Politecnico di Torino

Department of Mechanical and Aerospace Engineering (DIMEAS)

Master's Degree in Aerospace Engineering

Master Thesis

Nonlinear Estimation Algorithms for two CubeSat system

Supervisors:

Prof. Elisa Capello

Dr. Hyeongjun Park

Mr. Dario Ruggiero

Candidate:

Giovanni Stella

ID: 275540

Academic Year 2021 - 2022

Alla mia famiglia, a coloro che mi amano incondizionatamente, a coloro che mi hanno sostenuto in ogni momento della mia vita

"La famiglia è dove il cuore trova sempre una casa"

(Stephen Littleword)

"I metalli leggeri sono fruibili a fior di terra, mentre i più ricchi, la cui vena si cela in profondità, appagheranno più sostanzialmente la tenacia di chi li scava"

(Seneca)

Abstract

Technological progress changed the satellites' design over time: the modern tendency is to realize miniaturized spacecraft, best known as CubeSats, equipped by high-tech instrumentation and capable to perform transportation and objects' manipulation functions, as Unmanned Aerial Vehicles or similar on the Earth.

Because of their small size, these systems are subject to changes during all phases of the space mission, due to strong interaction with the external environment, internal torques and forces/torques generated by thrusters. The Attitude Determination and Control System (ADCS) has to be designed for reaching the desired state, exploiting information about inertia tensor elements, total mass and system's centre of mass position.

The purpose of this thesis is to estimate the mass properties of a rigid system, in which a collaborative satellite (chaser) and a non-cooperative one (target) are included, using nonlinear observer algorithms, for increasing the accuracy of the controller.

In particular, after the definition of the system dynamics, a simple orbital simulator is designed and two different estimation algorithms are performed. First, a Recursive Least Square (RLS) Method is implemented after an appropriate manipulation of the equations of motion. Then, an Adaptive Law, based on the same rearrangement of the plant, is derived and discussed in detail.

Finally, to demonstrate the effectiveness of the estimation processes, graphical and numerical results are shown after an appropriate mathematical analysis of the two defined models.

Keywords: CubeSats, ADCS, estimation, mass properties, rigid system, nonlinear observers, orbital simulator, RLS, adaptive law

Acknowledgments

With this thesis, I want to thank my supervisors Prof. Elisa Capello, Dr. Hyeongjun Park and Mr. Dario Ruggiero for this educational opportunity, aimed at the growth of my own professional career and at studying, researching and elaborating on an interesting Space Engineering sector: the development of navigation algorithms.

I thank all the professors who have contributed to improve my scientific knowledge, my skills and my mind, useful to became a good professional figure as engineer.

I am grateful to my family, who gave me a beautiful chance to make my dream come true, who supported me in the most difficult moments of my life, who gave me the courage to never give up, who taught me to fight when everything seems to be hostile. I love you.

I thank all my relatives, my friends and my colleagues who beared me in all these years, who helped make me the person I am today, with all his strengths and his weaknesses.

I am grateful to all members of Vescio-Marciano family: they were indispensable for the realization of this goal, especially during the first year of the Bachelor's Degree. Thank you for letting me join your family.

I want to dedicate a final thanks to those who are no longer here on this Earth: I feel they are next to me in every moment of my life; they give me courage in all my life choices. A piece of my heart will always be yours.

Con il presente lavoro di tesi, voglio ringraziare i miei relatori, la Prof.ssa Elisa Capello, il Dr. Hyeongjun Park e il Mr. Dario Ruggiero per avermi dato questa bellissima opportunità formativa, volta alla crescita della mia carriera professionale e allo studio, alla ricerca e all'approfondimento di un settore davvero interessante dell'Ingegneria Spaziale: lo sviluppo di algoritmi di navigazione.

Ringrazio tutti i professori che hanno contribuito a migliorare la mia conoscenza scientifica, le mie skills e la mia forma mentis, utili a diventare domani una buona figura professionale come ingegnere.

Sono molto grato alla mia famiglia, che mi ha dato la possibilità di avverare il mio sogno, che mi ha supportato nei momenti più difficili della mia vita, che mi ha dato il coraggio di non arrendermi mai, che mi ha insegnato a lottare anche quando ogni cosa sembra essere ostile. Vi amo.

Ringrazio tutti i parenti, amici e colleghi, i quali mi hanno sopportato in tutti questi anni, che hanno contribuito a rendermi la persona che sono oggi, con tutti i suoi pregi e difetti.

Sono grato a tutti i membri della famiglia Vescio-Marciano: essi sono stati indispensabili per il raggiungimento di questo traguardo, in particolar modo durante il primo anno di Laurea Triennale. Grazie per avermi accolto nella vostra famiglia.

Voglio dedicare un ultimo ringraziamento a coloro che non sono più qui sulla Terra: sento che essi mi stanno vicino in ogni momento della mia vita; mi infondono di coraggio in ogni scelta di vita. Un pezzo del mio cuore sarà sempre vostro.

Contents

List of Figures

List of Tables

Abbreviations

ADCS	Attitude Determination and Control System
AKF	Adaptive Kalman Filter
BRF	Body Reference Frame
CoM	Centre of Mass
COTS	Commercial Off-The-Shelf
EKF	Extended Kalman Filter
EVA	Extra Vehicular Activity
GNC	Guide Navigation and Control
IRF	Inertial Reference Frame
LEO	Low Earth Orbit
LVLH	Local Vertical Local Horizontal
MOS	Multiple Observer Synthesis
QFC	Quaternion Feedback Controller
RCS	Reaction Control System
R&D	Research & Development
RF	Reference Frame
RLSM	Recursive Least Square Method
RVD	Rendez-vous
RW	Reaction Wheel
S/C	Spacecraft
UAV	Unmanned Aerial Vehicle

Introduction

"*There is a driving force more powerful than stream, electricity and nuclear power: the will*" (Albert Einstein). What is science without devotion, constant research and sacrifice? Ambition and willpower are at the base of modern science and in particular they are foundation of Space Engineering: the great achievements, related to space exploration, are the results of the vision and the ability of human mind to overcome its limits.

Through the technological evolution, scientists and engineers developed "eyes" capable to observe the universe above that thin layer that separates us from Space. These eyes are artificial machines that not only orbit the Earth, but also operate in deep space.

Artificial satellites evolved a lot over time, reaching a very high technological quality inside a very small size: we are speaking about CubeSats, miniaturized satellites, generally composed by COTS components, characterized by dimensions not larger than a shoes box. Universities, Governments and scientific R&D centres developed CubeSats for different reasons:

- Costs' reduction, because of a simplification of satellite's structure and the standardization of satellite-rocket interface;
- Load unification between rockets and payloads is fundamental in case of a satellite's change;
- Introduction of a technological challenge for the present and next generations.

Because of their little size, CubeSats are really influenced by perturbations deriving from the external environment and internal disturbances, so It is necessary to implement robust and complex control law, with high accuracy levels.

According to Mohammed E.A. Cheriet et al (**[18]**) a precise control for rigid satellites is reached after good information about mass properties, such as total

system mass, inertia tensor elements and centre of mass position, subject to changes over time because of fuel sloshing and consumption, flexible appendages, etc…

Because of the recent tendency for developing cheaper and smaller spacecraft, an estimation algorithm is required, for obtaining a good control and because fault detection and isolation processes are affected by thrust properties and mass variations (**[19]**). Finally, in case of nadir pointing satellites, useful for Earth observations or similar missions, as discussed by R.E. Bordany et al (**[20]**), the ADCS subsystem is designed for reaching a good pointing capability, obtained also after a good estimation of thrusters' coefficients (thrust level and thrusters' alignment).

The modern tendency to build UAV and similar, useful for objects' transportation on Earth, has found application in recent space manipulators, indispensable for capturing non-cooperative targets, for removing debris from operational orbits, for repairing broken satellites or for servicing space stations (**[21]**).

Indeed, in recent years, human activity was responsible for accumulation of artificial material, such as pieces of old boosters of space segments or rockets, or, trivially, due to loss of instruments during EVA performed by astronauts.

The *Kessler syndrome*[1] indicates the consequence of this accumulation as a chain reaction of collisions, due to the high debris density around the Earth. In order to mitigate the formation of other debris many procedures had been proposed, such as the passivation of the spent stages released by rockets during ascent phase, for reducing the possible explosions, that can generate other artificial materials, or, for example, the use of a cooperative satellite that, after a grasping process and an orbital manoeuvre, the non-cooperative debris is positioned on a parking orbit; alternatively, if both satellites operate in LEO, after a rendezvous manoeuvre in which the chaser reaches the target, using the residual fuel of the first one, the connected system, after another manoeuvre, decays in atmosphere.

[1] Donald Kessler is an American scientist (NASA) who started to study the evolution of space debris.

In this context, the objective of this thesis is to implement an estimation algorithm for inertial properties identification of a rigid system composed by a collaborative satellite (chaser) and a non-cooperative one (target).

According to Q. Leboutet et al (**[27]**) the most used approaches to determine the mass properties of a physical system are:

- The Inverse Dynamic Identification Model with a classical Least-Square method or similar such as Recursive Least Square or Weighted Least-Square.
- Extended Kalman Filter (EKF) used not only for noise filtering problems, but also for dynamic system identification. The main problems connected to this estimator are related to its strong dependence to initial conditions and tuning parameters, such as the initial covariance, noise measurement and noise process matrices.
- Maximum Likelihood Methods.
- Closed-Loop Input / Output Error estimators.
- Neural Networks.

Finally, M. Kravis and R. Stolkin (**[22]**) in their essay explain innovative methods for estimating inertial properties of other objects when a robot gets in touch with these ones. They classify three different methods:

- *Purely visual*, based on different visual sensors, used for representing the object as a point cloud, information that will be elaborated for determining its maximum size and geometry; finally, through wave frequency It is possible to calculate the density of the object.
- *Exploratory methods,* based on an interaction between the object and the robot: this one applies a contact force on the target and, using cameras, It is possible to measure its relative motion; inertia tensor elements and mass are determined with an appropriate manipulation of the equations of motion. For these methods strong assumptions are necessary, in particular related to friction developed during contact.
- *Fixed object methods,* characterized by a rigid connection between robot and payload; the algorithms implemented are based on the classical estimators of dynamics parameters, after a rearrangements of the system's dynamics.

In this thesis project, analytical procedures similar to the proposed fixed-objects methods, described in **[22]**, are implemented. First, a RLS is designed for evaluating the mass properties for both chaser and target. Then, two Adaptive Laws are obtained using two different approaches: one is based on a parametrization of the system, expressed as function of masses, the other is derived starting from the same mathematical manipulation of RLSM.

To validate these algorithms, a simple scenario is studied: chaser and target, whose dynamics is influenced by a strong interaction with the external environment, operate in Low Earth Orbit, at 500 km of altitude.

The physical properties of these satellites are taken from **[3]**. In particular, target, a CubeSat weights 3.5 kg is rigidly grasped by another CubeSat, defined as service satellite (chaser). All the technical data are in Chapter 1. The Reaction Control System (RCS) of this one is chosen exploiting a comparison between the physical and geometrical properties of this collaborative satellite and those of other CubeSat with the same characteristics.

The Attitude Determination and Control System is designed to stabilize the total system, so that the actual quaternion is aligned to the ideal (unit) one. Exploiting all the information deriving from the kinematics expressed as function of quaternions, a simple control law, based on a Quaternion Feedback Controller, is implemented. Finally, the control input is generated by an actuation system composed by four Reaction Wheels in a pyramidal configuration. For simplicity, in this thesis the orbit determination problem is not considered, indeed, in the various simulation, the control forces are set to zero.

Finally, this work is organized as follow: in Chapter 1 the dynamics and kinematics models are discussed; in Chapter 2 a brief description of the most used algorithms for inertial parameters identification is analysed and the relative implementation, based on a rearrangement of the equations of motion, with the obtained graphical and numerical results are shown in Chapter 3. The last section shows the conclusions of this thesis project with the open questions that can be solved in future works.

Chapter 1

Dynamics and kinematics model

The purpose of this first section is to present the main elements for modelling a simple orbital simulator necessary for the definition and the implementation of an estimation process of the inertial parameters (mass, inertia and centre of mass) for a satellite system. In particular, in this chapter the technical data used for the various simulations, the mathematical model and the characterization of the space environment are discussed, not only for a system composed by two small satellites, but also for a single CubeSat, useful to validate the estimation algorithms: the unique difference between the two study cases consists in the plant (equations of motion), while the other elements of the discussed simulator are not subject to changes.

1.1. Spacecraft's Technical Data

The general properties of the analysed spacecraft are reported in the following tables. In particular, Table 1 shows all the technical requirements of a 6U CubeSat, in terms of total mass, maximum size, geometry and Reaction Control System, while Table 2 reports some information, such as geometry and mass, about two Small Satellites, adopted for running the various simulations.

Table 1. Technical data of 6U CubeSats (**[2]**)

PARAMETER	VALUE	UNIT OF MEASURE
Total mass	12	kg
Size	30x10x20	cm x cm x cm
REACTION WHEELS		
Number	4 Pyramidal configuration	/
Max torque	2	mN
Max angular momentum	30	mNms
REACTION CONTROL SYSTEM		
Thrust	1	mN
Max time on	25	s
Isp	60	s

Table 2. Technical data of two Small Satellites (**[3]**)

PARAMETER	VALUE	UNIT OF MEASURE
Service Satellite (chaser)		
Mass	10	kg
Dimensions	0.6 x 0.6 x 0.6	m x m x m
Target		
Mass	3.5	kg
Dimensions	0.4 x 0.4 x 0.4	m x m x m
Tether (connection element)		
Length	2.25	m
Diameter	5	mm
Young modulus	$4.456*10^7$	Pa

1.2. Dynamics models and main assumptions

1.2.1. Rotation dynamics model for a single CubeSat

At the base of the spacecraft dynamics model there is the Euler's equation (**[1] [28]**), that describes the satellite motion as a rigid body in terms of rotations.

The mathematical formulation, in a body reference frame, is

$$\dot{\omega}_B = J^{-1}\left(M_{ext} - M_{RW} - \omega_B \times (J\omega_B + h_{RW})\right) \tag{1.1}$$

$J \in R^{3x3}$is the inertia tensor (in this case It is a symmetric matrix), $\omega_B \in R^{3x1}$is the angular velocity vector in a BRF, $M_{RW} \in R^{3x1}$is the control vector in BRF, generated by the cluster of RWs in a pyramidal configuration, $h_{RW} \in R^{3x1}$is the angular momentum produced by RWs, $M_{ext} \in R^{3x1}$is the external disturbances vector.

In a real space simulator, another torque term[2] is responsible for motion changing: this vector occurs when, in a complex model, thrusters generate a control force, connected to variations for translational motion, whose distance from the system's CoM is not null. In our case, this value is not considered.

1.2.2. Translation dynamics for a single CubeSat

The rotation dynamics, studied in the previous section, is useful for estimating all the inertia tensor elements, because of the formulation of Euler's equation, in which the angular rate and the angular acceleration are related to this matrix.

A correct knowledge of translation dynamics can be considered for estimating the satellite's mass. In particular, for the first step of this thesis project, the position dynamics of a S/C is derived starting from the physical

[2] $M_{th} = \sum_{i=1}^{N_{th}}(F_{th} x l_i)$

laws for a simple rendezvous manoeuvre, in which the chaser has to complete far RVD manoeuvre, realizing a Hohmann transfer (**[2]**).

As proposed in **[6]**, after the definition of the IRF, centred in the Earth's CoM, and obtaining the target and chaser motion laws in base of the two body problem, the relative motion is linearized with a Taylor expansion and the first order term is obtained using the Jacobian matrix. The final equations are indispensable to represent the chaser motion in a Local Vertical Local Horizontal (LVLH) reference frame, centred in target CoM. The Hill's equations are formulated only if same assumptions are respected:

- The chaser and the target orbits are circular.
- The distance between the two satellites is lower than the orbital radius.
- The reference frame is LVLH.

The mathematical description of the Hill's equations is

$$\ddot{x} = \frac{1}{m_c} F_x + 2\omega\dot{z}$$

$$\ddot{y} = \frac{1}{m_c} F_y - \omega^2 y$$

$$\ddot{z} = \frac{1}{m_c} F_z - 2\omega\dot{x} + 3\omega^2 z \tag{1.2}$$

m_c is the chaser's mass, $F = \{F_x\ F_y\ F_z\}^T$ is the force vector that includes not only the forces from thrusters, but also the external disturbances, $\omega = \sqrt{\frac{\mu}{r^3}}$ is the orbital rate, where μ is the Earth gravitational parameter ($398600\,\frac{km^3}{s^2}$), r is the orbit radius.

The Hohmann transfer is the most efficient manoeuvre, if It is realized between two coplanar orbits, and It is an ellipse tangent to both. For simplicity, because of the low accuracy of the first phase of a far RVD, the transfer is

treated as an impulsive open-loop manoeuvre (in a real case It is a closed-loop one).

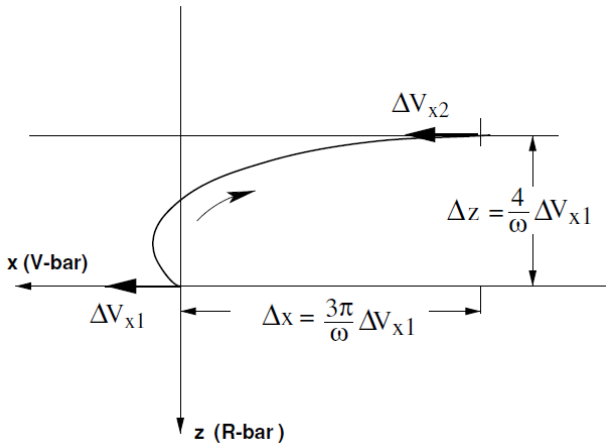

Figure 1. Example of far RVD manoeuvre (image taken from **[6]**)

With the hypothesis of an ideal case in which It could be possible to use the maximum thrusters' force, the purpose is to evaluate the time on, using the distance between the chaser and the target for the impulse determination in terms of speed variation. In symbols

$$\Delta v_{x1} = \Delta v_{x2} = \frac{\omega}{4}\Delta z$$

$$F_{x1} = F_{x2} = m_c \frac{\Delta v_x}{\Delta t} \leftrightarrow \Delta t = m_c \frac{\Delta v_x}{F_x} \tag{1.3}$$

1.2.3. Dynamics model for two connected CubeSats

After a brief description of the mathematical model for a single CubeSat, in this sub-section a new model, with its relative assumptions, for our study case is discussed.

The total system is composed by a cooperative satellite (chaser), influenced by external disturbances and forces/torques generated by actuation system, and by a non-cooperative satellite (target), whose changes in position and attitude depend on external disturbances and forces/torques

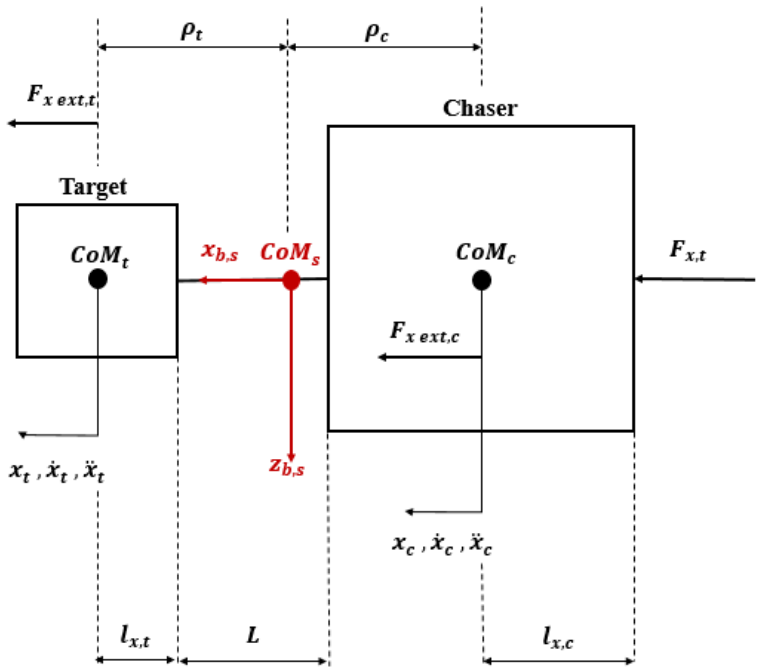

Figure 2. Chaser – Target system

produced by RCS and RWs of chaser; finally, a massless element, aligned along the direction vector of the two CoMs, connects the two rigid bodies.

For the derivation of the equations of motion, we consider Figure 2 (for simplicity, only the components along the x- body direction are shown).

For studying the total system, some preliminary steps and definitions are necessary. For a given system, composed by N material points, in which every point is characterized by a total mass m_i and position vector r_i, the total system's CoM is a position vector r_c, whose components are expressed by the following mathematical relations (**[4]**):

$$x_c = \frac{\sum_{i=1}^{N} m_i\, x_i}{\sum_{i=1}^{N} m_i}$$

$$y_c = \frac{\sum_{i=1}^{N} m_i\, y_i}{\sum_{i=1}^{N} m_i}$$

$$z_c = \frac{\sum_{i=1}^{N} m_i\, z_i}{\sum_{i=1}^{N} m_i} \tag{1.4}$$

Every position vector r_i is defined only if a preliminary RF is chosen; for this reason, as first step, we suppose to fix the origin of this reference frame at the chaser's CoM, the components of r_i are computed and then, knowing the total system's CoM, the RF is "transported" in this point. Finally, the distances from total CoM to chaser/target ones ρ_i are determined.

Since at the beginning of a space project, a mission design is required to reach, successfully, the final objective (in this case the capture of the target), using the maximum size of the target and chaser, It is possible to calculate the inertia tensor of the two rigid bodies, adopting, as first assumptions, these relations (**[28]**):

$$J_{x,i} = \frac{1}{12}\, m_i(l_{y,i}^2 + l_{z,i}^2)$$
$$J_{y,i} = \frac{1}{12}\, m_i(l_{x,i}^2 + l_{z,i}^2)$$
$$J_{z,i} = \frac{1}{12}\, m_i(l_{x,i}^2 + l_{y,i}^2)$$
$$J_i = \begin{bmatrix} J_{x,i} & 0 & 0 \\ 0 & J_{y,i} & 0 \\ 0 & 0 & J_{z,i} \end{bmatrix} \tag{1.5}$$

$i = t, c$

Obviously, these only three components define the inertia tensor as a diagonal matrix; in general, since a S/C has three symmetry axes, the tensor is symmetric and characterized by 6 independent elements, determined with a high accuracy after a detailed CAD model, composed by the final assembly of payload and subsystems.

Considering chaser and target as two bodies rigidly connected, the translational equation (described only for a full discussion) is defined as:

$$m_t a_t + m_c a_c = F_{ext,t} + F_{ext,c} + F_{th} = F_{ext} + F_{th} \tag{1.6}$$

m_t, m_c are the target and chaser mass respectively, $a_t, a_c \in R^{3x1}$ are the linear accelerations of target and chaser respectively, $F_{ext,t}, F_{ext,c} \in R^{3x1}$ are the external disturbances acting on target and chaser respectively, $F_{th} \in R^{3x1}$ is the thruster's force acting only on the chaser.

With the hypothesis that the total external disturbances vector is applied on system's CoM and that the control forces or torques act on chaser's one, It is possible to define all the quantities respect to the system's CoM:

$$a_t = \frac{d}{dt} v_t = \frac{d}{dt}(v_s + \omega_s \times \rho_{st}) = \frac{d}{dt} v_s + \frac{d}{dt}(\omega_s \times \rho_{st})$$
$$= \dot{v_s} + \omega_s \times v_s + \frac{d}{dt}\omega_s \times \rho_{st} + \omega_s \times \frac{d}{dt}\rho_{st}$$
$$\frac{d}{dt}\rho_{st} = \dot{\rho}_{st}{}^{3} + \omega_s \times \rho_{st} = \omega_s \times \rho_{st}$$

$$a_c = \frac{d}{dt} v_c = \frac{d}{dt}(v_s + \omega_s \times \rho_{sc}) = \frac{d}{dt} v_s + \frac{d}{dt}(\omega_s \times \rho_{sc})$$
$$= \dot{v_s} + \omega_s \times v_s + \frac{d}{dt}\omega_s \times \rho_{sc} + \omega_s \times \frac{d}{dt}\rho_{sc}$$

$$\frac{d}{dt}\omega_s = \dot{\omega_s} + \omega_s \times \omega_s{}^{4} = \dot{\omega_s}$$
$$\frac{d}{dt}\rho_{sc} = \dot{\rho}_{sc} + \omega_s \times \rho_{sc} = \omega_s \times \rho_{sc}$$

Uniting all the terms, for the target we have:

$$m_t a_t = m_t[\, \dot{v_s} + \omega_s \times v_s + \dot{\omega_s} \times \rho_{st} + \omega_s \times (\omega_s \times \rho_{st})]$$
$$= m_t[\, \dot{v_s} + \omega_s \times v_s + \dot{\omega_s} \times \rho_{st} - \omega_s \times (\rho_{st} \times \omega_s)] \quad (1.7)$$

[3] If the distances from the system's CoM are constant, or better they aren't subject to variations in time, the derivative is null.
[4] The cross product between two equal vectors is null.

With the same mathematical procedure, for the chaser we have:

$$m_c a_c = m_c[\,\dot{v}_s + \omega_s \times v_s + \dot{\omega}_s \times \rho_{sc} - \omega_s \times (\rho_{sc} \times \omega_s)]$$

(1.8)

Defining the state vector as the union of the linear and angular velocities, the final equation of translational motion in matrix form is

$$A\begin{Bmatrix}\dot{v}_s \\ \dot{\omega}_s\end{Bmatrix} + B\begin{Bmatrix}v_s \\ \omega_s\end{Bmatrix} = \{F_{th}\} + \{F_{ext}\}$$

$$A = [(m_t + m_c) \quad -m_t\rho_{st}^x - m_c\rho_{sc}^x]$$
$$B = [\omega_s^x(m_t + m_c) \quad -m_t\omega_s^x\rho_{st}^x - m_c\omega_s^x\rho_{sc}^x]$$

(1.9)

The conversion from cross product to a matrix form is possible by adopting the properties of the skew matrix (or antisymmetric matrix). For example:

$$\omega_s \times v_s = \omega_s^x v_s$$
$$\omega_s^x = \begin{bmatrix} 0 & -\omega_{s,z} & \omega_{s,y} \\ \omega_{s,z} & 0 & -\omega_{s,x} \\ -\omega_{s,y} & \omega_{s,x} & 0 \end{bmatrix}$$

For the rotation dynamics

$$\frac{d}{dt}h_{total} = \frac{d}{dt}(h_t + h_c) = M_{RW} + M_{ext}$$

(1.10)

$$h_t = J_t\omega_s + \rho_{st} \times m_t v_t{}^{5} = J_t\omega_s + \rho_{st} \times m_t(v_s + \omega_s \times \rho_{st})$$

[5] According to the physical laws of rigid bodies (**[4]**), the angular momentum of a system respect to a generic fixed pole is equal to the sum of the angular momentum of a material point concentrated in the centre of mass with the velocity of the CoM (first term) and the angular momentum respect to the CoM (second term)

$$h_c = J_c\omega_s + \rho_{sc} \times m_c v_c = J_c\omega_s + \rho_{sc} \times m_c(v_s + \omega_s \times \rho_{sc})$$

$$\begin{aligned}\frac{d}{dt}h_c &= J_c\dot{\omega}_s + \omega_s \times J_c\omega_s + \frac{d}{dt}(\rho_{sc} \times m_c v_s) + \frac{d}{dt}\{\rho_{sc} \times [m_c(\omega_s \times \rho_{sc})]\} \\ &= J_c\,\dot{\omega}_s + \omega_s \times J_c\omega_s + \frac{d}{dt}\rho_{sc} \times m_c v_s + \rho_{sc} \times \frac{d}{dt}(m_c v_s) + \frac{d}{dt}\rho_{sc} \\ &\quad \times (m_c\omega_s \times \rho_{sc}) + \rho_{sc} \times \frac{d}{dt}(m_c\omega_s \times \rho_{sc}) \\ &= J_c\,\dot{\omega}_s + \omega_s \times J_c\omega_s + (\omega_s \times \rho_{sc}) \times m_c v_s + \rho_{sc} \times m_c(\dot{v}_s + \omega_s \times v_s) \\ &\quad + (\omega_s \times \rho_{sc}) \times (m_c\omega_s \times \rho_{sc})^{6} + \rho_{sc} \times [\dot{\omega}_s \times m_c\rho_{sc} + m_c\omega_s \\ &\quad \times \omega_s \times \rho_{sc}]\end{aligned}$$

Manipulating the previous equation we have:

$$\begin{aligned}\frac{d}{dt}h_c &= J_c\,\dot{\omega}_s + \omega_s \times J_c\omega_s + m_c\rho_{sc} \times \dot{v}_s + m_c\rho_{sc} \times (\omega_s \times v_s) \\ &\quad + m_c(\omega_s \times \rho_{sc}) \times v_s{}^{7} - m_c\rho_{sc} \times (\rho_{sc} \times \dot{\omega}_s) - m_c\rho_{sc} \\ &\quad \times [\omega_s \times (\rho_{sc} \times \omega_s)] \\ &= m_c\rho_{sc} \times \dot{v}_s + J_c\,\dot{\omega}_s - m_c\rho_{sc} \times (\rho_{sc} \times \dot{\omega}_s) + m_c\rho_{sc} \times (\omega_s \times v_s) \\ &\quad + \omega_s \times J_c\omega_s - m_c\rho_{sc} \times [\omega_s \times (\rho_{sc} \times \omega_s)]\end{aligned} \tag{1.11}$$

Adopting the same procedure for the target we have:

$$\begin{aligned}\frac{d}{dt}h_t &= m_t\rho_{st} \times \dot{v}_s + J_t\,\dot{\omega}_s - m_t\rho_{st} \times (\rho_{st} \times \dot{\omega}_s) + m_t\rho_{st} \times (\omega_s \times v_s) + \omega_s \\ &\quad \times J_t\omega_s - m_t\rho_{st} \times [\omega_s \times (\rho_{st} \times \omega_s)]\end{aligned} \tag{1.12}$$

For analogy, the rotational dynamics can be written in matrix form, considering the same state vector and the sum of the various term deriving from the derivative of the target and chaser angular momentum. Adopting the

[6] This term is null because is in the form $a \times (ka) = k(a \times a) = 0$.

[7] This term is null because

$$(\omega_s \times \rho_{sc}) \times v_s = v_s \times v_s = 0$$

mathematical formulation proposed in **[5]**, It is possible to use a compact form, not only for rotations, but also for the total dynamics (plant).

In conclusion:

$$M\begin{Bmatrix}\dot{v}_s\\ \dot{\omega}_s\end{Bmatrix} + C\begin{Bmatrix}v_s\\ \omega_s\end{Bmatrix} = \begin{Bmatrix}F_{th}\\ \rho_{sc}^x F_{th} - M_{RW}\end{Bmatrix} + \begin{Bmatrix}F_{ext}\\ M_{ext}\end{Bmatrix}$$

$$M = \sum_{i=t,c}\begin{bmatrix} m_i & -m_i\rho_{si}^x \\ m_i\rho_{si}^x & J_i - m_i\rho_{si}^x\rho_{si}^x\end{bmatrix}$$

$$C = \sum_{i=t,c}\begin{bmatrix} m_i\omega_s^x & -m_i\omega_s^x\rho_{si}^x \\ m_i\rho_{si}^x\omega_s^x & \omega_s^x J_i - m_i\rho_{si}^x\omega_s^x\rho_{si}^x\end{bmatrix}$$

(1.13)

1.3. Kinematics model and Quaternion Feedback Controller

To guarantee the spacecraft active stabilization, for this thesis project a Quaternion Feedback Controller is implemented (**[1] [20]**). In general, the attitude of a rigid body is expressed using the Euler's angles, but in space applications a kinematics model, based on the use of quaternions, is preferred for:

- Avoiding geometric singularities.
- Reducing the computational cost.
- For increasing the numerical stability and the effectiveness of the algorithms.

The following equation (**[1]**), expressed in matrix form, describes the evolution of quaternions, starting from the knowledge of the angular velocities in a BRF

$$\begin{Bmatrix}\dot{q}_0\\ \dot{q}_1\\ \dot{q}_2\\ \dot{q}_4\end{Bmatrix} = \dot{q} = \frac{1}{2}\begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0\end{bmatrix}\begin{Bmatrix}q_0\\ q_1\\ q_2\\ q_3\end{Bmatrix}$$

(1.14)

$q_0 \in R$ is the scalar component of quaternions, while $q_v = \{q_1\ q_2\ q_3\} \in R^{3x1}$ represent the vector component. In another way, considering these two different components

$$\dot{q}_0 = -\frac{1}{2}\omega_B\ q_v$$

$$\dot{q}_v = \frac{1}{2}\ (q_0\omega_B - \omega_B \times q_v)$$

(1.15)

Since in this chapter the translational dynamics of the single CubeSat is described in a Local Orbital Reference Frame and in particular in Local Vertical Local Horizontal (LVLH) RF and the attitude dynamics is defined in a BRF, in this section is useful to report the implementation of the coordinate transformation matrix from BRF to IRF, expressed in function of quaternions (**[1]**):

$$L_{BI} = (q_0^2 - q_v \cdot q_v)I_3 + 2q_v q_v^T - 2q_0 Q^X$$

(1.16)

$I_3 \in R^{3x3}$ is the identity matrix, $Q^X \in R^{3x3}$is the skew matrix of q_v

$$Q^X = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}$$

After the definition of the satellite kinematics, It is possible to formulate the analytical discussion of the chosen controller (QFC).

The purpose of a control law is to calculate the necessary torque or force for reaching a desired attitude or position, after a series of rotations. In general, the reaching of the desired configuration requires a defined fuel consumption, so the QFC provides a nearly optimal strategy based on a complex logic control.

Using the easy computation of quaternions and all the attitude information from on-board sensors, the general law for a closed-loop control is

$$M_B = -K_p q_e - K_d \omega_B$$

(1.17)

$M_B \in R^{3x1}$ is the control torque, q_e[8] is the quaternion error, K_p and K_d are constant matrix that depend on the final controller formulation.

As proposed in **[1]**, there are four different types of controllers, but, since the objective of this thesis project is not correlated to a detailed controller design, the simplest control law is adopted

$$K_p = kI_3$$
$$K_d = diag[9](d_1\ d_2\ d_3)$$

(1.18)

Finally, because of the ideal quaternion is equal to the desired, It is possible to affirm that the quaternion error is about the vector component of current quaternion. So the mathematical description for QFC is

$$M_B = -sign[10](q_{des,0}) K_p q_v - K_d \omega_B$$

(1.19)

[8] The quaternion error is formulated as

$$q_e = q - q_{des}$$

q is the current quaternion, q_{des} is the desired.

The previous equation is not a simple difference, but It is expressed by a quaternion product **[1]**. In matrix form

$$\begin{Bmatrix} q_{e,0} \\ q_{e,1} \\ q_{e,2} \\ q_{e,3} \end{Bmatrix} = \begin{bmatrix} q_{des,0} & -q_{des,1} & -q_{des,2} & -q_{des,3} \\ q_{des,1} & q_{des,0} & -q_{des,3} & q_{des,2} \\ q_{des,2} & q_{des,3} & q_{des,0} & -q_{des,1} \\ q_{des,3} & -q_{des,2} & q_{des,1} & q_{des,0} \end{bmatrix} \begin{Bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{Bmatrix}$$

[9] "*Diag*" stands for the MatLab command used to indicate a diagonal matrix.

[10] The function "*sign*" is fundamental to accept as possible scalar quaternion component both ±1.

1.4. Actuators model

Every satellite is equipped by actuators for active position and attitude control; they are classified in (**[2]**):

- *Reaction-type actuators*, such as thrusters and magnetic torquers, for changing angular momentum.
- *Momentum exchange devices*, such as RW and control moment gyro, for maintaining constant the overall angular momentum.

In a real spacecraft, thrusters and jets generate both force and torque and for this reason they can be used not only for position but also for attitude control. In this thesis project, and in particular for the analysis of the single CubeSat, the RCS is used only for the guidance function.

For reaching the desired attitude and to guarantee a three axis control a cluster of four RWs in a pyramidal configuration is considered: these devices, that spin along a fixed axis, are brushless motors, connected to a flywheel with a high inertia; the disadvantages of these actuators are connected to torque and momentum saturation, because they cannot provide a torque higher than design value, due to electrical and mechanic limits.

According to the technical parameters reported in the Table 1, the RWs on-board of the analysed 6U CubeSat can be modelled considering the saturation limits, as shown in the following scheme:

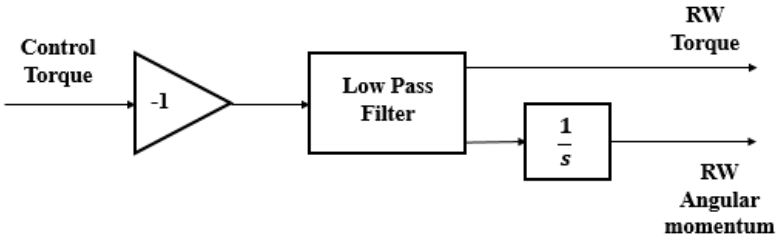

Figure 3. RW model

The equation of the Low Pass Filter is

$$\frac{k}{\tau s + 1}$$

$k, \tau \in R$.

The output of the actuator model is the necessary torque for reaching the desired state. Analysing the previous scheme, we can affirm that:

- The control torque, generated by the QFC, is expressed in BRF.
- The actuators model produces a torque in own RF.
- An actuation matrix is necessary to pass from BRF to RW RF for allocating the control torque in the RW assembly, and vice versa (from RW RF to BRF) for applying the Euler's equation.

Because the four actuators are allocated inside the spacecraft in a pyramidal configuration, the actuation matrix is defined as (**[2]**)

$$Z = \begin{bmatrix} \frac{\sqrt{3}}{2} & 0 & -\frac{\sqrt{3}}{2} & 0 \\ 0 & \frac{\sqrt{3}}{2} & 0 & -\frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$ [11]

Finally, considering all the above information, the real scheme for the torque determination as input of the Euler's equation is

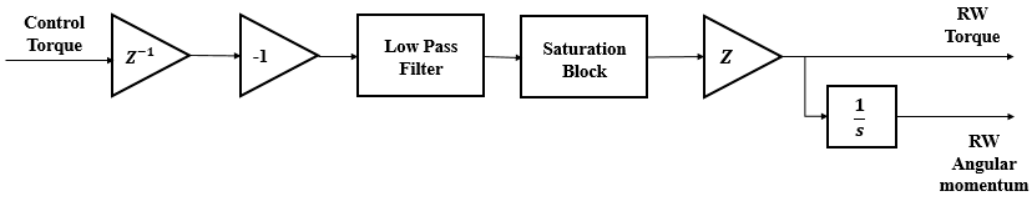

Figure 4. Block scheme for RW torque and angular momentum determination

[11] Because the matrix is not a square matrix, its inverse is defined pseudo-inverse and It can be calculated with the MATLAB command "*pinv*".

1.5. External disturbances

In addition to internal torques, that arise from the flexibility of the satellite structure or other phenomena such as propellant sloshing and uncertainty in S/C CoM, the CubeSat interacts with the external environment: the result is a series of torques and forces (cyclical or secular) responsible for variations in position and attitude. A classical CubeSat that operates in LEO, due to its small dimensions, is influenced by (**[2]**):

- **Atmospheric drag**, because of the presence of residual atmosphere. The conventional continuum theory of fluids decays and the interaction between the vehicle and the environment, due to very low density, can be considered at molecular level. Approximating this force as a constant, we have:

$$F_{atm} = \frac{1}{2}\rho v^2 S C_D \tag{1.20}$$

  ρ is the atmospheric density, S is the wet area, v is the S/C orbital velocity, C_D is the drag coefficient, depending on solar flux and altitude.

  Defining the correct motion's direction, the atmospheric drag vector is opposite to this one.

- **Solar pressure radiation**, due to solar activity. It is a sequence of photons with momentum, that produces a small pressure on the satellite surface, with relevant cumulative effect in a long period. The mathematical model of this effect is:

$$F_s = (1 + k)\frac{I_s}{c} S \tag{1.21}$$

k is the reflectivity and It can be equal to 0 in eclipse or 1 in case of illumination conditions, I_s is the solar constant (about 1367 $\frac{W}{m^2}$), c is the light speed ($3 * 10^8 \frac{m}{s}$), S is the frontal area.

This force acts on all motion's directions.

- **Gravity gradient** aligns the minimum inertia axis to the local vertical. Considering the spacecraft with a mass much smaller than the spherical principal body (Earth), the gravity gradient torque can be modelled as:

$$M_g = 3n^2 \hat{r} \times J\hat{r} \tag{1.22}$$

$n = \sqrt{\frac{\mu}{a^3}}$ is the orbital rate, where a is the major semi-axis of the orbit, μ is the Earth gravitational parameter ($398600 \frac{km^3}{s^2}$), $J \in R^{3x3}$ is the inertia tensor, $\hat{r} = \{-\sin\theta, \sin\phi, 1 - \sin^2\theta - \sin^2\phi\}^T$ is the unit vector from the planet to the satellite.

- **J2 effect**, related to the oblateness of the Earth. In a real mathematical model, the oblation on the planet's gravity is described as a mathematical series, in which every term J_k is defined as zonal coefficient; because of the J2 term is higher than others in terms of order of magnitude, in this project we consider only this perturbation, modelled as the following relation:

$$F_{J2} = m_c \frac{3J_2 \mu R_E^2}{2r^4} \tag{1.23}$$

J_2 is a constant ($1.08263 * 10^{-6}$), m_c is the S/C mass, R_E is the medium Earth radius, r is the orbit radius.

This force acts on the opposite reference motion's directions.

- **Magnetic torque**, because of the interaction with Earth's magnetic field.

In this thesis project, only the first four disturbances are considered for the implementation of a basic orbital simulator, necessary for the estimator design. In particular, the atmospheric drag, the J2 effect and the solar pressure are responsible for only position changes in translational motion, while the gravity gradient modifies only rotational dynamics. Really, this sentence is not true, because the external forces are not applied at CoM, but they can be distributed on all satellite external surface or on specific points, generating torques too.

Because of their small order of magnitude, the previous assumption is valid for our study case.

In conclusion, the total external force and torque are

$$F_{ext} = F_s + F_{atm} + F_{J2}$$

$$M_{ext} = (1 + 0.15)M_g \tag{1.24}$$

The surplus equal to 15% on the gravity gradient torque is applied for including the other perturbations on the rotational dynamics, that here are not modelled and calculated in detail.

Chapter 2

Parameters Estimation

In this chapter different algorithms used for inertial parameters estimation are discussed; in particular, after a brief analysis of the literature review, the mathematical description of the most common estimators and observers is explained step after step.

2.1. Literature review

The inertial parameters identification is fundamental for a good control of the spacecraft: different phenomena are responsible for variations in terms of mass, such as the fuel consumption after the realization of a particular manoeuvre, others are connected to inertia tensor and CoM, such as the internal torques (liquid sloshing, movements of the crew members in case of manned vehicles, etc…).

In literature, there are different studies in which these properties are analysed in detail, but are limited to UAVs or automotive. Recently, after the development of space manipulators, engineers and researchers applied the most used algorithms for inertial parameters identification to spacecraft, for increasing the accuracy and the stability of the control block, realizing a more automatized vehicle with the benefit of a great reduction of errors.

Exploiting all the results in bibliographic research, It is possible to affirm that the most used algorithms for mass and inertia estimation are:

- **Recursive Least Square Method** or similar;
- **Extended Kalman Filter** (used also for noise filtering problems);

- **Adaptive Laws**, derived from a manipulation and a rearrangement of the system's dynamics.

2.2. Recursive Least Square Method

In his thesis work, A. Eriksson (**[7]**) explained an estimation method based on an Adaptive Kalman Filter, known as RLS, for calculating the current mass of a heavy duty Scania Vehicle, starting from a manipulation of vehicle dynamics and exploiting the Newton's second law of motion.

Kim et al (**[8]**) propose an interesting estimation algorithm based on the analysis of the longitudinal and the roll dynamics: the results of the two different estimation processes are weighted adopting a Multiple Observer Synthesis (MOS).

Finally, F. Yuan et al (**[9]**) adopt the same algorithm (RLS) implemented not only for mass estimation, but also for the driving resistance; the main difference between **[8]** and **[9]** is related to a particular coefficient, called *forgetting factor* λ.

For implementing the RLS method, the first step consists in the definition and the minimization of a loss function, that depends on time and the estimated variable. Similar to the classical Least Square method, used to calculate the best solution of a linear system in the minimum square sense, the result of the derivative of the loss function can be written in the following form:

$$y(t) = \varphi^T(t)\theta(t) \tag{2.1}$$

$y(t)$ is the measured output, $\varphi(t)$ is the regressor matrix, $\theta(t)$ is the variable to be estimated. Really, for a nonlinear system, the regressor matrix is a function of time and $\theta(t)$.

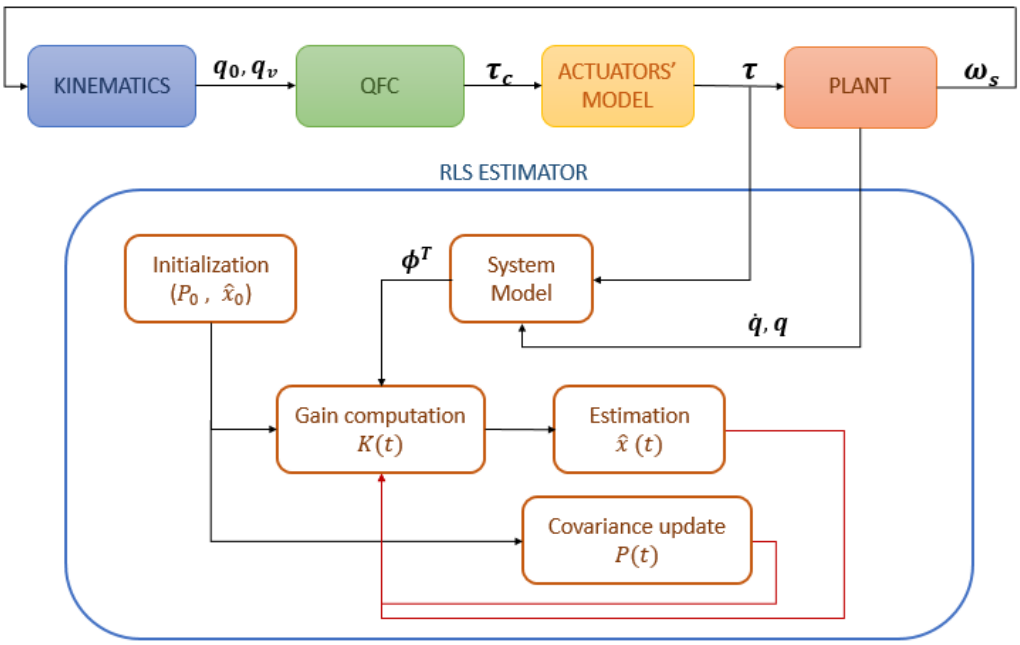

Figure 5. Block diagram of the RLSM

So, the second preliminary step is the manipulation of the plant, for determining the measured output and the regressor matrix. Now considering a generic case, in which the loss function depends also on forgetting factor, with the knowledge of all the quantities, It is possible to define the Kalman gain

$$K(t) = P(t)\varphi(t) = P(t-1)\varphi(t) * \left(\lambda I + \varphi^T(t)P(t-1)\varphi(t)\right)^{-1} \tag{2.2}$$

$P(t-1)$ is the covariance matrix calculated in the previous step (so, if we are calculating the Kalman gain at first iteration, we have to use an arbitrary value of P, necessary to initialize the filter), λ is the forgetting factor, I is the identity matrix.

The final and last steps consist in the calculation of the estimated parameter and in the updating of the covariance matrix:

$$P(t) = \left(I - K(t)\varphi^T(t)\right)P(t-1)\varphi(t)\frac{1}{\lambda}$$
$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)\left(y(t) - \varphi^T(t)\hat{\theta}(t-1)\right)$$

(2.3)

$\hat{\theta}(t-1)$ is the estimated value in the previous step.

In Figure 5, a simple block diagram of the RLS observer is shown: the filter begins the estimation process only if the user defines initial covariance and state vector; the calculation of the parameters reported in (2.3) is indispensable for initializing the next iteration (red lines of the graphic).

2.3. Extended Kalman Filter

If the scientific studies about RLS can be elaborated on not only for mass estimation, but also for inertia one, A. Bellar and Si Mohammed MA (**[10]**) propose an innovative algorithm for all inertia tensor elements identification, based on an EKF.

Generally, if the equations of motion are described in a linear form, a Kalman Filter is implemented for helping the navigation system in presence of degradation of on-board sensors, such as the star trackers. If this hypothesis is not respected, an Extended Kalman Filter is necessary. As the first, the EKF is composed by two different cycles: propagation and correction.

At the base of this observer there is the definition of the nonlinear system in the following form

$$\dot{x}(t) = f(x(t), t)$$
$$y(t) = Cx(t)$$

(2.4)

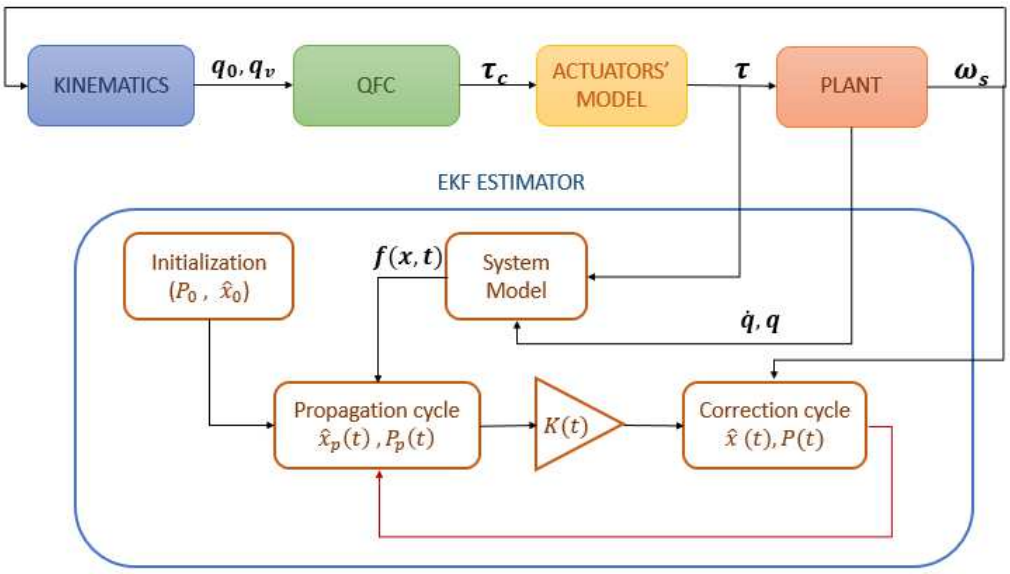

Figure 6. Block diagram of the EKF observer

$x(t)$ is the state vector to estimate, $\dot{x}(t)$ is the derivative of the state vector, f is the nonlinear function that describes the system's dynamics, expressed as a function of time and state vector.

For the propagation cycle in **[10]** the Adams integration method is applied, but for reducing the computational cost of the algorithm, It is possible to substitute a simple Explicit Euler Method to the proposed one. In other words, the main steps to be realized are

$$\hat{x}_{k,p} = \hat{x}_{k-1} + dt\ f(\hat{x}_{k-1}, t_{k-1})$$

$$P_{k,p} = \Phi_{k-1} P_{k-1} \Phi_{k-1}^T + Q_k$$

$$\Phi_{k-1} = I + dt\ F(\hat{x}_{k-1}, t_{k-1}) \tag{2.5}$$

dt is the fixed time step, Q is the process noise matrix, $\hat{x}_{k-1}$ and P_{k-1} are the estimated vector and the covariance matrix calculated in the previous

iteration, Φ is the transition matrix, I is the identity matrix, F is the Jacobian matrix, obtained as

$$F = \begin{bmatrix} \frac{\partial \dot{x}_1}{\partial x_1} & \cdots & \frac{\partial \dot{x}_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \dot{x}_m}{\partial x_1} & \cdots & \frac{\partial \dot{x}_m}{\partial x_n} \end{bmatrix}$$[12]

(2.6)

We can observe that only after the definition of this matrix, the transition matrix Φ, responsible for passing from the state k to state k+1 (or similarly from k-1 to k states) can be computed adopting an approximation based on the first order Taylor series expansion (see the set (2.5)).

If the propagation cycle produces *a priori* parameters, the correction cycle is fundamental to define *a posteriori* values of the estimated vector and the covariance matrix. At the base of this iteration there is the calculation of the optimal Kalman gain

$$K_k = P_{k,p} H_k^T \left[H_k P_{k,p} H_k^T + R \right]$$

$$\hat{x}_k = \hat{x}_{k,p} + K_k \left[z_k - H_k \hat{x}_{k,p} \right]$$

$$z_k = \omega_k$$

$$P_k = [I - K_k H_k] P_{k,p}$$

(2.7)

K_k is the Kalman gain, H_k is the observation matrix, I is the identity matrix, R is the measurement noise matrix, z_k is the measurement vector. Finally,

[12] This notation is based on the assumption that the state vector is composed by *n* variables, while the derivative of the state vector, that is equal to the function *f*, is characterized by *m* parameters. It is fundamental to pay attention to these variables: in general, they can be vector quantities.

H_k is calculated as F_k, so applying Jacobian to $y(t)$: if C is a constant matrix, $H_k \equiv C$.

A good system enjoys a particular property, called *observability*, that indicates the ability of the system to determine its own time history with a high accuracy; in other words, a dynamic system is observable only if it is capable to determine its state with the knowledge of the measured output. Mathematically, a dynamic system is observable only if the rank of the observability matrix is equal to the state vector's dimension, for each time step. This matrix, according to **[10]** is defined as:

$$OB = [H_k \;\; H_k F_k \;\; H_k F_k^2 \;\; H_k F_k^3 \ldots H_k F_k^{n-1}]^T \tag{2.8}$$

Where *n* is the state vector dimension.

2.4. Adaptive Laws

Generally, the most common used algorithms for inertial parameters identification are based on AKF, RLSM and in rare cases EKF. In all these studies, the total mass, the CoM allocation and the inertia tensor elements are constant during all the evolution of the physical phenomenon: this hypothesis is really restrictive, but It continues to remain valid for all the processes characterized by a small mass variation.

If we consider the equation (1.13), that describes the dynamics of target-chaser system in a body RF, It is easy to note an analogy with the classical dynamics of a UAV, for aerial manipulation scopes.

After a grasping process, the added mass of the unknown payload causes a change in dynamics, so the *Attitude Determination and Control System* has to evaluate the new total mass and total inertia for a good control of the vehicle.

Always with the hypothesis of constant parameters, H. Lee et al (**[11]**) designed an Augmented-Passivity controller based on an updating of inertial parameters. This process consists in a parametrization of the Newton-Euler equations in function of the unknown mass and the new allocation of the centre of mass.

N.A. Chaturvedi et al (**[14]**) elaborate a parameter identification method based on an Adaptive Control Law manipulating the classical rigid body dynamics, demonstrating the stability of the algorithm not only in case of a tracking problem characterized by a constant reference, but also when this one is a periodic signal.

J.Ahmed et al (**[15]**) proposed an alternative solution applied to the previous study case, adopting an adaptive feedback control based on the knowledge of kinematics expressed in terms of quaternions.

Exploiting all the results discussed in scientific literature and using the dynamics model of a multibody system formulated in the Chapter 1, It is possible to implement a mathematical model for defining an Adaptive Law for all inertial quantities identification. Since It is necessary a detailed analysis of the total system, the relative assumptions and the logical procedure to obtain a good estimation are treated in Chapter 3.

Chapter 3

Design and Implementation of the Algorithms

Using all the analytical results obtained in the previous chapter, It is possible to apply the studied algorithms to our study case: the inertial parameters estimation of a system composed by two satellites, modelled as two rigid bodies; in particular, the reference equation (1.13) represents the starting point for the definition of the various algorithms, that are performed with MATLAB interfaced with Simulink environment.

3.1. Numerical results

In this section, some numerical and graphic results are presented, in terms of angular velocities, quaternions and command torque, generated by RWs.

For simulating the evolution of the system's dynamics, in addition to the technical parameters reported in the Table 1, the simulation parameters are in Table 3. In particular, the gains shown in the following table are a trade-off to stabilize the total system and for reaching the desired attitude.

Table 3. Simulation parameters

PARAMETER	VALUE	UNIT OF MEASURE
Initial linear velocities	$[0\ 0\ 0]^T$	m/s
Initial angular velocities	$[0\ 0\ 0]^T$	rad/s
Chaser inertia tensor	0.15 eye(3)	$kg*m^2$
Target inertia tensor	0.0233 eye(3)	$kg*m^2$
System inertia tensor	0.1733 eye(3)	$kg*m^2$
Distance from CoM_c	$[-77/108\ 0\ 0]^T$	m
Distance from CoM_t	$[55/27\ 0\ 0]^T$	m
Proportional gain K_p	5.25 eye(3)	/
Derivative gain K_d	diag([150 150 150])	/
$\boldsymbol{\tau}$	0.01	

Knowing all these quantities, the simulation runs for 600 seconds.

The following results help us to establish that for this system the numerical values of the gain matrices are chosen correctly, because the dynamics quantities evolve over time, reaching the desired state after a time lapse of about 200 seconds.

Different simulations highlighted that both proportional and derivative gains are responsible for determining the accuracy of the controller and its own properties in terms of achieving of the desired attitude. In particular, as a classical Proportional-Integrative-Derivative (PID) controller, decreasing K_p the stabilization process becomes slower and the steady-state error (the error calculated as difference between the desired value and the actual one) increases. The derivative term K_d doesn't change a lot the steady-state error.

For example, assuming $K_p = 1.25 * eye(3)$ and $K_d = 100 * eye(3)$, the Quaternion Feedback Controller, according to actuators authority, evaluates necessary torques, reaching the desired attitude after about 500 seconds.

In general, the proposed orbital simulator is better designed, because the unit quaternion is achieved in the shortest time interval.

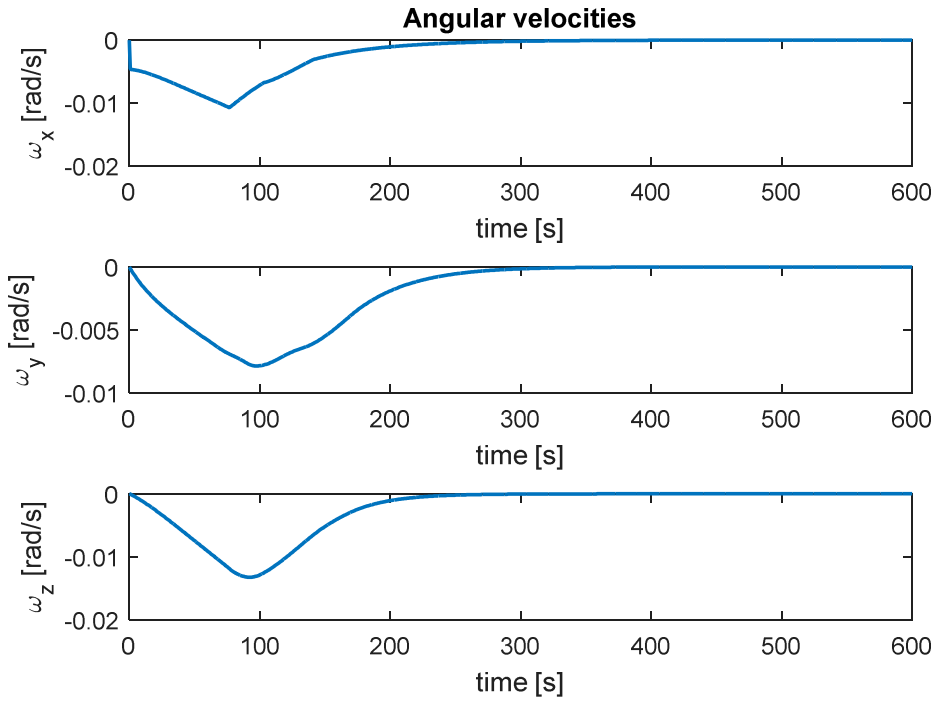

Figure 7. Angular velocities of the system

In Figure 7 the evolution of angular velocities is represented. The Attitude Determination and Control System is designed to reach the desired state. In our study case, the controller has to evaluate the necessary torques so that the desired angular velocities vector is equal to zero. According to the graphical results, this goal is achieved after about 250 seconds.

The time history of quaternions is shown in Figure 8. As discussed in Chapter 1, the QFC is implemented for evaluating the appropriate control vector, so that the system goes to the reference attitude (the ideal quaternion), in this case reached after about 200 seconds.

Finally, in Figure 9 the command torque in RW reference frame is reported. Despite they are the best actuators for controlling the spacecraft attitude, RWs suffer saturation limitations (**[1]**), as shown graphically. This depends on the absorption of external disturbances.

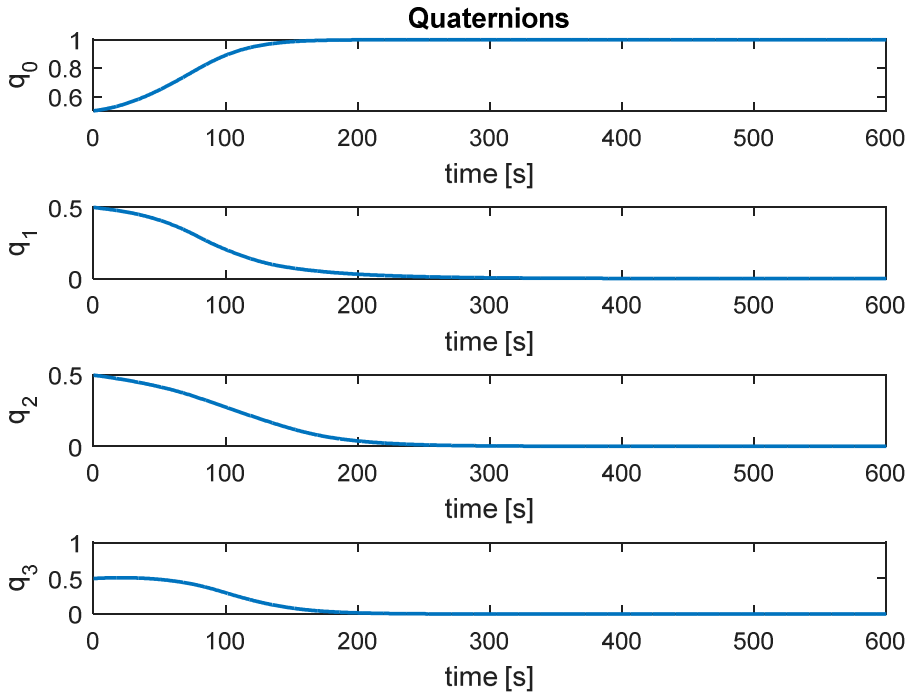

Figure 8. Quaternions' evolution

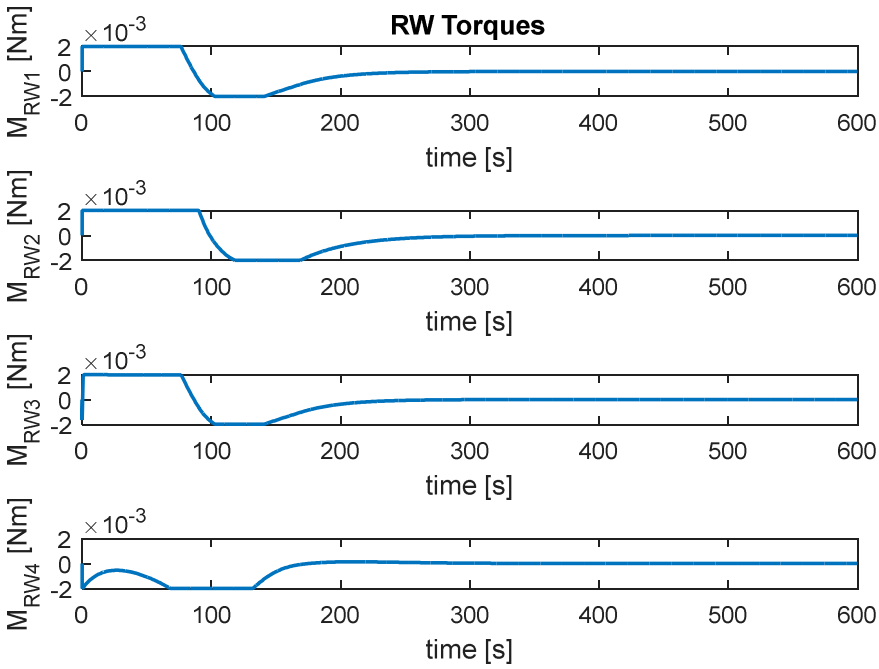

Figure 9. Command torques in RW reference frame

3.2. Recursive Least Square Method

The most common observer adopted for mass and inertia estimation is RLSM, but, in literature, It is implemented in a classical case in which quadrotors or similar systems carry an unknown payload, or when a free-flying robot grasps an object.

M. Ekal and R. Ventura (**[12]**) modelled an unknown mass grasped by a robot, by using a manipulator. After the definition of the reference frame, the equations of motion are written as functions of total mass, inertia tensor related to the robot and the offset of the centre of mass, after the connection with the payload.

In our case, both chaser and target are considered as rigid bodies, so the assumptions of the proposed mathematical model decay. However, using the same logical procedure, It is possible to formulate the dynamics problem as the classical formulation (2.1), also adopted by the authors.

We remember that the dynamics of our total system is described as:

$$M\begin{Bmatrix}\dot{v}_s\\ \dot{\omega}_s\end{Bmatrix} + C\begin{Bmatrix}v_s\\ \omega_s\end{Bmatrix} = \begin{Bmatrix}F_c\\ M_c\end{Bmatrix} + \begin{Bmatrix}F_{ext}\\ M_{ext}\end{Bmatrix}$$

$$M = \begin{bmatrix}(m_c + m_t) & -m_c\rho_{sc}^x - m_t\rho_{st}^x\\ m_c\rho_{sc}^x + m_t\rho_{st}^x & (J_c + J_t) - m_c\rho_{sc}^x\rho_{sc}^x - m_t\rho_{st}^x\rho_{st}^x\end{bmatrix}$$

$$C = \begin{bmatrix}(m_c + m_t)\omega_s^x & -m_c\omega_s^x\rho_{sc}^x - m_t\omega_s^x\rho_{st}^x\\ m_c\rho_{sc}^x\omega_s^x + m_t\rho_{st}^x\omega_s^x & \omega_s^x(J_c + J_t)^{13} - m_c\rho_{sc}^x\omega_s^x\rho_{sc}^x - m_t\rho_{st}^x\omega_s^x\rho_{st}^x\end{bmatrix} \quad (3.1)$$

$v_s \in R^{3x1}$ is linear velocities vector, $\omega_s \in R^{3x1}$ is the angular velocities vector, $F_c, M_c \in R^{3x1}$ and are forces and moments acting on chaser respectively, $F_{ext}, M_{ext} \in R^{3x1}$ are external disturbances (forces and torques) acting on the total system, m_c, m_t are chaser and target mass respectively, $J_c, J_t \in R^{3x3}$are the inertia tensor of chaser and target respectively, $\rho_{sc} \in R^{3x1}$is the distance

[13] This mathematical manipulation is possible because of the properties of matrices
$$AC + BC = (A + B)C$$

vector of the chaser's CoM from the total system's one, $\rho_{st} \in R^{3x1}$ is the distance vector of the target's CoM from the total system's one. It is fundamental to underline that M and C are expressed by $\omega_s^x, \rho_{sc}^x, \rho_{st}^x$, defined as skew matrices of the relative quantities described previously.

In general, given a vector characterized by three components, the skew matrix associated to this vector is

$$a^x = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{3.2}$$

Where $a = \{a_1\ a_2\ a_3\}^T \in R^{3x1}$

Because of the purpose is to estimate the inertial properties of the system, It is necessary to manipulate the (3.1), for applying (2.1).

We define:

$$\theta(t) = \{m_t\ m_c\ m_t\rho_{st}\ m_c\rho_{sc}\ J_{xx,s}\ J_{yy,s}\ J_{zz,s}\ J_{xy,s}\ J_{xz,s}\ J_{yz,s}\}^T \in R^{14x1}$$

$$\varphi^T(t) = \begin{bmatrix} A & A & B & B & 0_{3x6} \\ \rho_{st}^x A & \rho_{sc}^x A & \rho_{st}^x B & \rho_{sc}^x B & \dot{\Omega} + \omega_s^x \Omega^{14} \end{bmatrix} \in R^{6x14}$$

---

[14] Computing separately translation and rotation dynamics and focusing, in particular, on the last one, grouping all the terms that contain the total inertia tensor defined as:

$$J_s = J_t + J_c$$

We have:

$$J_s\ \dot{\omega}_s\ + \omega_s^x J_s \omega_s$$

We note that, formally, the previous equation is similar to (1.1). Since our purpose is to estimate all inertia tensor elements, It is possible to adopt the same results proposed in **[13]**. In particular

$$J_s\ \dot{\omega}_s\ + \omega_s^x J_s \omega_s = \left[\dot{\Omega} + \omega_s^x \Omega\right] \mathrm{J}$$

$$\Omega = \begin{bmatrix} \omega_{s,1} & 0 & 0 & \omega_{s,2} & \omega_{s,3} & 0 \\ 0 & \omega_{s,2} & 0 & \omega_{s,1} & 0 & \omega_{s,3} \\ 0 & 0 & \omega_{s,3} & 0 & \omega_{s,1} & \omega_{s,2} \end{bmatrix}$$

$$J = \{J_{xx,s}\ J_{yy,s}\ J_{zz,s}\ J_{xy,s}\ J_{xz,s}\ J_{yz,s}\}^T$$

$$A = A(t) = (\dot{v}_s + \omega_s^x v_s) \in R^{3x3}$$

$$B = B(t) = (\dot{\omega}_s^x + \omega_s^x \omega_s^x) \in R^{3x3}$$

$$u = u(t) = \begin{Bmatrix} F_c \\ M_c \end{Bmatrix} \in R^{6x1}$$

(3.3)

With this manipulation, We can rewrite (3.1) in a useful formulation for applying the classical RLSM, better known as Adaptive Kalman Filter, considering the equations (2.2) - (2.3).

Finally, initializing the filter with a good estimation of the state vector $\hat{\theta}(0)$ and a covariance matrix $P(0) = \alpha * cov(\hat{\theta}(0))$, whose numerical values are shown in Table 4, and underlining that in the future, these parameters have to be optimized, some numerical results are here reported.

Table 4. Simulation parameters for RLSM

PARAMETER	VALUE	UNIT OF MEASURE
$\widehat{\boldsymbol{m}}(\mathbf{0})$	[2.455 6.95]^T	kg
$\boldsymbol{m_t}\widehat{\boldsymbol{\rho_{st}}}(\mathbf{0})$	[4.98 1e-5 1e-5]^T	kg*m
$\boldsymbol{m_c}\widehat{\boldsymbol{\rho_{sc}}}(\mathbf{0})$	[-5.01 1e-5 1e-5]^T	kg*m
$\hat{\boldsymbol{J}}(\mathbf{0})$	[0.1205 0.1206 0.1205 1e-5 1e-5 1e-5]^T	kg*m^2
$\boldsymbol{P}(\mathbf{0})$	9.875e-4*cov($\hat{x}$(0))	/

As discussed in literature (**[27]**), the Recursive Least Square algorithm is a synthesis between a classical estimator, that determines the best solution in least square sense and a common Kalman Filter (indeed It is also known as Adaptive

Kalman Filter); really, because the system model is nonlinear, It is more correct to compare RLS to Extended Kalman Filter (EKF).

As an EKF, the proposed estimation process is strongly influenced not only by the initial conditions but also by the initial covariance matrix (**[27]**), but unlike this one, for implementing the Extended Kalman Filter It is necessary to design other two variables: the process noise and the measurement noise matrices.

Furthermore, this one requires a high computational cost, due to computation and evaluation of Jacobian matrix (**[10]**), that requires calculators with high computation power. An alternative is to define the system model in a mathematical formulation similar to state-space description and then an opportune approximation or linearization is applied (**[7]**).

In conclusion, the great advantage of RLSM consists in the evaluation of the state vector without calculating the Jacobian matrix, but the weaknesses of this numerical method are related to:

- The initialization of the filter; in general, the best solution consists in choosing an initial state vector next to the desired and then defining the covariance matrix as proposed in **[7]**. In our case, unlike this one, the addition of a corrective factor improves the convergence properties of the estimator. This correction is permitted because the choice of covariance matrix P is arbitrary.
- The high time lapse to converge to the desired state, as the classical Kalman filters.

For this reasons, other nonlinear observers are investigated, in order to improve the convergence rate.

The following graphic results (Figures 10-14) show how the estimation process evolves over time, while Table 5 reports the estimated parameters at the end of the simulation. In general, though the convergence to the desired state is achieved after about 200 seconds, analysing Table 10, that reports the evolution of the estimation error at the operating frequency (100 Hz), It is possible to establish that, under the previous assumptions, the estimation error is acceptable.

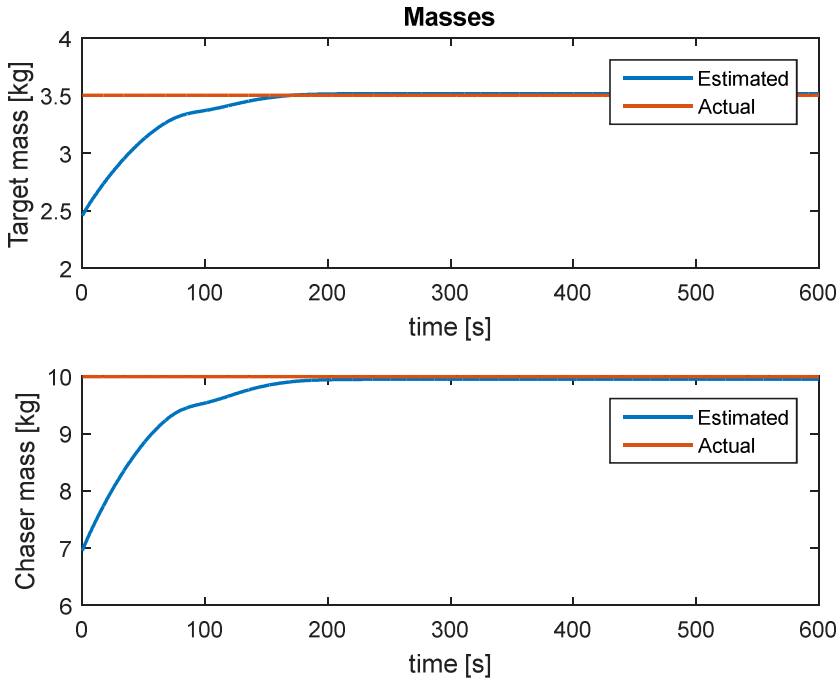

Figure 10. Mass estimation using RLSM

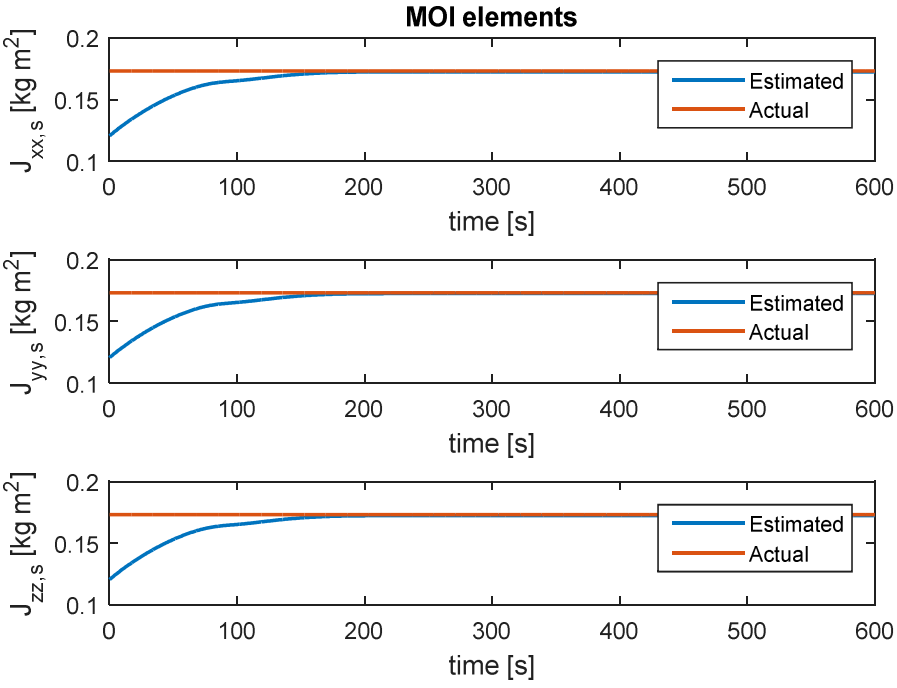

Figure 11. MOI elements estimation using RLSM

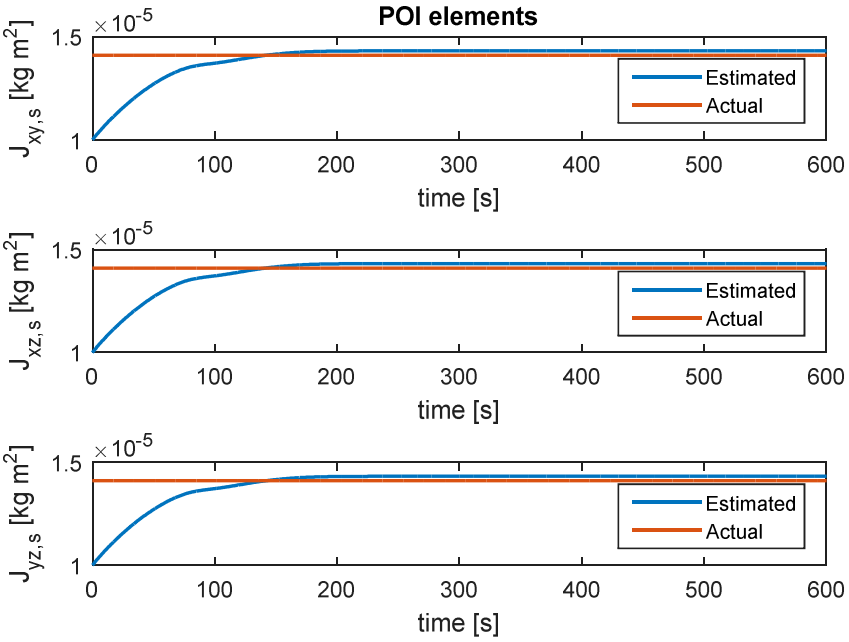

Figure 12. POI elements estimation using RLSM

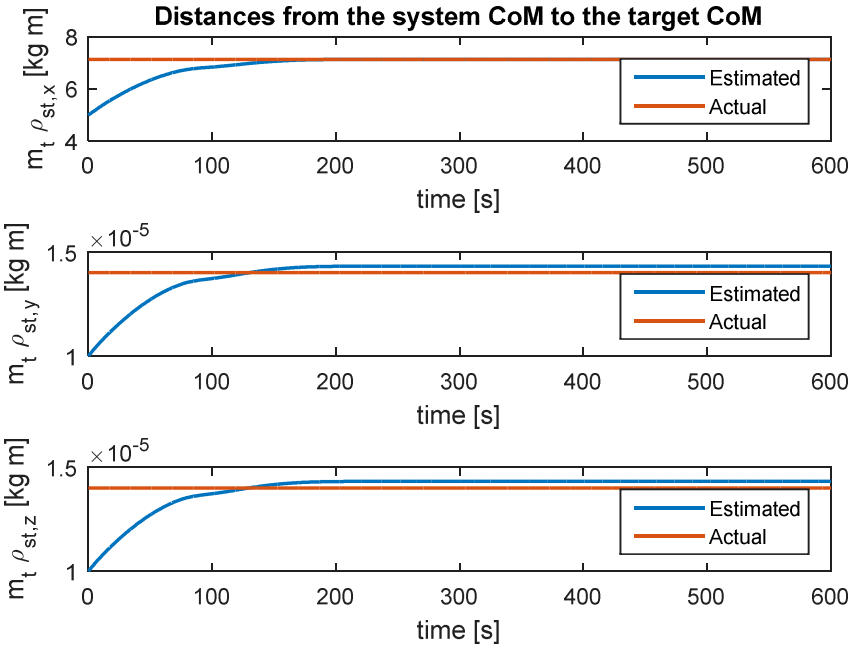

Figure 13. Distances between the system's CoM and target's one using RLSM

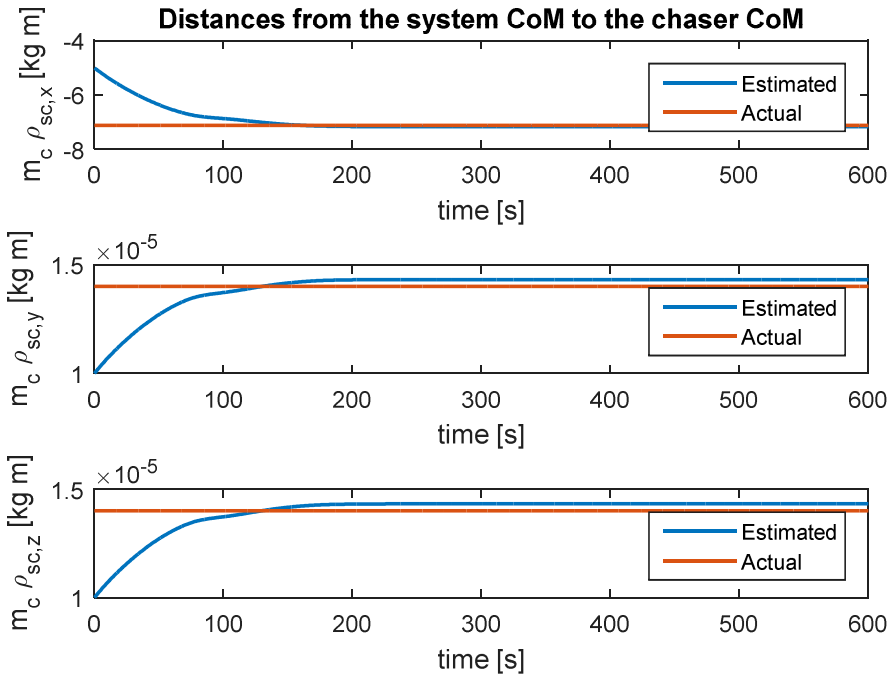

Figure 14. Distances between the system's CoM and chaser's one using RLSM

Table 5. Numerical results by using RLSM

PARAMETER	REAL	ESTIMATED	UNIT
$\boldsymbol{m_t}$	3.5	3.516	kg
$\boldsymbol{m_c}$	10	9.953	kg
$\boldsymbol{m_t\rho_t}$	$[7.12\ 0\ 0]^T$	$[7.132\ 1.4e\text{-}5\ 1.4e\text{-}5]^T$	kg*m
$\boldsymbol{m_c\rho_c}$	$[-7.12\ 0\ 0]^T$	$[-7.174\ 1.4e\text{-}5\ 1.4e\text{-}5]^T$	kg*m
$\boldsymbol{J_{xx,s}}$	0.1733	0.1726	$kg*m^2$
$\boldsymbol{J_{yy,s}}$	0.1733	0.1727	$kg*m^2$
$\boldsymbol{J_{zz,s}}$	0.1733	0.1726	$kg*m^2$
$\boldsymbol{J_{xy,s}}$	0	1.432e-5	$kg*m^2$
$\boldsymbol{J_{xz,s}}$	0	1.432e-5	$kg*m^2$
$\boldsymbol{J_{yz,s}}$	0	1.432e-5	$kg*m^2$

3.3. Adaptive Laws for system identification

In this section, two different methods are discussed in detail. First, the dynamics equations are parametrized as functions of satellites masses and the new centres of mass allocation after capture. Then, exploiting the same mathematical manipulations obtained with RLS, another observer is designed.

The aim of the first research is to apply the algorithm proposed in **[11]**, exploiting the similarity between the UAV dynamics and our study case. First of all, some restrictive assumptions are fundamental:

- Chaser and target are rigid bodies rigidly connected.
- The connection element is positioned so that the direction vector of this one is aligned with the vector joining the two centres of mass.
- The system's CoM is placed on this joining vector, so the distance between the chaser's CoM and the system's one (or similarly for the target) is defined as the following vector

$$\rho_{s,i} = \{\rho_{sx,i} \;\; 0 \;\; 0\}^T$$

- Masses and relative distances are constant over time.
- Inertia tensor elements are approximated as mass function (1.5).

We define

$$m_1 = m_t$$

$$m_2 = m_c$$

$$m_3 = m_t \rho_{sx,t}$$

$$m_4 = m_c \rho_{sx,c}$$

$$m_5 = m_t \rho_{sx,t}^2$$

$$m_6 = m_c \rho_{sx,c}^2$$

(3.4)

Considering (1.13) as our reference, the new matrices M and C will be:

$$M = m_1 M_{1t} + m_2 M_{1c} + m_3 M_{2t} + m_4 M_{2c} + m_5 M_{3t} + m_6 M_{3c}$$

$$C = m_1 C_{1t} + m_2 C_{1c} + m_3 C_{2t} + m_4 C_{2c} + m_5 C_{3t} + m_6 C_{3c} \tag{3.5}$$

$$M_{1t} = \begin{bmatrix} eye[15](3) & zeros[16](3,3) \\ zeros(3,3) & diag[17]([a\ b\ c]) \end{bmatrix}$$

$$M_{1c} = \begin{bmatrix} eye(3) & zeros(3,3) \\ zeros(3,3) & diag([d\ e\ f]) \end{bmatrix}$$

$$M_{2t} = \begin{bmatrix} & & & 0 & 0 & 0 \\ zeros(3,3) & & & 0 & 0 & 1 \\ & & & 0 & -1 & 0 \\ 0 & 0 & 0 & & & \\ 0 & 0 & -1 & zeros(3,3) & & \\ 0 & 1 & 0 & & & \end{bmatrix} = M_{2c}$$

$$M_{3t} = \begin{bmatrix} zeros(3,3) & zeros(3,3) \\ & 0\ \ 0\ \ 0 \\ zeros(3,3) & 0\ \ 1\ \ 0 \\ & 0\ \ 0\ \ 1 \end{bmatrix} = M_{3c}$$

$$C_{1t} = \begin{bmatrix} \omega^x & zeros(3,3) \\ zeros(3,3) & \omega^x diag([a\ b\ c]) \end{bmatrix}$$

$$C_{1c} = \begin{bmatrix} \omega^x & zeros(3,3) \\ zeros(3,3) & \omega^x diag([d\ e\ f]) \end{bmatrix}$$

$$C_{2t} = \begin{bmatrix} zeros(3,3) & \omega^x \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \omega^x & zeros(3,3) \end{bmatrix} = C_{2c}$$

$$C_{3t} = \begin{bmatrix} zeros(3,3) & zeros(3,3) \\ & 0\ \ 0\ \ 0 \\ zeros(3,3) & 0\ \ 0\ \ -\omega_x \\ & 0\ \ \omega_x\ \ 0 \end{bmatrix} = C_{3c}$$

(3.6)

[15] "*eye*" is the MATLAB command for the identity matrix definition.

[16] This is a MATLAB notation stands for a square matrix of dimension 3, composed by only elements equal to 0.

[17] The MATLAB command "*diag*" stands for the creation of a diagonal matrix.

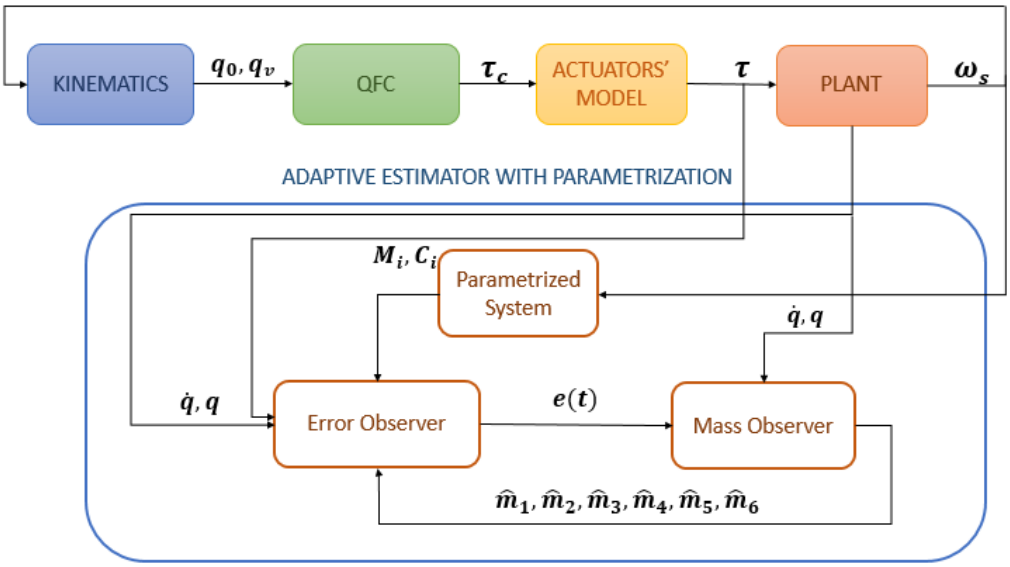

Figure 15. Block diagram of the adaptive estimator based on the system parametrization

$m_i \in R, M_{ij}, C_{ij} \in R^{6x6}$.

Grouping the similar terms, the final equation can be written as:

$$m_1(M_{1t}\,\dot{q} + C_{1t}q) + m_2(M_{1c}\,\dot{q} + C_{1c}q) + m_3(M_{2t}\,\dot{q} + C_{2t}q) +$$
$$+m_4(M_{2c}\,\dot{q} + C_{2c}q) + m_5(M_{3t}\,\dot{q} + C_{3t}q) + m_6(M_{3c}\,\dot{q} + C_{3c}q) = \tau \tag{3.7}$$

Finally, the on-line parameter estimator, whose a simple block diagram is shown in Figure 15, is formulated as:

$$C^*\dot{\hat{q}} + K^*\hat{q} + \hat{m}_1(M_{1t}\,\dot{q} + C_{1t}q) + \hat{m}_2(M_{1c}\,\dot{q} + C_{1c}q) + \hat{m}_3(M_{2t}\,\dot{q} + C_{2t}q) +$$
$$+\hat{m}_4(M_{2c}\,\dot{q} + C_{2c}q) + \hat{m}_5(M_{3t}\,\dot{q} + C_{3t}q) + \hat{m}_6(M_{3c}\,\dot{q} + C_{3c}q) =$$
$$= \tau + C^*\dot{q} + K^*q^{18} \tag{3.8}$$

[18] For a simple visualization of the equation, the notation $q(t), \hat{q}(t), \dot{q}(t), \dot{\hat{q}}(t)$ is omitted.

$$\dot{\hat{m}}_1 = \gamma_1 e^T(t)(M_{1t}\,\dot{q} + C_{1t}q)$$

$$\dot{\hat{m}}_2 = \gamma_2 e^T(t)(M_{1c}\,\dot{q} + C_{1c}q)$$

$$\dot{\hat{m}}_3 = \gamma_3 e^T(t)(M_{2t}\,\dot{q} + C_{2t}q)$$

$$\dot{\hat{m}}_4 = \gamma_4 e^T(t)(M_{2c}\,\dot{q} + C_{2c}q)$$

$$\dot{\hat{m}}_5 = \gamma_5 e^T(t)(M_{3t}\,\dot{q} + C_{3t}q)$$

$$\dot{\hat{m}}_6 = \gamma_6 e^T(t)(M_{3c}\,\dot{q} + C_{3c}q) \tag{3.9}$$

$$e(t) = \hat{q}(t) - q(t)$$

$C^*, K^* \in R^{6x6}$ are defined positive diagonal gain matrices, $\gamma_i \in R$ are user positive rates, $\hat{q}(t) \in R^{6x1}$is the estimated of the state vector $q(t)$.

The parameters adopted to run the simulation are reported in Table 6.

Table 6. Simulation parameters for Adaptive Estimator with the parametrization of the system

PARAMETER	VALUE	UNIT OF MEASURE
$\boldsymbol{C^*}$	50*eye(6)	/
$\boldsymbol{K^*}$	50*eye(6)	/
$\dot{\hat{\boldsymbol{q}}}(\mathbf{0})$	[10 10 10 10 10 10]^T	m/s^2 (for the first 3), rad/s^2 (for the last 3)
$\boldsymbol{\gamma}$	1e3*[1.5 1.6 0.45 1.1 0.68 0.75]^T	/
$\hat{\boldsymbol{m}}(\mathbf{0})$	[4.52 11.6 6.75 -8.05 17.65 8.85]	kg (for the first 2) kg*m^2 (for the last 3) kg*m (for the others)

The Adaptive Law obtained considering this parametrization of the system, like the reference paper **[11]**, shows good estimation properties: using the gain matrices described in Table 6, masses, relative distances between chaser and target centre of mass and the total system's one and inertia tensor elements converge to the desired values in a maximum time lapse of 10 seconds, as shown in Figures 16-18.

Though the simulation runs for 600 seconds, to understand how the rigid system evolves over time in terms of attitude, for greater visual clarity, the graphical results are presented in a short period (0-20 seconds).

However, the unique limitation of this algorithm is related to the strong assumptions at the base of the physical model. Since all these ones have to be respected for evaluating the described inertial parameters, the proposed estimator cannot be considered as an observer that could be applied in a general case.

Figure 16 shows the estimated masses of chaser and target satellites. Using the classical relation to calculate the percent estimation error (dividing the absolute value of the difference between the estimated and the actual values by the actual and then multiplying by 100), we note that at the end of the transitory, the estimation error for target mass is about 1.51%, while for chaser It is equal to 0.3 %.

Despite the proposed algorithm is designed for estimating the spacecraft masses (m_t, m_c) and the product between chaser or target mass and the relative distances from the centre of mass of the total system ($m_t\rho_{st,x}, m_c\rho_{sc,x}$), the results of Figure 17 are obtained dividing, for each sample time, the correlated estimated quantities (for example $m_t\rho_{st,x}$ and m_t).

Finally, because a typical space mission, connected to grasping processes of a debris or similar objects such as non-cooperative spacecraft, is designed knowing the maximum size of this one, exploiting the defined parametrization and the estimated masses, both inertia tensor elements of the two bodies can be evaluated, in case of symmetry, exploiting as first assumption the equations (1.5). Then, the total inertia tensor is calculated adding chaser and target inertia tensors. Figure 18 shows these inertial properties.

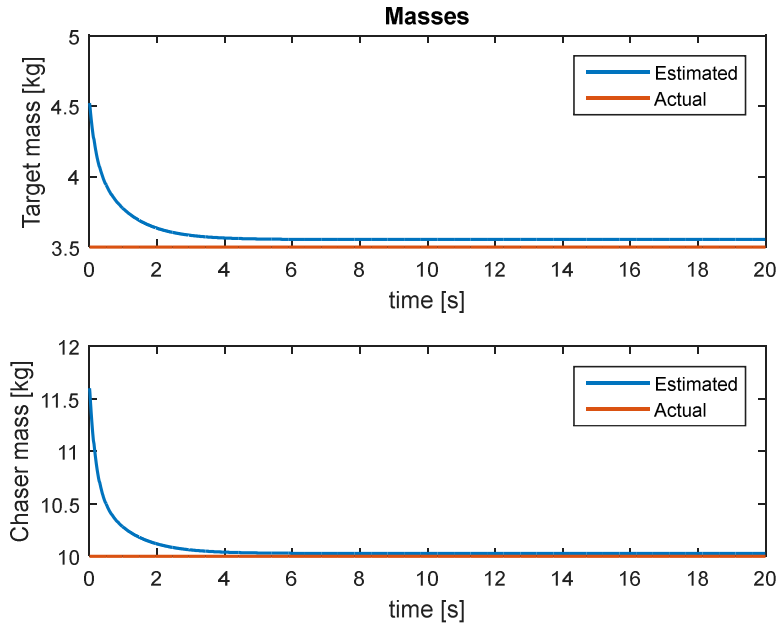

Figure 16. Mass estimation using an Adaptive Law based on a parametrization of the system

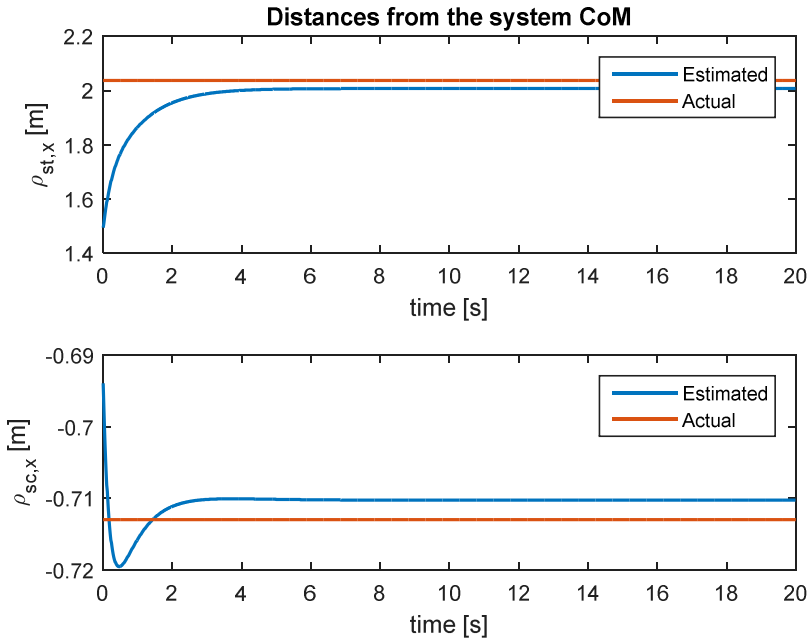

Figure 17. Estimation of the relative distances from the system's CoM using an Adaptive Law based on a parametrization of the system

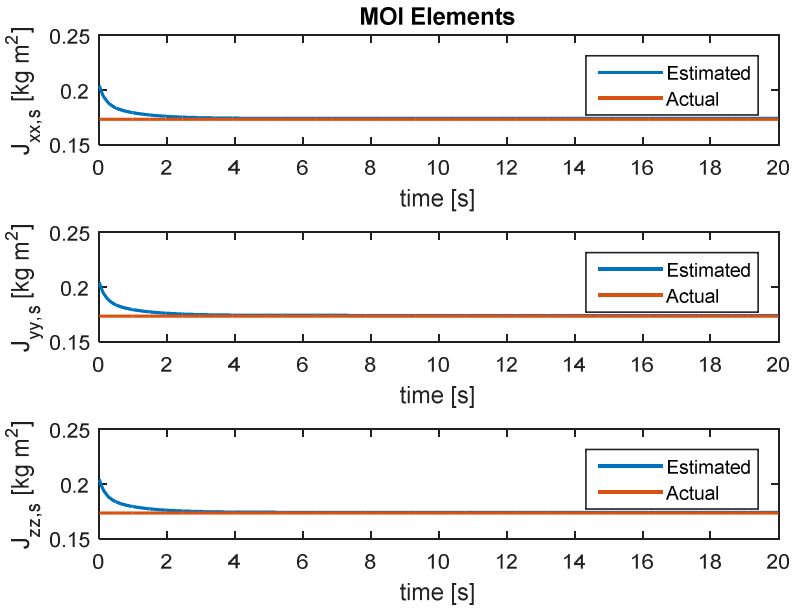

Figure 18. MOI elements estimation using an Adaptive Law based on a parametrization of the system

Table 7. Numerical results using an Adaptive Law applied to a parametrization of the dynamic system

PARAMETER	REAL	ESTIMATED	UNIT
$\boldsymbol{m_t}$	3.5	3.553	kg
$\boldsymbol{m_c}$	10	10.03	kg
$\boldsymbol{m_t\rho_{t,x}}$	7.12	7.131	kg*m
$\boldsymbol{m_c\rho_{c,x}}$	-7.12	-7.119	kg*m
$\boldsymbol{m_t\rho_{t,x}^2}$	14.523	14.21	kg*m^2
$\boldsymbol{m_c\rho_{c,x}^2}$	5.0832	5.056	kg*m^2

In the Appendix A, It is possible to consult the mathematical proof according to which the estimation error goes to zero when time goes to infinity.

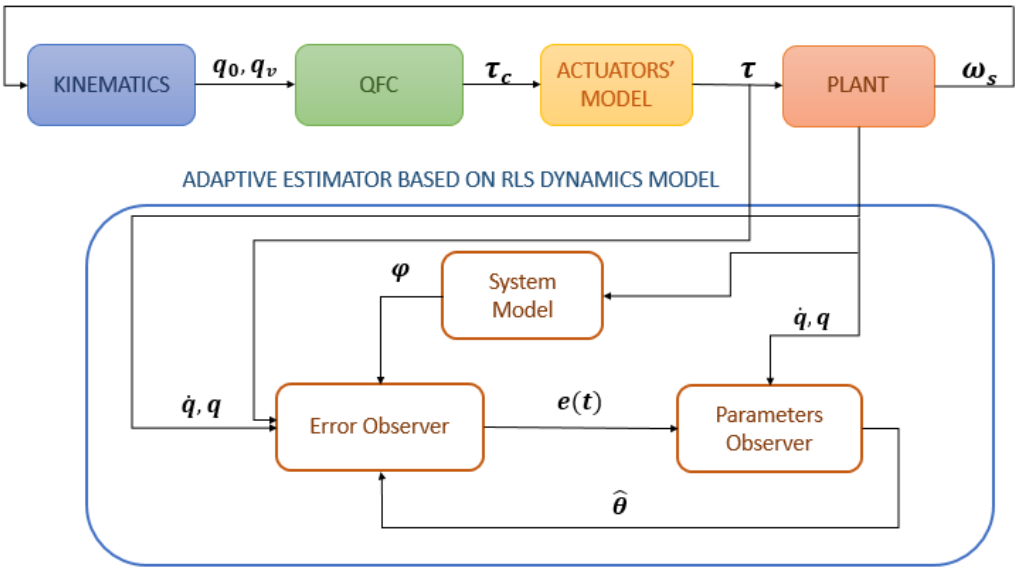

Figure 19. Block diagram of the Adaptive Law based on RLS dynamics model

Since the previous proposed algorithm imposes really restrictive assumptions, a system model's variation is required. The general idea, illustrated in Figure 19, is to apply the previous adaptive law to the dynamics system, formulated according to RLSM.

Using the same mathematical procedure explained in **[16]**, the robot dynamics model can be expressed in the following nonlinear[19] form:

$$M(q)\dot{q} + C(q)q = \tau$$

$$\tau = \varphi(q, \dot{q}, \theta)\theta \tag{3.10}$$

Exploiting (3.10), the final Adaptive Law will be:

[19] In the robot dynamics formulation described by professor A. De Luca (**[16]**), the regressor matrix, linearly dependent on $\ddot{q}$ and quadratically dependent on q, is a function of the state vector and of its derivative. In our case, because of dynamics formulation, It is not possible to reach a linear form of the system: the regressor matrix is a function of the estimate vector, too.

$$C^*\dot{q}(t) + K^*(t)q(t) + \tau = \widehat{M}(q)\dot{q}(t) + \hat{C}(q)q(t) + C^*\hat{\dot{q}}(t) + K^*\hat{q}(t)$$

$$C^*\dot{q}(t) + K^*(t)q(t) + \tau = \varphi(\dot{q}, q, \hat{\theta})\hat{\theta} + C^*\hat{\dot{q}}(t) + K^*\hat{q}(t)$$

$$\dot{\hat{\theta}} = Q\varphi^T(\dot{q}, q, \hat{\theta})(\hat{q} - q) \tag{3.11}$$

$C^*, K^* \in R^{6x6}$ and $Q \in R^{14x14}$ are user-defined positive gain matrices, $\widehat{M}(q), \hat{C}(q) \in R^{6x6}$ are the same matrices formulated in (3.1), $\tau \in R^{6x1}$ is the control vector, $\hat{\theta}(t) \in R^{14x1}$ is the estimated vector expressed in the same form of (3.3).

The parameters adopted for running the simulation are reported in Table 8.

Table 8. Simulation parameters for Adaptive Estimator based on dynamics model formulated for RLS

PARAMETER	VALUE	UNIT OF MEASURE
$\boldsymbol{C^*}$	50*eye(6)	/
$\boldsymbol{K^*}$	50*eye(6)	/
$\hat{\dot{\boldsymbol{q}}}(\mathbf{0})$	[7.5 7.5 7.5 7.5 7.5 7.5]^T	m/s^2 (for the first 3), rad/s^2 (for the last 3)
$\boldsymbol{Q}$	diag([600 670 1000 0.005 0.005 630 0.005 0.005 100.6 162 985 0.005 0.005 0.005])	/
$\hat{\boldsymbol{\theta}}(\mathbf{0})$	[3.68 10.2 8.75 1e-5 1e-5 -8.1 1e-5 1e-5 3.025 0.337 0.325 1e-5 1e-5 1e-5]	kg (for the first 2) kg*m^2 (for the last 6) kg*m (for the others)

In the Appendix A, It is possible to consult the mathematical proof according to which the estimation error goes to zero when time goes to infinity.

Rearranging the equations of motion, expressed in matrix form, and applying the same mathematical procedure described in **[11]**, the second Adaptive Law shows a mixture of the numerical properties common to RLSM and the first Adaptive Law, based on the parametrization of the system.

In particular, unlike the RLSM, the convergence time is reduced: this property is closely related to the effectiveness of the studied "*On-Line Parameter Estimator*" (**[11]**), assumed as reference.

As reported in Table 8, the simulation parameters are similar to those in Table 6. In detail:

- C^* and K^* are the same.
- γ_i gains are here replaced with the diagonal positive defined matrix Q.
- $\hat{\dot{q}}(t)$ is changed for improve the convergence rate.
- An initial estimation of the inertial properties is required.

In conclusion, similar to Recursive Least Square Method, the second Adaptive Law depends on both the chosen initial conditions and the gain matrices, but, unlike the first (RLS), this nonlinear observer shows a better convergence rate.

Figure 20 describes the evolution of the estimated masses: the identified values converge to the actual ones after about 10 seconds (like the first Adaptive Law).

Unfortunately, this property decays when $m_t\rho_{st,z}$ and $m_c\rho_{sc,z}$ are estimated (see Figures 21-22): despite the desired state is reached after about 100 seconds (less than RLSM), the estimation error at 25 s, reported in Table 13, is acceptable (the maximum error is equal to 8.77% and decreases up to 100 s).

Similarly, the same behaviour occurs for $J_{zz,s}$ (Figure 23).

To improve the estimation process of these parameters, It is necessary to investigate in more detail about C^* and K^* matrices.

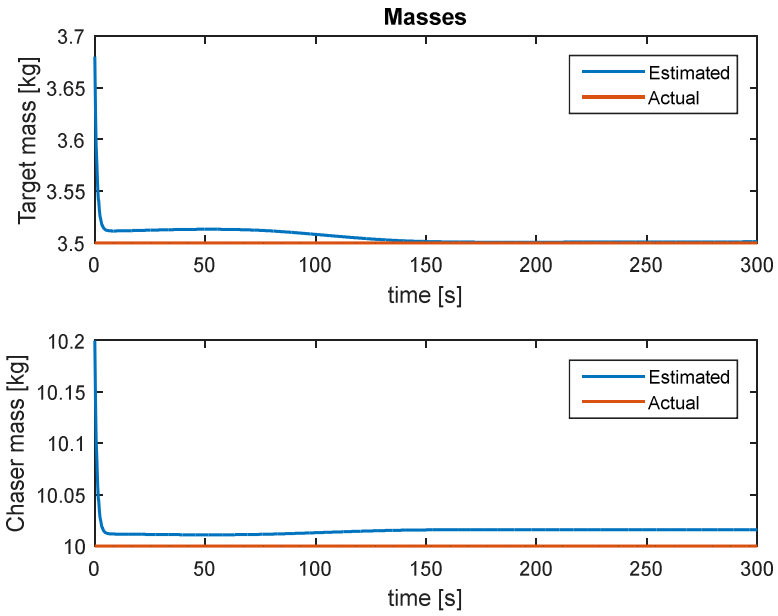

Figure 20. Mass estimation using Adaptive Law based on RLS dynamics formulation

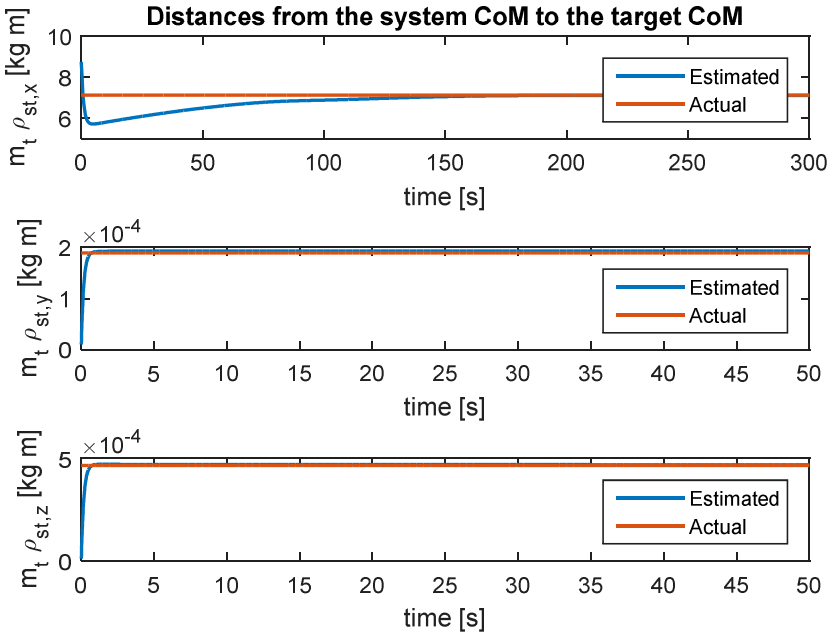

Figure 21. Estimation of the relative distances between the target's CoM and the system's one using the Adaptive Law based on RLS dynamics formulation

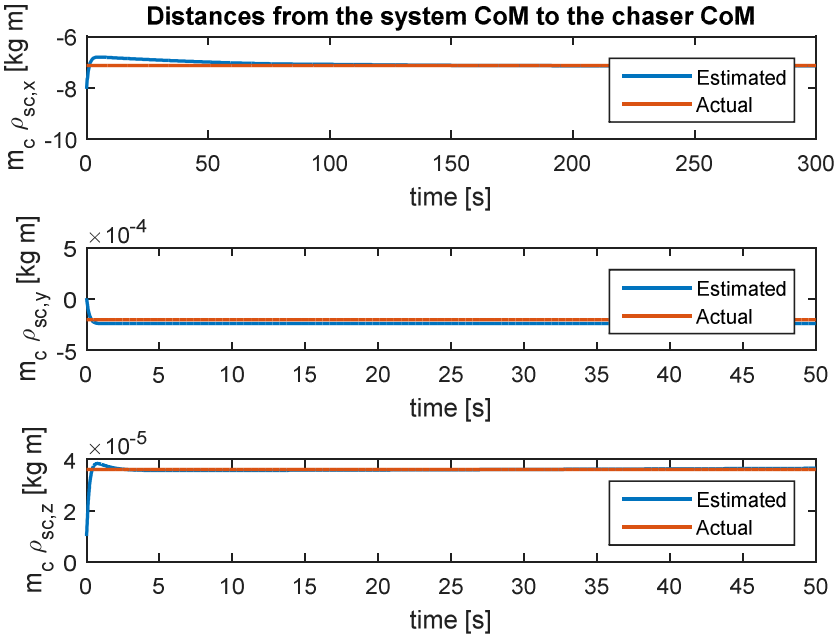

Figure 22. Estimation of the relative distances between the chaser's CoM and the system's one using the Adaptive Law based on RLS dynamics formulation

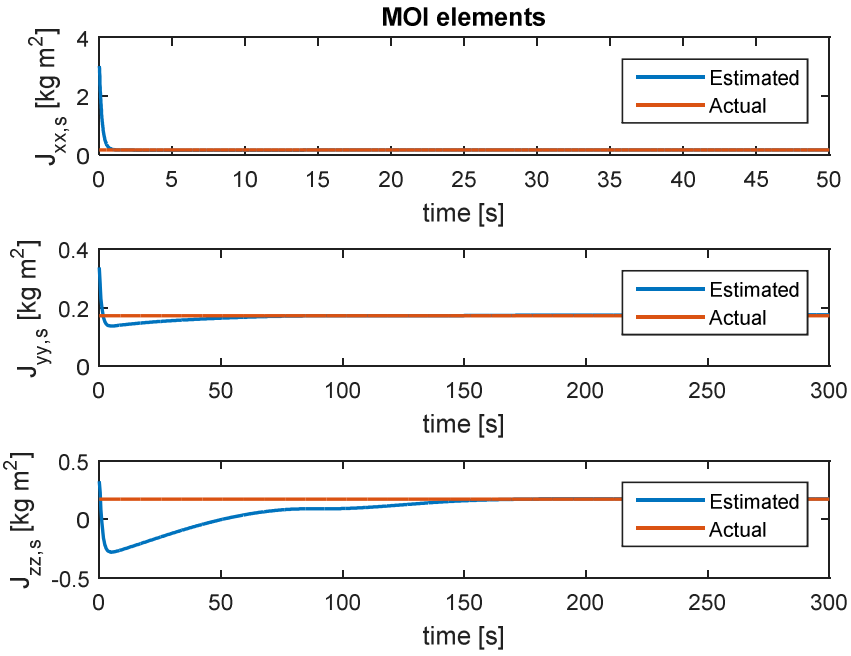

Figure 23. MOI elements estimation using the Adaptive Law based on RLS dynamics formulation

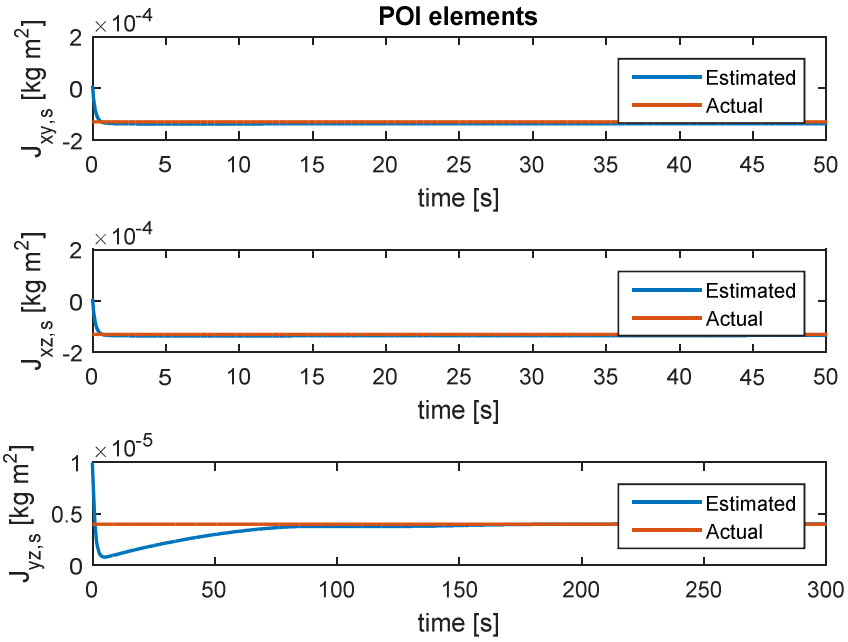

Figure 22. POI elements estimation using the Adaptive Law based on RLS dynamics formulation

Finally, a summary of the numerical results obtained by the last estimation process, are reported in Table 9.

Table 9. Numerical results using the Adaptive Law based on RLS dynamics formulation

PARAMETER	REAL	ESTIMATED	UNIT
$\boldsymbol{m_t}$	3.5	3.501	kg
$\boldsymbol{m_c}$	10	10.02	kg
$\boldsymbol{m_t\rho_t}$	$[7.12\ 0\ 0]^T$	$[7.125\ 0.00019\ 0.00047]^T$	kg*m
$\boldsymbol{m_c\rho_c}$	$[-7.12\ 0\ 0]^T$	$[-7.144\ -0.00024\ 3.724e\text{-}5]^T$	kg*m
$\boldsymbol{J_{xx,s}}$	0.1733	0.1747	kg*m^2
$\boldsymbol{J_{yy,s}}$	0.1733	0.1756	kg*m^2
$\boldsymbol{J_{zz,s}}$	0.1733	0.1759	kg*m^2
$\boldsymbol{J_{xy,s}}$	0	-0.000137	kg*m^2
$\boldsymbol{J_{xz,s}}$	0	-0.0001329	kg*m^2
$\boldsymbol{J_{yz,s}}$	0	4.007e-6	kg*m^2

3.4. Analysis of Results

For better understanding how the RSLM and the second Adaptive Law work over time, in this subsection a detailed analysis related to the behaviour of the two estimators is discussed, in case of variable frequency.

In particular, for each observer, gains and initial conditions don't change, because We want to study the evolution of the estimation process, in the event that a signal delay occurs. This hypotesis is realistic, because in a real simulator:

- GNC software runs at the specified frequency (in our case, the proposed observers are included in Navigation Toolbox).
- It is possible that a delay in the communication of information from the on-board sensors occurs.

For each estimator, a series of graphical and numerical results are reported; in particular, the estimation processes studied in the previous sections are compared to others, obtained using two different sample times (0.02 s and 0.04 s), or, in another way, adopting two different frequencies (50 Hz and 25 Hz respectively).

3.4.1. RLSM Analysis

Table 10. Numerical results for RLS algorithm, considering different time points @ dt=0.01 s (f=100 Hz)

	$t = 50\ s$		$t = 100\ s$		$t = 150\ s$	
m_t	3.3686	*3.75%*	3.5125	*0.36%*	3.5156	*0.45%*
m_c	9.5364	*4.64%*	9.9437	*0.56%*	9.9526	*0.47%*
$m_t \rho_{st,x}$	6.8333	*4.03%*	7.1251	*0.07%*	7.1315	*0.16%*
$m_t \rho_{st,y}$	0	*0%*	0	*0%*	0	*0%*
$m_t \rho_{st,z}$	0	*0%*	0	*0%*	0	*0%*
$m_c \rho_{sc,x}$	-6.8745	*3.45%*	-7.1680	*0.67%*	-7.1745	*0.77%*
$m_c \rho_{sc,y}$	0	*0%*	0	*0%*	0	*0%*
$m_c \rho_{sc,z}$	0	*0%*	0	*0%*	0	*0%*
$J_{xx,s}$	0.1653	*4.62%*	0.1724	*0.52%*	0.1726	*0.4%*
$J_{yy,s}$	0.1655	*4.5%*	0.1725	*0.46%*	0.1727	*0.35%*
$J_{zz,s}$	0.1653	*4.62%*	0.1724	*0.52%*	0.1726	*0.4%*
$J_{xy,s}$	0	*0%*	0	*0%*	0	*0%*
$J_{xz,s}$	0	*0%*	0	*0%*	0	*0%*
$J_{yz,s}$	0	*0%*	0	*0%*	0	*0%*

Table 10 describes the evolution of the estimation error when the operating frequency is set to 100 Hz. It is possible to note that the percent error calculated as:

$$error = \frac{estimated - actual}{actual} * 100$$[20]

[20] This formulation is applied in all the following tables.

Starts to go to zero after about 150 s. This is related not only to the appropriate choice of the initial conditions but also to covariance matrix.

This is not true if a variation of frequency occurs. Tables 11-12 are the mathematical proof that changing the sample time and fixing all the other quantities, the convergence property of the algorithm degradates: RLSM continues to converge to another actual value as calculated numerically.

Table 11. Numerical results for RLS algorithm, considering different time points @ dt=0.02 s (f=50 Hz)

	$t = 50\ s$		$t = 100\ s$		$t = 150\ s$	
m_t	2.9119	*16.81%*	2.9839	*14.75%*	2.9855	*14.7%*
m_c	8.2434	*17.57%*	8.4473	*15.53%*	8.4518	*15.48%*
$m_t\rho_{st,x}$	5.9068	*17.04%*	6.0529	*14.98%*	6.0561	*19.94%*
$m_t\rho_{st,y}$	0	*0%*	0	*0%*	0	*0%*
$m_t\rho_{st,z}$	0	*0%*	0	*0%*	0	*0%*
$m_c\rho_{sc,x}$	-5.9423	*16.54%*	-6.0894	*14.47%*	-6.0926	*14.43%*
$m_c\rho_{sc,y}$	0	*0%*	0	*0%*	0	*0%*
$m_c\rho_{sc,z}$	0	*0%*	0	*0%*	0	*0%*
$J_{xx,s}$	0.1429	*17.54%*	0.1465	*15.46%*	0.1465	*15.46%*
$J_{yy,s}$	0.1430	*17.48%*	0.1466	*15.41%*	0.1467	*15.35%*
$J_{zz,s}$	0.1429	*17.54%*	0.1465	*15.46%*	0.1465	*15.46%*
$J_{xy,s}$	0	*0%*	0	*0%*	0	*0%*
$J_{xz,s}$	0	*0%*	0	*0%*	0	*0%*
$J_{yz,s}$	0	*0%*	0	*0%*	0	*0%*

Table 12. Numerical results for RLS algorithm, considering different time points @ dt=0.04 s (f=25 Hz)

	$t = 50\ s$		$t = 100\ s$		$t = 150\ s$	
m_t	2.6834	*23.33%*	2.7195	*22.3%*	2.7202	*22.28%*
m_c	7.5966	*24.03%*	7.6987	*23.01%*	7.7009	*22.99%*
$m_t\rho_{st,x}$	5.4433	*23.55%*	5.5164	*22.52%*	5.5191	*22.48%*
$m_t\rho_{st,y}$	0	*0%*	0	*0%*	0	*0%*
$m_t\rho_{st,z}$	0	*0%*	0	*0%*	0	*0%*
$m_c\rho_{sc,x}$	-5.4761	*23.09%*	-5.5497	*22.05%*	-5.5191	*22.48%*
$m_c\rho_{sc,y}$	0	*0%*	0	*0%*	0	*0%*
$m_c\rho_{sc,z}$	0	*0%*	0	*0%*	0	*0%*
$J_{xx,s}$	0.1317	*24%*	0.1335	*22.97%*	0.1335	*22.97%*
$J_{yy,s}$	0.1318	*23.95%*	0.1336	*22.91%*	0.1336	*22.91%*
$J_{zz,s}$	0.1317	*24%*	0.1335	*22.97%*	0.1335	*22.97%*
$J_{xy,s}$	0	*0%*	0	*0%*	0	*0%*
$J_{xz,s}$	0	*0%*	0	*0%*	0	*0%*
$J_{yz,s}$	0	*0%*	0	*0%*	0	*0%*

After a brief study about the general properties of this estimator, It is possible to affirm that fixing initial conditions and initial covariance matrix and increasing the sample time (or alternatively decreasing the reference frequency), the RLS algorithm doesn't converge to the reference state.

Really, as you can see in all the following graphical results, under the previous assumptions, the proposed nonlinear observer converges to a different desired state, that decreases if the operating frequency decreases, when time goes to infinity.

If the main purpose is to design a Navigation Toolbox, that runs at smaller frequency, the convergence property is satisfied only increasing all the elements of the initial covariance matrix, exploiting the corrective factor α.

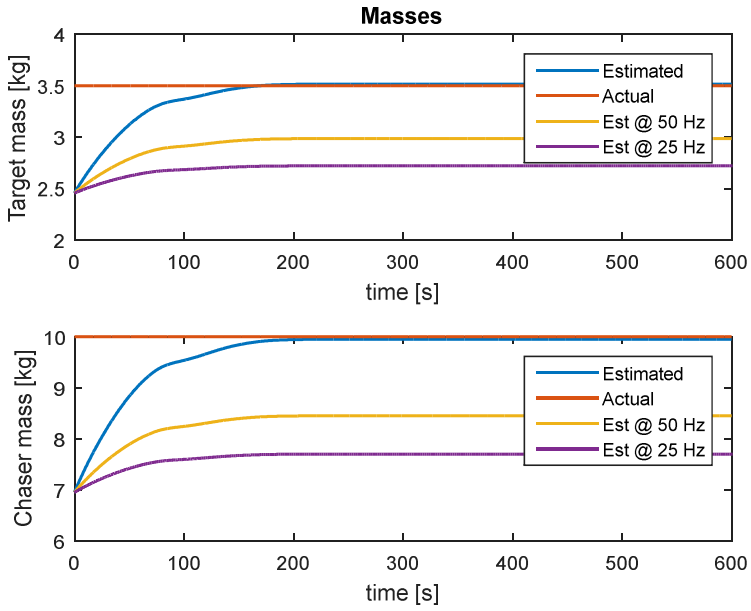

Figure 23. Mass estimation at different frequencies using RLSM

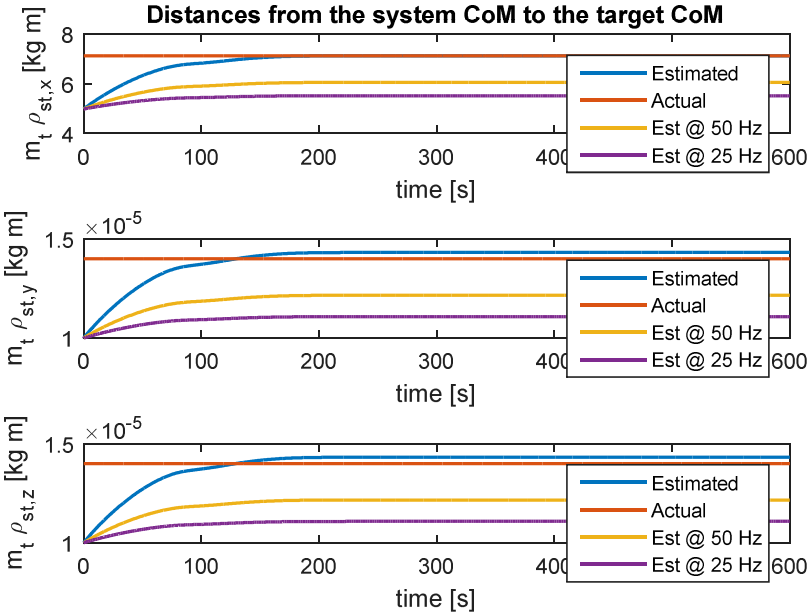

Figure 24. Estimation of relative distances between system's CoM and target's one at different frequencies using RLSM

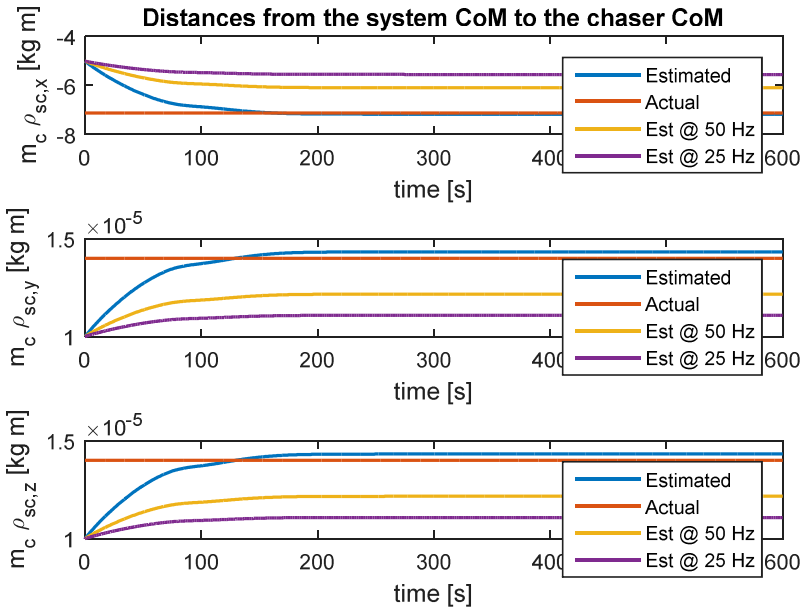

Figure 25. Estimation of relative distances between system's CoM and chaser's one at different frequencies using RLSM

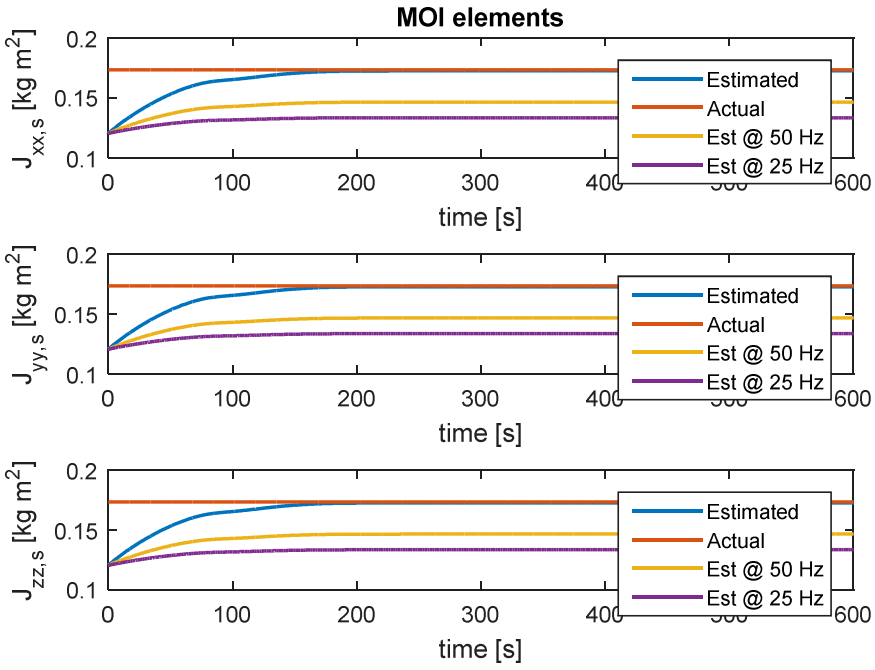

Figure 26. MOI elements estimation at different frequencies using RLSM

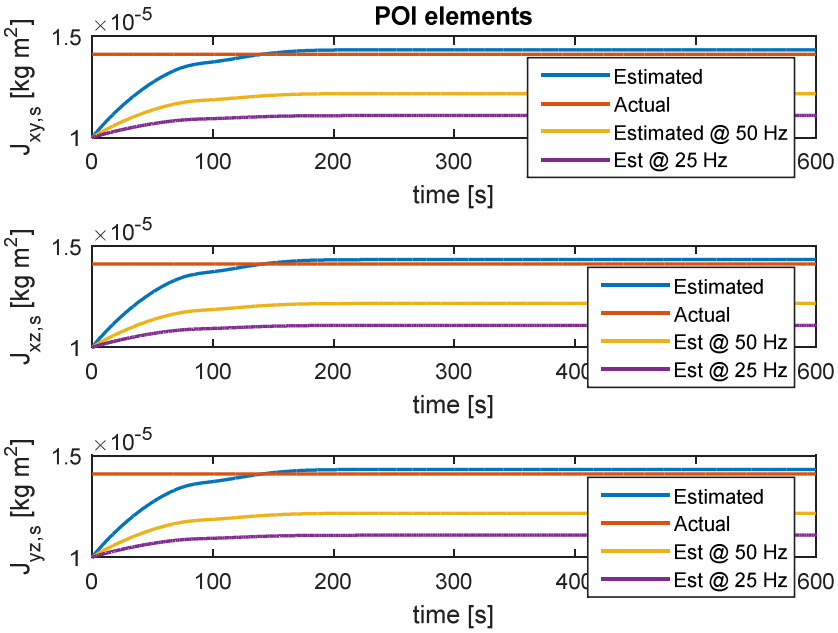

Figure 27. POI elements estimation at different frequencies using RLSM

3.4.2. Detailed analysis about the second Adaptive Law

Table 13. Numerical results related to the second Adaptive Law, considering different time points @ dt=0.01 s (f=100 Hz)

	$t = 25\ s$		$t = 50\ s$		$t = 100\ s$	
m_t	3.5132	*0.38%*	3.5082	*0.23%*	3.5007	*0.02%*
m_c	10.0110	*0.11%*	10.0130	*0.13%*	10.0160	*0.16%*
$m_t\rho_{st,x}$	6.4957	*8.77%*	6.8844	*3.31%*	7.1148	*0.07%*
$m_t\rho_{st,y}$	0.0002	*0%*	0.0002	*0%*	0.0002	*0%*
$m_t\rho_{st,z}$	0.0005	*0%*	0.0005	*0%*	0.0005	*0%*
$m_c\rho_{sc,x}$	-7.0004	*1.68%*	-7.0909	*0.41%*	-7.1425	*0.32%*
$m_c\rho_{sc,y}$	-0.0002	*0%*	-0.0002	*0%*	-0.0002	*0%*
$m_c\rho_{sc,z}$	0	*0%*	0	*0%*	0	*0.%*
$J_{xx,s}$	0.1719	*0.81%*	0.1744	*0.63%*	0.1747	*0.081%*
$J_{yy,s}$	0.1647	*4.96%*	0.1734	*0.06%*	0.1753	*1.15%*
$J_{zz,s}$	-0.0019	*101.09%*	0.0924	*46.68%*	0.1757	*1.38%*
$J_{xy,s}$	-0.0001	*0%*	-0.0001	*0%*	-0.0001	*0%*
$J_{xz,s}$	-0.0001	*0%*	-0.0001	*0%*	-0.0001	*0%*
$J_{yz,s}$	0	*0%*	0	*0%*	0	*0%*

Comparing the time history of the estimation error calculated in Table 13, for the second Adaptive Law, and in Table 10, for RLSM, It is clear that the first method, apart from $J_{zz,s}$, is more efficient in terms of parameters identification.

Tables 14-15 highlight that, unlike Recursive Least Square algorithm, the proposed nonlinear observer continues to estimate inertial properties with an acceptable accuracy. This fundamental condition decays if the update of the input information of the estimator is elaborated every 1s.

Table 14. Numerical results related to the second Adaptive Law, considering different time points @ dt=0.02 s (f=50 Hz)

	$t = 25\ s$		$t = 50\ s$		$t = 100\ s$	
m_t	3.5132	*0.38%*	3.5082	*0.23%*	3.5006	*0.017%*
m_c	10.0110	*0.11%*	10.0130	*0.13%*	10.0160	*0.16%*
$m_t\rho_{st,x}$	6.5316	*8.26%*	6.9229	*2.77%*	7.1547	*0.49%*
$m_t\rho_{st,y}$	0.0002	*0%*	0.0002	*0%*	0.0002	*0%*
$m_t\rho_{st,z}$	0.0005	*0%*	0.0005	*0%*	0.0005	*0%*
$m_c\rho_{sc,x}$	-7.0164	*1.46%*	-7.1072	*0.18%*	-7.1589	*0.55%*
$m_c\rho_{sc,y}$	-0.0002	*0%*	-0.0002	*0%*	-0.0002	*0%*
$m_c\rho_{sc,z}$	0	*0%*	0	*0%*	0	*0%*
$J_{xx,s}$	0.2176	*25.56%*	0.2201	*27%*	0.2204	*27.18%*
$J_{yy,s}$	0.1678	*3.17%*	0.1765	*1.85%*	0.1784	*2.94%*
$J_{zz,s}$	0.0123	*92%*	0.1069	*38.32%*	0.1901	*9.69%*
$J_{xy,s}$	-0.0001	*0%*	-0.0001	*0%*	-0.0001	*0%*
$J_{xz,s}$	-0.0001	*0%*	-0.0001	*0%*	-0.0001	*0%*
$J_{yz,s}$	0	*0%*	0	*0%*	0	*0%*

Table 15. Numerical results related to the second Adaptive Law, considering different time points @ dt=0.04 s (f=25 Hz)

	$t = 25\ s$		$t = 50\ s$		$t = 100\ s$	
m_t	3.5132	*0.38%*	3.5082	*0.23%*	3.5006	*0.017%*
m_c	10.0110	*0.11%*	10.0130	*0.13%*	10.0160	*0.16%*
$m_t\rho_{st,x}$	6.5668	*7.76%*	6.9608	*2.24%*	7.1940	*1.04%*
$m_t\rho_{st,y}$	0.0002	*0%*	0.0002	*0%*	0.0002	*0%*
$m_t\rho_{st,z}$	0.0005	*0%*	0.0004	*0%*	0.0004	*0%*
$m_c\rho_{sc,x}$	-7.0366	*1.17%*	-7.1278	*0.11%*	-7.1797	*0.84%*
$m_c\rho_{sc,y}$	-0.0002	*0%*	-0.0002	*0%*	-0.0002	*0%*
$m_c\rho_{sc,z}$	0	*0%*	0	*0%*	0	*0%*
$J_{xx,s}$	0.3084	*77.96%*	0.3109	*79.39%*	0.3112	*79.57%*
$J_{yy,s}$	0.1715	*1.04%*	0.1802	*3.98%*	0.1822	*5.14%*
$J_{zz,s}$	0.0246	*85.81%*	0.1193	*31.16%*	0.2026	*16.91%*
$J_{xy,s}$	-0.0001	*0%*	-0.0001	*0%*	-0.0001	*0%*
$J_{xz,s}$	-0.0001	*0%*	-0.0001	*0%*	-0.0001	*0%*
$J_{yz,s}$	0	*0%*	0	*0%*	0	*0%*

Exploiting all these results, some important conclusions can be extrapolated:

- Apart from some inertial parameters, such as $J_{zz,s}$ element, who converges to the desired value after 100 seconds, in general, the Adaptive Law shows good estimation properties.
- An optimization of the gain matrices (Q, K^*, C^*) can improve the convergence rate.
- Increasing the sample time, so increasing the signal delay, both algorithms are subject to changes during all the estimation process.

- As shown in the previous numerical and graphical results, running the simulations at these two different frequencies, unlike RLSM, the estimation error calculated adopting the Adaptive Law is acceptable.
- Using a sample time equal to 1 second (or alternatively a reference frequency for the observer equal to 1 Hz), the convergence properties of the adaptive Law decay, as shown in Appendix B.
- Finally, as demonstrated in Appendix A, both algorithms, under the appropriate assumptions, converge to the desired state, but, because of the similarity to a common Kalman Filter, RLS converges more slowly, if compared to the analysed Adaptive Law. Really, considering this one, some inertial properties converge to the desired after 100 seconds (smaller than RLSM convergence time), due to system nonlinearity.

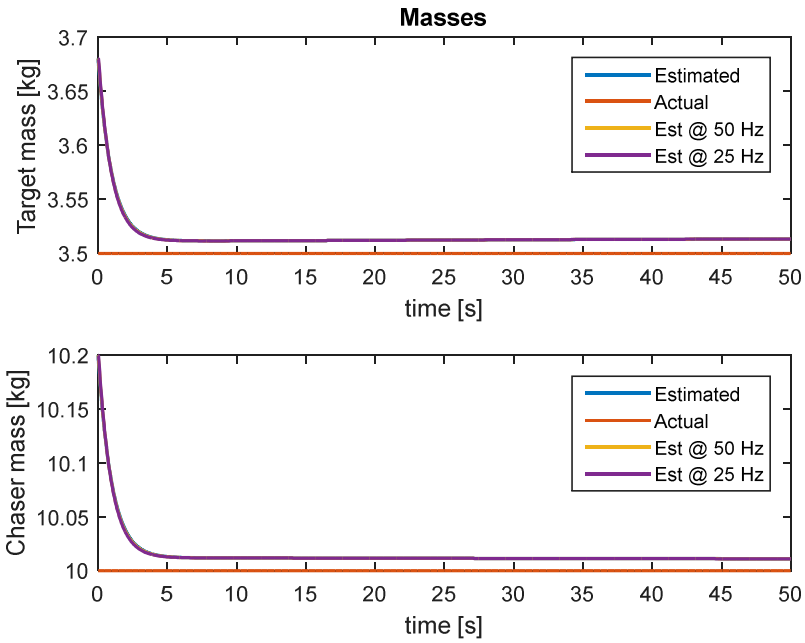

Figure 28. Mass estimation at different frequencies, considering the second Adaptive Law

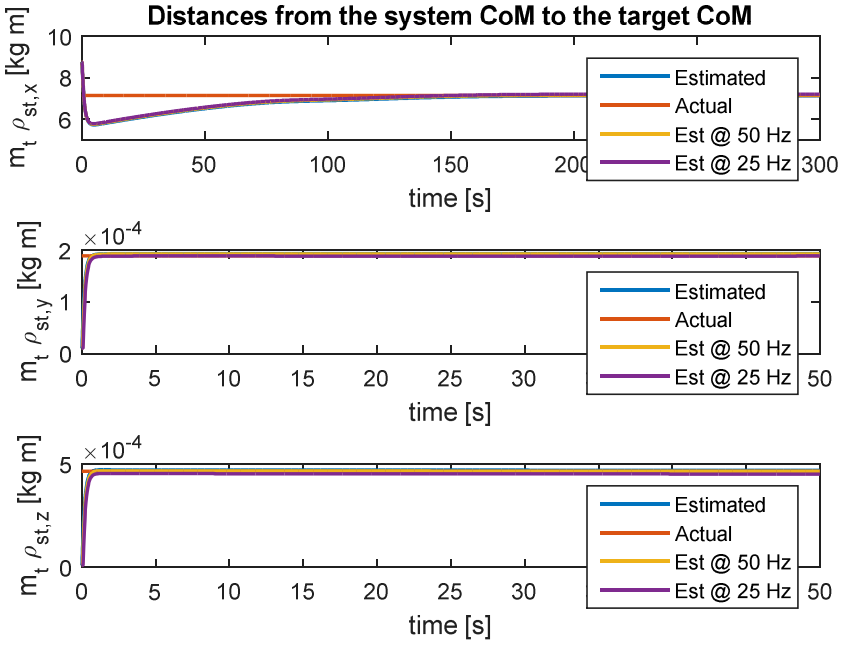

Figure 29. Estimation of relative distances between system's CoM and target's one at different frequencies considering the second Adaptive Law

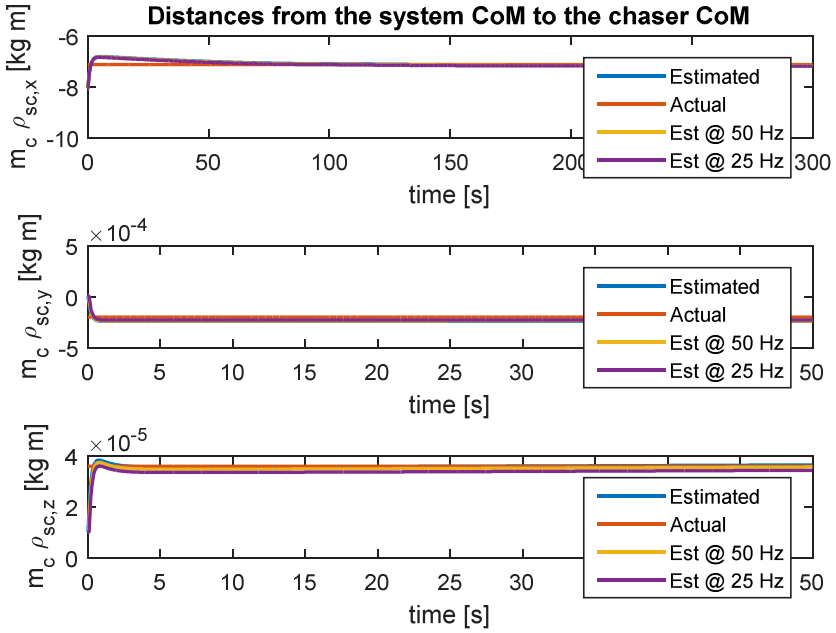

Figure 30. Estimation of relative distances between system's CoM and chaser's one at different frequencies considering the second Adaptive Law

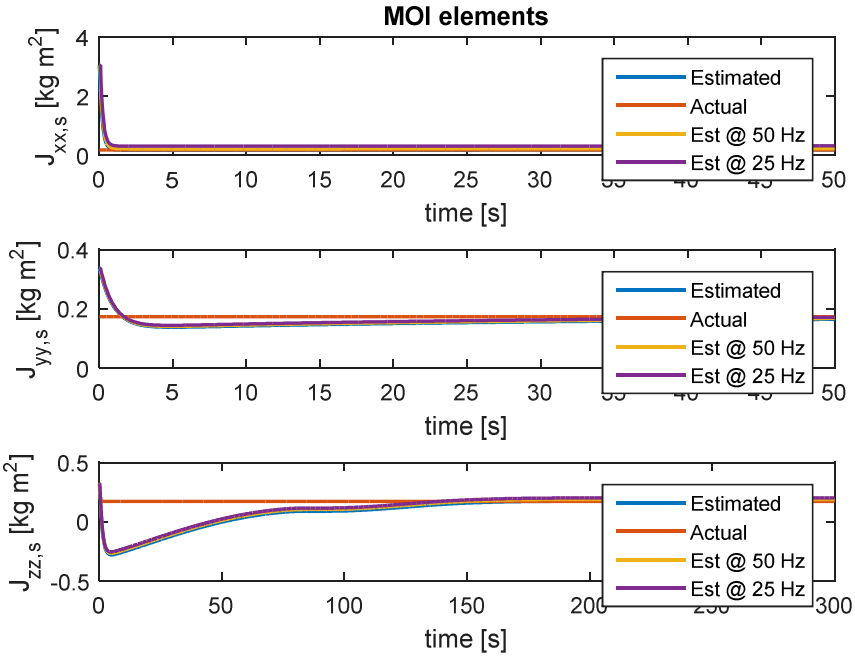

Figure 31. MOI elements estimation at different frequencies, considering the second Adaptive Law

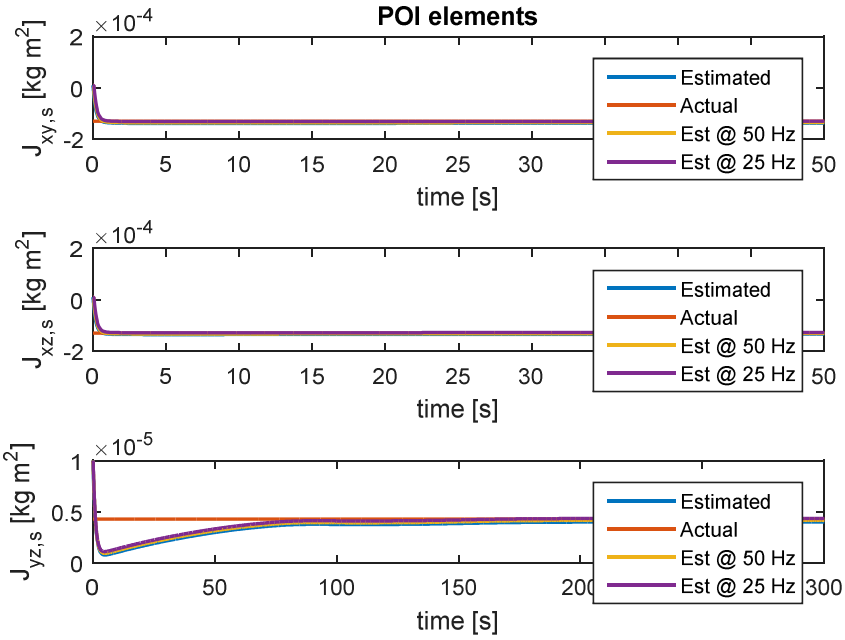

Figure 32. POI elements estimation at different frequencies, considering the second Adaptive Law

Conclusions and Future Works

In this work a mathematical model of a system composed by two rigid bodies, rigidly connected, is derived and compared to others discussed in scientific literature. Exploiting the results obtained from a preliminary study on a single satellite, three different nonlinear estimation algorithms are proposed and applied on the two CubeSat system.

A Recursive Least Square Method (RLSM) is implemented after an appropriate manipulation of the system dynamics. The initial conditions and the initial covariance matrix are determined tuning the simulation for better reaching the desired state.

A first Adaptive Law, based on a parametrization of the system, is derived exploiting the mathematical procedure analysed in **[11]**. In particular, unlike this one, under the appropriate assumptions, the plant is modelled for evaluating the mass properties not only of the grasped target, but also of the collaborative chaser. As shown in graphical and numerical results, unlike RLSM this observer converges rapidly to the desired state vector, but not all inertial properties can be evaluated, because of the strong assumptions of the physical model of reference.

A second Adaptive Law is implemented combining the system model formulated according to RLSM and the adaptive observer expressed in the previous analysis.

As shown in Chapter 3, a detailed study helps us to establish that:

- Unlike Recursive Least Square, similar to a common Kalman Filter, both Adaptive Laws converge to the desired solution in a short time lapse;
- A variation of the operating frequency of the nonlinear observer, due to a possible signal delay, is responsible of a great variation of the estimation error for RLSM, while the second Adaptive Law is subject to small changes of the same quantity. For the last algorithm, a significant loss of convergence properties is reached only if the operating frequency is equal

to 1 Hz; in this case It will be necessary to find other optimal tuning parameters.

In this context, further studies and other detailed analysis have to be conducted.

First, the connection element between chaser and target shall be modelled as a robotic arm, so that the proposed algorithms shall be applied on a generic space manipulator, or alternatively, the condition of rigid body can be replaced with a more complex mathematical model, considering the connection as a flexible appendage, characterized by elastic and dissipative terms, similar to tethered CubeSat case.

Finally, because the objective of this thesis is focusing only on the design of the inertial parameters estimator, in the next steps It will be necessary to design a more accurate controller, based on data received from the proposed nonlinear estimator.

Appendix A

A.1. Stability ([17])

According to Lyapunov's definition, an equilibrium point is stable only if

$$\forall \varepsilon, \exists \delta(\varepsilon): |x_0 - x_{eq}| < \delta \rightarrow \left|x(t) - x_{eq}\right| < \varepsilon, \forall t > 0$$

An equilibrium point is asymptotically only if It is stable and

$$\exists \delta: \left|x_0 - x_{eq}\right| < \delta \rightarrow \lim_{t \to \infty} \left|x(t) - x_{eq}\right| = 0$$

A.2. Lyapunov Criterion ([17])

For understanding the logic behind the mathematical discussion about stability of nonlinear systems, G.Oriolo (**[17]**) affirms that if we consider a mechanics system, in which the total energy is dissipated continuously, It goes to an equilibrium point. For this reason It is possible to define a scalar function, called Lyapunov function, for determining stability or instability of the examined system: It is an energy like function, analysed over the time.

Considering the following nonlinear system:

$$\dot{x} = f(x, t) \qquad x \in R^n$$

With an equilibrium point given by

$$f(x_{eq}, t) = 0$$

A general scalar function $V(x) \in C^1$ can be:

- Defined positive only if

  $V(x_{eq}) = 0$

  $V(x) > 0, \forall x \in S(x_{eq}, r), x \neq x\_eq$

- Semi-defined positive only if

  $V(x_{eq}) = 0$

  $V(x) \geq 0, \forall x \in S(x_{eq}, r), x \neq x\_eq$

- Defined negative only if $-V(x)$ is defined positive
- Semi-defined negative only if $-V(x)$ is semi-defined positive

$S(x_{eq}, r)$ is a spherical around of x_{eq} characterized by a radius r.

Theorem 1.

An equilibrium point x_{eq} of a nonlinear system $\dot{x} = f(x, t)$ is stable if

$$\exists V(x) \in C^1 : V(x) \text{ is defined positive in } S(x_{eq}, r)$$

$$\exists V(x) \in C^1 : \dot{V}(x) \text{ is semi-defined negative in the same } S(x_{eq}, r)$$

Theorem 2.

An equilibrium point x_{eq} of a nonlinear system $\dot{x} = f(x, t)$ is asymptotically stable if

$$\exists V(x) \in C^1 : V(x) \text{ is defined positive in } S(x_{eq}, r)$$

$$\exists V(x) \in C^1 \text{: } \dot{V}(x) \text{ is defined negative in the same } S(x_{eq}, r)$$

The mathematical proofs of theorems 1 and 2 are reported in **[17]**.

A.3. Proof: *Convergence of the estimation error calculated with the first Adaptive Law's implementation*

The aim of this section is to demonstrate the convergence of the estimation error obtained applying the first Adaptive Law on the parametrized dynamic system, according to all mathematical results described in A.1-A.2.

Introducing the mass and the state estimation errors as

$$\begin{aligned} \widetilde{m} &= \widehat{m} - m \\ \dot{\widetilde{m}} &= \dot{\widehat{m}}^{21} \\ e(t) &= \hat{q}(t) - q(t) \\ \dot{e}(t) &= \dot{\hat{q}}(t) - \dot{q}(t) \end{aligned} \tag{A.1}$$

Replacing (A.1) in the equation (3.8), we have:

$$C^* \dot{e}(t) + K^* e(t) + \widetilde{m}_1 (M_{1t} \dot{q} + C_{1t} q) + \widetilde{m}_2 (M_{1c}\, \dot{q} \; + C_{1c} q)$$

[21] This hypothesis is valid only if the real masses are constant over the time.

$$+\tilde{m}_3(M_{2t}\,\dot{q}\; + C_{2t}q) + \tilde{m}_4(M_{2c}\,\dot{q}\; + C_{2c}q) + \tilde{m}_5(M_{3t}\,\dot{q}\; + C_{3t}q) + \tilde{m}_6(M_{3c}\,\dot{q}\; + C_{3c}q) = 0 \tag{A.2}$$

Exploiting the same mathematical procedure discussed in **[11]**, We define the candidate Lyapunov function:

$$V = \frac{1}{2}\, e^T(t)C^*e(t) + \frac{1}{2\gamma_1}\tilde{m}_1^2 + \frac{1}{2\gamma_2}\tilde{m}_2^2 + \frac{1}{2\gamma_3}\tilde{m}_3^2 + \frac{1}{2\gamma_4}\tilde{m}_4^2 + \frac{1}{2\gamma_5}\tilde{m}_5^2 + \frac{1}{2\gamma_6}\tilde{m}_6^2 \tag{A.3}$$

$$\dot{V} = e^T(t)C^*\dot{e}(t) + \frac{1}{\gamma_1}\tilde{m}_1\dot{\tilde{m}}_1 + \frac{1}{\gamma_2}\tilde{m}_2\dot{\tilde{m}}_2 + \frac{1}{\gamma_3}\tilde{m}_3\dot{\tilde{m}}_3 + \frac{1}{\gamma_4}\tilde{m}_4\dot{\tilde{m}}_4 + \frac{1}{\gamma_5}\tilde{m}_5\dot{\tilde{m}}_5 + \frac{1}{\gamma_6}\tilde{m}_6\dot{\tilde{m}}_6 =$$

$$= e^T(t)C^*\dot{e}(t) + e^T(t)\left[\frac{1}{\gamma_1}\tilde{m}_1\gamma_1(M_{1t}\dot{q} + C_{1t}q) + \frac{1}{\gamma_2}\tilde{m}_2\gamma_2(M_{1c}\dot{q} + C_{1c}q) + \frac{1}{\gamma_3}\tilde{m}_3\gamma_3(M_{2t}\dot{q} + C_{2t}q) + \frac{1}{\gamma_4}\tilde{m}_4\gamma_4(M_{2c}\dot{q} + C_{2c}q) + \frac{1}{\gamma_5}\tilde{m}_5\gamma_5(M_{3t}\dot{q} + C_{3t}q) + \frac{1}{\gamma_6}\tilde{m}_6\gamma_6(M_{3c}\dot{q} + C_{3c}q)\right]$$

$$= e^T(t)C^*\dot{e}(t) + e^T(t)[-C^*\dot{e}(t) - K^*e(t)]$$

$$= -e^T(t)K^*e(t) \le -\lambda_{min}(K^*)\left|\left|e(t)\right|\right|^2 \le 0 \tag{A.4}$$

$$\ddot{V} = -2e^T(t)K^*\dot{e}(t) \le 0 \tag{A.5}$$

This result is true, because $\dot{e}(t)$ is bounded, so $q(t)$ is bounded: according to the Barbalat's lemma, this guarantees that the estimation error goes to 0, when time goes to infinity.

The graphical results, shown in the chapter 3, demonstrate the validity of the mathematical proof analysed previously.

A.4. Proof: *Convergence of the estimation error calculated with the second Adaptive Law's implementation*

The aim of this section is to demonstrate the convergence of the estimation error, exploiting the Lyapunov Criterion. We define

$$e(t) = \hat{q}(t) - q$$

$$\dot{e}(t) = \dot{\hat{q}}(t) - \dot{q}(t)$$

$$\tilde{\theta}(t) = \hat{\theta} - \theta$$

$$\dot{\tilde{\theta}} = \dot{\hat{\theta}}$$

(A.6)

The candidate Lyapunov function is:

$$V = \frac{1}{2}\, e^T(t) C^* e(t) + \frac{1}{2}\, \tilde{\theta}(t) Q^{-1} \tilde{\theta}(t)$$

(A.7)

$$\begin{aligned} \dot{V} &= e^T(t) C^* e(t) + \dot{\tilde{\theta}}^T(t) Q^{-1} \tilde{\theta}(t) \\ &= e^T(t)\left[-K^* e(t) - \varphi\tilde{\theta}\right] + \left(Q\varphi^T e(t)\right)^T Q^{-1}\tilde{\theta}(t) \\ &= e^T(t)\left[-K^* e(t) - \varphi\tilde{\theta}\right] + e^T(t)\varphi Q^T Q^{-1}\tilde{\theta}(t) \end{aligned}$$

(A.8)

If Q is a diagonal defined-positive matrix, $Q^T = Q$

For diagonal matrices: $Q^T Q^{-1} = QQ^{-1} = Q^{-1}Q = I$

$$\begin{aligned}\dot{V} &= e^T(t)\left[-K^* e(t) - \varphi\tilde{\theta}\right] + e^T(t)\varphi Q^T Q^{-1}\tilde{\theta}(t) \\ &= e^T(t)\left[-K^* e(t) - \varphi\tilde{\theta}\right] + e^T(t)\varphi\tilde{\theta}(t) \\ &= -e^T(t)K^* e(t) - e^T(t)\varphi\tilde{\theta} + e^T(t)\varphi\tilde{\theta}(t) = -e^T(t)K^* e(t)\end{aligned}$$

Using the same results of the appendix A.3, It is possible to affirm that:

$$\dot{V} = -e^T(t)K^* e(t) \leq -\lambda_{min}(K^*)\left||e(t)|\right|^2 \leq 0$$

$$\ddot{V} = -2e^T(t)K^*\dot{e}(t) \leq 0 \tag{A.9}$$

This result is true, because $\dot{e}(t)$ is bounded, so $q(t)$ is bounded: according to the Barbalat's lemma, this guarantees that the estimation error goes to 0, when time goes to infinity.

The graphical results, shown in the chapter 3, demonstrate the validity of the mathematical proof analysed previously.

Appendix B

In this appendix, the convergence properties of the second Adaptive Law are reported, fixing:

- The observer's reference frequency at 1 Hz;
- Initial conditions and gain matrices.

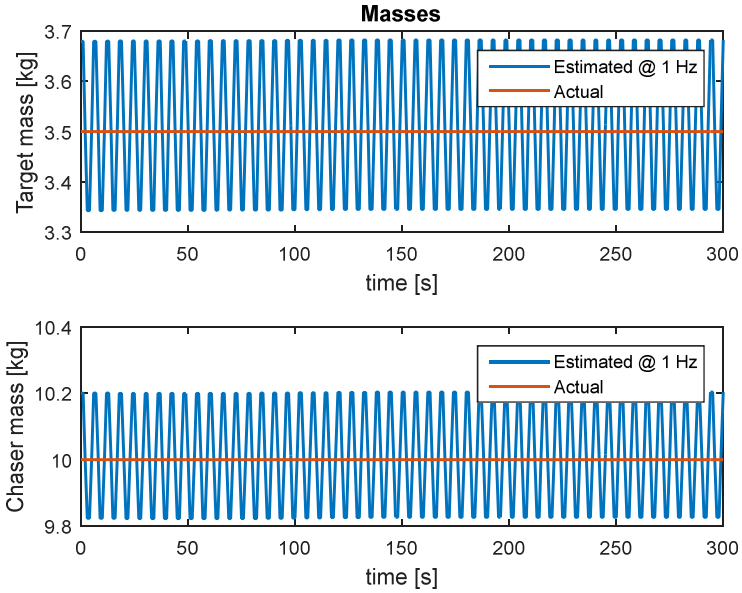

Figure 33. Mass estimation using the Adaptive Law @ 1 Hz

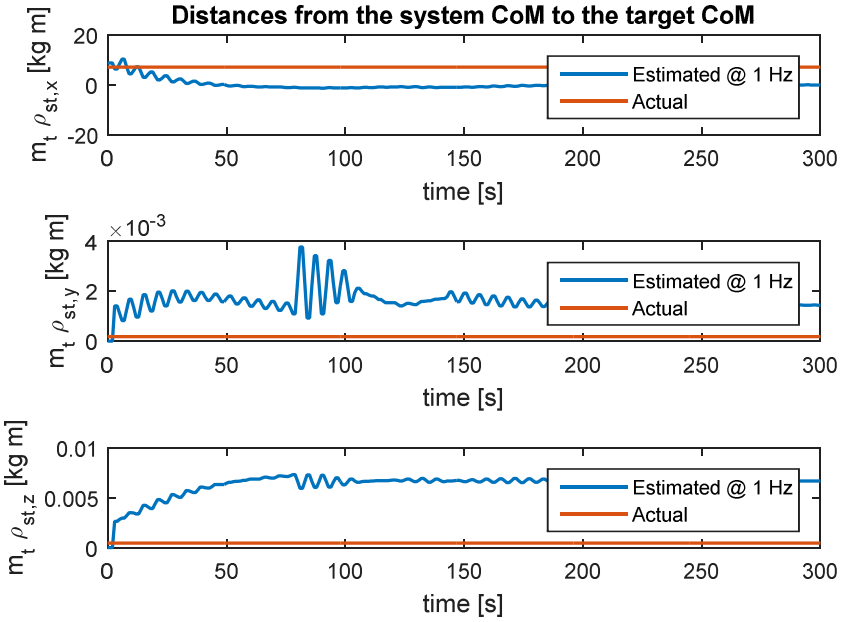

Figure 34. Estimation of the relative distances between the system's CoM and the target's one using the Adaptive Law @ 1 Hz

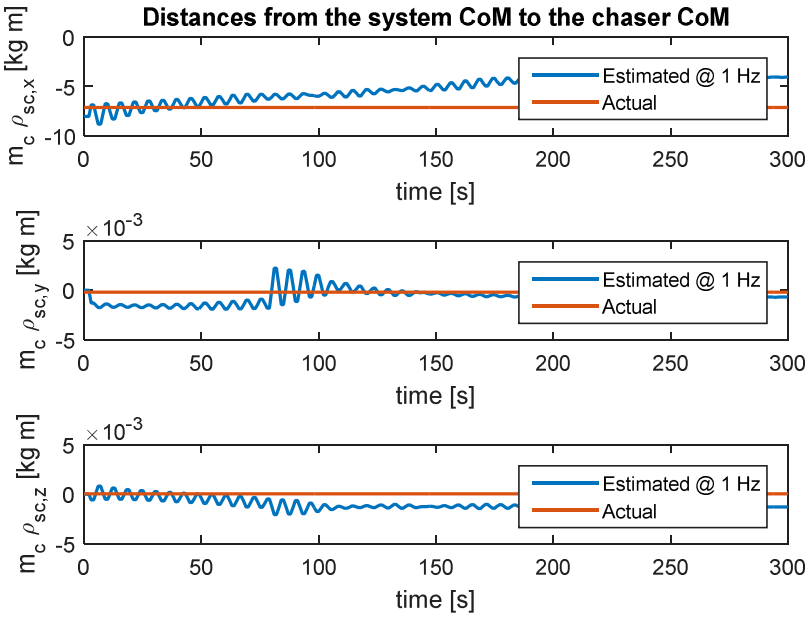

Figure 35. Estimation of the relative distances between the system's CoM and the chaser's one using the Adaptive Law @ 1 Hz

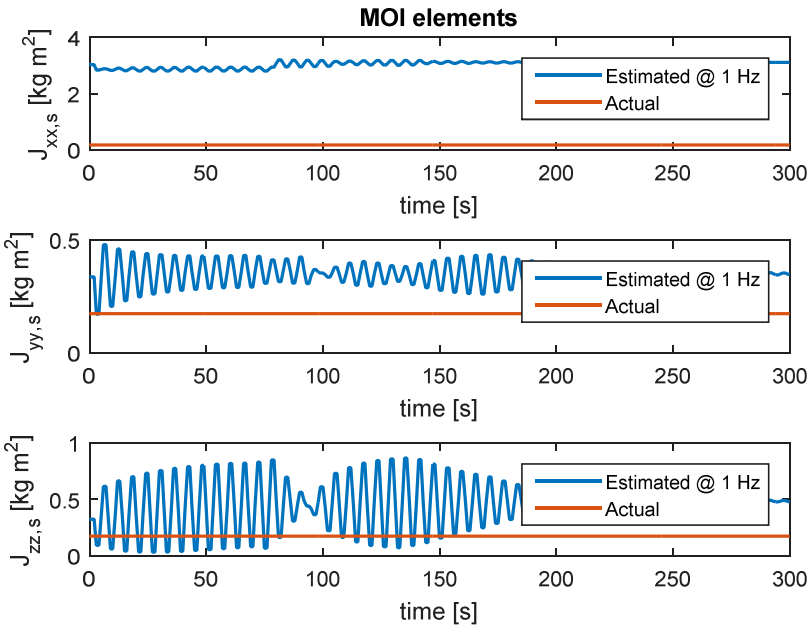

Figure 36. MOI elements estimation using the Adaptive Law @ 1 Hz

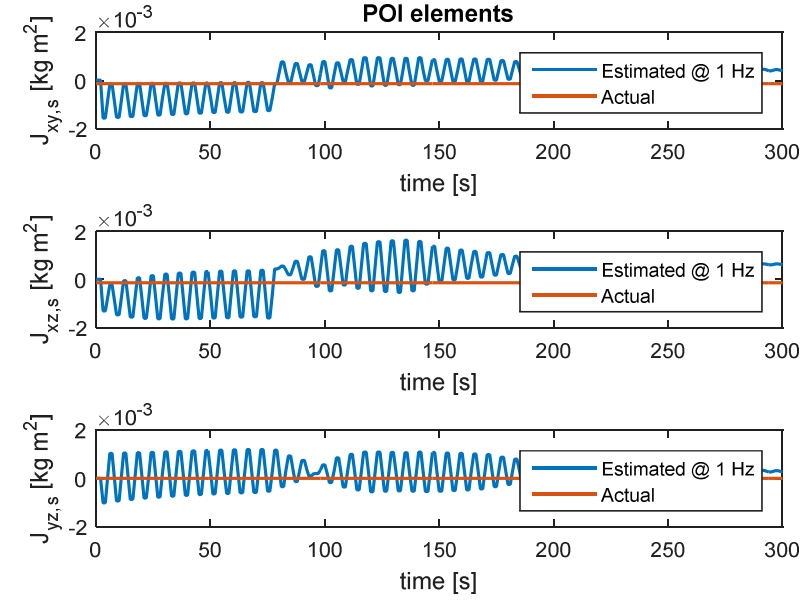

Figure 37. POI elements estimation using the Adaptive Law @ 1 Hz

References

[1] G. Avanzini, "*Spacecraft Attitude Dynamics and Control. Dispense del corso di Dinamica e Controllo di Assetto di Satelliti*", Vers. 3.0.0

[2] E. Capello, *"Dinamica e Controllo dei veicoli spaziali",* Academic material related to the Master's Degree in Aerospace Engineering course.

[3] Shan, M.; Shi, L. "*Comparison of Tethered Post-Capture System Models for Space Debris Removal*". Aerospace 2022, 9,33. https://doi.org/10.3390/aerospace9010033

[4] C. Mencuccini and V.Silvestrini, "*Fisica I, Meccanica e Termodinamica. Corso di fisica per le facoltà tecnico-scientifiche corredato di esempi ed esercizi*", Liguori Editore

[5] Y.Ikeda, T.Kida and T.Nagashio, "*Stabilizing Nonlinear Adaptive Control for Spacecraft Before and After Capture*".

[6] W. Fehse "*Automated Rendezvous and Docking of Spacecraft*", Cambridge Aerospace Series

[7] A. Eriksson, "*Implementation and Evaluation of a Mass Estimation Algorithm*"

[8] D. Kim, S. B. Choi and J. Oh, "*Integrated Vehicle Mass Estimation Using Longitudinal and Roll Dynamics*" 12th International Conference of Control, Automation and Systems

[9] F. Yuan, X. Lu, Y. Zhuoping and Q. Tong, "*Recursive Least Square Vehicle Mass Estimation Based on Acceleration Partition*", Chinese Journal of Mechanical Engineering

[10] Bellar A and Si Mohammed MA (2019), "*Satellite Inertia Parameters Estimation Based on Extended Kalman Filter*". J Aerosp Technol Manag, 11: e1619.
https://doi.org/10.5028/jatm.v11.1016

[11] H. Lee, S. Kim and J. Kim, "*Control of an Aerial Manipulator using On-line Parameter Estimator for an Unknown Payload*"

[12] M. Ekal and R. Ventura, "*An Energy Balance Based Method for Parameter Identification of a Free-Flying Robot Grasping an Unknown Object*"

[13] S. Yang, S. Lee, J.H. Lee and H. Oh, "*New Real-Time Estimation Method for Inertia Properties of STSAT-3 using Gyro Data*", Trans. Japan Soc. Aero. Space Sci. Vol. 58, No. 4, pp. 247-249, 2015

[14] N.A. Chaturvedi, D.S. Bernstein, J. Ahmed, F. Bacconi and N.H. McClamroch, "*Globally Convergent Adaptive Tracking of Angular Velocity and Inertia Identification for a 3-DOF Rigid Body*". IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, VOL. 14, NO. 5, SEPTEMBER 2006

[15] J. Ahmed, V. T. Coppola and D.S. Bernstein, "*Adaptive Asymptotic Tracking of Spacecraft Attitude Motion with Inertia Matrix Identification*". JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS Vol. 21, No. 5, September–October 1998

[16] A. De Luca, "*Adaptive Trajectory Control. Robotics 2*"

[17] G. Oriolo, "*Teoria della stabilità per sistemi non lineari*"

[18] M. E.A. Cheriet, A. Bellar, M.Y. Ghaffour, A. Adnane and M. A SI Mohammed, "*Inertia tensor estimation for a rigid nadir pointing satellite based on star tracker*" Advances in Aircraft and Spacecraft Science, Vol. 8, No. 2 (2021) 111-126
DOI: https://doi.org/10.12989/aas.2021.8.2.111

[19] E. Wilson, D.W. Sutter and R.W. Mah, "*MCRLS for On-Line Spacecraft Mass-and Thruster- Property Identification*" Proceedings of the IASTED International Conference on Intelligent Systems and Control, Honolulu, HI, Aug 2004

[20] R.E. Bordany, W.H. Steyn and M. Crwford, "*In-Orbit Estimation of the Inertia Matrix and Thruster Parameters of UoSAT-12*". 14th AIAA/USU Conference on Small Satellites

[21] Zhang, L. "*Configuration Optimization for Free-Floating Space Robot Capturing Tumbling Target*". Aerospace 2022, 9, 69.
https://doi.org/10.3390/aerospace9020069

[22] N. Mavrakis and R. Stolkin, "*Estimation and Exploitation of Objects' Inertial Parameters in Robotic Grasping and Manipulation: A Survey*"

[23] A.M. de Oliveira, H.K. Kuga and V. Carrara, "*Estimating the Mass Characteristics of a Dumbbell Air Bearing Satellite Simulator*". 22nd International Congress of Mechanical Engineering (COBEM 2013) November 3-7, 2013, Ribeirão Preto, SP, Brazil

[24] D. Kim, H.S. Oh and D. Choi, "*Inertia Estimation of Spacecraft Based on Modified Law of Conservation of Angular Momentum*". Journal of Astronomy & Space Sciences · December 2010
DOI: 10.5140/JASS.2010.27.4.353

[25] D.Kim, S. Yang and S. Lee, "*Rigid Body Inertia Estimation Using Extended Kalman and Savitzky-Golay Filters*". Hindawi Publishing Corporation Mathematical Problems in Engineering Volume 2016, Article ID 2962671, 7 pages http://dx.doi.org/10.1155/2016/2962671

[26] A.K. Sanyal, M. Chellappa, J.L. Valk, J. Ahmed, J. Shen, D.S. Bernstein, "*Globally Convergent Adaptive Tracking of Spacecraft Angular Velocity With Inertia Identification And Adaptive Linearization*". Proceedings of the 42nd IEEE Conference on Decision and Control Maui, Hawaii USA, December 2003

[27] Leboutet Q., Roux, J., Janot A., Guadarrama-Olvera, J.R. and Cheng, G. "*Inertial Parameter Identification in Robotics: A Survey*". Appl. Sci. 2021, 11, 4303.
https://doi.org/10.3390/app11094303

[28] H.D. Curtis, "*Orbital Mechanics for Engineering Students*", II edition, Elsevier Aerospace Engineering Series