1-1-2009

Application of the extended Kalman filter to Lidar pose estimation

Marcin Kuryllo
Ryerson University

APPLICATION OF THE EXTENDED KALMAN FILTER TO LIDAR POSE ESTIMATION

By

Marcin Kuryllo
BEng Ryerson University 2003

A thesis

presented to Ryerson University

In partial fulfillment of the

requirements for the degree of

Master of Applied Science

in the Program of

Aerospace Engineering

Toronto, Ontario, Canada, 2009

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Abstract

Marcin Kuryllo, Application of the Extended Kalman Filter to LIDAR Pose Estimation, MASc, Aerospace Engineering, Ryerson University of Toronto, 2009

The goal of this work is to investigate the benefits of using a well-known nonlinear motion estimator, an Extended Kalman Filter (EKF), in conjunction with the Iterative Closest Point algorithm (ICP), in particular, for the purpose of tracking the pose of a target satellite using a chaser satellite equipped with a LIDAR sensor. To accomplish this goal, two different architectures for tracking the pose of a target satellite were first implemented in MATLAB Simulink, and then implemented and tested on the Canadian Space Agency Automated Robotics Test Bed (CART) at the Canadian Space Agency (CSA) using a Neptec Laser Camera System as a sensor. The two architectures are: a) a pose tracking architecture that accepts the estimated pose supplied by the EKF to provide an initial pose guess to the ICP algorithm; and b) a pose tracking architecture that uses the pose supplied by the pervious pose measurement from the ICP algorithm as the initial pose guess for the ICP algorithm. The pose estimator combine with the EKF was able to track an object with a higher rate of motion then the rate possible without a nonlinear estimator. When the EKF estimate of the target satellite's states converges, a decrease in the number of ICP iterations per sensor measurement was also observed. Furthermore, the EKF increased the robustness of the system allowing the system to continue tracking after blackout periods. The test results showed an increased level of robustness of the tracking architecture that utilizes a nonlinear estimator in conjunction with the ICP algorithm. The advantages of the use of EKF were observed both in a simulated environment and experimentation.

Acknowledgements

I would like to acknowledge and thank my two professors at Ryerson University, Dr. Galina Okouneva and Dr. Donald McTavish, for their hard work, dedication and guidance without which this thesis would not be possible. I would also like to thank my supervisor at the Canadian Space Agency, Dr. Farhad Aghili, whose guidance and oversight helped lead the direction of my work. I would also like to thank my thesis defense committee, Dr. Yokota, Dr. Liu and Dr. Enright for overseeing my thesis defense. I would like to thank the department of Aerospace Engineering for its support of the project. Finally I would also like to thank the Canadian Space Agency for its hospitality and support of the project.

Dedication

Dedicated to my parents for their support throughout the years

Table of Contents

Abstract IV

Acknowledgements V

Dedication VI

Table of Figures IX

Notation, Abbreviations, Frames of Reference and Variables XI

Notation XI

Rotation XI

Quaternion Multiplication XI

Matrix Cross Product XII

Abbreviations XII

Frames of Reference XII

Variables and Constants XIII

Thesis Organization 1

Chapter I. Introduction 3

Chapter II. Theory and Previous Research 9

System Modeling 9

Models of Motion 10

Chaser Satellite Motion 10

Motion of Target Satellite 11

Previous Research 12

Extended Kalman Filter 16

Iterative Closest Point Algorithm 17

Chapter III. Extended Kalman Filter for Pose Estimation 21

Filter State Model 21

Measurement Model 25

Tracking Architectures 26

Chapter IV. Simulation, Experimental Setup and Results 31

Simulation 31

Experimental Setup 32

Simulation Results 39

Experimentation Results 41

Conclusion 55

Bibliography 57

Appendix 59

Results 59

Case I: Closed Loop 60

Case I: Open Loop 62

Case II: Closed Loop 64

Case II: Open Loop 66

Case III: Closed Loop 68

Case III: Open Loop Configuration 70

Case IV: Closed Loop 72

Table of Figures

Figure 1: Open Loop System ICP & EKF Architecture 7
Figure 2: Closed Loop System ICP & EKF Architecture 8
Figure 3: Chaser and Target Satellite System 9
Figure 4: Raster Scan of QuickSat Model 20
Figure 5: Physical System Representation 25
Figure 6: Architecture used in the open loop and closed loop configurations 28
Figure 7: CAD Model representation of target satellite, QuickSat, used in simulation and experimentation 31
Figure 8: Physical Experimental Setup 33
Figure 9: Graphical representation of apparatus and physical system simulated 34
Figure 10: Experimental system setup at initial conditions 36
Figure 11: Comparison between iterations required for ICP convergence 39
Figure 12: Error in Orientation for Open and Closed Configurations and Sensor Measurements 40
Figure 13: Case I Closed Loop Configuration 42
Figure 14: Case I Open Loop Configuration 42
Figure 15: Case II Closed Loop Configuration 43
Figure 16: Case II Open Loop Configuration 44
Figure 17: Case III Closed Loop Configuration 45
Figure 18: Case III Open Loop Configuration 45
Figure 19: Case IV Closed Loop Configuration 46
Figure 20: Case IV Open Loop Configuration 47
Figure 21: Case IV Open Loop Configuration, Target Position 48
Figure 22: Case IV Open Loop Configuration, Target Orientation 49
Figure 23: Case IV Open Loop Configuration, Target Angular Velocity 49
Figure 24: Case V Closed Loop Configuration, Target Position 51
Figure 25: Case V Closed Loop Configuration, Target Orientation 51
Figure 26: Case V Closed Loop Configuration, Target Angular Velocity 52
Figure 27: Case V Open Loop Configuration, Target Position 53
Figure 28: Case V Open Loop Configuration, Target Orientation 54
Figure 29: Case V Open Loop Configuration, Target Angular Rate 54
Figure 30: Case I: Estimated and sensor measurement for closed loop tracking architecture 60
Figure 31: Case I: Estimated and measured orientation for closed loop architecture 61
Figure 32: Case I: Estimated and true angular velocity of target satellite for closed loop architecture 61
Figure 33: Case I: Estimated and measured position of the target satellite for the open loop architecture 62
Figure 34: Case I: Estimated and measured orientation of target satellite for open loop architecture 63
Figure 35: Case I: Estimated and true angular velocity of target for open loop architecture 63
Figure 36: Case II: Estimated and measured position of target for closed loop architecture 64
Figure 37: Case II: Estimated and measured orientation of target for the closed loop architecture 65

Figure 38: Case II: Estimated and true angular velocity of the target for closed loop architecture 65
Figure 39: Case II: Estimated and measured position of target for the open loop architecture 66
Figure 40: Case II: Estimated and measured orientation of target for the open loop system 67
Figure 41: Case II: Estimated and true angular velocity of target for the open loop architecture 67
Figure 42: Case III: Estimated and measured position of target for the closed loop architecture 68
Figure 43: Case III: Estimated and measured orientation of target satellite for closed loop architecture 69
Figure 44: Case III: Estimated and true angular velocity of the target for the closed loop architecture ... 69
Figure 45: Case III: Estimated and measured position of the target for the open loop architecture 70
Figure 46: Case III: Estimated and measured orientation of the target satellite for the open loop architecture 71
Figure 47: Estimated and true angular velocity of the target for the open loop architecture 71
Figure 48: Case IV: Estimated and measured position of the target for the closed loop system 72
Figure 49: Case IV: Estimated and measured orientation of the target satellite for the closed loop architecture 73
Figure 50: Estimated and true angular velocity of the target for the closed loop architecture 73

Notation, Abbreviations, Frames of Reference and Variables

Notation

Rotation

There are several different methods used to represent a rotation. This work primarily uses Euler Parameter Quaternions (EPQ), Directional Cosine Matrices (DCM) and Euler Sequence Rotations represented as roll, pitch and yaw angles.

Where a rotational representation is used to transform a matrix representation of a vector, vectrix, from one frame to another the first subscript represents the frame which the vectrix will be expressed in after multiplication and the second subscript represents the frame which the vectrix was initially expressed in. For example the DCM, $\boldsymbol{C}_{ab}$, will transform a vectrix expressed in frame b, $\boldsymbol{F}_b$, to frame a, $\boldsymbol{F}_a$.

$$\boldsymbol{r}_a = \boldsymbol{C}_{ab}\boldsymbol{r}_b \quad (\boldsymbol{F}_b \rightarrow \boldsymbol{F}_a)$$

Quaternion Multiplication

A quaternion is a four parameter representation used in this work to represent rotation in 3D-space. It consists of a vector portion and a scalar portion. For example, if $\boldsymbol{a}$ is a quaternion, then the vector portion of $\boldsymbol{a}$ would be represented by using a subscript v, $\boldsymbol{a}_v$, and the scalar portion would be represented by using a subscript of o, a_o. In this work the two operators that denote quaternion multiplication and their formulation are defined in Equation 1 and Equation 2.

$$[\boldsymbol{a}\otimes] \triangleq \begin{bmatrix} -\boldsymbol{a}_v^{\times} + a_o\boldsymbol{I}_3 & \boldsymbol{a}_v \\ -\boldsymbol{a}_v^T & a_o \end{bmatrix} \qquad \text{Equation 1}$$

$$[\boldsymbol{a}\odot] \triangleq \begin{bmatrix} \boldsymbol{a}_v^{\times} + a_o\boldsymbol{I}_3 & \boldsymbol{a}_v \\ -\boldsymbol{a}_v^T & a_o \end{bmatrix} \qquad \text{Equation 2}$$

The two definitions are reverse processes of each other and their relation to each other can be seen in Equation 3.

$$\boldsymbol{q}_3 = \boldsymbol{q}_1 \otimes \boldsymbol{q}_2 = \boldsymbol{q}_2 \odot \boldsymbol{q}_1$$

Equation 3

Matrix Cross Product

The matrix version of the vector cross product is represented by Equation 4, where the vector portion of the cross product can be seen on the left hand side and the vector portion can be seen on the left hand side. The matrix cross product uses the skew symmetric matrix, and the corresponding vector has a superscript, $\boldsymbol{u}^{\times}$, as shown in Equation 5.

$$\vec{\boldsymbol{u}} \times \vec{\boldsymbol{v}} = \boldsymbol{u}^{\times} \boldsymbol{v}$$

Equation 4

$$\boldsymbol{u}^{\times} = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix}$$

Equation 5

Abbreviations

PoR	Point of Reference
EE	End Effecter
LCS	Laser Camera System
EPQ	Euler Parameter Quaternion
DCM	Directional Cosine Matrix
CAD	Computer Aided Design
CART	Canadian Space Agency Autonomous Robotics Test Bed
ICP	Iterative Closest Point
EKF	Extended Kalman Filter
CSA	Canadian Space Agency

Frames of Reference

$\boldsymbol{F_W}$ The World Frame located on the CSA Automated Robotics Test bed, CART

$\boldsymbol{F_{LCS}}$ The Laser Camera System, LCS, Frame located and oriented with the Neptec Tridar LCS

$\boldsymbol{F_T}$ The target satellite frame located at the center of mass, Center of mass, of the target satellite, QuickSat model, and aligned with the principal inertial axes

$\boldsymbol{F_C}$ The chaser frame, located at the chaser Center of mass and aligned with the chaser principal inertial axis, in experimentation the chaser model does not exist

$\boldsymbol{F_{EE}}$ The end effecter frame, EE, of reference located at the end of the STM1 arm of the CART which holds the QuickSat model mock up, aligned with $\boldsymbol{F}_T$

$\boldsymbol{F}_{PoR}$ The point of reference frame, PoR, located at the PoR on the target satellite, the [0, 0, 0] point in the CAD model of the QuickSat model.

Variables and Constants

$\boldsymbol{r}_{LCS,W}$ The position of the LCS with respect to $\boldsymbol{F}_W$

$\boldsymbol{r}_{T,W}$ The position of the target Center of mass with respect to $\boldsymbol{F}_W$

$\boldsymbol{r}_{C,W}$ The position of the chaser Center of mass with respect to $\boldsymbol{F}_W$

$\boldsymbol{r}_{PoR,W}$ The position of the PoR with respect to $\boldsymbol{F}_W$

$\boldsymbol{r}_{EE,W}$ The position of the EE with respect to $\boldsymbol{F}_W$

$\boldsymbol{\rho}_t$ The vectrix from $\boldsymbol{F}_T$ to $\boldsymbol{F}_{PoR}$ expressed in $\boldsymbol{F}_T$

$\boldsymbol{\rho}_c$ The vectrix from $\boldsymbol{F}_C$ to $\boldsymbol{F}_{LCS}$ expressed in $\boldsymbol{F}_C$

$\boldsymbol{o}$ The vectrix from $\boldsymbol{F}_{EE}$ to $\boldsymbol{F}_{PoR}$ expressed in $\boldsymbol{F}_T$

$\boldsymbol{q}_R$ Registration quaternion vector used in the ICP process, $q_0 \geq 0$

$\boldsymbol{r}_T$ Registration Translation vector used in the ICP process

$\boldsymbol{q}$ Registration vector used in the ICP process

$\boldsymbol{X}$ Model data point set used in the ICP algorithm

$\boldsymbol{P}$ Measured data point set used in the ICP algorithm

N_x Number of points in the model point set, $\boldsymbol{X}$

N_p Number of points in measured data point set, $\boldsymbol{P}$

$\boldsymbol{C}(\boldsymbol{q}_R)$ Rotation matrix constructed from $\boldsymbol{q}_R$

$\boldsymbol{\mu}_x$ The centriod of the model point set, $\boldsymbol{X}$

$\boldsymbol{\mu}_p$ The centriod of the measured point set, $\boldsymbol{P}$

$\boldsymbol{\Sigma}_{px}$ The cross covariance matrix constructed from the model point set

d The registration error, mean squared error used in the ICP algorithm

$\boldsymbol{A}$ The anti symmetric matrix used in the ICP algorithm

$\boldsymbol{\Delta}$ The cyclic components of the anti symmetric matrix used in the ICP algorithm

$\boldsymbol{L}$ Process noise Jacobian

$\boldsymbol{Q}$ The model noise covariance matrix

$\boldsymbol{K}$ The Kalman Filter gain matrix

$\boldsymbol{H}$ The output function Jacobian

$\boldsymbol{R}$ The sensor noise covariance matrix

Thesis Organization

This thesis is separated into five main portions, an introduction and four chapters. The introduction acquaints the reader with the variables, notation and an overview of the system being studied. This is then followed by the four chapters:

1. Chapter I. Introduction
2. Chapter II. Theory and Previous Research
3. Chapter III. Extended Kalman Filter for Pose Estimation
4. Chapter IV. Simulation, Experimental Setup and Results
5. Conclusion

In, Chapter I. Introduction, the goal of the thesis and the intended application are stated. It also provides background information on computer vision systems used, as well as a general over view of the architectures.

In Chapter II, the system modeled and the assumptions used are defined. The equations that are used to model the system are defined and explained. Once the system is defined and modeled, a general discussion of the EKF is introduced followed by a discussion of the ICP algorithm.

Chapter III discusses the specific state space equations used, first for the rotational motion of the system and then the translation motion. Then, the work undertaken in this thesis and its purpose is discussed in more details. In this Chapter, the two different tracking architectures used in this work are introduced and defined. In addition, the general method that was used to study the link between the EKF and ICP algorithm is introduced, as well as the method that was used to study the benefits of using the two in conjunction. Extensive details are provided on Simulation, Experimental Setup and Results, as well as on the method used for simulating the physical system and on the two phases of experimentation. The initial conditions and purpose for each test is presented, and any special events such as sensor occlusions are discussed. This is followed by a discussion about the data gathered through simulation and experimentation, first for the simulation and then for the laboratory experimentation.

The end of the thesis is dedicated to stating the conclusions drawn from an examination of the results. In addition, future work on improvements that can be made to the tracking architecture and Extended Kalman Filter is discussed.

Chapter I. Introduction

To accomplish autonomous orbital rendezvous for the purpose of satellite capture and docking, we need a system that is able to track the position and orientation of a target satellite. This system is comprised of two main components: a vision system, that is able to supply position and orientation measurements, and a state and parameter estimator, that can provide estimates of dynamic states. Progress in the fields of pose estimation and state and parameter estimation have created the opportunity to plan and design space missions, including autonomous spacecraft capture and rendezvous, with applications ranging from satellite service and repair to autonomous docking of spacecraft modules. The benefits of these missions include extended satellite life times, made possible by refueling missions (as the life time of a satellite is typically dictated by the amount of fuel available to maintain a desired orbit and attitude) and replacement of electronic modules. In addition to extended satellite life time, these missions would decrease the risk and cost associated with sending manned missions, as any maintenance preformed to satellites, such as the Hubble Space Telescope, is currently undertaken by a crew of astronauts who must spend an intense amount of time, in both preparation and performing the task, as well as putting their lives at high risk. There has been an extensive research in fields of computer vision such as pose estimation and state and parameter estimation. Research in pose estimation has focused primarily on the problem of determining an object's 3D position and orientation (pose) relative to a sensor's frame. Research in the fields of state and parameter estimation has focused primarily on the tasks of filtering sensor information using knowledge of the system's dynamics and the sensor's error statistics to provide an optimal or sub-optimal estimate.

Though there has been an extensive research in the areas of 3D pose, state and parameter estimation and methods that employ both for autonomous robotics, very little investigation has been done into the benefits of using a state and parameter estimator with a pose estimator.

The physical system for the intended application of this research, autonomous space craft rendezvous and capture, is represented as a two satellite system comprised of a target and chaser satellite. The target satellite's motion and pose is being estimated and tracked by the chaser satellite. It is assumed to be in a freefall orbit about the earth and in a passive tumble. The chaser satellite is equipped with a vision system that is measuring the pose of the target satellite relative to the vision system and an Extended Kalman Filter that is estimating the dynamics states of the target satellite. The chaser satellite is in a similar passive orbit to the target satellite, is nadir pointing and its states are assumed to be known.

Vision systems used for object pose estimation fall into two categories: passive and active. Passive sensors don't emit any signals in the process of observation like a single camera system or a stereo vision system. Active sensors emit a single that they then collect like Radio Frequency sensors or LIDAR sensors. The passive sensors that are used for object pose estimation are typically single camera or stereo systems, the active sensors that are used for object pose estimation are LIDARs.

The single camera system is able to provide a series of images, usually grayscale, that are used either individually or in conjunction with each other. Any single sensor reading using this type of sensor is able to only provide a two dimensional image and range data is not easily inferred from this type of image. In addition, cameras are also susceptible to the harsh and dynamic lighting present in space and when used, the time of operation has to be coordinated beforehand and in cases of poor lighting conditions self illumination would be required.

A stereo vision configuration uses multiple cameras. These cameras use the principal of binocular vision in order to extract depth information from two overlapping but distinct images. This method provides range data which can be used to match range data versus it. The data can be used as a method to construct a 3d model to be used for range data matching. The stereo vision configuration has similar advantages and disadvantages to that of the single camera vision system but is able to make use of the 3d range data to increase the methods available for object pose estimation and motion tracking.

A laser range scanner, for example, a Laser Camera System (LCS) developed by Neptec Design Group, is a single unit that provides 3D range data typically in the form of a point cloud of (x, y, z)- data in the frame of reference of the LCS. This point cloud can then be used with a priori knowledge of target geometry to estimate the pose or the data can be used to generate a model of the target object while matching subsequent data to the generated model. The benefits of using laser range scanners comes from the robustness of the imaging system; these scanners are able to work in multiple types of the lighting conditions present in space and have flight heritage. Laser scanners can be used at anytime and do not require coordination and restrictions on the mission due to lighting conditions. The disadvantage of using these scanners is the higher cost and power consumption. In the space environment where harsh and dynamic lighting conditions are present, the use of an LCS based vision system would be preferable to a system that uses cameras.

To determine the pose of an object, LCS uses the Iterative Closest Point (ICP) algorithm. The ICP algorithm accepts the point cloud generated by the LCS system, a priori knowledge of the target's geometry and an initial guess of the target's pose. A cost function is then minimized and used to calculate a rotation and translation between the two point clouds (data points from the scanner and the corresponding model points). This method is repeated iteratively until some predefined convergence criteria is met. The ICP algorithm is linearly convergent and not globally convergent so the initial guess of the target object's pose is important.

The initial pose guess is a core parameter for the ICP algorithm. If the initial guess is not close to the true pose, ICP can be stalled in a local minimum of the cost function. It was experimentally discovered that the initial guess must be within 10-15 degrees in roll, pitch and yaw of the true pose (Besl & McKay, 1992). In well coordinated missions such as space docking and rendezvous, ICP is used to refine the pose at the very final stage of operation. In these situations the initial guess can be accurately estimated from another source available to the mission specialists. The basic ICP algorithm is subject to two cases where using it alone will cause object tracking to fail. These two cases are: sensor blackout periods and fast rotational tumbling of the target object.

If the target object is unexpectedly not in the scanner's field of view, or the system does not function temporarily, the current solution to this problem is to store the last computed pose and use it as an initial guess when the system is back. It is a reasonable approach, but only in case when the system's blackout is short and the relative position of the docking modules did not change beyond the limit of the ICP algorithm. Or, if the target object is tumbling at a fast rotational velocity, the change in the target's pose will be beyond the limits of the ICP algorithm and the previous pose cannot be used as an initial guess to the ICP algorithm. If the last posies not available as an initial guess for ICP, then an estimator must be used to provide the improved initial guess while no tracking occurs. In this thesis, a nonlinear state estimator, namely the Extended Kalman Filter (EKF), is used to provide the estimation of the pose. The Kalman filter was first developed in 1960, (Kalman, 1960), and proved a novel method for estimating system states and parameters in the presence of measurement noise. Since then many different variations of the KF have been created including the EKF which is used in this work.

In addition to providing the ICP algorithm with an intelligent pose initial guess for each measurement cycle, the EKF is a generally beneficial for the application of autonomous rendezvous and control. The EKF can be used to provide a larger range of states which are not necessarily measured by the sensors. For example, when tracking tumbling of a target object, the sensor is able to measure the orientation of the target object but not its angular velocity, the use of the EKF allows the angular velocity of the tumbling object to be estimated even if it is not directly measured. Additionally, various parameters, such as the position of the center of mass of the target object, within the dynamic model of the system may only be known within certain accuracy, and these parameter estimates can be refined in real time.

The goal of this work is to implement the EKF functionality in the ICP algorithm, and then study the benefits of using the ICP pose estimator in conjunction with the EKF. The work was performed at the Canadian Space Agency (CSA) lab using Neptec's LCS and a QuickSat mock-up. In order to accomplish these goals, two different system architectures were implemented: first, in simulation using MATLAB Simulink environment, and then on the CSA Automated Robotics Test Bed (CART). The two different system architectures are variants on the combination of the ICP algorithm and the EKF. The first system architecture, referred to as the open loop system, is one where the ICP algorithm uses the pose from the previous measurement cycle as the initial guess for the current measurement cycle. The open loop system architecture is depicted in Figure 1.

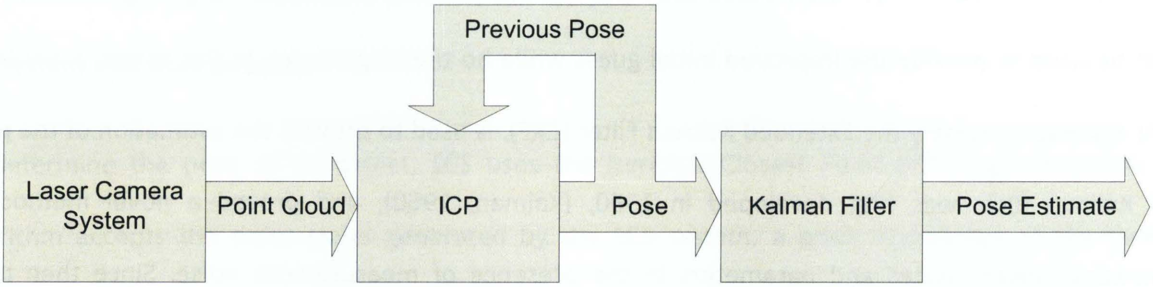

Figure 1: Open Loop System ICP & EKF Architecture

The second system architecture, referred to as the closed loop system, uses the EKF estimate at the time of the current measurement cycle as the initial condition for the ICP algorithm to find the current measurement pose which is then used in the EKF to perform a state estimate update. The closed loop system architecture can be seen in Figure 2.

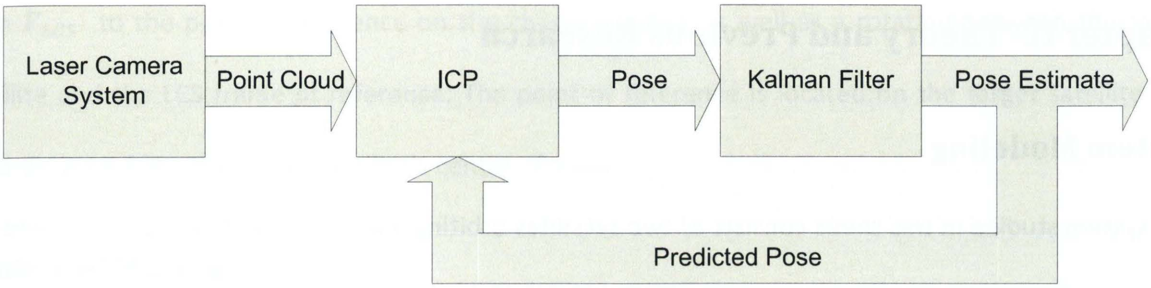

Figure 2: Closed Loop System ICP & EKF Architecture

Chapter II. Theory and Previous Research

System Modeling

The system studied in this thesis consists of two satellites orbiting about the earth: a chaser satellite and a target satellite. The satellites are assumed to be passively orbiting about the Earth. In total, there are five frames of reference associated with this system. The satellites orbit about the Earth Centered Inertial reference frame $\boldsymbol{F}_{ECI}$. There are two reference frames associated with the target satellite: the first reference frame is located at the center of mass of the target satellite and is aligned with the body principal axes' $\boldsymbol{F}_T$; the second reference frame is located at the point of reference on the target satellite and is aligned with $\boldsymbol{F}_T$. The final two reference frames are associated with the chaser satellite: the first reference frame is located at the center of mass of the chaser satellite and is x-axis nadir pointing with the z-axis normal to the orbital plane. The second frame of reference associated with the chaser satellite is located at a fixed position and orientation with respect to the chaser satellite.

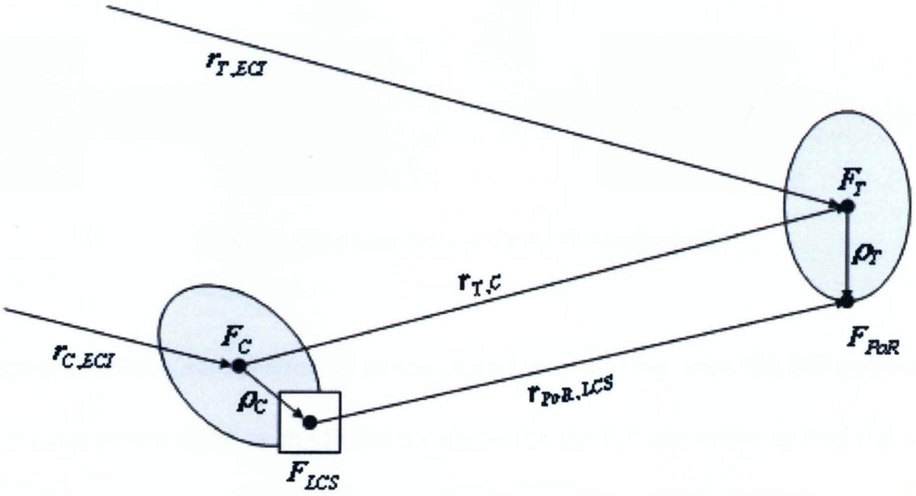

Figure 3: Chaser and Target Satellite System

The chaser satellite is equipped with a LCS mounted at a fixed distance $\boldsymbol{\rho}_C$ from the center of mass of the chaser satellite and a fixed orientation $\boldsymbol{C}_{LCS,C}$ with respect to $\boldsymbol{F}_C$. The LCS measures the distance from $\boldsymbol{F}_{LCS}$ to the point of reference on the chaser satellite as well as a rotation between the target satellite and the LCS frame of reference. The point of reference is located on the target satellite at a fixed distance from the target satellite's center of mass.

Models of Motion

The general models of motion for the system are based on: a) the assumption that the translational motion of the each satellite's center of mass is based on the satellite being in a passive orbit about the Earth. In other words, there are no external forces action on the satellites other then gravity; b) the attitude motion of the target satellite is torque free rotation about its center of mass, while the chaser satellite's attitude is specified as nadir pointing and the yaw axis is perpendicular to the orbital plane.

Chaser Satellite Motion

The chaser satellite's position is propagated through the inertial model of motion that can be seen in Equation 6, (Hughes, 1986).

$$\ddot{\boldsymbol{r}}_{C,ECI} = -\frac{\mu}{\left\|\boldsymbol{r}_{C,ECI}\right\|}\boldsymbol{r}_{C,ECI} + \boldsymbol{\delta a} \qquad \text{Equation 6}$$

where $\boldsymbol{\delta a}$ represents any orbital perturbation force that might affect the satellite, μ is the gravitational constant for earth and $\boldsymbol{r}_{C,ECI}$ is the inertial position of the chaser satellite in the earth centered inertial frame. This model can be used with numerical integration techniques to determine the inertial velocity and the inertial position of the chaser satellite with respect to the earth centered inertial frame. Since the magnitude of the orbit of the satellite is on the order of thousands of kilometers and the state estimated by the EKF is on the order of several meters, it is important to pick a numerical integration scheme that does not add numerical noise. Equation 7 and Equation 8 explain how the inertial velocity and the inertial position of the chaser satellite is propagated, where the velocity is the integration of the acceleration with time, and the position is the integration of the velocity with respect to time.

$$\dot{\boldsymbol{r}}_{C,ECI} = \int_0^t \ddot{\boldsymbol{r}}_{C,ECI} dt + c$$ Equation 7

$$\boldsymbol{r}_{C,ECI} = \iint_0^t \ddot{\boldsymbol{r}}_{C,ECI} dt + c$$ Equation 8

The attitude motion of the chaser satellite is specified as nadir pointing. Therefore, there are no dynamic models of motion associated with it that need to be included in the framework.

Motion of Target Satellite

The target satellite's position, velocity and acceleration are propagated through the same method that is used for the chaser satellite. Equation 9 describes the rate of change in angular velocity of the target satellite about its center of mass with respect to time, (Hughes, 1986). Equation 10 and Equation 11 describe the rate of change in the target satellite's orientation with respect to time when represented by a Euler Parameter Quaternion, (Hughes, 1986).

$$\dot{\boldsymbol{\omega}}_B = \boldsymbol{I}^{-1}(-\boldsymbol{\omega}_B^{\times}\boldsymbol{I}\boldsymbol{\omega}_B + \boldsymbol{g})$$ Equation 9

$$\dot{\boldsymbol{\epsilon}} = \frac{1}{2}(\boldsymbol{\epsilon}^{\times} + \eta\boldsymbol{1})\boldsymbol{\omega}_B$$ Equation 10

$$\dot{\eta} = -\frac{1}{2}\boldsymbol{\epsilon}^T\boldsymbol{\omega}_B$$ Equation 11

For the propagation of the angular velocity, $\boldsymbol{\omega}_B$ and $\dot{\boldsymbol{\omega}}_B$ are the angular rate and the derivative with respect to time of the target satellite in the target frame, $\boldsymbol{I}$ is the inertial matrix of the target satellite in the target satellite's frame, and $\boldsymbol{g}$ is the external torque applied to the target satellite. For the propagation of the quaternion representation of the target satellite's attitude, η and $\boldsymbol{\epsilon}$ are the scalar and vector portion of the quaternion, and $\dot{\eta}$ and $\dot{\boldsymbol{\epsilon}}$ are the derivatives with respect to time. The attitude motion of the target satellite is propagated first by using Equation 9 to propagate the target satellite's angular velocity, and Equation 10 and Equation 11 to propagate the Euler parameter quaternion that represents the attitude of the target satellite. For the rotational motion of the target satellite, there is no general solution to the attitude motion, and in order to obtain the attitude motion of the target satellite, a numerical integrator needs to be used in conjunction with Equation 9, Equation 10 and Equation 11.

Previous Research

Previous work in relation to the problem of object pose estimation and tracking includes that of (Hillenbrand & Lampariello, 2005), (Kelsey, Byrne, Cosgrove, Seereeram, & Mehra, 2002), (Lavin, Noyer, & Benjelloun, 2005), (Ruel, English, Anctil, & Church, 2005), (Ruiter & Benhabib, 2005), and (Simon, Hebert, & Kanade, 1994). In (Hillenbrand & Lampariello, 2005), the primary goal of the work presented is to investigate a method for state and parameter identification for the purpose of long term motion prediction. This is done to minimize problems in situations where occlusion are present, limited communications with the ground are available and for mission planning (autonomous grasping) are involved. This work is a study of three scenarios involving the number of points available from the vision system and the availability of point correspondences with respect to time. The method investigated does not presume a foreknowledge of the target's geometry. The vision data obtained from the vision system used in the study is the visible set of simulated range data uniformly distributed over the objects surface with 5mm standard deviation Gaussian noise and can be obtained through the use of stereo vision or a laser range scanner. The simulation is run for 200 seconds, for the first 100 seconds the target is observed at a rate of 1Hz and then for the next 100 seconds the motion is predicted without any observations being made. The parameters and state are identified by using a cost function to minimize an error function which incorporates all 100 scans and uses the incremental pose as their input. While this method uses a motion predictor it does not make use of it for the purpose of the pose estimate obtained from the use of a closed form solution to absolute orientation between two corresponding data sets but instead opts to use the pose estimate of the system at the previous observation.

In (Kelsey, Byrne, Cosgrove, Seereeram, & Mehra, 2002), Kelsey et al. present a system that can supply real-time estimates of relative pose between a target object and a chaser vehicle for the purposes of autonomous rendezvous and docking. The method studied uses a vision system in conjunction with a state and parameter estimator in order to estimate the targets pose and translational and angular motion. The system uses the predicted pose from the state estimator as the initial guess for the pose estimator used in the vision system. The state estimator used for the purpose of this system is the EKF and the vision system used is a single camera system. This single camera vision system implements a method to refine the predicted pose supplied by the EKF; this method is accomplished by rendering a view of the model from the a priori knowledge of the target and using an Iterative Recursive Least Squares (IRLS) method to match the rendered model to the image via a set of lines normal to the rendered model edges. This system Kelsey et al. present is tested in both simulation and using an assortment of scale models in the lab. The motion profiles tested using this system are a variety of 2D motion profiles, planar translation and rotation about the normal of the plane, including linear, spiral, Lissajous-type translation and a mixture of the three with a constant rotation. This paper demonstrates the use of a state and parameter estimation technique used with a pose estimator, the authors opt to concentrate on the pose refinement algorithm and the algorithms implementation and performance and do not explore the benefits of using one with the other. The authors state the potential benefits of this architecture as possible improvements to model correspondence and tracking performance but do not implement them and proceed to concentrate on the pose refinement technique.

In (Lavin, Noyer, & Benjelloun, 2005), the authors present an implementation method for a 3D object tracking and motion estimation algorithm. The method for object tracking and pose estimation uses a priori knowledge of the object, geometrical and texture, in conjunction with a single vision camera and a particle filter (PF) to predict the motion of the object. A predicted pose obtained through the PF is used to render the model of the target and then the pose is refined using the intensity values in the image and the rendered image of the model. The paper concentrates on the implementation of the algorithm on hardware, a standard desktop computer with a graphics processing unit (GPU), and the division of tasks between the CPU and GPU. This paper makes use of a nonlinear state estimator in order to reduce the computational time of the implementation; the focus is on the division of various tasks between the system hardware and not a study of the potential benefit of using a state estimator in conjunction with a computer vision system.

The work demonstrated in (Ruel, English, Anctil, & Church, 2005) varies from the previous references where the authors do not make use of a nonlinear state estimator, but choose to rely solely on a computer vision system to track the motion of a target for the purpose of autonomous satellite servicing. The method that Ruel, English, Anctil & Church present uses 3D data from a laser camera system (LCS) to obtain range data and uses an iterative closest point (ICP) algorithm to match that data to a geometrical representation of the target, assuming a priori knowledge. The method is tested both in a simulation environment and in a lab environment using a scale model of the Hubble space telescope. The system is able to demonstrate that object tracking and pose estimation is possible in real-time without the use of a motion predictor when the relative velocities of the target, translational and rotational, are small in comparison to the observation rate.

The method that Ruiter & Benhabib present in (Ruiter & Benhabib, 2005) uses a single camera vision system and a priori knowledge of the target. The camera obtains a single image and projects a texture mapped model of the object onto the image plane using a pose estimate supplied by the Kalman Filter. The two images are then compared using optical flow and the results are used to refine the pose predicted by the Kalman Filter. This method is tested both in simulation and on a 2D translational test bed. The work makes use of the Kalman Filter but chooses to concentrate on the implementation of the 3D vision system using optical flow.

The work that Simon, Hebert, & Kanade present in (Simon, Hebert, & Kanade, 1994) is similar to that of (Ruel, English, Anctil, & Church, 2005), but Simon, Hebert, & Kanade use of a light strip generator in conjunction with a camera to extract range data from the target instead of an LCS. The authors in this study demonstrate that real-time motion and pose tracking is possible without the use of a motion predictor but concede that to accommodate for faster relative motion of the target the accuracy of the ICP algorithm will have to be reduced in order to provide an increased pose estimation cycle rate.

The work presented by the authors of these papers all demonstrate the various capabilities of several different methods to track the pose of an object moving relative to a vision system through the use of computer vision by itself or coupled with a nonlinear estimation. The capability of using a nonlinear estimator with a computer vision system is shown but no investigation is done as to the potential benefits and drawbacks of using computer vision with a nonlinear estimator, with most papers concentrating on the development of the computer vision system. In addition to this the majority of the papers reviewed leave the coupling at the level of using the nonlinear state estimator to provide a predicted pose to the visions system algorithm to future work.

For the purpose of pose estimation and tracking, there are two different structures used. The first present and simplest is one where only a pose estimator is used and the next pose is obtained through using a pose refinement technique using the pose from the previous sensor cycle, as demonstrated in (Ruel, English, Anctil, & Church, 2005) and (Simon, Hebert, & Kanade). The second method used is more difficult to implement but offers a robust system for the purposes of pose estimation and motion tracking. This method employs the use of a state and parameter estimator for the purpose of providing an optimal or suboptimal prediction of the target pose at the time of a sensor update, as demonstrated in (Hillenbrand & Lampariello, 2005), (Kelsey, Byrne, Cosgrove, Seereeram, & Mehra, 2002), (Lavin, Noyer, & Benjelloun, 2005) and (Ruiter & Benhabib, 2005).

Extended Kalman Filter

The Extended Kalman Filter, the EKF, is a nonlinear recursive filter that propagates the mean and covariance of a system's states and/or parameters. The mean and covariance propagated by the EKF represents a conditional probability density function of a physical system's states and parameters. The reason why the EKF and other filters are used is because in real world situations the sensor readings of a system are always subject to noise. A physical system's states and parameters are typically known only with error, and in some cases some of the system's states at the start of the system's use are unknown and cannot be easily measured during the operation. The EKF uses a nonlinear dynamic model of the system to propagate the mean of the estimated states through state space, and linearized Jacobians of the nonlinear state functions to propagate the states covariance through the nonlinear space. The propagation of the system's states and covariance is accomplished by using Equation 12 and Equation 13.

$$\hat{\boldsymbol{x}}[t_k(-)] = \hat{\boldsymbol{x}}[t_{k-1}(-)] + \int_{t_{k-1}}^{t_k} \boldsymbol{f}\{\hat{\boldsymbol{x}}[\tau(-)], \boldsymbol{u}(\tau), \tau\}\, d\tau \qquad \text{Equation 12}$$

$$\boldsymbol{P}[t_k(-)] = \boldsymbol{P}[t_{k-1}(-)] + \int_{t_{k-1}}^{t_k} [\boldsymbol{F}(\tau)\boldsymbol{P}(\tau) + \boldsymbol{P}(\tau)\boldsymbol{F}^T(\tau) + \boldsymbol{L}(\tau)\boldsymbol{Q}_C(\tau)\boldsymbol{L}^T(\tau)]d\tau \qquad \text{Equation 13}$$

where $\hat{\boldsymbol{x}}$ is the estimate system state vector, $\boldsymbol{f}$ is the nonlinear system equation vector, $\boldsymbol{P}$ is the system covariance matrix, $\boldsymbol{F}$ is the system equation Jacobian defined in Equation 14 and $\boldsymbol{L}$ is the process noise Jacobian defined in Equation 15.

$$\boldsymbol{F}(t_k) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \qquad \text{Equation 14}$$

$$\boldsymbol{L}(t_k) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{w}} \qquad \text{Equation 15}$$

At the time of sensor measurements, the estimate of the probability density function and the sensors' error characteristics are used to calculate gains based on mean squared error statistics which are then used to calculate an optimal update of the system's probability density function to be propagated until the next sensor reading occurs. This process can be seen in Equation 16, Equation 17 and Equation 18.

$$\boldsymbol{K}(t_k) = \boldsymbol{P}[t_k(-)]\boldsymbol{H}^T(t_k)\{\boldsymbol{H}(t_k)\boldsymbol{P}[t_k(-)]\boldsymbol{H}^T(t_k) + \boldsymbol{R}(t_k)\}^{-1}$$ Equation 16

$$\widehat{\boldsymbol{x}}[t_k(+)] = \widehat{\boldsymbol{x}}[t_k(-)] + \boldsymbol{K}(t_k)\{\boldsymbol{z}(t_k) - \boldsymbol{h}[\widehat{\boldsymbol{x}}[t_k(-)], t_k]\}$$ Equation 17

$$\boldsymbol{P}[t_k(+)] = \left(\boldsymbol{1} - \boldsymbol{K}(t_k)\boldsymbol{H}(t_k)\right)\boldsymbol{P}_{\bar{t}_k}^{-}\left(\boldsymbol{1} - \boldsymbol{K}(t_k)\boldsymbol{H}(t_k)\right)^T + \boldsymbol{K}(t_k)\boldsymbol{R}(t_k)\boldsymbol{K}(t_k)^T$$ Equation 18

where $\boldsymbol{K}$ is mean squared gain matrix, $\boldsymbol{z}$ is the sensor measurement, $\boldsymbol{R}$ is the sensor error covariance matrix that represents the assumed error in sensor measurements and $\boldsymbol{h}$ and $\boldsymbol{H}$ is the output measurement equation and the output measurement Jacobian defined in Equation 19 respectively.

$$\boldsymbol{H}(t_k) = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}$$ Equation 19

Iterative Closest Point Algorithm

The formulation of the ICP algorithm used for the pose measurement of the target satellite was first introduced in (Besl & McKay, 1992). It is assumed that there is prior knowledge of the target object's shape in the form of a CAD model, a method to obtain a measured point cloud of the object and that there is an initial guess as to pose of the target object relative to the measurement frame. The measured point cloud consists of points on the target object that have been measured by a computer vision system.

The first step of the ICP algorithm is to match them versus the closest points on the CAD model of the target object. After this has been done, the two point clouds, the measured point set $\boldsymbol{P}$, and the closest points to the measured points on the target model called the target model point set $\boldsymbol{M}$, can be used to compute the least squares registration between the two sets of the point clouds. The vectors representing the pose quaternion and translation are:

$$\boldsymbol{q}_R = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}$$ Equation 20

$$\boldsymbol{r}_T = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}$$ Equation 21

Where $\boldsymbol{q}_R$ is the registration quaternion and $\boldsymbol{r}_T$ is the registration distance. The overall pose vector is expressed as the combination of the rotational and translational pose vectors and is expressed in Equation 22:

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{q}_R \\ \boldsymbol{r}_T \end{bmatrix}$$ Equation 22

The covariance matrix for the quaternion based on a least-squares objective function can be seen in Equation 23.

$$\boldsymbol{Q}(\boldsymbol{\Sigma}_{pm}) = \begin{bmatrix} tr(\boldsymbol{\Sigma}_{pm}) & \boldsymbol{\Delta}^T \\ \boldsymbol{\Delta} & \boldsymbol{\Sigma}_{pm} + \boldsymbol{\Sigma}_{pm}^T - tr(\boldsymbol{\Sigma}_{pm})\boldsymbol{I}_3 \end{bmatrix}$$ Equation 23

Where $\boldsymbol{\Sigma}_{pm}$, $\boldsymbol{\Delta}$, and $\boldsymbol{I}_3$ are the cross covariance matrix, which is a measure of the similarity between the point clouds, the cyclic components of the anti-symmetric matrix and a 3 by 3 identity matrix respectively. The eigen-values and vectors of matrix $\boldsymbol{Q}(\boldsymbol{\Sigma}_{pm})$ are then found, and the eigen-vector with the maximum eigen-value is taken to be the pose quaternion $\boldsymbol{q}_R$ after being normalized. The pose translation vector is then calculated by:

$$\boldsymbol{r}_T = \boldsymbol{\mu}_x - \boldsymbol{C}(\boldsymbol{q}_R)\boldsymbol{\mu}_p$$ Equation 24

Where $\boldsymbol{\mu}_x$ and $\boldsymbol{\mu}_p$ are the centriods of the two point clouds and $\boldsymbol{C}(\boldsymbol{q}_R)$ can be calculated by:

$$\boldsymbol{C}(\boldsymbol{q}_R) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 + q_2^2 - q_1^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 + q_3^2 - q_1^2 - q_2^2 \end{bmatrix}$$ Equation 25

The cross-covariance matrix $\boldsymbol{\Sigma}_{px}$ is:

$$\boldsymbol{\Sigma}_{px} = \frac{1}{N_p}\sum_{i=1}^{N_p}\left[(\boldsymbol{p}_i - \boldsymbol{m}_p)(\boldsymbol{m}_i - \boldsymbol{\mu}_x)^T\right] = \frac{1}{N_p}\sum_{i=1}^{N_p}\left[\boldsymbol{p}_i\boldsymbol{m}_i^T\right] - \boldsymbol{\mu}_p\boldsymbol{\mu}_x^T \qquad \text{Equation 26}$$

where $\boldsymbol{p}_i$ and $\boldsymbol{m}_i$ are individual points within the model point set and the measured point set. $\boldsymbol{\Delta}$ is comprised of the cyclic components of the anti-symmetric matrix $\boldsymbol{A}$:

$$\boldsymbol{\Delta} = \begin{bmatrix} A_{23} \\ A_{13} \\ A_{12} \end{bmatrix} \qquad \text{Equation 27}$$

The anti -symmetric matrix $\boldsymbol{A}$ is calculated using Equation 28:

$$\boldsymbol{A} = \boldsymbol{\Sigma}_{px} - \boldsymbol{\Sigma}_{\mathrm{px}}^{\mathrm{T}} \qquad \text{Equation 28}$$

The measured data point set $\boldsymbol{P}_k$ is then adjusted by the registration vector through:

$$\boldsymbol{p}_{i,k+1} = \boldsymbol{C}(\boldsymbol{q}_{R,k})\boldsymbol{p}_{i,k} - \boldsymbol{r}_{T,k} \qquad \text{Equation 29}$$

Once the pose has been calculated for the particular iteration, it is applied to the overall pose and then the mean squared error of the particular iteration can be calculated:

$$d = \frac{1}{N_p}\sum_{i=1}^{N_p}\left\|\boldsymbol{m}_{i,k} - \boldsymbol{p}_{i,k+1}\right\|^2 \qquad \text{Equation 30}$$

This process is continued until the difference between the registration error of the previous iteration and the current step is smaller than some predefined tolerance τ. For every iteration, k, the target model point set needs to be recalculated as the closest points on the target model to the measured point set will change when the registration is applied.

The ICP algorithm can converge to a local minimum based on the mean squared objective function. The initial guess of the pose of the object when the ICP is initiated determines if it will converge to the proper global minimum and supply the correct pose of the object.

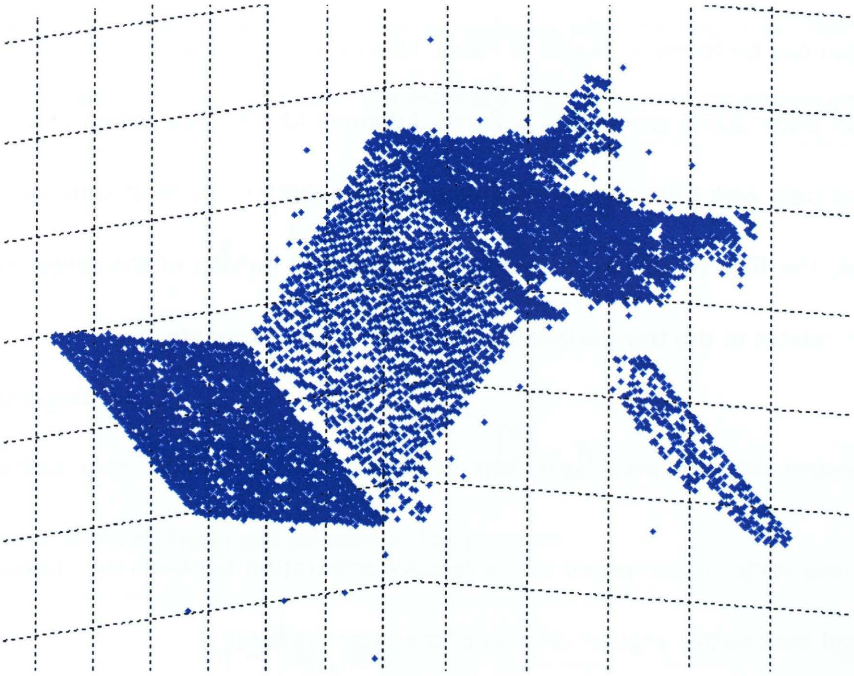

Figure 4: Raster Scan of QuickSat Model

In Figure 4 a typical raster scan of the QuickSat model used in this thesis using the Neptec LCS can be seen. In this figure the presence of outliers in the scan data can lead to poor convergence in the ICP algorithm, as the ICP algorithm uses a Least Squares error metric. In order to remove this effect the outliers are removed by Neptec LCS based on the range from the CAD model, Figure 7, and the initial guess of the models location for the ICP algorithm.

Chapter III. Extended Kalman Filter for Pose Estimation

Filter State Model

The specific filter state equations for the purpose of this work were developed by Aghili & Parsa and an in depth derivation can be found in (Aghili & Parsa, Motion and Parameter Estimation of Space Objects Using Laser-Vision Data, 2009) and (Aghili & Parsa, Adaptive Motion Estimation of a Tumbling Satellite Using Laser-Vision Data with Unknown Noise Characteristics, 2007). The filter state vector is divided into two state vectors, the first vector, $\boldsymbol{x}$, relates to the rotational motion of the target satellite while the second vector, $\boldsymbol{y}$, relates to the translational motion of the target satellite.

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} \qquad \text{Equation 31}$$

The rotational state vector is composed of the relative orientation between the chaser satellite and the target satellite and the relative angular velocity of the target satellite.

$$\boldsymbol{x} = \begin{bmatrix} (\boldsymbol{q})_v \\ \boldsymbol{\omega}_{rel} \end{bmatrix} \qquad \text{Equation 32}$$

The translational state vector of the system is comprised of the relative position of the target's center of mass with respect to the chaser's center of mass and the relative velocity between the target's and chaser's center of mass.

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{r}_{T,C} \\ \boldsymbol{v}_{T,C} \end{bmatrix} \qquad \text{Equation 33}$$

The overall state vector used for the propagation of the system estimate from one sensor reading to the next is:

$$\boldsymbol{X} = \begin{bmatrix} (\boldsymbol{q})_v \\ \boldsymbol{\omega}_{rel} \\ \boldsymbol{r}_{T,C} \\ \boldsymbol{v}_{T,C} \end{bmatrix} \qquad \text{Equation 34}$$

For the purpose of propagating the covariance matrix and performing the measurement update, an error state vector is used. The quaternion portion is replaced with the vector portion of a small error quaternion that represents the error between the estimate and the sensor measurement.

$$\boldsymbol{X}_e = \begin{bmatrix} (\delta \boldsymbol{q}_e)_v \\ \boldsymbol{\omega}_{rel} \\ \boldsymbol{r}_{T,C} \\ \boldsymbol{v}_{T,C} \end{bmatrix} \qquad \text{Equation 35}$$

Rotational State Equations

Using the quaternion $\boldsymbol{q}$ to represent the rotation of the target satellite with respect to the chaser satellite, the relation between them can be seen in Equation 36:

$$\dot{\boldsymbol{q}} = \frac{1}{2}\underline{\boldsymbol{\omega}}_{rel} \otimes \boldsymbol{q} \qquad \text{Equation 36}$$

where $\underline{\boldsymbol{\omega}}_{rel}$ represents the relative angular velocity between the two satellites as a 4 by 1 column matrix and is another formulation of Equation 10 and Equation 11:

$$\underline{\boldsymbol{\omega}}_{rel} = \begin{bmatrix} \boldsymbol{\omega}_{rel} \\ 0 \end{bmatrix} \qquad \text{Equation 37}$$

$$\boldsymbol{\omega}_{rel} = \boldsymbol{\omega} - \boldsymbol{\omega}_n \qquad \text{Equation 38}$$

The angular velocity of the chaser satellite is assumed to be equal to the angular velocity of the reference orbit so that the chaser satellite is earth pointing at all times.

$$\boldsymbol{n} = \begin{bmatrix} 0 \\ 0 \\ n_z \end{bmatrix} \qquad \text{Equation 39}$$

Using $\boldsymbol{n}$ to denote the angular velocity of the reference orbit, Equation 40 can be used to resolve the angular velocity of the chaser satellite in the appropriate frame:

$$\boldsymbol{\omega}_n = \boldsymbol{q} \otimes \underline{\boldsymbol{n}} \otimes \boldsymbol{q}^* \qquad \text{Equation 40}$$

where $\boldsymbol{q}^*$ is the reference quaternion representing the rotation of the target satellite with respect to the chaser satellite. Substituting Equation 40 into Equation 36 and linearizing about the estimated states $\overline{\boldsymbol{q}}$ and $\overline{\boldsymbol{\omega}}$, we arrive at:

$$\frac{d}{dt}\delta\boldsymbol{q}_v = -\boldsymbol{\omega}^{\times}\delta\boldsymbol{q}_v + \frac{1}{2}\delta\boldsymbol{\omega} \qquad \text{Equation 41}$$

$$\frac{d}{dt}\delta q_0 = 0 \qquad \text{Equation 42}$$

where $\delta\boldsymbol{q}_v$ and δq_0 represent small changes from the reference quaternion and the formulation for δq_0 can be ignored as it is not an independent variable.

Using Euler's formulation, the dynamics of the rotational motion of the target satellite can be expressed as Equation 43:

$$\dot{\boldsymbol{\omega}} = \boldsymbol{\psi}(\boldsymbol{\omega}) + \boldsymbol{J}\boldsymbol{\epsilon}_\tau \qquad \text{Equation 43}$$

Where $\boldsymbol{\psi}(\boldsymbol{\omega})$ is a 3 by 1 column matrix

$$\boldsymbol{\psi}(\boldsymbol{\omega}) = \begin{bmatrix} p_x\omega_y\omega_z \\ p_y\omega_x\omega_z \\ p_z\omega_x\omega_y \end{bmatrix} \qquad \text{Equation 44}$$

and

$$\boldsymbol{J} = diag\left(1, \frac{I_{xx}}{I_{yy}}, \frac{I_{xx}}{I_{zz}}\right) \qquad \text{Equation 45}$$

$$p_x = \frac{I_{yy} - I_{zz}}{I_{xx}} \qquad \text{Equation 46}$$

$$p_y = \frac{I_{zz} - I_{xx}}{I_{yy}} \qquad \text{Equation 47}$$

$$p_z = \frac{I_{xx} - I_{yy}}{I_{zz}} \qquad \text{Equation 48}$$

Above, I_{xx}, I_{yy} and I_{zz} are the principal moments of inertia of the target satellite. Using small perturbations we get:

$$\frac{d}{dt}\delta\boldsymbol{\omega} = \boldsymbol{A}(\overline{\boldsymbol{\omega}})\delta\boldsymbol{\omega} + \boldsymbol{J}\boldsymbol{\epsilon}_\tau \qquad \text{Equation 49}$$

where

$$\boldsymbol{A}(\overline{\boldsymbol{\omega}}) = \begin{bmatrix} 0 & p_x\omega_z & p_x\omega_y \\ p_y\omega_z & 0 & p_y\omega_y \\ p_z\omega_y & p_z\omega_x & 0 \end{bmatrix} \qquad \text{Equation 50}$$

Combining Equation 41 and Equation 50, the state space representation is:

$$\frac{d}{dt}\delta\boldsymbol{x}_f = \begin{bmatrix} -\overline{\boldsymbol{\omega}}^\times & \frac{1}{2}\boldsymbol{I}_3 \\ \boldsymbol{0}_{3x3} & \boldsymbol{A}(\overline{\boldsymbol{\omega}}) \end{bmatrix}\delta\boldsymbol{x}_f + \begin{bmatrix} \boldsymbol{0}_{3x1} \\ \boldsymbol{J}\boldsymbol{\epsilon}_f \end{bmatrix} \qquad \text{Equation 51}$$

Translational State Equations

The relative velocity between the center of mass of the chaser and target satellites can be described using Equation 52 as following:

$$\dot{\boldsymbol{v}} = -2\boldsymbol{n}^\times\boldsymbol{v} + \boldsymbol{\eta}(\boldsymbol{r},\boldsymbol{n}) + \boldsymbol{\epsilon}_f \qquad \text{Equation 52}$$

where

$$\boldsymbol{\eta}(\boldsymbol{r},\boldsymbol{n}) = \begin{bmatrix} 3r_x \\ 0 \\ -r_z \end{bmatrix} n_z^2 \qquad \text{Equation 53}$$

Assuming that $\dot{\boldsymbol{r}} = \boldsymbol{v}$ and arranging the above into the state space representation, we get:

$$\frac{d}{dt}\delta\boldsymbol{y}_f = \begin{bmatrix} \boldsymbol{0}_{3x3} & \boldsymbol{I}_3 \\ \boldsymbol{N} & -2\boldsymbol{n}^{\times} \end{bmatrix}\delta\boldsymbol{y}_f + \begin{bmatrix} \boldsymbol{0}_{3x1} \\ \boldsymbol{\epsilon}_f \end{bmatrix} \qquad \text{Equation 54}$$

where

$$\boldsymbol{N} = \frac{\partial\boldsymbol{\eta}}{\partial\boldsymbol{r}} = \begin{bmatrix} 3n_z^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -n_z^2 \end{bmatrix} \qquad \text{Equation 55}$$

Measurement Model

The vision system installed on the chaser satellite measures the orientation of the target satellite with respect to the orientation of the LCS and the position of the point of reference on the target satellite with respect to the position of the LCS frame of reference. The measurement vector, $\boldsymbol{z}$, used by the EKF consists of the relative distance between the center of mass of the target satellite and the chaser satellite and the vector portion of the quaternion that represents the rotation between the estimated orientation of the target satellite and the LCS measurement of the orientation of the target satellite.

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{r}_{T,C} \\ (\boldsymbol{\delta q})_v \end{bmatrix} \qquad \text{Equation 56}$$

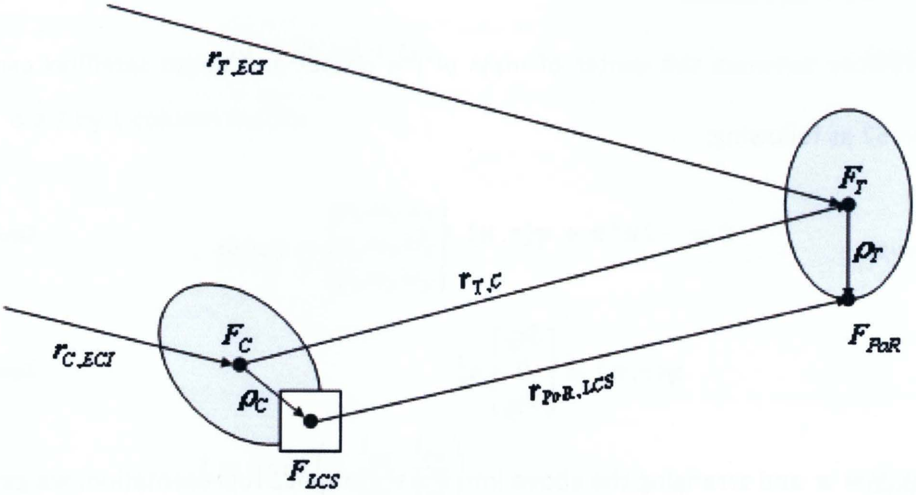

Figure 5: Physical System Representation

From the kinematics of the system defined in Figure 5 we have:

$$\boldsymbol{r}_{PoR,LCS} = -\boldsymbol{\rho}_C + \boldsymbol{r}_{T,C} + \boldsymbol{R}_{T,C}\rho_T \qquad \text{Equation 57}$$

Using Equation 56 and Equation 57, the linearized measurement sensitivity matrix can be found to be:

$$\boldsymbol{H} = \begin{bmatrix} -2\boldsymbol{R}(\boldsymbol{q})(\boldsymbol{\rho}_t \times \boldsymbol{\delta q}_v) & \boldsymbol{0}_{3x3} & \boldsymbol{I}_3 & \boldsymbol{0}_{3x3} \\ \boldsymbol{I}_3 & \boldsymbol{0}_{3x3} & \boldsymbol{0}_{3x3} & \boldsymbol{0}_{3x3} \end{bmatrix} \qquad \text{Equation 58}$$

Tracking Architectures

The main contribution made in this thesis is the implementation and examination of two different tracking architectures for the purpose of satellite tracking in the orbital environment. The two different tracking architectures can be seen in Figure 6. The first tracking architecture is referred to as the *open loop system* and consists of the LCS that uses the ICP algorithm to match a point cloud obtained from scanning the target satellite to its CAD model. The initial pose used for the ICP algorithm in the open loop system is the result of the previous sensor measurement.

The *closed loop system* consists of the LCS that uses the ICP algorithm to match a point cloud obtained from scanning the target satellite to its CAD model, and the EKF that is used to estimate the initial pose for the ICP algorithm, which then uses the ICP pose output as the sensor measurement to propagate the estimate of the target's pose for the next sensor reading.

The two tracking architectures differ in the respect that the closed loop tracking architecture establishes a two-way link between the EKF and the ICP algorithm. This link is comprised of using the predictive capability of the EKF within the framework for tracking. The EKF is used to predict the pose of the object at the next sensor cycle and use this prediction as an estimate of the target's relative pose for the ICP algorithm, this process provides an intelligent estimate of the target's relative pose.

In the open loop architecture, the system passively tracks the object using the previous pose of the object as the initial guess. In the closed loop architecture, the system actively tracks the object using an estimated pose of the object as the initial guess. With the closed loop, the system is provided with a more accurate pose initial guess of target's pose.

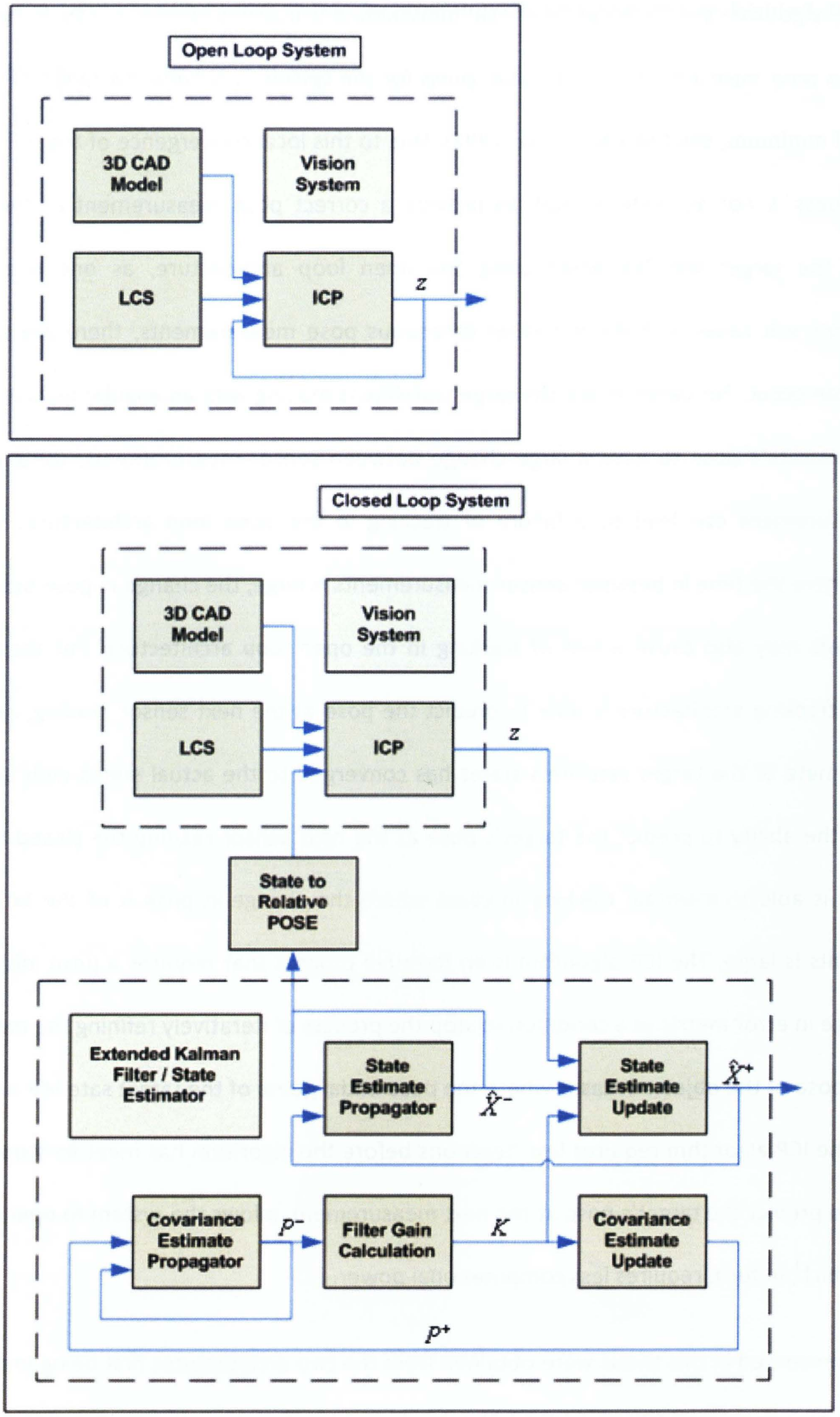

Figure 6: Architecture used in the open loop and closed loop configurations

Since the ICP algorithm can converge to a local minimum, it is possible for the ICP algorithm to provide an erroneous pose measurement if the initial guess for the system is outside the range of convergence of the global minimum, see (Besl & McKay, 1992). Due to this local convergence of the ICP algorithm, if the initial guess is not accurate enough to provide a correct pose measurement of the target, the tracking of the target will fail when using the open loop architecture, as one erroneous pose measurements will cause a chain of further erroneous pose measurements, there are several cases where this can occur. For cases where the target satellite is moving with an angular velocity that causes the target satellite's pose to have a large change between sensor measurements, using the previous sensor measurement can lead to a failure of tracking in the open loop architecture. Similarly, for situations where the time in between sensor measurements is large, the change in pose between sensor measurements may also cause a loss of tracking in the open loop architecture. For these cases, the closed loop tracking architecture is able to predict the pose at the next sensor reading, assuming that the EKF estimate of the target satellite's states has converged to the actual states with an acceptable error. With the ability to predict the target's pose at the next sensor reading the closed loop tracking architecture is able to maintain tracking in cases where the change in pose is of the target between measurements is large. The ICP algorithm is an iterative process that requires a pose initial guess and uses a change in error metric as a condition to stop the process of iteratively refining the measured pose to the true pose of the object. In cases where the pose initial guess of the target satellite is closer to the true pose, the ICP algorithm requires less iterations before the algorithm has meet its stopping criteria. Being able to predict the target's pose at the next measurement, allows the system to operate using less iterations which, in turn, requires less computational power.

The results presented in this thesis were obtained from the two architectures first being implemented in the MATLAB Simulink environment and then implemented in laboratory experimentation using the CSA CART system to create relative orbital movement between a QuickSat mock up used as the target satellite and the Neptec's LCS as the sensor to provide pose measurement to be used as the sensor readings for the EKF.

Chapter IV. Simulation, Experimental Setup and Results

The method used to examine the benefit of using the EKF in conjunction with the ICP algorithm consists of two phases. The first phase takes place within simulation in the MATLAB Simulink environment, while the second phase takes place using the CART system at CSA and the Neptec's LCS. For both phases, the two algorithms were tested versus each other in situations where fast target movement and full sensor occlusions were present.

Simulation

The first phase of study was preformed completely in simulation. The true orbital environment was simulated using MATLAB Simulink to drive a truth model of the physical system. The truth model was used to provide the position and orientation of the chaser satellite and the LCS as well as the position of the target satellite. Using the knowledge provided by the truth model, a simulated LCS and a CAD model of the target satellite, as seen in Figure 7, were used to generate a point cloud which was then used in the ICP algorithm to provide a sensor reading of the target satellite's relative position and orientation. For the open loop system, this sensor measurement was used as the initial guess of the target's pose for the next sensor cycle while for the closed loop architecture this sensor measurement was used as an input for the EKF.

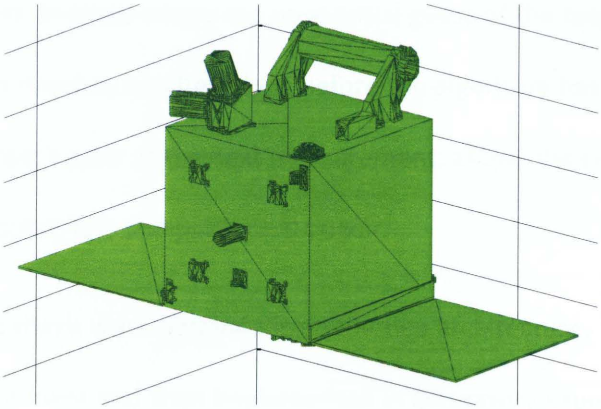

Figure 7: CAD Model representation of target satellite, QuickSat, used in simulation and experimentation

For the simulated phase both tracking architectures were implemented and tested using the described simulated environment above. The initial conditions for the test are:

$$\boldsymbol{q} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \boldsymbol{\omega} = \begin{bmatrix} 0.0175 \\ -0.0524 \\ 0.0524 \end{bmatrix} \frac{rad}{s}$$

$$\boldsymbol{r} = \begin{bmatrix} 0 \\ 0 \\ 10 \end{bmatrix} m \quad \boldsymbol{v} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \frac{m}{s}$$

where $\boldsymbol{q}$, $\boldsymbol{\omega}$, $\boldsymbol{r}$ and $\boldsymbol{v}$ are the target satellite's relative distance, orientation, linear and angular velocity to the chaser satellite. The simulation was run for 100 seconds with the sensor operating at 1 Hz and a sensor occlusion period starting at 40 seconds into the simulation and ending at 70 seconds. At 70 seconds the sensor occlusion was removed and sensor readings continued at a rate of 1 Hz until the end of the simulation time of 100 seconds.

Experimental Setup

For the second phase, the experimentation was performed in the laboratory environment at the CSA using the CART system. The CART system can be seen in the picture shown in Figure 8, a mock up of the QuickSat satellite and the Neptec's LCS, which can be seen in the picture shown in Figure 8.

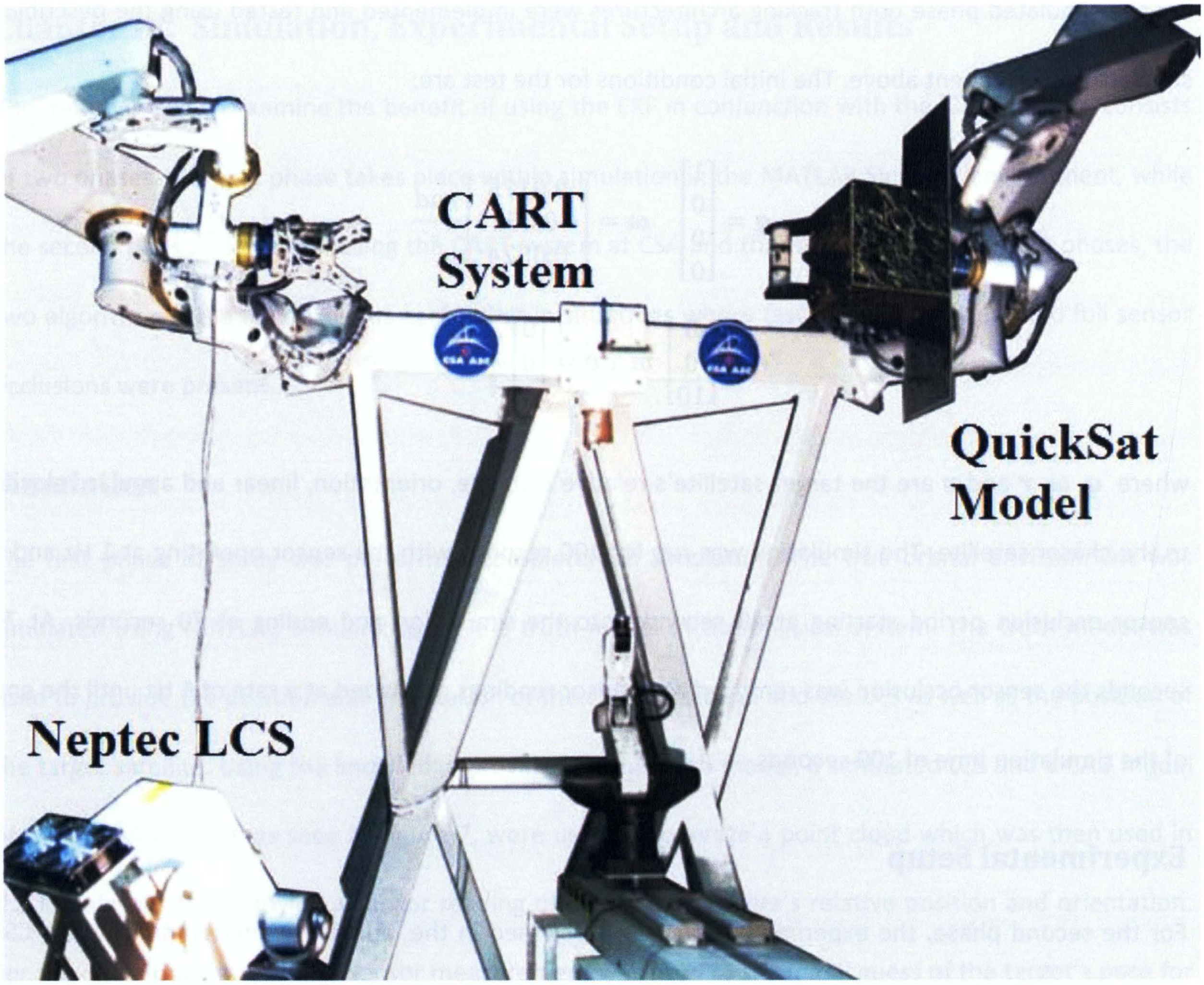

Figure 8: Physical Experimental Setup

The CART system consists of two robotic arms with seven prismatic joints attached to a frame. The robotic arms consist of seven revolute joints and seven spars. The QuickSat model is attached to one of the robotic arms of the CART system, while the LCS is positioned at a fixed distance and orientation from the CART System. The orbital environment is simulated by using the robotic manipulator to move the QuickSat model relative to the LCS in order to replicate the relative orbital motion of the target satellite with respect to the LCS. The QuickSat is moving relative to the LCS which scans the model and performs the ICP algorithm. The computed pose is then provided to the EKF.

The experimental setup is used to reproduce the orbital environment by moving the satellite model with respect to the LCS in the same manner that would be experienced in the orbital environment. Figure 9 shows a graphical representation of CART and the target and chaser satellites in the orbital environment.

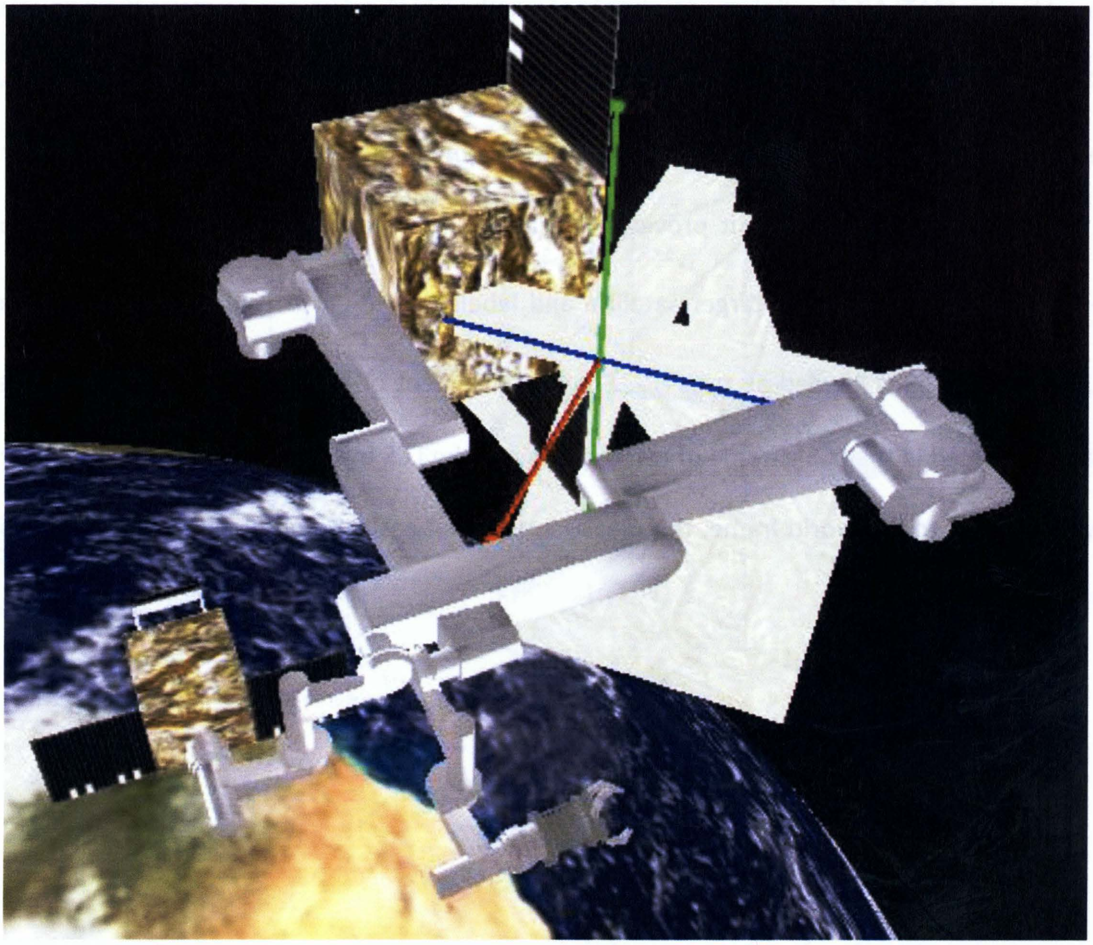

Figure 9: Graphical representation of apparatus and physical system simulated

This graphic demonstrates the relation between the true environment being replicated and the CART System. In the graphic the CART system is shown orbiting the earth while recreating the relative motion between the target satellite and the LCS. The axis seen in Figure 9 and labeled as F_W in Figure 10 is called the *world frame* and corresponds to the point on the CART system directly in between the STM1 and STM2 arms.

In order to replicate the physical system in experimentation, a trajectory file containing the kinematics and dynamics of the physical system at discrete time intervals is generated ahead of experimentation. The trajectory file contains the linear and angular position, velocity and acceleration of the of the end effecter of the STM1 arm and these values are expressed in the world frame of the robot. This file is then used by the MATLAB Simulink model that controls the STM1 robot arm to move the end effector of the STM1 robotic arm in the desired manner, which accurately replicates the relative orbital movement between the target satellite and the LCS. When the LCS measures the position and orientation of the target satellite, the measurement it provides is that of the distance and orientation of the point of reference frame located on the target satellite and labeled as F_{PoR} in Figure 10, the distance being measured by the LCS is labeled as $\boldsymbol{r}_{PoR,LCS}$. The measurement of the point of reference of the target satellite is obtained in the LCS's frame of reference. Since the LCS frame and chaser satellite's frames are fixed with respect to the world frame, the measurement can easily be obtained in the chaser satellite's frame.

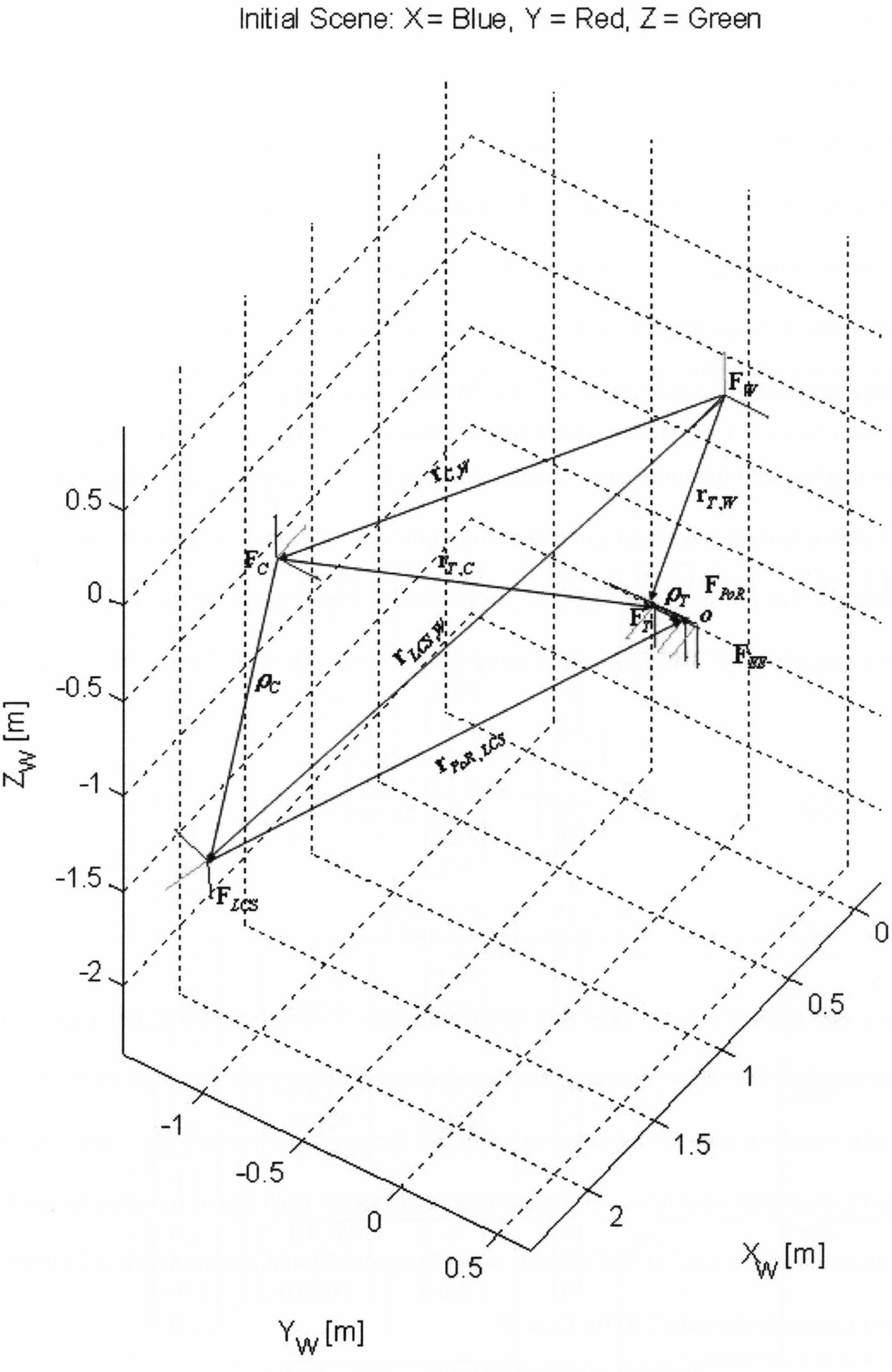

Figure 10: Experimental system setup at initial conditions

Figure 10 shows the experimental setup at initial conditions. In this figure the frames of reference of the system and constant kinematics are shown in the world frame labeled $\boldsymbol{F_W}$. The world frame is positioned directly between the two robotic arms on the CART system. The distance from the world frame to the chaser satellite frame, $\boldsymbol{r_{C,W}}$, and the distance from the world frame to the LCS frame, $\boldsymbol{r_{LCS,W}}$, is constant. The Neptec LCS measures the distance and orientation between the LCS frame, $\boldsymbol{F_{LCS}}$, and the Point of Reference frame, $\boldsymbol{F_{PoR}}$. The EKF estimates the distance, orientation, linear and angular velocity between the chaser frame, $\boldsymbol{F_C}$, and the target frame $\boldsymbol{F_T}$.

Two different tracking architectures were tested using five different cases. These cases tested the two architectures ability to track the target satellite at four different rotational rates and in situations where multiple occlusions are present. The first four cases focus on testing the ability of the two tracking architectures to track an object moving at increasing rates of speed. The initial conditions for Case I are:

$$\boldsymbol{q} = \begin{bmatrix} 0.5 \\ 0.5 \\ -0.5 \\ 0.5 \end{bmatrix} \quad \boldsymbol{\omega} = \begin{bmatrix} -0.0182 \\ 0.0455 \\ 0.0073 \end{bmatrix} \frac{rad}{s}$$

$$\boldsymbol{r} = \begin{bmatrix} 0 \\ 0.85 \\ 0.4 \end{bmatrix} m \quad \boldsymbol{v} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \frac{m}{s}$$

Where $\boldsymbol{q}$, $\boldsymbol{\omega}$, $\boldsymbol{r}$ and $\boldsymbol{v}$ are the target satellite's relative distance, orientation, linear and angular velocity to the chaser satellite. The initial conditions for Case II through Case IV are the same as those of Case I except that the rotational rate of the target satellite is a factor of the one used for Case I. For Case II through Case IV, the factor used is two, three and four respectively. Each case is run once for each of the tracking architectures for a total of 300 seconds with no occlusions and a sensor rate of 2 Hz for Case I through Case III and a sensor rate 1 Hz for Case IV.

The initial conditions for Case V are the same as those of Case IV and can be seen below:

$$q = \begin{bmatrix} 0.5 \\ 0.5 \\ -0.5 \\ 0.5 \end{bmatrix} \quad \omega = \begin{bmatrix} -0.0728 \\ 0.1820 \\ 0.0291 \end{bmatrix} \frac{rad}{s}$$

$$r = \begin{bmatrix} 0 \\ 0.85 \\ 0.4 \end{bmatrix} m \quad v = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \frac{m}{s}$$

Case V differs from that of Case IV in the way that now occlusions of the LCS are added in order to test the two different tracking architectures ability to manage periods where the sensor is not able to provide a measurement of the object. The conditions for the five laboratory test cases have been gathered and presented in Table 1.

Case	q	$\omega \left[\frac{rad}{s}\right]$	$r\ [m]$	$v \left[\frac{m}{s}\right]$	Occlusions	Time [s]
I	$\begin{bmatrix} 0.5 \\ 0.5 \\ -0.5 \\ 0.5 \end{bmatrix}$	$\begin{bmatrix} -0.0182 \\ 0.0455 \\ 0.0073 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0.85 \\ 0.4 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	No	300
II	$\begin{bmatrix} 0.5 \\ 0.5 \\ -0.5 \\ 0.5 \end{bmatrix}$	$\begin{bmatrix} -0.0364 \\ 0.0910 \\ 0.0146 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0.85 \\ 0.4 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	No	300
III	$\begin{bmatrix} 0.5 \\ 0.5 \\ -0.5 \\ 0.5 \end{bmatrix}$	$\begin{bmatrix} -0.0546 \\ 0.1365 \\ 0.0219 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0.85 \\ 0.4 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	No	300
IV	$\begin{bmatrix} 0.5 \\ 0.5 \\ -0.5 \\ 0.5 \end{bmatrix}$	$\begin{bmatrix} -0.0728 \\ 0.1820 \\ 0.0291 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0.85 \\ 0.4 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	No	300
V	$\begin{bmatrix} 0.5 \\ 0.5 \\ -0.5 \\ 0.5 \end{bmatrix}$	$\begin{bmatrix} -0.0728 \\ 0.1820 \\ 0.0291 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0.85 \\ 0.4 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$	Yes	300

Table 1: Initial conditions for test cases I through V

Simulation Results

In the MATLAB Simulink simulation part of the experimentation, the link between the EKF and the ICP algorithm was demonstrated. It was shown that in simulation the EKF was able to provide an improved pose initial guess for the ICP algorithm, high speed object pose tracking and an increase of robustness when using a closed loop configuration instead of the open loop configuration. The improved initial guess for the ICP algorithm using the closed loop configuration can be seen in Figure 11, where the number of iterations for each use of the ICP algorithm is plotted versus simulation time.

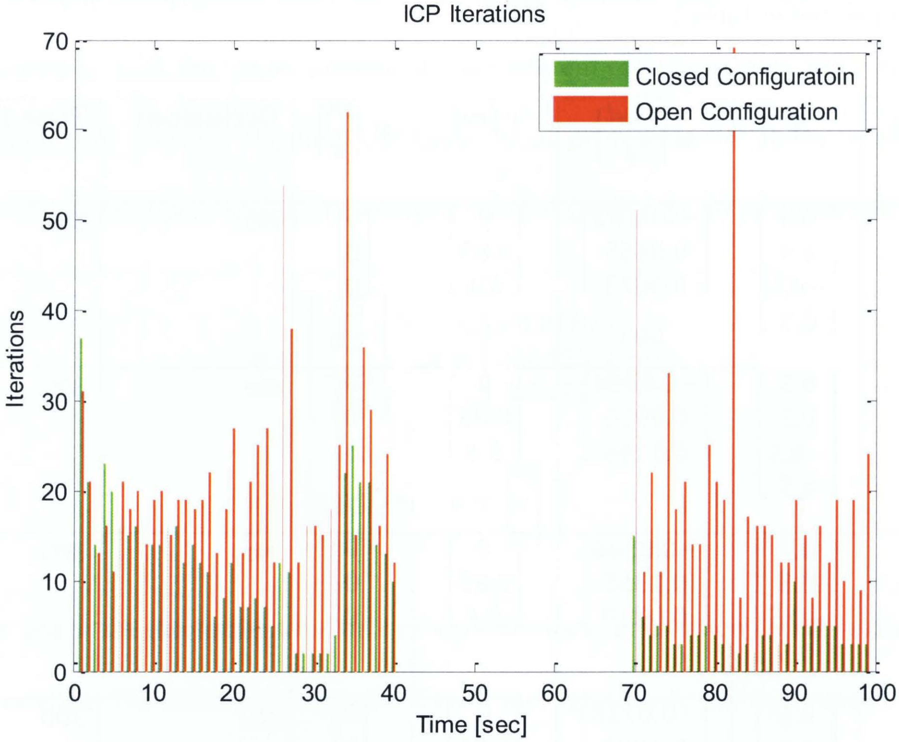

Figure 11: Comparison between iterations required for ICP convergence

It can be seen that once the EKF's estimate of the system states converges, the total number of iterations required to track the target object decreases. This is due to the estimate of the target object's relative pose to the simulated LCS being more accurate than the previous sensor reading, which is what is used as the initial guess in the open loop configuration. In addition to the increased accuracy in the initial guess for each sensor cycle, the increased robustness of the closed loop configuration can be seen in Figure 11. At a simulation time of 40 seconds, the simulated LCS is blocked and unable to provide a new sensor reading for 30 seconds. At a simulation time of 70 seconds, when the sensor reading is resumed tracking of the object is resumed in the closed loop configuration, while tracking of the target object fails in the open loop configuration, this can be seen in Figure 12.

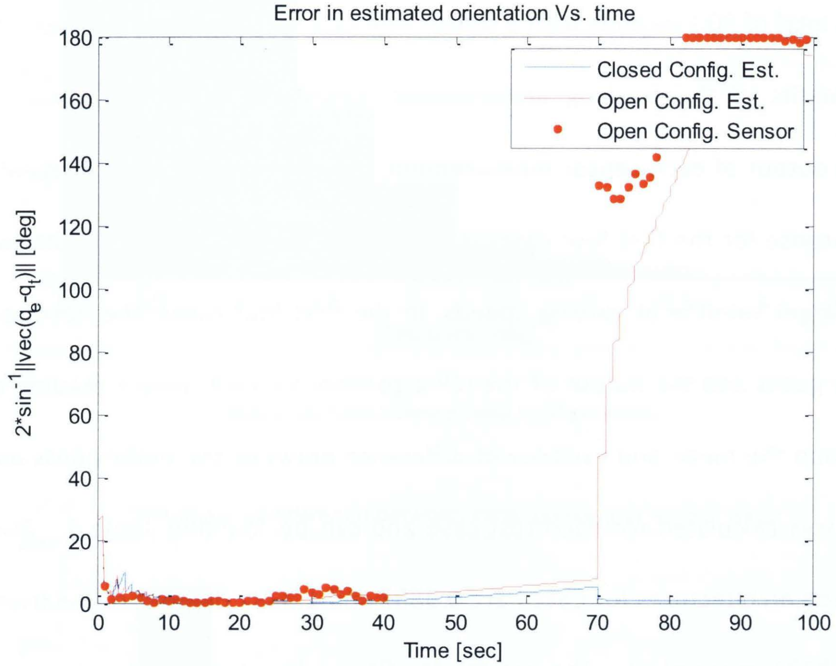

Figure 12: Error in Orientation for Open and Closed Configurations and Sensor Measurements

Figure 12 shows the error in orientation for the open and closed loop architectures as well as the error in the sensor measurements for the open loop architecture. In this figure the failure of the open loop tracking can be seen at 70 seconds into the simulation when the open configuration sensor reading error jumps to 131 degrees. This is a result of the EKF being able to provide an accurate enough estimate of the target's relative pose to the LCS for the ICP algorithm to converge to the correct local minimum. For the open loop configuration the tracking fails. This is because the relative pose used as the initial guess is the previous pose from the sensor reading that occurred 30 seconds. This causes the ICP algorithm to converge to the wrong local minimum providing an erroneous pose measurement. This erroneous pose measurement is then used as the initial guess for the next sensor reading which causes the next sensor reading to be erroneous which continues to be propagated from one sensor reading to the next.

Experimentation Results

In the laboratory experimentation, the five different test cases were run for each of the tracking architectures for a total of 300 seconds with a sensor rate of 1 Hz. The data gathered for the purpose of examining the benefits of the tracking architectures consists of the initial guess for each sensor measurement, the output of each sensor measurement and the states of the target satellite estimated by the EKF. The purpose for the first four cases is to observe the performance of the two architectures ability to track a target satellite at varying speeds. In the first four cases, the norm of the difference between the initial guess and the output of the ICP algorithm for each sensor reading was plotted as a histogram. In addition the mean and variance of difference between the initial guess and the output of the ICP algorithm was calculated for each test case and can be found in Table 2. The means for the closed and open loop architectures for Case I are 0.9807 and 1.3332 degrees respectively, the variances are 0.4334 and 0.8271 respectively. The results for Case I for the closed loop and the open loop architecture can be found in Figure 13 and Figure 14 respectively. It is seen that for cases where the rotational rate of the target satellite is relatively low, 1-2 degrees per second about the body axis the mean and variance between the initial guess for the ICP algorithm and the output for both cases is lower for the closed loop algorithm.

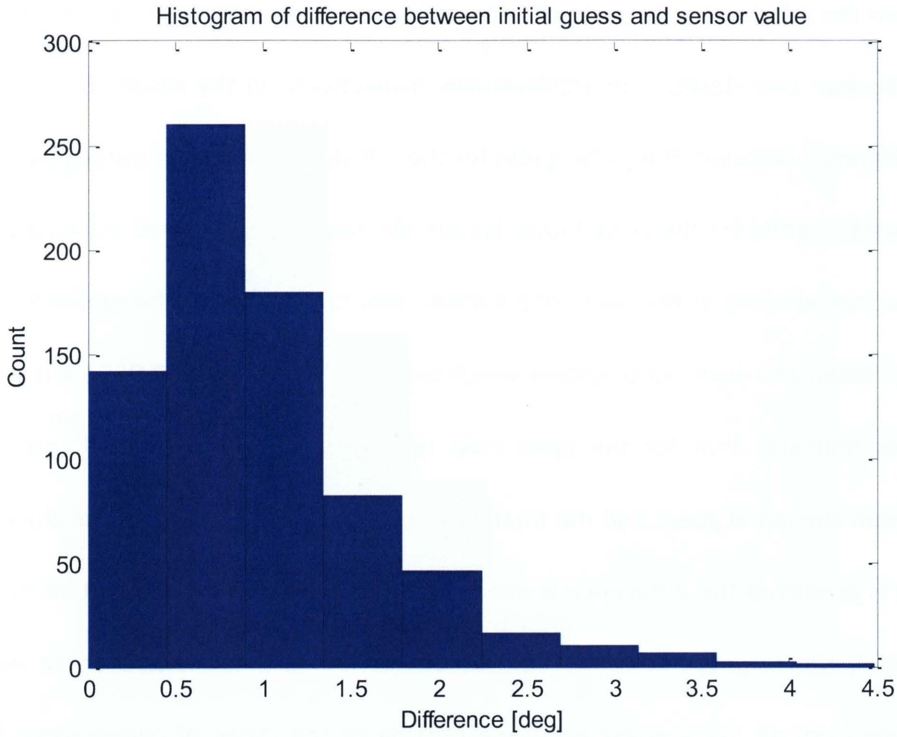

Figure 13: Case I Closed Loop Configuration

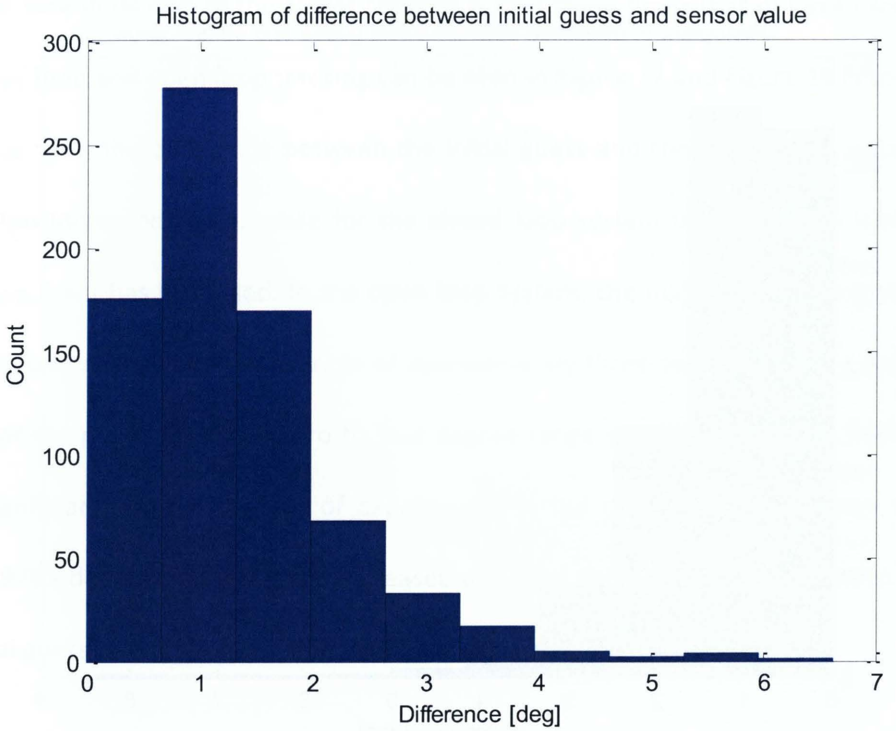

Figure 14: Case I Open Loop Configuration

For Case II, where the rotational rate is double that of Case I, the results can be seen in Figure 15 and Figure 16 for the open and closed loop architectures respectively. In the results for this set, it can be seen that the difference between the initial guess for the ICP algorithm and its output have increased for the open loop system while for the closed loop system the results have stayed increased to that of the pervious case but not as much as the open loop system. This can be seen in the values for the mean and variance for the closed and open loop system which are 1.311, 0.8933 and 2.0652 and 1.3319 degrees respectively. This indicates that for the open loop system, when the rotational rate increases, the difference between the initial guess and the final pose value increases, while for the closed loop system where the pose is predicted the difference is minimized as was seen in the simulation results. For both Case I and Case II, the target's motion is able to be tracked without failure or an erroneous pose estimate resulting from an initial guess occurring outside of the range of convergence for the global minimum.

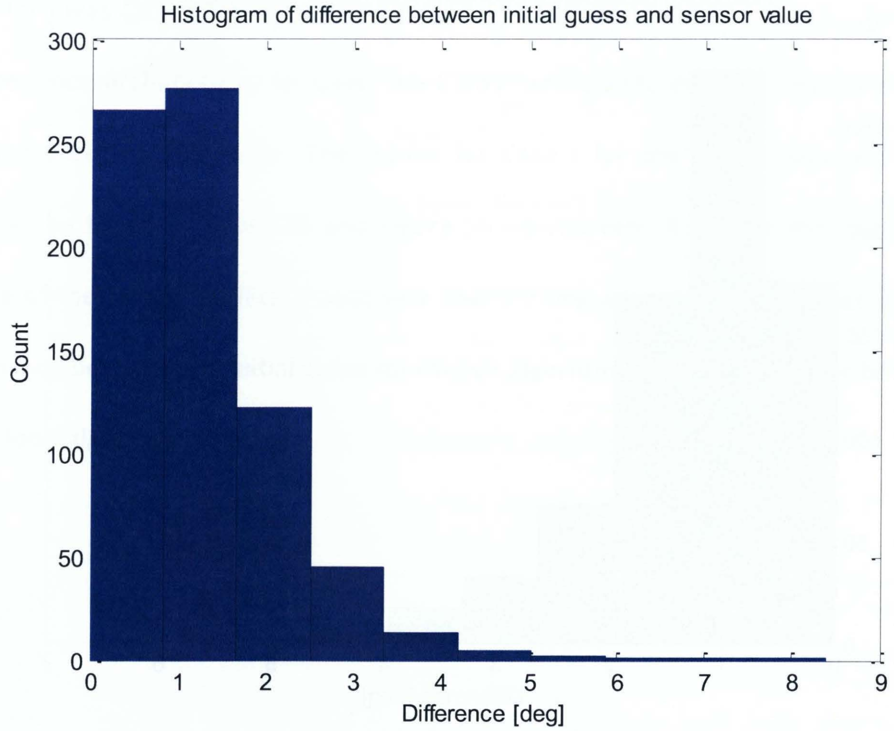

Figure 15: Case II Closed Loop Configuration

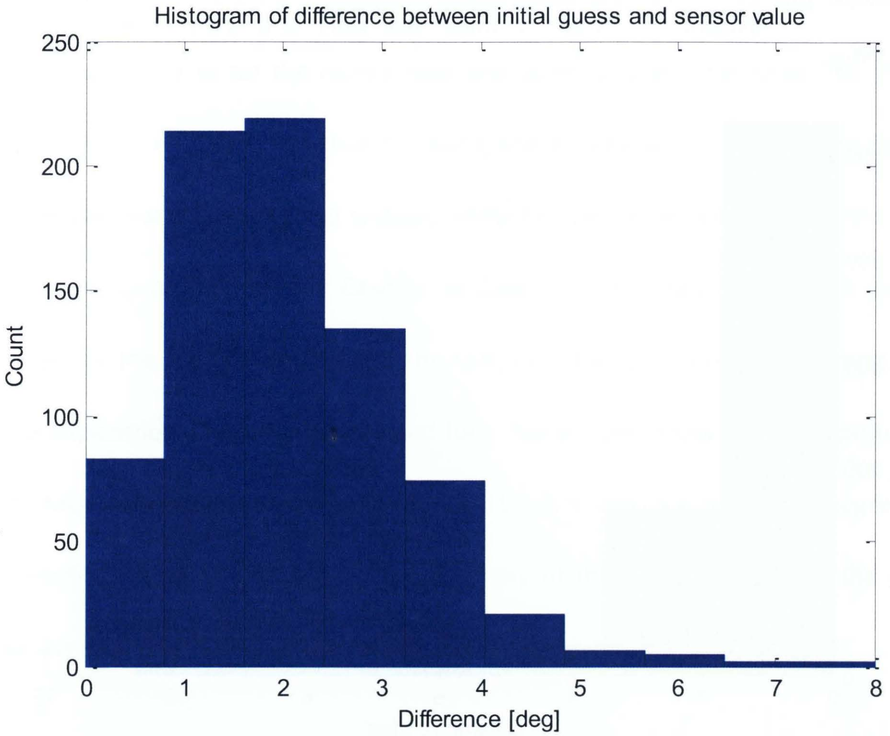

Figure 16: Case II Open Loop Configuration

In Case III, the rotational rate of the target satellite is now three times that of Case I and the results for both the closed loop and open loop tracking can be seen in Figure 17 and Figure 18 respectively. As seen in the previous case, the difference between the initial guess and the final sensor output for the open loop system has increased again, while for the closed loop system the difference is similar to that of Case I and Case II but has increased. In the open loop system, the majority of the points are now in the range seven to one degree with an average of approximately three degrees. For the closed loop system the majority of the points lie in the zero to four degree range with an average of 1.6665 degrees. This increase is significantly less than that of experienced in the case of the open loop system, with an average of 2.9785 degrees, showing the increased ability of the closed loop system to provide a more accurate initial guess to the ICP algorithm then the open loop architecture.

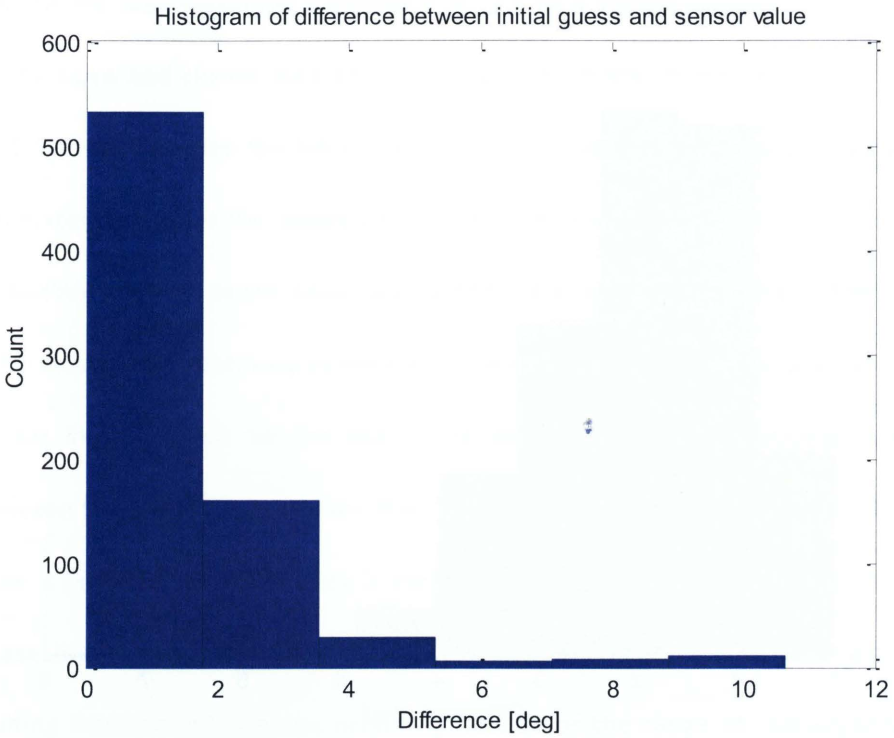

Figure 17: Case III Closed Loop Configuration

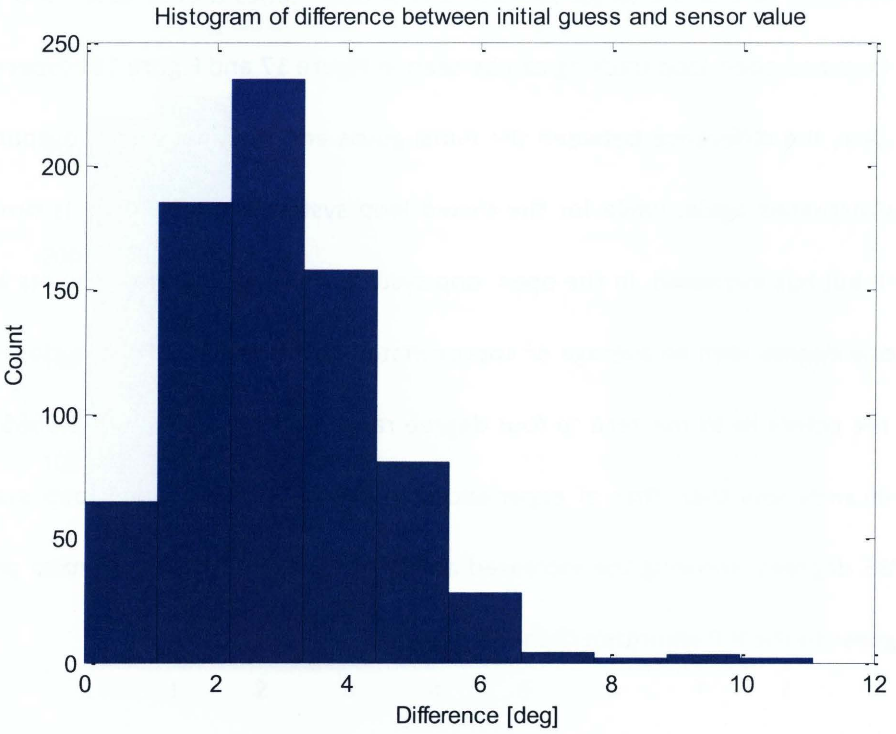

Figure 18: Case III Open Loop Configuration

In Case IV, the results of the difference between initial guess and output for the sensor reading can be seen in Figure 19 and Figure 20 for the closed loop and open loop architectures. The rotational velocity of the target satellite is now four times that of Case I, and as was seen in Case II and Case III the spread of points has increased for the open loop system, while for the closed loop system the spread of points has remained the similar as that seen in Case III. In Case IV, the closed loop system successfully tracks the target satellite for the full 300 seconds of the test, but the open loop architecture fails tracking 60 seconds into the experimentation. For the closed loop system the majority of the points fall within the range of zero to four degree with an average of 2.2981 and a variance of 4.4364 degrees. For the open loop system, before tracking failure occurs, the majority of the points fall within the range of 8 to 14 degrees with an average of 10.1919 and a variance of 13.1388 degrees.

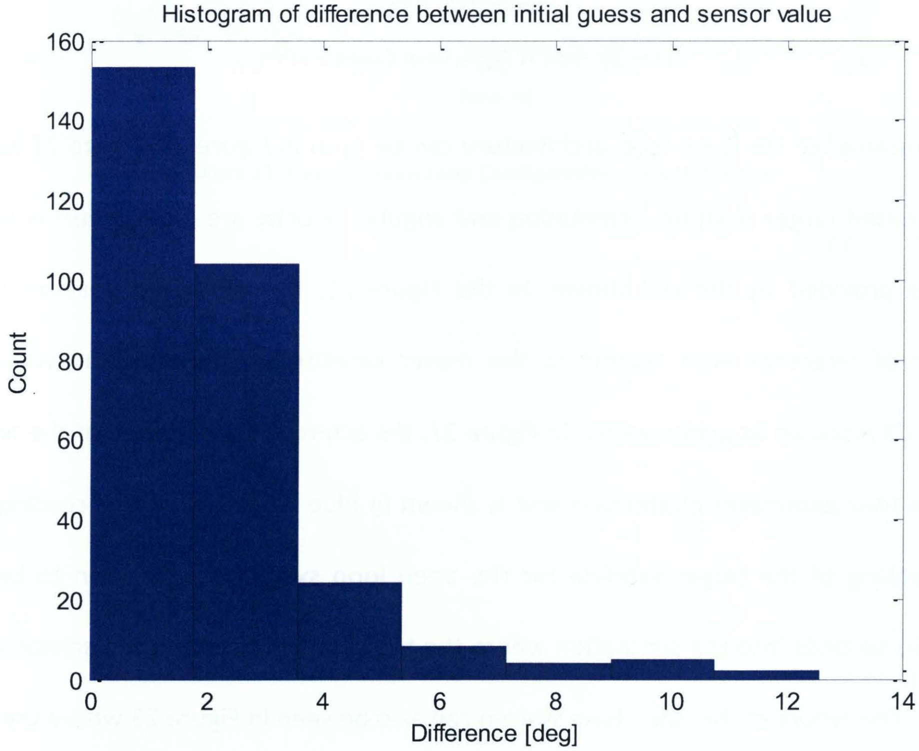

Figure 19: Case IV Closed Loop Configuration

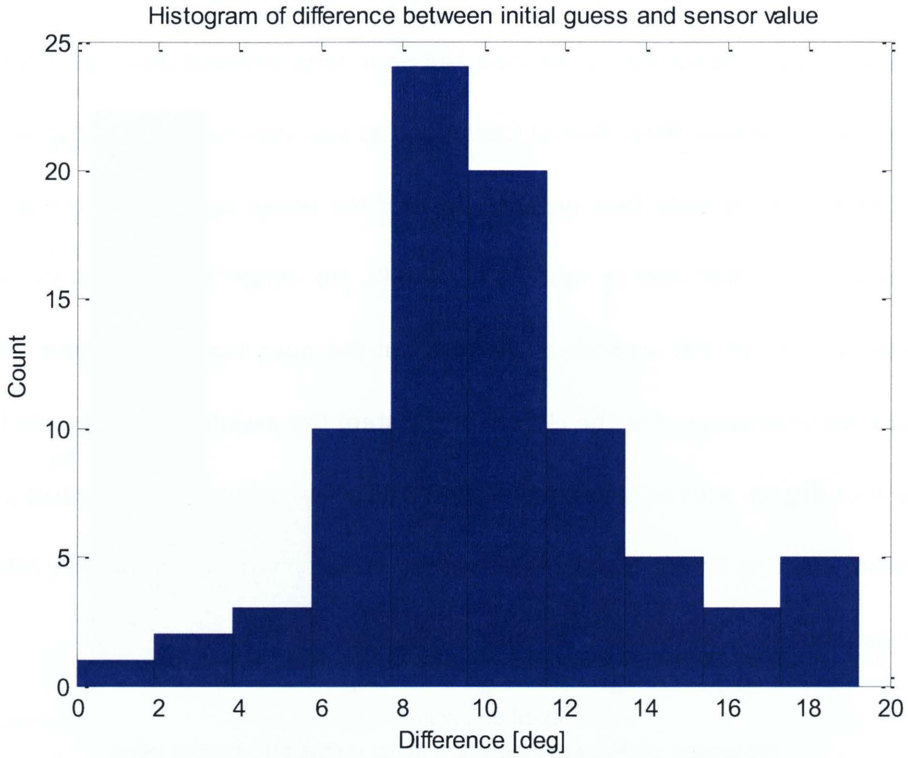

Figure 20: Case IV Open Loop Configuration

The failure in tracking for the open loop architecture can be seen in Figure 21, Figure 22 and Figure 23, where the estimated target position, orientation and angular velocity are plotted versus time with the sensor readings provided by the LCS shown. In the Figure 21, the estimated position of the target satellite's point of reference with respect to the chaser satellite is shown in blue while the sensor reading of the LCS is shown as green points. In Figure 22, the estimated orientation of the target satellite is shown as the four parameter quaternion and is shown in blue while the sensor reading is shown in green. The tracking of the target satellite for the open loop system can be seen to be performing properly until 60 seconds into the simulation where the trend of the filter and the sensor in the figures become erratic. The failure of the open loop system can also be seen in Figure 23 where the true angular velocity of the target satellite is shown in green and the estimated angular velocity of the target satellite is shown in green. In this figure it can be seen that the target satellite is being successfully tracked by the open loop system as the estimated angular velocity follows the trend of the true angular velocity, when the tracking of the system fails the chain of erroneous sensor readings causes the estimate of the target's angular velocity to diverge from the true angular velocity.

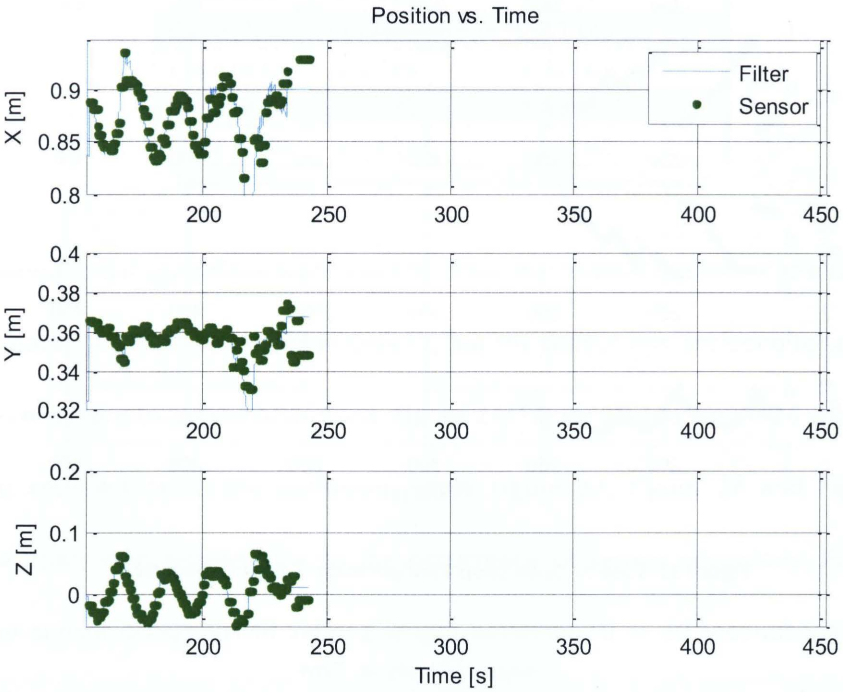

Figure 21: Case IV Open Loop Configuration, Target Position

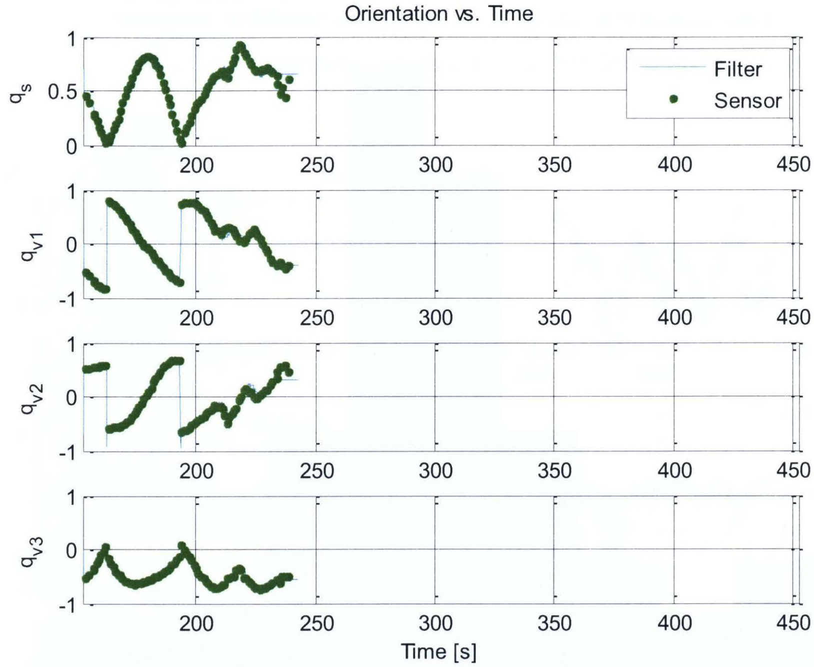

Figure 22: Case IV Open Loop Configuration, Target Orientation

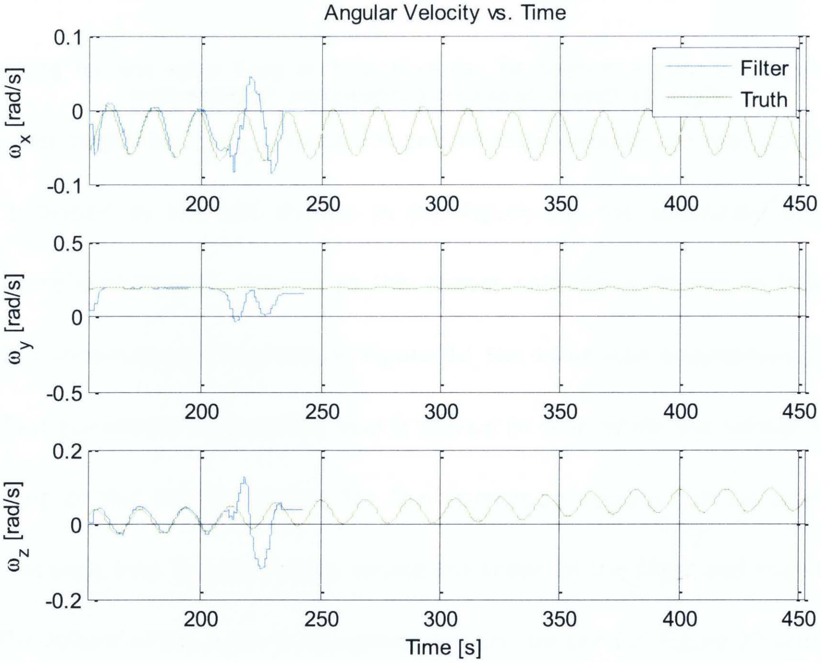

Figure 23: Case IV Open Loop Configuration, Target Angular Velocity

Case	Mean [deg]	Variance [deg^2]
Case I CL	0.9807	0.4334
Case I OL	1.3332	0.8271
Case II CL	1.311	0.8933
Case II OL	2.0652	1.3319
Case III CL	1.6665	2.81126
Case III OL	2.9785	2.3216
Case IV CL	2.2981	4.4364
Case IV OL	10.1919	13.1388

Table 2: Mean and variance for Case I through Case IV

For Case V the same initial conditions were used to drive the relative motion of the target satellite with respect to the Neptec LCS that were used in Case IV, but the sensor was occluded to test the response of each tracking architecture to sensor occlusions. Figure 24, Figure 25 and Figure 26 show the response of the closed loop architecture to the occlusion, while Figure 27, Figure 28 and Figure 29 show the response of the open loop architecture to the occurrence of sensor occlusions. For the closed loop architecture four sensor occlusions of varying length between 10 to 30 seconds were used. Figure 24 and Figure 25 show the estimated target satellite position and orientation as a blue line with the LCS sensor measurements shown as green dots, portions where there are no green dots indicate the occurrence of a sensor occlusion. In Figure 29 the estimated angular velocity of the target satellite is plotted against the true angular velocity of the target satellite.

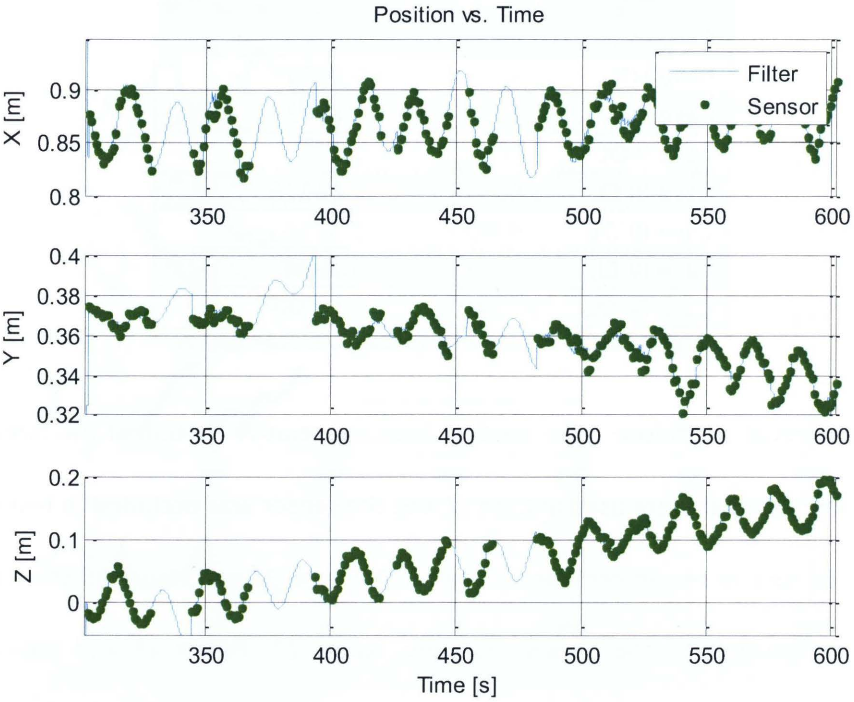

Figure 24: Case V Closed Loop Configuration, Target Position

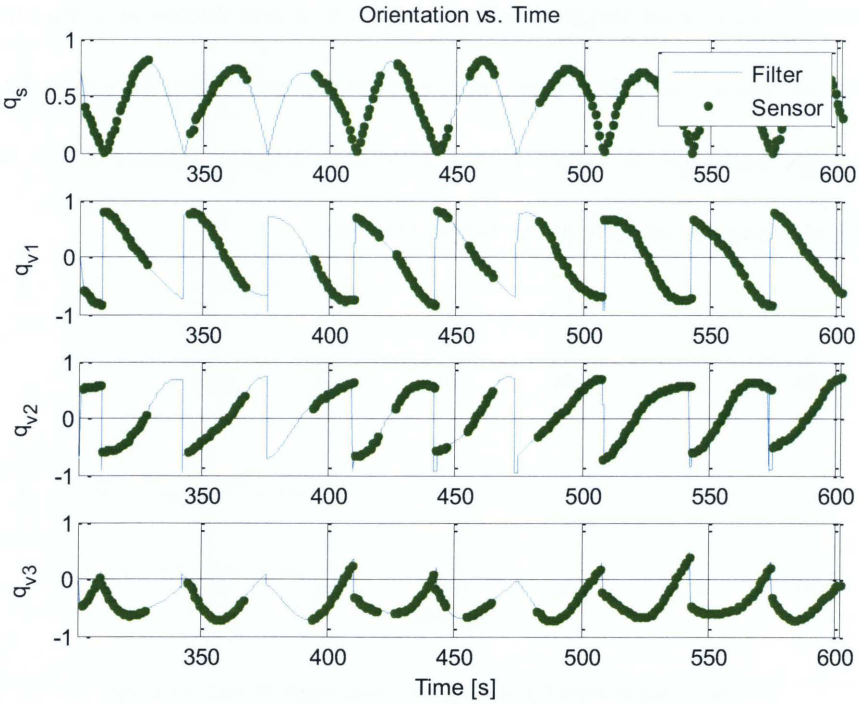

Figure 25: Case V Closed Loop Configuration, Target Orientation

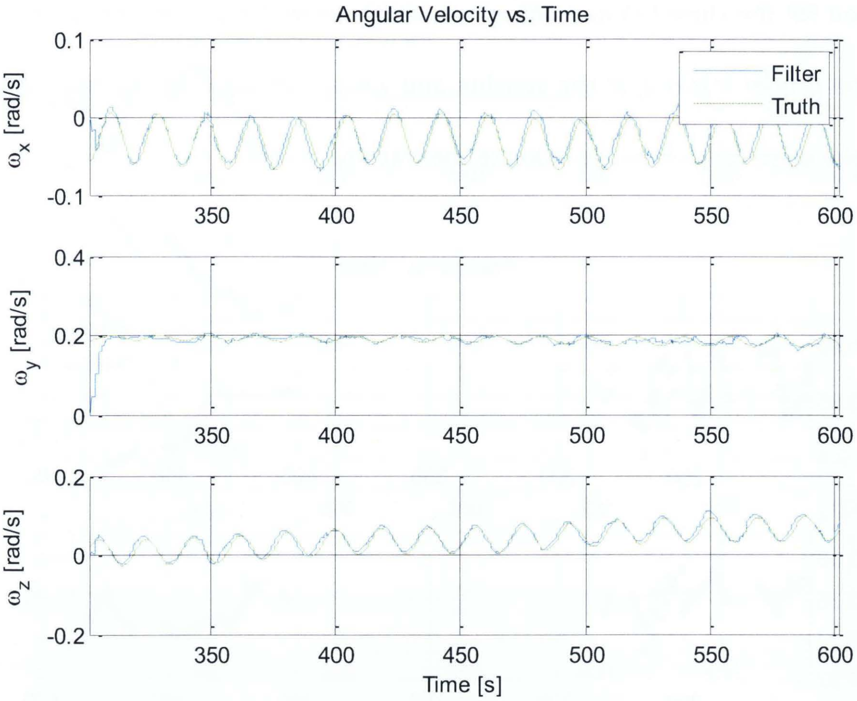

Figure 26: Case V Closed Loop Configuration, Target Angular Velocity

In Figure 24, Figure 25 and Figure 26 for the closed loop system, it can be seen that the target satellite is successfully tracked for the full 300 seconds of the experimentation, with the filter estimating the pose of the target satellite well enough to provide an accurate initial guess to continue tracking when the occlusion is removed. When the LCS is occluded and the sensor is unable to provide readings, the EKF uses the estimate of the angular velocity of the target satellite and a dynamic model to propagate its motion through time till the next sensor reading allowing for an accurate initial guess to be supplied to the ICP algorithm when the sensor occlusion is over.

Figure 27 and Figure 28 show the estimated relative position and orientation of the target satellite for the open loop system, with the LCS sensor output shown as green dots. Figure 29 shows the estimated target angular velocity plotted versus the true target angular velocity. For the open loop architecture, only one occlusion was used as when the occlusion was finished the tracking of the target satellite failed and data collection was ended. The tracking failure can be seen in the three figures when compared to the results obtained for the closed loop architecture which show the proper response. The closed loop response shows the proper tracking of the satellite and when compared to the open loop response the discontinuities in the open loop response indicate the tracking error.

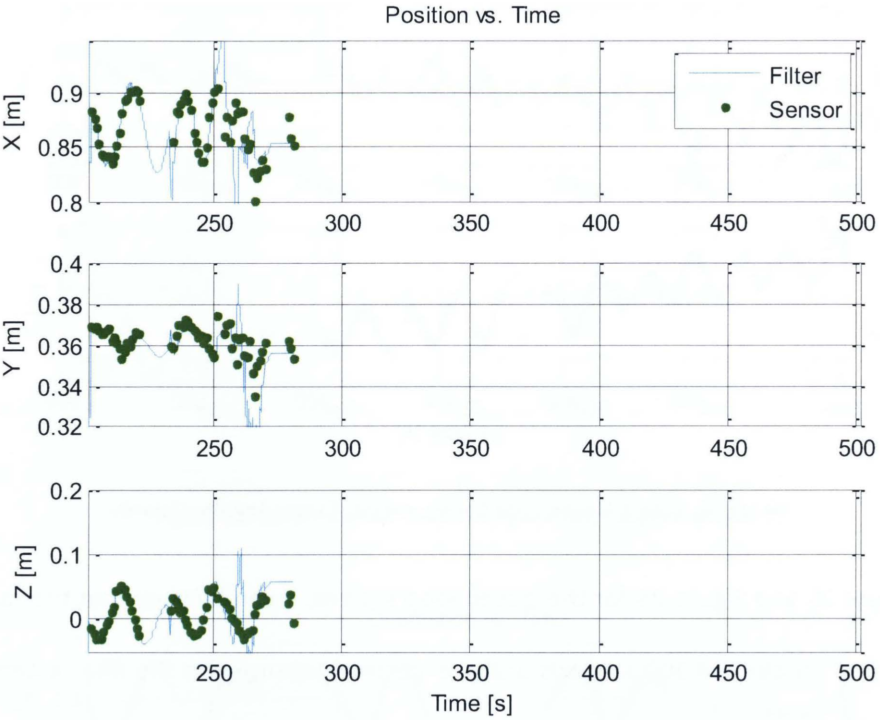

Figure 27: Case V Open Loop Configuration, Target Position

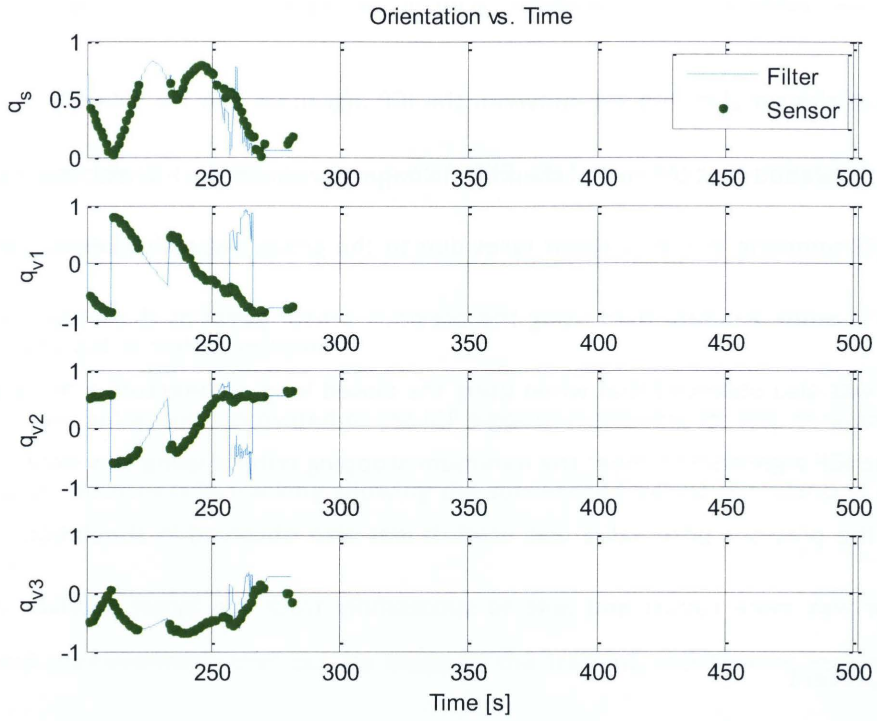

Figure 28: Case V Open Loop Configuration, Target Orientation

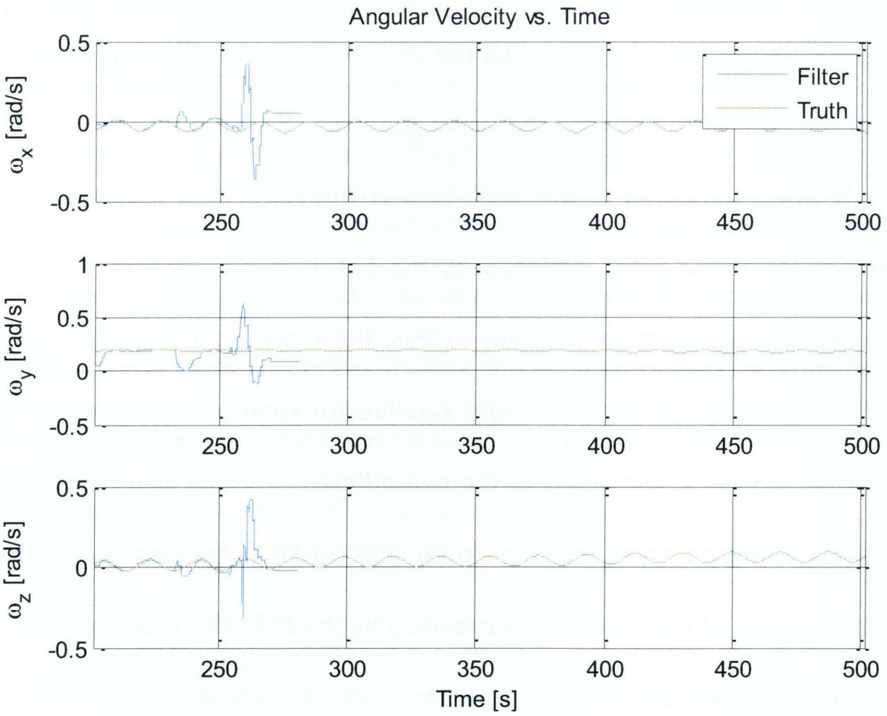

Figure 29: Case V Open Loop Configuration, Target Angular Rate

Conclusion

From the results of the simulation a link between the ICP algorithm and the EKF was demonstrated. It was observed in simulation that the use of the EKF in conjunction with the ICP algorithm allowed for the tracking of objects tumbling at high angular rates due to the predictive ability of the EKF providing an initial guess that is more accurate than using the previous sensor guess as the initial condition for the ICP algorithm. It was also observed that when using the closed loop architecture a more accurate initial guess allowed the ICP algorithm to meet the minimum stopping criteria using less iterations then in the situation where the previous pose value was used. It was also observed in simulation that the closed loop architecture was more robust and able to successfully track the target satellite in cases where occlusions were present.

In the laboratory phase of the experimentation, it was demonstrated in Case I through Case IV where the rotation rate of the target satellite was increased the difference between the initial guess and the output of the ICP algorithm increased when using the open loop architecture but when the closed loop algorithm was used the difference between the initial guess and the output of the ICP algorithm didn't increase significantly when compared to the open loop architecture. This shows that when using the closed loop algorithm the initial guess provided by the EKF is more accurate then when using the previous guess by the ICP algorithm. In addition to this, the more accurate initial guess of the pose provided by the EKF allows the tracking of target satellite for rotational rates where the open loop algorithm fails tracking. In Case V, the ability of the two different tracking algorithms was tested in the presence of occlusions. It was observed that the closed loop tracking algorithm was able to successfully deal with occlusions because of the initial guess provided by the EKF. This is because the initial guess is within the range of convergence and the ICP algorithm is able to provide a correct sensor reading. On the contrary, the open loop architecture is unable to maintain tracking because the target satellite's change in pose was large enough to cause the initial guess to fall outside the range of convergence. This causes the ICP algorithm to converge to a local minimum that provides an erroneous pose measurement.

In conclusion, it was found that the advantages of using the closed loop architecture over that of the open loop architecture included:

a) Tracking of a target at higher velocities

b) A more accurate initial guess provided to the ICP algorithm allowing for less iterations to be required

c) An increase in robustness in tracking allowing the successful tracking in situations where occlusions are present

Future work and improvements that can be made to the tracking architecture include the use of the covariance matrix from the ICP algorithm as the measurement error covariance matrix for the EKF. This would provide a dynamic measure of the measurement error for each sensor reading which will weight each sensor reading differently based on the output of the ICP algorithm. In addition to this if a constraint analysis of the target Object is performed to determine which features of the object are the best to scan to provide a good measurement, the EKF can be used to estimate which features will be in view of the camera at the time of the next measurement. If the features that are in view of the LCS are known then the feature that will provide the best measurement can be selected to be scanned. Improvements that can be made to the EKF include the addition of parameter estimation to the state vector, such as, the estimation of the inertial ratios of the system, or the center of mass location of the target satellite with respect to the point of reference.

Bibliography

Aghili, F., & Parsa, K. (2007). Adaptive Motion Estimation of a Tumbling Satellite Using Laser-Vision Data with Unknown Noise Characteristics. San Diego, CA: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems.

Aghili, F., & Parsa, K. (2009). Motion and Parameter Estimation of Space Objects Using Laser-Vision Data. *Journal of Guidance, Control and Dynamics , 32* (2), 538-550.

Besl, P. J., & McKay, N. D. (1992). A Method for Registration of 3-D Shapes. *IEEE Transactions on pattern analysis and machine intelligence* .

Blais, F., Beraldin, J. A., Cournoyer, L., Christie, I., Serafini, R., Mason, K., et al. (2000). Integration of a Tracking Laser Range Camera with the Photogrammetry based Space Vision System. *SPIE Proceedings AeroSense.* Orlando: NRC-CNRC.

Blais, F., Beraldin, J. A., Cournoyer, L., El-Hakim, S., Domey, J., Rioux, M., et al. (2001). Target Tracking, Object Pose Estimation, and Effects of the Sun on the NRC 3-D Laser Tracker. *Sixth International Symposium on Artificial Intelligence, Robotics and Automation in Space.* St-Hubert: NRC-CNRC.

Hillenbrand, U., & Lampariello, R. (2005). Motion and Parameter Estimation of a Free-Floating Space Object from Range Data for Motion Prediction. *8th International Symposium on Artificial Intelligence, Robotics and Automation in Space.* Munich.

Horn, B. K. (629). Closed-form solution of absolute orientation using unit quaternions. *Journal of the optical Society of America* , 1987.

Hughes, P. C. (1986). *Spacecraft Attitude Dynamics.* Jhon Wiley & Sons Inc.

Kalman, R. E. (1960). A New Approach to linear Filtering and Prediction Problems.

Kelsey, J. M., Byrne, J., Cosgrove, M., Seereeram, S., & Mehra, R. K. (2002). Vision-Based Relative Pose Estimation for Autonomous Rendezvous and Docking. *IEEE* .

Lavin, P., Noyer, J. C., & Benjelloun, M. (2005). An hardware architecture for 3D object tracking and motion estimation. *IEEE* .

Lyn, C., Mooney, G., Bush, D., Jasiobedzki, P., King, D., Krishnasamy, R., et al. (2007). Computer Vision Systems for Robotic Servicing of the Hubble Space Telescope. *AIAA SPACE 2007 Conference & Exposition.* Long Beach: AIAA.

McTavish, D. J., Schumacher, R., & Okouneva, G. (2007). Kalman filtering for dynamic pose and relative motion estimation in orbit. *53* (3).

Rouleau, G., Rekleitis, I., L'Archeveque, R., Martin, E., Parsa, K., & Dupuis, E. (2006). Autonomous Capture of a Tumbling Satellite. *IEEE International Conference on Robotics and Automation.* Orlando: IEEE.

Ruel, S., English, C., Anctil, M., & Church, P. (2005). LASSO: Real-time pose estimation from 3D data for autonomous satellite servicing. *ISAIRAS.* 2005: ESA.

Ruiter, H., & Benhabib, B. (2005). Tracking of Rigid-Bodies for autonomous Surveillance. *IEEE International Conference on Mechatronics & Automation.* Niagrara Falls: IEEE.

Samson, C., English, C., Deslauriers, A., Christie, I., Blais, F., & Ferrie, F. (2004). Neptec 3D Laser Camera System: From Space Mission STS-105. *to Terrestrial Applications* , NRC-CNRC.

Simon, D. A., Hebert, M., & Kanade, T. (1994). *Real-time 3-D Pose Estimation Using a High-Speed Range Sensor.* Pittsburgh: The Robotics Institute Carnegie Mellon University.

Timmons, K. K., & Ringelberg, J. C. (2007). Approach and Capture for Autonomous Rendezvous and Docking. *IEEE* .

Appendix

Results

This portion of the appendix contains graphs that aren't included in the results portion of the thesis but demonstrate the relative performance of the two tracking architectures. The graphs that appear in the appendix consist of three types:

1. The first type of graph shows the position estimated by the EKF plotted as a blue line with the sensor readings plotted as green dots. This type of graph demonstrates the accuracy of the EKF vs. the sensor readings for position.
2. The second type of graph shows the orientation estimated by the EKF as an Euler Parameter Quaternion plotted as a blue line with the sensor readings plotted as green dots. This type of graph demonstrates the accuracy of the EKF vs. the sensor readings for orientation.
3. The third type of graph shows the angular velocity of the estimated by the EKF of the target satellite as a blue line and the true angular velocity of the target satellite as a green line vs. time. This type of graph demonstrates the EKF ability to estimate the angular velocity of the target satellite with no direct measurements of the angular velocity.

Graphs for Case I through Case IV are provided for the open loop and closed loop configurations and any graphs that are missing are presented within the Experimental Results portion of the thesis.

Case I: Closed Loop

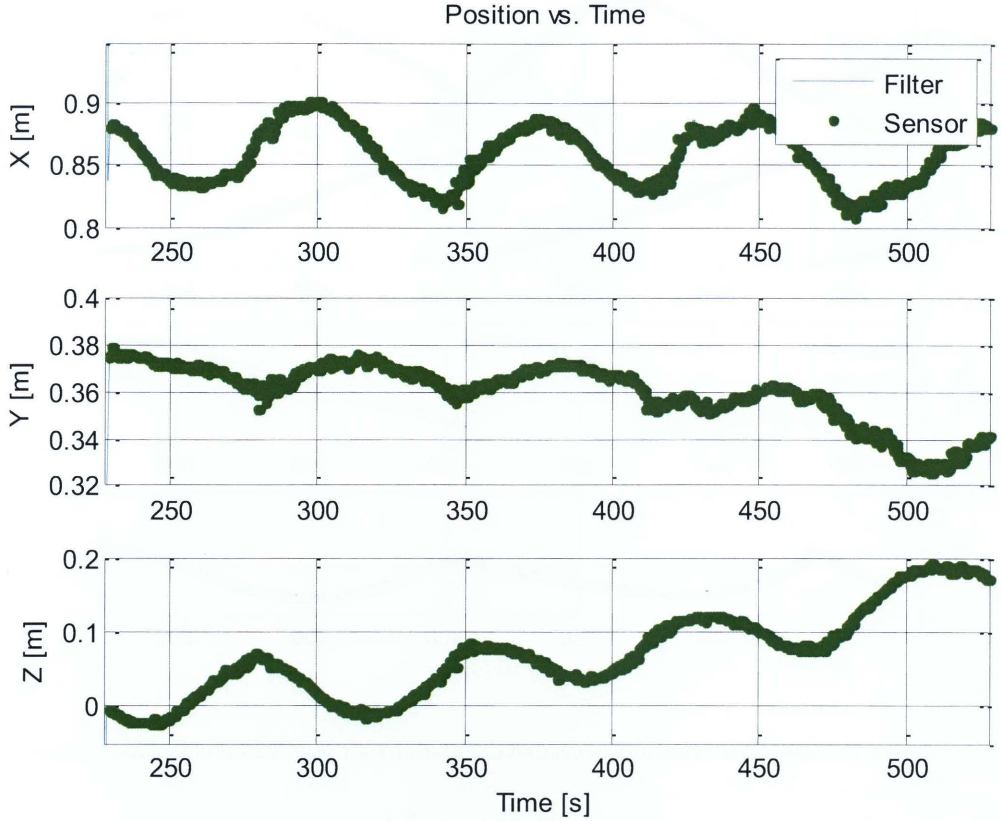

Figure 30: Case I: Estimated and sensor measurement for closed loop tracking architecture

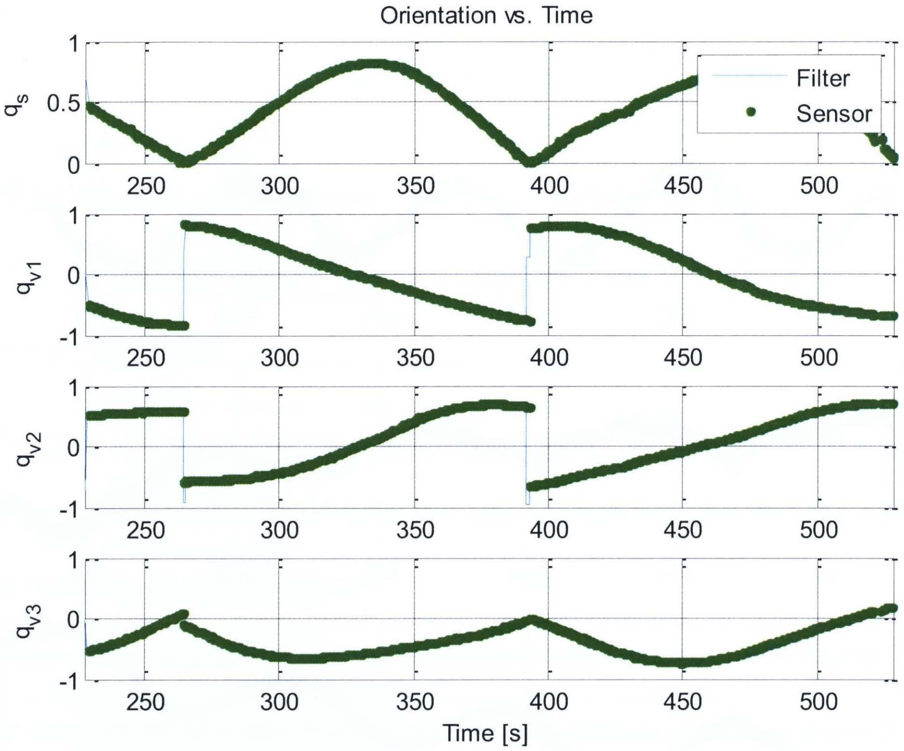

Figure 31: Case I: Estimated and measured orientation for closed loop architecture

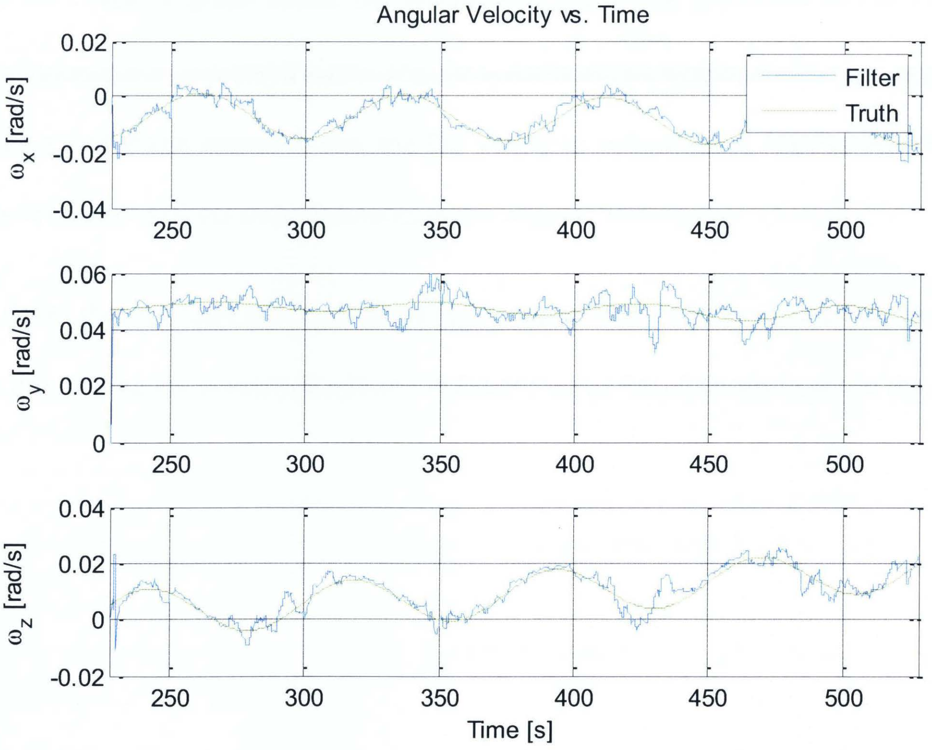

Figure 32: Case I: Estimated and true angular velocity of target satellite for closed loop architecture

Case I: Open Loop

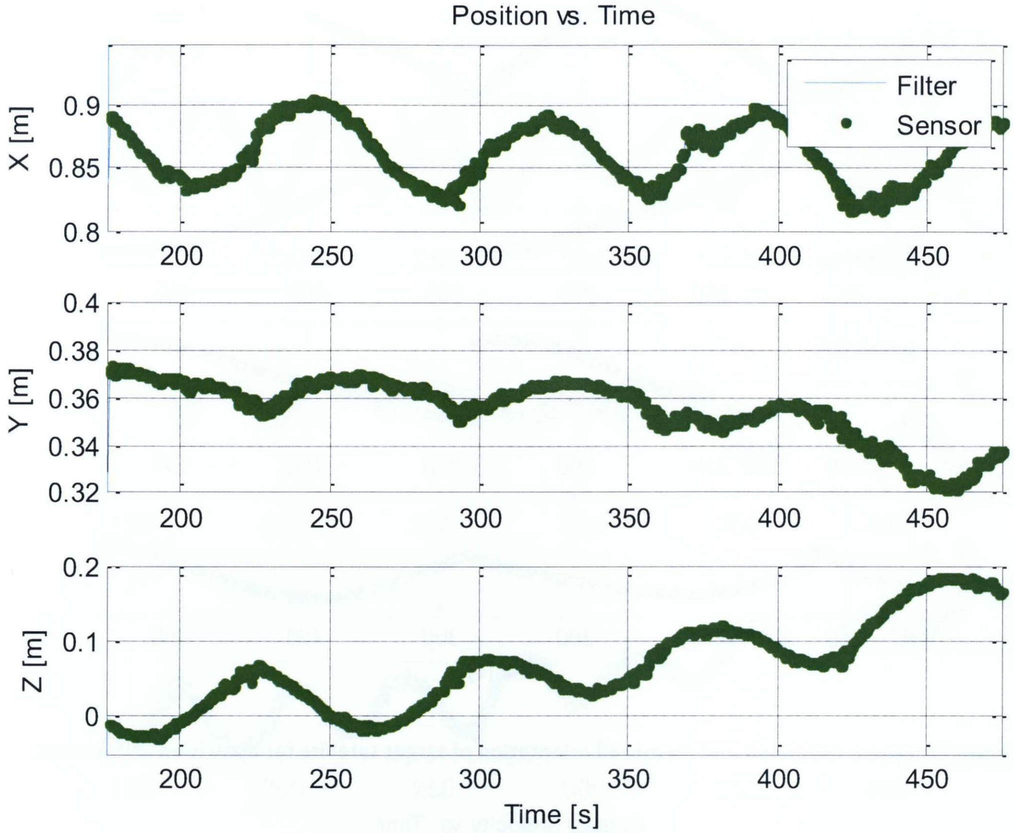

Figure 33: Case I: Estimated and measured position of the target satellite for the open loop architecture

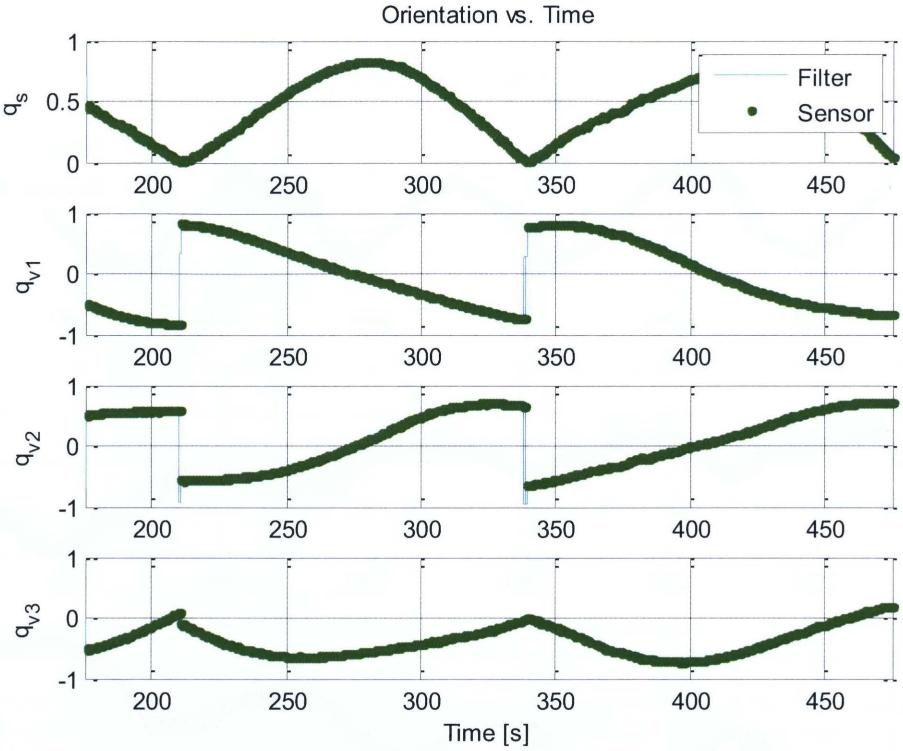

Figure 34: Case I: Estimated and measured orientation of target satellite for open loop architecture

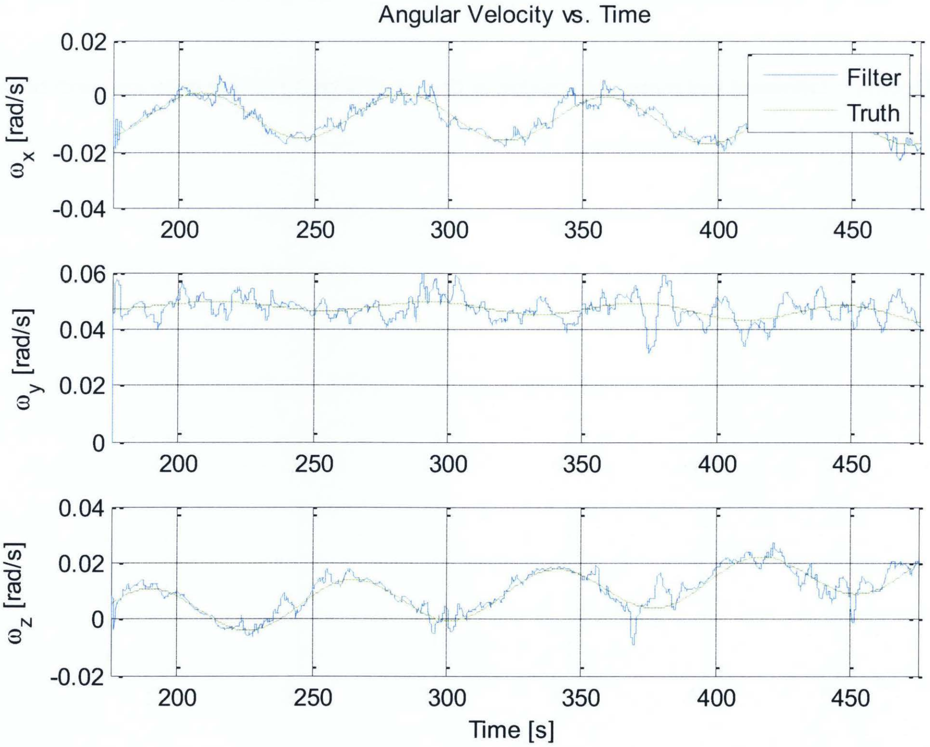

Figure 35: Case I: Estimated and true angular velocity of target for open loop architecture

Case II: Closed Loop

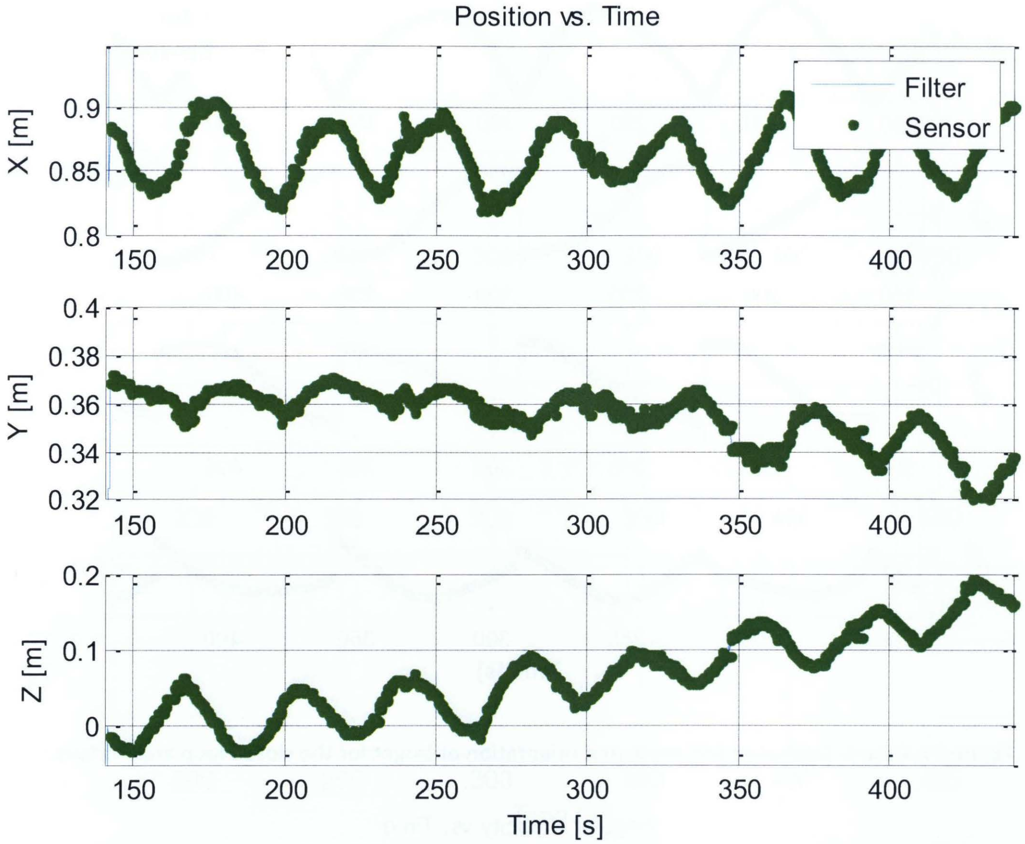

Figure 36: Case II: Estimated and measured position of target for closed loop architecture

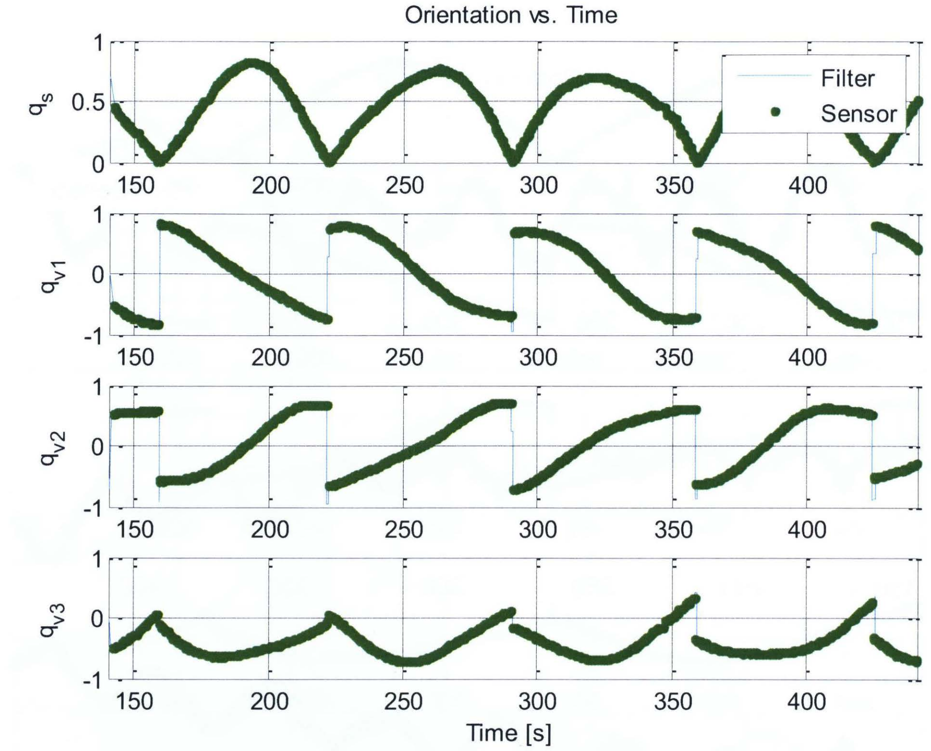

Figure 37: Case II: Estimated and measured orientation of target for the closed loop architecture

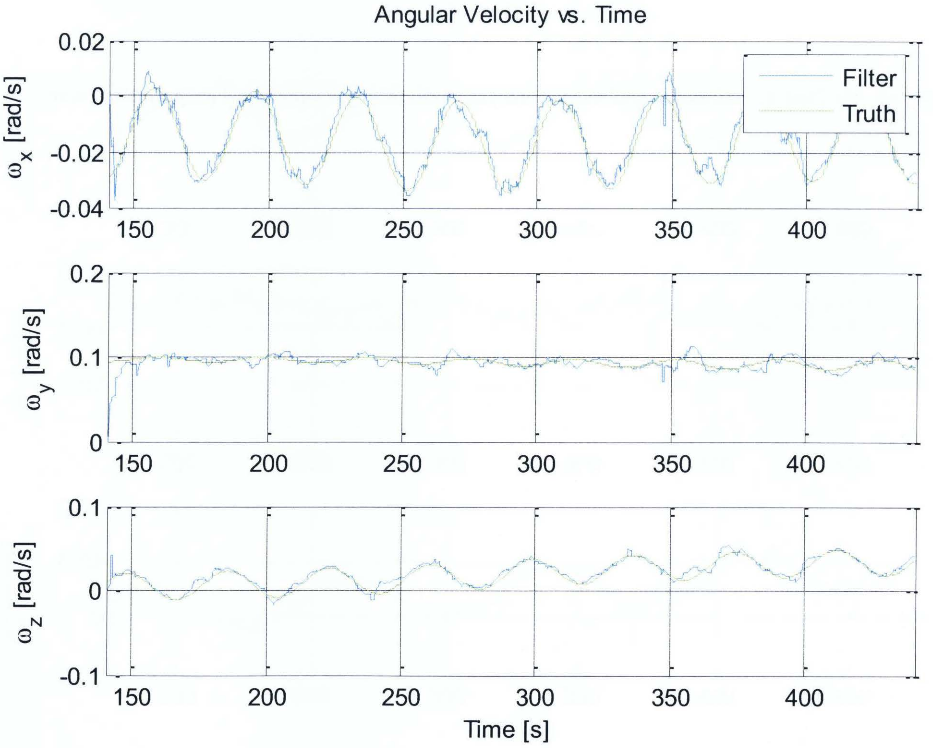

Figure 38: Case II: Estimated and true angular velocity of the target for closed loop architecture

Case II: Open Loop

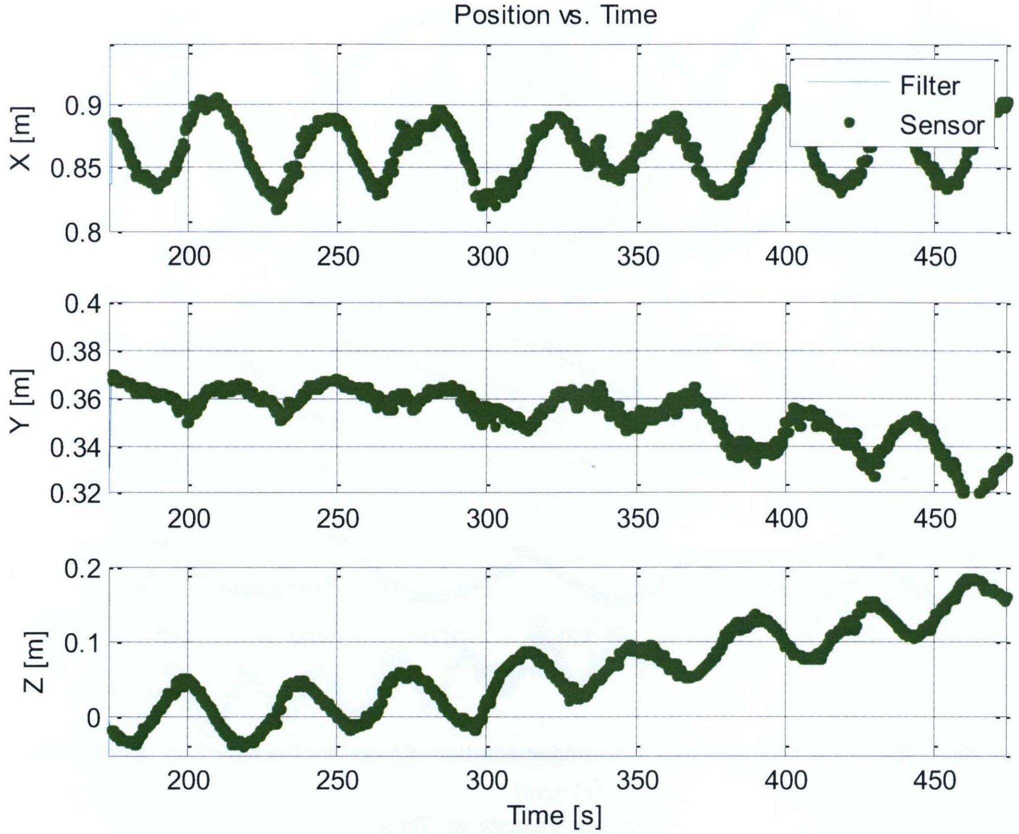

Figure 39: Case II: Estimated and measured position of target for the open loop architecture

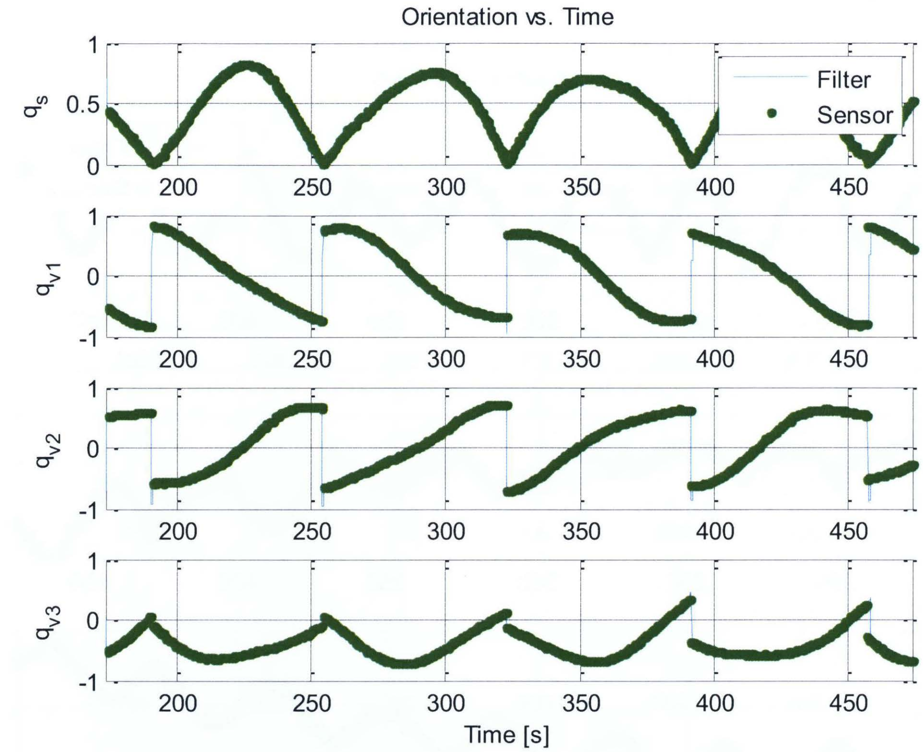

Figure 40: Case II: Estimated and measured orientation of target for the open loop system

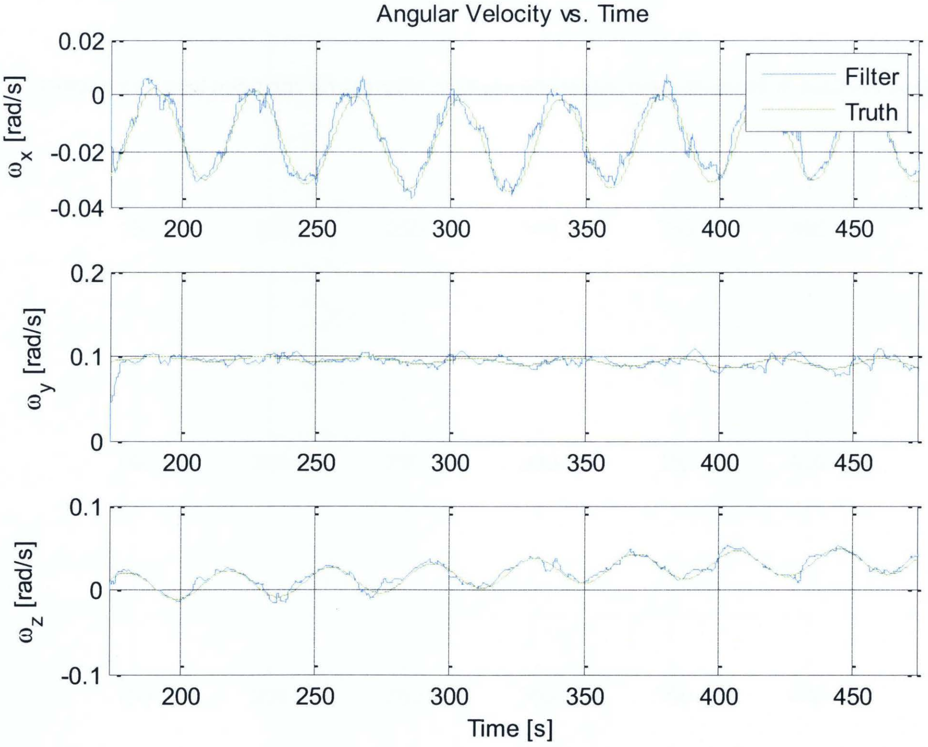

Figure 41: Case II: Estimated and true angular velocity of target for the open loop architecture

Case III: Closed Loop

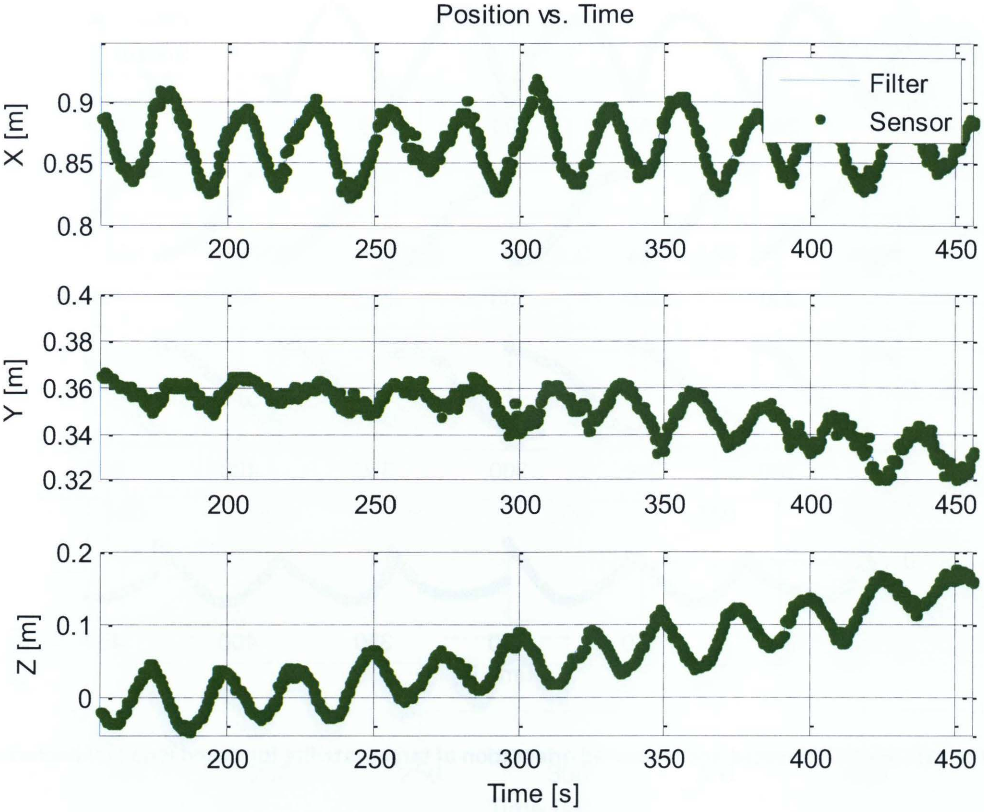

Figure 42: Case III: Estimated and measured position of target for the closed loop architecture

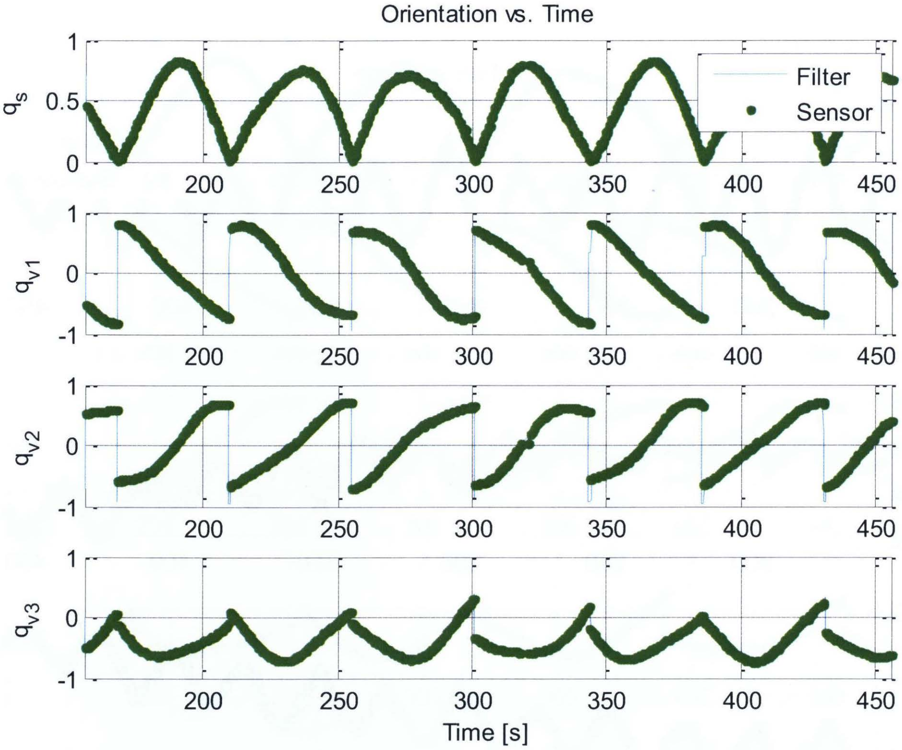

Figure 43: Case III: Estimated and measured orientation of target satellite for closed loop architecture

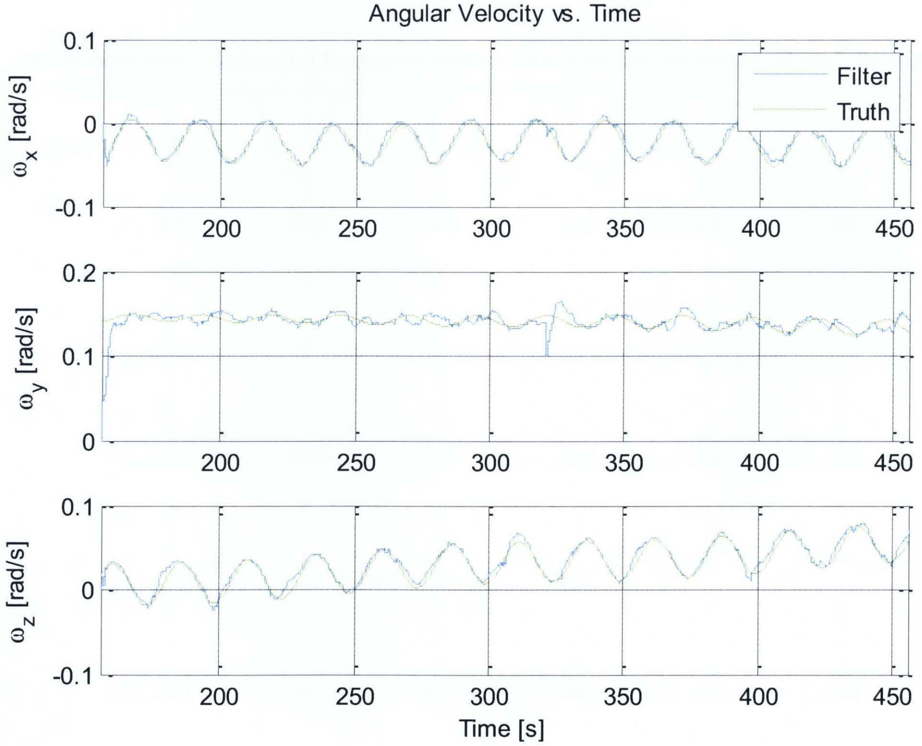

Figure 44: Case III: Estimated and true angular velocity of the target for the closed loop architecture

Case III: Open Loop Configuration

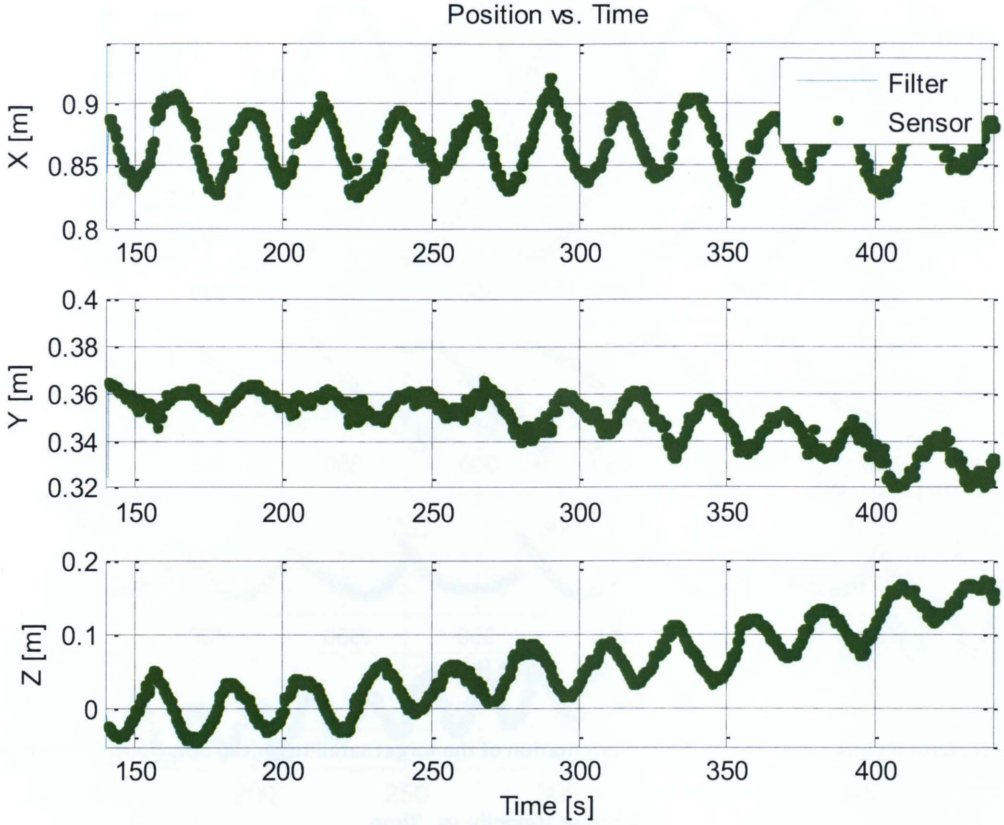

Figure 45: Case III: Estimated and measured position of the target for the open loop architecture

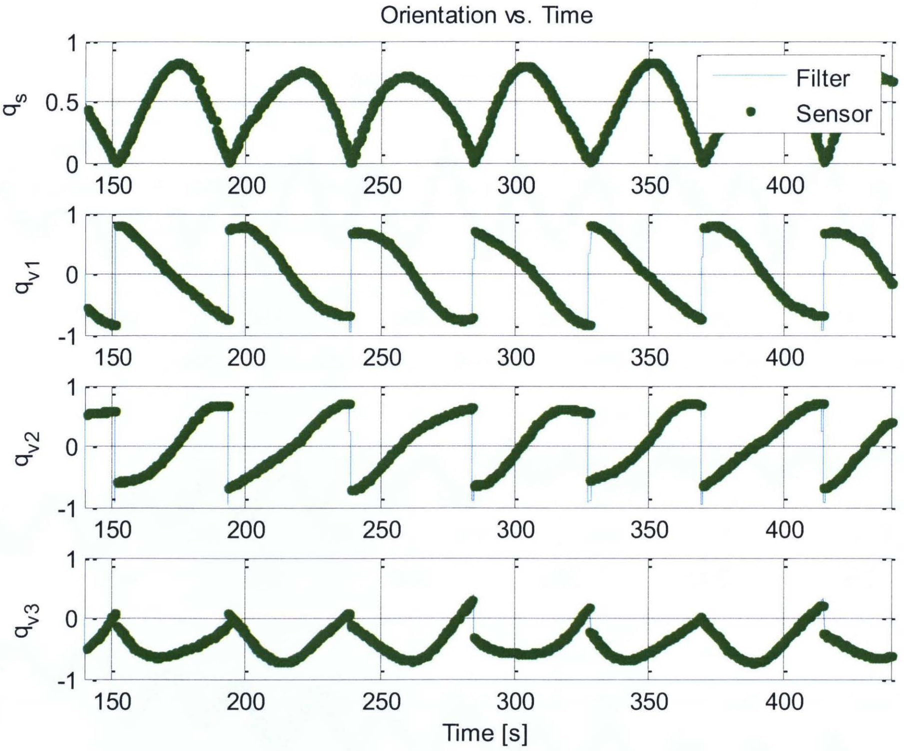

Figure 46: Case III: Estimated and measured orientation of the target satellite for the open loop architecture

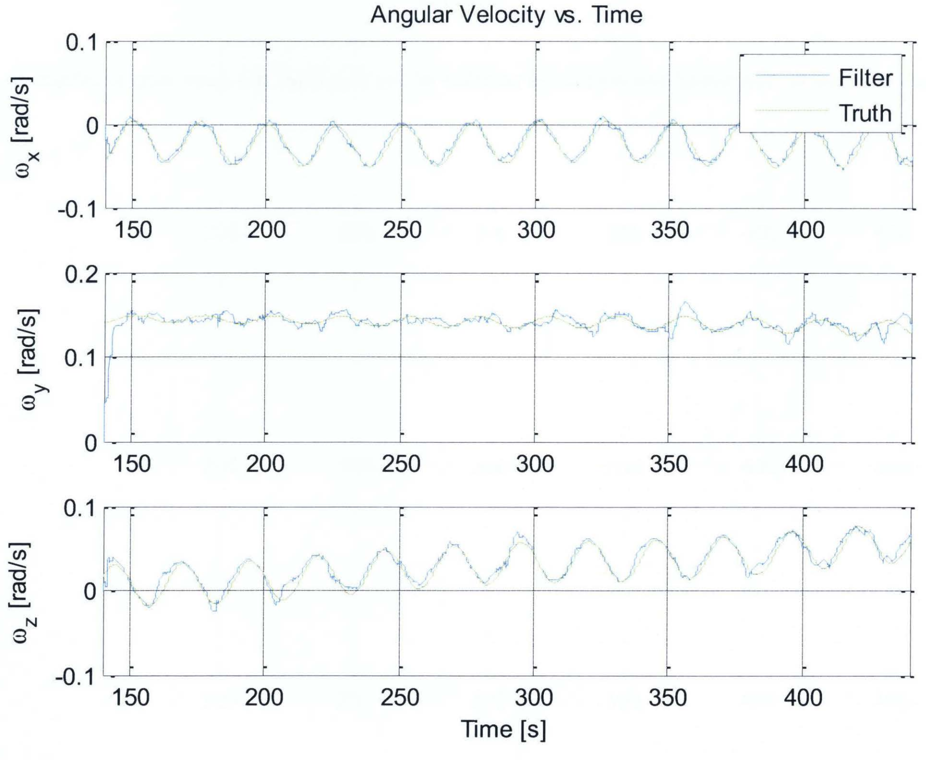

Figure 47: Estimated and true angular velocity of the target for the open loop architecture

Case IV: Closed Loop

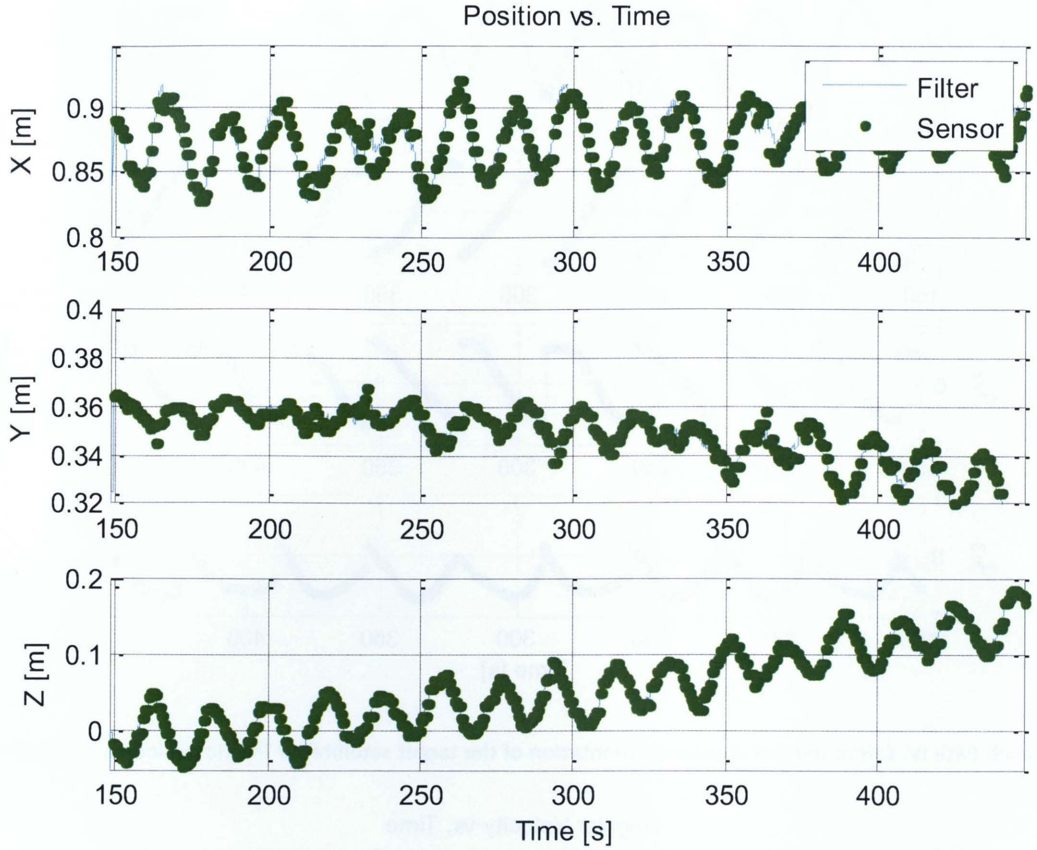

Figure 48: Case IV: Estimated and measured position of the target for the closed loop system

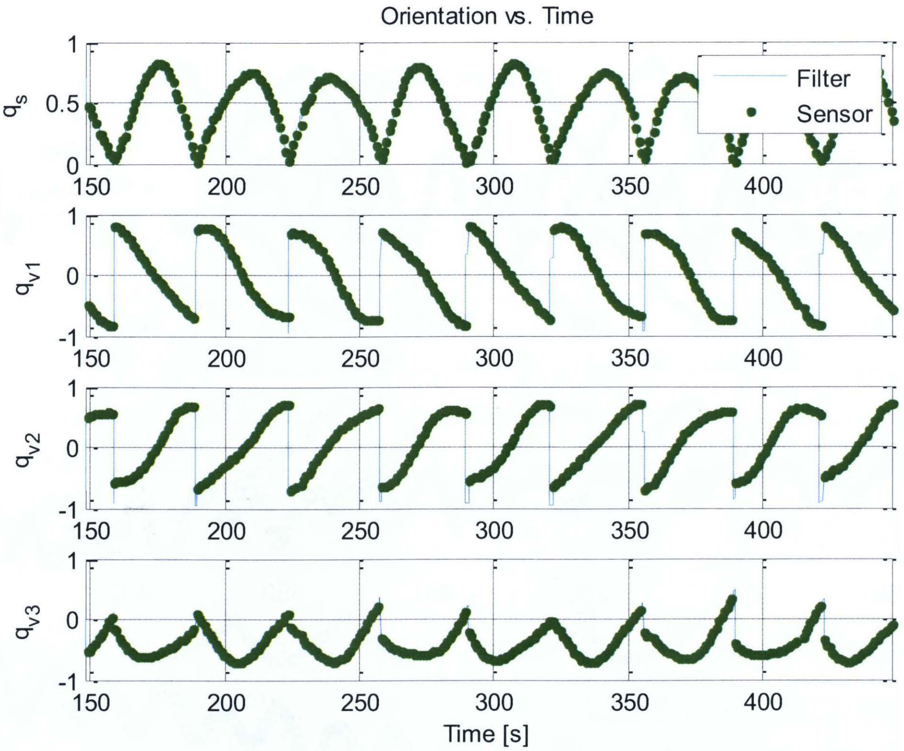

Figure 49: Case IV: Estimated and measured orientation of the target satellite for the closed loop architecture

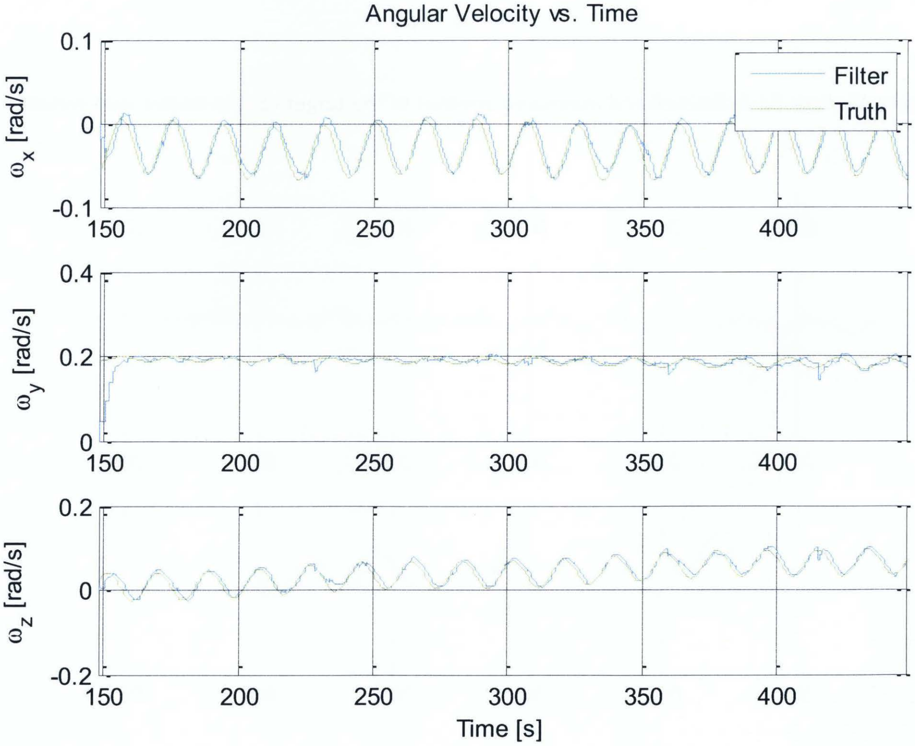

Figure 50: Estimated and true angular velocity of the target for the closed loop architecture

B) BC63-34