ORBITAID AEROSPACE

Proposal Report for Rendezvous, Proximity Operations, and Docking (RPOD) by a 3 Degree Of Freedom On Orbit Simulator

Authors
Sai Govardhan
Shayak Bhadraray

Project Advisor
Sakthi Kumar R
Founder and CEO,
OrbitAID Aerospace

November, 2022

Abstract

This report proposes an implementation of a 3-degree-of-freedom air-bearing satellite simulator testbed. This project proposal aims to implement the current algorithms for on-orbit RPOD operations and optimize the same. The major components of this proposed testbed are:

- A granite table or an epoxy resin mat.
- A pair of satellite simulator robots.
- An onboard camera over the chaser robot to introduce the novelty of vision-based tracking algorithms for RPOD.

The sole purpose of this implementation is to test soft docking between two satellites (chaser and target) and introduce obstacles in the HIL scenarios to further improve the algorithms emulating the real-time RPOD missions. The satellite simulator model is developed with light-weight components to facilitate economical use of fuel on board the compressed air tanks and also represent existing satellite constraints. The camera and Fiber Optic Gyroscope act as inputs to the computing system, where the Cold Gas Thrusters, Air Bearings and Reaction Wheel act as the outputs of the computing system. The computing system in this proposal comprises of a Raspberry Pi computer which is reponsible for executing all actions of the and handling the complete computational complexity of the RPOD simulation for various HIL scenarios presented.

Contents

1 Introduction

Summary: This report is a comprehensive proposal for building an On Orbit Simulator prototype for performing RPOD simulations. The hardware and software architecture of the prototype is designed such that it emulates real-time RPOD missions on three degrees of freedom ie. along the X, and Y axis (Translational) and around the Z axis (Rotational). The aim of the simulations is to ultimately test algorithms for image processing and satellite control systems for multiple scenarios of RPOD with and without obstacles and varying lighting conditions.

1.1 Keywords

DOF - Degree(s) Of Freedom

RPOD - Rendezvous, Proximity Operations and Docking

GNC - Guidance, Navigation, and Control

HIL - Hardware-In-The-Loop

OBC - On Board Computer

FOG - Fiber Optic Gyroscope

RW- Reaction Wheel

RPi- Raspberry Pi

PD -	Proportional Derivative
SS -	(small) Satellite Simulator

List of Figures

List of Tables

2 Literature Review

The authors in paper [1] discuss the small satellite simulator architecture, major software features, the tool's coordinate frames, and the simulator's application for estimating both ground and flight data. The paper then describes how the tool was validated using ground data obtained at the Jet Propulsion Laboratory on the testbed's air bearings and flight data from the Massachusetts Institute of Technology's MicroMAS satellite. The ground modules in this experiment, comprised of spherical and planar air bearings, were shown to provide accurate predictions of ground testing after a tuning process, while the space module was demonstrated to predict values similiar to the MIT MicroMAS on-orbit data. As a result, the simulation may give Small Satellite Dynamics Testbed (SSDT) operators with a way to estimate the performance of their hardware and software for both ground testing and flight.

According to the article in [2] an Air Bearing Floor (ABF) also known as a "flat floor" is a 70' X 98' epoxy surface designed to allow RPOD missions requiring low friction movement of small satellite like test interfaces over a flat surface.

The authors in paper [3] explore the design, construction, and testing of two air-bearing satellite simulators to perform manoeuvring in two linear and one rotational degree of freedom. These air bearing vehicles are outfitted with cameras, Microsoft Kinect sensors, an eight-thruster response control system, a reaction wheel, accelerometers, and different grippers and grasping capabilities.

The paper [4] presents a novel planar air-bearing microgravity simulator developed by the Polish Academy of Sciences' Space Research Centre. This simulator has a wide experiment area (2x3 metres) and independent air-bearings supporting the manipulator. The computer system consists of an ARM-based CPU that controls all of the computation on board before sending it to two Joint

Controller boards.

This study [5] evaluated the performance of RPOD missions of two alternative GNC algorithms based on the Model Predictive Control (MPC) and Inverse Dynamics in the Virtual Domain (IDVD) control in an experimental campaign. For the MPC formulation, a Linear Quadratic method with a Quadratic Programming solver was employed, and for the IDVD approach, a nonlinear Interior Point OPTimizer solver was used. The introduction of a keep-out zone and an entry cone restricts the docking scenarios. Control effort, time to target, and constraint management are among the performance indicators supplied for the experiments and simulations. The trials were carried out on a planar dynamic simulator, with spaceship simulators floating above a granite test-bed surface, resulting in reduced gravity and friction.

The authors in paper [6] presented an air-bearing testbed that was used to develop GNC systems for close-proximity operations in RPOD missions. Here, small satellites float on air bearings above a horizontally levelled granite test-bed and move with two translational DOF (X and Y axis) and one rotational DOF (Z axis) owing to thrusters and reaction wheel actuators. This process allowed a near-frictionless dynamic environment with reduced resultant acceleration. The test operations demonstrated the potential of the testbed and its proposed operations successfully.

For the RPOD of two small satellites in [7], an autonomous GNC based on optical navigation was developed, simulated, and tested. Two free-floating small satellite simulators move across a two-dimensional (testbed). The chaser satellite is equipped with a camera to identify its location coordinates and attitude in relation to the target by capturing and processing images related to a visual feature on the target. The approaches for reaching sufficiently fast computation times have been presented, as well as the measurement accuracy performance is described in detail. The two cases for stationary target and a gently falling target were both investigated and resulted in a successful manoeuvre, proving the validity and robustness of the proposed GNC approach.

This paper provides insights and experimental analysis for GNC of autonomous multiple spacecraft construction. This experiment has two translational degrees of freedom and one rotational degree of freedom. Each small satellite participating in the assembly doubles as both a chaser and a target, and equipped with symmetric algorithms. The relative navigation method is based on supplementing the target's state vector by providing the target's control inputs as extra state components. A chaser spacecraft determines the relative location, attitude, and control inputs of a target spacecraft flying in its vicinity using the suggested navigation approach. Through four autonomous three DOF robotic spacecraft simulators floating on a level floor, the suggested methodologies are successfully validated using hardware-in-the-loop testing.

In the study [9], a novel test bed is presented that enables HIL testing of a chaser spacecraft's autonomous approach and docking to a target spacecraft of identical mass. The test bed comprises of a chaser satellite simulator and a target satellite simulator hovering on a level floor through air bearings. The satellite simulators incorporates the prototype docking interface mechanism of DARPA's Orbital Express programme. The chaser spacecraft's relative navigation is achieved by combining signals from a single-camera vision sensor and an inertial measurement unit using Kalman filters. Three infrared light emitting LEDs are mounted on it for relative navigation. The chaser vehicle is moved by eight cold-gas on-off thrusters that are controlled by a non - linear control algorithm based on Schmitt triggers. A response wheel is also employed for vehicle rotation, using a proportional derivative linear control. The experimental findings of an automated proximity manoeuvre and an autonomous docking of the chaser simulator to the non floating target are given.

The thesis [10] presents a 3 DOF satellite simulator model which consists of a granite testbed, a pair of chaser and target satellite simulators and star infrared array. This implementation's aim was to establish soft docking between two satellite simulators using just hardware and systems aboard the satellite simulator. The computation was completely corresponded by the on board computer without any external computational assistance. The satellite simulators use compressed air stored in tanks onboard to power the air bearing and gas thrusters. The air bearing system created a small cushion of air on which the satellite simulator floats, eliminating surface contact and resistance between both the small satellite simulator and the granite table. Multiple Hardware in the Loop simulations were performed to test the capability of this setup, and its functionality was incrementally improved and successfully verified.

3 Overview of Proposed Architecture

The On-Orbit Simulators exercise selected functionality of real-time RPOD missions using small satellites. Due to varying environmental conditions on the ground compared to space, the proposed implementation uses relevant technology to overcome these additional forces, with priority on overcoming the effects of gravity. The main components of the Proposed Architecture are:

Component Name	**Description**
Simulator Testbed	A Granite Top which can be calibrated and is free from surface irregularites
Satellite Simulator Frame	Lightweight and Robust Metal Frame to encapsulate all hardware components of the Small Satellite
Air Bearings	Circular Pads to provide a cushion of air between the Small Satellite Simulator and the testbed to provide translational degree of freedom (Three Units)
Gas Thruster	Gas ejecting nozzle interfaces on the sides of the Satellite Simulator to provide rotational degree of freedom (Eight Units, two on each side except top and bottom)
Gas Tank	Refillable gas container containing cold compressed gas, which is used to move the satellite according to the RPOD control signals

Table 3.1: Components of the Proposed Architecture

Component Name	**Description**
Fuel Tank	Refillable gas container containing cold compressed gas, which is used to refuel the target satellite simulator after docking
Pressure Regulation Unit	Used to Monitor and Regulate the pressure output from Gas Tank sent to Air Bearings and Gas Thrusters uniformly through the Air Distribution Lines
Fiber Optic Gyroscope	A mechanical gyroscope used to collect data on the orientation of the Small Satellite Simulator
Reaction Wheel	An Actutator used to perform motion at higher degrees of precision during close range proximity operations
On Board Computing Unit	Consists of a Data Acquisition system, Raspberry Pi Unit, connecting interfaces and a Cooling Mechanism. This unit is responsible for all the computational activity of the RPOD simulations
Power Unit	Consists of the Rechargable Power Supply with step down mechanism to power components with different input voltage requirements
Docking Interface	The Satellite Simulator Pairs contain the Active and Passive docking interfaces in a cone and spear structure, with an output signal capability to confirm successful soft docking
Camera	A CMOS camera used for visual input during RPOD simulations

Table 3.2: Components of the Proposed Architecture

4 Hardware Architecture

The complete On Orbit Simulation environment comprises of the following Hardware components

1. **Simulation Testbed**
2. **Satellite Simulators**

4.1 Simulation Testbed

The RPOD simulations are performed on a platform that can provide an environment with negligible gravitational drifts and effects.
The following are the requirements to be satisfied by the Testbed, followed by the proposed solutions:

1. **The Testbed surface must be extremely smooth to enable drift free floating of the satellite simulators**
   A highly polished smooth glass, granite, or epoxy surface can be used for the RPOD simulations. The surface is expected to be free from friction caused by surface irregularities. Compared to the glass and epoxy surfaces, it is much more feasible to polish the granite top during regular maintenance sessions. The Epoxy surface alternatively is a much more scalable solution as it is easy to transport, replace or replicate.
2. **The Testbed surface must be perpendicular to the local gravitational vector**
   This can be satisfied by allowing adjustment of the surface on varying degrees of precision. In the case of the granite top, the frame and granite tops can be used to calibrate the surface perpendicularly to the gravity

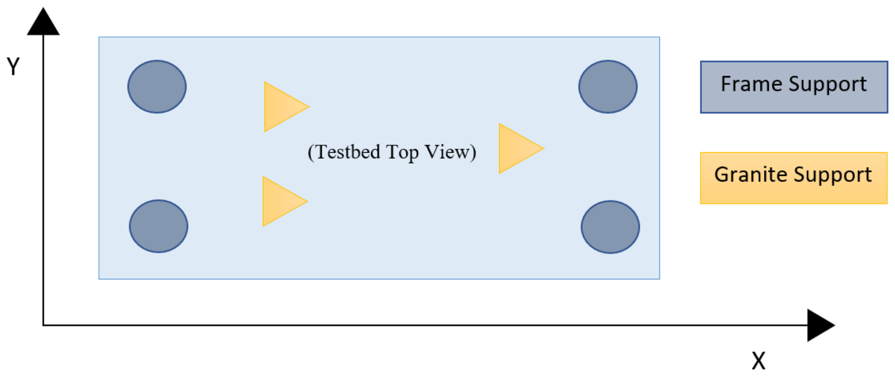

Figure 4.1: Testbed Top View

vector. The testbed's metal Frame Top consisting of four supports can be manually adjusted on both axes (indicated by the circle markers in figure 4.1), and this assists in easily calibrating the table for larger units of precision, where the granite top consisting of three-point supports (indicated by the triangle markers in figure 4.1)
Note that the table's top view can be considered as an X-Y plane, where the Z axis coincides with the local gravity vector when the table is ideally calibrated.

An ideally calibrated Testbed allows the Air Bearing Satellite Simulator to have no drift when floating when stationary and performs drift-free motion in all the 3 degrees of freedom.

4.2 Satellite Simulators

The satellite simulators are used to replicate a pair of servicing and serviceable real satellites performing RPOD missions for different test cases and obstacle avoidance situations.
The Structure of the Satellite simulators is made by using a lightweight metal frame, stable enough to facilitate reliable motion of the satellite simulator and spacious enough to equip all the hardware components.

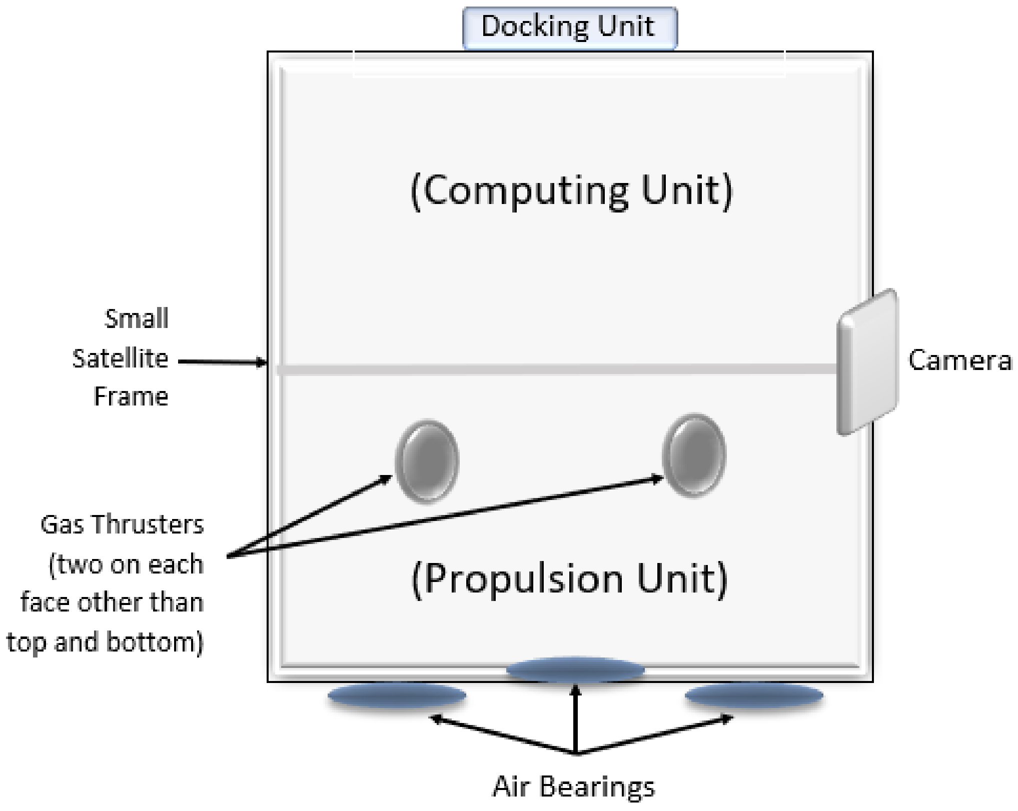

Figure 4.2: Satellite Simulator Basic Structure

Functionally, the satellite simulator comprises a propulsion system, guidance and navigation control system, docking interface, and onboard computing system. Each of these functional units is equipped with hardware systems to enable reliable and accurate simulations concerning space conditions.

The main constraint of a satellite system is its weight, which adds extra expense in terms of costs of launch, fuel consumption, and simulation accuracy. Hence the satellite simulator frame and components are selected in consistency with the purpose of keeping the satellite lightweight and computationally powerful as real-time small satellite systems.

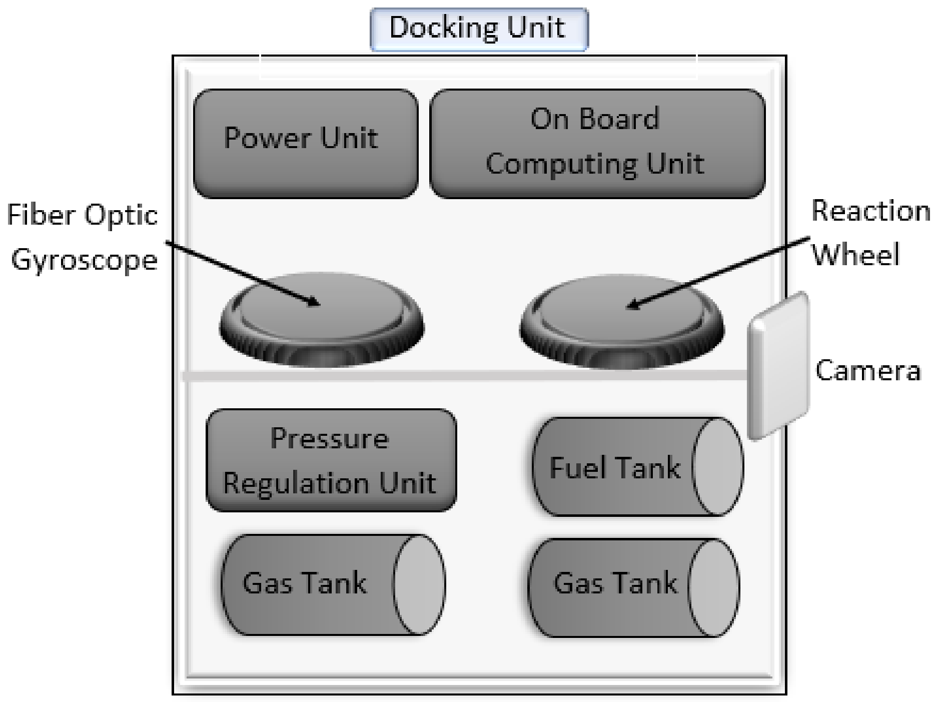

Figure 4.3: Satellite Simulator Functional Composition

Functional Category	**Sub Components**
Propulsion system	Gas Thrusters and Air Bearings
Guidance and Navigation Control	FOG, Reaction Wheel and Camera
Docking Interface	Spear and Cone-Hole structure
On Board Computing System	Raspberry Pi Computer

Table 4.1: Satellite Simulator components categorised functionally

4.2.1 Guidance, Navigation and Control

All the actions performed by the satellite simulator are an outcome of the feedback received by the sensors and the output of the Reaction wheel and actuators. Here the inputs of the satellite simulators are regarding the attitude of the satellite simulator and the visual location of the other satellite it's attempting to dock.
The position of the simulator is perceived using the Fiber Optic Gyroscope (FOG) and the camera. The visual feedback of where the serviceable satellite or obstacle is located is received by a camera. These two components comprise the Data Acquisition System (DAQ) and these inputs are then processed by the On Board Computing System to provide outputs to the air bearings (Translation along the X and Y axis) or gas thrusters(Rotation along the Z axis).

4.2.2 Propulsion System

At the interface with the testbed, the satellite simulators are equipped with three circular pads called Air-Bearings which eject air from two high-pressure gas tanks to maintain a thin layer of the air cushion to enable contact-less friction-free floating on top of the testbed.
The Propulsion system contains two tanks with a maximum capacity of 1 L each, where the air is stored at high pressure to emulate satellite fuel. These tanks store air at a pressure of 20 MPa. The pressure is then stepped down from 20 MPa to 1 MPa along the pressure regulating unit, followed by a further reduction to 0.4 MPa by the second pressure regulating valve. Finally, the air distribution lines
uniformly supply the stepped-down compressed gas to eight gas thrusters and three air bearings.

When the testbed is perfectly aligned, the satellite simulator has complete freedom from additional forces along the X and Y directions, while also allowing complete rotational freedom along the Z axis. The only prominent force on the satellite simulator is along the Z-axis, due to the local gravity vector.

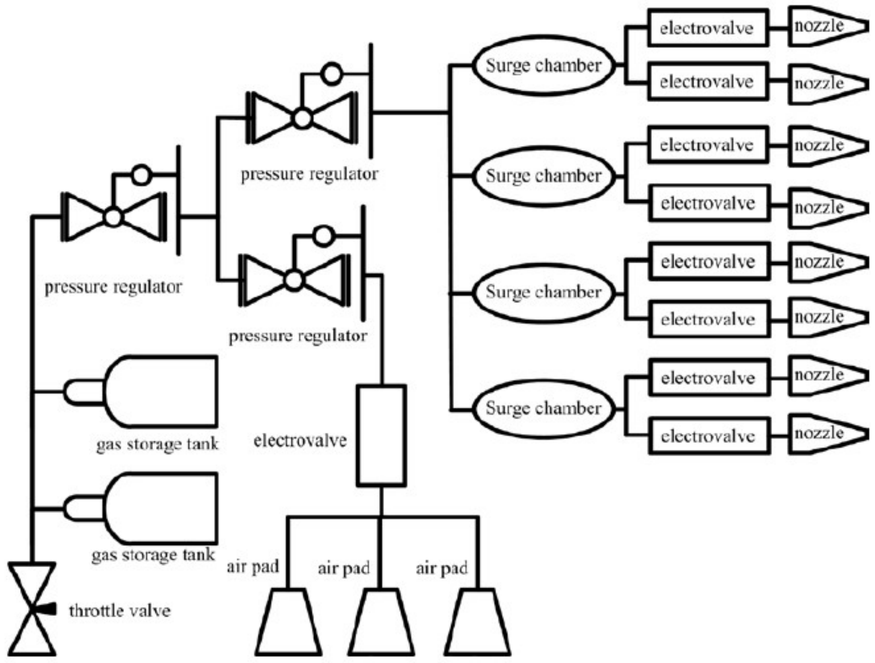

Figure 4.4: Propulsion System Overview

4.2.3 On Board Computing System

The On-Board Computing system receives inputs from the FOG, Reaction wheel, and Camera, for processing the present position and state using all its computing resources on board. The result of this computation is the values sent to the Air Bearings and Gas Thrusters.
The On-Board Computer comprises a Raspberry Pi (RPi) Computer which is powered by USB at 5v, received by the Power Unit.
The Raspberry Pi is a single-board computer that collects inputs from the FOG and camera, to compute the current position of the host satellite during far range and near-range proximity operations. The camera input is processed by the RPi computer to locate the docking satellite in terms of distance and angle to the docking interface.
The computer then sends inputs to the thrusters to move closer to the docking

satellite and continuously monitors the process with feedback from the Camera. The Camera also can detect obstacles in the path between the two satellites, and navigate away from them to prevent collisions. Once the satellite finds a clear path without obstacles in range, it starts to move closer to the other satellite simulator.
As the satellites are closing in, the Reaction Wheel is used to trigger motion in higher ranges of precision in near-range proximity. Once the Docking interfaces of both the host and docking satellite simulators are locked, the docking interface provides feedback to the Computing System, to confirm the completion of the RPOD simulation. This input can be used as an initial signal to perform servicing or fuelling operations between the two satellites.

4.2.4 Docking Interface

The Docking Mechanism contains an active and passive module on either of the Satellite Simulators. They are intended to emulate real-time docking at high levels of accuracy. The Satellite Simulator frames structures have slots for replacing the docking interface component, and for this particular experiment, a Spear and Cone Hole mechanism is deployed. The Spear structure is fixed to the servicing satellite, whereas the Cone Hole structure is attached to the serviceable satellite.
During RPOD simulations with obstacles, care has to be taken to prevent collision of the spear docking module with the obstacle surface, and hence a consistent distance has to be maintained.
After successful docking, a confirmation signal is processed, to display as a convenient visual confirmation.

5 Software Architecture

In the proposed software system, in light of the most recent successful implementations in the exact ventures of RPOD operations on simulator testbeds, we shall choose to implement the required algorithms using LabView. Raspberry Pi as an On Board Computer shall make it easy for us to interface the camera algorithms and the control system of the satellite simulator to be controlled using Python. LabView also provides a necessary interface to issue start-stop commands and set specific parameters for experimental simulations. The following section describes the various other implementations we look forward to implementing in the simulator testbed.

5.1 Control System

This subsection speaks in detail of the proposed implementation for position and attitude control for the satellite simulators in the X - Y plane of the simulator testbed.

5.1.1 Position Control

The proposed control system to propel the satellite simulators on application of their cold-gas thrusters, is a Proportional Integral Derivative (PID) based control system. The thrusters shall give the simulator testbed to get a 3 DOF freedom in both X and Y directions, the parameters for the control system shall be x, y and θ.

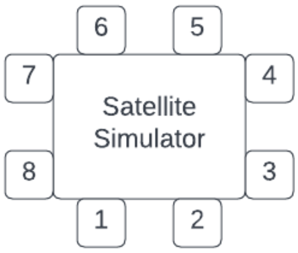

Figure 5.1: Thrusters, numbered over a Satellite Simulator

The 8 thrusters located on the satellite simulator are to be numbered from 1 - 8. The even numbered thrusters will be mapped to rotate the robot clockwise whereas the odd numbered thrusters would be responsible to move it anti-clockwise. Thus a mapping matrix would be devised to establish the relationship between each thruster and the 3 DOF movements.

$$\begin{bmatrix} t_{12} \\ t_{34} \\ t_{56} \\ t_{78} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ -1 & -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} U_x \\ U_y \\ U_\theta \end{bmatrix} \tag{5.1}$$

Since the thrusters can only operate at ON or OFF state; to achieve the desired output for the thrusters in order to ensure a more accurate measure of control, Pulse Width Modulation (PWM) is used.

5.1.2 Attitude Control

The two possible ways which the satellite simulators can achieve precise attitude control by the either the use of a reaction wheel or triggering specific gas thrusters as mapped by the matrix mentioned above (calibrated to precision).

While reaction wheels have proved to be quite precise for smaller angles. One of the drawbacks of using the same is the lack of feedback from the wheel. Having feedback enables the satellite simulator to monitor and adjust the reaction wheel's operation. Joshua Cookson et. al. [/ref] in their proposed simulator performed a comprehensive comparison between the two methods of attitude control for different ranges of angular maneuvering. For smaller angles, reaction wheel had proven to be more more precise. However as the attitude increases the thrusters prove to be more consistent in achieving the target angle and stabilize faster than reaction wheel. Thus the proposed attitude control system shall be a collaborative effort of reaction wheel and gas thrusters to achieve maximum efficiency.

5.2 Path Planning

The two major aspects of the implementation for the Path Planning and Following algorithm to be successful for docking operations are :-

- Positioning System
- Way point Determination

5.2.1 Positioning System

In order for the path planning and following algorithm to work efficiently, the satellite simulator needs to know of its estimated state/position. In a real implementation of RPOD mission, satellites are installed with star trackers in order to create a star based tracking system to estimate the relative position of the chaser satellite and its target w.r.t the stars in space. Joshua Cookson et. al. [/ref] have considered the implementation of an Pseudo Star System by installing IR LEDs in their implementation. Along with an Onboard camera with a bandpass filter to create a positioning system similar to a real star tracking system. However, the same implementation is quite financially high. In our proposed simulator testbed we propose the use of WhyCon marker based positioning and the use of an external overhead camera to track the satellite simulator in the testbed.

5.2.2 Path Planning System

The current position of the satellite simulator shall be estimated by the Positioning System, while the target position and attitude for the docking target is assumed to be known. The subsequent steps for waypoint determination and executing the path following system are shown below.

- Straight line determined from starting to final constructed with waypoints. To ensure a safe distance between the chaser and the target a threshold distance shall be kept between the waypoints.
- System shall then determine the shortest path around a circle with a radius matching the current distance from the target centre.
- Until a minimum decided threshold distance from the target is reached, the points are continued to be plotted.
- Current end position of the circular path means that a line connecting that point with the target's center then passes with through the center of the docking receiver.
- For docking to be successful, the attitude of both chaser and target should be aligned.
- Target attitude is specified at each waypoint, however not used as an active parameter.
- Hence, the satellite simulator can either point along the path's vector or can maintain a constant attitude as it travels over the path.

5.3 Target Pose Estimation

As discussed in the Path Planning section, the target's final position and attitude needs to be known. Previous implementations such as the '*ASTROS*' satellite simulator testbed at Georgia Institute of Technology and the '*POSEIDYN*' satellite simulator testbed at the Naval Postgraduate School demonstrate the same by assuming the final attitude and position of the target docking system. However, in consideration of making a simulator testbed more accurate and closer to the actual in space implementation for satellite Rendezvous, Proximity and Docking operations, we shall implement an Image Processing based Target Pose Estimation model from an on-board camera installed over the satellite simulator. The following section elucidates various implementation techniques for the same.

5.3.1 Visual Marker based Pose Estimation

Earlier implementations in the early 2000s, when the domain of RPOD based satellite exploration missions was at its primordial stage. Target satellites were installed with a visual marker which could be interpreted by an on-board camera from the chaser satellite, in order to extract the geometrical representation of the target.

A brief recapitulation of this method shall be :-

- A visual optical based marker shall be present on the target satellite body.
- The visual marker (shaped or color based) shall be detected by the on-board camera of the chaser vehicle, when the vehicle will be in a specific predetermined proximity distance from the target.
- Simple capture and Image Processing technique is used to interpret and determine the geometrical pose of the target.
- Once the final attitude is determined, the chaser vehicle shall proceed ahead with its Docking Operation.

However, in today's date visual marker based geometric interpretation is now proved to be redundant and inefficient. Illumination Angle, Shadowing, Brightness etc. are potential problems to this implementation. Moreover, estimating geometric pose of the target spacecraft on the basis of prior knowledge of pre-installed visual marker is considered a bad way to solve the pose estimation problem.

5.3.2 Image Processing/Feature Extraction

Moving further with the pose estimation problem, an improved algorithmic approach is to use a feature extraction based geometric pose estimation. In this process a heuristic function is chosen which takes in few specific gradient feature extraction. The extracted information is kept in correspondence with the actual geometric pose of the target spacecraft. A loop condition is initialized to keep the feature extraction and comparison method to keep running until a pre-decided threshold is met for the same. Once the threshold is met, the loop terminates thus confirming the extracted geometric pose.

In order to detect and remove outliers in the interpreted information, various error minimization procedures are executed. Our proposed outlier detection method on the basis of the literature survey conducted for the previous RPOD missions is preferably is '*Gaussian-Newton Minimization.*' The proposed algorithmic flow for the same is shown in the flow diagram below.

A brief recapitulation of the above explained algorithm is as follows :

- A new image is clicked from the on-board camera of the chaser spacecraft, If required certain amount of image pre-processing is performed to enhance the gradient extraction operations over the same.
- Image Processing libraries are used in order to extract features from the image (the most popular choice for the same as demonstrated in the flow above is line detection).
- A heuristic function, determined for comparison of these extracted values takes these gradient extraction parameters as the input in order to produce a suitable quantifiable result.
- Each of these function outputs are in correspondence with an actual pose/configuration of the target spacecraft, Hence, the function's output is used to estimate the pose of the target by choosing a subset of the mapped corresponding poses of the target.
- A certain threshold condition is used to decide the number of iterations for final position estimation of the target. If the threshold is not met, the loop repeats again after clicking a new picture from the on-board camera.

- If the threshold is met, than the loop terminates and the best estimate out of all the recorded estimates is chosen by Gauss-Newton Minimisation, or other equivalent methods.

While Image Processing techniques for pose estimation prove to be superior than the redundancy of using visual markers, the major issue which arises in its implementation is the randomness of illumination and brightness conditions which make it hard for the algorithm to succeed. Thus, in some situations pre-processing is required to enhance the image conditions.

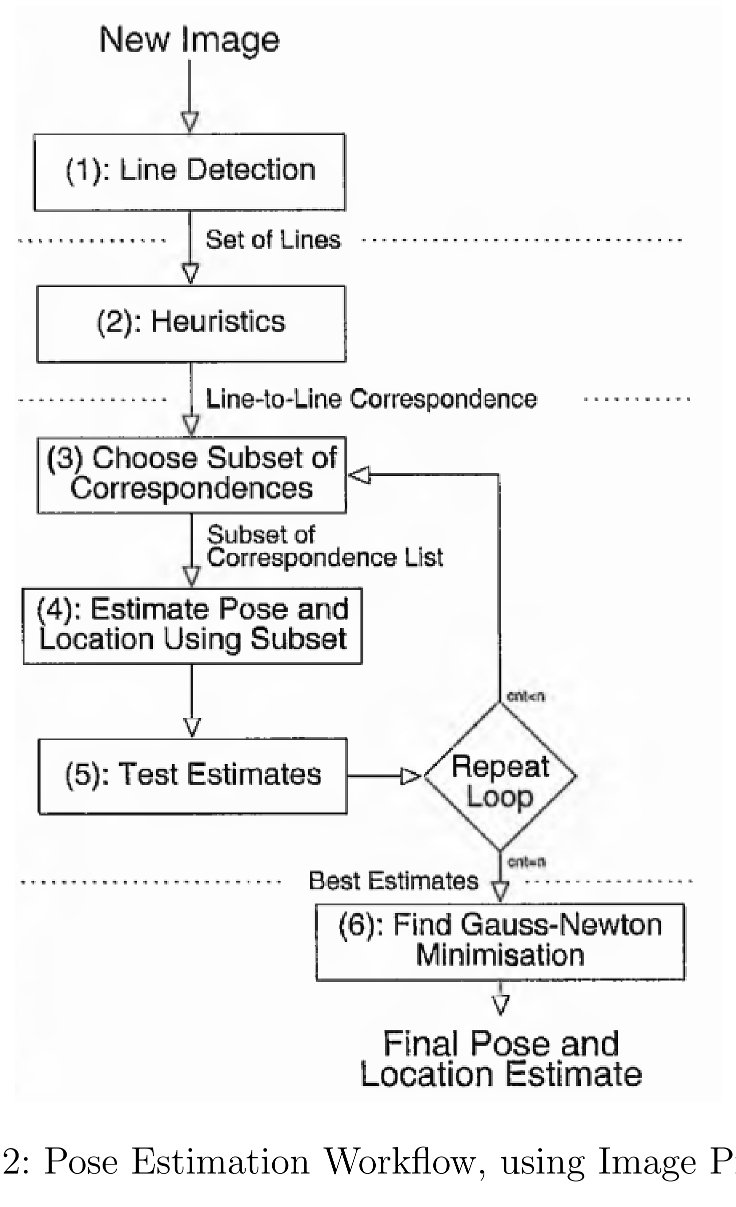
New Image
(1): Line Detection
Set of Lines
(2): Heuristics
Line-to-Line Correspondence
(3) Choose Subset of Correspondences
Subset of Correspondence List
(4): Estimate Pose and Location Using Subset
(5): Test Estimates
Repeat Loop
cnt<n
cnt=n
Best Estimates
(6): Find Gauss-Newton Minimisation
Final Pose and Location Estimate

Figure 5.2: Pose Estimation Workflow, using Image Processing

5.3.3 Machine Learning/Regression Models

In light of the problems encountered by pure Image Processing based pose estimation techniques, this method using pre-trained Machine Learning, Neural Networks and Regression Models comes in picture. On a critical comparison between the previously mentioned estimation techniques, this method is the most robust. Every anomaly which might arise and requires to be catered to by methods of Image Processing due to illumination, angle, lighting, shadowing, brightness etc. is overcome by this implementation. However, this method is a trade off for computational complexity.

Bo Chen et. al. [/ref] have demonstrated the use of regression models in order to predict the estimated pose of the target spacecraft on the basis of the novelty of Neural Networks. The methodology demonstrated by the same is shown in this image below.

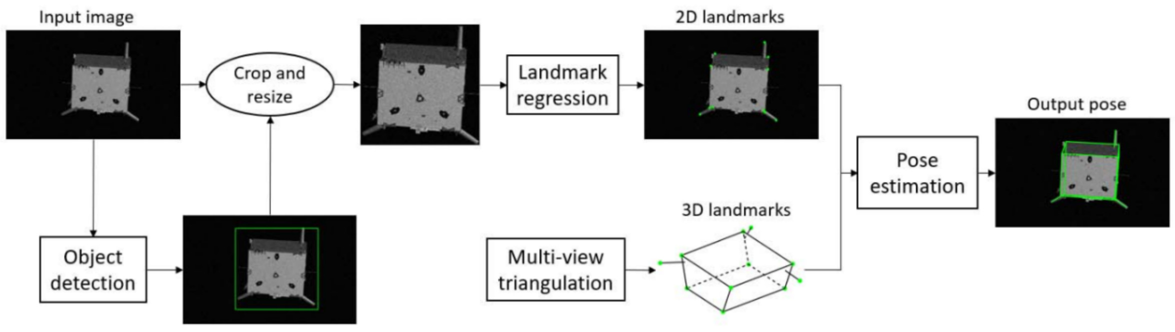

Figure 5.3: Pose Estimation using Regression Models

The primary novelty of the method is that it uses a trained deep network to regress the coordinates of 2D landmarks. Their proposed Convolutional Neural Network model is named as the '*Spacecraft Pose Network (SPN)*.' While their outputs over a critical comparison with feature extraction based processing models for pose estimation is proven to be more efficient. The demand for computational capability is higher for the same. Moreover, their outputs are not produced over a satellite simulator testbed, instead is performed over a live camera feed captured from an on board camera from the chaser spacecraft.

A brief overview of the proposed approach is shown as follows :-

- SPN first predicts the bounding box of the satellite in the image with an object detection sub-network
- Then, a classification sub-network retrieves the n most relevant base rotations from the feature map of the detected object.
- This regression sub-network learns a set of weights and outputs the predicted rotation as a weighted average of the n base rotations.
- Lastly, SPN solves the relative translation of the satellite utilising constraints from the predicted bounding box and rotation.

One of the notable observations which can be devised out from this method is their higher efficiency and accuracy than Image Processing and Feature based extraction methodology. However, the presented outcomes are with respect to actual video camera feeds captured from a chaser spacecraft, and not implemented on a real time simulator testbed. Hence, the external use of a strong Off Board Computer is justified, capable of processing the pre-trained models for pose estimation. However, in our proposed approach, we aim to obtain real time implementation and to make our simulator testbed more closer and relevant to actual RPOD missions, we are abstaining from using a high end On Board Computer.

Our proposed Hardware Architecture is composed of a coupled Raspberry Pi based system to simulate the conditions of On Board Computing more accurately for our proposed simulator testbed. Thus removing the dependency on an Off Board Computer.

To conclude, the earliest ways of geometric pose estimation included the use of optical markers over the surface of the target spacecraft. Captured images from the On Board Camera of the chaser satellite shall interpret the captured images to estimate the target's pose. However, this method, which first came in the early 2000s, back when RPOD missions had just started to venture out into space exploration, has been proven redundant and inefficient. Further advancements in the same domain gave rise to more robust methods of feature extraction and corresponding-based algorithms to interpret the geometric model of the target spacecraft and estimate its pose based on the same. However, the significant challenges in its implementation proved to be the varied illumination and shadowing conditions which might pose the need to pre-process the captured images before pose estimation to overcome erroneous prediction. In the spirit of advancement for the same, Machine Learning and Regression Model-based approaches came to light and overcame all these possible challenges. However, in cost of extra computational constraint. Thus, increasing the demand for a more vital On Board Computer for chaser satellites.

5.4 Proposed Vision System for Pose Estimation

The previous section gave us a comprehensive guide towards the major forms of vision algorithms and their implementations for target pose estimation. In due consideration of all the possible drawbacks and advantages for each of the above mentioned procedures, the two major aspects of consideration for our proposed Vision System for Pose Estimation are, '*Adaptation to varied illuminating conditions*' and '*Computational Constraints.*'

Our proposed satellite simulator testbed is planned to perform the necessary computations over a coupled system of two Raspberry Pis, in order to reduce the computational abilities demonstrated by other previously mentioned satellite simulator testbeds, in order to make our research more closer and relevant to the actual scenarios of space exploration. Hence, while ML/CNN models for pose estimation are proven to be the most robust implementation for pose estimation, in light of computational constraints of the Raspberry Pi, we shall propose to implement a Line Detection based Image Processing and Feature Extraction method to estimate the pose of the target spacecraft. This section is an in-depth overview for the same.

5.4.1 Important Frames Of References

By convention all reference frames are R.H.S co-ordinate frames. These reference frames are going to be the parameters of the necessary translational and rotational functions, in order to estimate the position of the target by the use of the decided Heuristic Function.

The Frames Of References used for the same are defined as :-

- Camera Centered Co-ordinate Frame : Origin is camera's optical center.
- World Co-ordinate Frame : Centered around chaser satellite's Centre Of Gravity and oriented w.r.t its orbital plane.
- Target Centered Co-ordinate Frame : Centered around target's Center Of Gravity.

Therefore, locating the target frame w.r.t the camera frame will locate the target satellite w.r.t the world frame.

5.4.2 Image Pre-Processing

In order to detect lines(parameter for the Heuristic Function) efficiently, we need to perform some Image Pre-Processing before the implementation of the algorithm. In certain cases the image obtained shall be highly pixelated which might make it hard for the Hough-Line Transformation (and other equivalent techniques for line detection) to detect and extract straight lines out of the target image. Hence, in order to tackle this problem we use Gaussian Blurring to reduce the heavily pixelated nature of the image.

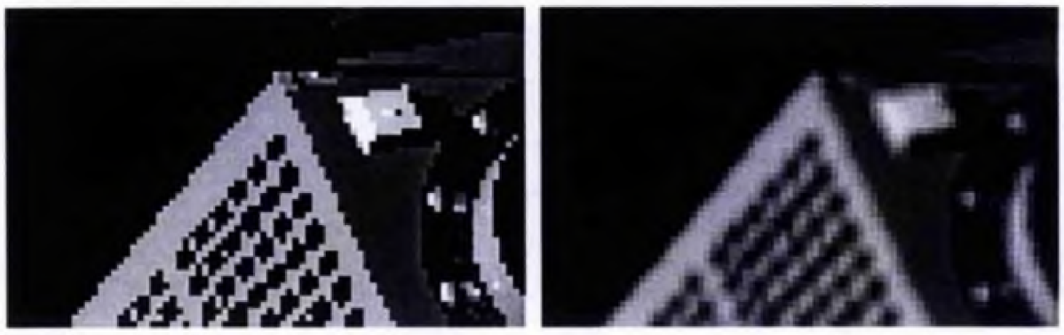

Figure 5.4: Pre-Processing of images to ensure efficient line detection

As observed in the test images above, the image on the left is the raw data obtained from the on-board camera of the chaser satellite. However its highly pixelated nature makes it harder for the Image Processing algorithm to detect the edges. Hence, blurring is performed over the edges of the target in order to tackle the same.

Other possible needs for Image Pre-Processing, before the step of gradient extraction shall be to reduce the anomalies caused due to varied illuminating conditions, shadowing, brightness, angle of illumination etc.

5.4.3 Canny Edge Detection

Certain amount of pre processing is required to be done in order to ensure detecting straight lines efficiently. *Canny Edge Detection* is used for the same.

The efficiency of line detection relies of the bin of accumulated pixel being distinct, e.g. a direct contrast between a pixel and its surrounding neighbours or if using a mask region a pixel region and its surrounds regions. If all pixels had similar accumulated values nothing would stand out as a line or circle. This leads to the reduction of colour (colour to grayscale, grayscale to black and white) in order to increase contract.

The number of parameters to the line detection algorithm also increase the spread of votes in the pixel bins and increase the complexity of the transform, which mean that normally only lines or circles are reliably detected using it as they have less than 3 parameters.

The edges need to be detected well before running the line detection, otherwise its efficiency suffers further. Also noisy images don't work well unless the noise is removed before hand.

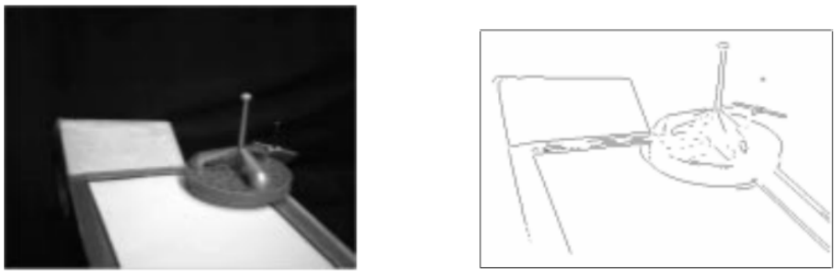

Figure 5.5: Canny Edge Detection over a target image

5.4.4 Image Heuristics

The Heuristic Function is responsible to judge and match lines detected in the image to the line model. The development of Heuristics for line correspondence is to developed using human involvement by mentioning the exact poses for and creating a reference for the extracted lines and their outputs. As shown below, a line model is developed in order to decide the necessary correspondences and match the extracted data from the detected lines to estimate the pose of the target.

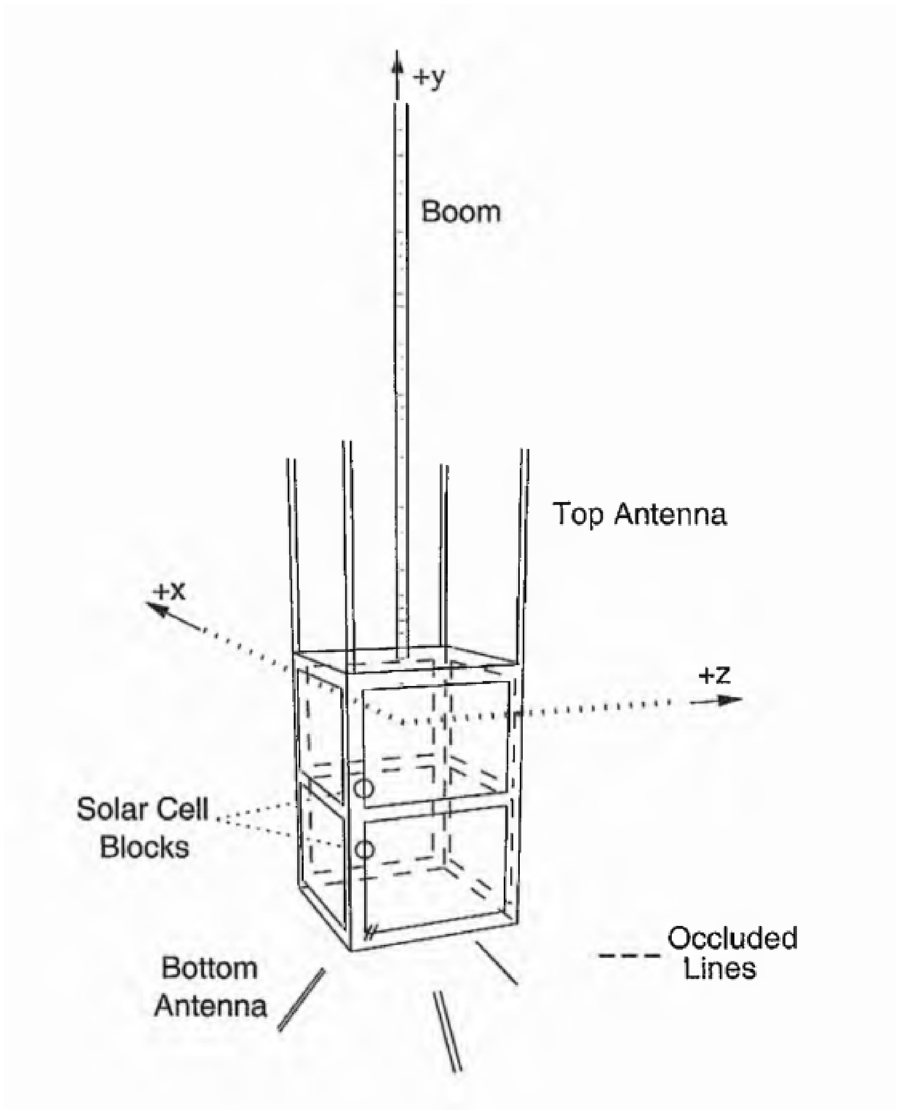

Figure 5.6: Line Model of the target spacecraft

The Heuristic Function is responsible for matching the lines extracted from image to the lines in the model. Chosen camera parameters influence the available ranges of estimated poses. A certain number of iterations for determining the pose is to be run, in order to ensure maximum possible correctness in estimating the position. The main target subset of the Heuristic Function to be developed is the set of lines parallel to the target axis. Hence, parallel lines detected from the image are further grouped w.r.t the axes they are parallel to in order to implement the same.

5.4.5 Line Detection

The most popular and efficient theory of performing Line Detection over an obtained Image is using '*Hough Line Transforms.*' Hough Line Transforms cam be used as pre built OpenCV libraries and can be implemented over Raspberry Pi using Python. Thus making the procedure desirable for us to implement over the proposed Hardware and Software Architecture. Hough Line Transforms are used to detect straight lines from the input image. It has two possible models of implementation in OpenCV :-

- Standard Hough Line Transform
- Probabilistic Hough Line Transform

Our proposed implementations over the satellite simulator testbed shall be a comprehensive comparison between both the techniques for the purpose of pose estimation.

The following steps summarize the working in brief :-

- It expresses lines in the Polar System. Hence, a line equation can be written as :-

$$y = -(\cos\theta/\sin\theta)x + (r/\sin\theta) \tag{5.2}$$

- In general for each point (x0,y0), we can define the family of lines that goes through that point as :-

$$r\theta = x0.\cos\theta + y0.\sin\theta \tag{5.3}$$

- If for a given (x0,y0) we plot the family of lines that goes through it, we get a sinusoid. For instance, for x0 = 8 and y0 = 6 we get the plot shown below.
- We can do the same operation above for all the points in an image. For instance, following with the example above and drawing the plot for two more points: x1 = 4, y1 = 9 and x2 = 12, y2 = 3, we get another plot, shown below.
- The more curves intersecting means that the line represented by that intersection have more points.
- This is what the Hough Line Transform does. It keeps track of the intersection between curves of every point in the image. If the number of intersections is above some threshold, then it declares it as a line.

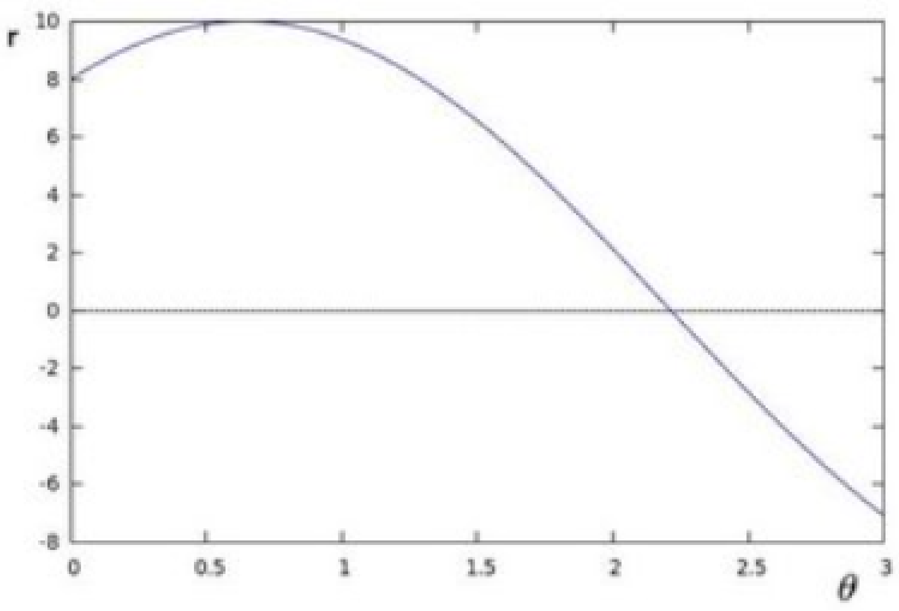

Figure 5.7: Plot for x0 = 8 and y0 = 6

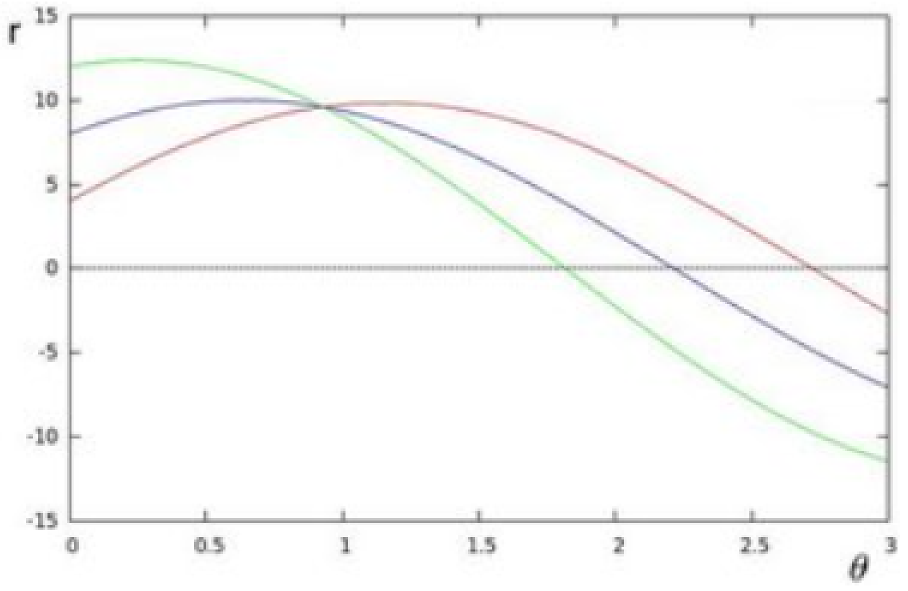

Figure 5.8: Plots for (x0 = 8, y0 = 6),(x1 = 4, y1 = 9) and (x2 = 12, y2 = 3)

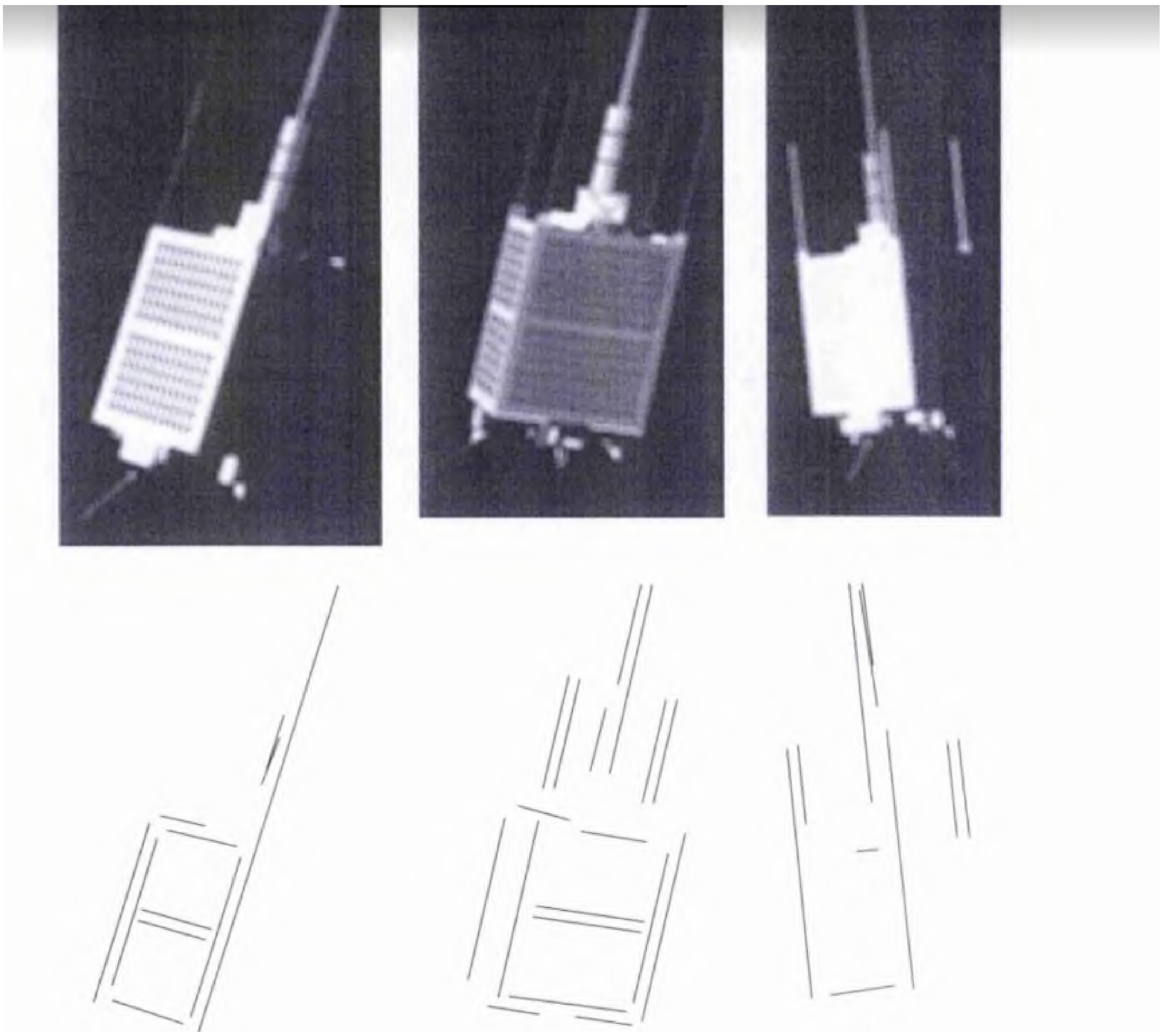
Figure 5.9: Hough Line Transforms to detect Straight Lines

5.5 Line Correspondence and Outlier Detection

5.5.1 Outlier Detection

In order to reduce the probability of estimating wrong pose of the target and increase the efficiency of the algorithm's output we need to identify and remove outliers form our predictions. Hence, we will use Random Sample Consensus (RANSAC), to fit a model and make it set for noisy measurements.

The parameters required for RANSAC to work in our specific case shall be :-

- Length of the chosen subset
- Camera calibration parameters
- Number of iterations

To determine the accuracy of the model fit, the model is projected back on the image, using estimated pose and known camera parameters. The number of iterations is dependent on length of the subset, chances of choosing outlier and confidence interval of the model.

5.5.2 Pose to Model Correspondence

The first step of corresponding line model is to estimate the rotational vector/matrix. The next step is to determine the translational aspect of the target. Relevant equations of translational and rotational matrices are thus used to find the corresponding estimated model from the target's line model.

Alexander Cropp shows the detailed implementation for the same in his satellite vision for pose estimation thesis. A brief recapitulation for the same is :-

- For each iteration, 4 image to model correspondences shall be considered.
- A series of rotational and translational matrix equations are used on the extracted set of line's data.

- Camera calibration parameters, Heuristic Function, Interpretation Plane and Image Plane are thus the important inputs for estimating the pose through these equations.
- Possible issue for the image-line correspondence shall be the convergence issue for 0 degree and 180 degree outputs.

5.5.3 Gauss-Newton Minimization

Gauss-Newton Minimization is used to improve the estimates pose by modelling and compensating for accumulated errors associated with line detection. However, in order to do the same it requires a pose estimate that is reasonably close to the true pose of the target.

Basically, Gauss-Newton Minimization is needed for providing a covariance matrix for the output parameter vector. Quaternions have been used to define the rotational vector for the parameters.

6 Conclusions

Rendezvous, Proximity and Docking operations are an essential part of space exploration missions. A brief overview of the history of RPOD mission highlight the importance of the same in refuelling spacecrafts and reducing space debris as well as maintenance of spacecrafts. In order to execute and optimize the algorithms aimed at RPOD space exploration missions, satellite simulator testbeds are made. Hence, they are used to simulate micro aerial conditions to launch robots called as satellite simulators, which work on cold gas thruster based propulsion mechanism.

Notable examples of the same shall be the 'POSEIDYN' satellite simulator proposed by the Naval Postgraduate School, a 3 Degree Of Freedom based satellite simulator which uses 10 *vicon cameras* and an external computer for the necessary calculations. 'ASTROS' satellite simulator, created by the Georgia Institute of Technology is another prime example of a 5 Degree Of Freedom based satellite simulator, with an additional air-bearing installed at the top. While most of these previous implementations assume the target position and attitude of the spacecraft to be docked known, our implementation brings out the novelty of vision based pose estimation and use less computational power by instead using a coupled system of 2 Raspberry Pis as the On Board Computer.

The satellite simulator testbed proposed in this report is composed of two satellite simulator robots with air bearing installed, operating over a granite table or a layer of resin, to simulate micro aerial and frictionless conditions. 8 Cold gas thrusters are installed on each side to form the propulsion control system, a specified mapping matrix is proposed to be implemented to control the pose and attitude of the chaser vehicle efficiently. LabView and Python, along with necessary OpenCV libraries shall be the majority of the proposed software architecture to be implemented. In addition, an on board camera will also be installed in the chaser robot to estimate target's pose through line detection and correspondence techniques to determine the target attitude for docking.

7 References

[1] Sternberg, David Pong, Christopher Filipe, Nuno Mohan, Swati Johnson, Shawn Jones-Wilson, Laura. (2017). Jet Propulsion Laboratory Small Satellite Dynamics Testbed Simulation: On-Orbit Performance Model Validation. Journal of Spacecraft and Rockets. 55. 1-13. 10.2514/1.A33806.

[2] www.nasa.gov/centers/johnson/engineering/integrated_environments/air_bearing_floor/index

[3] Kwok-Choon, Stephen T., K. Buchala, B. Blackwell, Susan E Lopresti, Markus Wilde and Tiauw Hiong Go. "Design , Fabrication , and Preliminary Testing of Air-Bearing Test Vehicles for the Study of Autonomous Satellite Maneuvers." (2018).

[4] Rybus, Tomasz, Janusz Nicolau-Kukli, Ski, Karol Seweryn, Tomasz Barci, Monika Ciesielska, Kamil Grassmann, Jerzy Grygorczuk, Michal Karczewski, Marek Kowalski, Marcin Krzewski, Tomasz Kuci, Jakub Lisowski, Rafał Przybyła, Konrad R. Skup, Tomasz Szewczyk and Roman Wawrzaszek. "NEW PLANAR AIR-BEARING MICROGRAVITY SIMULATOR FOR VERIFICATION OF SPACE ROBOTICS NUMERICAL SIMULATIONS AND CONTROL ALGORITHMS." (2013).

[5] Virgili-Llop, J., Zagaris, C., Park, H. et al. Experimental evaluation of model predictive control and inverse dynamics control for spacecraft proximity and docking maneuvers. CEAS Space J 10, 37–49 (2018). https://doi.org/10.1007/s12567-017-0155-7

[6] Dynamic Air-Bearing Hardware-in-the-Loop Testbed to Experimentally Evaluate Autonomous Spacecraft Proximity Maneuvers. Richard Zappulla II, Josep Virgili-Llop, Costantinos Zagaris, Hyeongjun Park, and Marcello Romano Journal of Spacecraft and Rockets 2017 54:4, 825-839

[7] Marco Sabatini, Giovanni B. Palmerini, Paolo Gasbarri, A testbed for visual based navigation and control during space rendezvous operations, Acta Astronautica, Volume 117, 2015, Pages 184-196, ISSN 0094-5765, https://doi.org/10.1016/j.actaastro.2015.07.026.

[8] Bevilacqua, Riccardo Romano, Marcello Curti, Fabio Caprari, Andrew Pellegrini, Veronica. (2011). Guidance Navigation and Control for Autonomous Multiple Spacecraft Assembly: Analysis and Experimentation. International Journal of Aerospace Engineering. 2011. 10.1155/2011/308245.

[9] Romano, Marcello Friedman, David Shay, Tracy. (2007). Laboratory Experimentation of Autonomous Spacecraft Approach and Docking to a Collaborative Target. Journal of Spacecraft and Rockets. 44. 164-173. 10.2514/1.22092.

[10] EXPERIMENTAL INVESTIGATION OF SPACECRAFT RENDEZVOUS AND DOCKING BY DEVELOPMENT OF A 3 DEGREE OF FREEDOM SATELLITE SIMULATOR TESTBED, JOSHUA COOKSON, York University, Toronto, Ontario, August 2019