

Erasmus Mundus SpaceMaster

Luleå University of Technology

Czech Technical University

Master Thesis

Study and Design of an Adaptive Control Law for Spacecraft Attitude Control

Author:

Lucas Almeida Cypriano

Supervisors:

Martin Hromcik

Dennis S. Bernstein

A thesis submitted in fulfilment of the requirements

for the degree of Master in Science

May 30

Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Control Engineering

DIPLOMA THESIS ASSIGNMENT

Student: **Almeida Cypriano Lucas**

Study programme: Cybernetics and Robotics
Specialisation: Systems and Control

Title of Diploma Thesis: **Study and Design of an Adaptive Control Law for Spacecraft Attitude Control**

Guidelines:

The thesis will consist of the study of adaptive control algorithms, the design of one such control law for spacecraft attitude control and simulation of the control law.

1. Work on the application of RCAC to the triaxial testbed in the Attitude Dynamics and Control Laboratory.
2. Implement and test RCAC to perform rigid body maneuvers. This will require simulation of RCAC and implementation of RCAC on the Quanser system in the lab.

Bibliography/Sources:

- [1] Sergio Esteban Roncero, "A Comparison of Nonlinear PI and PID Inertia-free Spacecraft Attitude Control Laws", BarSym 2012, Springer Verlag Edited Bound Volume.
- [2] A.E.Bryson Jr., Control of aircraft and spacecraft, Princeton, 2002.

Diploma Thesis Supervisor: doc.Ing. Martin Hromčík, Ph.D.

Valid until the summer semester 2014/2015


prof. Ing. Michael Šebek, DrSc.
Head of Department




prof. Ing. Pavel Ripka, CSc.
Dean

Prague, February 18, 2014

ERASMUS MUNDUS SPACEMASTER

Luleå University of Technology

Czech Technical University

Abstract

Martin Hromcik

Dennis Bernstein

Department of Control Engineering

Master in Science

Study and Design of an Adaptive Control Law for Spacecraft Attitude Control

by Lucas Almeida Cypriano

In this thesis the problem of spacecraft attitude control is investigated. The spacecraft dynamics are modeled as a rigid body system controlled by 4 thrusters, and the kinematics are represented by direction cosine matrix. In this thesis, the effectiveness of retrospective cost adaptive control for spacecraft attitude control is evaluated. The prominent feature of this adaptive controller is that it requires little knowledge of the system to be controlled. This makes this controller attractive since for most spacecraft knowledge of the spacecraft's inertia tensor is limited and will sometimes change over the spacecraft lifetime.

Acknowledgements

First I would like to thank my family and friends who encouraged me to follow my dream and embark in this journey. I would like to thank Prof. Kolmanovsky who accepted my request to research my thesis at University of Michigan, and Prof. Bernstein who accepted me as one of his students and offered me the Triax laboratory to perform my research. Finally I would like to thank my fellow students Frantisek Sobolic, Gerardo Cruz, Antai Xie and Liang Han who much helped me in this project.

Contents

List of Figures	5
List of Tables.....	7
Abbreviations	8
Chapter 1 - Introduction.....	9
1.1 – Motivation	9
1.2 – Spacecraft Attitude Determination	9
1.2.1 - Attitude Determination.....	10
1.3 – Attitude Control.....	15
1.3.1 – Basic Control Laws.....	16
1.3.2 – Direct Cosine Error Matrix Control Law	16
Chapter 2 – The RCAC	18
2.1 – Introduction.....	18
2.2 – The RCAC Algorithm	18
2.2.1 – Markov Parameters.....	18
2.2.2 – Retrospective Performance.....	19
2.2.3 – Retrospective Cost Function	20
2.3 – RCAC Design for Attitude Control	21
2.3.1 – Attitude Error Representation.....	21
2.3.2 – Markov Parameters.....	22
Chapter 3 – Simulation.....	25
3.1 – Preliminary Systems	25
3.1.1 – Aircraft Pitch.....	25
3.1.2 – DC Motor Position	27
3.1.3 – Inverted Pendulum.....	30
3.1.4 – Single Axis Spacecraft.....	33
3.2 – Triax System	35
3.2.1 – Torque Simulation	37
3.2.2 – Thruster Simulation	40
Chapter 4 - The Triax.....	44
4.1 – Description	44
4.2 – Mathematical Modeling.....	46
4.2.1 – Rigid Body Dynamics and Kinematics.....	46
4.2.2 – Inertia Tensor Estimate	48
4.2.3 – Inertia Tensor Calculation	48

4.3 – Simulink Model.....	50
Chapter 5 – Hardware Testing	52
Chapter 6 – Conclusion & Future Works.....	56
References.....	57

List of Figures

Figure 1 - Earth Sensor Diagram.....	10
Figure 2 - Analog Sun Sensor	10
Figure 3 - Star Sensor	11
Figure 4 - Mechanical Gyroscope	11
Figure 5 - MEMS Gyroscope [4]	11
Figure 6 - Basic problem of attitude representation	12
Figure 7 - Euler Angles Representation	14
Figure 8 - Simulink scheme for Markov Parameter extraction	23
Figure 9 - Simulink Scheme for Markov Parameter extraction	23
Figure 10 - Aircraft Pitch System	25
Figure 11 - Aircraft/DC Motor/Inverted Pendulum/Single axis spacecraft Simulation Scheme	26
Figure 12 - Aircraft Pitch state (system output) & reference 10 [deg].....	26
Figure 13 - Aircraft Pitch state (system output) & reference 180 [deg].....	27
Figure 14 - DC Motor Characteristics	28
Figure 15 - DC Motor Position (system output) & Reference 1 [rad].....	29
Figure 16 - DC Motor Position (system output)& Reference 1.5 [rad].....	29
Figure 17 - Inverted Pendulum System.....	30
Figure 18 - Pendulum Simulink Schematic	32
Figure 19 - Inverted Pendulum Position (system output)& Reference 5 [deg] initial condition.	32
Figure 20 - Inverted Pendulum Position (system output)& Reference 10 [deg] initial condition	33
Figure 21 - Single Axis spacecraft attitude (system output) no thruster saturation	34
Figure 22 - Single Axis Spacecraft attitude (system output) with thruster saturation on +5	35
Figure 23 - Single Axis Spacecraft attitude (system output) with thruster saturation on +2.5 ..	35
Figure 24 - Triax Simulation schematics	36
Figure 25 - Torque Simulations Triax Model	36
Figure 26 - Thruster Simulations Triax Model.....	36
Figure 27 – Eigen Angle & Control Parameters Pitch -175 [deg]	37
Figure 28 – Eigen Angle & Control Parameters Yaw 15 [deg]	37
Figure 29 – Eigen Angle & Control Parameters Roll -120 [deg].....	37
Figure 30 - Eigen Angle & Control Parameters Yaw 15-70 [deg].....	38
Figure 31 - Eigen Angle/Axis & Control Parameters Pitch 15 [deg] - Yaw 30 [deg].....	38
Figure 32 - Overestimated Markov parameter 100H	39
Figure 33 - Underestimated Markov parameter $10^{(-5)}H$	40
Figure 34 – Eigen Angle & Control Parameters Pitch 175 [deg].....	40
Figure 35 – Eigen Angle & Control Parameters Roll -120 [deg].....	41
Figure 36 – Eigen Angle & Control Parameters Yaw 45 [deg]	41
Figure 37 - Eigen Angle/Axis & Control Parameters Pitch 30 [deg] - Roll 45 [deg]	42
Figure 38 - Eigen Angle/Axis & Control Parameters Roll 10 [deg] - Yaw 90 [deg].....	42
Figure 39 – Triax.....	44
Figure 40 - Triax Batteries (left) and Actuators (right).....	45
Figure 41 - Optitrack Cameras	45
Figure 42 - Optitrack Camera Setup and Markers Trackable (Triax)	45
Figure 43 - Forces due to the thrusters on the Triax	47
Figure 44 - Solid Cylinder Inertia	48

Figure 45 - Thruster Curves, Linearized and Nonlinear.....	49
Figure 46 - Triax Model.....	50
Figure 47 - Spacecraft Model.....	50
Figure 48 – Thruster to Torque Model.....	51
Figure 49 - Main PID scheme	52
Figure 50 - PID Controller scheme	52
Figure 51 - Eigen Angle & Axis Pitch 10 [deg]	53
Figure 52 - Eigen Angle & Axis Roll -45 [deg]	53
Figure 53 - Eigen Angle & Axis Yaw 140 [deg]	54
Figure 54 - Eigen Angle & Axis [1,1,1] 20 [deg]	54

List of Tables

Table 1 - Potential Accuracies of Attitude Sensors [4]	12
Table 2 - Aircraft Pitch RCAC tuning parameters	27
Table 3 - DC Motor RCAC tuning parameters	30
Table 4 - Pendulum RCAC tuning parameters.....	33
Table 5 - Single Axis Spacecraft RCAC tuning parameters	34
Table 6 - Torque 3 Axis Stabilized Spacecraft RCAC tuning parameters	39
Table 7 - Thruster 3 Axis Stabilized Spacecraft RCAC tuning parameters	43
Table 8 - PID Control Law Gains	52

Abbreviations

RCAC – Retrospective Cost Adaptive Control

ADCS – Attitude Determination and Control System

LEO – Low Earth Orbit

DCM – Direct Cosine Matrix

MIMO – Multiple Input Multiple Output

MEMS – Micro Electrical Mechanical Systems

Chapter 1 - Introduction

1.1 – Motivation

Modern spacecraft have an attitude determination and control system (ADCS), one of the most challenging problems for control engineers is the modeling of the spacecraft. A precise estimate or measurement of the inertia tensor of the spacecraft is not possible or even not useful as the spacecraft might deploy antennas, solar panels, etc. While a measurement or estimate of the inertia tensor of the spacecraft in its launch configuration can be obtained, such knowledge is still lacking as the values obtained do not correspond to the spacecraft configuration in its main life cycle.

Research is been done in developing controllers that do not need an accurate model of the spacecraft inertia tensor or may even not need any modeling at all. The controller studied in this thesis is one such research topic, needing minimal knowledge of the spacecraft's inertia tensor. In this thesis, the only information necessary from the spacecraft model is the first non-zero Markov parameter. Furthermore even that can be forgone and used as a tuning parameter.

1.2 – Spacecraft Attitude Determination

The motion of a spacecraft is determined by its position, velocity, attitude and angular velocity. The first two, position and velocity, describe the translational motion of the center of mass of the spacecraft and are not studied in this thesis. The last two, attitude and angular velocity, describe the rotation of the spacecraft's body about its center of mass, relative to an inertial frame, and are of interest to the study in this thesis. Attitude of a spacecraft is its orientation in space.

Unlike orbit problems most of the advances in attitude determination and control are recent, having happened mostly since the launch of Sputnik on 1954. This can be exemplified by the fact that the prediction of the orbital motion of celestial bodies was the initial motivation for much of Newton's work. So while much has been researched and discovered in orbit analysis, much of the body of knowledge is centuries old unlike attitude analysis. [2]

The problem of a spacecraft's attitude can be divided in three parts: determination, prediction and control. Of these three, prediction will not be mentioned in this work, determination will be briefly mentioned about as it is not the object of study of this thesis, and control is the object of study of this thesis.

Attitude determination is the process of computing the orientation of the spacecraft in relation to another object such as the Earth, the Moon or the Sun for example.

Attitude prediction is the process of forecasting the future orientation of the spacecraft by using dynamical models to extrapolate the attitude history.

Attitude control is the process of orientating the spacecraft in a specific direction or spinning the spacecraft at a certain angular velocity around a specific axis.

1.2.1 - Attitude Determination

The attitude determination of a spacecraft is inherently dependent on its mission profile. A spacecraft that will be launched into Low Earth Orbit (LEO) does not have the same Attitude Determination system that an exploratory mission to Mars has. The sensors used, their number, the algorithms and data processing hardware are all different.

The accuracy of the determined attitude depends on the sensors, on the attitude determination algorithms used and on the hardware capability available. In order to fully determine a spacecraft's attitude it is necessary to have at least 2 linearly independent vectors [2]. Using just one vector it is not enough as it is not possible to know whether the spacecraft is spinning around said reference vector or not. This means that a single reference vector does not contain information to determine the spacecraft's rotation around this reference vector. Therefore another reference vector is necessary to unambiguously fix the spacecraft's attitude.

1.2.1.1 – Attitude Sensors

There exist many types of sensors to determine spacecraft attitude. Each of these sensors determine either angle or angular velocity with respect to a reference frame.

A magnetometer measures the angle between the spacecraft's vector orientation with respect to Earth's magnetic field vector.

A Sun sensor measures the angle between the spacecraft's vector orientation with respect to the vector the points from the spacecraft to the Sun, similarly to Earth sensors which measures the angle between the spacecraft's vector orientation with respect to the vector the points from the spacecraft to the Earth. Simplified schematics of an Earth sensor and a Sun sensor can be seen in Figures 1 and 2 respectively. [4]

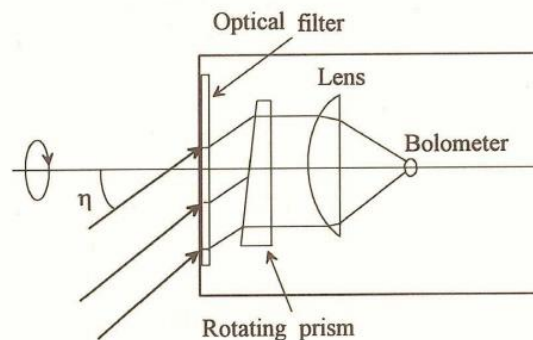


Figure 1 - Earth Sensor Diagram

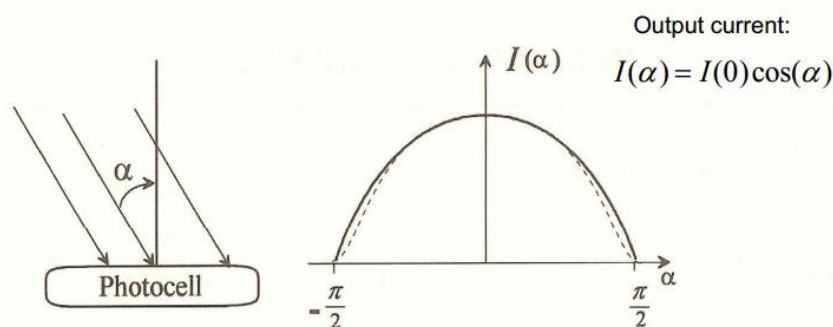


Figure 2 - Analog Sun Sensor

A star sensor measures angle with respect to the inertial frame of the universe. A simplified schematic of a star sensor can be seen in Figure 3. A CCD or other similar device captures the light coming from the stars in its field of view. This image is then amplified and compared with preloaded star charts in the memory of the sensor, when a match is found the sensor knows in which direction it is pointing.

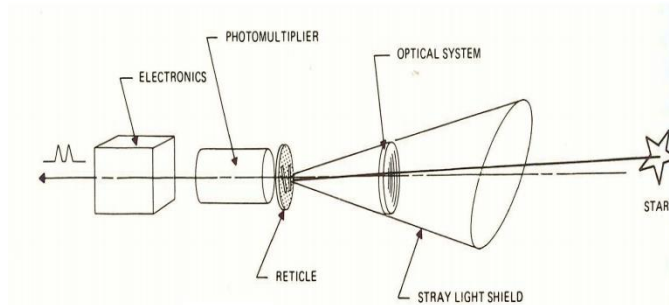


Figure 3 - Star Sensor

Gyroscopes, measuring angular velocity with respect to the inertial frame of the universe. Two examples of gyroscopes can be seen in Figures 4, a mechanical gyroscope, and 5, a MEMS gyroscope. [4]

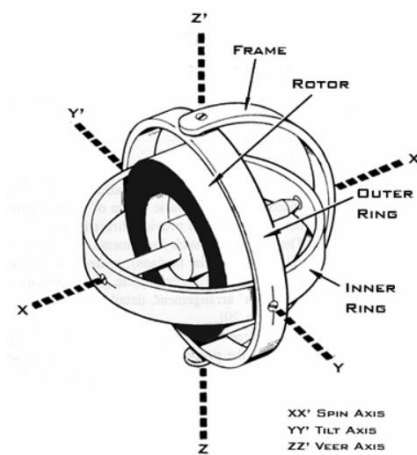


Figure 4 - Mechanical Gyroscope

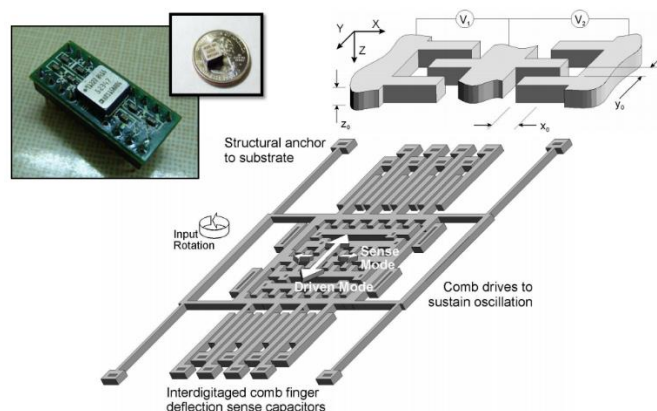


Figure 5 - MEMS Gyroscope [4]

Each of these sensors have different precision with which it is capable of measuring attitude. Table 1 shows the potential accuracies of the different types of sensors.

Reference Object	Potential Accuracy
Stars	1 arc second
Sun	1 arc minute
Earth (horizon)	6 arc minutes
RF beacon	1 arc minute
Magnetometer	30 arc minutes
Narstar Global Positioning System (GPS)	6 arc minutes

Table 1 - Potential Accuracies of Attitude Sensors [4]

1.2.1.2 – Attitude Representations

Consider a rigid body with an orthogonal coordinate system. Let the unit vectors \vec{u} , \vec{v} , \vec{w} be the triad of unit vector attached to the body that compose the body orthogonal coordinate system. The problem is then to specify the orientation of this vector triad with respect to a reference frame, as can be seen in Figure 6.

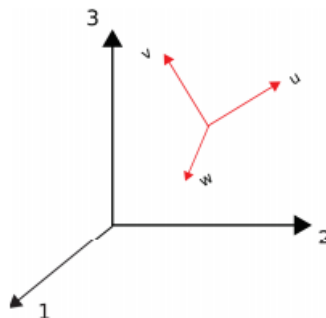


Figure 6 - Basic problem of attitude representation

The reference frame is composed of the vector triad $\vec{1}$, $\vec{2}$, $\vec{3}$. Thus in order to specify the vectors \vec{u} , \vec{v} , \vec{w} in this new reference frame 9 parameters are required. Each parameter is a component of one of the vectors \vec{u} , \vec{v} , \vec{w} resolved in the reference frame $\vec{1}$, $\vec{2}$, $\vec{3}$. These parameters can be represented in matrix form, where each of the matrix's elements is the cosine of the angle between a body unit vector and a reference unit vector. This way of representing attitude is known as Direction Cosine Matrix (DCM) or Attitude Matrix. [5]

$$A = \begin{bmatrix} u_1 & v_1 & w_1 \\ u_2 & v_2 & w_2 \\ u_3 & v_3 & w_3 \end{bmatrix} \quad (1.1)$$

Where $[u_1 \ u_2 \ u_3]^T$ is the vector \vec{u} expressed in the reference frame, and u_1, u_2, u_3 are the cosine of the angle between the vector \vec{u} and vectors $\vec{1}, \vec{2}, \vec{3}$ respectively.

The attitude matrix can be used to transform any vector $\vec{a} = [a_1 \ a_2 \ a_3]^T$ defined in the reference frame to the body frame by

$$\vec{a}_b = A\vec{a} \quad (1.2)$$

An important property of the attitude matrix that results from its orthogonality is

$$AA^T = I \rightarrow A^T = A^{-1} \quad (1.3)$$

where I is the identity matrix.

Thus any vector \vec{a}_b represented in body frame can be transformed back to the reference frame by

$$\vec{a} = A^T\vec{a}_b \quad (1.4)$$

Even though the attitude matrix is the fundamental quantity specifying the orientation, there are several other ways to represent attitude. A summary of attitude representation methods can be found in [1] and in table 1.1 in the appendix of [2].

The Euler Angle/Axis

The Euler angle/axis is a rotation defined around an axis, \vec{v} , by an angle θ . The simplest rotation in this representation is that of an angle θ around one of the orthonormal frame unit vectors. These simple rotations are called fundamental rotations. As there are three unit vectors in an the orthonormal coordinate frame there exists three fundamental rotations.

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{bmatrix} \quad (1.5)$$

$$R_y(\varphi) = \begin{bmatrix} \cos(\varphi) & 0 & -\sin(\varphi) \\ 0 & 1 & 0 \\ \sin(\varphi) & 0 & \cos(\varphi) \end{bmatrix} \quad (1.6)$$

$$R_z(\phi) = \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.7)$$

Where the notation is $R_{axis}(angle)$.

The rotation using this representation does not need to be done about one of the reference frame axis, it can be done about any axis \vec{v} .

The transformation from Euler angle/axis to attitude matrix (DCM) is given by [5]

$$R = \cos(\theta) I_{3 \times 3} + (1 - \cos(\theta))\vec{v}\vec{v}^T - \sin(\theta) E \quad (1.8)$$

Where E is the skew-symmetric operator of \vec{v} and $I_{3 \times 3}$ is a three by three identity matrix.

$$E = S(\vec{v}) = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \quad (1.9)$$

The Euler Angles

Another method of expressing the attitude of a body is through a sequence of fundamental rotations. Each fundamental rotation rotates the frame into a new position until it reaches the desired reference frame.

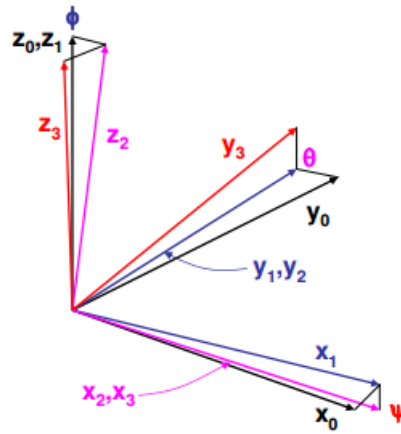


Figure 7 - Euler Angles Representation

The transformation to DCM can be done by successive fundamental rotations, with the total rotation given by the multiplication of the three successive matrices that represent the fundamental rotations. [5]

$$R_{total} = R_{xyz}(\phi, \varphi, \theta) = R_x(\theta)R_y(\varphi)R_z(\phi) \quad (1.10)$$

The Gibbs Vector

The Gibbs vector is easily expressed from the Euler angle/axis rotation information.

$$g_1 = e_1 \tan\left(\frac{\Phi}{2}\right) \quad (1.11)$$

$$g_2 = e_2 \tan\left(\frac{\Phi}{2}\right) \quad (1.12)$$

$$g_3 = e_3 \tan\left(\frac{\Phi}{2}\right) \quad (1.13)$$

The transformation to DCM is given by [5]

$$R = \frac{(1-\vec{g})\vec{1} + 2\vec{g}\vec{g}^T - G}{1+\vec{g}^2} \quad (1.14)$$

Where G is the skew-symmetric operator of \vec{g} .

The Quaternion

Quaternions are a number system that is an extension of complex numbers, which are very useful as an attitude representation. This usefulness derives from the fact that quaternion properties lead to linear kinematics equations and the quaternion unit norm allows for an easy way to preserve the orthogonality of the rotation matrix.

Quaternions are composed of a scalar part, q_0 , and a vector part, \vec{q} ,

$$\mathbf{q} = \begin{bmatrix} \vec{q} \\ q_0 \end{bmatrix} = q_0 + q_1\hat{i} + q_2\hat{j} + q_3\hat{k} \quad (1.15)$$

Parametrization of quaternions is easily done from Euler angle/axis

$$q_1 = e_1 \sin\left(\frac{\Phi}{2}\right) \quad (1.16)$$

$$q_2 = e_2 \sin\left(\frac{\Phi}{2}\right) \quad (1.17)$$

$$q_3 = e_3 \sin\left(\frac{\Phi}{2}\right) \quad (1.18)$$

$$q_0 = \cos\left(\frac{\Phi}{2}\right) \quad (1.19)$$

Transformation into DCM is given by [5]

$$R = (q_0^2 - \vec{q}^2)\vec{1} + 2\vec{q}\vec{q}^T - 2q_0Q \quad (1.20)$$

Where Q is the skew-symmetric operator of \vec{q} .

1.2.1.3 – Attitude Representation used in this project

In this thesis the attitude determination is done by an optical system consisting of 6 infrared cameras and 6 markers attached to the Triax testbed. After calibrated the Optitrack system gives the attitude of a trackable object composed of 6 markers attached to the Triax in quaternion form. Since the representation chosen for this project was the DCM it is necessary to transform the quaternion output from the Optitrack system to DCM. This can be done as follows

The transformation from quaternion representation to DCM representation is given by

$$R = \begin{bmatrix} 2q_0^2 + 2q_1^2 - 1 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_1q_2 + 2q_0q_3 & 2q_0^2 + 2q_2^2 - 1 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_2q_3 + 2q_0q_1 & 2q_0^2 + 2q_3^2 - 1 \end{bmatrix} \quad (1.21)$$

$$\vec{w} = \mathbf{q}\vec{v}\mathbf{q}^* = R\vec{v} \quad (1.22)$$

where \vec{v} is an arbitrary vector that we wish to rotate, \vec{w} is vector \vec{v} after rotation and \mathbf{q} is the quaternion.

1.3 – Attitude Control

Attitude control is the process of reorientating a spacecraft from one attitude to another attitude and maintaining this attitude irrespective of external disturbances such as external torques (gravity gradient, solar pressure, etc) or unintended torques (gas leak, crew movement, etc).

In general the attitude control system consists of attitude sensors, control algorithm/law and actuators. Sensors determine the current attitude, control algorithm/law determine the correction action in case the attitude is not the desired one. Actuators generate the necessary torque to change the spacecraft's attitude.

A control system can be either open- or closed-loop. An open-loop control system is an automatic control system that does not take into account the current state of the system, meaning it does not receive a feedback of the current output it is controlling. A closed-loop control system is one in which information about the state of the system being controlled is fed back into the controller so that it can perceive and compensate for any error or disturbances in the system to be controlled. [13]

1.3.1 – Basic Control Laws

A simple control law is based on the Euler angles attitude representation. For this representation the attitude error is given by

$$\phi_E = \phi_{Ref} - \phi \quad (1.23)$$

$$\theta_E = \theta_{Ref} - \theta \quad (1.24)$$

$$\psi_E = \psi_{Ref} - \psi \quad (1.25)$$

where ϕ_{Ref} , θ_{Ref} and ψ_{Ref} are the reference angles, and ϕ_E , θ_E and ψ_E are the error angles.

Considering a spacecraft with principal moments of inertia aligned with the Euler axes, meaning a spacecraft with a diagonal inertia matrix, using small angle approximation, the attitude dynamics are given by

$$T = [I] \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} \quad (1.26)$$

where $[I]$ is the inertia tensor.

As in this simplified system the equations are decoupled, thus the system can be broken down into 3 separate linear system with second order dynamics equations. A simple control law that stabilizes and tracks references is [13]

$$T_{C\phi} = k_{\phi P}(\phi_{Ref} - \phi) + k_{\phi D}\dot{\phi} \quad (1.27)$$

$$T_{C\theta} = k_{\theta P}(\theta_{Ref} - \theta) + k_{\theta D}\dot{\theta} \quad (1.28)$$

$$T_{C\psi} = k_{\psi P}(\psi_{Ref} - \psi) + k_{\psi D}\dot{\psi} \quad (1.29)$$

where $\dot{\phi}$, $\dot{\theta}$ and $\dot{\psi}$ are the angular velocities.

Designing this type control law is a well studied problem, treated in several control theory textbooks. It is apparent from the form of the above controllers that they are a PD controller.

The problem becomes more complex when large angle maneuvers are attempted. The simplified model given by (1.26) is no longer valid, the coupled nature of the system becomes apparent, and can no longer be solved by such simplifications. If the spacecraft does not need high maneuvering speeds, a slow change to the reference of the system can be used. By using a ramp instead of a step, or even a series of smaller steps what once was a large angle which would invalidate the simplification given by (1.26), becomes a problem that can still be treated using the controller in equations (1.27)-(1.29).

1.3.2 – Direct Cosine Error Matrix Control Law

As with the Euler angles control law the Direction Cosine Error Matrix control law is based on the error between the current orientation of the spacecraft and the desired orientation. Let $[R_C]$ be the current orientation of the spacecraft and $[R_{Ref}]$ be the desired orientation of the spacecraft.

$$\vec{a}_C = [R_C]\vec{a} \quad (1.30)$$

$$\vec{a}_{Ref} = [R_{Ref}]\vec{a} \quad (1.31)$$

by solving (43) for \vec{a} and substituting the results in (42) we have

$$\vec{a}_C = [R_C][R_{Ref}]^{-1}\vec{a}_{Ref} = [R_C][R_{Ref}]^T\vec{a}_{Ref} = [R_E]\vec{a}_{Ref} \quad (1.32)$$

$$[R_E] = [R_C][R_{Ref}]^{-1} = [R_C][R_{Ref}]^T \quad (1.33)$$

This indicates that when $\vec{a}_C = \vec{a}_{Ref}$, $[R_E]$ will be the identity matrix, as the vector \vec{a} will have the same representation on both the current coordinate system and in the desired coordinate system.

$$[R_E] = [R_C][R_{Ref}]^T \quad (1.34)$$

$$[R_E] = \begin{bmatrix} a_{11C} & a_{12C} & a_{13C} \\ a_{21C} & a_{22C} & a_{23C} \\ a_{31C} & a_{32C} & a_{33C} \end{bmatrix} \begin{bmatrix} a_{11Ref} & a_{12Ref} & a_{13Ref} \\ a_{21Ref} & a_{22Ref} & a_{23Ref} \\ a_{31Ref} & a_{32Ref} & a_{33Ref} \end{bmatrix} = \begin{bmatrix} a_{11E} & a_{12E} & a_{13E} \\ a_{21E} & a_{22E} & a_{23E} \\ a_{31E} & a_{32E} & a_{33E} \end{bmatrix} \quad (1.35)$$

For matrix $[R_E]$ to become identity its off-diagonal elements must be driven to zero.

To better understand why it is necessary to zero the off-diagonal elements of $[R_E]$ and how to do that take element a_{12E} for example. The element a_{12E} is the scalar dot product between the \vec{X}_C -axis of the current body frame and the \vec{Y}_{Ref} -axis of the desired frame.

$$a_{12E} = \vec{X}_C \cdot \vec{Y}_{Ref} \quad (1.36)$$

zeroing a_{12E} is analogous to make \vec{X}_C perpendicular to \vec{Y}_{Ref} , and this can be achieved by rotating the spacecraft about \vec{Z}_C -axis, body frame.

The same logic can be applied to $a_{13E} = \vec{X}_C \cdot \vec{Z}_{Ref}$ and $a_{23E} = \vec{Y}_C \cdot \vec{Z}_{Ref}$, thus by zeroing elements a_{12E} , a_{13E} and a_{23E} one brings the spacecraft body frame to the desired frame. Thus the control laws are [13]

$$T_x = k_{Px}a_{23E} + k_{Dx}p \quad (1.37)$$

$$T_y = k_{Py}a_{13E} + k_{Dy}q \quad (1.38)$$

$$T_z = k_{Pz}a_{12E} + k_{Dz}r \quad (1.39)$$

The terms p, q and r are the angular velocities of the body frame axis in the reference frame, which are used for damping purposes as the spacecraft dynamics consists of a double integrator on each axis. In equation (1.37), (1.38) and (1.39) it is possible to exchange a_{ij} by $-a_{ji}$. [13]

Chapter 2 – The RCAC

2.1 – Introduction

Retrospective Cost Adaptive Control (RCAC) is a MIMO controller that uses limited information regarding the system's model. In this thesis the only information on the system used in the controller was an estimation of the first Markov parameter. This estimation was obtained through the impulse response of the Triax model in a simulation.

RCAC basically works by collecting past data, both control inputs and system outputs, and calculating the best control action (with regards to the performance variable) it could have taken if it had the information it now has. Then it uses this knowledge to calculate a controller, which would have taken those actions. RCAC then uses this controller to calculate the next control action. This process repeats at every iteration.

Therefore RCAC is always taking past data and using it to try and find the controller that would have taken the optimal actions to minimize the cost function and thus the performance variable.

2.2 – The RCAC Algorithm

Given, the strictly casual discrete-time system

$$X(k+1) = AX(k) + Bu(k) + Fw(k) \quad (2.1)$$

$$Y(k) = CX(k) + v(k) \quad (2.2)$$

$$z(k) = EX(k) - r(k) \quad (2.3)$$

where X is the system state vector, $\in \mathbb{R}^{lx}$, Y is the system output, $\in \mathbb{R}^{ly}$, z is the system performance variable, $\in \mathbb{R}^{lz}$, u is the system input, $\in \mathbb{R}^{lu}$, r is reference, $\in \mathbb{R}^{lr}$, w is the process noise, $\in \mathbb{R}^{lw}$, v is the measurement noise, $\in \mathbb{R}^{lv}$.

The state at any time step k can be calculated by

$$X(k) = A^n X(k-n) + \sum_{i=1}^n A^{i-1} [Bu(k-i) + Fw(k-i)] \quad (2.4)$$

The performance variable at any time step k can be calculated by

$$z(k) = EA^n X(k-n) + E \sum_{i=1}^n A^{i-1} Bu(k-i) + E \sum_{i=1}^n A^{i-1} Fw(k-i) - r(k) \quad (2.5)$$

2.2.1 – Markov Parameters

Markov parameters sequence of a state space model is a matrix impulse response. [21]

The Markov parameters H_i of the system input to performance variable can be defined as

$$H_i \triangleq EA^{i-1}B, i \geq 1 \quad (2.6)$$

A transfer function, G , can be given by a Laurent Expansion

$$G(q) \triangleq E(q-A)^{-1}B \quad (2.7)$$

$$G(\mathbf{q}) = \sum_{i=0}^{\infty} \frac{H_i}{\mathbf{q}^i}, \forall \mathbf{q} > \rho \quad (2.8)$$

$$G(\mathbf{q}) \approx \sum_{i=0}^{n_H} \frac{H_i}{\mathbf{q}^i} \quad (2.9)$$

where \mathbf{q} is the forward-shift operator, ρ is the spectral radius of the plant.

Utilizing the Markov parameter definition above the performance variable at an arbitrary time step k can be written as

$$z(k) = \frac{E_1 A^n}{\mathbf{q}^n} X(k) + \sum_{i=0}^{n_H} \frac{H_i}{\mathbf{q}^i} u(k) + \sum_{i=0}^{n_H} \frac{H'_i}{\mathbf{q}^i} w(k) - r(k) \quad (2.10)$$

2.2.2 – Retrospective Performance

For a controller given by

$$u(k) = \sum_{i=1}^{n_c} P_i(k)u(k-i) + \sum_{i=0}^{n_c} Q_i(k)z(k-i) = \Phi(k)\theta(k) \quad (2.11)$$

where $\Phi(k)$ is the regressor matrix, $\theta(k)$ a vector containing the controller parameters

$$\Phi(k) \triangleq I_{l_u} \otimes \begin{bmatrix} u(k-1) \\ \vdots \\ u(k-n_c) \\ z(k) \\ \vdots \\ z(k-n_c) \end{bmatrix}^T \in \mathbb{R}^{l_u \times l_\theta} \quad (2.12)$$

where I_{l_u} is an $l_u \times l_u$ identity matrix, \otimes is the Kronecker product, $l_\theta = n_c l_u (l_u + l_z)$, $P_i(k)$ and $Q_i(k)$ are the controller gain matrices.

The retrospective control is defined as

$$\hat{u}(k) \triangleq \Phi(k)\hat{\theta}(k) \quad (2.13)$$

the retrospective performance is then

$$\hat{z}(k) \triangleq z_f(k) + \Phi_f(k)\hat{\theta} - u_f(k) \quad (2.14)$$

where $z_f(k)$, $\Phi_f(k)$ and $u_f(k)$ are the filtered $z(k)$, $\Phi(k)$ and $u(k)$ respectively.

In the version of RCAC used in this thesis

$$z_f(k) \triangleq G_z(\mathbf{q}^{-1})z(k) = D_z(\mathbf{q}^{-1})^{-1}N_z(\mathbf{q}^{-1})z(k) \quad (2.15)$$

$$z_f(k) = \sum_{i=0}^{n_z} N_{z_i}z(k-i) - \sum_{i=0}^{\bar{n}_z} D_{z_i}z_f(k-i) \quad (2.16)$$

$$z_f(k) = \bar{N}_z \bar{Z}(k) - \bar{D}_z \bar{Z}_f(k) \quad (2.17)$$

Since \bar{N}_z and \bar{D}_z were defined as I and 0 respectively we have

$$z_f(k) = \bar{Z}(k) \quad (2.18)$$

$$u_f(k) \triangleq G_u(\mathbf{q}^{-1})u(k) = D_u(\mathbf{q}^{-1})^{-1}N_u(\mathbf{q}^{-1})u(k) \quad (2.19)$$

$$u_f(k) = \sum_{i=0}^{n_u} N_{u_i}u(k-i) - \sum_{i=0}^{\bar{n}_u} D_{u_i}u_f(k-i) \quad (2.20)$$

$$u_f(k) = \bar{N}_u\bar{u}(k) - \bar{D}_u\bar{u}_f(k) \quad (2.21)$$

Since \bar{N}_u and \bar{D}_u were defined as H_1 and 0, respectively we have

$$u_f(k) = H_1\bar{u}(k) \quad (2.22)$$

$$\Phi_f(k) \triangleq G_\Phi(\mathbf{q}^{-1})\Phi(k) = D_\Phi(\mathbf{q}^{-1})^{-1}N_\Phi(\mathbf{q}^{-1})\Phi(k) \quad (2.23)$$

$$\Phi_f(k) = \sum_{i=0}^{n_\Phi} N_{\Phi_i}\Phi(k-i) - \sum_{i=0}^{\bar{n}_\Phi} D_{\Phi_i}\Phi_f(k-i) \quad (2.24)$$

$$\Phi_f(k) = \bar{N}_\Phi\bar{\Phi}(k) - \bar{D}_\Phi\bar{\Phi}_f(k) \quad (2.25)$$

Since \bar{N}_Φ and \bar{D}_Φ were defined as $[0 \ H_1]$ and 0, respectively we have

$$\Phi_f(k) = [0 \ H_1]\bar{\Phi}(k) \quad (2.26)$$

where

$$\bar{\Phi}(k) \triangleq \begin{bmatrix} \Phi(k-1) \\ \vdots \\ \Phi(k-n_f) \end{bmatrix} \in \mathbb{R}^{n_f l_u \times l_\theta} \quad (2.27)$$

$$\bar{U}(k) \triangleq \begin{bmatrix} u(k-1) \\ \vdots \\ u(k-n_f) \end{bmatrix} \in \mathbb{R}^{n_f l_u} \quad (2.28)$$

$$\bar{Z}(k) \triangleq \begin{bmatrix} z(k-1) \\ \vdots \\ z(k-n_f) \end{bmatrix} \in \mathbb{R}^{n_f l_z} \quad (2.29)$$

2.2.3 – Retrospective Cost Function

The cost function must be designed in such a way as to minimize the retrospective performance, as by consequence of that we minimize the performance variable of the system.

The cost function used was

$$J(k, \theta(k)) = \sum_{i=k_0}^k \left(\hat{z}(i)^T R_z(k, i) \hat{z}(i) + [\Phi_f(i)\hat{\theta}]^T R_{u_f}(k, i) [\Phi_f(i)\hat{\theta}] + [\Phi(i)\hat{\theta}]^T R_u(k, i) [\Phi(i)\hat{\theta}] \right) + [\hat{\theta} - \theta(0)]^T R_\theta(k) [\hat{\theta} - \theta(0)] + [\hat{\theta} - \theta(k-1)]^T R_\Delta(k) [\hat{\theta} - \theta(k-1)] \quad (2.30)$$

Differentiating (2.30) with respect to $\hat{\theta}$ we obtain

$$\frac{d}{d\hat{\theta}}J = 2 \sum_{i=k_0}^k \left(\hat{z}(i)^T R_z(k, i) \Phi_f(i) + [\Phi_f(i) \hat{\theta}]^T R_{u_f}(k, i) \Phi_f(i) + [\Phi(i) \hat{\theta}]^T R_u(k, i) \Phi(i) \right) + 2[\hat{\theta} - \theta(0)]^T R_\theta + 2[\hat{\theta} - \theta(k-1)]^T R_\Delta(k) \quad (2.31)$$

$$\frac{d}{d\hat{\theta}}J = 2\mathcal{A}(k)^T + 2\hat{\theta}^T \mathcal{P}(k)^{-1} \quad (2.32)$$

which has a minimizer given by [9]

$$\theta(k) = \hat{\theta} = -\mathcal{P}(k)\mathcal{A}(k) \quad (2.33)$$

where

$$\mathcal{P}(k) \triangleq \left[\sum_{i=k_0}^k \left(\Phi_f(i)^T \left[R_z(k, i) + R_{u_f}(k, i) \right] \Phi_f(i) + \Phi(i)^T R_u(k, i) \Phi(i) \right) + R_\theta(k) + R_\Delta(k) \right]^{-1} \quad (2.34)$$

$$\mathcal{A}(k) \triangleq \sum_{i=k_0}^k \left(\Phi_f(i)^T R_z(k, i) [z(i) - u_f(i)] \right) - R_\theta(k)\theta(0) - R_\Delta(k)\theta(k-1) \quad (2.35)$$

We define the penalty weights to be constant, then

$$\mathcal{P}(k) = \left[\mathcal{P}(k-1)^{-1} + \Phi_f(k)^T R_z(k) \Phi_f(k) + \Phi_f(k)^T R_{u_f}(k) \Phi_f(k) + \Phi(k)^T R_u(k) \Phi(k) \right]^{-1} \quad (2.36)$$

$$\mathcal{A}(k) = \mathcal{A}(k-1) + \Phi_f(k)^T R_z(k) [z(k) - u_f(k)] + R_\Delta[\theta(k-2) - \theta(k-1)] \quad (2.37)$$

The initial values are given by

$$\mathcal{P}(0) = [R_\theta + R_\Delta]^{-1} \quad (2.38)$$

and θ_0 is chosen arbitrarily.

2.3 – RCAC Design for Attitude Control

2.3.1 – Attitude Error Representation

RCAC is not capable of using a matrix as the performance variable, it can only use vectors. Thus it becomes necessary to find a useful transformation that from the error rotation matrix an error vector representative of the error is calculated.

As presented in [14], we can reformulate the error matrix using a vector parameter S .

$$\vec{S} \triangleq \sum_{i=1}^3 a_i (\tilde{R}^T e_i) \times e_i \quad (2.39)$$

where

a_i is a diagonal element of $A_{att} = \text{diag}(a_1, a_2, a_3)$, which is a positive definite weight matrix

e_i is the i th column of a 3x3 identity matrix

Using the result of equation (2.39) we can now define the performance variable of the controller

$$z \triangleq \begin{bmatrix} \vec{S} \\ \vec{w}_E \end{bmatrix} \quad (2.40)$$

where

$$\vec{w}_E = [\vec{w}_{Ref} - \vec{w}] \quad (2.41)$$

Another way to reformulate the error matrix into a vector is given in [15], first the attitude error matrix is converted into Euler axis/angle, then the following equation gives the error vector

$$\vec{E} = \theta \vec{v} \quad (2.42)$$

where θ is the Euler angle, \vec{v} is the Euler vector.

In both these reformulations the idea is that the vector goes to zero when the attitude error matrix goes to identity.

2.3.2 – Markov Parameters

As this control has as a main point of interest the fact that it does not require much in the way of a spacecraft model, the Markov parameter that it does require was considered a tuning parameter. Although for comparison purposes an estimative of the actual Markov parameter was calculated.

The first Markov parameter of the spacecraft can be obtained by giving the system an impulse and acquiring the first non-zero output. This must be done separately for every input the system has, thus in this case 3 times (for the system in which the controller output is directly related to the torques in all three axis) or 4 times (for the system in which the controller output is related to the thrusters that generate the torque). This method was used twice one with the Simulink model and one with the Triax equipment.

2.3.2.1 – Simulink Model Markov Parameter

The Simulink setup for the extraction of the first Markov parameter from the spacecraft's model can be seen in Figure 8.

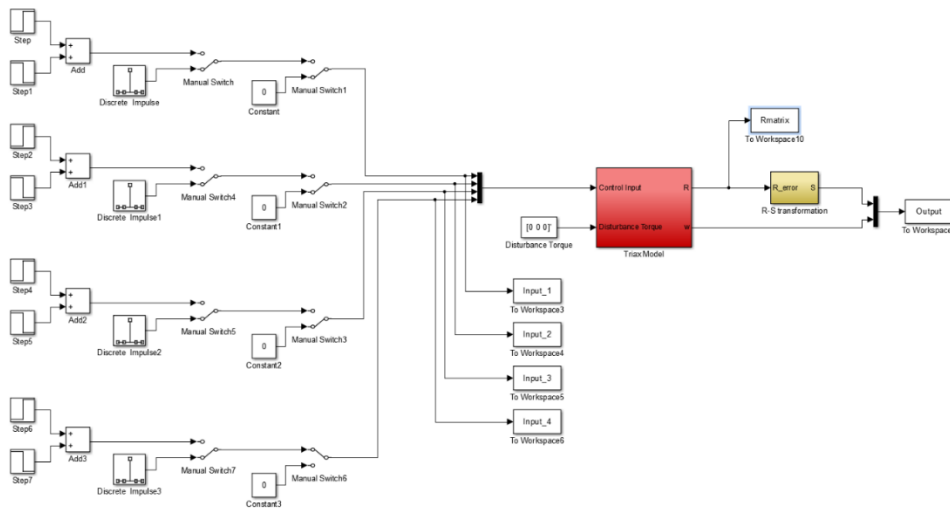


Figure 8 - Simulink scheme for Markov Parameter extraction

The method used to find the Markov parameter of the system model was to give each actuator an impulse, separately, and the first output series with non-zero values is one of the columns of the Markov parameter matrix. Giving an impulse to thruster one we had as output a 6x1 column vector, the first column of the Markov parameter matrix below. Afterwards thruster 2 was given an impulse and as system output we had another 6x1 column vector, the second column of the Markov parameter matrix below. This process was repeated for thrusters 3 and 4 to extract columns 3 and 4 of the Markov parameter matrix.

$$H = \begin{bmatrix} -1.349 \times 10^{-6} & -1.349 \times 10^{-6} & 0 & 0 \\ 0 & 0 & -4.477 \times 10^{-6} & -4.477 \times 10^{-6} \\ 1.645 \times 10^{-7} & -1.645 \times 10^{-7} & 1.641 \times 10^{-6} & -1.641 \times 10^{-6} \\ -2.009 \times 10^{-4} & -2.009 \times 10^{-4} & 0 & 0 \\ 0 & 0 & -2.009 \times 10^{-4} & -2.009 \times 10^{-4} \\ 2.216 \times 10^{-5} & -2.216 \times 10^{-5} & 7.017 \times 10^{-5} & -7.017 \times 10^{-5} \end{bmatrix}$$

2.3.2.2 – Triax Equipment Markov Parameter

The Simulink setup for the extraction of the first Markov parameter from the Triax equipment can be seen in Figure 9.

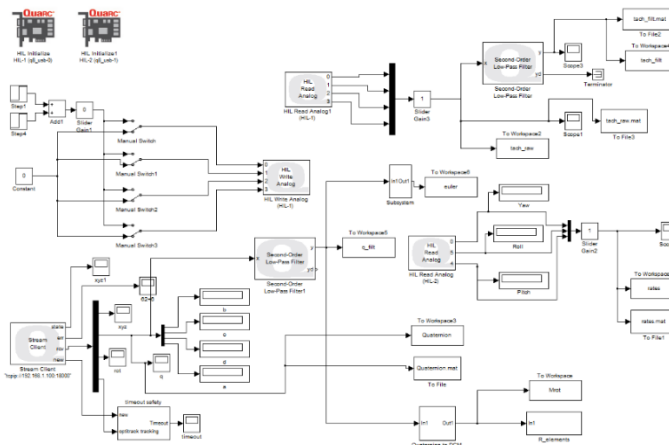


Figure 9 - Simulink Scheme for Markov Parameter extraction

Here the process to find the Markov parameter matrix was the same as in the previous section. The Triax was balanced and stabilized and then each thruster was separately given an impulse input and the output was collected. After each impulse input the output of the system was saved and the first non-zero output series of each input was stacked to form the Markov parameter matrix below.

$$H = \begin{bmatrix} -0.00013 & -0.00013 & 0 & 0 \\ 0 & 0 & -0.00045 & -0.00045 \\ 0.00002 & -0.00002 & 0.000160 & -0.000160 \\ -0.0201 & -0.0201 & 0 & 0 \\ 0 & 0 & -0.02010 & -0.02010 \\ 0.00222 & -0.00222 & 0.007020 & -0.00702 \end{bmatrix}$$

2.3.2.3 – Tuning Markov Parameter

As mentioned before the Markov parameter can be considered a tuning parameter of the controller. In the course of the various simulations performed in this project a great number of values of Markov parameters were attempted until a final value was found that fit the system and improved the controller considerably.

The tuning of the Markov parameter proceeded much like with any other tuning parameter, an arbitrary value was used as a first guess and depending on the performance of the controller such as rising time, overshoot, oscillation, the Markov parameter was modified accordingly. More about the tuning of Markov parameter in section 3.2.1.

This final value found was

$$H = \begin{bmatrix} -1 \times 10^{-6} & -1 \times 10^{-6} & 0 & 0 \\ 0 & 0 & -1 \times 10^{-6} & -1 \times 10^{-6} \\ 1 \times 10^{-6} & -1 \times 10^{-6} & 1 \times 10^{-6} & -1 \times 10^{-6} \\ -3 \times 10^{-4} & -3 \times 10^{-4} & 0 & 0 \\ 0 & 0 & -3 \times 10^{-4} & -3 \times 10^{-4} \\ 3 \times 10^{-4} & -3 \times 10^{-4} & 3 \times 10^{-4} & -3 \times 10^{-4} \end{bmatrix}$$

Chapter 3 – Simulation

To demonstrate the effectiveness of RCAC and to better understand the effects of the tuning parameters, several simple system were tested. The systems used for this purpose were, a simplified model of Aircraft pitch, a DC motor position, an inverted pendulum and a single degree of freedom spacecraft.

3.1 – Preliminary Systems

3.1.1 – Aircraft Pitch

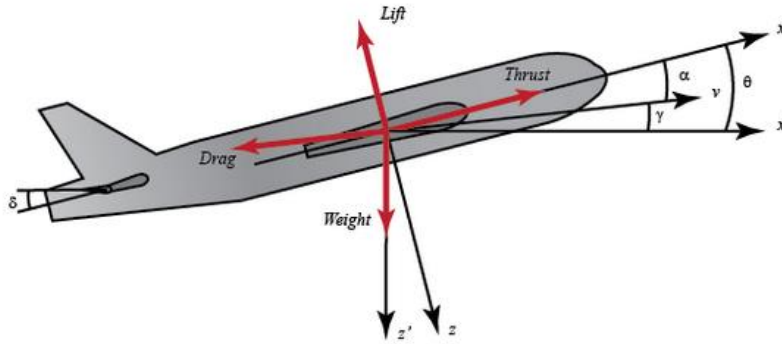


Figure 10 - Aircraft Pitch System

Let that the aircraft be in steady-cruise at constant altitude and velocity, therefore the drag, thrust, weight and lift forces are balanced in the x-y plane. The simplified model also assumes that the aircraft has a controller that maintains the aircraft's speed regardless of the pitch angles. With these assumptions the system dynamics are given by [17]

$$\dot{\alpha} = \mu\Omega\sigma \left[-(C_L + C_D)\alpha + \frac{q}{(\mu - C_L)} - (C_W \sin(\gamma))\theta + C_L \right] \quad (3.1)$$

$$\dot{q} = \frac{\mu\Omega}{2i_{yy}} \{ [C_M - \eta(C_L + C_D)]\alpha + [C_M + \sigma C_M(1 - \mu C_L)]q + (\eta C_W \sin(\gamma))\delta \} \quad (3.2)$$

$$\dot{\theta} = \Omega q \quad (3.3)$$

where α is the angle of attack, q is the pitch rate, θ is the pitch angle, δ is the elevator deflection angle, $\mu = \frac{\rho S \bar{c}}{4m}$, ρ is the density of air, S is the platform area of the wing, \bar{c} is the average chord length, m is the mass of the aircraft, $\Omega = \frac{2U}{\bar{c}}$, U is the equilibrium flight speed, C_T is the coefficient of thrust, C_D is the coefficient of drag, C_L is the coefficient of lift, C_W is the coefficient of weight, C_M is the coefficient of pitch moment, γ is the flight path angle, $\sigma = \frac{1}{1 + \mu C_L}$, i_{yy} is the normalized moment of inertia, $\eta = \mu \sigma C_M$.

Linearizing equations (3.1), (3.2) and (3.3) by small angle sine approximation and plugging numerical values in equation we have

$$\dot{\alpha} = -0.313\alpha + 56.7q + 0.232\delta \quad (3.4)$$

$$\dot{q} = -0.0139\alpha - 0.426q + 0.0203\delta \quad (3.5)$$

$$\dot{\theta} = 56.7q \quad (3.6)$$

The state space representation of the system is

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -0.313 & 56.7 & 0 \\ -0.0139 & -0.426 & 0 \\ 0 & 56.7 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} 0.232 \\ 0.0203 \\ 0 \end{bmatrix} \delta \quad (3.7)$$

$$y = [0 \quad 0 \quad 1] \begin{bmatrix} \alpha \\ q \\ \theta \end{bmatrix} \quad (3.8)$$

The Simulink scheme used in the simulation can be seen in Figure 11.

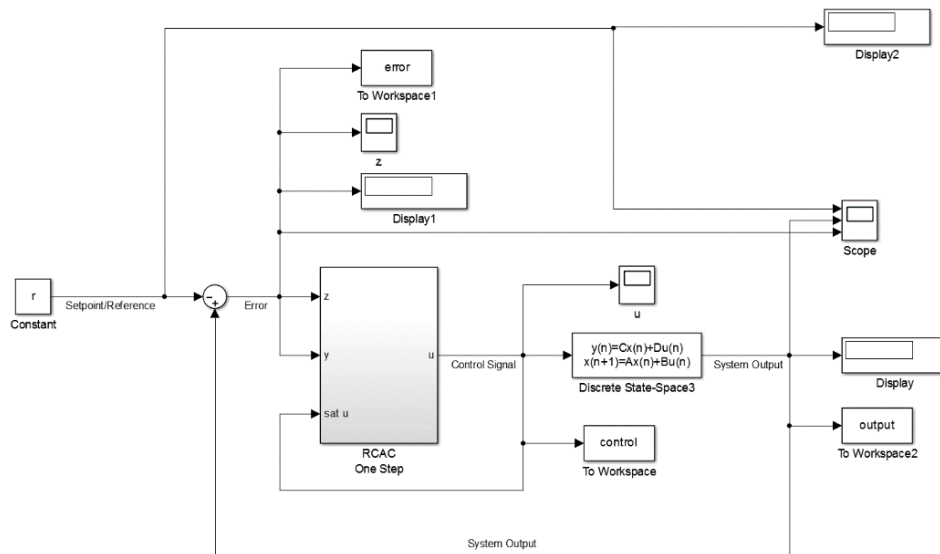


Figure 11 - Aircraft/DC Motor/Inverted Pendulum/Single axis spacecraft Simulation Scheme

The rising time and settling time of the Pitch angle were around 1 second as can be observed in Figure 12.

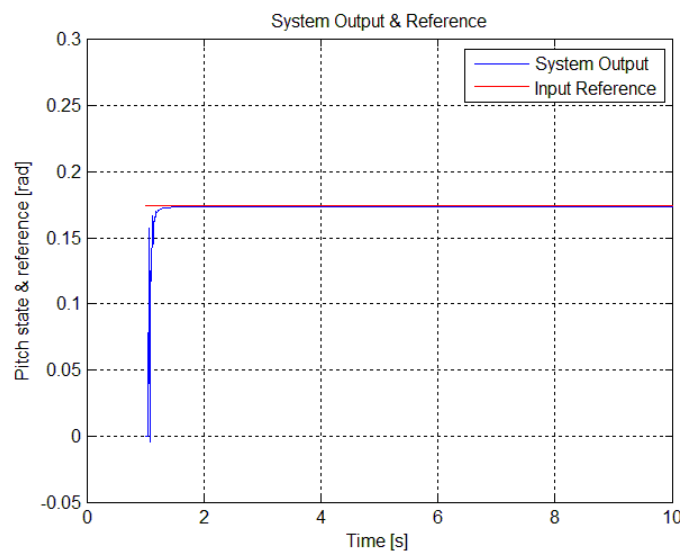


Figure 12 - Aircraft Pitch state (system output) & reference 10 [deg]

As the system is linear, RCAC can track the setpoint perfectly, although at large enough angles the system oscillates quite severely during the transitory as can be observed in Figure 13.

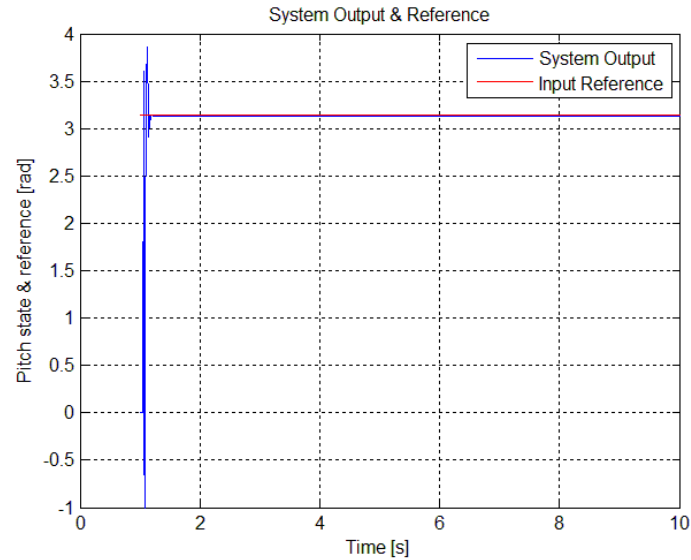


Figure 13 - Aircraft Pitch state (system output) & reference 180 [deg]

By increasing the control and transient penalties it was possible to reduce the over- and undershoot, and the oscillation, but not do away with it entirely.

The tuning parameters were

Performance penalty	$60 \times I_{l_z}$
Filtered Control penalty	$0 \times I_{l_z}$
Control penalty	$0.1 \times I_{l_u}$
Transient Penalty	$1 \times I_{l_\theta}$
Delta theta Penalty	$0 \times I_{l_\theta}$

Table 2 - Aircraft Pitch RCAC tuning parameters

3.1.2 – DC Motor Position

The DC motor is a widely used equipment in an equally wide variety of fields such as a computer HD, robotic servo actuators and many others. Its well behaved torque vs. rotation speed curve, see Figure 14 [19], allows for simple control schemes and by making small restrictions, maintaining the magnetic field constant, the system can be considered a linear system. The mathematical modeling of the motor is as follows.

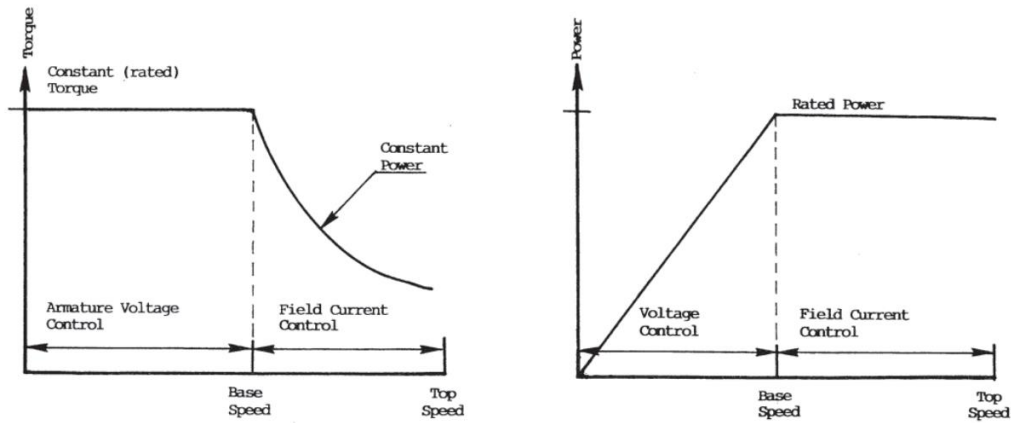


Figure 14 - DC Motor Characteristics

For an armature controlled motor, i.e. a motor controlled by the voltage applied to the motor armature, the torque is proportional to the armature current

$$T = K_t i \quad (3.9)$$

The back emf, e , is proportional to the angular velocity of the motor

$$e = K_e \dot{\theta} \quad (3.10)$$

When using SI units the motor torque constant and the back emf constant have the same value, therefore

$$K_t = K_e = K \quad (3.11)$$

Using Newton's second law we arrive at

$$J\ddot{\theta} + b\dot{\theta} = Ki \quad (3.12)$$

and using Kirchhoff's law we have

$$L \frac{di}{dt} + Ri = V - K\dot{\theta} \quad (3.13)$$

where J is the moment of inertia of the rotor, b is the motor viscous friction constant, K_b is the electromotive force constant, K_t is the motor torque constant, R is the electric resistance, L is the electric inductance.

The system's state space representation is

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{i}' \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{b}{J} & \frac{K}{L} \\ 0 & -\frac{K}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} V \quad (3.14)$$

$$y = [1 \quad 0 \quad 0] \begin{bmatrix} \theta \\ \dot{\theta} \\ i \end{bmatrix} \quad (3.15)$$

The Simulink scheme used in the simulation was the same as the one used in the Aircraft Pitch system in Figure 11 differing only in the system model (the discrete state space block).

The input for this system was a step of 1 radian, the results of RCAC on this system can be seen in Figure 15.

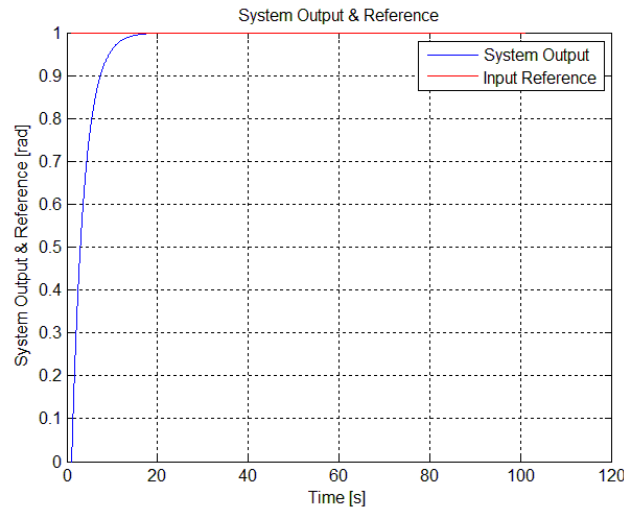


Figure 15 - DC Motor Position (system output) & Reference 1 [rad]

As expected for a linear system, RCAC had little problem to control this system. Regarding the tuning of the RCAC the biggest issue was in finding penalty weights that while aggressive did not cause overshoot. Most of the weights tried had satisfactory results, some with very high overshoot, others with very long settling times, but in general all reached the goal of bringing the position of the motor to the desired value.

Changing the size of the step input made little difference in regards to the output (motor position) curve, a small increase in rising time, as expected since the system is linear. This can be observed in Figure 16 where the step input was of 1.5 radians instead of 1 radian used in the first case.

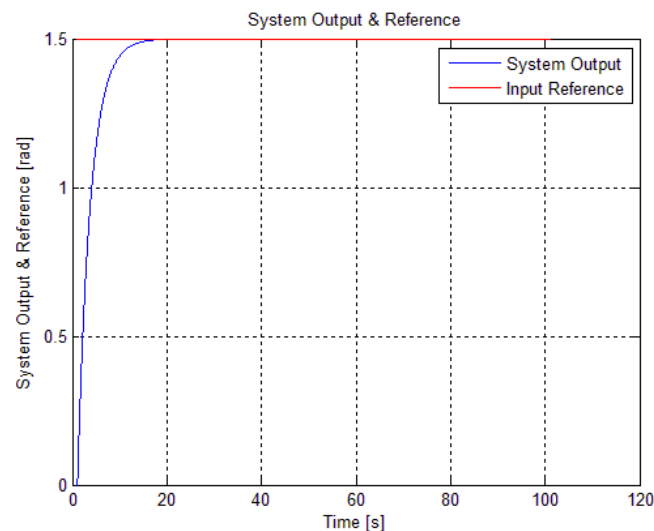


Figure 16 - DC Motor Position (system output) & Reference 1.5 [rad]

The RCAC tuning parameters for this case were

Performance penalty	$1 \times I_{l_z}$
Filtered Control penalty	$0 \times I_{l_z}$
Control penalty	$100 \times I_{l_u}$
Transient Penalty	$0.1 \times I_{l_\theta}$
Delta theta Penalty	$0 \times I_{l_\theta}$

Table 3 - DC Motor RCAC tuning parameters

3.1.3 – Inverted Pendulum

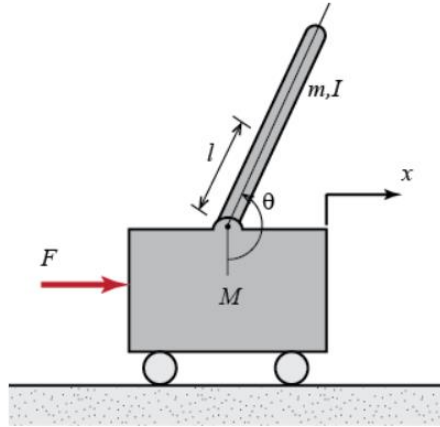


Figure 17 - Inverted Pendulum System

The inverted pendulum is a classic example commonly found in control system textbooks [17]. The system is unstable and its dynamics are nonlinear. The goal of the control is to maintain the balance of the pendulum at $\theta = 180^\circ$ by moving the cart to which the pendulum is attached to. This example is related to attitude control of a booster rocket at takeoff.

By summing the horizontal forces in action on the cart we have

$$M\ddot{x} + b\dot{x} + N = F \quad (3.16)$$

where N is the normal of the pendulum and F is the force applied to the cart adding the horizontal forces in the pendulum we have

$$N = m\ddot{x} + ml\ddot{\theta} \cos(\theta) - ml\dot{\theta}^2 \sin(\theta) \quad (3.17)$$

substituting equation (3.17) into equation (3.16) we have

$$(M + m)\ddot{x} + b\dot{x} + ml\ddot{\theta} \cos(\theta) - ml\dot{\theta}^2 \sin(\theta) = F \quad (3.18)$$

Now adding the forces perpendicular to the pendulum we have

$$P \sin(\theta) + N \cos(\theta) - mg \sin(\theta) = ml\ddot{\theta} + m\ddot{x} \cos(\theta) \quad (3.19)$$

combining equation (3.19) with the sum of the moments about the centroid of the pendulum

$$-Pl \sin(\theta) - Nl \cos(\theta) = I\ddot{\theta} \quad (3.20)$$

we have

$$(I + ml^2)\ddot{\theta} + mgl \sin(\theta) = -ml\ddot{x} \cos(\theta) \quad (3.21)$$

Linearizing the system around 180° , by making $\theta = \pi + \phi$, where ϕ is the deviation of the angle from 180° we can assume that

$$\cos(\theta) = \cos(\pi + \phi) \approx -1 \quad (3.22)$$

$$\sin(\theta) = \sin(\pi + \phi) \approx -\phi \quad (3.23)$$

$$\dot{\theta}^2 = \dot{\phi}^2 \approx 0 \quad (3.24)$$

thus we have

$$(I + ml^2)\ddot{\phi} + mgl\phi = -ml\ddot{x} \quad (3.25)$$

$$(M + m)\ddot{x} + b\dot{x} + ml\ddot{\phi} = F \quad (3.26)$$

where, M is the mass of the cart, m is the mass of the pendulum, b is the coefficient of friction for cart, l is the length to pendulum center of mass, I is the mass moment of inertia of the pendulum, F is the force applied to the cart, x is the cart position coordinate, θ is the pendulum angle from vertical (down).

The state space representation of the system is

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\phi} \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-(I+ml^2)b}{I(M+m)+Mml^2} & \frac{m^2gl^2}{I(M+m)+Mml^2} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-mlb}{I(M+m)+Mml^2} & \frac{mgl(M+m)}{I(M+m)+Mml^2} & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{I+ml^2}{I(M+m)+Mml^2} \\ 0 \\ \frac{ml}{I(M+m)+Mml^2} \end{bmatrix} u \quad (3.27)$$

$$y = [0 \quad 0 \quad 1 \quad 0] \begin{bmatrix} x \\ \dot{x} \\ \phi \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u \quad (3.28)$$

The Simulink scheme used in the simulation was the same as the one used in the Aircraft Pitch system in Figure 18.

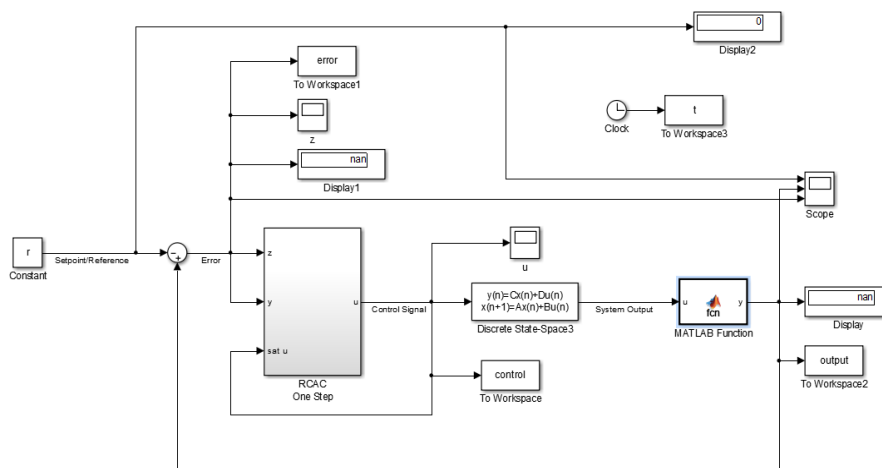


Figure 18 - Pendulum Simulink Schematic

The biggest difference between this simulation and the others is that in this case it is a stabilization problem and not a tracking problem. The system was given an initial state equal to 5° and RCAC's goal was to stabilize the system at 0° , which due to the linearization means that the pendulum is at 180° , pointing upwards.

The results of RCAC on this system can be seen in Figure 19.

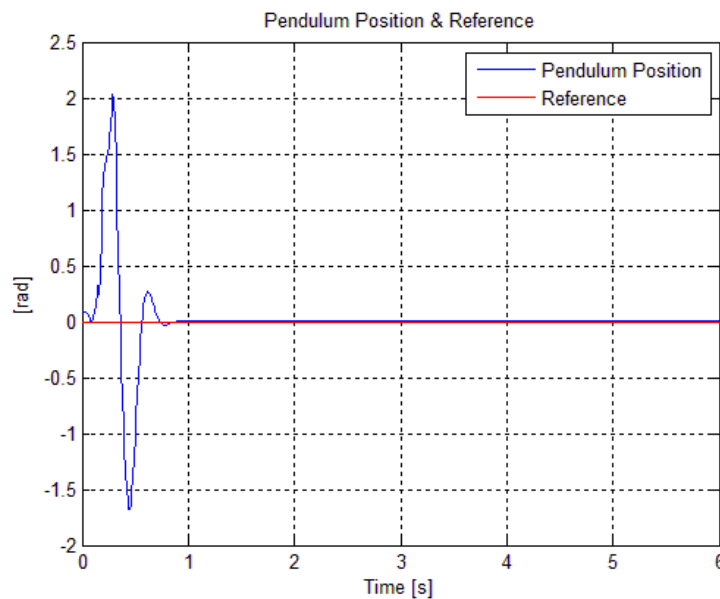


Figure 19 - Inverted Pendulum Position (system output) & Reference 5 [deg] initial condition

Other simulations were made in order to find the limits of the controller and it was found that when the initial condition is set to 10° the system oscillates severely before settling as can be seen in Figure 20.

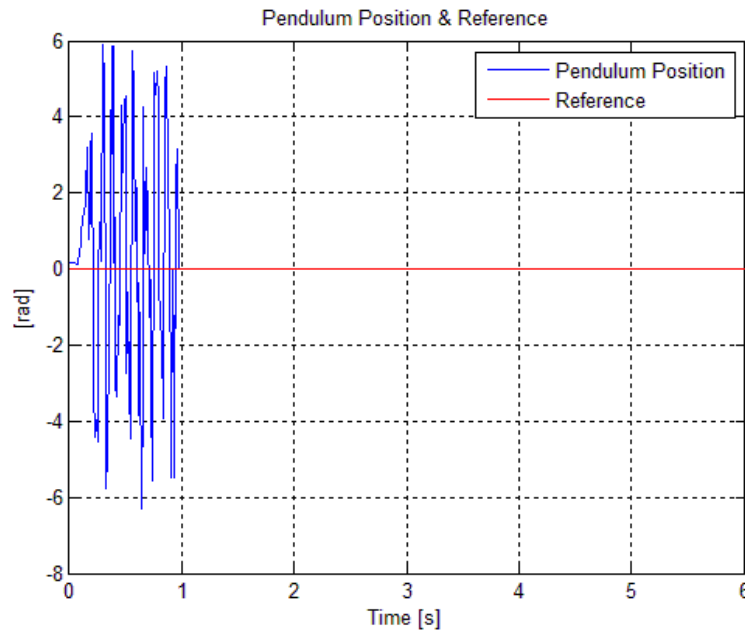


Figure 20 - Inverted Pendulum Position (system output) & Reference 10 [deg] initial condition

The RCAC tuning parameters for this result were

Performance penalty	$1 \times I_{l_z}$
Filtered Control penalty	$0 \times I_{l_z}$
Control penalty	$0 \times I_{l_u}$
Transient Penalty	$10^{-9} \times I_{l_\theta}$
Delta theta Penalty	$0 \times I_{l_\theta}$

Table 4 - Pendulum RCAC tuning parameters

3.1.4 – Single Axis Spacecraft

Before attempting to control a full 3 degrees of freedom spacecraft it is worth attempting to control a single axis spacecraft.

The spacecraft is modeled as a spinning solid disk. The moment of inertia of the disk is $20 \text{ [kg/m}^2\text{]}$. Using Newton's second law we have

$$\sum T = I\ddot{\theta} \quad (3.29)$$

The torques affecting the spacecraft is only the control torque generated by the thruster thus

$$T_c = I\ddot{\theta} \quad (3.30)$$

where $\ddot{\theta}$ is the angular acceleration of the spacecraft

The control torque is given by the equation

$$T_c = 8.12V \quad (3.31)$$

where V is the control signal given to the thruster, thus we have

$$\ddot{\theta} = \frac{T_c}{I} = \frac{8.12}{20} V = 0.406V \quad (3.32)$$

Therefore the state space representation of the system is

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0.406 \end{bmatrix} V \quad (3.33)$$

$$y = [1 \quad 0] \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} \quad (3.34)$$

With no limit to the thrust generated by the thruster even a 1 radian step can be tracked relatively closely as in Figure 21.

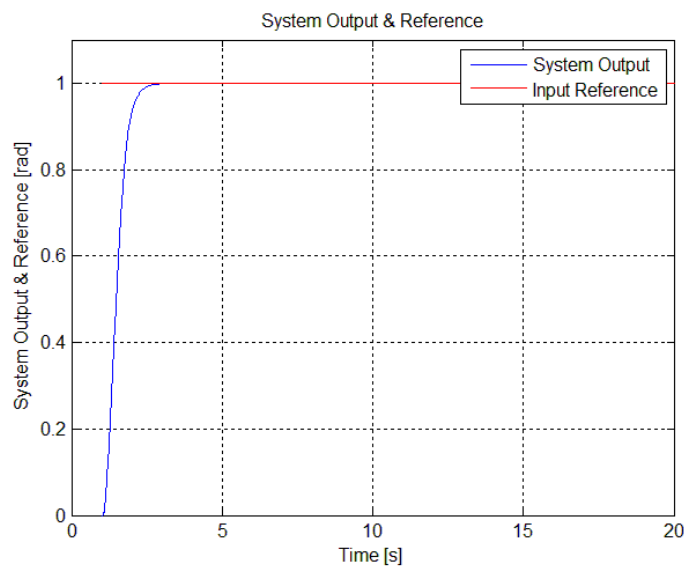


Figure 21 - Single Axis spacecraft attitude (system output) no thruster saturation

The RCAC tuning parameters for the result depicted in Figure 21 are

Performance penalty	$1 \times I_{l_z}$
Filtered Control penalty	$0 \times I_{l_z}$
Control penalty	$2 \times 10^{-7} \times I_{l_u}$
Transient Penalty	$10^{-9} \times I_{l_\theta}$
Delta theta Penalty	$0 \times I_{l_\theta}$

Table 5 - Single Axis Spacecraft RCAC tuning parameters

As the control authority provided by the thrusters diminish the system response starts oscillating as the thrusters do not have the power to bleed quickly enough the momentum generated to move the spacecraft from its initial attitude to the desired attitude. When the lack of control authority is still light, small overshoot and oscillation appear in the transitory of the system as can be seen in Figures 22 and 23.

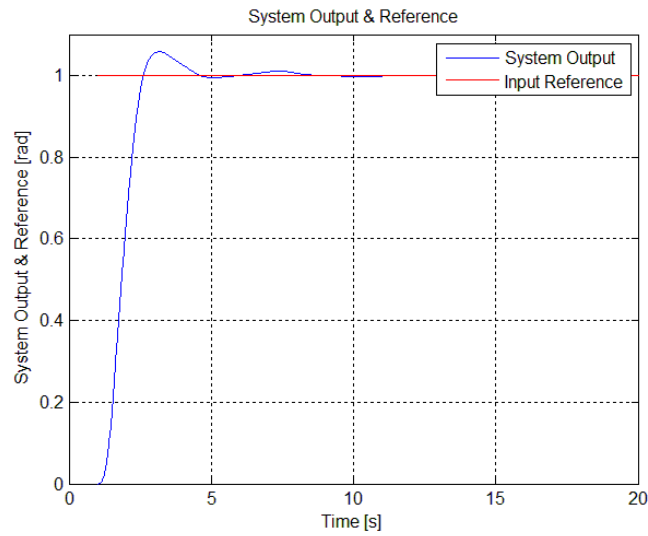


Figure 22 - Single Axis Spacecraft attitude (system output) with thruster saturation on +-5

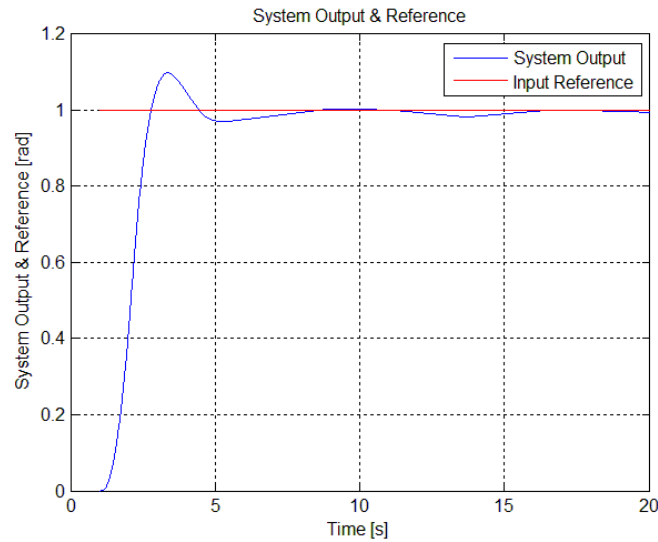


Figure 23 - Single Axis Spacecraft attitude (system output) with thruster saturation on +-2.5

3.2 – Triax System

For the Triax two types of simulations were made, one in which the RCAC has as control outputs the torques to each axis, i.e. the controller calculated directly the torques to be applied to each of the spacecraft body axes, and one in which the RCAC has as control outputs the voltage to each of the four thrusters that generate the torques to rotate the spacecraft. The general schematics for the simulation were the same and can be seen in Figure 24. The Triax model used for these simulations will be derived in chapter 4.

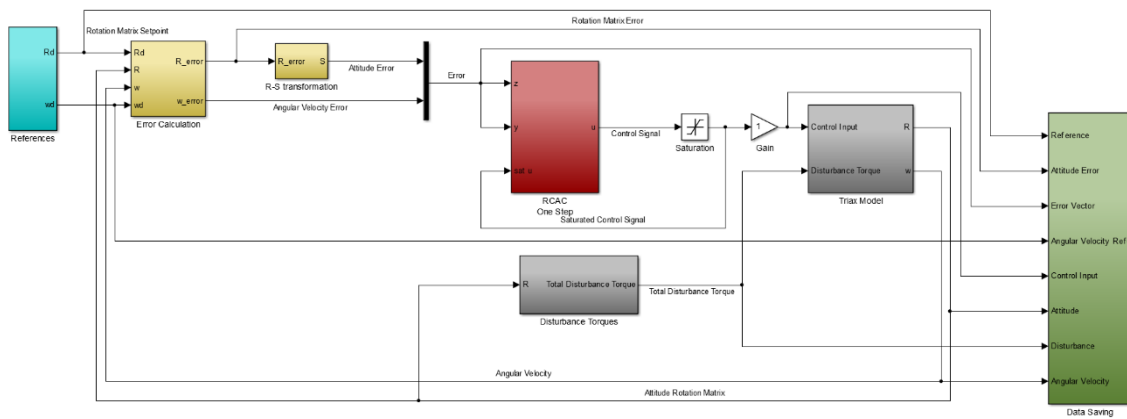


Figure 24 - Triax Simulation schematics

The main difference between the schematics used in these simulations, Figures 25 and 26, is the model of the thruster, which converts the thruster input voltages into torques that will act in each of the spacecraft's body axes. The thruster model will also be derived in chapter 4.

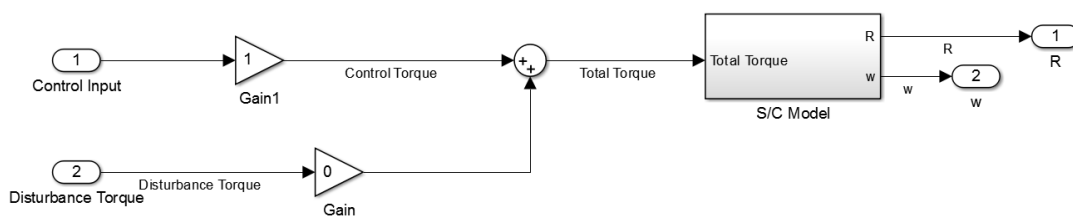


Figure 25 - Torque Simulations Triax Model

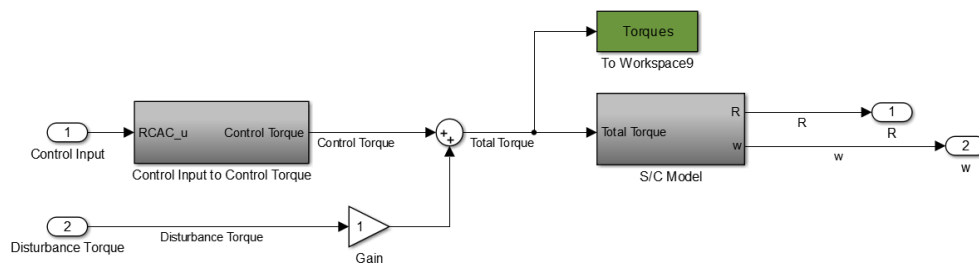


Figure 26 - Thruster Simulations Triax Model

3.2.1 – Torque Simulation

For the simulations in which the RCAC had as output the torque on each of the three axes simulations were done for several attitudes in order to ensure that the controller could operate in all directions. Since the method used to construct the DCM of the desired attitude was based on Euler axis/angle the setpoints all range from -180° to 180° . Below are Figures of some of the simulations made.

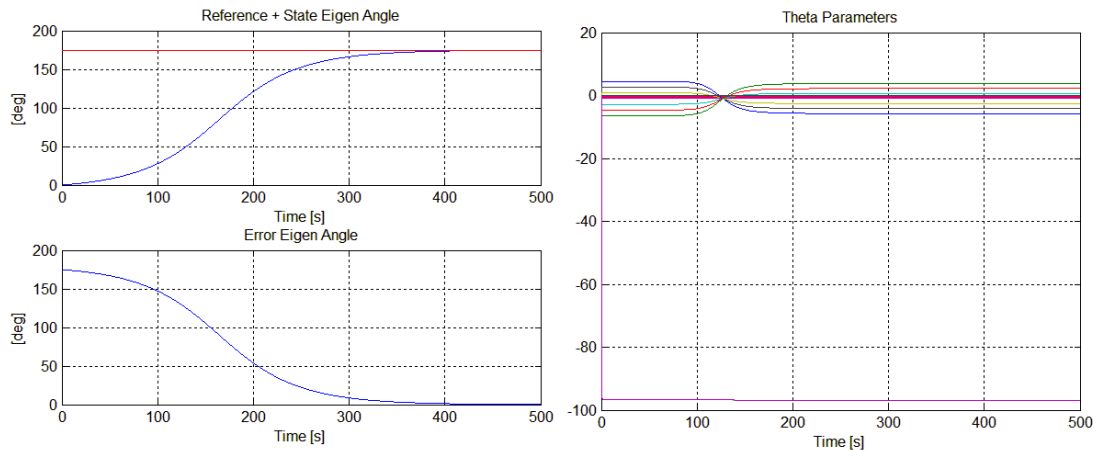


Figure 27 – Eigen Angle & Control Parameters Pitch -175 [deg]

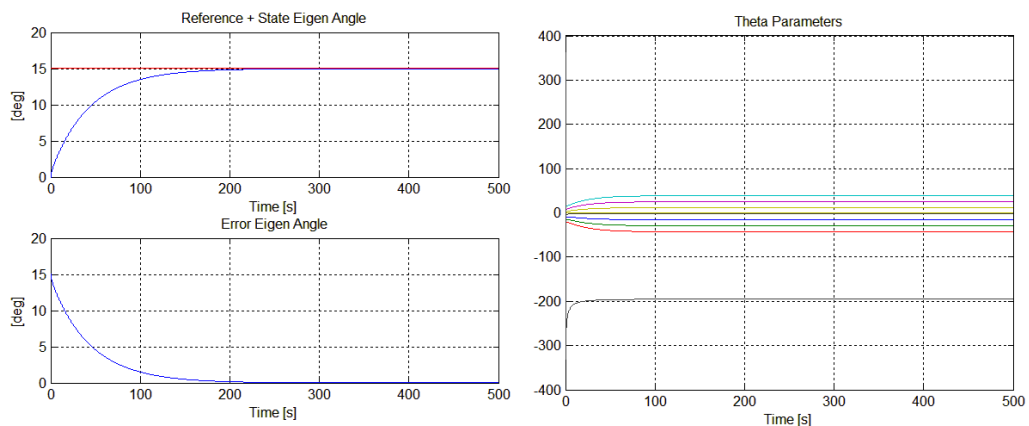


Figure 28 – Eigen Angle & Control Parameters Yaw 15 [deg]

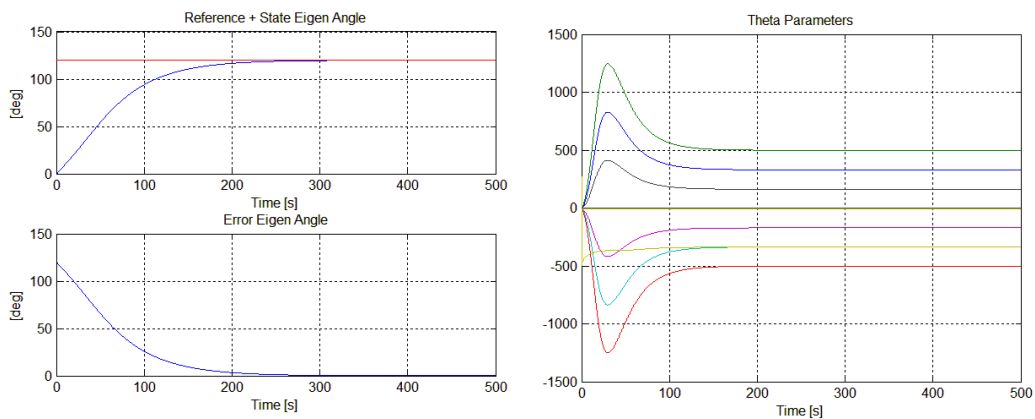


Figure 29 – Eigen Angle & Control Parameters Roll -120 [deg]

As is apparent in the above results the controller has no problem tracking setpoints on any direction. It can also be observed in the Figures that the RCAC converges fairly fast on the controller parameters.

RCAC, as expected, also adapts to changes in the setpoint.

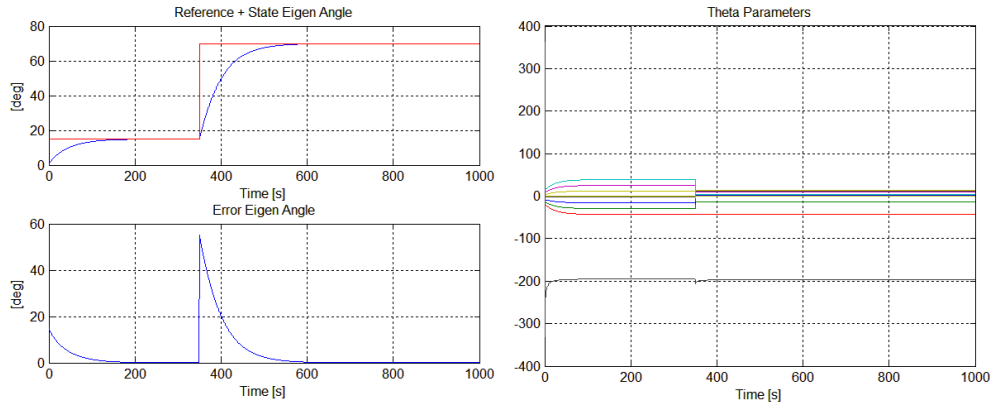


Figure 30 - Eigen Angle & Control Parameters Yaw 15-70 [deg]

even when the setpoints are given on different axes

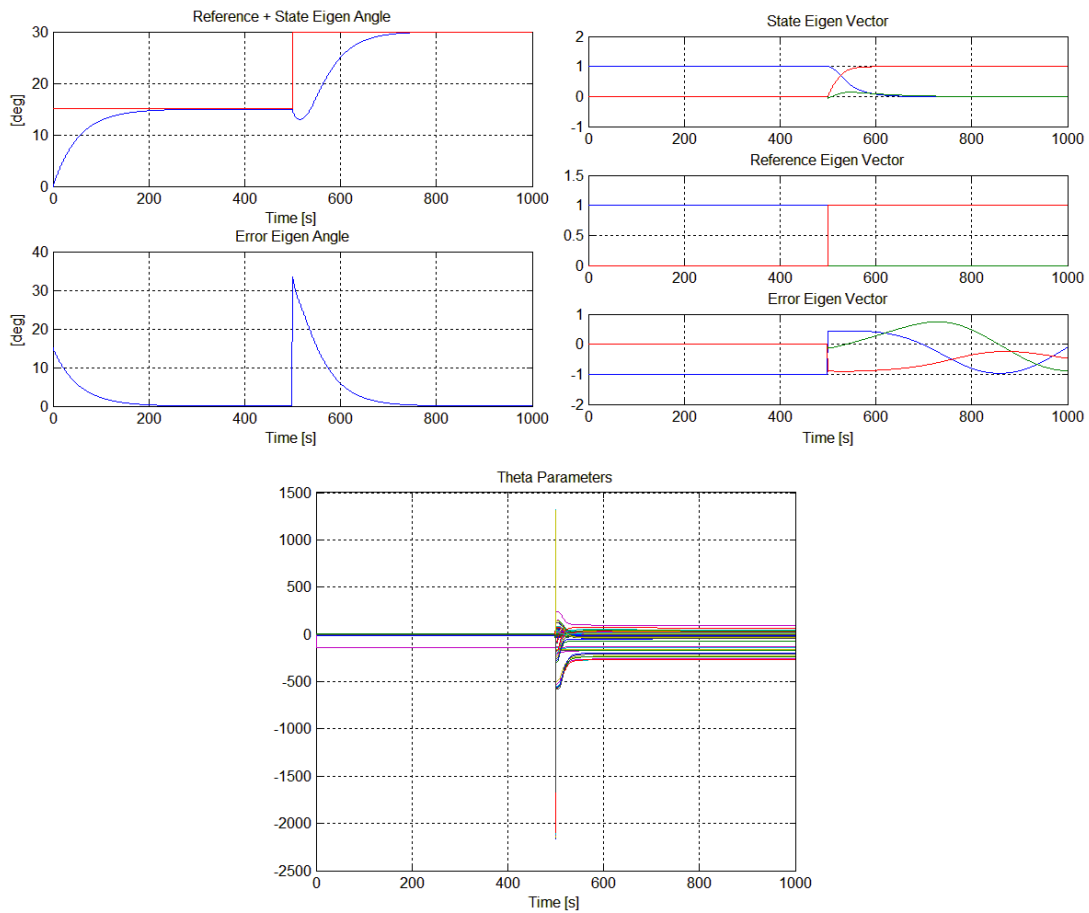


Figure 31 - Eigen Angle/Axis & Control Parameters Pitch 15 [deg] - Yaw 30 [deg]

The RCAC tuning parameters were

Performance penalty	$1 \times I_{l_z}$
Filtered Control penalty	$0 \times I_{l_z}$
Control penalty	$0 \times I_{l_u}$
Transient Penalty	$10^{-10} \times I_{l_\theta}$
Delta theta Penalty	$0 \times I_{l_\theta}$

Table 6 - Torque 3 Axis Stabilized Spacecraft RCAC tuning parameters

It is interesting to note that in this study it was found that the Markov parameter held information regarding the results of the actuator actions. The Markov parameter informed RCAC of the direction of the thrusters and thus in which axis it generated torque. The scale of the Markov parameter informed the RCAC of the order of magnitude of the torque generated by the thruster.

It was observed that when the Markov parameter was overestimated the rising time of the output increased significantly, see Figure 32, while when the Markov parameter was underestimated the output would have oscillations, see Figure 33, and become unstable and cause the simulation to crash if the difference between the real value of the Markov parameter and the estimated one were too large.

The Markov parameter gives RCAC a basic understanding of the effects of the actuators in the output of the system it is controlling.

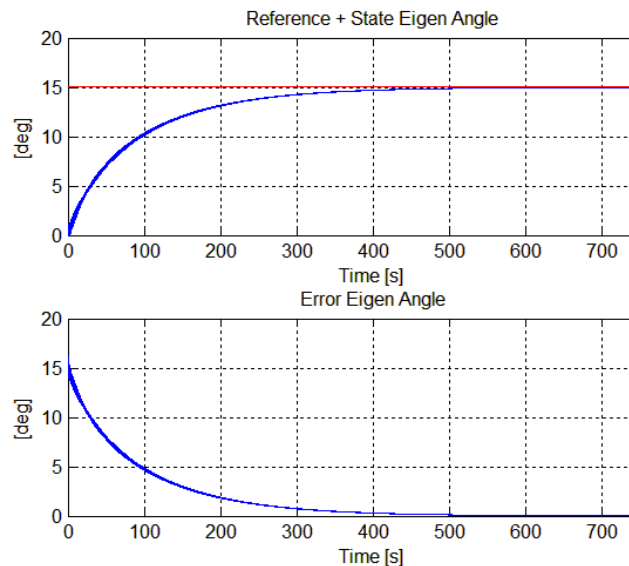


Figure 32 - Overestimated Markov parameter 100H

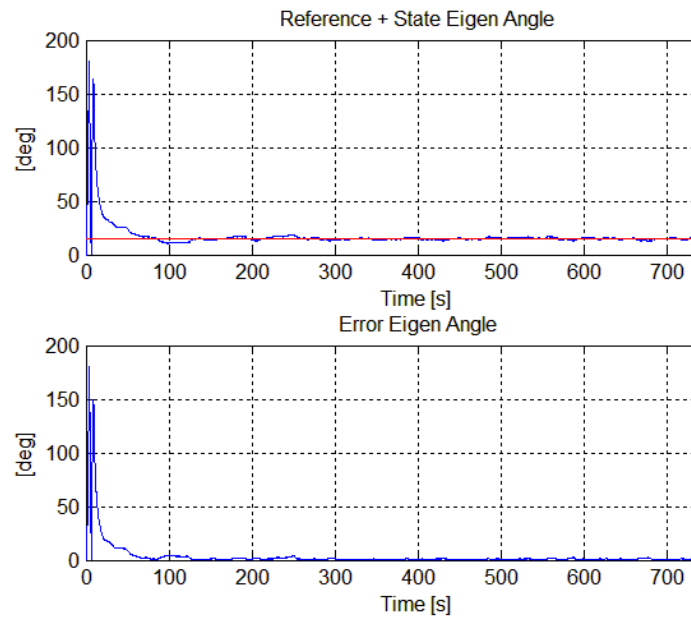


Figure 33 - Underestimated Markov parameter $10^{-5}H$

3.2.2 – Thruster Simulation

As with the simulations in which the RCAC had as output the torque to each axis, several simulations were done for a variety of attitudes to ensure that the RCAC could handle tracking setpoint on all directions.

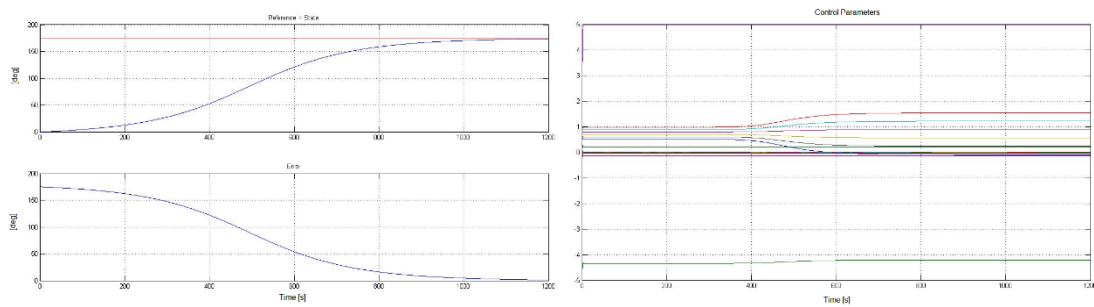


Figure 34 – Eigen Angle & Control Parameters Pitch 175 [deg]

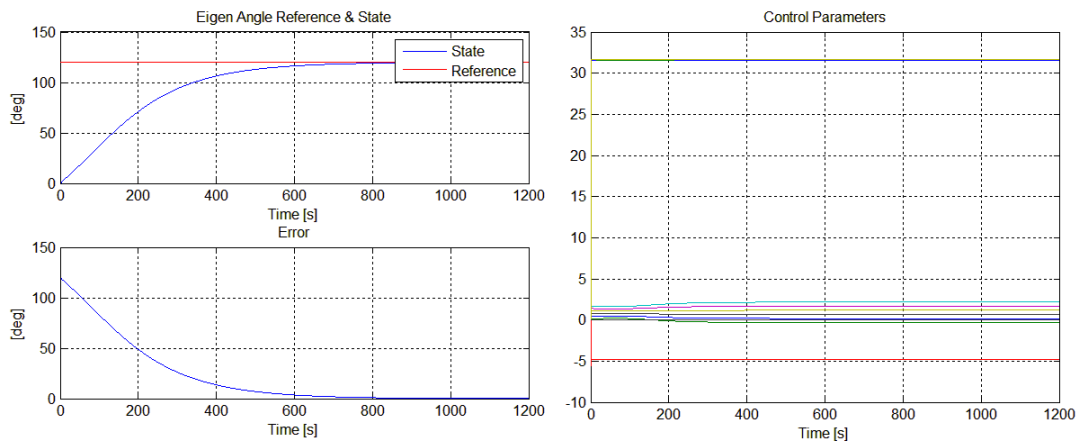


Figure 35 – Eigen Angle & Control Parameters Roll -120 [deg]

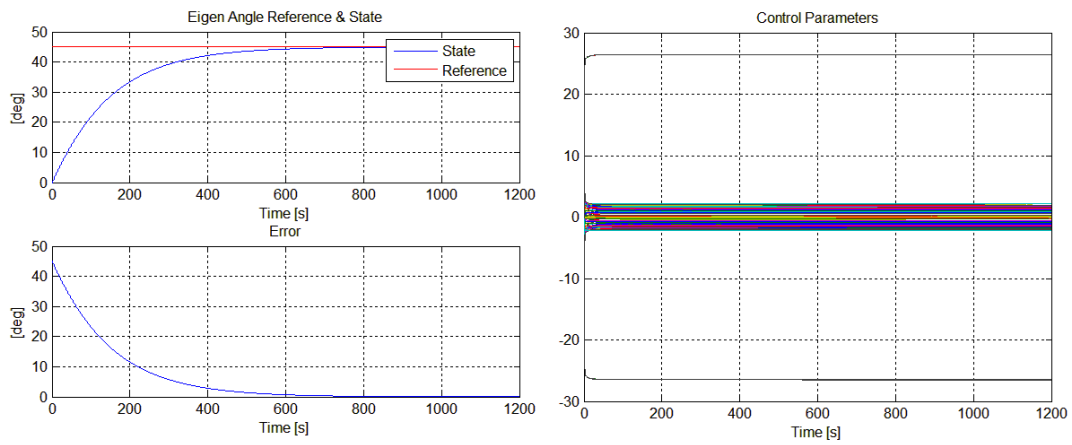


Figure 36 – Eigen Angle & Control Parameters Yaw 45 [deg]

Varying setpoints were also tested

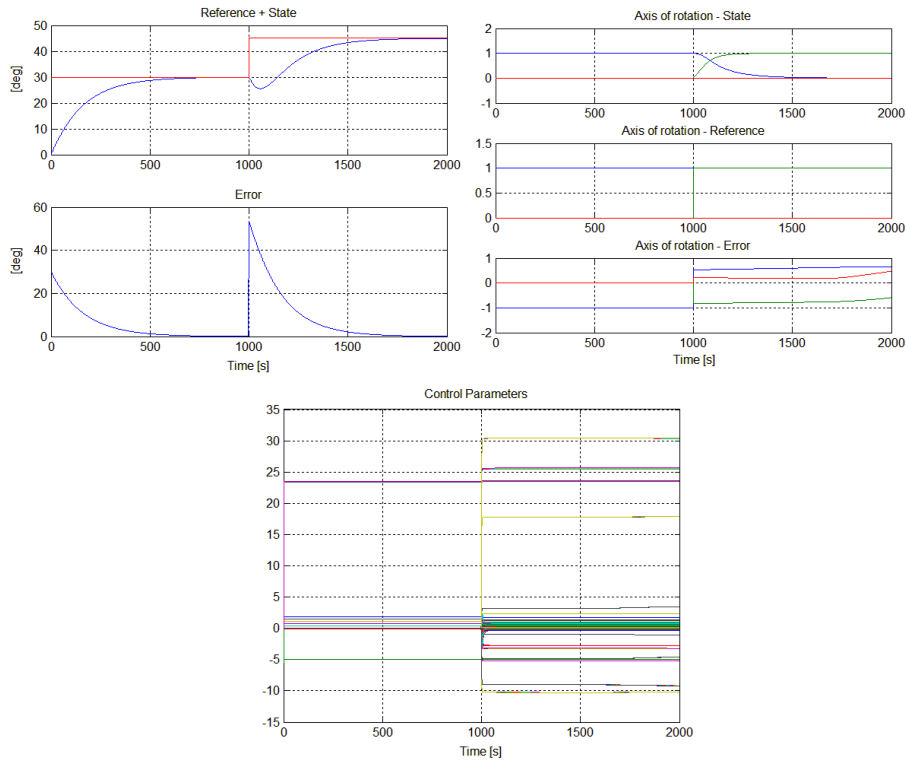


Figure 37 - Eigen Angle/Axis & Control Parameters Pitch 30 [deg] - Roll 45 [deg]

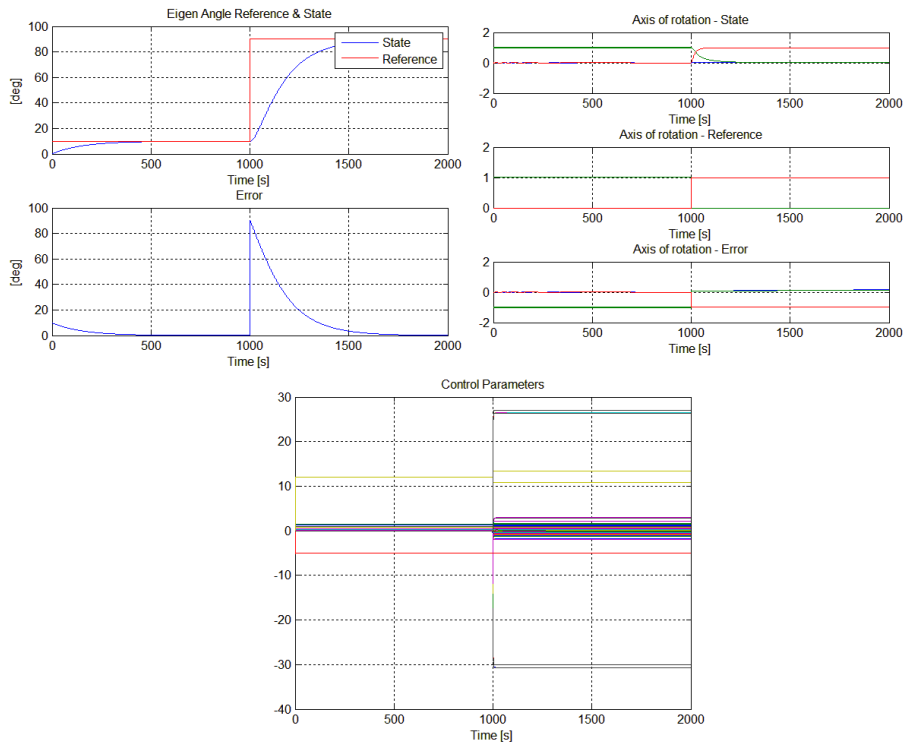


Figure 38 - Eigen Angle/Axis & Control Parameters Roll 10 [deg] - Yaw 90 [deg]

The RCAC tuning parameters were

Performance penalty	$1 \times I_{l_z}$
Filtered Control penalty	$0 \times I_{l_z}$
Control penalty	$10^{-6} \times I_{l_u}$
Transient Penalty	$10^{-9} \times I_{l_\theta}$
Delta theta Penalty	$0 \times I_{l_\theta}$

Table 7 - Thruster 3 Axis Stabilized Spacecraft RCAC tuning parameters

Chapter 4 - The Triax

4.1 – Description

The Triax is a testbed equipment designed to study rotational dynamics and control problems. It is based on a spherical air bearing which floats an 11 inch diameter aluminum sphere. The air is supplied by a compressor and exits at the cup on which the sphere is resting. The air then lifts the sphere drastically reducing the friction between the cup and the sphere. This allows the sphere to rotate freely and simulates a weightless environment.



Figure 39 – Triax

A shaft passes through the center of and is attached the sphere and connects both sides of the Triax both physically and electronically as this shaft is hollow and inside it wires are passed connecting both sides electrically and electronically. On each side of the Triax one circular plate and one square plate are mounted. On one side of the Triax, mounted on these plates are the batteries and electronics, and on the other side are the drivers and actuators. On the square plates on both sides of the Triax are also located weights for balancing the Triax.

Lead-acid batteries are used to power the actuators and lithium polymer batteries are used to power the electronics.

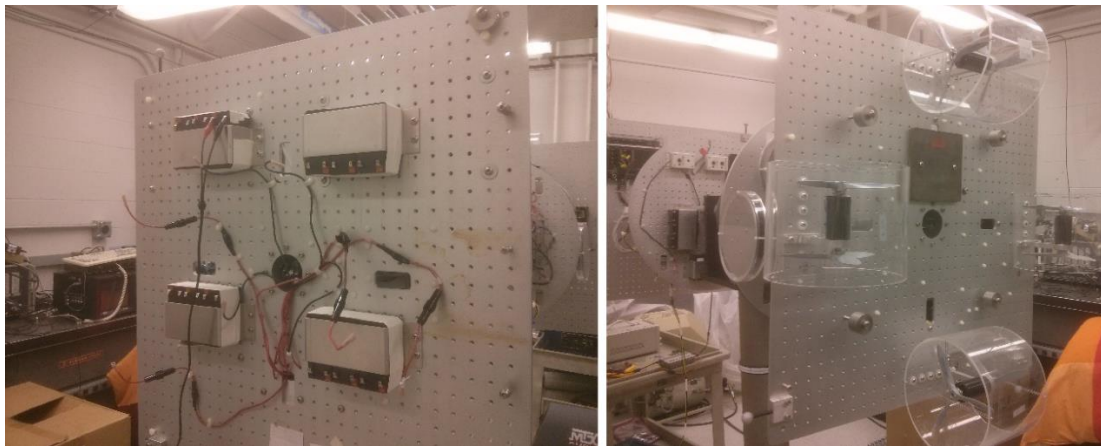


Figure 40 - Triax Batteries (left) and Actuators (right)

The air bearing allows for unrestricted motion of yaw and roll of the Triax, however due to the pedestal the pitch motion is restricted to $\pm 45^\circ$.

For his thesis the attitude sensor used is an Optitrack, a system that uses 6 cameras to capture the position of 6 markers placed in the Triax's physical frame and calculate its attitude with respect to the laboratory's frame of reference. This setup is similar to a spacecraft using star sensors. For the purpose of the experiment the laboratory, and thus the camera system, is considered an inertial frame of reference to which the attitude of the Triax is measured.

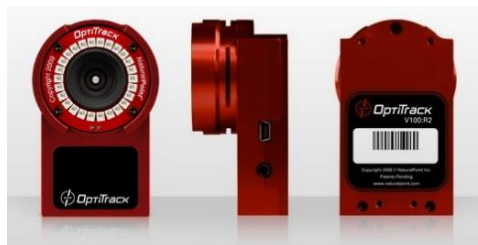


Figure 41 - Optitrack Cameras

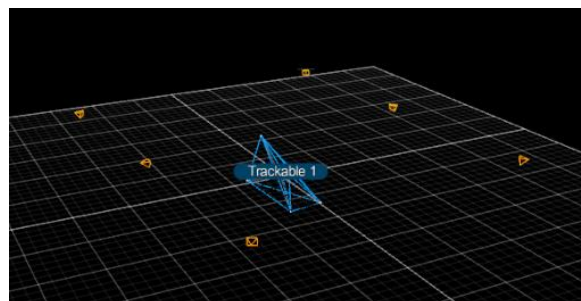


Figure 42 - Optitrack Camera Setup and Markers Trackable (Triax)

The Triax also possesses 3 MEMS gyroscopes which measure angular velocities in all 3 axis.

4.2 – Mathematical Modeling

4.2.1 – Rigid Body Dynamics and Kinematics

The kinematics of DCM's, the chosen attitude representation, is well known [20]

$$\dot{R} = R[w^\times] \quad (4.1)$$

where $[w^\times]$ is the skew-symmetric operator of \vec{w} which is the angular velocity and R is the rotation matrix from body to inertial frame.

The dynamics of the Triax are given by Euler's equations of Motion [2].

$$\dot{\vec{w}} = [I]^{-1}(-[w^\times][I]\vec{w}) + [I]^{-1}T_E \quad (4.2)$$

where $[I]$ is the inertia tensor.

Equation (4.2) stems from the conservation of angular momentum law, which states that

$$\dot{H} = \frac{d([I]\vec{w})}{dt} = \sum T \quad (4.3)$$

where

$$H = [I]\vec{w} \quad (4.4)$$

In an inertial frame of reference we can make use of a well-known vector operator [13]

$$\left. \frac{d\vec{A}}{dt} \right|_I = \left. \frac{d\vec{A}}{dt} \right|_B + \vec{w} \times \vec{A} \quad (4.5)$$

this states that the rate of change of vector \vec{A} observed from an inertial frame of reference equals its rate of change observed in the body frame of reference, with angular velocity w of the body frame with respect to the inertial reference frame, plus the vector product $\vec{w} \times \vec{A}$.

The rate of change of the angular momentum is then

$$\left. \frac{d\vec{H}}{dt} \right|_I = \left. \frac{d\vec{H}}{dt} \right|_B + \vec{w} \times \vec{H} = T_E \quad (4.6)$$

where $T_E = \sum T$ is the total torque and

$$\left. \frac{d\vec{H}}{dt} \right|_B = [I]\dot{\vec{w}} + [I]\dot{\vec{w}} \quad (4.7)$$

since in the body frame the inertia of the body is constant, $[\dot{I}] = 0$, thus

$$\left. \frac{d\vec{H}}{dt} \right|_B = [I]\dot{\vec{w}} \quad (4.8)$$

equation (4.6) becomes

$$[I]\dot{\vec{w}} + \vec{w} \times \vec{H} = T_E \quad (4.9)$$

substituting equation (4.4) in equation (4.9) we have

$$[I]\dot{\vec{w}} = T_E - \vec{w} \times [I]\vec{w} \quad (4.10)$$

$$\dot{\vec{w}} = [I]^{-1}T_E - [I]^{-1}(\vec{w} \times [I]\vec{w}) \quad (4.11)$$

By transforming the cross product on the second term of equation (4.11) into a skew matrix we have

$$\dot{\vec{w}} = -[I]^{-1}([w^\times][I]\vec{w}) + [I]^{-1}T_E \quad (4.12)$$

The only part of the Triax not modeled with these equations are the thrusters. The thrusters models are given in [12].

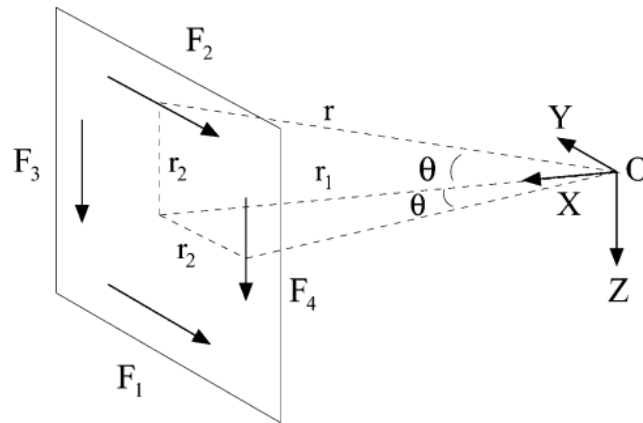


Figure 43 - Forces due to the thrusters on the Triax

The force generated by each of the thrusters given the input voltage to the motors has been experimentally found to be

$$F_{1,2}(V) = 0.12V^3 - 0.004V^2 + 0.26V + 0.01 \quad (4.13)$$

$$F_{3,4}(V) = 0.1V^3 - 0.0023V^2 + 0.28V - 0.01 \quad (4.14)$$

Therefore as can be seen in Figure 43, which shows a simplified schematic of the Triax physical construction, the torques generated by the thrusters are

$$T_{yaw} = -r_1(F_1(V_1) + F_2(V_2)) \quad (4.15)$$

$$T_{pitch} = -r_1(F_3(V_3) + F_4(V_4)) \quad (4.16)$$

$$T_{roll} = r_2[(F_1(V_1) - F_2(V_2)) + (F_3(V_3) - F_4(V_4))] \quad (4.17)$$

where r_1 and r_2 are shown in Figure 43, and are equal to 0.945 [m] and 0.33[m], respectively.

4.2.2 – Inertia Tensor Estimate

An estimate of the inertia tensor of the Triax is necessary for simulation purposes. Although RCAC does not require an accurate model of the system it is controlling the Triax inertia tensor was estimated by modeling it as a solid cylinder of mass 100 kg, length of 2 m and diameter of 0.5 m.

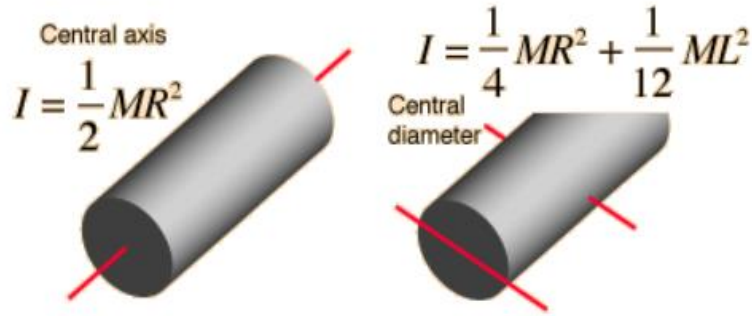


Figure 44 - Solid Cylinder Inertia

These values give an inertia tensor equal to

$$I = \begin{bmatrix} 39.58 & 0 & 0 \\ 0 & 12.5 & 0 \\ 0 & 0 & 39.58 \end{bmatrix} [kg.m^2]$$

4.2.3 – Inertia Tensor Calculation

It is also possible to calculate the inertia tensor of the Triax from it's first Markov parameter.

The first Markov parameter relating inputs to angular velocity is given by [10]

$$\mathcal{H}_1 \triangleq E_1 B = [hI^{-1}B] \quad (4.18)$$

thus if B is invertible

$$I = \left[\frac{1}{h} \mathcal{H}_1 B^{-1} \right]^{-1} \quad (4.19)$$

where

h is the sample time of the system

B is the actuator matrix of the state space representation of the system

Since the B matrix for the Triax is 3×4 , therefore not square, the following modifications are necessary

$$\mathcal{H}_1 B^T = [hI^{-1}BB^T] \quad (4.20)$$

$$I \approx \left[\frac{1}{h} \mathcal{H}_1 B^T (BB^T)^{-1} \right]^{-1} \quad (4.21)$$

An important issue here is that the B matrix is entirely dependent on the thruster model which is nonlinear it is necessary to linearize it before building the B matrix.

In order to linearize the thruster model MatLab's Curve Fitting Toolbox was used to interpolate a linear function that best fit the thruster in the input voltage interval, -10 to +10 [V]. The linearized models for the thrusters were

$$F_{1,2} = 8.12V \quad (4.22)$$

$$F_{3,4} = 5.8025V \quad (4.23)$$

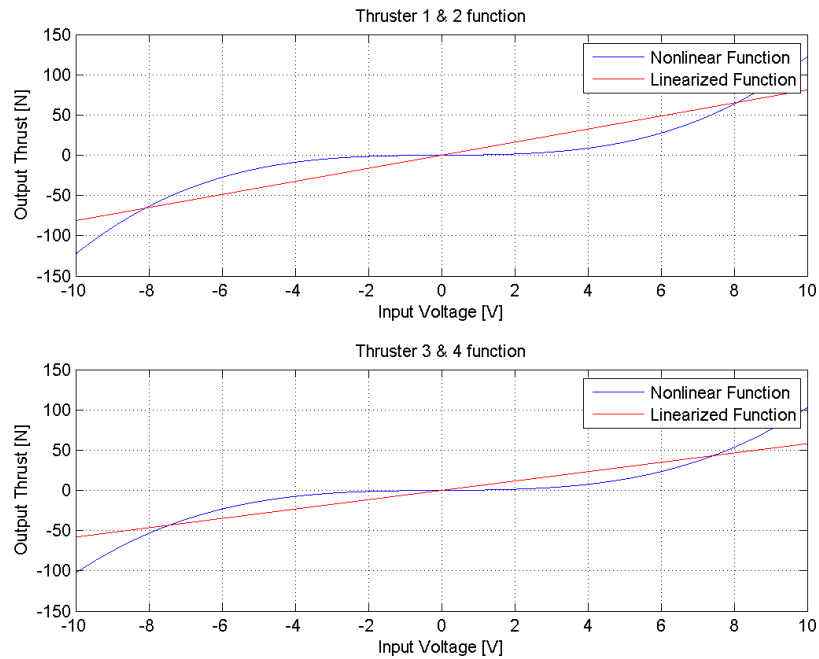


Figure 45 - Thruster Curves, Linearized and Nonlinear

This gives as B matrix

$$B = \begin{bmatrix} 0 & 0 & -5.4834 & -5.4834 \\ -7.6734 & -7.6734 & 0 & 0 \\ 7.6734 & -7.6734 & 5.4834 & -5.4834 \end{bmatrix}$$

Which in turn yield an approximate inertia tensor of

$$J = \begin{bmatrix} 365.6 & 516.5 & -875.4 \\ -511.6 & -721.6 & 1223.7 \\ 0 & 2651.2 & -2639.9 \end{bmatrix}$$

4.3 – Simulink Model

The Simulink model of the Triax used in the simulations of the RCAC was a block containing

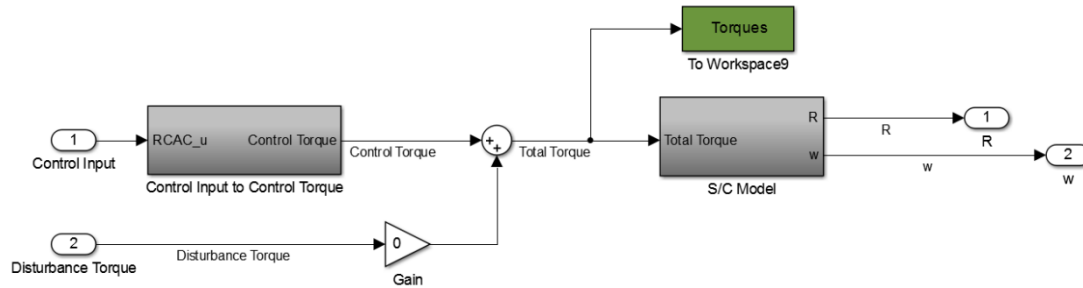


Figure 46 - Triax Model

The S/C Model block contained the kinematic (lower part of the block) and dynamic equations (upper part of the block).

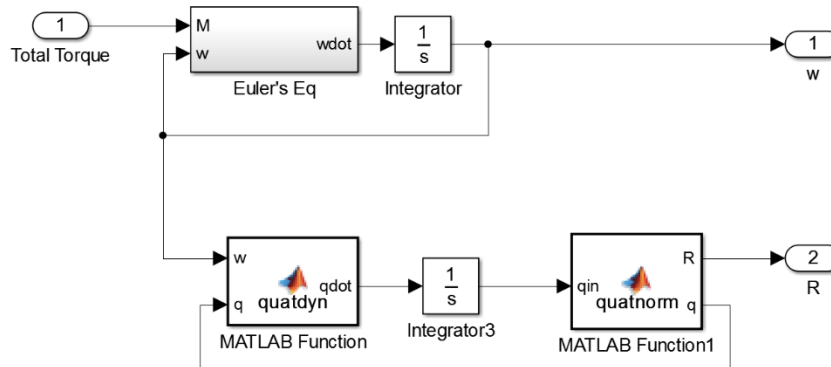


Figure 47 - Spacecraft Model

The Control Input to Control Torque is a block that takes the control inputs generated by RCAC and transforms it into thruster input voltage as shown in section 4.2

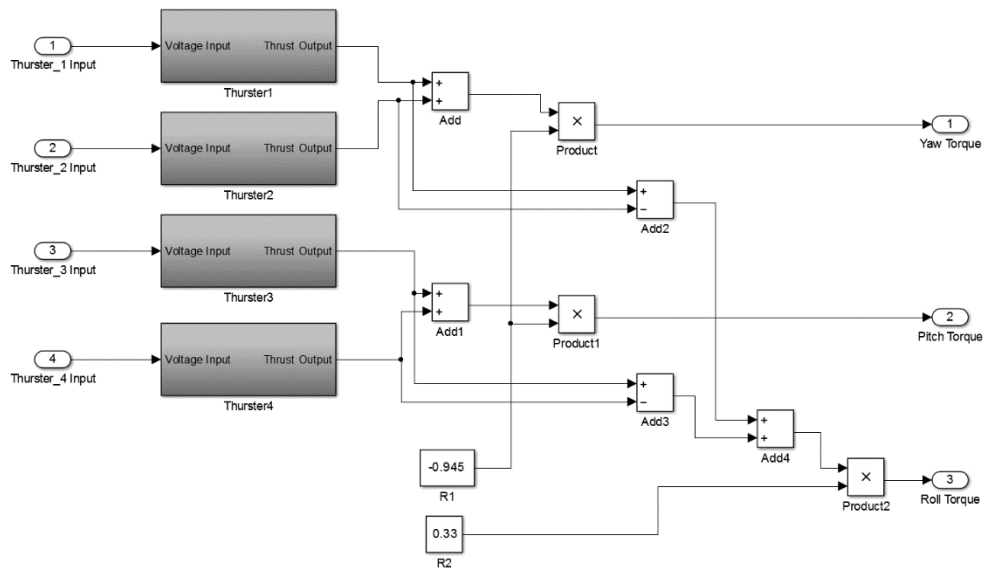


Figure 48 – Thruster to Torque Model

Chapter 5 – Hardware Testing

To check the Triax system and compare with RCAC a PID control law as defined in section 1.4.1 was designed. In order for the control law to be applicable and usable, the input command was not given as a step but as a ramp from zero to the desired eigen-angle.

The Simulink scheme used for this hardware testing was

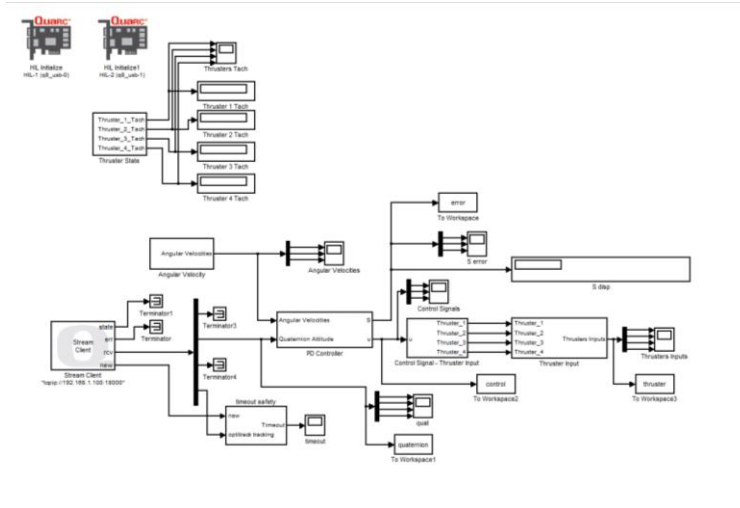


Figure 49 - Main PID scheme

and the controller block was

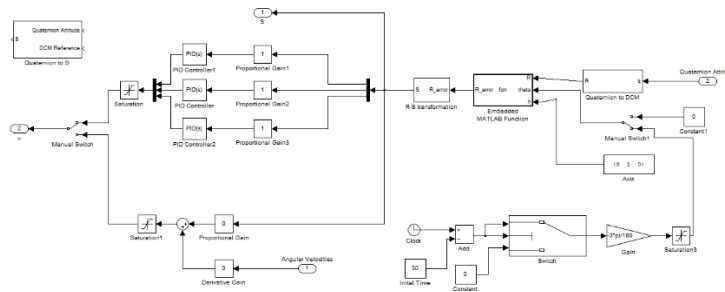


Figure 50 - PID Controller scheme

The PID architecture chosen was parallel, and the tuning parameters were:

Proportional Gain	5
Derivative Gain	3.5
Integral Gain	0.5

Table 8 - PID Control Law Gains

The first and last 3-8 seconds of the plot should be disregarded as the controller was not yet operating. Due to limitations in data acquisition in the Simulink scheme in which the controller was running it was necessary to put the data acquisition blocks in the OptiTrack Simulink scheme which runs parallel to the controller scheme, but are not initialized simultaneously, thus the synchrony problem with the data.

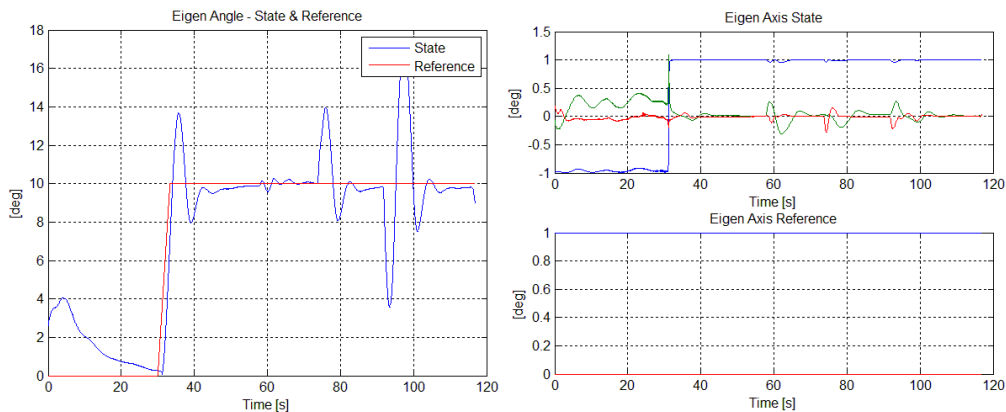


Figure 51 - Eigen Angle & Axis Pitch 10 [deg]

In Figure 51 it can be seen that the controller caused an appreciable overshoot during the transient, this is due to the necessity of an aggressive controller. For less aggressive controllers the Triax did not have a good stabilization effect and after reaching the desired setpoint the system oscillated lightly around it. It should also be noted that the Triax was powered by power cables and not batteries in this test, therefore a torque existed in the pitch axis.

When a test with the Triax powered by batteries was done with batteries instead of power cables it was verified that the batteries could not fully power the thrusters. This was most likely due to the batteries been old and no longer able to hold as much charge.

The two large oscillations present in the graph after the system settled were disturbances generated manually in order to test the controller's disturbance rejection. The first disturbance was in the roll axis and the second was in the yaw axis.

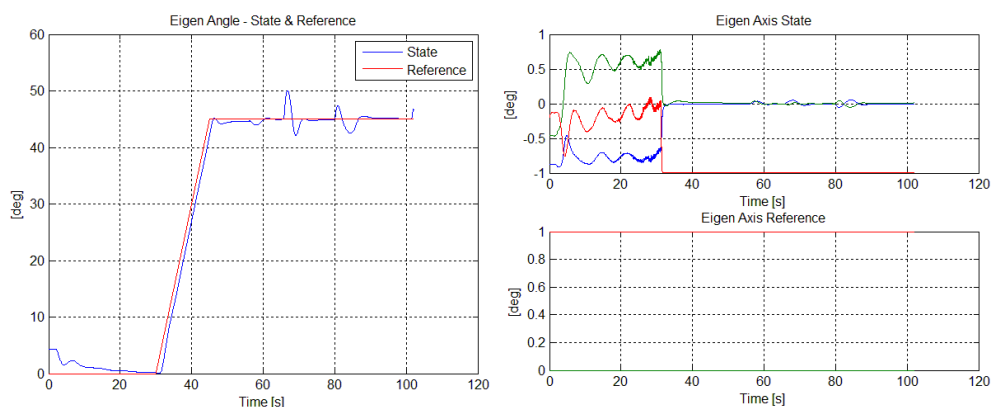


Figure 52 - Eigen Angle & Axis Roll -45 [deg]

In Figure 52 is the result of the roll test. Since the roll axis was affected to a lesser degree by the power cables used to power the Triax and due to the fact that for roll the controller has four instead of two thrusters to work with, thus has double the torque to make adjustment, the overshoot was minimal even considering the larger desired angle.

As with Figure 51, in Figure 52 the oscillations that appear after the system has stabilized in the desired attitude were disturbances generated manually as a test of disturbance rejection. The first disturbance was in the pitch axis and the second was in the yaw axis.

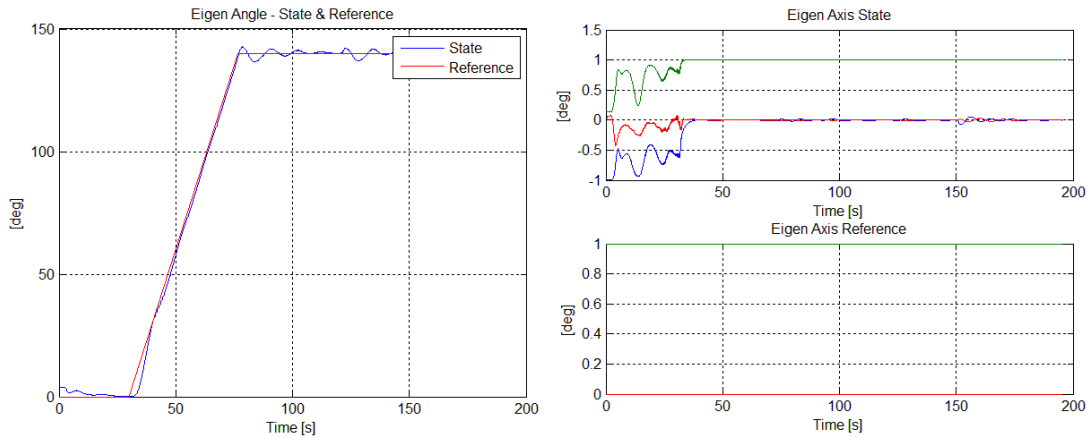


Figure 53 - Eigen Angle & Axis Yaw 140 [deg]

In Figure 53 we observe the result of the yaw test. Yaw was the least affected by the power cables and that translated in a far smaller overshoot then the one present on the pitch test results, even with also two thrusters as actuators.

As was the case in the pitch and roll tests the oscillations that appear after the system has been stabilized are disturbances generated manually to test the disturbance rejection. The first disturbance was in the pitch axis and the second was in the roll axis.

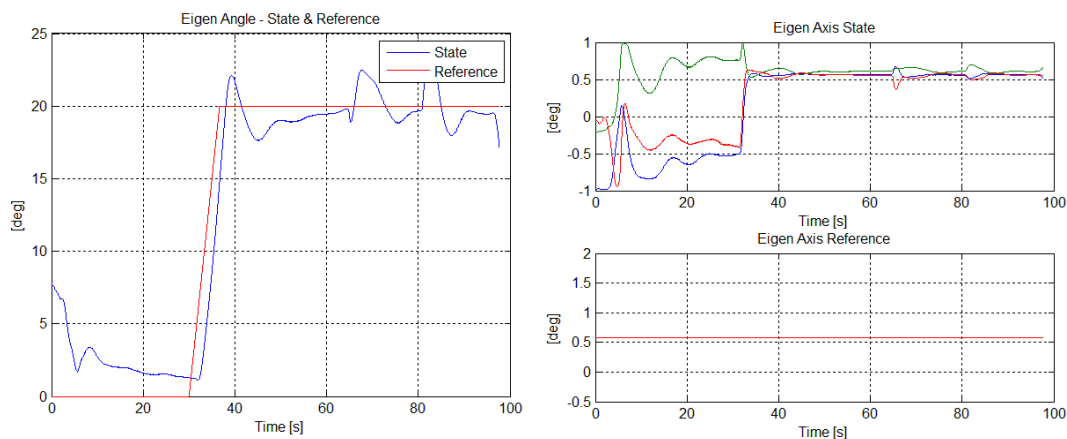


Figure 54 - Eigen Angle & Axis [1,1,1] 20 [deg]

The last test was to rotate the Triax 20 deg about all three axis, meaning around vector [1,1,1]. As expected, since this command required the Triax to lift the side in which the power cables were hanging, there was a significant overshoot and the system took longer to settle.

As was the case of all other tests the oscillations that appear after the system has been stabilized are disturbances generated manually to test the disturbance rejection. The first disturbance was in the roll axis and the second was in the yaw axis.

Attitude stabilization for initial condition with eigen-angles between -40° to 40° in under 10 seconds. For initial condition with eigen-angles bigger the $\pm 40^\circ$ generated instability, the system started oscillating out of control. This instability is due to the breakdown of the simplified model given in equation (1.26), the error eigen-angle was too large and the simplification upon which the controller was based is no longer valid.

Tracking of a setpoint with a ramp inclination higher then $3^\circ/s$ had overshoots around 10% and longer settling time. This is due to the small control authority given by the thrusters.

Chapter 6 – Conclusion & Future Works

In chapter 1 the motivation for this work was presented, the basics of spacecraft attitude determination and control were described. A brief description of attitude sensors for spacecrafts were presented. Simple control laws were derived.

In chapter 2 the RCAC algorithm was presented along with the necessary modifications for its application in spacecraft attitude control.

In chapter 3 a variety of simulations were done to both show the effectiveness of RCAC and to better understand the effects of the tuning parameters. Still in chapter 3 simulations were done for a 3 axis stabilized spacecraft to gauge RCAC's capabilities when dealing with spacecraft attitude control.

In chapter 4 the Triax testbed was described along with the attitude sensors used. Its mathematical model was derived and the necessary parameters were estimated.

In chapter 5 a simple PID control law was designed to control the Triax, and the results of this control law were presented. The disturbance rejection capabilities of the control law was also tested.

In conclusion RCAC was tested in various systems, such as Aircraft pitch, DC motor control and spacecraft attitude control, successfully.

RCAC presented itself as a viable control strategy for systems in which knowledge of specifics of systems dynamics are difficult to know.

Simulations indicate that RCAC would be a good control strategy for spacecraft attitude control. In special cases where the spacecraft configuration suffers many changes during its lifetime. Unfortunately due to time constraints and hardware issues it was not possible to tune RCAC on the Triax. Such work is left for future research.

Other possible venues for future research could be the tuning of RCAC for the Triax rate control, detumbling and spacecraft lifecycle changes such as a change from a spin stabilized to a 3 axis stabilized spacecraft. This is a common transition in the beginning stages of a spacecraft's lifetime, more specifically in the orbit maneuvering stage. [13]

References

- [1] – Malcom D. Shuster. A Survey of Attitude Representations. The Journal of the Astronautical Sciences, vol. 41, No. 4, 1993.
- [2] – James R. Wertz. Spacecraft Attitude determination and Control. Astrophysics and Space Science Library, 2002.
- [3] – Jack B. Kuipers. Quaternions & Rotation Sequences - A Primer with Applications to Orbits, Aerospace, and Virtual Reality. 1999.
- [4] – Jan Roháč. CTU Aircraft and Spacecraft Instrumentation lecture slides.
- [5] – Attitude Representation Methods Quick References Document. Spacecraft Dynamics and Controls Class. University of Wuerzburg
- [6] – Monson H Hayes. Statistical Digital Signal Processing. 1996
- [7] – Ali H Sayed. Fundamentals of Adaptive Filtering. 2003
- [8] – Cruz & Bernstein. The Retrospective Cost Adaptive Control: Algorithm Description and Pseudocode.
- [9] – Cruz. One-Step RCAC Algorithm Description. 2014
- [10] – Cruz & Bernstein. Adaptive Spacecraft Attitude Control with Reaction Wheel Actuation. 2013
- [11] – Cruz, D'Amato & Bernstein. Retrospective Cost Adaptive Control of Spacecraft Attitude. 2012. AIAA Guidance, Navigation, and Control Conference.
- [12] – Chaturvedi, Bernstein, Ahmed, Bacconi & McClamroch. Globally Convergent Adaptive Tracking of Angular Velocity and Inertia Identification for a 3-DOF Rigid Body. IEEE Transactions on Control Systems Technology, vol. 14, n°. 5, 2006
- [13] – Sidi M J. Spacecraft Dynamics and Control A Practical Engineering Approach. 1997
- [14] – Sanyal, Fosbury, Chaturvedi & Bernstein. Inertia-Free Spacecraft Attitude Tracking with Disturbance Rejection and Almost Global Stabilization. Journal of Guidance, Control, and Dynamics, vol. 3, n°4, July-August 2009
- [15] – Sobolic. Agile Flight Control Techniques for a Fixed-Wing Aircraft. 2006
- [16] – Venugopal & Bernstein. Adaptive disturbance rejection using armarkov/toeplitz models. Control Systems Technology, IEEE Transactions on, vol. 8, no. 2, pp. 257–269, 2000.
- [17] – <http://ctms.engin.umich.edu/CTMS/index.php?aux=Home>, 24/05/14
- [18] – Fitzgerald, Kingsley & Umans. Electric Machinery - 6th Edition.
- [19] – Toliyat & Kliman. Handbook of Electric Motors. 2004
- [20] – Kane, Likins & Levinson. Spacecraft Dynamics.
- [21] – Julius O. Smith. Introduction to Digital Filters with Audio Applications. W3K Publishing. <http://books.w3k.org/>. 2007. ISBN 978-0-9745607-1-7.