On-Orbit Servicing Satellite Docking Mechanism Modeling and Simulation

Karim Bondoky
2015

Master of Science (120 credits)
Space Engineering - Space Master

Luleå University of Technology
Department of Computer Science, Electrical and Space Engineering
On-Orbit Servicing Satellite Docking Mechanism Modeling and Simulation

by

Karim Bondoky

A thesis submitted in partial fulfilment for the degree of Masters of Science in Space Science and Technology

Supervisors:

Eng. Sebastian Schwarz  
Airbus Defence and Space
Prof. Dr. Sergio Montenegro  
University of Würzburg
Dr. Johnny Ejemalm  
Luleå University of Technology

October 2014
Declaration of Authorship

I, Karim Bondoky, declare that this thesis titled, ‘On-Orbit Servicing Satellite Docking Mechanism Modeling and Simulation’ and the work presented in it are my own. I confirm that:

■ This work was done wholly while in candidature for a research degree at both Universities.

■ Where any part of this thesis has previously been submitted for a degree or any other qualification, this has been clearly stated.

■ Where I have consulted the published work of others, this is always clearly attributed.

■ Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.

■ I have acknowledged all main sources of help.

Signed: 

Date: 15.10.2014
Docking mechanism of two spacecraft is considered as one of the main challenging aspects of an on-orbit servicing mission. This thesis presents the modeling, analysis and software simulation of one of the docking mechanisms called "probe and drogue". The aim of this thesis is to model the docking mechanism, simulate the docking process and use the results as a reference for the Hardware-In-the-Loop simulation (HIL) of the docking mechanism, in order to verify and validate it.

Modeling of the two spacecraft, docking mechanism and the contact dynamics are the most challenging parts of this thesis. The spacecraft motion’s modeling is done using Newton-Euler equations, with consideration of the homogeneous transformations of the different coordinate frames. Contact dynamics modeling included the Hertz dynamics, different types of damping models and a modified Coulomb’s friction model. Simulink/-Matlab 2013b are used to build two models, one for numerical analysis of the model and another one for the HIL simulation.

Results are analyzed in Matlab and visualized using the 3D visualization program that is built within the scope of this thesis. The results are then used as a guide for the requirements of the robotic hardware that will carry out the HIL simulation. Also, two different robotics hardware are proposed in this thesis to carry out the HIL simulation.

The work has a lot of aspects in common with the satellite of the German Space Agency (DLR) called, "DEutsche Orbitale Servicing Mission" (DEOS). This work took place in Friedrichshafen/Germany at the facilities of Airbus Defence and Space (the prime contractor of DEOS) in collaboration with Würzburg university in Germany and Luleå university of technology in Sweden.

**Keywords:** Docking, On-orbit servicing, Contact dynamics, Multi-body modeling, Multi-body dynamics, Newton-Euler, Homogeneous transformation.
Acknowledgments

I would like to express my gratitude to my supervisor in Airbus DS, Sebastian Schwarz, whose expertise, understanding, and patience, added considerably to my graduate experience. I appreciate his vast knowledge and skills in many areas (e.g., different fields of engineering, vision, project management, interaction with participants/collleagues), and his assistance in writing reports (i.e., proposals and applications, and this thesis), which have enlightened me on different levels. I have to say that without you, I would not have gone that far with this work. I would like to thank my supervisor in Würzburg University, Prof. Dr. Sergio Montenegro, whose experience, advices and consideration had a great contribution to the important decisions that were made during this work (i.e., thesis planning, thesis duration extension). I would like to thank my examiner in Luleå University of Technology, Dr. Johnny Ejemalm for his instructions and kind help.

I would like to express my appreciation for the DEOS family in Airbus DS who gave the time I spent in Friedrichshafen a special taste. Above all, I would like to express my sincere appreciation and gratitude to Dr. Andreas Rathke for his continuous support, advice, and for making everything easy for me. I appreciate his invaluable help for brainstorming, contacting people and writing proposals. Your help has definitely left a fingerprint in my life. Also, I would like to thank Arno Dietrich for every advice he gave me for my career life. I would like to express my gratitude to Patrick Bergner for his precious technical support and advices throughout the thesis. I acknowledge his ideas and support in this thesis (especially in the "collision point detection in the hardware") which have saved me a lot of energy and time.

I must also acknowledge Markus Huchler and Dr. Mathias von Alberti who opened the door for me to be in Airbus Defence and Space GmbH.

Special thanks to Anette Snällfot-Brändström, Maria Winnebäck, Ursula Shahmary, Heidi Schließus, Christine Buchholz and Anja Moser for making things administratively possible to work on this thesis.

A very special thanks to my family. Words cannot explain how grateful I am to my father and my mother, Nashaat and Nahed, for all the sacrifices, prayers, support you have made on my behalf. Your love and support to me (e.g., financially, physically, psychologically and spiritually) were what sustained me thus far. My special gratitude and appreciation also goes for my brother and my sister-in-law, Ramy and Thimar, for being always there for me. Finally, my gratitude goes for my awesome family members whose love have always been overwhelming me, thank you geddo Emile, teta Tuna, teta Monda, Hoda, Marinette, Misho, Hany, Glory, Sara and Dano. Also, Shadia, Nohad, Mounir, Hedy, Wafik, Fady and Marise.
I would like to acknowledge my SpaceMaster friends: Jānis Gailis and Simon Delchambre who took the time to proof-read all the master’s thesis.

Last but not least, a special thanks for my friends who were part of this phase of my life. You made me enjoy these 7 months with all the fun, events, support we had together. Thanks Christoph Holzapfel, Sebastian Fischer. I would like to thank my SpaceMaster friends in Würzburg: Jānis, John and Siddhath.
1 Introduction
  1.1 Background and Motivation ........................................ 1
  1.2 Problem and Goal Definition ..................................... 4
  1.3 Research Method ................................................. 5
  1.4 Thesis Structure ................................................ 7

2 Literature Review
  2.1 Docking Mechanisms History .................................... 8
  2.2 Multi-body Dynamics and Modeling of Docking Mechanisms .... 12
    2.2.1 Equations of Motion ....................................... 12
    2.2.2 Modeling aspects .......................................... 13
  2.3 Hardware-In-the-Loop Simulation Techniques .................. 14

3 Research Methodology
  3.1 System Overview ................................................ 18
  3.2 Hardware Model .................................................. 22
    3.2.1 Overview .................................................. 22
    3.2.2 Objective ................................................ 22
    3.2.3 Description .............................................. 22
    3.2.4 Case study ............................................... 23
  3.3 Software Numerical Model ....................................... 24
Contents

3.3.1 Overview .................................................. 24
3.3.2 Objective .................................................. 25
3.3.3 Description ................................................ 25
3.3.4 Case Study ................................................ 26

3.4 Dynamic Analysis and Modeling ................................ 27
3.4.1 6 DOF Equations of Motion .................................. 28
3.4.1.1 Newton-Euler Equations of Motion (Theoretical Back-
ground) .................................................. 28
3.4.1.2 Rigid Body Satellites Equations of Motions ............ 30
3.4.2 Coordinate Frame Transformations .......................... 31
3.4.2.1 Rotations ........................................... 32
3.4.2.2 Dynamic Model Homogeneous Transformation ........ 33
3.4.3 Collision Detection ......................................... 35
3.4.3.1 Software Model ...................................... 35
3.4.3.2 Hardware Model ..................................... 40
3.4.4 Contact Dynamics ........................................... 44
3.4.4.1 Hertzian Contact Force ................................ 45
3.4.4.2 Non-linear Contact Damping ......................... 46
3.4.4.3 Contact Friction ..................................... 48

4 3D Visualization ................................................. 50
4.1 Matlab 3D Visualization program ................................ 50

5 Hardware System .................................................. 54
5.1 6 DOF Forces/Torques Reaction Table .......................... 54
5.2 Proposed Robotic Hardware ..................................... 55
5.2.1 Robotic Manipulator ....................................... 55
5.2.2 6 DOF Stewart Platform (Hexapod) ....................... 55
5.2.3 Hardware Preliminary Requirements ....................... 56

6 Experiment and Results .......................................... 57
6.1 Simulation Scenario ............................................. 57
6.1.1 Chaser ..................................................... 58
6.1.2 Target ...................................................... 58
6.2 Simulation Results ............................................. 59
6.2.1 Contact Trajectory ....................................... 59
6.2.2 Relative Velocity ......................................... 60
6.2.3 Velocity .................................................... 61
6.2.4 Angular Velocity .......................................... 63
6.2.5 Forces ....................................................... 65
## Contents

Chaser ......................................................... 65  
Target ....................................................... 65  

6.2.6 Torques ............................................... 66  
Chaser ......................................................... 66  
Target ....................................................... 67  

6.2.7 Kinetic Energy ......................................... 68  
Chaser ......................................................... 68  
Target ....................................................... 68  

7 Conclusions and Future Work 70  
7.1 Conclusion .................................................. 70  
7.2 Future Work ............................................... 71  

A Software Numerical Analysis Simulink Model 72  
B Hardware-In-the-Loop Simulation Simulink Model 84  
C Miscellaneous Matlab Codes 89  

Bibliography 91
List of Figures

1.1 International Space Station (ISS) has been in the Low Earth Orbit (LEO) at altitude of 330 km since 1998. It is a joint project among five space agencies. ISS undergoes docking regularly when a module is sent to it. It also undergoes on-orbit servicing by its astronauts on board. [1] .................. 1

1.2 Space debris are one of the main threats against any satellite. The figure on the left shows a cut in the solar panel of the ISS because of the space debris. The other figure (right) depicts the ISS as it undergoes an on-orbit servicing by the astronaut Scott Parazynski. The cut of the solar panel was fixed after a space-walk that lasted 7 hours, 19 minutes. [1] .................. 3

1.3 The left figure depicts the ISS robotic arm while capturing the Orbital Sciences’ Cygnus cargo craft, and preparing it for docking. [2] The robotic arm is developed by Canadian Space Agency (CSA). The right figure depicts the cargo craft after it has been docked. [3] .......................... 5

1.4 This figure depicts the objective of the project (docking simulation) and its four sub-systems (e.g., System modeling, 3D Visualization, Multi-body dynamics tool and Hardware) ................................. 6

2.1 A family tree of the different docking mechanisms [4]. .................. 8

2.2 Left figure depicts the Gemini-Agena docking mechanism. While the probe and drogue mechanism (right figure) consists of a probe, which is inserted in a cone shaped receptacle. The docking is accomplished when the probe is captured softly by the latches at the end of the probe. Image courtesy of NASA [5] ................................. 9


2.4 Hybrid Russian docking mechanism consists of active (Left figure) [7] and passive (Right figure) parts. It is a combination of the “probe and drogue” and “APAS” systems. Left figure was taken from the ISS by one of the Expedition three crew members. In this figure, it is only seconds away from docking with the International Space Station (ISS). The vehicle was launched on September 14, 2001 and docking occurred on September 16. [8] ................................. 10

2.5 NASA conceptual design of the standard berthing and docking mechanism [9] ................................. 11

2.6 Left figure depicts the mechanical design of the International Berthing and Docking Mechanism (IBDM) [10]. In 2014, ESA announced that a new berthing and docking mechanism is being researched on. It senses the forces between the two vehicles undergoing the docking process. Its conceptual design is shown on the right figure [11]. ................................. 12
List of Figures

2.7 Apollo manned docking simulator was originally used by the astronauts preparing for Gemini missions. In this picture, the astronaut appears to be docking with a full-scale Apollo Command Module [12].

2.8 The left and the middle figure shows the Russian-American docking simulator based on two 6-DOF Stewart platforms, while the right figure shows the Chinese docking simulator [13].

2.9 HIL Simulation using Air-Bearing [25].

2.10 European Proximity Operations Simulator (EPOS 2.0) is one of the state-of-the-art of the HIL simulation of rendezvous, berthing and docking. HIL Simulation is based on two industrial KUKA robotic manipulators.

2.11 HIL Simulation of the International Berthing and Docking Mechanism (IBDM) by ESA.

3.1 The two spacecraft involved in the docking process are represented as rigid bodies. Many docking parameters are considered during the docking process. Some of these parameters are illustrated in this figure.

3.2 Simulink and Matlab are both used in this thesis to carry out the simulation. This illustration shows the architecture of the system. Simulink - acting as the main engine - connects all the sub-systems, process the inputs and the outputs. This illustration does not show the exact flow of the data through.

3.3 An overview of docking mechanism verification approach. Two models will be built: a software model and a hardware model. Results of the software numerical model will act as reference for the HIL simulation. The software model represents all the system of equations that the two spacecraft follow in space. Hardware represents the real testing of the docking.

3.4 An overview of the multi-body dynamic system with the coordinate frames assigned at the proper positions at each body. F0 is the World frame, F1 and F4 represent the Chaser and the Target’s Center of Mass (CoM) respectively.

3.5 Hardware-In-the-Loop (HIL) simulation block diagram. Forces and torques due to the contact are transmitted to the multi-body dynamic simulator. Then, the dynamic simulator outputs the relative motion of the Chaser (Manipulator) w.r.t. the 6 DOF reaction table (FTR Table). The loop keeps going until docking is accomplished.

3.6 The software numerical model is the modeling of the real life problem in mathematical and physical equations. This figure depicts the block diagram of the model.

3.7 Multi-body dynamics system consists of many different sub-systems, which are all handled in this section. The figure above shows the sub-systems of the Dynamic modelling system.

3.8 Transformation diagram shows the transformation from one frame to another. Homogeneous transformation is used to include both rotational and translational motion.

3.9 Collision detection and the contact point determination description. The approach is to split the model into two views during analysing. This figure depicts the X-Z plane.

3.10 The Y-Z plane view of the collision detection using geometrical analysis.
3.11 Three regions are defined inside the cone during the collision detection program. Each region follows the collision detection approach, however, the values change according to each region. ........................................... 39

3.12 The overview illustration of the X-Z Plane of the drogue. The marked part is zoomed for clarification, in the next Figure 3.13. ...................... 42

3.13 The X-Z plane view of the contact point detection of the hardware model. The main objective is to find the contact point position, knowing only the input forces and torques and the geometrical model dimensions. ........... 43

3.14 Hertzian contact stiffness parameters [15] .................................................. 45

3.15 A comparison between two reaction force curves due to the penetration depth of the probe in the drogue’s surface. The hysteresis represents the damping effect on the force (i.e., some energy is lost during collision). While the Hertz force model considers an elastic collision (i.e., no energy is lost). .......................................................... 47

3.16 A modified Coulomb friction model is presented in this thesis, which takes into consideration both the static ($\mu_s$) and the dynamic friction ($\mu_d$) coefficient. This model also avoids the singularity of Coulomb law, which occurs in the transition between negative and positive slip velocity. .......................... 49

4.1 3D overview of the two rigid bodies of both satellites using the 3D visualization program built using Matlab. The center of mass of each body is marked with the coordinate axes (X: red, Y: green, Z: blue) ..................... 51

4.2 The two rigid bodies satellites are shown here during the docking simulation. Each figure shows the two bodies w.r.t the world frame, but with a different angle of view. .............................................................. 52

4.3 The 3D visualization program also outputs the relative motion of the Chaser w.r.t. the Target. In these figures we can see the difference between the two types of outputs (e.g., world frame and relative frame) showing the same situation, in which the Target satellite (green) is rotated by 135° around Y-axis, while the Chaser satellite (yellow) is standing still. The left figure shows the ordinary motion of satellites as seen in space (world frame) in XZ plane. While, the right figure shows the relative motion of the Chaser w.r.t. the Target in XZ plane. ......................... 53

5.1 Two robotic hardware systems are possible to carry out the HIL simulation. Either a 6 DOF robotic manipulator (left figure) or a 6 DOF hexapod system (right figure). However, finding a robotic hardware that matches the specifications and the minimum requirements is not easy. Therefore, the standard hardware robotic systems, off-the-shelf hardware, are not recommended for this HIL simulation. .............................. 55

6.1 The worst case scenario of the docking approach is described in the figure above, with an approaching relative velocity of 0.1 [m/s]. The red sign ‘x’ represents the closest point of collision between the probe and the drogue. ...................... 58

6.2 A screen-shot from the simulation that depicts the configuration, the axes and the dimensions of assumed for the two rigid satellites. The Target lies at the origin of the inertial frame (F0), which is shown above. ............. 59
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.3</td>
<td>Multiple screen-shots from the simulation showing the satellite trajectory during the docking process. The axes of frame (F2) - base of the probe - and frame (F5) - base of the drogue - appear in the figure as (X: red, Y: green)</td>
</tr>
<tr>
<td>6.4</td>
<td>The relative velocity of the Chaser w.r.t. the Target throughout the docking simulation represented in the inertial frame (F0) axes.</td>
</tr>
<tr>
<td>6.5</td>
<td>Velocity of the Chaser satellite in body frame throughout the docking simulation.</td>
</tr>
<tr>
<td>6.6</td>
<td>Velocity of the Target satellite in body frame throughout the docking simulation.</td>
</tr>
<tr>
<td>6.7</td>
<td>Angular velocity of the Chaser satellite in body frame throughout the docking simulation.</td>
</tr>
<tr>
<td>6.8</td>
<td>Angular velocity of the Target satellite in body frame throughout the docking simulation.</td>
</tr>
<tr>
<td>6.9</td>
<td>Forces applied on the Chaser satellite during the docking process w.r.t. the local axes of the satellite.</td>
</tr>
<tr>
<td>6.10</td>
<td>Forces applied on the Target satellite during the docking process w.r.t. the local axes of the satellite.</td>
</tr>
<tr>
<td>6.11</td>
<td>Torques applied on the Chaser satellite (in body frame F1) during the docking process.</td>
</tr>
<tr>
<td>6.12</td>
<td>Torques applied on the Target satellite (in body frame F4) during the docking process.</td>
</tr>
<tr>
<td>6.13</td>
<td>Kinetic energy of the Chaser satellite (in body frame F1) throughout the docking simulation.</td>
</tr>
<tr>
<td>6.14</td>
<td>Kinetic energy of the Target satellite (in body frame F4) throughout the docking simulation.</td>
</tr>
<tr>
<td>6.15</td>
<td>The change of the kinetic energy of the Target satellite during the second impact throughout the docking simulation, shown in the (local) Y-axis.</td>
</tr>
</tbody>
</table>
List of Tables

3.1 Coordinate frames description ................................................ 21
3.2 Hardware Model Inputs and Outputs ....................................... 24
3.3 Software Numerical Model Inputs and Outputs .......................... 26
3.4 Essential Homogeneous Transformation Matrices ........................ 34

5.1 The preliminary - not final - hardware requirements from the results of the numerical analysis model of this thesis, different resources about docking analysis. ................................................................. 56
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISS</td>
<td>International Space Station</td>
</tr>
<tr>
<td>LEO</td>
<td>Low Earth Orbit</td>
</tr>
<tr>
<td>OOS</td>
<td>On-Orbit Servicing</td>
</tr>
<tr>
<td>HIL</td>
<td>Hardware-In-the-Loop</td>
</tr>
<tr>
<td>CSA</td>
<td>Canadian Space Agency</td>
</tr>
<tr>
<td>ESA</td>
<td>European Space Agency</td>
</tr>
<tr>
<td>NASA</td>
<td>National Aeronautics and Space Administration</td>
</tr>
<tr>
<td>APAS</td>
<td>Androgynous Peripheral Assembly System</td>
</tr>
<tr>
<td>ROS</td>
<td>Russian Orbital Segment</td>
</tr>
<tr>
<td>NDS</td>
<td>NASA Docking System</td>
</tr>
<tr>
<td>IDSS</td>
<td>International Docking System Standard</td>
</tr>
<tr>
<td>IBDM</td>
<td>International Berthing and Docking Mechanism</td>
</tr>
<tr>
<td>EPOS</td>
<td>European Proximity and Operations Simulator</td>
</tr>
<tr>
<td>DEOS</td>
<td>DEutsche Orbitale and Servicing Mission</td>
</tr>
<tr>
<td>DOF</td>
<td>Degrees Of Freedom</td>
</tr>
<tr>
<td>w.r.t.</td>
<td>With Respect To</td>
</tr>
<tr>
<td>FTR Table</td>
<td>Forces and Torques Reaction Table</td>
</tr>
<tr>
<td>CoM</td>
<td>Center of Mass</td>
</tr>
<tr>
<td>MOI</td>
<td>Moment Of Inertia</td>
</tr>
<tr>
<td>ODE</td>
<td>Ordinary Differential Equation</td>
</tr>
</tbody>
</table>
### Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta_d$</td>
<td>penetration distance</td>
<td>m</td>
</tr>
<tr>
<td>$\theta_{cone}$</td>
<td>Cone angle</td>
<td>rads</td>
</tr>
<tr>
<td>$I$</td>
<td>mass moment of inertia tensor</td>
<td>rads</td>
</tr>
<tr>
<td>$R$</td>
<td>Displacement</td>
<td>m</td>
</tr>
<tr>
<td>$v$</td>
<td>linear velocity</td>
<td>m \cdot s^{-1}</td>
</tr>
<tr>
<td>$\omega$</td>
<td>angular velocity</td>
<td>rad \cdot s^{-1}</td>
</tr>
<tr>
<td>$a$</td>
<td>linear acceleration</td>
<td>m \cdot s^{-2}</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>angular acceleration</td>
<td>rad \cdot s^{-2}</td>
</tr>
<tr>
<td>$L$</td>
<td>linear momentum</td>
<td>N \cdot sec</td>
</tr>
<tr>
<td>$H$</td>
<td>angular momentum</td>
<td>N \cdot m \cdot s</td>
</tr>
<tr>
<td>$F$</td>
<td>Force</td>
<td>N</td>
</tr>
<tr>
<td>$M$</td>
<td>Moment</td>
<td>N \cdot m</td>
</tr>
<tr>
<td>$T$</td>
<td>Torque</td>
<td>N \cdot m</td>
</tr>
</tbody>
</table>
This thesis is dedicated to my parents, Nashaat and Nahed, for everything you have done for me in my life.
Chapter 1

Introduction

1.1 Background and Motivation

On-Orbit Servicing

Figure 1.1: International Space Station (ISS) has been in the Low Earth Orbit (LEO) at altitude of 330 km since 1998. It is a joint project among five space agencies. ISS undergoes docking regularly when a module is sent to it. It also undergoes on-orbit servicing by its astronauts on board. [1]
Satellite docking is the most critical system in the servicing of an on-orbit satellite. On-Orbit Servicing (OOS) is an important research topic that has already been put into action, and lately has become realistic and promising for the space missions. Although spacecraft servicing has been totally neglected as its technology has been too expensive or has not been feasible, the interest of the space programs of US, Japan, Canada, Europe, Germany and China increased as they realized its importance for the future of space. ISS (International Space Station) - Figure 1.1 - is the best example for docking and on-orbit servicing. It often undergoes docking by the modules that are regularly sent to it to supply the astronauts and the station with the required supplies, materials and experiments. It also undergoes on-orbit servicing by the astronauts on board.

**On-Orbit Servicing Types and Objectives**

On-Orbit Servicing can be divided into two major types: Manned missions and Unmanned missions. The manned mission is carried out by at least one astronaut who goes out for a space-walk and starts to fix the issue. If you are an astronaut, on one hand you will be enjoying your space-walk, on the other hand, you won’t enjoy the amount of concentration you have to put into this task as it needs a lot of accuracy and it might last up to 8 hours! An example of a manned mission is shown in Figure 1.2.

The Unmanned mission is carried out by a robotic application. In most of the cases, a specific robotic manipulator is used to carry the experiment out. The Unmanned missions are more accurate, more efficient, have less risks, and many other advantages. Some of the examples of the Unmanned missions are ROTEX (1993), GETEX/ETS-VII (1999), OLEV (Orbital Lifetime Extension Vehicle) and DEOS (DEutsche Orbital Servicing Mission). On-Orbit Servicing has several objectives which includes:

- Life time extension
- Repair, maintenance and assembly tasks
- Refuelling
- Fleet management
- Disposal of space debris

Many valuable and expensive satellites have been left in wrong orbits uncontrolled and non-functional. While some others have ran out of fuel, even though they are still functioning properly. Not only that the satellite is not functioning anymore or it is abandoned uncontrolled in a wrong orbit, but also it acts as a huge space debris. This kind of a space debris (uncontrolled satellite) is considered as a ”Time-bomb”, as the probability that it crashes into another functioning satellite is very high. If this happens, many smaller debris will be created which makes the ”space debris” problem worse.
Chapter 1. Introduction

Figure 1.2: Space debris are one of the main threats against any satellite. The figure on the left shows a cut in the solar panel of the ISS because of the space debris. The other figure (right) depicts the ISS as it undergoes an on-orbit servicing by the astronaut Scott Parazynski. The cut of the solar panel was fixed after a space-walk that lasted 7 hours, 19 minutes. [1]

On-Orbit Servicing Challenges

*The greater the obstacle, the more glory in overcoming it.*

Molière

By researching into the issues of the uncontrolled satellites, space programs and agencies started to prepare for servicing missions to rescue these satellites that lie in different orbits: LEO (Low Earth Orbit) or GEO (Geostationary Orbit). Although several studies are still in progress, there are some challenges that managed to grab the attention of the researchers. Some of these challenges are:

1. Docking: is one of the most important phases of the servicing mission in order to establish a rigid connection between both satellites. Docking has to be done with any satellite, even the ones which do not have docking interface.

2. Rendezvous: in this phase, it has to deal with different satellites which have different equipment on it. Therefore, the rendezvous has to be able to cope with any kind of satellite in space.

3. Simulation on ground: in order to verify the docking and the rendezvous phases, simulations, tests and analyses have to be carried out to ensure a successful mission, as the numerical analysis will not be enough for this phase. Hence, a facility with a hardware has to be built to verify the system.

4. Autonomy of the satellites: missions with less ground communication capabilities require some degree of autonomy on board to accomplish the mission.
Thesis scope
In this thesis, we will discuss some of the main challenges of the on-orbit servicing. Namely, docking and the simulation of it on ground. The novelty of this work is represented in its simplicity. By combining different disciplines such as engineering, math and physics, a multi-body dynamics simulation will be built, tested and visualized in a 3D visualization program. This thesis is targeted at space agencies, space industries, members of universities with research interest into space, students and space amateurs.

It was chosen to research on this topic and to write the master’s thesis on it, because this research helps in the future space missions, space explorations and satellites servicing. Hence, it helps to have a better understanding of our Earth and Universe.

1.2 Problem and Goal Definition

Docking Process
Two satellites are usually involved in an On-Orbit Servicing mission: Chaser and Target. Target is the one which is already in space and needs maintenance or disposal. While the Chaser is the one which performs the necessary actions on the Target. Before carrying out any actions on the Target satellite, a rigid connection between both satellites has to be ensured. This is done by the on-orbit docking. The docking takes place at the end of the approach scenario, in which the two satellites start to approach each others in space. Figure 1.3 depicts a docking example.

Since this process is very critical for the whole mission, an end-to-end verification has to be performed on the docking mechanism that is proposed to verify the success of the process before carrying it out in space. This verification shall be demonstrated by software numerical simulation and a real-time Hardware-in-The-Loop (HIL) simulation. In the software numerical simulation, the multi-bodies dynamic parameters of both satellites are calculated due to the external forces acting on them. The results of the software model act as a guideline for the type of the hardware that will be used in the HIL simulation. Also, it acts as a reference for the results of the hardware model. On the other hand, the HIL simulation will be used to validate and verify the proposed docking mechanism in a real time simulation.

Goal
In the light of what is explained above, the objectives of the thesis are described as follows:
Chapter 1. Introduction

Figure 1.3: The left figure depicts the ISS robotic arm while capturing the Orbital Sciences’ Cygnus cargo craft, and preparing it for docking. [2] The robotic arm is developed by Canadian Space Agency (CSA). The right figure depicts the cargo craft after it has been docked. [3]

- Build a multi-body dynamics model of the system.
- Build two different Simulink models of the multi-body dynamics:
  - The first model carries out the numerical analysis simulation, which calculates the dynamic parameters of both spacecraft. It also acts as a guide for the hardware that will be used for the HIL simulation.
  - The second model carries out the Hardware-in-The-Loop simulation (HIL).
- Build a 3D visualization program.
- Define hardware specifications.
- Carry out simulations and analyze their results.

The work has a lot of aspects in common with the satellite of the German Space Agency (DLR) called, ”DEutsche Orbitale Servicing Mission” (DEOS). However, the naming of the spacecraft is different. In DEOS, the Target is called ”Servicer”, while the Chaser is called ”Client”.

1.3 Research Method

In this section, the methods of the research followed in this thesis are explained including the modeling, analysis, simulation and testing of all the sub-systems.

System Overview
To carry out a simulation of the docking mechanism, the main research topic is divided into different sub-topics as follows: multi-body dynamics tool, contact modelling, 3D visualization and hardware. A diagram illustrates the work breakdown structure in Figure 1.4.
Figure 1.4: This figure depicts the objective of the project (docking simulation) and its four sub-systems (e.g., System modeling, 3D Visualization, Multi-body dynamics tool and Hardware)

- Dynamic modeling is one of the biggest and the most important parts of this work. It is considered as the infrastructure of this work. It is used to build a model of the system in order to be able to mimic the real life scenario as close as possible.

- 3D visualization program visualizes both spacecraft in space (rigid bodies) and their docking mechanism. It shows how they move or rotate throughout the docking simulation. Also, it shows the relative motion of the Chaser w.r.t. the Target.

- The hardware is the robotic hardware that will carry out the experiment. It simulates the relative motion of the "Chaser" spacecraft w.r.t. the "Target". It takes its input (position and orientation) from the system dynamics model.

- The multi-body dynamics tool contains the equations of motion of both spacecraft in space and their reaction due to any forces/torques applied on them.

Two models are built to verify the docking mechanism (e.g., hardware model and software model). Although each model serves for a different objective, both of them work together to assure the correct results of the simulation of the docking mechanism. The overview of these two models is described here:

- **Software Numerical Model**
  This model simulates the real life scenario of both spacecraft during the docking process using systems of equations of math and physics. The objectives of this model are as follows:
Chapter 1. *Introduction*

- To get the dynamic parameters of both spacecraft, during the docking process.
- The output (numerical data) serves as a reference for the hardware that will be used in the HIL simulation (i.e., maximum forces, maximum torques and response time).
- The output data serve as a reference for the HIL simulation results.
- Software Numerical model will help to build a scheme of the HIL simulation Experiment.

**Hardware-In-the-Loop Model**

The main objective of the HIL simulation is to provide an end-to-end verification of the docking mechanism. In this model, two hardware applications are included, which represent the two spacecraft.

### 1.4 Thesis Structure

This thesis consists of seven chapters including the required background, motivation and the objectives of the research work in Chapter 1. Chapter 2 provides the related work of the previous work that are done by other researchers in the same area of the research. Also, it contains a review of the latest literature of docking mechanisms, multi-bodies dynamics and HIL simulations. In Chapter 3, the research methodology of all the sub-systems id explained in detail, including any assumptions that were made. Also, it includes the dynamic modeling of the multi-body system, a software numerical analysis model and a hardware model. It includes the main research effort of the thesis, especially in the dynamic modeling. Chapter 4 is concerned with the 3D visualization program and the explanation of its features and techniques. Chapter 5 covers the robotic hardware required to carry out the HIL simulation. It includes the hardware minimum requirements and the required specifications. The experiments and their results are discussed in Chapter 6 including the output graphs of the numerical analysis simulation. Finally, Chapter 7 provides conclusions and summary of the research work as well as the recommendations for the future research interest in this field.
Chapter 2

Literature Review

2.1 Docking Mechanisms History

Docking mechanism is of interest since 1964. Throughout the years new mechanisms and conceptual designs are proposed by the different space programs of the world. The different docking developments and designs are listed chronologically in figure 2.1.

![Family Tree of Docking and Berthing Mechanisms](image)

**Figure 2.1:** A family tree of the different docking mechanisms [4].
Chapter 2. Literature Review

In 1964, Langley Research Center (LaRC) - the oldest field center of NASA - hosted the first docking mechanism simulator of the Gemini VIII mission [16]. The passive/active docking interface - Figure 2.2 - allowed the Gemini spacecraft to dock to the Agena target vehicle. On March 16, 1966, NASA announced that the Gemini-Agena was the first successful manned on-orbit docking mission with the astronauts: Neil A. Armstrong and David R. Scott, on board [17]. The Gemini-Agena docking mission was a preparation for the Apollo project.

By the beginning of the American lunar program, it was not believed that docking would be planned. However, another complex docking mechanism was developed, which is known as Apollo docking mechanism. Its mechanism is a probe & drogue mechanism. This mechanism consists of a probe, which is inserted in a cone shaped receptacle. The docking is accomplished when the probe is captured softly by the latches at the end of the probe as seen in Figure 2.2. Apollo 9 was the first Apollo flight to dock, on March 3, 1969. This docking mechanism successfully allowed the "Apollo Command/Service Module" to dock to the "Apollo Lunar Module" and the "Skylab space station". A docking simulator was developed in LaRC to test this docking mechanism [18, 19].

The first Russian docking took place between two Soyuz spacecraft on January 4, 1969. The docking mechanism used was a probe and drogue docking mechanism, as shown in Figure 2.2. On April 19, 1971, the Russian space station "Salyut" was launched into space. Afterwards, the first docking between Salyut and Soyuz took place on June 30, 1971 in which an updated version of the probe and drogue design was used.

In 1975, a cooperation took place between two nations: America and the then Soviet Russia. Their first project was Apollo/Soyuz. Apollo Soyuz successful docking took place on July 17, using the new docking mechanism called Androgynous Peripheral
Attachment System, APAS-75. This mechanism is called ”Androgynous”, in which any of the spacecraft engaged in the docking, can act as passive or active. Different versions of the Androgynous mechanisms were designed. APAS-89, an upgrade of the APAS-75 was installed on Mir. Mir was a Russian Space Station that began on February 19, 1986. Another updated version was APAS-95, which was used to dock the Shuttle to the Mir space station, and later to the ISS. In Figure 2.3, the three types of the APAS are shown.

More docking mechanisms were developed to cross between the older ones. A ”Hybrid Docking System” was developed by Russia. Its name was derived from the combination of the ”probe and drogue” and ”APAS” systems. It was used by some of the modules of the Russian Orbital Segment (ROS) of the ISS. See Figure 2.4

In 1996, Johnson Space Center began the development of a new docking system called NASA Docking System (NDS), Figure 2.5. However, NASA’s design remained conceptual. The main aim was to implement a standard international docking system.
Currently, the International Docking System Standard (IDSS) \[20\] is released for the public to develop the physical features and the design loads of the standard docking interface.

In 2010, European Space Agency (ESA) started to research the new International Docking System Standard (IDSS). "This standard system builds on the heritage of the Russian developed APAS system (Androgynous Peripheral Attachment System) used for the space shuttle for the hard docking and the innovative soft-capture features of the new NASA and ESA systems", as ESA stated \[21\]. In 2014, the European Space Agency (ESA) announced that a new soft space docking is being researched, called International Berthing and Docking Mechanism (IBDM). Not only will the new European soft docking mechanism follow the international standards of the docking design, it will also sense the forces between the two spacecraft and adapt accordingly. A picture of the IBDM can be seen in Figure 2.6
Chapter 2. Literature Review

2.2 Multi-body Dynamics and Modeling of Docking Mechanisms

The first step for analysis or testing of the docking systems is modeling. Mathematical/Physical modeling of the docking system consists of different aspects. Some of the aspects considered in the modeling are: contact dynamics, stiffness, damping and friction force. The more aspects considered, the more complicated the modeling is. As docking is normally between two bodies, a multi-body dynamical model shall be built.

2.2.1 Equations of Motion

Newton-Euler Dynamics equations are used for the modeling of many docking mechanisms. Researchers who worked on European Proximity Operations Simulator (EPOS) [22], used Newton-Euler differential equations to model the two satellites as rigid bodies in space. Since Newton-Euler differential equations are the basics, they have been widely used by the other docking mechanisms researchers, such as: Canadian Space Agency [23], Chinese Docking system [13]. Although Newton-Euler is a fundamental method for dynamics, it is not very efficient in the heavy programming tasks. Therefore, some other methods are used such as: Lagrange-Euler method, or Kane method. Chinese researchers used Lagrange analytical mechanical theory while studying the probe and drogue docking mechanism [24]. Afterwards, the same Chinese researchers used the Kane method to represent their problem [25]. However, Newton-Euler method is still
enough for the modeling of a docking mechanism, which is not a very computationally-heavy problem.

2.2.2 Modeling aspects

Contact dynamics, stiffness and damping are some of the aspects that are considered by researchers of the docking mechanisms. Stiffness depends on the materials used in the process of the docking. It plays a very important role, as it affects the reaction of the bodies when they collide. For example, in the probe and drogue docking mechanism, the material of the probe and its stiffness affect the docking process. The stiffer the probe, the more impact force will be applied during the collision. While, a probe made of a less stiff material decreases the impact force on the spacecraft, due to the bending of the probe [24, 25]. Because of the high stiffness, simulations can easily become unstable unless the calculations are fast and accurate enough. This was achieved by the Canadian Space Agency (CSA), by using an external contact dynamics tool for their modeling and simulations. [23].

Another method was used by researchers who worked on European Proximity Operations Simulator (EPOS). The ordinary contact dynamics equation was used in the contact model, which is called “spring dashpot” model. This method was used due to its simplicity. In addition to that model, a virtual contact compliance model was added to facilitate the HIL simulation, and to avoid any high impact, which is very dangerous for the hardware [22, 26]. This device is represented in different springs and dampers. The stiffness of these models can be altered by changing the stiffness and damping coefficients in the impact force equation. The aim is to have a softer impact by having more impact time.
Figure 2.7: Apollo manned docking simulator was originally used by the astronauts preparing for Gemini missions. In this picture, the astronaut appears to be docking with a full-scale Apollo Command Module [12]

2.3 Hardware-In-the-Loop Simulation Techniques

Verification of the docking mechanism is an essential step before the docking process takes place in space. Hardware-In-the-Loop simulation has many objectives, above all, the following are the most important:

1. End-to-end verification of the whole system.
2. Ensures system’s reliability in space.
3. Determines the dynamic parameters of the docking mechanism.

In fact, simulating a space system or a space experiment on Earth is not an easy task. Different approaches are used in the HIL simulations. The first approach ever was done by NASA in Langley Research Center (LaRC); a manned simulation using two huge vehicles was carried out in 1964. [16, 19]. Figure 2.7 shows an example of the manned docking simulator. American and Russian scientists developed another simulator using two 6 DOF table (Hexapod), also known as Stewart-Gough platform. Each hexapod system represents a spacecraft which moves to represent the motion of the spacecraft during docking. Figure 2.8, shows the docking system simulator. Also, NASA developed another HIL Simulator using the 6 DOF simulators recently. Followed by Chinese
researchers who also built a docking simulator using Hexapods as seen in Figure 2.8 [13]. However, the size of the 6-DOF hexapods used in both cases was very big. Other researchers used air bearing technique to simulate the friction-less motion, on Earth as it is in space, as seen in Figure 2.9 [24, 25].

Moreover, industrial robots are recently involved in the HIL rendezvous, berthing and docking simulations. US Naval Research Lab used two industrial robots to carry out the HIL rendezvous simulation [27]. Canadian Space Agency (CSA) developed a facility for the testing of Special Purpose Dexterous Manipulator (SPDM) for the contact operations with the ISS [28]. Also, Chinese researchers developed simulation of free-floating robots, to simulate the on-orbit servicing of a satellite [29]. European Space Agency (ESA)
and German Aerospace Center (DLR) developed a huge facility called "EPOS 2.0", Figure 2.10, which is an improved version of the former facility called EPOS (European Proximity Operations Simulator) [30, 31]. EPOS uses two industrial KUKA robots to simulate the rendezvous, berthing and the docking of the satellites.

Also, ESA developed a validation docking test for the new International Berthing and Docking Mechanism (IBDM) by using a single KUKA robot [10], as shown in Figure 2.11. In this facility, they used the industrial robot to represent the motion of the spacecraft during the docking process in order to validate the docking mechanism. However, this test has many limitations. In the test, there is no feedback response of the forces or the torques, which makes it only realistic for very heavy satellites. Therefore, it is very difficult to test docking mechanisms of light satellites using this facility.
All existing HIL simulations for docking mechanisms (e.g., SDS (CSA), IBDM Testing (ESA) and EPOS (DLR)) have limitations. The shortcomings arise from the use of highly rigid industrial hardware for the generation of the multi-body dynamics combined with a dampening element that isolates the simulator hardware from the hardware under test. Such setups are satisfactory for the simulation of very high mass spacecraft as used in human space-flight, in which the contact dynamics leads to negligible back-reaction on the multi-body dynamics. However, for small spacecraft, the back-reaction of the contact on the relative trajectory is significant and many contact parameters (in particular, material friction, damping and stiffness) need to be considered in the simulation at high fidelity in order to represent the true behaviour. And this is the main concern of this thesis.
Chapter 3

Research Methodology

In this chapter, a thorough description of the docking system is explained and illustrated. The division of the system into different sub-systems will be included, along with the reasoning behind the division. Followed by the details of each sub-system and its contribution to the docking mechanism. These details include the inputs and the outputs of each sub-system and the novelty included in it.

3.1 System Overview

![Diagram of spacecraft](image)

**Figure 3.1:** The two spacecraft involved in the docking process are represented as rigid bodies. Many docking parameters are considered during the docking process. Some of these parameters are illustrated in this figure. [14]

As docking takes place between the two spacecraft (Chaser and Target), many factors are taken into consideration (e.g., relative position, relative orientation, relative approaching velocity and relative attitude). To make our problem simpler the bodies of
both spacecraft are considered as rigid bodies. In Figure 3.1 the rigid bodies of both spacecraft with the docking mechanism (probe and drogue) can be seen.

The docking process is accomplished when the probe of the Chaser spacecraft is docked inside the drogue of the Target spacecraft. This process happens after both spacecraft have approached each other with a certain relative velocity, position and attitude angle.

The software programs used in this thesis are: Matlab version “R2013b” and Simulink with only the default tool-boxes. Simulink is used as the main engine that connects all the sub-systems together. It takes different inputs, then performs the process defined inside of it and finally outputs the data in the required forms. Simulink supports real-time simulation, which helps later on to be used as an interface with the hardware included in the HIL simulation. Simulink supports the integration and the differentiation of different equations as well. Also, it handles the state space variable representation. Moreover, it can call external modules. Figure 3.2 depicts the architecture of the docking mechanism with inputs and outputs. However, this figure does not show the exact flow of the data, it shows only how the sub-systems are connected in our system.

The HIL simulation verifies the docking mechanism, by tests and experiments. The approach of the HIL simulation in this thesis is using two hardware systems to represent both spacecraft. However, before carrying out the HIL simulation, one needs reference
data to compare the results to it. Therefore, two models are built in this thesis (e.g., a hardware model (Section 3.2) and a software numerical analysis model (Section 3.3)). The hardware model represents the HIL simulation, while the software numerical model represents the system of equations that the two spacecraft submit to in space.

The output of the software model acts as a reference for the hardware model. Figure 3.3 illustrates the approach used in this thesis to verify the docking mechanism. In this chapter, both models (e.g., hardware model and software numerical model) will be explained in detail.

Before elaborating on the different models and on the modeling aspects, let’s first explain the overview picture of the different coordinate frames assigned.

For any multi-body dynamical system, coordinate frames play a very important role in the simplification of the problem. By arranging the coordinate frames in the proper positions of each of the bodies, the problem is more or less divided into different sections. In our multi-body docking system, 6 different co-ordinate frames are introduced. In Figure 3.4, an overview of the multi-body system is illustrated with the coordinate frames assigned at the proper position.
In Table 3.1, the different coordinate frames are presented. Later in this chapter, specifically in section 3.4.2, a thorough description of the coordinate frames and the methods of transformation are presented.

<table>
<thead>
<tr>
<th>Coordinate frame</th>
<th>Origin</th>
<th>Position</th>
<th>Axes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inertial Frame</td>
<td>$F_0$</td>
<td>World</td>
<td>$(X_0, Y_0, Z_0)$</td>
</tr>
<tr>
<td>Chaser’s Center</td>
<td>$F_1$</td>
<td>Chaser’s Center of Mass</td>
<td>$(X_1, Y_1, Z_1)$</td>
</tr>
<tr>
<td>Probe’s Base</td>
<td>$F_2$</td>
<td>Base of the Chaser’s probe</td>
<td>$(X_2, Y_2, Z_2)$</td>
</tr>
<tr>
<td>Probe’s Tip</td>
<td>$F_3$</td>
<td>Probe’s tip center</td>
<td>$(X_3, Y_3, Z_3)$</td>
</tr>
<tr>
<td>Target’s Center</td>
<td>$F_4$</td>
<td>Target’s Center of Mass</td>
<td>$(X_4, Y_4, Z_4)$</td>
</tr>
<tr>
<td>Drogue (Cone) Base</td>
<td>$F_5$</td>
<td>Base of the Target’s drogue</td>
<td>$(X_5, Y_5, Z_5)$</td>
</tr>
</tbody>
</table>

Table 3.1: Coordinate frames description
Chapter 3. Modeling

3.2 Hardware Model

3.2.1 Overview

The hardware model described here is the actual representation of the Hardware-In-the-Loop (HIL) simulation. First, a short introduction about the HIL simulation is recommended. HIL Simulation is a technique used to test and develop complex embedded system in real-time simulations. HIL simulation mainly consists of:

1. Plant simulation (which contains the mathematical representation of all the subsystems of the platform).
2. Hardware or the platform to be tested.
3. Embedded system.
4. Sensors, which act as an interface between all of the above systems.

3.2.2 Objective

Despite the complexity and the costly process of the HIL simulation, it is the most reliable way to develop and test the docking mechanism before flying in space. Therefore, the main objective of the HIL simulation in our project is to provide an end-to-end verification of the docking process.

3.2.3 Description

As seen in Figure 3.5, the HIL simulation loop of the docking system consists of:

1. A multi-body dynamic model implemented in Simulink (plant simulation),
2. 6 DOF reaction table & Manipulator or 6 DOF Hexapod (hardware system)
3. Forces/Torque sensor (sensor)
4. Robotic hardware controller
5. Application controller

Two hardware systems are included in the hardware model, which represent the two spacecraft. One of the hardware applications is a "forces/torques reaction table" (FTR
Figure 3.5: Hardware-In-the-Loop (HIL) simulation block diagram. Forces and torques due to the contact are transmitted to the multi-body dynamic simulator. Then, the dynamic simulator outputs the relative motion of the Chaser (Manipulator) w.r.t. the 6 DOF reaction table (FTR Table). The loop keeps going until docking is accomplished.

Table), which represents the Target spacecraft. In our case, the table is fixed. While, the other hardware, which represents the Chaser spacecraft, moves according to the relative motion of the Chaser with respect to the Target.

A cone shape, which represents the drogue, is fixed to the FTR table. And, a probe is connected to the manipulator as its end effector.

**Hardware Model I/O**

The inputs of the hardware model are the initialization of physical properties of the two rigid bodies. Also, the Ordinary Differential Equations (ODE) solver used during the simulation should be selected from a list of available solvers. While the outputs of the model are the relative position and relative orientation of the Chaser w.r.t. the Target, at each time-step. A summary of the inputs and the outputs of the model are described in Table 3.2

**3.2.4 Case study**

In simple words, both satellites are getting closer, while facing each other. Now, let’s assume more details, the Target satellite is initially at rest w.r.t. the world frame, and translated 10 [m] at +X direction of the world frame. The Chaser is initially at the origin
of the world frame. Chaser satellite is moving in the positive +X direction with certain velocity of 0.1 [m/s] (world frame). Target is rotated 180° at the +Y axis (body frame). While Chaser is not rotated at any of its axes. To visualize the scenario described above, Figure 3.4 shows the both bodies with the same orientation described above. Therefore, the drogue (cone shape) of the Target (fixed to the FTR table), is always at rest since the table itself is not moving in our experiment. The manipulator (which represents the Chaser) will be initially moving at a velocity of 0.1 [m/s]. The manipulator keeps moving closer to the cone shape (drogue) until collision happens. Contact is interpreted in the from of forces and torques. Meaning, when there are forces/torques input to the reaction table, that means there is a contact. The value of the forces and torques is given as an input to the plant simulation ”docking dynamic simulation”, which then calculates how the satellites will move. Afterwards, it outputs to the manipulator the relative motion of the Chaser (w.r.t the Target). Accordingly, the robotic arm moves to follow the trajectory. As soon as the probe gets inside the drogue’s inner cylinder, the HIL simulation is finished, and docking mission is accomplished.

3.3 Software Numerical Model

3.3.1 Overview

The software model represents the modeling of the docking system and both spacecraft using the mathematical and the physical laws. In other words, the software model is a system that represents most of the important relevant equations that two satellites are submitting to in space. The closer the software model to the real life situation of both satellites, the better it is.
This work considers the most important factors for the representation of the docking mechanisms.

The software model is the most essential, primitive and challenging part of this thesis. Why? Because the software model depends mainly on the dynamic modeling tool. Also, the dynamic modeling tool is the basic structure upon which the whole simulations are based. Therefore, the software model has to be as close to the real-life problem as possible.

### 3.3.2 Objective

The main objectives of the software model can be briefly described in three points:

1. Modeling of the real life docking problem by the implementation of the mathematical and physical laws.
2. Analysis of the physical properties (e.g., velocity, kinetic energy, etc...) of the bodies during the docking simulation.
3. The results of the software model are used as a reference for the HIL simulation.
4. The results of the model is used as a guide for the hardware requirements.

### 3.3.3 Description

In Figure 3.6, an illustration of the software numerical model’s block diagram can be seen. One of the main components of the software numerical model are (the sub-systems of the modeling system): multi-body dynamic system, coordinate frames transformation, collision detection, contact mechanics, stiffness and damping - as seen in Figure 3.7.

The outputs of the software numerical model are exported to Matlab in order to analyze the results. Also, the results are exported to the 3D visualization program to be visualized.

**Software Numerical Model I/O**

As the Software numerical model is using the same multi-body dynamics tool which is used in the hardware model, the inputs are almost the same. Unlike the hardware model, the software numerical model is not a real-time simulation, as the time-step size can be changed. It can be set to variable or manual. If the time-step is set to variable, it changes according to the type of the ODE solver used. However, if it is set to manual, it will be fixed throughout the docking simulation. By decreasing the time-step size
Chapter 3. Modeling

Figure 3.6: The software numerical model is the modeling of the real life problem in mathematical and physical equations. This figure depicts the block diagram of the model.

the software simulation, the resolution of the results are increased. Therefore, the inputs of the software numerical model are: the initialization of the two rigid bodies, their physical parameters, ODE solver type and the simulation time-step size. While outputs are the relative position and relative orientation of the Chaser w.r.t. the Target, at each time-step. Also, most of the physical states throughout the simulation are exported to Matlab for the analysis of the results. A summary of the inputs and the outputs of the model are described in Table 3.3

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Position (World frame)</td>
<td>Relative Position</td>
</tr>
<tr>
<td>Initial Orientation (World frame)</td>
<td>Relative Orientation</td>
</tr>
<tr>
<td>Initial Velocity (Body frame)</td>
<td></td>
</tr>
<tr>
<td>Initial Angular Velocity (Body frame)</td>
<td></td>
</tr>
<tr>
<td>Mass</td>
<td></td>
</tr>
<tr>
<td>Moment of Inertia</td>
<td></td>
</tr>
<tr>
<td>ODE Solver</td>
<td></td>
</tr>
<tr>
<td>time-step</td>
<td></td>
</tr>
<tr>
<td>(If set to manual)</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.3: Software Numerical Model Inputs and Outputs

3.3.4 Case Study

Let’s consider the same case study we have assumed before in the hardware model in section 3.2.4. Both spacecraft (Chaser and Target) are approaching each other in space.
Target satellite is initially at rest w.r.t. the world frame, and translated 10 [m] at +X direction from the origin of the world frame. While, the Chaser satellite is initially placed at the origin of the world frame. The Chaser satellite is moving in the +X direction with certain velocity of 0.1 [m/s] (world frame). Target is rotated 180° at the +Y axis (body frame). While Chaser is not rotated at any of its axes. To visualize the scenario described above, Figure 3.4 shows both spacecraft (shown as rigid bodies) with the same orientation described above.

Now, let’s keep in mind the block diagram of the software model shown in Figure 3.6. After the initial data are set, they are forwarded as inputs to the multi-body dynamics tool. The dynamics tool then calculates the position and the orientation of both satellites at the next time-step. Now, we will fast forward to the point where the satellite are almost getting in contact with each other. After the dynamics tool outputs the position and the orientation, these output data are inserted to the ”collision detection” sub-system, which calculates the relative position or penetration distance between the contacting bodies. The relative distance or the penetration depth is then sent to the contact mechanics sub-system.

The ”contact mechanics” sub-system then calculates the amount of force produced due to the penetration depth, with the consideration of the stiffness and the damping calculated for both of the materials used.

The forces applied at the edge position of both satellites (F3 and F5) are then transformed as the amount of forces and torques applied on the center of mass (CoM) of the satellites bodies (F1 and F4). This is done using the ”force and torque transformation” sub-system.

Afterwards, the applied forces and torques go through the feedback loop to the ”dynamics tool” again, to update the position and the orientation of each satellite.

The results of the whole simulation are exported to the 3D visualization program which helps to visualize the relative motion of the spacecraft in space. Also, these results are exported to Matlab for analysis purposes.

### 3.4 Dynamic Analysis and Modeling

In this section, dynamic modelling is explained. The dynamic modelling represents the core of this thesis, why? Because it is trying to mimic the real life situation of the docking mechanism on Earth as it is in space, which is very challenging. Also, the results of this section is taken as a reference for most of the objectives of this thesis. That makes the dynamic modelling system the biggest and the most important part.
Different sub-systems - as seen in Figure 3.7 - are included in the dynamic modeling system (e.g., rigid body dynamics, coordinate transformations, contact dynamics, collision detection and stiffness and damping system). In this section, all the dynamic modeling sub-systems are explained in detail.

3.4.1 6 DOF Equations of Motion

3.4.1.1 Newton-Euler Equations of Motion (Theoretical Background)

Newton’s equations of motion described the translational motion of a point particle due to any external forces acting on that point particle. However, it did not include the rotation motion, as rotation is irrelevant to a point particle (i.e., even though it rotates, it remains a point). Therefore, Euler introduced his laws (Euler’s laws) as an extension to Newton’s laws. Euler’s equations of motion extended Newton’s laws, from a point particle motion to a rigid body motion.

The motion of a rigid body is decomposed into two types: translational motion and rotational motion. Both types of motions have to be with respect to a point which is fixed to the rigid body. The equations are much more simple when this point of reference on the rigid body is the center of mass (CoM) of the rigid body.

The dynamic equations of the rigid bodies in this thesis are represented using Newton-Euler equations of motion. Newton-Euler equations describe the dynamics system in terms of force and momentum. It is derived by the direct representation of Newton’s second law of motion.

Derivation of Newton-Euler Equations

In this section, we will derive the Newton-Euler equations of motion. Assume that $I_C$
Chapter 3. Modeling

represents the mass moment of inertia tensor of the body on its center of mass (C), in
the body frame. Then, the angular momentum vector at the center of mass (C) is.

\[ \vec{H}_C = I_C \vec{\omega} \]  
(3.1)

and the linear momentum vector is

\[ \vec{L} = m\vec{v}_C \]  
(3.2)

According to Newton’s second law of motion and Euler’s first law, the sum of the external
forces on a center of gravity of a body equals to the rate of change of the body’s linear
momentum.

\[ \sum \vec{F} = \dot{\vec{L}} \]  
(3.3)

\[ \sum \vec{F} = m\vec{v}_C = m\vec{a}_C \]  
(3.4)

Also, the sum of external moments (Torques) at the center of mass equals the rate of
change of the angular momentum.

\[ \sum \vec{M}_C = \dot{\vec{H}}_C \]  
(3.5)

\[ \sum \vec{M}_C = I_C \vec{\alpha} + \vec{\omega} \times I_C \vec{\omega} \]  
(3.6)

Knowing that the rate of change of linear velocity \( \vec{v} \) equals the linear acceleration \( \vec{a} \),
\( \vec{v} = \vec{a} \). And the rate of change of the angular velocity \( \vec{\omega} \) equals the angular acceleration
\( \vec{\alpha} \), (i.e., \( \vec{\omega} = \vec{\alpha} \)).

Now, let’s use the spatial notation which is preferred due to its simplicity, especially
when dealing with rigid bodies. The Newton-Euler dynamics Equation 3.7 at the frame
which is coincident with the center of mass of a rigid body can be described as follows:

\[
\begin{bmatrix}
\vec{F}_C \\
\vec{M}_C
\end{bmatrix}
= 
\begin{bmatrix}
m \cdot E & 0 \\
0 & I_C
\end{bmatrix}
\begin{bmatrix}
\vec{a}_C \\
\vec{\alpha}_C
\end{bmatrix}
+ 
\begin{bmatrix}
0 \\
\vec{\omega} \times I_C \vec{\omega}
\end{bmatrix}
\]  
(3.7)

where \( E \) is a \([3 \times 3]\) identity matrix, \( \vec{F}_C \) and \( \vec{M}_C \) are \([3 \times 1]\) vectors.

However, the Newton-Euler equation, Eq. 3.7, is only valid when the coordinate frame
coincides with the body’s center of mass. In case we would like to express the equations
again on a frame ”K” which is not coincident with frame ”C”, certain transformations
are used.

\( \vec{ck} \), is the location of the center of mass expressed in the body-fixed frame, i.e., \( \vec{ck} = \vec{r}_C - \vec{r}_K \)
Then the acceleration w.r.t. the center of mass \( (\alpha_C) \) is:

\[
\vec{a}_C = \vec{a}_K + \vec{\alpha} \times \vec{c} + \vec{\omega} \times \vec{\omega} \times \vec{c} \tag{3.8}
\]

And, the net moment at the center of mass is:

\[
\sum \vec{M}_C = \sum \vec{M}_K + \vec{c} \times \sum \vec{F} \tag{3.9}
\]

Then, by substituting Eq. 3.8 into Eq. 3.4 and Eq. 3.9, we get the equations of motion as described by frame K, i.e., not coincident with the center of mass C, as follows:

\[
\sum \vec{F} = m \vec{a}_K - m \vec{c} \times \vec{\alpha} + m \vec{\omega} \times \vec{\omega} \times \vec{c} \tag{3.10}
\]

\[
\sum \vec{M}_K = I_C \vec{\alpha} + m \vec{c} \times \vec{a}_K - m \vec{c} \times \vec{c} \times \vec{c} + \vec{\omega} I_C \vec{\omega} + m \vec{c} \times (\vec{\omega} \times \vec{\omega} \times \vec{c}) \tag{3.11}
\]

As we can see from Eq. 3.10 and Eq. 3.11, the equations look complicated. But thanks to the spatial notation, both equations can be simplified as follows:

\[
\begin{bmatrix} \vec{F} \\ \vec{M}_K \end{bmatrix} = \begin{bmatrix} m \cdot E & -m [\vec{ck}]^x \\ m [\vec{ck}]^x I_C - m [\vec{ck}]^x [\vec{ck}]^x \end{bmatrix} \begin{bmatrix} \vec{a}_K \\ \vec{\alpha} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ \vec{ck} & 1 \end{bmatrix}_{6 \times 6} \begin{bmatrix} m \vec{\omega} \times \vec{\omega} \times \vec{c} \\ \vec{\omega} \times I_C \vec{\omega} \end{bmatrix}_{6 \times 1}
\]

\[
\begin{pmatrix} \vec{F} \\ \vec{M}_K \end{pmatrix} = \text{Spatial Inertia} \begin{pmatrix} \vec{a}_K \\ \vec{\alpha} \end{pmatrix} + \text{Fictitious forces} \begin{pmatrix} m \vec{\omega} \times \vec{\omega} \times \vec{c} \\ \vec{\omega} \times I_C \vec{\omega} \end{pmatrix}
\tag{3.12}
\]

Knowing that \( \mathbf{1} \) and \( \mathbf{0} \) are \([3 \times 3]\) matrices. While, \( [\vec{ck}]^x \) represents the skew-symmetric cross product matrix.

\[
[\vec{ck}]^x = \begin{bmatrix} ck_x \\ ck_y \\ ck_z \end{bmatrix}^x = \begin{bmatrix} 0 & -ck_z & ck_y \\ ck_z & 0 & -ck_x \\ -ck_y & ck_x & 0 \end{bmatrix}
\]

### 3.4.1.2 Rigid Body Satellites Equations of Motions

The dynamic models of the two satellites undergoing the docking process can be expressed using the Newton-Euler equations of motion which are derived in the previous Section 3.4.1.1. The dynamics models can be expressed in the body frame of both satellites (Target and Chaser) at their center of mass. We follow the same title of the coordinate frames that are assigned in our system and described in Section 3.1 and in Figure 3.4. The dynamic models are expressed in the body frames of the Chaser satellite \((F1)\) and the Target satellite \((F4)\) at their center of mass.

Some assumptions are made in our dynamics model to simplify the problem, e.g.,
1. The satellites are assumed to be free floating in space (i.e., the effect of the earth’s gravity is neglected).

2. The celestial mechanics effect is ignored when compared to the contact forces between the two satellites.

3. The orbital frame is assumed to be the inertial frame \((F_0)\) of both satellites during the docking. Because the two satellites are assumed to be in the same orbit when they are relatively close to each other, right before docking. Also, because the orbital dynamics are ignored.

Therefore, the motion of the satellites (rotational and translational) is described in the body fixed frames which are chosen to be at the center of mass of the satellites \((F_1, F_4)\). See Figure 3.4. Unless something else is noted, all the quantities are expressed in the Chaser satellite frame \((F_1)\) and the Target satellite frame \((F_4)\). Also, all the derivatives are w.r.t. the body frame.

Therefore the dynamics of the Chaser satellite are expressed using Newton-Euler equations as follows: \cite{22}

\[
\begin{bmatrix}
F^F_1 \\
M^F_1
\end{bmatrix} = \begin{bmatrix}
m_{ch} \cdot E & 0 \\
0 & I_{ch}
\end{bmatrix} \begin{bmatrix}
\dot{V}^F_1 \\
\dot{\omega}^F_1
\end{bmatrix} + \begin{bmatrix}
\overrightarrow{\omega}^F_1 \times m_{ch} \cdot \overrightarrow{V}^F_1 \\
\overrightarrow{\omega}^F_1 \times I_{ch} \cdot \overrightarrow{\omega}^F_1
\end{bmatrix} \tag{3.13}
\]

where \(\overrightarrow{V}^F_1\) is the velocity of the satellite w.r.t. frame \((F_1)\).

In the same way, the dynamics of the Target satellite model are defined as follows:

\[
\begin{bmatrix}
F^F_4 \\
M^F_4
\end{bmatrix} = \begin{bmatrix}
m_{ta} \cdot E & 0 \\
0 & I_{ta}
\end{bmatrix} \begin{bmatrix}
\dot{V}^F_4 \\
\dot{\omega}^F_4
\end{bmatrix} + \begin{bmatrix}
\overrightarrow{\omega}^F_4 \times m_{ta} \cdot \overrightarrow{V}^F_4 \\
\overrightarrow{\omega}^F_4 \times I_{ta} \cdot \overrightarrow{\omega}^F_4
\end{bmatrix} \tag{3.14}
\]

As mentioned, the input forces and torques in the above equations \((3.14, 3.13)\) are acting on the CoM of the body fixed frames. Therefore, any forces or torques applied to a different frame than the CoM frame have to be transformed into the body fixed frame (CoM). This is done using the rotation matrices and the homogeneous transformations. We will discuss this thoroughly in Section 3.4.2.

### 3.4.2 Coordinate Frame Transformations

In order to specify a position or an orientation or a velocity of a rigid body in terms of different coordinate frames, coordinate transformations are required. Coordinate transformations include rotations and translations. Using different coordinate frames is
also required for a multi-body dynamic model as it simplifies the problem by dividing
the whole scene into different sections. Coordinate transformations help in the HIL
simulation. Especially, in the calculation of the relative motion of the Chaser (robotic
manipulator) w.r.t. to the Target.

### 3.4.2.1 Rotations

Rotation transformation depends on 3 main rotation matrices, which are described be-
low.

**Rotations (Theoretical Background)** There are three rotations that describe the
rotation of a mobile frame (M) around another fixed frame (F), around the three axes.
The mobile frame (M) is originated from the fixed frame (F), by rotating it around one
of the unit vectors of (F). The result from the multiplication of these three rotation
matrices is called a "fundamental rotation matrix" and it is a $[3 \times 3]$ matrix. Following
the right hand rule, the three rotational matrices are described as follows:

\[
R_x(\theta) = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos\theta & -\sin\theta \\
0 & \sin\theta & \cos\theta
\end{bmatrix},
R_y(\theta) = \begin{bmatrix}
\cos\theta & 0 & \sin\theta \\
0 & 1 & 0 \\
-\sin\theta & 0 & \cos\theta
\end{bmatrix},
R_z(\theta) = \begin{bmatrix}
\cos\theta & -\sin\theta & 0 \\
\sin\theta & \cos\theta & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

Where $R_x(\theta)$ represents the rotation of a frame by an angle of $\theta$ around the X-axis.

When a number of fundamental rotation matrices are multiplied together, the resulting
matrix represents a sequence of rotations about the unit vectors. The form of these mul-
tiple rotations is called "composite rotations". The multiplication order of the rotation
matrices is very important, as it is different from one case to another. It was observed
that any two frames can coincide, if they are rotated around at least 2 axes. Therefore,
several certain composites are considered as fundamental. For example, Yaw-Pitch-Roll,
or Euler-Angle-Transformation, etc... Also, it is important to mention that there are two types of composite rotations, de-
pending mainly on the reference axis of rotation (fixed frame or local frame). "Pre-
multiplication" or "Extrinsic rotations" means that the sequence of rotations is per-
formed w.r.t. the fixed frame. While rotation around the local frame is called "Post-
multiplication" or "Intrinsic rotations". The extrinsic rotation is equivalent to the in-
trinsic rotation by the same angles, but with inverted order of the multiplication of the
rotation matrices.
Docking Mechanism Multi-bodies Model Rotations  In the docking multi-body system model, to calculate the "Direction Cosine Matrix" - which maps the world frame (F0) into the body fixed frame (F1 or F4) - from the Euler angles, post-multiplication (intrinsic rotations) is used. The sequence of the rotations is $R_x(\theta_1)\cdot R_y(\theta_2)\cdot R_z(\theta_3)$ which is noted as "Roll-Pitch-Yaw" or "X-Y-Z". Its composite rotation matrix is represented as follows [32]:

$$
DCM_{(XYZ)} = \begin{bmatrix}
C\theta_2 C\theta_3 & C\theta_2 S\theta_3 & -S\theta_2 \\
(S\theta_1 S\theta_2 C\theta_3 - C\theta_1 S\theta_3) & (S\theta_1 S\theta_2 S\theta_3 - C\theta_1 C\theta_3) & S\theta_1 C\theta_2 \\
(C\theta_1 S\theta_2 C\theta_3 + S\theta_1 S\theta_3) & (C\theta_1 S\theta_2 S\theta_3 - S\theta_1 C\theta_3) & C\theta_1 C\theta_2
\end{bmatrix} \tag{3.15}
$$

where $C\theta$ means $\cos(\theta)$.

While in the 3D visualization program, explained in Chapter 4, all rotations are done w.r.t the fixed frame (i.e., extrinsic or pre-multiplication). Therefore, the composite rotation is the inverse of Eq. 3.15. It is called "Yaw-Pitch-Roll" or "Z-Y-X". It is presented as the following rotations $R_z(\theta_1)\cdot R_y(\theta_2)\cdot R_x(\theta_3)$ as follows:

$$
DCM_{(ZYX)} = \begin{bmatrix}
C\theta_1 C\theta_2 & (C\theta_1 S\theta_2 S\theta_3 + S\theta_1 C\theta_3) & -C\theta_1 S\theta_2 C\theta_3 + S\theta_1 S\theta_3 \\
-S\theta_1 C\theta_2 & (-S\theta_1 S\theta_2 S\theta_3 + C\theta_1 C\theta_3) & (S\theta_1 S\theta_2 C\theta_3 + C\theta_1 S\theta_3) \\
S\theta_2 & C\theta_2 S\theta_3 & C\theta_2 C\theta_3
\end{bmatrix} \tag{3.16}
$$

3.4.2.2 Dynamic Model Homogeneous Transformation

Homogeneous Transformations include information about the required rotation and translation of a certain body to describe it w.r.t. any frame. Homogeneous transformation is a $[4 \times 4]$ matrix that includes the rotation matrix $[3 \times 3]$, translation vector $[1 \times 1]$ and some other information $[1 \times 4]$. The homogeneous transformation matrix is represented as follows:

$$
T = \begin{bmatrix}
R_{3 \times 3} & P_{3 \times 1} \\
\eta_{1 \times 3} & \sigma_{1 \times 1}
\end{bmatrix}
$$

where $R$ is the rotation matrix, $P$ is the translational vector, $\eta$ is the perspective vector and $\sigma$ is the scale value.
In the multi-body dynamic docking model, it is enough if the essential transformations and their inverses are calculated. Essential transformations mean all the possible transformations in the multi-body model. Using these essential transformations and their inverses, the transformation from any coordinate frame to another is possible. The five essential homogeneous transformation matrices are \((T_0^1, T_1^2, T_2^3, T_3^4, T_4^5)\). Where \(T_0^1\) means how the coordinate frame (F0) sees the coordinate frame (F1). In other words, it means the transformation of frame (F1) into frame (F0). Table 3.4, describes the five essential transformations, their description and inverses.

<table>
<thead>
<tr>
<th>Title</th>
<th>Description</th>
<th>Inverse</th>
</tr>
</thead>
<tbody>
<tr>
<td>(T_0^1)</td>
<td>Transformation of frame (F1) into frame (F0)</td>
<td>((T_0^1)^{-1} = T_1^0)</td>
</tr>
<tr>
<td>(T_1^2)</td>
<td>Transformation of frame (F2) into frame (F1)</td>
<td>((T_1^2)^{-1} = T_2^1)</td>
</tr>
<tr>
<td>(T_2^3)</td>
<td>Transformation of frame (F3) into frame (F2)</td>
<td>((T_2^3)^{-1} = T_3^2)</td>
</tr>
<tr>
<td>(T_3^4)</td>
<td>Transformation of frame (F4) into frame (F3)</td>
<td>((T_3^4)^{-1} = T_4^3)</td>
</tr>
<tr>
<td>(T_4^5)</td>
<td>Transformation of frame (F5) into frame (F4)</td>
<td>((T_4^5)^{-1} = T_5^4)</td>
</tr>
</tbody>
</table>

| Table 3.4: Essential Homogeneous Transformation Matrices |

**Transformation Diagram**

This diagram describes how to move from one coordinate to another. For example, to describe the relative motion of the Chaser’s probe tip (F3) w.r.t. the base of the Target’s drogue (F5), the position and orientation of probe tip described in (F3) has to be expressed w.r.t. frame (F5) (i.e., \(T_3^5\) has to be calculated). Because the probe tip is defined at the origin of the (F3) and the base of the drogue is defined at the origin of (F5), nothing has to be multiplied to \(T_3^5\). Hence, this is calculated using the correct sequence as follows: \(T_3^5 = T_5^0 \cdot T_0^3 = (T_5^4 \cdot T_4^0) \cdot (T_0^1 \cdot T_1^2 \cdot T_2^3)\).

Figure 3.8, depicts the transformation diagram, with some possible transformations.
3.4.3 Collision Detection

In this section, the collision detection sub-system is explained. As there are two models available (Software and Hardware), two different collision detection sub-systems are built to serve each accordingly. In the hardware model, it is used to determine the point of contact on the drogue, when a collision occurs between both rigid bodies satellites, while in the software model, it is used to calculate the closest point of collision. Also, to calculate the unit vectors of all the forces included in the system.

The main approach in this sub-system is using the geometrical and vectors analysis. Some assumptions are made in this sub-system, which are:

1. The collision between the probe and the drogue occurs only inside the cone shape. This helps in calculating the contact point from the force and torque input in the hardware system.

2. The exact dimensions of both the drogue and the probe are defined. However, a general solution is solved for.

3.4.3.1 Software Model

The collision detection of the software model is based on the calculation of the relative displacement or the penetration distance of the two satellites. Before explaining the method, the inputs and the outputs of the sub-systems are represented first.
Chapter 3. **Modeling**

### Inputs

$\vec{Pin}$: is the relative displacement between the drogue’s base (F5) and the probe’s tip center (F3), as seen in Figure 3.4.

### Outputs

- $C$: Contact point position
- $\delta_d$: relative displacement between probe’s surface and the drogue
- $F_N$: Normal force Unit Vector on the drogue’s surface of the Target
- $F_R$: Reaction force unit vector on the probe of the Chaser
- $F_{fr}^{Chaser}$: unit vector of the friction force acting on the probe
- $F_{fr}^{Target}$: unit vector of the friction force acting on the drogue

### Approach

The idea in this method is to split the analysis views of the cone in two planes: X-Z plane and Y-Z Plane. Although the dimensions are assumed, the proposed solution is for a general solution.

All the variables with "(t)", are changing at each time-step.

**Given**

- $\vec{Pin} = (Pin_x, Pin_y, Pin_z)$
- An assumption that the contact force is inside the cone.

**Solution**

**Plane X-Z**

First, to start analysing the X-Z plane - Figure 3.9. We will have to calculate the outer radius of the frustum $R_{Frustum}(t)$ in the YZ plane (i.e., the radius of the circle created due to intersection of the Y-Z plane with the $\vec{Pin}$). In other words, $R_{frustum}$ changes as long as the position of the probe inside the cone changes. See Figure 3.10

\[
\therefore \text{dist}_x(t) = Pin_x - L_{cyl} \quad (3.17)
\]
\[
\therefore R_{cone}(t) = \text{dist}_x(t) \cdot \tan(\theta_{cone}) \quad (3.18)
\]
\[
\therefore R_{Frustum}(t) = R_{cyl} + R_{cone}(t). \quad (3.19)
\]

**Plane Y-Z**

In this plane, we calculate the closest distance from probe’s center to the surface of the
drogue in (YZ-section).

\[
\therefore P_{inYZ} = \sqrt{P_{in_y}^2 + P_{in_z}^2} \quad (3.20)
\]

\[
\therefore d_{surf}(t) = R_{Frustum}(t) - P_{inYZ} \quad (3.21)
\]

Now, back to X-Z Plane, we calculate the magnitude of the perpendicular distance \(d_\perp\) between the tip of the probe’s center and the drogue’s surface. Afterwards, we subtract the radius of the probe’s tip (hemisphere) to get the (\(\delta_d\)) relative displacement or penetration depth.

\[
\therefore ||d_\perp|| = d_{surf}(t) \cdot \cos(\theta_{cone}) \quad (3.22)
\]

\[
\therefore \delta_d = ||d_\perp|| - 25[\text{mm}] \quad (3.23)
\]

All the above variables are the scalar magnitudes. Therefore, to calculate the vectors, we will multiply these scalar variables with the unit vectors of the desired outputs. The unit vectors are calculated according to the geometry analysis, as follows:
Figure 3.10: The Y-Z plane view of the collision detection using geometrical analysis.

\[ \overrightarrow{f_N} = [-\sin(\theta_{cone}), \cos(\theta_{cone})\cos(\alpha_y(t)), \cos(\theta_{cone})\cos(\alpha_z(t))]^T \]

\[ \overrightarrow{f_R} = [\sin(\theta_{cone}), \cos(\theta_{cone})\cos(\pi + \alpha_y(t)), \sin(\theta_{cone})\cos(\pi + \alpha_z(t))]^T \]

\[ \overrightarrow{f_{Target}} = [\cos(\theta_{cone}), \sin(\theta_{cone})\cos(\alpha_y(t)), \sin(\theta_{cone})\cos(\alpha_z(t))]^T \]

\[ \overrightarrow{f_{Chaser}} = [-\cos(\theta_{cone}), -\sin(\theta)\cos(\alpha_y(t)), -\sin(\theta_{cone})\cos(\alpha_z(t))]^T \]

Where \( \phi_y, \phi_z \) are the angles that are shown in Figure 3.10.

As we notice, the unit vector of the vector \( \overrightarrow{d_{\perp}} \) is the same unit vector of the \( \overrightarrow{F_R} \). Therefore, we get the perpendicular distance vector \( \overrightarrow{d_{\perp}} \) by multiplying the magnitude of the perpendicular distance (\( \delta_d \)) to the unit vector. Afterwards, the contact position on the drogue (\( \overrightarrow{C} \)) is calculated.

\[ \overrightarrow{C} = \overrightarrow{Pin} - \overrightarrow{d_{\perp}} \]  \hspace{1cm} (3.24)

In this collision detection model, three zones are defined in which the values of the above approach change accordingly. These zones are: cone outer zone, cylinder tip zone and docking cylinder zone, as seen in Figure 3.11.
Figure 3.11: Three regions are defined inside the cone during the collision detection program. Each region follows the collision detection approach, however, the values changes according to each region.

Cone outer zone
When the probe is moving in the region between the outer tip of the cone and the docking cylinder, the above approach is working exactly as described without changes.

Cylinder tip zone
When the probe approaches the docking cylinder inside the drogue, the expected collision point is going to be always the tip of the inner docking cylinder. Therefore, in this case, the magnitude of the closest collision point magnitude is calculated using the same approach above, however, the variable $R_{frustum}(t)$ becomes constant and equals to $R_{cyl}$, i.e.,

$$R_{frustum}(t) = R_{cyl}$$

and the variable $\theta_{cone}$ becomes variable at each time-step, in order to make the predicted point of collision always the cylinder tip. One way to understand this is to imagine that
the surface of the cone is tilted at every time-step, such that it remains perpendicular to the probe’s tip center. This is done as follows:

\[
\theta_{\text{cone}} = \arctan\left( \frac{\text{dist}_x(t)}{d_{\text{surf}}} \right)
\] (3.25)

where, \(d_{\text{surf}} = R_{\text{frustum}} - P_{\text{inYZ}}\). Knowing that \(R_{\text{frustum}} = R_{\text{cyl}}\) in this zone. Also, the penetration magnitude is calculated in a different way,

\[
\therefore \|d_\perp\| = \frac{d_{\text{surf}}}{\cos(\theta_{\text{cone}})}
\] (3.26)

\[
\therefore \delta_d = \|d_\perp\| - 25[mm]
\] (3.27)

**Docking cylinder zone**

When the probe gets in the inner cylinder of the drogue, the system of equations stays exactly the same except for the value of \(\theta_{\text{cone}}\) which then becomes \(90[^\circ]\), i.e.,

\[
\theta_{\text{cone}} = 90[^\circ]
\]

Also, the variable \(R_{\text{frustum}}(t)\) becomes constant and equals to the radius of the inner cylinder. i.e.,

\[
R_{\text{frustum}}(t) = R_{\text{cyl}}.
\]

### 3.4.3.2 Hardware Model

The contact point and collision detection sub-system of the hardware model depends mainly on the input forces and torques at each time-step of the simulation. By analysing these input data and the geometry of our system, it is possible to calculate where exactly the collision point on the drogue is. In the next part, the inputs, the outputs and the method of this sub-system are explained. This sub-system is only valid with the assumption that the forces and torques applied to the drogue of the Target satellite are all inside the cone. No outer forces or torques included.

**Inputs**

\(\vec{F}_{C}^{5}\)  Force measured by the sensor fixed at the drogue’s base (w.r.t. frame F5)

\(\vec{T}_{C}^{5}\)  Torque measured by the sensor fixed at the drogue’s base (w.r.t. frame F5)
Chapter 3. Modeling

Outputs

$\vec{C}$  The calculated position of the contact point on the drogue.

Approach

The main approach here is the same as the approach in the software model, which is splitting the analysis views of the cone in two planes: X-Z plane and Y-Z Plane. Simply, plane Y-Z is for the contact position in the circle that represents the drogue. While plane X-Z is for the detection of contact point along the surface of the drogue.

Although the approach is the same as the software, the technique used is totally different. The inputs of this sub-system are vectors of Force ($\vec{F}_C$) and Torque ($\vec{T}_C$). Therefore, it is possible to find the exact location of the contact point on the cone by the analysis of the geometry and the fundamental equation of the torque, i.e.,

$$\vec{T} = \vec{R} \times \vec{F}$$  \hspace{1cm} (3.28)

$$\|T\| = \|R\| \cdot \|F\| \cdot \sin(\pi - \phi)$$  \hspace{1cm} (3.29)

where $\vec{R}$ is the displacement vector and ($\pi - \phi$) is the angle between the force and the displacement vectors.

Although the dimensions of the model are assumed, the proposed solution is for a general solution. Also, all the variables with "(t)" are changing at each time-step.

Given

- $\vec{F}_C = (F_{Cx}, F_{Cy}, F_{Cz})$
- $\vec{T}_C = (T_{Cx}, T_{Cy}, T_{Cz})$
- An assumption that the forces and torques are applied only inside the cone.

Solution

As mentioned before, the system is divided into two different planes for analysis, e.g., X-Z plane & Y-Z plane. First of all, the contact displacement unit vector $\vec{r}(t)$ is formulated from the geometrical description of the model. Contact displacement vector is defined as the vector starting from the base of the cone (located at frame F5) to the colliding point on the surface of the cone. In other words, how the frame F5 sees the contact point. Its unit vector is represented as follows:

$$\vec{r}(t) = \begin{bmatrix} \cos(\phi(t) + \theta_{cone}) \\ \sin(\phi(t) + \theta_{cone}) \cos(\alpha_y(t)) \\ \sin(\phi(t) + \theta_{cone}) \cos(\alpha_z(t)) \end{bmatrix}$$  \hspace{1cm} (3.30)
**Plane Y-Z**

In this plane, the angles $(\alpha_y(t), \alpha_z(t))$ are calculated. These are the angles that the point of collision makes with respect to the Y-axis and the Z-axis respectively, see Figure 3.10. First, the unit vector of the input force ($\vec{f}_C$) is calculated, as follows:

$$\vec{f}_C = \frac{\vec{F}_C}{\sqrt{F_{Cx}^2 + F_{Cy}^2 + F_{Cz}^2}}$$

And then we calculate the angles:

$$\alpha_y(t) = \tan^{-1}\left(\frac{F_{Cy}}{F_{Cx}}\right) \quad (3.31)$$

$$\alpha_z(t) = \tan^{-1}\left(\frac{F_{Cy}}{F_{Cz}}\right) \quad (3.32)$$

**Plane X-Z**

Now, in plane X-Z (as seen in Figure 3.12), the main objective is to find the contact point position. Therefore, the contact point position is known, if the displacement vector
The X-Z plane view of the contact point detection of the hardware model. The main objective is to find the contact point position, knowing only the input forces and torques and the geometrical model dimensions.

$\overrightarrow{R}(t)$ is calculated. We will first define it w.r.t. the base of the cone (Frame F5), then transform it to any frame needed.

\[
\overrightarrow{R}(t) = \| R \| \cdot \overrightarrow{r}_i
\] (3.33)

By combining Eq. 3.33 and Eq. 3.30 together, we get the following equation:

\[
\overrightarrow{R}(t) = \| R \| \cdot \begin{bmatrix}
\cos(\phi(t) + \theta_{cone}) \\
\sin(\phi(t) + \theta_{cone}) \cos(\alpha_y(t)) \\
\sin(\phi(t) + \theta_{cone}) \cos(\alpha_z(t))
\end{bmatrix}
\] (3.34)

$\theta_{cone}$ is a constant which depends on the shape of the drogue (i.e. $\theta_{cone} = \text{atan}(\frac{R_{cone(max)}}{L_{cone}})$). And, $\alpha_y(t) & \alpha_z(t)$ are calculated in Eq. 3.31 and 3.32. Then, the unknowns are the magnitude of the displacement vector $\| R \|$ and the angle $\phi(t)$.

By substituting the inputs ($F_C$ and $T_C$) in Eq. 3.29, we get the value of $(\| R \| \cdot \sin(\pi - \phi)$ which is given a variable name ”M”), as follows:

\[
M = \| R \| \cdot \sin(\pi - \phi) = \frac{\| T \|}{\| F \|}
\] (3.35)
Following what we achieved above, the triangle M-K-R, as seen in Figure 3.13, is a right angled triangle, where \( K \) is always constant. \( K \) represents the distance between the internal surface of the cone and the extended ray from base of the cone with inclination of \( \theta_{cone} \) with x-axis. The constant \( K \) is calculated in Eq.3.38 at the end of this section. Therefore, using the triangulation rules, the angle \( \phi \) can be calculated.

\[
\phi(t) = \arctan\left(\frac{K}{M}\right) 
\]

(3.36)

Also, from the same triangle, the magnitude of the displacement vector can be calculated:

\[
\|R\| = \sqrt{K^2 + M^2} 
\]

(3.37)

Finally, the displacement vector \( \vec{R}(t) \) is calculated by substituting in Eq.3.34 all the calculated variables, e.g. \( \|R\|, \phi(t), \alpha_y(t) \) and \( \alpha_z(t) \)

\( K \), is calculated using the geometrical specification of the drogue, as follows: (See Figure 3.13 for illustration)

\[
\therefore K = K_1 \\
\therefore L_1 = L_{cyl} \cdot \tan(\theta_{cone}) \\
\therefore L_2 = R_{cyl} - L_1 \\
\therefore K = K_1 = L_2 \cdot \cos(\theta_{cone}) 
\]

(3.38)

### 3.4.4 Contact Dynamics

In this section, the different contact dynamics aspects are explained. These aspects include stiffness, damping and friction. They are in action only when a collision starts to occur. It is observed that there is a collision when the relative distance between the probe and the drogue’s surface is more than or equal to zero (i.e., during penetration). These aspects play a very significant role in the whole numerical simulation, as the results of the simulation are directly dependant on them. To be more specific, the magnitude of the rebound force (impact force) of the multi-body system is calculated based on these aspects. Therefore, the output of this sub-system is fed into the two bodies dynamic models, after doing the required transformations, at each time-step.

Different models of contact dynamics have been proposed. However, almost all of these models contain information about contact stiffness, contact damping. The general form
of the equation can be represented in Equation 3.39, as follows,

$$F = f_s + f_d = \underbrace{k_c \delta_d^3}_{\text{Stiffness}} + \underbrace{b_c \delta_d \dot{\delta}_d}_{\text{Damping}} \quad (3.39)$$

The most simple and primitive model is called a "spring-dashpot" model, in which the

$$q = n = g(\delta_d) = 1.$$ This model is a popular choice for many researchers due to its

simplicity. [22, 26, 33, 34]. Therefore, the impact force $F$ is represented as:

$$F = k_c \delta_d + b_c \dot{\delta}_d \quad (3.40)$$

### 3.4.4.1 Hertzian Contact Force

A non-linear force displacement law is the most widely used stiffness model for impact
modelling. The contact stiffness model used in this thesis is Hertz stiffness model. This
model depends on the shape of the contact bodies (e.g., sphere-plane, sphere-sphere,
cylinder-cylinder, etc.), the material of the bodies (Aluminium, Steel, Vespel, etc.) along
with the material’s properties (Elastic modulus, Poisson’s ratio), the penetration dis-
tance and radius of the body.

The objective of this Hertzian contact force is to calculate the magnitude of the impact
force generated during the contact of the probe with the drogue’s surface.

In this thesis, the contact is assumed to be between a sphere (Vespel SP3) and a plane
(Aluminium). In this type of collision, the indentation depth $\delta_d$ is the input while the
impact force magnitude ($F$) is calculated and is given as an output.

The radius of the contact area is directly proportional to the indentation depth $\delta_d$.
Meaning, the deeper the probe is penetrating inside the plane’s surface, the bigger the
contact force is. This can be explained using the following equation,

$$a = \sqrt{2R\delta_d - \delta_d^2} \text{ or } a = \sqrt{2R\delta_d} \text{ when } \delta_d << R.$$ Figure 3.14 depicts the previous equation.

![Figure 3.14: Hertzian contact stiffness parameters][15]
Equation 3.41 represents the Hertz force equation. Where \( f_s \) [N] is the impact force due to the indentation depth \( \delta_d \) [m]. \( R \) [m] is the contact sphere radius and \( (E^*) \) is the factor that contains the information about the the elastic moduli \( E \) and the Poisson’s ratios \( \nu \) of the different materials in contact. To calculate this factor, the following equation is used,

\[
\frac{1}{E^*} = \frac{1 - \nu_1^2}{E_1} + \frac{1 - \nu_2^2}{E_2}
\]

(3.42)

where \( E_1, E_2 \) are the elastic moduli and \( \nu_1, \nu_2 \) are the Poisson’s ratios associated with each body’s material.

In the model presented in this thesis, two materials are used: Aluminium and Vespel SP3. Therefore, Equation 3.42 are calculated according to these material properties. The elastic moduli are \( E_{al} = 68 \cdot 10^9 \) [Pa], \( E_{ve} = 3.275 \cdot 10^9 \) [Pa], while the Poisson’s ratios are \( \nu_{al} = 0.33 \), \( \nu_{ve} = 0.41 \). Afterword, the result \( (E^*) \) is used compute the Hertzian force in Equation 3.41, knowing that the radius of the hemisphere of the probe is \( R = 25 \) [mm].

### 3.4.4.2 Non-linear Contact Damping

Contact Damping is the aspect that helps the bodies to lose some energy during the impact, which is the one of the most challenging parts in contact dynamics part. Without the damping component, introduced in Equation 3.39, the model will be simulating elastic collision of the bodies (i.e., no energy lost due to the collision). Damping is not a straight forward equation that can be used. Its coefficient has to be tuned and verified using experiments and tests. That is why this aspect is challenging!

In the same sense that stiffness depends on the indentation depth \( \delta_d \), damping depends mainly on the relative velocity of indentation \( \dot{\delta}_d \). After trying several models of the non-linear damping, such as [33, 35], the Hunt and Crossley [36] model was used as it fits more within the scope of this thesis.

**Hunt and Crossley Method** A non linear expression for the damping force was originally proposed by Hunt and Crossley [36], where the damping force has the following general form:
Chapter 3. Modeling

\[ f_d = \beta \delta d \delta_d^\gamma \] (3.43)

Such that the force equation including the damping would be as follows:

\[ F = f_s + f_d = k_c \delta_d^n + \beta \delta_d \delta_d^\gamma \] (3.44)

where \( n, \beta \) and \( \gamma \) are constants, which in our case are equal to 1.5, \( k_c/100 \) and 1.4 respectively. The force deformation curve then takes the shape of a hysteresis curve, because of the damping effect of the material. Although the material damping effect in our case is not remarkable, it does create a hysteresis that can be seen in Figure 3.15. In this figure, two curves can be shown: un-damped force curve and damped force curve.

![Figure 3.15: A comparison between two reaction force curves due to the penetration depth of the probe in the drogue’s surface. The hysteresis represents the damping effect on the force (i.e., some energy is lost during collision). While the Hertz force model consider an elastic collision (i.e., no energy is lost).](image)

The hysteresis curve above means that there is some energy lost during the collision. While on the other curve (un-damped one) that is enclosed inside the damping curve, the forces before collision is exactly the same after collision (i.e., no energy losses!)
3.4.4.3 Contact Friction

Friction occurs in all mechanical systems. There are many different components that can be taken into consideration to take care of different aspects of the friction force. Some of these aspects are: velocity, normal force and contact area. The main equation of friction can be described as $F = \mu F_N$.

The main description of the friction is explained by Coulomb friction law. Although Coulomb friction is seen as an ideal model, it has a singularity. It does not specify the friction force for zero velocity, whether it is zero or it lies between $F_C$ and $-F_C$. However, the Coulomb friction model has been used in different researches as the friction compensation, due to its simplicity. Many other different models have been proposed, some of them have been explained in [37].

As the dynamic model of this thesis is a continuous model (non-discrete model), the accuracy of the friction is acceptable and easier to be implemented than a discrete dynamic model. However, to increase the accuracy of the results, a modified Coulomb friction model is presented in this thesis, which has been applied before in the multi-body dynamics software (MSC Adams) [38]. This modified model takes into consideration the static friction coefficient ($\mu_s$), the dynamic friction coefficient ($\mu_d$), and the transition velocity ($V_t$) between both coefficients. In this model, a smooth regularisation is applied in the vicinity of zero relative slip velocity between the contact bodies, in order to avoid the singularity of the original coulomb model and to include the resistive effect of static versus dynamic friction. A description of this modified model is shown in Figure 3.16. When the relative slip velocity exceeds the static transition velocity, the friction coefficient starts to switch gradually from the static coefficient to the dynamic coefficient.
Figure 3.16: A modified Coulomb friction model is presented in this thesis, which takes into consideration both the static ($\mu_s$) and the dynamic friction ($\mu_d$) coefficient. This model also avoids the singularity of Coulomb law, which occurs in the transition between negative and positive slip velocity.
3D visualization is very important for both models (software numerical analysis model and hardware model). Especially, the numerical analysis model, as it needs a visualization program, to visualize the motion of both spacecraft. Each motion - whether translation or rotation - of each spacecraft is a result of the system of equations governing the multi-body model.

The 3D visualization program is built using Matlab from scratch as part of the thesis objectives which are defined in Section 1.2. Figure 4.1 shows an example picture of the visualization program during the docking simulation.

4.1 Matlab 3D Visualization program

The 3D Matlab visualization program is built using Matlab version R2013b. In the following, the description of the inputs, the outputs and the specifications of the program are explained. The program has the following features:

- 3D model of both satellites (Chaser and Target) as rigid bodies.
- Transformation and rotation of all the bodies and parts are included.
- Axes of each satellite (X: red, Y: green, Z: blue) are shown, to make it easier to visualize the orientation of the satellites.
- Visualization of both satellites in space, seen from the world frame.
- Visualization of the relative motion of the Chaser satellite w.r.t. the Target.
- Visualization of the closest point of collision on the drogue.
Figure 4.1: 3D overview of the two rigid bodies of both satellites using the 3D visualization program built using Matlab. The center of mass of each body is marked with the coordinate axes (X: red, Y: green, Z: blue)

The inputs of the 3D visualization program are: the length of the side of the rigid body \( L_{cube} \), position and orientation of each satellites w.r.t the world frame (initial frame) \( (X_0^{Chaser}, \theta_0^{Chaser}, X_0^{Target}, \theta_0^{Target}) \), position of the probe’s base (center of frame F2) with respect to the world frame \( (X_0^{Probe}) \), position of the drogue’s base (center of frame F5) with respect to the world frame \( (X_0^{Drogue}) \), relative position of the Chaser w.r.t. the Target \( (X_{Chaser}^{Target}) \) and the position of the probe’s tip w.r.t. the Target \( (X_{Probelip}^{Target}) \).

On the other hand, the outputs of the program are two Matlab windows (figures) which are updated at each time-step. One represents the view of both satellites in space from the world frame (initial frame F0), while the other one represents the relative motion of the Chaser satellite w.r.t. the Target satellite. In the latter type, visualization is very useful as it visualizes the exact same case of the HIL simulation approach. Meaning, the 6 DOF reaction table (Target) is fixed, while the robotic manipulator (Chaser) shows the relative motion. At each time-step, the two figures are updated. Hence, the position and the orientation of both satellites, and their special parts (e.g., probe and drogue) are updated.

Some of the main challenges of building this 3D visualization program are: building a function to represent the rotation and translation matrices to adjust the pose of the rigid
bodies, representing special shapes of the rigid bodies mathematically, such as: probe of the Chaser satellite (which is represented as a cylinder with a hemisphere attached at the end of the cylinder) and the drogue (which is represented as a cylinder with a frustum of a cone attached to the tip of the cylinder).

In Figure 4.3, we can see the difference between the two views (e.g., normal motion of both satellites in space and the relative motion of the Chaser w.r.t. the Target). In this example, the Chaser (yellow body) is standing still, while the Target rotates 135° around its local Y-axis (green).
Figure 4.3: The 3D visualization program also outputs the relative motion of the Chaser w.r.t. the Target. In these figures we can see the difference between the two types of outputs (e.g., world frame and relative frame) showing the same situation, in which the Target satellite (green) is rotated by 135° around Y-axis, while the Chaser satellite (yellow) is standing still.

The left figure shows the ordinary motion of satellites as seen in space (world frame) in XZ plane. While, the right figure shows the relative motion of the Chaser w.r.t. the Target in XZ plane.
Chapter 5

Hardware System

In this chapter, the different hardware options proposed for the HIL simulation are explained. As mentioned before, the HIL simulation consists of two hardware systems that represent the two rigid satellites. One of these hardware systems is available in the facilities of Airbus DS in Friedrichshafen - where this work took place. It is a 6 DOF forces/torques reaction table (FTR table). The FTR table represents the Target satellite. The table is intended to be fixed and not moving. While the other hardware system, simulates the relative motion of the Chaser w.r.t. the Target. Hence, one of the objectives of this thesis is to define the hardware required to act as the Chaser satellite in the HIL simulation.

The main requirements of the hardware are the capability to withstand the forces and torques applied during the simulation, and the capability to have a reasonable band-width and update frequency that matches the requirements made by software numerical analysis model.

5.1 6 DOF Forces/Torques Reaction Table

The 6 DOF reaction table is connected to Simulink via xPC Target interface. The input of the table is the applied force on its surface. While the outputs are the measured force and torque around the center of the table. The fixed table represents the Target spacecraft, on which the drogue will be fixed to simulate the real life model.
Section 5

5.2 Proposed Robotic Hardware

After a comprehensive survey, involving 6 DOF Stewart platforms as well as industrial manipulator systems, an off-the-shelf solution (or a solution only requiring modest modification) for the HIL docking simulator could not be identified. Consequently, a concept for the HIL simulator needs to be developed starting from the underlying contact dynamics and the required control system design. Appropriate hardware needs to be identified, procured and adapted in terms of control capabilities, a system setup needs to be established and validated.

5.2.1 Robotic Manipulator

One of the approaches to carry out the HIL simulation of the docking mechanism, is to fix the docking probe to a 6 DOF robotic manipulator as an end-effector. While the drogue is fixed on the 6 DOF FTR Table. The docking probe is supposed to approach the drogue by moving the manipulator until it hits the inner surface of the drogue. The manipulator moves according to the relative motion of the Chaser w.r.t. the Target, as the FTR table is fixed. A description of the system can be seen in Figure 5.1.

5.2.2 6 DOF Stewart Platform (Hexapod)

The other approach for the HIL simulation of the docking mechanism, is to fix the docking probe on a 6 DOF Stewart-Gough platform (Hexapod). While the drogue is
fixed on the 6 DOF FTR Table. The docking probe is supposed to approach the drogue by extending the hexapod in the defined position and orientation in order to hit the cone, and try to accomplish the docking. Knowing that, the 6 DOF hexapod platform simulates the relative motion of the Chaser w.r.t. the Target. A description of the system can be seen in Figure 5.1.

5.2.3 Hardware Preliminary Requirements

In this section the preliminary requirements of the hardware system are summarized in Table 5.1. These requirements are a conclusion from the results of the numerical simulation of this thesis, from other studies that were made on this topic in Airbus DS and from an experienced advisor in this field. These requirements are not final, but it is the closest to desired hardware.

<table>
<thead>
<tr>
<th>Description</th>
<th>Numerical Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work Envelope</td>
<td>[X Y Z] = [300 300 300] (mm) *Axes are shown in Figure 5.1</td>
</tr>
<tr>
<td>Rotation</td>
<td>30° around each axis</td>
</tr>
<tr>
<td>Contact Force</td>
<td>4.5 (kN)</td>
</tr>
<tr>
<td>TCP Positioning Velocity</td>
<td>&lt;0.1 (m/sec)</td>
</tr>
<tr>
<td>Positioning Accuracy</td>
<td>&lt;0.3 (mm) in all axes</td>
</tr>
<tr>
<td>Data Interfaces</td>
<td>EtherCat or analogue signals</td>
</tr>
</tbody>
</table>

Table 5.1: The preliminary - not final - hardware requirements from the results of the numerical analysis model of this thesis, different resources about docking analysis.
Chapter 6

Experiment and Results

In this section, the results of the multi-body dynamic simulation are presented, analyzed and explained. After testing and simulating the two rigid satellites for several times, it was possible to define a worst case scenario of the successful docking process. The worst case scenario is the worst relative position/orientation/velocity of both satellites during the close approach scenario such that the docking can be successfully accomplished. A preview of the docking mechanism simulation is available at http://youtu.be/ukSTXFfjER0

6.1 Simulation Scenario

In this simulation scenario, the Chaser is assumed to approach with relative angular displacement of 7.5°, relative displacement of (75, 0, 0) [mm] w.r.t the Target satellite, as shown in Figure 6.1. The approach relative velocity of the Chaser w.r.t. the Target in this case is (0.1, 0, 0) [m/s] represented in the Chaser frame axes.
Chapter 6. Experiments and Results

Figure 6.1: The worst case scenario of the docking approach is described in the figure above, with an approaching relative velocity of 0.1 [m/s]. The red sign ‘x’ represents the closest point of collision between the probe and the drogue.

Both satellites are assumed to be rigid bodies of a cube shape, as seen in Figure 6.2.

**Chaser** The COG of the Chaser satellite is (−1977, 0, 0) [mm] w.r.t. frame (F3), which lies at the center of the probe tip. The length of the probe is 230 [mm], and the radius of the tip of the probe is 25 [mm]. The mass of the Chaser satellite is assumed to be 350 [kg]. Finally, the Moment of Inertia (MOI) of the Chaser satellite ($I_{xx}, I_{yy}, I_{zz}$) is assumed (118, 96, 92) [kgm$^2$].

The initial position of the Chaser is (−3.0651, −0.3331, 0) [m] in the inertial frame, shown in Figure 6.2, and the initial velocity is (0, 0, 0) [m/s] in the body frame. Also, the initial orientation is (0, 0, 7.5) [°] in the inertial frame, and the initial angular velocity is (0, 0, 0) [rad/s] in the body frame.

**Target** The COG of the Target satellite is (−860, 0, 0) [mm] w.r.t. frame (F5), which lies at the base of the drogue. The drogue consists of a cone and a cylinder, their heights are assumed to be: 195 [mm] and 30 [mm] respectively. Therefore, in total, the height of the drogue (cone and cylinder) is 225 [mm]. The radius of the cylinder is 35 [mm]. Also, the radius of the cone base is the same as the cylinder’s radius 35 [mm]. While,
the top radius of the cone is 125 [mm]. The mass of the Target satellite is assumed to be 759 [kg]. Finally, the Moment of Inertia (MOI) of the Target satellite \((I_{xx}, I_{yy}, I_{zz})\) is assumed as \((720, 791, 990) [kgm^2]\).

The initial position of the Target is \((0, 0, 0) [m]\) in the inertial frame, shown in Figure 6.2, and the initial velocity is \((0, 0, 0) [m/s]\) in the body frame. Also, the initial orientation is \((0, 180, 0) [^\circ]\) in the inertial frame, and the initial angular velocity is \((0, 0, 0) [rad/s]\) in the body frame.

Simulink is used to carry out the simulation of the software numerical docking model. The time-step of the simulation is set to variable, with a maximum time-step size of 0.0001 [s]. Meaning, when a variable difference between two consecutive time-steps is relatively small, the time-step size decreases. While, when the difference is big relative to the tolerance value, the time-step size increases with a maximum value of 0.0001 [s].

6.2 Simulation Results

6.2.1 Contact Trajectory

Figure 6.3 shows the trajectory of the probe tip motion inside the drogue and the multiple contacts that occurred. These contacts are shown from the inertial frame (F0) point of view. The probe tip motion was only in the X-Y plane, with rotation around the Z axis, since the contact forces were detected only in only the X-axis and the Y-axis.
6.2.2 Relative Velocity

In this section, the relative velocity of the Chaser w.r.t. the Target during the docking simulation are shown. All velocities are shown in the axes of the inertial frame (F0), which are shown in Figure 6.2.
6.2.3 Velocity

In this section, the velocity of both satellites during the docking simulation are shown and analyzed. All velocities are shown w.r.t. the body frame of each satellite.

**Chaser** The velocity of the COG of the Chaser satellite in the body frame (F1) is shown in Figure 6.5. It is observed that the velocity in X-axis is always negative, this is because all the collisions acted with a reaction force in negative X-direction on the Chaser. While in the Y-axis, the velocity is always fluctuating from positive to negative and vice versa. The reasons is that for every collision, the reaction force acts in a different Y-direction.
Chapter 6. *Experiments and Results*

**Figure 6.5:** Velocity of the Chaser satellite in body frame throughout the docking simulation.

**Target**  The velocity of the COG of the Target satellite in the body frame (F4) is shown in Figure 6.6. In this case, it is also observed that the velocity in X-axis (body frame) is always negative, this is because for every collision, the Chaser pushes the Target backward (body frame), i.e., imposing more velocity to it in the -ve X direction of its local frame. While in the Y-axis, the velocity is also fluctuating like the Target. Moreover, the signs of the velocities of both satellites in the Y-direction are always opposite to each other at the same time-step.
6.2.4 Angular Velocity

In this section, the angular velocity of both satellites during the docking simulation are shown and analyzed. Angular velocities are shown in the body frame. As the motion is observed only in the X-Y plane, the rotation occurs only in the Z-axis of each satellite. Moreover, the signs of the angular velocities of both satellites are always opposite to each other at the same time-step.

**Chaser**  The angular velocity of the Chaser satellite is shown in Figure 6.7. The velocities are shown w.r.t. the local body frame.
Figure 6.7: Angular velocity of the Chaser satellite in body frame throughout the docking simulation.

**Target** The angular velocity of the Target satellite is shown in Figure 6.8. The velocities are shown w.r.t. the local body frame.

Figure 6.8: Angular velocity of the Target satellite in body frame throughout the docking simulation.
6.2.5 Forces

In this section, the forces which acted on the CoM of each of the rigid satellites due to the different collisions throughout the docking simulation are shown. All the forces are shown in the body frame of each satellite.

It is observed that the forces appear only in the X-axis and the Y-axis, as the initial relative motion was only in the X-Y plane.

Moreover, the forces in the X-axis are always negative, which makes sense, because the initial relative velocity of the Chaser is in positive X-axis of local body frame (F1), while the Target is standing still with its local X-axis facing the Chaser. The axes of this configuration can be seen in Figure 6.2.

Also, the values of the force in the Y-axis are always fluctuating from negative to the positive and vice versa. The reason is that the collision points are always changing from positive side to the lower side of the (local) Y-axis of the Target and then to the positive side again, and so on. Therefore, the forces are also changing accordingly.

**Chaser**  The net forces acting on the CoM of the Chaser satellite are shown in Figure 6.9.

**Target**  The net forces acting on the CoM of the Target satellite are shown in Figure 6.10.

![Figure 6.9: Forces applied on the Chaser satellite during the docking process w.r.t. the local axes of the satellite.](image)
6.2.6 Torques

In this section, the torques applied on the CoM of the satellites because of the several collisions throughout the docking simulation are explained and shown.

The torques applied on the CoM of both satellites were observed only in the (local) Z-axis of each of the satellites. This is because the initial relative motion was only in the X-Y plane.

It is also observed how the torques change from negative to positive and vice versa. This is due to the contact trajectory, in which the point of contact flips continuously from one side of the (local) Y-axis of the Target to the other. Consequently, the torques values change accordingly.

Moreover, it is observed that the sign of the Target’s torque is the opposite of the Chaser’s torque at the same unit of time of the simulation.

**Chaser**  The net torques acting on the Chaser rigid satellite are shown in figure 6.11.
Chapter 6. Experiments and Results

Figure 6.11: Torques applied on the Chaser satellite (in body frame F1) during the docking process.

Target The net torques acting on the Target rigid satellite are shown in figure 6.12.

Figure 6.12: Torques applied on the Target satellite (in body frame F4) during the docking process.
6.2.7 Kinetic Energy

In this section, the kinetic energies of both satellites are analyzed and explained. It is very observable how the kinetic energy changes after each impact. The results are shown in the body frame of each satellite.

**Chaser**  The Kinetic energy of the Chaser satellite is shown in Figure 6.13

![Figure 6.13: Kinetic energy of the Chaser satellite (in body frame F1) throughout the docking simulation.](image)

**Target**  The Kinetic energy of the Target satellite is shown in Figure 6.14.
Also, it is observed that the kinetic energy in the (local) Y-axis of the Target goes back to zero at each impact, as seen in Figure 6.15. This is due to the reason that the potential energy builds up,
Chapter 7

Conclusions and Future Work

7.1 Conclusion

This work focuses on modeling and simulation of an on-orbit servicing satellite’s docking mechanism. The results of the work will be used in order to validate and verify the docking mechanism using a Hardware-In-the-Loop (HIL) simulation. The main conclusions that are found from this work are given as follows:

- Two models are built to serve for the objectives of this work, (e.g., software numerical analysis model and hardware model). The results of the software numerical model will be used as a reference later on for the hardware model during the Hardware-In-the-Loop Simulation.

- Simulink is chosen as the best option to be used as the main engine of the software and the hardware model. Matlab is also used for the initialization of the Simulink simulation and for the analysis of the results from the Simulink model. Moreover, results from the Simulink are used to visualize the 3D motion of both satellites in space, using the 3D visualization program built using Matlab within the scope of this thesis.

- Modeling of the system is the most challenging part of this thesis, it included three main aspects, (e.g., satellites’ equations of motion, collision detection and contact dynamics). After considering different approaches for these aspects, they are built as follows:

  1. Satellites equations of motion are built using Newton-Euler equations to simulate the motion of the satellites in space.
2. Collision detection program is built for this type of the docking mechanism from scratch using Matlab, based on geometrical description of the probe and drogue, and by using geometrical and vector analysis.

3. Contact dynamics included different aspects (e.g., impact stiffness, damping and friction).

- After doing a market survey, two robotic hardware systems are proposed to act as the Chaser satellite in the HIL simulation. The Target satellite is represented using a 6 DOF forces/torques reaction table.

- All the results of the numerical analysis simulation are analyzed and the main findings are:
  
  1. A worst case scenario is defined in which the maximum displacement of the probe from the center of the drogue is 75 [mm] with maximum inclination is 7.5 [$^\circ$] and maximum relative approach velocity of 0.1 [m/s].
  
  2. The maximum axial force acting on the Chaser satellite is 1500 [N]. While the shear force on the same satellite is 4000 [N]. This means that the robotic hardware that will be used in the HIL simulation has to support a maximum force load of 4300 [N].
  
  3. The maximum simulation time-step size used was 0.0001 [s].

### 7.2 Future Work

The recommendations for future work in the scope of this thesis topic can be summarized as follows:

- The whole system of equations can be implemented using State Space Variables which will simplify the model later on, when the control theory comes into consideration in this model.

- Modeling of the contact dynamics should be tested once again with different types of stiffness, damping and friction until the best model can be found.

- Modeling of the probe stiffness should be included as it will affect the reaction of the Chaser due to the impact forces.

- In order to verify the docking mechanism and to have an end-to-end verification, a Hardware-In-the-Loop (HIL) simulation with a model of the two satellites have to be carried out.
Appendix A

Software Numerical Analysis
Simulink Model

Software Numerical Analysis Model
Appendix A. Software Numerical Analysis Simulink Model
Appendix A. Software Numerical Analysis Simulink Model

Chaser Dynamics Block

Target Dynamics Block
Appendix A. Software Numerical Analysis Simulink Model

Equations of Motion Blocks (Newton-Euler)

Transformations Block
Homogeneous Transformations Block

The point here is to output a 4x4 homogeneous matrix from the 3x3 rotation matrix and the 3x1 translation.
Collision Detection Code

```matlab
function [Contact, pt, D_perp, Pin, Distance, F_Reaction_UV_Client, ...
    Friction_UV_Client, F_Normal_UV_Servicer, Friction_UV_Servicer, ...
    Docking_flag] = DistanceToCollision (PinInEdgeCoordinates)

   %% Constants %%
   Pin = [PinInEdgeCoordinates(1); PinInEdgeCoordinates(2); ...
       PinInEdgeCoordinates(3)];
   R_pin = 25.e-3; % Radius of the ... probe's tip
   Docking_flag = 0; % Flag for the ... ending of the docking
   Tip_pt_flag = 0; % Flag for the ... detection of which zone the probe is

   %% Drogue
   ConeAngle = atan(98/195); % Angle of the cone
   L_cyl = 30.e-3; % Length of the ... docking cylinder
   R_cyl = 35.e-3; % Radius of the ... docking cylinder

   %% Variable Data
   X_dis = Pin(1) - L_cyl; % distance between ... pin and drogue's base in Drogue's local X-axis
   R_cone = X_dis.*tan(ConeAngle); % Radius(t) of the ... cone only, without the docking cylinder.
   R_frustum = R_cyl + R_cone; % Radius(t) of ... frustum (Cone and the cylinder)
   Pin_YZ = sqrt(Pin(2).^2 + Pin(3).^2); % magnitude of the ... pin distance from the origin in plane YZ
   d_surf = R_frustum - Pin_YZ; % distance from ... (surface tip of the drogue's circle) to the pin position in YZ plane
```
Appendix A. Software Numerical Analysis Simulink Model

```matlab
% Constraint to get in the docking cylinder zone

tip_pt_constr = L_cyl+(d_surf*sin(ConeAngle));

% Extracting angles
Pin_UV = Pin / norm(Pin); % Unit Vector of the probe
theta_x = real(acos(Pin_UV(1))); % Getting the angle x

% Handling Errors
if (theta_x == 0)
    theta_y = 0;
    theta_z = 0;
else
    theta_y = real(acos(Pin_UV(2)/sin(complex(theta_x))));
    theta_z = real(acos(Pin_UV(3)/sin(complex(theta_x))));
end

% Zone detection
if (Pin(1) ≤ R_pin) % Docking accomplished zone
    disp('Docking accomplished')
    Docking_flag = 1;
else if (Pin(1) ≤ L_cyl) % Docking cylinder zone
    disp('I am in the Docking Cylinder')
    ConeAngle = 0;
    R_frustum = R_cyl;
else if (Pin(1) > L_cyl)
    if (Pin(1) ≤ tip_pt_constr) % Cylinder tip zone
        disp('I am in the Cylinder tip zone')
        Tip_pt_flag = 1;
        R_frustum = R_cyl;
        d_surf = R_frustum - Pin_YZ;
        ConeAngle = atan(X_dis/d_surf);
    else % Cone Outer zone
        disp('I am in the Cone outer zone')
    end
end

% Printing the angles for testing
y = theta_y*180/pi;
z = theta_z*180/pi;
```
%% Calculation of the different Unit Vectors %
F_Reaction_UV_Client = [ sin(ConeAngle) ... 
cos(ConeAngle)*cos(pi+theta_y) ... 
cos(ConeAngle)*cos(pi+theta_z)]; % unit vector of the reaction ... force due to the force applied (Force that will act on the Client)
Friction_UV_Client = [ cos(ConeAngle) ... 
sin(ConeAngle)*cos(theta_y) sin(ConeAngle)*cos(theta_z)]; ... 
% unit vector of the friction force due to the force applied ... (Force that will act on the Client)
F_Normal_UV_Servicer = [-sin(ConeAngle) ... 
cos(ConeAngle)*cos(theta_y) cos(ConeAngle)*cos(theta_z)]; ... 
% unit vector of the reaction force due to the force applied (Force ... that will act on the Servicer)
Friction_UV_Servicer = [-cos(ConeAngle) ... 
-sin(ConeAngle)*cos(theta_y) -sin(ConeAngle)*cos(theta_z)]; ... 
% unit vector of the friction force due to the force applied ... (Force that will act on the Servicer)

%% Calculation of Perpendicular Distance
if Tip_pt_flag == 1
    d_perp_mag = d_surf/cos(ConeAngle);
else
    d_perp_mag = d_surf.*cos(ConeAngle); % The magnitude of ... the perpindicular angle
end

if Tip_pt_flag == 1
    d_perp_UV = F_Reaction_UV_Client; % Unit Vector of ... the Perpendicular distance
else
    d_perp_UV = F_Normal_UV_Servicer; % Unit Vector of ... the Perpendicular distance
end

Trans_mat = [1 0 0 Pin(1);... 
0 1 0 Pin(2);... 
0 0 1 Pin(3);... 
0 0 0 1];

D_perp_UV,tra_temp = Trans_mat* transpose([d_perp_UV(1) ... 
D_perp_UV(2) d_perp_UV(3) 1]);
D_perp_UV,tra = d_perp_UV,tra_temp(1:3); ... 
% Unit vector of Perp. distance after translation
D_perp = d_perp_mag * (D_perp_UV,tra-(Pin)); ... 
% Perp. distance in vector form

%% Return Outputs
Contact_pt = Pin - D_perp; % Contact point ... w.r.t. the drogue base
Distance = d_perp_mag - R_pin; % Closest distance ... to collision
%% Correcting the values
for i = 1:3
    if abs(F_Reaction_UV_Client(i)) \leq \text{eps}
        F_Reaction_UV_Client(i) = 0;
    end
    if abs(Friction_UV_Client(i)) \leq \text{eps}
        Friction_UV_Client(i) = 0;
    end
    if abs(F_Normal_UV_Servicer(i)) \leq \text{eps}
        F_Normal_UV_Servicer(i) = 0;
    end
    if abs(Friction_UV_Servicer(i)) \leq \text{eps}
        Friction_UV_Servicer(i) = 0;
    end
    if abs(Contact_pt(i)) \leq \text{eps}
        Contact_pt(i) = 0;
    end
    if abs(Distance) < 0
        Distance(i) = 0;
    end
end
disp('--------------------')
Contact Dynamics Code

```matlab
function [Force_Magnitude, Servicer_Normal, Servicer_Friction, ... Servicer_Total_Forces, Client_Reaction, Client_Friction, ... Client_Total_Forces, F_Hertz, F_damp, F_Dashpot, F_fr] = ... Contact_Dynamics(du, penetration_distance, penetration_velocity, ... Client_Reaction_UV, Client_Friction_UV, Servicer_Normal_UV, ... Servicer_Friction_UV)

% Constants
R_hemisphere = 25*1.e-3; % Radius of the probe[m]
rho1 = 0.33; % possion's ratio for Aluminum
rho2 = 0.41; % possion's ratio for Vespel
E1 = 6.8e10; % Elastic module of Aluminum [Pa]
E2 = 3.275e9; % Elastic module of Vespel [Pa]
Miu_d = 0.1; % Dynamic friction Coefficient
Miu_s = 0.15; % static friction Coefficient
V_dyn = 5e-4; % The transition velocity from static ... friction to dynamic one

% Stiffness Coefficient
E = 1 / ( ((1-(rho1)^2)/E1) + ((1-(rho2)^2)/E2) );
Kc = 4/3 * E * sqrt(R_hemisphere); % ... Stiffness coefficient
```

% Stiffness & Damping Force Model

if (Penetration_distance < 0) % ...
    If it is not penetrating
    B_d = 0;
    Kc = 0;
    P_D = -1 * (abs(Penetration_distance).^1.5));
    P_DB = -1 * (abs(Penetration_distance).^1.4));
else % ...
    If penetration Started
    B_d = Kc*0.01; % ...
    Damping Coefficient (Hunt and Crossley model)
    P_D = abs(Penetration_distance).^1.5); % ...
    penetration distance^1.5, for calculation of stiffness coefficient
    P_DB = abs(Penetration_distance).^1.4); % ...
    penetration distance^1.4, for calculation of damping coefficient
end

%% ------------------------------------------% % Calculation of the force magnitude % % ------------------------------------------ %
F_Hertz = (Kc * P_D); % ...
    Hertz Stiffness Force Component
F_damp = -1 * (B_d * P_DB * Penetration_velocity); % ...
    Damping Force Component
F_Dashpot = (Kc * P_D) + F_damp; % ...
    Force with stiffness and damping

if F_Dashpot ≥ 0 % ...
    If condition, to assure that no % ...
    negative force is acting on the probe
    Force_Magnitude = F_Dashpot;
else
    Force_Magnitude = 0;
end

%% ----------------------------------------% % Modified Coulomb Friction Model % % ---------------------------------------- %
if abs(A Ve(1)) < V_dyn
    F_fr = Miu_s * sin(3/8 * 2*pi * A Ve(1)/V_dyn);
else
    F_fr = Miu_d;
end

%% ------------------------------------------% % Forces expressed in vector format % % ------------------------------------------ %
Servicer\_Normal = transpose(Force\_Magnitude \ast ... 
\text{Servicer\_Normal\_UV}); \quad \% \text{Chaser Normal Force}

Servicer\_Friction = transpose(F_{fr} \ast Force\_Magnitude \ast ... 
\text{Servicer\_Friction\_UV}); \quad \% \text{Friction on the Chaser}

Servicer\_Total\_Forces = Servicer\_Friction + Servicer\_Normal; \quad \% \text{Total forces on the Chaser}

Client\_Reaction = transpose(Force\_Magnitude \ast ... 
\text{Client\_Reaction\_UV}); \quad \% \text{Target Reaction Force}

Client\_Friction = transpose(F_{fr} \ast Force\_Magnitude \ast ... 
\text{Client\_Friction\_UV}); \quad \% \text{Friction on the Target}

Client\_Total\_Forces = Client\_Reaction + Client\_Friction; \quad \% \text{Total forces on the Target}
Appendix B

Hardware-In-the-Loop Simulation

Simulink Model

Hardware-In-the-Loop Simulation Model
Appendix B. Hardware-In-the-Loop Simulation Simulink Model
Collision Point Detection Code

```matlab
function [Contact_pt, Docking_flag] = Cone.Contact_Det(Torque, Force)

%% Error Handling
Force = [Force(1); Force(2); Force(3)];
Torque = [Torque(1); Torque(2); Torque(3)];

if ( Force (1) == 0 && Force (2) == 0 && Force (3) == 0 )
    Contact_pt = [0; 0; 0];
else
    % Beginning of the...
    contact point detection tool

    % Setting Constants
    L_cyl = 30.e-3; % Length of the...
    docking cylinder
    R_cyl = 35.e-3; % Radius of the...
    docking cylinder
    L = sqrt((R_cyl.^2)+(L_cyl.^2)); % line connecting...
    the cone base with
    % the tip of the...
    docking cylinder
    ConeAngle = atan(98/195); % angle between...
    surface of the cone
    % and the base...
    plane that contains the origin
    L1 = L_cyl*tan(ConeAngle);
    L2 = R_cyl-L1;
    K = L2*cos(ConeAngle); % K is a constant;...
    perp- distance between
    % drogue's inner...
    surface and ext. line from
    % drogue's base...
    with inclination of cone angle
    Force_UV = Force / norm(Force); % Force Unit vector
```
F_angle.x = real(acos(Force_uv(1))); % x = F_angle.x*180/pi

% Angles calculation in Y-Z Plane
F_angle.y = real(atan2(Force_uv(3), Force_uv(2))); % The angle between the point of contact cone & y-axis. Seen from Y-Z plane.
F_angle.z = real(atan2(Force_uv(2), Force_uv(3)));

% M calculation (distance between the contact point and the origin of the cone)
M = norm(Torque)/norm(Force); % The displacement magnitude assuming the phi is 90.

% In other words, this is R_mag/cos(phi).

% Calculate Phi angle ({pi/2 - phi} is the angle between the force and the perpendicular distance)
phi = atan(K/M);

% R unit vector
R_uv = [cos(ConeAngle+phi) sin(ConeAngle+phi)*cos(F_angle.y) sin(ConeAngle+phi)*cos(F_angle.z)];

% displacement Magnitude
R_mag = sqrt((M^2) + (K^2));

% displacement vector
R = R_mag * R_uv;

% Docking condition
if (phi >= (atan(K/L)))
    disp('Docking accomplished')
    Docking_flag = 1;
end

% Contact point
Contact_pt = [R(1); R(2); R(3)];
end

% Testing
Fn_uv = [cos(pi/2+theta) sin(pi/2+theta)*cos(phi_y) ...
    sin(pi/2+theta)*sin(phi_y)];

Fn_mag = 1; % Force = 1 Newton
F = Fn_mag*Fn_uv;
\[ R_{test \_uv} = [\cos(zeta+theta) \sin(zeta+theta) \cos(phi_y) ... \\
\sin(zeta+theta) \sin(phi_y)]]; \]
\[
R_{test \_mag} = 100; % Distance to point of contact = 100 mm
R_{test} = R_{test \_mag} \times R_{test \_uv};
\]
\[
\% Torque\_analysis = \text{cross}(R_{test}, F)
R_{test \_mag\_result} = \text{norm}(Torque\_analysis)/\text{norm}(F)/\sin(pi/2-sigma)\]
Appendix C

Miscellaneous Matlab Codes

Initialization Matlab Code

```matlab
# Appendix C
# Miscellaneous Matlab Codes

# Initialization Matlab Code

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%% -------- Simulink Initialization -----------%%%%
%%%% -------------------------------------------------%%%%
%%%% Initialization of the Simulink models ---------%%%%
%%%% -------------------------------------------------%%%%
%%%% September 2014 %%%%
%%%% Karim Bondoky %%%%
%%%% bondoky@gmail.com %%%%
%%%% -------------------------------------------------%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

1
2 $% Length of the probe [m]
3 $% Length of the Chaser extension, which ...
4 $% of the probe, so that it matches the COG
5 $% Length of half side of the Target's ... cube rigid body [m]
6 $% Length of half side of the Chaser's ... cube rigid body [m]
7 $% Length of the cone [m]
8
9 CLT_ini.pos = -1*[ (20.e-3 + ConeLength + SRVLength + ...
10 = ((CLTLength+PinBaseExt+Pin-Length)*cos(7.5*pi/180))...  
11 = (CLTLength+PinBaseExt+Pin-Length)*sin(7.5*pi/180))...  
12 = 0];
13
14 MOI_CLT = [ 118 0 0;...  
15 = 0 96 0;...  
```
Appendix C. Miscellaneous Matlab Codes

25 0 0 92];
26
27 MOI_SRV = [ 720 0 0;...
28 0 791 0;...
29 0 0 990];
30
31 Mass_Client = 350;
32
33 Mass_Servicer = 759;
Bibliography


[10] Michael Hardt, Carlos Mas, Antonio Ayusi, Daniel Cocho, Luis Mollinedo, Oscar Gracia, and Peter Urmston. Validation of space vehicle docking with the


