sid.inpe.br/mtc-m21c/2020/08.09.05.32-TDI

OPTIMIZATION IN BERTHING MANEUVERS OF SATELLITES ENDOWED WITH ROBOTIC MANIPULATORS

Anderson Brazil Nardin

Doctorate Thesis of the Graduate Course in Space Engineering and Technology / Space Mechanics and Control Division, guided by Dr. Evandro Marconi Rocco, and Dr. Heike Benninghoff, approved on August 07, 2020

URL of the original document:
<http://urlib.net/8JMKD3MGP3W34R/432UMHP>

INPE
São José dos Campos
2020

PUBLISHED BY:

Instituto Nacional de Pesquisas Espaciais - INPE
Gabinete do Diretor (GBDIR)
Serviço de Informação e Documentação (SESID)
CEP 12.227-010
São José dos Campos - SP - Brasil
Tel.:(012) 3208-6923/7348
E-mail: pubtc@inpe.br

sid.inpe.br/mtc-m21c/2020/08.09.05.32-TDI

OPTIMIZATION IN BERTHING MANEUVERS OF SATELLITES ENDOWED WITH ROBOTIC MANIPULATORS

Anderson Brazil Nardin

Doctorate Thesis of the Graduate Course in Space Engineering and Technology / Space Mechanics and Control Division, guided by Dr. Evandro Marconi Rocco, and Dr. Heike Benninghoff, approved on August 07, 2020

URL of the original document:
<http://urlib.net/8JMKD3MGP3W34R/432UMHP>

INPE
São José dos Campos
2020

Cataloging in Publication Data

Nardin, Anderson Brazil.

N166o Optimization in berthing maneuvers of satellites endowed with robotic manipulators / Anderson Brazil Nardin. – São José dos Campos : INPE, 2020.

xxx + 131 p. ; (sid.inpe.br/mtc-m21c/2020/08.09.05.32-TDI)

Thesis (Doctorate in Engineering and Space Technology/Spatial Mechanics and Control) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2020.

Guiding : Drs. Evandro Marconi Rocco and Heike Benninghoff.

1. Spacecraft maneuvers. 2. Hardware-in-the-loop simulation. 3. Robotics. 4. Orbital servicing. 5. Optimization. I.Title.

CDU 629.7.017.3:629.78

Aluno (a): *Anderson Brazil Nardin*

Título: "OPTIMIZATION IN BERTHING MANEUVERS OF SATELLITES ENDOWED WITH ROBOTIC MANIPULATORS"

Aprovado (a) pela Banca Examinadora em cumprimento ao requisito exigido para obtenção do Título de *Doutor(a)* **em**

Engenharia e Tecnologia Espaciais/Mecânica Espacial e Controle

Dr. Mario Cesar Ricci

Presidente / INPE / SJCampos - SP

(X) Participação por Video - Conferência

(X) Aprovado ( ) Reprovado

Dr. Evandro Marconi Rocco

p/

Orientador(a) / INPE / SJCampos - SP

(X) Participação por Video - Conferência

(X) Aprovado ( ) Reprovado

Dra. Heike Benninghoff

p/

Orientador(a) / DLR / Alemanha - DE

(X) Participação por Video - Conferência

(X) Aprovado ( ) Reprovado

Dr. Ijar Milagre da Fonseca

p/

Convidado(a) / ITA/DCTA / São José dos Campos - SP

(X) Participação por Video - Conferência

(X) Aprovado ( ) Reprovado

Dr. Aguinaldo Cardozo da Costa Filho

p/

Convidado(a) / IFSP / São José dos Campos - SP

(X) Participação por Video - Conferência

(X) Aprovado ( ) Reprovado

Este trabalho foi aprovado por:

( ) maioria simples

(X) unanimidade

São José dos Campos, 07 de agosto de 2020

Aprovado (a) pela Banca Examinadora em cumprimento ao requisito exigido para obtenção do Título de *Doutor(a)* **em**

Engenharia e Tecnologia Espaciais/Mecânica Espacial e Controle

Dr. Alexandre Carvalho Leite

Convidado(a) / IFF / Campos dos Goytacazes - RJ

(X) Participação por Video - Conferência

(X) Aprovado *( ) Reprovado*

Este trabalho foi aprovado por:

( ) maioria simples

(X) unanimidade

São José dos Campos, 07 de agosto de 2020

“Computers are useless. They can only give you answers.”

Pablo Picasso

To the future minds to whom it may be useful.

ACKNOWLEDGEMENTS

There is no nobler profession among human beings than that of a teacher/professor, and knowledge is the greatest of all possessions that one can possess; defies mathematics, and multiplies when one divides; at a certain point, it is impossible to distinguish between what one is and what one knows; possession and possessor enjoy the same existence. Therefore, those who subdue their vanities, and choose to share what they are, reaffirm what is the essence of bestowal and live while they last in the memories of those who once dared to teach.

I would like to thank professors Dr. Mário César Ricci, Dr. Ijar Milagre da Fonseca and especially my advisor Dr. Evandro Marconi Rocco for their advice, support, and friendship.

I am grateful to many people. Everybody from my family who understood my often absence. All friends who helped me while I was in Germany, especially Paulo Victor Padrão Lopes and André Fialho Coelho.

I would also like to thank my colleagues from the On-Orbit Servicing and Autonomy group of the Space Flight Technology department at DLR, Heike Benninghoff, Florian Rems, Eicke-Alexander Risse, Kurt Schwenk and Ksenia Klionovska for their assistance and friendship, without them the results presented here would not have been possible.

Finally, but not less important, I thank CAPES and CNPq for its financial support.

ABSTRACT

The study of On-Orbit Servicing (OOS) missions has shown the necessity for developing a methodology able to provide solutions for berthing maneuvers of artificial satellites, being the chaser endowed with a robotic manipulator, applying the multi-objective optimization of conflicting objectives. This work aspires to develop a multi-objective optimization approach that aims to find a balanced solution among conflicting objectives. Movement accuracy, attitude maintenance, maneuver time, and energy consumption from different sources are going to be the selected criteria for optimization due to their great importance, despite the inherent difficulty for simultaneous optimization that they impose on berthing maneuvers of artificial satellites. The approach of this work focuses on the disturbances the robotic arm and base satellite cause to each other. Such disturbances are considered torques generated by the coupling between the robotic manipulator and its base satellite through their distinct control systems. The robotic arm configuration allows diverse applications and notable usefulness in the accomplishment of OOS. The results showed that it was possible to test and validate the developed simulation environment for berthing maneuvers through real-time and hardware-in-the-loop (HIL) simulations using the European Proximity Operations Simulator (EPOS) at German Aerospace Center (DLR). In this scenario, two physical robots play the role of chaser and target satellites involved in the maneuver, while a virtual robotic manipulator coupled to the chaser satellite is simulated by software. This work was successful in creating reliable software for tests of berthing maneuvers since the developed algorithms found balanced solutions among conflicting objectives.

Keywords: Spacecraft maneuvers. Hardware-in-the-loop simulation. Robotics. Orbital servicing. Optimization.

OTIMIZAÇÃO EM MANOBRAS DE ATRACAÇÃO DE SATÉLITES DOTADOS DE MANIPULADORES ROBÓTICOS

RESUMO

O estudo de missões de *On-Orbit Servicing* (OOS) tem mostrado a necessidade de desenvolver uma metodologia capaz de fornecer soluções para manobras de atracação de satélites artificiais, sendo o perseguidor dotado de um manipulador robótico, aplicando a otimização multiobjetivo de objetivos conflitantes. Este trabalho aspira a desenvolver uma abordagem de otimização multiobjetivo que visa encontrar uma solução equilibrada entre objetivos conflitantes. A precisão do movimento, a manutenção da atitude, o tempo de manobra e o consumo de energia de diferentes fontes serão os critérios selecionados para otimização devido à sua grande importância, apesar da dificuldade inerente à otimização simultânea que eles impõem às manobras de atracação de satélites artificiais. A abordagem deste trabalho enfoca os distúrbios que o braço robótico e o satélite base causam um ao outro. Tais distúrbios são considerados torques gerados pelo acoplamento entre o manipulador robótico e seu satélite base por meio de seus distintos sistemas de controle. A configuração do braço robótico permite diversas aplicações e notável utilidade na realização de OOS. Os resultados mostraram que foi possível testar e validar o ambiente de simulação desenvolvido para manobras de atracação por meio de simulações *real-time* e com *hardware-in-the-loop* (HIL) usando o *European Proximity Operations Simulator* (EPOS) no Centro Aeroespacial Alemão (DLR). Nesse cenário, dois robôs físicos desempenham o papel de satélites caçador e alvo envolvidos na manobra, enquanto um manipulador robótico virtual acoplado ao satélite caçador é simulado por software. Este trabalho foi bem-sucedido na criação de software confiável para testes de manobras de atracação, uma vez que os algoritmos desenvolvidos encontraram soluções equilibradas entre objetivos conflitantes.

Palavras-chave: Manobras de espaçonaves. Simulação com hardware na malha. Robótica. Serviços em órbita. Otimização.

LIST OF FIGURES

Page

LIST OF TABLES

Page.

LIST OF SYMBOLS

a_n	length of link n
c_i	cosine of θ_i
$c(t)$	control signal
$\boldsymbol{C}_{ba}$	transformation matrix from system a to b
$\boldsymbol{C}_{ba}^{T}$	transpose of transformation matrix from system a to b
$\boldsymbol{C}_{ba_n}$	transformation matrix from system a to b in step n
$\boldsymbol{C}_{ca}$	satellite attitude matrix
$\boldsymbol{C}_n$	rotation matrix around axis n
C_m	center of mass
dm	infinitesimal mass element
$er(t)$	error signal
Er	joint error
$\boldsymbol{f}$	resulting external forces applied to the rigid body
${}^{i}\vec{f}_i$	force exerted by joint i with relation to the system i
${}^{i+1}\vec{F}_{i+1}$	actuating force on the link $i+1$ with relation to the system $i+1$
$\boldsymbol{F}_d$	feasible area of the decision space
$g(\boldsymbol{x})$	constraint function
$\boldsymbol{g}_c$	resulting external torques applied to the rigid body on its center of mass

h object height

$\boldsymbol{h}_c$ angular momentum with respect to the center of mass

$\dot{\boldsymbol{h}}^{b}{}_{c}$ time derivative of angular momentum on the center of mass

$\boldsymbol{I}$ moment of inertia

$\boldsymbol{I}_a$ moment of inertia in the system a

${}^{Cs}\boldsymbol{I}_a$ moment of inertia tensor of the link a on the center of mass

${}^{Cs}\boldsymbol{I}_S$ moment of inertia tensor of the satellite on the center of mass

${}^{Cs}\boldsymbol{I}_T$ total moment of inertia tensor of the arrangement of robotic arm and satellite with respect to the satellite's center of mass

${}^{A}\boldsymbol{I}$ moment of inertia tensor with respect to system A

I_{ab} inertia matrix element from row a and column b

$\boldsymbol{I}_3$ identity matrix with dimensions *3x3*

J_n joint n

K_D derivative gain

K_I integral gain

K_P proportional gain

l object length

m object mass

M_A link A mass

M_S satellite mass

$^{i}\vec{n}_{i}$ torque exerted by joint i with respect to system i

$^{i+1}\vec{N}_{i+1}$actuating torque on the link $i+1$ with respect to the system $i+1$

O inertial system origin

O' spinning system origin

p number of objective functions

$\boldsymbol{p}$ linear momentum

$\dot{\boldsymbol{p}}^{b}$ time derivative of linear momentum of the rigid body

$\boldsymbol{P}_{c}$ matrix representing the vector of center of mass

$\vec{P}_{CA}$ vector center of mass of the link A

$\vec{P}_{CM}$ vector center of mass of the arrangement of robotic arm and satellite

$\vec{P}_{CS}$ vector center of mass of the satellite

$\vec{P}_{SR}$ vector distance between satellite center and the robot

$^{i}\vec{P}_{C_i}$ vector distance between system i origin to the link i center

$^{i}\vec{P}_{i+1}$ vector distance between consecutive systems

${}^{B}_{A}\boldsymbol{R}$ rotation matrix from system A to B

s_i sine of θ_i

$\dot{\mathbf{v}}_{\mathrm{c}}^{\mathrm{b}}$ time derivative of the linear velocity of the body’s center of mass

$^{i}\dot{\vec{v}}_{j}$ time derivative of the linear velocity of the link j in the system i

${}^{i}\dot{\vec{v}}_{C_i}$ time derivative of the linear velocity of the link i center of mass in the system i

${}^{i}\vec{v}_{j}$ linear velocity of the link j in the system i

$\mathbf{v}_{\mathrm{c}}$ linear velocity of the center of mass

$\vec{V}_{ag}$ target vector in the robot system, desired vector for the wrist

$\vec{V}_{CT}$ target vector in the arrangement center of mass system

$\vec{V}_{OT}$ target vector in the system O

w object width

w_k objective weight number k

x_c coordinate of the center of mass position vector in the axis x

$\hat{X}_i$ unit vector (versor) i in the axis x

$\mathbf{x}$ vector of decision variables

y_c coordinate of the center of mass position vector in the axis y

$\hat{Y}_i$ unit vector (versor) i in the axis y

$\mathbf{z}$ objective vector of individually optimized objectives

z_c coordinate of the center of mass position vector in the axis z

$\hat{Z}_i$ unit vector (versor) i in the axis z

$\mathbf{Z}(\mathbf{x})$ objective function to be optimized

$\mathbf{Z}(\mathbf{x}^{\mathrm{b}})$ the most balanced candidate in the objective space

$\mathbf{Z}(\mathbf{x}^*)$	solution with the smallest loss for all objectives
$\dot{\theta}_i$	time derivative of the angular position of joint i
$\ddot{\theta}_i$	second time derivative of the angular position of joint i
$\boldsymbol{\theta}^\times$	skew-symmetric matrix of the infinitesimal angles vector
θ_i	joint variable i
ρ	object volumetric density
$\boldsymbol{\omega}$	angular velocity
${}^i\dot{\vec{\omega}}_j$	time derivative of the angular velocity of the link j in the system i
$\dot{\boldsymbol{\omega}}^b$	time derivative of the angular velocity of the rigid body
${}^i\vec{\omega}_j$	angular velocity of the link j with respect to the system i
$\boldsymbol{\omega}^\times$	skew-symmetric matrix of the angular velocities

CONTENTS

Page

1 INTRODUCTION

Optimization certainly plays a prominent role in engineering. It is the branch of mathematics that deals with the question "what is the best way to do this?". The search for the answer represents a challenge commonly overcome by nature and faced by designers who dare to subject their creations to the strict criteria that define the best.

Optimization, therefore, consists in the maximization or minimization, according to the defined objective, of objective functions, aiming to find an “optimal solution”, that is, understood as the best possible (WISMER; CHATTERGY, 1978).

By considering more than one goal concurrently in an optimization problem, we add complexity. This raises the difficulty of obtaining a solution, especially in the case of conflicting objectives, i.e., the respective optimizations are mutually exclusive.

The minimization of maneuvering time combined with the minimization of fuel consumption is a notable multi-objective optimization problem in space environment. As we have noted, time and consumption are often conflicting objectives since minimizing time requires higher execution speed and, consequently, increased consumption by satellite actuators. Minimizing consumption is usually related to decreased speed, thus increasing the time required to perform the maneuver (ROCCO, 2002).

Similarly to optimization, Guidance, Navigation, and Control (GNC), as well as robotic systems, play essential roles in many spacecraft missions. In On-Orbit Servicing (OOS) missions, a service satellite (chaser) approaches a client satellite (target) aiming, for example, to extend client’s lifetime or for safe end-of-life de-orbiting (NISHIDA et al., 2009; ELLERY; KREISEL; SOMMER, 2008; FLORES-ABAD et al., 2014). For example, the Mission Extension Vehicle (MEV) which can dock with existing satellites providing the attitude control needed to extend their lives (GEBHARDT, 2020). The client may be damaged or out of fuel and it is necessary to perform a re-entry in the Earth's atmosphere or to take the satellite to a graveyard orbit. OOS contributes to space debris removal since non-operational satellites may represent a hazard in space environment (NISHIDA et al., 2009). Finally, OOS can be done for assembly of large structures in space, such as modular spacecrafts, to enhance future space exploration missions (GRALLA; WECK, 2007).

Many other OOS missions have been planned and currently have their development being achieved for future launch. For example, the On-Orbit Servicing, Assembly and Manufacturing mission 1 (OSAM-1) (NASA, 2020), the Robotic Servicing of Geosynchronous Satellites (RSGS) (DARPA, 2020), the End-of-Life Service by Astroscale (ELSA) (FORSHAW et al., 2019) and the ClearSpace-1 (ESA, 2019).

The issue that will be addressed in this work is the analysis of artificial satellites equipped with robotic manipulators for the execution of berthing maneuvers, i.e., reaching target points. In other words, the way robotic manipulators that have artificial satellites as their base perform the task of taking their end effectors to specific points in space.

Accuracy can be understood as the ability of a manipulator (arm) to bring its end effector (wrist) to a target point within the workspace, while the reach is defined by the farthest attainable point from its base (respecting imposed constraints).

When designers focus on the task of designing a robotic manipulator, a recurring difficulty emerges from the core of the problem. How to deal with the accuracy of the movement that is impaired as we meet another desired goal? We are keen to maximize the reach of the manipulator, unfortunately, this procedure is related to the loss of accuracy. Such an inextricable duality of the problem occurs because of the physical characteristics of the manipulators, especially those of anthropomorphic configuration, which resemble a human arm (KUTTAN, 2007; GROOVER, 2019; SICILIANO et al., 2009).

On the other hand, if we restrict the region in which the movements are performed, to improve accuracy, we face the inseparable decrease of reach, which limits the manipulator's ability to perform tasks and requires greater action by the satellite, so that the target point becomes achievable.

In general, the action of actuators is associated with the duality of time and energy required to perform the maneuver. We realize, therefore, that if the robotic manipulator and artificial satellite are jointly responsible for the execution of the total maneuver, that is, the berthing, the total action is divided between them and the portion of responsibility of the manipulator has impact on its necessary reach.

Accuracy and reach are always on the agenda when roboticists analyze the requirements of a project. Both concepts represent conflicting goals, as accuracy varies within the workspace and the shorter the distance between the end effector and base, the better the accuracy.

In this work, more detailed explanations of reach and accuracy duality will be presented, but at this point, intuition naturally leads us to understand the problem. Who has never approached the workbench to perform a delicate task? Something like a solder, for example. Accuracy has to do with the addressable points of the robotic manipulator, which define the motion resolution, and the steps of its joints. Respecting trigonometric ratios, the accuracy degrades in the pursuit of longer reaches.

It is in this scenario of conflicts among the objectives such as energy consumption from different sources, maneuver time, attitude maintenance and accuracy of movement that this work is inserted, dealing with how optimization can be obtained in this kind of problem.

Based on simulations, observations, and previous works, it was hypothesized that a novel methodology for the approach of multi-objective optimization problems would bring benefits to the chore of finding balanced solutions avoiding inherent and unwanted characteristics of older methods.

This document also contains descriptions of the experiments conducted with the European Proximity Operations Simulator (EPOS) at German Aerospace Center (DLR). Given the research power of the DLR facilities and the enormous empirical capacity that EPOS represents, it was possible to introduce the intended experimentation ideas.

The EPOS facility's robots were used to obtain outcomes that confirm those from previous computer simulations, adding realism and reliability to the whole research, thanks to the inclusion of hardware in the control loop. In the laboratory, two robotic manipulators emulate the moves to which the target and chaser satellites are subject to promote the attainment of valuable results (BENNINGHOFF; REMS; BOGE, 2014; BENNINGHOFF et al., 2018; BOGE; BENNINGHOFF; TZSCHICHHOLZ, 2011; SANTOS et al., 2016).

In the simulations, one of the physical robots acts as the chaser satellite that serves as base to the attached virtual robotic manipulator, which, by its turn, is simulated by software. The second physical robot acts as target satellite being pursued. The chaser and virtual arm have their distinct control systems to fulfill the berthing maneuver, i.e., to achieve a target point within robotic arm work volume. Such a point is the target satellite position.

The system’s components, being the robotic manipulator role played by the developed software simulator and the chaser satellite role played by one of the robots physically available, are going to be subject to disturbances resulting from the other’s movement (NARDIN, 2019).

The ensemble formed by the chaser satellite and robotic arm is a hybrid system formed by a physically emulated satellite, a robot performs its pose, and a virtual simulated manipulator, a software provides its motion. Once both parts are attached (virtually), the software-based arm and the hardware-in-the-loop (HIL) experience a situation of mutual sensitivity. In this case, the HIL concept is explored, aiming to ascertain the maneuver effectiveness.

Note that, in this work, the term HIL refers to the mentioned hardware inserted in the control loop accordingly to its use by the On-Orbit Servicing and Autonomy group of the Space Flight Technology department at DLR. Indeed, the satellites involved in the maneuvers are emulated by two robots in a hybrid simulation with the software-based arm.

This work can be understood as a natural sequel to the results obtained in Nardin (2015), where the berthing maneuver simulation tool was elaborated and had its functionalities explored, however, without the perspective of applied optimization. Such an approach is finally brought to light in this document.

1.1. Hypotheses

To emphasize, the main hypothesis is the possibility of application of a multi-objective optimization methodology to the problem of berthing maneuvers of artificial satellites endowed with robotic manipulators. To test this, it has been developed the necessary theory for modeling of all different aspects involved with the question: satellite attitude and robot dynamics, control systems, multi-objective optimization algorithms, etc.

In Nardin (2015), with the implementation of computational models, it was obtained simulation outcomes allowing to assess the system behavior. Such satellite plus robotic manipulator system simulations enabled the verification of implications and forecasts in satellite berthing maneuvers. These results confirmed the modeling performed.

Using hardware-in-the-loop simulations, it has been generated new results to verify implications in berthing maneuvers and the possibility of usage of the developed models jointly with the available hardware.

Once all results have corroborated the proposed theory and models, the hypotheses are going to be considered true. This allows forecasting the system behavior when it is respected the defined modeling limitations, e.g., their control gains, workspace limit, the existence of necessary optimization variables, hardware demands, environmental disturbances, etc.

1.2. Objectives

The objective of this work is to develop a methodology able to provide solutions for berthing maneuvers of artificial satellites in space, being the chaser endowed with a robotic manipulator considering the multi-objective optimization to meet, in a balanced way, conflicting objectives.

This work contributes to the development of an innovative methodology based on a common metric for all objectives to deal with multi-objective optimizations. It will be shown how a new method, called *Mutual Metric Method* (MMM), is useful to find balanced solutions for an, until now, unexploited environment under a novel perspective of berthing maneuvers. Its performance may present advantages, from the standpoint of computational implementation, when compared to the *Smallest Loss Criterion* (SLC) (ROCCO, 2002).

Additionally, it will be presented how it was possible to test and validate the developed simulation environment for berthing maneuvers through real-time and hardware-in-the-loop simulations using the European Proximity Operations Simulator (EPOS) with a workaround solution for the absence of a physical berthing manipulator.

2 REVIEW OF LITERATURE

This chapter aims to present works found in the literature within the topic discussed. The works cited here deal essentially with studies with robotic manipulators or applied optimization.

2.1. Robotic manipulators and space robotics

Wang and Chen (1991) developed a new method for computing numerical solutions to the inverse kinematic problem of robotic manipulators based on optimization techniques.

Ellery (2000) developed studies on the dynamics of satellites equipped with robotic manipulators. The work cites several references in which the different formulations of robotic dynamics were tested and establishes a comparative considering the computational efficiency of each one.

The works of Yoshida (2000, 2001, 2003) ratified important concepts tested in the Engineering Test Satellite No. 7 (ETS-VII), including the treatment given to the attitude motion of the base satellite according to the motion of the manipulator arm. The ETS-VII was equipped with a 2-meter-long, 6 degrees of freedom robotic arm, which was used to carry out several experiments.

Fonseca, Arantes Junior and Bainum (2004) presented a mathematical model for a large space structure containing a robotic manipulator that moves throughout the station and is used to perform satellite grabbing operations to be fixed by the astronaut. Control efforts were also evaluated.

In Fonseca and Bainum (2004), the model for a low-orbiting space station was developed to analyze attitude behavior during maneuvers performed with a robotic manipulator. The results showed that for short duration maneuvers the attitude did not change significantly.

Bagchi (2005) used a technique that optimizes the joint angle by means of an inverse kinematics program for an appropriate size of the manipulator to reach a given set of target points within the workspace without singularities. The optimization process is based on maximizing the manipulability. The objective function is subject to link length and joint angle constraints.

Pettersson (2008) studied industrial robot design optimization strategies. Regarding the inherent objectives of robot design, such as performance, cost, and quality, these are treated as compromises in optimization.

Ayten (2012) has developed methods to generate trajectories for point-to-point movement of robotic manipulators under previously defined kinematic and dynamic constraints. In his literature review, he discusses aspects of multi-objective optimization.

The work (PIRES; OLIVEIRA; MACHADO, 2010) approached the trajectory planning problem considering: robots with two and three degrees of freedom, the inclusion of obstacles in the work volume, more than five criteria that are used to qualify the trajectory. Such criteria are used to minimize: the distance of travel of the joints and end effector, the path oscillation, and the energy required by the manipulator to reach a destination point. The perspective of multi-objective optimization revealed a set of solutions belonging to the Pareto frontier. The proposed method aims to assist the decision maker in choosing the best solution since such a method provides a set of non-dominated solutions.

In Fonseca et al. (2015), the problem of a robotic manipulator operating on a non-fixed base was studied considering a scenario in which the base is free to react in response to the manipulator movement and in another scenario where the manipulator moves its links to compensate for reaction forces on the platform maintaining stability.

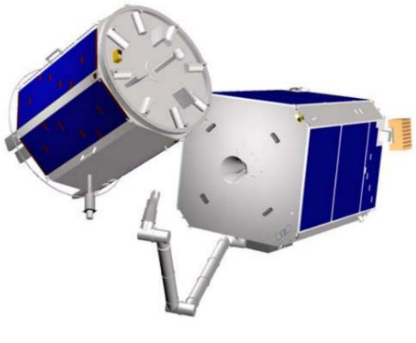
Fonseca et al. (2017) considered a spacecraft and a manipulator to analyze the impact of robotic dynamics on attitude motion and the associated control effort to maintain a stable attitude during operations of the manipulator.

2.2. Simulations and optimization

In Boge et al. (2010), it is stated that satellites in orbit can severely be affected by aging or degradation as well as by consumption of available resources. According to them, these problems can be solved by On-Orbit Servicing (OOS) missions. Additionally, it is said that a critical issue is to ensure a reliable Rendezvous and Docking (RvD) operation performed autonomously in space. Therefore, it must be carefully simulated before the real mission. The European Proximity Operations Simulator (EPOS) is presented for this purpose.

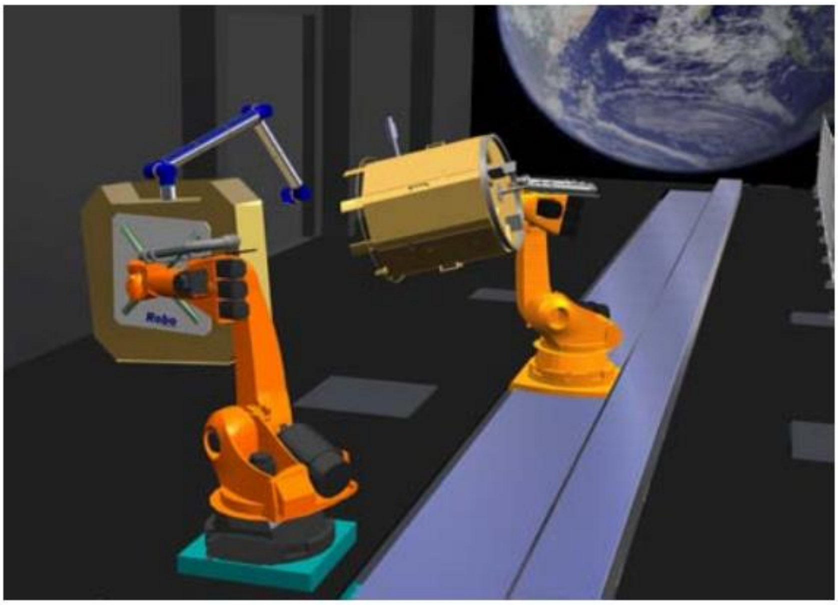
One of the goals of the cited work, was to perform rendezvous, capture and docking scenarios for the German OOS technology demonstration mission (DEOS) pictured in Figure 2.1. A configuration of the EPOS facility for a DEOS RvD simulation with a robotic manipulator arm mounted on one of the mockups is presented in Figure 2.2. It was thought the possibility of tests of hardware-in-the-loop scenarios with sensors measuring relative position and attitude while the onboard computer calculates the necessary thrusters or reaction wheels commands.

Figure 2.1 - Servicer and client satellite of DEOS.



Source: Boge et al. (2010).

Figure 2.2 - EPOS simulation for DEOS.



Source: Boge et al. (2010).

In the literature (SANTOS; STEFFEN; SARAMAGO, 2010) a strategy is proposed to optimize the energy consumption and manipulation of a robot by defining a relationship between the objectives.

Rocco, Souza and Prado (2000, 2001, 2002, 2003, 2005a, 2005b, 2013) studied the problem of maneuvers performed by satellites belonging to a constellation. In order to perform maneuvers while minimizing fuel consumption, maneuvering time and position deviation. Such an approach configures multi-objective optimization with conflicting objectives. Therefore, the *Smallest Loss Criterion* (ROCCO, 2002) was defined and applied to achieve a balanced solution that equally considers all objectives. This method was also used in several other works, such as: (SANTOS et al., 2016; VENDITTI et al., 2010; SANTOS; ROCCO; BOGE, 2015; LAU et al., 2014; AMORIM TERCEIRO, 2013; GRANZIERA JUNIOR., 2015; SANTOS, 2015).

Granziera Junior (2015) formulated attitude control and determination as a multi-objective problem, considering random and non-random errors, as well as acknowledging the existence of conflicts in objectives and the approach by the *Smallest Loss Criterion*.

Santos (2015) explored the problem of spacecraft control using actuators with conflicting characteristics in a scenario of a final rendezvous maneuver. It was proposed a strategy based on multi-objective optimization for operating a group of actuators. The developed software presented effectiveness and robustness, proving to be able to generate reliable outcomes in both non-real-time and real-time simulations at EPOS.

The EPOS facility has been used to test and validate proposed models employing real sensors for guidance, navigation and control in real-time (RT - VxWorks operating system) and it is thoroughly explained in the works (BENNINGHOFF; BOGE; REMS, 2014; BENNINGHOFF; REMS; BOGE, 2014; BENNINGHOFF et al., 2017). Additionally, in (BENNINGHOFF et al., 2018), it is described how simulations have been done using two robotic hardware-in-the-loop test beds, the EPOS, as the robotic rendezvous simulator, and the OOS-Sim, as the robotic test facility for berthing, where a robotic arm is used for capturing the client satellite. It is stated that robot-based facilities, i.e., hardware-in-the-loop simulators, can help to implement active gravity compensation, accommodate complex systems, e.g., a free-flying robot, and they provide unlimited time for simulations.

2.3. Positioning and relevance

As it is possible to realize, the works described in this chapter do not represent essentially a research line in which the present document complements or disproves. For all implications, it is possible to say they are complete in their scope, that is why they were selected as references here, while the present work proposes a comprehensive scenario considering space environment, multi-objective optimization, conflicting objectives, satellite and robotic manipulator dynamics all together in a way it was not addressed before.

From the analysis of the works found in the literature, it is recognizable that the employment of robotic manipulators for space applications is a current research topic with immediate application in OOS missions. Therefore, the present work verifies the possibility of optimization applied to the use of a robotic manipulator in berthing maneuvers.

3 MODELS, METHODS AND TOOLS

3.1. Mathematical models

It was developed a simulator for robotic systems in space. The base satellite movement dynamically changes the distance between end effector and target point due to the coupled manipulator’s moves.

In a berthing maneuver simulation, it is possible to follow the center of mass of the compound (satellite that serves as base plus the robotic-three-jointed manipulator) changing along the time. Then, it is proposed an approach based on the ongoing changes in moments of inertia matrix and its consequences to the servicer satellite dynamics using the iterative Newton-Euler algorithm.

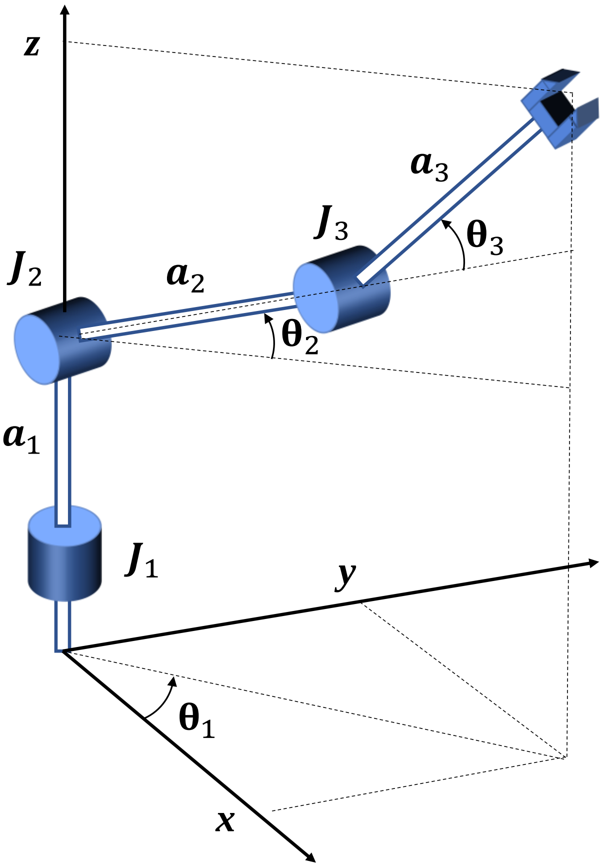
If you move the manipulator arm during the berthing maneuver, the servicer base satellite pose will change since you change the center of mass and the moments of inertia of the coupled system. Such center of mass and moments of inertia modifications were already considered in the developed simulator. Through animations, it is possible to see satellite and robot interfering with each other during a berthing maneuver.

The proposed software was named SAtellite and RObot Simulator, SAROS as an evolution of SAS (*Satellite Attitude Simulator*) developed by Rocco, Costa Filho and Carrara (2011); and Rocco and Costa Filho (2015).

3.1.1. Kinematics

The robotic arm is an anthropomorphic robot, in this case a Twisting - Rotational – Rotational (TRR) robot Figure 3.1, from which we obtained equations to solve the direct kinematics (given the angles of each joint, it is possible to calculate the end effector position) and inverse kinematics (given the desired position to the end effector, it finds the joint angles able to take the robot’s wrist there).

Figure 3.1 - Robotic arm TRR.



Source: Nardin (2015).

Using simple trigonometric relations, it is possible to find Equations 3.1, 3.2 and 3.3 for inverse kinematics and Equations 3.4, 3.5 and 3.6 for direct kinematics, considering each joint respectively from *1* to *3*.

$$\theta_1 = \arctan\frac{y}{x} \tag{3.1}$$

$$\theta_2 = \arctan\left[\frac{(z-a_1)(a_2+a_3\cos\theta_3)-\sqrt{x^2+y^2}\,a_3\sin\theta_3}{\sqrt{x^2+y^2}\,(a_2+a_3\cos\theta_3)+(z-a_1)a_3\sin\theta_3}\right] \tag{3.2}$$

$$\theta_3 = \arccos\left(\frac{x^2+y^2+(z-a_1)^2-a_2^2-a_3^2}{2a_2a_3}\right) \tag{3.3}$$

$$x = [a_2 \cos\theta_2 + a_3 \cos(\theta_2 + \theta_3)] \cos\theta_1 \tag{3.4}$$

$$y = [a_2 \cos\theta_2 + a_3 \cos(\theta_2 + \theta_3)] \sin\theta_1 \tag{3.5}$$

$$z = a_1 + a_2 \sin\theta_2 + a_3 \sin(\theta_2 + \theta_3) \tag{3.6}$$

3.1.2. Moments of inertia

The moment of inertia tensor relative to the reference system A is expressed in Equation 3.7 (CRAIG, 2005).

$${}^{A}\mathbf{I} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \tag{3.7}$$

Elements are given by Equations 3.8, 3.9, 3.10, 3.11, 3.12, and 3.13:

$$I_{xx} = \iiint_V (y^2 + z^2)\rho dv \tag{3.8}$$

$$I_{yy} = \iiint_V (x^2 + z^2)\rho dv \tag{3.9}$$

$$I_{zz} = \iiint_V (x^2 + y^2)\rho dv \tag{3.10}$$

$$I_{xy} = \iiint_V xy\rho dv \tag{3.11}$$

$$I_{xz} = \iiint_V xz\rho dv \tag{3.12}$$

$$I_{yz} = \iiint_V yz\rho dv \tag{3.13}$$

Being ρ the homogenous volumetric mass density. Sometimes, it is necessary to calculate the moment of inertia tensor with relation to the center of mass, which is given by Equation 3.14.

$$\mathbf{P}_c = [x_c, y_c, z_c]^T \tag{3.14}$$

Given that C is the body center of mass system and A a non-particular system, the Steiner theorem application brings the Equation 3.15 (CRAIG, 2005):

$$^{A}\mathbf{I} = {}^{C}\mathbf{I} + m[\mathbf{P}_c{}^T \mathbf{P}_c \mathbf{I}_3 - \mathbf{P}_c \mathbf{P}_c{}^T] \tag{3.15}$$

Where $\boldsymbol{I}_3$ is the *3* by *3* identity matrix.

The moment of inertia tensor, Equation 3.16, relative to the center of mass of a parallelepiped will be useful because that is the standard links' shape (CRAIG, 2005).

$$^{C}\mathbf{I} = \begin{bmatrix} \frac{m}{12}(h^2 + l^2) & 0 & 0 \\ 0 & \frac{m}{12}(w^2 + h^2) & 0 \\ 0 & 0 & \frac{m}{12}(l^2 + w^2) \end{bmatrix} \tag{3.16}$$

Where m*,* l*,* w and h are body's mass, length, width and height, respectively.

3.1.3. Newton-Euler algorithm

If a rigid body has some acceleration in its center of mass it must have been caused by a force with respect to a system i, Equation 3.17. If this body is in rotation, with angular velocity and acceleration, then there exists a torque which acts on the body in order to cause a movement, Equation 3.18.

$$\vec{F}_i = m\dot{\vec{v}}_{C_i} \tag{3.17}$$

$$\vec{N}_i = {}^{C_i}\mathbf{I}\dot{\vec{\omega}}_i + \vec{\omega}_i \times {}^{C_i}\mathbf{I}\vec{\omega}_i \tag{3.18}$$

Where C has its origin on the link center of mass and the same orientation of the reference system i. We use the iterative Newton-Euler to calculate torques along the manipulator movement. The algorithm works in two stages: Outward, velocities and accelerations propagation, forces and torques calculation on each link from the first system to the last; Inward, forces and torques executed by each joint, from end effector to the manipulator’s base (CRAIG, 2005; ELLERY, 2000). Equations 3.19, 3.20, 3.21, 3.22, 3.23, and 3.24 for Outward:

$${}^{i+1}\vec{\omega}_{i+1} = {}^{i+1}_{i}\mathbf{R}\,{}^{i}\vec{\omega}_{i} + \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1} \tag{3.19}$$

$${}^{i+1}\dot{\vec{\omega}}_{i+1} = {}^{i+1}_{i}\mathbf{R}\,{}^{i}\dot{\vec{\omega}}_{i} + {}^{i+1}_{i}\mathbf{R}\,{}^{i}\vec{\omega}_{i} \times \dot{\theta}_{i+1}\,{}^{i+1}\hat{Z}_{i+1} + \ddot{\theta}_{i}\,{}^{i+1}\hat{Z}_{i+1} \tag{3.20}$$

$${}^{i+1}\dot{\vec{v}}_{i+1} = {}^{i+1}_{i}\mathbf{R}({}^{i}\dot{\vec{\omega}}_{i} \times {}^{i}\vec{P}_{i+1} + {}^{i}\vec{\omega}_{i} \times ({}^{i}\vec{\omega}_{i} \times {}^{i}\vec{P}_{i+1}) + {}^{i}\dot{\vec{v}}_{i}) \tag{3.21}$$

$${}^{i+1}\dot{\vec{v}}_{C_{i+1}} = {}^{i+1}\dot{\vec{\omega}}_{i+1} \times {}^{i+1}\vec{P}_{C_{i+1}} + {}^{i+1}\vec{\omega}_{i+1} \times ({}^{i+1}\vec{\omega}_{i+1} \times {}^{i+1}\vec{P}_{C_{i+1}}) + {}^{i+1}\dot{\vec{v}}_{i+1} \tag{3.22}$$

$${}^{i+1}\vec{F}_{i+1} = m_{i+1}\,{}^{i+1}\dot{\vec{v}}_{C_{i+1}} \tag{3.23}$$

$${}^{i+1}\vec{N}_{i+1} = {}^{C_{i+1}}I_{i+1}\,{}^{i+1}\dot{\vec{\omega}}_{i+1} + {}^{i+1}\vec{\omega}_{i+1} \times {}^{C_{i+1}}I_{i+1}\,{}^{i+1}\vec{\omega}_{i+1} \tag{3.24}$$

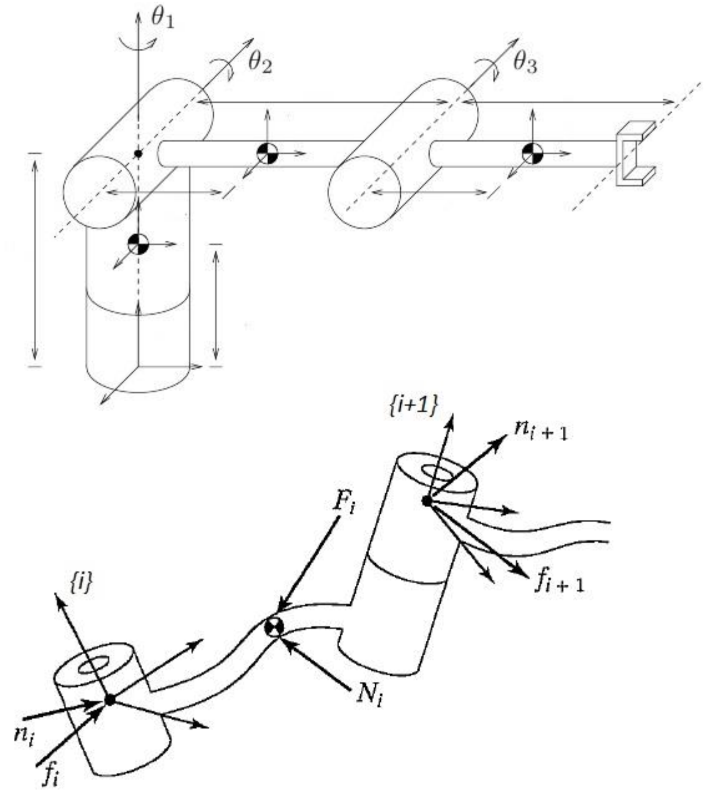
Equations 3.25 and 3.26 for Inward:

$${}^{i}\vec{f}_{i} = {}^{i}_{i+1}\mathbf{R}\,{}^{i+1}\vec{f}_{i+1} + {}^{i}\vec{F}_{i} \tag{3.25}$$

$${}^{i}\vec{n}_{i} = {}^{i}\vec{N}_{i} + {}^{i}_{i+1}\mathbf{R}\,{}^{i+1}\vec{n}_{i+1} + {}^{i}\vec{P}_{C_{i}} \times {}^{i}\vec{F}_{i} + {}^{i}\vec{P}_{i+1} \times {}^{i}_{i+1}\mathbf{R}\,{}^{i+1}\vec{f}_{i+1} \tag{3.26}$$

The Newton-Euler formulation presents advantages when compared to similar others, being, for example, computationally more efficient. Ellery (2000) dedicated special attention to works that evaluated the Newton-Euler formulation superiority against others, such as Lagrangian. Figure 3.2 shows reference systems fixed to each joint center of mass (above) and the reference systems fixed to two successive joints (below).

Figure 3.2 - References on the robot (above) and on two joints (below).



Source: Adapted from Craig (2005).

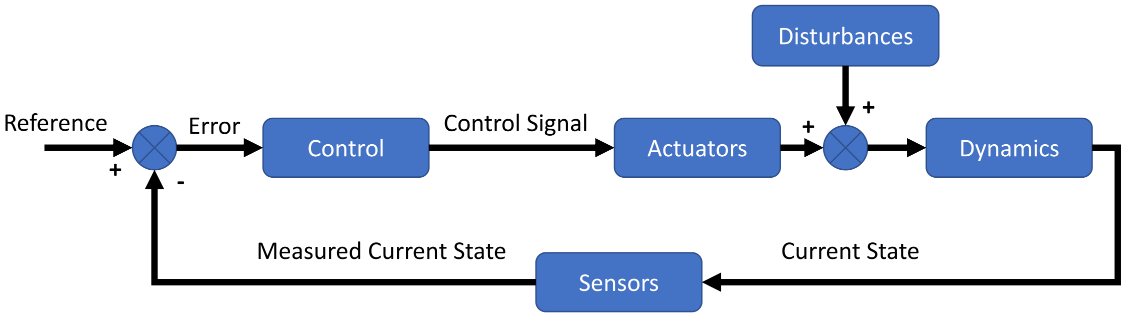
3.1.4. Satellite dynamics

The satellite control system has a PID controller (Proportional - Integral - Derivative) with its respective gains K_P, K_I, K_D, which follows the control law defined by Equation 3.27, where $er(t)$ is the error signal. These gains were used for controlling satellite orientation in roll, pitch and yaw. It is not aim, however, of this work to discuss the requirements and the control system design devised for this simulator, whereas Nardin (2015) addressed the control features in detail.

The control system configuration is shown in Figure 3.3, while Figure 3.4 presents the ensemble formed by cubic shape base satellite plus robotic manipulator. Table 3.1 provides the physical characteristics of each manipulator’s link and base satellite for the example treated in this document and it does not represent a constraint to other manipulators.

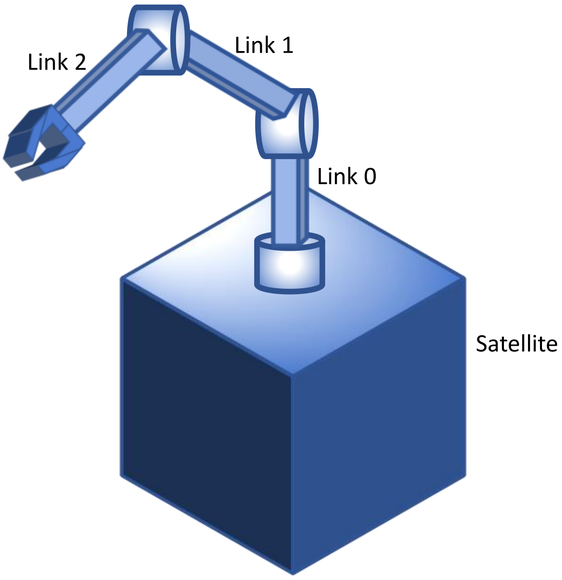
$$c(t) = K_P er(t) + K_I \int er(t)dt + K_D e\dot{r}(t) \tag{3.27}$$

Figure 3.3 - Control configuration.



Source: Nardin (2015).

Figure 3.4 - Ensemble arm and satellite (out of scale).



Source: Nardin (2015).

Table 3.1 - Physical characteristics.

	Length(m)	Width(m)	Height(m)	Mass (kg)
Link 0	0.5	0.1	0.1	40
Link 1	1	0.1	0.1	20
Link 2	1	0.1	0.1	20
Satellite	2	2	2	500

Source: Corrected from Nardin (2015).

Using direct cosine matrix, we can obtain any final satellite attitude. For example, rotations around axis X, Y, Z (or *1-2-3* in this order) produce a matrix as seen in Equations 3.28 and 3.29 (HUGHES, 2004).

$$\mathbf{C}_{ba} = \mathbf{C}_3(\theta_3)\mathbf{C}_2(\theta_2)\mathbf{C}_1(\theta_1) \tag{3.28}$$

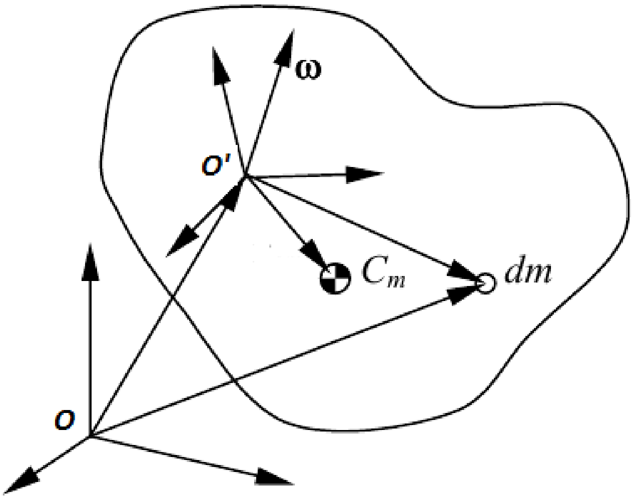
$$\mathbf{C}_{ba} = \begin{bmatrix} c_3c_2 & s_3c_1 + c_3s_2s_1 & s_3s_1 - c_3s_2c_1 \\ -s_3c_2 & c_3c_1 - s_3s_2s_1 & c_3s_1 + s_3s_2c_1 \\ s_2 & -c_2s_1 & c_2c_1 \end{bmatrix} \tag{3.29}$$

Where c represents cosine, s represents sine and indexes *1, 2, 3* represent angles $\theta_1, \theta_2, \theta_3$ respectively. Equation 3.30 is an attitude matrix considering infinitesimal angular displacements (HUGHES, 2004).

$$\mathbf{C}_{ba} \cong \begin{bmatrix} 1 & \theta_3 & -\theta_2 \\ -\theta_3 & 1 & \theta_1 \\ \theta_2 & -\theta_1 & 1 \end{bmatrix} = \mathbf{I}_3 - \boldsymbol{\theta}^{\times} \tag{3.30}$$

A rigid body spins in the system O', with angular velocity with relation to the inertial system O, as shown in Figure 3.5. Being dm an infinitesimal mass element.

Figure 3.5 - Rigid body movement.



Source: Adapted from Hughes (2004).

If the origin of the system O' fixed to the body coincides with the center of mass, the dynamics equations of the satellite are deducted as follows. Equations 3.31, 3.32 , 3.33, and 3.34 use the concept of vectrices from Hughes (2004).

$$\mathbf{p} = m\mathbf{v}_c \tag{3.31}$$

$$\mathbf{h}_c = \mathbf{I}\boldsymbol{\omega} \tag{3.32}$$

$$\dot{\mathbf{p}}^b = \mathbf{f} - \boldsymbol{\omega}^{\times}\mathbf{p} \tag{3.33}$$

$$\dot{\mathbf{h}}_c^b = \mathbf{g}_c - \boldsymbol{\omega}^{\times}\mathbf{h}_c \tag{3.34}$$

Where $\boldsymbol{p}$ is linear momentum, $\boldsymbol{h_c}$ is the angular momentum with relation to the center of mass, $\boldsymbol{v_c}$ is the center of mass linear velocity and $\boldsymbol{\omega}$ is the angular velocity.

We define, $\boldsymbol{f}$ as the resulting of external forces and $\boldsymbol{g_c}$ as the resulting of external torques, both applied to the rigid body on its center of mass. Considering Equations 3.35 and 3.36, we find Equations 3.37 and 3.38, again using the vectrices (HUGHES, 2004).

$$\dot{\mathbf{p}}^b = m\dot{\mathbf{v}}_c^b \tag{3.35}$$

$$\dot{\mathbf{h}}_c^b = \mathbf{I}\dot{\boldsymbol{\omega}}^b \tag{3.36}$$

$$\dot{\mathbf{v}}_c^b = m^{-1}\mathbf{f} - \boldsymbol{\omega}^{\times}\mathbf{v}_c \tag{3.37}$$

$$\dot{\boldsymbol{\omega}}^b = \mathbf{I}^{-1}(\mathbf{g}_c - \boldsymbol{\omega}^{\times}\mathbf{I}\boldsymbol{\omega}) \tag{3.38}$$

Figure 3.6 shows vectors and reference systems used in order to obtain the target vector for the manipulator’s tip. In a real mission, the robot’s wrist could be equipped with a claw for example. There are several tools which could be used as end effector, depending on which kind of task should be performed. Being T

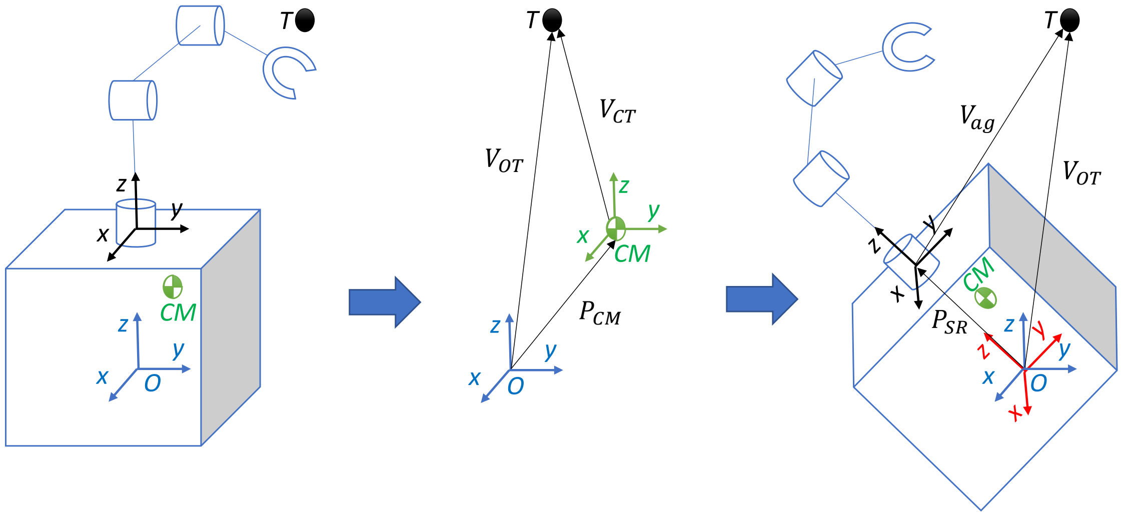
the representation of a target position in space, we obtain Equations 3.39, 3.40, and 3.41.

$$\vec{V}_{OT} = \vec{V}_{CT} + \vec{P}_{CM} \tag{3.39}$$

$$\mathbf{C}_{ca} = \prod_{i=1}^{n} \mathbf{C}_{ba_i} = \mathbf{C}_{ba_n} \mathbf{C}_{ba_{n-1}} ... \mathbf{C}_{ba_2} \mathbf{C}_{ba_1} \tag{3.40}$$

$$\vec{V}_{ag} = \mathbf{C}_{ca} \vec{V}_{OT} - \vec{P}_{SR} \tag{3.41}$$

Figure 3.6 - Attainment of the target vector.



Source: Nardin (2015).

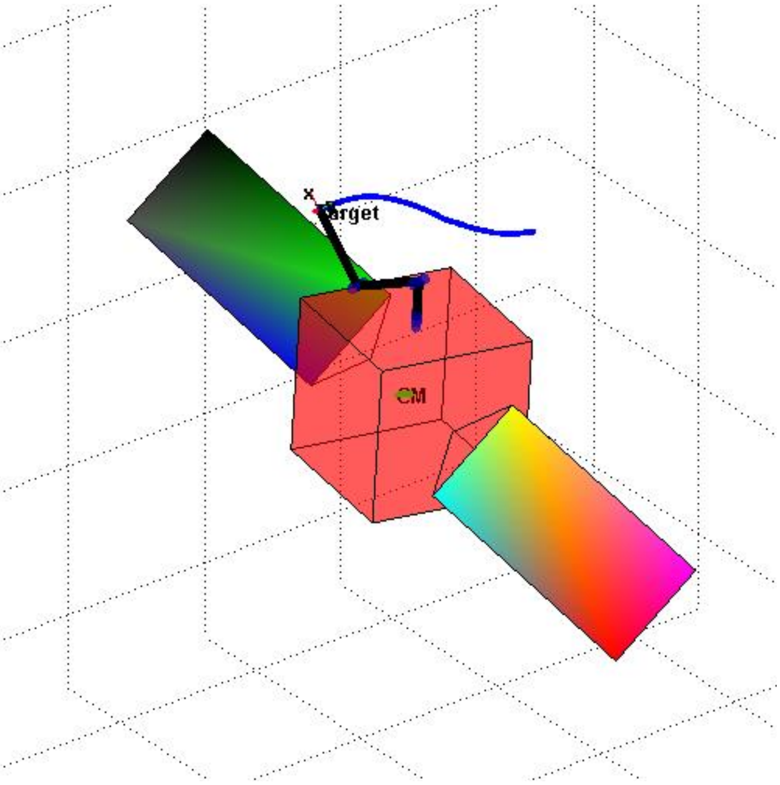
3.2. Simulation tool

A unique characteristic of robot-like satellites is their motion dynamics. Due to the action-to-reaction principle or the momentum conservation, the base satellite moves according to the motion of the manipulator (YOSHIDA, 2000). It is also true that the movement of the robot base, due to satellite repositioning through its actuators, alters dynamically the distance to the target point.

It is investigated the modeling of a robotic system in space. The robotic arm consists of a revolute manipulator with three rotating joints and three degrees of freedom in a Twisting-Rotational-Rotational (TRR) configuration that allows diverse applicability and notable usefulness in the accomplishment of On-Orbit Servicing (OOS).

The proposed simulator, SAROS, generates useful animations for inspecting the maneuver and, occasionally, for confirming that the manipulator has achieved a specified target point. It was used the *Robotics Toolbox for MATLAB* with modifications (CORKE, 2017). Figure 3.7 illustrates a frame of the chaser satellite operating a maneuver. In this work, the solar panels do not have their influence calculated in the dynamics, they are used merely for visualization.

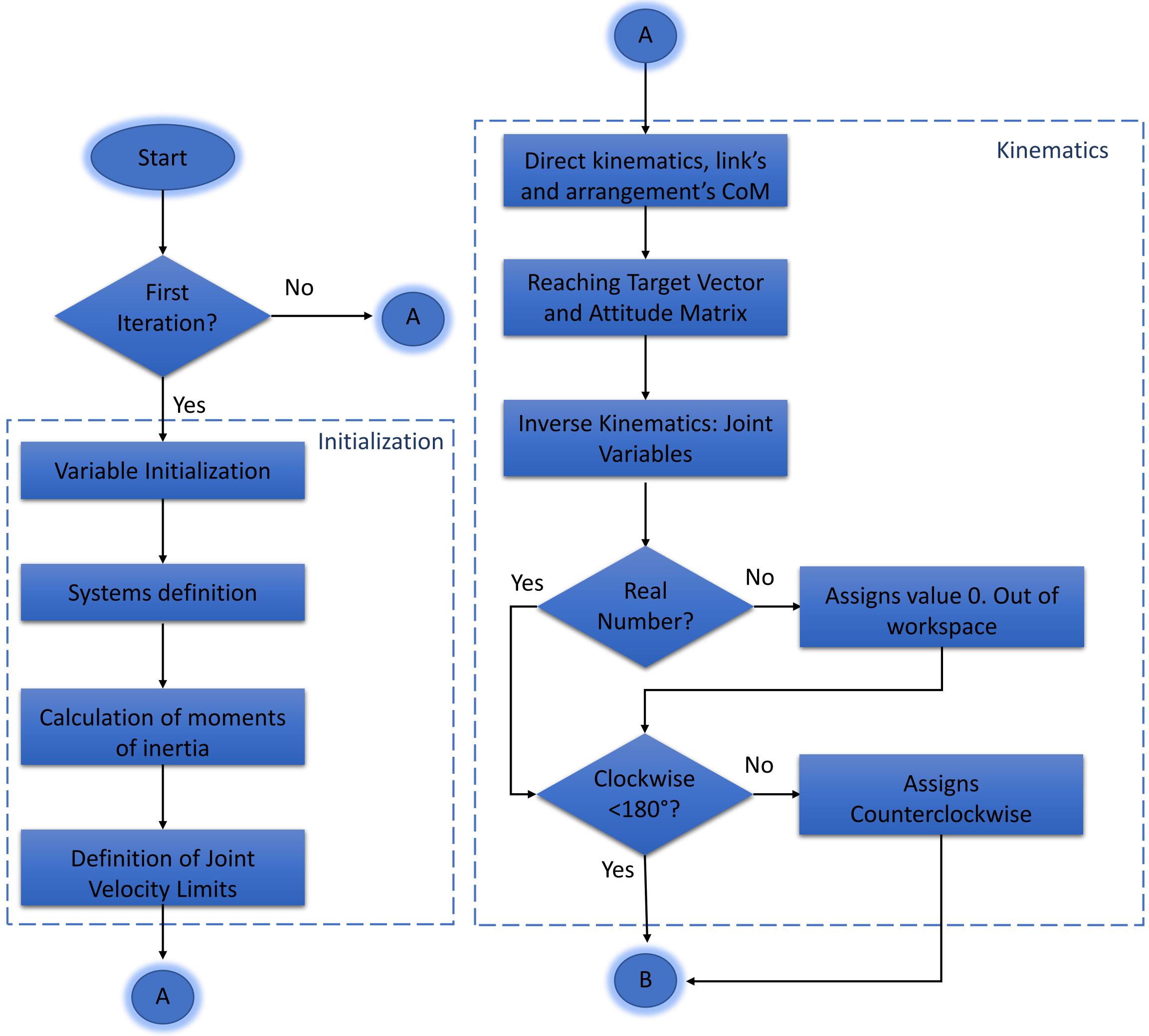
Figure 3.7 - Robotic Manipulator and Satellite.



Source: Nardin (2019).

The simplified flowchart of the software has been divided into two parts and is presented in Figure 3.8 and Figure 3.9. Given the complexity of the program, such a simplification was necessary, but without prejudice to the understanding of the operation in general.

Figure 3.8 - Simplified flowchart (part 1).

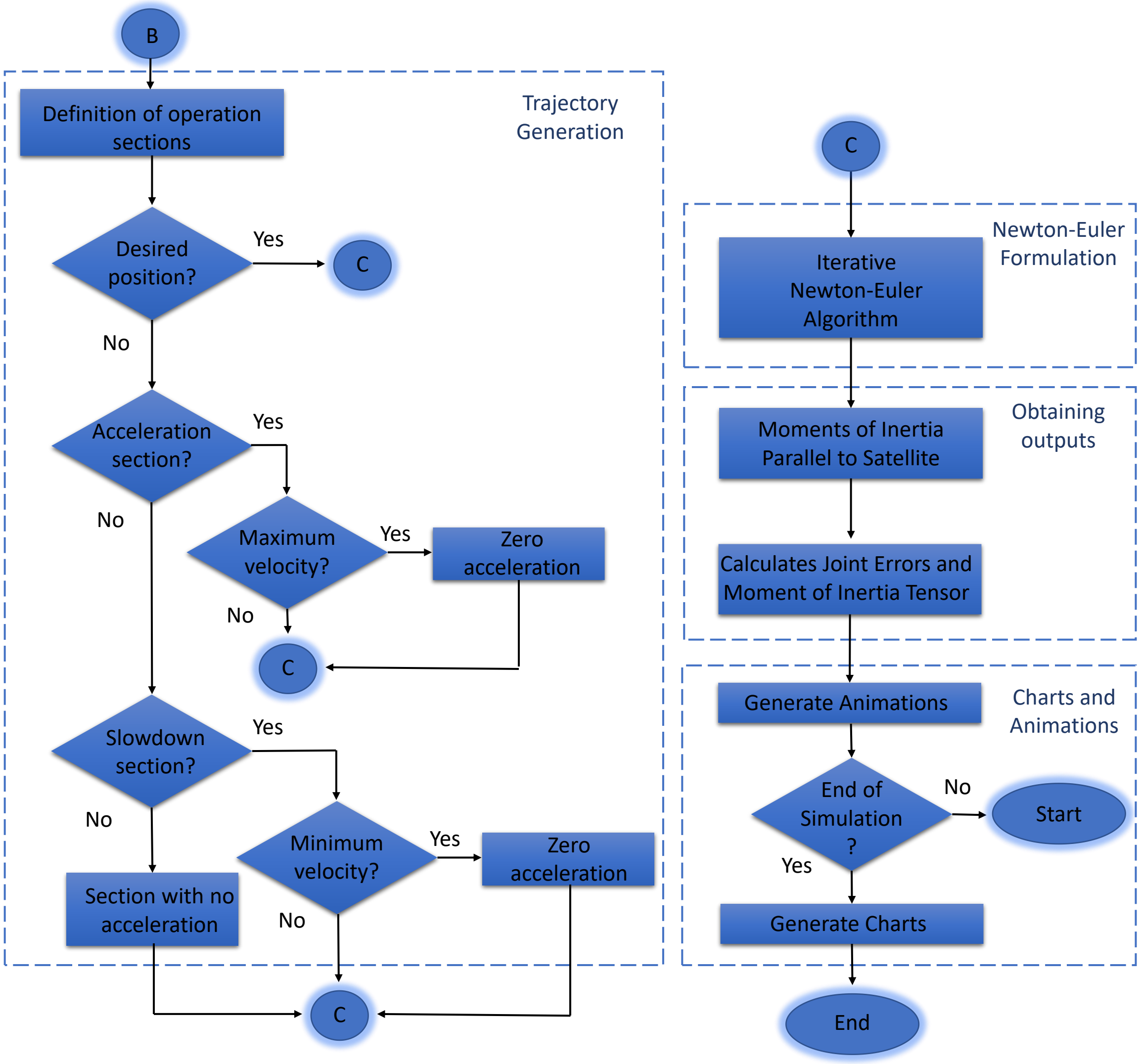


Start
First Iteration?
No
A
Yes
Initialization
Variable Initialization
Systems definition
Calculation of moments of inertia
Definition of Joint Velocity Limits
A
A
Kinematics
Direct kinematics, link's and arrangement's CoM
Reaching Target Vector and Attitude Matrix
Inverse Kinematics: Joint Variables
Yes
Real Number?
No
Assigns value 0. Out of workspace
Clockwise <180°?
No
Assigns Counterclockwise
Yes
B

Source: Adapted from Nardin (2015).

The software performs initialization of variables, kinematic calculations and path generation by defining the sections to which each acceleration is associated. The robotic manipulator dynamics is calculated using the Newton-Euler formulation. The outputs are obtained, and finally the graphics and animations are generated. A more detailed explanation of this software can be found in Nardin (2015).

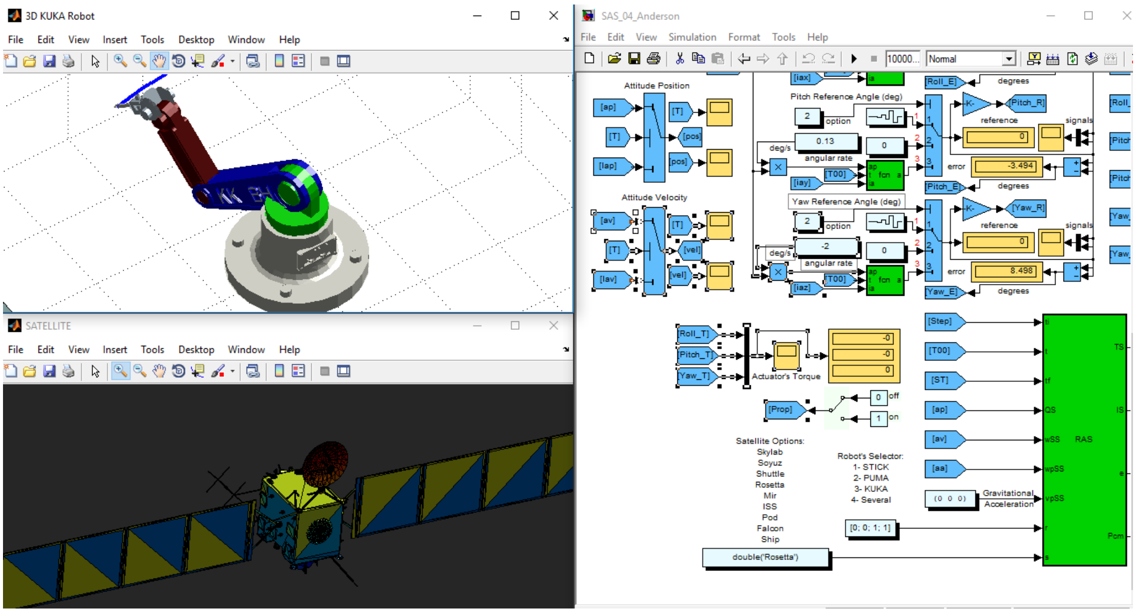
Figure 3.9 - Simplified flowchart (part 2).



Source: Adapted from Nardin (2015).

Figure 3.10 presents a frame of the animated simulation performed by a computer environment simulator designed by Nardin (2015) with its enhancements, which consists of the integration of the SAS (Satellite Attitude Simulator) and the RAS (Robot Attitude Simulator) subsystem. The RAS model calculates the dynamics of the robotic manipulator and provides for the SAS the disturbing torques applied to the base satellite due to the movements of the robotic arm while the SAS provides for RAS the movements stemming from the base satellite dynamics.

Figure 3.10 - Simulator fulfills a berthing.



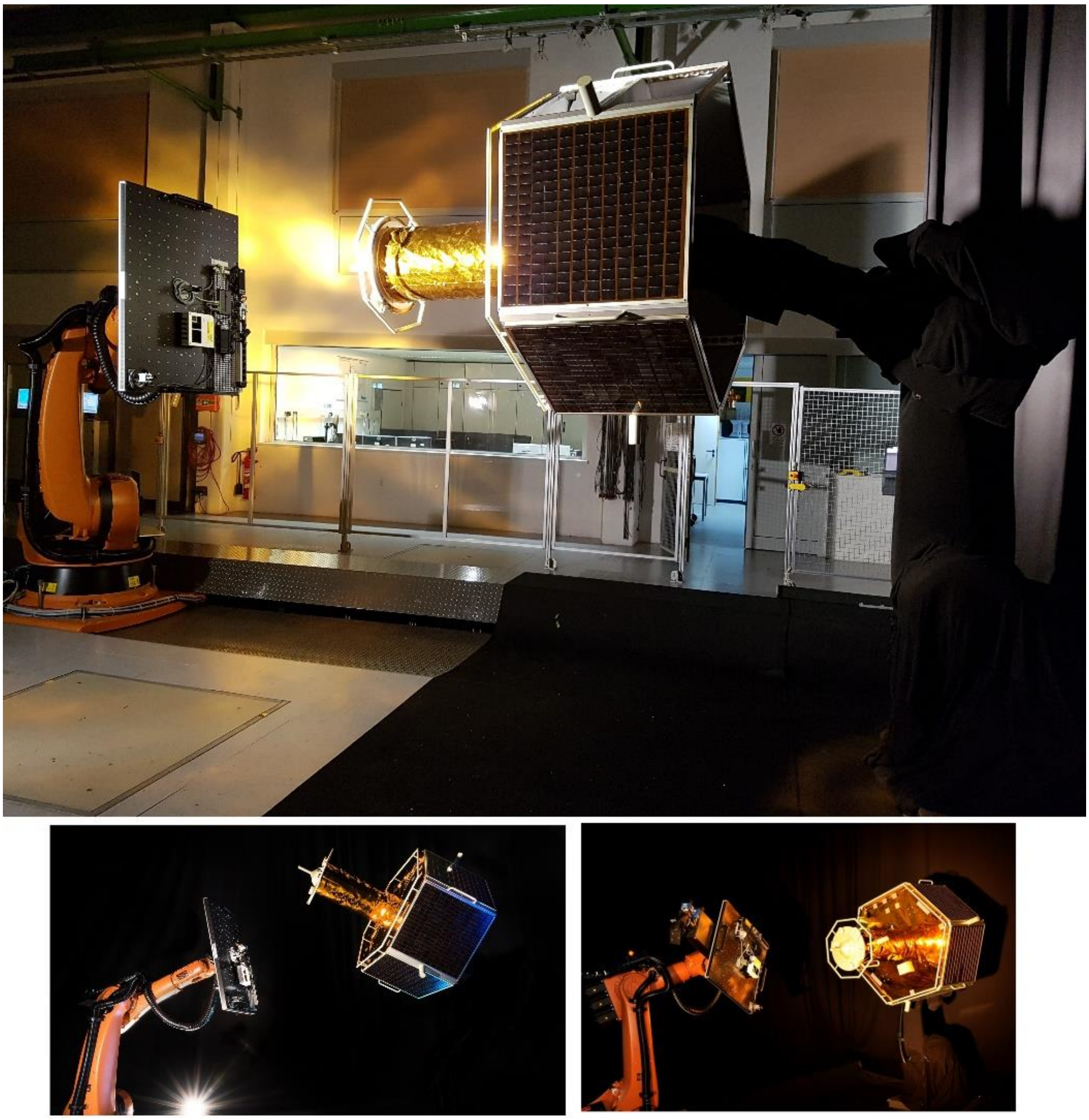
Source: Nardin (2019).

3.3. Hardware-in-the-loop at EPOS

Hardware-in-the-loop simulation is a kind of real-time simulation. It can determine if the physical models are valid and shows how the control system responds to the applied stimuli in real-time. Its basic concept sums up to include real hardware in the simulation loop. HIL simulations have been used for performing Model-Based Design (MBD) since it offers practicality in inaccessible environments like deep space.

The European Proximity Operations Simulator, EPOS, uses two industrial robots to execute the rotational and translational movements of two docking satellites, so-called chaser and target. It is also useful for carrying out testing and verification capabilities of On-Orbit servicing (OOS) missions. The EPOS robots are separated from *0* to *25* meters, they are utilized on realistic simulations of rendezvous and docking process (see Figure 3.11).

Figure 3.11 - EPOS Robots.

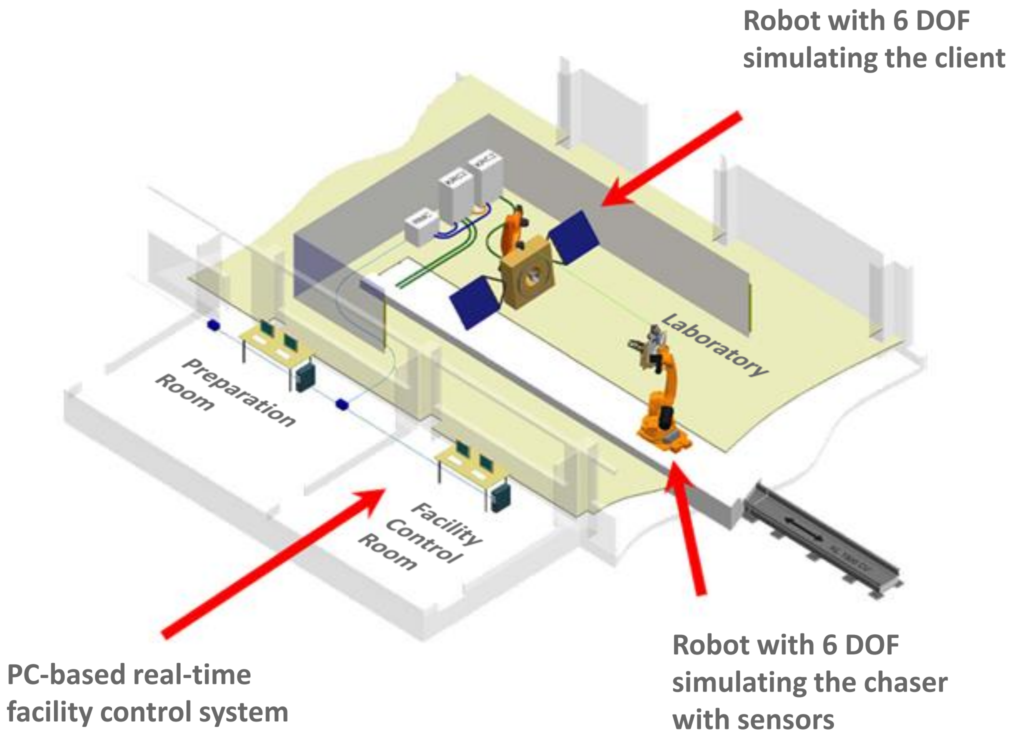


Source: Courtesy of DLR.

In general, experiments of integration, testing and verification of rendezvous maneuvers are managed by the German Aerospace Center using the EPOS hardware-in-the-loop simulations. This facility has been described as an important tool and its potential has been exploited through many test campaigns, including end-to-end simulation environment for rendezvous (BENNINGHOFF et al., 2018).

Figure 3.12 presents the facility’s layout. There is a rail system assembled on the floor to move one of the robots. The so-called Robot 1, a KUKA KR 100 HA, is assembled on a linear rail. The Robot 1 is responsible for *6* degrees of freedom of one of the satellites (chaser), while Robot 2, a KUKA KR 240-2, represents the other satellite (client) with its own *6* degrees of freedom. There is also a control and monitoring system that can manage the whole simulation. More detailed technical data will be presented ahead.

Figure 3.12 - EPOS system components.



Robot with 6 DOF simulating the client

PC-based real-time facility control system

Robot with 6 DOF simulating the chaser with sensors

Source: Boge et al. (2009).

This simulation facility has been used to demonstrate and validate outcomes obtained by computer-based simulations, such as those that will be described here. The EPOS experiments reported in this document contributed to the validation of models previously developed.

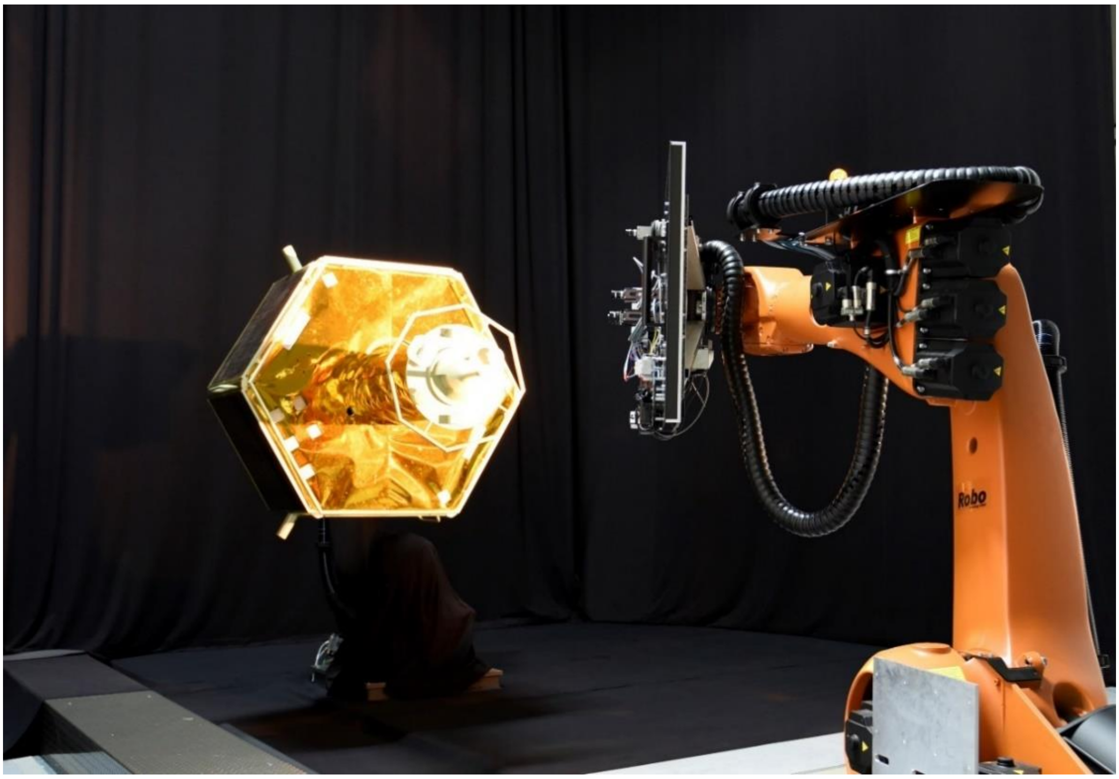
3.3.1. EPOS technical features

The European Proximity Operations Simulator (EPOS) provides numerical results for simulations of satellite dynamics on ground. Each robot can carry different satellite mockups, sensor breadboards, rockets upper stage mockups, etc.

It is also possible to simulate larger distance maneuvers using scaled models. Position and attitude controls must be provided to the facility with a command rate of *250* Hz.

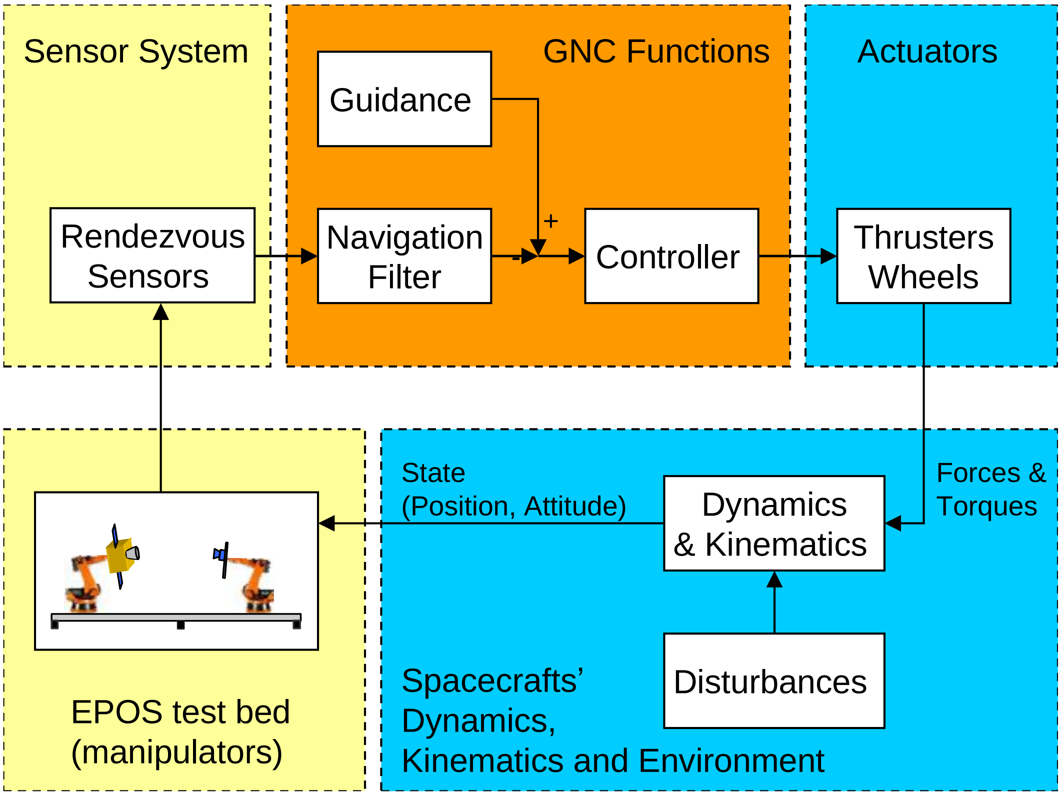
A typical setup of the EPOS robotic test bed is shown by Figure 3.13, whereas Figure 3.14 presents an example of control loop for HIL simulations including hardware elements (yellow area), on-board software (orange area), and numerical simulator (blue area). The technical data of the rail and the robots is presented in Table 3.2 and Table 3.3, respectively.

Figure 3.13 - EPOS robotic test bed.



Source: Courtesy of DLR.

Figure 3.14 - Control loop for HIL simulation.



Source: Benninghoff; Boge; Rems (2014).

Table 3.2 - Technical data of the rail.

Parameter	KUKA KL 1500 (rail)
Type	Linear axis with rack-and-pinion drive
Payload mass maximum	3000 kg
Rail mass (without robot)	ca. 13500 kg
Repeatability	±0.02 mm (ISO 9283)

Source: Benninghoff et al. (2017).

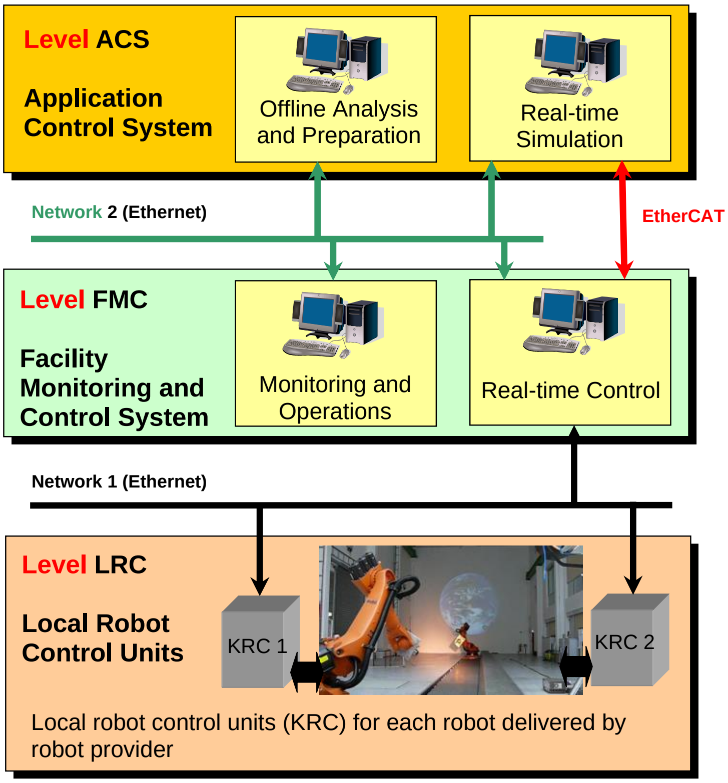
Table 3.3 - Technical data of the robots.

Parameter	KUKA KR 100 HA (robot 1)	KUKA KR 240-2 (robot 2)
Type	6-axis articulated robot	6-axis articulated robot
Payload mass maximum	100 kg	240 kg
Robot mass (without control part)	1200 kg	1267 kg
Reach (approx.)	2600 mm	2700 mm
Repeatability	±0.12 mm (ISO 9283)	±0.12 mm (ISO 9283)
Mounting	Rail	Floor

Source: Benninghoff et al. (2017).

The facility has a computer-based control and monitoring system, Figure 3.15. Each computer fulfills a specific task. The first level is the Application Control System (ACS). The second level corresponds to the Facility Monitoring Control System (FMC). The third is the Local Robot Control Unit (LRC).

Figure 3.15 - EPOS control system.


Level ACS
Application Control System
Offline Analysis and Preparation
Real-time Simulation
Network 2 (Ethernet)
EtherCAT
Level FMC
Facility Monitoring and Control System
Monitoring and Operations
Real-time Control
Network 1 (Ethernet)
Level LRC
Local Robot Control Units
KRC 1
KRC 2
Local robot control units (KRC) for each robot delivered by robot provider

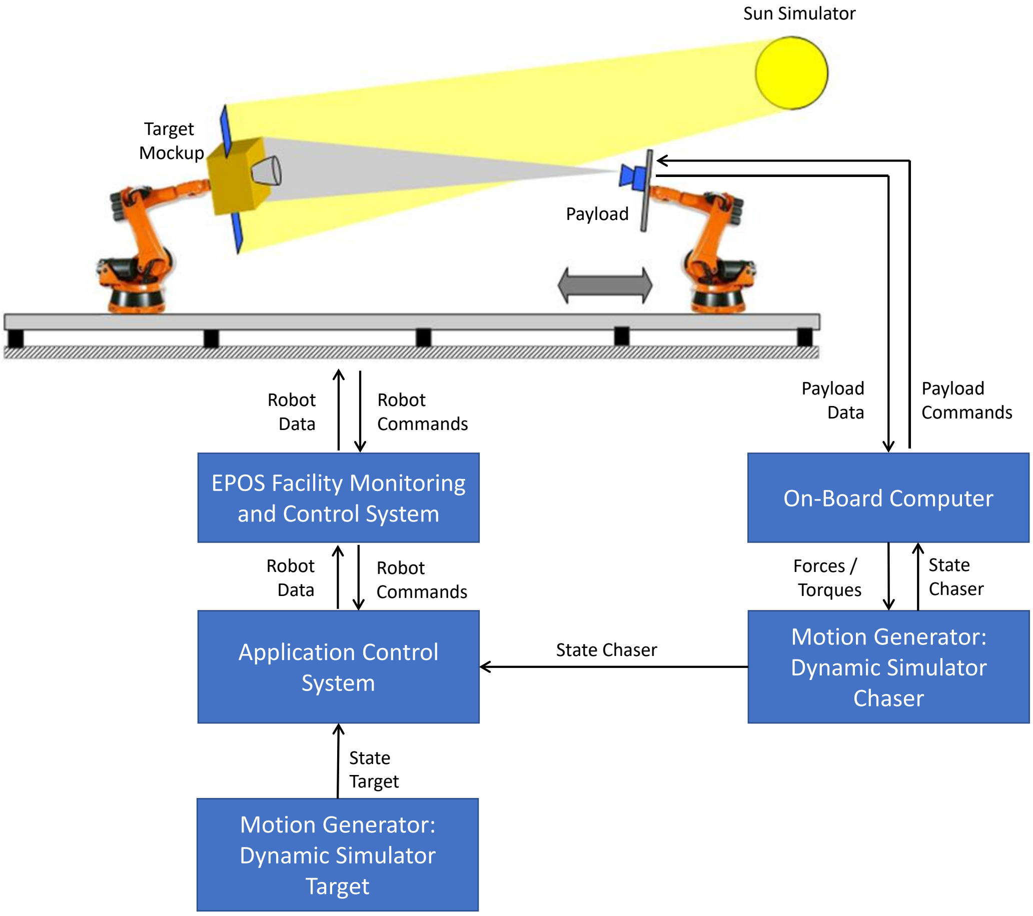
Source: Benninghoff et al. (2017).

The FMC and ACS systems consist of two computers each, a real-time computer with VxWorks as operating system running the real-time executable and a non-real-time Windows computer for Man Machine Interface (MMI). The LRC consists of two KUKA Robot Controllers (KRC) connected to the robots.

On the ACS-MMI computer, it is developed the application using MATLAB/Simulink, while the ACS-RT is configured as the external target. The MATLAB Real-Time Workshop (RTW) is used to generate the executable. A possible model is a numerical satellite simulator (BENNINGHOFF et al., 2017).

Figure 3.16 illustrates a closed-loop simulation of a rendezvous. The system is based on measures of relative positioning between satellite models, and it can control the complete maneuver simulation.

Figure 3.16 - EPOS configuration in a closed loop.

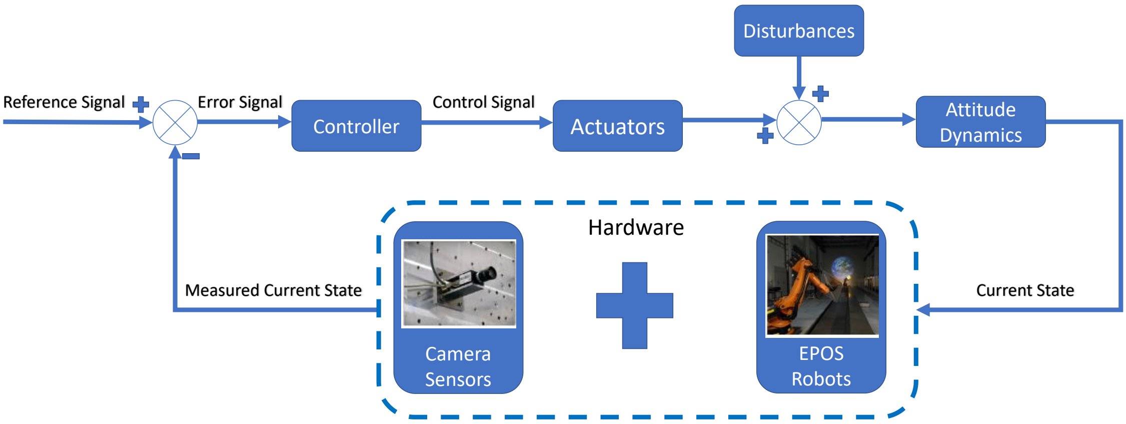


Source: Boge; Benninghoff; Tzschichholz (2011).

3.3.2. EPOS simulations

The SAROS project was presented to the On-Orbit Servicing and Autonomy group of the Space Flight Technology department at DLR, it was proposed a few modifications to be fulfilled in the software in order to improve chances of having it working correctly with the EPOS hardware. Indeed, those and many other modifications were done aiming to improve performance (simulation rate). Figure 3.17 shows the control system configuration.

Figure 3.17 - Control System Configuration.



Source: Adapted from Nardin (2019).

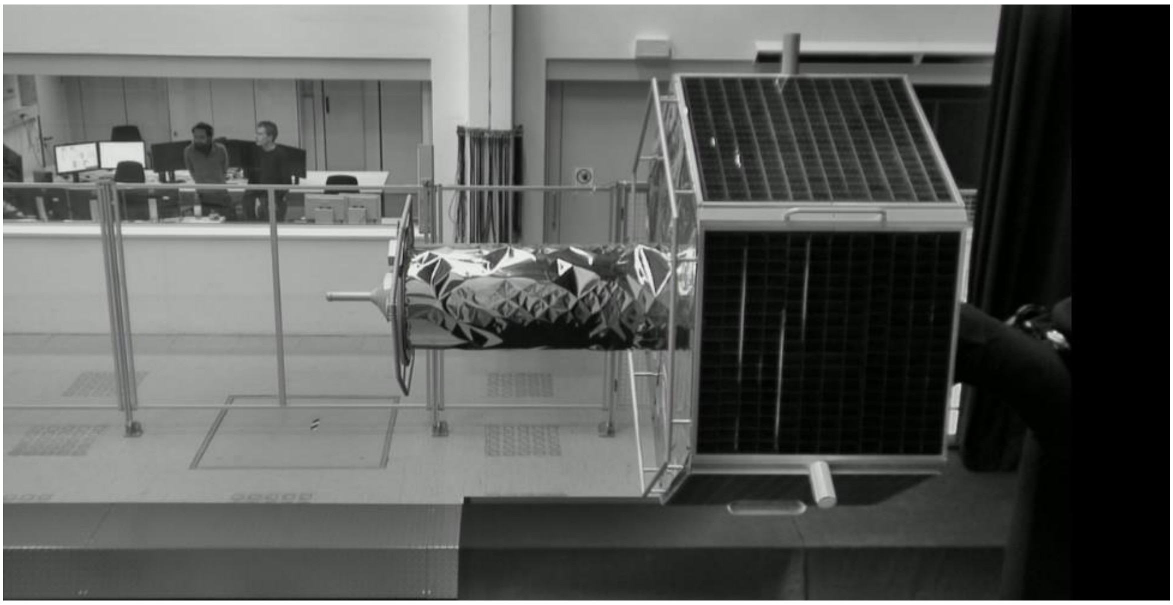
To be able to test the developed SAROS algorithms in the EPOS facility, it was composed a test routine. This plan of actions, when executed in the correct order, could improve results and at the same time avoid mistakes or unsafe actions.

The mentioned test plan was improved after some attempts, its final version was called “Test Plan version 2.0”. Each test was thought to be useful in some way to prove a simulator capability. After each test, a document called “Test Protocol” had to be properly filled in with a report format, examples are presented in (NARDIN, 2019).

Every software modification was properly documented using a tool for code sources management and software version control (*Git*), which has the intents: speed development, data integrity and support for distributed, non-linear workflows. The complete history of modifications fulfilled can be seen in (NARDIN, 2019).

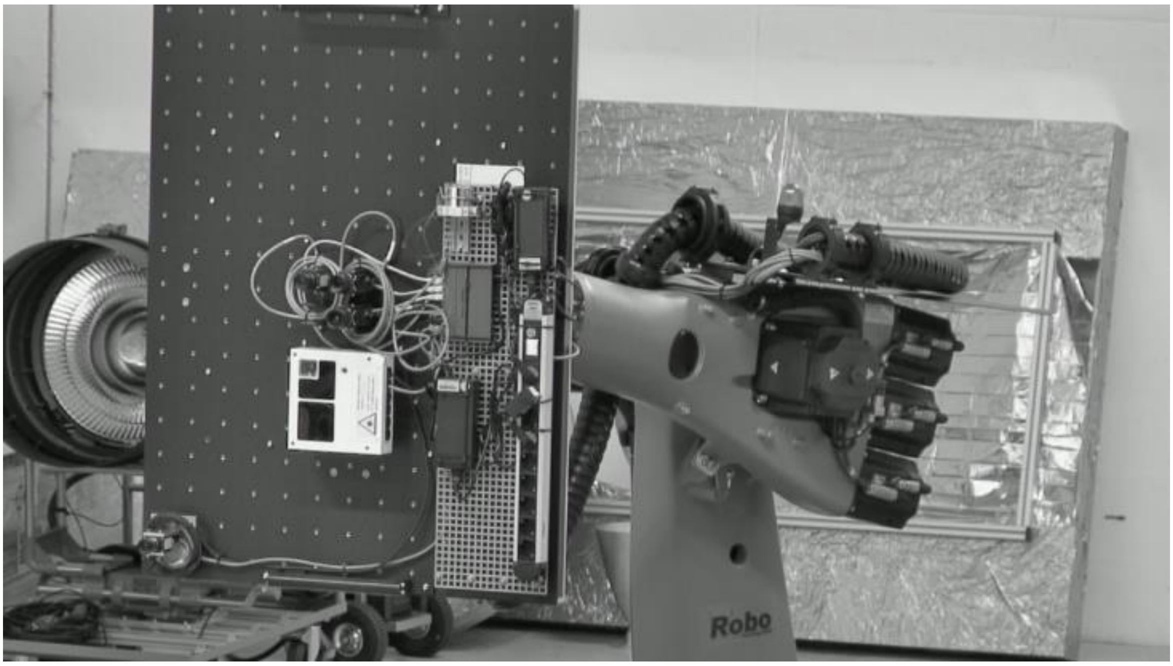
All robots’ movements in EPOS facility during simulations have been recorded by security cameras, Figure 3.18 shows a frame from the recorded video of the robot which performs the target satellite’s movements, while Figure 3.19 shows the robot responsible for chaser satellite’s movements.

Figure 3.18 - Target Satellite and its robot.



Source: Nardin (2019).

Figure 3.19 - Chaser Satellite and its robot.



Source: Nardin (2019).

Once berthing maneuvers are expected to be performed when target and chaser satellites are close enough, considering the manipulator workspace, the orbits of both satellites are synchronized, and the simulations described here are only dedicated to the attitude motion.

The general objective of the EPOS simulations is to increase the level of complexity of the experiments which the software simulator can operate with. The routines that will be described were planned as a genuine effort of, as it is required by the scientific method, continuous enhancement in rigor and sophistication of tests. We are interested in verifying whether the software simulator will be able to solve dynamic problems and, under different circumstances, perform the desired berthing maneuver.

Most of the results presented in this section were previously published in a technical report by Nardin (2019), some of them with corrections. The parameters of the simulations for the chaser satellite are defined according to the Table 3.4, where the initial angular position is defined using a Direct Cosine Matrix (DCM). The simulations were defined to gradually test the implemented system, increasing the complexity with each simulation.

Firstly, the procedure defined in the test plan was done trying to maintain the initial orientation with satellite's and virtual manipulator's control systems turned off in order to demonstrate the possibility of connection between the developed system and the devices of EPOS facility.

We examine the collected data to verify the success of each test, mainly, looking at results generated on graphs. Figure 3.20, 3.21 and 3.22 demonstrate how the chaser satellite orientation was maintained steady along all simulation, as wanted for this simulation. Despite there are no torques or forces in this simulation, it was important to ensure the software would not generate and propagate unexpected and dangerous numerical errors to the hardware.

At this point, it was desirable that satellite and virtual manipulator control systems were remained turned off. This demonstrates reliability when the software runs, without crashes on hardware. We can verify this results in Figures 3.23 and 3.24. These results represent a good general operation at this moment.

Table 3.4 - Parameters for EPOS simulations.

Parameter	Value
Initial Angular Position	[-0.899156722810816, -0.142371232009588, 0.413820758445229; -0.413463218401238, -0.033497699477740, -0.909904429684233; 0.143406258069175, -0.989246347693926, -0.028745586110327]
Initial Angular Velocity (rad/s)	[0; 0; 0]
Initial Angular Acceleration (rad/s^2)	[0; 0; 0]
Limitless Simulation Time	On
Fixed Step Size (s)	0.004
Gravitational Acceleration (m/s^2)	[0; 0; 0]
Inertial Vector Target (outside workspace) (m)	[-13.4873508401215; -6.20194827020168; 2.15109387098346]
Inertial Vector Target (inside workspace) (m)	[0.2; -1.5; -1.7]
Initial Manipulator’s Joint Variables (deg)	[0;0;0]
Gains [Proportional; Integral; Derivative]	[1; 0.0002; 100]

Figure 3.20 - Angle in roll remains constant.

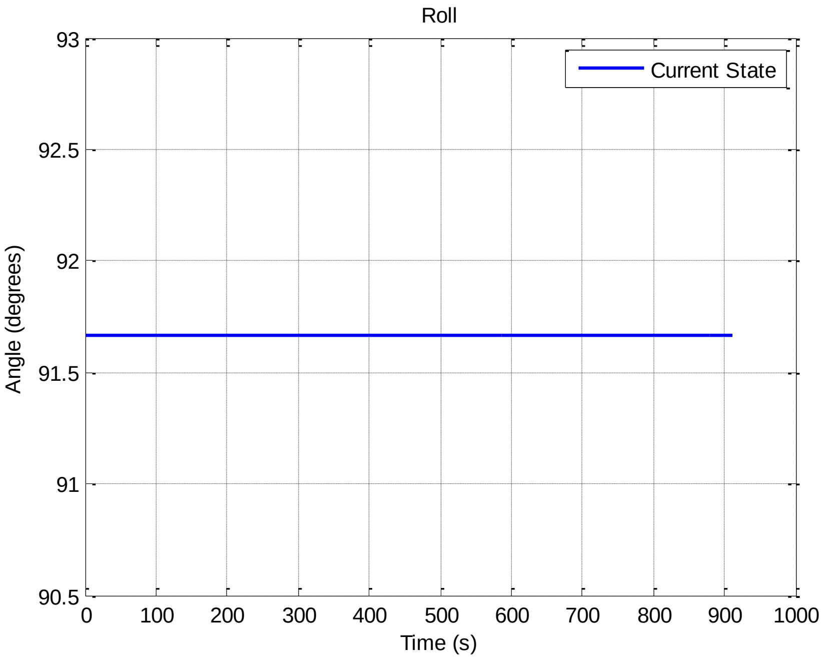


Figure 3.21 - Angle in pitch remains constant.

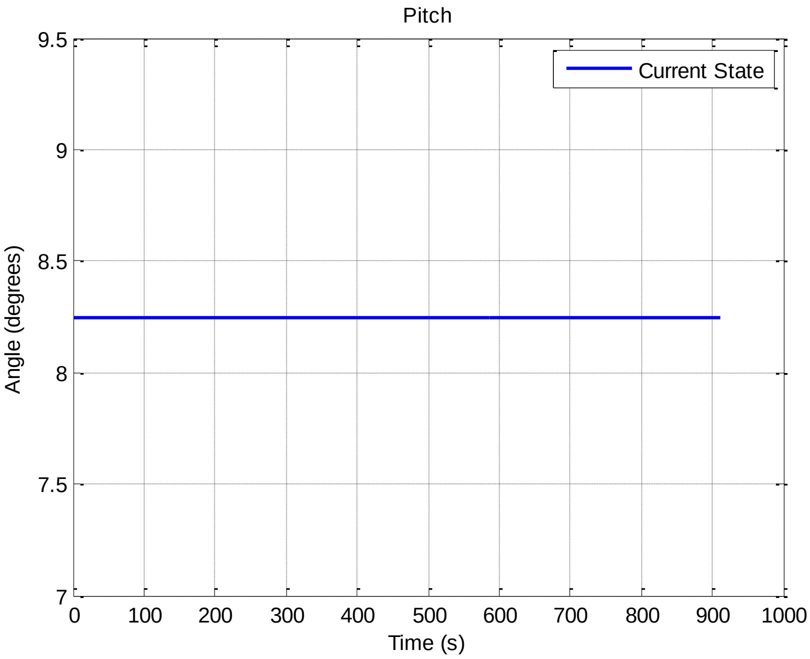


Figure 3.22 - Angle in yaw remains constant.

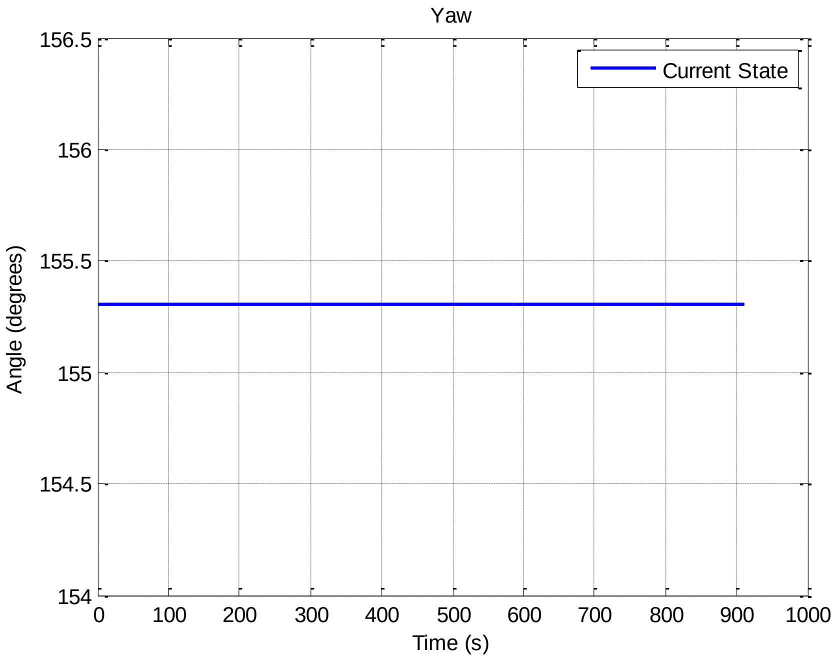


Figure 3.23 - Actuators torque maintained at zero.

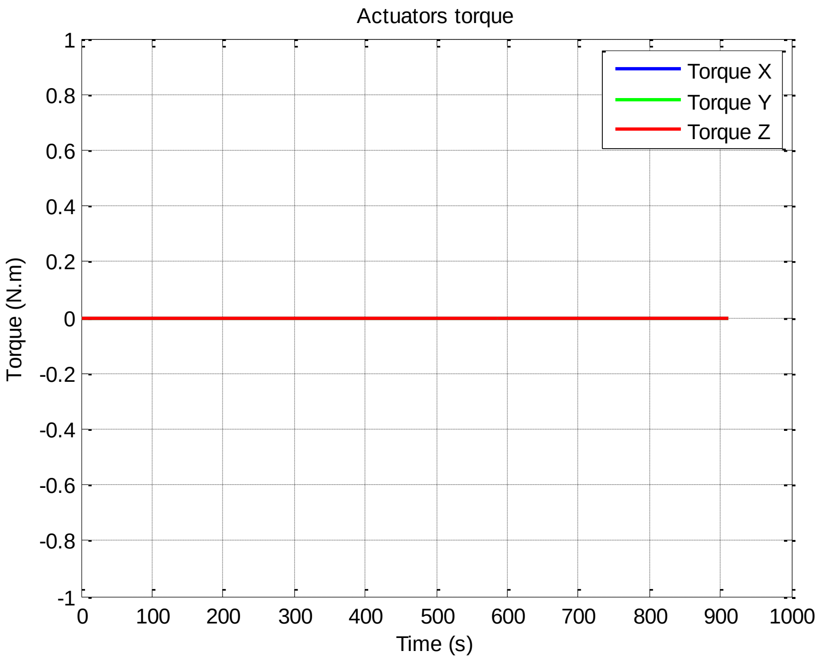
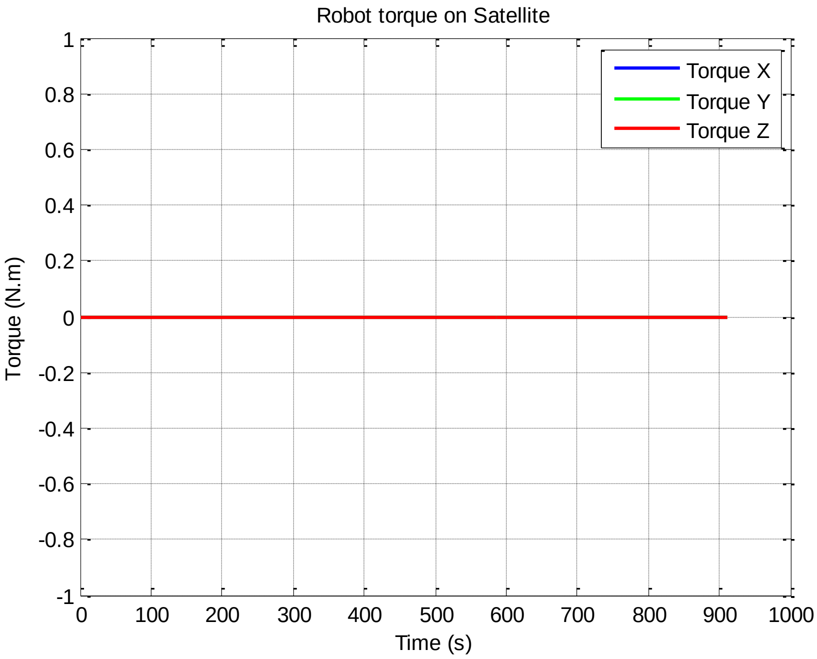


Figure 3.24 - Robot torque on Satellite maintained at zero.



After that, we should turn on the chaser satellite control system and then operate roll, pitch and yaw maneuver angles in order to show that the developed system can control the EPOS robots and emulate movements of the satellites.

Taking the movement of yaw as parameter, at approximately *200* seconds of simulation, the satellite control system was turned on. Figures 3.25, 3.26 and 3.27 show this task has been properly executed.

The two subsequent peaks in roll and pitch represent the coupling behavior of bonded dimensions. They occur after changing the yaw reference. Then the control system has shown to be able to pursue the reference, i.e., the devised control system commanded the current state in the sense of a different reference.

The changes on yaw reference can be identified by the presence of torques prevalently being performed in dimension *Z*, when we look at the actuators torque along simulation time, Figure 3.28.

Figure 3.25 - Angle in roll testing commands.

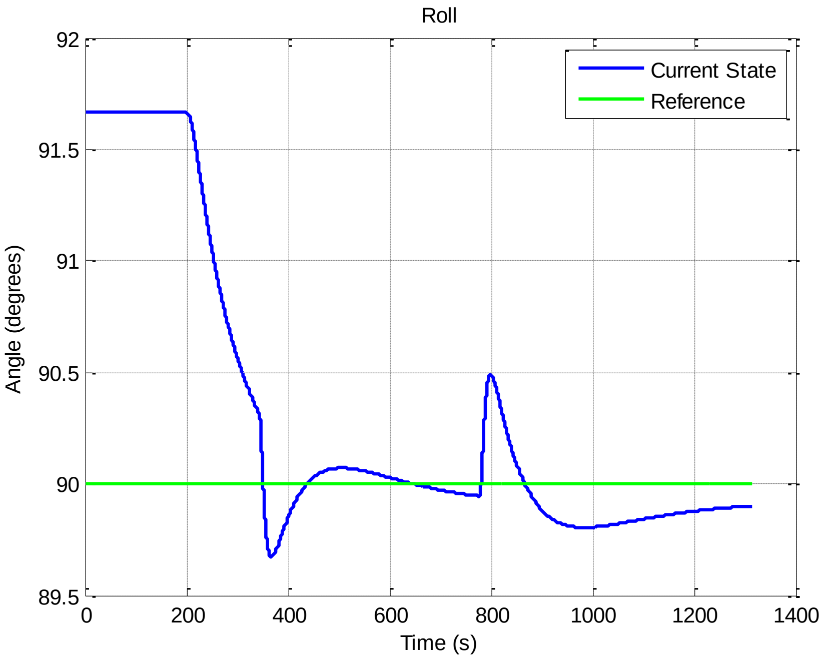


Figure 3.26 - Angle in pitch testing commands.

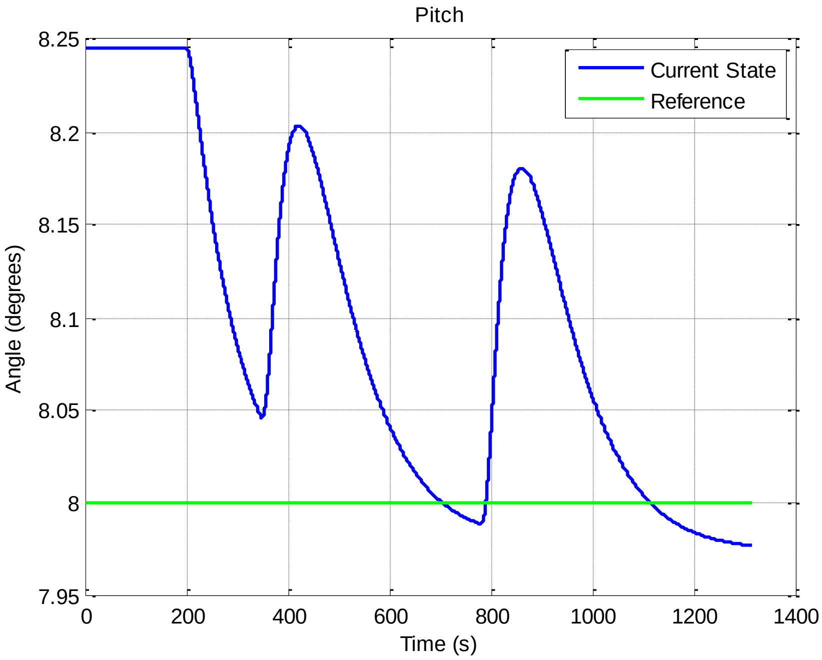


Figure 3.27 - Angle in yaw testing commands.

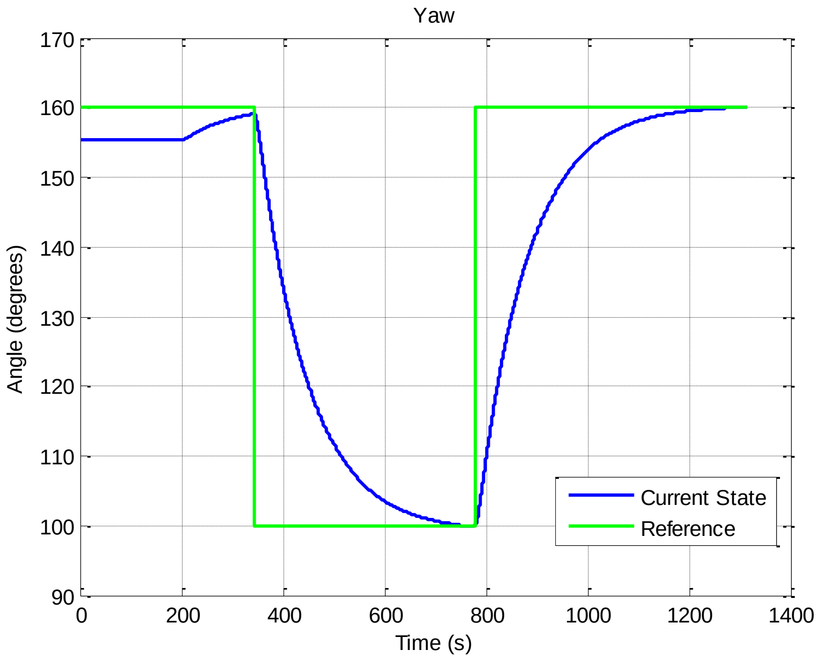


Figure 3.28 - Actuators torque testing commands.

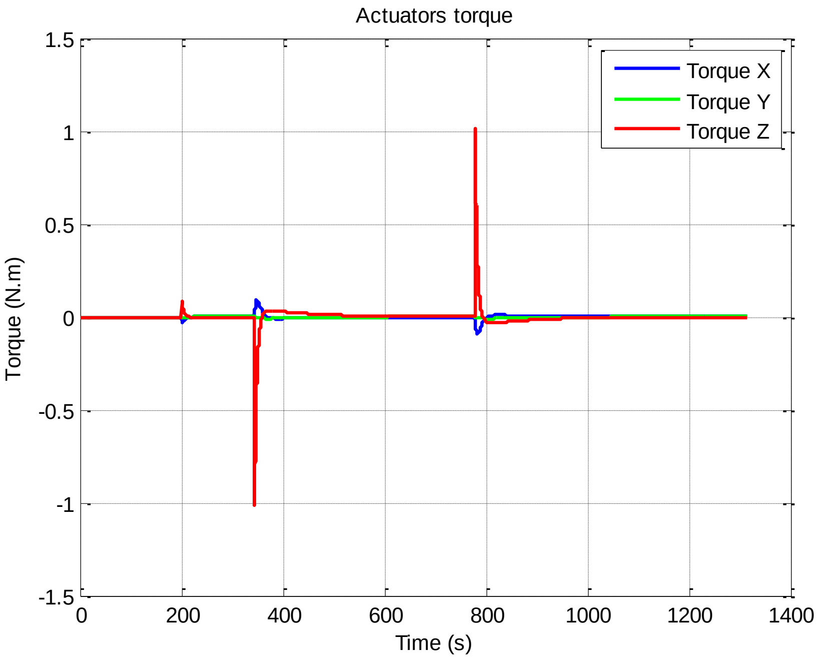
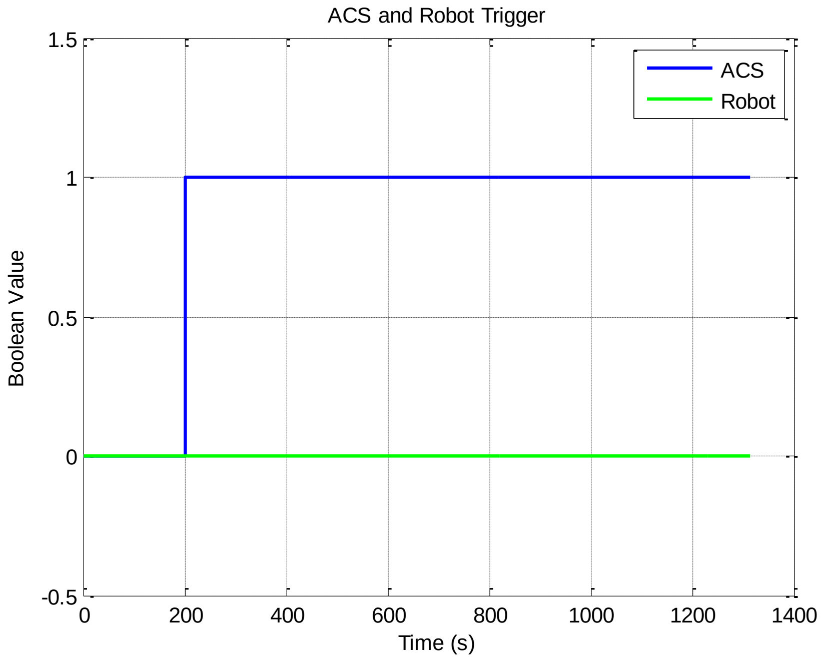


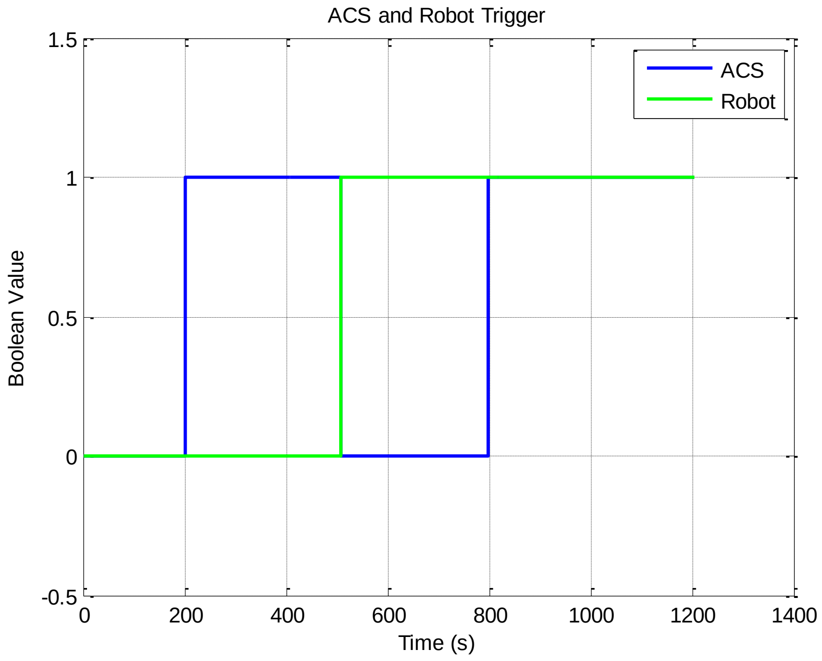
Figure 3.29 shows us when the attitude control system was turned on and confirms that the robotic manipulator was maintained turned off.

Figure 3.29 - ACS and robot trigger testing commands.



To see the virtual manipulator working, its control system was turned on while the satellite had its control system alternated between on and off. The satellite control system executed the capacity of achieving a commanded orientation at about *500* seconds when it was turned off. Simultaneously, the manipulator control system was turned on. Figure 3.30 shows us the order of actions.

Figure 3.30 - ACS and robot trigger with target out of workspace.



Despite the satellite had its control system turned off, we could verify its orientation changing due to manipulators movement toward the target vector. Figures 3.31, 3.32 and 3.33 present the satellite orientation changing along time in each direction roll, pitch and yaw, respectively.

Turning on again the satellite control system at about *800* seconds, its orientation was led to the desired reference. This test aimed to show us how the virtual manipulator behaves in the scenario of simultaneous motion of robot and satellite.

The manipulator took its end effector as close of the defined target vector as possible. In this case, the commanded target vector was out of the virtual manipulator workspace. Figure 3.34 shows that, as expected, the target point was not achieved by the manipulator.

Figure 3.31 - Angle in roll with target out of workspace.

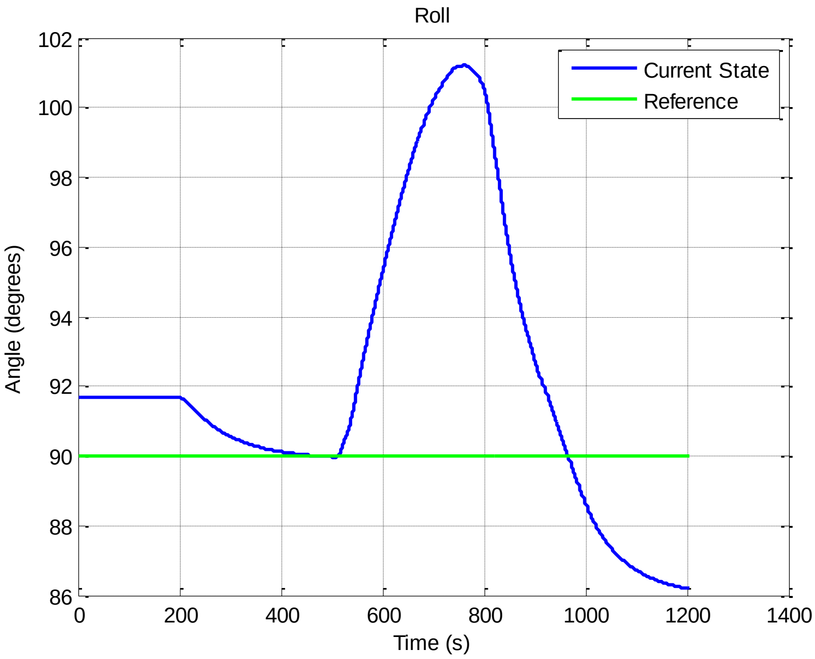


Figure 3.32 - Angle in pitch with target out of workspace.

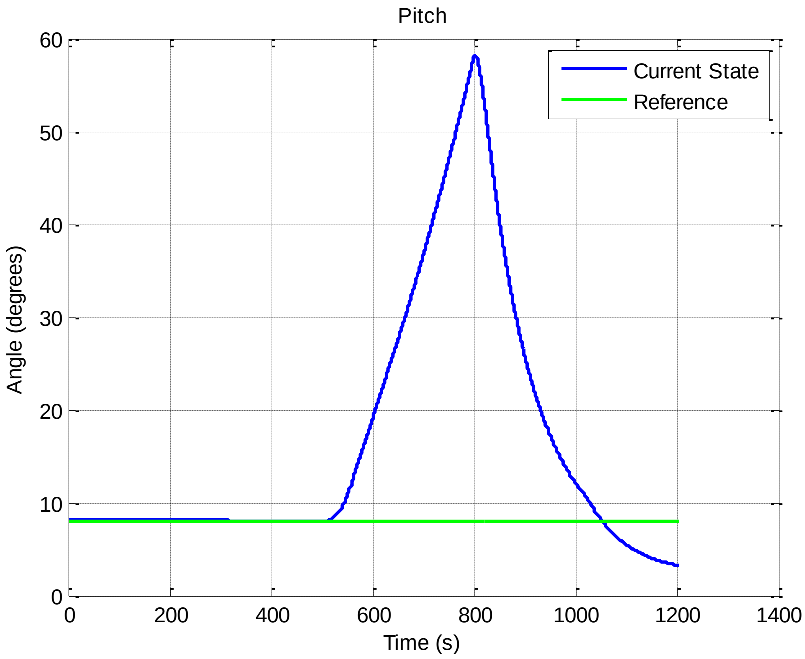


Figure 3.33 - Angle in yaw with target out of workspace.

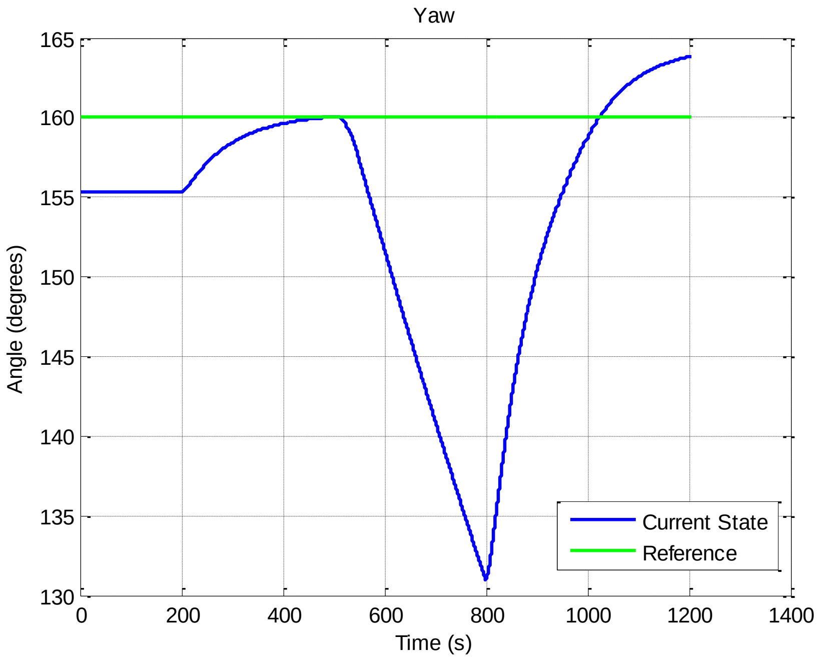
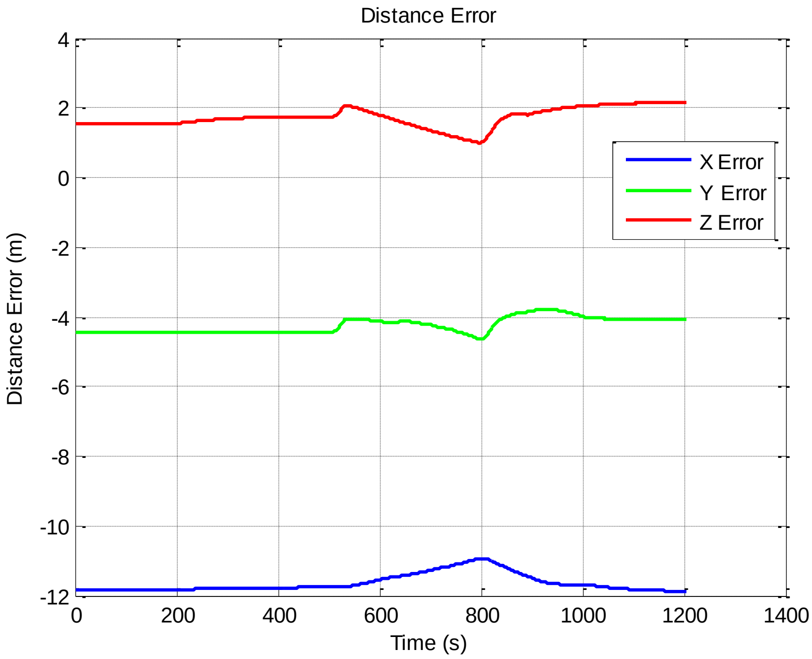


Figure 3.34 - Distance error with target out of workspace.



The actuators changed the satellite orientation exerting a higher torque at the second activation than at the first one, Figure 3.35. As the virtual manipulator was trigged, Figures 3.36, 3.37 and 3.38 show angular positions, velocities and accelerations, respectively, of each robot's joints moving along time.

Figure 3.35 - Actuators torque with target out of workspace.

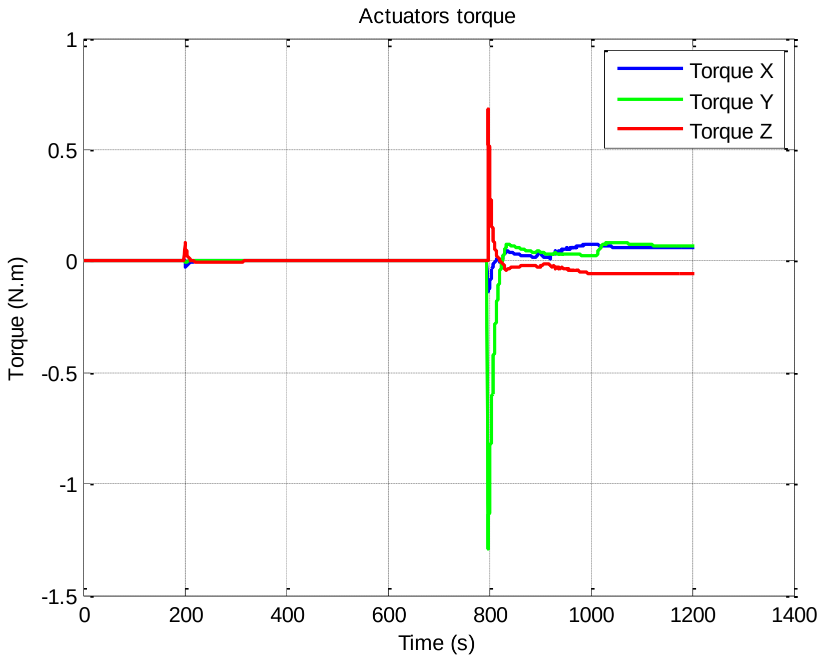


Figure 3.36 - Angular positions with target out of workspace.

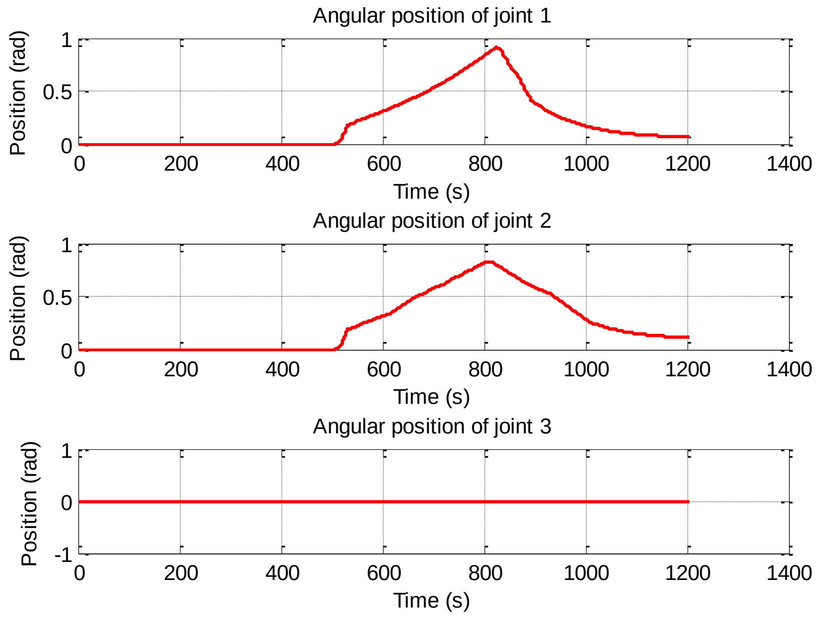


Figure 3.37 - Angular velocities with target out of workspace.

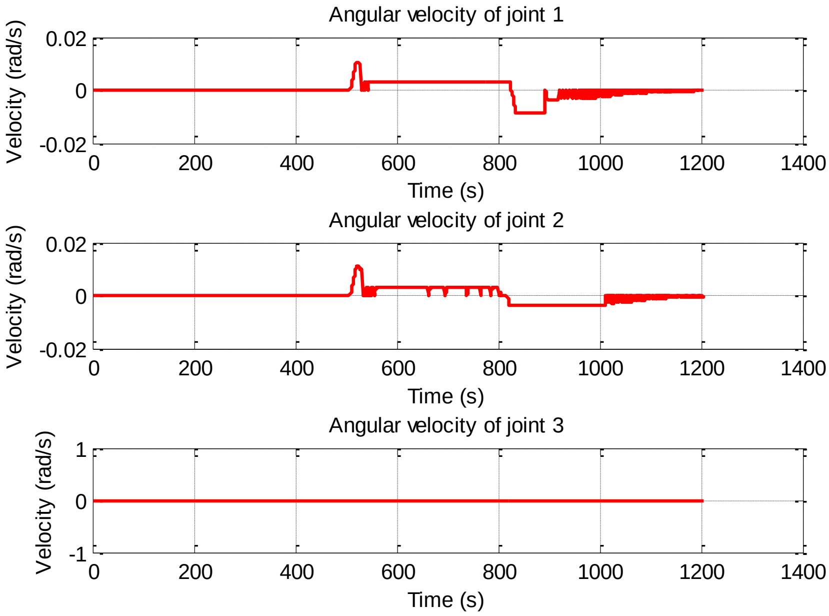


Figure 3.38 - Angular accelerations with target out of workspace.

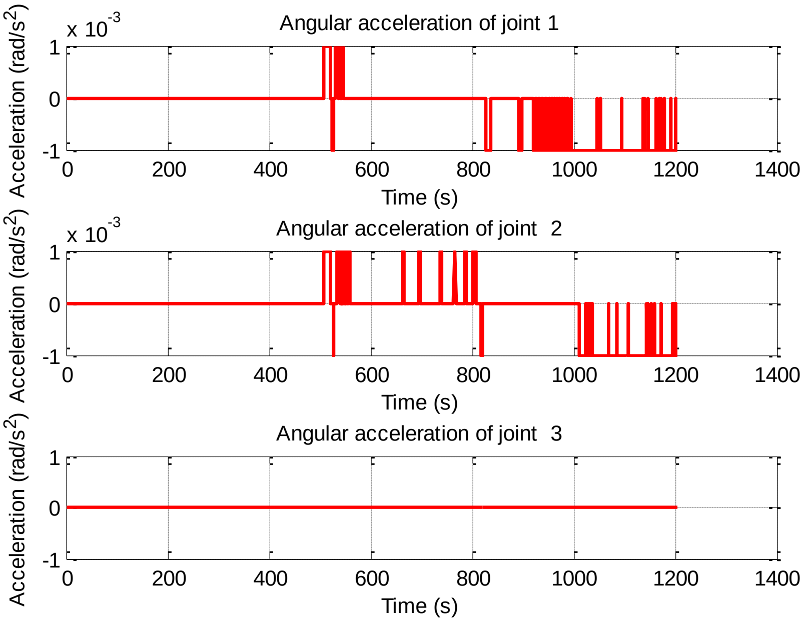
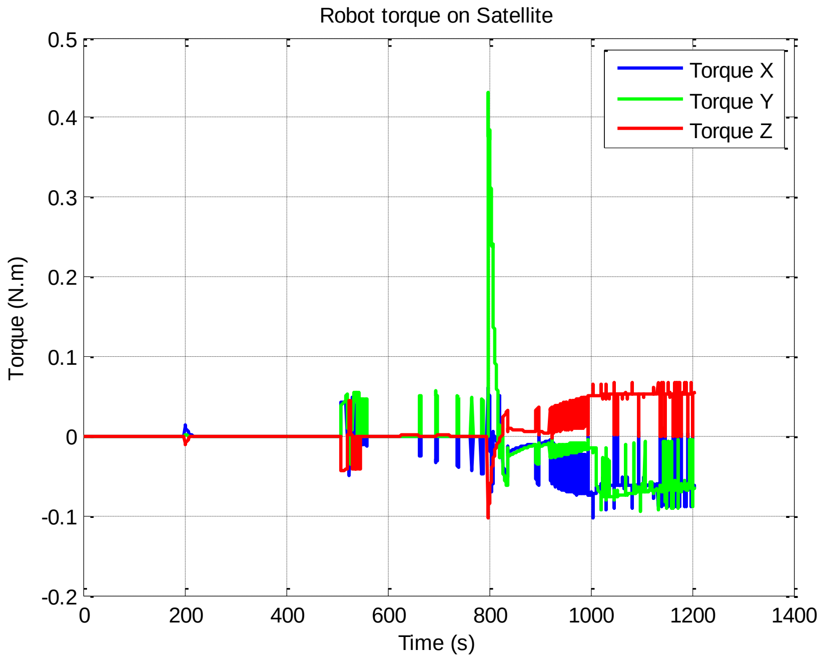


Figure 3.39 shows the torques applied on the base satellite by the manipulator with their noticeable points at *500* and *800* seconds.

Figure 3.39 - Robot torque on satellite with target out of workspace.



In a different experiment, Figure 3.40 shows the triggers when the target vector was defined to be inside the virtual manipulator workspace to demonstrate that it was able to reach such a point.

Firstly, the attitude control system was turned on while the virtual manipulator was remained turned off. Around *400* seconds, the ACS was turned off and the robotic manipulator had its control system turned on to pursue the target point inside its workspace, see Figure 3.41.

Figures 3.42, 3.43 and 3.44 show the satellite orientation changing along time in roll, pitch and yaw, respectively. It is possible to notice when the attitude control system was turned on at *100* seconds, achieved the reference and, after being turned off, the satellite orientation became solely subject to the virtual manipulator movements until when the ACS was turned on again, after *600* seconds. Figure 3.45 shows the actuators torque exerted by the attitude control system at the cited moments along simulation.

Figure 3.40 - ACS and robot trigger with target in workspace.

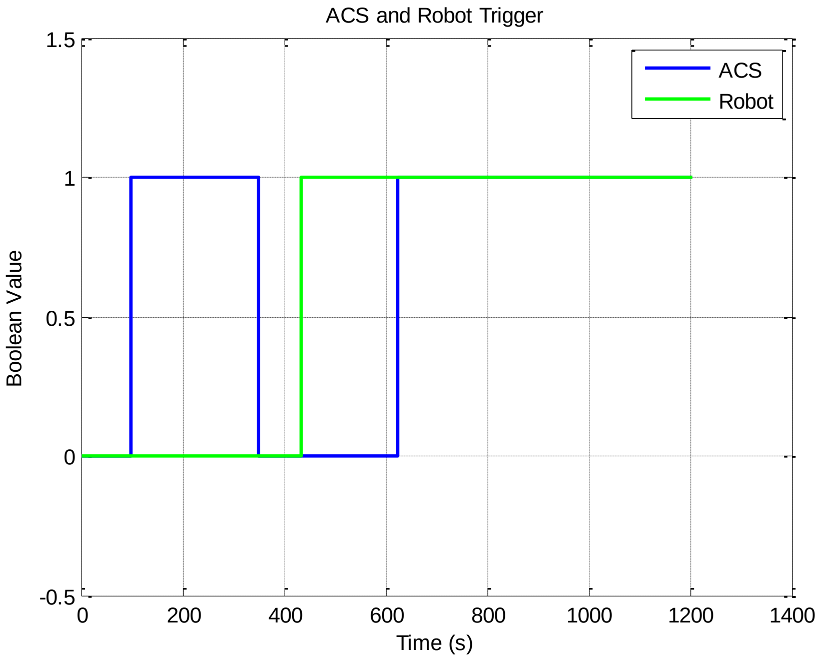


Figure 3.41 - Vector target along time with target in workspace.

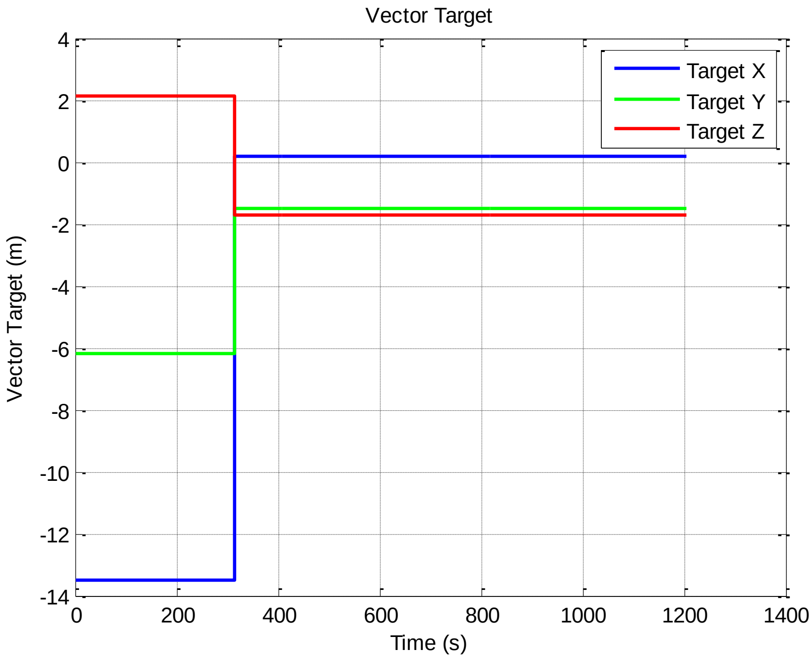


Figure 3.42 - Angle in roll with target in workspace.

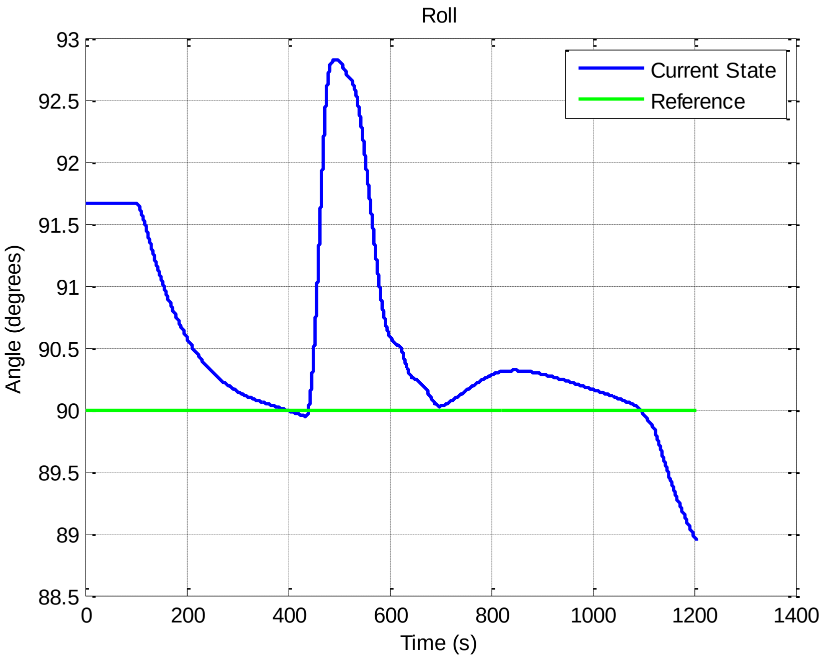


Figure 3.43 - Angle in pitch with target in workspace.

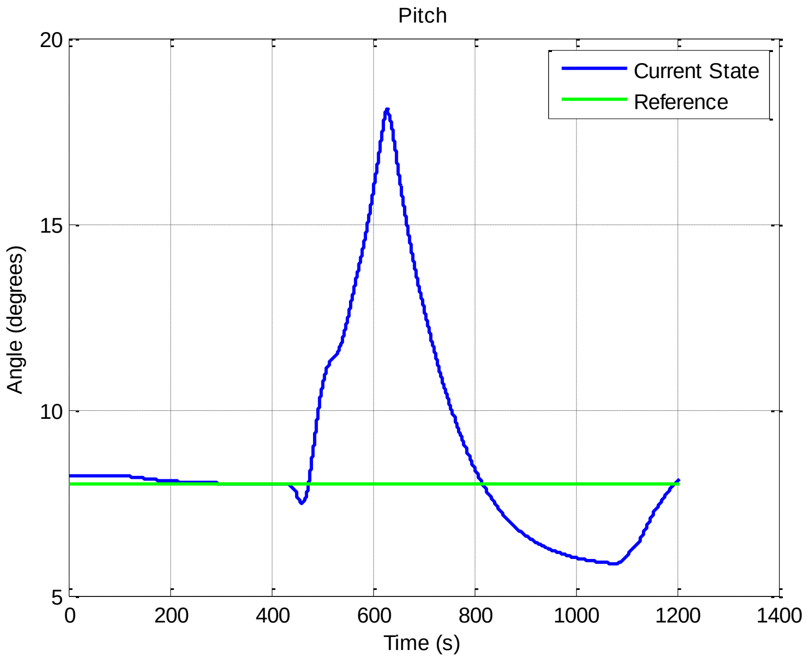


Figure 3.44 - Angle in yaw with target in workspace.

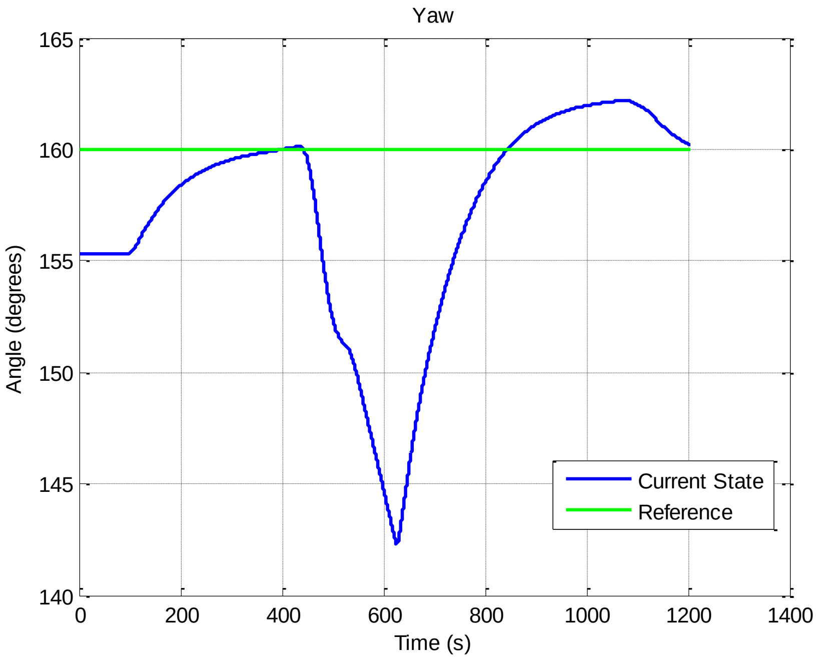


Figure 3.45 - Actuators torque with target in workspace.

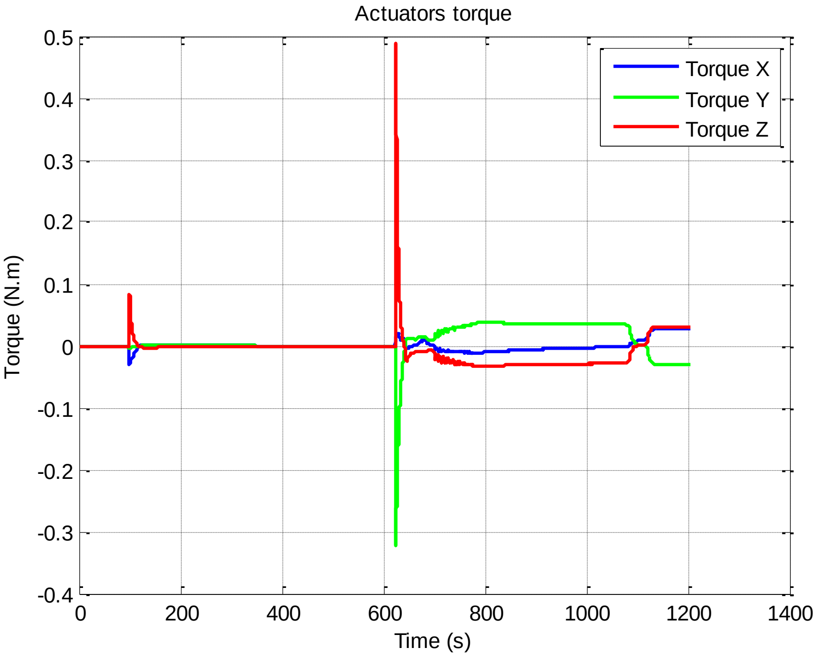


Figure 3.46 exemplifies results that the simulator is able to provide for the moments of inertia of the arrangement, formed by base satellite plus robotic

manipulator, while Figure 3.47 illustrates results from other calculations, the center of mass for the ensemble.

Figure 3.46 - Arrangement moments of inertia with target in workspace.

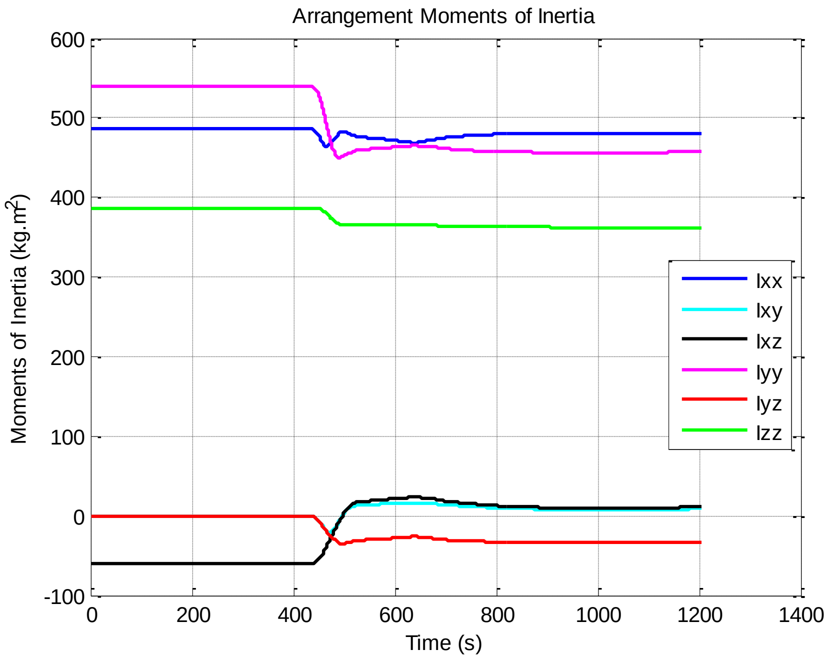
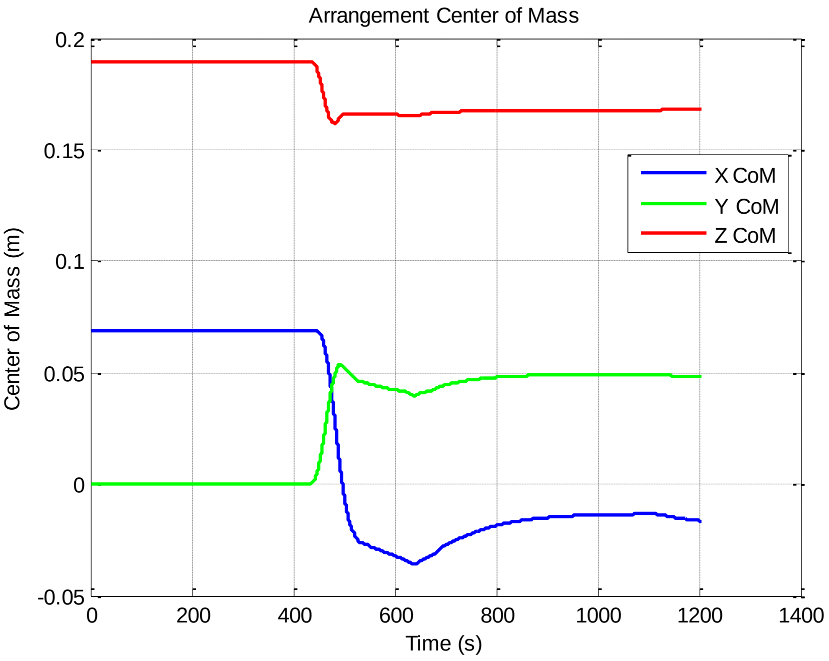


Figure 3.47 - Arrangement center of mass with target in workspace.



Figures 3.48 and 3.49 prove that the manipulator was able to achieve the target point since the distance error tended to zero and joints had velocity zero.

Figure 3.48 - Distance error to the target with target in workspace.

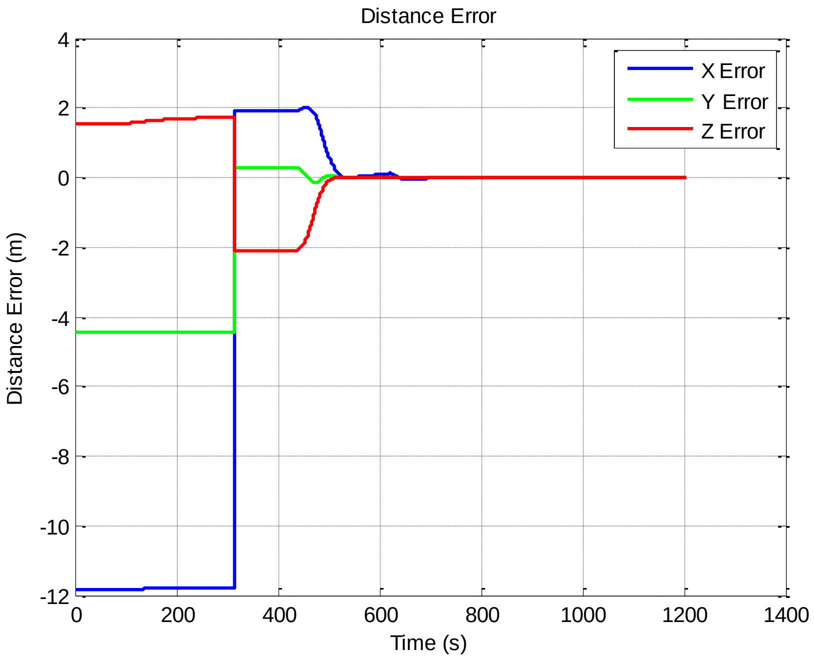


Figure 3.49 - Joint angular velocities with target in workspace.

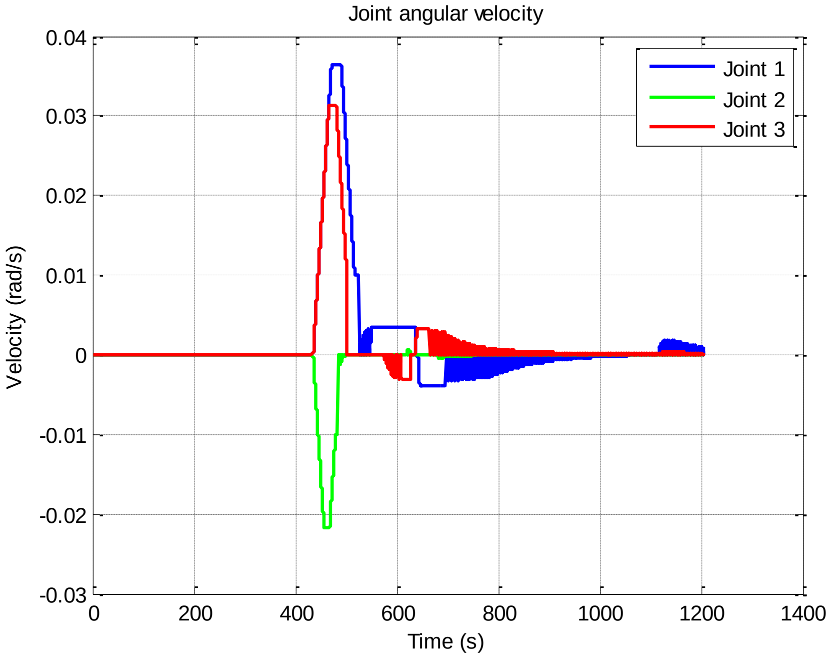
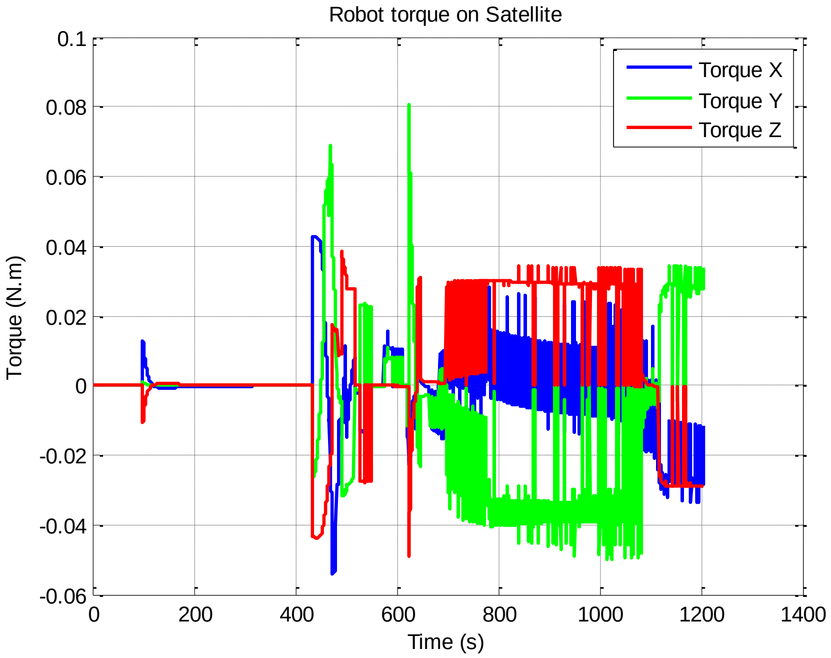


Figure 3.50 presents the torques applied on the satellite by the robotic manipulator. The torques presented here were treated as disturbances by the satellite control system.

Figure 3.50 - Robot torque on satellite with target in workspace.



The previous tests were replicated, now with more steps during a longer simulation to assure reliability for the simulator.

Figure 3.51 shows the triggering history for the virtual robot and the attitude control system. After *1500* seconds in this chart, where we see just a green line edge, it is instead a double change with the blue line going down. Figure 3.52 shows the vector target displacement, from being defined outside to inside the virtual robotic manipulator workspace.

Figure 3.53 presents the actuators torque, where we can see how the attitude control system behaved to satisfy those angles of attitude commanded at each time. The remarkable moments are at the peaks when the actuators were required more. Figure 3.54 presents in a tridimensional space how the arrangement center of mass has changed along the simulation.

Figure 3.51 - ACS and robot trigger in a longer simulation.

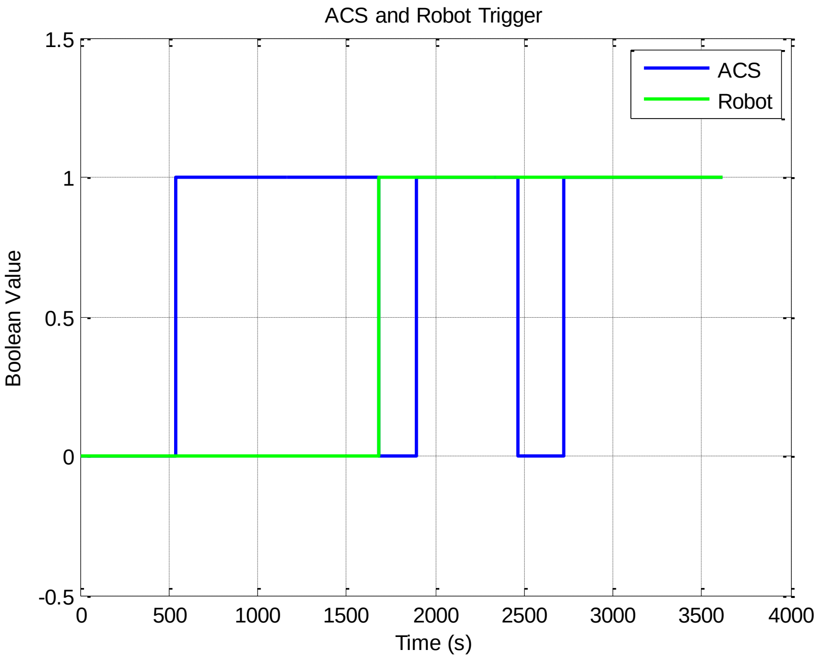


Figure 3.52 - Vector target along time in a longer simulation.

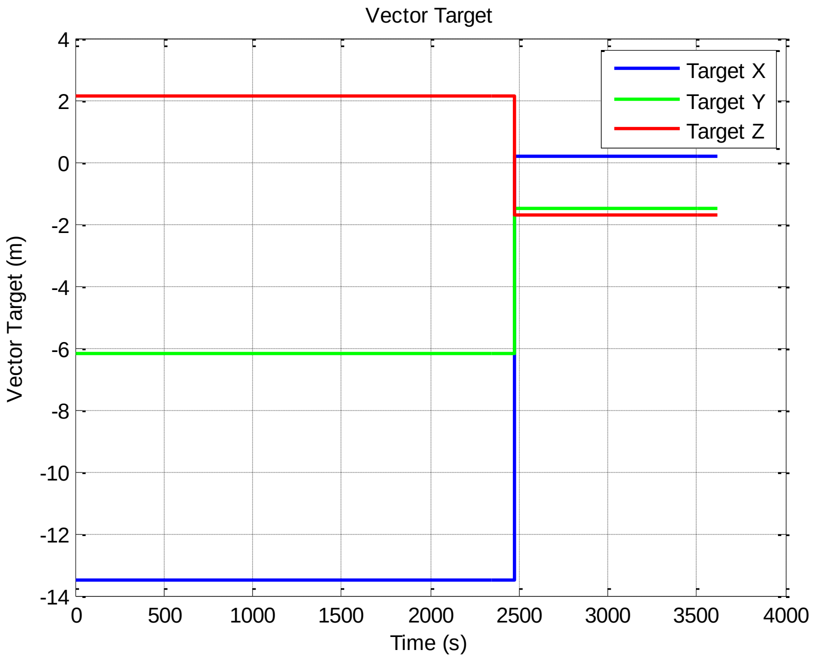


Figure 3.53 - Actuators torque in a longer simulation.

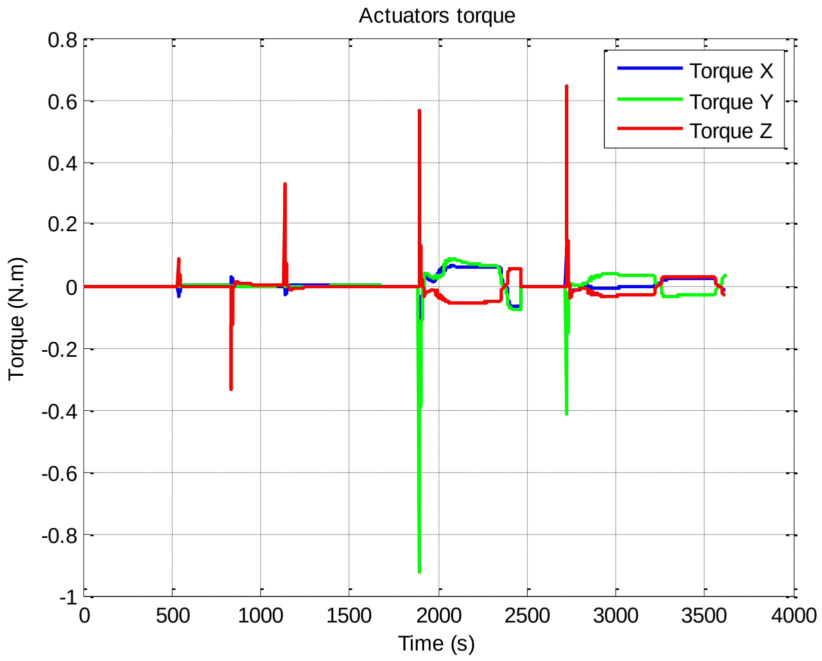


Figure 3.54 - Arrangement center of mass 3D in a longer simulation.

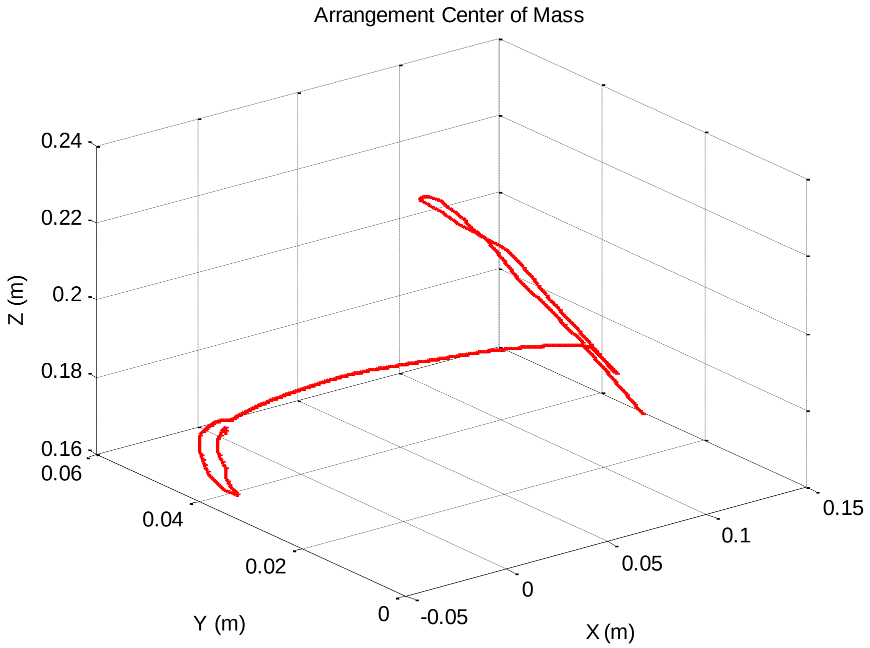


Figure 3.55 shows the distance error tending to zero, this proves that the target point was reached by the virtual manipulator end effector. Also, this can be

confirmed by Figure 3.56, where the joints velocities are presented and at some point all joints had null velocities.

Figure 3.55 - Distance error to the target in a longer simulation.

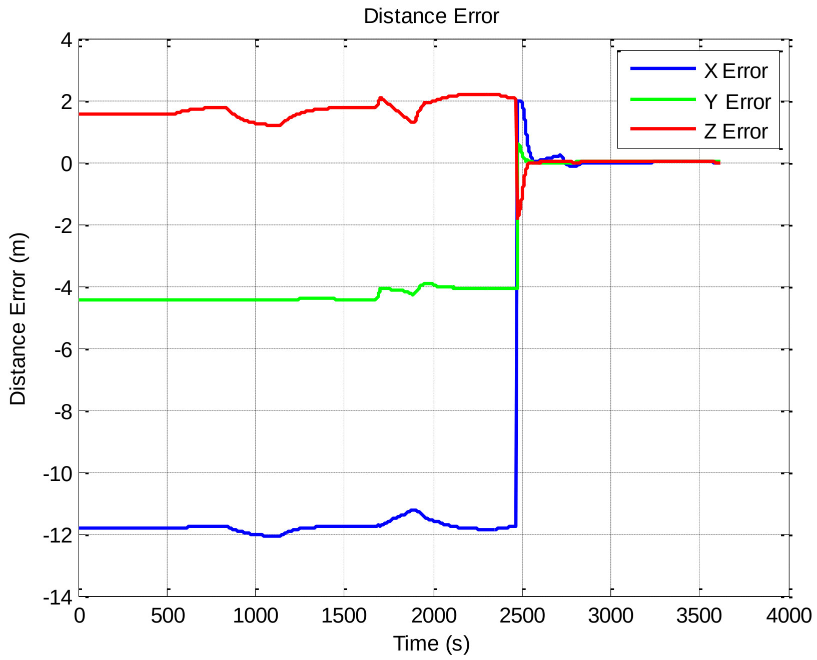
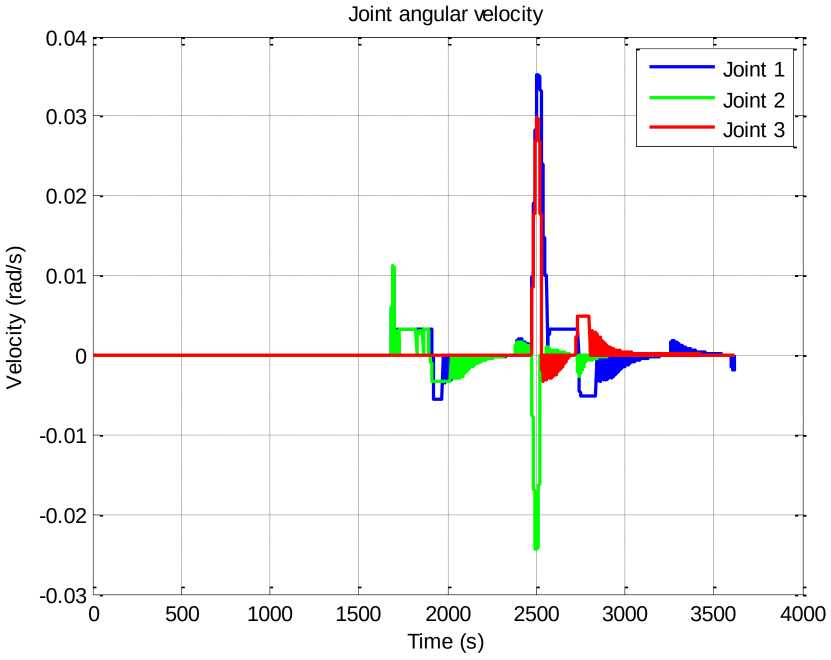


Figure 3.56 - Joint angular velocities in a longer simulation.



In order to evaluate the satellite control system, Figures 3.57, 3.58 and 3.59 show that the most acute peaks match with moments when the ACS was turned off.

Figure 3.57 - Angle in roll in a longer simulation.

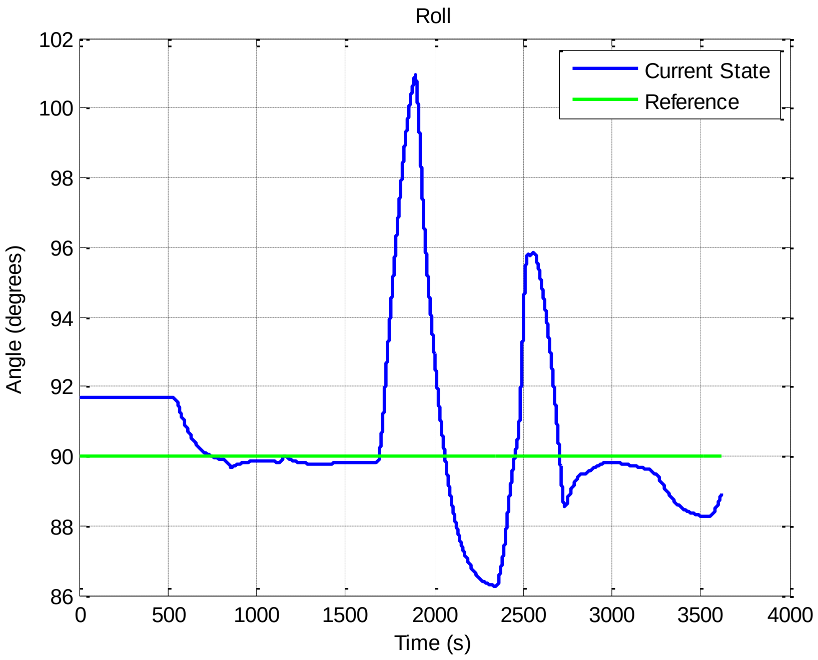


Figure 3.58 - Angle in pitch in a longer simulation.

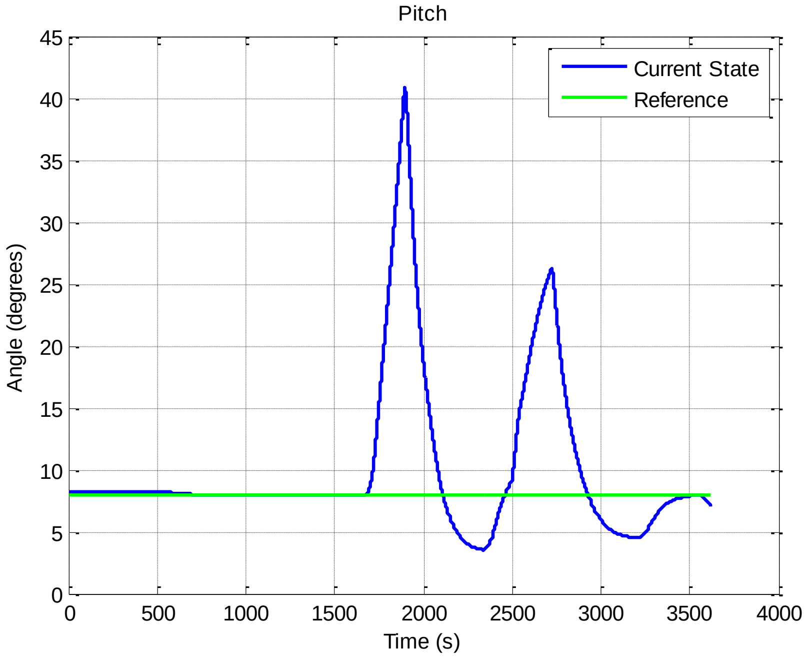
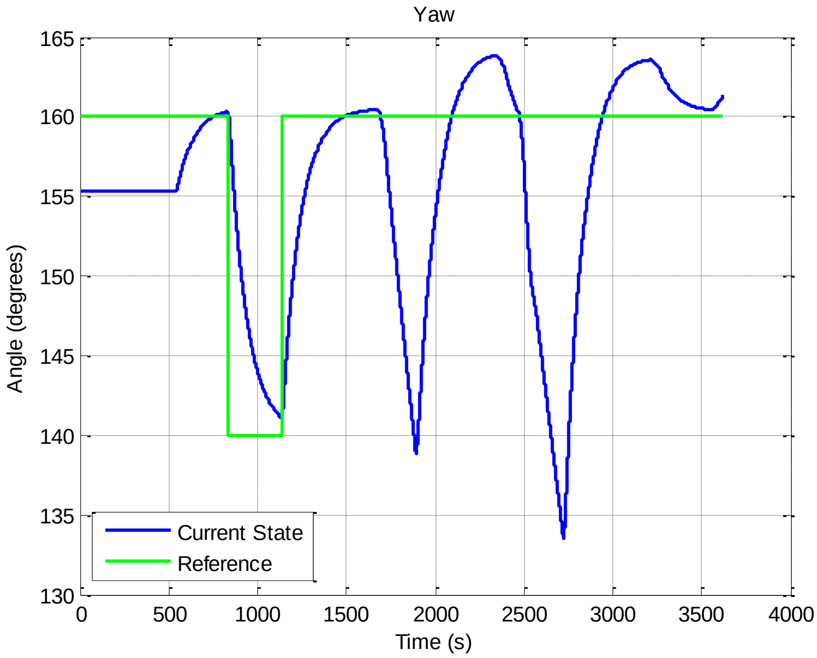
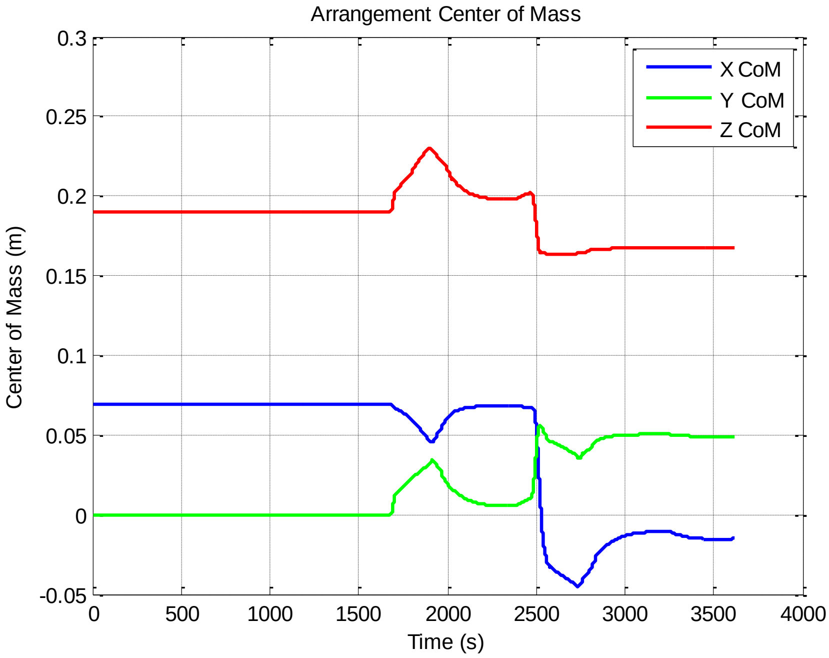


Figure 3.59 - Angle in yaw in a longer simulation.



The simulator also provides the changes in the arrangement center of mass position along time splitting it for each dimension, Figure 3.60.

Figure 3.60 - Arrangement center of mass along time in a longer simulation.



The experiments using EPOS demonstrated the consistency of algorithms and methods developed for model dynamics as well as for the manipulator and satellite combined motion in the software simulator.

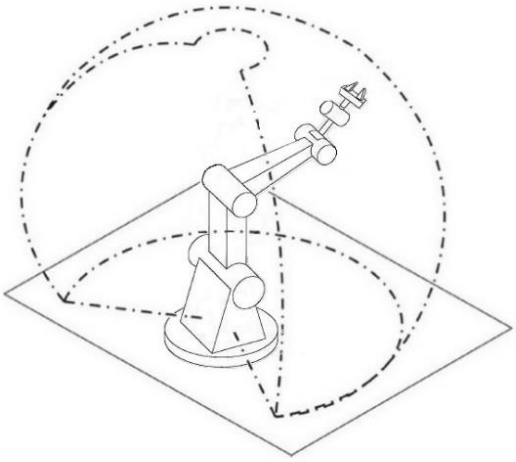
Indeed, the EPOS experiments provided reliability for the developed models, i.e., the results contributed to the simulation software suitability. At EPOS it was ensured that the software was able to work with HIL. The increase in experimental complexity fomented further applications employing optimization.

3.4. Accuracy and other terms

Some terms must be defined to properly understand their implications for the future results presented here. These terms are accuracy, resolution, repeatability, and workspace (KUTTAN, 2007; GROOVER, 2019; SICILIANO et al., 2009).

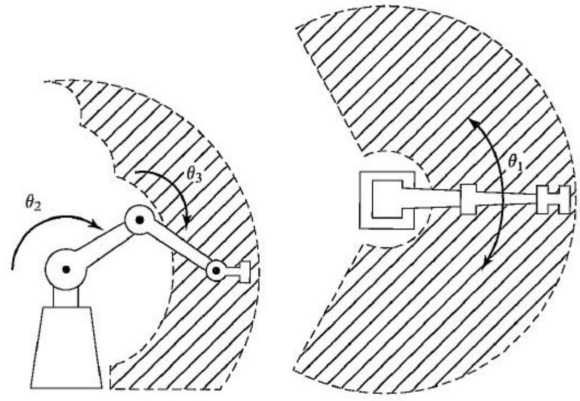
The workspace, or work volume, is the space within which the robot can manipulate its wrist. The workspace determination considers the robot physical configuration, the size of the various parts that compound the shape, and the limits of the robot joints. Figure 3.61 shows the space around a robot comprising the workspace and Figure 3.62 presents the workspace as a shaded area from different views.

Figure 3.61 - Workspace of a revolute robot.



Source: Adapted from Craig (2005).

Figure 3.62 - Workspace from different views.



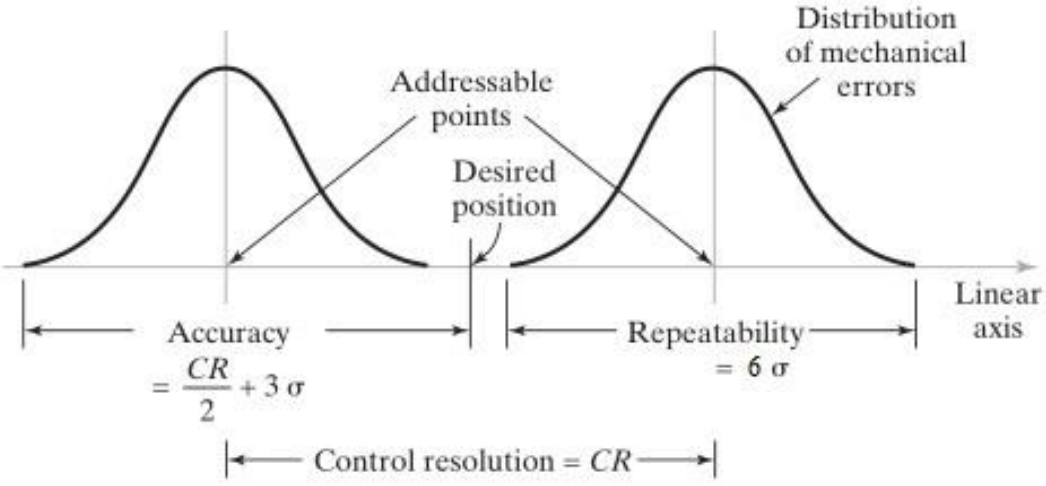
Source: Adapted from Craig (2005).

Resolution, or control resolution, is defined as the smallest increment of motion in which the robot can divide its workspace based on the minimum increments of its joints, which defines closely spaced points, called addressable points for the wrist. It is the distance separating two adjacent addressable points, it is smaller when the desired position is closer to the robot's workspace center.

Repeatability is the ability to repeatedly position the wrist on a desired target point inside the workspace. Each time the manipulator returns to the controlled point, it is subject to some variations due to mechanical errors such as gear backlash, link deflection, hydraulic fluid leaks etc. These factors create a slight difference in positioning. Mechanical inaccuracies are represented by the assumed normal statistical distribution of errors.

Accuracy is the robot's ability to position its wrist at a desired location within the workspace. It represents the difference between the aimed point and the effectively achieved one. It is defined for the worst case in which the desired point lies in the middle between two addressable points. Figure 3.63 shows accuracy, repeatability and resolution for a portion of space that includes two nearby addressable points.

Figure 3.63 - Accuracy, repeatability, resolution.


Distribution of mechanical errors
Addressable points
Desired position
Linear axis
Accuracy $= \frac{CR}{2} + 3\sigma$
Repeatability $= 6\sigma$
Control resolution = CR

Source: Adapted from Groover (2019).

Accuracy varies within the work volume, tending to be worse in the outer space of its workspace, away from its base and better when closer to its base (KUTTAN, 2007; GROOVER, 2019; SICILIANO et al., 2009). This results that the mechanical inaccuracies are more significant when the manipulator is completely extended. It is generated an error map that characterizes the accuracy of the manipulator as a function in the workspace.

3.5. Multi-objective optimization

A classic optimization problem is finding solutions that represent the optimal value for an objective function. It is important to draw attention to the issue of obtaining an optimal solution, considering its possible existence and uniqueness, which characterizes the answer that could be called the solution of the classical optimization problem.

For example, finding the optimal value for a given process whose evolution describes a downward concave parabola function (second temporal derivative is negative) over time corresponds to finding the minimum value of this function. To do so, we investigate the point at which the slope is zero.

However, in engineering applications (and other areas of knowledge), in many cases, problems occur that require simultaneous optimization of more than one goal. We are faced with the possibility that there is no solution that produces an optimal one for each of the goals simultaneously.

A real problem associated with multi-objective optimization is where the desired objectives are conflicting, such as speed and fuel consumption of any vehicle. For cases like this, designers use compromise solutions in which all objectives are acceptable to a certain degree.

Multi-objective optimization is, therefore, an area of decision making with multiple criteria. It is inserted in the optimization of mathematical problems involving more two or more objective functions to be simultaneously optimized, i.e., it is the process of systematically and simultaneously optimizing a group of objective functions. Sometimes such objective functions do not present possible simultaneous optimization, this occurs when dealing with conflicting objectives (COHON, 1978).

Algorithms have been created to search for multi-objective problem solving. Proper programming of such algorithms is a fundamental part of finding acceptable solutions. Multi-objective programming and planning is concerned with decision making problems in which there are several conflicting objectives. It would, therefore, be up to the decision maker (DM) to attribute the relative importance of the objectives (COHON, 1978).

In engineering, many problems are not simply described as “the bigger, the better” or “the smaller, the better”, sometimes there is a certain desired value for each objective and the desire is to get as close to this value as possible at any given time. An example would be to control the position of a satellite and its fuel usage.

The general multi-objective optimization problem comprises n decision variables (forming vector $\boldsymbol{x}$) for which values must be selected, i.e., n parameters that allows the search for an optimum of the objective function $\mathbf{Z}(\mathbf{x})$, m constraint functions $g(\boldsymbol{x})$ and p objectives. Such a problem can be defined by Equation 3.42 (COHON, 1978):

$$\begin{aligned} &\max \mathbf{Z}(\mathbf{x}) = [Z_1(\mathbf{x}), Z_2(\mathbf{x}), ..., Z_p(\mathbf{x})] \\ &\mathbf{x} \in \mathbf{F}_d \end{aligned} \tag{3.42}$$

Where the vector $\boldsymbol{Z}: \boldsymbol{F}_d \rightarrow \mathbb{R}^n$ consists of the objective functions and $\boldsymbol{F}_d$ is the feasible area of the decision space, defined by Equation 3.43:

$$\mathbf{F}_d = \{\mathbf{x} \in \mathbb{R}^n \mid g_i(\mathbf{x}) \leq 0, i = 1, 2, ..., m; \mathbf{x} \geq 0\} \tag{3.43}$$

A solution of $\boldsymbol{x}$ can only be considered optimal for a certain group of objectives if a better solution of $\boldsymbol{y}$, considering all the objectives, does not exist. It is said that a solution of $\boldsymbol{x}$ dominates another solution of $\boldsymbol{y}$ if (Equations 3.44 and 3.45):

$$Z_j(\mathbf{x}) \geq Z_j(\mathbf{y}) \tag{3.44}$$

$$Z_k(\mathbf{x}) > Z_k(\mathbf{y}) \tag{3.45}$$

For all indexes $j = 1,2, \dots, p$. And for at least one of the indexes $k = 1,2, \dots, p$.

A solution is called Pareto's solution if there is no other that dominates it. This optimal solution can also be called non-inferior or non-dominated solution. The set of optimal solutions, or non-inferior set, is called Pareto frontier. Any candidate that belongs to the frontier could be chosen as a solution for the multi-objective problem, i.e., the degree of optimality would be the same for any solution in this group. Therefore, the solution choice would be made by a DM and not automatically (PARETO; MONTESANO, 2014).

Without adding subjective preference information, all Pareto optimal solutions are considered equally good. From a different point of view, solving such optimization problems can be understood as finding a set of Pareto solutions, quantifying requirements that meet different objectives, or finding a single solution that meets the preferences subjectively defined by a human decision maker.

The decision maker is regarded as an expert in the problem domain supposedly able to provide better solutions than anyone else and would play an important role in selecting appropriate solutions. A significant advantage of the *Smallest Loss Criterion* (ROCCO, 2002) lays on the fact of not depending on such a DM.

Many methods convert the original multi-objective problem into a mono-objective optimization problem. Some examples of multi-objective approach will be presented.

3.5.1. Weighting Method

Weighting Method considers the combination of objectives through influence factors. The objective weights are given by $w_k > 0$. Thus, we have Equation 3.46:

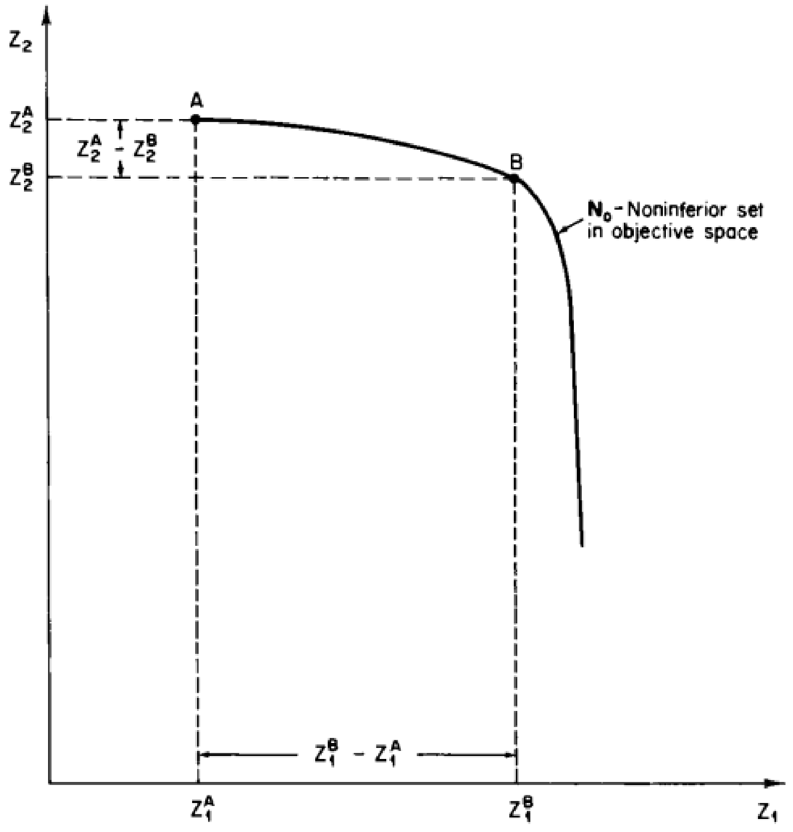
$$\max Z(\mathbf{x},\mathbf{w}) = \sum_{k=1}^{p} w_k Z_k(\mathbf{x}) \tag{3.46}$$

According to Cohon (1978) the generation of the optimal solution set for a multi-objective problem is particularly effective when the set exhibits a “bent” shape like the one in Figure 3.64.

The best compromise solution is obtained in the vicinity of point B. For example, going from point B to point A, a relatively small improvement in the vertically represented objective is associated with a large decrease, comparatively, to the objective described on the horizontal axis.

This method consists of a non-inferior solution set generator. It is the oldest technique of obtaining multi-objective solutions (COHON, 1978).

Figure 3.64 - Set non-inferior “bent”.



Source: Cohon (1978).

3.5.2. Smallest Loss Criterion

Pareto's multi-objective optimization methods provide a group of solutions that are equally rated in terms of their quality in meeting the various objectives that are conflicting in their realization, but in practical applications it would be interesting to apply a methodology capable of finding a solution that meets all objectives simultaneously as well as possible.

Considering Table 3.5, for a case of satellite orbital maneuvers, each maneuver requires a velocity increment, a time and generates a positioning error. We want to find a maneuver that minimizes such quantities treated as conflicting objectives.

Table 3.5 - Orbital Maneuvers.

	$\delta\theta$ (rad)	Δv (km/s)	T (s)
1	0.565637	1.07333	1660
2	0.445267	1.02667	1700
3	0.314329	0.99600	1800
4	0.118357	0.88733	2110
5	0.217551	0.85493	2250
6	0.241902	0.84267	2305
7	0.275673	0.80400	2550
8	0.278017	0.79867	2600
9	0.291305	0.78667	2705
10	0.289064	0.77867	2800
11	0.314594	0.76400	2910
12	0.337307	0.76212	2990

Source: Rocco; Souza; Prado (2003).

A solution that minimizes all three objectives simultaneously does not exist. Therefore, according to the application of Pareto's methodology, we find the non-dominated solutions to the problem. Solutions *4, 12*, and *1* are non-dominated, each minimizing one of the objectives. If we choose any of these solutions, we would be prioritizing one objective over the others.

In the *Smallest Loss Criterion* (SLC), a balanced optimal solution definition for a multi-objective problem is proposed. Therefore, the optimal solution in this case would be the lowest loss solution for all objectives (ROCCO, 2002; ROCCO; SOUZA; PRADO, 2003).

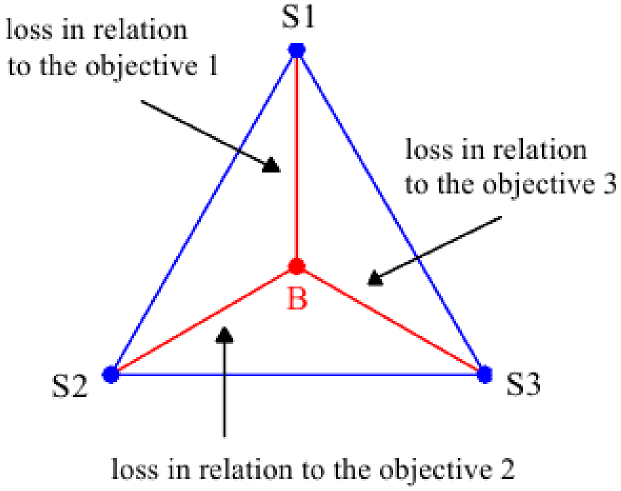
To get the solution that does not take any objective as a priority, you would need to choose an intermediate solution. Such a choice is based on the symmetry argument observed in various natural manifestations (ROCCO, 2002).

In multi-objective problems, solutions with the same symmetry are expected. For this reason, an extreme solution cannot be considered as an optimal solution. Only an intermediate solution can consider the symmetry between the optimal solution candidates. One possibility to find such a solution is given by the barycenter (also called centroid or geometric center) method.

Barycenter is the solution that generates the least loss in relation to all objectives. Thus, the best solution to the multi-objective problem would be the central point of the figure that has as its vertex the optimal solutions for each objective. This holds true regardless of the number of dimensions (objectives) we are dealing with. Sometimes the barycenter is not among the possible solutions, for these cases it is gotten the closest solution possible.

In Figure 3.65 we have the example for three conflicting objectives, where B is the barycenter of the figure, i.e., the line segments between B and $S1$, $S2$ or $S3$, (which represent the solutions that optimize each of the objectives individually) are equal. Each of these segments represents the loss relative to one of the objectives. When the *Smallest Loss Criterion* reveals a barycenter that is not among the feasible solutions, we choose the closest option to the barycenter.

Figure 3.65 - Losses with relation to the objectives.



Source: Rocco; Souza; Prado (2003).

For the example addressed (ROCCO; SOUZA; PRADO, 2003), the objectives had their values normalized dividing them by the maximum value, Table 3.6. This normalization is necessary as the solution depends on the size of each objective. With $\delta\theta_{max} = 0.7\ rad$, $\Delta v_{max} = 1.2\ km/s$, and $T_{max} = 3000\ s$.

Table 3.6 - Normalized solutions.

	$\delta\theta/\delta\theta_{max}$	$\Delta v/\Delta v_{max}$	T/T_{max}
1	0.808053	0.894442	0.533333
2	0.636096	0.855558	0.566667
3	0.449041	0.830000	0.600000
4	0.169081	0.739442	0.703333
5	0.310787	0.712442	0.750000
6	0.345574	0.702225	0.768333
7	0.393819	0.670000	0.850000
8	0.397167	0.665558	0.866667
9	0.416150	0.655558	0.901667
10	0.412949	0.648892	0.933333
11	0.449420	0.636667	0.970000
12	0.481867	0.635100	0.996667

Source: Rocco; Souza; Prado (2003).

We realize that, using this method, the solution does not depend on a human decision maker's preference because a systematic seek for equilibrium is applied. Besides the SLC, we could cite the Nash arbitration method (NASH, 1950) as another example of such an approach for multi-objective problems.

The SLC tackles a natural problem of other methods that prioritize a given objective, which sometimes reduce the problem to a single objective problem, by pursuing a single best compromise solution associated to the smallest loss for all objectives.

The SLC finds the barycenter of a normalized *p*-dimensional figure. Being p the number of objective functions. For all indexes $k = 1,2,\ldots,p$, the barycenter solution, $\mathbf{Z}(\mathbf{x}^*) \in \mathbb{R}^p$, represents the equilibrium point in the objective space and it is computed by Equation 3.47.

$$\mathbf{Z}(\mathbf{x}^*) = \frac{\sum_{k=1}^{p} \mathbf{z}_k(\mathbf{x})}{p} \tag{3.47}$$

Where $\boldsymbol{z} \in \mathbb{R}^p$ is the objective vector formed by individually optimized objectives normalized by the maximum value of each objective, e.g., solutions *4*, *12*, and *1*, which are non-dominated, each minimizing one of the objectives.

The normalization process is necessary to disregard the dimensions of each objective. Evaluating the Euclidean distance in the objective space, from the barycenter to all candidates, it is possible to find the most balanced candidate. The function to be minimized, $\boldsymbol{Z}(\boldsymbol{x}^B) \in \mathbb{R}^p$, is given by Equation 3.48:

$$\mathbf{Z}(\mathbf{x}^B) = \min\left|\mathbf{Z}(\mathbf{x}) - \mathbf{Z}(\mathbf{x}^*)\right| = \min\left\{\sum_{k=1}^{p}\left[Z_k(\mathbf{x}) - Z_k(\mathbf{x}^*)\right]^2\right\}^{\frac{1}{2}} \tag{3.48}$$

3.5.3. Mutual Metric Method

Following, it is presented the generalized algorithm to apply the proposed *Mutual Metric Method* (MMM) for a generic multi-objective optimization problem. Differently from the *Smallest Loss Criterion* (SLC), MMM uses a common variable as metric instead of a normalization step.

Firstly, it must be found a common metric that can be measured in every simulation and is used to represent the objectives individually. Eventually, there will be more than one possible metric; in this case, it is up to the optimization designer to select a choice. Not necessarily the metric will be a geometric representation. This procedure is essential for the method.

Then, simulations individually optimizing each of the objectives are done. The task of optimizing each objective is not always simple and sometimes it requires many simulations since the parameters which the objectives depend on can be extremely sensitive or even partially unknown.

As mentioned, MMM does not use the normalization division as SLC does, although MMM utilizes an equivalent procedure to deal with the different nature of the objectives. Therefore, $\mathbf{z}$ now is the vector formed by individually optimized objectives given by Equation 3.49:

$$\begin{aligned}&\mathbf{z}_{kMMM}(\mathbf{x}) = \left[Z_a(\mathbf{x}), \min Z_k(\mathbf{x}), Z_b(\mathbf{x})\right]\\&a \in \mathbb{N} \mid 1 \leq a < k\\&b \in \mathbb{N} \mid k < b \leq p\end{aligned} \tag{3.49}$$

The selected mutual metric is collected after each simulation and serves as the vertices of a polygon. Mathematically, the definition of the function f responsible for providing the vertices of the mutual metric $\mathbf{v}_{\mathrm{k}}(\mathbf{x})$ based on the individual optimization is highly cumbersome, although it can be, relatively, easy to obtain it once we have at our disposal the powerful dynamics software simulator described previously. Then Equation 3.50 is:

$$\mathbf{v}_k(\mathbf{x}) = f(\mathbf{z}_{kMMM}(\mathbf{x})) \tag{3.50}$$

The barycenter of the polygon is calculated using Equation 3.51:

$$\mathbf{V}(\mathbf{x}^*) = \frac{\sum_{k=1}^{p} \mathbf{v}_k(\mathbf{x})}{p} \tag{3.51}$$

The simulation, mathematically represented by function h, must be able to pursue the calculated barycenter, generating a solution $\mathbf{Z}_{MMM}(\mathbf{x}^*)$. Then Equation 3.52 is:

$$\mathbf{Z}_{MMM}(\mathbf{x}^*) = h(\mathbf{V}(\mathbf{x}^*)) \tag{3.52}$$

For those objectives that cannot be directly determined by a simple scalar in the simulation, it is needed to find the Root Mean Square (RMS) value corresponding to those objectives, in every dimension (if applicable).

The RMS result for a discrete collection of *N* values is given by Equations 3.53, 3.54 and 3.55. It was done for each component of the vectors. Later, the norm of the RMS vector $\|a\|$ is obtained by Equation 3.56.

$$x_{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2} \tag{3.53}$$

$$y_{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} y_i^2} \tag{3.54}$$

$$z_{RMS} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} z_i^2} \quad (3.55)$$

$$\|a\| = \sqrt{x_{RMS}^2 + y_{RMS}^2 + z_{RMS}^2} \quad (3.56)$$

Eventually, the *Mutual Metric Method* solution will not coincide with the barycenter provided by SLC or mean value of the objectives. In this case, the vertices of the polygon can be corrected by a suitable factor defined by the solutions provided by the ratio between simulator and the mean value (SLC). New simulations are done using the vertices properly modified.

The procedure is repeated a determined number of times, generating a set of possible solutions. Finally, one can be interested in calculating the closest solution in the set to the mean value (SLC barycenter).

In general terms, the MMM approach can be applied to any multi-objective problem; however, in this work, it will be employed in berthing maneuvers.

3.5.4. An introduction to berthing applications

Inside the SAROS simulator, the block called RAS (Robot Attitude Simulator) is responsible, basically, for receiving a vector as input and providing the proper references for the satellite control system (roll, pitch, yaw). This block is also responsible for calculating the disturbances conveyed to the satellite dynamics block.

The vector that serves as input to RAS, $\mathbf{A}$, is composed of information about maneuver time, satellite attitude, satellite angular velocity, satellite angular acceleration, target position, gravitational acceleration, objective vertices, center of mass displacement, among other variables used for configuration proposes.

During multi-objective optimization simulations, the objective function to be minimized depending on its inputs can be understood as Equation 3.57.

$$\mathrm{R}(\mathbf{A}) = \min\left|\vec{V}_r - \vec{V}_c\right| \tag{3.57}$$

For each objective, the vectors whose distance must be minimized are properly defined. Deeper explanations will be provided with examples in the next chapter, but for a while, here it is possible to understand the objectives' assumptions.

A matrix of coefficients can be used to sum up the necessary vectors in two general formulas. Equations 3.58 and 3.59.

$$\begin{aligned}\vec{V}_r = a(1-b)(1-c)(1-d)\vec{J}_2 + b(1-a)(1-c)(1-d)\vec{J}_{2R} + \\ c(1-a)(1-b)(1-d)\vec{V}_{ST} + d(1-a)(1-b)(1-c)\vec{V}_{MMM}\end{aligned} \tag{3.58}$$

$$\vec{V}_c = e(1-f)\vec{J}_2 + f(1-e)\vec{V}_{SC} \tag{3.59}$$

The joint 2 position is a notable structural point, it is a symmetrical point out of the satellite body, which shares the satellite rotations once the distance between them is constant (link 0 length), and it is coincident with the manipulator workspace center. It will be the selected mutual metric.

The first objective is to minimize the satellite energy consumption. The satellite control system is turned off. This mode is called "Float" and it requires Equations 3.60, 3.61 and 3.62, where $\vec{J}_2$ is the vector from the satellite center of mass to the joint 2 position. The satellite rotates freely.

$$\left[a,b,c,d,e,f\right] = [1,0,0,0,0,0] \tag{3.60}$$

$$\vec{V}_r = \vec{J}_2 \tag{3.61}$$

$$\vec{V}_c = \vec{0} \tag{3.62}$$

For the second objective, minimizing satellite attitude motion, the output provided is the initial vector of references for joint 2 position, $\vec{J}_{2R}$. It is used Equations 3.63, 3.64 and 3.65. This mode is called “Angle”. The satellite maintains its initial attitude.

$$\left[a, b, c, d, e, f\right] = [0, 1, 0, 0, 0, 0] \tag{3.63}$$

$$\vec{V}_r = \vec{J}_{2R} \tag{3.64}$$

$$\vec{V}_c = \vec{0} \tag{3.65}$$

The objective of maximizing accuracy utilizes Equations 3.66, 3.67 and 3.68, where $\vec{V}_{ST}$ is the vector from the satellite center of mass to the target position. This mode is called “Core”. The satellite is moved to minimize the distance between the manipulator workspace center, joint 2 position, and the target point.

$$\left[a, b, c, d, e, f\right] = [0, 0, 1, 0, 1, 0] \tag{3.66}$$

$$\vec{V}_r = \vec{V}_{ST} \tag{3.67}$$

$$\vec{V}_c = \vec{J}_2 \tag{3.68}$$

Then the objective of minimizing manipulator energy consumption uses Equations 3.69, 3.70 and 3.71, where $\vec{V}_{SC}$ is the vector from the satellite center of mass to the manipulator claw. The manipulator control system is turned off. This mode is called “Claw robot off”. The satellite is moved to minimize the distance between the manipulator claw and the target point.

$$\left[a, b, c, d, e, f\right] = [0, 0, 1, 0, 0, 1] \tag{3.69}$$

$$\vec{V}_r = \vec{V}_{ST} \tag{3.70}$$

$$\vec{V}_c = \vec{V}_{SC} \tag{3.71}$$

For the objective of minimizing maneuver time, the vectors are defined exactly as for the previous objective, but this turn with the manipulator control system turned on. This mode is called “Claw robot on”.

Finally, for the MMM multi-objective optimization, it is considered Equations 3.72, 3.73 and 3.74. The elements of $\mathbf{V}(\mathbf{x}^{*})$ are converted into components of the vector $\vec{V}_{MMM}$. This mode is called “Center”. The satellite moves to minimize the distance between the joint 2 position and the barycenter obtained by the MMM.

$$\left[a,b,c,d,e,f\right]=[0,0,0,1,1,0] \tag{3.72}$$

$$\vec{V}_r=\vec{V}_{MMM} \tag{3.73}$$

$$\vec{V}_c=\vec{J}_2 \tag{3.74}$$

4 SIMULATIONS AND RESULTS

In complex systems, it can be difficult to obtain mathematical functions to describe and establish the objectives once such functions may have their dynamics coupled or even partially unknown decision variables. Simplifications could insert diverse errors and lead us to unreal results. Therefore, the approach presented here considers the relevance of model simulations based on parameters defined as well as possible.

In this chapter, the *Mutual Metric Method* (MMM) will be applied to the satellite and manipulator problem. The mentioned conflicting objectives are presented at each section with its respective simulation until the multi-objective optimization is addressed.

4.1. Multi-objective optimization simulations

Some enhancements were fulfilled in the simulator to make it able to test the multi-objective optimization scenarios. It was aimed to achieve five objectives at all, minimizing energy consumption for the satellite, minimizing energy consumption for the manipulator, minimizing satellite attitude motion, maximizing manipulator's accuracy, and minimizing the total maneuver time.

For the objectives of minimizing the energy consumption, the solution is basically limit it to zero by turning off the control system that requires such a resource. This is the procedure done for both, manipulator and satellite, control systems separately.

A similar procedure is taken to minimize the satellite attitude motion, it is basically done by defining the initial satellite attitude as its control system reference.

To maximize the accuracy, the target point is achieved as close to the workspace sphere center as possible. Finally, to minimize the maneuver time, both control systems are defined to work simultaneously and as fast as possible, given their features, to achieve the target point.

More detailed explanations about these procedures are presented in the sections ahead for each simulation.

In the multi-objective simulations, it was defined the following parameters as presented by Table 4.1. It is considered that the robotic manipulator wrist reaches the target point inside its workspace, if the distance between them is smaller than *0.1* meters.

Table 4.1 - Parameters for multi-objective simulations.

Parameter	Value
Initial Angular Position (deg)	[0; 0; 0]
Initial Angular Velocity (deg/s)	[0; 0; 0]
Initial Angular Acceleration (deg/s^2)	[0; 0; 0]
Limitless Simulation Time	On
Fixed Step Size (s)	0.15
Gravitational Acceleration (m/s^2)	[0; 0; 0]
Inertial Vector Target (inside workspace) (m)	[-0.2; 1.4; 1.7]
Initial Manipulator's Joint Variables (deg)	[0;0;0]
Gains [Proportional; Integral; Derivative]	[200; 0.0002; 1000]

Manipulator's and satellite's control systems are simultaneously activated in all simulations, except when it is desired to minimize the respective energy consumption, when the proper control is turned off. It means that base satellite attitude is subject to occasional movements of the robot, when satellite control system is turned off, and the manipulator holds its initial joint variables, when robot control system is turned off.

Once the robotic manipulator, by means of a claw on its wrist, reaches the defined target point, we register the time required and the joint 2 inertial position. This position is particularly important, because we are going to assume it as the spatial representation for all objectives of our simulations and later the collection of these points will be used to apply a multi-objective optimization technique.

The joint 2 position was selected as the point for comparison because of its structural symmetry. Theoretically, any point in the chaser body could be chosen, excepting the appendage (arm) parts whose distance to the satellite (base) center may change since such parts are subject to different dynamics and control. Nevertheless, the joint 2 position presents an easier visualization.

4.1.1. Simulation 1 (minimizing satellite energy consumption)

We are interested in minimizing the energy consumption from a source from which the satellite control system depends on, for example a fuel that powers some actuators. For this purpose, it is executed a simulation where the satellite has its control system continuously turned off. Just the manipulator performs its trajectory to achieve the aimed point, the inertial vector target inside its workspace. Performed in this way, the maneuver is said to be in free-floating mode. Figure 4.1 and Figure 4.2 show as the attitude control system of the satellite and the robotic manipulator’s joints have been, respectively, activated.

Figure 4.1 - Actuators torque (simulation 1).

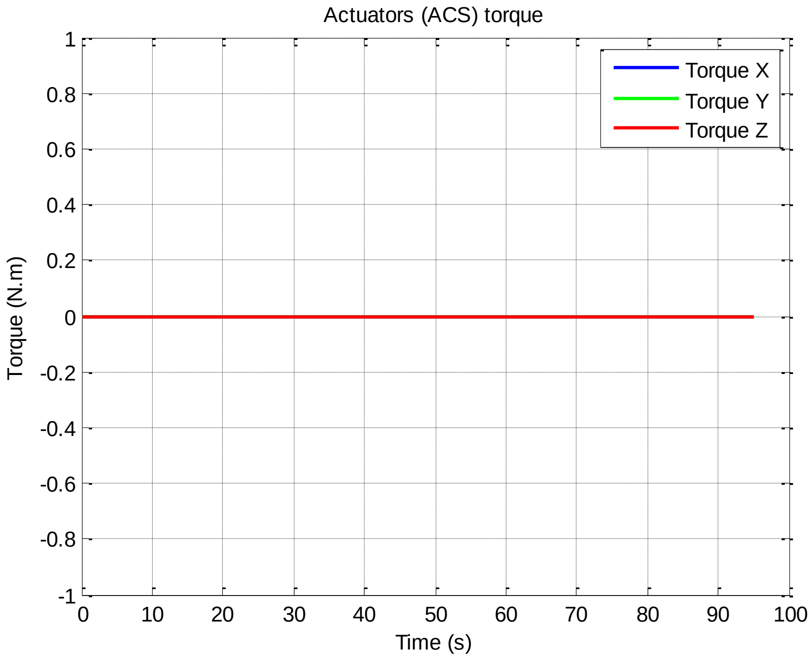


Figure 4.2 - Joint angular positions (simulation 1).

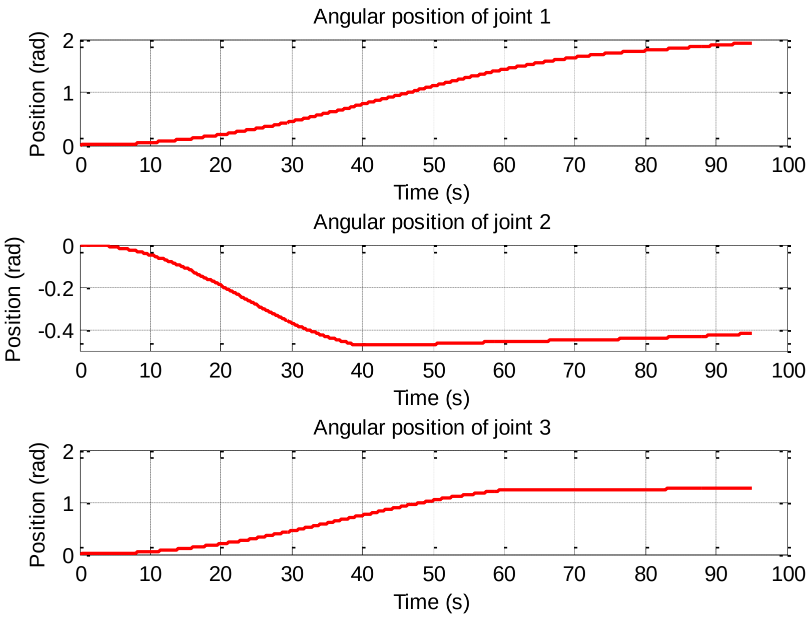


Figure 4.3 and Figure 4.4 show the final pose, defined as position and orientation of an object, in the satellite frame and in the inertial frame respectively.

Figure 4.3 - SAROS in satellite frame (simulation 1).

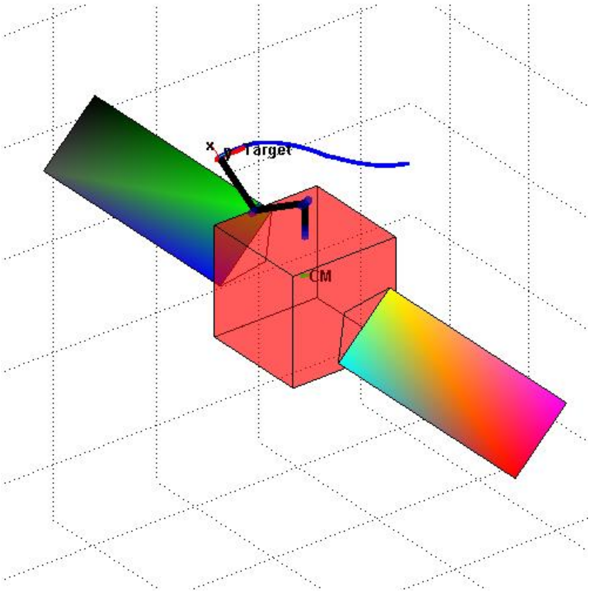


Figure 4.4 - SAROS in inertial frame (simulation 1).

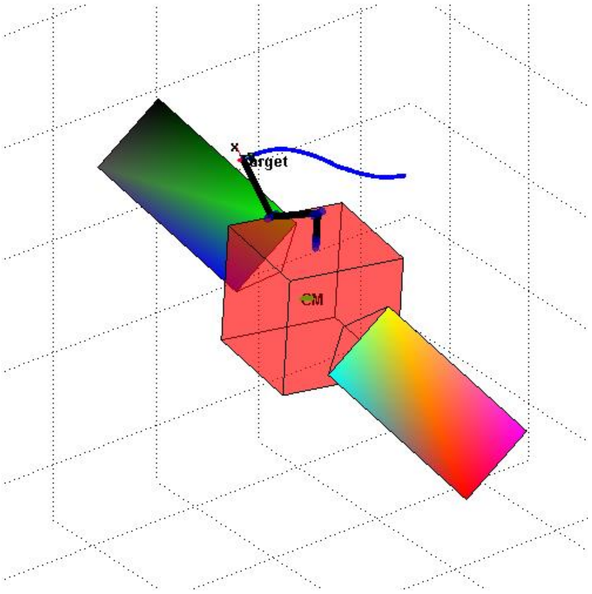


Figure 4.5 shows how the distance from the wrist to the target point, distance error, lessened along time.

Figure 4.5 - Distance error to the target (simulation 1).

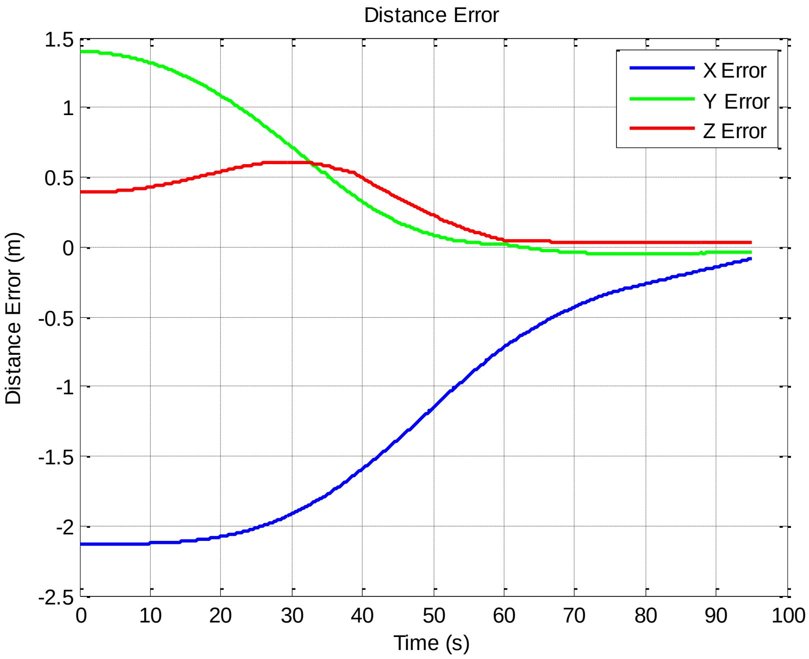
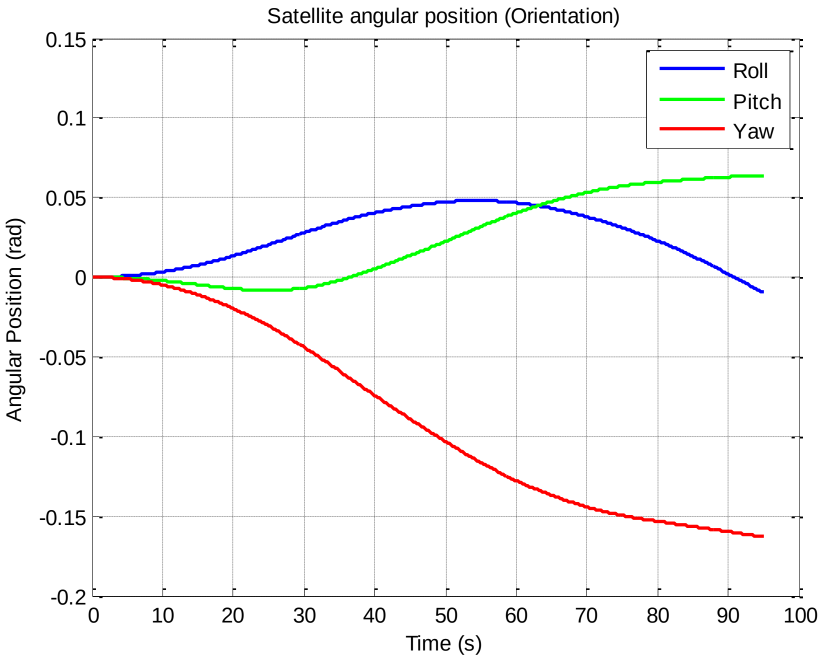


Figure 4.6 shows how the base satellite had its orientation changed along time due to manipulator's movements.

Figure 4.6 - Satellite orientation (simulation 1).



The software provided the precise moment when the target was reached, at *94.95* seconds of simulation, and at this moment the joint 2 was in the inertial position represented by the vector *[0.1737;-0.0959;1.3005]* in meters.

4.1.2. Simulation 2 (minimizing satellite attitude motion)

Our second objective is to minimize the attitude motion of the satellite. This might be desirable in situations where the slosh dynamics can be harmful, for example. Slosh refers to the movement of a fluid inside a body (AGOSTINHO, 2019; CARNEIRO JUNIOR, 2017). Examples include propellant in spacecraft tanks. It can be especially dangerous when dealing with liquids in microgravity because it can alter the mass distribution and center of mass of the satellite. Propellant slosh can introduce uncertainties in the attitude or cause problematic interaction with the attitude control system. Another good reason to be interested in maintaining the satellite attitude is for safety of eventual experiments being conducted inside the satellite.

In this simulation, it was turned on the attitude control system and provided as reference *0* degrees in roll, pitch and yaw, same from beginning to end of the simulation. In this configuration, the manipulator fulfills the berthing maneuver taking its wrist to the target point in a rotation free-flying mode.

To verify that the target point was reached, note Figure 4.7, which shows how the distance decreased along the simulation.

Figure 4.8 and Figure 4.9 show the final pose in the satellite frame and in the inertial frame, respectively.

Figure 4.10 and Figure 4.11 show the torques exerted by the attitude control system and by the robotic manipulator and we can realize a sort of symmetry between them since the ACS had to react to those torques generated by the robot, such torques would disturb the satellite attitude, but the satellite control succeed in maintaining the attitude.

Figure 4.7 - Distance error to the target (simulation 2).

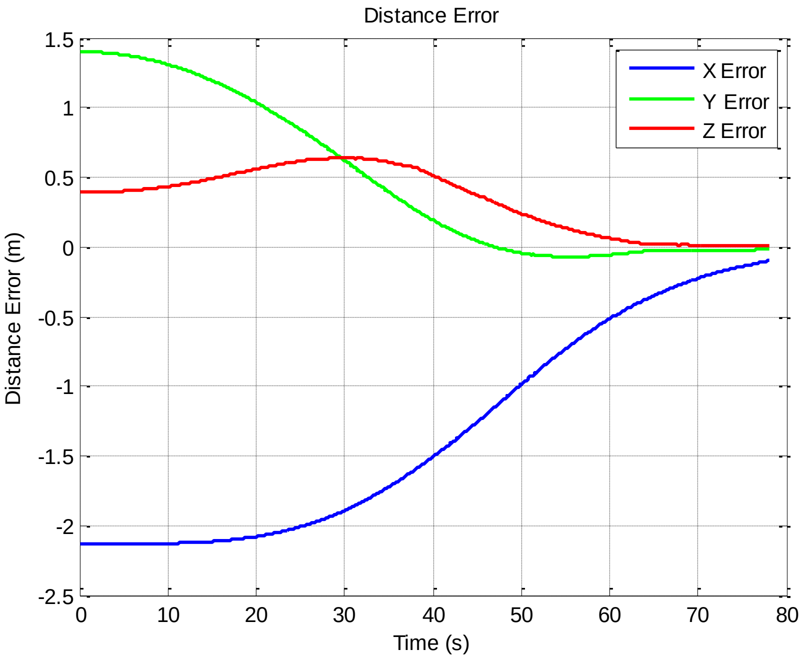


Figure 4.8 - SAROS in satellite frame (simulation 2).

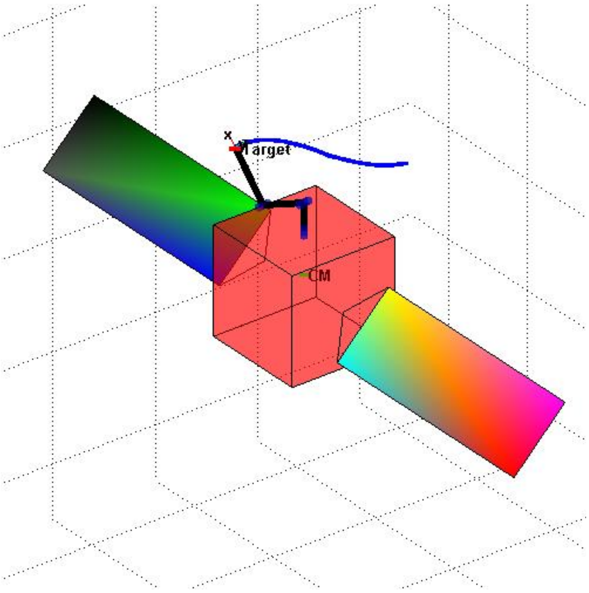


Figure 4.9 - SAROS in inertial frame (simulation 2).

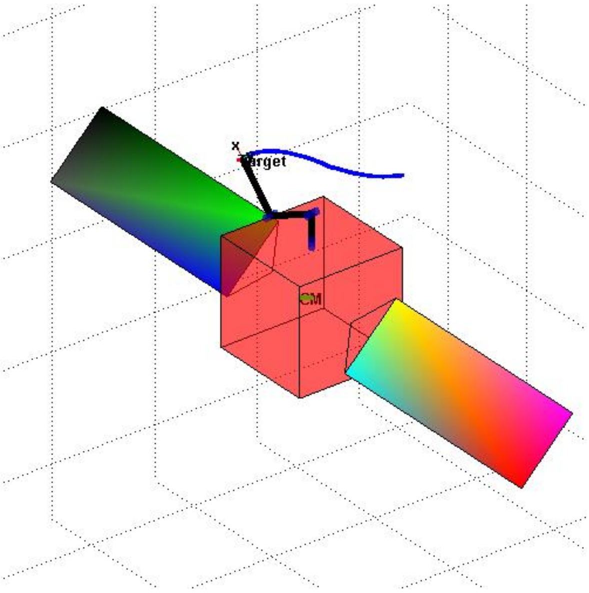


Figure 4.10 - Actuators torque (simulation 2).

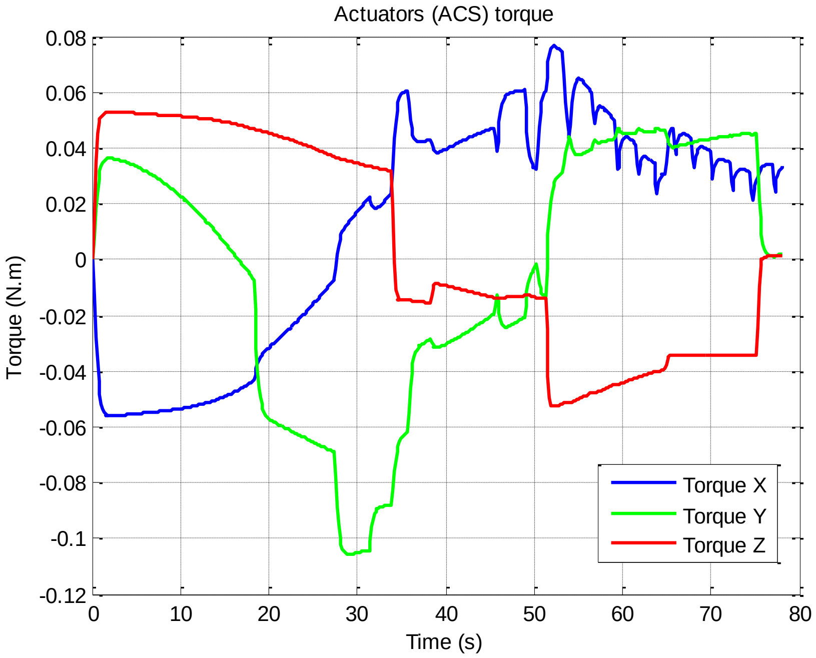
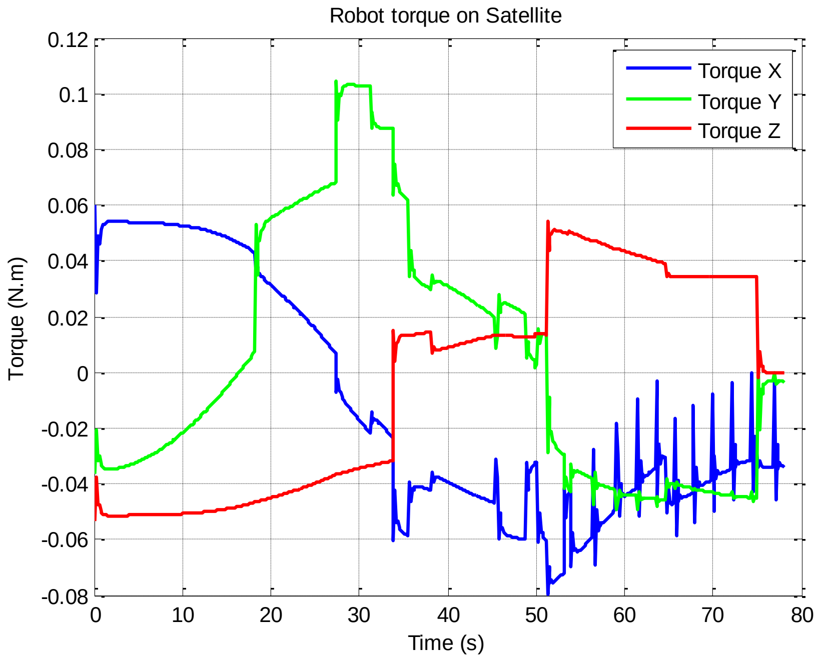


Figure 4.11 - Robot torque on satellite (simulation 2).



Figures 4.12, 4.13 and 4.14 show the attitude control system behavior to pursue its given reference during the simulation in roll, pitch and yaw, respectively.

Figure 4.12 - Angle in roll (simulation 2).

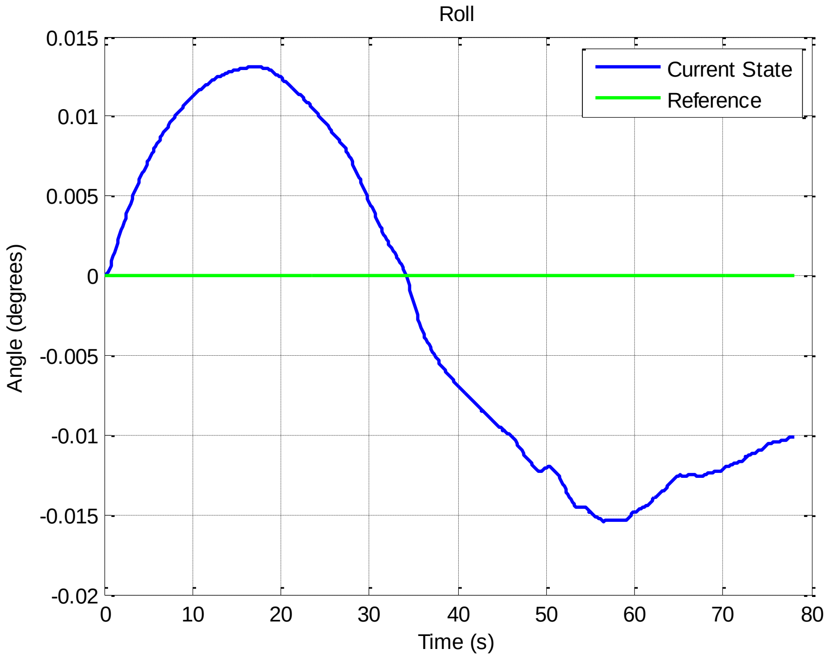


Figure 4.13 - Angle in pitch (simulation 2).

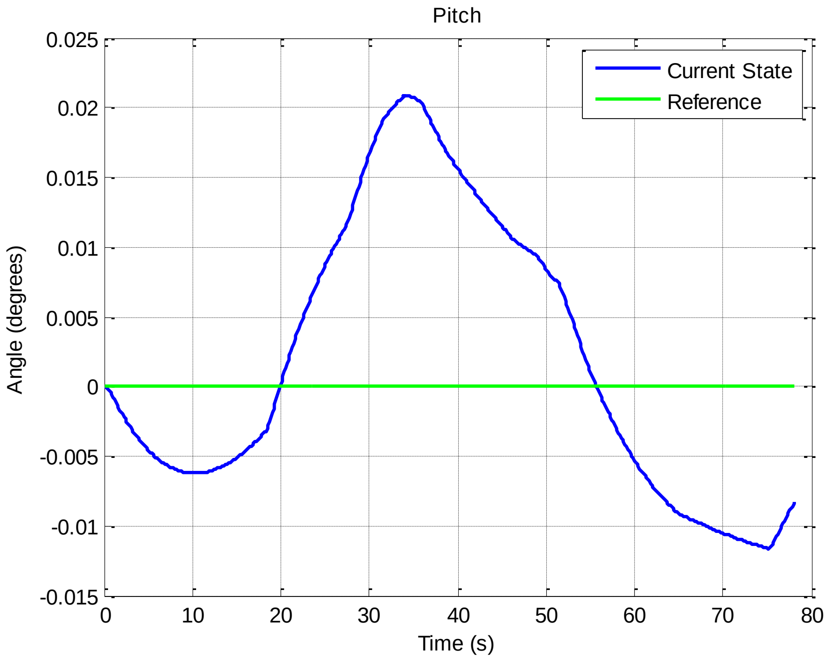


Figure 4.14 - Angle in yaw (simulation 2).

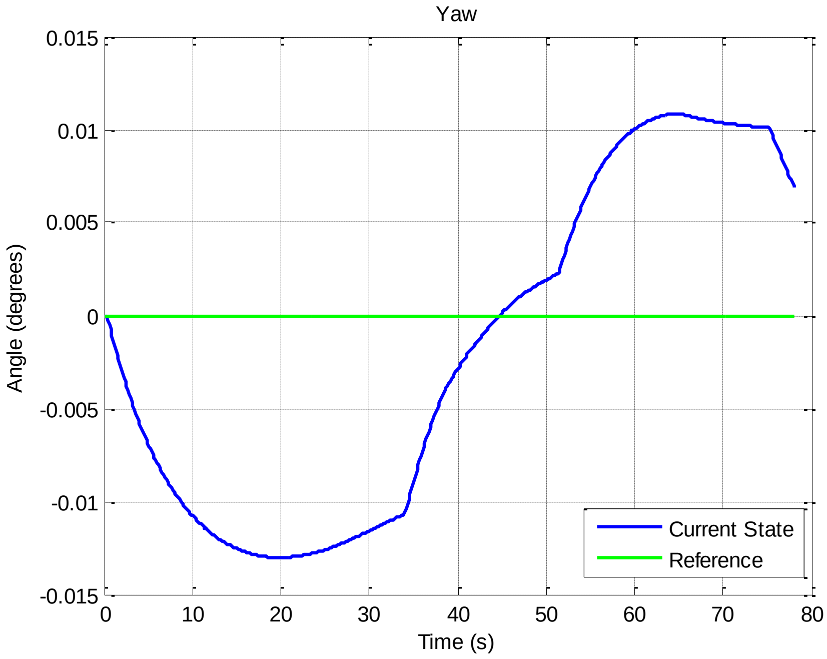
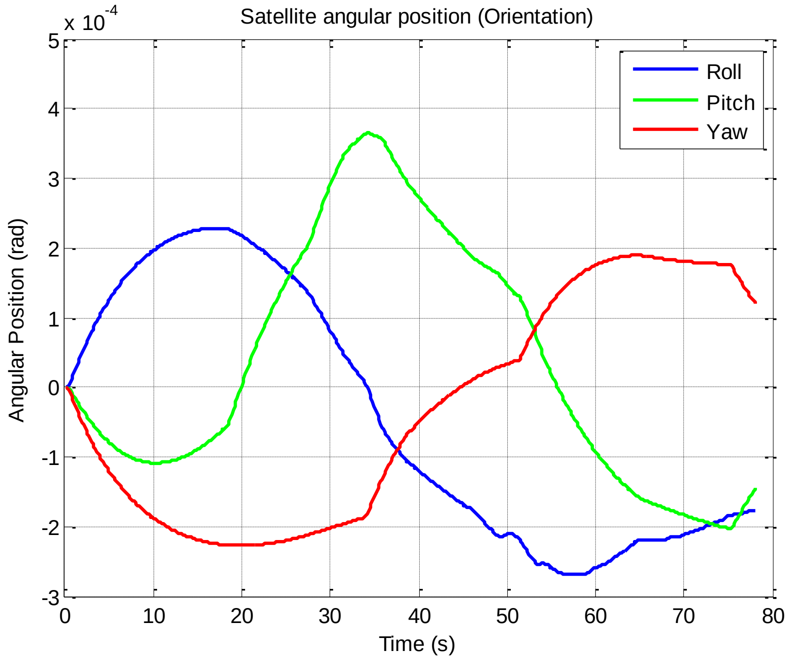


Figure 4.15 shows how the satellite orientation barely changes along time. See its magnitude.

Figure 4.15 - Satellite orientation (simulation 2).



According to the software response, it took *78.15* seconds to the manipulator achieves the target point. At this time, the joint 2 was at a position in the inertial frame of *[0.0784;-0.1051;1.3247]* meters.

4.1.3. Simulation 3 (maximizing accuracy)

For the third objective, we are interested in maximizing the accuracy. We can do it commanding the attitude control system to maneuver the satellite, so that the target point be as close as possible to the center of the sphere that defines the manipulator's workspace, i.e., this workspace center is located at the joint 2. It is as if we had rotated the satellite for the manipulator to find the target next to its workspace center. Figure 4.16 proves that the target point was achieved, while Figure 4.17 presents the satellite orientation changes along time.

Figure 4.18 and Figure 4.19 show the final pose in the satellite frame and in the inertial frame, respectively. The dashed line represents an axis that coincides with the link 1 and, in this case, aims to match the target point (red point). In blue, we see the robot’s wrist trajectory, while the red line represents the target trajectory (in satellite frame).

Figure 4.16 - Distance error to the target (simulation 3).

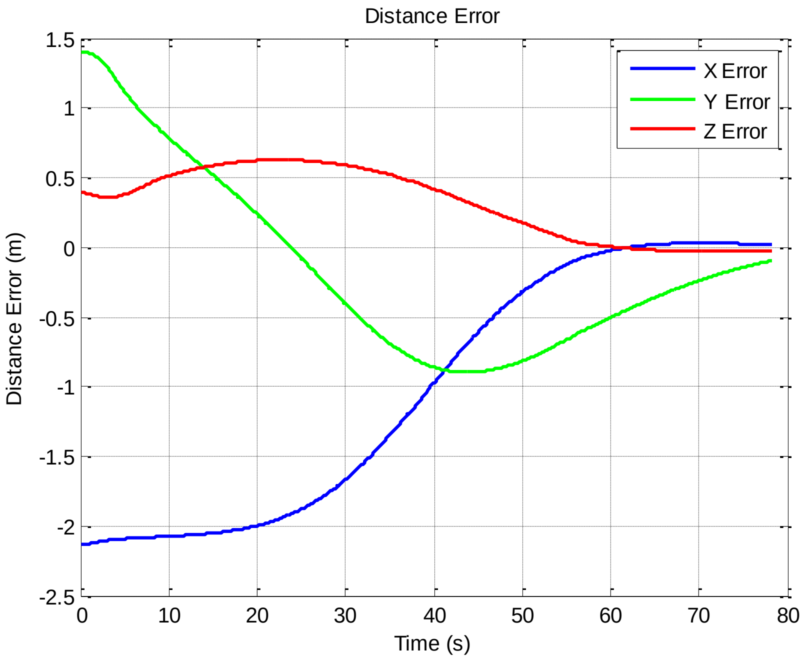


Figure 4.17 - Satellite orientation (simulation 3).

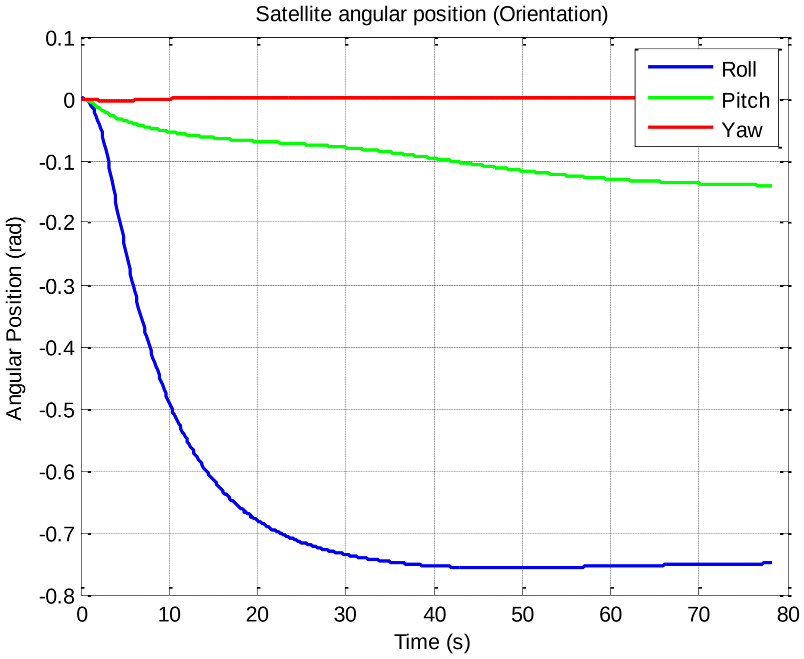


Figure 4.18 - SAROS in satellite frame (simulation 3).

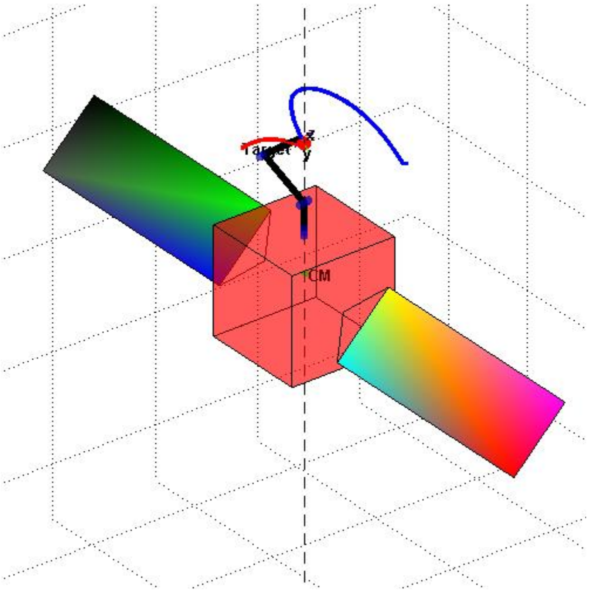
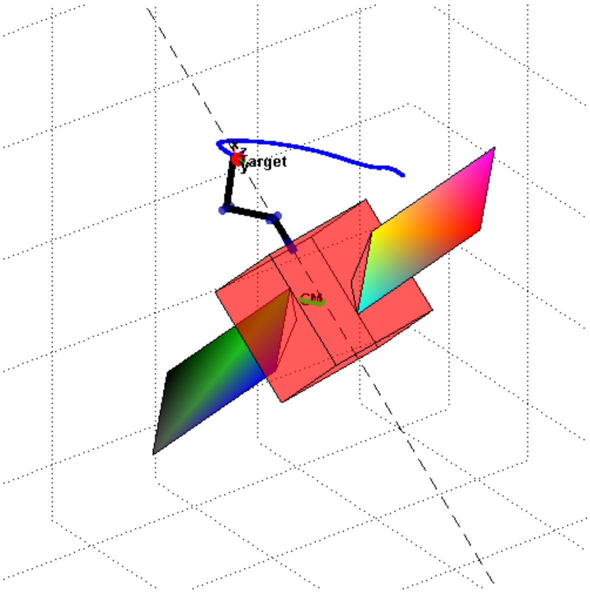


Figure 4.19 - SAROS in inertial frame (simulation 3).



Figures 4.20, 4.21 and 4.22 show the attitude control system behavior to pursue its given reference during the simulation in roll, pitch and yaw, respectively.

Figure 4.20 - Angle in roll (simulation 3).

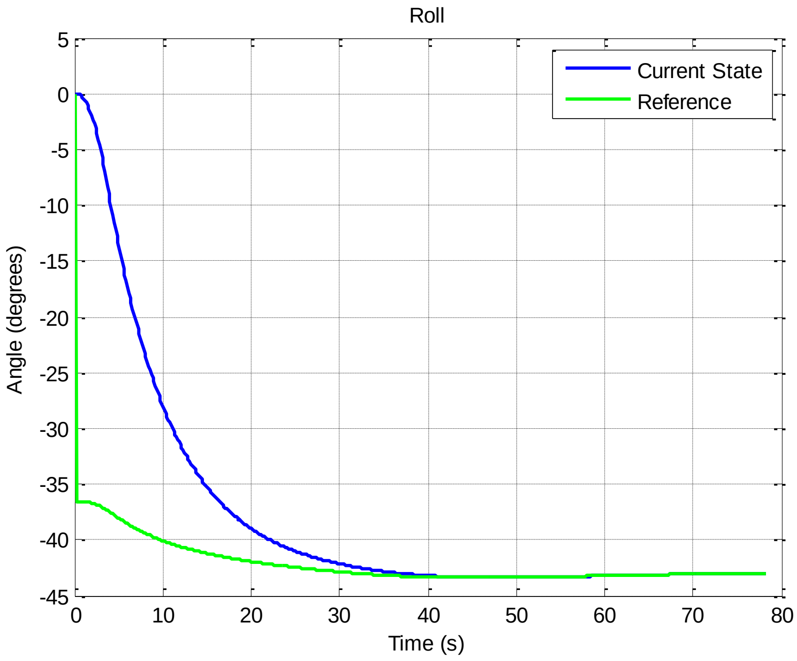


Figure 4.21 - Angle in pitch (simulation 3).

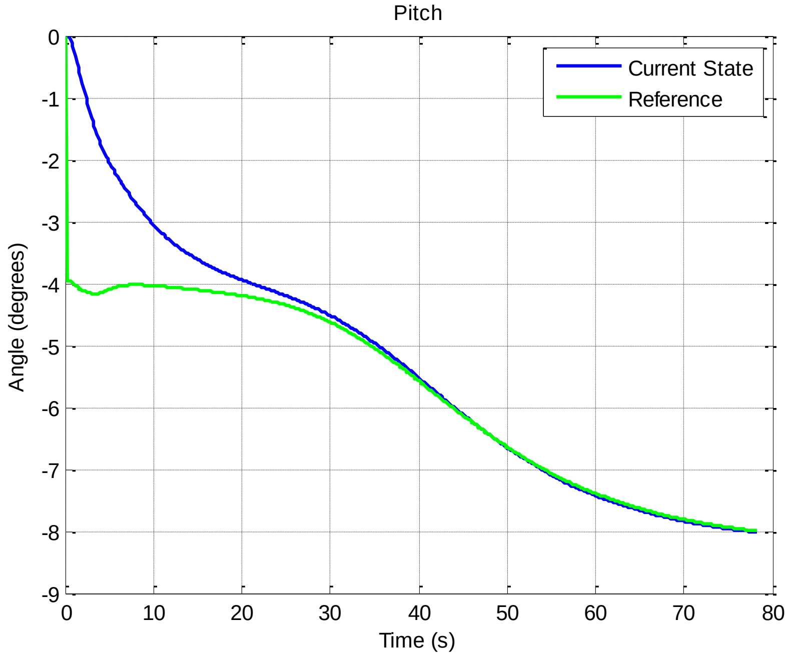
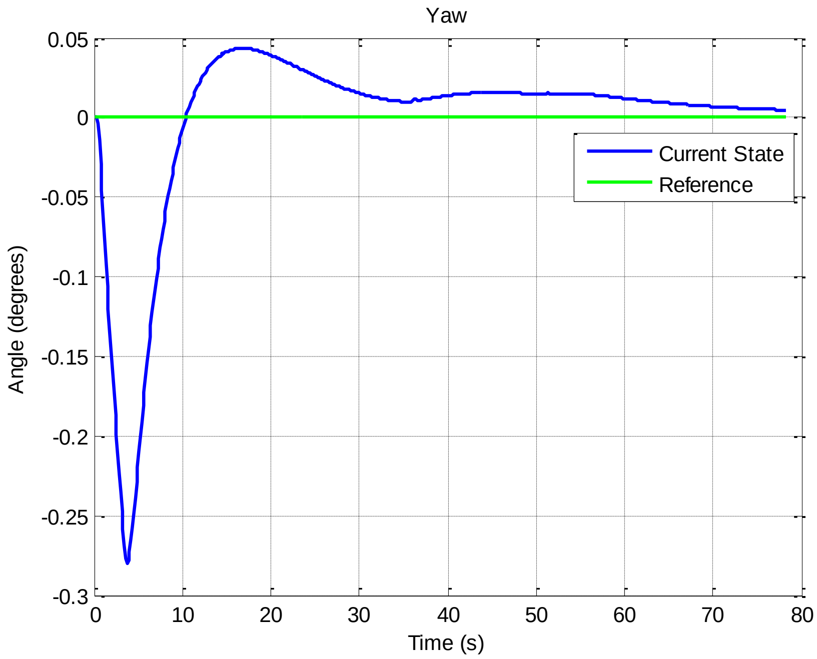


Figure 4.22 - Angle in yaw (simulation 3).



According to the software response, it was necessary *78.30* seconds for the manipulator reaches the target point with its joint 2 at *[-0.0648;0.6880;0.9410]* meters in the inertial frame.

4.1.4. Simulation 4 (minimizing manipulator energy consumption)

For the fourth objective, we wanted to minimize the energy consumption of a source that powers the robotic manipulator, for example a battery. To achieve such a requirement, we can minimize the manipulator action. The manipulator has its control system continuously turned off and we command the ACS to maneuver the satellite, so that the target point be as close as possible to the wrist initial position, i.e., it is as if we had rotated the satellite for the manipulator to find the target at a point that coincides with its wrist initial position.

Figure 4.23 proves that the target point was achieved, while Figure 4.24 presents the satellite orientation changes along time.

Figure 4.23 - Distance error to the target (simulation 4).

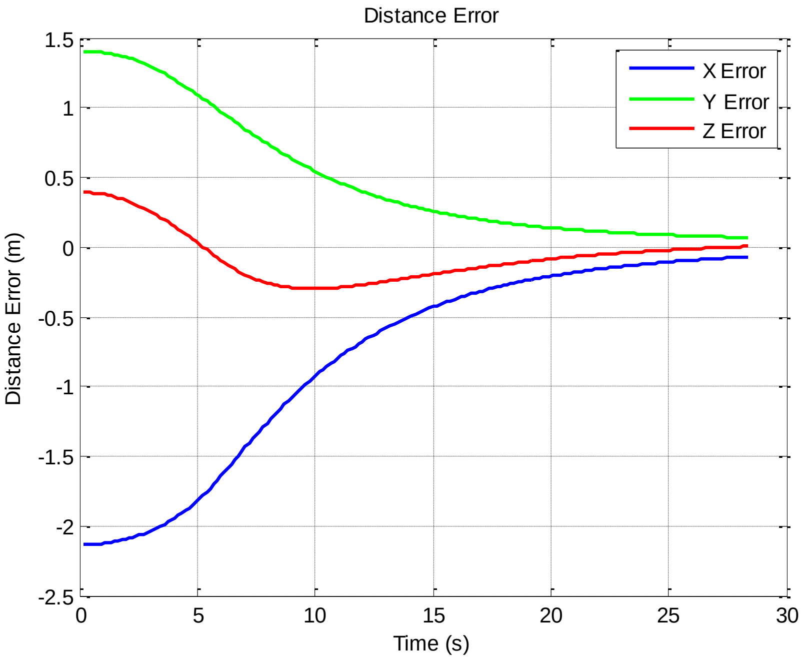


Figure 4.24 - Satellite orientation (simulation 4).

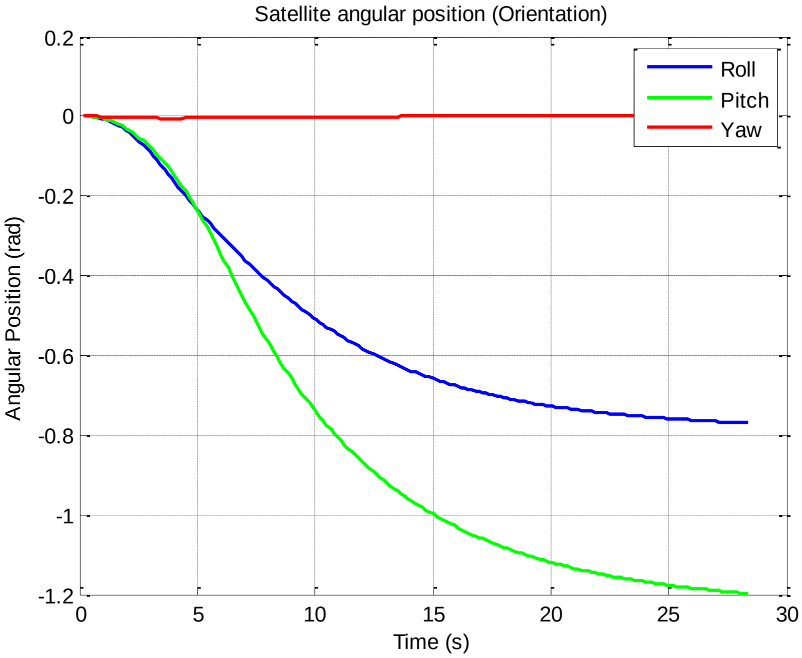


Figure 4.25 and Figure 4.26 show the final pose in the satellite frame and in the inertial frame, respectively. The dashed line represents an axis that contains the

satellite's center and the wrist initial position and, in this case, aims to match the target point (red point).

Figure 4.25 - SAROS in satellite frame (simulation 4).

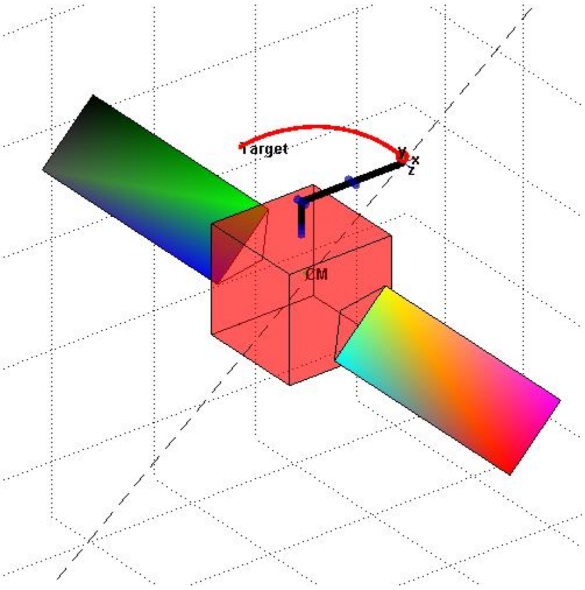


Figure 4.26 - SAROS in inertial frame (simulation 4).

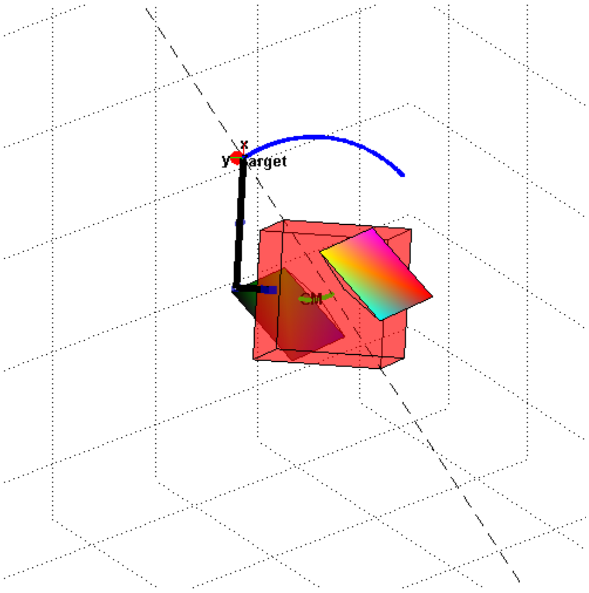


Figure 4.27 and Figure 4.28 show as the attitude control system of the satellite and the robotic manipulator's joints have been, respectively, activated.

Figure 4.27 - Actuators torque (simulation 4).

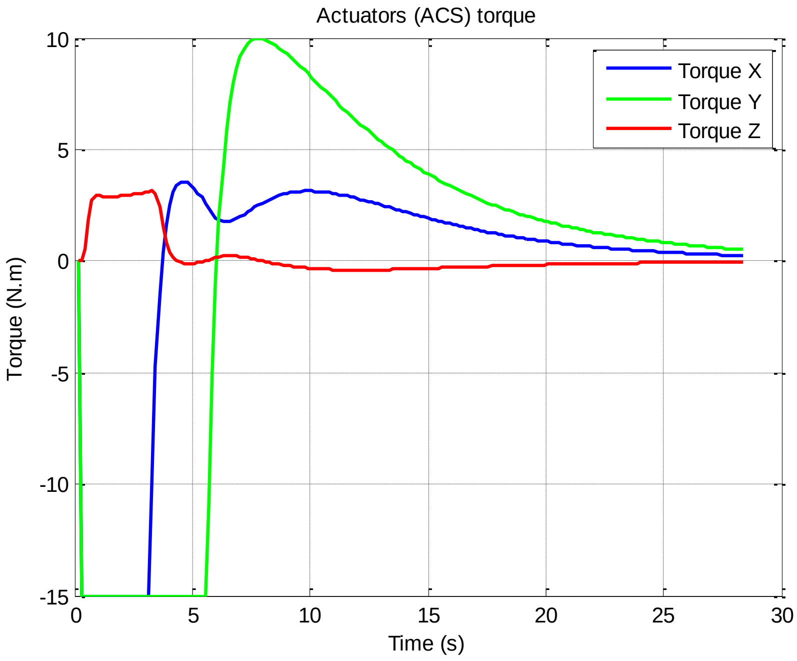
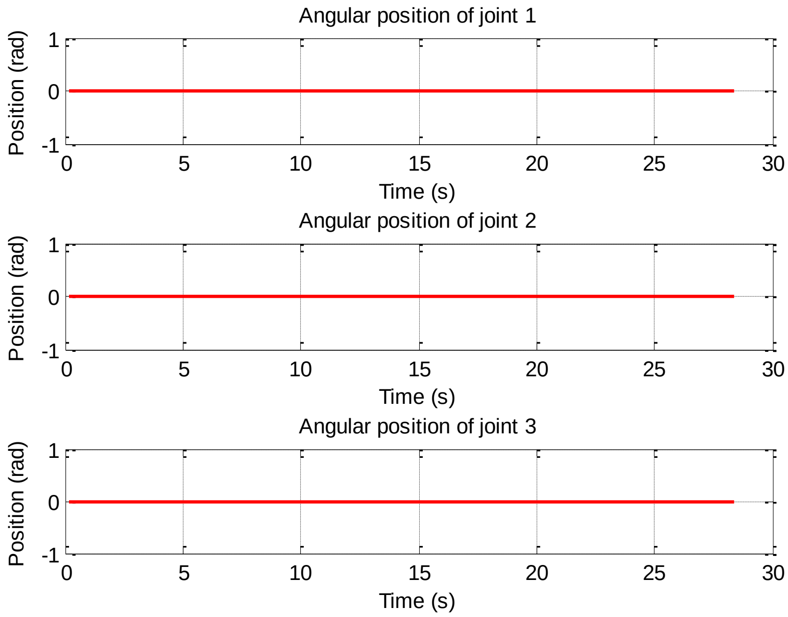


Figure 4.28 - Joint angular positions (simulation 4).



Figures 4.29, 4.30 and 4.31 show the attitude control system behavior to pursue its given reference during the simulation in roll, pitch and yaw, respectively.

Figure 4.29 - Angle in roll (simulation 4).

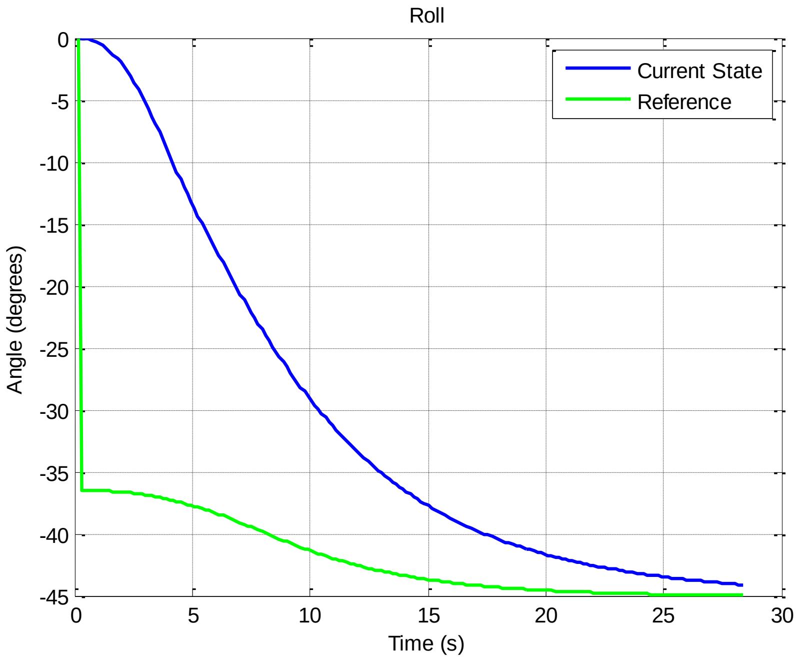


Figure 4.30 - Angle in pitch (simulation 4).

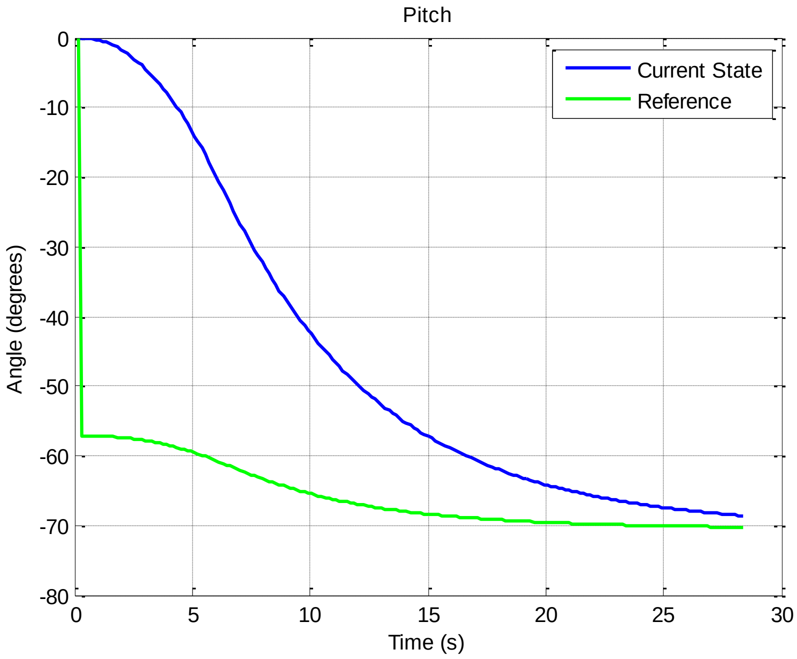
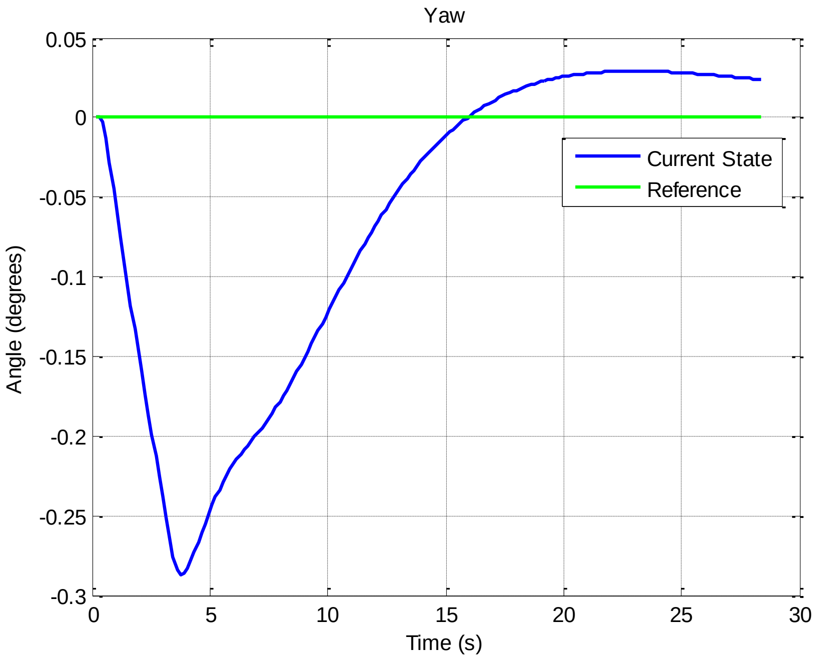


Figure 4.31 - Angle in yaw (simulation 4).



According to the software response, it was necessary *28.35* seconds for the manipulator reaches the target point with its joint 2 at *[-0.9378;0.5013;0.0828]* meters in the inertial frame.

4.1.5. Simulation 5 (minimizing maneuver time)

For the fifth objective, we wanted, for example, to minimize the maneuver time. To achieve such a requirement, we make satellite's and robot's control systems to pursue the target position as fast as they can. The velocity limits for each of them is defined based on the characteristics of the configured model, especially the acceleration ramps and control parameters, see Nardin (2015). We command the ACS to maneuver the satellite, so that the target point be as close as possible to the wrist current position while the manipulator moves toward the target point, i.e., it is as if we had rotated the satellite for the robot to find the target at a point that coincides with its wrist current position.

Figure 4.32 proves that the target point was achieved, while Figure 4.33 presents the satellite orientation changes along time.

Figure 4.32 - Distance error to the target (simulation 5).

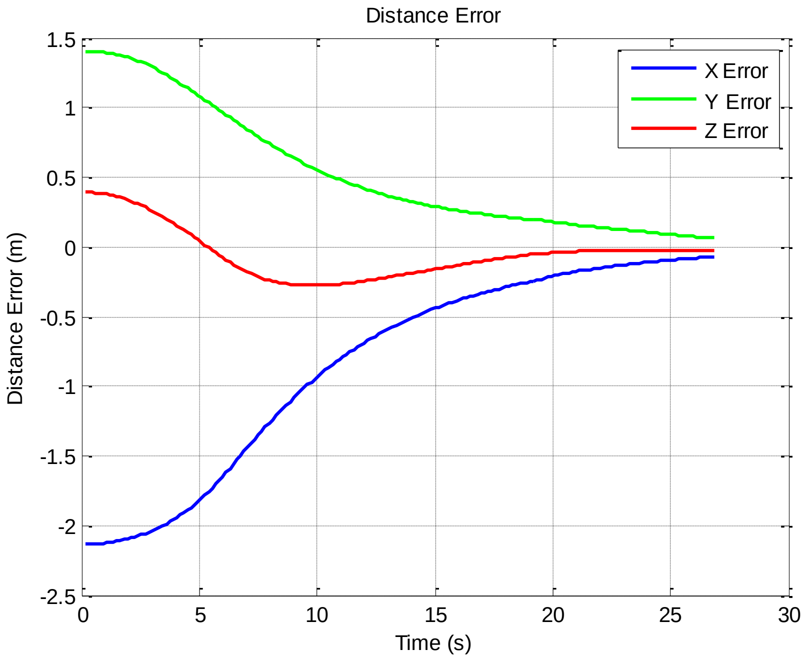


Figure 4.33 - Satellite orientation (simulation 5).

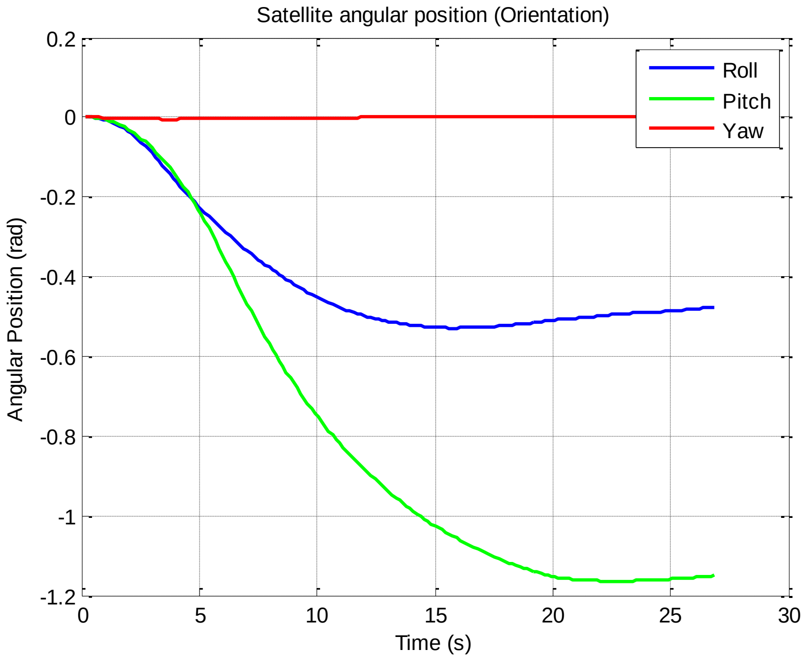


Figure 4.34 and Figure 4.35 show the final pose in the satellite frame and in the inertial frame, respectively. The dashed line represents an axis that contains the

satellite's center and the wrist current position and, in this case, aims to match the target point (red point).

Figure 4.34 - SAROS in satellite frame (simulation 5).

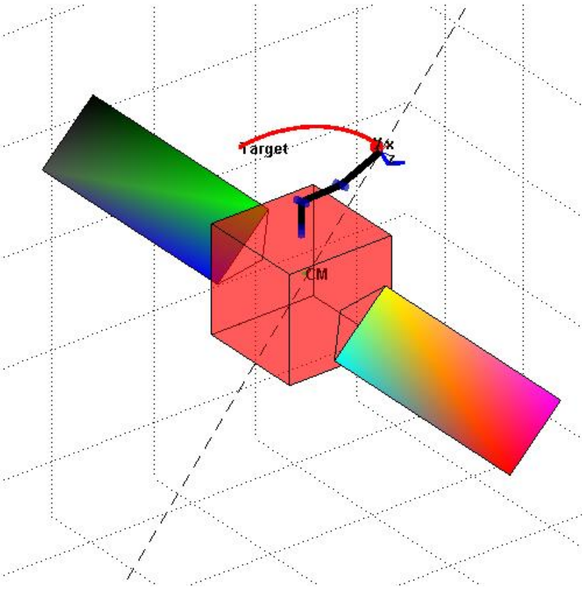
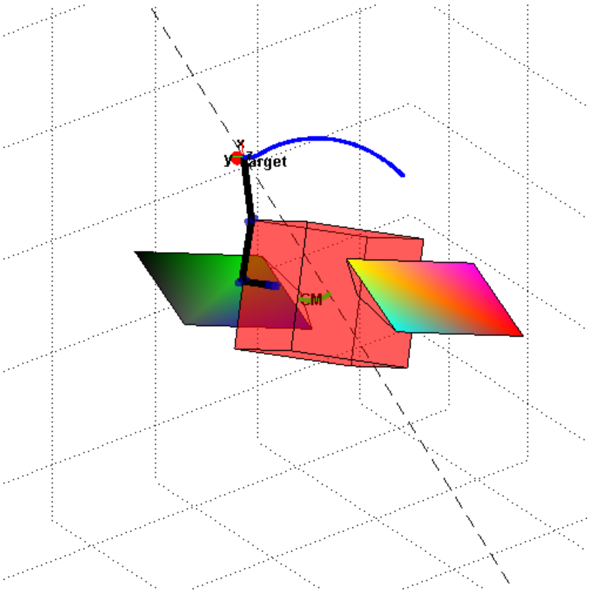


Figure 4.35 - SAROS in inertial frame (simulation 5).



Figures 4.36, 4.37 and 4.38 show the attitude control system behavior to pursue its given reference during the simulation in roll, pitch and yaw, respectively.

Figure 4.36 - Angle in roll (simulation 5).

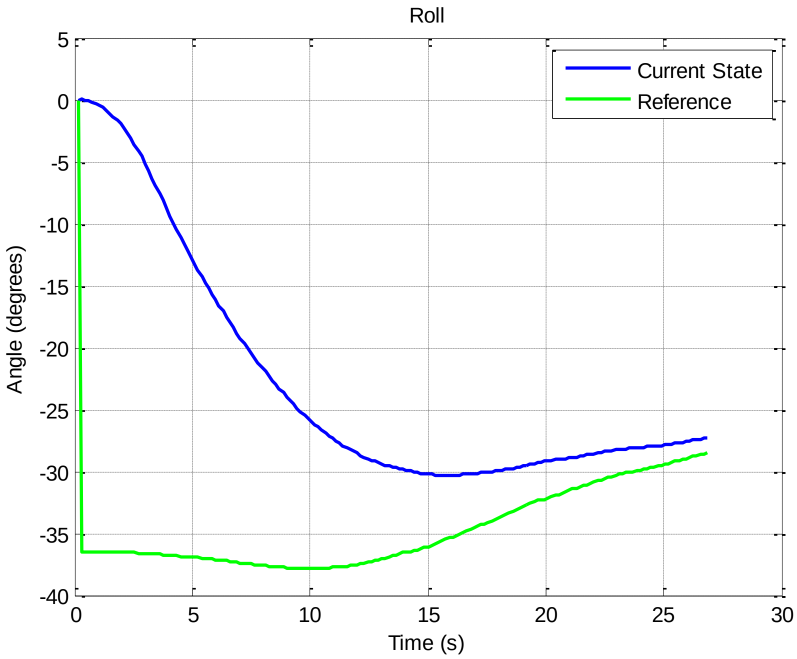


Figure 4.37 - Angle in pitch (simulation 5).

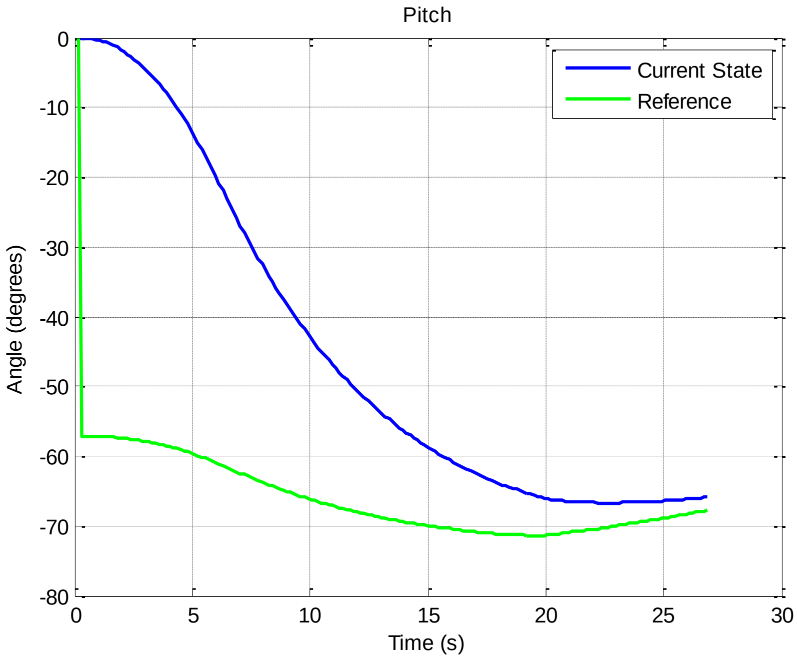
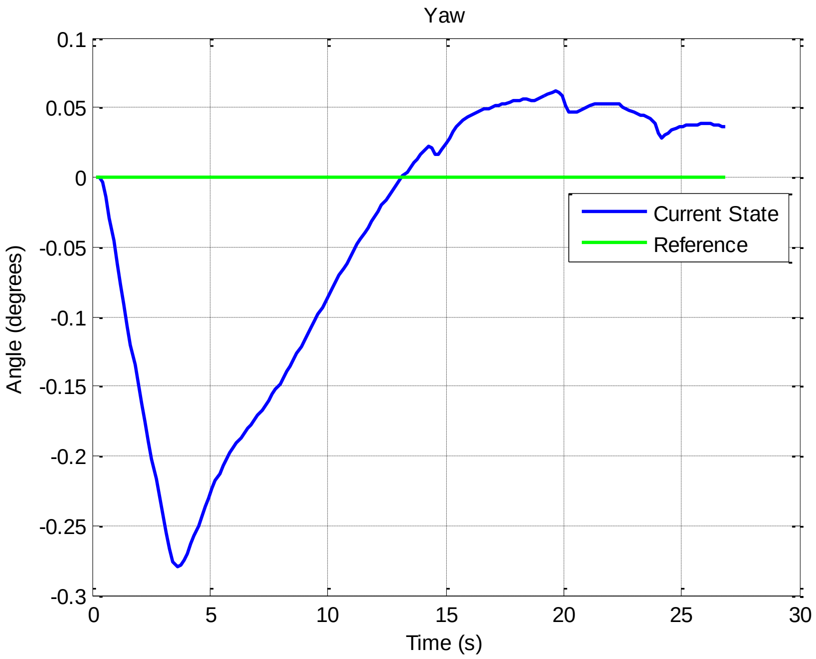


Figure 4.38 - Angle in yaw (simulation 5).



According to the software response, it was necessary *26.85* seconds for the manipulator reaches the target point with its joint 2 at *[-1.0141;0.2588;0.3433]* meters in the inertial frame. Indeed, as we could predict, this was the fastest simulation, slightly faster than the previous simulation mainly due to the manipulator joints' velocities.

4.1.6. Simulation 6 (MMM multi-objective optimization)

In this simulation, we wanted to apply the proposed multi-objective optimization approach. Firstly, we feed the algorithm with all five previous results of joint 2 position that represent individually each of our five distinct objectives in the three-dimensional space. Then, we command the ACS to maneuver the satellite, so that the balanced solution be as close as possible to the joint 2.

Figure 4.39 and Figure 4.40 show, in the inertial frame, simulation results considering three (triangle) and four (tetrahedron) arbitrary objectives, respectively. The simulator allows up to five objectives.

Figure 4.39 - SAROS in inertial frame with three objectives (simulation 6).

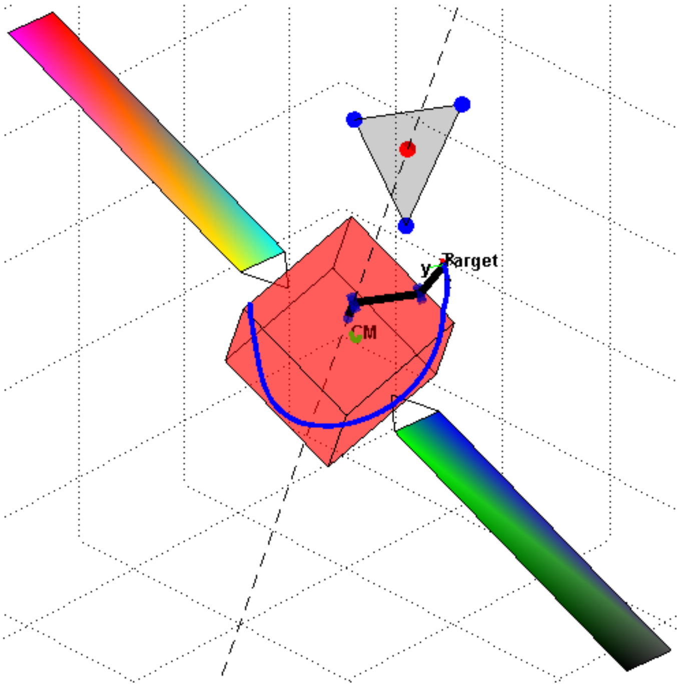
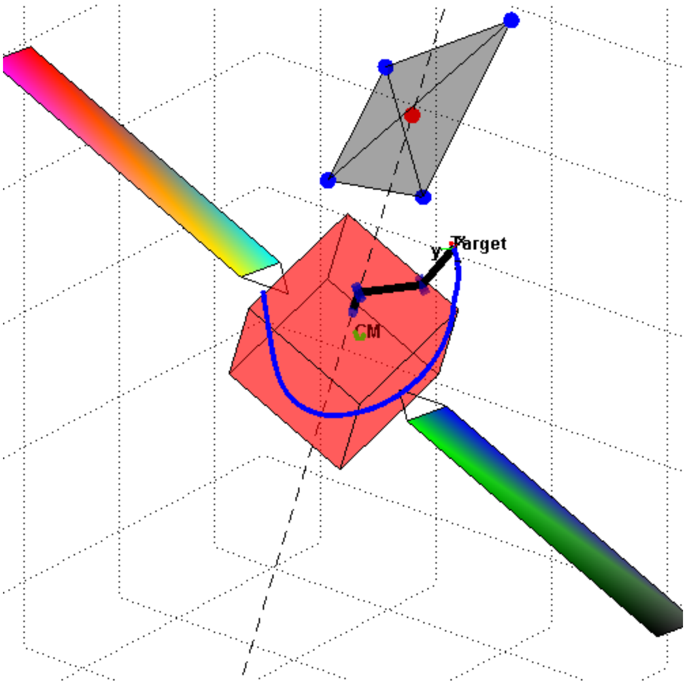


Figure 4.40 - SAROS in inertial frame with four objectives (simulation 6).



Now considering all five objectives, Figure 4.41 proves that the target point was achieved, while Figure 4.42 presents the satellite orientation changes along time.

Figure 4.41 - Distance error to the target (simulation 6).

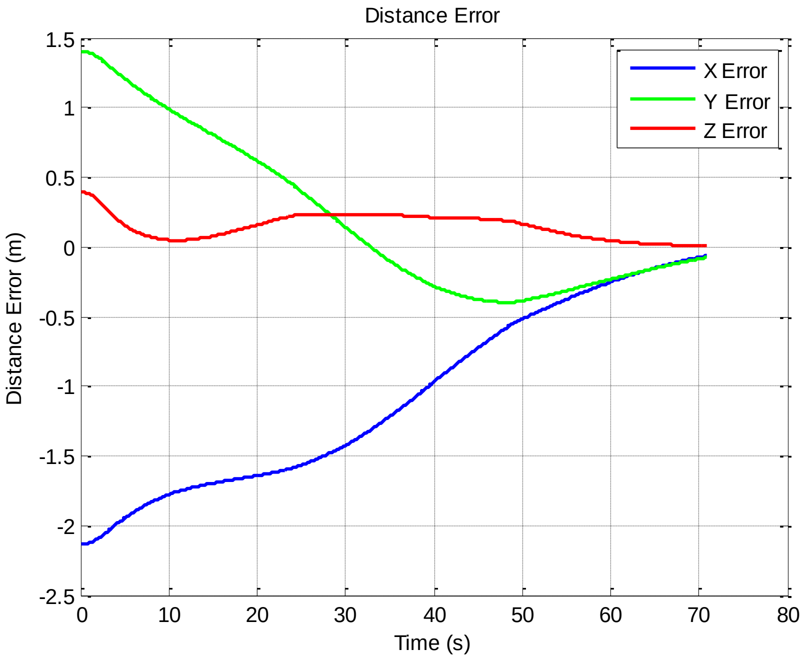


Figure 4.42 - Satellite orientation (simulation 6).

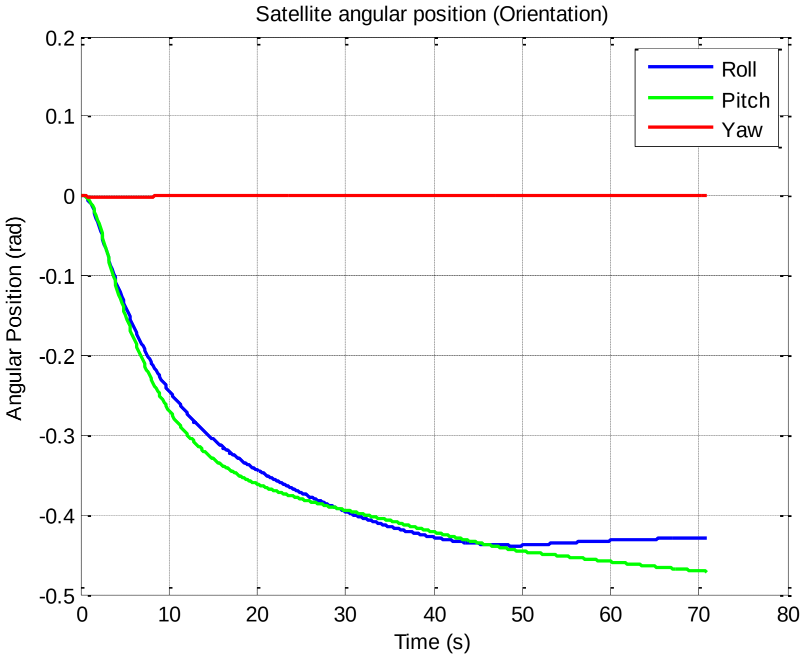


Figure 4.43 and Figure 4.44 show the final pose in the satellite frame and in the inertial frame, respectively. The dashed line represents an axis that coincides

with link 1 and, in this case, aims to match the balanced solution among all objectives (red point).

Figure 4.43 - SAROS in satellite frame with five objectives (simulation 6).

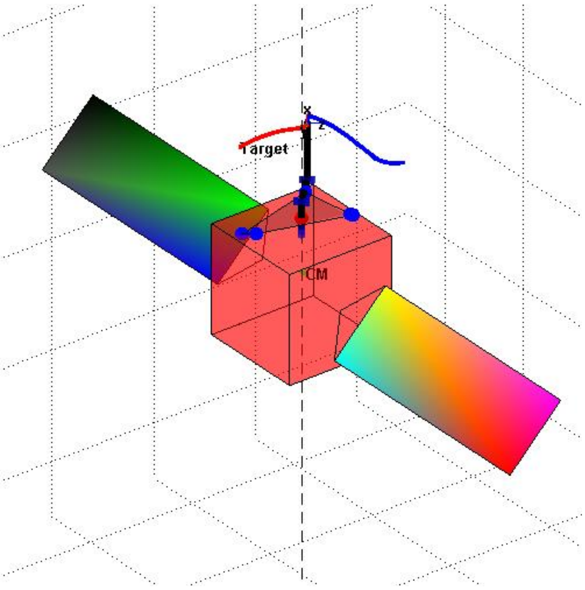
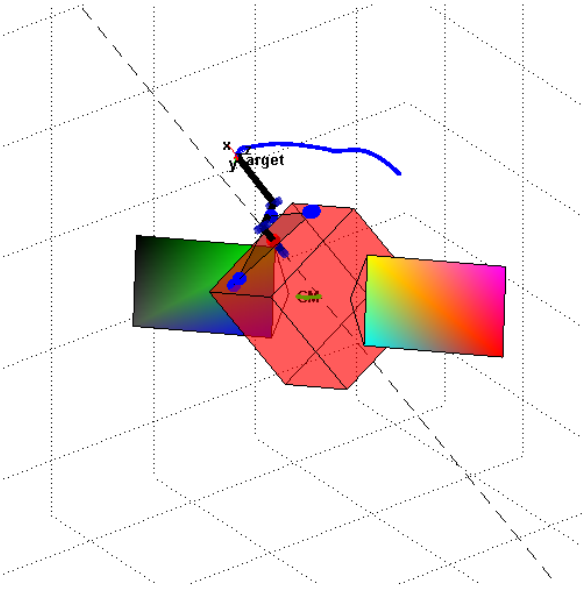


Figure 4.44 - SAROS in inertial frame with five objectives (simulation 6).



Figures 4.45, 4.46 and 4.47 show the attitude control system behavior to pursue its given reference during the simulation in roll, pitch and yaw, respectively.

Figure 4.45 - Angle in roll (simulation 6).

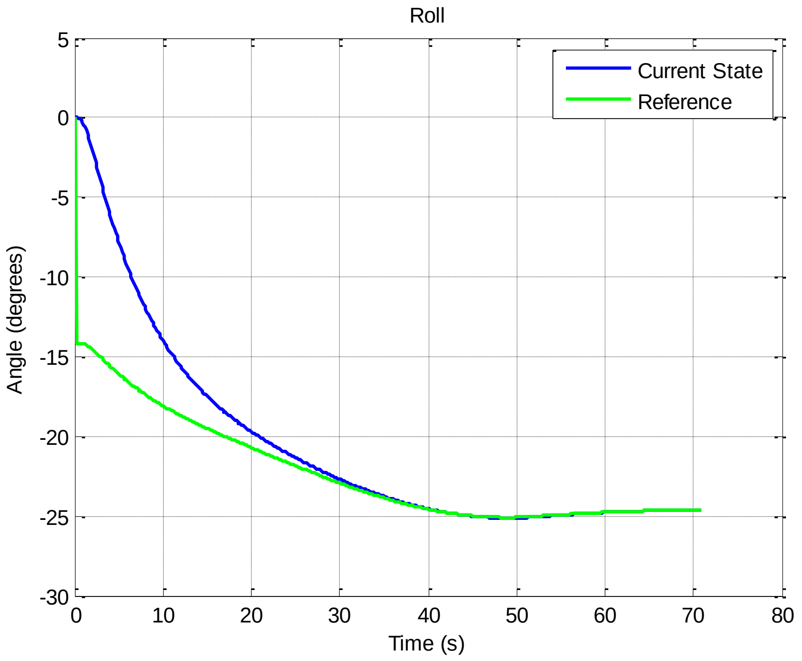


Figure 4.46 - Angle in pitch (simulation 6).

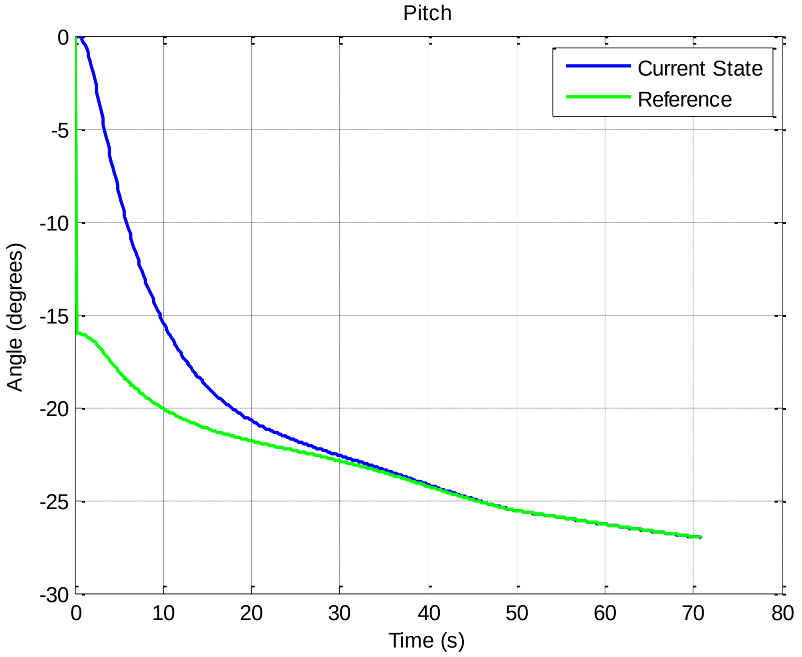
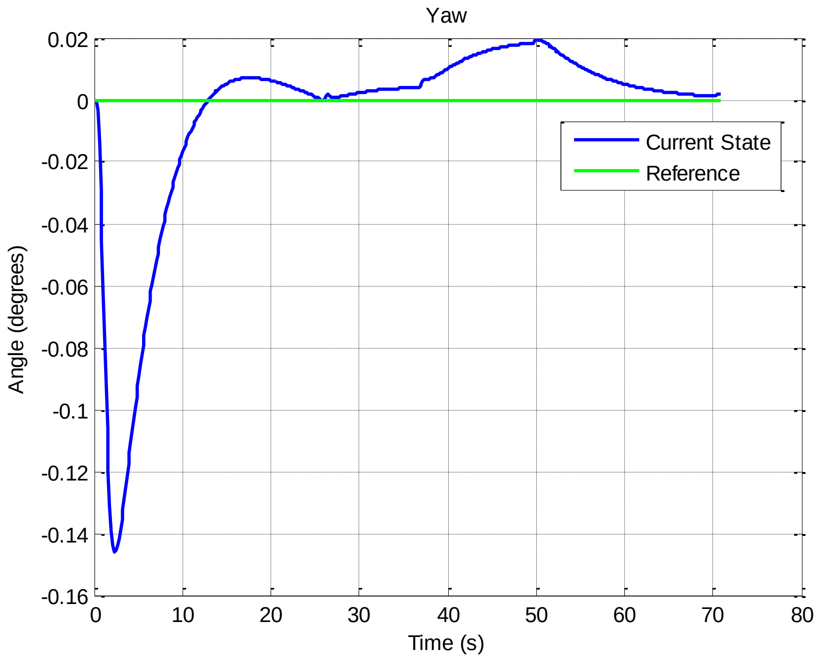


Figure 4.47 - Angle in yaw (simulation 6).



According to the software response, it was necessary *70.95* seconds for the manipulator reaches the target point with its joint 2 at *[-0.4818;0.3666;1.0336]* meters in the inertial frame.

4.2. Results analysis

To verify whether the simulations had the desirable consistency, for example having optimized the objectives it should, it will be analyzed results with relation to some variables of interest.

These variables are going to be the attitude control system, satellite orientation, distance between joint 2 and target, joint angular position, and time. The proposed variables allow an assessment of the performance improvements in the real world due to multi-objective optimization.

Given the joint 2 inertial position and knowing where the target is placed it is simple to calculate the distance between them. This determines the quality of solutions in satisfying the objective of maximizing accuracy, equivalent to minimize the distance between joint 2, workspace sphere center, and target point.

For every simulation, the simulation time can be easily compared. To evaluate the other objectives and compare solutions, it was devised a procedure that guarantees reliability.

To assess the energy consumption required by the attitude control system, it is possible to calculate the Root Mean Square (RMS) value of the torques along time. For the robot's energy consumption assessment, the RMS was calculated based on the joint angular position. The smaller these values, the better the quality of the solution to meet the objective.

To evaluate the quality of solutions in meeting the objective of maintaining the base satellite attitude, the RMS was obtained based on the satellite orientation along simulation time. This number represents how much the satellite changed its attitude during the maneuver. The best solution is that in which this value is the smallest.

Therefore, this procedure of finding the RMS values can be done to evaluate the attitude control system, satellite orientation and joint angular position.

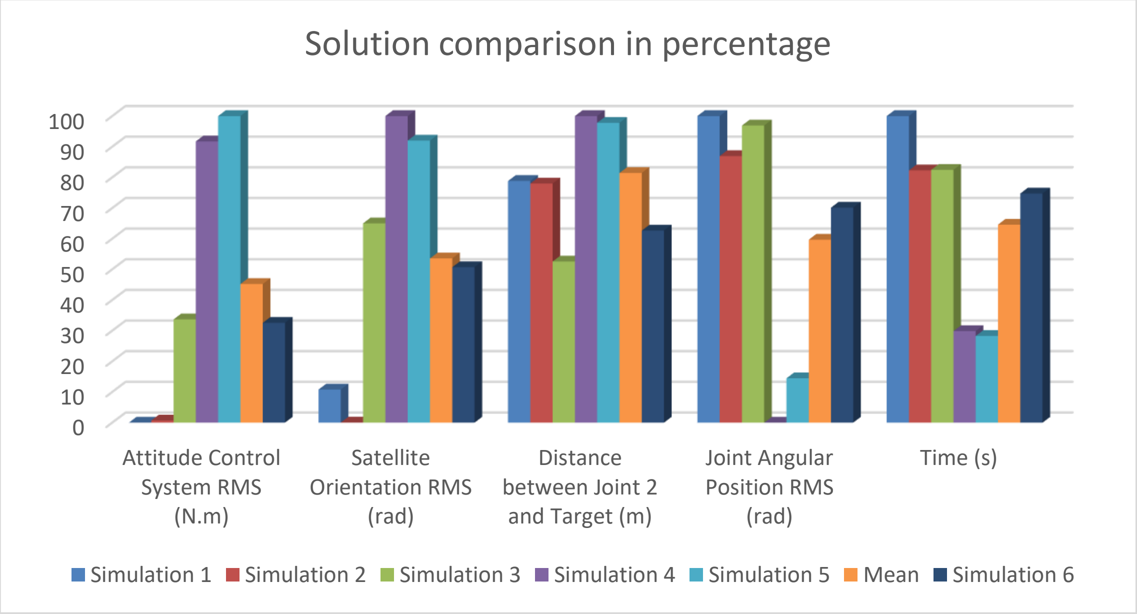
Table 4.2 summarizes all time simulation results obtained with their respective joint 2 position in the inertial frame used as parameter for comparison. As we can see, each objective was optimized by the simulation it was expected to be.

If the biggest value in a column is taken as one hundred percent, we can compare all the other values as a percentage of that. Simulation 6 is the most balanced of all since it meets simultaneously all objectives minimizing the losses. Therefore, it should be elected for a practical multi-objective mission. Figure 4.48 shows the comparative performance of all solutions.

Table 4.2 - Solutions summary.

Simulation	Objective	Joint 2 Inertial Position (m)	Attitude Control System RMS (N.m)	Satellite Orientation RMS (rad)	Distance between Joint 2 and Target (m)	Joint Angular Position RMS (rad)	Time (s)
1	Satellite Consumption	[0.1737; -0.0959; 1.3005]	0	0.1153	1.5929	1.5580	94.95
2	Attitude Maintenance	[0.0784; -0.1051; 1.3247]	0.0744	3.0592e-004	1.5760	1.3548	78.15
3	Manipulator Accuracy	[-0.0648; 0.6880; 0.9410]	3.5056	0.6944	1.0620	1.5100	78.30
4	Manipulator Consumption	[-0.9378; 0.5013; 0.0828]	9.5498	1.0682	2.0206	0	28.35
5	Maneuver Time	[-1.0141; 0.2588; 0.3433]	10.4101	0.9827	1.9762	0.2253	26.85
Mean	SLC - Multi-objective	-	4.70798	0.57218118	1.64554	0.92962	61.32
6	MMM - Multi-objective	[-0.4818; 0.3666; 1.0336]	3.3890	0.5418	1.2657	1.0926	70.95

Figure 4.48 - Solution comparison in percentage.



When the mean values of each objective for the first five simulations are calculated, it is possible to realize how close it is to the simulation *6*. Although simulation *6* evidences its superiority among the possible solutions in the set, due to the propagation of numerical errors and dynamics coupling, it is difficult to find exactly a solution coincident with the mean values.

The mean values represent the multi-objective solution related to the smallest loss of all objectives simultaneously, i.e., it is the solution provided by the *Smallest Loss Criterion* (SLC).

4.3. Successive simulations

One can think it is worth an attempt to get closer solutions to the mean values. Here, it will be presented a possible approach for that and it will be investigated how costly it can be.

To obtain a multi-objective solution closer to the mean value (SLC), it is possible to develop a systematic process of corrections for the vertices, whose places are defined by joint 2 inertial position, using a factor calculated based on the distance between the multi-objective solution and the desired mean value. Then, less or more importance is properly defined to each objective displacing the vertices accordingly to the correction factor.

For example, the MMM multi-objective simulation resulted in a time maneuver of *70.95* seconds while the mean value is *61.35* seconds, i.e., it should be approximately *15.7* percent faster. It is possible to correct the proper vertex, joint 2 inertial position, changing the polygon barycenter location of the current solution.

The described process can be applied to every objective and always when the multi-objective solution is different from the mean value (SLC). Then, it can be determined the closest solution in the set of possible options.

The process was applied updating the results from a set of fifty simulations generating the values provide by Table 4.3, where solutions *A, B, C, D, E* represent the polygon vertices or simulations *1, 2, 3, 4, 5* respectively and solution *F* is the simulation *6* (first multi-objective simulation).

Other solutions numbered from *1* to *50* form a set of possible solutions. After analyzing all simulations, it resulted that the forty-second (*42nd*) solution was the closest to the mean values (SLC) among all possible solutions in the set.

Theoretically, the proposed set might be updated indefinitely, however, we must plan a reasonable number of simulations for practical purposes. The computational cost can be considered an issue when applying this approach to find solutions nearer to the mean values (SLC).

By means of Figures 4.49, 4.50, 4.51, 4.52 and 4.53, it is possible to see the evolution of each objective as simulations change. The blue line represents the updated value for the proper objective while the red line is the mean value obtained from the solutions *A, B, C, D, E*. Insofar as each objective has its associated vertex on the polygon corrected by changing the desired position for joint 2, we verify a general enhancement in approximating to the mean value.

The solution *F* (MMM) presents overall advantages in time simulation and computational effort since it does not require multiple simulations, while its price is a solution slightly farther from the mean values (SLC).

Strictly comparing, the MMM avoids the SLC's inherent normalization achieving an approximated solution to the mean values, not representing exactly the smallest loss solution. Through systematic corrections presented here, it is possible to see how the distinct outcomes get closer.

The MMM application, jointly to these corrections, presented computational advantage with poorer precision comparatively to the exact application of SLC.

Table 4.3 - Set of possible solutions.

Solution	Attitude Control System RMS (N.m)	Satellite Orientation RMS (rad)	Distance between Joint 2 and Target (m)	Joint Angular Position RMS (rad)	Time (s)
A	0	0.1153	1.5929	1.558	94.95
B	0.0744	3.06E-04	1.576	1.3548	78.15
C	3.5056	0.6944	1.062	1.51	78.3
D	9.5498	1.0682	2.0206	0	28.35
E	10.4101	0.9827	1.9762	0.2253	26.85
F	3.389	0.5418	1.2657	1.0926	70.95
1	3.9216	0.6464	1.3446	0.9895	68.55
2	4.096	0.6764	1.3872	0.959	67.2
3	4.1354	0.6766	1.4098	0.9522	66.45
4	4.1042	0.6654	1.4237	0.956	66.15
5	4.069	0.6526	1.4344	0.9606	65.85
6	4.0413	0.6411	1.4427	0.9644	65.55
7	4.0104	0.6316	1.4497	0.9682	65.4
8	3.993	0.6243	1.4567	0.9702	65.25
9	3.9891	0.6186	1.4628	0.9701	64.95
10	3.9736	0.6138	1.468	0.9724	64.95
11	3.9722	0.6107	1.4733	0.9725	64.8
12	3.9776	0.6076	1.4775	0.9701	64.5
13	3.968	0.6051	1.4812	0.9732	64.65
14	3.9768	0.6039	1.4851	0.9724	64.5
15	3.9773	0.6026	1.4882	0.9723	64.5
16	3.9833	0.6014	1.4911	0.9709	64.35
17	3.9931	0.5997	1.4943	0.9682	64.05
18	3.9915	0.5981	1.4957	0.969	64.05
19	3.9888	0.5976	1.498	0.9707	64.2
20	4.0039	0.5974	1.5003	0.967	63.9
21	4.0035	0.596	1.5019	0.9674	63.9
22	4.0111	0.5954	1.5034	0.966	63.75
23	4.0124	0.5946	1.5043	0.9665	63.75
24	4.0158	0.5944	1.5063	0.9663	63.75
25	4.0146	0.5944	1.5075	0.9684	63.9

(Continues)

Table 4.3 - Continuation.

Solution	Attitude Control System RMS (N.m)	Satellite Orientation RMS (rad)	Distance between Joint 2 and Target (m)	Joint Angular Position RMS (rad)	Time (s)
26	4.0264	0.5948	1.5087	0.9665	63.75
27	4.0336	0.5941	1.5104	0.9646	63.6
28	4.0391	0.5931	1.5109	0.963	63.45
29	4.0343	0.5923	1.5112	0.9652	63.6
30	4.0401	0.5928	1.5122	0.9653	63.6
31	4.0399	0.5931	1.5134	0.9664	63.75
32	4.0514	0.5936	1.5144	0.9643	63.6
33	4.0528	0.593	1.5148	0.9645	63.6
34	4.0608	0.5926	1.5164	0.9625	63.45
35	4.0613	0.592	1.5166	0.9632	63.45
36	4.0592	0.592	1.5171	0.9651	63.6
37	4.0698	0.5925	1.5177	0.963	63.45
38	4.0703	0.592	1.5179	0.9632	63.45
39	4.072	0.5919	1.5183	0.9633	63.45
40	4.0788	0.5917	1.5186	0.9612	63.3
41	4.0782	0.5909	1.5198	0.9614	63.3
42	4.0843	0.5905	1.5203	0.9593	63.15
43	4.0787	0.5899	1.5201	0.9621	63.3
44	4.0826	0.5906	1.5207	0.9612	63.3
45	4.0844	0.5906	1.5211	0.9617	63.3
46	4.0816	0.5909	1.5212	0.9631	63.45
47	4.0809	0.5917	1.5217	0.9644	63.6
48	4.0906	0.5924	1.5225	0.9622	63.45
49	4.0906	0.5919	1.5225	0.9623	63.45
50	4.0918	0.5918	1.5225	0.9624	63.45

Figure 4.49 - Satellite consumption.

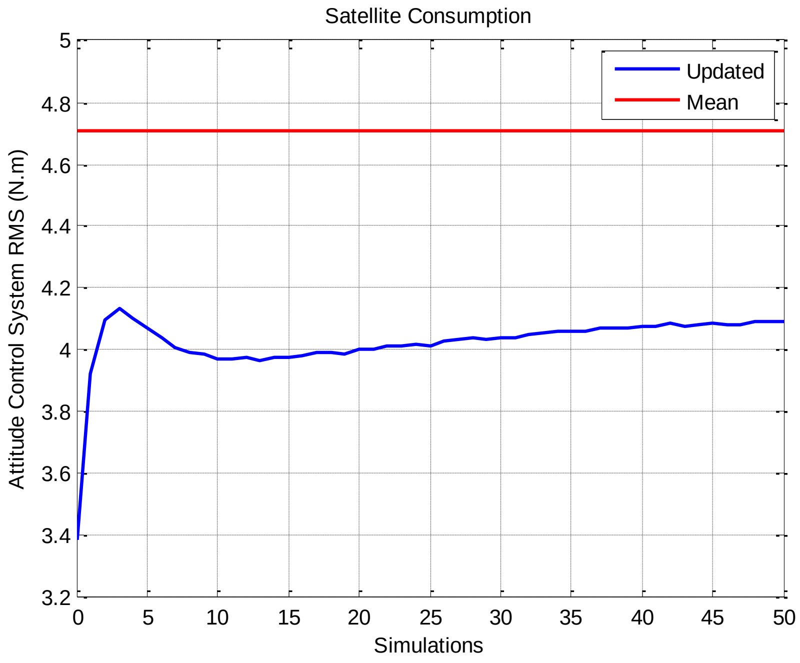


Figure 4.50 - Attitude maintenance.

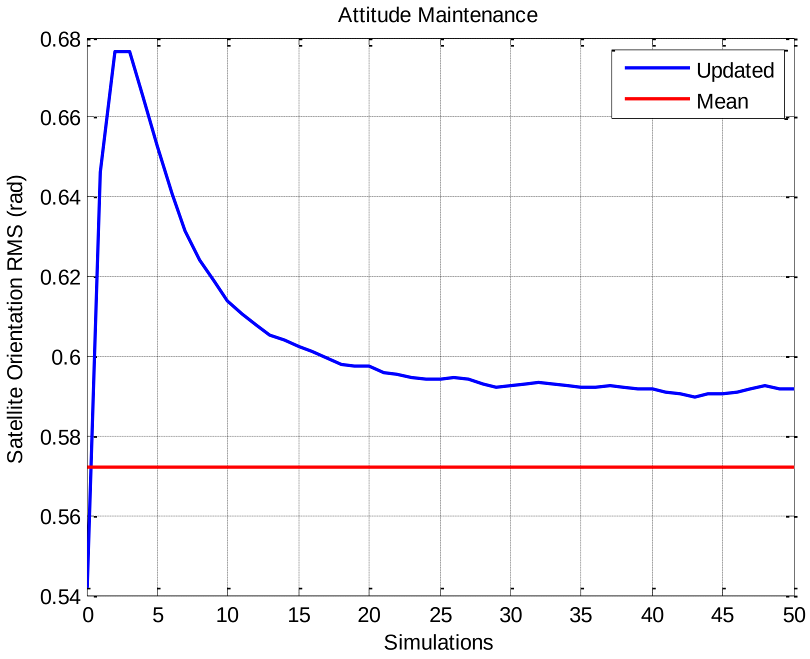


Figure 4.51 - Manipulator accuracy.

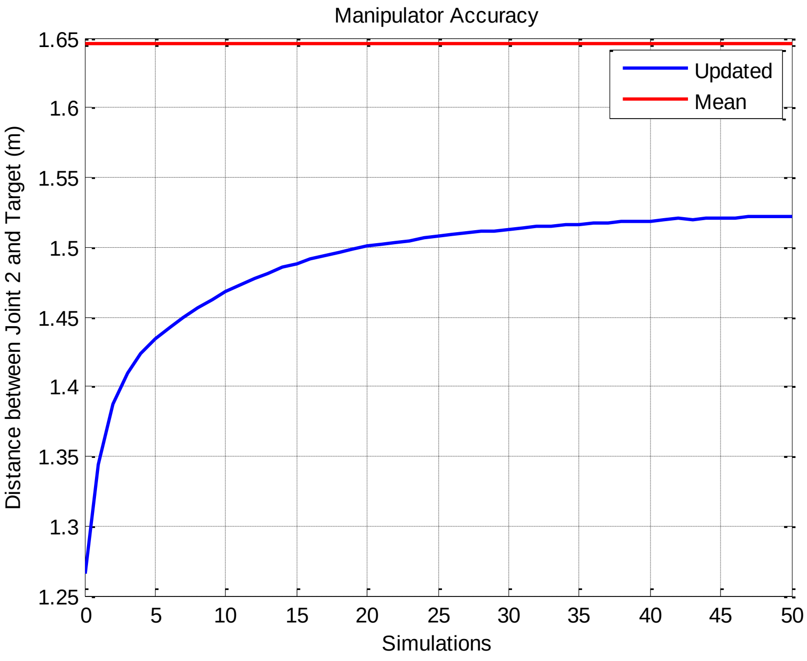


Figure 4.52 - Manipulator consumption.

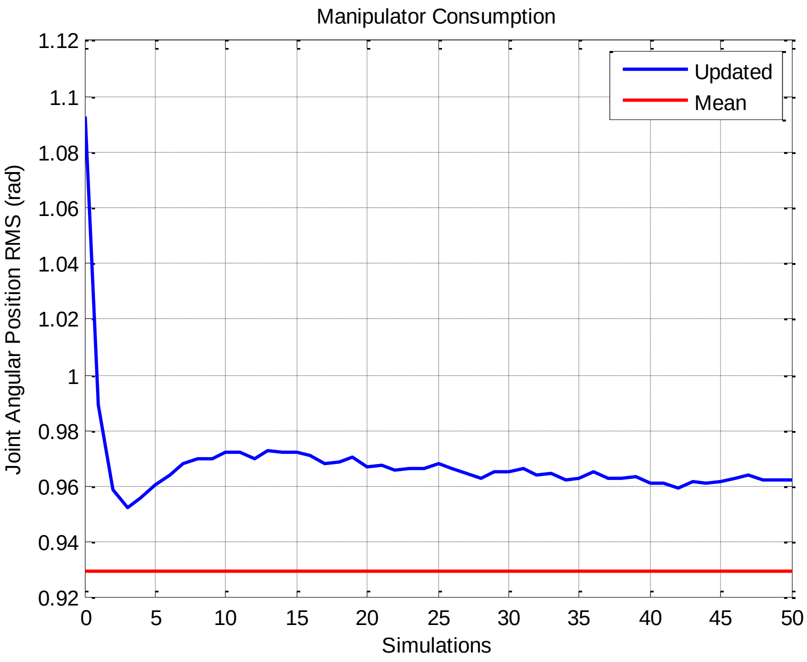
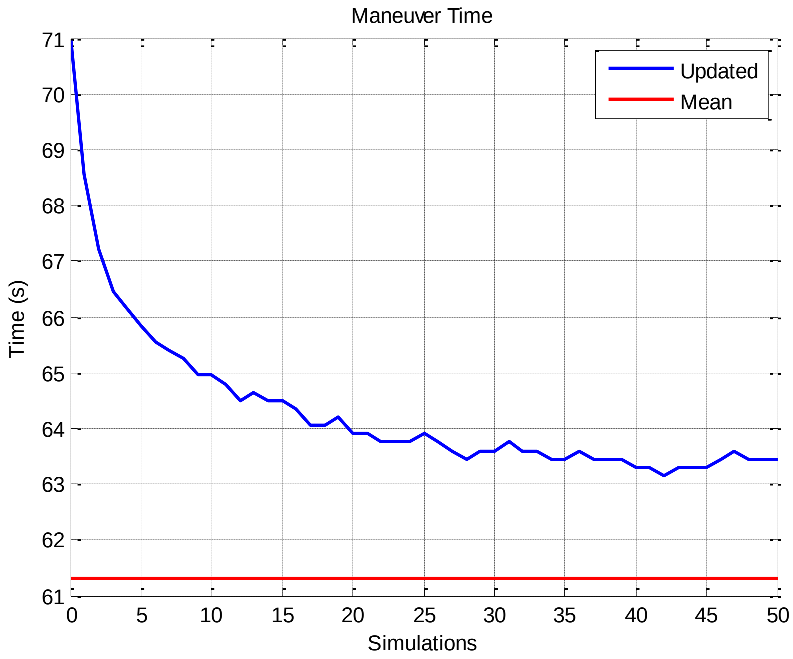


Figure 4.53 - Maneuver time.



4.4. Final considerations

In multi-objective problems, we cannot infer which solution that optimizes a number of objectives smaller than the total is better than any other, i.e., we cannot say, for example, a solution that optimizes two objectives is better than other that optimizes one or none (ROCCO, 2002). Any solution where not all the objectives are optimized might present at least one harmful value for a specific objective.

For example, considering a biological experiment, where water, food and oxygen are three necessary objectives for a colony growth. If a solution maximizes two of the objectives and overshadow one of them, it is possible that the neglected supply becomes insufficient and lead the entire colony to succumb (ROCCO, 2002). In scenarios where the objectives are conflicting, a balanced solution represents a decisive strategy for success.

The same reasoning can be applied to the berthing scenario presented in the past sections. For example, if we slightly change the target point position in space from *[-0.2;1.4;1.7]* to *[-0.2;1.5;1.7]*, we realized that a solution which optimizes the manipulator's energy consumption wouldn't be able to achieve our primary objective, of performing a berthing capturing the target, i.e., turning off the manipulator control system makes the target point unachievable, out of the robot workspace. Figures 4.54, 4.55 and 4.56 show such a scenario.

Figure 4.54 - Target out of workspace in satellite frame.

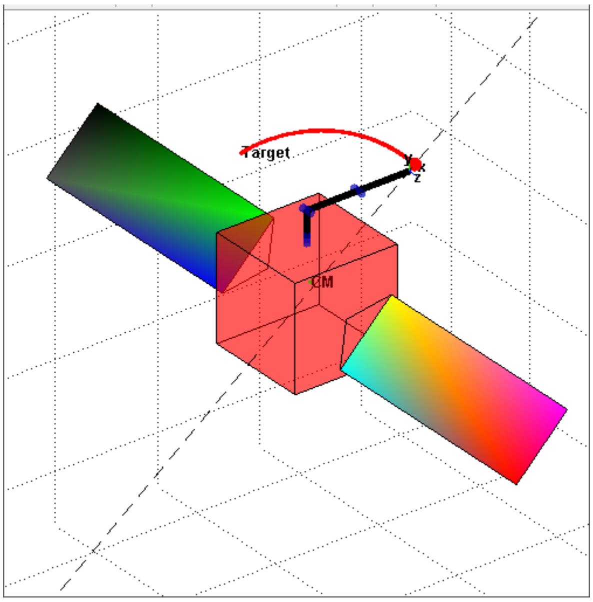


Figure 4.55 - Target out of workspace in inertial frame.

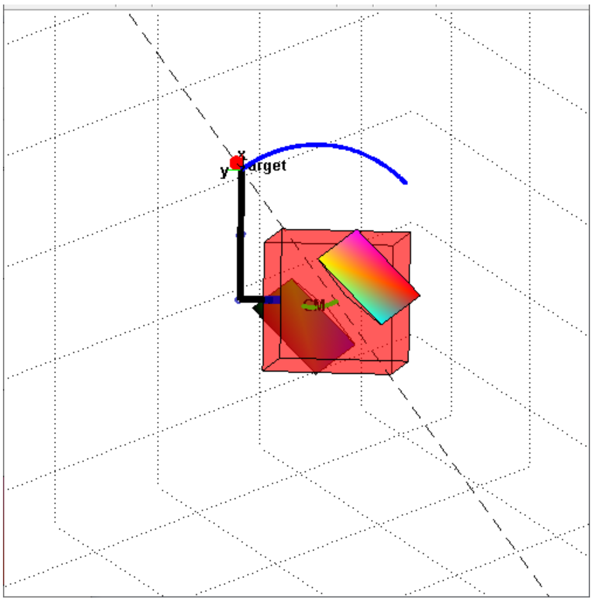
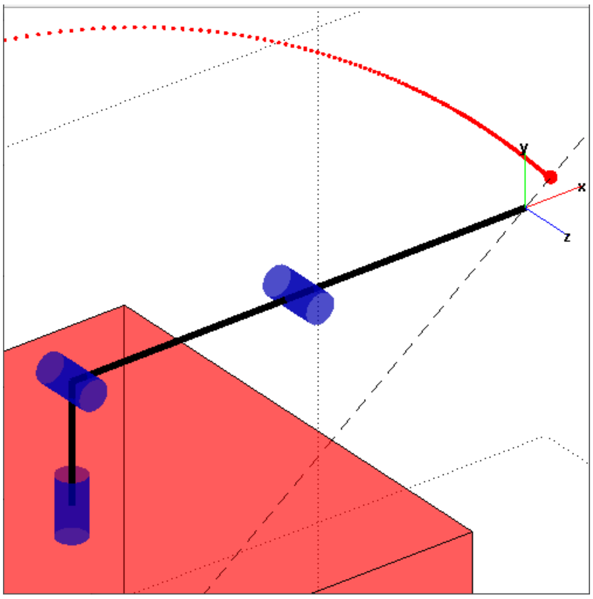


Figure 4.56 - Target out of workspace zoomed.



Recollecting, it is considered that the robotic manipulator wrist reaches the target point inside its workspace, if the distance between them is smaller than *0.1* meters. Even considering such a tolerance, the target was not achieved.

5 CONCLUSIONS

The objective of this work has been achieved since it has been developed a methodology able to provide solutions for berthing maneuvers of artificial satellites in space. In this scenario, the chaser satellite is endowed with a robotic manipulator and it is considered the multi-objective optimization to meet, in a balanced way, conflicting objectives.

In the field of multi-objective optimization, renowned methods (*Weighting method*, *Constraint method*, etc.) have been used for years; however, such methods have vital participation of a decision maker (DM) to select weights. The approach proposed here avoids DMs finding the most balanced solutions; this ensures automation and, as required by the scientific method, reproducibility of results.

This work contributed to the development of an innovative methodology based on a common metric for all objectives to deal with multi-objective optimizations. It has been shown how the *Mutual Metric Method* (MMM) is useful to find balanced solutions for an, until now, unexploited environment under a novel perspective of berthing maneuvers. Its performance has presented advantages, from the point of view of computational implementation, when compared to the *Smallest Loss Criterion* (SLC).

Considering the On-Orbit Servicing field, laboratories that enable scientists to validate planned routines and previously confirm results are fundamental tools for such critical and dangerous missions. In this work, a reliable software working together with proper hardware was used to demonstrate the usefulness of such laboratories for exploiting the HIL concept.

The European Proximity Operations Simulator (EPOS) provided improvements in the reliability of the developed models by increasing the experimental complexity. EPOS experiments demonstrated the consistency of the algorithms developed once its results contributed to the simulation software adequacy and fomented further applications harnessing multi-objective optimization.

The available facility, EPOS, was used to verify the software simulator in an environment for space berthing maneuvers through real-time and HIL simulations. Therefore, a virtual berthing manipulator was used as a workaround solution to compensate for the absence of a physical one.

These results presented movements being performed by the chaser satellite, through the robot that played its role in the simulation, notwithstanding its control system being turned off. Such movements were due to disturbances generated by the virtual manipulator, not physically portrayed (like a so-called "Phantom Limb"), as what would happen to a spacecraft endowed with a physical robotic manipulator.

As predicted by simulations solely based on software for the same inputs of target position, the hardware reacted to actions generated by a virtual part, berthing arm emulated. All things considered, this work was successful in creating a reliable software environment for berthing maneuvers jointly with EPOS robots at DLR.

The emulated berthing manipulator makes it easier (quicker, cheaper, etc.) to change design features when compared to the implementation of a physical robot. This helps the engineers to better plan space missions.

Employing the multi-objective optimization simulations, it was obtained points that served as geometrical parameters for comparison of solutions, avoiding the procedure of normalization naturally required by the *Smallest Loss Criterion*.

These points determined the attitude adopted by the satellite, through its attitude control system, aiming to optimize each of the objectives. In the context of multi-objective optimization, those points can be understood as extreme non-dominated solutions, individually optimizing each of the objectives.

In the final simulation, a balanced solution has been found. For sure, this solution is not the best if we consider one of the objectives, for example, it is not the fastest. However, it is correct to claim that this solution is the solution that minimizes the losses when dealing with conflicting objectives. Many other results with different numbers of objectives could be generated using the developed software tool.

The *Mutual Metric Method* proposed here presents noticeable generality for applications in other multi-objective optimization problems in space area or even in other fields of knowledge. However, it is vital to point out its limitations of employment. For example, it demands the verification of existence of a variable that represents a metric for every objective avoiding the SLC normalization. If this variable, or mutual metric, does not exist, the MMM application will not be possible.

During simulations, the only space environment characteristic considered was the microgravity, although other environmental disturbances could have been implemented in the simulator. In a real capturing mission, with only one try at your disposal, the results provided by the multi-objective simulation could be useful to control the real satellite and to meet the various objectives in the best way.

In the future, collision avoidance and contact dynamics will be important issues to be addressed and upgrades to the software tool can be implemented. For example, the manipulator used here had three joints to avoid higher complexity; more complex configurations could be simulated for evaluation of computational effort and assessment of the real-time capacity.

Finally, the study of balanced solutions among objectives can assist the roboticist during the design phase to select or rework characteristics of the manipulators in question. For example, investigating possibilities for changes in degrees of freedom, limits of joint variables that relate to reach, changes in spatial resolution, and constructive characteristics that impact accuracy, or redesign of controller gains that influence the total velocity.

REFERENCES

AGOSTINHO, A. C. **Controle por modos deslizantes aplicado ao movimento de atitude de satélites submetidos aos torques perturbadores gerados pelo movimento de um líquido**. 2019. 199 p. IBI: <8JMKD3MGP3W34R/3T633UL>. (sid.inpe.br/mtc-m21c/2019/04.17.18.21-TDI). Tese (Doutorado em Mecânica Espacial e Controle) - Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2019. Available from: <http://urlib.net/rep/8JMKD3MGP3W34R/3T633UL>.

AMORIM TERCEIRO, F. C. **Um método automático para desenvolver arquiteturas funcionais e físicas de sistemas de controle por otimização multi-objetivo baseada em modelos, atributos e métricas sistêmicas**. 2013. 196 p. IBI: <8JMKD3MGP7W/3EMDHQ8>. (sid.inpe.br/mtc-m19/2013/08.22.10.40-TDI). Tese (Doutorado em Mecânica Espacial e Controle) - Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2013. Available from: <http://urlib.net/rep/8JMKD3MGP7W/3EMDHQ8>.

AYTEN, K. K. **Optimum trajectory planning for redundant manipulators through inverse dynamics**. Thesis (Doctorate Mechanical Engineering) — University of Bath, Bath, 2012.

BAGCHI, S. **Design optimization and synthesis of manipulators based on various manipulation indices**. Dissertation (Master of Science in Mechanical Engineering) — The University of Texas at Arlington, Arlington, 2005.

BENNINGHOFF, H.; BOGE, T.; REMS, F. Autonomous navigation for on-orbit servicing. **KI - Künstliche Intelligenz**, v. 28, n. 2, p. 77–83, june 2014.

BENNINGHOFF, H.; REMS, F.; BOGE, T. Development and hardware-in-the-loop test of a guidance, navigation and control system for on-orbit servicing. **Acta Astronautica**, v. 102, p. 67–80, sept. 2014.

BENNINGHOFF, H. et al. European Proximity Operations Simulator 2.0 (EPOS) - a robotic-based rendezvous and docking simulator. **Journal of Large-Scale Research Facilities JLSRF**, v. 3, p. A107, apr. 2017.

BENNINGHOFF, H. et al. End-to-end simulation and verification of GNC and robotic systems considering both space segment and ground segment. **CEAS Space Journal**, v. 10, n. 4, p. 535–553, dec. 2018.

BOGE, T. et al. Hardware in the loop simulator für rendezvous und docking manöver. In: GERMAN AEROSPACE CONGRESS OF DGLR, 2009, Aachen, Germany. **Proceedings…** 2009.

BOGE, T. et al. EPOS - using robotics for RvD simulation of on-orbit servicing missions. In: AIAA MODELING AND SIMULATION TECHNOLOGIES CONFERENCE, 2010. **Proceedings…** Toronto, Ontario, Canada: American Institute of Aeronautics and Astronautics, 2010. Available from: <http://arc.aiaa.org/doi/10.2514/6.2010-7788>. Access in: 11 jan. 2020.

BOGE, T.; BENNINGHOFF, H.; TZSCHICHHOLZ, T. Hardware-in-the-loop rendezvous simulation using a vision based sensor. In: INTERNATIONAL ESA CONFERENCE ON GUIDANCE, NAVIGATION AND CONTROL SYSTEMS, 8., 2011. **Proceedings...** Karlovy Vary, Czech Republic, 2011.

CARNEIRO JUNIOR, D. P. **Comportamento do sloshing em ambiente de microgravidade**. 2017. 144 p. IBI: <8JMKD3MGP3W34P/3NS5DM8>. (sid.inpe.br/mtc-m21b/2017/05.10.22.40-TDI). Tese (Doutorado em Mecânica Espacial e Controle) - Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2017. Available from: <http://urlib.net/rep/8JMKD3MGP3W34P/3NS5DM8>.

COHON, J. L. **Multiobjective programming and planning**. New York: Academic Press, 1978.

CORKE, P. **Robotics, vision and control**. Cham: Springer International, 2017. v. 118.

CRAIG, J. J. **Introduction to robotics:** mechanics and control. 3.ed. Upper Saddle River, N.J: Pearson; Prentice Hall, 2005.

DEFENSE ADVANCED RESEARCH PROJECTS AGENCY - DARPA. **In-space robotic servicing program moves forward with new commercial partner**. 2020. Available from: <https://www.darpa.mil/news-events/2020-03-04>. Access in: 15 jun. 2020.

ELLERY, A. **An introduction to space robotics**. London; New York : Chichester: Springer; Praxis Pub, 2000.

ELLERY, A.; KREISEL, J.; SOMMER, B. The case for robotic on-orbit servicing of spacecraft: spacecraft reliability is a myth. **Acta Astronautica**, v. 63, n. 5–6, p. 632–648, sept. 2008.

EUROPEAN SPACE AGENCY - ESA. **ESA commissions world's first space debris removal**. 2019. Available from: <https://www.darpa.mil/news-events/2020-03-04>. Access in: 15 jun. 2020.

FORSHAW, J. et al. The ELSA-d end-of-life debris removal mission: mission design, in-flight safety, and preparations for launch. In: ADVANCED MAUI OPTICAL AND SPACE SURVEILLANCE TECHNOLOGIES CONFERENCE, 2019. **Proceedings…** Wailea, Maui, Hawaii: S. Ryan, The Maui Economic Development Board, 2019.

FLORES-ABAD, A. et al. A review of space robotics technologies for on-orbit servicing. **Progress in Aerospace Sciences**, v. 68, p. 1–26, july 2014.

FONSECA, I. M. D.; ARANTES JUNIOR, G.; BAINUM, P. M. Attitude dynamics and control of a LSS during the translation of the robot manipulator and astronaut walk along the space structure. In: INTERNATIONAL

ASTRONAUTICAL CONGRESS, 55, 2004, Vancouver, Canada. **Proceedings…** 2004.

FONSECA, I. M. D.; BAINUM, P. M. Space robotics application for large space structures. In: CONGRESSO TEMÁTICO DE DINÂMICA E CONTROLE DA SBMAC, 3., 2004, Ilha Solteira. **Anais...** Ilha Solteira: Unesp, 2004.

FONSECA, I. M. D. et al. Interaction between motions of robotic manipulator arms and the non-fixed base in on-orbit operations. **Journal of Aerospace Technology and Management**, v. 7, n. 4, p. 443–453, 2015.

FONSECA, I. M. D. et al. Attitude dynamics and control of a spacecraft like a robotic manipulator when implementing on-orbit servicing. **Acta Astronautica**, v. 137, p. 490–497, aug. 2017.

GEBHARDT, C. **Northrop grumman makes history, mission extension vehicle docks to target satellite**. 2020. Available from: <https://www.nasaspaceflight.com/2020/02/northrop-grumman-history-mission-extension-vehicle-docks-satellite/>. Access in: 15 jun. 2020.

GRALLA, E. L.; WECK, O. L. D. Strategies for on-orbit assembly of modular spacecraft. **Journal of the British Interplanetary Society**, v. 60, n. 6, p. 219–227, 2007.

GRANZIERA JUNIOR, F. **Mapeamento de conflitos na determinação e controle de atitude e estratégia para sua mitigação considerando erros dos sensores e atuadores e requisitos de missão**. 2015. 185 p. IBI: <8JMKD3MGP3W34P/3JCJQHL>. (sid.inpe.br/mtc-m21b/2015/04.24.14.00-TDI). Tese (Doutorado em Mecânica Espacial e Controle) - Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2015. Available from: <http://urlib.net/rep/8JMKD3MGP3W34P/3JCJQHL>.

GROOVER, M. P. **Automation, production systems, and computer-integrated manufacturing**. 5.ed. New York: Pearson Education, 2019.

HUGHES, P. C. **Spacecraft attitude dynamics**. Mineola, N.Y: Dover, 2004.

KUTTAN, A. **Robotics**. New Delhi: I.K. International House, 2007.

LAU, V. et al. A Multidisciplinary design optimization tool for spacecraft equipment layout conception. **Journal of Aerospace Technology and** Management, v. 6, n. 4, p. 431–446, 23 nov. 2014.

NARDIN, A. B. **Análise de manobras de atracação de satélites dotados de manipuladores robóticos**. 2015. 181 p. IBI: <8JMKD3MGP3W34P/3J259P2>. (sid.inpe.br/mtc-m21b/2015/02.18.23.42-TDI). Dissertação (Mestrado em Mecânica Espacial e Controle) - Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2015. Available from: <http://urlib.net/rep/8JMKD3MGP3W34P/3J259P2>.

NARDIN, A. B. **Internship report**: software environment for berthing maneuvers of satellites endowed with robotic manipulators. São José dos Campos: INPE, 2019. 58 p. IBI: <8JMKD3MGP3W34R/3U6K6T5>. (sid.inpe.br/mtc-m21c/2019/10.04.12.14-RPQ). Available from: <http://urlib.net/rep/8JMKD3MGP3W34R/3U6K6T5>.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION - NASA. OSAM-1: **On-orbit servicing, assembly and manufacturing mission**. 2020. Available from: <https://nexis.gsfc.nasa.gov/OSAM-1.html>. Access in: 15 jun. 2020.

NASH, J. F. The bargaining problem. **Econometrica**, v. 18, n. 2, p. 155, apr. 1950.

NISHIDA, S.-I. et al. Space debris removal system using a small satellite. **Acta Astronautica**, v. 65, n. 1–2, p. 95–102, july 2009.

PARETO, V.; MONTESANO, A. **Manual of political economy**: a critical and variorum edition. Oxford: Oxford University Press, 2014.

PETTERSSON, M. **Design optimization in industrial robotics**: methods and algorithms for drive train design. Linköping: Linköpings Universitet, 2008.

ROCCO, E. M.; SOUZA, M. L. O.; PRADO, A. F. B. A. Constellation station keeping using optimum impulsive maneuvers with time constraint. In: INTERNATITONAL ASTRONAUTICAL CONGRESS, 51., 2000, Rio de Janeiro, Brasil. **Proceedings…** Rio de Janeiro, 2000.

ROCCO, E. M.; SOUZA, M. L. O.; PRADO, A. F. B. A. Multi-objective optimization approach applied to station keeping of satellite constellations. In: AAS/AIAA ASTRODYNAMICS SPECIALISTS CONFERENCE, 2001, Quebec, Canada. **Proceedings…** Quebec, 2000. AAS 01-346.

ROCCO, E. M.; SOUZA, M. L. O.; PRADO, A. F. B. A. Station keeping of satellite constellations with time constraint using optimal bi-impulsive maneuvers. In: WINTER, O. C.; PRADO, A. F. B. A. (Ed.). **Advances in space dynamics 3 aplications in astrounautics**. São José dos Campos, SP: Instituto Nacional de Pesquisas Espaciais, 2002.

ROCCO, E. M. **Manutenção orbital de constelações simétricas de satélites utilizando manobras impulsivas ótimas com vínculo de tempo**. 2002. IBI: <6qtX3pFwXQZ3r59YCT/H3MrR>. Tese (Doutorado em Mecânica Espacial e Controle) - Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2002.

ROCCO, E. M.; SOUZA, M. L. DE O. E; PRADO, A. F. B. DE A. Multi-objective optimization applied to satellite constellations I: formulation of the smallest loss criterion. In: IAC INTERNATIONAL ASTRONAUTICAL CONGRESS, 54., 2003. Bremen, Germany. **Proceedings…** 2003.

ROCCO, E. M.; SOUZA, M. L. O.; PRADO, A. F. B. A. Further applications of the smallest loss criterion in the multi-objective optimization of a satellite constellations. In: INTERNATIONAL ASTRONAUTICAL CONGRESS (IAC 2005), 56., 2005, Fukuoka, Japan. **Proceedings…** Fukuoka, 2005a.

ROCCO, E. M.; SOUZA, M. L. O.; PRADO, A. F. B. A. Multi-objective optimization applied to satellite constellations II: initial applications of the smallest loss criterion. In: IWSCFF INTERNATIONAL WORKSHOP ON SATELLITE CONSTELLATIONS AND FORMATION FLYING, 4., 2005, São José dos Campos, Brasil. **Proceedings…** São José dos Campos, 2005b. p. 123-132. Available from: <http://urlib.net/sid.inpe.br/iris@1916/2005/10.06.13.14>.

ROCCO, E. M.; COSTA FILHO, A. C; CARRARA, V. Effect of the coupling between attitude and orbital control in maneuvers using continuous thrust. In: CONFERÊNCIA BRASILEIRA DE DINÂMICA, CONTROLE E APLICAÇÕES, 10., 2011, Águas de Lindóia. **Anais...** Águas de Lindóia: [s.n.], 2011.

ROCCO, E. M.; SOUZA, M. L. DE O. E; PRADO, A. F. B. DE A. Station keeping of constellations using multiobjective strategies. **Mathematical Problems in Engineering**, v. 2013, p. 1–15, 2013.

ROCCO, E. M.; COSTA FILHO, A. C. DA. Avaliação dos desvios na trajetória originados pelo acoplamento entre o controle de atitude e de órbita em manobras orbitais com propulsão contínua. In: CONGRESSO NACIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL, 35., 2015. **Proceedings...** 2015. Available from: <https://proceedings.sbmac.org.br/sbmac/article/view/585>. Access in: 11 jan. 2020.

SANTOS, R. R.; STEFFEN, V.; SARAMAGO, S. F. P. Optimal task placement of a serial robot manipulator for manipulability and mechanical power optimization. **Intelligent Information Management**, v. 2, n. 9, p. 512–525, 2010.

SANTOS, W. G. **Discrete multiobjective optimization applied to the spacecraft actuators command problem and tested in a hardware-in-the-loop rendezvous simulator**. 2015. 160 p.
IBI: <8JMKD3MGP3W34P/3HRTNES>. (sid.inpe.br/mtc-m21b/2015/01.30.12.04-TDI). Tese (Doutorado em Mecânica Espacial e Controle) - Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2015. Available from: <http://urlib.net/rep/8JMKD3MGP3W34P/3HRTNES>.

SANTOS, W. G.; ROCCO, E. M.; BOGE, T. Design of a linear time-invariant control system based on a multiobjective optimization approach. **Computational and Applied Mathematics**, v. 35, n. 3, p. 789–801, june 2015.

SANTOS, W. G. et al. Hardware-in-the-Loop rendezvous tests of a novel actuators command concept. **The Journal of the Astronautical Sciences**, v. 63, n. 4, p. 287–307, dec. 2016.

SICILIANO, B. et al. **Robotics**. London: Springer, 2009.

PIRES, E. J. S.; OLIVEIRA, P. B. M.; MACHADO, J. A. T. Multi-criteria optimization manipulator trajectory planning. In: LAZINICA, A.; KAWAI, H. (Ed.). **Robot manipulators new schievements**. [S.l.]: InTech, 2010.

VENDITTI, F. C. F. et al. Gravity-assisted maneuvers applied in the multi-objective optimization of interplanetary trajectories. **Acta Astronautica**, v. 67, n. 9–10, p. 1255–1271, nov. 2010.

WANG, L.-C. T.; CHEN, C. C. A combined optimization method for solving the inverse kinematics problems of mechanical manipulators. **IEEE Transactions on Robotics and Automation**, v. 7, n. 4, p. 489–499, aug. 1991.

WISMER, D. A.; CHATTERGY, R. **Introduction to nonlinear optimization**: a problem solving approach. New York: North-Holland, 1978.

YOSHIDA, K. Space robot dynamics and control: to orbit, from orbit, and future. In: HOLLERBACH, J. M.; KODITSCHEK, D. E. (Ed.). **Robotics research**. London: Springer, 2000. p. 449–456.

YOSHIDA, K. ETS-VII Flight experiments for space robot dynamics and control. In: RUS, D.; SINGH, S. (Ed.). **Experimental robotics VII**. Berlin, Heidelberg: Springer, 2001. v. 271. p. 209–218.

YOSHIDA, K. Engineering test satellite VII flight experiments for space robot dynamics and control: theories on laboratory test beds ten years ago, now in orbit. **The International Journal of Robotics Research**, v. 22, n. 5, p. 321–335, may 2003.