetilde{\mathbf{x}}^{i-1}(k-1)(\widetilde{\mathbf{x}}^i(k-1))^T\right\}\\ &\quad +\mathbf{K}(k)E\left\{\widetilde{\mathbf{x}}^i(k-1)(\widetilde{\mathbf{x}}^i(k-1))^T\right\}\mathbf{K}^T(k) + \mathbf{K}(k)E\left\{\widetilde{\mathbf{x}}^{i-1}(k-1)(\widetilde{\mathbf{x}}^{i-1}(k-1))^T\right\}\mathbf{K}^T(k)\\ &\quad +\mathbf{K}(k)E\left\{\widetilde{\mathbf{x}}^i(k-1)(\widetilde{\mathbf{x}}^{i-1}(k-1))^T\right\}\mathbf{K}^T(k) + \mathbf{K}(k)E\left\{\widetilde{\mathbf{x}}^{i-1}(k-1)(\widetilde{\mathbf{x}}^i(k-1))^T\right\}\mathbf{K}^T(k)\end{aligned}$$

and using $\mathbf{P}^{i-1,i} = \left\{\widetilde{\mathbf{x}}^{i-1}\left(\widetilde{\mathbf{x}}^i\right)^T\right\}$ and $\mathbf{P}^{i,i-1} = \left\{\widetilde{\mathbf{x}}^i\left(\widetilde{\mathbf{x}}^{i-1}\right)^T\right\}$,

$$\begin{aligned}\mathbf{P}^i(k|k) &= \mathbf{P}^i(k|k-1) - \mathbf{P}^i(k|k-1)\mathbf{K}^T(k) - \mathbf{P}^{i,i-1}(k|k-1)\mathbf{K}^T(k) - \mathbf{K}(k)\mathbf{P}^i(k|k-1)\\ &\quad -\mathbf{K}(k)\mathbf{P}^{i-1,i}(k|k-1) + \mathbf{K}(k)\mathbf{P}^i(k|k-1)\mathbf{K}^T(k) + \mathbf{K}(k)\mathbf{P}^{i-1}(k|k-1)\mathbf{K}^T(k)\\ &\quad +\mathbf{K}(k)\mathbf{P}^{i,i-1}(k|k-1)\mathbf{K}^T(k) + \mathbf{K}(k)\mathbf{P}^{i-1,i}(k|k-1)\mathbf{K}^T(k)\end{aligned}$$

Knowing that

$$\mathbf{P} - \mathbf{P}\mathbf{K}^\mathbf{T} - \mathbf{K}\mathbf{P} + \mathbf{K}\mathbf{P}\mathbf{K}^\mathbf{T} = [\mathbf{1} - \mathbf{K}]\,\mathbf{P}\,[\mathbf{1} - \mathbf{K}]^T$$

then,

$$\begin{aligned}\mathbf{P}^i(k|k) &= [\mathbf{1} - \mathbf{K}(k)]\,\mathbf{P}^i(k|k-1)\,[\mathbf{1} - \mathbf{K}(k)]^T + \mathbf{K}(k)\mathbf{P}^{i-1}(k|k-1)\mathbf{K}^T(k)\\ &\quad -\mathbf{P}^{i,i-1}(k|k-1)\mathbf{K}^T(k) - \mathbf{K}(k)\mathbf{P}^{i-1,i}(k|k-1)\\ &\quad +\mathbf{K}(k)\mathbf{P}^{i,i-1}(k|k-1)\mathbf{K}^T(k) + \mathbf{K}(k)\mathbf{P}^{i-1,i}(k|k-1)\mathbf{K}^T(k)\end{aligned} \tag{4.45}$$

From (4.41), we can observe that in this special measurement model, the error covariance matrix ($\mathbf{P}^{i-1} = E\left\{\widetilde{\mathbf{x}}^{i-1}(\widetilde{\mathbf{x}}^{i-1})^T\right\}$) replaces the usual place of the measurement noise covariance matrix ($\mathbf{R}^i = E\left\{\boldsymbol{\nu}^i(\boldsymbol{\nu}^i)^T\right\}$) and the cross-covariance matrices $\mathbf{P}^{i,i-1} = \left\{\widetilde{\mathbf{x}}^i\left(\widetilde{\mathbf{x}}^{i-1}\right)^T\right\}$ and $\mathbf{P}^{i-1,i} = \left\{\widetilde{\mathbf{x}}^{i-1}\left(\widetilde{\mathbf{x}}^i\right)^T\right\}$ replace $\mathbf{E}\left\{\widetilde{\mathbf{x}}\mathbf{v}^T\right\}$ and $\mathbf{E}\left\{\mathbf{v}\widetilde{\mathbf{x}}^T\right\}$, respectively. Thus, comparing with a generic linearized observation model (4.44) and assuming that the observation error is independent of the estimation error,

$$\begin{aligned}\mathbf{E}\left\{\widetilde{\mathbf{x}}\mathbf{v}^T\right\} &= 0\\ \mathbf{E}\left\{\mathbf{v}\widetilde{\mathbf{x}}^T\right\} &= 0\end{aligned}$$

then, (4.45) will be reduced to the Joseph form (4.36), with $\mathbf{H} = 1$,

$$\mathbf{P}^i(k|k) = [\mathbf{1} - \mathbf{K}(k)]\,\mathbf{P}^i(k|k-1)\,[\mathbf{1} - \mathbf{K}(k)]^T + \mathbf{K}(k)\mathbf{R}(k)\mathbf{K}^T(k) \tag{4.46}$$

Therefore, equation (4.45) will reduce to the Joseph form if the cross-covariance matrices are zero, $\mathbf{P}^{i,i-1}(k|k-1) = \mathbf{0}$ and $\mathbf{P}^{i-1,i}(k|k-1) = \mathbf{0}$, *i.e.,* the state variables are uncorrelated.

Let us consider the case of relative state vector where their components express relative relations among vehicles. For example, the state vector,

$$\mathbf{x}_{trans} = \begin{bmatrix} x_{12} & x_{23} & x_{13} \end{bmatrix}^T \tag{4.47}$$

meaning that the state vector variables are dependent on each other, then its error covariance matrix for discrete-time k is,

$$\mathbf{P}^i = \begin{bmatrix} E\left\{\tilde{x}^i_{12}\left(\tilde{x}x^i_{12}\right)^T\right\} & E\left\{\tilde{x}^i_{12}\left(\tilde{x}^i_{23}\right)^T\right\} & E\left\{\tilde{x}^i_{12}\left(\tilde{x}^i_{13}\right)^T\right\} \\ E\left\{\tilde{x}^i_{23}\left(\tilde{x}x^i_{12}\right)^T\right\} & E\left\{\tilde{x}^i_{23}\left(\tilde{x}^i_{23}\right)^T\right\} & E\left\{\tilde{x}^i_{23}\left(\tilde{x}^i_{13}\right)^T\right\} \\ E\left\{\tilde{x}^i_{13}\left(\tilde{x}x^i_{12}\right)^T\right\} & E\left\{\tilde{x}^i_{13}\left(\tilde{x}^i_{23}\right)^T\right\} & E\left\{\tilde{x}^i_{13}\left(\tilde{x}^i_{13}\right)^T\right\} \end{bmatrix}$$

then for $\mathbf{P}^{i,i-1}$ and $\mathbf{P}^{i-1,i}$, we have,

$$\mathbf{P}^{i,i-1} = \begin{bmatrix} E\left\{\tilde{x}^i_{12}\left(\tilde{x}^{i-1}_{12}\right)^T\right\} & E\left\{\tilde{x}^i_{12}\left(\tilde{x}^{i-1}_{23}\right)^T\right\} & E\left\{\tilde{x}^i_{12}\left(\tilde{x}^{i-1}_{13}\right)^T\right\} \\ E\left\{\tilde{x}^i_{23}\left(\tilde{x}^{i-1}_{12}\right)^T\right\} & E\left\{\tilde{x}^i_{23}\left(\tilde{x}^{i-1}_{23}\right)^T\right\} & E\left\{\tilde{x}^i_{23}\left(\tilde{x}^{i-1}_{13}\right)^T\right\} \\ E\left\{\tilde{x}^i_{13}\left(\tilde{x}^{i-1}_{12}\right)^T\right\} & E\left\{\tilde{x}^i_{13}\left(\tilde{x}^{i-1}_{23}\right)^T\right\} & E\left\{\tilde{x}^i_{13}\left(\tilde{x}^{i-1}_{13}\right)^T\right\} \end{bmatrix}$$

$$\mathbf{P}^{i-1,i} = \begin{bmatrix} E\left\{\tilde{x}^{i-1}_{12}\left(\tilde{x}^i_{12}\right)^T\right\} & E\left\{\tilde{x}^{i-1}_{12}\left(\tilde{x}^i_{23}\right)^T\right\} & E\left\{\tilde{x}^{i-1}_{12}\left(\tilde{x}^i_{13}\right)^T\right\} \\ E\left\{\tilde{x}^{i-1}_{23}\left(\tilde{x}^i_{12}\right)^T\right\} & E\left\{\tilde{x}^{i-1}_{23}\left(\tilde{x}^i_{23}\right)^T\right\} & E\left\{\tilde{x}^{i-1}_{23}\left(\tilde{x}^i_{13}\right)^T\right\} \\ E\left\{\tilde{x}^{i-1}_{13}\left(\tilde{x}^i_{12}\right)^T\right\} & E\left\{\tilde{x}^{i-1}_{13}\left(\tilde{x}^i_{23}\right)^T\right\} & E\left\{\tilde{x}^{i-1}_{13}\left(\tilde{x}^i_{13}\right)^T\right\} \end{bmatrix}$$

From the matrices we can observe that not only the cross-covariance terms are not zero, but also the variances as not zero, due to the relative relations among quantities,

$$x_{12} = x_{13} - x_{23}$$

Thus, the update of the respective estimates is at the expense of the same relative measurements which are a function of each other vehicle state estimates, meaning that the error covariances matrices become correlated, because they share common information (4.45).

The problem is to compute the degree of correlation, ignored by the traditional optimal estimation techniques. The consequence is to under-estimate the covariance matrix which causes instability problems, as shown in [3]. With the Covariance Intersection algorithm it is possible to combine random variables with unknown degrees of correlation, thus all off-diagonal elements of the error covariance ma-

trix can have values not known *a priori*, as opposed to the assumptions of the Kalman filter estimator. Therefore, the CI method and the Kalman filter are applicable to opposite classes of problems in the sense of the correlation between entities that are meant to be fused, as stressed by [54].

4.5.2 Covariance intersection

When we want to fuse the estimates in the vehicle i, $\hat{\mathbf{x}}^i_{trans}(k|k-1)$, with the estimates in the previous vehicle $i-1$, $\hat{\mathbf{x}}^{i-1}_{trans}(k|k-1)$, when there is some unknown degree of correlation between the state vectors, the covariance intersection algorithm as defined in [3], is as follows,

$$\begin{aligned}\hat{\mathbf{x}}^i_{trans}(k|k) &= \mathbf{P}^i(k|k)\Big\{w_c(\mathbf{P}^i(k|k-1))^{-1}\hat{\mathbf{x}}^i_{trans}(k|k-1) + (1-w_c)\left(\mathbf{P}^{i-1}(k|k-1)\right)^{-1} \\ &\quad \hat{\mathbf{x}}^{i-1}_{trans}(k|k-1)\Big\} \qquad (4.48)\\ (\mathbf{P}^i(k|k))^{-1} &= w_c(\mathbf{P}^i(k|k-1))^{-1} + (1-w_c)(\mathbf{P}^{i-1}(k|k-1))^{-1} \qquad (4.49)\end{aligned}$$

where $\mathbf{P}^i(k|k)$ is the error covariance matrix associated with state estimates $\hat{\mathbf{x}}^i_{trans}(k|k-1)$, and $\mathbf{P}^{i-1}(k|k)$ is the error covariance matrix associated with state estimates $\hat{\mathbf{x}}^{i-1}_{trans}(k|k-1)$, and $w_c \in [0,1]$, which represents a weight that is computed so as to minimize the trace of the resulting error covariance matrix. In the CI algorithm, the weight parameter w_c can be manipulated in such a way that modifies the weights assigned to variables $\hat{\mathbf{x}}^i_{trans}(k|k-1)$ and $\hat{\mathbf{x}}^{i-1}_{trans}(k|k-1)$, combined to improve the estimation of the variable $\hat{\mathbf{x}}^i_{trans}(k|k)$**.** Also the weighted parameter can be used to optimize the resulting error covariance estimate based on different performance criteria, such as minimizing the trace of the resulting error covariance matrix $\mathbf{P}^i(k|k)$, or its determinant, as suggested in [3]. It follows an analysis of the error covariances for the interval of w_c, based on minimizing the trace of the resulting error covariance matrix. It is assumed, for the sake of simplicity, that there is no correlation between the state estimates, meaning the covariances are zero.

1. For $0 < w_c < 1$, w_c is obtained so that the trace of the covariance of the resulting state vector, $\mathbf{P}^i(k|k)$ is minimized. Let us consider a vehicle filter i, with two state vector components,

$$\hat{\mathbf{x}}^i_{trans}(k|k-1) = \begin{bmatrix} \hat{x}_{ij} & \hat{x}_{i-1,j} \end{bmatrix}^T$$

   Vehicle filter $i-1$ also has the same state vector. The first component represents relative quantities between vehicle i and vehicle $i-1$, and it is observable in vehicle filter i, since it is updated by the local measurements represented by links in the *measurement* subgraph $¥^i$. The second component represents relative quantities between vehicle $i-1$ and vehicle j, and is not observable in vehicle

i but is observable in vehicle $i-1$. Thus, the error covariance matrices for the state vector on vehicle filter i and vehicle filter $i-1$ are, in an example, as follows

$$\begin{aligned}\mathbf{P}^{i}(k|k-1) &= \begin{bmatrix} 0.6 & 0 \\ 0 & 1.1 \end{bmatrix} \\ \mathbf{P}^{i-1}(k|k-1) &= \begin{bmatrix} 0.9 & 0 \\ 0 & 0.9 \end{bmatrix}\end{aligned}$$

By fusing the estimates from vehicle i and $i-1$, and using (4.49), the resulting error covariance matrix in vehicle filter i will be,

$$\mathbf{P}^{i}(k|k) = \begin{bmatrix} \frac{0.9}{0.5w_c+1} & 0 \\ 0 & \frac{4.95}{5.5-w_c} \end{bmatrix}$$

The value of the parameter w_c that minimizes the trace of $\mathbf{P}^{i}(k|k)$ is $w_c = 0.8213$, thus

$$\mathbf{P}^{i}(k|k) = \begin{bmatrix} 0.64 & 0 \\ 0 & 1.06 \end{bmatrix}$$

$\mathbf{P}^{i}(k|k)$ contains the intersection of the covariances of the two variables, $\hat{\mathbf{x}}^{i}_{trans}(k|k-1)$ and $\hat{\mathbf{x}}^{i-1}_{trans}(k|k-1)$, and is such that there will be a trade off among the variances of all the components. The analysis for the extreme cases in the interval, $w_c = 0$ and $w_c = 1$, follows.

2. For $w_c = 0$, Eq. (4.49) is reduced to,

$$\mathbf{P}^{i}(k|k) = \mathbf{P}^{i-1}(k|k-1)$$

and replacing the result in (4.48), and also with $w_c = 0$, the state vector equation is reduced to

$$\hat{\mathbf{x}}^{i}_{trans}(k|k) = \mathbf{P}^{i}(k|k)\left\{\left(\mathbf{P}^{i-1}(k|k-1)\right)^{-1}\hat{\mathbf{x}}^{i-1}_{trans}(k|k-1)\right\}$$

and knowing $\mathbf{P}^{i}(k|k) = \mathbf{P}^{i-1}(k|k-1)$, then the previous equation becomes,

$$\hat{\mathbf{x}}^{i}_{trans}(k|k) = \hat{\mathbf{x}}^{i-1}_{trans}(k|k-1)$$

The state vector update in vehicle filter i is going to be performed exclusively at the expense of the state estimate from the previous vehicle $i-1$. This situation will occur when combining two

state vectors with one common observable component, while the other component is observable in one filter but not in the other. The components that are observable will update the non-observable components of the other state vector. An example of a state vector with two components, follows

$$\hat{\mathbf{x}}^i_{trans}(k|k-1) = \begin{bmatrix} \hat{x}_{i,i-1} & \hat{x}_{i-1,j} \end{bmatrix}^T$$

where the first component represents relative quantities between vehicles i and $i-1$ and is computed locally by the Kalman filter through the local measurements represented by links in its *measurement* subgraph $\overleftarrow{\mathbf{Y}}^i$. The second component represents relative quantities between vehicle filters $i-1$ and j and is not observable in vehicle filter i. In vehicle filter $i-1$, both state vector component are observable. Thus, the state error variances are equal in both vehicles for the first component since they are computed by the same relative measurements, and the state error variance is higher for the second component in vehicle filter i, because this is a non-observable component,

$$\begin{aligned} \mathbf{P}^i(k|k-1) &= \begin{bmatrix} 0.6 & 0 \\ 0 & 1.2 \end{bmatrix} \\ \mathbf{P}^{i-1}(k|k-1) &= \begin{bmatrix} 0.6 & 0 \\ 0 & 0.6 \end{bmatrix} \end{aligned}$$

By fusing the estimates from vehicle filter i and $i-1$ in vehicle filter i, and using (4.49), the resulting error covariance matrix in vehicle filter i will be,

$$\mathbf{P}^i(k|k) = \begin{bmatrix} 0.6 & 0 \\ 0 & \frac{1.2}{2-w_c} \end{bmatrix}$$

The trace of the matrix is,

$$trace = 0.6 + \frac{1.2}{2-w_c}$$

and the minimum is computed as follows,

$$\begin{aligned} \frac{\partial trace}{\partial w_c} &= 0 \\ \frac{-1.2}{(w_c-2)^2} &= 0 \end{aligned}$$

Thus, in this case, the value of the parameter w_c that minimizes the trace of $\mathbf{P}^i(k|k)$ is indeed

$w_c = 0$, thus

$$\mathbf{P}^i(k|k) = \begin{bmatrix} 0.6 & 0 \\ 0 & 0.6 \end{bmatrix}$$

This means that the observation from the predecessor vehicle, where the variance was improved due to the measurements, will improve the non-observable component estimate of vehicle filter i. Despite the assumption made *i.e.,* a cross-covariance of zero, indicating no correlation between the components of the state vector results are also extendable to correlated state vector fusion. The main reason why the CI method is used is because there is a degree of correlation between the entities which is unknown and, in the case of EKF, its performance will degrade or even diverge.

3. For $w_c = 1$, Eq. (4.49) is reduced to,

$$\mathbf{P}^i(k|k) = \mathbf{P}^i(k|k-1)$$

and replacing the result in (4.48), and also with $w_c = 1$, the state vector equation is reduced to

$$\hat{\mathbf{x}}^i_{trans}(k|k) = \hat{\mathbf{x}}^i_{trans}(k|k-1)$$

Therefore, the state estimate from the previous vehicle $i-1$ does not contribute to the state vector update in vehicle i, which is going to be performed exclusively at the expense of the predicted state estimate in vehicle i from the previous step. Considering the same example as before but for the opposite situation: in vehicle filter i, both state vector components are observable, and in vehicle filter j only the first component is observable, leading to the following error covariance matrices,

$$\begin{aligned} \mathbf{P}^i(k|k-1) &= \begin{bmatrix} 0.6 & 0 \\ 0 & 0.6 \end{bmatrix} \\ \mathbf{P}^{i-1}(k|k-1) &= \begin{bmatrix} 0.6 & 0 \\ 0 & 1.2 \end{bmatrix} \end{aligned}$$

Thus, the resulting error covariance matrix in vehicle i, will be,

$$\mathbf{P}^i(k|k) = \begin{bmatrix} 0.6 & 0 \\ 0 & \frac{1.2}{w_c+1} \end{bmatrix}$$

and the value of the parameter w_c that minimizes the trace of $\mathbf{P}^i(k|k)$ is $w_c = 1$, thus

$$\mathbf{P}^i(k|k) = \begin{bmatrix} 0.6 & 0 \\ 0 & 0.6 \end{bmatrix}$$

This means that the variance from the previous vehicle filter $i-1$, higher than the variance from the vehicle filter i, does not contribute to the component estimate of vehicle filter i.

Covariance intersection parameter

From the previous discussion for $0 < w_c < 1$ the resulting error covariance matrix is a balance of the estimates obtained at the predecessor vehicle filter based on its measurements, and the estimates of state variables measured at the current vehicle. Since there are dependence among the variables some price is expected to be paid, since the improvement of the component errors is obtained at the expense of reduced accuracy of the remaining variables. CI provides a resulting covariance matrix which is a trade-off among the variance of all the components, the state estimate and the new "measurement" (*i.e.,* state estimates). Since the resulting covariance ellipsoid encloses the intersection region of the covariance ellipsoids of the variables that correspond to the state estimate and the new state "measurements", it is expected that the resulting covariance matrix will have some components reduced, at the expense of a higher variance for the other components, as shown in Fig. 4-11.

4.6 Translational State Vector Estimation Algorithm

The study carried out in Section 4.2 and the formation navigation algorithm proposed in Section 4.3 have shown the feasibility of using an EKF+CI filter to estimate adequately the relative position and translational relative velocity of a N-vehicle formation. In this section, we will summarize the algorithm for a translational relative state vector determined in a local frame from relative measurements.

Recall the notation of the full 6DoF state vector, as defined in Section 4.1.1 : $\mathbf{x}_{trans}$ denotes the translational vector, $\mathbf{x}_{trans} = \begin{bmatrix} \boldsymbol{\chi}^T & \dot{\boldsymbol{\chi}}^T \end{bmatrix}^T$. Throughout this section, for the sake of simplicity, we will use $\mathbf{x}$ to denote the translational state $\mathbf{x}_{trans}$.

4.6.1 Relative position and translational velocity

Knowing that the relative state vector is defined in a non-inertial frame, denoted by a local frame as defined in (4.4), and repeated here for convenience, with the local frame assumed to be the body frame

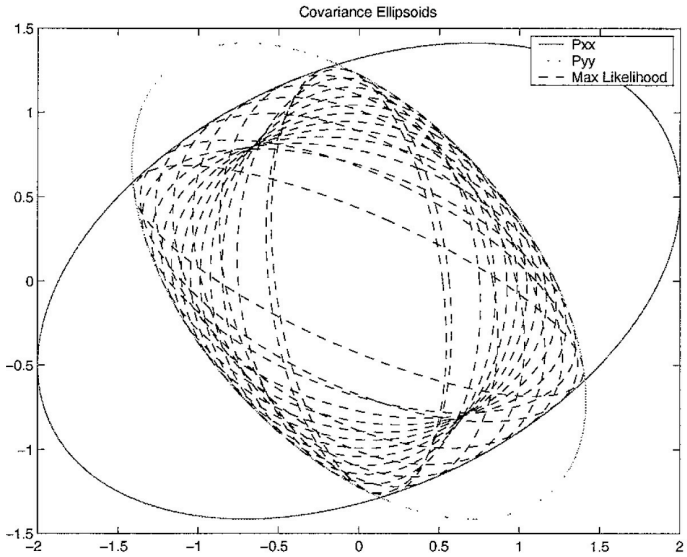

Figure 4-11: The error ellipse (for a given threshold) of the cross-covariance $\mathbf{P}_{xy}$ (dashed) for different cross-correlations of the linear combination of two randoms variables, x and y, always lies within the intersection of the error ellipses (for a given threshold) of its covariances $\mathbf{P}_{xx}$ (solid) and $\mathbf{P}_{yy}$ (doted) respectively, for a Maximum likehood estimator. Reprinted from [49].

b_i of vehicle i, abbreviated as i, then,

$$\mathbf{x}^i_{trans} = \begin{bmatrix} \left(\boldsymbol{\chi}^i_{ij}\right)^T & \left(\dot{\boldsymbol{\chi}}^i_{ij}\right)^T \end{bmatrix}^T$$

with $i, j = \{1, ..., N\}, i \neq j$, for each vehicle $i = 1, ..., N$.

Algorithm

To determine the components $\boldsymbol{\chi}^i_{ij}$ and $\dot{\boldsymbol{\chi}}^i_{ij}$ the usual filtering part of the EKF is employed, as explained in subsection 4.4.1, where the *measurement* subgraph Υ^i at time k is used. The nonlinear observation model and process model must be linearized for the i^{th} vehicle using the entire relative state estimate about time k to compute the error covariance matrix. At this instant of time, the non-observable components $\boldsymbol{\chi}^i_{jk}$ and $\dot{\boldsymbol{\chi}}^i_{kj}$ are just propagated through the equation of motion and only at time $k+1$ are then updated using the CI method. Thus, at the instant of time $k+1$ the state vector estimates from the predecessor vehicle are transmitted to the next vehicle, as established by the peer-to-peer *information flow* graph in subsection 4.2.3, and combined with the local state vector estimates, as shown in Fig. 4-12. In this way, the *measurement* subgraph is indirectly transmitted through the state vector estimates until the union of *measurement* subgraphs becomes the *formation estimation* graph, and thus the whole state vector is

determined. The algorithm is written in pseudo-code in Appendix B.2.

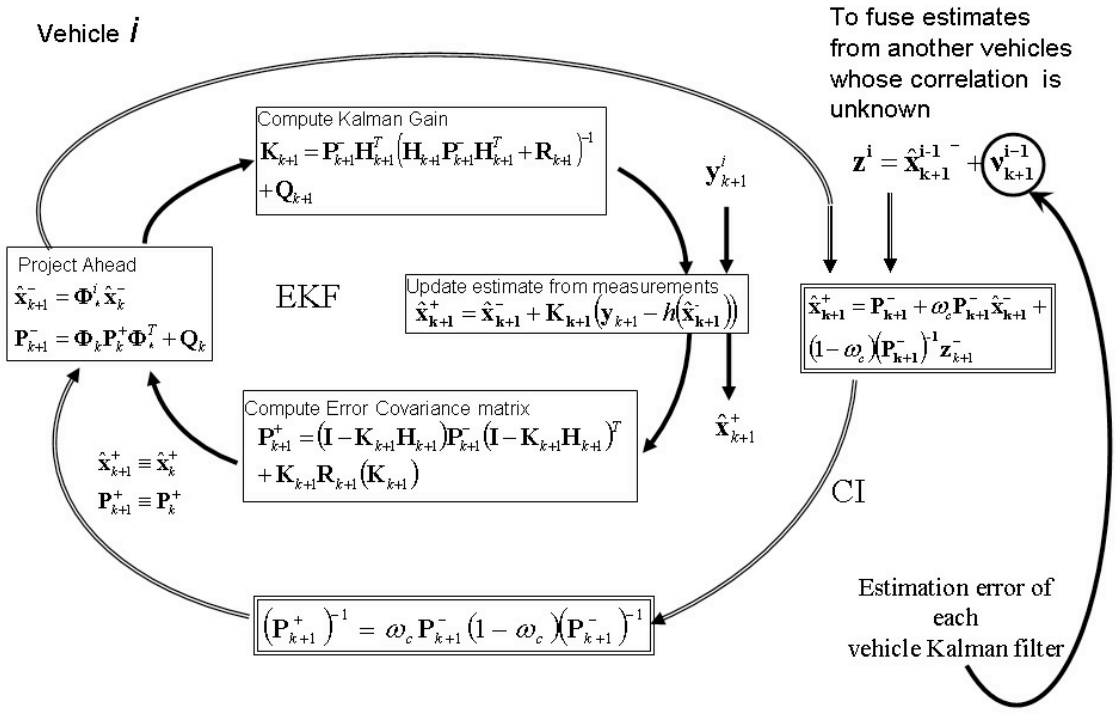

Figure 4-12: Block diagram of the EKF algorithm with CI used in the state vector estimation of relative position and relative translational velocity.

4.7 Rotational State Vector Estimation Algorithm

The rotational part of the state vector differs from the relative part of the state vector in terms of the operations allowed for the quaternions. Therefore, the equations that are used in EKF are revised adequately for the quaternion components of the state vector. On the other hand, the sensors used to update a rotational state vector are usually inertial, therefore it is simpler to use an absolute state vector, since the dynamic equations are decoupled, as well as the equations in the observation model, presented in the following subsection.

Using absolute attitude and angular velocity in the estimator eliminates the correlations between measurements, and the CI algorithm is no longer required. This section describes the EKF Fusion filter used to estimate the rotational components of the full state vector.

Recall the notation of the full 6DoF state vector, as defined in subsection 4.1.1 of the last chapter, where $\mathbf{x}_{rot}$ denotes the rotational vector, $\mathbf{x}_{rot} = \begin{bmatrix} q^T & \boldsymbol{\omega}^T \end{bmatrix}^T$. Throughout this section, for the sake of

simplifying the notation, we will use $\mathbf{x}$ to denote the rotational state $\mathbf{x}_{rot}$, dropping all the frames, and to refer to the rotational state vector i^{th} vehicle and to its vehicle i components as, $\mathbf{x}^i = \begin{bmatrix} \left(q^i\right)^T & \left(\boldsymbol{\omega}^i\right)^T \end{bmatrix}^T$.

4.7.1 Absolute attitude and angular velocity

The orientation part of the state vector for each vehicle includes its attitude expressed in quaternions, q, and the angular velocity, $\boldsymbol{\omega}$, was defined in (4.5), and it is repeated here for convenience,

$$\mathbf{x}_{rot}^{glob} = \begin{bmatrix} \left(q_{glob}^i\right)^T & \left(\boldsymbol{\omega}_i^{glob}\right)^T \end{bmatrix}^T \tag{4.50}$$

where $\boldsymbol{\omega}$ is the angular velocity vector with components ω_x, ω_y and ω_z, and the quaternion q is a three element vector with a possible fourth element depending on the other vector components, as defined in subsection 4.1.1. With respect to the quaternion, the subscript *glob* and superscript i represents the rotation between global and body i frames; with respect to the angular velocity, the subscript i represents the angular velocity of vehicle i, and the superscript represents the frame in which the angular velocity is written.

Related to the quaternion state, two problems arise:

- the *unit constraint*, $q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1$, has to be ensured because it represents the orthogonality in the rotation. This is difficult to maintain due to rounding errors, because of the dependence of the scalar quaternion component on the other components.
- the scalar quaternion leads to a singularity in the error covariance matrix and thus is not considered in the filter computation, as shown in [55] .

One way to overcome these, is to reduce the state vector to the vector part and rebuild the scalar part during the update of the quaternion estimate in the filtering part of the Kalman filter algorithm,

$$\hat{q}(k|k) = \hat{q}(k) \otimes \hat{q}(k|k-1) \tag{4.51}$$

where the quaternion $\hat{q}(k)$ is the update term for the rotation part of the state (see below). In Eq. (4.51), the scalar part of $\hat{q}(k)$ can be rebuilt according to

$$\hat{q}(k) = \begin{bmatrix} (\delta\hat{\mathbf{x}}_{\mathbf{q}})^T & \sqrt{1 - \|\delta\hat{\mathbf{x}}_{\mathbf{q}}\|} \end{bmatrix}^T$$

where $\delta\mathbf{x_q}$ is the vector part of the perturbation quaternion estimate, obtained as follows,

$$\begin{bmatrix} \delta\hat{\mathbf{x}}_{\mathbf{q}} \\ \delta\hat{\mathbf{x}}_{\boldsymbol{\omega}} \end{bmatrix} = \mathbf{K}\delta\hat{\mathbf{q}} \tag{4.52}$$

where $\mathbf{K}$ is the Kalman gain, $\delta\mathbf{x}_{\boldsymbol{\omega}}$ is the perturbation angular velocity estimate and $\delta\hat{\mathbf{q}}$ is the quaternion innovation or measurement residual obtained from the quaternions measurements and the quaternion prediction, as in the Kalman filter algorithm but in a special way due to quaternion operations, as follows,

$$\delta\hat{q}(k) = \begin{bmatrix} \delta\hat{\mathbf{q}} \\ \delta\hat{q}_4 \end{bmatrix} = \boldsymbol{\Xi}^T\left(\hat{q}\left(k|k-1\right)\right) q_{glob}(k) \tag{4.53}$$

where q_{glob} are the quaternion measurements.

Thus, the update operation becomes,

$$\begin{aligned} \hat{q}(k|k) &= \hat{q}(k) \otimes \hat{q}(k|k-1) \\ &= \begin{bmatrix} \left(\delta\hat{\mathbf{x}}_{\mathbf{q}}\right)^T & \sqrt{1 - \|\delta\hat{\mathbf{x}}_{\mathbf{q}}\|} \end{bmatrix}^T \otimes \hat{q}(k|k-1) \end{aligned} \tag{4.54}$$

The error quaternion, δq, a small perturbation of the quaternion attitude, is defined as follows,

$$q = \delta q \otimes \hat{q} <=> \delta q = q \otimes \hat{q}^{-1} \tag{4.55}$$

where $\hat{q}$ is the estimated quaternion. According to the definition of the quaternion in terms of the Euler axis/angle given by Eq. (4.7), and since δq is a small perturbation,

$$\delta q = \begin{bmatrix} \overrightarrow{\mathbf{u}}_1 \sin\left(\frac{\delta\alpha}{2}\right) \\ \overrightarrow{\mathbf{u}}_2 \sin\left(\frac{\delta\alpha}{2}\right) \\ \overrightarrow{\mathbf{u}}_3 \sin\left(\frac{\delta\alpha}{2}\right) \\ \cos\left(\frac{\delta\alpha}{2}\right) \end{bmatrix} \approx \begin{bmatrix} \delta q_1 \\ \delta q_2 \\ \delta q_3 \\ 1 \end{bmatrix} = \begin{bmatrix} \delta\mathbf{q} \\ 1 \end{bmatrix} \tag{4.56}$$

Therefore, the rotational part of the N-vehicle formation absolute state vector, will be given by the shortened version

$$\mathbf{x}_{rot'}^{glob} = \begin{bmatrix} \left(q_1\right)_{glob}^{i} & \left(q_2\right)_{glob}^{i} & \left(q_3\right)_{glob}^{i} & \left(\omega_x\right)_{i}^{glob} & \left(\omega_y\right)_{i}^{glob} & \left(\omega_z\right)_{i}^{glob} \end{bmatrix}^T \tag{4.57}$$

where q_4, required for the state vector, can be rebuilt through Eq. (4.54).

Algorithm

Local extended Kalman filter For each vehicle $i = 1, ..., N$, the usual filtering part of the EKF is used, where the local measurement update and estimation is based on the *measurement* subgraph $¥^i$. The linear observation model is used for the error covariance matrix but the update of the local state estimate must be performed carefully as discussed before. The error innovation in the filtering equations of the Kalman filter is usually computed as the difference between the actual measurements and those predicted by the observation model. However, because quaternions must respect the rotation operation, according to (4.10)-(4.11), one must compute the innovation differently for the quaternion components of the state. Denoting the actual global measurement by $q^i_{glob}(k)$ and its prediction by $\hat{q}^i(k|k-1)$, the innovation or measurement residual becomes,

$$\begin{bmatrix} \delta\hat{\mathbf{q}}^i(k) \\ 1 \end{bmatrix} = q^i_{glob}(k) \otimes \left(\hat{q}^i\left(k|k-1\right)\right)^{-1}$$

where $\hat{q}^{-1} = \begin{bmatrix} -\hat{\mathbf{q}}^T & \hat{q}_4 \end{bmatrix}^T$, or with after some algebra manipulation,

$$\delta\hat{q}^i(k) = \mathbf{\Xi}^T\left(\hat{q}^i\left(k|k-1\right)\right) q^i_{glob}(k)$$

The complete state vector is split in two parts, corresponding to the angular velocity and the quaternion, *e.g.*, $\mathbf{x}^i(k|k) = \begin{bmatrix} q^i\left(k|k\right) \\ \boldsymbol{\omega}^i\left(k|k\right) \end{bmatrix}$ vehicle i components. For the sake of simplifying the notation, the frames from (4.50) were dropped and the superscript i refers to vehicle i and not to global frame as stated in Eq. (4.50). Therefore, the state estimate perturbation, computed as the innovation multiplied by the Kalman gain can be expressed as

$$\begin{bmatrix} \delta\hat{\mathbf{x}}^i_{\mathbf{q}}(k) \\ \delta\hat{\mathbf{x}}^i_{\boldsymbol{\omega}}(k) \end{bmatrix} = \mathbf{K}^i(k)\delta\hat{\mathbf{q}}^i(k)$$

Hence, the local state estimate update is performed according to,

$$\hat{\mathbf{x}}^i(k|k) = \begin{bmatrix} \hat{q}^i\left(k|k\right) \\ \hat{\boldsymbol{\omega}}^i\left(k|k\right) \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \delta\hat{\mathbf{x}}^i_{\mathbf{q}}(k) \\ \sqrt{1 - \left\|\delta\hat{\mathbf{x}}^i_{\mathbf{q}}(k)\right\|} \end{bmatrix} \otimes \hat{q}^i\left(k|k-1\right) \\ \hat{\boldsymbol{\omega}}^i\left(k|k-1\right) + \delta\hat{\mathbf{x}}^i_{\boldsymbol{\omega}}(k) \end{bmatrix}$$

In the prediction, the nonlinear process model $\frac{d}{dt}\hat{\mathbf{x}}^i(t) = f(t, \hat{\mathbf{x}}^i(t))$, approximated by a fourth-order four-stage Runge-Kutta method, is used to update local state estimate perturbation (according to Wertz

[53]). The linearized process model for the i^{th} vehicle, uses the entire relative state estimate $\hat{\mathbf{x}}^i$ with $i \in [1, N]$. This model assumes that the angular velocity is constant over the sampling time.

EKF Fusion Since the state vector includes quaternion components, the Covariance Intersection algorithm is no longer applicable, as the weighted sum of both estimates is meaningless for quaternions. Furthermore, because we use the absolute attitude, no correlations in measurements occur, in contrast to the relative state estimation case. Therefore, the alternative is to fuse the state estimates from each vehicle using an EKF.

Considering the generalized observation model for inertial measurements,

$$\mathbf{y}^i(k) = \mathbf{H}^i \mathbf{x}_{rot}(k|k-1) + \mathbf{v}^i(k)$$

where $\mathbf{x}_{rot}(k)$ is defined in Eq. (4.57), and for the sake of simplifying the notation, the frames were dropped and the superscript i refers to vehicle i and not to global frame as stated in the equation. The "measurements" in the observation equation are now the state estimates from the predecessor vehicle without the components that are locally updated by the inertial sensor in vehicle i. Thus, the covariance matrix of the measurement is now the error covariance matrix from the predecessor state estimates, $E\left\{\mathbf{v}^i(k)\left(\mathbf{v}^i(k)\right)^T\right\} = \mathbf{P}^{i-1}(k|k-1)$, which must also be communicated. Taking advantage of $\mathbf{P}$ symmetry, only the lower or upper triangular part plus the diagonal need to be transmitted. For the fusion of vehicle i and the previous vehicle $i-1$ information, the usual EKF equations are applied such that the observation is now the predecessor vehicle state estimate, $\mathbf{z}^i(k) = \mathbf{x}^{i-1}(k|k-1) + \mathbf{v}^{i-1}(k)$, and the observation noise is the error covariance matrix from the predecessor vehicle, $i-1$. Therefore, the state estimate perturbation is performed according to,

$$\begin{bmatrix} \delta\hat{\mathbf{x}}_q^i(k) \\ \delta\hat{\mathbf{x}}_{\boldsymbol{\omega}}^i(k) \end{bmatrix} = \mathbf{K}^i(k) \begin{bmatrix} \delta\hat{q}^i(k) \\ \delta\hat{\boldsymbol{\omega}}^i(k) \end{bmatrix} \tag{4.58}$$

with,

$$\begin{aligned} \delta\hat{q}^i(k) &= \hat{q}^{i-1}(k|k-1) \otimes \left(\hat{q}^i(k|k-1)\right)^{-1} = \boldsymbol{\Xi}^T\left(\hat{q}^i(k|k-1)\right)\hat{q}^{i-1}(k|k-1) \\ \delta\hat{\boldsymbol{\omega}}^i(k) &= \hat{\boldsymbol{\omega}}^{i-1}(k|k-1) - \hat{\boldsymbol{\omega}}^i(k|k-1) \end{aligned}$$

Thus, the local state estimate update becomes

$$\hat{\mathbf{x}}^i(k|k) = \begin{bmatrix} \hat{q}^i(k|k) \\ \hat{\boldsymbol{\omega}}^i(k|k) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Xi}\left(\hat{q}^i(k|k-1)\right)\delta\hat{\mathbf{x}}_q^i(k) \\ \hat{\boldsymbol{\omega}}^i(k|k-1) + \delta\hat{\mathbf{x}}_{\boldsymbol{\omega}}^i(k) \end{bmatrix}$$

To summarize, the filtering part of the estimator algorithm that computes the inertial attitude and the inertial angular velocity is divided into two parts: Kalman filtering and the Kalman filtering fusion method, depending on the source of the measurements. As soon as a measurement of the sensor arrives, the filtering part is computed and the result is the state vector estimation at that instant, which is then sent to the next vehicle, shown in Fig. 4-13. At the next time, the predecessor state vector is combined with the local state vector through the Kalman filter fusion (denoted by Kalman Filter Fusion), now adapted for the fact that the observations for this filter are now the predecessor state estimates, resulting in a new state vector estimate. The algorithm is detailed in Appendix B.3.

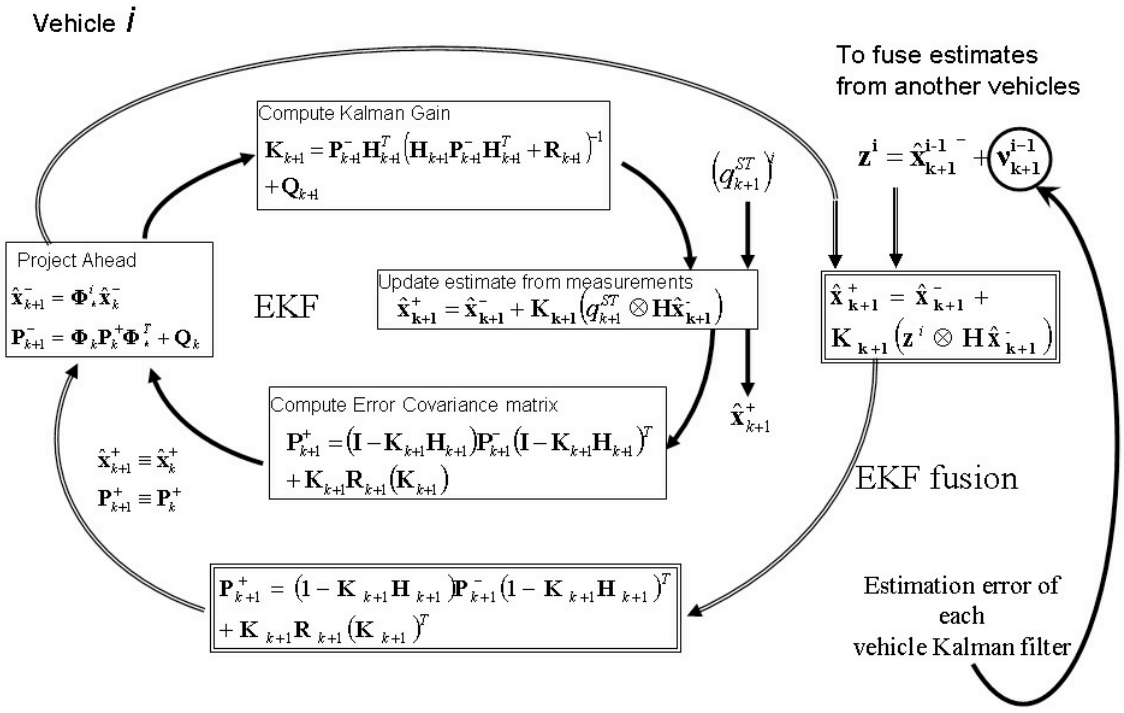

Figure 4-13: Block diagram of the EKF algorithm with EKF Fusion used in the state vector estimation of inertial attitude and angular velocity.

4.8 Summary

The determination of the 6DoF state vector (relative position, translational velocity, absolute attitude and angular velocity) estimates for a N-vehicle formation represented by the *formation estimation* graph, presented in this chapter, relies on the key idea that the filter estimation algorithm proposed uses a *measurement* graph together with a peer-to-peer *information flow* graph, in other words with relative

measurements only available locally and low communication within the formation. The design of the Navigation algorithm is based on:

- **for the translational components:** a combination of an EKF and a CI algorithm at each vehicle, with inter-vehicle measurements obtained from a local sensor in the *measurement* graph. Each vehicle communicates its most recent state estimate to its successor vehicle in the peer-to-peer *information flow* graph. The filtering part of the state estimation process is performed at alternate sampling times, by an EKF –when local measurements are used– and a CI algorithm when observations (*i.e.,* the state estimates) come from another vehicle. The prediction part of the state estimation process is performed by the EKF;

- **for the rotational components:** a so-called EKF Fusion at each vehicle, and on absolute attitude measurements obtained from a global sensor, according to the *measurement* graph. Each vehicle communicates its most recent state estimate to its successor vehicle in the peer-to-peer *information flow* graph. The filtering part of the state estimation process is performed, at alternate sampling times, by an EKF –using local measurements– and another EKF –when observations (*i.e.,* the state estimates) come from another vehicle. The prediction part of the state estimation process is performed by an EKF.

The resulting architecture achieves improvements in estimating the full relative components that represent the full vehicle inter-relations within the formation and the angular variables that characterize each vehicle. These improvements are achieved in less than one complete cycle, in a decentralized topology, in the sense that there is no master filter, so each vehicle has access, not to all measurements in the formation, only to those sensed locally. The designed filter overcomes the correlation problem, due to estimated information in two vehicles being shared. The quaternion operations are also taken into account in the filter equations.

Chapter 5

EXTENDED KALMAN FILTER ANALYSIS ISSUES

5.1 Introduction

This chapter is devoted to the study of the EKF and to the performance analysis of the solution obtained by the EKF when applied to the nonlinear problem of the full state estimation for N-vehicle in formation discussed in previous chapters.

In the first section 5.2, the filter stability for linear and nonlinear systems and their relationship to observability is presented in subsection 5.2.2. A linear time-invariant system is given as a basic example to gain more insight into the influence of the process and measurement error covariance matrices. Also, their influence on the solution of the algebraic Riccati equation is addressed. In Section 5.3, the Posterior Cramer-Rao Lower Bound (PCRLB), which determines the optimal performance lower bound, is presented for nonlinear systems. The PCRLB allows evaluation of whether the suboptimal EKF is close to the optimal estimate or not. This study also includes, in Section 5.4, the consistency test, very important for nonlinear systems due to the approximations made, which tests for possible bias and for whether the EKF actual MSE matrix matches the covariance matrix assumptions.

5.2 Stability

Observability is an important concept for the stability of optimal estimation problems, especially in stochastic systems, where observability is related to the convergence of the covariance matrix.

In open loop, *i.e.,* when the estimator is not in a closed-loop with a controller, the observability

plays an important role in the estimation problem. According to Gelb [56] and Bar-Shalom [57], - "*A continuous-time (deterministic) system is completely observable if its initial state can be fully an uniquely recovered from its output, observed over a finite interval, and the knowledge of the input.*" -. So if a system is observable at time $k_1 > k_0$ if it is possible to determine the state $x(k_0)$ by observing $y(k)$ in the interval (k_0, k_1). If all states $x(k)$ corresponding to all $y(k)$ are observable, the system is completely observable. This means that there should be enough information in the sensor measurements, such that the state vector can be determined distinctly at each time k.

The observability in stochastic systems is verified if the covariance matrix associated with the conditional probability density function of the state vector $\mathbf{x}(k)$ given $\mathbf{y}(k)$, $p(\mathbf{x}(k)|\mathbf{y}(k))$, remains bounded as the time k approaches infinity, $k \to \infty$. In practice, this means that the incoming sensor information is such that the uncertainty around the state vector is bounded. According to Bar-Shalom [57], observability of the system guarantees the positive definiteness of the inverse covariance matrix, which makes it invertible, *i.e.,* it guarantees the existence (boundeness) of the covariance matrix, which is given by the solution of the Riccati equation.

For nonlinear systems there is no rule for determining observability, since the asymptotic properties are no longer valid, thus the best option is to compute the observation matrix for nonlinear systems and ensure the consistency of the algorithms with practical tests, as in Section 5.4.

5.2.1 Riccati difference equation

The evolution of the error covariance matrix $\mathbf{P}(k|k)$ which will determine the Kalman Gain, is the solution of the *difference matrix Riccati equation* (RDE), here written for discrete time-variant systems,

$$\begin{aligned}\mathbf{P}(k|k) &= \mathbf{F}(k)\left[\mathbf{1} - \mathbf{P}(k|k-1)\mathbf{H}^T(k)\left[\mathbf{H}(k)\mathbf{P}(k|k-1)\mathbf{H}^T(k) + \mathbf{R}(k)\right]^{-1}\mathbf{H}(k)\right] \\ &\quad \cdot\mathbf{P}(k|k-1)\mathbf{F}^T(k) + \mathbf{Q}(k)\end{aligned}$$

or

$$\begin{aligned}\mathbf{P}(k|k) &= \mathbf{F}(k)\mathbf{P}(k|k-1)\mathbf{F}^T(k) - \mathbf{F}(k)\mathbf{P}(k|k-1)\mathbf{H}^T(k)\left[\mathbf{H}(k)\mathbf{P}(k|k-1)\mathbf{H}^T(k) + \right. \\ &\quad \left. + \mathbf{R}(k)\right]^{-1}\mathbf{H}(k)\mathbf{P}(k|k-1)\mathbf{F}^T(k) + \mathbf{Q}(k) \qquad (5.1)\end{aligned}$$

where $\mathbf{F}(k)$ and $\mathbf{H}(k)$ are the Jacobian of $f(.)$ and $h(.)$, given by Eqs. (4.39) and (4.33) respectively.

Alternatively, it is shown in [57], that there is a recursive expression for the inverse error covariance matrix, $\mathbf{P}^{-1}(k|k)$,

$$\mathbf{P}^{-1}(k|k) = \mathbf{P}^{-1}(k|k-1) + \mathbf{H}^T(k)\mathbf{R}^{-1}(k)\mathbf{H}(k) \qquad (5.2)$$

which is known as the information matrix filter. The information matrix filter is interpreted as the sum of information up to step $k-1$, plus the new information about the state vector obtained from measurements at time k, according to [57].

In time-invariant systems it is possible to compute of-line the steady-state error covariance matrix, which is independent of its initial value, and is the solution of the *algebraic Riccati equation*,

$$\mathbf{P}(k|k) = \mathbf{FP}\left(k|k-1\right)\mathbf{F}^T - \mathbf{FP}\left(k|k-1\right)\mathbf{H}^T\left[\mathbf{HP}\left(k|k-1\right)\mathbf{H}^T + \mathbf{R}\right]^{-1}\mathbf{HP}\left(k|k-1\right)\mathbf{F}^T + \mathbf{Q} \quad (5.3)$$

yielding to the steady-state gain for the Kalman filter,

$$\lim_{\mathbf{k}\longrightarrow\infty}\mathbf{K}(k) = \mathbf{P}_{\infty}^{-}\mathbf{H}^T\left[\mathbf{HP}_{\infty}^{-}\mathbf{H}^T + \mathbf{R}\right]^{-1}$$

with $\mathbf{P}_{\infty}^{-}$ from Eq. (5.3),

$$\lim_{\mathbf{k}\longrightarrow\infty}\mathbf{P}\left(k|k-1\right) = \mathbf{P}_{\infty}^{-} \quad (5.4)$$

For nonlinear systems, the solutions for $\mathbf{P}(k|k-1)$ may be unstable because of the errors resulting from linearization errors.

5.2.2 Linear time-invariant example

In this subsection, a KF is applied to a LTI scalar example in order to provide some insight into the influence of the variances of the measurement error and of the process error in the error covariance matrix and thus the accuracy obtained by the filter.

Consider an example of a scalar, linear time-invariant system with the following dynamic,

$$\begin{aligned} x(k+1) &= Ax(k) + w(k) \\ y(k) &= Cx(k) + v(k) \end{aligned}$$

where $k \in \mathbb{N}$ is the time step, $x(k) \in \mathbb{R}$, is the state, $y(k) \in \mathbb{R}$, is the output, $A = 0.2$, and $C = 1$, and $w(k)$, $v(k)$ are assumed to be uncorrelated with zero mean white noise with known variance,

$$\begin{aligned} E\left\{w(k)w^T(k)\right\} &= Q(k) \\ E\left\{w(k)\right\} &= 0 \\ E\left\{v(k)v^T(k)\right\} &= R(k) \\ E\left\{v(k)\right\} &= 0 \end{aligned} \quad (5.5)$$

with,

$$Q = \sigma_Q^2 \tag{5.6}$$

$$R = \sigma_R^2 \tag{5.7}$$

A discrete-time Kalman filter is given by the following equations,

Initial conditions

It is assumed that the filter knows exactly the system since here convergency is not an issue.

$$\begin{aligned} \hat{x}(0|0) &= x(0) \\ P(0|0) &= 1 \end{aligned}$$

Filtering

The innovation is,

$$\begin{aligned} v(k) &= y(k) - C\hat{x}(k|k-1) \\ &= y(k) - \hat{x}(k|k-1) \end{aligned}$$

thus the state estimate can be computed as,

$$\begin{aligned} \hat{x}(k|k) &= \hat{x}(k|k-1) + K(k)v(k) \\ &= \hat{x}(k|k-1) + K(k)v(k) \end{aligned}$$

with Kalman Gain,

$$\begin{aligned} K(k) &= P(k|k-1)C^T \left[CP(k|k-1)C^T + R(k)\right]^{-1} \\ &= \frac{\sigma^2_{P(k|k-1)}}{\sigma^2_{P(k|k-1)} + \sigma_R^2} = \frac{1}{1 + \frac{\sigma_R^2}{\sigma^2_{P(k|k-1)}}} \end{aligned} \tag{5.8}$$

where $P(k|k-1) = \sigma^2_{P(k|k-1)}$ and the update equations for state estimate error variance,

$$P\left(k|k\right) = \left[1 - K(k)C\right] P\left(k|k-1\right)$$

$$\sigma^2_{P(k|k)} = \left[1 - \frac{\sigma^2_{P(k|k-1)}}{\sigma^2_{P(k|k-1)} + \sigma_R^2}\right] \sigma^2_{P(k|k-1)} = \frac{\sigma^2_{P(k|k-1)}\sigma_R^2}{\sigma^2_{P(k|k-1)} + \sigma_R^2} = \frac{\sigma^2_{P(k|k-1)}}{\frac{\sigma^2_{P(k|k-1)}}{\sigma_R^2} + 1} \tag{5.9}$$

Prediction

The state prediction equation is,

$$\begin{aligned} \hat{x}(k+1|k) &= A\hat{x}(k|k) \\ &= 0.2\hat{x}(k|k) \end{aligned}$$

The error variance is predicted through the following equations,

$$\begin{aligned} P(k+1|k) &= AP(k|k)A^T + Q(k) \\ \sigma^2_{P(k+1|k)} &= 0.04\sigma^2_{P(k|k-1)} + \sigma^2_Q \end{aligned} \tag{5.10}$$

For each simulation, where the step time $k = \{\mathbb{N} : 1 \leq k \leq 1000\}$, the standard deviations of the measurement and process noises, σ_R, σ_Q, change in the integer interval $\{n \in \mathbb{N} : 1 \leq n \leq 1000\}$, according to Table 5.1.

1^{st} case	$\{\sigma_R \in \mathbb{N} : 1 \leq \sigma_R \leq 1000\}$, keeping $\sigma_Q$constant, $\sigma_Q = 10$
2^{nd} case	$\{\sigma_Q \in \mathbb{N} : 1 \leq \sigma_Q \leq 1000\}$, keeping σ_R constant, $\sigma_R = 10$

Table 5.1: 1^{st} case: Range value for the variance of the measurement noise, while the variance of the process noise is constant. 2^{nd} case: Range value for the variance of the process noise, while the variance of the measurement noise is constant.

At the end of each simulation (1000 time steps), the Root Mean Square (RMS) of the estimation error is computed according to,

$$RMS = \sqrt{\frac{1}{1000}\sum_{k=1}^{1000}\left(x(k) - \hat{x}(k)\right)^2} \tag{5.11}$$

and for the two cases, each of which corresponds to 1000 simulations, the mean of the RMS for all experiments is computed,

$$\overline{RMS} = \frac{1}{1000}\sum_{n=1}^{1000} RMS(n) \tag{5.12}$$

The steady-state posterior error variance, *i.e.,* the $\lim_{\mathbf{k}\longrightarrow\infty} \mathbf{P}\left(k|k-1\right) = \mathbf{P}^-_\infty$ is obtained and shown in Figure 5-2. Figure 5-1 shows the results of the Kalman filter, the RMS of the state estimate, and Fig. 5-2 the steady-sate solution to the Riccati equation as σ_Q and σ_R change. In both graphics, the top figure shows the evolution (of RMS in Fig. 5-1 and of the error standard deviation in Fig. 5-2 obtained from KF) when the process noise standard deviation is fixed to 10 and the standard deviation of the measurements takes the values, $1 \leq \sigma_R \leq 1000$. Likewise, the bottom figures show the evolution

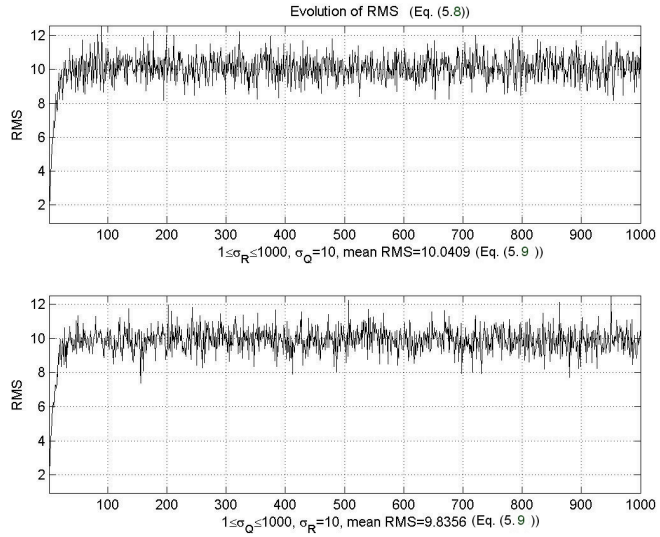

Figure 5-1: The plot shows the results (RMS) obtained by the KF for a scalar system. The top plot shows the evolution of RMS obtained by the KF when the process error is constant, $\sigma_Q^2 = (10)^2$, and the variance of the measurements ranges through the values presented in the table. The bottom figure shows the evolution of RMS obtained from KF when the error in the measurements is constant, $\sigma_R^2 = (10)^2$, and the variance of the process noise ranges through the values in the table 5.1. The mean RMS values are computed according to equation 5.12.

when the measurement noise standard deviation is fixed to 10 and the variance of the process takes the values, $1 \leq \sigma_Q \leq 1000$. It is interesting to notice that in both plots the standard deviation of the error variance, obtained from Eq. (5.9) in the end of each simulation, is the standard deviation of the lowest of the process and measurement standard deviations. When one of the variances grows, the filter state error variance converges to the smallest variance. That is the reason why the standard deviation of $\mathbf{P}_\infty^-$ remains constant, despite the growth of the standard deviation of the measurement error in the top plot, or the growth of the standard deviation of the process error in the bottom plot. This can be confirmed by the equations derived for this example. In the first case, as σ_R^2 grows, from Eq. (5.9) σ_P^2 is constant, and when predicted by Eq. (5.10) the term σ_Q^2 (which is constant) is dominating. In the second case, the growth of σ_Q^2 in (5.9) is cancelled by the first term of the equation, $0.04\sigma_P^2$. So, the accuracy of the state estimates depends on the process or measurement error with reduced uncertainty.

5.2.3 Observability of nonlinear systems

The stability conditions for the KF are no longer sufficient for the EKF, since the estimate computed is an approximate conditional mean, and its covariance matrix, computed by the filter $\mathbf{P}(k|k)$, is a MSE

matrix and not the exact covariance matrix. Thus, the approximations resulting from linearizing the system can lead the filter to diverge due to unbounded estimation errors. So, the observability for a linear system enables the KF estimation error $E\{\|\widetilde{\mathbf{x}}(k|k)\|^2\} = E\{(\mathbf{x}(k) - \hat{\mathbf{x}}(k|k))^2\} = var\left[\widetilde{\mathbf{x}}(k|k)\right]$ to be bounded, thus it guarantees an asymptotic solution to the covariance matrix. However, in the nonlinear case only local observability exists and thus no asymptotic properties can be inferred for the covariance matrix. Nevertheless, under some conditions, in the case of small initial estimation error and small noise, the estimation error of the discrete-time EKF remains bounded [58], [57]. We will first investigate the observability of linear (time-varying) systems and then present the results for the nonlinear case.

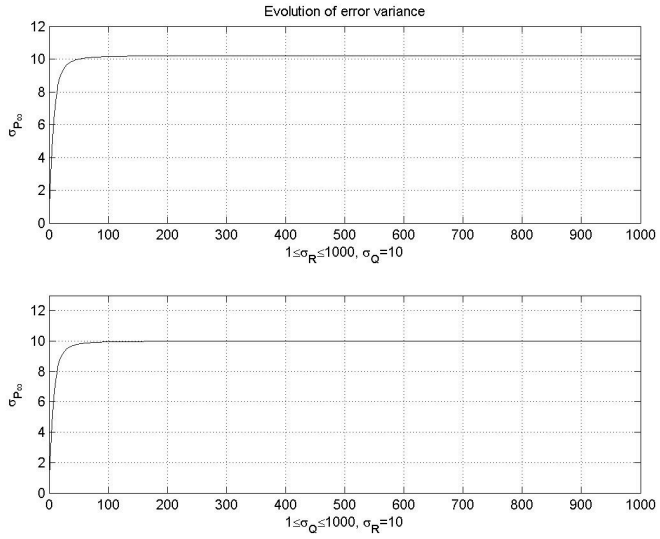

Figure 5-2: The plots show the results (error variance) obtained by the KF for a scalar system. The top plot shows the evolution of error variance obtained by the KF when the process error is constant, $\sigma_Q^2 = (10)^2$, and the variance of the measurements ranges through the values presented in the table. The bottom figure shows the evolution of error variance obtained from KF when the error in the measurements is constant, $\sigma_R^2 = (10)^2$, and the variance of the process noise ranges through the values in the table 5.1.

Linear time-varying systems

Deterministic linear dynamic systems can be described in discrete-time by the following state equations,

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{\Phi}(k+1,k)\mathbf{x}(k) + \mathbf{w}(k) \\ \mathbf{y}(k) &= \mathbf{C}(k)\mathbf{x}(k) + \boldsymbol{\nu}(k) \text{ with } k = 1,\, 2,\, ... \end{aligned} \tag{5.13}$$

the observability is given in terms of the *observability Gramian* [52] or stochastic observability matrix,

$$\mathbf{\Psi}(k_1, k_0) \triangleq \sum_{k=k_0}^{k_1} \mathbf{\Phi}^T(k, k_0)\mathbf{C}^T(k)\mathbf{R}^{-1}(k)\mathbf{C}(k)\mathbf{\Phi}(k, k_0) \tag{5.14}$$

where $k \in \mathbb{N}^0$ is the discrete-time, and $\mathbf{\Phi}$ is the $(n_x \times n_x)$ state transition matrix , defined for all $k \geq k_0$,

$$\mathbf{\Phi}(k, k_0) \triangleq e^{\int_{t_{k_0}}^{t_k} \mathbf{A}(\tau)d\tau}$$

with the following properties,

$$\begin{aligned} \mathbf{\Phi}(k, k) &= \mathbf{1} \\ \mathbf{\Phi}(k, k_0) &= \mathbf{A}(k-1)\mathbf{A}(k-2)...\mathbf{A}(k_0) \end{aligned}$$

where $\mathbf{A}(k)$ is the system matrix from the linear dynamic system.

If no process noise is considered, $\mathbf{w}(k) \equiv 0$, and assuming that $\mathbf{P}^{-1}(k|k)$ is not zero, the observability Gramian is related to the KF covariance matrix by,

$$\mathbf{P}^{-1}(k+1|k) \triangleq \mathbf{\Phi}^T(k+1, k)\mathbf{P}^{-1}(k|k)\mathbf{\Phi}(k+1, k) + \mathbf{\Psi}(k+1, k)$$

where $\mathbf{P}^{-1}(k+1|k)$ is the information matrix. It can be seen that as $k \longrightarrow \infty$ the information matrix can also tend to infinity, $\mathbf{P}^{-1} \longrightarrow \infty$, which results in the covariance matrix tending to zero.

Still regarding linear time-varying systems but now without measurement noise, $\boldsymbol{\nu}(k) \equiv 0$, the observability Gramian changes to,

$$\mathbf{\Psi}(k_1, k_0) = \sum_{k=k_0}^{k_1} \mathbf{\Phi}^T(k, k_0)\mathbf{C}^T(k)\mathbf{C}(k)\mathbf{\Phi}(k, k_0)$$

Instead of the verifying the condition observability Gramian, $\mathbf{\Psi}(k_1, k_0) > 0$, the observability can instead be verified according to $\mathbf{U}(\mathbf{x}(k))$, as cited by [59], and according to,

$$rank[\mathbf{C}(k)\mathbf{\Phi}(k, k_0)] = rank[\mathbf{U}(\mathbf{x}(k))] = n_x \tag{5.15}$$

where the matrix $\mathbf{U}$ is computed from the Gramian observability as follows,

$$\begin{aligned}\boldsymbol{\Psi}(k_1,k_0) \quad = \quad & \\ & \left[\begin{array}{cccc} \boldsymbol{\Phi}^T(k_1,k_0)\mathbf{C}^T(k_1) & \boldsymbol{\Phi}^T(k_1-1,k_0)\mathbf{C}^T(k_1-1) & \cdots & \boldsymbol{\Phi}^T(k,k_0)\mathbf{C}^T(k) \quad \cdots \end{array}\right. \\ & \left.\begin{array}{cc} \boldsymbol{\Phi}^T(k_0+1,k_0)\mathbf{C}^T(k_0+1) & \mathbf{C}^T(k_0) \end{array}\right] \left[\begin{array}{c} \mathbf{C}(k_0) \\ \mathbf{C}(k_0+1)\boldsymbol{\Phi}(k_0+1,k_0) \\ \vdots \\ \mathbf{C}(k)\boldsymbol{\Phi}(k,k_0) \\ \vdots \\ \mathbf{C}(k_1-1)\boldsymbol{\Phi}(k_1-1,k_0) \\ \vdots \\ \mathbf{C}(k_1)\boldsymbol{\Phi}(k_1,k_0) \end{array}\right]\end{aligned}$$

or

$$\boldsymbol{\Psi}(k+n_x-1,k) = \mathbf{U}^T\left(\mathbf{x}(k)\right)\mathbf{U}\left(\mathbf{x}\left(k\right)\right)$$

where $\mathbf{U}\left(\mathbf{x}\left(k\right)\right) = \mathbf{C}(k)\boldsymbol{\Phi}(k,k_0)$. Thus, the observability of the linear system and therefore the convergence of the error covariance matrix can be verified through the condition (5.15).

Stability of nonlinear time-varying systems

For nonlinear systems as defined in subsection (4.1.2) and detailed in subsection B.1, the observation matrix is nonlinear and takes the form,

$$\mathbf{U}\left(\mathbf{x}\left(k\right)\right) = \left[\begin{array}{l} \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(k)} \\ \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(k+1)} \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(k)} \\ \vdots \\ \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(k+n_x-1)} \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(k+n_x-2)} \cdots \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(k)} \end{array}\right] \tag{5.16}$$

where $\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}(k|k)$ and $\mathbf{H}(k) = \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(k)}$, $\mathbf{F}\left(k\right) = \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(k)}$. With $k_0 = k$ and $k_1 = k_0 + n_x - 1$ the Gramian observability is,

$$\boldsymbol{\Psi}(k+n_x-1,k) = \mathbf{U}^T\left(\mathbf{x}(k)\right)\mathbf{U}\left(\mathbf{x}\left(k\right)\right) \tag{5.17}$$

According to [60], [61] or [62], a nonlinear system without process noise (*i.e.,* with $\mathbf{w}(k) \equiv 0$) is observable on κ if there is a $\mathbf{x}_{\boldsymbol{\Psi}}\left(k\right) \in \kappa \subset \Re^{n_x}$ that satisfies the nonlinear observability condition, *i.e.,* $rank[\mathbf{U}\left(\mathbf{x}\left(k\right)\right)] = n_x$ at $\mathbf{x}_{\boldsymbol{\Psi}}(k)$, *i.e.,* $\mathbf{x}_{\boldsymbol{\Psi}}(k) \subset \Re^{n_x}$. Should $\mathbf{x}_{\boldsymbol{\Psi}}(k)$ be "sufficiently" close to the true state

$\mathbf{x}(k)$, this can be recovered from the measurements $y(1)...y(k)$. In fact, the observability conditions are used to prove observability in terms of bounded state estimate error covariances at each linearized point, and thus are helpful in proving local convergence, but not asymptotic observability, as we know from LTI systems. However, it can be used as a condition to prove stochastic stability as suggested by [58] and [57]. According to [58] and [57], if the nonlinear system satisfies the conditions of sufficiently small initial estimation error and small noise, then the estimation error of the discrete-time EKF remains bounded. However, as concluded by [58], the sufficient conditions for stochastic stability are so conservative that they have theoretical interest only.

5.3 Posterior Cramer-Rao Lower Bound

The Cramer-Rao Bound (CRB) provides a lower bound on the performance of an unbiased estimator of a vector, $\mathbf{x}$, based on the measurement vector, $\mathbf{y}$. For nonlinear filters the Minimum Mean Square Error (MMSE) may not be achievable, and thus the PCRLB is a lower bound on the minimum MSE. Typically the Cramer-Rao bound is used to evaluate whether the suboptimal estimator is close to the optimal, simultaneously providing an indication of whether the imposed performance requirements are realistic or not.

The Posterior Cramer-Rao Lower Bound (PCRLB) or Bayesian Cramer-Rao Bound states that the error covariance matrix is lower bounded by the inverse of the Fisher Information Matrix (FIM) [63]. The Cramer-Rao inequality is,

$$\mathbf{PCRLB} \triangleq E\left\{\left(\hat{\mathbf{x}}-\mathbf{x}\right)\left(\hat{\mathbf{x}}-\mathbf{x}\right)^{T}\right\} \geq \mathbf{J}^{-1} \tag{5.18}$$

which means that $\mathbf{PCRLB} - \mathbf{J}^{-1} \geq 0$, *i.e.,* the difference between the two matrices is positive semidefinite. The matrix $\mathbf{J}$ is the Fisher information

$$\mathbf{J} \triangleq E\left\{\nabla_x \ln p\left(\mathbf{y},\mathbf{x}\right) \nabla_x^T \ln p\left(\mathbf{y},\mathbf{x}\right)\right\} \tag{5.19}$$

where $\mathbf{x}$ is a random state present in the set of observations, $\mathbf{y}$, with an unbiased estimator $\hat{\mathbf{x}}(\mathbf{y})$, and where ∇_x is the gradient operator, $p\left(\mathbf{y},\mathbf{x}\right)$ is the joint probability density function of $(\mathbf{y},\mathbf{x})$, and $E\{.\}$ is the expectation *w.r.t.* $\mathbf{y}$ and $\mathbf{x}$. The FIM elements are given according to,

$$J_{ij} = E\left\{-\frac{\partial^2 \ln p\left(\mathbf{y},\mathbf{x}\right)}{\partial x_i \partial x_j}\right\} \quad \text{for } i,\, j = 1,...,n_x$$

In [64], Tichavsky *et al.* provides a Ricatti-like recursion for the calculation of $\mathbf{J}(k)$ for linear and

nonlinear systems. Knowing that $p(\mathbf{y},\mathbf{x}) = p(\mathbf{y}|\mathbf{x}).p(\mathbf{x})$ then the FIM can be decomposed in two terms,

$$\mathbf{J} = \mathbf{J}_y + \mathbf{J}_p$$

where $\mathbf{J}_p$ represents the a *prori* information matrix and $\mathbf{J}_y$ represents the information obtained from the data, *i.e.,* measurements from the sensors, of which there can be more than one. In the latter case, if $\imath$ denotes the sensor number, then $\mathbf{J}_y = \sum_{\imath} \mathbf{J}_y^{\imath}$. For simplicity we assume that $\imath = 1$, unless noted otherwise.

For a time-varying linear discrete-time system as defined in (5.13) with Gaussian process and observation noises $\mathbf{Q}$ and $\mathbf{R}$, respectively, the sequence of posterior FIM, $\mathbf{J}(k)$, is as follows,

$$\begin{aligned}
\mathbf{J}(k+1) &= \mathbf{J}_y(k+1) + \mathbf{J}_p(k+1) \\
\mathbf{J}_y(k+1) &= \mathbf{Q}^{-1}(k) - \mathbf{Q}^{-1}(k)\mathbf{A}(k)\left(\mathbf{J}(k) + \mathbf{\Phi}^T(k)\mathbf{Q}^{-1}(k)\mathbf{\Phi}(k)\right)^{-1}\mathbf{\Phi}^T(k)\mathbf{Q}^{-1}(k) \\
\mathbf{J}_p(k+1) &= \mathbf{C}^T(k)\mathbf{R}^{-1}(k)\mathbf{C}(k)
\end{aligned}$$

Applying the matrix inversion lemma, $(\mathbf{Z}+\mathbf{BVD})^{-1} = \mathbf{Z}^{-1} - \mathbf{Z}^{-1}\mathbf{B}(\mathbf{DA}^{-1}\mathbf{B}+\mathbf{V})^{-1}\mathbf{DZ}^{-1}$, the FIM can be written as,

$$\begin{aligned}
\mathbf{J}(k+1) &= \mathbf{J}_y(k+1) + \mathbf{J}_p(k+1) \\
\mathbf{J}_y(k+1) &= \left(\mathbf{Q}(k) + \mathbf{\Phi}(k)\mathbf{J}^{-1}(k)\mathbf{\Phi}^T(k)\right)^{-1} \\
\mathbf{J}_p(k+1) &= \mathbf{C}^T(k)\mathbf{R}^{-1}(k)\mathbf{C}(k)
\end{aligned}$$

For a time-invariant linear discrete-time system, where the matrices $\mathbf{A}(k)$ and $\mathbf{C}(k)$ are independent of k, it is possible to compute $\mathbf{J}(k) \longrightarrow \mathbf{J}_\infty$, with $\mathbf{J}_\infty$ being the solution of an algebraic Riccati equation. In this case, the FIM is in fact the error covariance matrix, *i.e.,* the lowest possible bound is given by the error covariance matrix obtained from the Kalman Filter.

For **nonlinear systems,** as defined in subsection (4.1.2) and with the EKF equations derived in subsection (B.1), where there is a linearization of the dynamic equations about the estimated trajectory, *i.e.,* an approximation continuously updated with the state estimates based on the measurements, the EKF is used. The error covariance matrix obtained from the EKF filter, which is not optimal anymore, is not the lowest bound in the Fisher sense, but is bounded below by the CRLB.

For nonlinear systems, as defined in subsection 4.4.1, the sequence of posterior FIM, $\mathbf{J}(k)$, is as follows,

$$\mathbf{J}(k) = \mathbf{J}_y(k) + \mathbf{J}_p(k)$$

$$\begin{aligned}\mathbf{J}_y(k) &= \mathbf{Q}^{-1}(k-1) - \mathbf{Q}^{-1}(k-1)E\{\mathbf{F}(k-1)\}\left(\mathbf{J}(k-1) + E\{\mathbf{F}^T(k-1)\mathbf{Q}^{-1}(k)\mathbf{F}(k-1)\}\right)^{-1} \\ &\quad \cdot E\{\mathbf{F}^T(k-1)\}\mathbf{Q}^{-1}(k-1) \\ \mathbf{J}_p(k) &= E\{\mathbf{H}^T(k)\mathbf{R}^{-1}(k)\mathbf{H}(k)\}\end{aligned}$$

where $\mathbf{F}(k)$ is the Jacobian matrix of $f(.)$, $\mathbf{H}(k)$ is the Jacobian matrix of $h(.)$, and $\mathbf{Q}(k)$ is the process noise matrix obtained from the power spectrum of the continuous noise.

The evolution of the FIM (which provides the inverse of PCRLB and also the inverse of the filter error covariance matrix, $\mathbf{P}^{-1}(k|k)$) according to [65] and [66]) is given by,

$$\mathbf{J}(k) = \left[\mathbf{Q}(k-1) + \mathbf{F}(k-1)\mathbf{J}^{-1}(k-1)\mathbf{F}^T(k-1)\right]^{-1} + \mathbf{H}^T(k)\mathbf{R}^{-1}(k)\mathbf{H}(k)$$

Replacing $\mathbf{J}(k)$ by $\mathbf{P}^{-1}(k|k)$ gives,

$$\mathbf{P}^{-1}(k|k) = \left[\mathbf{Q}(k-1) + \mathbf{F}(k-1)\mathbf{P}(k-1|k-1)\mathbf{F}^T(k-1)\right]^{-1} + \mathbf{H}^T(k)\mathbf{R}^{-1}(k)\mathbf{H}(k)$$

and knowing from the Extended Kalman filter equations (error covariance prediction) that,

$$\mathbf{P}(k|k-1) = \mathbf{Q}(k-1) + \mathbf{F}(k-1)\mathbf{P}^{-1}(k-1|k-1)\mathbf{F}^T(k-1)$$

thus,

$$\mathbf{P}^{-1}(k|k) = \mathbf{P}^{-1}(k|k-1) + \mathbf{H}^T(k)\mathbf{R}^{-1}(k)\mathbf{H}(k)$$

which is identical to the information matrix filter recursion, Eq. (5.2), and equivalent to the error covariance matrix for the EKF. The difference is that now the Jacobians $\mathbf{H}(k)$ and $\mathbf{F}(k)$ are evaluated about the true trajectory and not the estimated one. This means that true values of the state vector are used to evaluate the Jacobians $\mathbf{H}(k)$ and $\mathbf{F}(k)$, and thus the PCRLB, which is the inverse of the FIM,

$$\begin{aligned}(\mathbf{PCRLB})^{-1} &= \mathbf{J}(k) = \mathbf{P}^{-1}(k|k) = \\ &= \mathbf{P}^{-1}(k|k-1) + \mathbf{H}^T(k)\mathbf{R}(k)^{-1}\mathbf{H}(k)\end{aligned}$$

converges to zero as $k \to \infty$ just as the amount of information, in the sense of Fisher, tends to infinity as $k \to \infty$. Thus the error variance also converges to zero. This gives a necessary condition for an estimator to be *consistent* in the mean square sense. Also, if the filter's variance is equal to the CRLB, which happens for the Kalman Filter, then the estimator is said to be efficient [57].

Since this work is focus in nonlinear systems, the PCRLB is an important tool to be used in the analysis of the results an to identify if the algorithms proposed are far from the optimal or not.

5.4 Filter Consistency

Because the EKF is not optimal, checking its consistency is essential in order to understand its performance. Statistical consistency tests such as the Normalized Estimation Error Squared (NEES) and the Normalized Innovation Squared (NIS) tests help us on this task. A statistical examination of the variability of the nonlinear system is quantified through the Monte-Carlo simulations for the NEES and NIS tests and its results will be presented in Chapter 6, applied to the case study of 3-FF s/c in a GTO mission.

5.4.1 Normalized estimation error squared test

An estimator is called consistent if its state estimation error, $\widetilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$ satisfies the following conditions [57]:

$$\begin{aligned} E\left\{\mathbf{x}(k) - \hat{\mathbf{x}}(k|k)\right\} &= 0 \\ E\left\{\left(\mathbf{x}(k) - \hat{\mathbf{x}}(k|k)\right)\left(\mathbf{x}(k) - \hat{\mathbf{x}}(k|k)\right)^T\right\} &= \mathbf{P}(k|k) \end{aligned}$$

which means that the estimator is unbiased, and the actual MSE matrix matches the error covariance matrix. Whenever the system is nonlinear, checking consistency of the Kalman Filter is extremely important and, if the true state vector is available, a statistical test for filter consistency can be carried out on the NEES, which can be used to characterize the filter performance. Thus, if the filter is consistent a chi-square test on NEES for the parameter x,

$$\epsilon_x \triangleq \widetilde{\mathbf{x}}^T \mathbf{P}^{-1} \widetilde{\mathbf{x}} \tag{5.20}$$

should give ϵ_x a chi-square distribution with n_x DoF, where n_x is the state dimension. That is,

$$\epsilon_x \sim \chi^2_{n_x, 1-\pi}$$

where π is the desired significance level (usually 0.05).

The NEES plot given by $\widetilde{\mathbf{x}}^T \mathbf{P}^{-1} \widetilde{\mathbf{x}}$ should be inside the region indicated by the chi-square test to be considered consistent, *i.e.,* the state vector error must be commensurate with its error covariance matrix (see Fig. 5-3 a)). In the case where the plot is above the upper limit, the error covariance matrix is too small compared to the error state vector, meaning that the precision is high but the accuracy is low (see Fig. 5-3 b)). In the case where the plot is below the lower limit, the error covariance matrix is too large compared to the error state vector, meaning that the precision is low but the accuracy is comparably

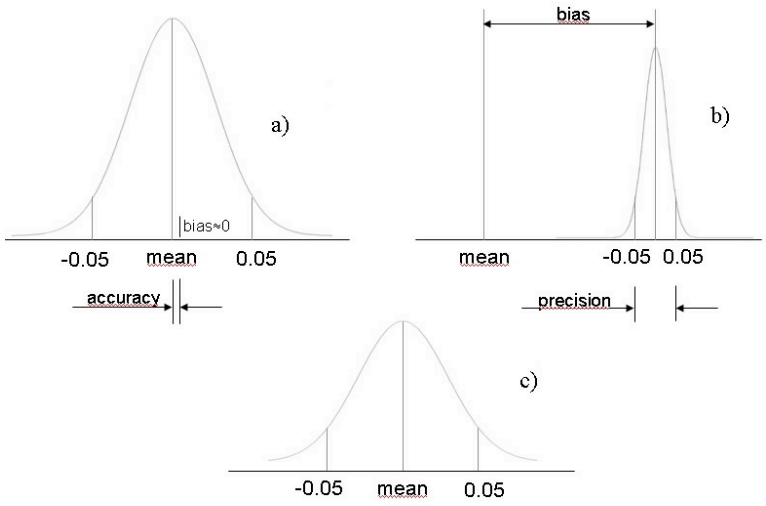

Figure 5-3: **Error variance analysis,** scalar case where the interval $[-0.05, 0.05]$ represents the region indicated by the chi-square test for two-sided 95% probability region. a) the filter is consistent, the error covariance matrix is commensurate with the error. The bias is greatly reduced or zero. b) the filter is inconsistent, its precision is high but its accuracy is very low, corresponding to a NEES plot above the upper limit of the 95% probability region of the chi-square distribution. In practice the error covariance is too small for the corresponding state vector error. c) the filter is pessismistic, the precision is low due for example to bias. The NEES curve is below the upper limit of the 95% probability region of the chi-square distribution. In practice the error covariance is too big compared to a small state vector error.

higher (as illustrated in Fig. 5-3 c)).

In summary, an estimator is

a) *consistent*, when the error covariance matrix is commensurate with its error;

b) *inconsistent*, when the estimator accuracy is very low (bias is large). In this case, *e.g.*, when uncertainty is small (low error covariance), leading to a high estimator precision, the NEES plot is above the upper limit of the probability region of the chi-square distribution;

c) *pessimistic* when the accuracy is high and uncertainty is large, *i.e.,* the error covariance is too large compared to a small state error, corresponding to a NEES plot below the lower limit of the probability region of the chi-square distribution.

Under a Linear Gaussian assumption, $\mathbf{P}^{-1} = \mathbf{J}$, where $\mathbf{J}$ is according to Eq. (5.19), and the average of the NEES must be equal to the dimension of the corresponding vector since it has a chi-square distribution,

$$E\left\{\epsilon_{\mathbf{x}}\right\} = E\left\{\tilde{\mathbf{x}}^T \mathbf{J} \tilde{\mathbf{x}}\right\} = n_x \tag{5.21}$$

In the case where there exists bias, the mean of the error is not zero, which increases the NEES values, causing the chi-square test to fail and the filter to be inconsistent. Since the mean of the error is not zero, there is a bias, and the MSE is increased by the bias,

$$MSE[\hat{\mathbf{x}}(k|k)] = E\left\{(\hat{\mathbf{x}}(k|k) - \mathbf{x}(k))^2\right\}] + bias^2(\hat{\mathbf{x}}(k|k))$$

where $bias(\hat{\mathbf{x}}(k|k)) = \mathbf{x}(k) - E\{\hat{\mathbf{x}}(k|k)\}$.

5.4.2 Normalized innovation squared test

According to [57], under the hypothesis that the Kalman filter is consistent, given by the NEES test, its innovation should also have zero mean, the characteristics of white noise and a MSE magnitude commensurate with the innovation matrix computed by the filter, $\mathbf{S}(k) = \mathbf{H}(k)\mathbf{P}(k)\mathbf{H}(k)^T + \mathbf{R}(k)$. Knowing that the innovation error is $\boldsymbol{\nu}(k) = \mathbf{y}(k) - h(\hat{\mathbf{x}}(k|k))$, the filter satisfies the following conditions,

$$\begin{aligned} E\{\boldsymbol{\nu}(k) - \hat{\boldsymbol{\nu}}(k|k)\} &= 0 \\ E\left\{(\boldsymbol{\nu}(k) - \hat{\boldsymbol{\nu}}(k|k))(\boldsymbol{\nu}(k) - \hat{\boldsymbol{\nu}}(k|k))^T\right\} &= \mathbf{S}(k|k) \end{aligned}$$

which are similar to those of the NEES test but now applied to the measurements. Thus the innovation error consistency test can be carried out on the NIS, which can be used to characterize the filter performance. Thus, if the filter is consistent with its innovations, a chi-square test on NIS for the parameter $\boldsymbol{\nu}$,

$$\epsilon_{\boldsymbol{\nu}} \triangleq \boldsymbol{\nu}^T\mathbf{S}^{-1}\boldsymbol{\nu} \tag{5.22}$$

should give $\epsilon_{\boldsymbol{\nu}}$ a chi-square distribution with n_y DoF, where n_y is the measurements dimension. That is,

$$\epsilon_{\boldsymbol{\nu}} \sim \chi^2_{n_y,1-\pi}$$

where π is the desired significance level (usually 0.05).

Under a Linear Gaussian assumption, the average of the NIS must be equal to the dimension of the corresponding vector since it has a chi-square distribution,

$$E\{\epsilon_{\boldsymbol{\nu}}\} = E\left\{\boldsymbol{\nu}^T\mathbf{S}^{-1}\boldsymbol{\nu}\right\} = n_y$$

which means that the innovation error and the innovation matrix should have commensurate magnitudes.

In the case where there exists bias, the mean of the innovation error is not zero, which will increase NIS, causing the chi-square test to fail and the filter to be inconsistent with regards to the measurement

model. Since the mean of the error is not zero, there is a bias, and the covariance matrix is increased by the bias,

$$MSE[\hat{\boldsymbol{\nu}}(k)] = E\left\{(\hat{\boldsymbol{\nu}}(k) - \boldsymbol{\nu}(k))^2\right\} + bias^2(\hat{\boldsymbol{\nu}}(k))$$

where $bias(\hat{\boldsymbol{\nu}}(k|k)) = \boldsymbol{\nu}(k) - E\{\hat{\boldsymbol{\nu}}(k|k)\}$.

5.5 Summary

In this chapter, the stability of the EKF, applied to a nonlinear discrete-time system, was discussed. Observability was addressed since it is related to the existence of the error covariance matrix and thus related to the stability of the system. In nonlinear systems the observability condition can no longer be used as it is in linear systems due to the uncertainty around the linearized state, and thus it is used to prove local convergence only.

The PCRLB was discussed for nonlinear systems to obtain the optimal performance limits, providing a benchmark against which to compare the EKF performance. Thus, comparing the PCRLB with the EKF results we can infer how close the suboptimal EKF performance comes to the optimal performance, and noting whether the specified requirements are demanding for EKF or not. With this mathematical tool, and knowing that EKF gives a suboptimal solution, we can analyse: a) if the EKF results are accurate or not; b) how far it is from the optimal performance; c) if there exist bias or not; d) how the compensation of nonlinearities can improve the EKF performance.

Moreover, the consistency tests are addressed, showing whether the filter is being optimistic or pessimist to the presence of bias in the state estimates or to unavoidable errors from the linearization. These are important issues in the EKF assumptions since it could cause the filter to be inconsistent and thus give poor performance or even diverge.

Chapter 6

SIMULATION SETUP

To validate the Navigation algorithms described in this work, a FF s/c simulator, Formation Flying - Functional Engineering Simulator (FF-FES)[1], was used under the ESA FEMDS[2] project whose goal is the development and evaluation of GNC algorithms and concepts for a FF demonstration mission in a GTO, since for LEO or GEO the gravity gradient represents great expense of thrusters and is therefore considered unsuitable for multiple spacecraft demonstration missions. The focus of the implementation and analysis of this work is done in great detail around apogee. The case study which is simulated in FF-FES simulator is a formation of 3-FF s/c in GTO, aiming at doing science experiments in the apogee, using high resolution imaging by aperture synthesis with minimum control action, which is very similar to the Darwin infrared interferometer mission to be launched in 2014. Due to the position of the formation in GTO and to the different science requirements, the formation was divided in three different phases:

- Deployment phase.
- Loose Formation phase.
- Constellation/formation phase.

Each of the phases is defined by one or two modes with different GNC scenarios. Here, the deployment phase, characterized by the Formation Acquisition mode (FAC), and the Baseline Control Mode (BCM), was chosen to run the GNC algorithms. FAC is the first mode in which the formation enters after launch, representing a demanding goal of controlling the formation state from a disperse configuration within a sphere of 8 $[km]$ into an equilateral triangle of 250 metre formation. The BCM mode is related to

[1] developed by DEIMOS Engenharia Lda.
[2] Formation Estimation Methodologies for Distributed Spacecraft.

the period of the orbit where the science experiment takes place, and each s/c has to be controlled in absolute attitude, in order to reach the goal of pointing at target stars. Full state estimates are required to achieve such goals, namely to feed controller during BCM and keep track of the state to detect events that trigger mode changes, for all modes.

The present study uses the RF sensor for relative state estimation of position and translational velocity, and uses ST measurements for inertial attitude and angular velocity estimation, focusing on the problem of determining the full state vector estimates at each element of the formation in a decentralized architecture, as proposed in Chapter 4.

A description of the FF-FES simulator with its assumptions and constraints as well as some of its capabilities, is provided in Section 6.1. The next section introduces a detailed description of the different modes of the GTO and of the characteristics of the formation orbit and composing s/c. In Section 6.3, the estimates of the full 6DoF state vector are computed according to the filter design.

The full algorithm is implemented in 3-FF s/c, which is the minimum number of s/c that sets up a formation with some degree of complexity. The case study, which includes the EKF and the CI filters (as shown in Fig. 4-12), applied to relative position and translational velocity, as described in subsection 4.6.1, and the filters EKF and the EKF Fusion (as shown in Fig. 4-13), applied to absolute attitude and angular velocity, as described in subsection 4.7.1, has the purpose of illustrating the full state vector estimates determination using the peer-to-peer *information flow* graph for the information flow graph, as defined in Section 3.4, and the *measurement* graph, as defined in Section 3.3, with special attention to the technical issues concerning the CI algorithm and to the implementation of the quaternion variables in the EKF.

The study carried out considered a formation orbiting in GTO, which is an eccentric orbit, where inertial measurements are not available for most of the orbit, making the relative state vector estimates depend on sensors measured on each s/c, whose measurements relate to the s/c in the formation. The coordinate systems relevant to this work as well as the parameterization of their relations are defined in subsection 6.3.1, as well as the equations of motion, the relative dynamic equations for eccentric orbit (subsection 6.3.4) and the kinematic equations (subsection 6.3.5). The dynamic equations of the preceding section are re-cast in subsection 6.3.6, so as to obtain a relative state representation, more suited for the estimator. In this section, the RF subsystem measurements and the EKF observation model to estimate the translational relative state vector, are also presented. Also the kinematic equations of the preceding section are linearized in subsection 6.3.6, taking care of the quaternion operations, and in this section the ST measurements and the EKF observation model to estimate the rotational absolute state vector are also presented. Afterwards, the description of the experiments with the estimator algorithm in the loop, and with the estimator as an observer without feeding the GC algorithms with its estimates,

addressed in Section 6.4. Finally, the statistical information used to analyse the errors of the estimator algorithm in simulation is defined and explained in Section 6.5, with definition of the performance criteria to evaluate the consistency of the navigation algorithm results, discussed in Chapter 5.

6.1 Formation Flying Functional Engineering Simulator

The FF-FES platform is a MATLAB/Simulink version 6.5 program, developed by DEIMOS SA [67], which currently reproduces a realistic environment of three spacecraft flying, but possible to extend to more s/c, in GTO, which supports the inclusion of GNC algorithms in decentralized scheme. The three s/c motion and its time-varying behaviour relative to each other and to the Earth is realistically simulated through the propagation of the s/c state, using the dynamic and kinematic equations of motion. Thus, considering both translational and rotational motion, each s/c is simulated with 6DoF including attitude dynamics and kinematics. Are also simulated equations of motion which include perturbations from different sources. Thus, Solar radiation pressure, $J2$ effect due to the Earth oblateness, third body perturbation from the Sun, and Moon are considered. Air drag due to the Earth atmosphere and micrometeorids that have relevant impact in the s/c dynamics are also simulated, as well as gravitational attraction, including high order gravitational acceleration harmonics. For the sensors, each s/c carrying the RF metrology, ST, gyros and coarse lateral sensor (divergent laser) is simulated in a frame that includes relative and absolute measurements *w.r.t.* to Earth. Moreover, functional models of the micro-N cold gas thrusters are simulated as the unique type of actuator available. A communication channel to exchange data information among the s/c is also simulated. Thus, the FF-FES simulator performs open-loop simulation and also closed-loop of the formation in the different phases using provided GNC algorithms in a decentralized scheme and in an environment very close to reality.

A block diagram illustrating a global view of the main blocks composing the simulator is shown in Figure 6-1. Figure 6-2 details this block diagram for one of the s/c, including the GNC block, where the developed algorithms run. The connections between the Navigation and the GC algorithms are depicted in Fig. 6-3.

6.2 Geostationary Transfer Orbit Mission

In order to accomplish the science mission the formation has to be study since launch, with random disposition within a sphere of 8 $[km]$ diameter, till the final disposition, in which the s/c are 250 metre apart from each other and in a formation plane required for the science interferometer experiment, going through a natural motion around perigee and returning to a coarse formation configuration ready to

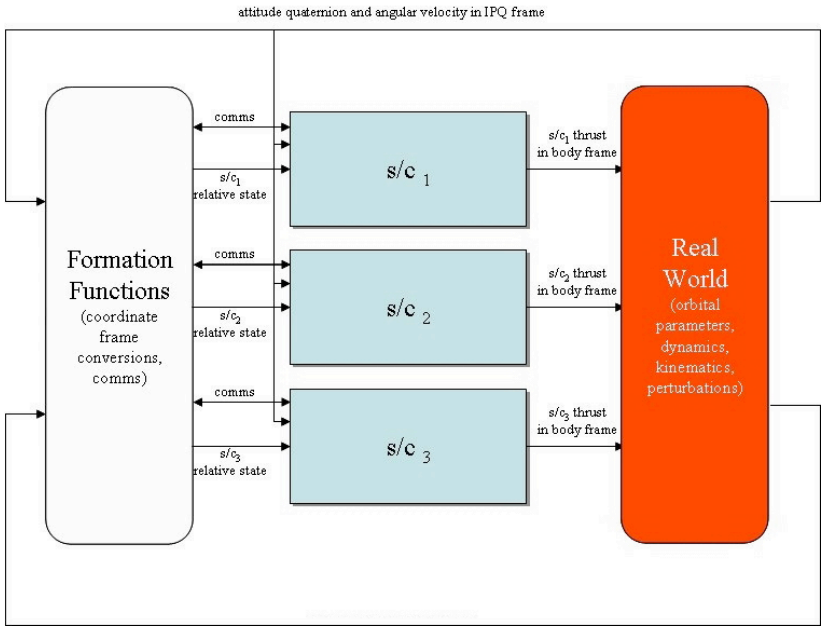

Figure 6-1: Block diagram of the main simulator blocks.

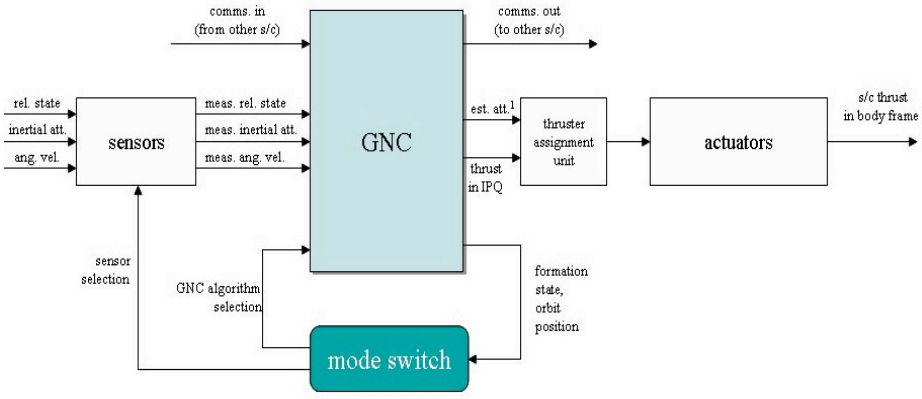

Figure 6-2: Block diagram of the main subsystems inside each s/c in the simulator.

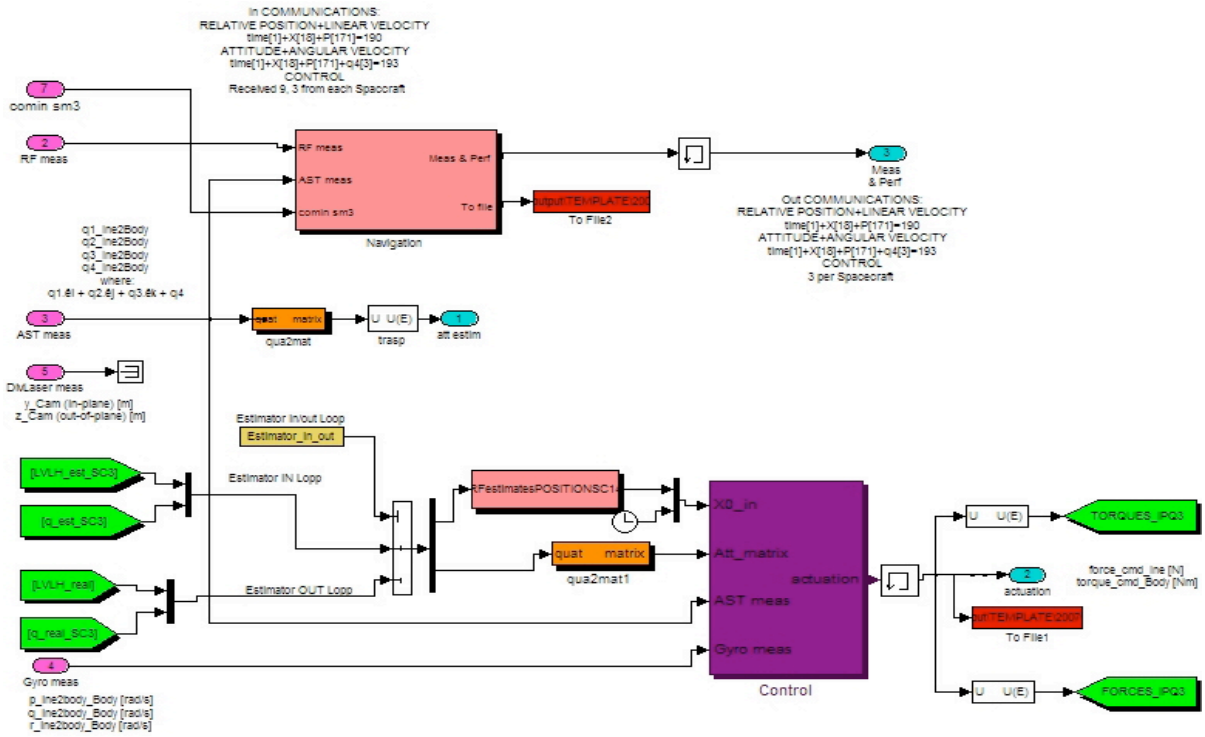

Figure 6-3: GNC block diagram inside each s/c in the simulator.

recover the configuration appropriate to do science experiment. All these phases lead to the definition of different modes with different requirements.

6.2.1 Modes definition

The GTO is divided in three phases depending on the science experiment accuracy and the position of the formation in the orbit. The three phases are in turn defined in terms of modes, depending if the formation is doing science or not and if the control requirement are fine or coarse. Thus, the initial mode corresponds to the formation deployment and is defined as the Formation Acquisition Mode (FAC). Afterwards the formation can go to Interferometer Construction (ICM) mode or to Loose formation mode (LFM), depending on if in the apogee the formation is meant to do science or not. Also the science experiment is defined as Baseline Control Mode (BCM) divided in two modes (Fine BCM or Coarse BCM) depending on the demanding accuracy.

Deployment phase In this case the formation is in acquisition mode (FAC) and the goal is to bring the s/c from a random disposition in a sphere of 8 $[km]$ after dispenser, to a configuration suitable for LFM or ICM depending on the location in the orbit.

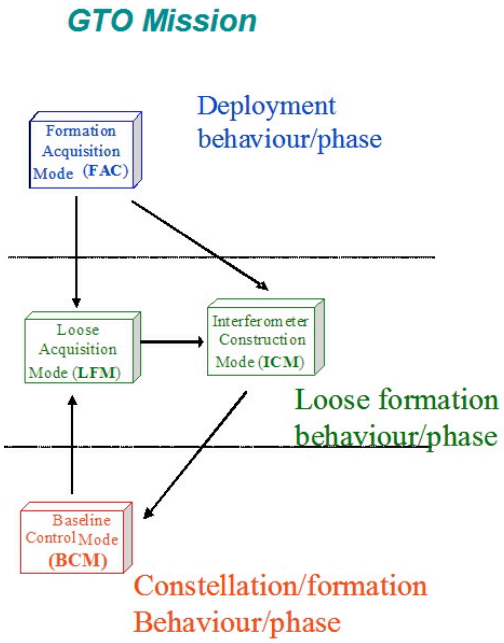

Figure 6-4: Proposed mode sequence. Reprinted from [68].

Loose formation phase is divided in two modes:

LFM the Loose Formation Mode (LFM), in which the s/c enters after the science experiment has ended or after FAC mode, the formation is kept in natural motion during the rest of the orbit until next pass over the apogee, where the ICM mode begins;

ICM the Interferometer Construction Mode (ICM) that prepares the formation conditions to perform the science experiment. The s/c are put at a distance of 250 metre, starting from the outcome position of the LFM.

Constelation/formation phase The formation is in Baseline Control Mode (BCM), during which the science experiment takes place, and is divided into two different sub-modes:

Coarse BCM in which transition is made from control requirements at the level of ICM to the fine requirements of BCM.

Fine BCM in which the s/c control requirements are in line with typical requirements of a BCM (*e.g.*, Darwin).

All the modes can be visualized in GTO in Figs. 6-4 and 6-5. Figure 6-5 shows the zone of application of each mode during the GTO.

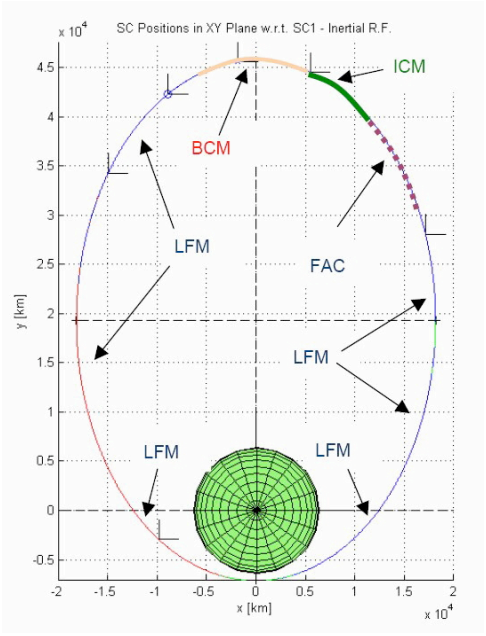

Figure 6-5: Considered modes upon the baseline GTO (not at scale). Reprinted from [68].

6.2.2 Requirements

Note that the GC techniques for ICM and BCM are similar to the techniques used for FAC, the main difference being that for FAC the initial formation is more dispersed than for a normal ICM, and in BCM the goal is to keep the formation, with accuracy of $0.01[m]$ for relative position, $0.001[m/s]$ for velocity, $1[arcsec]$ for inertial attitude, $0.1[arcsec/s]$ for angular velocity and $0.01[m]$, $0.001[m/s]$ for lateral position and velocity, respectively, as shown in Table 6.1.

6.2.3 Orbital parameters simulation setup

The GTO reference orbit is characterized by [68], where the additional orbital parameters are defined in Tables 6.3 and 6.4, for FAC and BCM modes, respectively. The natural frequency n_t of the reference orbit is,

$$n_t = \sqrt{\frac{\mu_t}{a_{smx}^3}} \tag{6.1}$$

where a_{smx} stands for the semi-major axis, and defined in Table 6.2, μ_t stands for the product between the universal gravitation constant G_{rt} and the Earth's mass, m_{Earth}

$$\mu_t = G_{rt}.m_{\text{Earth}} = 3.9860044.10^5 \left[\frac{\text{km}^3}{s^2}\right] \tag{6.2}$$

Formation	**FAC**	**LFM**	**ICM**	**BCM**
Navigation sensors	RF Star tracker	RF Star tracker	RF Star tracker	RF, Star tracker Divergent laser
Control requirements:				
·Longitudinal relative position	$0.1[m]$	N/A	$0.1[m]$	$0.01[m]$
·Longitudinal translational velocity	$0.01[m/s]$	N/A	$0.01[m/s]$	$0.01[m/s]$
·Lateral relative position	N/A	N/A	N/A	$0.01[m]$ $0.001[m/s]$
·Inertial attitude	$1[^o]$	$1[^o]$	$1[^o]$	1[arcsec]
·Inertial Angular velocity	$0.1[^o/s]$	$0.1[^o/s]$	$0.1[^o/s]$	0.1[arcsec /s]

Table 6.1: Specifications for all the modes of the GTO mission: Formation Acquisition Mode (FAC), Loose Formation Mode (LFM), Interferometer Construction Mode (ICM) and Baseline Control Mode (BCM). $1deg = 1^o$, $1' = 1/60th$ of 1^o, $1_J = 1/60th$ of $1'$.

a_{smx}	semi-major axis	26624.137[km]
e_{cc}	eccentricity	0.730389
R_{aan}	Right ascension of the ascending node	0^o

Table 6.2: Specifications of the orbital parameters defined in the dynamic equations for GTO orbit.

Darwin spacecraft are used at the simulator with a mass of $250[kg]$ each, and with a cube shape, as described by the Table 6.5.

System initial conditions - FAC mode

The initial true state vector for relative position and translational relative velocity corresponds to an initial random disposition of position components within a sphere of $8\,[km]$ diameter, centered in the dispenser, as in Table 6.6.

The initial true state vector (before FAC mode) corresponds to approximately $0[rad/s]$ for the angular velocity components and $0[^o]$ for Euler angles. The quaternions can be obtained from the Euler angles using (6.48), which equal values for all the s/c, as listed in Table 6.7.

System initial conditions - BCM mode

The initial true state vector for relative position and translational relative velocity corresponds to a triangle configuration with disposition of position components within a sphere of $250\,[m]$ diameter, centered in the dispenser, as in Table 6.8.

The initial true state vector (before BCM mode) corresponds to approximately $0[rad/s]$ for the angular velocity components, and ($89\,[^o]$, $0.71\,[^o]$, $89\,[^o]$) for ψ, θ, ϕ, the Euler angles, respectively. The

i_{cc}	inclination	$7[^o]$
ϖ	argument of perigee	$-90[^o]$
θ_t	true anomaly	$156.2700853[^o]$

Table 6.3: Specifications of the additonal GTO orbital parameters for FAC mode.

i_{cc}	inclination	$6.999982532[^o]$
ϖ	argument of perigee	$-89.996375[^o]$
θ_t	true anomaly	$173.0721[^o]$

Table 6.4: Specifications of the additional GTO orbital parameters for BCM mode.

quaternions can be obtained from the Euler angles using (6.48), which have equal values for all the s/c, as listed in Table 6.7.

Sensors description

The characteristics of the simulated RF sensors in FF-FES are listed in Table 6.9. As for the ST sensor simulated in the FF-FES, its major characteristics are listed in Table 6.10.

6.3 Six Degrees of Freedom Integrated Navigation Implementation

To validate the estimation algorithms described in this chapter, a 3-FF s/c is used. The goal is the development and evaluation of estimation algorithms for a FF demonstration mission in a GTO. The sensors used are a RF system and a ST on board each the 3-s/c. We start with the description of the coordinate systems used, followed by the relation between them.

The local frame is defined as the reference orbit of the formation, and denoted as the Local Vertical Local Horizontal (LVLH) frame. Making use of the reference frames defined, the translational relative state vector is defined *w.r.t.* the LVLH while the rotational state vector is an absolute state vector defined *w.r.t.* the inertial frame, both described in subsections 6.3.2 and 6.3.3, respectively. Also, the s/c equations of relative motion are described for a GTO for both translational and rotational motions using the translational and rotational state vector, as described in subsections 6.3.4 and 6.3.5, in relative and inertial frames, respectively. In subsection 6.3.6, implementation of the overall algorithms (CI+EKF+EKF Fusion) for full state estimate vector is described. Details of EKF equations are provided, in the following subsection, focused on the RF measurements for the filtering part of the translational relative state vector, and on the ST measurements for the filtering part of the rotational absolute state vector, as well

	Dimensions s/c 1	Dimensions s/c 2	Dimensions s/c 3
mass	$250[kg]$	$250[kg]$	$250[kg]$
cube	$1.5 \times 1.5 \times 1.5[m^3]$	$1.5 \times 1.5 \times 1.5[m^3]$	$1.5 \times 1.5 \times 1.5[m^3]$
Inertial Moments	$x \quad 375[kg/m^2]$ $y \quad -140.625[kg/m^2]$ $z \quad -140.625[kg/m^2]$	$x \quad -140.625[kg/m^2]$ $y \quad 375[kg/m^2]$ $z \quad -140.625[kg/m^2]$	$x \quad -140.625[kg/m^2]$ $y \quad -140.625[kg/m^2]$ $z \quad 375[kg/m^2]$

Table 6.5: Specifications of each s/c in the 3-FF Darwin mission.

relative position	$\mathbf{x}_{trans}(\theta_0)$	relative velocity	$\mathbf{x}'_{trans}(\theta_0)$
x_{12}= x_2−x_1[m]	3000	$\dot{x}_{12}$=$\dot{x}_2$−$\dot{x}_1$[m/ s]	-0.04
y_{12}= y_2−y_1[m]	300	$\dot{y}_{12}$=$\dot{y}_2$−$\dot{y}_1$[m/ s]	-0.04
z_{12}= z_2−z_1[m]	-864	$\dot{z}_{12}$=$\dot{z}_2$−$\dot{z}_1$[m/ s]	-0.04
x_{32}= x_2−x_3[m]	2875	$\dot{x}_{32}$=$\dot{x}_2$−$\dot{x}_3$[m/ s]	-0.02
y_{32}= y_2−y_3[m]	175	$\dot{y}_{32}$=$\dot{y}_2$−$\dot{y}_3$[m/ s]	-0.02
z_{32}= z_2−z_3[m]	-3749	$\dot{z}_{32}$=$\dot{z}_2$−$\dot{z}_3$[m/ s]	-0.06
x_{13}= x_3−x_1[m]	125	$\dot{x}_{13}$=$\dot{x}_3$−$\dot{x}_1$[m/ s]	-0.02
y_{13}= y_3−y_1[m]	125	$\dot{y}_{13}$=$\dot{y}_3$−$\dot{y}_1$[m/ s]	-0.02
z_{13}= z_3−z_1[m]	2885	$\dot{z}_{13}$=$\dot{z}_3$−$\dot{z}_1$[m/ s]	0.02

Table 6.6: FAC mode relative position and translational velocity initial conditions for the FF. s/c1≡s/c2≡s/c3.

as details of the prediction part of the EKF, with the calculation of the Jacobians.

6.3.1 Coordinate systems and rotations matrix

The orientation and position of a s/c must be known *w.r.t.* the Earth, the Sun or the stars. The relative sensor measurements, provided by the RF sensor, must also be described *w.r.t.* a reference s/c body frame or any other frame suitable to the problem. Three reference coordinate systems are used throughout this work. They are described in the following subsection.

Reference frames

We define the following reference frames:

1) The **Inertial Planet Frame** (IPQ) is the reference inertial coordinate system, with versors $\overrightarrow{\mathbf{I}}$, $\overrightarrow{\mathbf{J}}$ and $\overrightarrow{\mathbf{K}}$ as shown in Fig. (6-6), is defined as follows:

- Origin: Earth center;
- $+x^{IPQ}$ axis: in the equator plane, parallel to the Earth vernal equinox direction;
- $+y^{IPQ}$ axis: is in the equatorial plane and completes the frame;

s/c1≡s/c2≡s/c3	$\mathbf{x}_{rot}(0)$
$\omega_x^{b_1}[rad/s]$	0
$\omega_y^{b_1}[rad/s]$	0
$\omega_z^{b_1}[rad/s]$	0
$q_1^{b_1}$	0
$q_2^{b_1}$	0
$q_3^{b_1}$	0
$q_4^{b_1}$	1

Table 6.7: FAC mode and BCM attitude and angular velocity initial conditions for the FF. s/c1≡s/c2≡s/c3.

relative position	$\mathbf{x}_{trans}(\theta_0)$	relative velocity	$\mathbf{x}'_{trans}(\theta_0)$
x_{12}= x_2−x_1[m]	-248.1704	$\dot{x}_{12}$=$\dot{x}_2$−$\dot{x}_1$[m/ s]	-0.004
y_{12}= y_2−y_1[m]	0	$\dot{y}_{12}$=$\dot{y}_2$−$\dot{y}_1$[m/ s]	-0.002
z_{12}= z_2−z_1[m]	-30.1519	$\dot{z}_{12}$=$\dot{z}_2$−$\dot{z}_1$[m/ s]	-0.011
x_{32}= x_2−x_3[m]	-372.2606	$\dot{x}_{32}$=$\dot{x}_2$−$\dot{x}_3$[m/ s]	-0.02
y_{32}= y_2−y_3[m]	216.51	$\dot{y}_{32}$=$\dot{y}_2$−$\dot{y}_3$[m/ s]	-0.02
z_{32}= z_2−z_3[m]	-45.2274	$\dot{z}_{32}$=$\dot{z}_2$−$\dot{z}_3$[m/ s]	-0.06
x_{13}= x_3−x_1[m]	124.0902	$\dot{x}_{13}$=$\dot{x}_3$−$\dot{x}_1$[m/ s]	0.0022
y_{13}= y_3−y_1[m]	-216.51	$\dot{y}_{13}$=$\dot{y}_3$−$\dot{y}_1$[m/ s]	0.0004
z_{13}= z_3−z_1[m]	15.0770	$\dot{z}_{13}$=$\dot{z}_3$−$\dot{z}_1$[m/ s]	-0.0049

Table 6.8: BCM mode relative position and translational velocity initial conditions for the FF. s/c1≡s/c2≡s/c3.

- $+z^{IPQ}$ axis: from the Earth mass center towards North (*i.e.,* along the Earth angular velocity vector, perpendicular to equatorial plane).

2) The **Local Vertical Local Horizon** (LVLH) frame used to locate each s/c with respect to the formation center, f_c, which is placed on the reference orbit frame. LVLH is defined as follows (see Fig. 6-8):

- Origin: formation center (placed on the reference orbit frame and each of the axis normal to the face of a cube modeling the s/c, so that $\vec{\mathbf{e}}_x$, $\vec{\mathbf{e}}_y$, $\vec{\mathbf{e}}_z$ form an orthogonal triad);
- $+x^{LVLH}$ axis: completes the right-hand system with the other two axes;
- $+y^{LVLH}$ axis: is normal to the orbital plane, opposite the angular momentum vector of the reference orbit;
- $+z^{LVLH}$ axis: points in the nadir direction, that is, towards the Earth mass center.

Radio Frequency	
Number of receiving antennas on the s/c	3
chip length $[m]$	3.0
RF wave longitude $[m]$	0.01
temperature on the receiving antenna $[K]$	100
Gain of receiving antennas	1.0
Gain of transmitting antennas	1.0
RF signal power	10^{-3}

Table 6.9: RF subsystem features simulated in the FF-FES.

Star Tracker	
Rotation from body to star tracker frame	$\begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$
Random bias, for each attitude angle $[rad]$	$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$
Noise equivalent angle (NEAx) for sensitive x-axis	10^{-5}
Noise equivalent angle (NEAy) for sensitive y-axis	10^{-5}
Noise equivalent angle (NEAz) for sensitive z-axis	10^{-5}
Noise correlation time $[s]$	0.1
initial seed value for the random number generator	$\begin{bmatrix} 1213 & 2423 & 2131 \end{bmatrix}$

Table 6.10: ST system features simulated in the FF-FES.

3) For each s/c i, with $i = 1, ..., N$, the **Body Frame** (b_i) [69]**,** with its center placed in the center of gravity of the s/c and each of the axis normal to the face of a cube modeling the s/c, so that x^{b_i}, y^{b_i}, z^{b_i} form an orthogonal triad, with versors $\overrightarrow{\mathbf{b}}_x$, $\overrightarrow{\mathbf{b}}_y$, $\overrightarrow{\mathbf{b}}_z$ as shown in Fig. 6-9.

4) One usual representation of rotations, due to its clear physical interpretation, is the **Euler axis/angle**. Using Euler's theorem - "*The most general displacement of a rigid body with one point fixed is equivalent to a single rotation of an angle about some axis through that point.*" - and considering the angle as α and the unit vector $\overrightarrow{\mathbf{u}} = \begin{bmatrix} \overrightarrow{\mathbf{u}}_1 & \overrightarrow{\mathbf{u}}_2 & \overrightarrow{\mathbf{u}}_3 \end{bmatrix}^T$ about the z axis of rotation, as shown in Fig. 6-7, the quaternions can be defined in terms of the Euler axis/angle by,

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \equiv \begin{bmatrix} \overrightarrow{\mathbf{u}}_1 \sin\left(\frac{\alpha}{2}\right) \\ \overrightarrow{\mathbf{u}}_2 \sin\left(\frac{\alpha}{2}\right) \\ \overrightarrow{\mathbf{u}}_3 \sin\left(\frac{\alpha}{2}\right) \\ \cos\left(\frac{\alpha}{2}\right) \end{bmatrix}$$

Notice that the LVLH frame or body frame defined here, can be set as the local frames at the definition of the relative state vector in Section 4.1.

Since we will take as a starting point the relative dynamic equations provided by [70] and [10], Fig. 6-8 also presents the frame used by these authors, named here as Tillerson's frame. Tillerson's frame

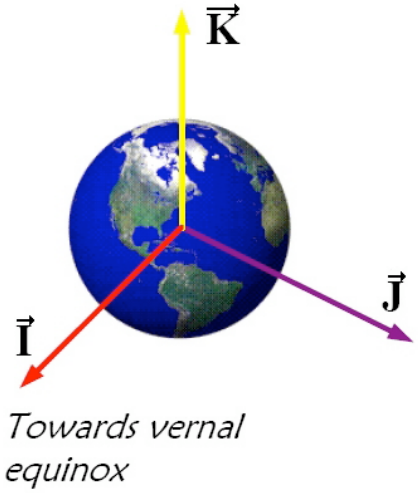

Figure 6-6: Inertial Planet Frame (IPQ) is the reference inertial coordinate system also known as Earth Centered Inertial (ECI) frame.

has also the formation center as origin, its axis being $\overrightarrow{\mathbf{k}}_x$, $\overrightarrow{\mathbf{k}}_y$ and $\overrightarrow{\mathbf{k}}_z$. We have:

$$\overrightarrow{\mathbf{k}}_x, = -\overrightarrow{\mathbf{e}}_z,\ \overrightarrow{\mathbf{k}}_y, = -\overrightarrow{\mathbf{e}}_x,\ \overrightarrow{\mathbf{k}}_z, = -\overrightarrow{\mathbf{e}}_y$$

The transformation matrix

$$\mathbf{A}_{LVLH}^{Tillerson} = \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

allows us to transform the representation $\boldsymbol{\chi}^{LVLH}$ in the LVLH frame of the position vector $\boldsymbol{\chi}$ into its representation $\boldsymbol{\chi}_{Tillerson}$ in Tillerson's frame:

$$\boldsymbol{\chi}^{Tillerson} = \mathbf{A}_{LVLH}^{Tillerson} \boldsymbol{\chi}^{LVLH}$$

Transformation between IPQ and LVLH reference frames

By taking into account the following orbital parameters

- the right ascension of the ascending node R_{aan} (acronym RAAN).
- the inclination i_{cl}.

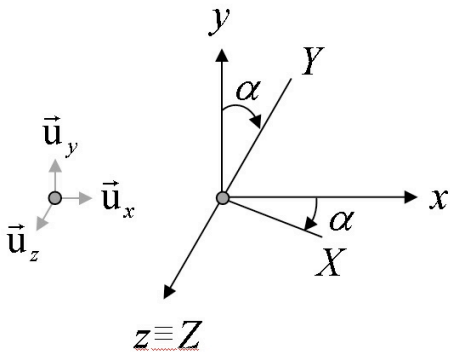

Figure 6-7: Angle/axis rotation about z axis. x, y, z are the original axis and X, Y, Z are the rotating x, y, z axis around z axis.

- the argument of perigee ϖ.
- the true anomaly θ_t.

We can obtain [51, p.24-26] the transformation matrix between LVLH frame and IPQ frame, which is designed as a rotation matrix since it is not obtained directly from quaternions, $\mathbf{A}_{LVLH}^{IPQ}$:

$$\begin{aligned} \mathbf{A}_{LVLH}^{IPQ} &= \left[\mathbf{A}_z(\varpi+\theta_t)\mathbf{A}_x(i_{cl})\mathbf{A}_z(R_{aan})\right]\mathbf{A}_{LVLH}^{Tillerson} = \\ &= \begin{bmatrix} \vdots & \vdots & \vdots \\ \mathbf{a}_x & \mathbf{a}_y & \mathbf{a}_z \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \end{aligned} \tag{6.3}$$

with

$$\mathbf{a}_x = \begin{bmatrix} \cos R_{aan}\cos(\varpi+\theta_t) - \sin R_{aan}\cos i_{cl}\sin(\varpi+\theta_t) \\ \sin R_{aan}\cos(\varpi+\theta_t) + \cos R_{aan}\cos i_{cl}\sin(\varpi+\theta_t) \\ \sin i_{cl}\sin(\varpi+\theta_t) \end{bmatrix}$$

$$\mathbf{a}_y = \begin{bmatrix} -\cos R_{aan}\sin(\varpi+\theta_t) - \sin R_{aan}\cos i_{cl}\cos(\varpi+\theta_t) \\ -\sin R_{aan}\sin(\varpi+\theta_t) + \cos R_{aan}\cos i_{cl}\cos(\varpi+\theta_t) \\ \sin i_{cl}\cos(\varpi+\theta_t) \end{bmatrix}$$

$$\mathbf{a}_z = \begin{bmatrix} \sin R_{aan}\sin i_{cl} \\ -\cos R_{aan}\sin i_{cl} \\ \cos i_{cl} \end{bmatrix}$$

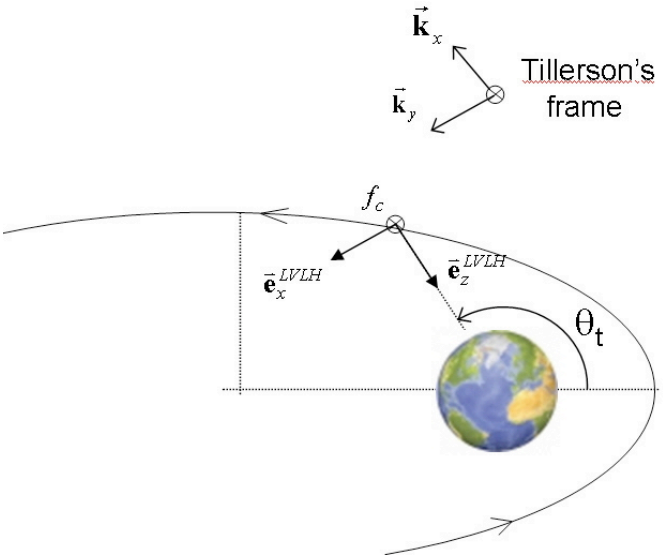

Figure 6-8: LVLH frame, view from above the orbital plane.

Thus,

$$\mathbf{A}_{LVLH}^{IPQ} = \begin{bmatrix} \vdots & \vdots & \vdots \\ \mathbf{a}_x & \mathbf{a}_y & \mathbf{a}_z \\ \vdots & \vdots & \vdots \end{bmatrix}$$

$$(\mathbf{a}_x)_{LVLH}^{IPQ} = \begin{bmatrix} -\cos R_{aan}\sin(\varpi+\theta_t) - \sin R_{aan}\cos i_{cl}\cos(\varpi+\theta_t) \\ -\sin R_{aan}\sin(\varpi+\theta_t) + \cos R_{aan}\cos i_{cl}\cos(\varpi+\theta_t) \\ \sin i_{cl}\cos(\varpi+\theta_t) \end{bmatrix}$$

$$(\mathbf{a}_y)_{LVLH}^{IPQ} = \begin{bmatrix} -\sin R_{aan}\sin i_{cl} \\ \cos R_{aan}\sin i_{cl} \\ -\cos i_{cl} \end{bmatrix}$$

$$(\mathbf{a}_z)_{LVLH}^{IPQ} = \begin{bmatrix} -\cos R_{aan}\cos(\varpi+\theta_t) + \sin R_{aan}\cos i_{cl}\sin(\varpi+\theta_t) \\ -\sin R_{aan}\cos(\varpi+\theta_t) - \cos R_{aan}\cos i_{cl}\sin(\varpi+\theta_t) \\ -\sin i_{cl}\sin(\varpi+\theta_t) \end{bmatrix}$$

The two following equivalent identities are describing the transformations between representations of a vector $\boldsymbol{\chi}$ in the IPQ frame and in the LVLH frame:

$$\boldsymbol{\chi}^{IPQ} = \mathbf{A}_{LVLH}^{IPQ}\boldsymbol{\chi}^{LVLH} \Longleftrightarrow \boldsymbol{\chi}^{LVLH} = \left(\mathbf{A}_{LVLH}^{IPQ}\right)^T \boldsymbol{\chi}^{IPQ} \tag{6.4}$$

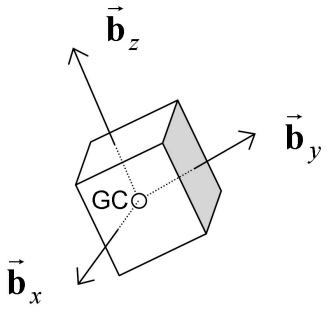

Figure 6-9: Body frame versors of a spacecraft.

where $(.)^T$ denotes the transpose of matrix. Transformation 6.4 is valid only for "static" vectors (position and external force vectors). If we are dealing with transformations of velocity or acceleration vectors between frames, the use of kinematics would require taking into consideration additional terms (Coriolis acceleration, centripetal acceleration, accelerating rotation of LVLH with respect to IPQ).

6.3.2 Translational state vector

Using the frames defined before, we define the translational relative state vector that relates body frame b_i *w.r.t.* body frame b_j, $\mathbf{x}_{trans}^{LVLH}$ expressed in LVLH frame. According to the state vector definition performed in subsection 4.1.1, the translational relative vector for the s/c i, is

$$\mathbf{x}_{trans}^{LVLH}(k) = \begin{bmatrix} \left(\boldsymbol{\chi}_{b_i b_j}^{LVLH}\right)^T & \left(\dot{\boldsymbol{\chi}}_{b_i b_j}^{LVLH}\right)^T \end{bmatrix}^T$$

with $i = 1, ..., N, i \neq j$. Knowing that the formation is composed by 3-s/c, $N = 3$, the size of the *formation estimation* graph is $\frac{N(N-1)}{2} = 3$ components for each vector denoting relative position and translational relative velocity between pairs of s/c, as follows,

$$\begin{aligned} \boldsymbol{\chi}_{b_i b_j}^{LVLH} &= \begin{bmatrix} (\boldsymbol{\chi}_{12})^T & (\boldsymbol{\chi}_{32})^T & (\boldsymbol{\chi}_{13})^T \end{bmatrix}^T \\ &= \begin{bmatrix} (\boldsymbol{\chi}_2 - \boldsymbol{\chi}_1)^T & (\boldsymbol{\chi}_2 - \boldsymbol{\chi}_3)^T & (\boldsymbol{\chi}_3 - \boldsymbol{\chi}_1)^T \end{bmatrix}^T \\ &= \begin{bmatrix} \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{bmatrix}^T & \begin{bmatrix} x_2 - x_3 \\ y_2 - y_3 \\ z_2 - z_3 \end{bmatrix}^T & \begin{bmatrix} x_3 - x_1 \\ y_3 - y_1 \\ z_3 - z_1 \end{bmatrix}^T \end{bmatrix}^T \end{aligned} \tag{6.5}$$

$$\boldsymbol{\chi}_{b_i b_j}^{LVLH} = \left[\begin{bmatrix} x_{12} \\ y_{12} \\ z_{12} \end{bmatrix}^T \quad \begin{bmatrix} x_{32} \\ y_{32} \\ z_{32} \end{bmatrix}^T \quad \begin{bmatrix} x_{13} \\ y_{13} \\ z_{13} \end{bmatrix}^T \right]^T$$

where x_i, y_i and z_i are the coordinates written in LVLH frame of the relative position vector $\boldsymbol{\chi}_i$ of the i^{th} s/c ($i = 1, 2, ..., N$) with respect to the LVLH frame, as shown in Fig. 6-10, and specified as follows,

$$\boldsymbol{\chi}_{LVLHi}^{LVLH} = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T \tag{6.6}$$

or written as $\boldsymbol{\chi}_{LVLHi}^{LVLH} = \boldsymbol{\chi}_i^{LVLH}$, where $\boldsymbol{\chi}_{ij}$, $i, j = 1, 2, 3$ $i \neq j$ the relative position between s/c i and s/c j,

$$\boldsymbol{\chi}_{ij} = \boldsymbol{\chi}_j - \boldsymbol{\chi}_i = -\boldsymbol{\chi}_{ji} \tag{6.7}$$

and the relative distance as follows,

$$\rho_{ij} = \|\boldsymbol{\chi}_i - \boldsymbol{\chi}_j\| = \|\boldsymbol{\chi}_j - \boldsymbol{\chi}_i\| \tag{6.8}$$

The state variables are related to the LVLH frame which is located, for now, in a generic point, as shown in Fig. 6-10. Thus, using (6.6)

$$\boldsymbol{\chi}_{ij} = \begin{bmatrix} x_j - x_i & y_j - y_i & z_j - z_i \end{bmatrix}^T$$

as shown in Fig. 6-11.

The same is applied for the translational velocity,

$$\begin{aligned} \dot{\boldsymbol{\chi}}_{b_i b_j}^{LVLH} &= \begin{bmatrix} (\dot{\boldsymbol{\chi}}_{12})^T & (\dot{\boldsymbol{\chi}}_{32})^T & (\dot{\boldsymbol{\chi}}_{13})^T \end{bmatrix}^T \\ &= \begin{bmatrix} (\dot{\boldsymbol{\chi}}_2 - \dot{\boldsymbol{\chi}}_1)^T & (\dot{\boldsymbol{\chi}}_2 - \dot{\boldsymbol{\chi}}_3)^T & (\dot{\boldsymbol{\chi}}_3 - \dot{\boldsymbol{\chi}}_1)^T \end{bmatrix}^T \\ &= \left[\begin{bmatrix} \dot{x}_2 - \dot{x}_1 \\ \dot{y}_2 - \dot{y}_1 \\ \dot{z}_2 - \dot{z}_1 \end{bmatrix}^T \quad \begin{bmatrix} \dot{x}_2 - \dot{x}_3 \\ \dot{y}_2 - \dot{y}_3 \\ \dot{z}_2 - \dot{z}_3 \end{bmatrix}^T \quad \begin{bmatrix} \dot{x}_3 - \dot{x}_1 \\ \dot{y}_3 - \dot{y}_1 \\ \dot{z}_3 - \dot{z}_1 \end{bmatrix}^T \right]^T \\ &= \left[\begin{bmatrix} \dot{x}_{12} \\ \dot{y}_{12} \\ \dot{z}_{12} \end{bmatrix}^T \quad \begin{bmatrix} \dot{x}_{32} \\ \dot{y}_{32} \\ \dot{z}_{32} \end{bmatrix}^T \quad \begin{bmatrix} \dot{x}_{13} \\ \dot{y}_{13} \\ \dot{z}_{13} \end{bmatrix}^T \right]^T \end{aligned} \tag{6.9}$$

6.3.3 Rotational state vector

Assuming that the rotational part of the 3-FF s/c state vector is defined in the inertial IPQ frame, defined in the subsection 6.3.1, as the state vector that relates body frame b_i written inertial frame IPQ,

$$\begin{aligned}\mathbf{x}_{rot'}(k) \quad = \quad \Big[\; & (q_1)^{b_1}_{IPQ} \quad (q_2)^{b_1}_{IPQ} \quad (q_3)^{b_1}_{IPQ} \quad (\omega_x)^{IPQ}_{b_i} \quad (\omega_y)^{IPQ}_{b_i} \quad (\omega_z)^{IPQ}_{b_i} \\ & (q_1)^{b_2}_{IPQ} \quad (q_2)^{b_2}_{IPQ} \quad (q_3)^{b_2}_{IPQ} \quad (\omega_x)^{IPQ}_{b_i} \quad (\omega_y)^{IPQ}_{b_i} \quad (\omega_z)^{IPQ}_{b_i} \\ & (q_1)^{b_3}_{IPQ} \quad (q_2)^{b_3}_{IPQ} \quad (q_3)^{b_3}_{IPQ} \quad (\omega_x)^{IPQ}_{b_i} \quad (\omega_y)^{IPQ}_{b_i} \quad (\omega_z)^{IPQ}_{b_i} \; \Big]^T\end{aligned} \tag{6.10}$$

The fourth component of the quaternion, q_4, which is required for the state vector, can be rebuilt through (4.54).

The attitude matrix that transforms vectors from LVLH frame in body frame can be computed by,

$$\mathbf{A}\left(q^{b_i}_{LVLH}\right) = \mathbf{A}\left(q^{b_i}_{IPQ}\right)\mathbf{A}^{IPQ}_{LVLH}$$

where the rotation matrix $\mathbf{A}^{IPQ}_{LVLH}$ depends on orbital parameters from (6.3), and where $\mathbf{A}\left(q^{b_i}_{IPQ}\right)$ is the attitude matrix (4.8), expressed in terms of the quaternion estimated in the rotational absolute state vector (6.10). In terms of quaternions, knowing that the quaternion q^{IPQ}_{LVLH} can be obtained from the rotation matrix $\mathbf{A}^{IPQ}_{LVLH}$, then the quaternion $q^{b_i}_{LVLH}$ is,

$$q^{b_i}_{LVLH} = q^{b_i}_{IPQ} \otimes q^{IPQ}_{LVLH} \tag{6.11}$$

where $\otimes$ is the quaternion multiplication, defined in (4.9).

6.3.4 Translational linearized equations of motion

As derived in detail in the appendix C, the differential equations of motion will be expressed with respect to the true anomaly θ_t because the true anomaly of the reference orbit increases monotonically with time t and provides a natural basis for parameterizing the fleet time and motion [10]. For elliptic orbits, the equation relating the time $t - t_p$ directly with the true anomaly is [51, p.22]:

$$t - t_p = \frac{1}{n_t}\left[2\arctan\left(\sqrt{\frac{1-e_{cc}}{1+e_{cc}}}\tan\frac{\theta_t}{2}\right) - \frac{e\sqrt{1-e^2_{cc}}\sin\theta_t}{1+e_{cc}\cos\theta_t}\right] \tag{6.12}$$

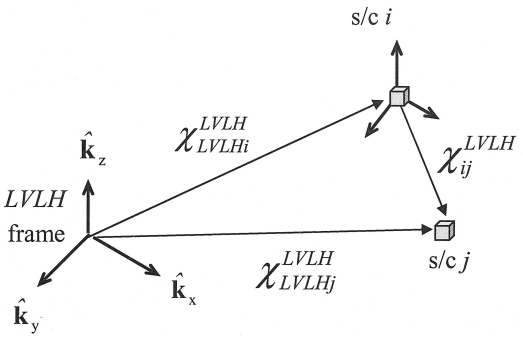

Figure 6-10: Generic position of two spacecraft and the position representation of each spacecraft *w.r.t.* the LVLH frame. The relative distance between spacecraft is also represented.

where t_p is the passage time at the Perigee. By deriving this equation, we obtain:

$$\frac{dt}{d\theta_t} = \frac{\sqrt{1-e_{cc}^2}}{n_t}\left[\frac{1}{(1+e_{cc})\cos^2\frac{\theta_t}{2} + (1-e_{cc})\sin^2\frac{\theta_t}{2}} - \frac{e_{cc}(e_{cc}+\cos\theta_t)}{(1+e_{cc}\cos\theta_t)^2}\right] \tag{6.13}$$

Therefore, for this problem we will have the translational relative velocity as a function of the true anomaly and not from time as in (6.9),

$$\begin{aligned}
\boldsymbol{\chi}'^{\,LVLH}_{b_i b_j} &= \left[\begin{array}{ccc} (\boldsymbol{\chi}'_{12})^T & (\boldsymbol{\chi}'_{32})^T & (\boldsymbol{\chi}'_{13})^T \end{array}\right]^T \\
&= \left[\begin{array}{ccc} (\boldsymbol{\chi}'_2 - \boldsymbol{\chi}'_1)^T & (\boldsymbol{\chi}'_2 - \boldsymbol{\chi}'_3)^T & (\boldsymbol{\chi}'_3 - \boldsymbol{\chi}'_1)^T \end{array}\right]^T \\
&= \left[\begin{array}{ccc} \begin{bmatrix} x'_2 - x'_1 \\ y'_2 - y'_1 \\ z'_2 - z'_1 \end{bmatrix}^T & \begin{bmatrix} x'_2 - x'_3 \\ y'_2 - y'_3 \\ z'_2 - z'_3 \end{bmatrix}^T & \begin{bmatrix} x'_3 - x'_1 \\ y'_3 - y'_1 \\ z'_3 - z'_1 \end{bmatrix}^T \end{array}\right]^T
\end{aligned} \tag{6.14}$$

where x'_i, y'_i, z'_i, denotes the derivative *w.r.t* the true anomaly θ_t, respectively, $x'_i = \frac{dx_i}{d\theta_t}$, $y'_i = \frac{dy_i}{d\theta_t}$, $z'_i = \frac{dz_i}{d\theta_t}$.

The set of linearized θ_t-varying equations which describes the relative motion between the i^{th} s/c and LVLH, in an eccentric orbit, expressed in LVLH is [70], [10], and deducted in appendix C and given by (C.21),

$$\frac{d}{d\theta_t}\begin{bmatrix} y_i \\ y'_i \end{bmatrix}_i^{LVLH} = \begin{bmatrix} 0 & 1 \\ f_{52} & f_{43} \end{bmatrix}\begin{bmatrix} y_i \\ y'_i \end{bmatrix}_i^{LVLH} + \xi(\theta_t)\begin{bmatrix} 0 \\ 1 \end{bmatrix}\iota_{i,y}^{IPQ} \tag{6.15a}$$

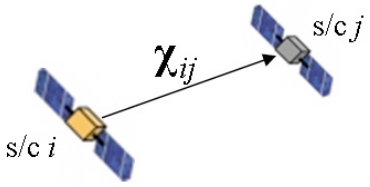

Figure 6-11: Representation of a relative vector where $\boldsymbol{\chi}_{ij} = \boldsymbol{\chi}_j - \boldsymbol{\chi}_i = -\boldsymbol{\chi}_{ji}$ and its absolute value is $\boldsymbol{\rho}_{ij} = \|\boldsymbol{\chi}_i - \boldsymbol{\chi}_j\| = \|\boldsymbol{\chi}_j - \boldsymbol{\chi}_i\|$.

$$\frac{d}{d\theta_t}\begin{bmatrix} x_i \\ x_i' \\ z_i \\ z_i' \end{bmatrix}_i^{LVLH} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ f_{41} & f_{43} & -f_{43} & 2 \\ 0 & 0 & 0 & 1 \\ f_{43} & -2 & f_{63} & f_{43} \end{bmatrix} \begin{bmatrix} x_i \\ x_i' \\ z_i \\ z_i' \end{bmatrix}_i^{LVLH} + \xi(\theta_t) \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \iota_x \\ \iota_z \end{bmatrix}_i^{IPQ} \tag{6.16}$$

with $\xi(\theta_t) = \frac{(1-e_{cc}^2)^3}{(1+e_{cc}\cos\theta_t)^4 n_t^2}$, $f_{41} = \frac{e_{cc}\cos\theta_t}{1+e_{cc}\cos\theta_t}$, $f_{43} = \frac{2e_{cc}\sin\theta_t}{1+e_{cc}\cos\theta_t}$, $f_{52} = \frac{-1}{1+e_{cc}\cos\theta_t}$, $f_{63} = \frac{3+e_{cc}\cos\theta_t}{1+e_{cc}\cos\theta_t}$, with $\boldsymbol{\chi}_{iLVLH}^{LVLH} = \boldsymbol{\chi}_i^{LVLH}$.

The relative positions x_i, y_i and z_i and the relative velocities x_i', y_i' and z_i' characterize the state of the i^{th} s/c with respect to the formation center.

$\iota_{i,x}^{IPQ}$, $\iota_{i,y}^{IPQ}$ and $\iota_{i,z}^{IPQ}$ are the components in IPQ frame of the external forces vector ι_i, which includes the control inputs $\mathbf{u}_i$ acting on the i^{th} s/c and the differential (relative) perturbations experienced by the i^{th} s/c *w.r.t.* the perturbations affecting the formation center:

$$\boldsymbol{\iota}_i = \mathbf{u}_i + \mathbf{w}_{i,J2} + \mathbf{w}_{i,sun} + \mathbf{w}_{i,moon} \tag{6.17}$$

In expression (6.17), $\mathbf{w}_{i,J2}$ is the differential perturbation due to $J2$ effect; $\mathbf{w}_{i,sun}$ and $\mathbf{w}_{i,moon}$ are the third-body (Sun, Moon) gravitational differential perturbations. Of course, other perturbations (atmospheric drag, solar radiation pressure, micrometeoroids) may be taken into account, but for the moment no disturbances are considered at all.

Letting aside the control inputs and perturbations, the in-plane (x and z) and out-of-plane (y) motions are decoupled and their equations can be integrated separately (equations 6.16) for in-plane motion and equations (6.15a) for out-of-plane motion. The θ_t-varying relative dynamic equations (6.16) and (6.15a) are nonlinear. This nonlinearity comes from the differential perturbation due to $J2$ effect and from the third-body gravitational perturbations. Letting aside the perturbations and the control inputs, the θ_t-varying relative dynamic equations (6.16) and (6.15a) are linear and we are able to find their homogenous solutions.

6.3.5 Rotational equations of motion

In IPQ, the attitude dynamics is straightforward to compute and there is no correlation between measurements compared to when relative attitude is used. The full nonlinear motion model of the system is given by the kinematic and dynamic equations, respectively. The kinematic equation,

$$\begin{aligned} \frac{d}{dt} q_{IPQ}^{b_i}(t) &= \frac{1}{2} \boldsymbol{\Xi} \left(q_{IPQ}^{b_i} \right) \omega_{b_i}^{IPQ} = \frac{1}{2} \boldsymbol{\Lambda} \left(\omega_{b_i}^{IPQ} \right) q_{IPQ}^{b_i} \\ &= \frac{1}{2} \omega_{b_i}^{IPQ} \otimes q_{IPQ}^{b_i} \end{aligned}$$

where the angular velocity rate components concern the rotation of the body frame written in the IPQ frame, and defined as a quaternion,

$$\begin{aligned} \omega_{b_i}^{IPQ} &= \begin{bmatrix} (\omega_x)_{b_i}^{IPQ} & (\omega_y)_{b_i}^{IPQ} & (\omega_z)_{b_i}^{IPQ} & 0 \end{bmatrix}^T \\ &= \begin{bmatrix} \left(\boldsymbol{\omega}_{b_i}^{IPQ} \right)^T & 0 \end{bmatrix}^T \end{aligned}$$

The dynamic rotational equation of motion for a s/c considered as a rigid body, relates its angular velocity with the external forces. Assuming that the principle of conservation of angular momentum applies and the body axis of rotation is parallel to the body angular velocity, then the dynamic rotational equation of motion for an orbiting s/c, also known as Euler's equations of motion, is given by,

$$\mathbf{I} \dot{\boldsymbol{\omega}} = \sum \mathbf{m}_{ext} - \boldsymbol{\omega} \times \mathbf{I} \boldsymbol{\omega} \tag{6.18}$$

where $\boldsymbol{\omega}$ is the angular velocity of the rigid body written in the IPQ frame, $\mathbf{I}$ is the moment of inertia and $\sum \mathbf{m}_{ext}$ are the sum of the external moments acting on the s/c j. Solving the equation in order to the rate of change of the angular velocity, follows,

$$\dot{\boldsymbol{\omega}}_{b_i}^{IPQ} = \mathbf{I}^{-1} \left(\sum \mathbf{m}_{ext} - \boldsymbol{\omega}_{b_i}^{IPQ} \times \mathbf{I} \boldsymbol{\omega}_{b_i}^{IPQ} \right) \tag{6.19}$$

6.3.6 Implementation

In this subsection, the implementation details of the estimators for the full 6DoF state vector using RF measurements to update the translational relative state vector and the ST measurements for the filtering part of the rotational absolute state vector, are presented. The translational and rotational components of the vector can be estimated separately, but the translational part requires knowledge of the current estimates of the rotational components, which are required in the filtering part for the computation of the nonlinear observation function, in (6.25).

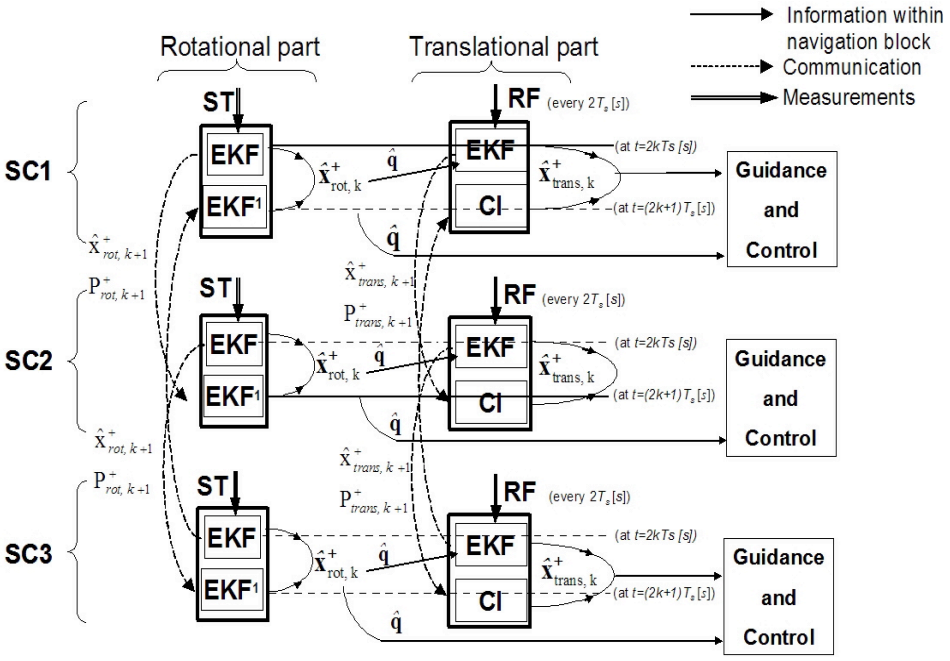

Figure 6-12: Diagramatic view of the communication between s/c and between navigation and GC, filtering and prediction times. The estimates of attitude and angular velocity are used online by the relative position and translational velocity navigation algorithm.

At each sampling time, the relative position and translational velocity (translational) algorithm, receives the estimates from the absolute inertial attitude and angular velocity (6.10), computed by the rotational algorithm, which are required in the filtering part for the computation of the observation model. The filtering part of the estimator algorithm that computes the translational relative state vector, is divided in two parts: Kalman filtering and CI method, depending on the source of the measurements. As soon as a measurement of the RF subsystem arrives, the filtering part is computed (using Kalman filtering) and the result is the state vector estimation for that time instant, which is then sent to the next s/c, as shown in Fig. 4-12. In the next sampling time the predecessor state vector is combined with the local state vector through the covariance intersection method, resulting in a new state vector estimate.

The filtering part of the rotational estimator algorithm that computes the inertial attitude and the inertial angular velocity is divided in two parts: Kalman filtering and Kalman filtering Fusion method, depending on the source of the measurements. As soon as a measurement of the ST sensor arrives, the filtering part is computed and the result is the state vector estimation for that time instant, which is then sent to the next s/c, as shown in Fig. 4-13. In the next sampling time the predecessor state vector is combined with the local state vector through the Kalman filter Fusion, adapted to the fact that

the observations for this filter are now the predecessor state estimates, resulting in a new state vector estimate.

The sampling times of the estimator algorithms were synchronized and run every T_s. At instants $t = 2k.T_s[s]$, with $k = 0, 1, 2, ...$ a new translational relative state vector estimate is computed in s/c i, $\hat{\mathbf{x}}^i_{trans}$, based on the RF measurements defined by the *measurement* graph, $¥^i$, as well as a new rotational absolute state vector estimate, $\hat{\mathbf{x}}^i_{rot}$. Then, it is communicated to the next s/c $(i+1)$, represented by links in the peer-to-peer *information flow* graph, $\wp^P$, and processed at $t = (2k+1).T_s[s]$, $\mathbf{z}^{i+1}(t+T_s) = \hat{\mathbf{x}}^i_{trans}$, for the translational filter estimator, using the CI method, and $\mathbf{z}^{i+1}(t+T_s) = \hat{\mathbf{x}}^i_{rot}$, for the rotational filter estimator, using EKF Fusion, as shown in Fig. 6-12. The RF/ST measurements are received every $2T_s$ seconds by the translational relative filter estimator and the rotational absolute filter estimator, respectively, and the measurements coming from the current estimate of the previous s/c are processed in between, as shown in Fig 6-12. After the filtering step the state vector is sent to the GC algorithm.

Translational state vector

Here the derivation of the equations related to EKF implementation using RF measurements, are performed. The nonlinear observation model is computed using the translational relative state vector. The linearization of the process and observation models are performed as well.

Throughout this subsection, for the sake of simplifying the notation, we will use $\mathbf{x}$ to denote the translational state $\mathbf{x}_{trans}$.

Extended Kalman filter filtering - radio frequency measurements The notation for the relative distance will differ because it will depend on whether the spacecraft is transmitting or receiving signals and the transformation from body reference frame to LVLH reference frame. Thus the RF measurements in the case of **pseudo-range** signals are represented as follows [69]:

$$\begin{aligned} \rho^i_j &= \left\|\boldsymbol{\chi}_j - \boldsymbol{\chi}_i\right\| + c\left(T^i - T^j\right) + E^{i,j}_{multipath\rho} + \varepsilon^{i,j}_{\rho} \\ &= \rho_{ji} + c\left(T^i - T^j\right) + \nu^j \end{aligned} \tag{6.20}$$

$\rho^i_j$where,

$\boldsymbol{\chi}_j$ is the position state vector of vehicle j (likewise for $\boldsymbol{\chi}_i$)

ρ^i_j is the code phase between transmitting spacecraft j and receiver spacecraft i,

$\rho_{ji} = \left\|\boldsymbol{\chi}_j - \boldsymbol{\chi}_i\right\| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$ is the equation that relates the true distance, between the spacecraft i at time of signal transmission and spacecraft j at measurement time,

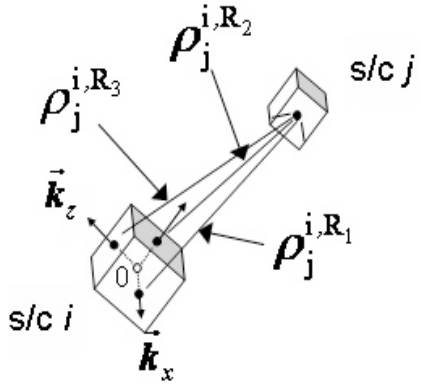

Figure 6-13: Three measurements, ρ_j^{i,R_1}, ρ_j^{i,R_2}, ρ_j^{i,R_3}, transmitted from spacecraft i are received by each antenna, R_1, R_2, R_3, on spacecraft i.

with the formation state, in body reference frame

T^i and T^j are the time bias of the receiving and transmitting s/c clock, respectively,

c is the speed of light $(3 \times 10^8\,[m/s])$,

$\varepsilon_\rho^{i,j}$ is the pseudorange measurement noise due to the receiver thermal noise,

$E_{multipath\rho}^{i,j}$ represents the multipath error.

The three antennas, R_1, R_2, R_3, are placed in the sides of the spacecraft in the positions $\begin{bmatrix} ap & 0 & 0 \end{bmatrix}$, $\begin{bmatrix} 0 & ap & 0 \end{bmatrix}$, $\begin{bmatrix} 0 & 0 & ap \end{bmatrix}$ meters, respectively, *w.r.t.* the s/c body frame shown in Fig. 6-9. Therefore, when s/c i is receiving signals from s/c j:

for receiving antenna 1:

$$\rho_j^{i,R_1} = \left\|\boldsymbol{\chi}_j - \boldsymbol{\chi}_i\right\|^{R_1} + c\left(T^j - T^i\right) + E_{multipath\rho}^{i,j} + \varepsilon_\rho^{i,j}$$

for receiving antenna 2:

$$\rho_j^{i,R_2} = \left\|\boldsymbol{\chi}_j - \boldsymbol{\chi}_i\right\|^{R_2} + c\left(T^j - T^i\right) + E_{multipath\rho}^{i,j} + \varepsilon_\rho^{i,j}$$

for receiving antenna 3:

$$\rho_j^{i,R_3} = \left\|\boldsymbol{\chi}_j - \boldsymbol{\chi}_i\right\|^{R_3} + c\left(T^j - T^i\right) + E_{multipath\rho}^{i,j} + \varepsilon_\rho^{i,j}$$

as shown in Fig. 6-13.

However the receiver antennas are not placed in the center of the s/c, but placed ap metre ahead on each side of the s/c as shown in Fig 6-14. Antenna number 1 is placed ap metre ahead from the gravity center in x direction of the body frame, antenna number 2 is placed ap metre ahead from the gravity center in y direction of the body frame, antenna number 3 is placed ap metre ahead from the gravity

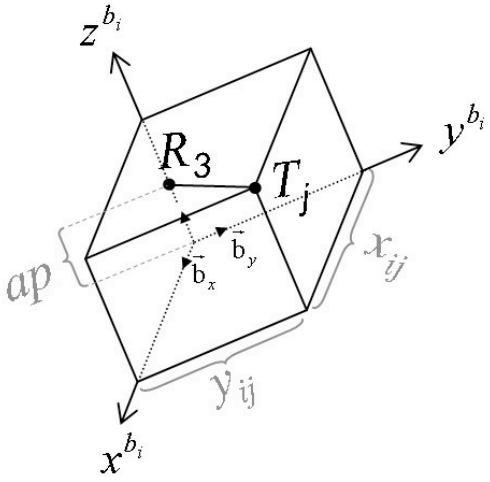

Figure 6-14: Geometric representation of the measurement received by antenna 3 in the s/c i and transmitted by the s/c j.

center in z direction of the body frame, as follows,

$$\begin{aligned} \mathbf{c}_{R_1}^{b_i} &= \begin{bmatrix} ap & 0 & 0 \end{bmatrix}^T \\ \mathbf{c}_{R_2}^{b_i} &= \begin{bmatrix} 0 & ap & 0 \end{bmatrix}^T \\ \mathbf{c}_{R_3}^{b_i} &= \begin{bmatrix} 0 & 0 & ap \end{bmatrix}^T \end{aligned} \tag{6.21}$$

with $i = 1, ..., N$, all expressed in body reference frame.

Thus, there are three measurements instead of just one, as shown in (6.20). Therefore, the relation between measurements received in the three antennas, $\rho_i^{j,R_1}, \rho_i^{j,R_2}, \rho_i^{j,R_3}$ and states, expressed in IPQ frame, is (see Fig. 6-14):

$$\left\|\boldsymbol{\chi}_j - \boldsymbol{\chi}_i\right\|^{R_1} = \sqrt{\left(x_{ij}^{IPQ} - c_{R_1,x}^{IPQ}\right)^2 + \left(y_{ij}^{IPQ} - c_{R_1,y}^{IPQ}\right)^2 + \left(z_{ij}^{IPQ} - c_{R_1,z}^{IPQ}\right)^2} \tag{6.22}$$

$$\left\|\boldsymbol{\chi}_j - \boldsymbol{\chi}_i\right\|^{R_2} = \sqrt{\left(x_{ij}^{IPQ} - c_{R_2,x}^{IPQ}\right)^2 + \left(y_{ij}^{IPQ} - c_{R_2,y}^{IPQ}\right)^2 + \left(z_{ij}^{IPQ} - c_{R_2,z}^{IPQ}\right)^2} \tag{6.23}$$

$$\left\|\boldsymbol{\chi}_j - \boldsymbol{\chi}_i\right\|^{R_3} = \sqrt{\left(x_{ij}^{IPQ} - c_{R_3,x}^{IPQ}\right)^2 + \left(c_{ij}^{IPQ} - c_{R_3,y}^{IPQ}\right)^2 + \left(z_{ij}^{IPQ} - c_{R_3,z}^{IPQ}\right)^2} \tag{6.24}$$

where $\mathbf{c}_{R_l}^{IPQ}$, $l = 1, 2, 3$ can be built up from $\mathbf{c}_{R_l}^{b_i}$ defined in (6.21) through the following transformation,

$$\mathbf{c}_{R_l}^{IPQ} = \mathbf{A}^T\left(q_{IPQ}^{b_i}\right)\mathbf{c}_{R_l}^{b_i},\ i = 1, .., N, \text{ and, } l = 1, 2, 3 \text{ with } i \neq l \tag{6.25}$$

$$\left[\begin{array}{ccc} c_{R_l,x}^{IPQ} & c_{R_l,y}^{IPQ} & c_{R_l,y}^{IPQ} \end{array}\right]^T = \mathbf{A}^T\left(q_{IPQ}^{b_i}\right)\left[\begin{array}{ccc} c_{R_l,x}^{b_i} & c_{R_l,y}^{b_i} & c_{R_l,y}^{b_i} \end{array}\right]^T$$

with $\mathbf{A}\left(q_{IPQ}^{b_i}\right)$ is the attitude matrix (4.8), expressed in terms of the quaternion estimated in the rotational state vector (6.10).

Since the state vector is expressed in LVLH reference frame, the state variables should be transformed into the LVLH reference frame. Thus, any vector $\mathbf{x}$ in IPQ reference frame can be transformed into the LVLH frame by

$$\mathbf{x}^{LVLH} = \left(\mathbf{A}_{LVLH}^{IPQ}\right)^T \mathbf{x}^{IPQ}$$

where $\mathbf{A}_{LVLH}^{IPQ}$ is defined in Eq. (6.3).

In particular,

$$\left[\begin{array}{ccc} x_{ij}^{IPQ} & y_{ij}^{IPQ} & z_{ij}^{IPQ} \end{array}\right]^T = \left(\mathbf{A}_{LVLH}^{IPQ}\right)\left[\begin{array}{ccc} x_{ij}^{LVLH} & y_{ij}^{LVLH} & z_{ij}^{LVLH} \end{array}\right]^T$$

or

$$\begin{aligned} x_{ij}^{IPQ} &= a_{11}x_{ij}^{LVLH} + a_{12}y_{ij}^{LVLH} + a_{13}z_{ij}^{LVLH} \\ y_{ij}^{IPQ} &= a_{21}x_{ij}^{LVLH} + a_{22}y_{ij}^{LVLH} + a_{23}z_{ij}^{LVLH} \\ z_{ij}^{IPQ} &= a_{31}x_{ij}^{LVLH} + a_{32}y_{ij}^{LVLH} + a_{33}z_{ij}^{LVLH} \end{aligned}$$

where $a_{n,m}$ are the components of the matrix $\mathbf{A}_{LVLH}^{IPQ}$.

Applying the previous equation to the measurements given by (6.22) - (6.24), the observations obtained by each receiving antenna are then expressed as a function of the state variables in LVLH frame as follows,

$$\begin{aligned} \left(\left\|\boldsymbol{\chi}_j - \boldsymbol{\chi}_i\right\|^{R_1}\right)^2 &= \left(a_{11}x_{ij}^{LVLH} + a_{12}y_{ij}^{LVLH} + a_{13}z_{ij}^{LVLH} - c_{R_1,x}^{IPQ}\right)^2 + \\ &\quad \left(a_{21}x_{ij}^{LVLH} + a_{22}y_{ij}^{LVLH} + a_{23}z_{ij}^{LVLH} - c_{R_1,y}^{IPQ}\right)^2 + \\ &\quad \left(a_{31}x_{ij}^{LVLH} + a_{32}y_{ij}^{LVLH} + a_{33}z_{ij}^{LVLH} - c_{R_1,z}^{IPQ}\right)^2 \\ \left(\left\|\boldsymbol{\chi}_j - \boldsymbol{\chi}_i\right\|^{R_2}\right)^2 &= \left(a_{11}x_{ij}^{LVLH} + a_{12}y_{ij}^{LVLH} + a_{13}z_{ij}^{LVLH} - c_{R_2,x}^{IPQ}\right)^2 + \\ &\quad \left(a_{21}x_{ij}^{LVLH} + a_{22}y_{ij}^{LVLH} + a_{23}z_{ij}^{LVLH} - c_{R_2,y}^{IPQ}\right)^2 + \\ &\quad \left(a_{31}x_{ij}^{LVLH} + a_{32}y_{ij}^{LVLH} + a_{33}z_{ij}^{LVLH} - c_{R_2,z}^{IPQ}\right)^2 \end{aligned}$$

$$\begin{aligned}\left(\left\|\boldsymbol{\chi}_j-\boldsymbol{\chi}_i\right\|^{R_3}\right)^2 &= \left(a_{11}x_{ij}^{LVLH}+a_{12}y_{ij}^{LVLH}+a_{13}z_{ij}^{LVLH}-c_{R_3,x}^{IPQ}\right)^2+\\ &\left(a_{21}x_{ij}^{LVLH}+a_{22}y_{ij}^{LVLH}+a_{23}z_{ij}^{LVLH}-c_{R_3,y}^{IPQ}\right)^2+\\ &\left(a_{31}x_{ij}^{LVLH}+a_{32}y_{ij}^{LVLH}+a_{33}z_{ij}^{LVLH}-c_{R_3,z}^{IPQ}\right)^2\end{aligned}$$

Radio frequency observation model The observation model is nonlinear, according to the observation equations given by (6.20).

$$\mathbf{y}^i(k)=h^i(\mathbf{x}(k))+\boldsymbol{\nu}^i(k)$$

where $\mathbf{y}(k)$ is the n_y-dimensional observation vector of vehicle i at time k, and where $\boldsymbol{\nu}(k)$ is observation noise, assumed to be white Gaussian with,

$$\begin{aligned}E\left\{\boldsymbol{\nu}(k)\boldsymbol{\nu}^T(k)\right\} &= \mathbf{R}(k)\\ E\left\{\boldsymbol{\nu}(k)\right\} &= 0\end{aligned}$$

The linearized model to be used by the EKF is derived in appendix B, Eq. (B.7),

$$\boldsymbol{\delta}\mathbf{y}^i(k)=\mathbf{H}^i(\hat{\mathbf{x}}^i(k),k)\delta\mathbf{x}(k)+\boldsymbol{\nu}^i(k)$$

where $\delta\mathbf{x}$ is the small perturbation, defined as follows,

$$\delta\mathbf{x}\left(k\right)\triangleq\mathbf{x}\left(k\right)-\hat{\mathbf{x}}\left(k\right) \tag{6.26}$$

The previous equations lead to the linear pseudo-measurement equation for the Kalman filter,

$$\mathbf{y}^i(k)\triangleq\mathbf{H}^i(k)\boldsymbol{\chi}(k)+\boldsymbol{\nu}^i(k)$$

which comparing with (6.20), i is the receiver spacecraft and $\boldsymbol{\nu}^i$ represents the combination of the pseudo-range measurement noise due to the receiver thermal noise, $\varepsilon_\rho^{i,j}$ and the multipath error $E_{multipath,\rho}^{i,j}$. No time bias is considered for the RF measurements. Also, for each row of the linearized observation matrix $\mathbf{H}^i$ for receiving s/c i is given by:

$$\mathbf{H}^i\left(k\right)=\left.\frac{\partial\rho_j^i}{\partial\chi}\right|_{\chi=\hat{\chi}(k|k-1)}=\begin{bmatrix}\vdots & \vdots & \vdots\\ \mathbf{h}_{j,x}^i(k) & \mathbf{h}_{j,y}^i\left(k\right) & \mathbf{h}_{j,z}^i\left(k\right)\\ \vdots & \vdots & \vdots\end{bmatrix} \tag{6.27}$$

so that, where ρ_j^i is defined in (6.20).

$$\mathbf{h}_{j,x}^i(k) = \begin{bmatrix} \frac{\partial \rho_j^{i,R_1}}{\partial x_{ij}^{LVLH}}\Big|_{\chi=\hat{\chi}(k|k-1)} \\ \frac{\partial \rho_j^{i,R_2}}{\partial x_{ij}^{LVLH}}\Big|_{\chi=\hat{\chi}(k|k-1)} \\ \frac{\partial \rho_j^{i,R_3}}{\partial x_{ij}^{LVLH}}\Big|_{\chi=\hat{\chi}(k|k-1)} \end{bmatrix}, \mathbf{h}_{i,y}^j(k) = \begin{bmatrix} \frac{\partial \rho_j^{i,R_1}}{\partial y_{ij}^{LVLH}}\Big|_{\chi=\hat{\chi}(k|k-1)} \\ \frac{\partial \rho_j^{i,R_2}}{\partial y_{ij}^{LVLH}}\Big|_{\chi=\hat{\chi}(k|k-1)} \\ \frac{\partial \rho_j^{i,R_3}}{\partial y_{ij}^{LVLH}}\Big|_{\chi=\hat{\chi}(k|k-1)} \end{bmatrix}, \mathbf{h}_{i,z}^j(k) = \begin{bmatrix} \frac{\partial \rho_j^{i,R_1}}{\partial z_{ij}^{LVLH}}\Big|_{\chi=\hat{\chi}(k|k-1)} \\ \frac{\partial \rho_j^{i,R_2}}{\partial z_{ij}^{LVLH}}\Big|_{\chi=\hat{\chi}(k|k-1)} \\ \frac{\partial \rho_j^{i,R_3}}{\partial z_{ij}^{LVLH}}\Big|_{\chi=\hat{\chi}(k|k-1)} \end{bmatrix}$$

where j is the index number of the transmitting spacecraft, $j = 1, 2, 3, i \neq j$. Since in our case each receiving s/c has 3 antennas, each term $\mathbf{h}_{j,*}^i$ has 3 rows, and as each s/c measures the distance to 2 s/c, each $\mathbf{H}^i$, $i = 1$, 2, 3 has 6 rows. The detailed expressions for the $\mathbf{h}_{j,*}^i$ follows,

$$\mathbf{h}_{j,x}^i(k) = \begin{bmatrix} \frac{\partial(\rho_j^{i,R_1})}{\partial(x_{ij}^b)}\frac{\partial(x_{ij}^b)}{\partial(x_{ij}^{LVLH})} + \frac{\partial(\rho_j^{i,R_1})}{\partial(y_{ij}^b)}\frac{\partial(y_{ij}^b)}{\partial(x_{ij}^{LVLH})} + \frac{\partial(\rho_j^{i,R_1})}{\partial(z_{ij}^b)}\frac{\partial(z_{ij}^b)}{\partial(x_{ij}^{LVLH})}\Big|_{\chi=\hat{\chi}(k|k-1)} \\ \frac{\partial(\rho_j^{i,R_2})}{\partial(x_{ij}^b)}\frac{\partial(x_{ij}^b)}{\partial(x_{ij}^{LVLH})} + \frac{\partial(\rho_j^{i,R_2})}{\partial(y_{ij}^b)}\frac{\partial(y_{ij}^b)}{\partial(x_{ij}^{LVLH})} + \frac{\partial(\rho_j^{i,R_2})}{\partial(z_{ij}^b)}\frac{\partial(z_{ij}^b)}{\partial(x_{ij}^{LVLH})}\Big|_{\chi=\hat{\chi}(k|k-1)} \\ \frac{\partial(\rho_j^{i,R_3})}{\partial(x_{ij}^b)}\frac{\partial(x_{ij}^b)}{\partial(x_{ij}^{LVLH})} + \frac{\partial(\rho_j^{i,R_3})}{\partial(y_{ij}^b)}\frac{\partial(y_{ij}^b)}{\partial(x_{ij}^{LVLH})} + \frac{\partial(\rho_j^{i,R_3})}{\partial(z_{ij}^b)}\frac{\partial(z_{ij}^b)}{\partial(x_{ij}^{LVLH})}\Big|_{\chi=\hat{\chi}(k|k-1)} \end{bmatrix}$$

$$\mathbf{h}_{j,x}^i(k) = \begin{bmatrix} \frac{h1_x}{\|\boldsymbol{\chi}_i - \boldsymbol{\chi}_j\|^{R_1}} \\ \frac{h2_x}{\|\boldsymbol{\chi}_i - \boldsymbol{\chi}_j\|^{R_2}} \\ \frac{h3_x}{\|\boldsymbol{\chi}_i - \boldsymbol{\chi}_j\|^{R_3}} \end{bmatrix}$$

with

$$\begin{aligned} h1_x &= a_{11}(a_{11}x_{ij}^{LVLH} + a_{12}y_{ij}^{LVLH} + a_{13}z_{ij}^{LVLH} - c_{R_1,x}^{IPQ}) + \\ &\quad a_{21}(a_{21}x_{ij}^{LVLH} + a_{22}y_{ij}^{LVLH} + a_{23}z_{ij}^{LVLH} - c_{R_1,y}^{IPQ}) + \\ &\quad a_{31}(a_{31}x_{ij}^{LVLH} + a_{32}y_{ij}^{LVLH} + a_{33}z_{ij}^{LVLH} - c_{R_1,z}^{IPQ}) \\ h2_x &= a_{11}(a_{11}x_{ij}^{LVLH} + a_{12}y_{ij}^{LVLH} + a_{13}z_{ij}^{LVLH} - c_{R_2,x}^{IPQ}) + \\ &\quad a_{21}(a_{21}x_{ij}^{LVLH} + a_{22}y_{ij}^{LVLH} + a_{23}z_{ij}^{LVLH} - c_{R_2,y}^{IPQ}) + \\ &\quad a_{31}(a_{31}x_{ij}^{LVLH} + a_{32}y_{ij}^{LVLH} + a_{33}z_{ij}^{LVLH} - c_{R_2,z}^{IPQ}) \\ h3_x &= a_{11}(a_{11}x_{ij}^{LVLH} + a_{12}y_{ij}^{LVLH} + a_{13}z_{ij}^{LVLH} - c_{R_3,x}^{IPQ}) + \\ &\quad a_{21}(a_{21}x_{ij}^{LVLH} + a_{22}y_{ij}^{LVLH} + a_{23}z_{ij}^{LVLH} - c_{R_3,y}^{IPQ}) + \\ &\quad a_{31}(a_{31}x_{ij}^{LVLH} + a_{32}y_{ij}^{LVLH} + a_{33}z_{ij}^{LVLH} - c_{R_3,z}^{IPQ}) \end{aligned}$$

and

$$\mathbf{h}_{i,y}^{j}(k) = \left[\begin{array}{l} \left. \frac{\partial(\rho_j^{i,R_1})}{\partial(x_{ij}^b)} \frac{\partial(x_{ij}^b)}{\partial(y_{ij}^{LVLH})} + \frac{\partial(\rho_j^{i,R_1})}{\partial(y_{ij}^b)} \frac{\partial(y_{ij}^b)}{\partial(y_{ij}^{LVLH})} + \frac{\partial(\rho_j^{i,R_1})}{\partial(z_{ij}^b)} \frac{\partial(z_{ij}^b)}{\partial(y_{ij}^{LVLH})} \right|_{\chi=\hat{\chi}(k|k-1)} \\ \left. \frac{\partial(\rho_j^{i,R_2})}{\partial(x_{ij}^b)} \frac{\partial(x_{ij}^b)}{\partial(y_{ij}^{LVLH})} + \frac{\partial(\rho_j^{j,R_2})}{\partial(y_{ij}^b)} \frac{\partial(y_{ij}^b)}{\partial(y_{ij}^{LVLH})} + \frac{\partial(\rho_j^{i,R_2})}{\partial(z_{ij}^b)} \frac{\partial(z_{ij}^b)}{\partial(y_{ij}^{LVLH})} \right|_{\chi=\hat{\chi}(k|k-1)} \\ \left. \frac{\partial(\rho_j^{i,R_3})}{\partial(x_{ij}^b)} \frac{\partial(x_{ij}^b)}{\partial(z_{ij}^{LVLH})} + \frac{\partial(\rho_j^{i,R_3})}{\partial(y_{ij}^b)} \frac{\partial(y_{ij}^b)}{\partial(y_{ij}^{LVLH})} + \frac{\partial(\rho_j^{i,R_3})}{\partial(z_{ij}^b)} \frac{\partial(z_{ij}^b)}{\partial(y_{ij}^{LVLH})} \right|_{\chi=\hat{\chi}(k|k-1)} \end{array} \right]$$

$$\mathbf{h}_{j,y}^{i}(k) = \left[\begin{array}{l} \frac{h1_y}{\left\|\boldsymbol{\chi}_i - \boldsymbol{\chi}_j\right\|^{R_1}} \\ \frac{h2_y}{\left\|\boldsymbol{\chi}_i - \boldsymbol{\chi}_j\right\|^{R_2}} \\ \frac{h3_y}{\left\|\boldsymbol{\chi}_i - \boldsymbol{\chi}_j\right\|^{R_3}} \end{array} \right]$$

with

$$\begin{aligned} h1_y &= a_{12}(a_{11}x_{ij}^{LVLH} + a_{12}y_{ij}^{LVLH} + a_{13}z_{ij}^{LVLH} - c_{R_1,x}^{IPQ}) + \\ &\quad a_{22}(a_{21}x_{ij}^{LVLH} + a_{22}y_{ij}^{LVLH} + a_{23}z_{ij}^{LVLH} - c_{R_1,y}^{IPQ}) + \\ &\quad a_{32}(a_{31}x_{ij}^{LVLH} + a_{32}y_{ij}^{LVLH} + a_{33}z_{ij}^{LVLH} - c_{R_1,z}^{IPQ}) \\ h2_y &= a_{12}(a_{11}x_{ij}^{LVLH} + a_{12}y_{ij}^{LVLH} + a_{13}z_{ij}^{LVLH} - c_{R_2,x}^{IPQ}) + \\ &\quad a_{22}(a_{21}x_{ij}^{LVLH} + a_{22}y_{ij}^{LVLH} + a_{23}z_{ij}^{LVLH} - c_{R_2,y}^{IPQ}) + \\ &\quad a_{32}(a_{31}x_{ij}^{LVLH} + a_{32}y_{ij}^{LVLH} + a_{33}z_{ij}^{LVLH} - c_{R_2,z}^{IPQ}) \\ h3_y &= a_{12}(a_{11}x_{ij}^{LVLH} + a_{12}y_{ij}^{LVLH} + a_{13}z_{ij}^{LVLH} - c_{R_3,x}^{IPQ}) + \\ &\quad a_{22}(a_{21}x_{ij}^{LVLH} + a_{22}y_{ij}^{LVLH} + a_{23}z_{ij}^{LVLH} - c_{R_3,y}^{IPQ}) + \\ &\quad a_{32}(a_{31}x_{ij}^{LVLH} + a_{32}y_{ij}^{LVLH} + a_{33}z_{ij}^{LVLH} - c_{R_3,z}^{IPQ}) \end{aligned}$$

and

$$\mathbf{h}_{j,z}^{i}(k) = \left[\begin{array}{l} \left. \frac{\partial(\rho_j^{i,R_1})}{\partial(x_{ij}^b)} \frac{\partial(x_{ij}^b)}{\partial(z_{ij}^{LVLH})} + \frac{\partial(\rho_j^{i,R_1})}{\partial(y_{ij}^b)} \frac{\partial(y_{ij}^b)}{\partial(z_{ij}^{LVLH})} + \frac{\partial(\rho_j^{i,R_1})}{\partial(z_{ij}^b)} \frac{\partial(z_{ij}^b)}{\partial(z_{ij}^{LVLH})} \right|_{\chi=\hat{\chi}(k|k-1)} \\ \left. \frac{\partial(\rho_j^{i,R_2})}{\partial(x_{ij}^b)} \frac{\partial(x_{ij}^b)}{\partial(z_{ij}^{LVLH})} + \frac{\partial(\rho_j^{i,R_2})}{\partial(y_{ij}^b)} \frac{\partial(y_{ij}^b)}{\partial(z_{ij}^{LVLH})} + \frac{\partial(\rho_j^{i,R_2})}{\partial(z_{ij}^b)} \frac{\partial(z_{ij}^b)}{\partial(z_{ij}^{LVLH})} \right|_{\chi=\hat{\chi}(k|k-1)} \\ \left. \frac{\partial(\rho_j^{i,R_3})}{\partial(x_{ij}^b)} \frac{\partial(x_{ij}^b)}{\partial(z_{ij}^{LVLH})} + \frac{\partial(\rho_j^{j,R_3})}{\partial(y_{ij}^b)} \frac{\partial(y_{ij}^b)}{\partial(z_{ij}^{LVLH})} + \frac{\partial(\rho_j^{i,R_3})}{\partial(z_{ij}^b)} \frac{\partial(z_{ij}^b)}{\partial(z_{ij}^{LVLH})} \right|_{\chi=\hat{\chi}(k|k-1)} \end{array} \right]$$

$$\mathbf{h}_{j,z}^{i}(k) = \left[\begin{array}{l} \frac{h1_z}{\left\|\boldsymbol{\chi}_i - \boldsymbol{\chi}_j\right\|^{R_1}} \\ \frac{h2_z}{\left\|\boldsymbol{\chi}_i - \boldsymbol{\chi}_j\right\|^{R_2}} \\ \frac{h3_z}{\left\|\boldsymbol{\chi}_i - \boldsymbol{\chi}_j\right\|^{R_3}} \end{array} \right]$$

with

$$
\begin{aligned}
h1_z &= a_{13}(a_{11}x_{ij}^{LVLH} + a_{12}y_{ij}^{LVLH} + a_{13}z_{ij}^{LVLH} - c_{R_1,x}^{IPQ}) + \\
&\quad a_{23}(a_{21}x_{ij}^{LVLH} + a_{22}y_{ij}^{LVLH} + a_{23}z_{ij}^{LVLH} - c_{R_1,y}^{IPQ}) + \\
&\quad a_{33}(a_{31}x_{ij}^{LVLH} + a_{32}y_{ij}^{LVLH} + a_{33}z_{ij}^{LVLH} - c_{R_1,z}^{IPQ}) \\
h2_z &= a_{13}(a_{11}x_{ij}^{LVLH} + a_{12}y_{ij}^{LVLH} + a_{13}z_{ij}^{LVLH} - c_{R_2,x}^{IPQ}) + \\
&\quad a_{23}(a_{21}x_{ij}^{LVLH} + a_{22}y_{ij}^{LVLH} + a_{23}z_{ij}^{LVLH} - c_{R_2,y}^{IPQ}) + \\
&\quad a_{33}(a_{31}x_{ij}^{LVLH} + a_{32}y_{ij}^{LVLH} + a_{33}z_{ij}^{LVLH} - c_{R_2,z}^{IPQ}) \\
h3_z &= a_{13}(a_{11}x_{ij}^{LVLH} + a_{12}y_{ij}^{LVLH} + a_{13}z_{ij}^{LVLH} - c_{R_3,x}^{IPQ}) + \\
&\quad a_{23}(a_{21}x_{ij}^{LVLH} + a_{22}y_{ij}^{LVLH} + a_{23}z_{ij}^{LVLH} - c_{R_3,y}^{IPQ}) + \\
&\quad a_{33}(a_{31}x_{ij}^{LVLH} + a_{32}y_{ij}^{LVLH} + a_{33}z_{ij}^{LVLH} - c_{R_3,z}^{IPQ})
\end{aligned}
$$

Prediction The linearized formation dynamic equations can be presented as the relative motion of the i^{th} spacecraft in an eccentric orbit, *w.r.t.* the orbiting formation reference frame, LVLH, as presented in (6.16)-(6.15a),

$$
\frac{d}{d\theta_t}\begin{bmatrix} x_i \\ y_i \\ z_i \\ x_i' \\ y_i' \\ z_i' \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ f_{41} & 0 & -f_{43} & f_{43} & 0 & 2 \\ 0 & f_{52} & 0 & 0 & f_{43} & 0 \\ f_{43} & 0 & f_{63} & -2 & 0 & f_{43} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ x_i' \\ y_i' \\ z_i' \end{bmatrix} + \xi(\theta_t) \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \iota_{i,x} \\ \iota_{i,y} \\ \iota_{i,z} \end{bmatrix} \tag{6.28}
$$

with $f_{41} = \frac{e_{cc}\cos\theta_t}{1+e_{cc}\cos\theta_t}$, $f_{43} = \frac{2e_{cc}\sin\theta_t}{1+e_{cc}\cos\theta_t}$, $f_{52} = \frac{-1}{1+e_{cc}\cos\theta_t}$, $f_{63} = \frac{3+e_{cc}\cos\theta_t}{1+e_{cc}\cos\theta_t}$, and where ι is the sum of perturbation and control inputs, $\boldsymbol{\iota} = \mathbf{u}+\mathbf{w}$. The perturbations are $\mathbf{w}_{i,(x,y,z)} = \mathbf{w}_{i,J2(x,y,z)}+\mathbf{w}_{i,sun(x,y,z)}+\mathbf{w}_{i,moon(x,y,z)}$. The previous equation corresponds to a linearized θ_t-varying equation in a compact format,

$$
\delta\mathbf{x}'_{trans}(\theta_t) = \mathbf{F}(\hat{\mathbf{x}}_{trans},\theta_t)\delta\mathbf{x}_{trans} + \mathbf{B}(\theta_t)\mathbf{u}(\theta_t) + \mathbf{B}(\theta_t)\mathbf{w}(\mathbf{x}_{trans}(\theta_t)) \tag{6.29}
$$

For the *relative* state representation, and since the matrices $\mathbf{F}$, $\mathbf{B}_u$ and $\mathbf{B}_d$ do not depend on $\boldsymbol{\chi}$, the

dynamic equations become,

$$\frac{d}{d\theta_t}\begin{bmatrix} x_i - x_j \\ y_i - y_j \\ z_i - z_j \\ x_i^{'} - x_j^{'} \\ y_i^{'} - y_j^{'} \\ z_i^{'} - z_j^{'} \end{bmatrix} = \mathbf{F}(\theta_t)\begin{bmatrix} x_i - x_j \\ y_i - y_j \\ z_i - z_j \\ x_i^{'} - x_j^{'} \\ y_i^{'} - y_j^{'} \\ z_i^{'} - z_j^{'} \end{bmatrix} + \mathbf{B}(\theta_t)\begin{bmatrix} \iota_x \\ \iota_y \\ \iota_z \end{bmatrix}_i \tag{6.30}$$

with ,

$$\mathbf{F}(\theta_t) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ f_{41} & 0 & -f_{43} & f_{43} & 0 & 2 \\ 0 & f_{52} & 0 & 0 & f_{43} & 0 \\ f_{43} & 0 & f_{63} & -2 & 0 & f_{43} \end{bmatrix}, \quad \begin{aligned} f_{41} &= \tfrac{e_{cc}\cos\theta_t}{1+e_{cc}\cos\theta_t} \\ f_{43} &= \tfrac{2e_{cc}\sin\theta_t}{1+e_{cc}\cos\theta_t} \\ f_{52} &= \tfrac{-1}{1+e_{cc}\cos\theta_t} \\ f_{63} &= \tfrac{3+e_{cc}\cos\theta_t}{1+e_{cc}\cos\theta_t} \end{aligned}$$

$$\mathbf{B}(\theta_t) = \frac{(1-e_{cc}^2)^3}{(1+e_{cc}\cos\theta_t)^4 n_t^2}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and

$$\begin{bmatrix} \iota_{ij,x} \\ \iota_{ij,y} \\ \iota_{ij,z} \end{bmatrix} = \begin{bmatrix} u_{i,x} - u_{j,x} \\ u_{i,y} - u_{j,y} \\ u_{i,z} - u_{j,z} \end{bmatrix} + \begin{bmatrix} w_{i,x} - w_{j,x} \\ w_{i,y} - w_{j,y} \\ w_{i,z} - w_{j,z} \end{bmatrix}$$

Care must be taken because the perturbations could depend on the state (*e.g.*, $w_{(x,y,z)} = w(\mathbf{x})$).

Rotational state vector

Here the derivation of the equations related to EKF implementation using ST sensor, are performed. The nonlinear observation model is computed using the rotational absolute state vector. The linearization of the process and observation models are performed as well. Also the EKF Fusion filter used to estimate the rotational components of the full state vector, is described.

Throughout this subsection, for the sake of simplifying the notation, we will use $\mathbf{x}$ to denote the

rotational state $\mathbf{x}_{rot}$, dropping all the frames, and to refer to the rotational state vector i^{th} vehicle and to its vehicle i components as, $\mathbf{x}^i = \left[\begin{array}{cc} \left(q^i\right)^T & \left(\boldsymbol{\omega}^i\right)^T \end{array} \right]^T$.

Extended Kalman filter Filtering - star tracker observation model The ST provides directly the quaternion expressing the rotation from body to inertial planet frame (IPQ), with some associated error, so the observation model is, for each s/c i,

$$\mathbf{y}^i(k) = q^i_{IPQ}(k) + \mathbf{v}^i(k) \tag{6.31}$$

where $\mathbf{v}(k)$ is the measurement noise for each time k.

The generalized observation model for s/c i is,

$$\mathbf{y}^i(k) = \mathbf{H}^i \mathbf{x}^i_{rot'}(k) + \mathbf{v}^i(k)$$

where $\mathbf{x}^i_{rot'}(k)$ is the vector of s/c i state components from $\mathbf{x}_{rot'}(k)$ defined in Eq. (6.10). For the sake of simplifying the notation, the frames were dropped and the superscript i refers to vehicle i and not to global frame as stated in Eq. (6.10). Also, and $\mathbf{v}^i(k)$ is the sensor noise for s/c i ST, considered zero mean Gaussian, and characterized by $E\{\mathbf{v}^i(k)\left(\mathbf{v}^i(k)\right)^T\} = \mathbf{R}^i(k)$. Since the ST directly provides measurements of the quaternion, $\mathbf{y}^i(k) = q^i_{ST}(k) = q^{b_i}_{IPQ}(k)$, the observation matrices for s/c $i = 1$, 2 and 3, are as follows

$$\mathbf{H}^1 = \left[\begin{array}{cccccc} \mathbf{1}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{array} \right] \tag{6.32}$$

$$\mathbf{H}^2 = \left[\begin{array}{cccccc} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{1}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{array} \right] \tag{6.33}$$

$$\mathbf{H}^3 = \left[\begin{array}{cccccc} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{1}_{3\times3} & \mathbf{0}_{3\times3} \end{array} \right] \tag{6.34}$$

Error innovation The error innovation in the filtering equations of the Kalman filter is usually computed as the difference between the actual measurements and those predicted by the observation model. However, because quaternions must respect the rotation operation, according to Eqs. (4.10)-(4.11), one must compute the innovation differently for the quaternion components of the state. Denoting the actual ST measurement by $q^i_{ST}(k)$ and its prediction by $\hat{q}^i(k|k-1)$, the innovation or measurement residual becomes,

$$\left[\begin{array}{c} \delta\hat{\mathbf{q}}^i(k) \\ 1 \end{array} \right] = q^i_{ST}(k) \otimes \left(\hat{q}^i\left(k|k-1\right)\right)^{-1}$$

where $\hat{q}^{-1} = \begin{bmatrix} -\hat{\mathbf{q}} & \hat{q}_4 \end{bmatrix}^T$, or after some algebra manipulation,

$$\delta\hat{q}^i(k) = \mathbf{\Xi}^T \left(\hat{q}^i\left(k|k-1\right)\right) q^i_{ST}(k)$$

The complete state vector is split in two parts, corresponding to the angular velocity and the quaternion, *e.g.*, $\mathbf{x}^i(k|k) = \begin{bmatrix} q^i\left(k|k\right) \\ \boldsymbol{\omega}^i\left(k|k\right) \end{bmatrix}$ s/c i components. Therefore, the innovation multiplied by the Kalman gain can be expressed as

$$\begin{bmatrix} \delta\hat{\mathbf{x}}^i_{\mathbf{q}}(k) \\ \delta\hat{\mathbf{x}}^i_{\boldsymbol{\omega}}(k) \end{bmatrix} = \mathbf{K}^i(k)\delta\hat{\mathbf{q}}^i(k)$$

Hence, the local state estimate update is performed according to,

$$\hat{\mathbf{x}}^i(k|k) = \begin{bmatrix} \hat{q}^i\left(k|k\right) \\ \hat{\boldsymbol{\omega}}^i\left(k|k\right) \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \delta\hat{\mathbf{x}}^i_q(k) \\ \sqrt{1 - \left\|\delta\hat{\mathbf{x}}^i_q(k)\right\|} \end{bmatrix} \otimes \hat{q}^i\left(k|k-1\right) \\ \hat{\boldsymbol{\omega}}^i\left(k|k-1\right) + \delta\hat{\mathbf{x}}^i_{\boldsymbol{\omega}}(k) \end{bmatrix}$$

Prediction In order to linearize the nonlinear equations of motion, the state perturbation defined in (B.9) is used, yet it has to be split into two parts, due to the presence of quaternions in state vector components,

$$\begin{aligned} \delta q &= q \otimes \hat{q}^{-1} \\ \delta\boldsymbol{\omega}(k) &= \boldsymbol{\omega}(k) - \hat{\boldsymbol{\omega}}(k) \end{aligned}$$

as already defined in (B.9) and (4.55).

The linearized perturbation state model can be used by the EKF according to,

$$\boldsymbol{\delta}\mathbf{x}\left(k+1\right) = \mathbf{\Phi}(k)\delta\mathbf{x}\left(k\right) + \mathbf{w}(k)$$

where $\mathbf{\Phi}(k)$ is the transition matrix, computed as follow,

$$\mathbf{\Phi}(k) \approx \mathbf{1} + \mathbf{F}(\hat{\mathbf{x}}(t), t)T_s + \frac{(\mathbf{F}(\hat{\mathbf{x}}(t), t)T_s)^2}{2!}$$

and $\mathbf{F}(\hat{\mathbf{x}}(t), t)$ is the Jacobian of $f(.)$ (4.39).

Kinematic equation - linearization Dropping the frame indices for simplicity of notation, the angular velocity of the body frame written in the IPQ frame $\boldsymbol{\omega}_{b_i}^{IPQ}$, is denoted as $\boldsymbol{\omega}$,

$$\boldsymbol{\omega} = \hat{\boldsymbol{\omega}} + \delta\boldsymbol{\omega} \tag{6.35}$$

and from the application of the chain rule follows,

$$\frac{d}{dt}q(t) = \frac{d}{dt}\left(\delta q \otimes \hat{q}\right) = \delta\dot{q} \otimes \hat{q} + \delta q \otimes \dot{\hat{q}}$$

Using the previous expression in the kinematic equation, $\delta\dot{q}\otimes\hat{q}+\delta q\otimes\dot{\hat{q}} = \frac{1}{2}\omega\otimes q$ and right multiplying the previous expression by $\hat{q}^{-1}$, it follows immediately,

$$\delta\dot{q} = \frac{1}{2}\omega \otimes q \otimes \hat{q}^{-1} - \delta q \otimes \dot{\hat{q}} \otimes \hat{q}^{-1}$$

From the kinematics,

$$\begin{aligned} \frac{d}{dt}\hat{q} &= \frac{1}{2}\hat{\omega} \otimes \hat{q} \\ \dot{\hat{q}} \otimes \hat{q}^{-1} &= \frac{1}{2}\hat{\omega} \end{aligned}$$

and using again the chain rule, the expression becomes,

$$\delta\dot{q} = \frac{1}{2}\omega \otimes \delta q - \frac{1}{2}\delta q \otimes \hat{\omega}$$

Expressing the angular velocity as a perturbation around the estimate, as in (6.35),

$$\delta\dot{q} = \frac{1}{2}\hat{\omega} \otimes \delta q + \frac{1}{2}\delta\omega \otimes \delta q - \frac{1}{2}\delta q \otimes \hat{\omega}$$

Representing the quaternion multiplication with the help of the left and right operators $\boldsymbol{\Lambda}(),\ \boldsymbol{\Xi}()$, defined in (4.11) and (4.10), respectively

$$\begin{aligned} 2\delta\dot{q} &= \boldsymbol{\Lambda}\left(\hat{\omega}\right)\delta q + \boldsymbol{\Lambda}\left(\delta\omega\right)\delta q - \boldsymbol{\Xi}\left(\hat{\omega}\right)\delta q \\ &= \begin{bmatrix} -\left[\hat{\boldsymbol{\omega}}\times\right] & \hat{\boldsymbol{\omega}} \\ -\hat{\boldsymbol{\omega}}^T & 0 \end{bmatrix}\delta q + \begin{bmatrix} -\left[\delta\boldsymbol{\omega}\times\right] & \delta\boldsymbol{\omega} \\ -\delta\boldsymbol{\omega}^T & 0 \end{bmatrix}\delta q - \begin{bmatrix} -\left[\hat{\boldsymbol{\omega}}\times\right] & \hat{\boldsymbol{\omega}} \\ -\hat{\boldsymbol{\omega}}^T & 0 \end{bmatrix}\delta q \\ &= \begin{bmatrix} -2\left[\hat{\boldsymbol{\omega}}\times\right] - \left[\delta\boldsymbol{\omega}\times\right] & \delta\boldsymbol{\omega} \\ -\delta\boldsymbol{\omega}^T & 0 \end{bmatrix}\delta q \end{aligned}$$

Applying small quaternion perturbation from (B.9), the previous equation becomes,

$$2\delta\dot{q} = \begin{bmatrix} (-2\left[\hat{\boldsymbol{\omega}}\times\right] - \left[\delta\boldsymbol{\omega}\times\right])\delta\mathbf{q} + \delta\boldsymbol{\omega} \\ -\delta\boldsymbol{\omega}^T\delta\mathbf{q} \end{bmatrix}$$

and neglecting second order terms, $-\left[\delta\boldsymbol{\omega}\times\right]\delta\mathbf{q}$ and $-\delta\boldsymbol{\omega}^T\delta\mathbf{q}$, the resulting linearization of the kinematic equation is,

$$\begin{aligned} \delta\dot{\mathbf{q}}_{IPQ}^{b_i} &= -\left[\hat{\boldsymbol{\omega}}_{b_i}^{IPQ}\times\right]\delta\mathbf{q}_{IPQ}^{b_i} + \frac{1}{2}\delta\boldsymbol{\omega}_{b_i}^{IPQ} \\ \delta\left(\dot{q}_4\right)_{IPQ}^{b_i} &= 0 \end{aligned} \tag{6.36}$$

Dynamic equation - linearization The dynamic equations, derived from Euler's second law, assuming a rigid body with all products of inertia zero and a angular momentum parallel to the body angular velocity, reduces to,

$$\dot{\boldsymbol{\omega}} = \mathbf{I}^{-1}\left(\sum \mathbf{m}_{ext} + \boldsymbol{\omega}\times\mathbf{I}\boldsymbol{\omega}\right) \tag{6.37}$$

where $\mathbf{I}$ is the inertial matrix, and $\mathbf{M}_{ext}$ is the external moments vector, when linearized becomes,

$$\delta\dot{\boldsymbol{\omega}} = \mathbf{I}^{-1}\left(\delta\sum \mathbf{m}_{ext} + \left(\left[\mathbf{I}\hat{\boldsymbol{\omega}}\times\right] - \left[\hat{\boldsymbol{\omega}}\times\right]\mathbf{I}\right)\delta\boldsymbol{\omega}\right) \tag{6.38}$$

thus,

$$\left[\mathbf{I}\hat{\boldsymbol{\omega}}_{b_i}^{IPQ}\times\right] - \left[\hat{\boldsymbol{\omega}}_{b_i}^{IPQ}\times\right]\mathbf{I} = \begin{bmatrix} 0 & (I_y - I_z)\,\hat{\omega}_z & (I_y - I_z)\,\hat{\omega}_z \\ (I_z - I_x)\,\hat{\omega}_z & 0 & (I_z - I_x)\,\hat{\omega}_y \\ (I_x - I_y)\,\hat{\omega}_y & (I_x - I_y)\,\hat{\omega}_x & 0 \end{bmatrix} \tag{6.39}$$

The external moments are neglected and for the open loop case no control is applied to the s/c, resulting in $\sum \mathbf{m}_{ext} = 0$.

The linearization of the **dynamic and kinematic equations** of motion, (6.39) and (6.36) respectively, results into the following matrix for s/c i,

$$\mathbf{F}(\hat{\mathbf{x}}^i(t), t) = \begin{bmatrix} \mathbf{F}_1 & \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} \\ \mathbf{0}_{6\times6} & \mathbf{F}_2 & \mathbf{0}_{6\times6} \\ \mathbf{0}_{6\times6} & \mathbf{0}_{6\times6} & \mathbf{F}_3 \end{bmatrix} \tag{6.40}$$

with

$$\mathbf{F}_1 = \begin{bmatrix} \frac{1}{2}\mathbf{1}_{3\times3} & -\left[\hat{\boldsymbol{\omega}}\times\right] \\ \mathbf{I}^{-1}\left(\left[\mathbf{I}\hat{\boldsymbol{\omega}}\right] - \left[\hat{\boldsymbol{\omega}}\times\right]\mathbf{I}\right) & \mathbf{0}_{3\times3} \end{bmatrix}$$

$$\mathbf{F}_2 = \begin{bmatrix} \frac{1}{2}\mathbf{1}_{3\times3} & -[\hat{\boldsymbol{\omega}}\times] \\ \mathbf{I}^{-1}\left([\mathbf{I}\hat{\boldsymbol{\omega}}] - [\hat{\boldsymbol{\omega}}\times]\mathbf{I}\right) & \mathbf{0}_{3\times3} \end{bmatrix}$$

$$\mathbf{F}_3 = \begin{bmatrix} \frac{1}{2}\mathbf{1}_{3\times3} & -[\hat{\boldsymbol{\omega}}\times] \\ \mathbf{I}^{-1}\left([\mathbf{I}\hat{\boldsymbol{\omega}}] - [\hat{\boldsymbol{\omega}}\times]\mathbf{I}\right) & \mathbf{0}_{3\times3} \end{bmatrix}$$

Extended Kalman filter fusion - rotational state vector

Since the state vector includes quaternion components, the Covariance Intersection algorithm is not applicable anymore, as the weighted sum of both estimates is meaningless for quaternions. Furthermore, because we use the absolute attitude, no correlations in measurements occur, as opposed to the translational state estimation case. Therefore, the alternative is to fuse the state estimates from each s/c using an EKF.

Considering the generalized observation model,

$$\mathbf{y}^i(k) = \mathbf{H}^i\mathbf{x}_{rot'}(k|k-1) + \mathbf{v}^i(k) \tag{6.41}$$

where $\mathbf{x}_{rot'}(k)$ is defined in Eq. (6.10). The "measurements" in the observation equation are now the state estimates from the predecessor s/c, $\mathbf{z}^i(k) = \mathbf{x}_{rot}^{i-1}(k|k-1) + \mathbf{v}^{i-1}(k)$, and the observation noise is the error covariance matrix from the predecessor vehicle, $i-1$, without the components which are locally by the ST sensor in s/c i. Thus, the observation matrix is for s/c $i = 1$, 2 and 3, as follows

$$\mathbf{H}^1 = \mathbf{H}^2 = \mathbf{H}^2 = \mathbf{1}_{18\times18} \tag{6.42}$$

and the covariance matrix of the measurement is now the error covariance matrix from the predecessor state estimates, $E\left\{\mathbf{v}^i(k)\left(\mathbf{v}^i(k)\right)^T\right\} = \mathbf{P}^{i-1}(k|k-1)$, which must also be communicated.

6.3.7 Gramian observability

Translation state vector

From the nonlinear equations for a formation of N-s/c, described in detail in subsection 6.3.4, we have the Jacobian matrix of $f(\mathbf{x}_{trans}, t)$, and (6.27) for the Jacobian matrix of $h(\mathbf{x}_{trans}, k)$. Computing the observation matrix (5.16), its rank is 12, for each s/c. This means that there are 6 state vector components that are non-observable on each s/c, as previously verified in subsection 4.3.3, *i.e.,* there is not enough information in the RF sensor measurements and the state vector cannot be determined distinctly at every time k. Thus, as discussed previously, more information must be communicated, which is performed via the *information flow* graph, resulting into all state vector components observability. Thus,

when the fusion with other s/c estimates occurs (using the CI algorithm) the rank of the observability matrix is full, *i.e.,* 18. Results will be presented in Chapter 7.

Rotational state vector

Applying the observability concept to the rotational part of the state vector, where the Jacobian matrix of $f(\mathbf{x}_{trans}, t)$, given by (6.40), and the linear observation matrix is given by (6.41), we can deduce that the rank is 6. This means that there are only 6 columns of the observability matrix that are linearly independent. This in fact correspond to the quaternions and angular velocity components from the considered s/c, which are updated due to the quaternion measurements coming from their ST sensor. The rest of the state vector variables are non-observable, meaning that there is not enough information in the sensor measurements and the state vector cannot be determined distinctly at every time k. When the fusion with other s/c estimates occurs (using the EKF fusion algorithm) the rank of the observability matrix is full, *i.e.,* 18. This is verified in the results obtain in Chapter 7.

6.3.8 Filtering and prediction

The filtering and prediction blocks of the algorithm run every sampling time, $T_s = 10[s]$ for FAC mode and $T_s = 1[s]$ for BCM mode. At instants $t = 2k.T_s\,[s]$ with $k = 0, 1, 2, ...$ the estimators use the s/c own RF measurements (translational algorithm) through the links represented by the *measurement* graph and the s/c own ST measurements (rotational algorithm), to obtain by filtering a state estimate. Also, at the same instant, the translational algorithm is using the attitude estimates computed by the rotational algorithm. At instants $t = (2k + 1).T_s\,[s]$ with $k = 0, 1, 2, ...$ the estimators communicate to the successor s/c through the links represented by the *information flow* graph (see Figure 4-9) its predicted state estimate. The successor s/c uses the communicated estimate from the predecessor s/c in the *information flow* graph to obtain (by filtering + prediction) a state estimate. Every T_s, the state estimated by filtering is sent to the s/c controller, as shown in Fig. 6-12.

6.3.9 Modularity

For the sake of modularity, the MATLAB/Simulink Navigation blocks, depicted in Fig. 6-15, representing the estimator algorithms are implemented such that one can copy and paste it from one s/c to another. To ensure this, the observation matrix of all s/c in the formation should be available for calculation inside the filter, as linearized observation equations lead to structurally different matrices from s/c to s/c. Then a mask should be used to select which matrix is to be used, depending on the s/c number and which s/c is connected to it. However, this modularity is limited since if the number of formation

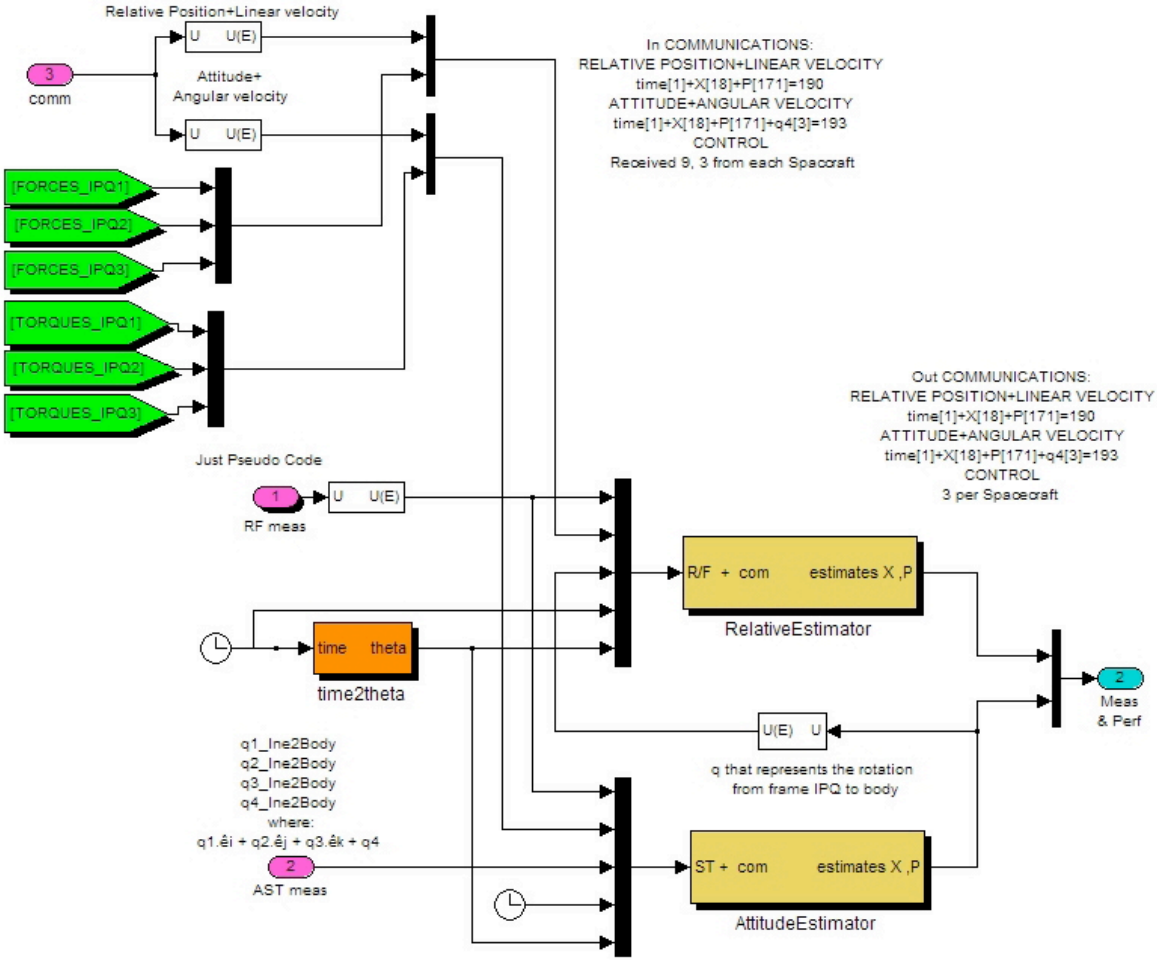

Figure 6-15: Translational and rotational block diagrams inside each Navigation block in the simulator.

s/c increases or decreases, the state vector should be extended as the rest of the variables: matrices, measurements and all the mathematical elements involved in the calculations.

6.4 Tests Description

The most interesting and ambitious goal of the GNC methods is to have the satellites pointing to the inertial direction during an interval starting several degrees before arriving to apogee and ending the same time after apogee passage and with tight accuracy, as well as to achieve and maintain a high accuracy during the science experiment. Therefore, the two segments of the GTO trajectory chosen to test the navigation algorithms are the FAC and BCM. The FAC mode represents a demanding goal of controlling the formation from an initial disperse configuration, a sphere of $8\,[km]$ diameter into a

triangle formation, where each 250 metre apart from each other. The required final accuracy is of $0.1[m]$ for relative position, $0.01[m/s]$ for velocity and $1[deg]$ for inertial attitude and $0.1[deg/s]$ for angular velocity as described in Fig. 6-16 or can also be visualized in GTO in Fig. 6-17. The trajectory is centered around apogee, starting at $\theta_t(0) = 156,27\,[rad]$ ($t(0) = 10716.94[s]$) and ending 6 hours later, as visualized in Table 6.11.

The BCM mode has a duration of 2 hours, centered in apogee and the required accuracy is to keep the formation, with accuracy of $0.01[m]$ for relative position, $0.001[m/s]$ for velocity, $1[arcsec]$ for inertial attitude, $0.1[arcsec/s]$ for angular velocity is, as described in more detail in Table 6.12. In addition to this, the most important results are also presented in BCM mode in closed-loop with the navigation algorithms out of the GC loop and in the GC loop. This mode is even more demanding, in terms of estimation accuracy then the FAC mode as described in Table 6.12, but the formation in the BCM mode, contrary to the FAC mode, is already starting in an initial tight configuration.

Formation Concepts	**Formation Acquisition Mode (FAC)**
Description	-Bring the formation from initial conditions to final conditions which according to location in the orbit is the LFM or ICM mode
Initial conditions	-position: s/c's are dispersed within a sphere of $8[km]$, centered at the reference orbit -velocity: each s/c has the reference orbital velocity with an error of $+/-0.1[m/s]$ in a random direction
Final conditions	-position: relative distance $< 500[m]$ for each s/c -attitude: attitude error *w.r.t.* nominal pointing $< 5[^o]$ for the formation
GNC Objective	-acquire a formation configuration corresponding to LFM -avoid collisions
Guidance	-initial random disposition in sphere of $8[km]$
Attitude Guidance	-inertial anti-sun
Navigation	-RF sensor
Control requirements	-relative position: $0.1[m]$, $0.01[m/s]$ -inertial attitude: $1[^o]$, $0.1[^o/s]$
·**Actuators**	-$[mN]$ thrusters
·**Sensors**	-RF sensor -Star Tracker

Table 6.11: Formation Acquisition Mode (FAC) specifications.

Details of the set of experiments used to test and analyse the proposed navigation algorithm, which will be presented in the next chapter, follow. A set of simulation tests were set up to evaluate the position estimation performance under different scenarios.

- the FAC mode, which lasts 6 hours orbit run, is performed always in closed-loop in order to avoid

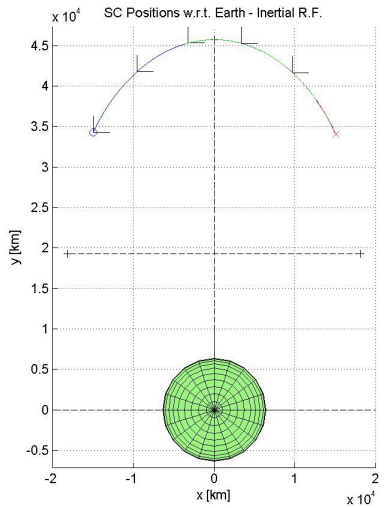

Figure 6-16: View from above the orbital plane of the 3 spacecraft absolute positions *w.r.t.* Earth, in IPQ. Reprinted from [68].

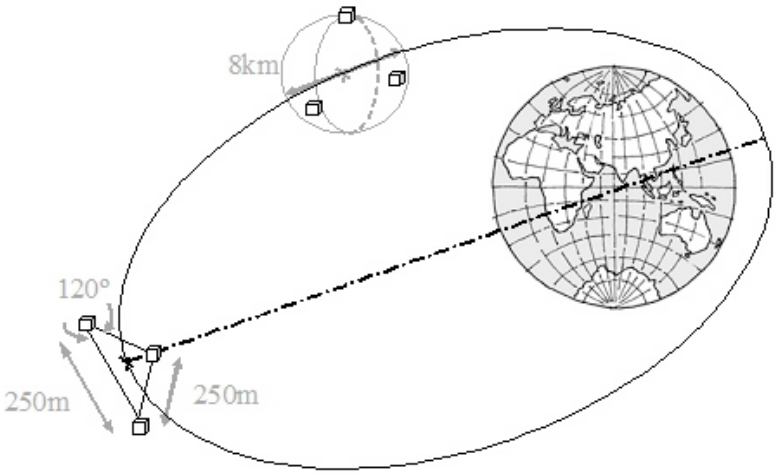

Figure 6-17: Representation of the Formation Acquisition mode (FAC) in the GTO orbit. The initial conditions coincide with the deployment of the s/c, so the s/c can be within a sphere of 8km diameter and the final conditions corresponds to building the formation to perform the science experiment, i.e., the initial conditions of the ICM where the s/c should be 25 meter away from each other.

Formation Concepts	**Baseline Control Mode (BCM)**
Description	-Bring the formation to a configuration in attitude and position just before the final control loops required for interferometry, and and maintain it during what would be the experiment time.
Initial conditions	-inertial pointing tolerance of 60[*arcsec*]
Final conditions	-*position*: accuracy $< 10[cm]$ (*i.e.*, improve performance for RF alone) -*absolute attitude* pointing accuracy $< 60[arcsec]$ (*i.e.*, improve ST accuracy) -*relative attitude* should be higher than for absolute accuracy
GNC Objective	-to achieve coarse inertial attitude pointing -to achieve coarse lateral positioning
Guidance	-keep the formation triangle configuration
Attitude Guidance	-inertial attitude
Navigation	-RF sensor for ranging -coarse lateral
Control requirements	-*longitudinal relative position*: $0.01[m]$, $0.1[cm/s]$ -*lateral relative position*: $0.01[m]$, $0.1[cm/s]$ -*inertial attitude*: $1[arcsec]$, $0.1[arcsec/s]$
·**Actuators**	-$[mN]$ thrusters
·**Sensors**	-RF sensor -Star Tracker -Gyroscopes for attitute control

Table 6.12: Baseline Control Mode (BCM) specifications.

the formation to drift apart, as concluded from the two orbits run, and explained in Section 7.2.4. These closed-loop tests are divided in two groups: the GC experiment in closed-loop but without the state estimator algorithms included in the loop and the GC experiment with the estimators included in the closed-loop. In former case, the GC are in closed-loop in order to have the estimators results independent of the RF distance dependency. Thus, the estimators attempt to follow the real relative position and velocity and do not feed the control with estimates of the full formation state estimation. In latter case, the estimators are included in the closed-loop, feeding the GC part. The test performed is meant to show the navigation algorithms performance when they are put together in the closed-loop. Also some preliminary tests are performed using perturbations, in order to demonstrate how does the navigation algorithm perform in a demanding and closer to reality scenario. The matrices $\mathbf{R}$ and $\mathbf{Q}$ are determined in order to fulfill the NEES and NIS consistency tests. The PCRLB is also computed for this analysis in order to show the optimality of the resulting navigation algorithms.

- The BCM mode, which lasts 2 hour orbit run, is performed always in open-loop for the estimator.

Since the goal of this mode is to control the absolute attitude, there is only guidance and no control for the translational motion. Again the matrices **R** and **Q** are determined for this segment of the GTO, in order to fulfill the NEES and NIS consistency tests.

- Monte-Carlo simulations, $M = 10$ runs, are performed for the BCM mode of the GTO, since this is part of the orbit corresponds to the science-experiment and thus it is of higher interest to analyse regarding parameters uncertainty. Again the matrices **R** and **Q** are determined for the Monte-Carlo runs, in order to fulfill the NEES and NIS tests limits.

6.4.1 Guidance and control

The GC algorithm used in FAC, was developed by Dumitriu *et al.* [8], and is based on the Pontryagin Maximum Principle formulation of a model-based optimal trajectory planning problem with collision avoidance. Its model-based nature has the advantage of providing not only the optimal trajectories, but also the corresponding optimal thrusts.

The implemented GC algorithm computes a quasi optimal (which does neither consider collision avoidance nor the nonlinear perturbation terms), simple and robust-to-perturbations algebraic solution. The control inputs limitations and collision avoidance are considered a posteriori, so as to avoid using any time consuming iterative technique. Also, perturbations are not considered in the dynamics model used by the algorithm. Perturbations must be linearized first in order to be included.

The trajectory planning algorithm is regularly computed at given time instants. After each of those time instants the obtained optimal inputs are applied at a faster rate than the trajectory planning computation, until the next time instant where the algorithm is re-computed, thus updating again the control inputs and the optimal trajectory.

During FAC, the goal of the GC is to bring the 3-s/c:

- from an *initial* random disposition within a sphere of 8 $[km]$ in diameter centered in the dispenser,
- to a suitable *final* disposition, so that the formation should afterwards reach the configuration desired at the beginning of BCM, or end of first ICM, mostly by means of natural gravity motion, with as little control inputs as possible during this first ICM, in order to save on propellant. This configuration will be an equilateral triangle where each s/c is 250 metre apart from each other, around apogee such that the observation time will last for 5500 seconds centred on the apogee (see Fig 6-18). The formation plane must also be rotated to the desired plane for the science measurements. Moreover, the goal of control, besides obtaining an accurate final formation configuration, is to avoid collision among the spacecraft during the lifetime of the mission and

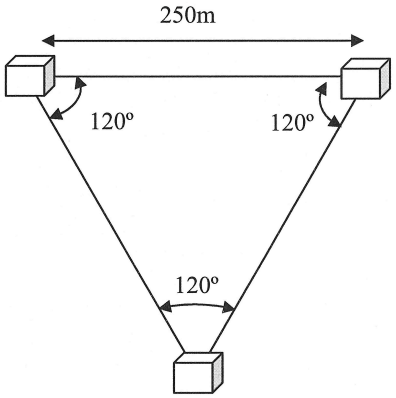

Figure 6-18: Formation Flying geometry to accomplish Mission Science.

minimize propellant consumption among the elements of the formation. This must be performed by minimizing the propellant consumption of all spacecraft and by avoiding collisions.

For this purpose, Dumitriu *et al.* [8] [71], proposed an algebraic algorithm that minimizes the propellant consumption and ensuring collision avoidance. The Pontryagin's maximum principle for Guidance, is used to generate the control inputs required to obtain the optimal trajectory at spaced time instants along the orbit. Simultaneously, the control block applies the optimal inputs until the next regularly spaced time instant, updated again by the Guidance block. The GC algorithms were implemented as C MEX-file S-functions from specifications written in Ansi-C compatible source code.

The GC algorithm used in BCM, was developed by DEIMOS SA [9], and it uses an optimized low thrust profile, and relative position plus attitude PID control. For the translational motion there is only guidance and no control.

6.4.2 Experiments description

A set of 6 experiments were set up to test the estimator plus 1 experiment to illustrate the GC behaviour in FAC mode [72], and another for the GC behaviour in BCM mode, and 1 experiment for GNC. They are listed in Table 6.13 and described in the sequel.

On **Experiment 1**, we present the estimator results for FAC mode with the navigation algorithms not included in the loop with the GC algorithms. The s/c state estimates are initialized subtracting $1[m]$ from the true relative position components, $1[m/rad]$ from the true relative velocity component, $10[arcsec/s]$ from the true angular velocity and $10[^o]$ from the Euler angles, for the formation full state vector, with sampling time $T_s = 10[s]$. The measurement and process noise covariance matrices were not

Exp.	State estimate initial conditions	Local Filter	Filter Fusion	Mode
1	$\hat{x}_{trans} = a)$ $\hat{x}_{rot} = b)$	EKFc) EKF	- -	FAC
2	$\hat{x}_{trans} = a)$ $\hat{x}_{rot} = b)$	EKF EKF	CI EKF fusion	FAC
3	$\hat{x}_{trans} = a)$ $\hat{x}_{rot} = b)$	EKF EKF	- -	FAC
4 d)	$\hat{x}_{trans} = a)$ $\hat{x}_{rot} = b)$	EKF EKF	CI EKF fusion	FAC
5	GC no Navigation	-	-	FAC
6	GN (= exp. 4) C	-	-	FAC
7	GC, no Navigation	-	-	BCM
8 d)	$\hat{x}_{trans} = e)$ $\hat{x}_{rot} = f)$	EKF EKF	CI -	BCM
9	G, N observer (= exp. 8) C	-	-	BCM

Table 6.13: Experiments setup with *a)* $\begin{cases} \hat{\chi}(\theta_0) = \chi(\theta_0) - 1[m] \\ \hat{\chi}'(\theta_0) = \chi'(\theta_0) - 1[m/rad] \end{cases}$ *b)* $\begin{cases} \hat{\psi}, \hat{\theta}, \hat{\phi}(t_0) = \psi, \theta, \phi(t_0) - 10[^o] \\ \omega(t_0) = \hat{\omega}(t_0) - 10[\text{arcsec}/s] \end{cases}$ *c)* with nominal measurement noise covariance matrix *d)* with perturbations included in the simulator *e)* $\begin{cases} \hat{\chi}(\theta_0) = \chi(\theta_0) \\ \hat{\chi}'(\theta_0) = \chi'(\theta_0) \end{cases}$ *f)* $\begin{cases} \hat{\psi}, \hat{\theta}, \hat{\phi}(t_0) = \psi, \theta, \phi(t_0) \\ \omega(t_0) = \hat{\omega}(t_0) \end{cases}$. Experiments 1-4 and 8, 9 are with the estimator as observer, the navigation estimates are not fedback into the loop. Experiments 6 is closed-loop with navigation included, GNC with the navigation algorithm, where the true values of the variables are fed to the GC algorithms.

tuned, but instead the nominal matrices are used and correspond to those given directly by simulator for the process noise and for the measurement noise given directly by the physical characteristics of the sensors (presented in subsections 7.2.2 and 7.2.3 respectively). RF sensor for the translational relative state vector and ST sensor for the rotational absolute state vector.

At times $t = 2k.T_s[s]$, with $k = 0, 1, 2, ...$, each s/c computes the state vector relative position and translational relative velocity estimates based on its local RF measurements and the inertial attitude and inertial angular velocity estimates based on its local ST measurements, both using EKF. The non-observable components of the state vector are propagated through the dynamic equations of motion. The translational and rotational algorithms are working independently, *i.e.,* the state vector estimates obtained in the rotational absolute algorithm are not being used by the translational algorithm, since for now, we want to study the algorithms characteristics without influence on each other.

On **Experiment 2**, we present the results of the estimator with the same conditions as in experiment 1, except that the measurement and process noise covariance matrices were tuned for a 6-hour orbit in order for the filter to be as consistent as possible in the FAC mode of the orbit, according to the NEES and NIS tests, as described in detail in subsections 7.3.2 and 7.3.3. As in previous experiment, in s/c 1

there are no measurements to update the relative position and velocity between s/c 2 and 3 and thus these components are just propagated through the dynamics if only a local EKF are used. As a result, not only its initial estimation error is propagated but it also accumulates over time the error of the dynamic equations linearization.

On **Experiment 3**, we present the results of the estimator for FAC mode with the navigation algorithms not included in the loop with the GC algorithms. The state vector is initialized by the same values of the experiment 2, but at times $t = (2k + 1).T_s[s]$, with $k = 0, 1, 2,...$ the predecessor state vector acts as an observation in the filtering part, using the CI algorithm for the relative position and translational relative velocity, while an EKF, properly modified and thus called EKF fusion, is used to combine the predecessor state vector attitude and angular velocity with the local attitude and angular velocity estimates. The measurement noise covariance matrix and process noise covariance matrix used, were those tuned for 6 orbit example, obtained from previous experiment. In contrast with experiment 2, using CI for the relatives states and EKF fusion for the attitude and angular velocity, enables the filter to update the components of the state vector estimates for the relative position between s/c 2 and 3, with improved estimates made by predecessor s/c since, at some point in the communications flow, the actual distance measurement is used, *e.g.*, when s/c 3 measures its distance to s/c 2. In the implemented CI algorithm, the weight parameter w is computed at each step based on the performance criterion of minimizing the trace of the resulting error covariance matrix. In this situation, the translational and rotational algorithms are working together, *i.e.,* the state vector estimates obtained in the rotational absolute algorithm are used by the translational algorithm.

On **Experiment 4**, we present the results of the estimator with the same conditions as in experiment 3, except that the following perturbations were added to the simulator,

- Solar Radiation Pressure
- High order gravitational anomalies
- Sun and Moon third body perturbations
- J2 perturbations
- Micrometeorids

On **Experiment 5**, in this test, we present results from tests performed, in the FF-FES simulator for FAC mode, with the GC algorithms according to [71], with the closed-loop GC algorithm but without navigation. This experiment shows the performance level attained by the GC algorithm when the estimator is not in the loop, *i.e.,* with access to full and noiseless state information. The final conditions for the FAC mode are the distances of 208 metre for s/c 2 and 216 metre for s/c 3 *w.r.t.* to s/c 1,

provided by DEIMOS SA, and computed so that the natural motion of the FF s/c during the LFM is stable and will lead to a FF s/c relative distance of 250 metre apart from each other in the beginning of BCM. Thus, the goal of the GC algorithm is to reach the distances of 208 metre for s/c 2 and 216 metre for s/c 3 *w.r.t.* to s/c 1, when the FAC mode ends.

On **Experiment 6**, we present the results of the estimator for FAC mode with the navigation algorithms included in the loop with the GC algorithms, GNC. The navigation algorithms are according to experiment 4.

On **Experiment 7**, we present the results of the GC for BCM mode with the navigation algorithms turned off. The sampling time is $T_s = 1\,[s]$. The target is to control the absolute attitude for each s/c, thus the attitude as to be represented in inertial frame. Therefore, a PID control from [9] are used to control just the inertial attitude.

On **Experiment 8,** we present results from the tests performed in the FF-FES simulator for BCM mode, with the GC algorithms from [9], with estimator as observer, its estimates not being feedback into the closed-loop GC algorithm. Notice that since in the translational motion there is only guidance, and no control, the navigation estimates are not feedback into the closed-loop. The estimator at times $t = 2k.T_s[s]$, with $k = 0, 1, 2, ...$, each s/c computes the state vector relative position and translational relative velocity estimates based on its local RF measurements and the inertial attitude and inertial angular velocity estimates based on its local ST measurements, both using EKF. At times $t = (2k + 1).T_s[s]$, with $k = 0, 1, 2,...$ the predecessor state vector acts as an observation in the filtering part, using the CI algorithm for the relative position and translational relative velocity, while for the non-observable rotational components of the state vector are propagated through the dynamic equations of motion, since the translational state vector components just need the observable components of the state vector. As in previous experiment, the sampling time used is $T_s = 1\,[s]$. This experiment shows the performance level attained by the GC algorithm when the estimator is not in the loop, *i.e.,* with access to full and noiseless state information. The initial conditions for the BCM mode are the distances of 250 metre apart from each other. Since the aim of the s/c is science experiment, stars pointing, the goal of the GC algorithms in this mode, is to control the inertial attitude. Thus the Euler angle has the requirement of being less then $5\,[^o]$ for the formation. The translational part of the state vector is not controlled but only have guidance algorithm applied. The FF-FE simulator also have the perturbations turned on. The process and measurement noise matrices are those tuned in the BCM mode, and as described in subsection 7.3.2.

On **Experiment 9**, The BCM trajectory segment is considered, with $M = 10$ Monte-Carlo simulations for the assessment of the performance of the navigation algorithms design *w.r.t.* model parameters uncertainties and nonlinearities not considered during the navigation algorithms design. They are listed

in Table 6.14 and described in the sequel.

Parameters description	**initial value**
Confidence level in which Monte-Carlo results	95 [%]
Percentage of uncertainty (3σ) on the gravitational Sun constant	1 [%]
Percentage of uncertainty (3σ) on the gravitational Earth constant	1 [%]
Uncertainty (3σ) in the initial true anomaly of the GTO	1 [^{o}]
Uncertainty (3σ) in the initial translational state vector	1 [m] ,1 [m/rad]
Uncertainty (3σ) in the initial rotational state vector	1 [$arsec/s$], 10 [^{o}]
Percentage of uncertainty (3σ) on the s/c mass	1 [%]
Percentage of uncertainty (3σ) on the calculated SRP acting on each s/c	1 [%]
Uncertainty (3σ) of the s/c thrusters each s/c	0.01 [m]
Uncertainty (3σ) in the drag coefficient of the s/c	5 [%]

Table 6.14: Description of the Monte-Carlo uncertainty parameters used on the simulator.

The estimator algorithms are according to conditions of experiment 8. The navigation algorithms are out of the closed-loop, however the state vector estimates obtained in the rotational absolute algorithm are used by the translational algorithm.

6.5 Error Analysis

There are several ways to analyse the state estimates coming from the algorithms. The state vector considered is the relative position in meters and the translational relative velocity in meters per radians. The relative velocity is estimated in metre per second due to the θ_t-varying relative dynamic equations, thus the algorithm is developed related to the true anomaly θ_t, instead of time, t. Moreover, since the quaternions have no clear physical meaning, as opposite to the Euler angles, the quaternions are then transformed to Euler angles (6.48), according to the sequence of rotations considered, for this work. However, whenever the standard deviation envelope (2σ, are calculated by taking the square root of the diagonals of the estimated error covariance matrix and multiplying the result by two) for attitude is presented, the quaternions must be shown instead the Euler angles since the translational navigation

algorithm is performed in terms of the quaternions.

6.5.1 Relative position and translational velocity error

The error in the position is calculated, at each sampling time, as follows,

$$\widetilde{\boldsymbol{\chi}}_{12,32,13} = \boldsymbol{\chi}_{12,32,13} - \hat{\boldsymbol{\chi}}_{12,32,13} \tag{6.43}$$

and for the translational relative velocity,

$$\widetilde{\boldsymbol{\chi}}'_{12,32,13} = \boldsymbol{\chi}'_{12,32,13} - \hat{\boldsymbol{\chi}}'_{12,32,13} \tag{6.44}$$

where the subscript 12, 32 and 13 refer to its relative components.

6.5.2 Angular velocity error

The error in the angular velocity is calculated, at each sampling time, as follows,

$$\widetilde{\omega}_{1,2,3} = \omega_{1,2,3} - \hat{\omega}_{1,2,3} \tag{6.45}$$

where the subscript 12, 32 and 13 refer to its relative components.

6.5.3 Quaternion error

From Eq. (4.55), the error in the quaternion is calculated, at each sampling time, as follows,

$$\begin{bmatrix} \widetilde{\mathbf{q}} \\ 1 \end{bmatrix} = q \otimes \hat{q}^{-1} \tag{6.46}$$

where $\hat{q}^{-1} = \begin{bmatrix} -\hat{\mathbf{q}}^T & \hat{q}_4 \end{bmatrix}^T$,

$$\tilde{q} = \boldsymbol{\Xi}\left(\hat{q}^{-1}\right) q$$

or with after some algebra manipulation,

$$\tilde{q} = \boldsymbol{\Xi}^T\left(\hat{q}\right) q$$

6.5.4 Euler angles error

The Euler angles are computed in terms of the elements of the attitude matrix that maps any vector from the IPQ to the body frame, Eq. (4.8), or in terms of the Euler angles depending on the sequence

of rotations used. For the 123 sequence considered,

$$\begin{aligned} A_{123}(\psi,\theta,\phi) &= A_1(\psi)A_2(\theta)A_3(\phi) \\ &= \begin{bmatrix} \cos\theta\cos\phi & \cos\phi\sin\theta\sin\psi+\sin\phi\cos\psi & -\cos\phi\sin\theta\cos\psi+\sin\phi\sin\psi \\ -\cos\theta\sin\phi & -\sin\phi\sin\theta\sin\psi+\cos\phi\cos\psi & \sin\phi\sin\theta\cos\psi+\cos\phi\sin\psi \\ \sin\theta & -\cos\theta\sin\psi & \cos\theta\cos\psi \end{bmatrix} \end{aligned} \tag{6.47}$$

the following relation can be established between quaternions $q = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^T$ and Euler angles, ψ, θ, ϕ:

$$\begin{aligned} \psi &= \arctan 2\left\{\frac{-2\left(q_2q_3-q_1q_4\right)}{-q_1^2-q_2^2+q_3^2+q_4^2}\right\}[rad],\ 0[^o] \le \psi \le 360[^o] \\ &= \arctan2\left(-\frac{A_{32}}{A_{33}}\right) \\ \theta &= \arcsin\left(2\left(q_1q_3+q_2q_4\right)\right)[rad],\ 90[^o] \le \theta \le 90[^o] \\ &= \arcsin\left(A_{31}\right) \\ \phi &= \arctan 2\left\{\frac{-2\left(q_1q_2-q_3q_4\right)}{q_1^2-q_2^2-q_3^2+q_4^2}\right\}[rad],\ 0[^o] \le \phi \le 360[^o] \\ &= \arctan2\left(-\frac{A_{21}}{A_{11}}\right) \end{aligned} \tag{6.48}$$

where the Matlab function atan2(Y,X) is the four quadrant arctangent of the real parts of the elements of X and Y. $-\pi$ <=atan2(Y,X) <= π. Considering other elements in the attitude matrix, different expressions can be obtained relating quaternions and Euler angles. Thus, the error in the Euler angles is calculated, at each sampling time, as follows,

$$\widetilde{\psi},\widetilde{\theta},\widetilde{\phi} = \psi,\theta,\phi - \hat{\psi},\hat{\theta},\hat{\phi} \tag{6.49}$$

6.5.5 Standard deviation

The estimates errors defined before can be bounded by a standard deviation, σ_l which provides realistic error estimates assuming that the measurements are uncorrelated. The standard deviation is calculated from the corresponding diagonal elements of the state error covariance matrix, P,

$$\sigma_{ll} = \sqrt{P_{ll}}$$

Since the diagonal of the error covariance matrix is defined as $P_{ll} = E[\widetilde{x}_l\widetilde{x}_l^T]$ where $\widetilde{x} = x - \hat{x}$ is the error of the corresponding parameter, relative position, relative angular velocity and absolute angular

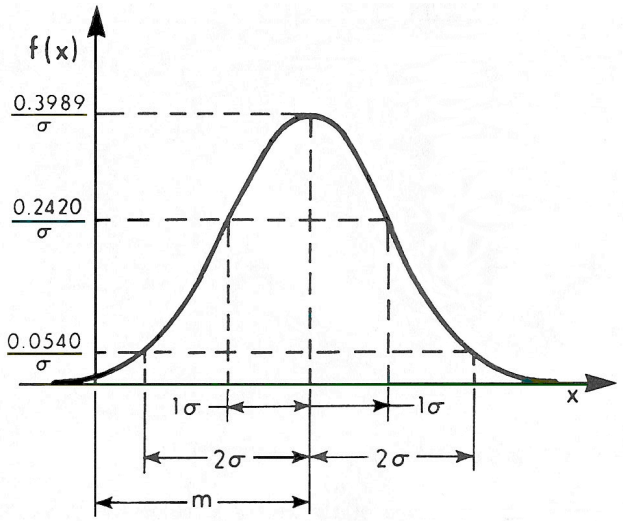

Figure 6-19: **Normal probability density function** defined by the mean, m, and by the standard deviation, σ. The probability of a random value being inside the $\pm 2\sigma$ bounds (σ is the standard deviation) centered around the mean is 95%. Figure reprinted from [56].

velocity. For these parameters,

$$\begin{aligned} P_{\tilde{x}} &= E[(x-\hat{x})(x-\hat{x})^T] \\ &= E[\tilde{x}\tilde{x}^T] \\ \sigma_x &= \sqrt{P_x} \end{aligned} \tag{6.50}$$

while for the quaternion, the error covariance matrix is defined as $P_{ll} = E[\tilde{q}_{ll}^T]$ where $\tilde{q}$ is the error of the quaternion, defined in Eq. (6.46). Thus for the quaternion,

$$P_{q_1,q_2,q_3} = E\left[\begin{array}{cc} \tilde{q}_{11,22,33} & \tilde{q}_{11,22,33}^T \end{array} \right]$$

In the results shown in next chapter, the estimates errors are presented with twice of their standard deviation $\pm 2\sigma$ resulting in an envelope, which corresponds to the estimates probability of 95% of being centered around the mean, as shown in Fig. 6-19.

6.5.6 Root mean square error

The absolute errors are interesting for the analysis of the navigation algorithms performance, as stressed by [57]. Therefore, the expressions for the relative position, translational relative velocity, distance,

Euler angles and absolute angular velocity are given.

The absolute error in the relative **position** is calculated as the root mean square (RMS) error, for a simulation with ℓ time steps,

$$\begin{aligned} RMS\left(\widetilde{x}_{12,32,13}\right) &\simeq \lim_{\ell\to\infty}\sqrt{\frac{1}{\ell}\sum_{t=1}^{\ell}(x_{12,32,13}-\hat{x}_{12,32,13}))^2} \\ RMS\left(\widetilde{y}_{12,32,13}\right) &\simeq \lim_{\ell\to\infty}\sqrt{\frac{1}{\ell}\sum_{t=1}^{\ell}(y_{12,32,13}-\hat{y}_{12,32,13})^2} \\ RMS\left(\widetilde{z}_{12,32,13}\right) &\simeq \lim_{\ell\to\infty}\sqrt{\frac{1}{\ell}\sum_{t=1}^{\ell}(z_{12,32,13}-\hat{z}_{12,32,13})^2} \end{aligned} \tag{6.51}$$

where the subscript 12, 32 and 13 refer to its relative components.

The absolute error in the translational relative **velocity** is calculated as the RMS,

$$\begin{aligned} RMS(\dot{\widetilde{x}}_{12,32,13}) &\simeq \lim_{\ell\to\infty}\sqrt{\frac{1}{\ell}\sum_{t=1}^{\ell}(\dot{x}_{12,32,13}-\dot{\hat{x}}_{12,32,13})^2} \\ RMS(\dot{\widetilde{y}}_{12,32,13}) &\simeq \lim_{\ell\to\infty}\sqrt{\frac{1}{\ell}\sum_{t=1}^{\ell}(\dot{y}_{12,32,13}-\dot{\hat{y}}_{12,32,13})^2} \\ RMS(\dot{\widetilde{z}}_{12,32,13}) &\simeq \lim_{\ell\to\infty}\sqrt{\frac{1}{\ell}\sum_{t=1}^{m}(\dot{z}_{12,32,13}-\dot{\hat{z}}_{12,32,13})^2} \end{aligned}$$

where the subscript 12, 32 and 13 refer to its relative components.

The absolute error in the **distance**, $\widetilde{\rho}_{ij}$, with $ij = 12, 23, 13$, is calculated as the RMS,

$$RMS\left(\widetilde{\rho}_{ij}\right) \simeq \lim_{\ell\to\infty}\sqrt{\frac{1}{\ell}\sum_{t=1}^{\ell}(\rho_{ij}-\hat{\rho}_{ij})^2}$$

where the relative distance (6.8) is estimated as,

$$\hat{\rho}_{ij} = \sqrt{\hat{x}_{ij}^2+\hat{y}_{ij}^2+\hat{z}_{ij}^2}$$

The absolute error in the **Euler angles** is calculated as the RMS,

$$RMS\left(\widetilde{\psi}_{1,2,3}\right) \simeq \lim_{\ell\to\infty}\sqrt{\frac{1}{\ell}\sum_{t=1}^{\ell}(\psi_{1,2,3}-\hat{\psi}_{1,2,3})^2}$$

$$RMS\left(\widetilde{\theta}_{1,2,3}\right) \simeq \lim_{\ell\to\infty}\sqrt{\frac{1}{\ell}\sum_{t=1}^{\ell}(\theta_{1,2,3}-\hat{\theta}_{1,2,3})^2}$$

$$RMS\left(\widetilde{\phi}_{1,2,3}\right) \simeq \lim_{\ell\to\infty}\sqrt{\frac{1}{\ell}\sum_{t=1}^{\ell}(\phi_{1,2,3}-\hat{\phi}_{1,2,3})^2}$$

The absolute error in the absolute **angular velocity** is calculated as the RMS,

$$RMS\left(\widetilde{\omega}_{(1,2,3),x}\right) \simeq \lim_{\ell\to\infty}\sqrt{\frac{1}{\ell}\sum_{t=1}^{\ell}((\omega_x)_{1,2,3}-(\hat{\omega}_x)_{1,2,3})^2}$$

$$RMS\left(\widetilde{\omega}_{(1,2,3),y}\right) \simeq \lim_{\ell\to\infty}\sqrt{\frac{1}{\ell}\sum_{t=1}^{m}((\omega_y)_{1,2,3}-(\hat{\omega}_y)_{1,2,3})^2}$$

$$RMS\left((\widetilde{\omega}_z)_{1,2,3}\right) \simeq \lim_{\ell\to\infty}\sqrt{\frac{1}{\ell}\sum_{t=1}^{\ell}((\omega_z)_{1,2,3}-(\hat{\omega}_z)_{1,2,3})^2}$$

6.5.7 Monte-Carlo

The RMS from M Monte-Carlo runs for relative **position** is,

$$RMS\left(\widetilde{x}_{12,32,13}\right) \simeq \lim_{M}\sqrt{\frac{1}{M}\sum_{t=1}^{M}(\tilde{x}_{12,32,13})^2}$$

$$RMS\left(\widetilde{y}_{12,32,13}\right) \simeq \lim_{M\to\infty}\sqrt{\frac{1}{M}\sum_{t=1}^{M}(y_{12,32,13}-\hat{y}_{12,32,13})^2}$$

$$RMS\left(\widetilde{z}_{12,32,13}\right) \simeq \lim_{M\to\infty}\sqrt{\frac{1}{M}\sum_{t=1}^{M}(z_{12,32,13}-\hat{z}_{12,32,13})^2}$$

The RMS from M Monte-Carlo runs for translational relative **velocity** is,

$$RMS\left(\dot{\widetilde{x}}_{12,32,13}\right) \simeq \lim_{M\to\infty}\sqrt{\frac{1}{M}\sum_{t=1}^{M}(\dot{x}_{12,32,13}-\dot{\hat{x}}_{12,32,13})^2}$$

$$RMS\left(\dot{\widetilde{y}}_{12,32,13}\right) \simeq \lim_{M\to\infty}\sqrt{\frac{1}{M}\sum_{t=1}^{M}(\dot{y}_{12,32,13}-\dot{\hat{y}}_{12,32,13})^2}$$

$$RMS\left(\dot{\widetilde{z}}\widetilde{z}_{12,32,13}\right) \simeq \lim_{M\to\infty}\sqrt{\frac{1}{M}\sum_{t=1}^{M}(\dot{z}_{12,32,13}-\dot{\hat{z}}_{12,32,13})^2}$$

The absolute error in the **relative distance**, $\tilde{\rho}_{ij}$, with $ij = 12, 23, 13$, is calculated as the RMS,

$$RMS\left(\widetilde{\rho}_{ij}\right) \simeq \lim_{M\to\infty}\sqrt{\frac{1}{M}\sum_{t=1}^{M}(\rho_{ij}-\hat{\rho}_{ij})^2}$$

The RMS from M Monte-Carlo runs for **Euler** angles is,

$$\begin{aligned}
RMS\left(\widetilde{\psi}_{_{1,2,3}}\right) &\simeq \lim_{M\to\infty}\sqrt{\frac{1}{M}\sum_{t=1}^{M}(\psi_{1,2,3}-\hat{\psi}_{1,2,3})^2}\\
RMS\left(\widetilde{\theta}_{1,2,3}\right) &\simeq \lim_{M\to\infty}\sqrt{\frac{1}{M}\sum_{t=1}^{M}(\theta_{1,2,3}-\hat{\theta}_{1,2,3})^2}\\
RMS\left(\widetilde{\phi}_{1,2,3}\right) &\simeq \lim_{M\to\infty}\sqrt{\frac{1}{M}\sum_{t=1}^{M}(\phi_{1,2,3}-\hat{\phi}_{1,2,3})^2}
\end{aligned}$$

The RMS from M Monte-Carlo runs for absolute **angular velocity** is,

$$\begin{aligned}
RMS\left(\left(\widetilde{\omega}_x\right)_{_{1,2,3}}\right) &\simeq \lim_{M\to\infty}\sqrt{\frac{1}{M}\sum_{t=1}^{M}((\omega_x)_{1,2,3}-(\hat{\omega}_x)_{1,2,3})^2}\\
RMS\left(\left(\widetilde{\omega}_y\right)_{1,2,3}\right) &\simeq \lim_{M\to\infty}\sqrt{\frac{1}{M}\sum_{t=1}^{M}((\omega_y)_{1,2,3}-(\hat{\omega}_y)_{1,2,3})^2}
\end{aligned}$$

$$RMS\left(\left(\widetilde{\omega}_z\right)_{1,2,3}\right) \simeq \lim_{M\to\infty}\sqrt{\frac{1}{M}\sum_{t=1}^{M}((\omega_z)_{1,2,3}-(\hat{\omega}_z)_{1,2,3})^2}$$

6.5.8 Normalized estimation error squared test

According to the discussion performed in Section 5.4 for the NEES test, we will use the limits of the 95% probability region of the chi-square distribution to state about the navigation algorithms consistency. The limits applied for the case study follow.

Relative position and translational velocity

For a $M = 1$ run Monte-Carlo test and with a dimension of the state vector $n_x = 18$ (relative position and translational velocity state vector only), the NEES test is defined as the two-sided probability region of 95% used for a 18DoF chi-square random variable:

$$\left[\chi^2_{18}(0.05), \chi^2_{18}(0.95)\right] = [9.40; 28.9] \tag{6.52}$$

In case the state vector has the dimension $n_x = 12$, due to the observable components only, the NEES test is defined as the two-sided probability region of 97.5% used for a 18DoF chi-square random variable:

$$\left[\chi^2_{12}(0.05), \chi^2_{12}(0.95)\right] = [5.23; 21]$$

For a $M = 10$ run Monte-Carlo test and with a dimension of the state vector $n_x = 18$ (relative position and translational velocity state vector only), the NEES test is defined as the two-sided probability region of 95% used for a 18DoF chi-square random variable:

$$\left[\chi^2_{180}(0.05), \chi^2_{180}(0.95)\right] = [149,97; 212,30]$$

which dividing by $M = 10$, the 95% probability region for the average normalized state estimation error squared becomes

$$\left[\chi^2_{M=10}(0.05), \chi^2_{M=10}(0.95)\right] = [15; 21,2] \tag{6.53}$$

Attitude and angular velocity

For a $M = 1$ run Monte-Carlo test and with a dimension of the state vector $n_x = 6$ (applicable to attitude and angular velocity state vector in s/c 1 and *w.r.t.* variables from s/c 1 only), the NEES test is defined as the two-sided probability region of 95% used for a 6DoF chi-square random variable,

$$\left[\chi^2_{6}(0.05), \chi^2_{6}(0.95)\right] = [1.64; 12.6] \tag{6.54}$$

For the situation where there is contribution of the predecessor state estimates to compute the non-observable components, then the dimension of the state vector is 18 and the two-sided probability region use is according to Eq. (6.52).

For a $M = 10$ run Monte-Carlo test and with a dimension of the state vector $n_x = 6$ (applicable to attitude and angular velocity state vector in s/c 1 and *w.r.t.* variables from s/c 1 only), the NEES test is defined as the two-sided probability region of 95%, used for a 6DoF chi-square random variable according to

$$\left[\chi^2_{60}(0.05), \chi^2_{60}(0.95)\right] = [43.2; 79.1]$$

which dividing by $M = 10$, the 95% probability region for the average normalized state estimation error squared becomes

$$\left[\chi^2_{M=10}(0.05), \chi^2_{M=10}(0.95)\right] = [4.3; 7.9] \tag{6.55}$$

6.5.9 Normalized innovation squared test

According to the discussion performed in Section 5.4 for the NIS test, we will use the limits of the 95% probability region of the chi-square distribution to state about the innovation consistency. It follows the limits applied for the case study.

Relative position and translational velocity

For a $M = 1$ run Monte-Carlo test and with a dimension of the state vector $n_y = 6$, the NIS test is defined as the two-sided probability region of 95%, used for a 6DoF chi-square random variable according to Eq. (6.54).

For a $M = 10$ run Monte-Carlo test and with a dimension of the state vector $n_y = 6$, the NIS test is defined as the two-sided probability region of 95%, used for a 6DoF chi-square random variable according to Eq. (6.55).

Attitude and angular velocity

For a $M = 1$ run Monte-Carlo test and with a dimension of the state vector $n_y = 3$, the NIS test is defined as the two-sided probability region of 95%, used for a 3DoF chi-square random variable according to,

$$\left[\chi^2_3(0.05), \chi^2_3(0.95)\right] = [0.352; 7.81]$$

For a $M = 10$ run Monte-Carlo test and with a dimension of the state vector $n_y = 3$, the NIS test is defined as the two-sided probability region of 95%, used for a 3DoF chi-square random variable according to

$$\left[\chi^2_{30}(0.05), \chi^2_{30}(0.95)\right] = [18.5; 43.8]$$

which dividing by $M = 10$, the 95% probability region for the average normalized state estimation error squared becomes,

$$\left[\chi^2_{M=10}(0.05), \chi^2_{M=10}(0.95)\right] = [1.9; 4.4]$$

6.6 Summary

In this chapter we presented the methodology discussed in Chapter 4 of applying the 6DoF Navigation algorithms to the FAC mode of the GTO experiment for 3-FF s/c. We started with a brief summary of the GTO mission, its modes, initial conditions for FAC mode in the simulation, a description of the sensors used and the GNC requirements. Also, the general form of the dynamic equations that govern the system, including their implementation is presented.

The chapter closed with description of the simulation setup, a summary of the experiments, and the error tests used to analyse the navigation algorithms results.

Chapter 7

RESULTS

7.1 Introduction

In this chapter, the results from applying the 6DoF Navigation algorithms to the FAC and BCM mode of the GTO experiment for 3-FF s/c, are presented. The FF-FES simulator is described in detail in Section 6.1.

We start in Section 7.2 with a summary of the system initial conditions and we present, the calculation of the process and measurement noise covariance matrices from the sensors and noise characteristics. The measurements accuracy study attained for both RF and ST sensors, follow. For this purpose the simulator runs in two orbits trajectory starting with FAC mode initial conditions, showing the behaviour of the s/c in the formation in open-loop and closed-loop. In both situations, it is shown how the distance among s/c influences the RF measurements accuracy. As a result of the sensor measurements study the system is always in closed-loop with or without the navigation algorithms within the loop. In Section 7.3, the initial conditions used on the navigation algorithms, which are different from the real ones, are presented. A method to tune the process and measurement noise covariance matrices follows, where given the bounds of the consistency tests, specified in Section 6.5, the process and measurement noises matrices are obtained in such a way that the NEES and NIS tests are satisfied. Therefore, for each segment of the GTO trajectory being studied, these matrices are tuned according to the explanation provided. Section 7.2.4 and the remainder of the chapter is dedicated to the case study of a formation of 3-s/c, results are presented and performed according to the detailed description in Section 6.4. The navigation algorithm conditions applied to the FAC and BCM modes of the GTO orbit are summarized in Table 6.13. Therefore, this section is divided in two parts: results of the navigation algorithms application to FAC and BCM segments of the orbit. Thus, the results of the closed-loop on FAC mode,

with the navigation algorithm as observer, for different case tests (experiments 1-4) are presented. In experiments 5 and 7 we present the results of the GC control used in FAC and BCM mode, respectively, with no estimator in the loop. In experiment 6, the formation state estimator is in closed-loop with the GC algorithm for FAC mode. In experiment 8, the GC is used for the BCM mode, and again the formation state estimator is used only as observer. The experiments finalize with experiment 9 where 10-run Monte-Carlo results for BCM mode are presented. The chapter closes with a summary of the main achievements obtained. Without loss of generality, all the results concerning relative position and translational relative velocity shown in this chapter concern the estimated state of the formation at s/c 1 and are presented in the LVLH frame. Also, all the results concerning absolute attitude and absolute angular velocity shown in this chapter concern the estimated state at s/c 1 but they are presented in the inertial IPQ frame.

7.2 System Conditions

In this section we present the system initial conditions for the FAC and BCM segments of the orbit. In subsection 7.2.2 is explained how the process noise covariance matrix is obtained directly from the simulator and in subsection 7.2.3 the same procedure is applied in order to obtain the measurement noise covariance matrices when the RF sensor or the ST sensor are considered. An analysis of the sensor measurements accuracy as the motion of 3-FF s/c in the formation changes, during two orbits trajectory on a GTO mission, follows in subsection 7.2.4. It is shown how much the RF measurements accuracy depends directly on the distance among s/c. This is achieved by comparing the RF measurements accuracy attained for open-loop and closed-loop since in closed-loop the control and guidance algorithms will keep the s/c from drifting apart from each other. This is an important issue in navigation algorithms due to the measurements accuracy impact on the navigation results.

7.2.1 Initial conditions

In this chapter, two parts of the GTO are considered: FAC mode and BCM mode.

- FAC mode corresponds to 6 hours trajectory centered around apogee (more detail in subsection 6.2.3). For FAC mode, the trajectory starts at

$$\theta_t\left(0\right) = 156,27[^o](t_o = 10716.94[s]) \tag{7.1}$$

  as shown in Figure 7-1.

- BCM mode corresponds to 2 hours trajectory centered around apogee (more detail in subsection 6.2.3). For BCM mode, the trajectory starts at

$$\theta_t\left(0\right) = 173,0721[^o](t_o = 18016.94[s])$$

and lasts for 2 hours, centered in the apogee.

7.2.2 Process noise covariance matrix

The covariance matrix of the process noise, $\mathbf{Q}(k)$ is calculated for the signal obtained by subtracting the linearized version of the equation of the relative dynamics (C.22) from the true value of the state which is delivered by the simulator, which has the perturbations switched off,

$$\mathbf{w}_{trans}(t) = \dot{\mathbf{x}}_{trans}(t) - f(\mathbf{x}_{trans}(t), \mathbf{u}(t), t)$$

Using at each step the true anomaly corresponding to time k, the previous equation becomes,

$$\mathbf{w}_{trans}(\theta_t) = \delta\mathbf{x}_{trans}(\theta_t + \Delta\theta_t) - (\mathbf{\Phi}(\theta_t)\delta\mathbf{x}_{trans}(\theta_t) + \mathbf{B}(\theta_t)\mathbf{u}(\theta_t))$$

or

$$\mathbf{w}_{trans}(\theta_t + \Delta\theta_t) = \mathbf{x}_{trans}(\theta_t) - (\mathbf{\Phi}(\theta_t + \Delta\theta_t)\mathbf{x}_{trans}(\theta_t + \Delta\theta_t) + \mathbf{B}(\theta_t + \Delta\theta_t)\mathbf{u}(\theta_t + \Delta\theta_t))$$

where $\Delta\theta_t$ is the θ_t step corresponding to the sampling time T_s, $\mathbf{x}_{trans}(\theta_t)$ and $\mathbf{x}_{trans}(\theta_t + \Delta\theta_t)$ are given by the simulator, $\mathbf{\Phi} \approx \mathbf{1} + \mathbf{F}(\hat{\mathbf{x}}_{trans}\left(\theta_t + \Delta\theta_t\right))T_s + \frac{(\mathbf{F}(\hat{\mathbf{x}}_{trans}(\theta_t + \Delta\theta_t))T_s)^2}{2!}$, with $\mathbf{F}(\hat{\mathbf{x}}_{trans}\left(\theta_t + \Delta\theta_t\right))$ and $\mathbf{B}(\theta_t + \Delta\theta_t)$ are computed according to Eq. (6.30), and $\mathbf{u}(\theta_t + \Delta\theta_t)$ is given by the control signal.

For attitude and angular velocity, the same concept is applied but the quaternion part of the vector has to be handled according to the quaternion rules (subsection 4.1.1). Thus, it follows,

$$\mathbf{w}_{rot}(k) = \begin{bmatrix} \mathbf{q}(k+1) \otimes \left(f\left(q(k), k\right)\right)^{-1} \\ \boldsymbol{\omega}(k+1) - f(\boldsymbol{\omega}(k), \mathbf{u}(k), k) \end{bmatrix}$$

For simplicity all the process noise covariance matrices obtained are diagonal. This is the starting point for the process error analysis and therefore, used as nominal matrix (results shown in experiment 1) in the tuning procedure of the process noise covariance matrix described in subsection 7.3.2, such that the NEES consistency test is attained, defined in subsection 5.4.1.

7.2.3 Measurement noise covariance matrix

RF subsystem

Following the same path of the previous section, the actual measurement noise covariance matrix can then be computed as follows: considering the measurement equation

$$\boldsymbol{\nu}(k) = \mathbf{y}(k) - h^i(\mathbf{x}_{trans}(k), k) \tag{7.2}$$

where $\mathbf{y}(k)$ are the RF sensor measurements, $\mathbf{x}_{trans}(k)$ is the true value of the state vector given by the simulator, and the nonlinear observation function $h(\mathbf{x}(k), k)$ is, for s/c i,

$$h^i(\mathbf{x}_{trans}(k), k) = \left[\begin{array}{ccc} \left\|\boldsymbol{\chi}_j - \boldsymbol{\chi}_i\right\|^{R_1} & \left\|\boldsymbol{\chi}_j - \boldsymbol{\chi}_i\right\|^{R_2} & \left\|\boldsymbol{\chi}_j - \boldsymbol{\chi}_i\right\|^{R_3} \end{array} \right]^T$$

with

$$\begin{aligned}
\left(\left\|\boldsymbol{\chi}_j - \boldsymbol{\chi}i\right\|^{R_1}\right)^2 &= \left(a_{11}x_{ij}^{LVLH} + a_{12}y_{ij}^{LVLH} + a_{13}z_{ij}^{LVLH} - c_{R_1,x}^{IPQ}\right)^2 + \\
&\quad \left(a_{21}x_{ij}^{LVLH} + a_{22}y_{ij}^{LVLH} + a_{23}z_{ij}^{LVLH} - c_{R_1,y}^{IPQ}\right)^2 + \\
&\quad \left(a_{31}x_{ij}^{LVLH} + a_{32}y_{ij}^{LVLH} + a_{33}z_{ij}^{LVLH} - c_{R_1,z}^{IPQ}\right)^2 \\
\left(\left\|\boldsymbol{\chi}_j - \boldsymbol{\chi}_i\right\|^{R_2}\right)^2 &= \left(a_{11}x_{ij}^{LVLH} + a_{12}y_{ij}^{LVLH} + a_{13}z_{ij}^{LVLH} - c_{R_2,x}^{IPQ}\right)^2 + \\
&\quad \left(a_{21}x_{ij}^{LVLH} + a_{22}y_{ij}^{LVLH} + a_{23}z_{ij}^{LVLH} - c_{R_2,y}^{IPQ}\right)^2 + \\
&\quad \left(a_{31}x_{ij}^{LVLH} + a_{32}y_{ij}^{LVLH} + a_{33}z_{ij}^{LVLH} - c_{R_2,z}^{IPQ}\right)^2 \\
\left(\left\|\boldsymbol{\chi}_j - \boldsymbol{\chi}_i\right\|^{R_3}\right)^2 &= \left(a_{11}x_{ij}^{LVLH} + a_{12}y_{ij}^{LVLH} + a_{13}z_{ij}^{LVLH} - c_{R_3,x}^{IPQ}\right)^2 + \\
&\quad \left(a_{21}x_{ij}^{LVLH} + a_{22}y_{ij}^{LVLH} + a_{23}z_{ij}^{LVLH} - c_{R_3,y}^{IPQ}\right)^2 + \\
&\quad \left(a_{31}x_{ij}^{LVLH} + a_{32}y_{ij}^{LVLH} + a_{33}z_{ij}^{LVLH} - c_{R_3,z}^{IPQ}\right)^2
\end{aligned}$$

where the a_{kl} are the entries of matrix $\mathbf{A}_{LVLH}^{IPQ}$ (attitude matrix that transforms vectors from LVLH frame to IPQ frame). The measurement noise was assumed to be zero-mean and Gaussian, and the covariance matrix of the error noise $\boldsymbol{\nu}(k)$ was assumed diagonal. On the other hand, the RF subsystem measurement noise can be obtained from its physical characteristics, reported in Section 6.2, Table 6.9. Thus, the variance of the measurement noise is,

$$\sigma_v^2 = 4.2149 \cdot 10^{-7} d[m^2]$$

where d is the distance measured between the transmitter s/c and the receiver s/c antennas. For instance, the measurement covariance matrix used in s/c i is

$$\begin{aligned} R^i &= 4.2149 \times 10^{-7} diag \Big(\left\|\boldsymbol{\chi}_j - \boldsymbol{\chi}_i\right\|^{R_1} \quad \left\|\boldsymbol{\chi}_j - \boldsymbol{\chi}_i\right\|^{R_2} \quad \left\|\boldsymbol{\chi}_j - \boldsymbol{\chi}_i\right\|^{R_3} \\ &\qquad \left\|\boldsymbol{\chi}_j - \boldsymbol{\chi}_i\right\|^{R_1} \quad \left\|\boldsymbol{\chi}_j - \boldsymbol{\chi}_i\right\|^{R_2} \quad \left\|\boldsymbol{\chi}_j - \boldsymbol{\chi}_i\right\|^{R_3} \Big) \end{aligned} \tag{7.3}$$

where $i = 1, 2, 3 \neq j$. This value is the starting point for the sensor error analysis and therefore, used as nominal matrix (results shown in experiment 1) in the tuning procedure of the measurement noise covariance matrix described in subsection 7.3.3, such that the NIS consistency test is attained, defined in subsection 5.4.2. The standard deviation envelope (2σ, where σ is computed by taking the square root of the diagonal elements of the measurement noise covariance matrix (7.3)) has to enclose the error (7.2), in other to have consistent measurements, given by the NIS test.

Star tracker sensor

Applying the same concept as in previous section, the measurement noise covariance matrix can be obtained as follows: considering for each of the s/c 1, 2, 3, the measurement equation,

$$\begin{aligned} \mathbf{q}_{ST}^1 &= \left[\begin{array}{ccc} (q_1)_{IPQ}^{b_1} & (q_2)_{IPQ}^{b_1} & (q_3)_{IPQ}^{b_1} \end{array} \right]^T \text{ for s/c 1} \\ \mathbf{q}_{ST}^2 &= \left[\begin{array}{ccc} (q_1)_{IPQ}^{b_2} & (q_2)_{IPQ}^{b_2} & (q_3)_{IPQ}^{b_2} \end{array} \right]^T \text{ for s/c 2} \\ \mathbf{q}_{ST}^3 &= \left[\begin{array}{ccc} (q_1)_{IPQ}^{b_3} & (q_2)_{IPQ}^{b_3} & (q_3)_{IPQ}^{b_3} \end{array} \right]^T \text{ for s/c 3} \end{aligned}$$

the measurement noise covariance matrix is obtained according to,

$$\boldsymbol{\nu}^i(k) = q_{ST}^i(k) \otimes \left(q^j\left(k\right)\right)^{-1} \tag{7.4}$$

where $q_{ST}^i(k)$ are the measurements provided by the ST sensor and $q^i\left(k\right)$ corresponds to the true value of the state vector provided by the simulator. Also, the ST subsystem measurement noise was obtained from its physical characteristics, reported in subsection 6.2.3, Table 6.10. Thus, the variance of the measurement noise is,

$$\sigma_v^2 = \frac{1}{3} \cdot 10^{-10} \tag{7.5}$$

This value is the starting point for the sensor error analysis and therefore, used as nominal matrix (results shown in experiment 1) in the tuning procedure of the measurement noise covariance matrix described in subsection 7.3.3, such that the NIS consistency test is attained, defined in subsection 5.4.2.

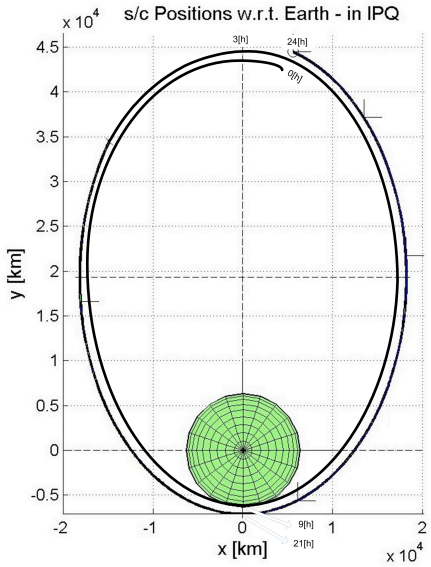

Figure 7-1: 24 hour trajectory for GTO orbit, where the counting of time starts with true anomaly θ_0 and the perigee occurs at 9 and 21 hour.

The standard deviation envelope (2σ, where σ is computed by taking the square root of the diagonal elements of the measurement noise covariance matrix (7.3)) has to enclose the error (7.4), in other to have consistent measurements, given by the NIS test.

7.2.4 Sensor measurements analysis

Radio frequency measurements accuracy study

Running the simulator during two orbits, corresponding to 24 hours in open-loop, starting from θ_o in (7.1) as shown in Fig. 7-1, the diagonal variances were obtained based on the deviation from the nominal performances. The average of the error samples and its envelope are shown in Fig. 7-2. Similar results are obtained for s/c 2 and s/c 3. From the figure, it is clear that the measurements provided by RF subsystem between s/c 1 and 2 (as shown in the top row of Figs. 7-2) are more accurate then the measurements between s/c 1 and 3 (as shown in the bottom row of Fig. 7-2). This is due to the s/c 3 drifting apart from the others s/c (as shown in Fig. 7-3 and Fig. 7-4). Moreover, is clear from Fig. 7-2 that the measurement noise is unbiased.

One way to avoid the formation to drift apart and thus reducing the effect of distance among s/c in the accuracy of the RF observations, is to add to the system the closed-loop algorithm. In fact, in Figs. 7-5 and 7-6 is clear that the formation with GC added, after the initial distance among the s/c

is overcome, the relative distance is kept with a distance of approximately 250 meters from each other. As a consequence, it was verified that the accuracy of all RF measurements is now improved compared to the open-loop run and the innovation is small in all measurements for the closed-loop and inside the bounds of the standard deviation envelope. Moreover, the more precise are the RF measurements

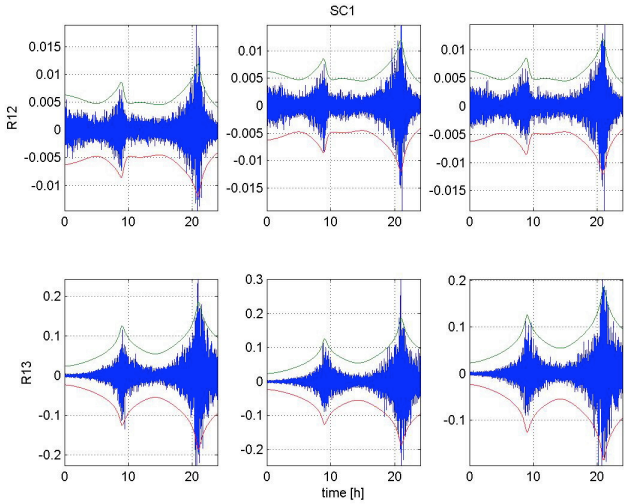

Figure 7-2: **RF measurements real noise at s/c 1**. The real error noise obtained between the receiving antennas in s/c 1 and s/c 2 (top row) and s/c 1 and s/c 3 (bottom row) for RF subsystem. The standard deviation envelope is also plotted and was computed from the measurement noise covariance matrix diagonal elements.

between s/c 1 and 2, the more accurate are the state estimate components. This also indicates that the error from the measurements is smaller than the process error for RF measurements between s/c 1 and 2, making the filter "trust" more the measurements rather than the dynamic model.

As a result and in order to evaluate the navigation algorithms under conditions where the effect of the distance among s/c in the accuracy of the RF observations is reduced, the GC algorithm was added to the system to avoid the formation to drift apart. Therefore, the formation in the experiments presented in Sections 7.4 for FAC mode and 7.5 for BCM segment of the orbit will always run in closed-loop, *i.e.,* the GC algorithms are always added to the system, but the navigation algorithms can be included or not in the loop, depending on the goal of the experiments, according to Table 6.13.

Star tracker measurements accuracy study

Running the simulator during two orbits, corresponding to 24 hours, starting from $\theta_t(0)$ in (7.1) as shown in Fig. 7-1, the diagonal variances were obtained based on the deviation from the nominal

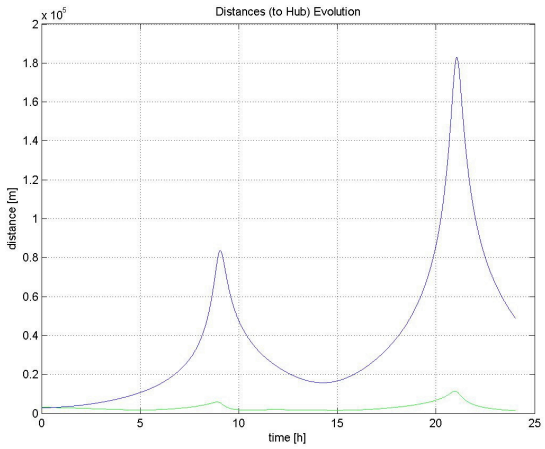

Figure 7-3: **2 orbits run open-loop.** The evolution of the distances between the s/c 1 and s/c 2 (green solid line), and between s/c 1 and s/c 3 (blue solid line) without any GC applied to the formation.

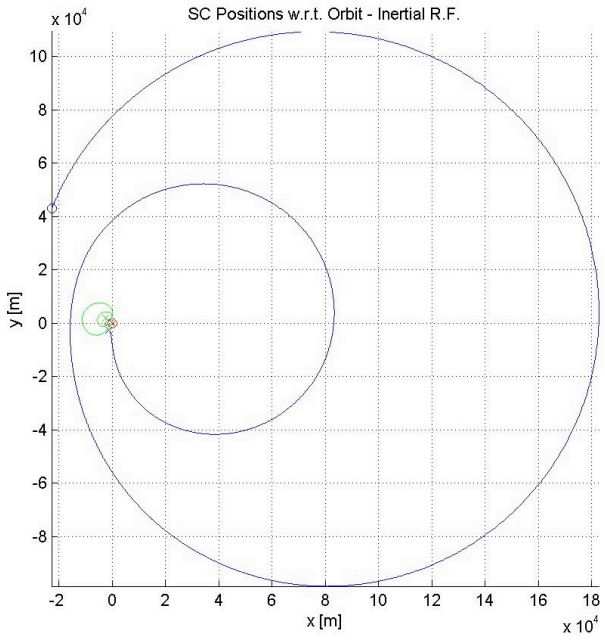

Figure 7-4: **2 orbits run open-loop.** Projection in the x-y plane of the 3 s/c relative trajectories in IPQ without GC applied to the formation. s/c 1 (red solid line), s/c 2 (green solid lie) and s/c 3 (blue solid lie).

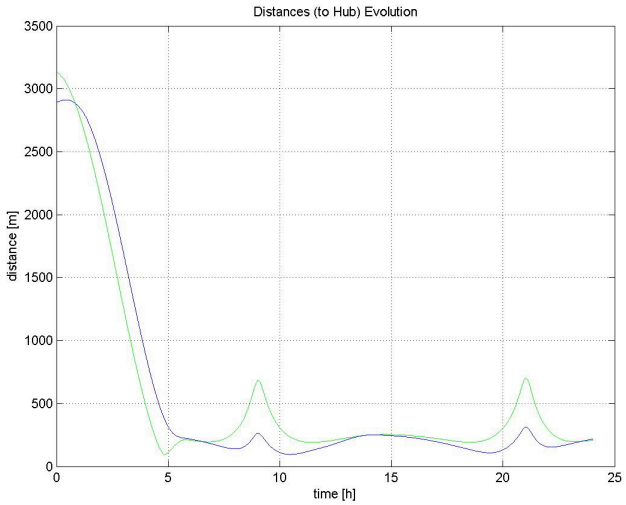

Figure 7-5: **2-orbits run with GC.** The evolution of the distances between the s/c 1 and s/c 2 (green solid lie), and between s/c 1 and s/c 3 (blue solid line) without any GC applied to the formation.

performances. The average of the error samples and its envelope are shown in Fig. 7-7. From the figure it is clear that the error (7.4) is enclosed by its standard deviation computed by (7.5). Therefore, for this trajectory, there is no need to change the variance of the diagonal terms of the measurement noise covariance matrix in order to improve the accuracy as for the RF sensor. Also notice, that the change in the relative quantities does not affect the ST measurements accuracy, since the inertial rotational quantities are not affected by changes in the relative quantities.

7.3 Navigation Algorithms Conditions

In this section we start with the navigation algorithms initial conditions for the FAC and BCM segments of the orbit. Since tuning the process and measurement noise covariance matrices are very important to attain the NEES and NIS consistency tests, a method to tune the process and the measurement noise covariance matrices is presented in subsections 7.3.2 and 7.3.3, respectively.

7.3.1 EKF initial conditions

Relative position and translational velocity initial conditions

For the FAC mode, the initial true state vector corresponds to an initial random disposition of position components within a sphere of $8\,[km]$ diameter, centered in the dispenser, as listed in Table 6.6. For

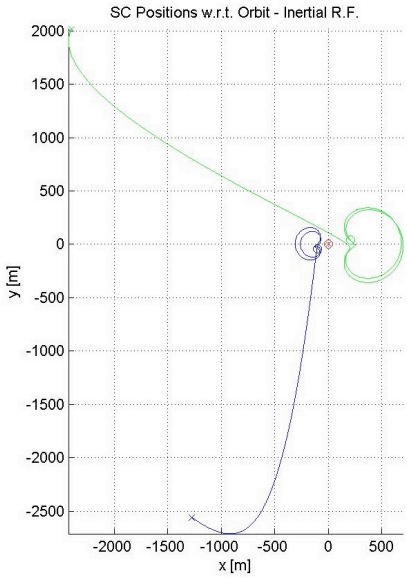

Figure 7-6: **2-orbits run with GC.** Projection in the x-y plane of the 3 s/c relative trajectories in IPQ with the GC applied to the formation. s/c 1 (red solid line), s/c 2 (green solid line) and s/c 3 (blue solid lie).

the BCM mode the initial true state vector corresponds to an initial triangle disposition of position components within a sphere of 250 $[km]$ diameter, centered in the dispenser, as listed in Table 6.8. The initial estimates $\widehat{\boldsymbol{\chi}}(\theta_t\,(0))$ of the state components used in the EKF were, obtained by subtracting $1[m]$[1] from the true relative position components and $1[m/rad]$[1] from the true relative velocity components

$$\begin{aligned}\hat{\boldsymbol{\chi}}(0|0) &= \boldsymbol{\chi}(0|0) - \mathbf{1}[m] \\ \hat{\boldsymbol{\chi}}'(0|0) &= \boldsymbol{\chi}'(0|0) - \mathbf{1}[m/rad]\end{aligned}$$

The state covariance matrix $\mathbf{P}$ was initialized based on a chi-square test for the individual initial estimation errors, according to,

$$\widetilde{\boldsymbol{\chi}}_i(0|0)^T \mathbf{P}_i(0|0)^{-1} \widetilde{\boldsymbol{\chi}}_i(0|0) \leq c$$

where c represents the upper limit of the 95% confidence region for the chi-square distribution with one degree of freedom. $\widetilde{\boldsymbol{\chi}}_i(0|0)$ is the initial error for i^{th} component of the state variable and $\mathbf{P}_i(0|0)$ its associated covariance. Since the initial deviation of the estimate from the true value is not higher then $1[m]$ for relative position and $1[m/rad]$ for translational relative velocity, this is the initial error value

[1] The initial estimation error was part of the ESA project (Formation Estimation Methodologies for Distributed Spacecraft) requirements.

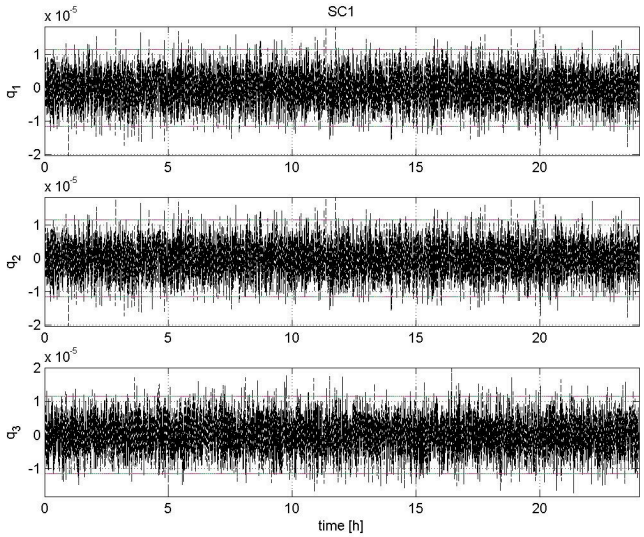

Figure 7-7: **ST measurements real noise at s/c 1.** The real error noise obtained for the star tracker measurements in s/c 1, 2 or 3 enclosed by the standard deviation envelope.

adopted. Thus,

$$\mathbf{P}_i(0|0)^{-1} \leq 3.84$$

The simulations were carried out used

$$\mathbf{P}(0|0) = \mathbf{1}$$

which satisfies the previous inequality.

For the BCM mode, since the formation is already starting in an initial tight configuration, the initial estimates $\widehat{\boldsymbol{\chi}}(\theta_t\,(0))$ of the state components used in the EKF were equal to the initial true state vector.

Attitude and angular velocity initial conditions

The initial true state vector corresponds to approximately $0[rad/s]$ for the angular velocity components and $0[^o]$ for Euler angles, with equal values for all the s/c, as listed in Table 6.7 for FAC mode.

Knowing that the quaternions can parameterize the Euler angles through the Attitude matrix, according to (6.48), thus the initial state vector corresponds to $(\phi, \theta, \psi) = (0[^o], 0[^o], 0[^o])$. The initial estimates $\widehat{\mathbf{x}}(0|0)$ of the state components used in the EKF were, except when noted, obtained by subtracting $10[\text{arcsec}\,/s]^2$ from the true angular velocity components and $\sim 10[^o]$(exactly $9.9981[^o]^2$) from

[2]The initial estimation error was part of the ESA project (Formation Estimation Methodologies for Distributed Spacecraft) requirements.

s/c1≡s/c2≡s/c3	$\hat{\mathbf{x}}_{rot}(0\|0)$
$\omega_x^{b_1}[rad/s]$	4.848×10^{-5}
$\omega_y^{b_1}[rad/s]$	4.848×10^{-5}
$\omega_z^{b_1}[rad/s]$	4.848×10^{-5}
$q_1^{b_1}$	0.0941
$q_2^{b_1}$	0.0789
$q_3^{b_1}$	0.0941
$q_4^{b_1}$	0.9880

Table 7.1: Initial Navigation conditions for attitude and angular velocity. s/c 1 is equal to s/c 2 and s/c 3.

the true Euler angle components which corresponds to a quaternion

$$q = \begin{bmatrix} 0.0941 & 0.0789 & 0.0941 & 0.988 \end{bmatrix}^T$$

Thus, the initial state vector estimates assumed are,

$$\begin{aligned} \hat{\mathbf{x}}_{\omega}(0|0) &= \mathbf{x}_{\omega}(0|0) - \mathbf{10}[\text{arcsec}/s] \\ \hat{\mathbf{x}}_{q}(0|0) &= \mathbf{x}_{q}(0|0) - \mathbf{10}[^o] \end{aligned}$$

as listed in Table 7.1.

For the BCM mode, since the formation is already starting in an initial tight configuration, the initial estimates $\hat{\mathbf{x}}_{rot}$ of the state components used in the EKF were equal to the initial true state vector.

7.3.2 Tuning the process noise covariance matrix

To tune the process noise covariance matrix, three different situations are considered depending on the amount of state vector components being estimated and the Monte-Carlo simulations. For $M = 1$ Monte-Carlo test considering only the observable components (state vector dimension is $n_x = 12$ for the translational state vector and $n_x = 6$ for the rotational state vector), the two-side probability regions of 95% of the NEES test used are defined in subsection 6.5.8. On the other hand, if all state vector components are computed, the state vector dimension is $n_x = 18$ and the limits for the NEES plot change, as presented in the same subsection. For $M = 10$ Monte-Carlo test, all components are computed, but the bounds of the NEES test are now narrower, illustrating the variability reduction resulting from averaging those 10 simulations. Again, the limits are defined in subsection 6.5.8.

The NEES plot given by $\tilde{\mathbf{x}}^T\mathbf{P}^{-1}\tilde{\mathbf{x}}$ should be inside the region indicated by the chi-square test to be considered consistent, *i.e.,* the state vector error must be commensurate to its error covariance matrix.

In case the plot is above the upper limit, that the error covariance matrix is too small compared to the error state vector, meaning that the precision is high but the accuracy is low. In case the plot is below the lower limit, the error covariance matrix is too large compared to the error state vector, meaning that the precision is low but the accuracy is comparably higher, as discussed in subsection 5.4.1.

For a scalar time-invariant linear system, as shown in subsection 5.2.2, the standard deviation of the error obtained is the standard deviation corresponding to the minimum between the process and measurement standard deviations. In nonlinear systems the values change of one entity influence the other entity. Thus, the lower (higher) the standard deviation of the diagonal elements of the error covariance matrix, the higher (lower) the NEES plot gets, so the standard deviation of the diagonal elements of the error covariance matrix has to be increased (decreased). This is achieved by increasing (decreasing) the measurement noise and the process noise covariance matrices. The algorithm to compute the process noise covariance matrix is described in subsection 7.2.2,

1. the real process error is computed from the simulator;
2. the process noise covariance matrix is computed, based on the previous error;
3. plot the standard deviation envelope (2σ, where σ is calculated by taking the square root of the diagonals of the process noise covariance matrix) with the process error, previously determined. The envelope has to enclose the error, in order to have consistent estimations;
4. Run the navigation algorithms;
5. plot the estimation error of the state vector variables and their standard deviation envelope (2σ, where σ is calculated by taking the square root of the diagonals of the error covariance matrix). The envelope has to enclose the error, in order to have consistent estimations;
6. compute the NEES and NIS plots to confirm or to tune the values of the process noise covariance matrix and repeat the procedure, *i.e.,* return to step 4.

Items 1, 2 and 3 correspond to the algorithm to compute the process noise covariance matrix is described in subsection 7.2.2.

7.3.3 Tuning the measurement noise covariance matrix

To tune the measurement noise covariance matrix, three different situations are considered depending on the sensors being estimated and the Monte-Carlo simulations. For $M = 1$ Monte-Carlo test considering the RF sensor (measurement vector dimension is $n_y = 6$) and the ST sensor (measurement vector dimension is $n_y = 3$), the two-side probability regions of 95% of the NIS test used are defined in

subsection 6.5.9. For $M = 10$ Monte-Carlo test the bounds of the NIS test are now narrower, illustrating the variability reduction resulting from averaging those 10. Again, the limits are defined in subsection 6.5.9.

The NIS curve given by $\boldsymbol{\nu}^T\mathbf{S}^{-1}\boldsymbol{\nu}$ should be inside the region indicated by the chi-square test to be considered consistent concerning the observation model and the measurements, *i.e.,* for the innovation process to be commensurate with its innovation covariance matrix, $\mathbf{S}(k) = \mathbf{H}(k)\mathbf{P}(k)\mathbf{H}(k)^T + \mathbf{R}(k)$. In case the NIS plot is above the region upper limit, the innovation covariance matrix is too small compared to the innovation error vector, meaning that the precision of measurements is high but its accuracy is low. In case the NIS plot is below the lower limit, the innovation covariance matrix is too large compared to the innovation error vector meaning that the measurements precision is low but the accuracy is comparably higher, as discussed in subsection 5.4.2.

For a scalar time-invariant linear system, as shown in subsection 5.2.2 the standard deviation of the innovation signal obtained is the standard deviation of to the minimum between the error and measurement standard deviations. In nonlinear systems the values change of one entity influence the other entity. Thus, the lower (higher) the standard deviation of each of the diagonal elements of the innovation covariance matrix, the higher (lower) the NIS plot gets, larger (smaller) the standard deviation of each of the innovation covariance matrix diagonal elements must be. This is accomplished by increasing (decreasing) the measurement noise and the process noise error covariance matrices.

1. the real measurement error is computed from the simulator;
2. the measurement noise covariance matrix is computed, based on the sensor physical characteristics;
3. plot the diagonal variance envelope (2σ, where σ is calculated by taking the square root of the diagonals of the measurement noise covariance matrix) with the measurement error, previously determined. The envelope has to enclose the error, in order to have consistent measurements;
4. Run the navigation algorithms;
5. plot the innovation process and their standard deviation envelope (2σ, where σ is calculated by taking the square root of the diagonals of the innovation covariance matrix). The envelope has to enclose the innovation process, in order to have consistent estimations;
6. and compute the NEES and NIS plots to confirm or to tune the values of the measurement noise covariance matrix and repeat the procedure, *i.e.,* return to step 4.

Items 1, 2 and 3 correspond to the algorithm to compute the measurement noise covariance matrix is described in subsection 7.2.3.

7.4 Results for Formation Acquisition Mode

All the results concerning relative position and translational relative velocity shown in this section refer to the estimated full state of the formation at s/c 1 and are represented in the LVLH frame. All the results concerning absolute attitude and absolute angular velocity shown in this section concern the estimated state at s/c 1 and are represented in the inertial IPQ frame.

Since, in subsection 7.2.4 the RF measurements study have shown the need to add the GC algorithm to the system all the time in order to avoid the s/c formation to drift apart and thus degradating the RF accuracy. Therefore all the results presented are in closed-loop with the navigation algorithms within or not the closed-loop, depending on the goal of the experiments, according to Table 6.13.

7.4.1 Experiment 1 - local extended Kalman filter only with nominal measurement noise

We can observe from Figs. 7-8 to 7-11 the results in s/c 1 when the nominal process and measurement error covariance matrices of translational relative algorithm and rotational absolute algorithm, as presented in subsections 7.2.2 and 7.2.3. For the navigation algorithms every $t = 10[s]$ the filtering part of the EKF receives measurements from the RF subsystem concerning the distances from s/c 1 to s/c 2 and 3. As there are no measurements to update the state components concerning the distance between s/c 2 and 3, these components are propagated through the dynamic equations of motion. The translational and rotational algorithms are working independently, *i.e.,* the state vector estimates obtained in the rotational absolute algorithm are not being used by the translational algorithm, since for now, we want to study the navigation algorithms characteristics without influence on each other.

In Fig. 7-8 shows the innovation process obtained in local EKF, obtained in s/c 1 which is computed as the difference between the RF measurements at the receiving antennas in s/c 1 and s/c 2 (top row) or s/c 1 and s/c 3 (bottom row) and a sensor model, $\nu^1 = y^1 - h(\hat{\chi}^1)$. The standard deviation envelope shown is obtained from the innovation matrix ($\mathbf{S} = \mathbf{HPH}^T + \mathbf{R}$) diagonal elements. We can observe from Fig. 7-8 that the envelope is far from the mean measurement error attained. This is also confirmed by the NIS test (as shown in the bottom of Fig. 7-10), indicating that the measurement noise covariance matrix should be decreased. Also the NEES test (as shown in the top of Fig. 7-10), indicating that the process noise covariance matrix should be decreased. Also, we can observe that the EKF error covariance matrix diagonal elements are far from the lower bound given by the PCRLB (as shown in Fig. 7-9). Thus, there is still room for improvement and thus either the process and measurement noise covariance matrices could be tuned, as detailed in subsection 7.3.2 and 7.3.3, respectively. However, in nonlinear systems changing one of the matrices (process or measurement noise covariance matrices) is going to influence

both NEES and NIS results. Since the measurement noise was computed from the sensor physical characteristics, thus close to reality, the main change of value will be performed in the process noise covariance matrix and small value changes in the process noise covariance matrix. Therefore, starting with the dynamic equations, as explained in subsection 7.2.2, the process noise covariance matrix was obtained by trial and error (for a 6 hour orbit segment), in order to fulfill the NEES test as explained in detail in subsection 7.3.2. We then plot the estimation error obtained, its envelope and the resulting NEES plot. Only after the NEES test have been close to be fulfilled the measurement noise covariance matrix is turned, as explained in subsection 7.3.3 in order to fulfill the NIS test. Knowing that the RF measurements are not biased, the measurement noise covariance matrix was obtained by trial and error (for a 6 hour orbit segment), starting from the diagonal matrix provided by DEIMOS SA and from the physical characteristics, as explained in subsection 7.2.3. We then plot the innovation process, its envelope and the resulting NIS plot, as explained in detail in subsection 7.3.2. Results from the tuning process are presented in next experiment. We can also observe from Fig. 7-9 (middle row of Fig. 7-9) the results in s/c 1 when there are no measurements to update the state components concerning the distance between s/c 2 and 3, therefore these components are propagated through the dynamic equations of motion. Thus, these relative position non-observable components from the state vector will have its error increased due to the linearization errors without correction by the measurements. Moreover, it is interesting to notice that the y component of relative position (middle column and row of Fig. 7-9) remain within a small range error during the entire 6 hours segment of the orbit. This is due to the dynamics of the s/c fleet: the in-plane motion is x-z and the out-of-plane motion is y. As a consequence, the two motion components are decoupled, and the out-of-plane motion is due to the typical cyclic behavior that comes from small changes in the orbit parameters: inclination, i_{cl} and/or right ascension of the ascending node, R_{aan} of the s/c's *w.r.t.* the formation reference frame, as described by [10].

For the rotational part of the state vector, we can verify that for the quaternions the envelope enclosed the ST error, according to the discussion in subsection 7.2.4. From Fig. 7-11 the NEES-NIS tests show that the NEES test (as shown in the top of Fig. 7-11) is fulfilled while for the NIS test (as shown in the bottom of Fig. 7-11) there is still room for improvement, suggesting that the diagonal of the measurement noise covariance could be decreased. However, in nonlinear systems the change of value of the process noise covariance matrix is going to influence the innovation process and the innovation matrix and thus resulting in changing both NEES and NIS tests. As stated before, since the measurement noise was computed from the sensor physical characteristics, thus close to reality, the main change of value will be performed in the process noise covariance matrix and small value changes in the process noise covariance matrix. Moreover, the process noise covariance matrix obtained in subsection 7.2.2 was computed under nominal conditions and thus not taking into account perturbations, unexpected changes in the simulator

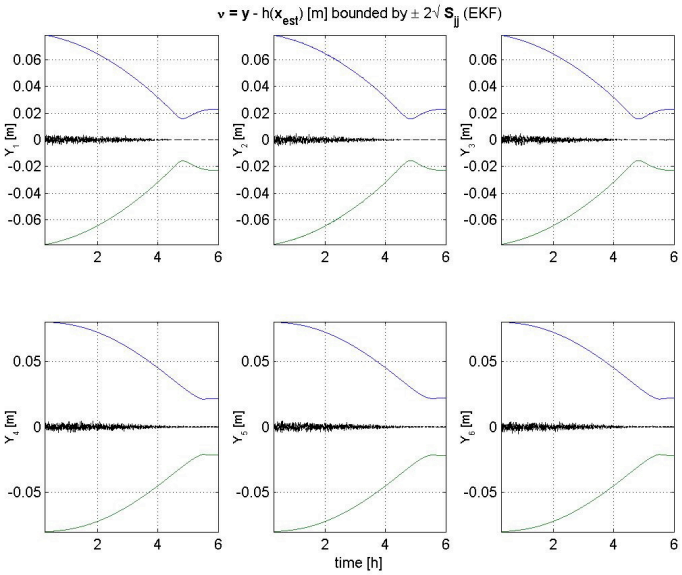

Figure 7-8: **Experiment 1** - The innovation process obtained in local EKF which is computed as the difference between the RF measurements at the receiving antennas in s/c 1 and s/c 2 (top row) or s/c 1 and s/c 3 (bottom row) and a sensor model. Standard deviation envelope obtained from the Innovation matrix ($\mathbf{S} = \mathbf{HPH}^T + \mathbf{R}$) diagonal elements.

or uncertainties. Therefore, the process noise covariance matrix was obtained by trial and error (for a 6 hour orbit), starting from the nominal diagonal matrix, in order to fulfill the NEES test as explained in detail in subsection 7.3.2. We then plot the estimation error obtained, its envelope, and the resulting NEES-NIS plot, as shown in next experiment. Also, knowing that the ST measurements are not biased, the measurement noise covariance matrix was obtained by trial and error (for a 6 hour orbit segment), starting from the diagonal matrix provided by DEIMOS SA and from the physical characteristics, as explained in subsection 7.2.3. We then plot the innovation process, its envelope and the resulting NIS plot, as explained in detail in subsection 7.3.2. Results from the tuning process are presented in next experiment.

7.4.2 Experiment 2 - local extended Kalman filter only with tuned measurement and process noises

This experiment uses the same conditions of experiment 1 but the measurement and process error covariance matrices of translational relative algorithm and rotational absolute algorithm are tuned to fulfill the consistency tests, as explained in previous experiment. We can observe from Fig. 7-12 the

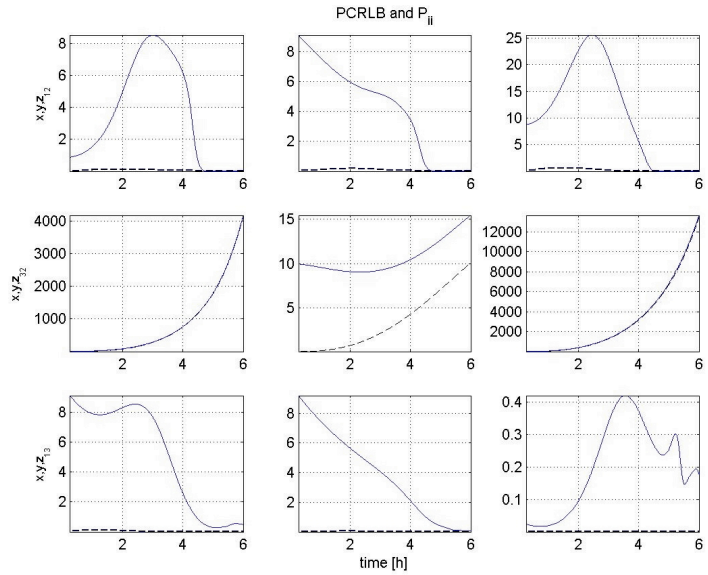

Figure 7-9: **Experiment 1** - Results for the $\mathbf{P}_{EKF}$ (solid line) of relative position obtained in s/c 1 for a 6 hours orbit segment and the PCRLB (dotted line).

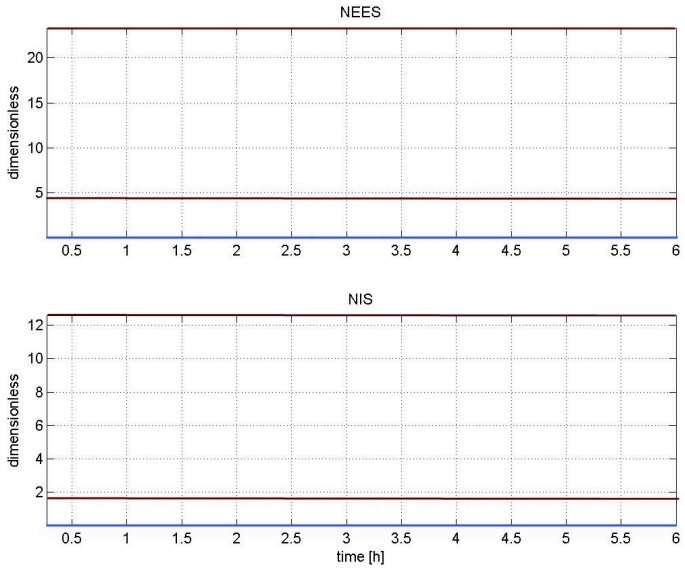

Figure 7-10: **Experiment 1** - NEES-NIS plots for translational state vector. Top plot: NEES test, Bottom plot: NIS test.The plots are computed for all state vector components, $n_x = 12$ corresponding to limits $[5.23; 21]$, and $n_y = 6$ corresponding to limits $[1.64; 12.6]$, on each s/c.

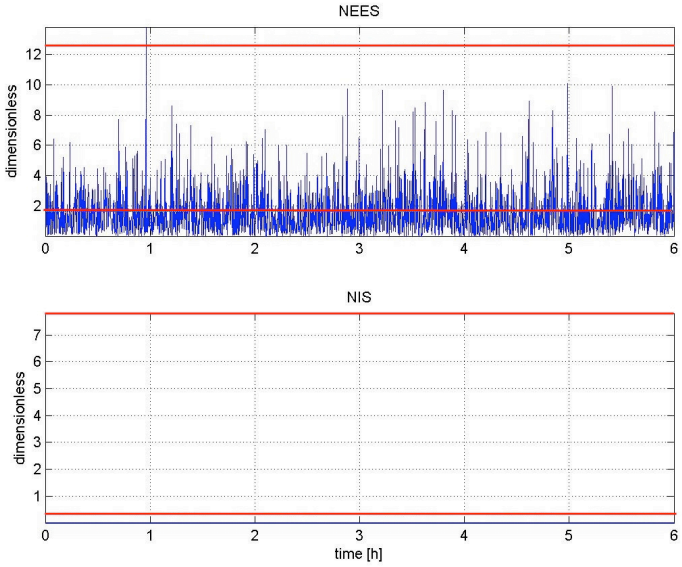

Figure 7-11: **Experiment 1** - NEES-NIS plots for rotational state vector. Top plot: NEES test, Bottom plot: NIS test.The plots are computed for all state vector components, $n_x = 6$ corresponding to limits $[1.64; 12.6]$, and $n_y = 3$ corresponding to limits $[0.352; 7.81]$, on each s/c.

results obtained in s/c 2 where, comparing to experiment 1 (shown in Fig. 7-8) the envelope is closer to the mean error in experiment 2 than in experiment 1. This is to demonstrate the discussion performed in previous experiment, where the process error covariance matrix is decreased in order to have the envelope closer to the estimation error mean, leading to more accurate estimates. Less significantly the measurement noise error covariance is decreased leading to more accurate measurements. This is also confirmed by the NEES-NIS tests in Fig. 7-10, that indicate a very pessimistic filter compared to the NEES-NIS tests from experiment 2 (as shown in Fig. 7-16). So, as we can observe from Fig. 7-16 that most of the points are inside the 95% probability region, except for the initialization of the filter where the error is not commensurable with its error variance, but as soon as this initial period is overcomed, the filter estimates goes bellow the 0.05% probability region, indicating that the state estimates are pessimistic, *i.e.,* the state estimates error are much smaller, comparable to its error variance. This happens when the s/c are quite close, meaning that the error from measurements is even smaller than initially expected, as shown in Fig. 7-12. In this figure the innovation process is shown enclosed by the standard deviation envelope obtained from by taking the square root of the diagonals of the innovation matrix ($\mathbf{S} = \mathbf{HPH}^T + \mathbf{R}$) diagonal elements. The innovation error is computed as the difference between the RF measurements at the receiving antennas in s/c 1 and s/c 2 (top row) or s/c

1 and s/c 3 (bottom row) and a sensor model. Thus, the solution would be to tight the envelope of the measurement noise covariance matrix for this segment of orbit. Moreover, the EKF error covariance matrix diagonal elements are closer in experiment 2 to the lower limit provided by the PCRLB (as shown in Fig. 7-13 and comparing with Fig. 7-9), showing that there is not much more room for improvement in the navigation filter, comparing with the nominal PCRLB curve obtain for experiment 2.

The same way we can verify that the envelope enclosing the quaternions had to be increased for the attitude and angular velocity, leading to improvement in the algorithm performance, as explained in experiment 1. With the process and measurement noise covariance matrices tuned, the EKF error variance is relatively more close to the PCRLB curve, indicating that by tuning the matrices there is still sufficient information in the data to compute the state vector estimates, thus the navigation algorithms are more close to the optimality.

As in experiment 1, we can observe from Fig. 7-14 the results in s/c 1 when there are no measurements to update the state components concerning the distance between s/c 2 and 3, therefore these components are propagated through the dynamic equations of motion. Thus, these relative position non-observable components from the state vector will have its error increased due to the linearization errors, as shown in Fig. 7-13, without correction by the measurements. This is also confirmed by the RMS shown in Fig. 7-15.

For the attitude and angular velocity, only the state vector components with associated measurements in s/c 1 are updated, *i.e.,* only the attitude and the angular velocity of s/c 1 are updated, since only s/c 1 ST is used. The rest of the state vector components, *i.e.,* the angular velocity and quaternions from s/c 2 and 3 are propagated through the dynamic equations of motion with the corresponding error increase. This is confirmed by the Euler angles RMS shown in Fig. 7-20 and the angular velocity RMS shown in Fig. 7-21.

7.4.3 Experiment 3 - local extended Kalman filter and fusion with covariance intersection

Figures 7-23 to 7-34 depict the results obtained in s/c 1 when at times $t = 2k.10\,[s]$, $k = 0, 1, 2, ...$, each s/c computes the state vector estimates based on its local RF measurements using EKF, and at times $t = (2k + 1).10\,[s]$, with $k = 0, 1, 2, ...$, the predecessor state vector is used as an observation in the filtering part, and fused with the local estimate using the CI algorithm. The results are presented for s/c 1 and it can be seen that the behavior of the components concerning the distance between s/c 3 and 2 are improved by the CI algorithm. In Fig. 7-23 and 7-24, the error plots for the relative position and translational relative velocity in s/c 1 are respectively depicted. The results denote that the requirements of $0.1[m]$ and $0.01[m/s]$ are fulfilled for all components at the end of the FAC mode. However this

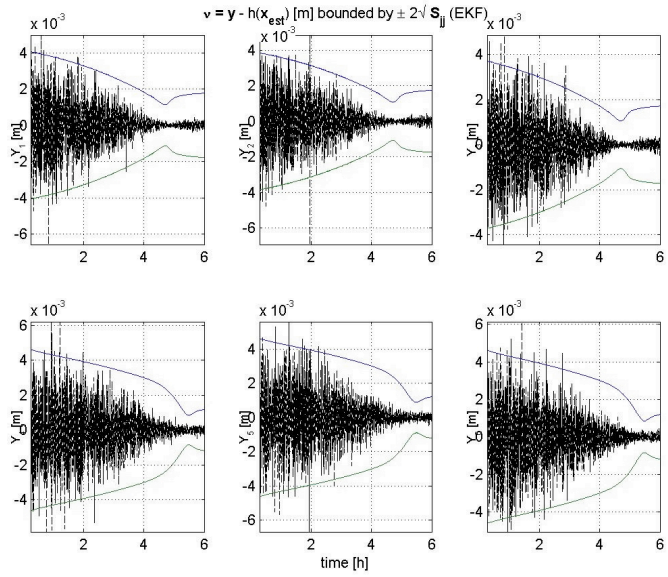

Figure 7-12: **Experiment 2**. The innovation process obtained in local EKF enclosed by the standard deviation envelope. The innovation process is computed as the difference between the RF measuments at the receiving antennas in s/c 1 and s/c 2 (top row) or s/c 1 and s/c 3 (bottom row) and a sensor model.

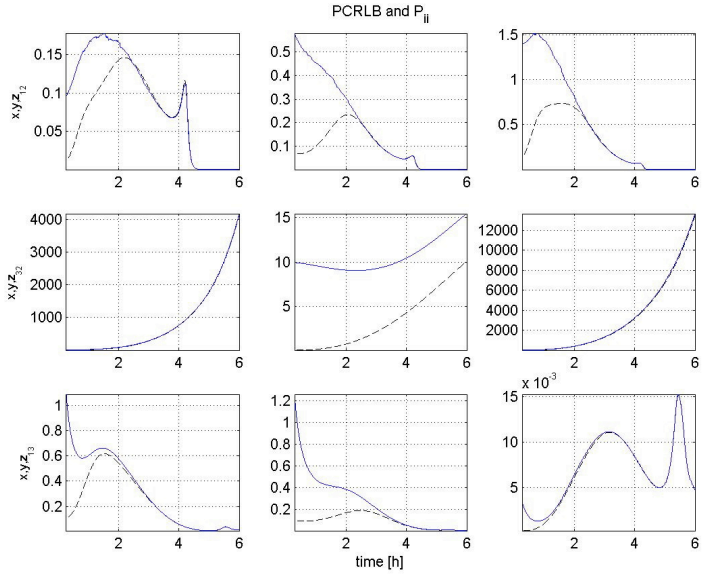

Figure 7-13: **Experiment 2** - Results for the $\mathbf{P}_{EKF}$ (solid line) of relative position obtained in s/c 1 for a 6 hours orbit segment and the PCRLB (dotted line).

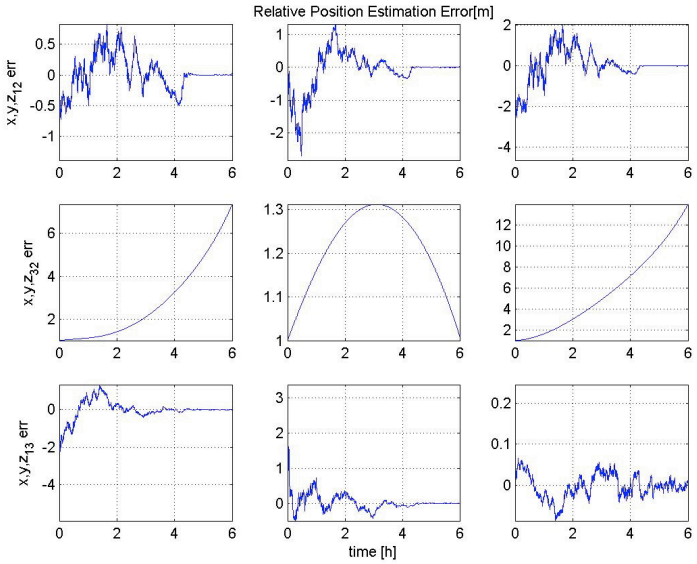

Figure 7-14: **Experiment 2** - Results for the estimation error of relative position obtained in s/c 1 for a 6 hours orbit segment. The xyz_{12} and xyz_{13} components are updated locally with the RF measurements while the components xyz_{32} are just propagated through the equations of motion.

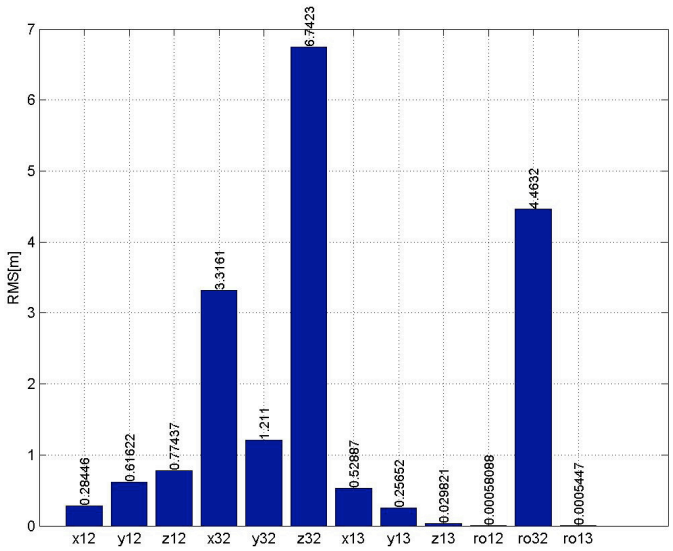

Figure 7-15: **Experiment 2** - Results for the RMS of relative position obtained in s/c 1 for a 6 hours orbit segment. The xyz_{12} and xyz_{13} components are updated locally with the RF measurements while the components xyz_{32} are just propagated through the equations of motion.

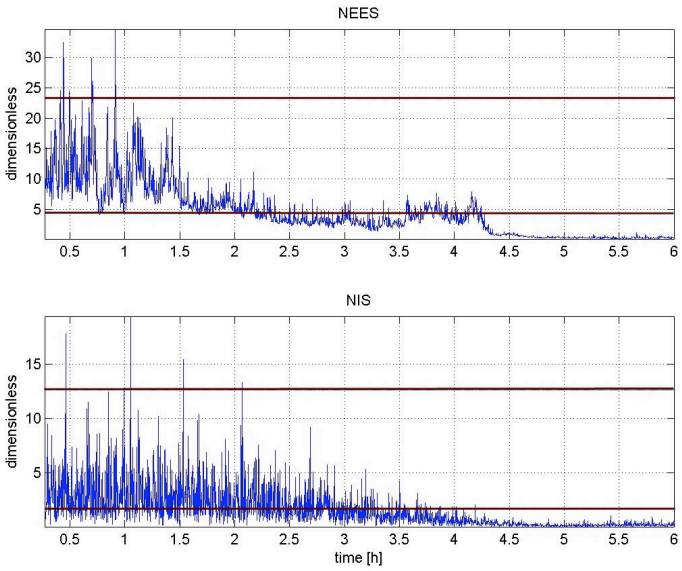

Figure 7-16: **Experiment 2** - NEES-NIS plots for trsnalational state vector. Top plot: NEES test, Bottom plot: NIS test.The plots are computed for all state vector components, $n_x = 12$ corresponding to limits $[5.23; 21]$, and $n_y = 6$ corresponding to limits $[1.64; 12.6]$, on each s/c.

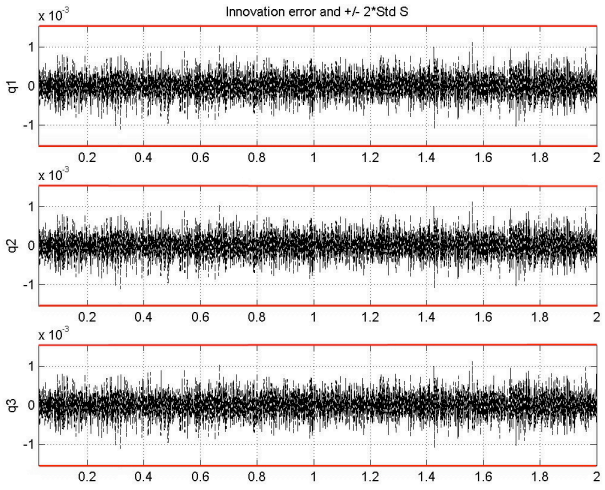

Figure 7-17: **Experiment 2**. The innovation process obtained in local EKF, enclosed by the standard deviation envelope . This is computed as the difference between the ST measaments at the receiving antennas in s/c 1 and s/c 2 (top row) or s/c 1 and s/c 3 (bottom row) and a sensor model.

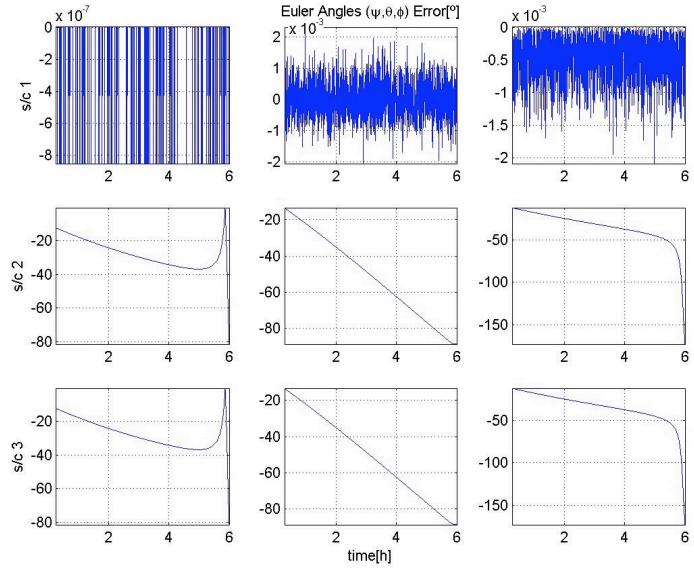

Figure 7-18: **Experiment 2** - Results for the estimation error of the Euler angles obtained in s/c 1 for a 6 hours orbit segment. The components of s/c 1 are updated locally with the ST measurements while the components of s/c 2 and 3 are just propagated through the equations of motion.

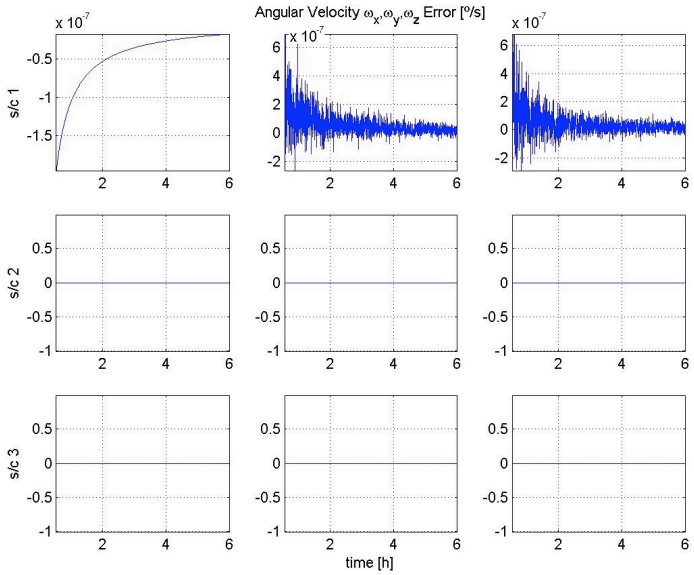

Figure 7-19: **Experiment 2** - Results for the estimation error of the angular velocity for s/c 1 for a 6 hours segment. The components of s/c 1 are updated locally with the ST measurements while the components of s/c 2 and 3 are just propagated through the equations of motion.

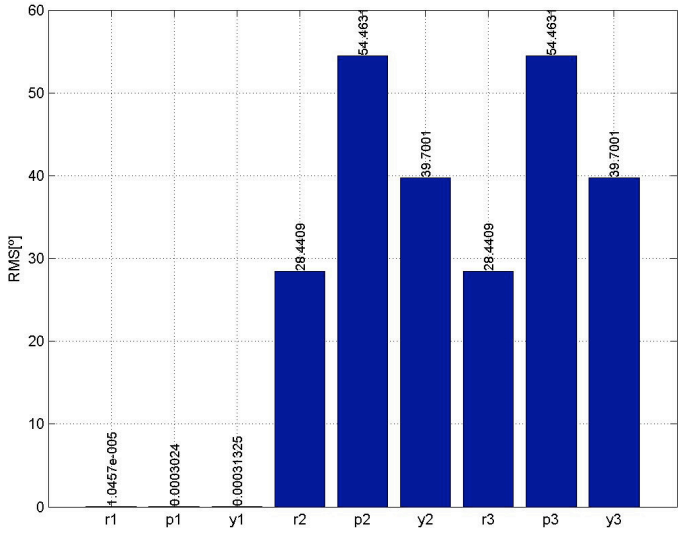

Figure 7-20: **Experiment 2** - Results for the RMS of Euler angles obtained in s/c 1 for a 6 hours orbit segment. The ψ(r1), θ(p1), ϕ(y1) for s/c 1 are updated locally with the ST measurements while for s/c 2 and 3, the angular velocity components are just propagated through the equations of motion.

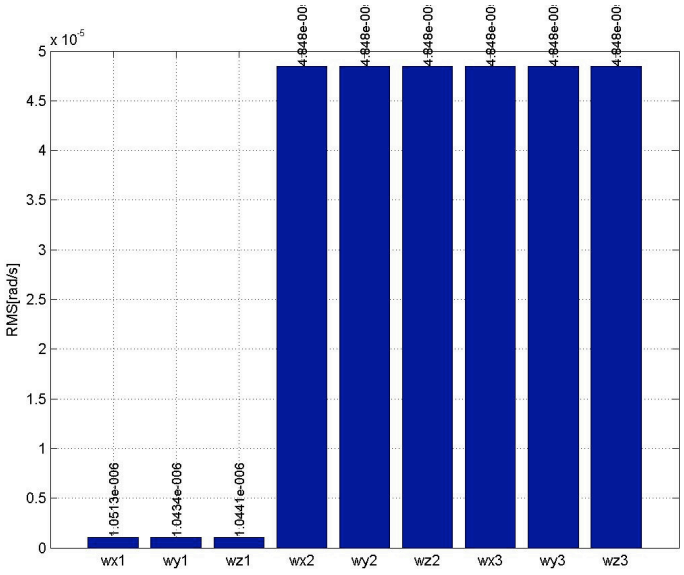

Figure 7-21: **Experiment 2** - Results for the RMS of angular velocity $[rad/s]$ obtained in s/c 1 for a 6 hours orbit segment. The ω_{xyz} for s/c 1 are updated locally with the ST measurements while for s/c 2 and 3, the angular velocity components are just propagated through the equations of motion.

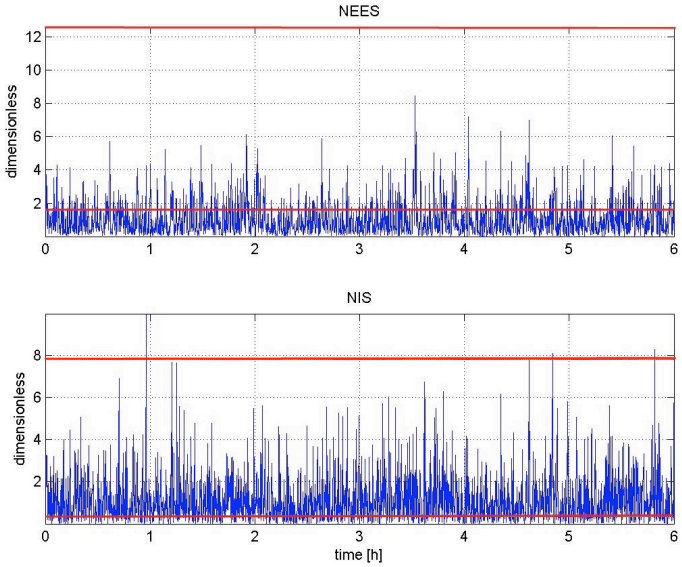

Figure 7-22: **Experiment 2** - NEES-NIS plots for rotational state vector. Top plot: NEES test, Bottom plot: NIS test.The plots are computed for all state vector components, $n_x = 6$ corresponding to limits $[1.64; 12.6]$, and $n_y = 3$ corresponding to limits $[0.352; 7.81]$, on each s/c.

improvement is obtained at the expense of the degradation of the local observable components, since the weighted parameter from the CI method will be a trade off among the variances of all the components of the translational state vector. This affects all components, since the computation of the parameter w is made to minimize the trace of the whole matrix, as shown in Fig. 7-26. In Figs. 7-30 and 7-32, the inertial attitude and angular velocity components are shown, denoting that the specifications are fulfilled for the rotational state components. The NEES and NIS tests for a 6-hour orbit are shown in Fig. 7-29, but now for all translational state vector components and not for the observable components only as presented in experiment 1, shown in Fig. 7-16. It can be seen that, according to the NEES test, the filter is consistent in part of the orbit and then changes to pessimistic. Also notice in the in Fig. 7-28, that since the state vector components $\mathbf{x}_{23}$ are now updated through the CI method, its error variance obtained from the diagonal elements of the error covariance matrix have an improved curve comparing with the optimal given by the PCRLB, after the initialization. This is as expected since now besides having the process and measurement covariance matrices tuned, the navigation algorithms also have "measurements" (predecessor state vector estimates) to update the local non-observable components. The RMS for the translational state vector, Euler angles and the angular velocity are shown in Figs. 7-27, 7-31 and 7-33 respectively, showing that there is an improvement on the state vector components that

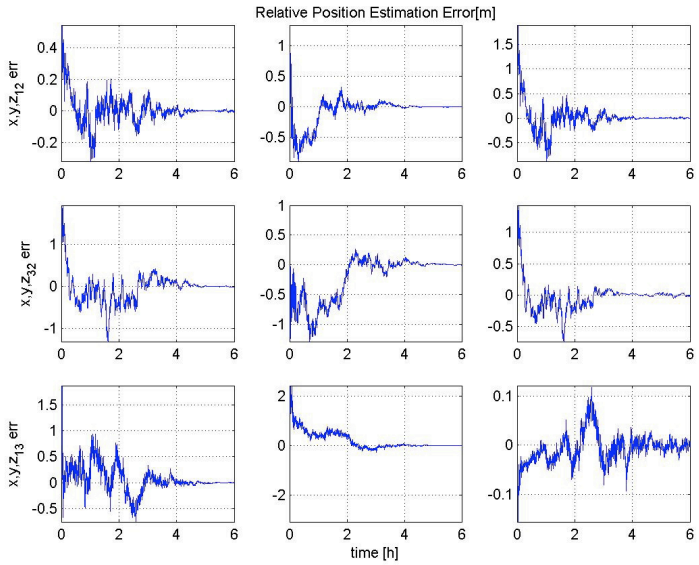

Figure 7-23: **Experiment 3** - Results for the estimation error of relative position obtained in s/c 1 for a 6 hours orbit segment. The components of xyz_{12} and xyz_{13} are updated locally with the RF measurements while the components xyz_{32} are updated due to the CI algorithm.

are not locally observable at the expense of the increased error on the other components, comparing to the equivalent plots from experiment 2. Notice that the RMS values includes also the initial estimation error increasing its values, however, we can observer from Fig. 7-23 that the FAC requirement of $0.1[m]$ for the relative state vector is achieved after the initialization of the navigation filters. The same successful result is obtained for the translational relative velocity, as shown in Fig. 7-24. Also, the RMS of the relative distance, calculated according to subsection 6.5.6, and shown in last three columns of Fig. 7-27 attain the requirements. For the rotational part of the state vector, we can also observe successful results obtained for the Euler angles, which have the requirement of $1[^o]$, as shown in Figs. 7-30 and 7-31 and for the angular velocity, which have the requirement of $0.1[^o/s]$ or $\simeq 0.018[rad/s]$, as shown in Fig. 7-32 and 7-33.

7.4.4 Experiment 4 - local extended Kalman filter and fusion with covariance intersection and simulator with perturbations

This experiment follows the same conditions of experiment 3 but the following nonlinear perturbations are added to the simulator:

- Solar Radiation Pressure

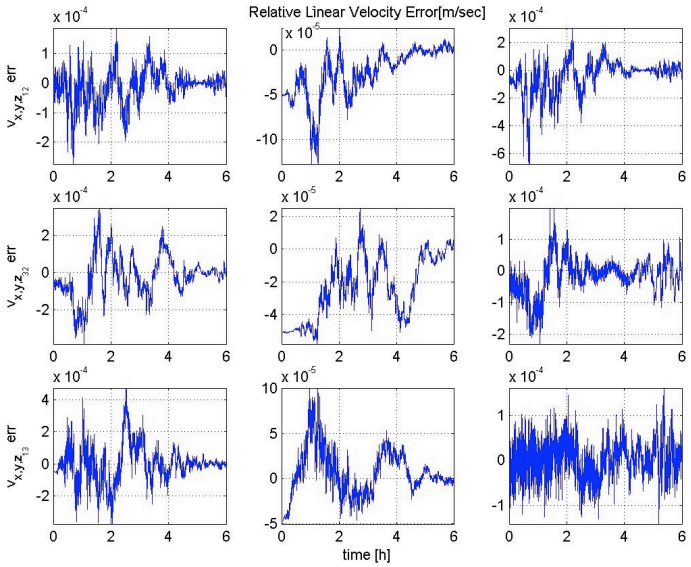

Figure 7-24: **Experiment 3** - Results for the estimation error of relative translational velocity obtained in s/c 1 for a 6 hours orbit segment. The components of xyz_{12} and xyz_{13} are updated locally with the RF measurements while the components xyz_{32} are updated due to the CI algorithm.

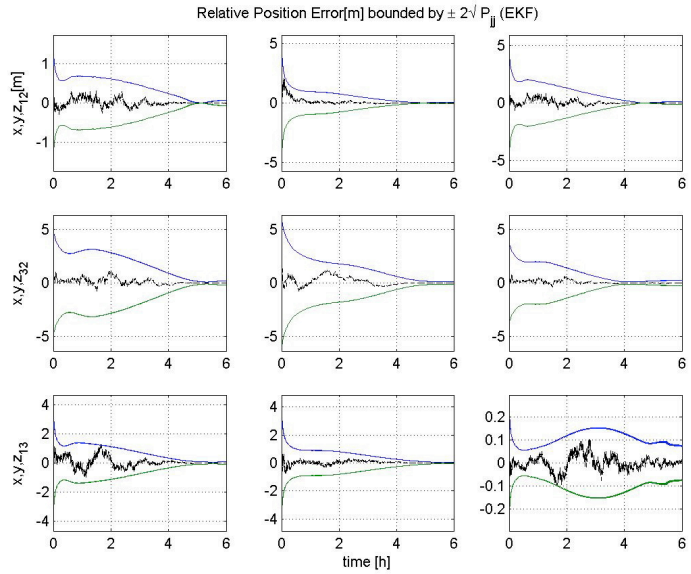

Figure 7-25: **Experiment 3** - Results for the estimation error of relative position enclosed by the standard deviation envelope obtained in s/c 1 for a 6 hours orbit segment. The components of xyz_{12} and xyz_{13} are updated locally with the RF measurements while the components xyz_{32} are updated due to the CI algorithm.

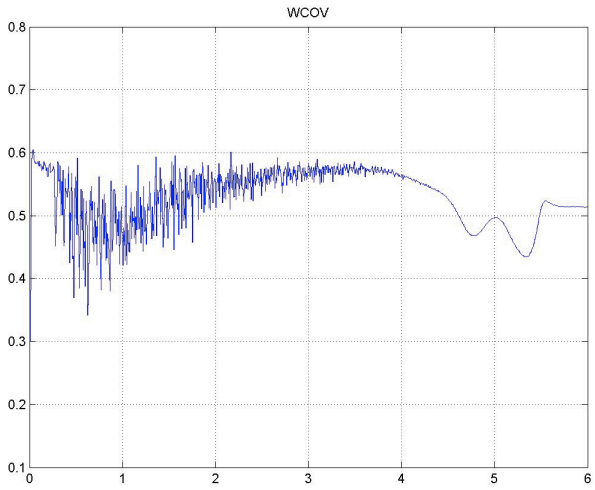

Figure 7-26: **Experiment 3** - Results for the weighted parameter w for s/c 1.

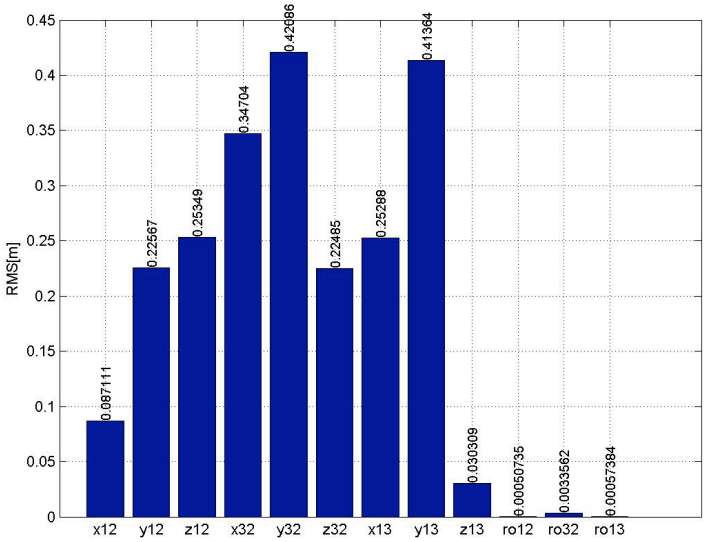

Figure 7-27: **Experiment 3** - Results for the RMS of relative position obtained in s/c 1 for a 6 hours orbit segment. The components of xyz_{12} and xyz_{13} are updated locally with the RF measurements while the components xyz_{32} are updated due to the CI algorithm.

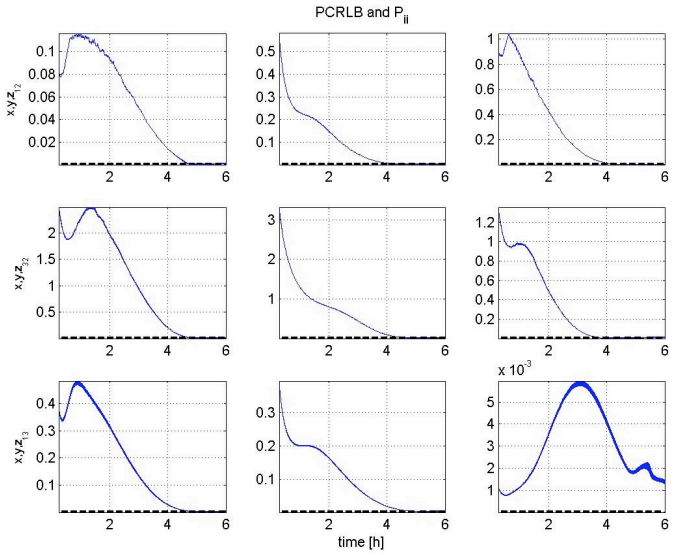

Figure 7-28: **Experiment 3** - Results for the $\mathbf{P}_{EKF}$ (solid line) of relative position obtained in s/c 1 for a 6 hours orbit segment and the PCRLB (dotted line). The components of xyz_{12} and xyz_{13} are updated locally with the RF measurements while the components xyz_{32} are updated due to the CI algorithm.

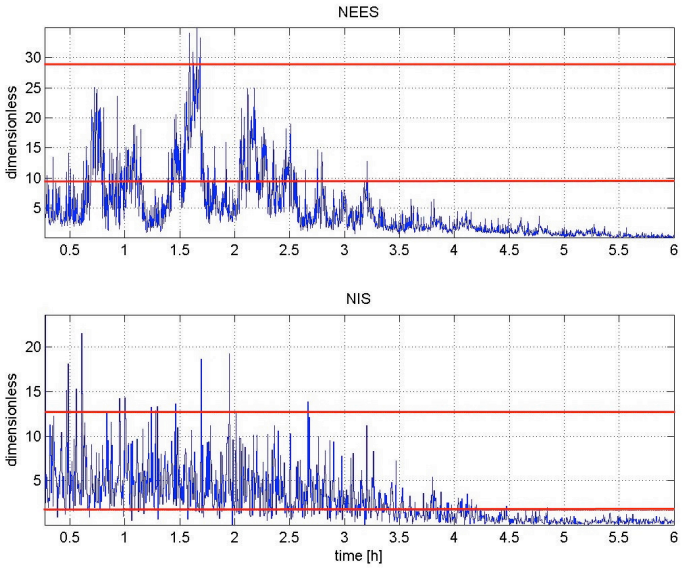

Figure 7-29: **Experiment 3** - NEES-NIS plots for the translational state vector. Top plot: NEES test, Bottom plot: NIS test. The plots are computed for all state vector components, $n_x = 18$ of s/c 1 and $n_y = 18$ corresponding to limits $[9.40; 28.9]$, on each s/c.

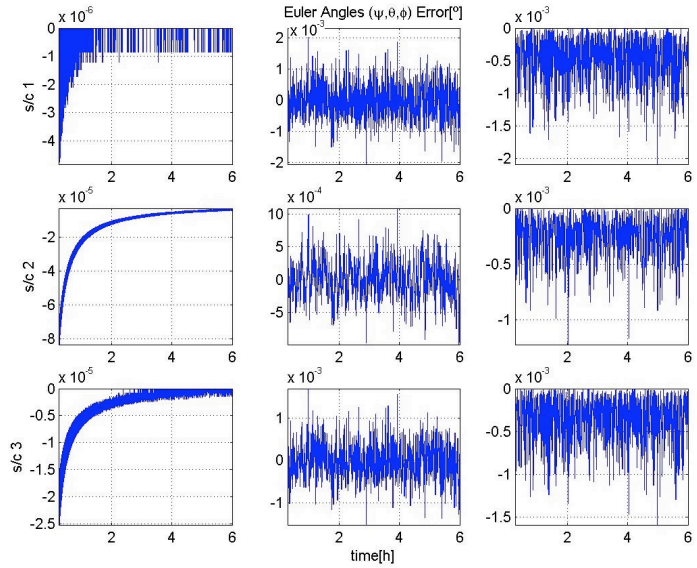

Figure 7-30: **Experiment 3** - Results for the estimation of Euler angles obtained in s/c 1 for a 6 hours orbit segment. The components of s/c 1 are updated locally with the ST measurements while the componentsof s/c 2 and 3 are updated due to the CI algorithm.

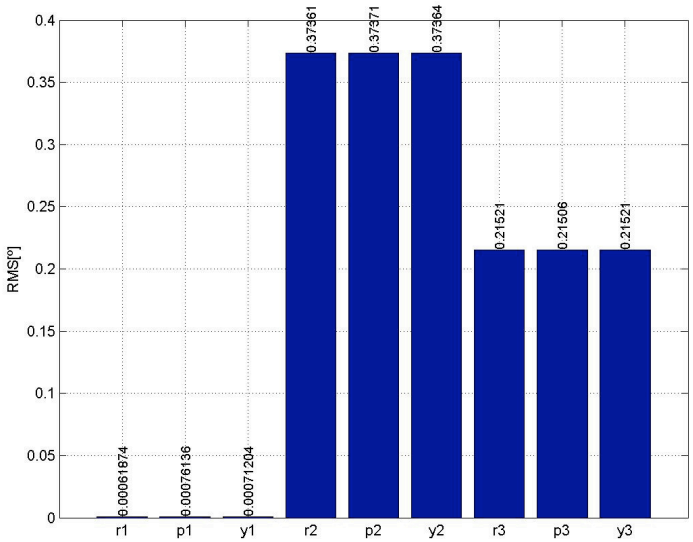

Figure 7-31: **Experiment 3** - Results for the RMS of Euler angles obtained in s/c 1 for a 6 hours orbit segment. The ψ(r1), θ(p1), ϕ(y1) for s/c 1 are updated locally with the ST measurements while for s/c 2 and 3, the angular velocity components are updated due to the CI algorithm.

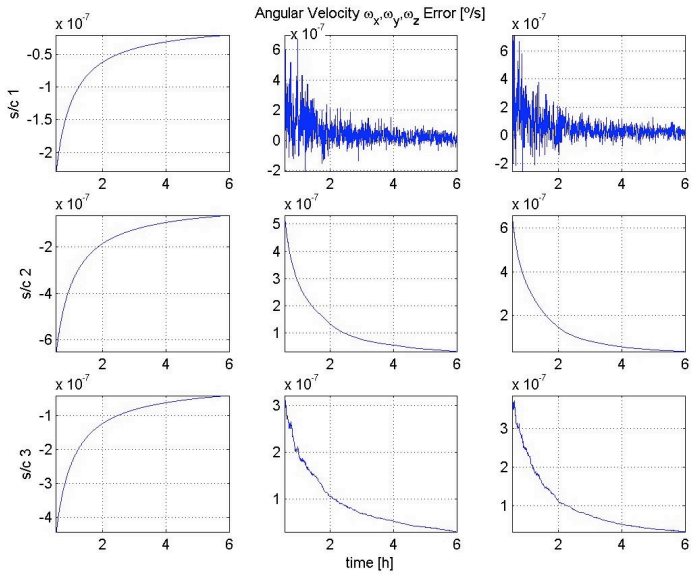

Figure 7-32: **Experiment 3** - Results for the estimation error of the angular velocity for s/c 1 for a 6 hours segment. The components of xyz_{12} and xyz_{13} are updated locally with the RF measurements while the components xyz_{32} are updated due to the CI algorithm.

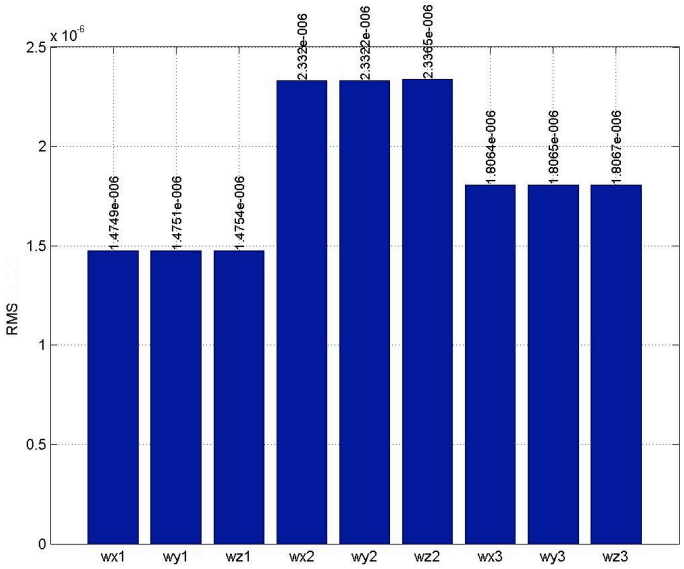

Figure 7-33: **Experiment 3** - Results for the RMS of angular velocity $[rad/s]$ obtained in s/c 1 for a 6 hours orbit segment. The components of xyz_{12} and xyz_{13} are updated locally with the RF measurements while the components xyz_{32} are updated due to the CI algorithm.

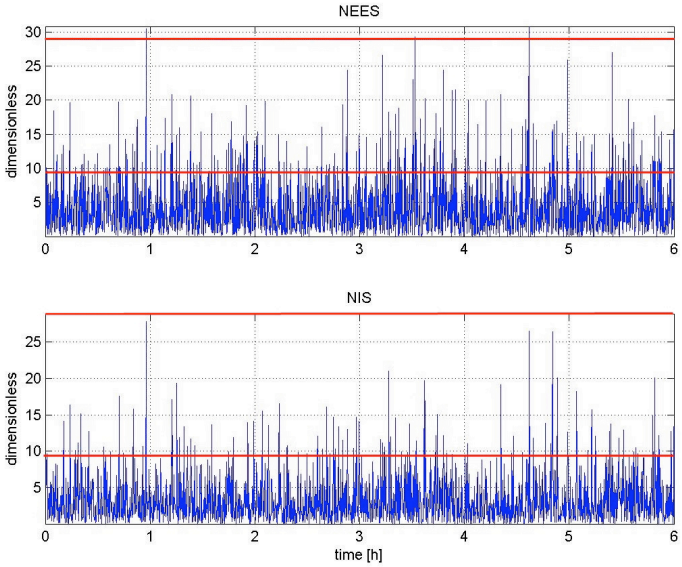

Figure 7-34: **Experiment 3** - NEES-NIS plots for the rotational state vector. Top plot: NEES test, Bottom plot: NIS test.The plots are computed for all state vector components, $n_x = 18$ and $n_y = 18$ corresponding to limits $[9.40; 28.9]$, on each s/c.

- High order gravitational anomalies
- Sun and Moon third body perturbations
- J2 perturbations
- Micrometeorids

As we can observe from the results presented in Figs. 7-35 and 7-40, there is no clear degradation of the algorithms performance compared with the results from experiment 3, with the algorithms reacting well to the perturbations.

7.4.5 Experiment 5 - implementation of guidance and control

The goal of this experiment is to bring the formation to a distance among s/c of approximately 250 meters, more exactly 208 metre for s/c 2 and 216 metre for s/c 3 relative to s/c 1 as explained in last chapter. This is achieved with an maximum error of $0.06[m]$ as plotted in Figures 7-41 and 7-42.

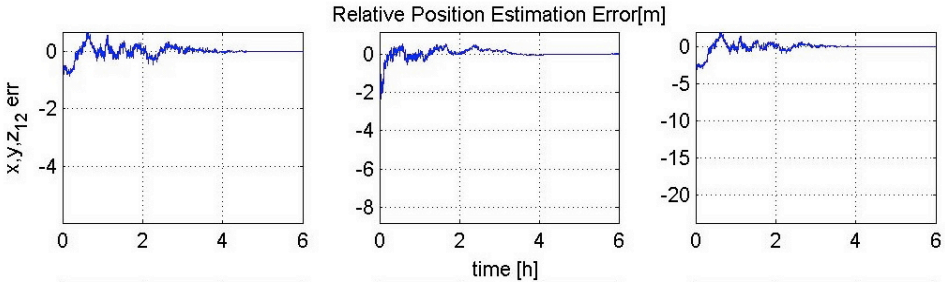

Figure 7-35: **Experiment 4** - Results for the estimation error of relative position obtained in s/c 1 for 6 hours orbit segment. The xyz_{12} and xyz_{13}components are updated locally with the RF measurements while the xyz_{32} components are updated due to the CI algorithm. Perturbations are added to the simulator.

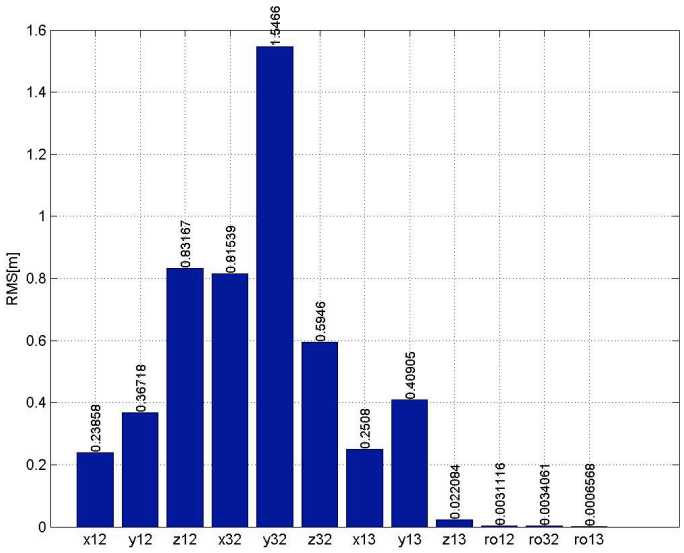

Figure 7-36: **Experiment 4** - Results for the RMS of relative position obtained in s/c 1 for a 6 hours orbit segment. The xyz_{12} and xyz_{13}components are updated locally with the RF measurements while the xyz_{32} components are updated due to the CI algorithm which combines the local state vector estimates with the predecessor state vector estimates. Perturbations are added to the simulator.

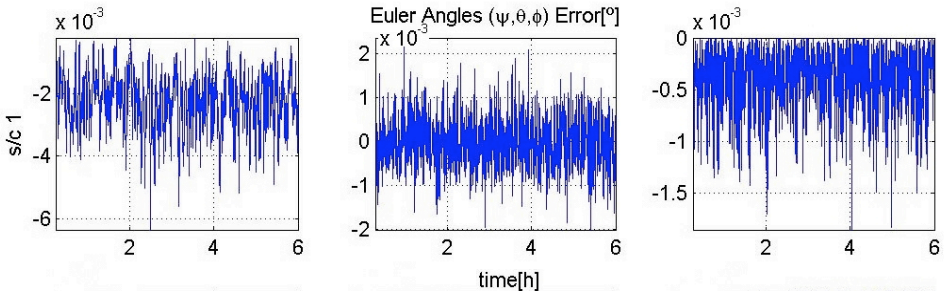

Figure 7-37: **Experiment 4** - Results for the estimation error of Euler angles obtained in s/c 1 for a 6 hours orbit segment. The components of s/c 1 are updated locally with the ST measurements while the components of s/c 2 and 3 are updated with CI algorithm which combines the local state vector estimates with the predecessor state vector estimates. Perturbations are added to the simulator.

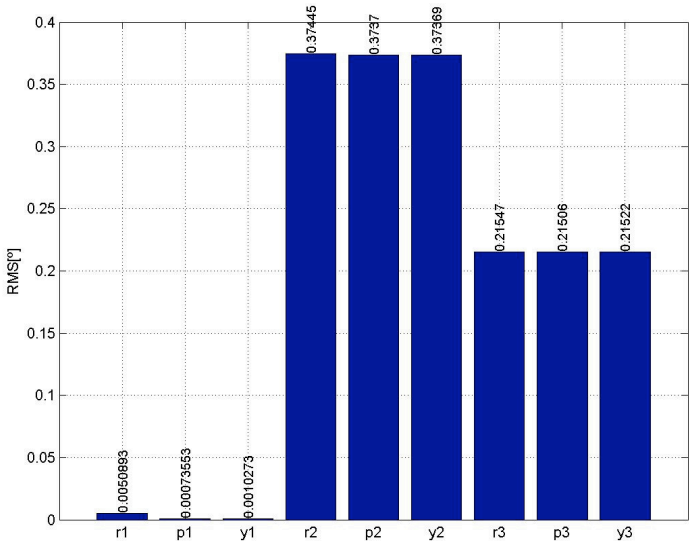

Figure 7-38: **Experiment 4** - Results for the RMS of Euler angles obtained in s/c 1 for a 6 hours orbit segment. The ψ(r1), θ(p1), ϕ(y1) for s/c 1 are updated locally with the ST measurements while the components of s/c 2 and 3 are updated with CI. Perturbations are added to the simulator.

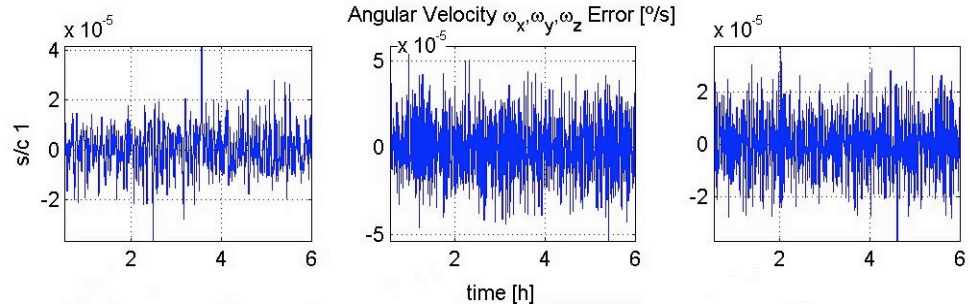

Figure 7-39: **Experiment 4** - Results for the estimation error of angular velocity for s/c 1 for a 6 hours segment. The components of s/c 1 are updated locally with the ST measurements while the components of s/c 2 and 3 are updated due to the CI algorithm. Perturbations are added to the simulator.

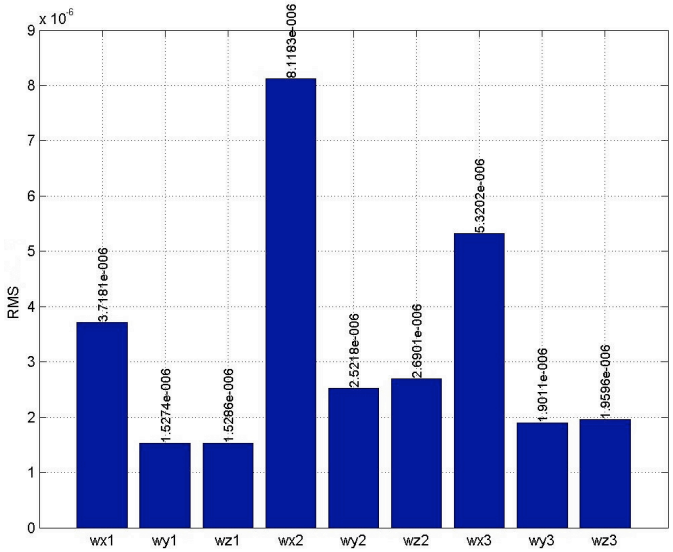

Figure 7-40: **Experiment 4** - Results for the RMS of angular velocity obtained in s/c 1 for a 6 hours orbit segment. The ω_{xyz} for s/c 1 are updated locally with the ST measurements while for s/c 2 and 3, the angular velocity components are updated due to the CI algorithm. Perturbations are added to the simulator.

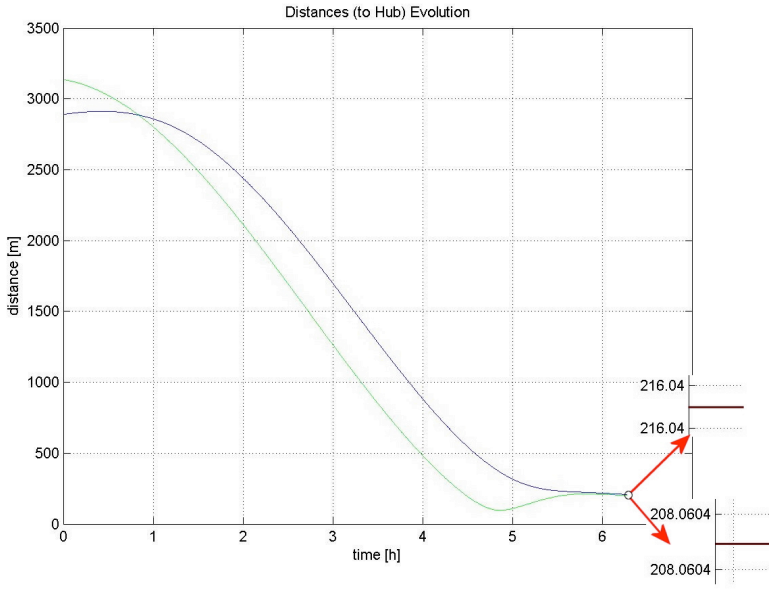

Figure 7-41: **Experiment 5** - The evolution of the distances between the s/c 1 and s/c 2 (green solid lie), and between s/c 1 and s/c 3 (blue solid line).

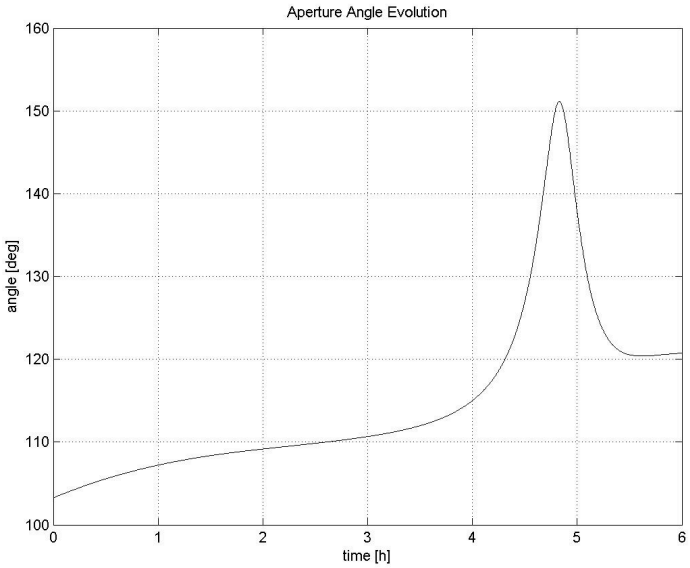

Figure 7-42: **Experiment 5** - The evolution of the aperture angle of the triangle formation.

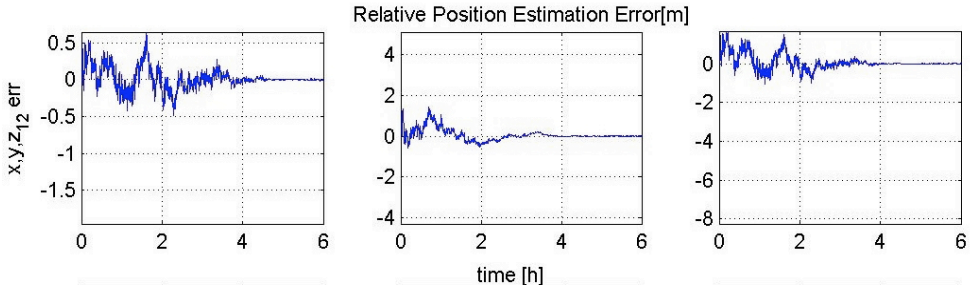

Figure 7-43: **Experiment 6** - Results for the estimation error of relative position obtained in s/c 1 for GNC 6 hours orbit segment. The xyz_{12} and xyz_{13}components are updated locally with the RF measurements while the xyz_{32} components are updated due to the CI algorithm. Perturbations are added to the simulator.

7.4.6 Experiment 6 - implementation of guidance, navigation and control

This experiment follows the same conditions of experiment 5 but now the navigation algorithms are included in the control closed-loop. Results of the relative position estimation error are displayed in Fig. 7-43. This is a more realistic implementation of the full GNC loop, since the full formation state vector components are estimated locally by the navigation algorithms, using information from the previous s/c, according to a peer-to-peer *information flow* graph, as discussed in Chapter 3 and 4, and the control algorithm is using the estimates from the navigation algorithms. The goal is to bring the formation to a distance among s/c of 250 meters, more exactly 208 metre for s/c 2 and 216 metre for s/c 3 relative to s/c 1 as explained in last chapter. This is achieved with an total error of $0.074[m]$, showing that adding the navigation algorithms does not degrade the GNC performance significantly. This is as expected since now the GC algorithms consider the navigation algorithms estimations as real, and in fact the estimated state vector components have an error. Thus, the results are expected to be worse then when the navigation algorithms are not in the closed-loop.

The evolution of the aperture angle of the triangle formation is similar to the one obtained in experiment 5 (shown in Fig. 7-42).

7.5 Results for Baseline Control Mode

All the results concerning relative position and translational relative velocity shown in this section concern the estimated full state of the formation at s/c 1 and are represented in the LVLH frame. All the results concerning absolute attitude and absolute angular velocity shown in this section concern the estimated state at s/c 1 and are represented in the inertial IPQ frame.

The process and measurement noise covariance matrices are tuned in order fulfill the NEES-NIS tests respectively, as performed for FAC mode trajectory, in experiment 2.

7.5.1 Experiment 7 - guidance implementation

Results are plotted in Figs. 7-44 to 7-51. The goal is to maintain the translational variables and to control the attitude of the s/c, therefore, for the translational part of the state vector there is only guidance, and no control and as for the rotational part of the state vector there is a PID control. The relative position is given by means of feed-forward profiles on 2^{nd} and 3^{rd} s/c, since the hub is the s/c 1. Figure 7-44 shows the triangle that is made by the 3-FF s/c. Figure 7-45 shows the evolution of the distance of s/c 2 and 3 *w.r.t.* s/c 1 and in Fig. 7-46 the aperture angle for both s/c is depicted. In Fig. 7-47 is plotted the BCM orbit and the s/c attitude. At the beginning of the BCM orbit (cross), it is clear that the body frames of the s/c are not coincident and this is due to the initial s/c 1 attitude being different from the inertial attitude, *i.e.,* null Euler angles. However, soon the control corrects this situation and it can be seen that the following frames are coincident. In Figs. 7-48 and 7-49 are plotted the evolutions the of the 3-s/c in the BCM orbit in the local reference frame. We can observe that the beginning (cross) and the end (circle) are very close, as expected since they are guided by the GC algorithms. This is also possible to be seen from Figs. 7-45 and 7-46, but in terms of the distance to s/c 1 and aperture angle evolution of the s/c, respectively. In Figs. 7-50 and 7-51 are highlighted the LFM, *i.e.,* the end of the BCM mode, and the rest of the orbit, until when the 3-FF s/c reach the beginning of the ICM orbit segment. During this mode the s/c are not guided but left to natural motion. Studies carried out by DEIMOS SA and reported in [68] have shown the orbit parameters range that the FF s/c have to take in the beginning of the LFM such that the dynamic itself will lead the FF s/c close to each other in the end of the LFM.

7.5.2 Experiment 8 - integrated guidance and navigation as observer

GC algorithms previously described are integrated with the RF-based relative position navigation to assess its performance during the BCM mode. It shall be highlighted that since in the translational motion there is only guidance, and no control, the navigation estimates is not feedback into de loop, is just an observer. The following environment perturbations have been taken into account:

- Solar Radiation Pressure
- High order gravitational anomalies
- Sun and Moon third body perturbations

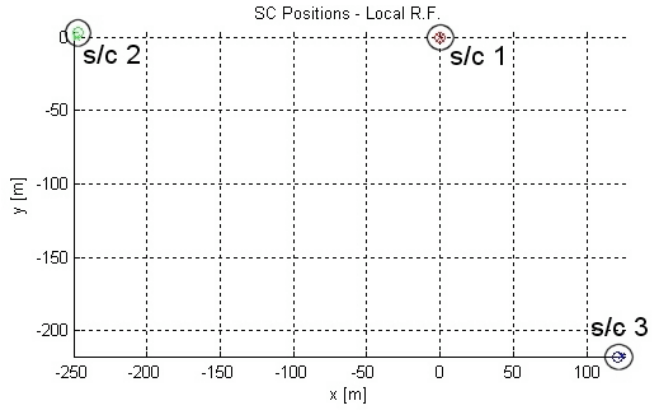

Figure 7-44: **Experiment 7 -** Evolution of the s/c position, which remains a triangle, in the local reference frame, for 2 hour BCM orbit. Reprinted from [9].

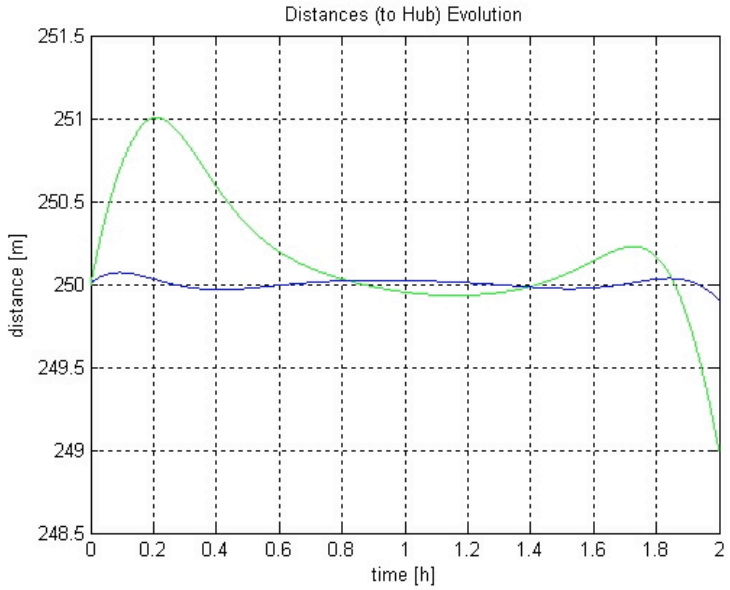

Figure 7-45: **Experiment 7** - The evolution of the distances between the s/c 1 and s/c 2 (green solid line), and between s/c 1 and s/c 3 (blue solid line) for BCM 2 hour orbit. Reprinted from [9].

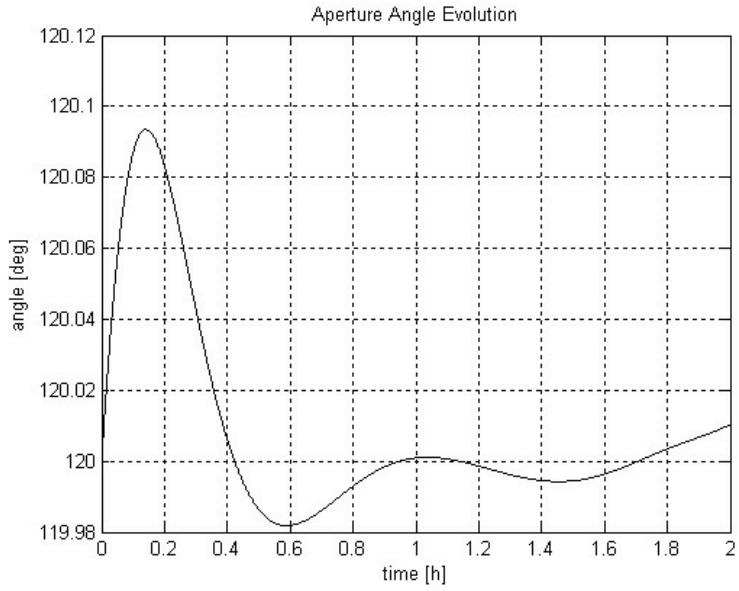

Figure 7-46: **Experiment 7** - The evolution of the aperture angle of the triangle formation in BCM mode. Reprinted from [9].

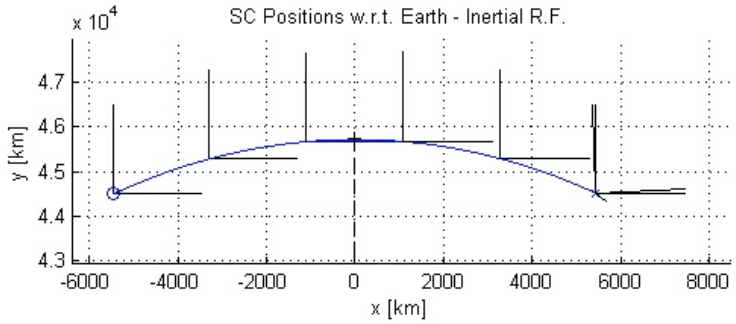

Figure 7-47: **Experiment 7** - View of the BCM 2 hours orbit of the 3 s/c absolute positions *w.r.t.* inertial reference frame. Reprinted from [9].

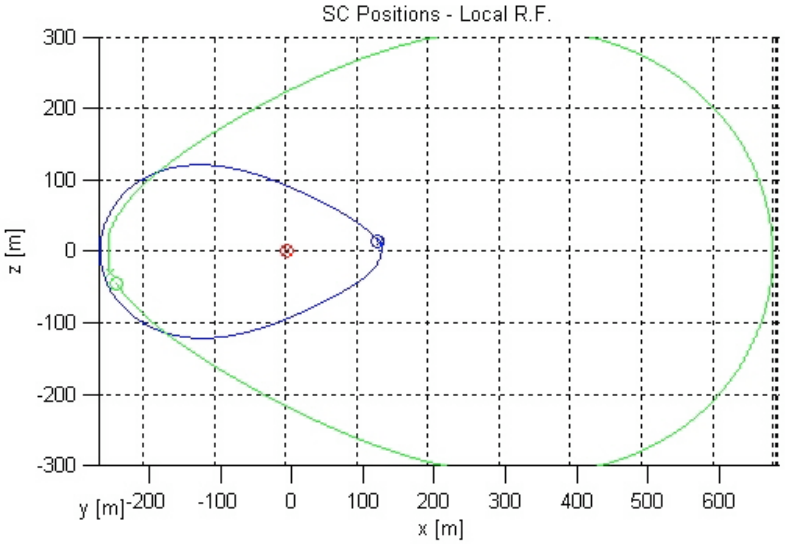

Figure 7-48: **Experiment 7** - Projection in the $z-x$ plane of the 3 s/c relative trajectories in IPQ with the GC applied to the formation. s/c 1 (red solid line) , s/c 2 (green solid line), s/c 3 (blue solid line). Reprinted from [9].

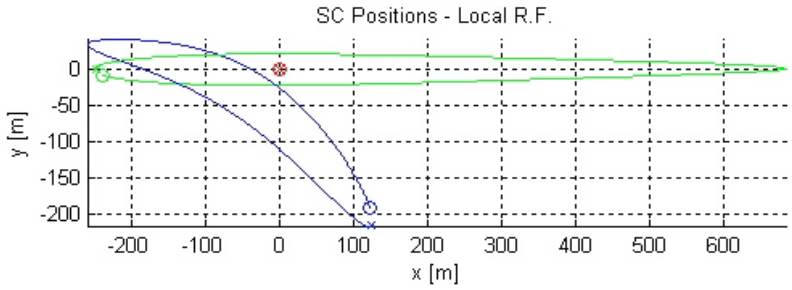

Figure 7-49: **Experiment 7** - Projection in the $x-y$ plane of the 3 s/c relative trajectories in IPQ with the GC applied to the formation. s/c 1 (red solid line) , s/c 2 (green solid line), s/c 3 (blue solid line). Reprinted from [9].

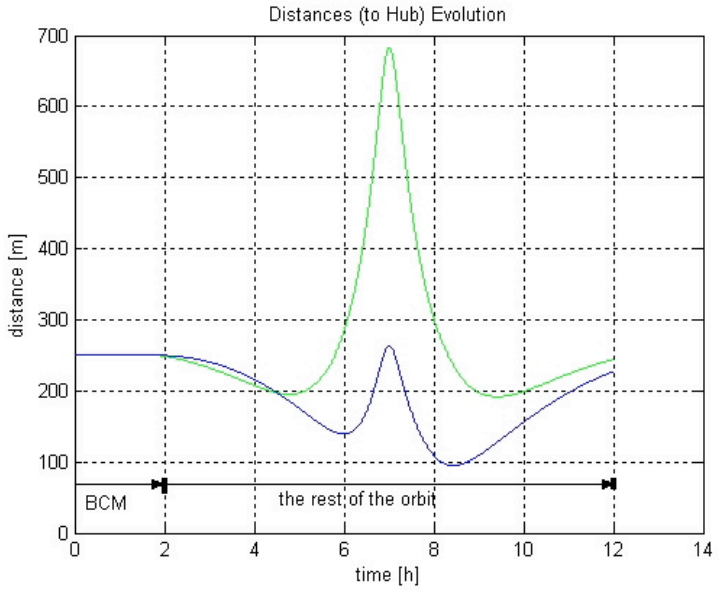

Figure 7-50: **Experiment 7** - The evolution of the distances between the s/c 1 and the other between the s/c 1 and s/c 2 (green solid line), and between s/c 1 and s/c 3 (blue solid line) for the GC applied to the formation in BCM and the rest of the orbit in LFM. Reprinted from [9].

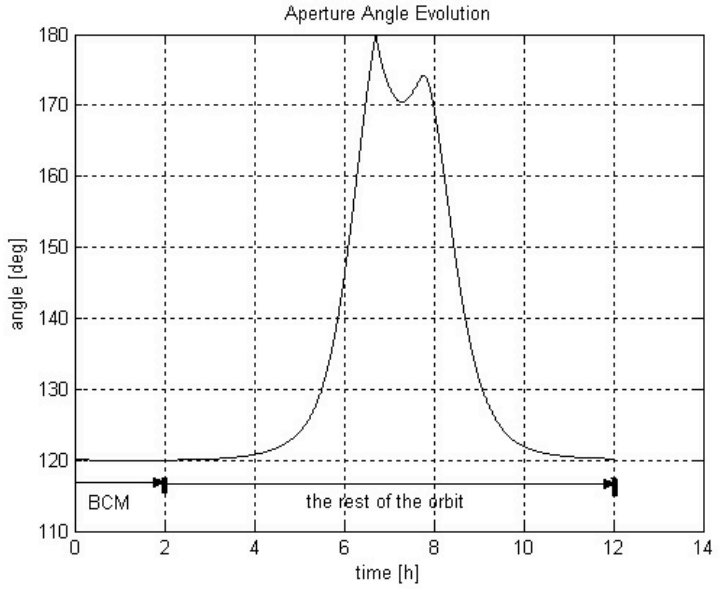

Figure 7-51: **Experiment 7** - The evolution of the aperture angle of the triangle formation in BCM and in the rest of the orbit in LFM. Reprinted from [9].

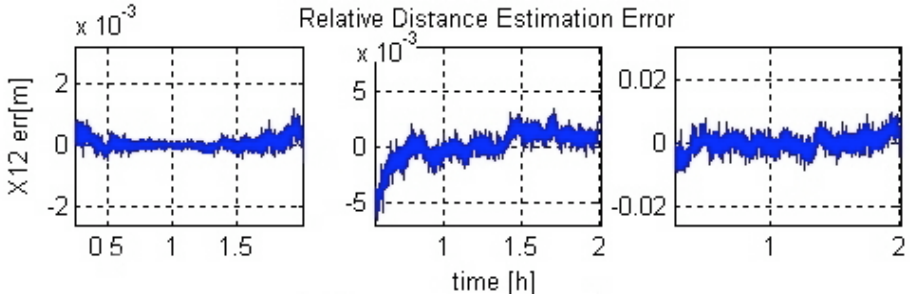

Figure 7-52: **Experiment 8 -** Results (detail plot) for the estimation error of relative position obtained in s/c 1 for a 2 hours orbit segment. The components of xyz_{12} and xyz_{13} are updated locally with the RF measurements while the components xyz_{32} are updated due to the CI algorithm. Reprinted from [9].

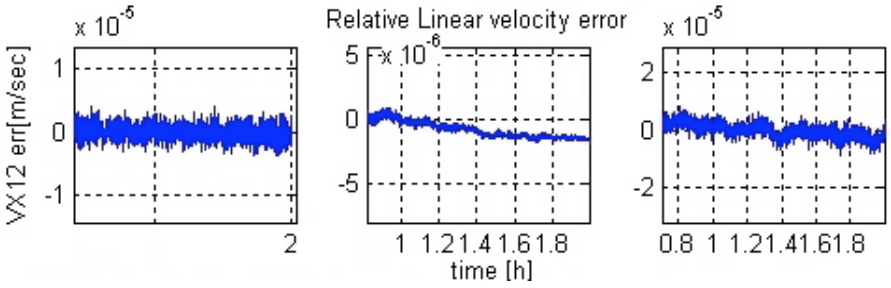

Figure 7-53: **Experiment 8 -** Results (detail plot) for the estimation error of relative translational velocity obtained in s/c 1 for a 2 hours orbit segment. The components of xyz_{12} and xyz_{13} are updated locally with the RF measurements while the components of s/c 2 and 3 are just propagated through the equations of motion. Reprinted from [9].

Results from the translational state vector are shown in Figs. 7-52 to 7-55, and as for the FAC mode segment of the orbit, we can observe a successful implementation of the translational navigation algorithm, since the specification of $0.01[m]$ and $0.01[m/s]$ for relative position and translational relative velocity respectively, described in Table 6.1 and shown in Figs. 7-52 and 7-53, were met in the end of the BCM mode. For the rotational part of the state vector the results show a successful implementation of the rotational navigation algorithm as well. The specification of 1 [arcsec] ($= 2.7778 \cdot 10^{-4}[^o]$) for the inertial Euler angles and 0.1 [arcsec /s] ($= 4.848 \cdot 10^{-7}[rad/s]$) for the angular velocity, respectively were met, and shown in Figs. 7-54 and 7-55.

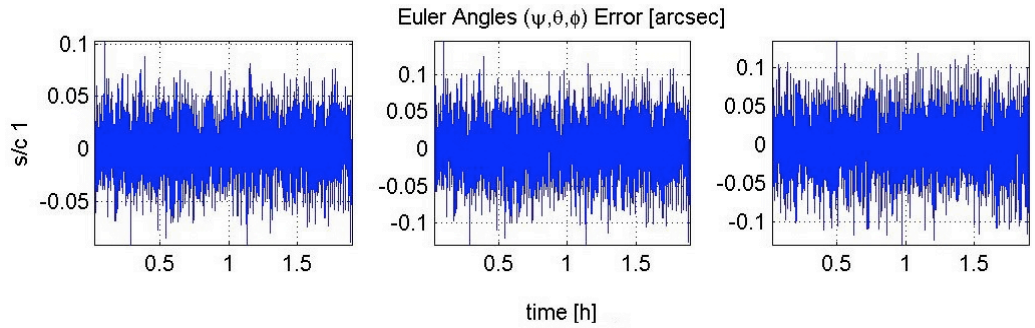

Figure 7-54: **Experiment 8** - Results (detail plot) for the estimation of Euler angles obtained in s/c 1 for a 2 hours orbit segment. The components of s/c 1 are updated locally with the ST measurements while the components of s/c 2 and 3 are just propagated through the equations of motion.

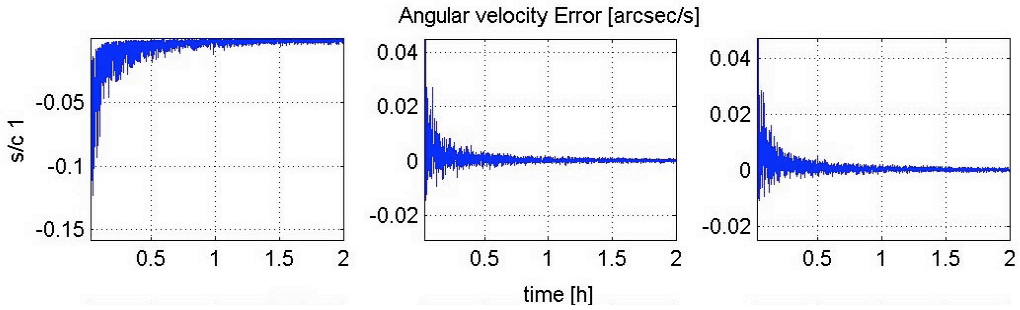

Figure 7-55: **Experiment 8** - Results (detail plot) for the estimation of angular velocity obtained in s/c 1 for a 2 hours orbit segment. The components of s/c 1 are updated locally with the ST measurements while the components of s/c 2 and 3 are just propagated through the equations of motion.

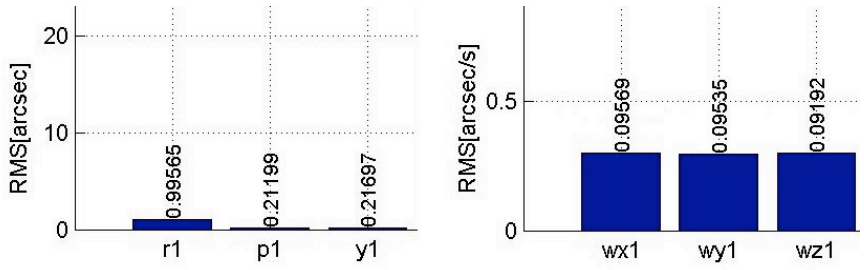

Figure 7-56: **Experiment 9** - Results for the RMS of Euler angles (left plot) and angular velocity (right plot) obtained in s/c 1 for a 6 hours orbit segment. The ψ(r1), θ(p1), ϕ(y1) and ω_{xyz} for s/c 1 are updated locally with the ST measurements while for s/c 2 and 3, the quaternions and angular velocity components are just propagated through the equations of motion.

7.5.3 Experiment 9 - Monte-Carlo results for guidance and control with navigation as observer

The average results of the 10 runs are computed at each step and plotted in Fig. 7-56. The process and measurement noise covariance matrices were obtained in order to fulfill the NEES and NIS limits, as performed for FAC and BCM mode trajectory. The rotational algorithm is in conditions of experiment 2, *i.e.,* the EKF fusion algorithm is not used to update the state vector local non-observable components. This is so, because the translational algorithm just need the estimates of the rotational state vector related of its one s/c and also the feasibility of the EKF Fusion algorithm was already tested and discussed in experiment 3. The Figure show the RMS obtained for the 10 Monte-Carlo runs, and the values are according to expected, *i.e.,* worse then experiment 8. The variability introduced in the Monte-Carlo simulations introduce unexpected errors in the navigation algorithms resulting in worse performance than the nominal performance studied so far. The relative position and translational relative velocity RMS are from one order worse in Monte-Carlo run, as well as the Euler angles and the angular velocity RMS are in Monte-Carlo experiment two order higher than for nominal performance.

7.6 Summary

The results of two orbits trajectory and 12 experiments (for FAC and BCM segments of the orbit) are presented for a formation of 3-s/c in a GTO mission. The results are divided in two parts, which correspond to the initial phase deployment mode, the FAC mode, and to the science experiment, the BCM mode. For the FAC mode the results presented are always in closed-loop guidance and control but split in two parts:

- without the estimator in the loop, working just as an observer, without feeding its estimates to the CG algorithm;
- With the estimator in the loop.

For the BCM mode only results of the estimator in the loop are presented. There is no control applied to the translational state vector estimates, but just to the rotational variables. Results with perturbations disturbing the system are also presented to asses the navigation algorithm robustness. Also, the calculation of the process and measurement noise covariance matrices from the sensors and noise characteristics and tuned to fulfill the NEES and NIS consistency tests, is addressed. Finally, results for Monte-Carlo runs, are presented for the BCM segment of the orbit.

The main relevant results are:

- a successful implementation of the proposed 6DoF navigation algorithms, *i.e.,* the specifications were met;
- the importance and feasibility of using information from another vehicle estimates, represented by links in a peer-to-peer *information flow* graph, to compute the non-observable state vector components that are not estimated locally;
- the importance of using the CI method to fuse correlated entities and to improve the estimation accuracy for locally non-observable state components;
- the accuracy of the achieved full state vector estimates and the navigation optimality by comparison with the PCRLB curve;
- the capability of the navigation algorithms to overcome the initial error, disturbances and to track control changes;
- the navigation's robustness to uncertainty of the parameters, presented in the Monte-Carlo results;
- the importance of tuning the process and measurement noise covariance matrices using the NEES-NIS consistency tests and the PCRLB curve.

Chapter 8

CONCLUSIONS AND FUTURE WORK

In this thesis, we proposed an EKF+CI algorithm to compute a 6DoF full formation state vector estimate, using low bandwidth communication and a decentralized solution. A graph framework is proposed, to analyse the feasibility of decentralized 6DoF formation state vector estimation on each vehicle. This also enables to assessment of the robustness of the navigation algorithm to communications loss or the occlusion of one vehicle *w.r.t.* another. The measurements available on each vehicle are modelled by links in a *measurement* graph and the state vector estimates transmitted between vehicles are modelled by links in the *information flow* graph. Thus, using only local measurements and restricting the number of communication links to a peer-to-peer topology, the proposed navigation filter estimates the 6DoF full formation state vector locally, at each vehicle.

In order to estimate the full formation state vector at each vehicle, the navigation algorithm has to cope with different sources of measurements: sensors and state "measurements" (*i.e.,* the state estimates) from other vehicles. The proposed filter is based on a combination of an EKF and a CI algorithm at each element of the formation, and on inter-vehicle measurements obtained from range sensors to compute the translational part of the state vector and on absolute sensors located on each vehicle to estimate the rotational state part of the vector. Each vehicle communicates its most recent state estimate to the next vehicle in the communication network. **For the translational components,** the filtering part of the state estimation process is performed by either an EKF, when local measurements are used, or by a CI algorithm, when state "measurements" come from another vehicle. **For the rotational components** the filtering part of the state estimation process is performed by an EKF at each vehicle, based on absolute attitude measurements obtained from an inertial sensor, and at alternative sampling times by

another EKF (EKF fusion), when observations (*i.e.,* the state estimates) come from another vehicle. The prediction part of the state estimation process in both filters is performed by an EKF.

The EKF observability and consistency is studied and, based on the consistency tests, a process of tuning the process and observation noise matrices is proposed. Study of the PCRLB to evaluate the EKF suboptimal solution, as well as the EKF compensation techniques, is also reported.

The proposed navigation algorithm was applied to a 3 FF s/c mission. This estimates the 6DoF formation state (relative position, translational velocity, attitude and angular velocity) in each of the s/c, and was tested for the simulated FAC and BCM modes of the GTO mission. In order to reduce the influence of the RF accuracy degradation with large distances on the state estimates, the GC algorithm is used with the true state as input so as to reduce the inter-s/c distances. For FAC, an optimal guidance and control algorithm including collision avoidance was used [8]. For BCM, an optimized low thruster profile, and relative position plus attitude PID control were used. Sensor models for the ST and RF sensors were included in the FF-FES simulator for this purpose. The results show:

- a successful implementation of the 6DoF navigation filter, providing an integrated estimate of the relative translational and rotational state components, which meets the GNC specifications for accuracy (according to the GTO mission specifications);
- the importance and feasibility of using information from other's vehicles estimates, represented by links in the peer-to-peer *information flow* graph to compute the non-observable state vector components that are not estimated locally;
- very good accuracy achieved by the relative position and relative translational velocity estimator, as the result of eliminating biases and employing systematic tuning of noise covariance matrices based on consistency tests. The estimation error covariance matrix of the translational components was reduced close to the lower bounds of uncertainty;
- that the GNC requirements (according to the GTO mission specifications) are achieved for BCM using an optimized low thruster profile and relative position plus attitude PID control;
- the ability to simulate successfully (according to the GTO mission specifications) the full closed-loop GNC algorithm, *i.e.,* including the estimator in the loop;
- the ability of the proposed full-order decentralized navigation algorithm to track the formation state, starting from an initial value different from the true state value, and under several unmodelled perturbations;
- the efficiency of the proposed algorithm by comparing its performance to the PCRLB;

- the navigation algorithms robustness for the Monte-Carlo runs.

The results show that it is possible to have an integrated estimation of the relative translational and rotational components of the full-order state vector of a formation of vehicles in a decentralized framework, and with information flowing among its elements using a minimum-communication topology.

8.1 Future Work

Despite the sucess of this work, there are still areas where improvements can be made. Suggestions for the extension of this study are:

Improving accuracy by using multi-modal distributions - to improve the decentralized estimator accuracy by using methods that deal with multimodal probabilistic distribution associated with the estimates, instead of the unimodal Gaussian distribution, *e.g.*, Monte-Carlo localization methods for 6DoF [73], [18]. The Particle filters, or Sequential Monte-Carlo Methods, are estimation tools that approximate the representation of the uncertainty instead of approximating the nonlinearity of the system representation which is performed by the EKF. Particle filters can deal effectively with probability distributions that are neither Gaussian nor unimodal, which is one of the drawbacks of the EKF. Particle filters are reported in the work of Thrun [46], [18], [74], which proposes [75] the Unscented Transform (UT), a more attractive method to use than Particle Filters, since it uses a much smaller number of sigma points to approximate the probability distribution. An optimal recursive estimator of the UT is called the Unscented Kalman Filter (UKF) that uses the UT to implement the nonlinear transformations for both nonlinear process and measurement models without going through the linearization steps required by the EKF. This method has already been applied to underwater vehicles in [76] where the goal is to compute the relative position between a moving observer and a stationary object, by fusing measurements from monocular vision of a single feature with measurements from inertial rate sensors. The UT is used to propagate the nonlinear state dynamics, avoiding the error introduced by the linearizations. Also, the UT has been used in a FF scenario by [77] for relative navigation of multiple vehicles, where the goal is to compute the relative orbital navigation based on Carrier-phase Differential GPS in a centralized architecture, taking special care with the error effects in the semi-major axis. Nevertheless, as concluded by [66], care should be taken because particle filters are computationally expensive and therefore should be used only for difficult nonlinear/non-Gaussian problems, where conventional methods fail. Thus, for future work, to assess navigation algorithms for decentralized 6DoF full formation state estimation with UT instead EKF, would also be interesting to address;

Extend Covariance Intersection method to attitude variable - to extend or propose an algorithm able to estimate the attitude variable which represents the relationship between vehicles obtained directly from range measurements, as for example carrier-phase measurements, and not from absolute measurements such as those from the ST sensor;

Extend Navigation to all GTO modes - the implementation of the GNC algorithms for all the modes through a mode switching mechanism using as threshold the error covariance matrix to change the modes, in order to provide a systematic approach to mission level navigation, including better failure detection and handling;

Include perturbation model - to include in the formation dynamics model of the navigation algorithm the non-linear perturbations and, if possible, add velocity measurements (*e.g.,* RF doppler) to provide more information, capable of correcting errors inherent to the dynamic propagation;

Multiple sensor measurements - to include more sensor information, not only to study the sensor fusion but also to improve, if possible, the accuracy of the estimates. Examples of sensors are the divergent laser sensor, and/or the carrier-phase measurements, depending on the orbit modes;

Sensor comparison - to use the carrier-phase to provide more accurate information compared to the RF measurements, to improve the relative state vector errors, and to compare the impact of sensor measurements in the algorithm performance;

Aysnchronous communications - to study the asynchronous communication graph, including time-delays.

References

[1] P. Fergunson, "Distributed Estimation and Control Technologies for Formation Flying Spacecraft," Masters Thesis, Massachusetts Institute of Technology, Jan. 2003.

[2] S. Julier and J. Uhlmann, "A Non-Divergent Estimation Algorithm in the Presence of Unkown Correlations," in *Proceedings of the 1997 American Control Conference (ACC'97)*, June 1997.

[3] P. Arambel, C. Rago, and R. Mehra, "Covariance Intersection Algorithm for Distributed Spacecraft State Estimation," in *Proceedings of the American Control Conference*, Arlington, VA, June 2001.

[4] A. Mutambara, *Decentralized Estimation Control for Multisensor Systems*. CRC Press, 2001.

[5] B. Kang, F. Hadaegh, and D. Scharf, "Decentralized and Self-Centered Estimation Architecture for Formation Flying of Spacecraft," in *16th International Symposium on Space Flight Dynamics*, Pasadena, California, EUA, Dec. 2001.

[6] H. Durrant-Whyte, "Sensor Models and Multi-Sensor Integration," *Int. Jounal Robotics Research*, vol. 7, no. 6, pp. 97–113, 1998.

[7] H. Hu and P. Probert, "Distributed architectures for sensing and control in obstacle avoidance for autonomous vehicles," in *IARP Int. Conf. Multi-Sensor Data Fusion*, 1989.

[8] D. Dumitriu, P. Lima, and B. Udrea, "Optimal Trajectory Planning of Formation Flying Spacecraft," in *16th IFAC World Congress in Prague*, ser. No Fr-M07-TP/3, no. Fr-M07-TP/3, 2005.

[9] P. Lima, A. Caramagno, S. Marques, and L. Peñin, "Coarse Hybrid Relatve Navigation CCN3-4 Final Report," Instituto de Sistemas e Robótica, Instituto Superior Técnico, Tech. Rep. ESA RFQ3-10624/03/NL/LvH/bj, 2006.

[10] M. Tillerson, "Coordination and Control of Multiple Spacecraft Using Convex Optimization Techniques," Master's thesis, Massachusetts Institute of Technology, June 2002.

[11] J. How and M. Tillerson, "Analysis of the Impact of Sensor Noise on Formation Flying Control," in *IEEE American Control Conference*, P. of the American Control Conference, Ed., Arlington, VA, 2001, pp. 3986–3991.

[12] M. Colavita, K. Lau, and M. Shao, "The New Millennium Separated Spacecraft Interferometer," in *In Space Technology and Applications International Forum*, Albuquerque, NM, 1997.

[13] K. Lau, S. Lichten, L. Young, and B. Haines, "An Innovative Deep Space Application of GPS Technology for Formation Flying Spacecraft," in *Conference of AIAA Guidance, Navigation and Control*, July 1996.

[14] J. Leitner, F. Bauer, D. Folta, M. Moreau, R. Carpenter, and J. How, "Distributed Spacecraft Systems Develop New GPS Capabilities," *GPS World*, February 2002.

[15] B. Engberg and R. Twiggs, "The Orion Microsatellite: A Testbed for Command, Control, and Communication for Formation Fleets," in *14th Annual AIAA/USU Conference on Small Satelites*, ser. SSC00-II-4, september 2000.

[16] D. Nardi, M. Riedmiller, C. Sammut, and J. Santos-Victor, *RoboCup 2004: Robot Soccer World Cup VIII*. Lecture Notes in Computer Science, Sprimger, 2005, vol. 3276.

[17] S. Roumeliotis and G. Bekey, "Distributed multi-robot localization," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, October 2002.

[18] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, "A Probabilistic Approach to Collaborative Multi-Robot Localization," *In Special issue of Autonomous Robots on Heterogeneous Multi-Robot Systems*, vol. 8, no. 3, pp. 255–266, 2000.

[19] J. Leonard and H. Durrant-Whyte, "Simultaneous Map Building and Localization for an Autonomous Mobile Robot," in *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, vol. 3, 1991, pp. 1442–1447.

[20] I. Rekleitis, G. Dudek, and E. Milios, "Multi-Robot Cooperative Localization: A Study of Trade-offs Between Efficiency and Accuracy," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, EPFL, Switzerland, October 2002, pp. 2690–2695.

[21] A. Howard, M. Mataric, and G. Sukhatme, "Localization for Mobile Robot Teams Using Maximum Likelihood Estimation," in *In IEEE/RSJ Intrenational Conference on Robotics and Intelligent Systems (IROS)*, EPFL, Switzerland, 2002, pp. 434–459.

[22] A. Howard and M. Mataric, "Localization for Mobile Robot Teams: A Distributed MLE Approach," Extended Abstract for the 6th International Symposium on Experimental Robotics (ISER'02), Sant'Angelo d'Ischia, Italy, pp. 781–795, July 2002.

[23] M. Marco, A. Garulli, A. Giannitrapani, and A. Vicino, "Simultaneous Localization and Map Building for a Team of Cooperating Robots: A Set Membership Approach," *IEEE Transactions of Robotics and Automation*, vol. 19, no. 2, pp. 238–248, december 2003.

[24] C.-W. Park and J. How, "Decentralized Relative Navigation for Formation Flying Spacecraft using Augmented CDGPS," in *Proceedings of the ION*, september 2001.

[25] U. Hanebeck and K. Briechle, "New Results for Stochastic Prediction and Filtering with Unkown Correlations," in *Proceedings of the IEEE Conference on Multisensor Fusion and Integration Systems (MFI'2001)*, Baden-Baden in Germany, 2001, pp. 147–152.

[26] J. Crassidis, E. Lightsey, and F. Markley, "Efficient and Optimal Attitude Determination Using Recursive Global Positioning System Signal Operations," *AIAA Journal of guidance, Control and Dynamics*, vol. 22, no. 2, pp. 193–201, Mar. 1999.

[27] J. Crassidis and F. Markley, "A New Algorithm for Attitude Determination Using Global Positioning System Signals," *AIAA Journal of Guidance Control and Dynamics*, vol. 20, no. 5, pp. 891–896, September-October 1997.

[28] C. Cohen, "Attitude Determination Using GPS," Ph.D. dissertation, Stanford University, Sept. 1992.

[29] G. Lu, "Development of a GPS Multi-Antenna System for Attitude Determination," Ph.D. dissertation, Dept. of Geomatics Engineering in University of Calgary, Canada, 1995.

[30] F. Busse, "Precise Formation-State Estimation in Low Earth Orbit Using Carrier Differential GPS," Ph.D. dissertation, Stanford University, Nov. 2002.

[31] C. Altmayer, S. Martin, and S. Theil, "Autonomous Onboard Orbit and Attitude Control of Geostationary Satellites Using Pseudolites," in *ION GPS-98*, Tagung in Nashville Tenesse EUA, Sept. 1998.

[32] T. Corazzini, A. Robertson, J. Adams, A. Hassibi, and J. How, "GPS Sensing for Spacecraft Formation Flying," in *Institute of Navigation GPS-97*, Kansas City, Sept. 1997.

[33] T. Corazzini and J. How, "Onboard GPS Signal Augmentation for Spacecraft Formation Flying," in *Proc. Of the ION-GPS Conference*, Sept. 1999, pp. 1559–1568.

[34] S. Marques, R. Clements, and P. Lima, "Comparison of Small Satellite Attitude Determination Methods," in *Proceedings of 2000 AIAA Conference on Navigation, Guidance and Control*, Colorado, USA, August 2000.

[35] J. Crassidis and F. Markley, "Predictive Filtering for Nonlinear Systems," *Journal of Guidance, Control and Dynamics*, vol. 20, no. 3, pp. 566–572, May-June 1997.

[36] S. Marques, "Small Satellites Attitude Determination Methods," Masters Thesis, Instituto Superior Técnico from Universidade Técnica de Lisboa, Lisbon, 2001.

[37] ——, "Small Satellites Attitude Determination Using a Predictive Algorithm for Attitude Stabilization and Spin Control," in *Mediterranean Conference of IEEE in Control and Automation*, Instituto Superior Técnico from Univeversidade Técnica de Lisboa, Lisbon, July 2002.

[38] D. Mortari, "Moon-Sun Sensor," *Journal of Spacecraft and Rockets*, vol. 34, no. 3, pp. 360–364, May-June 1997.

[39] ——, "ESOQ: A Closed-Form Solution to the Wahba Problem," *Journal of the Astronautical Sciences*, vol. 45, no. 2, pp. 195–204, 1997.

[40] E. Gill and H. Runge, "Tight Formation Flying for an Along-Track SAR Interferometer," in *IAC-03-A.3.03*, Bremen, Germany, Sep. 29 - Oct. 3 2003.

[41] E. Gill, M. Steckling, and P. Butz, "Gemini: A Milestone Towards Autonomous Formation Flying," *ESA Workshop on On-Board Autonomy*, vol. WPP-191, pp. 415–419, Oct. 2001.

[42] M. Aung, G. Purcell, J. Tien, L.Young, L. Amaro, J. Srinvasan, M. Ciminera, and Y. Chong, "Autonomous Formation-Flying Sensor for the Starlight Mission," IPN progress report, Tech. Rep., Feb. 2003.

[43] G. Purcell, D. Kuang, S. Lichten, S. Wu, and L. Young, "Autonomous Formation Flying (AFF) Sensor Technology Development," in *In Proceedings of the 21st Annual AAS Guidance and Control Conference*, Breckenridge CO, Feb. 1998.

[44] J. Junkins, D. Hughes, K. Wazni, and V. Pariyapong, "Vision-Based Navigation for Rendezvous Docking and Proximity Operations," in *22nd Annual AAS Guidance and Control Conference*, Breckenrige, CO, February 1999.

[45] R. Alonso, J.-Y. Du, D. Hughes, and J. Junkins, "Relative Navigation for Formation Flying of Spacecraft," *Proceedings of the Flight Mechanics Symposium*, pp. 115–129, June 2001.

[46] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, September 2005.

[47] J. Zhipu and R. Murray, "Stability and Performance Analysis with Double-Graph Model of Vehicle Formations," in *Proceedings of the 2003 American Control Conference*, vol. 3, June 2003, pp. 2223–2228.

[48] C. Mehlen, "RF Sensor," Alcatel Alenia Space, Tech. Rep. 1000089709L, 2006.

[49] S. Julier and J.Uhlmann, *General Decentralized Data Fusion with Covariance Intersection (CI)*, I. D. Hall and J. Llians, Eds. CRC Press, 2001.

[50] J. Uhlmann, S. Julier, M. Csorba, and H. Durrant-White, "A Culminating Advance in the Theory and Practice of Data Fusion, Filtering and Decentralized Estimation," Covariance Intersection Working Group, Tech. Rep., 1997.

[51] M. Sidi, *Spacecraft Dynamics and Control - A Practical Engineering Approach*. Cambridge University Press, 1997.

[52] A. Jazwinski, *Stochastic Processes and Filtering Theory*, ser. Volume 64 of Mathematical in Science and Engineering. Academic Press, Inc. London Ltd., 1970, vol. 64.

[53] J. Wertz, *Spacecraft Attitude Determination and Control*, ser. Astropysics and Space Science Library. Kluwer Academic Publishers, 1978, vol. 73.

[54] J. Uhlmann, "Covariance Consistency Methods for Fault-Tolerant Distributed Data Fusion," *Information Fusion*, vol. 4, pp. 201–215, 2003.

[55] T. Bak, "Spacecraft Attitude Determination - a Magnetometer Approach," Ph.D. dissertation, Aalborg University, Dept. of Control Engineering, Aug 1999.

[56] A. Gelb, *Applied Optimal Estimation*. MIT Press, Cambridge, MA, 1974.

[57] Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation, Theory Algorithms and Software*. John Wiley and Sons, 2001.

[58] K. Reif, S. Günther, E. Yaz, and R. Unbehauen, "Stochastic Stability of Discrete-Time Extended Kalman Filter," *IEEE Transactions on automatic Control*, vol. 44, no. 4, April 1999.

[59] M. Gopal, *Modern Control System Theory*. Wiley Eastern Ltd., 1984.

[60] J. Fitts, "On the Observability of Non-Linear Systems with Applications to Non-Linear Regression Analysis," *Information Sciences*, vol. 4, no. 2, pp. 129–156, April 1972.

[61] H. Nijmeijer, "Observability of Autonomous Discrete Time Non-Linear Systems: A Geometric Approach," *International Journal Control*, vol. 36, no. 5, pp. 867–874, 1982.

[62] E. Sontag, "On the Observability of Polynomial Systems, I: Finite-Time Problems," *SIAM Journal of Control and Optimization*, vol. 17, no. 1, pp. 139–151, 1979.

[63] H. V. Trees, *Detection, Estimation, and Modulation Theory: Part.* Wiley, New York, 1968.

[64] P. Tichasvsky, C. Muravchik, and A. Nehorai, "Posterior Cramér-Rao Bounds for Discrete Time Nonlinear Filtering," *IEEE Transactions on Signal Processing*, vol. 46, no. 5, May 1998.

[65] N. Bergman, "Recursive Bayesian Estimation: Navigation and Tracking Applications," Ph.D. dissertation, Linköping University, 1999.

[66] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter, Particle Filters for Tracking Applications.* Artech House, 2004.

[67] L. Peñin and J. Araújo, "FFFES (v1.1) Software User's Manual- SUM -FEMDIS-DME-SUM," DEIMOS Engenharia, Tech. Rep., 2004.

[68] A. Caramagno, J. Bastante, and L. Peñin, "Mission Analysis and Scenario Characterization of a Formation Flying Experiment in GTO," DEIMOS Engenharia Lda, Tech. Rep., 2004.

[69] J. Araujo and L. Peñin, "RF Measurements Generation," DEIMOS Engenharia FEMDIS-DME-MEM-002, Tech. Rep., 2004.

[70] G. Inalhan, M. Tillerson, and J. How, "Relative Dynamics and Control and Spacecraft Formations in Eccentric Orbits," *Journal of Guidance, Control and Dynamics*, vol. 25, no. 1, pp. 48–59, Jan-Feb 2005.

[71] P. Lima, D. Dumitriu, and S. Marques, "Candidate Formation Algorithms Simulation Results and Performance Assessment," Instituto Sistemas e Robótica, Instituto Superior Técnico, Lisbon, Tech. Rep. ESA RFQ3-10624/03/NL/LvH/bj, 2004.

[72] D. Dumitriu, S. Marques, P. Lima, J. Bastante, J. Araújo, L. Peñin, A. Caramagno, and B. Udrea, "Optimal Guidance and Decentralised State Estimation Applied to a Formation Flying Demonstration Mission in GTO," *IET Control Theory Appl.*, vol. 1, no. 2, pp. 532–544, 2007.

[73] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle Filters for Positioning, Navigation and Tracking," *IEEE Transactions on Signal Processing, Special issue on Monte Carlo methods for statistical signal processing*, vol. 50, no. 2, pp. 425–437, 2002.

[74] D. Fox, S. Thrun, W. Burgard, and F. Dellaert, *Particle Filters for Mobile Robot Localization*. In A. Docet and N. De Freitas and N. Gordon Editors Sequential Monte Carlo in Practice, 2000.

[75] S. Julier and J. Uhlmann, "Unscented Filtering and Nonlinear Estimation," *In Proceedings of the IEEE*, vol. 92, no. 3, 2004.

[76] A. Huster, "Relative Position Sensing by Fusing Monocular Vision and Inertial," Ph.D. dissertation, Stanford University, July 2004.

[77] M. Mitchell, "CDGPS-Based Relative Navigation for Multiple Spacecraft," Master's thesis, Massachusetts Institute of Technology, May 2004.

[78] R. Diestel, *Graph Theory*, ser. Gradute Texs in Mathematics. Springer-Verlag, 2000, vol. 173.

[79] J. Fax and R. Murray, "Graph Laplacians and Stabilization of Vehicles Formations," in *15th IFAC World Congress*, Barcelona, Spain, 2002.

[80] T. Harju, "Lecture Notes on Graph Theory," FIN-20014 Turku, Finland, 2007.

[81] J. Bondy and U. Murty, *Graph Theory with Applications*. North-Holland: North-Holland, 1982.

[82] F. Chung, *Spectral Graph Theory*, ser. No. 92 in Regional Conf. Series in Mathematics. American Mathematical Society, 1994, no. 92.

[83] R. Smith and F. Hadaegh, "Closed-Loop Dynamics of Cooperative Vehicle Formations with Parallel Estimators and Communication," *IEEE Trans. Automatic Control*, vol. 52, no. 8, pp. 1404–1414, 2007.

[84] J. Fax and R. Murray, "Information Flow and Cooperative Control of Vehicle Formations," *IEEE Trans. Auto. Control*, vol. 49, no. 9, pp. 1465–1476, 2004.

[85] G. Lafferriere, A. Williams, J. Caughman, and J. Veeraman, "Decentralized Control of Vehicles Formations," *Syst. and Control Letters*, vol. 54, no. 9, pp. 899–910, 2005.

[86] "http://www.physics.ncsu.edu/courses/astron/orbits.html."

[87] "http://www.srrb.noaa.gov."

[88] "http://en.wikipedia.org/wiki/celestialequator."

Appendix A

GRAPHS

Most of the definition follows, [78], [79], [80] and [81].

A.1 Graph Theory

Definition 5 (graph) *A graph consists of a pair* $\aleph = (V(\aleph), E(\aleph))$ *of nonempty set of vertices,* $V(\aleph) = \{v_i | \, i = 1, 2, ...N\}$ *where* N *is the order of the graph (definition 7), and a set of edges* $E(\aleph)$*, such that* $E(\aleph)$ *disjoint from* $V(\aleph)$*,* $E(\aleph) \cap V(\aleph) = \varnothing$*, satisfying* $E(\aleph) \subset V(\aleph) \times V(\aleph)$*. The set of edges of the graph,* $E(\aleph) \{(v_i, v_j) | \, v_i, v_j \in V(\aleph)\}$*, where* $i, j = 1, 2, ..N$*,* $i \neq j$ *each edge is denoted by* $e(\aleph)$ *and* v_i *and* v_j *are vertices,* $e(\aleph) = (v_i, v_j)$*.*

Definition 6 (size of a graph) *The number of edges of a graph* $\aleph$ *is called size of graph* $\aleph$*, and it is written as* $|E(\aleph)| = N_e$*.*

Definition 7 (*order* of a graph) *The order of a graph* $\aleph$ *is the number of vertices of the graph, called order of a graph* $\aleph$*, and it is written as* $|V(\aleph)| = N$*.*

Definition 8 (*trivial* graph) *Trivial graph is a graph of order* 0 *or* 1*.*

Definition 9 (ends) *For an edge* $e(\aleph) = (v_i, v_j)$ *with* $i, j = 1, 2, ..N$*,* $i \neq j$*, the vertices* v_i *and* v_j *are its ends.*

Definition 10 (tail/head) *To an edge* $e(\aleph) = (v_i, v_j)$ *with* $i, j = 1, 2, ..N$*,* $i \neq j$*, we refer to* v_i*, as the head and to* v_j*, as the tail of the edge* $e(\aleph)$*.*

Definition 11 (*incident* vertex) *A vertex* v *is said incident with the edge* $e(\aleph)$ *if* v *is the end of the edge, if* $v \in e(\aleph)$*.*

Definition 12 (neighbors or adjacent vertices) *Two vertices, v_i and v_j with $i, j = 1, 2, ..N$, $i \neq j$, of graph $\aleph$ are adjacent or neighbors, if (v_i, v_j) is an edge of $\aleph$, in which case the common edge is said to join the two vertices, $N_v(\aleph) = \{ v_i \in V(\aleph) | (v_i, v_j) \in E(\aleph) \}$.*

Definition 13 (*adjacent* edges) *Two edges are adjacent, with $e(\aleph) \neq f(\aleph)$, if they are incident with a common vertice.*

Definition 14 (*multiple* edges) *Multiple edges are such that there is more than one edge between the same two vertices.*

Definition 15 (loop) *An edge (v_i, v_j) with identical vertices forms a loop, v_i, $v_j \in V(\aleph)$, $v_i = v_j$.*

Definition 16 (*simple* graph) *A simple graph is a graph which contains no loops or multiple edges.*

Definition 17 (*complete* graph) *In a complete graph each pair of vertices is joined by an edge, that is, the graph contains all possible edges.*

Definition 18 (*degree* of a vertex) *The degree (or valency) of a vertex v is the number of edges of $\aleph$ incident to the vertex or the number of its neighbors, $deg(v) = |N_v(\aleph)|$ with loops being counted twice.*

Definition 19 (*maximum/minimum* degrees) *The maximum and minimum degrees are respectively, $\Delta(\aleph)$ and $\delta(\aleph)$, which corresponds to the largest (smallest) degree over all vertices, respectively,*

$$\begin{aligned} \Delta(\aleph) &= \min \{ deg(v) | v \in V(\aleph) \} \\ \delta(\aleph) &= \max \{ deg(v) | v \in V(\aleph) \} \end{aligned}$$

Definition 20 (*isolated* vertex) *A vertex of degree 0 is an isolated vertice in graph $\aleph$.*

Definition 21 (κ-*regular* graph) *A graph is regular if every vertice of the graph $\aleph$ has the same degree. If this degree is equal to κ, then the graph is called a κ-regular graph and the graph itself is said to have degree κ. In a complete graph with N vertices, the graph has degree $\kappa = N - 1$.*

Definition 22 (*walk*) *A walk in a graph κ, called W, is an alternating sequence of vertices and edges, beginning and ending with vertices, such that for any edge in the sequence, e_i for $1 \leq i \leq N$, the vertices of e_i are v_{i-1} and v_i. A walk is closed if its first and last vertices are the same, $v_0 = v_N$, otherwise open, $v_0 \neq v_N$.*

Definition 23 (*path*) *A path is an open walk which is simple, meaning that no vertex appears more than once, $v_i \neq v_j$ for all $i \neq j$ with $1 \leq i \leq j \leq N$.*

Definition 24 (*cycle*) *A cycle is a closed walk which is simple, except for the first and last vertices which are equal, $v_i \neq v_j$ for all $i \neq j$, except $v_0 = v_N$. A length of a cycle is its number of vertices or edges and is called N-cycle and denoted by C^N.*

Definition 25 (*acyclic*) *A graph without cycles is said to be acyclic.*

Definition 26 (*subgraph* of a graph) *A graph $\aleph'$ is a subgraph of $\aleph$, $\aleph' \subseteq \aleph$ if its vertices belong to the sets of vertices of $\aleph$, $V(\aleph') \subseteq V(\aleph)$ and the set of edges contain all the edges of H', $E(\aleph') \subseteq E(\aleph)$.*

Definition 27 (union/intersection) *Let $\Im$ be a graph, then $\aleph \cup \Im := (V(\aleph) \cup V(\Im), E(\aleph) \cup E(\Im))$ and $\aleph \cap \Im := (V(\aleph) \cap V(\Im), E(\aleph) \cap E(\Im))$*

Definition 28 (sum) *For a subset V_1 of $V \times V$ we write $\aleph + V_1 := (V, E \cup V_1)$.*

Definition 29 (*disjoint* graph) *Considering the graph $\aleph$ and subgraph $\aleph$, if $\aleph \cap \aleph' = \varnothing$, then $\aleph$ and $\aleph\prime$ are disjoint.*

Definition 30 (*spanning* graph) *A spanning graph of $\aleph$ is a subgraph, $\aleph'$, with vertex set $V(\aleph)$, such that $V(\aleph') = V(\aleph)$.*

Definition 31 (undirected graph) *The graph $\aleph$ is said indirected if $(v_i, v_j) \in E(\aleph) \Leftrightarrow (v_j, v_i) \in E(\aleph)$.*

Definition 32 (*connected* graph) *An undirected graph that has a path between every pair of vertices.*

Definition 33 (*component* graph) *An n-component of a graph is a maximal connected subgraph. A componenet being connected is always non-empty, therefore, an empty graph has no components.*

Definition 34 (*tree*) *Tree is an undirected, acyclic, connected graph, i.e., if $|V(\aleph)| = N$, then $|E(\aleph)| = N - 1$.*

Definition 35 (strongly connected graph) *A graph $\aleph$ with the property that for any v_i, $v_j \in V(\aleph)$, with $i, j = 1, 2, ..N$, $i \neq j$, there exists a path from v_i to v_j is said strongly connected.*

Definition 36 (*disconnected* graph) *A graph in which disjoint subsets of vertices exists whose elements do not have access to one another is termed disconnected. An undirected graph is either strongly connected or disconnected.*

Definition 37 (digraph) *A directed graph, digraph or oriented graph, Ð, is a pair Ð= $(V(Ð), E(Ð))$, where the edges are ordered pairs of vertices. The edge is said to be directed from the head v_i to the tail v_j with $i, j = 1, 2, ..N$, $i \neq j$, where $v_i = head(e(Ð))$ and $v_j = tail(e(Ð))$. Directed graphs allow both (v_i, v_j) and (v_j, v_i), with $(v_i, v_j) \neq (v_j, v_i)$ and the inverse of $e(Ð)$ as $e^{-1}(Ð) = (v_j, v_i)$.*

Definition 38 (*strongly connected*digraph) *A directed graph is called strongly connected if for every pair of vertices u and v there is a path from u to v and a path from v to u. The vertices in a strongly connected graph must all have indegree and outdegree of at least 1.*

Definition 39 (*hypergraph*) *If the same edge can connect any pair of vertices then the graph is said hypergraph. Graphs can be seen as special hypergraphs.*

Definition 40 (*underlying* graph) *To each digraph Ð we can associate an underlying graph $Ð^U$ on the same vertex set of Ð, $V(Ð)$, where to each edge of $Ð^U$ there is the corresponding undirected edge in Ð. Sum and union are the same for underlying graphs.*

Definition 41 (source vertex) *A vertex of a directed graph with no incoming edges, v_r. More formaly, a vertex with indegree 0 ($deg_{in}(v_i) = 0$ definition 48).*

Definition 42 (sink vertex) *A vertex of a directed graph with no outcoming edges, v_s. More formaly, a vertex with outdegree 0 ($deg_{out}(v_i) = 0$ definition 48).*

Definition 43 (*directed* tree) *A directed tree is a digraph whose underlying graph is a tree, in fact a tree can be viewed as a minimal connected graph.*

Definition 44 (*rooted* tree) *if the tree is called rooted is because there is a distinguished fixed vertice, v_r, called the source or root, such that there is a directed path from vertex v_r to any other vertice v_i, of a graph.*

Definition 45 (*network*) *A network is an underlying digraph with two distinct subsets of vertices, called sources and sinks. Intermediate vertices are those that are neither sources or sinks and there exists a capacity function associated to each edge.*

A.2 Algebraic Graph Theory

Algebraic graph theory studies the relationships between the structure of the graphs and different matrix representations of graphs. The part of algebraic graph theory that studies relationships between the properties of the graph and its adjacency matrix is also called spectral graph theory. It studies the spectrum, which are the set of eigenvalues, of the adjacency matrix, the Kirchhoff matrix, or the Laplacian matrix of a graph. The definitions presented here follow closely [79]. An extended work about spectral graph theory can be found in [82].

Definition 46 (*Adjacency Matrix*) *Let Ð= $(V(Ð),E(Ð))$ be a digraph where the vertices of Ð are enumerated according to $V(Ð) = \{v_i \mid i \in [1, N]\}$ and the edges $E(Ð) = \{e_{i,i+1}(Ð) = (v_i, v_{i+1}) \mid v \in V(Ð)\}$, so $|V(Ð)| = N$ and $|E(Ð)| = M$. The adjacency matrix associated to the undirected or directed graph Ðis a square matrix of size $|V(Ð)|$, the $N \times N$ matrix $\mathbf{T}_g(Ð) = [t_{ij}]$ in which t_{ij} is the number of edges joining vertex v_j and v_i, and the diagonal entry a_{ii} is the number of loops, for all $1 \leq i \leq j \leq N$,*

$$t_{ij} = \begin{cases} 1 \text{ if } (v_j, v_i) \in V \\ 0 \text{ if } (v_j, v_i) \notin V \end{cases}$$

Using the definition of [83], and as we will see later in this chapter, it is interesting to associate to the adjacency matrix components to the existing communication links from vertex j to the vertex i, where the estimator j/i associated to the respective vertices are repectively the transmitter/receiver. In this situation, the indegree of each vertex corresponds to the number of receiving links.

When the graph in question is clear, the adjacency matrix will be denoted as $\mathbf{T}_g(Ð)$. Note that $\mathbf{T}_g(Ð)$ uniquely specifies a graph, and it is not the adjacency matrix of any other graph, although $\mathbf{T}_g(Ð)$ itself is not unique for a given graph because it depends on the enumeration of the vertices. As such, it is clear that the eigenvalues of $\mathbf{T}_g(Ð)$ are uniquely specified by the graph (thought the converse is not true).

Definition 47 (*Incidence* matrix) *The incidence matrix of a directed graph Ð is a $N \times M$ matrix $B_g(Ð)$ where N and M are the number of vertices and edges respectively, such that,*

$$b_{ij} = \begin{cases} -1 \text{ if } v_i = tail(e_j) \\ 0 \text{ otherwise} \\ 1 \text{ if } v_i = head(e_j) \end{cases}$$

Definition 48 (*indegree/outdegree*) *The indegree $\deg_{in}(v_i)$ of a vertex v_i, is the number of edges*

that come in v_i. The sum of tail endpoints count toward the outdegree,

$$\begin{aligned} \deg_{in}(v_i) &= \sum_{j=1}^{N} e_{ji}(Ð) \\ \deg_{out}(v_i) &= \sum_{j=1}^{N} e_{ij}(Ð) \end{aligned}$$

Definition 49 (*dominated/undominated* vertex) *A vertex v_i will be called undominated if $\deg_{in}(v_i) = 0$ and dominated if $\deg_{in}(v_i) \geq 1$.*

Definition 50 (*Laplacian Matrix*) *Let $\mathbf{D}_g(G)$ be the degree matrix, which is a diagonal matrix with the indegree of each vertice along the diagonal,*

$$d_{ij} = \begin{cases} \deg_{in}(v_i) \text{ if } v_i = v_j \\ 0 \text{ if } v_i \neq v_j \end{cases}$$

Thus, according to the adjacency defintion, the degree matrix represents the receiving links in each vertex of the graph. The Laplacian of the graph is defined as,

$$\mathbf{L}(Ð) = \mathbf{D}_g(Ð) - \mathbf{T}_g(Ð) \tag{A.1}$$

where $\mathbf{T}_g$ is the adjacency matrix and represents the communication link between vertex j (transmitter) and vertex i (receiver). The Laplacian satisfies the property,

$$\mathbf{L}(Ð) = \mathbf{B}_g(Ð)\left(\mathbf{B}_g(Ð)\right)^T$$

independenlyt of the graph $\mathbf{D}_g(Ð)$ being directed or not.

Interest in Laplacian and it properties had increased in the recent years, where researchers identified many links between the Laplacian eigenvalues and graph theoretic properties. Here we present some relevant for this work, which can be found in [84].

Laplacian eigenvalues

The Laplacian matrix is of great interest because its eigenvalues are related to the structural properties of the graph Ð. Let us define the Laplacian eigenvalues as, $eig\{\mathbf{L}(Ð)\} = \{eig^0, eig^1, \ldots, eig^{N-1}\}$, with $eig^0 \leq eig^1 \leq \ldots \leq eig^{N-1}$. The multiplicity λ of an eigenvalue is indicated in superscript in round brackets, $eig^{(\lambda)}$.

1. Every row sum of the Laplacian matrix is zero, which implies that there is always a zero eigenvalue corresponding to an eigenvector $[1 \cdots 1]^T$.

2. If Ð is an acyclic digraph, one of the Laplacian's eigenvalues is always zero.

3. If the directed graph has a root directed spanning tree, then the eigenvalue 0 has multiplicity 1, (proof can be found in [85]).

4. Also, the graph is regular if and only if $[1 \cdots 1]^T$is an eigenvector of the Laplacian.

5. If Ð is a strongly connected graph, the zero eigenvalue of its Laplacian is simple but the other way around is not true.

6. The multiplicity of the eigenvalue 0 in the Laplacian of a graph is equal to the number of connected components in the graph.

7. For an undirected graph, the Laplacian matrix is symmetric positive semi-definite. A symmetric Laplacian has real eigenvalues, meaning that its eigenspaces are mutually orthogonal. Thus, undirected graphs have mutual orthogonal eigenspaces.

8. An eigenvalue with multiplicity $\imath$ has $\imath-$dimensional eigenspaces.

Example 1 *Let us consider the digraph Ð in Fig. A-1. The adjacency matrix according to the previous definition, is*

$$\mathbf{T}_g\left(Ð\right) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

and the degree matrix is

$$\mathbf{D}_g\left(Ð\right) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Thus the Laplacian is given by the following matrix,

$$\mathbf{L}(\not{D}) = \mathbf{D}_g(\not{D}) - \mathbf{T}_g(\not{D}) = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 2 & -1 & 0 & 0 & -1 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 0 & 2 & -1 & 0 \\ 0 & -1 & 0 & -1 & 3 & -1 \\ -1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

whose eigenvalues are, $0^{(1)}$*,* $3.7049^{(1)}$ *,* $(1.2190 \pm 0.4925i)^{(1)}$ *and* $(2.4285 \pm 0.7102i)^{(1)}$*. Notice that the sum of each row is zero implying that there is an zero eigenvalue, also any vertex can reach any other vertex, thus the graph is strongly connected, the eigenvalue zero is simple. Now, considering the underlying graph,* G^U *of the digraph* G*, the adjacency matrix is obtained independently of the orientation of the edges, thus it is symmetric*

$$\mathbf{T}_g\left(\not{D}^U\right) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

and the degree matrix is obtained without making distinction between outdegree or indegree, thus

$$\mathbf{D}_g\left(\not{D}^U\right) = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

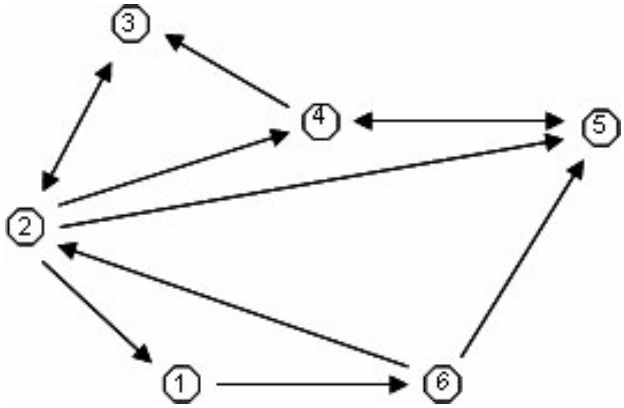

Figure A-1: Example of a graph.

and the Laplacian matrix is now,

$$\mathbf{L}\left(\mathcal{D}^{U}\right) = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & -1 \\ -1 & 5 & -1 & -1 & -1 & -1 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 & 3 & -1 & 0 \\ 0 & -1 & 0 & -1 & 3 & -1 \\ -1 & -1 & 0 & 0 & -1 & 3 \end{bmatrix}$$

whose eigenvalues are, $0^{(1)}$*,* $3.7049^{(1)}$*,* $1.3820^{(1)}$*,* $2.3820^{(1)}$*,* $3.6180^{(1)}$*,* $4.6180^{(1)}$ *and* $6.0000^{(1)}$*. Notice that for an underlying graph the eigenvalues of the Laplacian matrix are always real.*

Appendix B

EXTENDED KALMAN FILTER

B.1 Continuous-Discrete Extended Kalman Filter Equations

Consider the nonlinear equation written in a compact form,

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), t) + \breve{\mathbf{w}}(t) \tag{B.1}$$

while the output is given by,

$$\mathbf{y}(k) = h(\mathbf{x}(k), k) + \boldsymbol{\nu}(k) \tag{B.2}$$

where $\mathbf{y}(k)$ is the n_y-dimensional observation vector at time k, and where $\boldsymbol{\nu}(k)$ is observation noise, assumed to be white Gaussian with,

$$\begin{aligned} E\left\{\boldsymbol{\nu}(k)\boldsymbol{\nu}^T(k)\right\} &= \mathbf{R}(k) \\ E\left\{\boldsymbol{\nu}(k)\right\} &= 0 \end{aligned}$$

Moreover, we assume that the process noise, $\breve{\mathbf{w}}(t)$, is independent of the observation noise, $\boldsymbol{\nu}(k)$, and is also assumed, to be white Gaussian with,

$$\begin{aligned} E\left\{\breve{\mathbf{w}}(t)\breve{\mathbf{w}}^T(\tau)\right\} &= \breve{\mathbf{Q}}(t)\delta(t-\tau) \\ E\left\{\breve{\mathbf{w}}(t)\right\} &= 0 \end{aligned}$$

Consider the nonlinear equation being expanded by the Taylor series about a nominal or reference

trajectory (in case of a formation of vehicles we consider the formation center, f_c), $x_{f_c}(t)$

$$\dot{\mathbf{x}}(t) - \dot{\mathbf{x}}_{f_c}(t) = \left.\frac{\partial f(\mathbf{x}(t),t)}{\partial \mathbf{x}(t)}\right|_{\mathbf{x}(t)=\mathbf{x}_{f_c}(t)} (\mathbf{x}(t) - \mathbf{x}_{f_c}(t)) + h.o.t + \breve{\mathbf{w}}(t)$$

where $h.o.t$ are higher order terms then $\mathbf{x}(t) - \mathbf{x}_{f_c}(t)$, assumed to be neglected, and which can always be masked by the observation noise, $\breve{\mathbf{w}}(t)$, which may be increased to account for this effect. Defining $\delta\mathbf{x}(t) \triangleq \mathbf{x}(t) - \mathbf{x}_{f_c}(t)$ as the deviation or small perturbation from the nominal trajectory, and assuming that these deviations are small, given by the state estimate, we obtain, the first order linear approximation of the original nonlinear system,

$$\boldsymbol{\delta}\dot{\mathbf{x}}(t) = \mathbf{F}(\hat{\mathbf{x}}(t),t)\delta\mathbf{x}(t) + \breve{\mathbf{w}}(t) \tag{B.3}$$

where $\mathbf{F}(\hat{\mathbf{x}}(t),t)$ is the Jacobian of $f(.)$, denoted by,

$$\mathbf{F}(\hat{\mathbf{x}}(t),t) \triangleq \left.\left[\nabla f(\mathbf{x},t)^T\right]^T\right|_{\mathbf{x}=\hat{\mathbf{x}}(t)} = \left.\frac{\partial f(\mathbf{x},t)}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(t)} \tag{B.4}$$

We can now discretize the approximated linear equation (B.3), in [52], obtaining the following linearized discrete system,

$$\boldsymbol{\delta}\mathbf{x}(k+1) = \boldsymbol{\Phi}(k+1,k)\delta\mathbf{x}(k) + \mathbf{w}(k)$$

evaluated for the nominal trajectory $\mathbf{x}_{f_c}(k)$ given by the state estimate at time k, and where:

$\boldsymbol{\Phi}(k+1,k)$ is the state transition matrix,

$$\boldsymbol{\Phi}(k+1,k) \triangleq e^{\int_{t_k}^{t_{k+1}} \mathbf{F}(\tau)d\tau}$$

which can be be approximated by,

$$\boldsymbol{\Phi}(k) \approx \mathbf{1} + \mathbf{F}(\hat{\mathbf{x}}(t),t)T_s + \frac{(\mathbf{F}(\hat{\mathbf{x}}(t),t)T_s)^2}{2!}$$

where T_s is the sampling period used to propagate the estimate

$\mathbf{w}(k)$ $(n_x \times n_x)$-dimensional, is a zero-mean white Gaussian sequence,

$$\begin{aligned} E\left\{\mathbf{w}(k)\mathbf{w}^T(j)\right\} &= \mathbf{Q}(k)\delta_{kj} \\ E\left\{\mathbf{w}(k)\right\} &= 0 \end{aligned}$$

$\mathbf{Q}(k)$ is obtained from the power spectrum of the continuous noise,

$$\mathbf{Q}(k) \triangleq \mathbf{Q}(t_k) = \int_{t_k}^{t_{k+1}} \mathbf{\Phi}(t_{k+1},\tau)\breve{\mathbf{Q}}(\tau)\mathbf{\Phi}^T(t_{k+1},\tau)d\tau \tag{B.5}$$

In order to use the EKF the nonlinear measurement model must be linearized. Using the perturbation definition $\delta\mathbf{x}(k) \triangleq \mathbf{x}(k) - \hat{\mathbf{x}}(k)$, in the measurement model, it follows,

$$\begin{aligned} \mathbf{y}(k) &= h(\mathbf{x}(k)) + \boldsymbol{\nu}(k) \\ &= h(\hat{\mathbf{x}}(k) + \delta\mathbf{x}(k)) + \boldsymbol{\nu}^i(k) \\ &\approx h(\hat{\mathbf{x}}(k),k) + \left.\frac{\partial h^i}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(k)} \delta\mathbf{x}(k) + \boldsymbol{\nu}(k) \\ &= \hat{\mathbf{y}}(k) + \mathbf{H}(\hat{\mathbf{x}}(k),k)\delta\mathbf{x}(k) + \boldsymbol{\nu}(k) \end{aligned} \tag{B.6}$$

The previous equations lead to the linearized measurement equation for the Kalman filter,

$$\boldsymbol{\delta}\mathbf{y}(k) = \mathbf{H}(\hat{\mathbf{x}}(k),k)\delta\mathbf{x}(k) + \boldsymbol{\nu}(k) \tag{B.7}$$

where $\delta\mathbf{y}(k)$ is the measurement perturbation from the nominal measurement, $\delta\mathbf{y}(k) \triangleq \mathbf{y}(k) - \hat{\mathbf{y}}(k)$ and $\mathbf{H}(k)$ is the Jacobian of $h(.)$, denoted by

$$\mathbf{H}(\hat{\mathbf{x}}(k),k) \triangleq \left[\nabla h(\mathbf{x},k)^T\right]^T\Big|_{\mathbf{x}=\hat{\mathbf{x}}(k)} = \left.\frac{\partial h(\mathbf{x},k)}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(k)} \tag{B.8}$$

evaluated about the nominal trajectory, given by the state estimate.

To summarize, the linearized version of the nonlinear system of (B.1) with output (B.2), is,

$$\begin{aligned} \boldsymbol{\delta}\mathbf{x}(k+1) &= \mathbf{\Phi}(k+1,k)\delta\mathbf{x}(k) + \mathbf{w}(k) \\ \boldsymbol{\delta}\mathbf{y}(k) &= \mathbf{H}(k)\delta\mathbf{x}(k) + \boldsymbol{\nu}(k) \end{aligned}$$

Now, we can apply the equations for the linear KF, provided that the Jacobian matrices exist, yet the output of the filter will be the estimate of the state perturbation instead the state estimate. Replacing $\mathbf{x}_{f_c}(k) = \hat{\mathbf{x}}(k)$ in the perturbation definition $\delta\mathbf{x}(k) \triangleq \mathbf{x}(k) - \mathbf{x}_{f_c}(k)$, follows,

$$\delta\mathbf{x}(k) \triangleq \mathbf{x}(k) - \hat{\mathbf{x}}(k) \tag{B.9}$$

so that,

$$\mathbf{P}\left(k|k\right) = E\left[\widetilde{\mathbf{x}}\left(k|k\right)\widetilde{\mathbf{x}}^{T}\left(k|k\right)\right]$$

$$\mathbf{P}\left(k|k\right) = E\left[\delta\widetilde{\mathbf{x}}\left(k|k\right)\delta\widetilde{\mathbf{x}}^{T}\left(k|k\right)\right]$$

Using (B.9) we have $\delta\hat{\mathbf{x}}\left(k|k\right) \triangleq \hat{\mathbf{x}}\left(k|k\right) - \hat{\mathbf{x}}\left(k|k-1\right)$, meaning that with the state perturbation estimates from the EKF we obtain an estimate of the full state

$$\hat{\mathbf{x}}\left(k|k\right) = \hat{\mathbf{x}}\left(k|k-1\right) + \delta\hat{\mathbf{x}}\left(k|k\right)$$

The linearization is performed every time there is a new measurement, and thus obtained a better nominal trajectory, which will prevent the large initial state estimates to propagate through time. The point is to use a better nominal trajectory as soon as one is available. However, the linearization is going to depend also on how good approximation is the linearized dynamics, and in the stability properties of the system. The full state estimate between measurements is propagated according to the nonlinear function with initial condition $\hat{\mathbf{x}}(0)$, with $\mathbf{w}(k) = 0$ and error covariance matrix $\mathbf{P}(0)$,

$$\delta\mathbf{x}(0) \sim N\left(0, \mathbf{P}(0)\right)$$

If we have a good initial state estimate, its initial error covariance matrix will be small, and if the linearized dynamics represent a good approximation to the nonlinear system, the process noise covariance matrix is small. Under these conditions the assumptions of state perturbations will remain small, or in other words the state estimates obtained for each time will be the nominal trajectory.

To summarize: we started with the continuous nonlinear system,

System model:	$\dot{\mathbf{x}}\left(t\right) = f(\mathbf{x}\left(t\right), t) + \breve{\mathbf{w}}(t)$
Measurement model:	$\mathbf{y}(k) = h(\mathbf{x}(k), k) + \boldsymbol{\nu}(k)$

Initial conditions

	$E\left\{\breve{\mathbf{w}}(t)\breve{\mathbf{w}}^{T}(\tau)\right\} = \breve{\mathbf{Q}}(t)\delta(t-\tau)$, $E\left\{\breve{\mathbf{w}}(t)\right\} = 0$
	$E\left\{\boldsymbol{\nu}(k)\boldsymbol{\nu}^{T}(k)\right\} = \mathbf{R}(t)$, $E\left\{\boldsymbol{\nu}(k)\right\} = 0$
	$E\left\{\breve{\mathbf{w}}(t)\boldsymbol{\nu}^{T}(k)\right\} = 0$ for all k and t

Filtering

State estimate update:	$\hat{\mathbf{x}}(k\|k) = \hat{\mathbf{x}}(k\|k-1) + \mathbf{K}(k)\left(\mathbf{y}(k) - h(k, \hat{\mathbf{x}}(k\|k-1))\right)$
Error covariance update:	$\mathbf{P}\left(k\|k\right) = \left[\mathbf{1} - \mathbf{K}(k)\mathbf{H}(k)\right]\mathbf{P}(k\|k-1)\left[\mathbf{1} - \mathbf{K}(k)\mathbf{H}(k)\right]^{T} +$
	$+\mathbf{K}(k)\mathbf{R}(k)\mathbf{K}^{T}(k)$

Kalman gain equation:	$\mathbf{K}(k) = \mathbf{P}(k\|k-1)\mathbf{H}^T(k)\left[\mathbf{H}(k)\mathbf{P}(k\|k-1)\mathbf{H}^T(k) + \mathbf{R}(k)\right]^{-1}$

Prediction

State estimate propagation:	$\frac{d}{dt}\hat{\mathbf{x}}(t) = f(\hat{\mathbf{x}}(t), t)$
Error covariance propagation:	$\mathbf{P}(k+1\|k) = \mathbf{\Phi}(k)\mathbf{P}(k\|k)(\mathbf{\Phi}(k))^T + \mathbf{Q}(k)$

Definitions

Jacobian of $f(\cdot)$	$\mathbf{F}(\hat{\mathbf{x}}(t),t) \triangleq \left[\nabla f(\mathbf{x},t)^T\right]^T\Big\|_{\mathbf{x}=\hat{\mathbf{x}}(t)}$
Jacobian of $h(\cdot)$	$\mathbf{H}(\hat{\mathbf{x}}(k),k) \triangleq \left[\nabla h(\mathbf{x},k)^T\right]^T\Big\|_{\mathbf{x}=\hat{\mathbf{x}}(k)}$
State Transition matrix	$\mathbf{\Phi}(k) \approx \mathbf{1} + \mathbf{F}(\hat{\mathbf{x}}(t),t)T_s + \frac{(\mathbf{F}(\hat{\mathbf{x}}(t),t)T_s)^2}{2!}.$

B.2 Algorithm for Relative Position and Translational Velocity Estimation

For each s/c $i = 1, 2, .., N$, circularly (*i.e.* 1 comes after $N-1$) and with $N = 3$. For sake of simplicity, we will denote $\mathbf{x}$ to denote the relative translational state vector $\mathbf{x}_{trans}$.

Filtering - Local Measurement Update and Estimation

For each time t is computed the corresponding true anomaly, θ_t according to (6.13).

B.2.1 Sensor observation

Calculation of the sensor observation $\mathbf{y}^i(k)$

1. Linearize the measurement equation for the i^{th} spacecraft using the entire formation state estimate

$$\hat{\mathbf{x}}^i(k|k-1) = \left[\begin{array}{cccccc} (\hat{\boldsymbol{\chi}}_{12})^T & (\hat{\boldsymbol{\chi}}_{32})^T & (\hat{\boldsymbol{\chi}}_{31})^T & (\hat{\boldsymbol{\chi}}'_{12})^T & (\hat{\boldsymbol{\chi}}'_{32})^T & (\hat{\boldsymbol{\chi}}'_{31})^T \end{array}\right]^T$$

   and the k^{th} measurement, $\mathbf{y}^i(k) = \mathbf{h}(\hat{\mathbf{x}}^i(k|k-1)) + \mathbf{v}^i(k)$ in order to compute the local observation matrix (6.27), $\mathbf{H}^i(k, \hat{\mathbf{x}}^i(k|k-1))$ (denoted as $\mathbf{H}^i(k)$ to simplify).

2. Compute local innovation covariance matrix

$$\mathbf{S}^i(k) = \mathbf{H}^i(k)\mathbf{P}^i(k|k-1)(\mathbf{H}^i(k))^T + \mathbf{R}^i(k)$$

   where $\mathbf{R}^i$ is the covariance matrix of the measurement noise.

3. Compute local Kalman Gain

$$\mathbf{K}^i(k) = \mathbf{P}^i(k|k-1)\mathbf{H}^i(k)(\mathbf{S}^i(k))^{-1}$$

4. Update local state estimate perturbation

$$\delta\hat{\mathbf{x}}^i(k) = \mathbf{K}^i(k)\left(\mathbf{y}^i(k) - \mathbf{H}^i(k, \hat{\mathbf{x}}^i(k|k-1))\right)$$

5. Update local state estimate

$$\hat{\mathbf{x}}^i(k|k) = \hat{\mathbf{x}}^i(k|k-1) + \delta\hat{\mathbf{x}}^i(k)$$

6. Compute the error covariance matrix

$$\mathbf{P}^i(k|k) = (\mathbf{1} - \mathbf{K}^i(k)\mathbf{H}^i(k))\mathbf{P}^i(k|k-1)(\mathbf{1} - \mathbf{K}^i(k)\mathbf{H}^i(k))^T + \mathbf{K}^i(k)\mathbf{R}^i(k)(\mathbf{K}^i(k))^T$$

B.2.2 Fusion of state vectors estimates

Fusion of state vectors estimates $\mathbf{z}^i(k) = \hat{\mathbf{x}}^{i-1}(k|k-1) + \mathbf{v}^{i-1}(k)$ where the superscript $i-1$ stands for the predecessor vehicle.

1. Compute the error covariance matrix

$$(\mathbf{P}^i(k|k))^{-1} = w_c(\mathbf{P}^i(k|k-1))^{-1} + (1-w_c)(\mathbf{P}^{i-1}(k|k-1))^{-1}$$

where the parameter w_c is chosen at every step such that the trace of the matrix $\mathbf{P}^i(k|k)$ is minimized.

2. Update local state estimate

$$\hat{\mathbf{x}}^i(k|k) = \mathbf{P}^i(k|k)\left\{w_c\left(\mathbf{P}^i(k|k-1)\right)^{-1}\hat{\mathbf{x}}^i(k|k-1) + (1-w_c)\left(\mathbf{P}^{i-1}(k|k-1)\right)^{-1}\mathbf{z}^i(k)\right\}$$

Prediction - Project Ahead

1. The relative dynamics (6.30) $\delta\mathbf{x}'(t) = \mathbf{F}(\hat{\mathbf{x}}, t)\delta\mathbf{x} + \mathbf{B}(t)\mathbf{u}(t)$ are approximated by a fourth-order four-stage Runge-Kutta method, [53], as follows,

$$\hat{\mathbf{x}}_{\ell+1} = \hat{\mathbf{x}}_\ell + \frac{\Delta t}{6}\left(\mathbf{p}_1 + 2\mathbf{p}_2 + 2\mathbf{p}_3 + \mathbf{p}_4\right)$$

with,

$$\begin{aligned}\mathbf{p}_1 &= \mathbf{F}(\hat{\mathbf{x}}_\ell, t_\ell)\hat{\mathbf{x}}_\ell + \mathbf{B}(t_\ell)\mathbf{u}(t_\ell)\\ \mathbf{p}_2 &= \mathbf{F}(t_\ell + \frac{\Delta t}{2})\left(\hat{\mathbf{x}}_\ell + \frac{\Delta t}{2}\mathbf{p}_1\right) + \mathbf{B}(t_\ell + \frac{\Delta t}{2})\mathbf{u}(t_\ell)\end{aligned}$$

$$\begin{aligned}\mathbf{Þ}_3 &= \mathbf{F}(t_\ell + \frac{\Delta t}{2})\left(\hat{\mathbf{x}}_\ell + \frac{\Delta t}{2}\mathbf{Þ}_2\right) + \mathbf{B}(t_\ell + \frac{\Delta t}{2})\mathbf{u}(t_\ell) \\ \mathbf{Þ}_4 &= \mathbf{F}(t_\ell + \Delta t)\left(\hat{\mathbf{x}}_\ell + \Delta t\mathbf{Þ}_3\right) + \mathbf{B}(t_\ell + \Delta t)\mathbf{u}(t_\ell)\end{aligned}$$

where Δt is the Runge-Kutta step size and $\ell \in \mathbb{N}^0$.

2. $\mathbf{P}^i(k+1|k) = \mathbf{\Phi}(k)\,\mathbf{P}^i(k|k)(\mathbf{\Phi}(k))^T + \mathbf{Q}(k)$

where $\mathbf{Q}$ is the covariance matrix of the process noise, mainly due to unmodeled dynamics, $\mathbf{\Phi}(k) \approx \mathbf{1} + \mathbf{F}(\hat{\mathbf{x}}^i(t), t)T_s + \frac{(\mathbf{F}(\hat{\mathbf{x}}^i(t),t)T_s)^2}{2!}$ is the transition matrix, and T_s is the sampling period used to propagate the estimate.

B.2.3 Algorithm in pseudo-code form

For $i = 1, 2, ..., N-1$, circularly (*i.e.*, 1 comes after $N-1$).

Filtering

if **Sensor observation $\mathbf{y}^i(k)$ on**

compute the local observation matrix (6.27), $\mathbf{H}^i(k, \hat{\mathbf{x}}^i(k|k-1))$ (equal to $\mathbf{H}^i(k)$ to simplify).

compute local innovation covariance matrix

$$\mathbf{S}^i(k) = \mathbf{H}^i(k)\mathbf{P}^i(k|k-1)(\mathbf{H}^i(k))^T + \mathbf{R}^i(k)$$

compute local Kalman Gain

$$\mathbf{K}^i(k) = \mathbf{P}^i(k|k-1)\mathbf{H}^i(k)(\mathbf{S}^i(k))^{-1}$$

update local state estimate perturbation

$$\delta\hat{\mathbf{x}}^i(k) = \mathbf{K}^i(k)\left(y^i(k) - \mathbf{H}^i(k, \hat{\mathbf{x}}^i(k|k-1))\right)$$

update local state estimate

$$\hat{\mathbf{x}}^i(k|k) = \hat{\mathbf{x}}^i(k|k-1) + \delta\hat{\mathbf{x}}^i(k)$$

compute the error covariance matrix

$$\mathbf{P}^i(k|k) = (\mathbf{1} - \mathbf{K}^i(k)\mathbf{H}^i(k))\mathbf{P}^i(k|k-1)(\mathbf{1} - \mathbf{K}^i(k)\mathbf{H}^i(k))^T + \mathbf{K}^i(k)\mathbf{R}^i(k)(\mathbf{K}^i(k))^T$$

else if **State estimate from previous spacecraft,** $\mathbf{z}^i(k) = \hat{\mathbf{x}}^{i-1}(k|k-1) + \mathbf{v}^{i-1}(k)$

compute the error covariance matrix

$$(\mathbf{P}^i(k|k))^{-1} = w_c(\mathbf{P}^i(k|k-1))^{-1} + (1-w_c)(\mathbf{P}^{i-1}(k|k-1))^{-1}$$

where the parameter w_c is chosen at every step such that the trace of the matrix $\mathbf{P}^i(k|k)$ is minimized.

update local state estimate

$$\hat{\mathbf{x}}^i(k|k) = \mathbf{P}^i(k|k)\left\{w_c\left(\mathbf{P}^i(k|k-1)\right)^{-1}\hat{\mathbf{x}}^i(k|k-1) + (1-w_c)\left(\mathbf{P}^{i-1}(k|k-1)\right)^{-1}\mathbf{z}^i(k)\right\}$$

end

Prediction

$\hat{\mathbf{x}}^i(k+1|k) = \mathbf{F}(k, \hat{\mathbf{x}}^i(k|k))$ approximated by a fourth-order four-stage Runge-Kutta method,

$$\hat{\mathbf{x}}_{\ell+1} = \hat{\mathbf{x}}_\ell + \frac{\Delta t}{6}\left(\mathbf{Þ}_1 + 2\mathbf{Þ}_2 + 2\mathbf{Þ}_3 + \mathbf{Þ}_4\right)$$

with,

$$\begin{aligned}
\mathbf{Þ}_1 &= \mathbf{F}(\hat{\mathbf{x}}_\ell, t_\ell)\hat{\mathbf{x}}_\ell + \mathbf{B}(t_\ell)\mathbf{u}(t_\ell) \\
\mathbf{Þ}_2 &= \mathbf{F}(t_\ell + \frac{\Delta t}{2})\left(\hat{\mathbf{x}}_\ell + \frac{\Delta t}{2}\mathbf{Þ}_1\right) + \mathbf{B}(t_\ell + \frac{\Delta t}{2})\mathbf{u}(t_\ell) \\
\mathbf{Þ}_3 &= \mathbf{F}(t_\ell + \frac{\Delta t}{2})\left(\hat{\mathbf{x}}_\ell + \frac{\Delta t}{2}\mathbf{Þ}_2\right) + \mathbf{B}(t_\ell + \frac{\Delta t}{2})\mathbf{u}(t_\ell) \\
\mathbf{Þ}_4 &= \mathbf{F}(t_\ell + \Delta t)\left(\hat{\mathbf{x}}_\ell + \Delta t\mathbf{Þ}_3\right) + \mathbf{B}(t_\ell + \Delta t)\mathbf{u}(t_\ell)
\end{aligned}$$

where $\mathbf{F}$ and $\mathbf{B}$ matrices are computed according to (6.28).

$\mathbf{P}^i(k+1|k) = \mathbf{\Phi}(k)\,\mathbf{P}^i(k|k)(\mathbf{\Phi}(k))^T + \mathbf{Q}(k)$ where the transition matrix is approximated by $\mathbf{\Phi}(k) \approx \mathbf{1} + \mathbf{F}(\hat{\mathbf{x}}(t), t)T_s + \frac{(\mathbf{F}(\hat{\mathbf{x}}(t),t)T_s)^2}{2!}$.

B.3 Algorithm for Attitude and Angular Velocity Estimation

For each s/c $i = 1, 2, ..., N-1$, circularly (*i.e.*, 1 comes after $N-1$) and with $N = 3$. For sake of simplicity, we will denote $\mathbf{x}$ to denote the absolute rotational state vector estimates

$$\mathbf{x}^i_{rot} = \begin{bmatrix} q^i_1 & q^i_2 & q^i_3 & q^i_4 & \omega^i_x & \omega^i_y & \omega^i_z \end{bmatrix}^T$$

Filtering - Local Measurement Update and Estimation

B.3.1 Sensor observation

Calculation of Sensor observation $\mathbf{y}^i(k) = q^i_{ST}(k)$

1. Compute local innovation covariance matrix

$$\mathbf{S}^i(k) = \mathbf{H}^i\mathbf{P}^i(k|k-1)(\mathbf{H}^i)^T + \mathbf{R}^i(k)$$

where $\mathbf{R}^i(k)$ is the covariance matrix of the measurement noise and $\mathbf{H}^i$ according to Eqs. (6.32)-(6.34).

2. Compute local Kalman Gain

$$\mathbf{K}^i(k) = \mathbf{P}^i(k|k-1)\mathbf{H}^i(\mathbf{S}^i(k))^{-1}$$

3. Update local state estimate perturbation

$$\begin{bmatrix} \delta\hat{\mathbf{x}}^i_{\mathbf{q}}(k) \\ \delta\hat{\mathbf{x}}^i_{\boldsymbol{\omega}}(k) \end{bmatrix} = \mathbf{K}^i(k)\delta\hat{\mathbf{q}}$$

with,

$$\delta\hat{q} = \begin{bmatrix} \delta\hat{\mathbf{q}}^i(k) \\ 1 \end{bmatrix} = q^i_{ST}(k) \otimes \hat{q}^{-1}(k|k-1) = \boldsymbol{\Xi}^T\left(\hat{q}(k|k-1)\right) q^i_{ST}(k)$$

4. Update local state estimate

$$\hat{\mathbf{x}}^i(k|k) = \begin{bmatrix} \hat{q}(k|k) \\ \hat{\boldsymbol{\omega}}(k|k) \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \delta\hat{\mathbf{x}}^i_q(k) \\ \sqrt{1 - \left\|\delta\hat{\mathbf{x}}^i_q(k)\right\|} \end{bmatrix} \otimes \hat{q}^i(k|k-1) \\ \hat{\boldsymbol{\omega}}^i(k|k-1) + \delta\hat{\mathbf{x}}^i_{\boldsymbol{\omega}}(k) \end{bmatrix}$$

5. Compute the error covariance matrix

$$\mathbf{P}^i(k|k) = (\mathbf{1} - \mathbf{K}^i(k)\mathbf{H}^i)\mathbf{P}^i(k|k-1)(\mathbf{1} - \mathbf{K}^i(k)\mathbf{H}^i)^T + \mathbf{K}^i(k)\mathbf{R}^i(k)(\mathbf{K}^i(k))^T$$

B.3.2 Fusion of state vectors estimates

Fusion of state vectors estimates $\mathbf{z}^i(k) = \hat{\mathbf{x}}^{i-1}(k|k-1) + \boldsymbol{\nu}^{i-1}(k)$ where the superscript $i-1$ stands for the predecessor vehicle.

1. Compute local innovation covariance matrix

$$\mathbf{S}^i(k) = \mathbf{H}^i\mathbf{P}^i(k|k-1)(\mathbf{H}^i)^T + \mathbf{P}^{i-1}(k|k-1)$$

where $\mathbf{H}^i$ according to Eqs. (6.42).

2. Compute local Kalman Gain

$$\mathbf{K}^i(k) = \mathbf{P}^i(k|k-1)\mathbf{H}^i(\mathbf{S}^i(k))^{-1}$$

3. Update local state estimate perturbation

$$\begin{bmatrix} \delta\hat{\mathbf{x}}^i_q(k) \\ \delta\hat{\mathbf{x}}^i_{\boldsymbol{\omega}}(k) \end{bmatrix} = \mathbf{K}^i(k) \begin{bmatrix} \delta\hat{q} \\ \delta\hat{\boldsymbol{\omega}} \end{bmatrix}$$

with,

$$\begin{aligned} \delta\hat{q} &= \hat{q}^{i-1}(k|k-1) \otimes \left(\hat{q}^i\left(k|k-1\right)\right)^{-1} = \boldsymbol{\Xi}^T\left(\hat{q}^i\left(k|k-1\right)\right)\hat{q}^{i-1}(k|k-1) \\ \delta\hat{\boldsymbol{\omega}} &= \hat{\boldsymbol{\omega}}^{i-1}(k|k-1) - \hat{\boldsymbol{\omega}}^i(k|k-1) \end{aligned}$$

4. Update local state estimate

$$\hat{\mathbf{x}}^i(k|k) = \begin{bmatrix} \hat{q}^i\left(k|k\right) \\ \hat{\boldsymbol{\omega}}^i\left(k|k\right) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Xi}\left(\hat{q}^i\left(k|k-1\right)\right)\delta\hat{\mathbf{x}}^i_q(k) \\ \hat{\boldsymbol{\omega}}^i\left(k|k-1\right) + \delta\hat{\mathbf{x}}^i_{\boldsymbol{\omega}}(k) \end{bmatrix}$$

5. Compute the error covariance matrix

$$\mathbf{P}^i(k|k) = (\mathbf{1} - \mathbf{K}^i(k)\mathbf{H}^i)\mathbf{P}^i(k/k-1)(\mathbf{1} - \mathbf{K}^i(k)\mathbf{H}^i)^T + \mathbf{K}^i(k)\mathbf{P}^{i-1}(k|k-1)(\mathbf{K}^i(k))^T$$

Prediction - Project Ahead

1. $\frac{d}{dt}\hat{\mathbf{x}}^i(t) = f(\hat{\mathbf{x}}^i(t), t)$ approximated by a fourth-order four-stage Runge-Kutta method, according to Wertz [53],

$$\hat{\boldsymbol{\omega}}_{\ell+1} = \hat{\boldsymbol{\omega}}_\ell + \frac{\Delta t}{6}\left(\mathbf{P}_1 + 2\mathbf{P}_2 + 2\mathbf{P}_3 + \mathbf{P}_4\right) \tag{B.10}$$

with,

$$\begin{aligned}
\mathbf{\mathfrak{P}}_1 &= f(\hat{\boldsymbol{\omega}}_\ell, t_\ell) \\
\mathbf{\mathfrak{P}}_2 &= f(\hat{\boldsymbol{\omega}}_\ell + \frac{\Delta t}{2}\mathbf{\mathfrak{P}}_1, t_\ell + \frac{\Delta t}{2}) \\
\mathbf{\mathfrak{P}}_3 &= f(\hat{\boldsymbol{\omega}}_\ell + \frac{\Delta t}{2}\mathbf{\mathfrak{P}}_2, t_\ell + \frac{\Delta t}{2}) \\
\mathbf{\mathfrak{P}}_4 &= f(\hat{\boldsymbol{\omega}}_\ell + \Delta t\mathbf{\mathfrak{P}}_3, t_\ell + \Delta t)
\end{aligned}$$

where $f(\hat{\boldsymbol{\omega}}(k|k), k)$ is computed according to (6.37), and the quaternion,

$$\hat{q}(k+1|k) = \left[\cos\left(\frac{\bar{\omega}_k T_s}{2}\right)\mathbf{1}_{4\times 4} + \frac{1}{\bar{\omega}_k}\sin\left(\frac{\bar{\omega}_k T_s}{2}\right)\boldsymbol{\Lambda}(\boldsymbol{\omega})\right]\hat{q}(k)$$

where $\bar{\omega}_k = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}$ and $\boldsymbol{\Lambda}(\boldsymbol{\omega}) = \begin{bmatrix} -[\boldsymbol{\omega}\times] & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix}$.

2. $\mathbf{P}^i(k+1|k) = \boldsymbol{\Phi}(k)\mathbf{P}^i(k)(\boldsymbol{\Phi}(k))^T + \mathbf{Q}(k)$, the propagation of the state error covariance matrix, where $\mathbf{Q}$ is the covariance matrix of the process noise (mainly due to unmodeled dynamics), $\boldsymbol{\Phi}(k) \approx \mathbf{1} + \mathbf{F}(\hat{\mathbf{x}}^i(t), t)T_s + \frac{(\mathbf{F}(\hat{\mathbf{x}}^i(t),t)T_s)^2}{2!}$ is the transition matrix, where $\mathbf{F}(\hat{\mathbf{x}}^i(t), t)$ is computed from Eq. (6.40) and T_s is the sampling period used to propagate the estimate.

B.3.3 Algorithm in pseudo-code form

For $i = 1, 2, ..., N-1$, circularly (*i.e.*, 1 comes after $N-1$).

Filtering

if **Sensor observation** $\mathbf{y}^i(k) = q^i_{ST}(k)$

compute the local observation matrix according to

Eqs. (6.32)-(6.34), $\mathbf{H}^i$

compute local innovation covariance matrix

$$\mathbf{S}^i(k) = \mathbf{H}^i\mathbf{P}^i(k|k-1)(\mathbf{H}^i)^T + \mathbf{R}^i(k)$$

compute local Kalman Gain

$$\mathbf{K}^i(k) = \mathbf{P}^i(k|k-1)\mathbf{H}^i(\mathbf{S}^i(k))^{-1}$$

update local state estimate perturbation

$$\delta\hat{\mathbf{x}}^i(k) = \begin{bmatrix} \delta\hat{\mathbf{x}}^i_{\mathbf{q}}(k) \\ \delta\hat{\mathbf{x}}^i_{\boldsymbol{\omega}}(k) \end{bmatrix} = \mathbf{K}^i(k)\delta\hat{\mathbf{q}}$$

with,

$$\delta\hat{q} = \begin{bmatrix} \delta\hat{\mathbf{q}} \\ 1 \end{bmatrix} = q^i_{ST}(k) \otimes \hat{q}^{-1}\left(k|k-1\right) = \boldsymbol{\Xi}^T\left(\hat{q}\left(k|k-1\right)\right) q^i_{ST}(k)$$

update local state estimate

$$\hat{\mathbf{x}}^i(k|k) = \begin{bmatrix} \hat{q}\left(k|k\right) \\ \hat{\boldsymbol{\omega}}\left(k|k\right) \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \delta\hat{\mathbf{x}}^i_{\mathbf{q}}(k) \\ \sqrt{1 - \left\|\delta\hat{\mathbf{x}}^i_{\mathbf{q}}(k)\right\|} \end{bmatrix} \otimes \hat{q}^i\left(k|k-1\right) \\ \hat{\boldsymbol{\omega}}^i\left(k|k-1\right) + \delta\hat{\mathbf{x}}^i_{\boldsymbol{\omega}}(k) \end{bmatrix}$$

compute the error covariance matrix

$$\mathbf{P}^i(k|k) = (\mathbf{1} - \mathbf{K}^i(k)\mathbf{H}^i)\mathbf{P}^i(k|k-1)(\mathbf{1} - \mathbf{K}^i(k)\mathbf{H}^i)^T + \mathbf{K}^i(k)\mathbf{R}^i(k)(\mathbf{K}^i(k))^T$$

else if **State estimate from previous s/c,** $\mathbf{z}^i(k) = \mathbf{x}^i(k|k-1) + \mathbf{v}^{i-1}(k)$

compute the local observation matrix according to Eqs. (6.42), $\mathbf{H}^i$

compute local innovation covariance matrix

$$\mathbf{S}^i(k) = \mathbf{H}^i\mathbf{P}^i(k|k-1)(\mathbf{H}^i)^T + \mathbf{R}^i(k)$$

compute local Kalman Gain

$$\mathbf{K}^i(k) = \mathbf{P}^i(k|k-1)\mathbf{H}^i(\mathbf{S}^i(k))^{-1}$$

update local state estimate perturbation

$$\delta\hat{\mathbf{x}}^i(k) = \begin{bmatrix} \delta\hat{\mathbf{x}}^i_q(k) \\ \delta\hat{\mathbf{x}}^i_{\boldsymbol{\omega}}(k) \end{bmatrix} = \mathbf{K}^i(k) \begin{bmatrix} \delta\hat{q} \\ \delta\hat{\boldsymbol{\omega}} \end{bmatrix}$$

with,

$$\begin{aligned}\delta\hat{q} &= \hat{q}^{i-1}(k|k-1) \otimes \left(\hat{q}^{i}\left(k|k-1\right)\right)^{-1} = \mathbf{\Xi}^{T}\left(\hat{q}^{i}\left(k|k-1\right)\right)\hat{q}^{i-1}(k|k-1)\\ \delta\hat{\boldsymbol{\omega}} &= \hat{\boldsymbol{\omega}}^{i-1}(k|k-1) - \hat{\boldsymbol{\omega}}^{i}(k|k-1)\end{aligned}$$

update local state estimate

$$\hat{\mathbf{x}}^{i}(k|k) = \begin{bmatrix} \hat{q}\left(k|k\right) \\ \hat{\boldsymbol{\omega}}\left(k|k\right) \end{bmatrix} = \begin{bmatrix} \mathbf{\Xi}\left(\hat{q}^{i}\left(k|k-1\right)\right)\delta\hat{\mathbf{x}}^{i}_{\mathbf{q}}(k) \\ \hat{\boldsymbol{\omega}}^{i}\left(k|k-1\right) + \delta\hat{\mathbf{x}}^{i}_{\boldsymbol{\omega}}(k) \end{bmatrix}$$

compute the error covariance matrix

$$\mathbf{P}^{i}(k|k) = (\mathbf{1} - \mathbf{K}^{i}(k)\mathbf{H}^{i})\mathbf{P}^{i}(k|k-1)(\mathbf{1} - \mathbf{K}^{i}(k)\mathbf{H}^{i})^{T} + \mathbf{K}^{i}(k)\mathbf{R}^{i}(k)(\mathbf{K}^{i}(k))^{T}$$

end

Prediction

$\frac{d}{dt}\hat{\mathbf{x}}^{i}(t) = f(\hat{\mathbf{x}}^{i}(t), t)$ approximated by a fourth-order four-stage Runge-Kutta method, according to Wertz [53],

$$\hat{\boldsymbol{\omega}}_{\ell+1} = \hat{\boldsymbol{\omega}}_{\ell} + \frac{\Delta t}{6}\left(\mathbf{\Phi}_1 + 2\mathbf{\Phi}_2 + 2\mathbf{\Phi}_3 + \mathbf{\Phi}_4\right)$$

approximated by a fourth-order four-stage Runge-Kutta method, and the quaternion,

$$\hat{q}(k+1|k) = \left[\cos\left(\frac{\bar{\omega}_k T_s}{2}\right)\mathbf{1}_{4\times 4} + \frac{1}{\bar{\omega}_k}\sin\left(\frac{\bar{\omega}_k T_s}{2}\right)\mathbf{\Lambda}\left(\boldsymbol{\omega}\right)\right]\hat{q}\left(k\right)$$

where $\bar{\omega}_k = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}$ and $\mathbf{\Lambda}\left(\boldsymbol{\omega}\right) = \begin{bmatrix} -\left[\boldsymbol{\omega}\times\right] & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^{T} & 0 \end{bmatrix}$

$\mathbf{P}^{i}(k+1|k) = \mathbf{\Phi}\left(k\right)\mathbf{P}^{i}(k|k)(\mathbf{\Phi}\left(k\right))^{T} + \mathbf{Q}(k)$.

Appendix C

RELATIVE DYNAMICS

In this appendix the translational relative equations of motion among s/c in a formation are obtained. First, in Section C.1 the parametrization of an elliptic orbit is performed, then the relative motion of two s/c is described in Section C.3, using the two-body problem, presented in Section C.2. These second-order differential equations provide an incomplete description of the s/c Earth motion, since the acceleration caused by Earth's gravitational harmonics acting on the s/c is not taking into account. This is accomplished through the angular momentum equation between the s/c and the angular momentum of the rotational reference frame, presented in subsection C.3.2. Assuming that the reference frame has a quasi circular orbit and that the distance between s/c is small compared to the distance of the formation orbit reference frame in inertial frame, then the equations are reduced to the approximate linearized equations of Clohessy-Wiltshire or Hill's equations. Because the true anomaly provides a natural basis for parameterizing of the formation time and motion, the Hill's equations are further deducted for true anomaly varying instead time, described in subsection C.3.3. The definitions presented in this appendix follow closely [79]. The reference frames used are defined in detail in subsection 6.3.1.

C.1 Elliptic Orbit Parameterization

This section follows closely [86].

An eccentric orbit can be represented by the standard orbital parameters, shown in Figs. (C-1)-(C-2), which are:

- **semi-major axis**, a_{smx}: half of the longer of the two axes of the orbital ellipse;
- **eccentricity**, e_{cc}: half of the distance between the foci of an ellipse divided by the semi-major axis. An eccentricity of 0 would be a circle orbit;

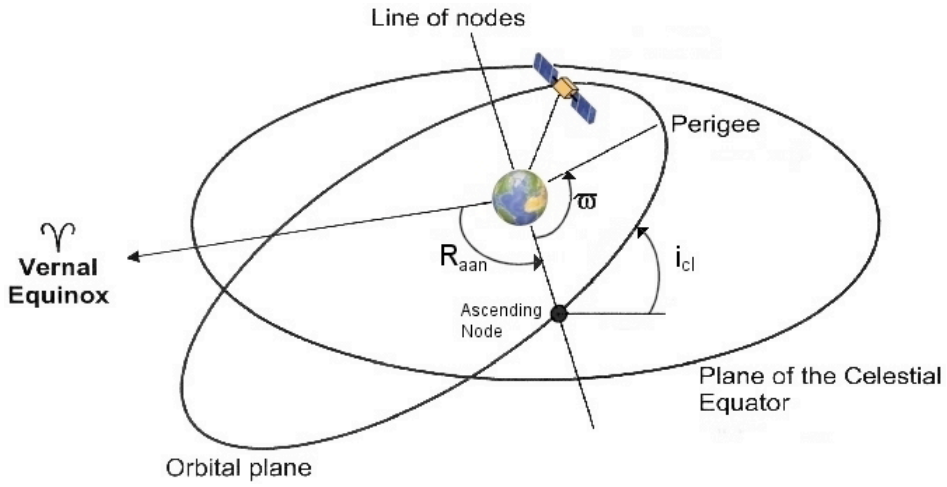

Figure C-1: Geometry representation of the elliptical orbit parameters inclination, i_{cl}, right ascension of the ascending node**,** R_{aan}, and argument of perigee, ϖ. Reprinted from [86].

- **inclination**, i_{cl}: the angle between the plane of the orbit and the plane of the *celestial equator* for Earth orbiting spacecraft;
- **right ascension of the ascending node**, R_{aan} (acronym RAAN): it is the angle of the ascending node measured EAST of the *vernal equinox* along the *celestial equator*;
- **argument of perigee**, ϖ: the angle between the ascending node and perigee, measured counter clockwise along the plane of the orbit;
- **true anomaly**, θ_t: the actual angle that a spacecraft has moved since last passing perigee.

More useful terms in the aerospace engineering:

- **Perigee** is the point in an orbit when the spacecraft is closest to the Earth, as shown in Fig. (C-2);
- **Ecliptic** is the plane of the Earth's orbit around the sun;
- **Vernal Equinox** (vernal is derived from Latin *ver* = spring) is the point where the *ecliptic* crosses the *celestial equator* with the sun passing from south to north;
- **Celestial Sphere**, a imaginary sphere surrounding the Earth, as shown in Fig. C-3, at some arbitrary great distance, upon which the stars are considered to be fixed for the purpose of position measurement. The celestial sphere is divided by projecting the *celestial equator* into space. This divides the sphere into the north celestial hemisphere and the south celestial hemisphere;

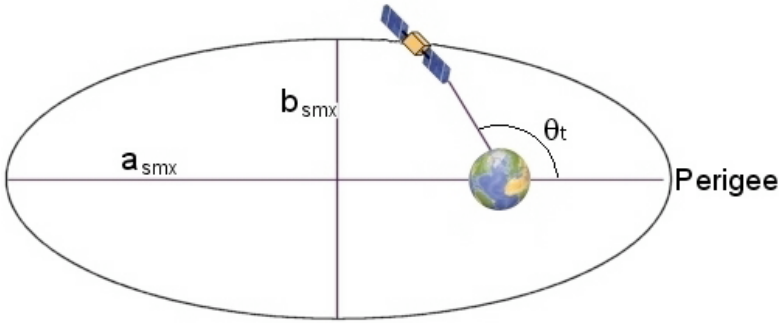

Figure C-2: Geometry representation of the elliptical orbit parameters, θ_t, true anomaly, and semi-major axis, a_{smx}, semi-minor axis, b_{smx}, in the orbital plane. Reprinted from [86].

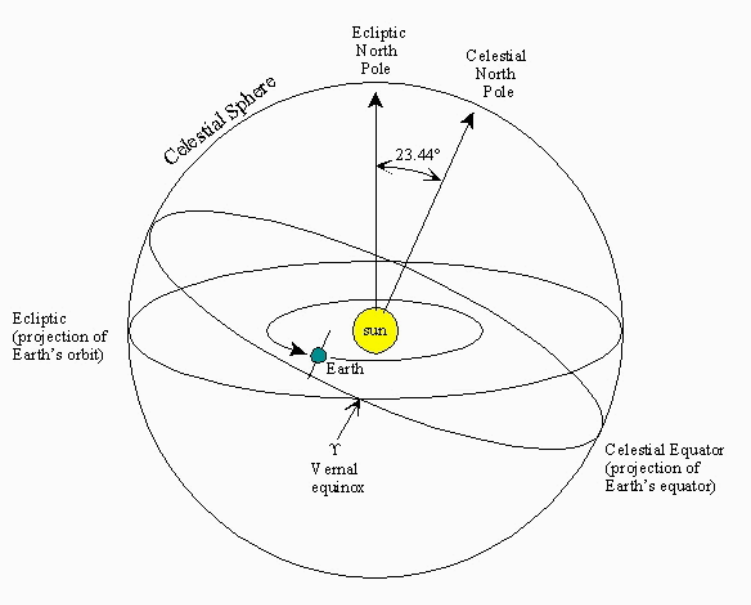

Figure C-3: The ecliptic or plane of the Earth's orbit and the celestial equator cross at two points, the vernal equinox and the autumnal equinox. The celestial equator is tilted 23.5 degrees in relation to the ecliptic. Reprinted from [87].

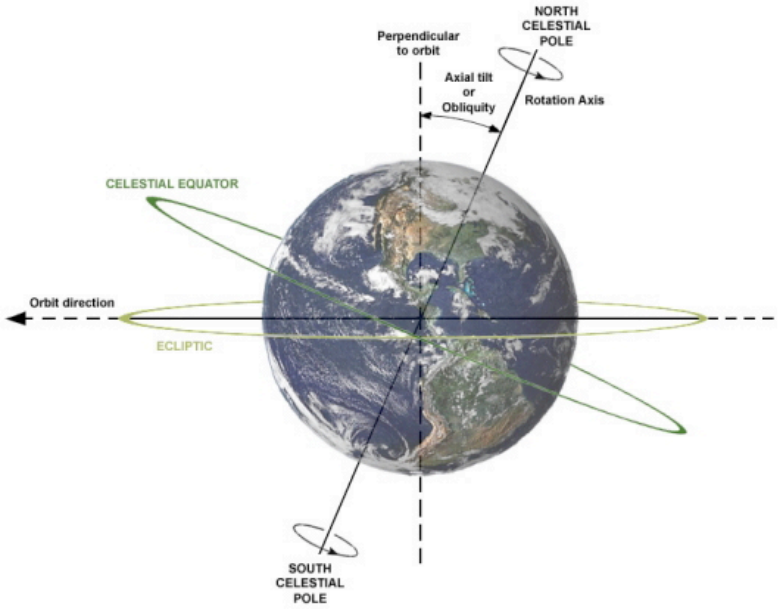

Figure C-4: The celestial equator represented on Earth. The celestial sphere is divided in the celestial equator. Reprinted from [88].

- **Celestial Equator** is the plane of the Earth's equator, as shown in Fig. C-4 projected onto the *celestial sphere*. The celestial equator is tilted 23.5 degrees in relation to the plane of the Earth's orbit known as the *ecliptic* (which is 360° or a full circle in one year, the annual motion of the Sun). The ecliptic and the celestial equator cross at two points, the *vernal equinox* and the autumnal equinox.

C.2 Two-Body Problem

The two-body problem considers two point masses in orbit about each other under the influence of their mutual gravitation. The general equation of motion of the two-body problem in Cartesian coordinates is

$$\ddot{\boldsymbol{\chi}}_{f_c}^{IPQ} = -\frac{\mu_t}{\left\|\boldsymbol{\chi}_{f_c}^{IPQ}\right\|^3}\boldsymbol{\chi}_{f_c}^{IPQ} \tag{C.1}$$

where $\boldsymbol{\chi}_{f_c}$ is the location of s/c f_c, at formation center *w.r.t.* IPQ, *i.e.*, the vector from the center of mass of the Earth to the s/c f_c and μ_t is the gravitational parameter, $\mu_t = G_{rt}(M_{Earth} + m_{s/c})$, as shown in Fig. (C-5).

Let us consider a formation with N s/c, as shown in Fig. (C-6) the formation center is the reference

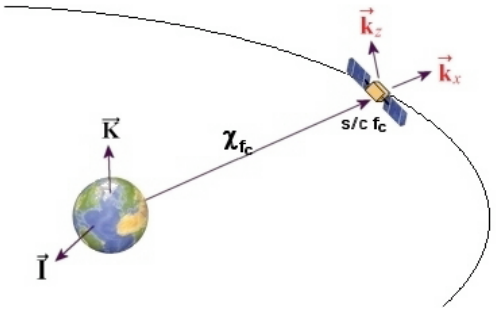

Figure C-5: Position of the s/c f_c placed in reference frame, *w.r.t.* Earth ant its IPQ frame.

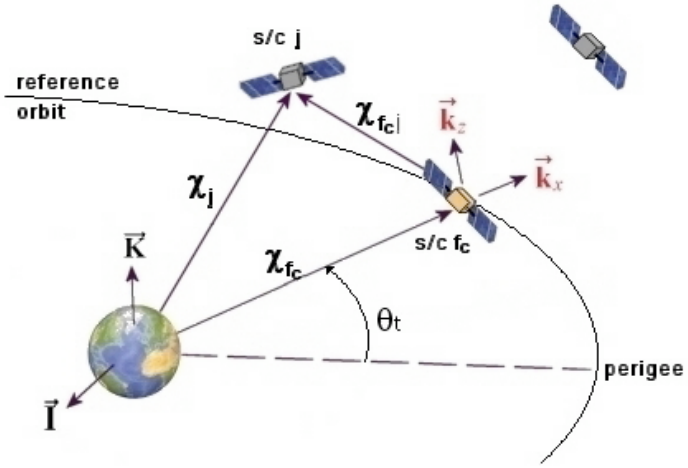

Figure C-6: relative motion of a s/c j *w.r.t.* the formation center which describes a reference orbit.

orbit frame and can be either a local point or fixed to a s/c. The s/c f_c is located in the reference orbit which is described by its orbital elements as defined in Section C.1. As for s/c f_c, in formation center, the s/c j from the formation has the following equation of motion,

$$\ddot{\boldsymbol{\chi}}_j^{IPQ} = -\frac{\mu_t}{\left\|\boldsymbol{\chi}_j^{IPQ}\right\|^3}\boldsymbol{\chi}_j^{IPQ} + \boldsymbol{\iota}_j^{IPQ} \tag{C.2}$$

where $\boldsymbol{\iota}_j$ are the sum of the perturbations $\mathbf{w}_j$ and the control inputs $\mathbf{u_j}$ acting on the j^{th} s/c, contrary to the formation center which by definition is assumed not perturbed.

C.3 Relative Motion

C.3.1 Geometric equation

From Figure (C-6) the s/c f_c, have a absolute translation vector, $\boldsymbol{\chi}_{f_c}^{IPQ}$ written in Earth Inertial frame, IPQ, and the j^{th} s/c in a formation of N s/c have relative translation vector, $\boldsymbol{\chi}_{f_c,j}^{IPQ}$ which relates to the s/c j with the s/c f_c, written in IPQ frame. The equation is given by,

$$\boldsymbol{\chi}_{j}^{IPQ} = \boldsymbol{\chi}_{fc}^{IPQ} + \boldsymbol{\chi}_{f_c,j}^{IPQ}$$

which can also be written as,

$$\ddot{\boldsymbol{\chi}}_{f_c,j}^{IPQ} = \ddot{\boldsymbol{\chi}}_{j}^{IPQ} - \ddot{\boldsymbol{\chi}}_{f_c}^{IPQ}$$

From the previous equation and Eqs. (C.2) and (C.1), we have,

$$\begin{aligned}
\ddot{\boldsymbol{\chi}}_{f_c,j}^{IPQ} &= -\frac{\mu_t}{\left\|\boldsymbol{\chi}_{j}^{IPQ}\right\|^3}\boldsymbol{\chi}_{j}^{IPQ} + \boldsymbol{\iota}_{j}^{IPQ} + \frac{\mu_t}{\left\|\boldsymbol{\chi}_{f_c}^{IPQ}\right\|^3}\boldsymbol{\chi}_{f_c}^{IPQ} \\
\ddot{\boldsymbol{\chi}}_{f_c,j}^{IPQ} &= \frac{\mu_t}{\left\|\boldsymbol{\chi}_{f_c}^{IPQ}\right\|^3}\left(-\left\|\boldsymbol{\chi}_{f_c}^{IPQ}\right\|^3 \frac{\boldsymbol{\chi}_{j}^{IPQ}}{\left\|\boldsymbol{\chi}_{j}^{IPQ}\right\|^3} + \boldsymbol{\chi}_{f_c}^{IPQ}\right) + \boldsymbol{\iota}_{j}^{IPQ} \\
\ddot{\boldsymbol{\chi}}_{f_c,j}^{IPQ} &= \frac{\mu_t}{\left\|\boldsymbol{\chi}_{f_c}^{IPQ}\right\|^3}\left(\boldsymbol{\chi}_{f_c}^{IPQ} - \left\|\boldsymbol{\chi}_{f_c}^{IPQ}\right\|^3 \frac{\boldsymbol{\chi}_{j}^{IPQ}}{\left\|\boldsymbol{\chi}_{j}^{IPQ}\right\|^3}\right) + \boldsymbol{\iota}_{j}^{IPQ}
\end{aligned} \tag{C.3}$$

Linearized model

Since $\boldsymbol{\chi}_{f_c,j}^{IPQ} = \boldsymbol{\chi}_{f_c}^{IPQ} + \boldsymbol{\chi}_{j}^{IPQ}$, and dropping the IPQ frame for sake of simplicity, we have,

$$\begin{aligned}
\frac{\boldsymbol{\chi}_j}{\left\|\boldsymbol{\chi}_j\right\|^3} &= \frac{\boldsymbol{\chi}_{f_c} + \boldsymbol{\chi}_{f_c,j}}{\left\|\boldsymbol{\chi}_{f_c} + \boldsymbol{\chi}_{f_c,j}\right\|^3} \\
\frac{\boldsymbol{\chi}_j}{\left\|\boldsymbol{\chi}_j\right\|^3} &= \frac{\boldsymbol{\chi}_{f_c} + \boldsymbol{\chi}_{f_c,j}}{\left(\left\|\boldsymbol{\chi}_{f_c}\right\|^2 + 2\boldsymbol{\chi}_{f_c}\cdot\boldsymbol{\chi}_{f_c,j} + \left\|\boldsymbol{\chi}_{f_c,j}\right\|^2\right)^{\frac{3}{2}}}
\end{aligned} \tag{C.4}$$

which may be written as follows,

$$\begin{aligned}
\frac{\boldsymbol{\chi}_j}{\left\|\boldsymbol{\chi}_j\right\|^3} &= \frac{\boldsymbol{\chi}_{f_c} + \boldsymbol{\chi}_{f_c,j}}{\left\|\boldsymbol{\chi}_{f_c}\right\|^3\left(\frac{\left\|\boldsymbol{\chi}_{f_c}\right\|^2 + 2\boldsymbol{\chi}_{f_c}\cdot\boldsymbol{\chi}_{f_c,j} + \left|\boldsymbol{\chi}_{f_c,j}\right|^2}{\left|\boldsymbol{\chi}_{f_c}\right|^2}\right)^{\frac{3}{2}}} \\
&= \frac{\boldsymbol{\chi}_{f_c} + \boldsymbol{\chi}_{f_c,j}}{\left\|\boldsymbol{\chi}_{f_c}\right\|^3\left(1 + \frac{2\boldsymbol{\chi}_{f_c}\cdot\boldsymbol{\chi}_{f_c,j}}{\left\|\boldsymbol{\chi}_{f_c}\right\|^2} + \frac{\left\|\boldsymbol{\chi}_{f_c,j}\right\|^2}{\left\|\boldsymbol{\chi}_{f_c}\right\|^2}\right)^{\frac{3}{2}}}
\end{aligned}$$

$$\frac{\boldsymbol{\chi}_j}{\left\|\boldsymbol{\chi}_j\right\|^3} = \frac{\boldsymbol{\chi}_{f_c} + \boldsymbol{\chi}_{f_c,j}}{\left\|\boldsymbol{\chi}_{f_c}\right\|^3} \left(1 + \frac{2\boldsymbol{\chi}_{f_c} \cdot \boldsymbol{\chi}_{f_c,j}}{\left\|\boldsymbol{\chi}_{f_c}\right\|^2} + \frac{\left\|\boldsymbol{\chi}_{f_c,j}\right\|^2}{\left\|\boldsymbol{\chi}_{f_c}\right\|^2}\right)^{-\frac{3}{2}} \tag{C.5}$$

Using binomial series $(1+x)^n = 1 + nx + \frac{n(n-1)}{2!}x^2 + ...$ the term $(.)^{-\frac{3}{2}}$ from the previous equation becomes,

$$\left(1 + \frac{2\boldsymbol{\chi}_{f_c} \cdot \boldsymbol{\chi}_{f_c,j}}{\left\|\boldsymbol{\chi}_{f_c}\right\|^2} + \frac{\left\|\boldsymbol{\chi}_{f_c,j}\right\|^2}{\left\|\boldsymbol{\chi}_{f_c}\right\|^2}\right)^{-\frac{3}{2}} = 1 - \frac{3}{2}\left(\frac{2\boldsymbol{\chi}_{f_c} \cdot \boldsymbol{\chi}_{f_c,j}}{\left\|\boldsymbol{\chi}_{f_c}\right\|^2} + \frac{\left\|\boldsymbol{\chi}_{f_c,j}\right\|^2}{\left\|\boldsymbol{\chi}_{f_c}\right\|^2}\right) + ... \tag{C.6}$$

Thus, Eq. (C.5) becomes,

$$\frac{\boldsymbol{\chi}_j}{\left\|\boldsymbol{\chi}_j\right\|^3} = \frac{\boldsymbol{\chi}_{f_c} + \boldsymbol{\chi}_{f_c,j}}{\left\|\boldsymbol{\chi}_{f_c}\right\|^3} \left[1 - \frac{3}{2}\left(\frac{2\boldsymbol{\chi}_{f_c} \cdot \boldsymbol{\chi}_{f_c,j}}{\left\|\boldsymbol{\chi}_{f_c}\right\|^2}\right) + \bigcirc\left(\left(\boldsymbol{\chi}_{f_c,j}\right)^2\right)\right]$$

Assuming that $\boldsymbol{\chi}_{f_c,j} << \boldsymbol{\chi}_{f_c}$, and neglecting terms of order greater than 1, the equation becomes,

$$\frac{\boldsymbol{\chi}_j}{\left\|\boldsymbol{\chi}_j\right\|^3} \approx \frac{\boldsymbol{\chi}_{f_c} + \boldsymbol{\chi}_{f_c,j}}{\left\|\boldsymbol{\chi}_{f_c}\right\|^3} \left[1 - \frac{3}{2}\left(\frac{2\boldsymbol{\chi}_{f_c} \cdot \boldsymbol{\chi}_{f_c,j}}{\left\|\boldsymbol{\chi}_{f_c}\right\|^2}\right)\right]$$

Replacing in Eq. (C.3) it follows that,

$$\begin{aligned}
\ddot{\boldsymbol{\chi}}_{f_c,j} &\approx \frac{\mu_t}{\left\|\boldsymbol{\chi}_{f_c}\right\|^3}\left(\boldsymbol{\chi}_{f_c} - \left\|\boldsymbol{\chi}_{f_c}\right\|^3 \frac{\boldsymbol{\chi}_{f_c} + \boldsymbol{\chi}_{f_c,j}}{\left\|\boldsymbol{\chi}_{f_c}\right\|^3}\left[1 - \frac{3}{2}\left(\frac{2\boldsymbol{\chi}_{f_c} \cdot \boldsymbol{\chi}_{f_c,j}}{\left\|\boldsymbol{\chi}_{f_c}\right\|^2}\right)\right]\right) + \boldsymbol{\iota}_j \\
\ddot{\boldsymbol{\chi}}_{f_c,j} &\approx \frac{\mu_t}{\left\|\boldsymbol{\chi}_{f_c}\right\|^3}\left(\boldsymbol{\chi}_{f_c} - \left(\boldsymbol{\chi}_{f_c} + \boldsymbol{\chi}_{f_c,j}\right)\left[1 - \frac{3}{2}\left(\frac{2\boldsymbol{\chi}_{f_c} \cdot \boldsymbol{\chi}_{f_c,j}}{\left\|\boldsymbol{\chi}_{f_c}\right\|^2}\right)\right]\right) + \boldsymbol{\iota}_j \\
\ddot{\boldsymbol{\chi}}_{f_c,j} &\approx \frac{\mu_t}{\left\|\boldsymbol{\chi}_{f_c}\right\|^3}\left(-\boldsymbol{\chi}_{f_c,j} + \frac{3\boldsymbol{\chi}_{f_c} \cdot \boldsymbol{\chi}_{f_c,j}}{\left\|\boldsymbol{\chi}_{f_c}\right\|^2}\boldsymbol{\chi}_{f_c} + \frac{3\boldsymbol{\chi}_{f_c} \cdot \boldsymbol{\chi}_{f_c,j}}{\left\|\boldsymbol{\chi}_{f_c}\right\|^2}\boldsymbol{\chi}_{f_c,j}\right) + \boldsymbol{\iota}_j
\end{aligned}$$

Notice that since the term $\frac{3\boldsymbol{\chi}_{f_c} \cdot \boldsymbol{\chi}_{f_c,j}}{\left\|\boldsymbol{\chi}_{f_c}\right\|^2}\boldsymbol{\chi}_{f_c,j}$ is of order greater than one, and thus neglected in previous equation,

$$\ddot{\boldsymbol{\chi}}_{f_c,j} = -\frac{\mu_t}{\left\|\boldsymbol{\chi}_{f_c}\right\|^3}\left(\boldsymbol{\chi}_{f_c,j} - \frac{3\boldsymbol{\chi}_{f_c} \cdot \boldsymbol{\chi}_{f_c,j}}{\left\|\boldsymbol{\chi}_{f_c}\right\|^2}\boldsymbol{\chi}_{f_c}\right) + \boldsymbol{\iota}_j \tag{C.7}$$

Knowing that

$$\boldsymbol{\chi}_{f_c,j} = x_j \overrightarrow{\mathbf{k}}_x + y_j \overrightarrow{\mathbf{k}}_y + z_j \overrightarrow{\mathbf{k}}_z \tag{C.8}$$

$$\boldsymbol{\chi}_{f_c} = R_{f_c} \overrightarrow{\mathbf{k}}_x \tag{C.9}$$

Using Eqs. (C.8) and (C.9) in Eq. (C.7), then,

$$\ddot{\boldsymbol{\chi}}_{f_c,j} = -\frac{\mu_t}{\left\|\boldsymbol{\chi}_{f_c}\right\|^3}\left(x_j \overrightarrow{\mathbf{k}}_x + y_j \overrightarrow{\mathbf{k}}_y + z_j \overrightarrow{\mathbf{k}}_z - \frac{3R_{f_c} \overrightarrow{\mathbf{k}}_x \cdot \left(x_j \overrightarrow{\mathbf{k}}_x + y_j \overrightarrow{\mathbf{k}}_y + z_j \overrightarrow{\mathbf{k}}_z\right)}{\left\|\boldsymbol{\chi}_{f_c}\right\|^2}\boldsymbol{\chi}_{f_c}\right) + \boldsymbol{\iota}_j$$

$$\begin{aligned}
\ddot{\boldsymbol{\chi}}_{f_c,j} &= -\frac{\mu_t}{\left\|\boldsymbol{\chi}_{f_c}\right\|^3}\left(x_j\overrightarrow{\mathbf{k}}_x + y_j\overrightarrow{\mathbf{k}}_y + z_j\overrightarrow{\mathbf{k}}_z - \frac{3R_{f_c}x_j\overrightarrow{\mathbf{k}}_x}{\left\|\boldsymbol{\chi}_{f_c}\right\|^2}R_{f_c}\overrightarrow{\mathbf{k}}_x\right) + \boldsymbol{\iota}_j \\
\ddot{\boldsymbol{\chi}}_{f_c,j} &= -\frac{\mu_t}{\left\|\boldsymbol{\chi}_{f_c}\right\|^3}\left(x_j\overrightarrow{\mathbf{k}}_x + y_j\overrightarrow{\mathbf{k}}_y + z_j\overrightarrow{\mathbf{k}}_z - 3x_j\overrightarrow{\mathbf{k}}_x\right) + \boldsymbol{\iota}_j \\
\ddot{\boldsymbol{\chi}}_{f_c,j} &= \frac{\mu_t}{\left\|\boldsymbol{\chi}_{f_c}\right\|^3}\left(2x_j\overrightarrow{\mathbf{k}}_x - y_j\overrightarrow{\mathbf{k}}_y - z_j\overrightarrow{\mathbf{k}}_z\right) + \boldsymbol{\iota}_j
\end{aligned} \tag{C.10}$$

Using fundamental orbit mechanics describing planetary motion, as cited by [79], the radius and angular velocity of the formation center can be written as,

$$\left\|\boldsymbol{\chi}_{f_c}\right\| = \frac{a_{smx}\left(1-e_{cc}^2\right)}{1+e_{cc}\cos\theta_t}$$

where a_{smx} corresponds to the semi-major axis, e_{cc} to the orbit eccentricity, $\dot{\theta}_t = \frac{n_t(1+e_{cc}\cos\theta_t)^2}{(1-e_{cc}^2)^{\frac{3}{2}}}$ and $n_t = \sqrt{\frac{\mu_t}{a_{smx}^3}}$ is the natural frequency,μ_t stands for the product between the universal gravitation constant G_{rt} and the Earth's mass, m_{Earth}, $\mu_t = G_{rt}.m_{\text{Earth}} = 3.9860044.10^5\left[\frac{\text{km}^3}{s^2}\right]$. Thus,

$$\begin{aligned}
\frac{\mu_t}{\left\|\boldsymbol{\chi}_{f_c,j}\right\|^3} &= \frac{\mu_t\left(1+e_{cc}\cos\theta_t\right)^3}{a_t^3\left(1-e_{cc}^2\right)^3} \\
&= n_t^2\left(\frac{1+e_{cc}\cos\theta_t}{1-e_{cc}^2}\right)^3
\end{aligned} \tag{C.11}$$

and Eq. (C.10) becomes,

$$\ddot{\boldsymbol{\chi}}_{f_c,j}^{IPQ} = n_t^2\left(\frac{1+e_{cc}\cos\theta_t}{1-e_{cc}^2}\right)^3\left(+2x_j\overrightarrow{\mathbf{k}}_x - y_j\overrightarrow{\mathbf{k}}_y - z_j\overrightarrow{\mathbf{k}}_z\right) + \boldsymbol{\iota}_j^{IPQ} \tag{C.12}$$

Instead of having the relative translation vector, $\boldsymbol{\chi}_{f_c,j}^{IPQ}$ written in IPQ frame, would be preferable to have it written in the formation center or formation orbit reference frame, which is a rotating frame. Therefore, we will use the Hamiltonian equations of motion in order to obtain an expression that relates the relative translation vector, $\boldsymbol{\chi}_{f_c,j}^{IPQ}$ written in formation center.

C.3.2 Clohessy-Wiltshire (Hill's) equations, time dependent

The equation that relates the relative acceleration between j^{th} s/c and s/c i (at formation center) written in IPQ, $\ddot{\boldsymbol{\chi}}_{f_c,j}^{IPQ}$, and the acceleration written in formation center frame, $\ddot{\boldsymbol{\chi}}_{f_c,j}^{f_c}$, which has translational motion and rotational motion is given the general motion of a rigid body rotating about a reference frame with translational motion (Newton's law) and rotational motion (Euler's law), as shown in Fig. C-6, and as follows,

$$\ddot{\boldsymbol{\chi}}^{IPQ}_{f_c,j} = \ddot{\boldsymbol{\chi}}^{f_c}_{f_c,j} + 2\dot{\Theta}^{IPQ} \times \boldsymbol{\chi}^{f_c}_{f_c,j} + \dot{\Theta}^{IPQ} \times \left(\dot{\Theta}^{IPQ} \times \boldsymbol{\chi}^{f_c}_{f_c,j}\right) + \left(\ddot{\Theta}^{IPQ} \times \boldsymbol{\chi}^{f_c}_{f_c,j}\right) \tag{C.13}$$

where,

$\dot{\Theta}^{IPQ}$ is the angular velocity of moving frame (formation center),

$\ddot{\Theta}^{IPQ}$ is the acceleration of moving frame (formation center),

$2\dot{\Theta}^{IPQ} \times \boldsymbol{\chi}^{f_c}_{f_c,j}$ is the Coriolis acceleration due to the motion of the s/c in the moving frame (formation center),

$\dot{\Theta}^{IPQ} \times \left(\dot{\Theta}^{IPQ} \times \boldsymbol{\chi}^{f_c}_{f_c,j}\right)$ is the centrifugal acceleration due to the angle between $\dot{\Theta}^{IPQ}$, $\boldsymbol{\chi}^{f_c}_{f_c,j}$, and the term $\left(\ddot{\Theta}^{IPQ} \times \boldsymbol{\chi}^{f_c}_{f_c,j}\right)$ is the acceleration of the s/c due to $\ddot{\Theta}^{IPQ}$ changing.

Using (C.8), and knowing that

$$\dot{\Theta}^{IPQ} = \dot{\theta}_t \overrightarrow{\mathbf{k}}_z$$

then the terms in Eq. (C.13) are,

$$\begin{aligned}
\dot{\Theta}^{IPQ} \times \boldsymbol{\chi}^{f_c}_{f_c,j} &= \begin{bmatrix} 0 & -\dot{\theta}_t & 0 \\ \dot{\theta}_t & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \\
\ddot{\Theta}^{IPQ} \times \boldsymbol{\chi}^{f_c}_{f_c,j} &= \begin{bmatrix} 0 & -\ddot{\theta}_t & 0 \\ \ddot{\theta}_t & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \\
\dot{\Theta}^{IPQ} \times \left(\dot{\Theta}^{IPQ} \times \boldsymbol{\chi}^{f_c}_{f_c,j}\right) &= \begin{bmatrix} -\dot{\theta}_t^2 & 0 & 0 \\ 0 & -\dot{\theta}_t^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}
\end{aligned}$$

with θ_t corresponding to the true anomaly of an elliptic orbit. Then Eq. (C.13) becomes,

$$\begin{aligned}
\frac{d}{dt} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}^{f_c}_{f_c,j} &= \ddot{\boldsymbol{\chi}}^{IPQ}_{f_c,j} - 2 \begin{bmatrix} 0 & -\dot{\theta}_t & 0 \\ \dot{\theta}_t & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}^{f_c}_{f_c,j} \\
&\quad - \begin{bmatrix} -\dot{\theta}_t^2 & 0 & 0 \\ 0 & -\dot{\theta}_t^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}^{f_c}_{f_c,j} - \begin{bmatrix} 0 & -\ddot{\theta}_t & 0 \\ \ddot{\theta}_t & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}^{f_c}_{f_c,j}
\end{aligned} \tag{C.14}$$

Using the result from Eq. (C.12), the previous equation becomes,

$$\frac{d}{dt}\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{bmatrix} = n_t^2\left(\frac{1+e_{cc}\cos\theta_t}{1-e_{cc}^2}\right)^3\begin{bmatrix}2x\\ -y\\ -z\end{bmatrix}+\begin{bmatrix}\iota_x\\ \iota_y\\ \iota_z\end{bmatrix}-2\begin{bmatrix}0 & -\dot{\theta}_t & 0\\ \dot{\theta}_t & 0 & 0\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{bmatrix}+ \\ -\begin{bmatrix}-\dot{\theta}_t^2 & 0 & 0\\ 0 & -\dot{\theta}_t^2 & 0\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}x\\ y\\ z\end{bmatrix}-\begin{bmatrix}0 & -\ddot{\theta}_t & 0\\ \ddot{\theta}_t & 0 & 0\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}x\\ y\\ z\end{bmatrix} \tag{C.15}$$

Knowing that $n_t^2\left(\frac{1+e_{cc}\cos\theta_t}{1-e_{cc}^2}\right)^3 = \frac{\dot{\theta}_t^2}{1+e_{cc}\cos\theta_t}$ and rearranging the terms taking into account the position and velocity, the previous equation may be written as,

$$\frac{d}{dt}\begin{bmatrix}x\\ y\\ z\\ \dot{x}\\ \dot{y}\\ \dot{z}\end{bmatrix}_{f_c,j} = \begin{bmatrix}0 & 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 0 & 1\\ \dot{\theta}_t^2+\frac{2\dot{\theta}_t^2}{1+e_{cc}\cos\theta_t} & \ddot{\theta}_t & 0 & 0 & 2\dot{\theta}_t & 0\\ -\ddot{\theta}_t & \dot{\theta}_t^2-\frac{\dot{\theta}_t^2}{1+e_{cc}\cos\theta_t} & 0 & -2\dot{\theta}_t & 0 & 0\\ 0 & 0 & \frac{-\dot{\theta}_t^2}{1+e_{cc}\cos\theta_t} & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}x\\ y\\ z\\ \dot{x}\\ \dot{y}\\ \dot{z}\end{bmatrix}_{f_c,j}+\begin{bmatrix}\iota_x\\ \iota_y\\ \iota_z\end{bmatrix}_j^{IPQ} \tag{C.16}$$

also know as the Clohessy-Wiltshire or Hill's equations. These set of equations describe the problem in a first order approximation of relative motion with respect to the moving reference frame. The equations are derived assuming a quasi circular reference orbit, and that s/c relative position errors within the formation are small compared to the orbit radius and with Earth being spherical. These assumptions introduce modeling errors. So for a model to be valid in all type of relative motion orbits, it must include all the nonlinearities.

C.3.3 Clohessy-Wiltshire (Hill's) equations, true anomaly dependent

Time evolution of the true anomaly

From Kepler's law, the true anomaly θ_t of the elliptic orbit grows with time, *i.e.,* the reference orbit increases monotonically with time t. This provides a natural basis for parameterizing the formation time and motion [10]. For elliptic orbits, the equation relating the time $t-t_p$ directly with the true anomaly is [51, p.22]:

$$t-t_p = \frac{1}{n_t}\left[2\arctan\left(\sqrt{\frac{1-e_{cc}}{1+e_{cc}}}\tan\frac{\theta_t}{2}\right)-\frac{e\sqrt{1-e_{cc}^2}\sin\theta_t}{1+e_{cc}\cos\theta_t}\right]$$

where t_p is the passage time at the Perigee. By deriving this equation, we obtain:

$$\frac{dt}{d\theta_t} = \frac{\sqrt{1-e_{cc}^2}}{n_t}\left[\frac{1}{(1+e_{cc})\cos^2\frac{\theta_t}{2}+(1-e_{cc})\sin^2\frac{\theta_t}{2}} - \frac{e_{cc}(e_{cc}+\cos\theta_t)}{(1+e_{cc}\cos\theta_t)^2}\right]$$

Using the true anomaly as free variable instead time, and using the relationships,

$$\begin{aligned}(\dot{.}) &= (.)^{'}\dot{\theta}_t &\text{(C.17)}\\ (\ddot{.}) &= (.)^{''}\dot{\theta}_t^2 + \dot{\theta}_t\dot{\theta}_t^{'}(.)^{'} &\text{(C.18)}\end{aligned}$$

as well as the fact that $\dot{\theta}_t = \frac{n_t(1+e_{cc}\cos\theta_t)^2}{(1-e_{cc}^2)^{\frac{3}{2}}}, \dot{\theta}_t^2 = \frac{n_t^2(1+e_{cc}\cos\theta_t)^4}{(1-e_{cc}^2)^3}$, then we have,

$$\begin{aligned}\dot{x} &= x^{'}\dot{\theta}_t\\ \ddot{x} &= x^{''}\dot{\theta}_t^2 + \dot{\theta}_t\dot{\theta}_t^{'}x^{'}\end{aligned}$$

The differential equations of motion will be expressed with respect to the true anomaly θ_t which become,

$$\begin{aligned}\dot{\theta}_t x^{'} &= \dot{\theta}_t x^{'}\\ \dot{\theta}_t y^{'} &= \dot{\theta}_t y^{'}\\ \dot{\theta}_t z^{'} &= \dot{\theta}_t z^{'}\\ \dot{\theta}_t^2 x^{''} + \dot{\theta}_t\dot{\theta}_t^{'}x^{'} &= \dot{\theta}_t^2\left[1+\frac{2}{1+e_{cc}\cos\theta_t}\right]x + \ddot{\theta}_t y + +2\dot{\theta}_t^2 y^{'} + \iota_x\\ \dot{\theta}_t^2 y^{''} + \dot{\theta}_t\dot{\theta}_t^{'}y^{'} &= -\ddot{\theta}_t x + \dot{\theta}_t^2\left[1-\frac{1}{1+e_{cc}\cos\theta_t}\right]y - 2\dot{\theta}_t^2 x^{'} + \iota_y\\ \dot{\theta}_t^2 z^{''} + \dot{\theta}_t\dot{\theta}_t^{'}z^{'} &= \frac{-\dot{\theta}_t^2}{1+e_{cc}\cos\theta_t}z + \iota_z\end{aligned}$$

which solving,

$$\begin{aligned}x^{'} &= x^{'}\\ y^{'} &= y^{'}\\ z^{'} &= z^{'}\\ x^{''} &= \left[\frac{3+e_{cc}\cos\theta_t}{1+e_{cc}\cos\theta_t}\right]x + \frac{\ddot{\theta}_t}{\dot{\theta}_t^2}y + \frac{-\dot{\theta}_t^{'}}{\dot{\theta}_t}x^{'} + 2y^{'} + \frac{1}{\dot{\theta}_t^2}\iota_x\\ y^{''} &= -\frac{\ddot{\theta}_t}{\dot{\theta}_t^2}x + \left[\frac{e_{cc}\cos\theta_t}{1+e_{cc}\cos\theta_t}\right]y - 2x^{'} + \frac{-\dot{\theta}_t^{'}}{\dot{\theta}_t}y^{'} + \frac{1}{\dot{\theta}_t^2}\iota_y\end{aligned}$$

$$z^{''} = -\frac{1}{1+e_{cc}\cos\theta_t}z + \frac{-\dot{\theta}_t^{'}}{\dot{\theta}_t}z^{'} + \frac{1}{\dot{\theta}_t^2}\iota_z$$

and since, $\frac{\dot{\theta}_t^{'}}{\dot{\theta}_t} = \frac{-2e_{cc}\sin\theta_t}{1+e_{cc}\cos\theta_t}$, $\frac{\ddot{\theta}_t}{\dot{\theta}_t^2} = \frac{-2e_{cc}\sin\theta_t}{1+e_{cc}\cos\theta_t}$, $\frac{1}{\dot{\theta}_t^2} = \frac{\left(1-e_{cc}^2\right)^3}{n_t^2(1+e_{cc}\cos\theta_t)^4}$ and $\iota_y = \mathbf{w}_j - \mathbf{u}_j$ then,

$$\frac{d}{d\theta_t}\begin{bmatrix} x \\ y \\ z \\ x^{'} \\ y^{'} \\ z^{'} \end{bmatrix}_{f_c,j}^{f_c} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{3+e_{cc}\cos\theta_t}{1+e_{cc}\cos\theta_t} & \frac{-2e_{cc}\sin\theta_t}{1+e_{cc}\cos\theta_t} & 0 & \frac{2e_{cc}\sin\theta_t}{1+e_{cc}\cos\theta_t} & 2 & 0 \\ \frac{2e_{cc}\sin\theta_t}{1+e_{cc}\cos\theta_t} & \frac{e_{cc}\cos\theta_t}{1+e_{cc}\cos\theta_t} & 0 & -2 & \frac{2e_{cc}\sin\theta_t}{1+e_{cc}\cos\theta_t} & 0 \\ 0 & 0 & \frac{-1}{1+e_{cc}\cos\theta_t} & 0 & 0 & \frac{2e_{cc}\sin\theta_t}{1+e_{cc}\cos\theta_t} \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ x^{'} \\ y^{'} \\ z^{'} \end{bmatrix}_{f_c,j}^{f_c} +$$

$$+\frac{\left(1-e_{cc}^2\right)^3}{n_t^2\left(1+e_{cc}\cos\theta_t\right)^4}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} w_x & u_x \\ w_y & + \; u_y \\ w_z & u_z \end{bmatrix}_j^{IPQ}$$

The parametrization of the relative equations of motion in true anomaly show that when the orbital perturbations or disturbing forces are assumed zero, from Eq. (C.2) $\mathbf{w}_j = 0$, there is no dependence of the s/c attitude or angular velocity explicitly. Therefore the equation of motion for rotation are decoupled from the translational relative equations of motion and consequently can be treated separately. However, if gravity gradient stabilization or drag for s/c at lower altitudes are considered, the attitude plays an important role and therefore the translational relative motion cannot be decoupled from the rotational equations of motion which are associated with the rotational state variables, quaternion and angular velocity of each s/c within the formation.

Relative equations of motion in LVLH frame

The frame used in this work for the reference orbit where the formation center is placed is called Local Vertical Local Horizon (LVLH) frame, defined in subsection (6.3.1). The LVLH has its origin located in the formation center so that the versor $\overrightarrow{\mathbf{e}}_y$ is normal to the orbital plane, opposite the angular momentum vector of the reference orbit, $\overrightarrow{\mathbf{e}}_z$ points in the nadir direction, that is, towards the Earth mass center and the versor $\overrightarrow{\mathbf{e}}_x$ form an orthogonal triad, as shown in Fig. (C-7).

The deduction for the relative dynamic equations of motion follow [70] and [10] also use the frame

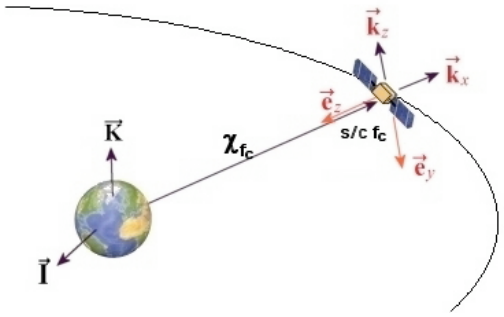

Figure C-7: Relation between LVLH frame versors $\vec{\mathbf{e}}_x$, $\vec{\mathbf{e}}_y$, $\vec{\mathbf{e}}_z$, and Tillerson's frame versors $\vec{\mathbf{k}}_x$, $\vec{\mathbf{k}}_y$, $\vec{\mathbf{k}}_z$.

used by these authors, named here as Tillerson's frame. From Fig. (C-7) we can deduct the relation between versors from Tillerson's frame and LVLH frame,

$$\vec{\mathbf{k}}_x, = -\vec{\mathbf{e}}_z,\ \vec{\mathbf{k}}_y, = -\vec{\mathbf{e}}_x,\ \vec{\mathbf{k}}_z, = -\vec{\mathbf{e}}_y$$

which leads to the following transformation matrix

$$\mathbf{A}_{LVLH}^{Tillerson} = \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

Therefore, the representation $\boldsymbol{\chi}^{LVLH}$ in the LVLH frame of the position vector $\boldsymbol{\chi}$ can be transformed into its representation $\boldsymbol{\chi}^{Tillerson}$ in Tillerson's frame according to,

$$\boldsymbol{\chi}^{Tillerson} = \mathbf{A}_{LVLH}^{Tillerson} \boldsymbol{\chi}^{LVLH}$$

Therefore, the LVLH frame used in this work is related to the frame used by Tillerson throughout the following transformation,

$$\begin{bmatrix} x^{Tillerson} \\ y^{Tillerson} \\ z^{Tillerson} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x^{LVLH} \\ y^{LVLH} \\ z^{LVLH} \end{bmatrix} \tag{C.19}$$

then the first term of the previous equation of motion is transformed as follows,

$$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix} \frac{d}{d\theta_t} \begin{bmatrix} x^{LVLH} \\ y^{LVLH} \\ z^{LVLH} \\ x'^{LVLH} \\ y'^{LVLH} \\ z'^{LVLH} \end{bmatrix} =$$

$$= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{3+e_{cc}\cos\theta_t}{1+e_{cc}\cos\theta_t} & \frac{-2e_{cc}\sin\theta_t}{1+e_{cc}\cos\theta_t} & 0 & \frac{2e_{cc}\sin\theta_t}{1+e_{cc}\cos\theta_t} & 2 & 0 \\ \frac{2e_{cc}\sin\theta_t}{1+e_{cc}\cos\theta_t} & \frac{e_{cc}\cos\theta_t}{1+e_{cc}\cos\theta_t} & 0 & -2 & \frac{2e_{cc}\sin\theta_t}{1+e_{cc}\cos\theta_t} & 0 \\ 0 & 0 & \frac{-1}{1+e_{cc}\cos\theta_t} & 0 & 0 & \frac{2e_{cc}\sin\theta_t}{1+e_{cc}\cos\theta_t} \end{bmatrix}$$
$$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x^{LVLH} \\ y^{LVLH} \\ z^{LVLH} \\ x'^{LVLH} \\ y'^{LVLH} \\ z'^{LVLH} \end{bmatrix}$$

which results in,

$$\frac{d}{d\theta_t}\begin{bmatrix} -z^{LVLH} \\ x^{LVLH} \\ -y^{LVLH} \\ -z'^{LVLH} \\ x'^{LVLH} \\ -y'^{LVLH} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{3+e_{cc}\cos\theta_t}{1+e_{cc}\cos\theta_t} & \frac{-2e_{cc}\sin\theta_t}{1+e_{cc}\cos\theta_t} & 0 & \frac{2e_{cc}\sin\theta_t}{1+e_{cc}\cos\theta_t} & 2 & 0 \\ \frac{2e_{cc}\sin\theta_t}{1+e_{cc}\cos\theta_t} & \frac{e_{cc}\cos\theta_t}{1+e_{cc}\cos\theta_t} & 0 & -2 & \frac{2e_{cc}\sin\theta_t}{1+e_{cc}\cos\theta_t} & 0 \\ 0 & 0 & \frac{-1}{1+e_{cc}\cos\theta_t} & 0 & 0 & \frac{2e_{cc}\sin\theta_t}{1+e_{cc}\cos\theta_t} \end{bmatrix} \begin{bmatrix} -z^{LVLH} \\ x^{LVLH} \\ -y^{LVLH} \\ -z'^{LVLH} \\ x'^{LVLH} \\ -y'^{LVLH} \end{bmatrix} \tag{C.20}$$

Rearranging the terms, it follows,

$$\frac{d}{d\theta_t}\begin{bmatrix} x^{LVLH} \\ y^{LVLH} \\ z^{LVLH} \\ x'^{LVLH} \\ y'^{LVLH} \\ z'^{LVLH} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{e_{cc}\cos\theta_t}{1+e_{cc}\cos\theta_t} & 0 & \frac{-2e_{cc}\sin\theta_t}{1+e_{cc}\cos\theta_t} & \frac{2e_{cc}\sin\theta_t}{1+e_{cc}\cos\theta_t} & 0 & 2 \\ 0 & \frac{-1}{1+e_{cc}\cos\theta} & 0 & 0 & \frac{2e_{cc}\sin\theta_t}{1+e_{cc}\cos\theta_t} & 0 \\ \frac{2e_{cc}\sin\theta_t}{1+e_{cc}\cos\theta_t} & 0 & \frac{3+e_{cc}\cos\theta}{1+e_{cc}\cos\theta} & -2 & 0 & \frac{2e_{cc}\sin\theta_t}{1+e_{cc}\cos\theta_t} \end{bmatrix} \begin{bmatrix} x^{LVLH} \\ y^{LVLH} \\ z^{LVLH} \\ x'^{LVLH} \\ y'^{LVLH} \\ z'^{LVLH} \end{bmatrix} + \frac{\left(1-e_{cc}^2\right)^3}{n_t^2\left(1+e_{cc}\cos\theta_t\right)^4}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix}$$

or,

$$\frac{d}{d\theta_t}\begin{bmatrix} x \\ y \\ z \\ x' \\ y' \\ z' \end{bmatrix}^{LVLH}_{LVLH,j} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ f_{41} & 0 & -f_{43} & f_{43} & 0 & 2 \\ 0 & f_{52} & 0 & 0 & f_{43} & 0 \\ f_{43} & 0 & f_{63} & -2 & 0 & f_{43} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ x' \\ y' \\ z' \end{bmatrix}^{LVLH}_{LVLH,j} + \xi(\theta_t) \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \iota_x \\ \iota_y \\ \iota_z \end{bmatrix}^{IPQ}_{j} \tag{C.21}$$

with $f_{41} = \frac{e_{cc}\cos\theta_t}{1+e_{cc}\cos\theta_t}$, $f_{43} = \frac{2e_{cc}\sin\theta_t}{1+e_{cc}\cos\theta_t}$,$f_{52} = \frac{-1}{1+e_{cc}\cos\theta_t}$, $f_{63} = \frac{3+e_{cc}\cos\theta_t}{1+e_{cc}\cos\theta_t}$, $\xi(\theta_t) = \frac{(1-e_{cc}^2)^3}{(1+e_{cc}\cos\theta_t)^4 n_t^2}$ which are the relative position linearized motion equations. For $\mathbf{w}_j = 0$, the previous equation corresponds to a linear θ_t-varying equation in a compact format,

$$\delta\mathbf{x}'_{trans}(\theta_t) = \mathbf{F}(\hat{\mathbf{x}}_{trans}(\theta_t), \theta_t)\delta\mathbf{x}_{trans}(\theta_t) + \mathbf{B}(\theta_t)\mathbf{u}(\theta_t) \tag{C.22}$$

with,

$$\mathbf{F}(\theta_t) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ f_{41} & 0 & -f_{43} & f_{43} & 0 & 2 \\ 0 & f_{52} & 0 & 0 & f_{43} & 0 \\ f_{43} & 0 & f_{63} & -2 & 0 & f_{43} \end{bmatrix}, \mathbf{B}(\theta_t) = \frac{(1-e_{cc}^2)^3}{(1+e_{cc}\cos\theta_t)^4 n_t^2} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$