DEVELOPMENT OF A GENERIC FUZZY LOGIC MIMO CONTROLLER FOR SATELLITE ATTITUDE CONTROL

By

KEVIN J. WALCHKO

A THESIS PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

UNIVERSITY OF FLORIDA

1999

ACKNOWLEDGMENTS

I would like to thank Dr. Paul Mason for his help and support during this work. His knowledge and understanding of controls and fuzzy logic have aided my work greatly. Also his willingness to make time for me whenever I had a question and concern for me were greatly appreciated. He is truly a great friend.

I would also like to thank Nina C. Massie for her understanding during this time. She was always willing to make sacrifices, so I could have enough time to do my work. She always showed enthusiasm for my work even though she did not totally understand what I was doing. She is the best friend and partner in life that I could ever have.

I would like to thank my parents, Jack and Barbara Walchko, who always had faith in me. They knew I could do anything and always supported me in whatever endeavor I chose. I owe them more than these few sentences could ever convey.

Finally I would like to thank the Horde (Conan, Maxamillion, Elvis, Jonsey, and Tabitha), who were always there for me with their unconditional love. They always cheered me up when I was sad or helped me to feel better when I was sick. Their kitty hijinks were always the cure for what ailed me.

TABLE OF CONTENTS

Abstract of Thesis Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
requirements for the Degree of Master of Science

DEVELOPMENT OF A GENERIC FUZZY LOGIC MIMO CONTROLLER FOR SATELLITE ATTITUDE CONTROL

By

Kevin J. Walchko

May 1999

Chair Person: Dr. Paul Mason
Major Department: Mechanical Engineer

This work describes the development and implementation of a generic hybrid MIMO fuzzy logic controller for NASA satellites. Due to its generic nature, this control scheme can be adapted to a variety of existing and future satellites with minimal or no effort. The hybrid structure of the controller takes advantage of classical proportional-integral-derivative control logic while maintaining a significant degree of robustness, performance and portability. When the fuzzy controller was compared to the current Liapunov controller for the SAMPEX satellite, the fuzzy provided much better performance in pointing and complex tracking. Furthermore, the fuzzy controller out performed the Liapunov scheme in the presence of noise and disturbance rejection. Finally, the fuzzy controller was more portable than the Liapunov controller.

CHAPTER 1
INTRODUCTION AND BACKGROUND

Introduction

Why a Generic Controller?

Many industries are faced with the financial and production burden of producing complex multi-functional systems, which require customized controller. The problem of designing multiple control laws for systems with the same structure or range of dynamics is the primary motivation for the development of generic control schemes. These schemes should produce a reasonable response when applied to a variety of systems. Later, only minimal modifications may be required to fine-tune the system. Classical techniques, on the other hand, do not afford this advantage since they are model dependent. This results in the need for a complete redesign to implement the controller on a new plant.

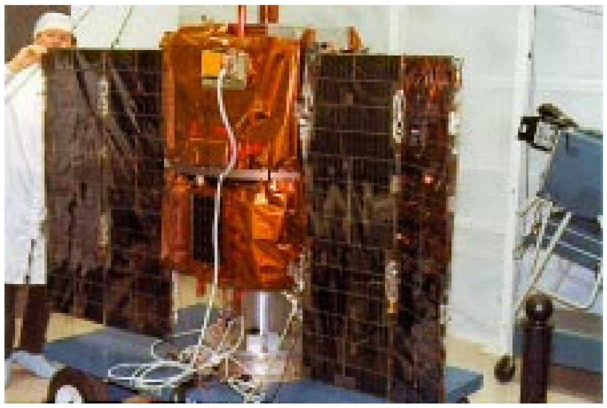
This work deals with the development of intelligent generic controllers that utilize heuristic knowledge as well as any mathematical description of the system to provide a reasonable control response. The proposed controller combines the proportional-derivative (PD) control with fuzzy logic to create a generic scheme, which can be integrated into a variety of existing systems with minimal effort.

Emphasis / Direction of this Research

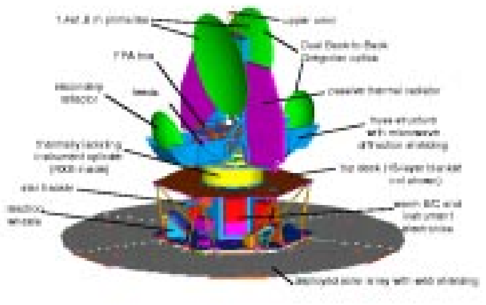
The initial focus of this work is the design of a generic controller, which will produce a reasonable, realizable response for simple linear systems. The source of this idea comes from the need for open architecture manufacturing systems (proposed by the University of Michigan and funded by NIST). Currently, all manufacturing architecture is proprietary, which results in higher costs and longer lead-time in retooling for new products. The University of Michigan (UM) plans to set a standard for which all machine tools are controlled. They are planning to use Linux as the operating system and design a communication standard to talk to a CNC machine, lathe, hexapod, etc. This universal standard and open architecture is intended to reduce cost and time for upgrades and development. One of the technologies called for by UM is a generic controller that can one day run a conveyor belt and the next operate a drill press. This is where conventional control fails, and fuzzy logic succeeds. Thus with generic controllers and a standard for communicating to machine tools, factories will be better able to go from manufacturing toilet paper one day to bombers the next.

Recently, Mason [1] successfully applied fuzzy logic to the satellite attitude problem. Today, NASA is of the belief: cheaper, faster, and smarter. Given the capability to reduce the design and tuning time of current control schemes, NASA could reduce the cost of developing and maintaining a satellites with no or minimal change.

The controller in this work will be implemented on several systems. Specifically, this work uses the SAMPEX satellite (shown below) to illustrate the robustness of the controller. Next, it is applied to the MAP satellite to demonstrate the generic capabilities.



(a)


upper omni
Dual Back-to-Back Gregorian optics
FPA box
secondary reflector
feeds
passive thermal radiator
top deck
deployed solar array with web shielding

(b)

Figure 1-1. The (a) SAMPEX; (b) MAP satellites

Background – Literature Review

Fuzzy logic control is the application of expert knowledge to control system. Fuzzy logic controllers are developed without use of analytic models. Unlike fuzzy controllers, classical control systems use error to determine exact control action and are model dependent. Walchko et al [2] performed numerous experiments with a hybrid fuzzy PID controller on linear system. In cases where the inputs and outputs had the same ranges, the hybrid fuzzy controller produced better results than its classical counter part. The systems studied were the DC servomotor, the hydraulic piston, the pneumatic linear actuator, and a ball screw. These bench mark experiments illustrate the flexibility of a single controller to successfully control different systems with few to no modifications, thus showing fuzzy logic to be model independent. Although other authors [3,4,5,6] have developed hybrid fuzzy PD and PID controllers, none have investigated their generic capabilities. In addition, few researchers have applied fuzzy logic to satellite control.

Wie and Barba [7] developed several computationally efficient control schemes for large angle maneuvers on satellites. Many of these schemes utilize quaternion and angular velocity feedback to provide stability control. Quaternions, the primary methods used for determining attitude, allow for more realistic, large angle maneuver control schemes. Most spacecraft are spin stabilized during deployment, which could produce large attitude errors. For example, if deployment occurs from the space shuttle, then the reaction jets must be turned off until the spacecraft reaches a minimum safe distance of 200 ft. Since the deployment rate is approximately one ft/sec, then no corrections can occur for 200 sec. An imperfect deployment (as little as .5 deg/sec) could cause the spacecraft to drift away from the desired attitude where a large angle correction (up to 180°) would be needed to reorient to the proper attitude. The control schemes developed by Wie and Barba are based on a Liapunov analysis, which only produces a range of positive stable gains for that control law. Thus in order to meet desired performance, engineers must iterate through a significant number of gain combinations to obtain the desired response. However, even when a satisfactory response is finally obtained, there are no guarantees how the satellite will behave in the presence of disturbances, noise, or uncertainties.

Crassidis and Markley [8] developed a model based nonlinear predictive control method for spacecraft, which allowed large-angle maneuvers. Their method utilized a one-time step ahead trajectory to predict control effort. This predictive control scheme determined the torque input required to make the predicted trajectories match the desired trajectories by minimizing the norm-squared error between the two. This method was robust against model error, and was simulated on the Microwave Anisotropy Probe

(MAP). Although this control scheme performed better than other traditional controllers presented in the paper, the design method was very complicated and time consuming to design. Thus, the design method would require an expert with in-depth knowledge to redesign a control system for each new satellite.

SAMPEX currently uses the Minimum Model Error (MME) estimation algorithm, which attempts to optimally estimate the dynamics of a system that was poorly modeled. This method designed by Pena et al [9], accurately models the state trajectories, and the state estimates are free of jump discontinuities that are present in Kalman filters. This method is used along with the current Liapunov controller and in the fuzzy controller to determine attitude rates since SAMPEX was not equipped with rate measuring devices.

Woodard [10] developed a fuzzy controller for the Fast Auroral Snapshot Explorer (FAST) and compared it to the traditional controller (AGSS). FAST was launched in August 1996 utilizing the Attitude Ground Support System (AGSS). AGSS is a group of algorithms developed in the 1970’s for the Dynamics Explorer mission. Woodard’s fuzzy controller proved to be inferior to the traditional AGSS. During pointing maneuvers the fuzzy controller took 15% longer to point in the desired direction and had a larger range of errors during pointing. However Woodard did point out that the fuzzy logic controller was mathematically simpler and more flexible, but not as accurate as the traditional method. “The performance of the fuzzy logic controller was slightly less desirable than that of the AGSS. This reinforces the general notion that performance with fuzzy logic controllers is sacrificed somewhat.” [10, p.103] This is an unfair notion of fuzzy logic’s performance capabilities. Fuzzy logic does have the capability to achieve or surpass the performance level of other traditional control schemes, as shown in this work.

Recently, Buijtenen et al [11] developed an adaptive fuzzy logic PD controller. Their method involves a critic that predicts the future system performance and a stochastic exploration module to explore the space of possible actions. The actual adaptation process is produced by reinforcement learning. Reinforcement learning (RL) is actually a family of biologically inspired algorithms. RL indirectly evaluates a controllers action and rewards desired outcomes and punishes undesirable outcomes. However, their method only looks at one attitude while the method proposed here looks at all four quaternions at once. Since the system is a coupled nonlinear system, changing one element of the attitude quaternion will have a dramatic effect on the other values. In addition, their method relies on the critic accurately predicting several steps into the future. This is a critical component of the RL, since the critic is used in the reinforcement process. Thus if the critic is inaccurate in predicting the future, this process has no hope of properly adapting.

Outline of Thesis

This thesis is laid out in the following manner. Chapter 2 presence the theory pertaining to classical controls and fuzzy logic. Chapter 3 describes the systems that the controller has been applied too. Chapter 4 is the design of the multi-input/multi-output (MIMO) fuzzy controller. Chapter 5 is the results of various simulations performed on the SAMPEX and MAP satellite. Chapter 6 contains conclusions, and recommendations.

CHAPTER 2
THEORY

Introduction

This chapter will provide simple understanding of both classical control and fuzzy logic. First, the classical methodology will be described as it pertains to a PID controller. Then, fuzzy logic will be introduced. Next, the hybrid structure, which is a composite of the two, will be described. This chapter lays the foundation and ground work for the next chapter which covers the controller itself. Finally, an introduction to quaternions and quaternion math will be provided

Classical Methodology

The fundamentals of classical control can be found in any undergraduate controls textbook (such as Dorf and Bishop [12] and Ogata [13]). Thus, this chapter will not attempt to explain all aspects of classical control, but rather a simple overview with emphasis on PID control.

Classical control theory is typically confined to the realm of single-input-single-output (SISO) control methodologies. The objective is to obtain a mathematical model of the closed loop system to be controlled and manipulate the dynamics of the system (via controller gains) to achieve the desired response. The modeling is based on the Laplace domain analysis (transfer functions).

Transfer Functions and Block Diagrams

Classical controls typically uses transfer functions, which are ratios of the output Laplace transform over the input Laplace transform. A Laplace transform is a transformation of a function f(t) from the time domain into the complex domain F(s). Laplace transforms convert Ordinary Differential Equations (ODE) into linear algebraic equations, which can easily be solved or manipulated.

Block diagrams are graphical representations of signal flow through a system of transfer functions. Since a system maybe composed of many transfer functions, a complex block diagram can be simplified down through block diagram manipulation. Block diagrams also provide a significant amount of physical insight on the interface between sub systems.

Feedback

To properly control a system in the presence of disturbance or uncertainty, you need continuous knowledge of the actual response and the desired response. Thus by comparing these two bits of information, an error is developed:

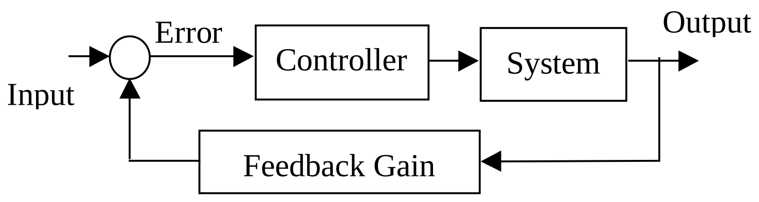


Figure 2-1. Feedback loop

Error = desired – actual = Input – Output

This error signal is the driving force for all classical feedback controllers. However, just knowing what your error is at any particular point in time does not guarantee that the controller will provide the desired response. Depending on the type of controller, one may or may not achieve the desired response. The most common classical controllers are lead, lag, P, PD, PI, and PID.

Root Locus

To achieve the desired controller response, the gains of the controller must be determined via some type of design procedure. The most commonly used procedure is the root locus.

The root locus method, which is a graphical method for drawing the locus of roots in the s-plane as a parameter is varied from zero to positive infinity [12], was originally developed by Evans in 1948. Since then, root locus has been used extensively in the development of controls system. Since the stability and transient performance of the closed loop system is directly related to the positions of the closed loop poles, the root locus enables a control engineer the ability to see the trend in the system dynamics and decide where he needs to move the poles to get the desired system response. The root locus is based on the characteristic equation of the closed loop system.

$$\text{Characteristic equation} = 1 + K_P\,(\text{Controller})(\text{System})$$

The root locus displays the pole locations as K_P varies from zero to positive infinity.

PID Control

Proportional, integral, derivative control (PID) is the most commonly used classical controller in industry. This is due to the PID's ability to provide a fast response with no steady state error (SSE). A PID controller can be designed by use of the root locus method, and the desired system characteristics.

Proportional control. Proportional control is the heart of the PID. Proportional control magnifies the error seen by the system. The higher the proportional gain, the faster the response time. If the proportional gain is too high, the system will overshoot the desired state and the system can become unstable.

Integral control. The integral control is a function of the integral of the error seen by the system. This gain is designed to eliminate SSE. However, this control decreases the stability of the system as the gain increases. Thus, a system with a high integral gain can become unstable.

Derivative control. The derivative control is designed to reduce the velocity error to zero. This type of control has the effect of increasing the dampening of the system, thus producing a stiffer system response.

Fuzzy Logic

Fuzzy logic was conceived by Lotfi Zadeh and brought to the attention of the world in his paper “Fuzzy Set” in 1965. At that time, fuzzy logic was a radical deviation from classical logic, and initially received little interest and attention. Slowly fuzzy logic is gaining more and more credit in the United States as a legitimate method for controls. The driving forces behind the adoption of fuzzy logic to controls applications are: model independence, use of expert knowledge to control systems, robustness to noise/disturbances, capable of controlling nonlinear systems, etc.

Fuzzy logic was biologically inspired by a world of living organism which do not face life in the classical binary logic sense of true or false. Living organisms deal with sometimes-subtle gradual changes of truth, where there is no crisp black and white, but rather shades of gray. Fuzzy logic is a method which allows computers to deviate from the classical binary decision making method, and embrace the shades of gray world of human existence.

Fuzzy logic is a vast and complex topic. Thus, the purpose of this section is not to cover all aspects of fuzzy logic, but rather introduce an unfamiliar reader to the concept of fuzzy logic and therefore lay a foundation for material throughout the rest of this chapter. For more information on fuzzy logic, the reader is referred to Kosko [14]

Classical and Fuzzy Set Theory

In the sections below, a description of classical set theory, fuzzy set theory and fuzzy logic as a whole is presented. With this basic understanding of the difference between classical and fuzzy, a better understanding of the under lying principles of fuzzy logic should emerge.

Classical set theory. Classical set theory is marked by a true or false, on or off, 1 or 0, $FF or $00 relationship. There is a clear and unambiguous line in the sand where false ends and truth begins. For example, if Y is a classical set composed of real numbers whose value is greater than 10, then

$$Y = \{x \mid x > 10\}$$

There is a crisp boundary where 10.00000 does not belong to Y, but 10.0000001 does belong to Y. This type of logic is simple for computers to understand and use, but classical sets are unintuitive to humans who do not see real life situations in black and white but rather shades of gray.

Fuzzy set theory. Fuzzy set theory on the other hand is quite different from the classical form. Fuzzy set theory introduces these shades of gray that is inherent to human perception of everyday life and allows computers to participate in this ambiguity of life. For example, suppose that Y is the set tall trees and x is the height of an individual tree greater than ten meters tall.

$$Y = \{x \mid x > 10 \text{ m}\}$$

Like the previous example, classical set theory would give drastically different answers for a tree 10.00000 m tall not belonging in set Y (false) and a tree 10.00001 m tall belonging to set Y (true). However, humans would not be so drastic in their choice of answers. A person would look at the two trees and say that there is little real difference between 10.00000 m and 10.00001 m. Thus if the 10.00001 m tall tree belongs to set Y then a 10.00000 m tall tree should also to some degree belong to set Y.

Another way to look at fuzzy set theory is, if fifty people where asked what they considered to be fast driving, there would be fifty different answers. This is because the human experience is a very subjective one. What one person considers fast, someone else might think is too fast or not fast enough. This also shows the shades of gray that humans see, but classically logical computers do not.

Fuzzy Rule Base

Now that there is a better understanding between classical and fuzzy set theory, we need to define the framework for how we are going to use fuzzy logic. Membership functions are fuzzy sets (as described previously) and fuzzy rules are a method of joining sets together. A series of rules define what is referred to as the rule base. In controls, this would be your input/output control surface.

From the previous examples, let X be the sample space (universe of discourse) from which x is drawn, and Y X where Y is a collection of elements of x. In the classical instance, x will either belong or not belong to Y (true - 1 or false - 0). Thus for each element x in X, a set of ordered pairs which will denote the degree of membership, (x,0) and (x,1), can be constructed to represent the classical set Y.

Now in fuzzy logic, a similar pairing will occur to denote the degree of membership when given a fuzzy set Y in X where X contains a collection of elements of x. Now each element x in X can be mapped to Y by a set of ordered pairs:

$$Y = \{(x,\mu_Y(x)) \mid x \in X\}$$

where $\mu_Y(x)$ is the membership function (MF) for fuzzy set Y. The MF will map each element x to a membership value between 0 (false) and 1 (true).

Fuzzy Inference

Fuzzification and fuzzy reasoning. The fuzzy logic process starts with obtaining some crisp values from user input or sensor on the system that is being controlled. The first thing that has to be done is to fuzzify these crisp numbers into fuzzy numbers. This process is of course accomplished by use of the appropriate MFs to which the crisp numbers belong.

Once all of the inputs are fuzzified, the rules need to be applied to the now fuzzy numbers. In fuzzy logic, all rules are executed in parallel; thus no rules in the rule base are ever not evaluated.

Rules. Fuzzy logic rules describe the relationship between the input and the output, and (from a controls standpoint) defines an input/output surface of control effort.

This surface has been used to quantify stability by Petroff et al [15] of fuzzy controllers. Fuzzy rules take the standard if then form, i.e.

IF (Y is x) then (B is a)

IF (antecedent) then (premise)

where x is an input, Y and B are a fuzzy sets, and a is an output. A fuzzy system can have multiple inputs and multiple outputs (MIMO), where the inputs and outputs are joined together by ANDs and ORs.

IF (A is a) AND (B is b) AND (C is c) AND … then (AA is aa) AND …

In addition, many other operators can be used to combine and manipulate fuzzy rules. One example is hedges (i.e.. Not, Very, Somewhat, below, above, etc) [16]. These hedges will not be described here since they were not used in this work.

Defuzzification. Once the fuzzy reasoning is complete, a fuzzy output can be produced. However, this output is only meaningful to humans and not to computers and other classical logic equipment. Thus, a method is necessary to transform the resulting fuzzy output into a crisp numerical output. There are several methods to accomplish this. Some of the most meaningful schemes are described below and illustrated in Figure 2-2.

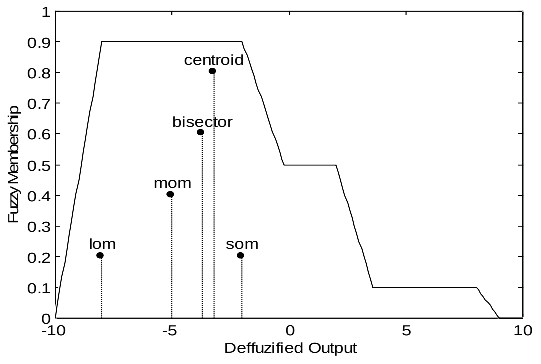


Figure 2-2. Comparison of various defuzzification techniques [16]

Winner takes all / Singleton – this is the simplest defuzzification method to implement. The output MF with the highest membership wins.

Mean of max (mom) – this is the average.

Bisector of area – this is a partition of the resulting area into two regions.

Centroid of area – this is the most popular defuzzification method. This method finds the center of mass of the output region.

The entire process is shown in Figure 2-3. The n inputs on the left are fuzzified by MFs. Then each of the fuzzified values is put into the rules that produce m outputs. Each of these m outputs are combined (there are various methods) and the defuzzification of the resulting area is found. Now the whole process of fuzzification, rule application, and defuzzification can be put together.

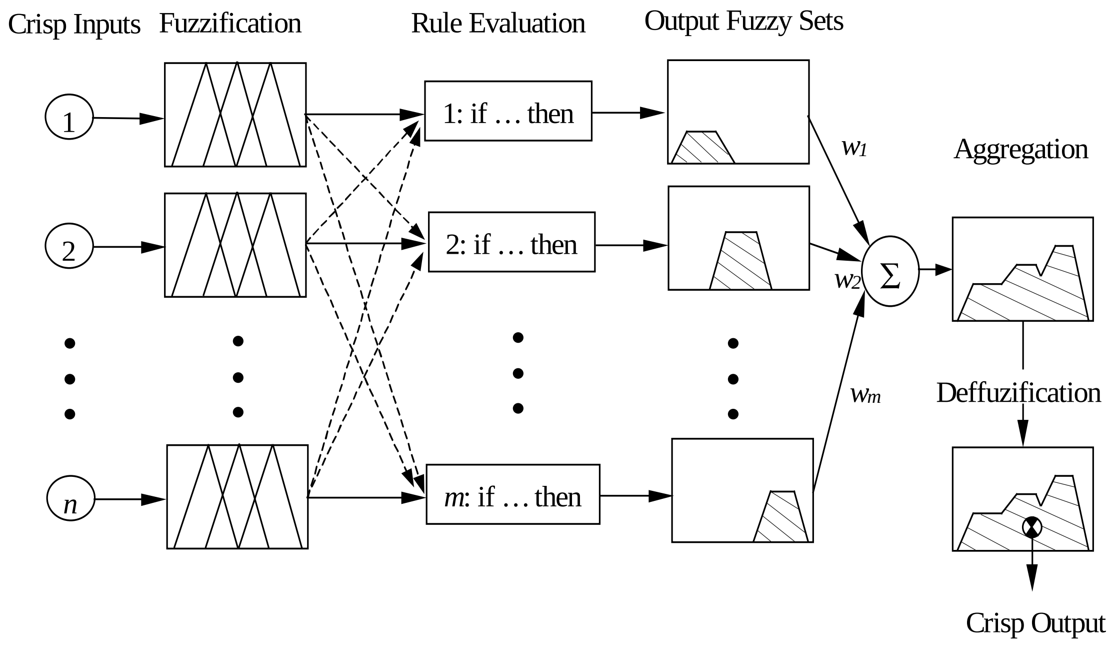


Figure 2-3. Fuzzy Logic from input to output [16]

Fuzzy Stability

Stability is still one of the major obstacles against the wide spread adoption of fuzzy logic. As of now, there is no general method of determining whether a fuzzy controller is stable or unstable. However, there is one method described by Petroff et al [15] to linearize small areas of the control surface and then determine whether the controller is stable over that area. However, fuzzy logic control surfaces tend to be very complex and analyzing each small area to prove general stability of the whole surface is not always practical.

Stability for the proposed fuzzy controller in this work will be accomplished via Liapunov stability analysis. A Liapunov function will be found, which is stable, and the fuzzy controller will be shown to mimic this function. The down side of this method is that the Liapunov function will be assumed to contain a PD controller; thus this method would not work for a fuzzy controller where similarity between classical and fuzzy control can not be made.

Hybrid Structure

In this work, the hybrid controller structure is based upon an open architecture such that it can be inserted into any error-based SISO control system. This hybrid algorithm is described graphically in Figure 2-4. These fuzzy gains are time varying.

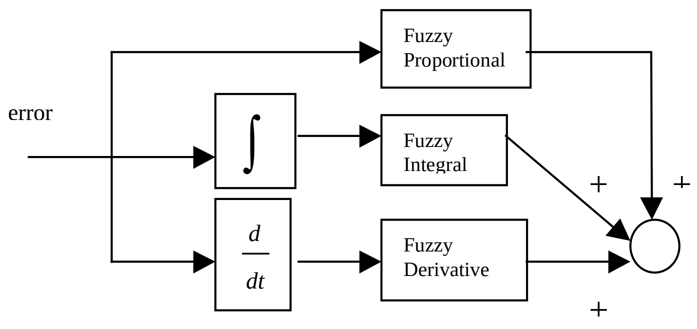


Figure 2-4. The Hybrid structure of the fuzzy controller [2]

Generic Characteristics

Due to the structure of the current SISO hybrid schemes, generic characteristics are embedded into the hybrid algorithm such that it can be integrated into a large number of systems to enhance performance without a significant amount of effort. A generic controller would simplify the design of new control systems while still maintaining performance. This characteristic introduces a degree of “plug and play” to controller design. The paragraph below illustrates this attribute for a SISO system. Later, further work will show the generic properties of the controller as they pertain to MIMO satellites.

Given a second order system with a standard PID controller, the closed-loop transfer function is

$$TF = \frac{(K_D s^2 + K_P s + K_I)\omega_n^2}{s^3 + (2\zeta\omega_n + K_D\omega_n^2)s^2 + (1+K_P)\omega_n^2 s + K_I\omega_n^2} = \frac{N(s)}{D(s)}$$

Now let $\tilde{\omega}_n = (1+\Delta)\omega_n$ and $\tilde{\zeta} = (1+\beta)\zeta$

Where Δ and β are uncertainties in the system parameter from the nominal or modeled values. The characteristic equation is

$$D(s) = s^3 + (2\tilde{\zeta}\tilde{\omega}_n + K_D\tilde{\omega}_n^2)s^2 + (1+K_P)\tilde{\omega}_n^2 s + K_I\tilde{\omega}_n^2$$

Based on the fuzzy PID structure, the control effort can be written as

$$u_C = \mathrm{fuzzy}_P(e) + \mathrm{fuzzy}_D(\dot{e}) + \mathrm{fuzzy}_{\int}(\textstyle\int e) \cong K_P(1+\Delta_e)e + K_I(1+\Delta_I)\textstyle\int e + K_D(1+\Delta_D)\dot{e}$$

where Δ_e, Δ_I , and Δ_D are functions of the universe of discourse.

The fuzzy universe of discourse can be adjusted to account for variations in the model. This conclusion can be extended to state that the fuzzy PID is a generalized controller (generic) like its classical controls counter part. Although there may be variations, systems having similar levels of controllability and dominant dynamics will yield similar controlled responses. Therefore, this hybrid algorithm can be used as a generic controller.

Quaternions and Euler Angles

Euler Angles

Euler angles are typically though of in terms of roll (ϕ), pitch (θ), and yaw (ψ). These terms are shown graphically below.

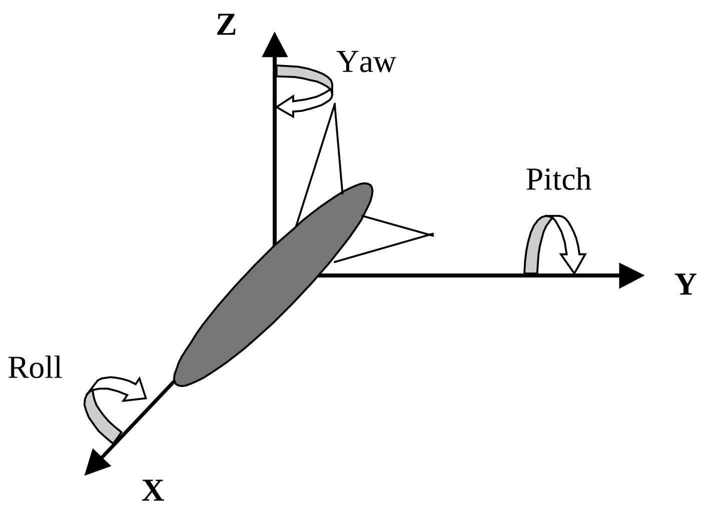


Figure 2-5. Spaceship with pitch, yaw, and roll

In order to perform these operations on the spacecraft, an attitude matrix (**A**) is used to find the new orientation of the spacecraft given the old orientation. Given below are the attitude matrices for rotations about the x, y, and z-axes.

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

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

$$A_z(\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Now, subsequent rotations about these primary axes can be accomplished by multiplying the matrices together. Thus, successive rotations about the z-axis, x-axis, and y-axis are given by:

$$A_{zxy}(\psi, \theta, \phi) = A_z(\psi)A_x(\theta)A_y(\phi) =$$

$$\begin{bmatrix} \cos(\phi)\cos(\psi) - \sin(\theta)\sin(\phi)\sin(\psi) & \cos(\phi)\sin(\psi) - \sin(\theta)\sin(\phi)\cos(\psi) & -\cos(\theta)\sin(\phi) \\ -\cos(\theta)\sin(\psi) & \cos(\theta)\cos(\psi) & \sin(\theta) \\ \sin(\phi)\cos(\psi) + \sin(\theta)\cos(\phi)\sin(\psi) & \sin(\phi)\sin(\psi) - \sin(\theta)\cos(\phi)\cos(\psi) & \cos(\theta)\cos(\phi) \end{bmatrix}$$

If small angle approximation is used, **A** can be simplified down too:

$$A = \begin{bmatrix} 1 & \psi & -\phi \\ -\psi & 1 & \theta \\ \phi & -\theta & 1 \end{bmatrix}$$

However, throughout this work large angle maneuvers will be done. Thus this simplification is not of interest to us.

Finally, note that these rotations presented here were about body fixed axes. If a rotation was desired about an arbitrary axis (**e**) in space by angle (Φ) could be accomplished by the following attitude matrix.

$$A = \begin{bmatrix} \cos(\Phi)+e_1^2(1-\cos(\Phi)) & e_1e_2(1-\cos(\Phi))+e_3\sin(\Phi) & e_1e_3(1-\cos(\Phi))+e_2\sin(\Phi) \\ e_1e_2(1-\cos(\Phi))-e_3\sin(\Phi) & \cos(\Phi)+e_2^2(1-\cos(\Phi)) & e_2e_3(1-\cos(\Phi))+e_1\sin(\Phi) \\ e_1e_3(1-\cos(\Phi))+e_2\sin(\Phi) & e_2e_3(1-\cos(\Phi))-e_1\sin(\Phi) & \cos(\Phi)+e_3^2(1-\cos(\Phi)) \end{bmatrix}$$

$$e \equiv \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix}$$

Quaternions

Quaternions, also known as Euler symmetric parameters, are in many instances a more mathematically efficient way to compute rotations of rigid and non-rigid body systems than traditional methods involving standard rotational matrices or Euler angles. Quaternions have the advantage of no trigonometric functions needed to compute attitude. Also, there exists a product rule for successive rotations that greatly simplifies the math, thus reducing processor computation time. The major disadvantage of quaternions is the lack of intuitive physical meaning. Most people would understand where a point was if they were given {1,2,3}, however, few would comprehend where a point was if given the quaternion {1,2,3,4}. This section does not attempt to provide the extensive understanding needed to employ quaternions but rather a simple introduction. Further information can be found in Wertz [17] or Crane and Duffy [18].

Mathematics. The quaternion is composed of a scalar and a vector part. The scalar is a redundant element that prevents singularities from occurring since the four elements are all dependent upon each other.

$$q=(q1,q2,q3, q4)=(a\mathbf{i}, b\mathbf{j}, c\mathbf{k}, d)$$

Quaternion addition behaves as normal vector addition.

$$q1+q2=(a1+a2, b1+b2, c1+c2, d1+d2)$$

Multiplication of a quaternion by a scalar value also behaves much like vector scalar multiplication.

$$\lambda q=(\lambda a, \lambda b, \lambda c, \lambda d)$$

However, the multiplication of two quaternions behaves much differently from traditional vector multiplication.

$$q1q2 = \Xi(q1)q2 = d1d2-a1a2-b1b2-c1c2+d1(a2\mathbf{i}+b2\mathbf{j}+c2\mathbf{k})+d2(a1\mathbf{i}+b1\mathbf{j}+c1\mathbf{k})+\mathbf{det}[q1,q2]$$

To calculate the inverse of a quaternion constitutes several steps. First, the conjugate of the quaternion (qq) is found by negating the vector part.

$$qq = (-a\mathbf{i}, -b\mathbf{j}, -c\mathbf{k}, d)$$

Next, the norm of the quaternion is calculated by multiplying the quaternion by its conjugate.

$$\text{norm} = q(qq)=d^2+a^2+b^2+c^2$$

Finally, the inverse is found by scalar division of the conjugate by the norm.

$$q^{-1} = qq/norm$$

Rotations of rigid bodies in space. In order to perform a rotation (Φ) of a rigid body about an arbitrary moving/fixed axis (**e**) in space, where **e** is defined by

$$e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix}$$

The quaternion representation of this operation is

$$q_1 \equiv e_1 \sin\left(\frac{\Phi}{2}\right)$$
$$q_2 \equiv e_2 \sin\left(\frac{\Phi}{2}\right)$$
$$q_3 \equiv e_3 \sin\left(\frac{\Phi}{2}\right)$$
$$q_4 \equiv \cos\left(\frac{\Phi}{2}\right)$$

The attitude matrix (**A**) for this is

$$A(q) = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} =$$

$$\left(q_4^2 - q^2\right) \cdot I + 2qq^T - 2q_4Q$$

$$Q \equiv \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}$$

where I is the identity matrix.

Successive rotations can be accomplished by multiplying the attitude matrices together. Thus for two successive rotations, the attitude matrix that would accomplish both would be:

$$A(\mathbf{q}'')=A(\mathbf{q}')A(\mathbf{q})$$

This process though can be simplified computationally by using the following expression:

$$q''=\begin{bmatrix} q'_4 & q'_3 & -q'_2 & q'_1 \\ -q'_3 & q'_4 & q'_1 & q'_2 \\ q'_2 & -q'_1 & q'_4 & q'_3 \\ -q'_1 & -q'_2 & -q'_3 & q'_4 \end{bmatrix} \cdot q$$

Thus, the above equation will provide the quaternion with the two successive rotations already imbedded into it. Had the two attitude matrices been calculated and multiplied together, that would have resulted in twenty-seven multiplication (the same number as for Euler) operations rather than the sixteen multiplication operations shown here.

Hopefully it can be seen that quaternions are better than other Euler operations to determine attitude. Quaternions lack singularities due to one redundant element. They also lack the computationally intensive trigonometric functions, and contain a simplified way to determine successive rotations about an arbitrary axis.

CHAPTER 3
SYSTEM MODELS AND DYNAMICS

In this chapter, a brief description of the two satellites used in this study, SAMPEX and MAP, is given. Their background and mission are briefly stated for familiarity. Also, there is a quick review of satellite dynamics at the end of the chapter.

MAP Satellite

The MAP (Microwave Anisotropy Probe) satellite's primary mission is to probe conditions in the early universe by measuring the properties of the cosmic microwave background radiation over the full sky. MAP will accomplish this by using passively cooled differential microwave radiometers with dual Gregorian 1.4 x 1.6 meter primary reflectors. Questions that MAP could answer:

How old is the universe?

How fast is the Universe expanding?

Is the universe infinite?

Is there a cosmological constant?

What is the density of ordinary ("baryonic") matter?

When did the first stars form?

What is the origin of structure in the universe?

The MAP satellite is scheduled to launch from Cape Canaveral, Florida in Fall 2000. The launch vehicle is a Delta II 7425-10 rocket. Once the rocket launches, it will revolve around its axis at a rate of about 60 rpm. This spinning will slow by a process called the "yo-yo despin."

Figure 3-1. Delta rocket.

The yo-yo despin is a technique used to stop the Delta rocket from revolving around its axis after it has launched. Weights on strings are attached to the side of the rocket. Once it is activated, the strings unwind like a yo-yo and the weights are released. Due to the conservation of angular momentum, the rocket's angular velocity (its spin rate) must decrease. This process is analogous to a figure skater who is spinning at high speed on ice and extends her arms to slow herself down.

After MAP is released from the rocket, its own propulsion system carries it into a lunar assisted trajectory to the Sun-Earth L2 liberation point for a nominal 27 month mission. L2 is an abbreviation for "Lagrange point #2." At this point, MAP will orbit the Sun along with Earth (i.e., the Sun is always behind the Earth relative to the satellite. This is possible because L2 is an (unstable) equilibrium point in space. It is a point on the border between the last orbit around the Earth and the first orbit around the Sun. The L2 point is especially useful for the MAP project because

L2 is far enough away from the Earth to be fairly well protected from the Earth's microwave emission, magnetic fields, and other disturbances.

The Sun, Earth, and Moon are always behind the instrument's field of view.

MAP will get to L2 by using both the Earth's and the Moon's gravity to "loop" around the moon between 2.5 and 3.5 times until it is in a favorable position to continue on its trip to the L2 point. This trip will take about one hundred days, or a little more than three months. With this trajectory, it will consume as little fuel as possible.

MAP is a large satellite, whose inertia matrix is shown below.

$$I = \begin{bmatrix} 399 & -2.81 & -1.31 \\ -2.81 & 377 & 2.54 \\ -1.31 & 2.54 & 377 \end{bmatrix} \cdot \mathrm{kg} - \mathrm{m}^2$$

SAMPEX Satellite

SAMPEX (Solar, Anomalous, and Magnetospheric Particle Explorer) is a product of the SMEX (Small Explorer Program). This program realizes the advantages of small, quick turnaround projects. The SAMPEX satellite is designed and equipped to study the energy, composition, and charge states of particles from the explosions of supernovas, solar flares, and from the depths of interstellar space. Closer to home, SAMPEX monitors the earth's middle atmosphere as magnetospheric particle populations occasionally plunge into it.

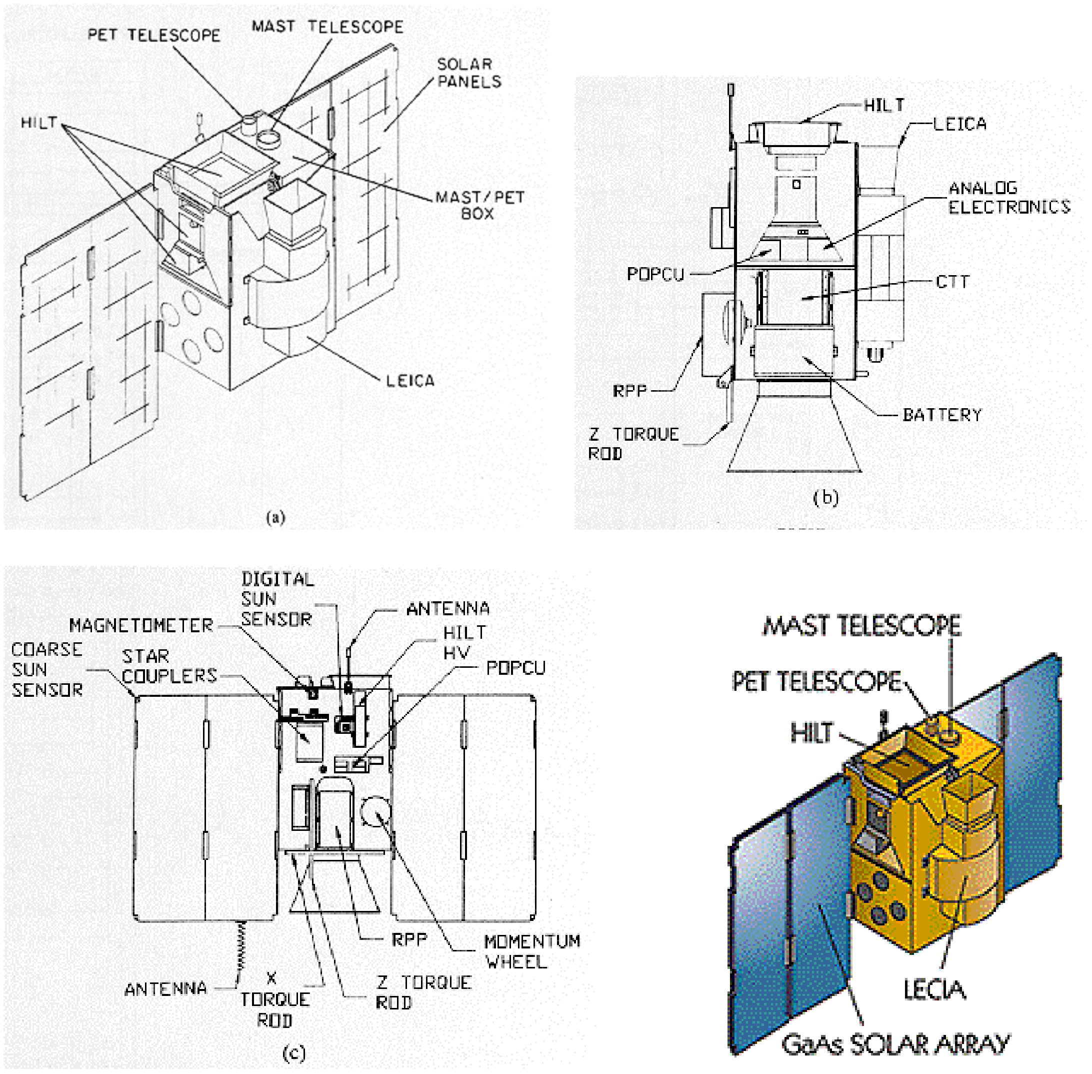


Figure 3-2. The SAMPEX satellite

Although SAMPEX was manufactured with limited resources, it contains many innovations such as: powerful processors (80386/80387), optical fiber buses, solid state memory, and a highly integrated mechanical design. SAMPEX was limited both monetarily and physically. It was launched into orbit using the scout launch vehicle, which is limited to a cargo of less than 372 lb. To help reduce the bulk, SAMPEX is a single string architecture spacecraft. SAMPEX consists of the primary structure, a deployable solar array system, and the yo-yo despin system.

SAMPEX and MAP both use the attitude control subsystem (ACS) which is a solar-pointed/momentum-bias system designed to always point towards the sun while rotating about the sunline once per orbit. Pointing the satellite is accomplished by choosing sensor, torquers, and system configurations from a standard set of electronics, sensors, and actuators. One momentum wheel and three electromagnetic torque rods are used to orient the experiment-viewing axis by the ACS system. Real-time attitude determination and control is accomplished by the onboard data systems. The attitude is determined by comparing the local measured magnetic field vector with an on-board ephemeris model. Appropriate digital control signals are sent across the optical bus to pyrotechnic actuators to control the spacecraft. The ACS also contains various operational modes that the satellite can be in. Figure 3-3 shows the state diagram of the ACS mode transitions.

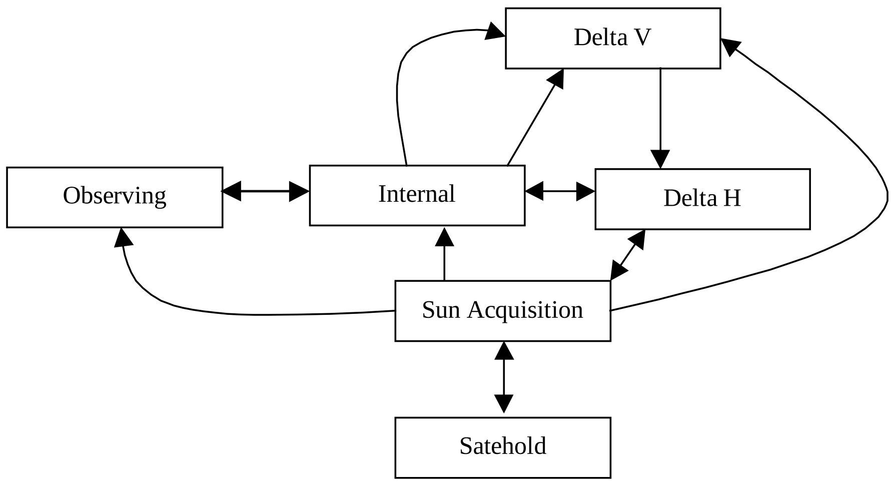


Figure 3-3. ACS State diagram [19]

For further explanation of the various ACS states, the reader is referred to Baker et al [19].

SAMPEX is also operated by POCC (project operations control center) located at Goddard Space Flight Center. The mission is constantly manned 24 hrs a day, seven days a week. Data are sent here twice a day by SAMPEX and commands are sent up to SAMPEX once a day.

SAMPEX is a small symmetric satellite compared to MAP. The inertia tensor and reaction wheel inertia for the SAMPEX satellite is shown below

$$I = \begin{bmatrix} 15.516 & 0 & 0 \\ 0 & 21.621 & -.194 \\ 0 & -.194 & 15.234 \end{bmatrix} \cdot \text{kg} - \text{m}^2$$

$$I_{rw} = .0041488 \text{ kg} - \text{m}^2$$

Spacecraft Dynamics

Next, a brief review of the kinematic and dynamic equations of motion for a three-axis stabilized spacecraft is presented [20]. The attitude is assumed to be represented by the quaternion, defined as

$$\underline{q} \equiv \begin{bmatrix} \underline{q}_{13} \\ q_4 \end{bmatrix} \tag{3.1}$$

with

$$\underline{q}_{13} \equiv \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \underline{\hat{n}} \sin(\theta / 2) \tag{3.2a}$$

$$q_4 = \cos(\theta / 2) \tag{3.2b}$$

where $\underline{\hat{n}}$ is a unit vector corresponding to the axis of rotation and θ is the angle of rotation. In Wertz [17] the quaternion kinematic equations of motion are derived by using the spacecraft's angular velocity ($\underline{\omega}$),

$$\dot{\underline{q}} = \frac{1}{2}\Omega(\underline{\omega})\underline{q} = \frac{1}{2}\Xi(\underline{q})\underline{\omega} \tag{3.3}$$

where ω, $\Omega(\underline{\omega})$ and $\Xi(\underline{q})$ are defined as

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = I^{-1}\begin{bmatrix} L_1 \\ L_2 - I_{rw} \times \omega_{rw} \\ L_3 \end{bmatrix}$$

$$\Omega(\underline{\omega}) \equiv \begin{bmatrix} -[\underline{\omega}\times] & \vdots & \underline{\omega} \\ \cdots & \vdots & \cdots \\ -\underline{\omega}^T & \vdots & 0 \end{bmatrix} \tag{3.4a}$$

$$\Xi(\underline{q}) \equiv \begin{bmatrix} q_4 I_{3\times3} + [\underline{q}_{13}\times] \\ \cdots \\ -\underline{q}_{13}^T \end{bmatrix} \tag{3.4b}$$

where $I_{n\times n}$ represents an $n \times n$ identity matrix (also, $0_{n\times m}$ will represent a $n \times m$ zero matrix), L is the angular momentum, and ω_{rw} is the angular velocity of the reaction wheels. The 3 x 3 dimensional matrices $[\underline{\omega}\times]$ and $[\underline{q}_{13}\times]$ are referred to as cross product matrices since $\underline{a} \times \underline{b} = [\underline{a}\times]\underline{b}$, with

$$[\underline{a}\times] \equiv \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{3.5}$$

Since a three degree-of-freedom attitude system is represented by a four-dimensional vector, the quaternion components cannot be independent. This condition leads to the following normalization constraint

$$\underline{q}^T \underline{q} = \underline{q}_{13}^T \underline{q}_{13} + q_4^2 = 1 \tag{3.6}$$

In addition, the matrix $\Xi(\underline{q})$ obeys the following helpful relations

$$\Xi^T(\underline{q})\Xi(\underline{q}) = \underline{q}^T \underline{q} I_{3\times3} \tag{3.7a}$$

$$\Xi(\underline{q})\Xi^T(\underline{q}) = \underline{q}^T \underline{q} I_{4\times4} - \underline{q}\,\underline{q}^T \tag{3.7b}$$

$$\Xi^T(\underline{q})\underline{q} = \underline{0}_{3\times1} \tag{3.7c}$$

$$\Xi^T(\underline{q})\underline{\lambda} = -\Xi^T(\underline{\lambda})\underline{q} \text{ for any } \underline{\lambda}_{4\times1} \tag{3.7d}$$

Moreover, the error quaternion of two quaternions, $\underline{q}$ and $\underline{\tilde{q}}$, is defined by

$$\underline{\delta q} \equiv \begin{bmatrix} \underline{\delta q}_{13} \\ \cdots \\ \delta q_4 \end{bmatrix} = \underline{q} \otimes \underline{\tilde{q}}^{-1} \tag{3.8}$$

where the operator $\otimes$ denotes quaternion multiplication (see Ref. 16 for details), and the inverse quaternion is defined by

$$\underline{\tilde{q}}^{-1} = \begin{bmatrix} -\tilde{q}_1 & -\tilde{q}_2 & -\tilde{q}_3 & \tilde{q}_4 \end{bmatrix}^T \tag{3.9}$$

Another useful identity is given by

$$\underline{\delta q}_{13} = \Xi^T(\underline{\tilde{q}}^{-1})\underline{q} \tag{3.10}$$

In addition, if Equation (3.8) represent a small rotation then $\delta q_4 \approx 1$, and $\underline{\delta q}_{13}$ corresponds to half-angles of rotation.

The dynamic equations of motion, also known as Euler's equations, for a rotating spacecraft are given by

$$\dot{\underline{H}} = -\underline{\omega} \times \underline{H} + \underline{u}_{ext} \tag{3.11}$$

where $\underline{H}$ is the total system angular momentum, $\underline{u}_{ext}$ is the total external torque (which includes, control torques, aerodynamic drag torques, solar pressure torques, etc.). The angular velocity from of Euler's equation could be used, given by

$$J\dot{\underline{\omega}} = -\underline{\omega} \times (J\underline{\omega}) + \underline{u} \tag{3.12}$$

where J is the inertia matrix of the spacecraft, and $\underline{u}$ is the total torque. Equations (3.8), (3.9), and (3.12) can be used to show that rotational motion without nutation only occurs if the rotation is about a principal axis of the rigid body.

CHAPTER 4
FUZZY CONTROLLER DESIGN

Introduction

This chapter covers the basic design and layout of the fuzzy controller. Next, a description of the method used to extend the SISO fuzzy controller to a MIMO controller is described. Finally, the stability and robustness of the controller are examined.

The Controller

The fuzzy satellite controller utilizes the same rules and structures as Walchko et al [2] to control various linear systems. The rule base was developed by first designing a conventional PID controller using the root locus method, and developing the relationship between errors (error, integral of error, and derivative of error) and the control. The first set of rules was constructed to mimic this relationship.

Like the conventional PID, the fuzzy PID hybrid consists of three inputs (error, integral of error, and derivative of error) and one output (control effort). Since this is a multi dimensional fuzzy controller, it is very difficult to display the rule tables. Thus, only some the possible rules will be shown. The rule tables for the controller are shown below in Figure 4-1, where NEG, SN, SSN, Z, SSP, SP, POS are negative, small negative, small small negative, zero, small small positive, small positive, positive respectively:

Table 4-1. Fuzzy PID controller outputs for selected inputs. (a) Error and Integral of Error; (b) Error and Derivative of Error; (c) Integral and Derivative of Error.

INT / E	NEG	SN	SSN	Z	SSP	SP	POS
POS	Z	SSP	SP	POS	POS	POS	POS
SP	SSN	Z	SSP	SP	POS	POS	POS
SSP	SN	SSN	Z	SSP	SP	POS	POS
Z	NEG	SN	SSN	Z	SSP	SP	POS
SSN	NEG	NEG	SN	SSN	Z	SSP	SP
SN	NEG	NEG	NEG	SN	SSN	Z	SSP
POS	NEG	NEG	NEG	NEG	SN	SSN	Z

(a)

DER / E	NEG	Z	POS
POS	POS	POS	POS
SP	SP	SP	POS
SSP	SSP	SSP	POS
Z	Z	Z	Z
SSN	NEG	SSN	SSN
SN	NEG	SN	SN
NEG	NEG	NEG	NEG

(b)

INT / DER	NEG	Z	POS
POS	POS	POS	POS
SP	SP	POS	POS
SSP	SSP	SP	SP
Z	Z	Z	Z
SSN	SN	SN	SSN
SN	NEG	NEG	SN
NEG	NEG	NEG	NEG

(c)

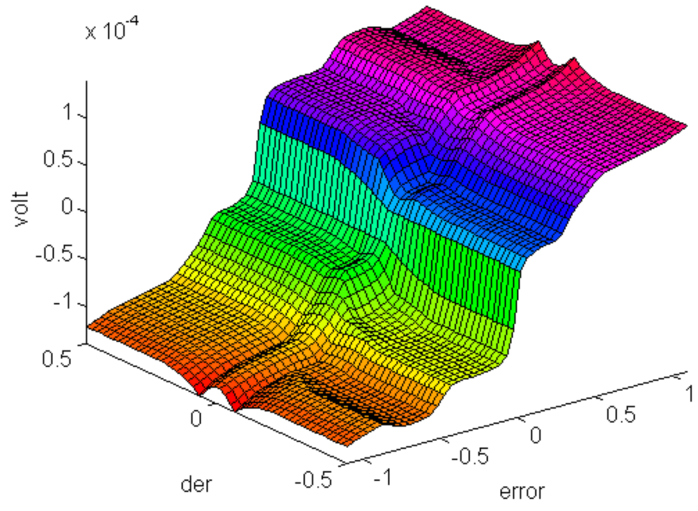


Figure 4-1. Control Surface

Since the satellites have the ability to get rid of steady state error, only a PD controller is used. This is accomplished by using a momentum wheel for pitch error angle control. The system momentum vector can be computed from the momentum wheel speed.

Extension of the Fuzzy Hybrid Controller to MIMO Applications

The fuzzy PID controller was originally designed for SISO systems. In order to apply it to the satellite problem, a technique to extend the SISO scheme into the multi input realm must be developed. This conversion (or extension) can be accomplished through the manipulation of the error vector, which is used to define the direction and magnitude of the control effort to reduce the error in a minimal amount of time. The motivation and the formalization of this technique are given below.

The error used by the initial fuzzy PD is a scalar. However, in the MIMO case, it is a vector. In order to maintain the fuzzy PID structure, the vector is transformed into a scalar representation. This scalar representation is used to calculate the MIMO control effort. The transformation is based on the norm of the error vector.

$$|\vec{e}| = \text{scalar representation} \Rightarrow \text{Fuzzy(scalar representation)}$$

This scalar representation is used by the fuzzy inference system to determine control magnitude, which is used along with a given direction to create the MIMO control vector. The effort level must be based on the ratios of the error magnitudes, which are defined by the direction. This direction is determined from the normalized error vector. This vector was normalized because of the different levels of possible errors. For example, if the first element was .001 and the third element was 4000, the input error to the fuzzy controller would be dominated by 4000 and the output control effort would be large. This would result in over controlling of the system, or result in instability.

The control effort vector is defined by magnitude and direction. This relationship is used to extend the single control effort to multi output control effort.

$$\text{direction} = \frac{e}{|e|}$$

$$\text{control\_vector} = u_{\text{fuzzy}} \cdot \text{direction}$$

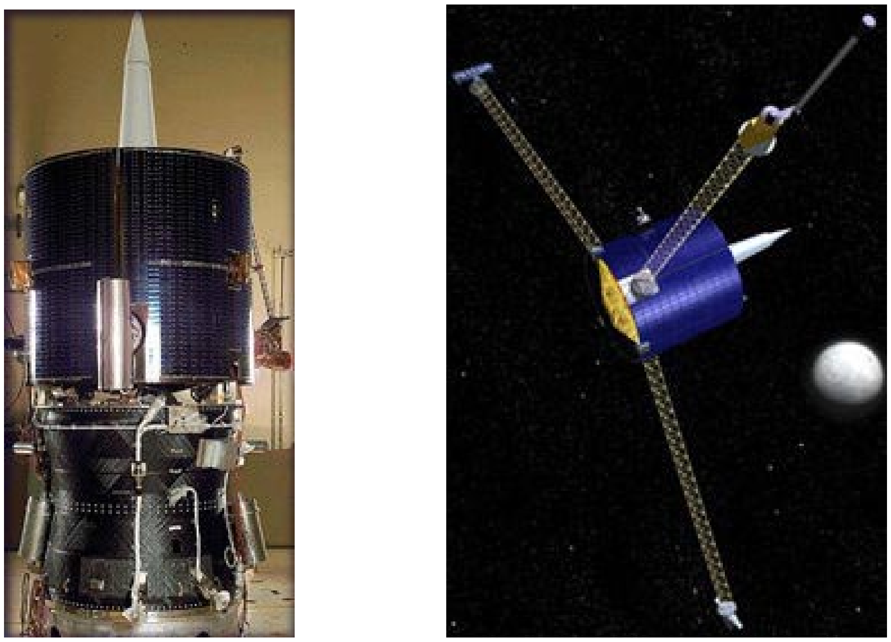
Since the fuzzy input is always positive, the complexity of the fuzzy inference system is significantly reduced. The number of rules is cut in half, which alleviates the need for all of the negative MF's on the inputs and output (it is important to remember this fact when we discuss stability later). This does not mean that there will never be a negative control effort. Negatives are reintroduced back into the system by the direction vector.

Why Use This Error Vector Thing, Why Not Just Use Multiple Fuzzy Controllers?

By using the error vector, the computational load on the satellite's control system is reduced. This is a very important factor since NASA wait's until processors are space certified, which can take up to ten years. Thus, while military and other commercial satellites may use better faster processors, NASA uses proven space worthy technology. This is obviously good and bad. Unfortunately, the current level of processor power is on the order of an Intel 486/early Pentium. Thus, it is important for this, or any, control design to be implemented in a computationally efficient manner.

For example, the most recent visit to the moon by the Lunar Prospector was done remotely from the ground.

> Prospector has no on-board computer, only an electronics box that executes commands it receives from a team on Earth. "It's a very stupid spacecraft which is great because nothing ever goes wrong," says Alan Binder, director of the Lunar Research Institute and designer of the Lunar Prospector Mission. The spacecraft's simplicity also has benefits for the research, since it produces little background noise to interfere with the scientific experiments, Binder says. [19, p.501]



(a) (b)

Figure 4-2. Lunar Prospector (a) at Lockheed and Martin; (b) deployed in space

Thus in the pursuit of lower cost missions, slower processors may be used by NASA which provide low computational abilities. For example, the SAMPEX satellite is equipped with an 80386/80387 processor.

Is This Error Formulation Valid?

Normally, the classical PD control effort would be

$$u = K_P \cdot e + K_D \cdot \dot{e}$$

However, in the MIMO case, the error (e) and error velocity ($\dot{e}$) are vectors, which produce the following control effort.

$$u = K_P \cdot \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} + K_D \cdot \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix}$$

In order to illustrate this concept, the following numerical example is presented.

$$e = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \cdot \text{meters}$$
$$\dot{e} = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} \cdot \frac{\text{meters}}{\text{sec}}$$

Normalizing both vectors.

$$u = K_P \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \cdot \text{meters} + K_D \cdot \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} \cdot \frac{\text{meters}}{\text{sec}} =$$
$$K_P \cdot 3.7417 \frac{1}{\text{meters}} \cdot \begin{bmatrix} .2673 \\ .5345 \\ .8018 \end{bmatrix} \cdot \text{meters} + K_D \cdot 8.7750 \frac{\text{sec}}{\text{meter}} \cdot \begin{bmatrix} .4558 \\ .5698 \\ .6838 \end{bmatrix} \cdot \frac{\text{meters}}{\text{sec}}$$

Notice the conversion factor for error and error velocity is incorporated into the control effort along with the norm. Thus the units cancel and the control effort results in the following.

$$u = K_P \cdot 3.7417 \cdot \begin{bmatrix} .2673 \\ .5345 \\ .8018 \end{bmatrix} + K_D \cdot 8.7750 \cdot \begin{bmatrix} .4558 \\ .5698 \\ .6838 \end{bmatrix}$$

$$u = \tilde{K}_p \cdot \begin{bmatrix} .2673 \\ .5345 \\ .8018 \end{bmatrix} + \tilde{K}_D \cdot \begin{bmatrix} .4558 \\ .5698 \\ .6838 \end{bmatrix}$$

Next, $\tilde{K}_P$ is factored out.

$$u = \tilde{K}_P \left(\begin{bmatrix} .2673 \\ .5345 \\ .8018 \end{bmatrix} + \frac{\tilde{K}_D}{\tilde{K}_P} \cdot \begin{bmatrix} .4558 \\ .5698 \\ .6838 \end{bmatrix} \right)$$

$$= \tilde{K}_P \left(\begin{bmatrix} .2673 \\ .5345 \\ .8018 \end{bmatrix} + T_D \cdot \begin{bmatrix} .4558 \\ .5698 \\ .6838 \end{bmatrix} \right)$$

Now, if the two gain's $\tilde{K}_P$ and $\tilde{K}_D$ were equal to each other, then $T_D = 1$ and the control effort would be

$$u = \tilde{K}_P \cdot \left(\begin{bmatrix} .2673 \\ .5345 \\ .8018 \end{bmatrix} + \begin{bmatrix} .4558 \\ .5698 \\ .6838 \end{bmatrix} \right) = u_{fuzzy} \cdot \left(\begin{bmatrix} .2673 \\ .5345 \\ .8018 \end{bmatrix} + \begin{bmatrix} .4558 \\ .5698 \\ .6838 \end{bmatrix} \right) = u_{fuzzy} \cdot \text{direction}$$

Therefore, it is possible to extend the SISO concept into a MIMO formulation assuming that the gains K_P and K_D are the same value, which is legitimate.

Is This Stable?

The next section addresses the issue of stability and robustness of the controller. The stability will be examined via the Liapunov stability criteria. For SAMPEX, the dynamics and control is given by

$$\dot{q} = \Xi(q) \cdot \omega$$

$$\mathrm{J}\dot{\omega} = \dot{\mathrm{H}} = (\omega \cdot \mathrm{x})\mathrm{H}$$

$$\mathrm{u} = -\mathrm{K}_{\mathrm{P}} \cdot \Xi^{\mathrm{T}}(\mathrm{q}_{\mathrm{ref}})\mathrm{q} - \mathrm{K}_{\mathrm{D}}(\omega - 0)$$

The following Liapunov function is chosen.

$$\mathrm{V} = \frac{1}{2}\omega^{\mathrm{T}} \cdot \mathrm{J} \cdot \omega + \frac{1}{2}\left(\mathrm{K}_{\mathrm{P}}(\mathrm{q} - \mathrm{q}_{\mathrm{ref}})^{\mathrm{T}}(\mathrm{q} - \mathrm{q}_{\mathrm{ref}})\right)$$

Assuming a pointing maneuver ($q_{ref} = 0$) and taking the derivative yields

$$\dot{\mathrm{V}} = \omega^{\mathrm{T}}\mathrm{J}\dot{\omega} + \frac{2}{2}\left(\mathrm{K}_{\mathrm{P}}(\mathrm{q} - \mathrm{q}_{\mathrm{ref}})^{\mathrm{T}}(\dot{\mathrm{q}} - 0)\right)$$

$$= \omega^{\mathrm{T}}\mathrm{J}\dot{\omega} + \mathrm{K}_{\mathrm{P}}(\mathrm{q} - \mathrm{q}_{\mathrm{ref}})^{\mathrm{T}}\dot{\mathrm{q}}$$

$$= \omega^{\mathrm{T}}\mathrm{J}\dot{\omega} + \mathrm{K}_{\mathrm{P}}\mathrm{q}^{\mathrm{T}}\dot{\mathrm{q}} - \mathrm{K}_{\mathrm{P}}\mathrm{q}_{\mathrm{ref}}^{\mathrm{T}}\dot{\mathrm{q}}$$

$$= \omega^{\mathrm{T}}\mathrm{J}\left\{\mathrm{J}^{-1}(\omega \cdot \mathrm{x})\mathrm{J}\omega - \mathrm{J}^{-1}\mathrm{K}_{\mathrm{P}}\Xi^{\mathrm{T}}(\mathrm{q}_{\mathrm{ref}})\mathrm{q} - \mathrm{J}^{-1}\mathrm{K}_{\mathrm{D}}\omega\right\} + \mathrm{K}_{\mathrm{P}}\mathrm{q}^{\mathrm{T}}\dot{\mathrm{q}} - \mathrm{K}_{\mathrm{P}}\mathrm{q}_{\mathrm{ref}}^{\mathrm{T}}\dot{\mathrm{q}}$$

$$= \omega^{\mathrm{T}}\cancel{\mathrm{J}\mathrm{J}^{-1}}(\omega \cdot \mathrm{x})\mathrm{J}\omega - \mathrm{K}_{\mathrm{P}} \cdot \omega^{\mathrm{T}}\cancel{\mathrm{J}\mathrm{J}^{-1}} \cdot \Xi^{\mathrm{T}}(\mathrm{q}_{\mathrm{ref}})\mathrm{q} - \omega^{\mathrm{T}}\cancel{\mathrm{J}\mathrm{J}^{-1}}\mathrm{K}_{\mathrm{D}}\omega + \mathrm{K}_{\mathrm{P}}\mathrm{q}^{\mathrm{T}}\dot{\mathrm{q}} - \mathrm{K}_{\mathrm{P}}\mathrm{q}_{\mathrm{ref}}^{\mathrm{T}}\dot{\mathrm{q}}$$

$$= \omega^{\mathrm{T}}(\omega \cdot \mathrm{x})\mathrm{J}\omega - \mathrm{K}_{\mathrm{P}} \cdot \omega^{\mathrm{T}}\Xi^{\mathrm{T}}(\mathrm{q}_{\mathrm{ref}})\mathrm{q} - \omega^{\mathrm{T}}\mathrm{K}_{\mathrm{D}}\omega + \mathrm{K}_{\mathrm{P}}\mathrm{q}^{\mathrm{T}}\dot{\mathrm{q}} - \mathrm{K}_{\mathrm{P}}\mathrm{q}_{\mathrm{ref}}^{\mathrm{T}}\dot{\mathrm{q}}$$

$$= \omega^{\mathrm{T}}(\omega \cdot \mathrm{x})\mathrm{J}\omega - \mathrm{K}_{\mathrm{P}} \cdot \omega^{\mathrm{T}}\Xi^{\mathrm{T}}(\mathrm{q}_{\mathrm{ref}})\mathrm{q} - \mathrm{K}_{\mathrm{D}}\omega^{\mathrm{T}}\omega + \mathrm{K}_{\mathrm{P}}\mathrm{q}^{\mathrm{T}}\Xi(\mathrm{q}) - \mathrm{K}_{\mathrm{P}}\mathrm{q}_{\mathrm{ref}}^{\mathrm{T}}\Xi(\mathrm{q})\omega$$

note

$$\omega^T(\omega \cdot x) = 0$$

$$q^T \Xi(q) = 0$$

$$\Xi^T(q) \cdot q_{ref} = -\Xi^T(q_{ref}) \cdot q$$

therefore

$$= 0 - K_P \cdot \omega^T \Xi^T(q_{ref})q - K_D \omega^T \omega + K_P q^T \Xi(q)\omega - K_P q_{ref}^T \Xi(q)\omega$$

$$= -K_P \cdot \omega^T \Xi^T(q_{ref})q - K_D \omega^T \omega + K_P q^T \Xi(q)\omega - K_P(\Xi^T(q)q_{ref})\omega$$

Now the last term can be transformed into

$$+K_P(\omega^T(\Xi^T(q)q_{ref}))^T$$

Therefore, we now have

$$= -K_P \cdot \omega^T \Xi^T(q_{ref})q - K_D \omega^T \omega + K_P \cdot 0 \cdot \omega + K_P(\omega^T(\Xi^T(q)q_{ref}))^T$$

$$= K_P(-\omega^T \Xi^T(q_{ref})q + (\omega^T(\Xi^T(q)q_{ref}))^T) - K_D \omega^T \omega$$

$$= -K_D \omega^T \omega$$

Therefore, the $\dot{V}$ term will always be negative, and thus stable, if K_D is always positive. Therefore, the system is stable if K_D ($\dot{V}$ = –) is positive and K_P is positive (V = +)

To determine the robustness of the controller, assume that the inertia matrix (J) for the satellite has some uncertainty defined by α. Therefore the inertia matrix will be

$$J = \alpha J$$

$$J\dot{\omega} = [\omega x]J\omega + u$$

$$\alpha J\dot{\omega} = [\omega x]\alpha J\omega + u$$

$$\dot{\omega} = \frac{1}{\alpha}J^{-1}[\omega x]J\omega + \frac{1}{\alpha}J^{-1}u$$

The control effort u is

$$u = -K_P I^T(q_{ref})q - K_D(\omega - 0)$$

The Liapunov function is given by

$$V = \frac{1}{2}\omega^T \alpha J \omega + \frac{1}{2}(K_P(q - q_{ref})^T(q - q_{ref}))$$

$$\dot{V} = \omega^T \alpha J \dot{\omega} + (K_P(q - q_{ref})^T(\dot{q} - \dot{q}_{ref}))$$

However q_{ref} dot is zero, therefore the equation simplifies to

$$\dot{V} = \omega^T \alpha J \dot{\omega} + K_P q^T \dot{q} - K_P q_{ref}^T \dot{q}$$

Now substituting in angular acceleration and the control effort gives

$$\dot{V} = \omega^T \alpha J(\frac{1}{\alpha} J^{-1}[\omega x]J\omega + \frac{1}{\alpha} J^{-1}(-K_P I^T(q_{ref})q - K_D(\omega - 0))) + K_P q^T \dot{q} - K_P q_{ref}^T \dot{q}$$

Upon simplification

$$\dot{V} = \omega^T[\omega x]J\omega - K_P \omega^T I^T(q_{ref}) - K_D \omega^T \omega + K_P q^T \Xi(q)\omega - K_P q_{ref}^T \Xi(q)\omega$$

but

$$[\omega x] = 0$$

Therefore $\dot{V}$ is not effected by α, but V is effected. However, V is always positive as long as K_P is positive and V dot is always negative as long as K_P and K_D are positive which were determined earlier to be the criteria needed for stability. This is easy to verify, since it was stated in the beginning that the magnitude of the error vector is sent into the fuzzy controller (always a positive value) and thus always a positive value will emerge from the fuzzy controller. Therefore, the system is stable and robust.

CHAPTER 5
RESULTS AND CONCLUSIONS

SAMPEX Results

Attitude control is an area of concern for a variety of applications ranging from autonomous vehicles to spacecraft. Accurate attitude control of spacecraft and other vehicles is crucial to the success of their mission. However, due to the high degree of non-linearity, it is very difficult to ensure performance or stability without a significant amount of effort and luck. This work provides a more accurate and robust control solution in addition to a methodology for developing controllers in a faster, more cost effective manner. Because of its flexibility or generic nature the attitude controller, which utilizes fuzzy logic and quaternion feedback, can be easily ported to other systems.

Throughout this section, various simulation results are presented comparing the generic fuzzy controller to a PD type nonlinear controller derived using Liapunov analysis. The states are the quaternions (q1, q2, q3, q4) and angular momentum (L1, L2, L3). The error states are the quaternion errors and the angular velocity errors (w1 – x-axis, w2 – y-axis, w3 – z-axis).

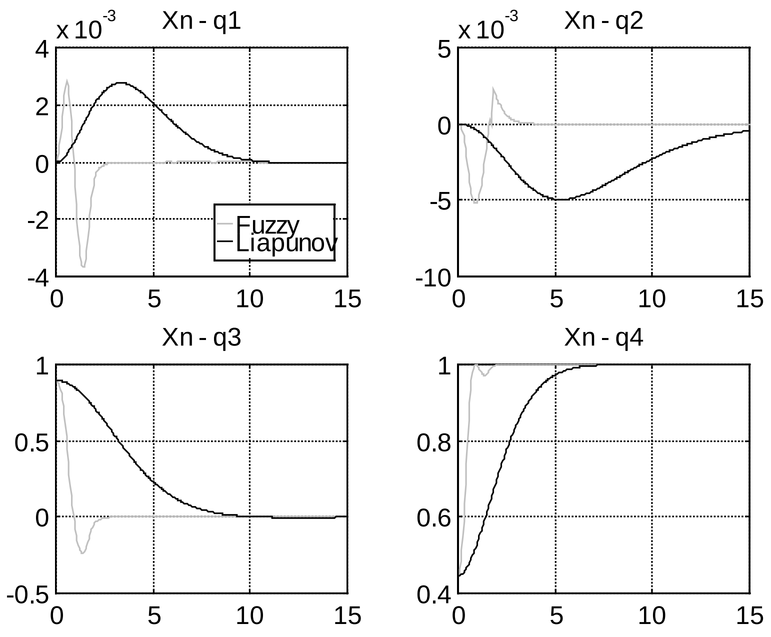
$$\text{Inputs:}\quad X = \begin{bmatrix} q \\ L \end{bmatrix} \qquad\qquad \text{Errors:}\quad X = \begin{bmatrix} e_q \\ \omega \end{bmatrix}$$

In order to test the robustness and flexibility of this scheme several test cases are examined: simple pointing maneuvers, disturbance, random noise, and tracking. Note that for most cases, initial conditions were set to: q = [0;0;2;1] L = [0;0;0] and desired were set to q = [0;0;0;1] and w = [0;0;0].

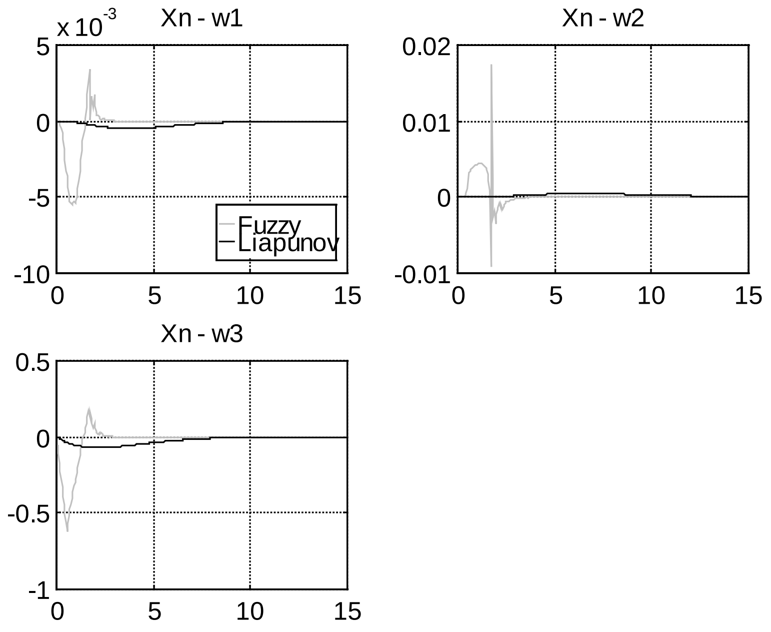
Basic Pointing Maneuver

The pointing maneuver, which is analogous to a regulator problem, is the simplest manuever to perform. Basically, the satellite is oriented (pointing) in one direction and then commanded to point in another direction. This is equivelent to a step response on the satellite. As shown in Figure 5-1, the fuzzy PD controller performs basically the same as the Liapunov based controller under ideal circumstances. Here the satellite was given the command to point in the quaternion direction [0;0;0;1] with angular velocities of [0;0;0]. Both of the controllers were capable of sending the system to the desired values. However, the fuzzy controller achieved the desired response quicker with greater overshoot, while the Liapunov controller produced a stiffer, slower response. Ultimately though, both controllers attained the desired orientation. Thus under ideal circumstances either controller could be used with satisfactory results.

The control efforts, as seen in Figure 5-2, of the two were very different. The fuzzy controller produced efforts that were one order of magnitude greater than those of the Liapunov controller. This however can be rectified to some extent by changing the range of the output MF to a range that is more acceptable. However, this would result in a slower settling times, etc.

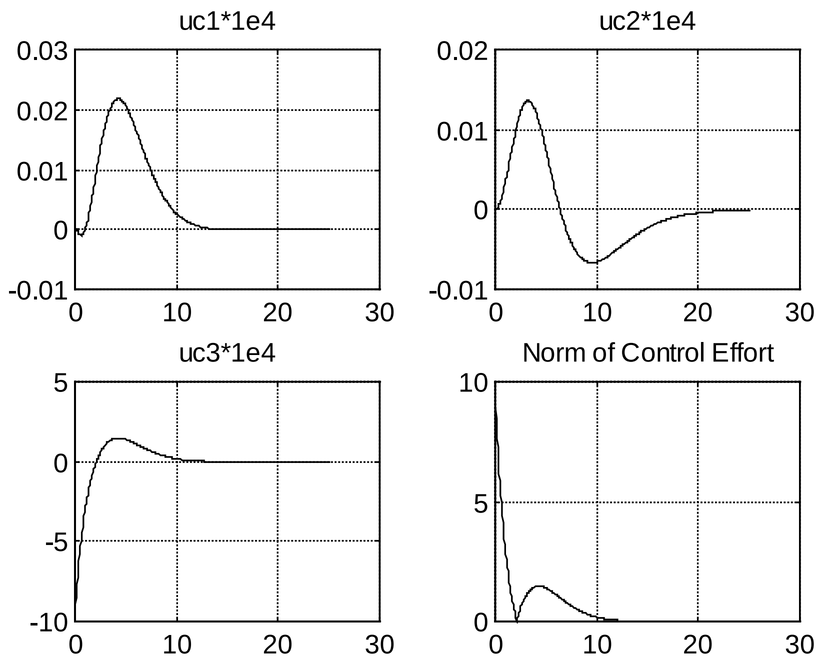


(a)

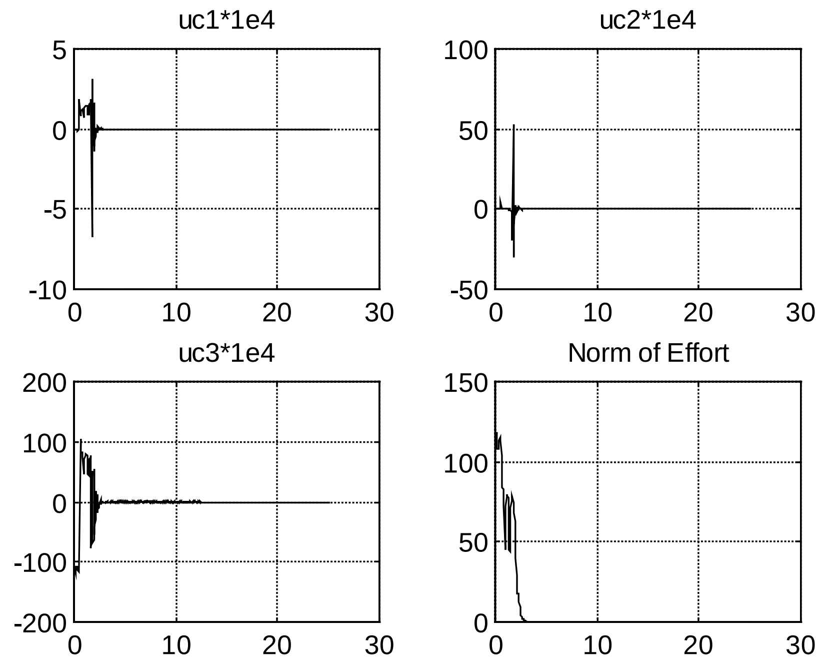


(b)

Figure 5-1. Quaternion values during a pointing manuver. The initial q=[0;0;2;1] L=[0;0;0] and the desired is q=[0;0;0;1] w=[0;0;0], (a) is the quaternions; (b) is the angular velocities.



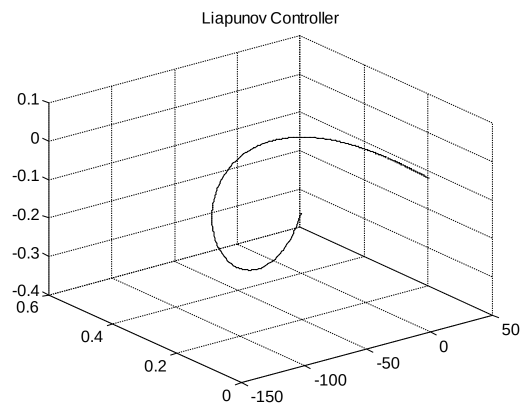
(a)

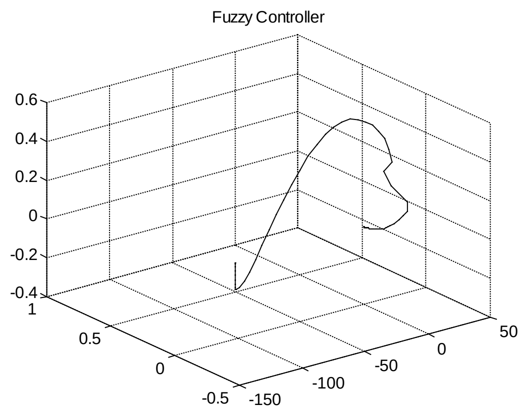


(b)

Figure 5-2. The control efforts of (a) Liapunov controller; (b) fuzzy controller

These differences in control effort can also be shown in the phase plots of both controllers, shown below. Optimal controls tells us that the most efficient method



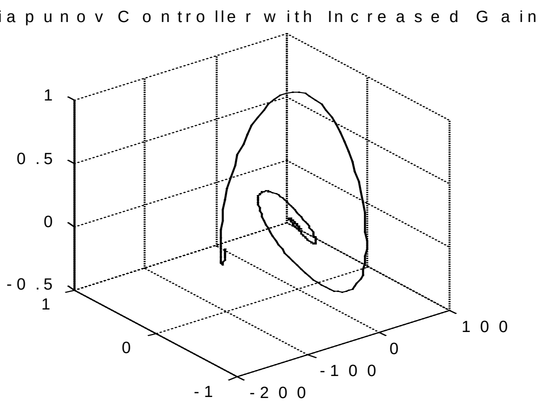


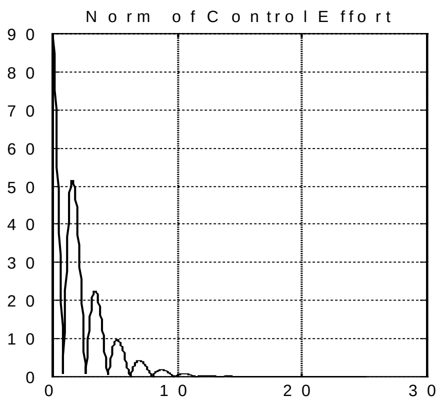
(a) (b)

Figure 5-3. Error Phase plots for the (a) Liapunov controller and; (b) fuzzy controller

to reduce the error would be an eigenaxis maneuver (straight line) in the phase plot. However, neither of these controllers produces that response. The Liapunov controller however does produce a smoother and smaller spiral to the desired states (zero error). This means that the Liapunov controller produces a more efficient control maneuver to get to the desired orientation. Also, notice the Liapunov controller produces a tighter spiral to the desired states, thus there is smaller percent overshoot than the fuzzy controller.

Since the fuzzy controller was capable of producing a higher controller effort, the next simulation was done to see what would happen if the Liapunov controller's proportional gain was increased by a factor of ten. The results are shown below.

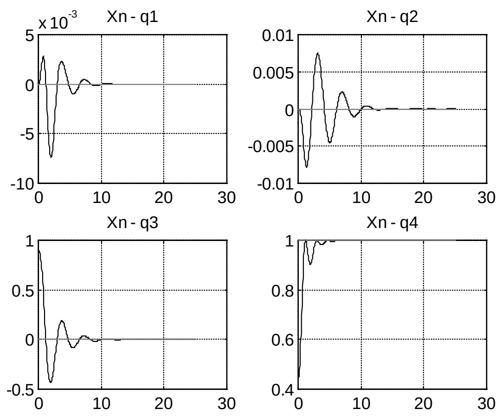




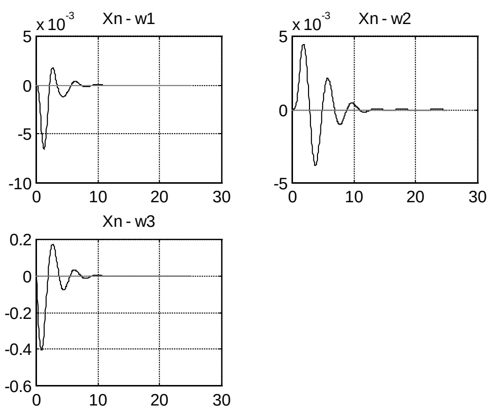
(a) (b)

Figure 5-4. The (a) phase plot; (b) control effort of the Liapunov controller when the proportional gain was increased by a factor of ten.

Notice how much bigger the spiral to zero error was compared to the original controller. Not only does the larger spiral mean that more energy was consumed controlling the satellite, but the spiraling through positive and negative values means that the satellite was wobbling through space during this maneuver. The states for the increased gain controller are shown below to illustrate the oscillatory response. The controller effort was in the same order of magnitude as that of the fuzzy controller, however the controller was unable to reduce the error any faster with the higher gain.



(a)



(b)

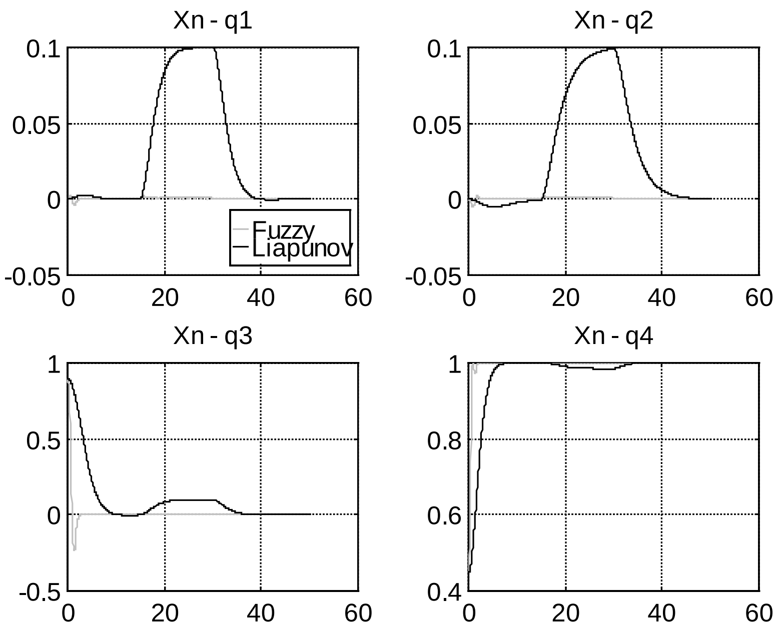
Figure 5-5. Liapunov controller with increased gains
(a) quaternions; (b) angular velocities

Effects of Disturbances

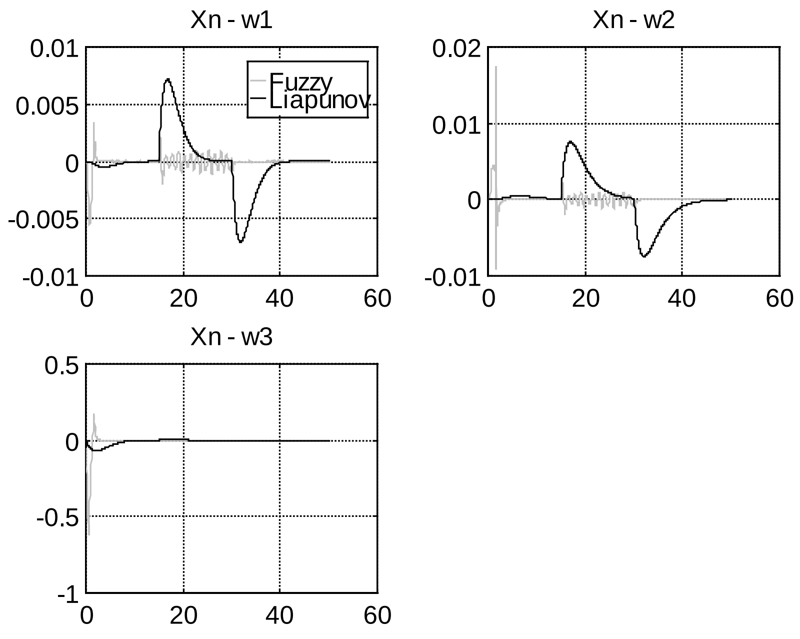
The universe is not an ideal place and things like uncertainties or disturbances are constantly present. Disturbances in space come in a variety of forms. Currently there are 6,650 known objects (debris) weighing approxiametly 3,000,000 kilograms orbiting earth. The debris is composed of paint flecks, refuse, waste, solid rocket effluent, etc. There are also millions of smaller (under 10 cm) untrackable objects with velocities on the order of 10 km/sec [21].

Other disturbances are due to magnetic flux, solar winds, solar radiation pressure torque (approxiatly 1 x 10 E-5 N-m). In addition, an aging space station Mir wobbling through space ready to smack an unsuspecting satellite, are all problems faced in space.

In the next simulation, a disturbance which is one magnitude greater than solar radiation pressure torque is applied to the satellite. Now, with the introduction of a disturbance (1 x 10 E-4 N-m for a short period of time), the results of the two controllers differs greatly from the previous section as shown below.



(a)



(b)

Figure 5-6. States when a disturbance ([.001;.001,.001]) is present where are the quaternions; (b) is the angular velocities; (c) control effort and initial conditions were: q = [0;0;2;1] L = [0;0;0]

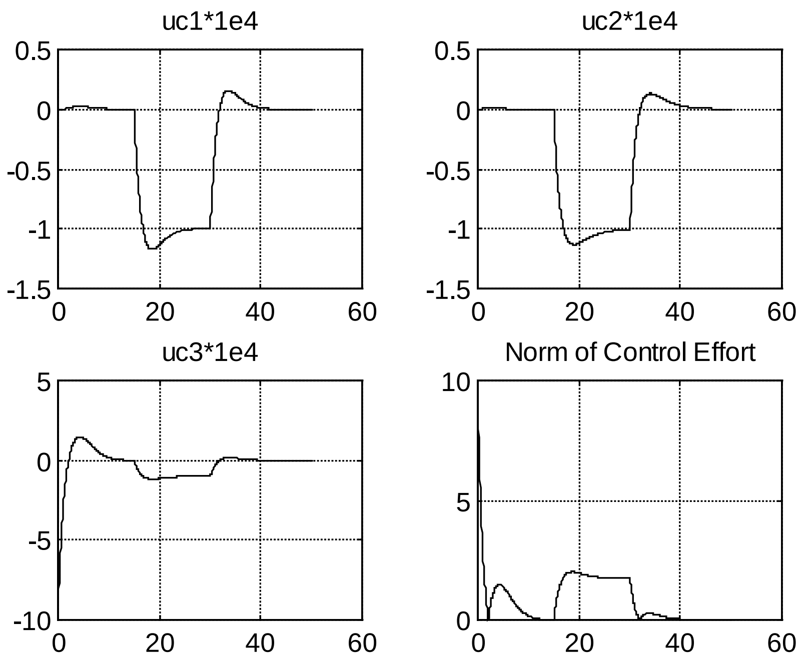
Because the Liapunov controller is a fixed gain controller, it had the tendency to significantly amplify and follow the disturbance. While the same disturbance given to the fuzzy controller however, did not produce this amplification. Only a small jittering effect could be notice in the quaternion and the angular velocities. In addition, one should note that the initial response before the disturbance is the same.

The control effort, Figure 5-7, again reveals that the fuzzy controller produced effort approximately one order of magnitude higher than the Liapunov controller. The control effort produced by the Liapunov controller was much smoother than that of the fuzzy controller.

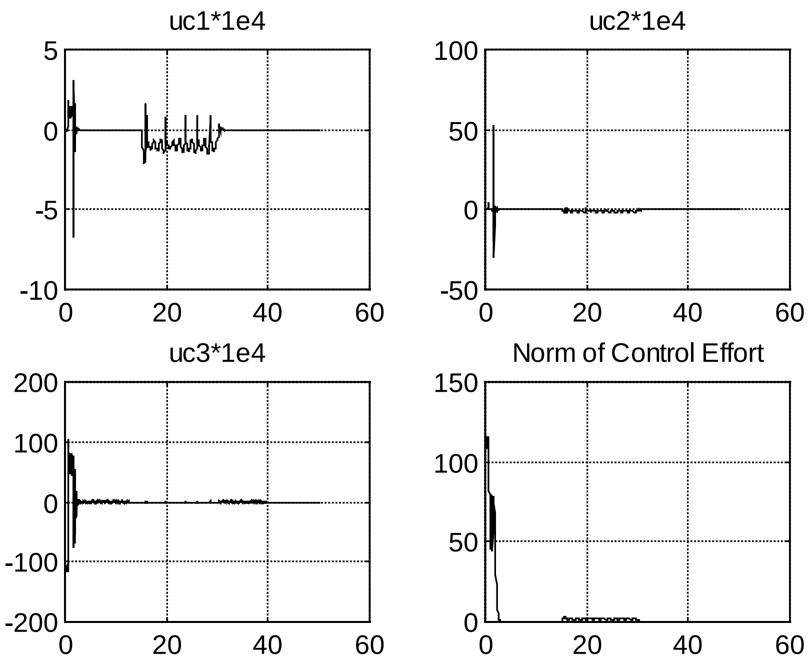
The phase plots, in Figure 5-8, look similar to the original pointing maneuvers performed without a disturbance (see Figure 5-3). However, notice how the disturbance effects the Liapunov controller more. There was greater energy wasted by the Liapunov controller than the fuzzy controller.

As before, the gain for the Liapunov controller was increased by a factor of ten in order to compare the two controllers another way. The phase plot and the control effort, Figure 5-9, again prove that increasing the gains of the Liapunov controller does not result in better performance. The increased gain produces a very oscillatory response, as shown by the spiraling path of the error curve.

The phase plot is shown again below for greater detail. The spacecraft starts on the left side of Figure 5-10 and proceeds to the desired value in the typical oscillatory fashion that has already been identified. Just after there was zero error in the system, the disturbance was injected into the system and the satellite spirals out to a new point defined by the disturbance. Once the disturbance was eliminated, the spacecraft spiraled back down to the desired point. Thus once again, even with higher control efforts the Liapunov controller does not improve the performance.

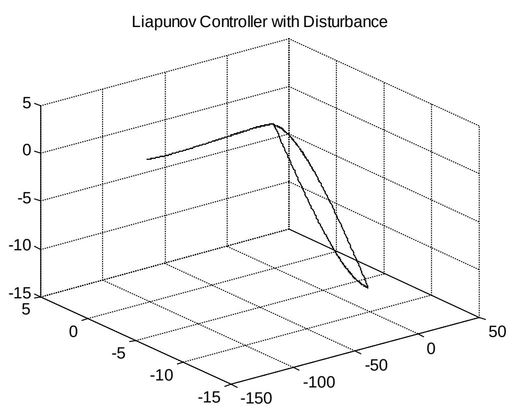


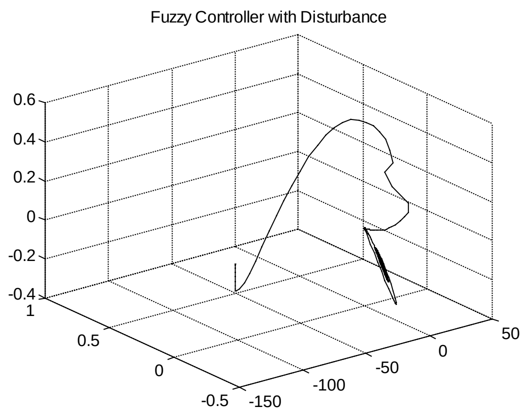
(a)



(b)

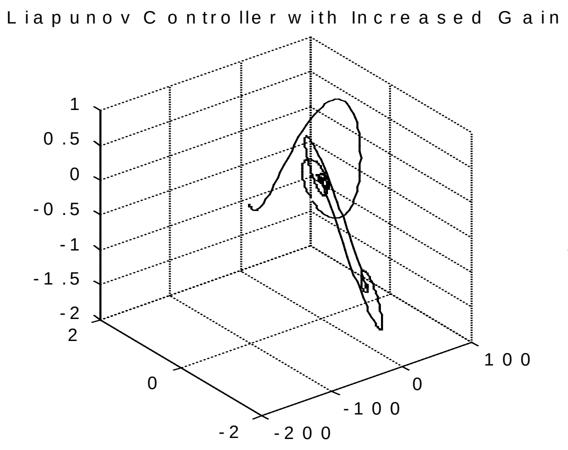
Figure 5-7. Control efforts for the (a) Liapunov controller; (b) the fuzzy controller

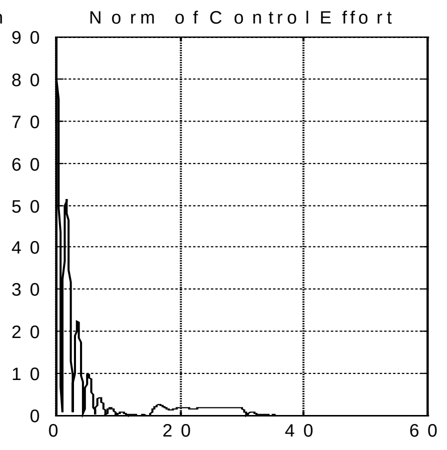




(a) (b)

Figure 5-8. Phase plots for the (a) Liapunov controller; (b) fuzzy controller





(a) (b)

Figure 5-9. The Liapunov controller with the gains increase by a factor of ten (a) phase plot; (b) control effort

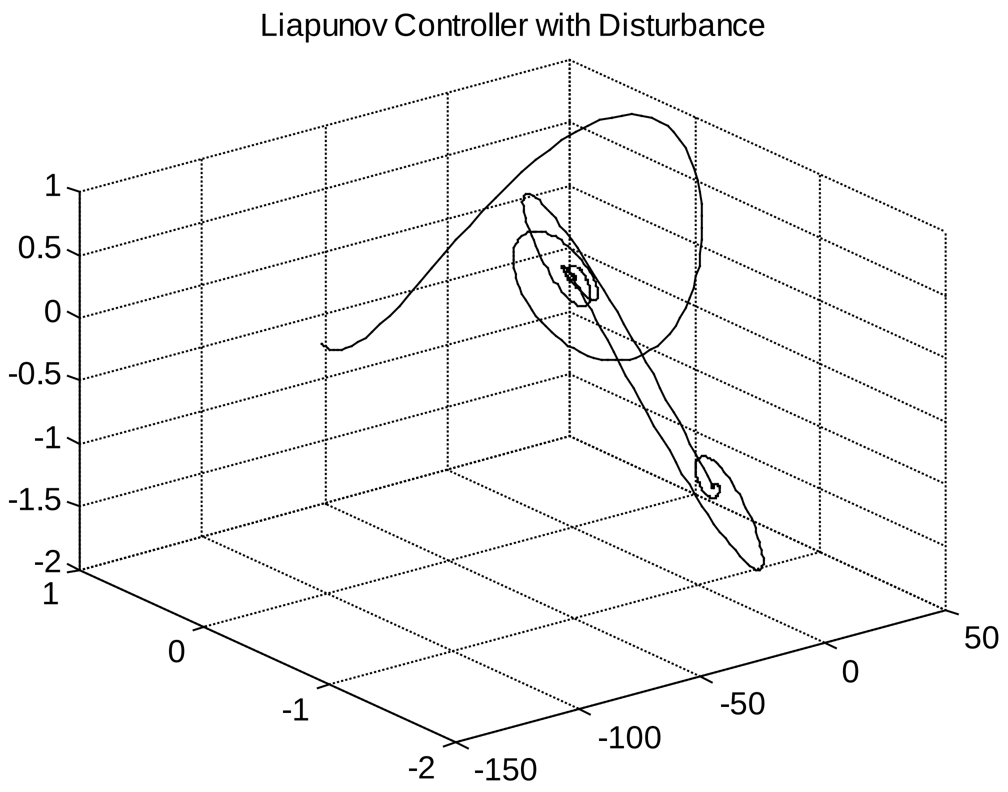
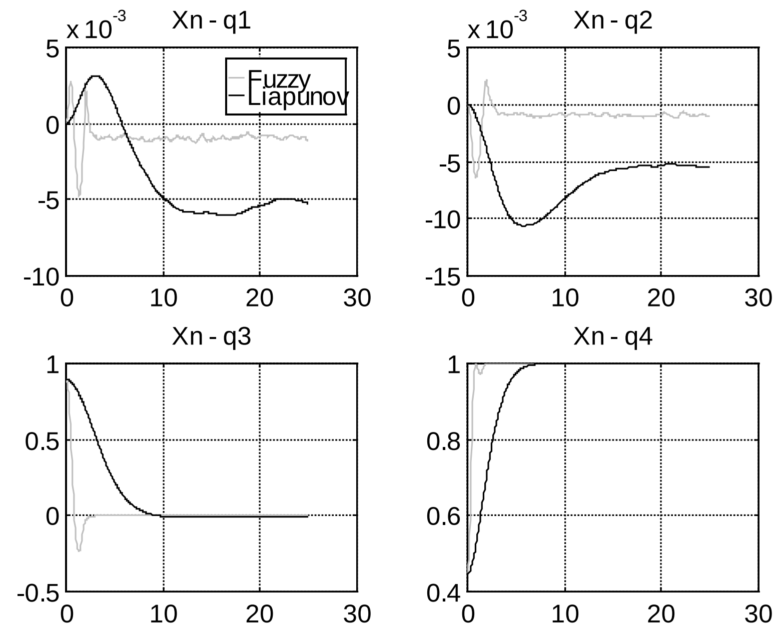


Figure 5-10. Liapunov controller's phase plot with increase gain

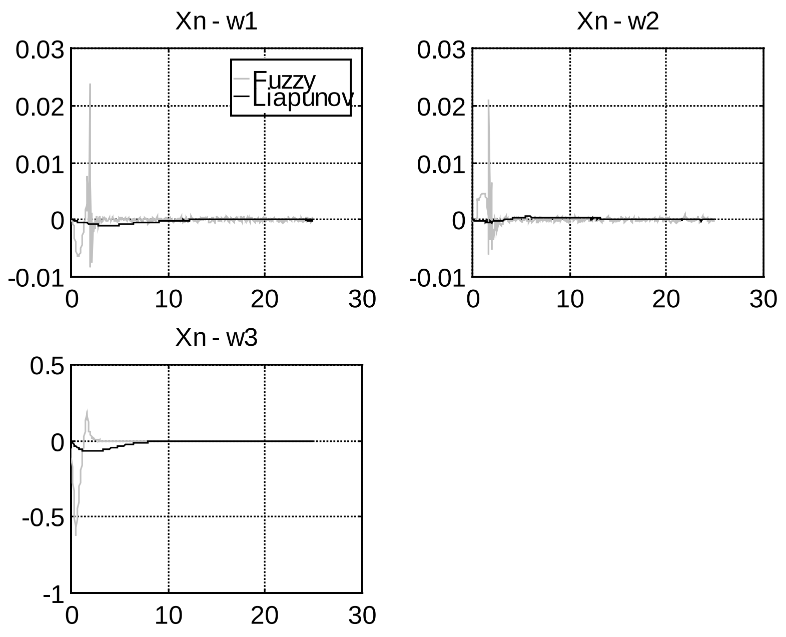
Effects of Random Noise

Noise is a problem with many aging systems. Satellites are exposed to extreme temperature changes as they rotate around the planet in gyrosynchronous orbit causing expansion and contraction of vital electrical and mechanical components. Thus the percision of sensors and equipment inside of a satellite deteriorates with time giving rise to increased noise seen in the system.

As before with the disturbance, the performance changes when random noise was introduced into the system. Random noise of N(0, σ = .001) was introduced into the system's sensors and the fuzzy controller produced better results than did the Liapunov controller. The results are shown below.



(a)



(b)

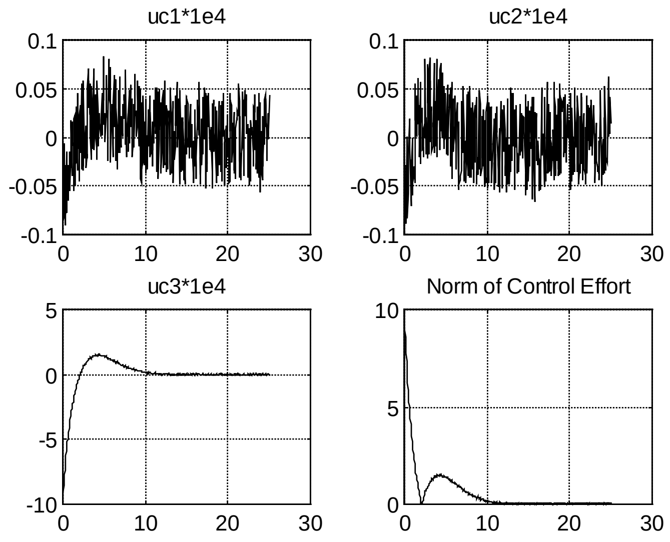
Figure 5-11. States when random noise is introduced where (a) are the quaternions and (b) are the angular velocites and initial conditions were: q=[0;0;2;1] L=[0;0;0]

Notice how the Liapunov controller oscillates in the quaternions when random noise is present. While the fuzzy controller only shows a small offset of .001 in q1 and q2.

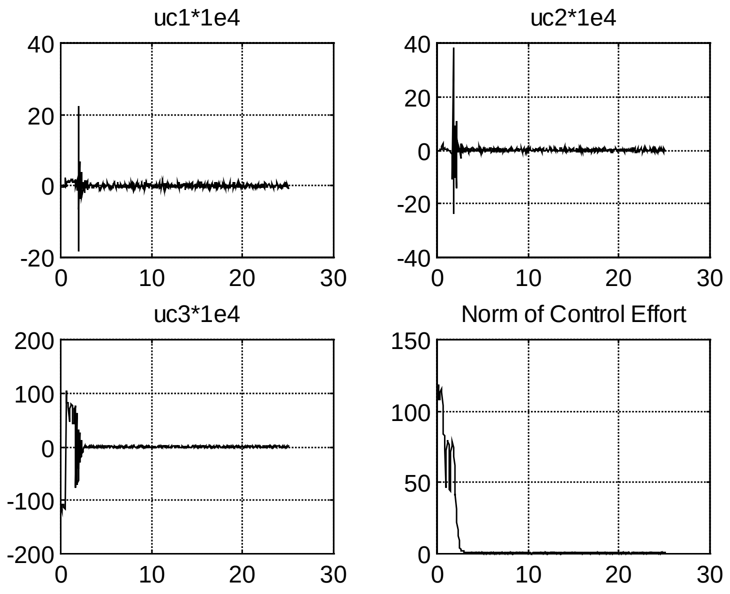
The control efforts for both the Liapunov and fuzzy controller are shown in Figure 5-12. Both controllers exhibit chatter once steady state is reached. The liapunov controller however produced a much smoother and lower control effort. The fuzzy controller produced higher and more erratic control effort.

The phase plots for the two controllers, shown in Figure 5-13, look very similar to the pointing manuevers without random noise intoduced (see Figure 5-3). As before, the Liapunov controller heads to the desired more directly, while the fuzzy controller arcs around more inefficiently. This inefficiency is not necessarily bad. This type of controller could result in quicker responses and in higher energy consumption.

The proportional gain for the Liapunov controller was again increased to see if there was a performance gain. Again the controller produced the same results with an oscillatory response. The error phase plot and control effort are shown in Figure 5-14 for the incrased gain.

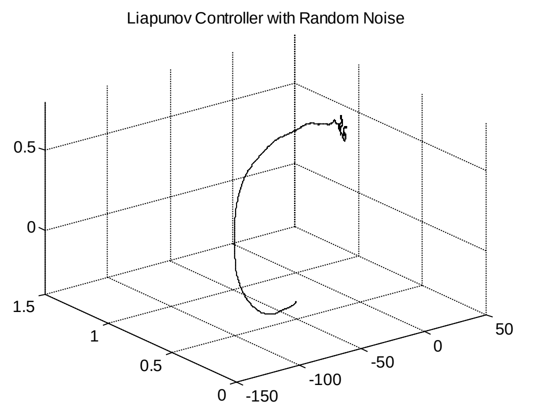


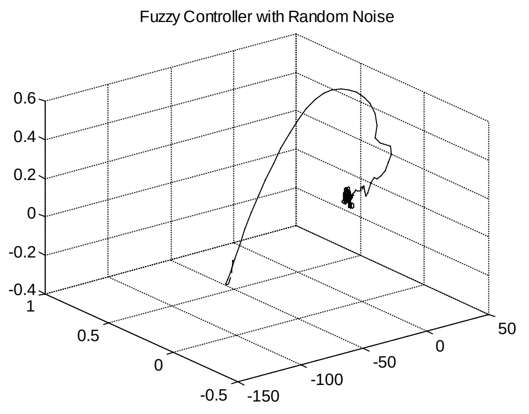
(a)



(b)

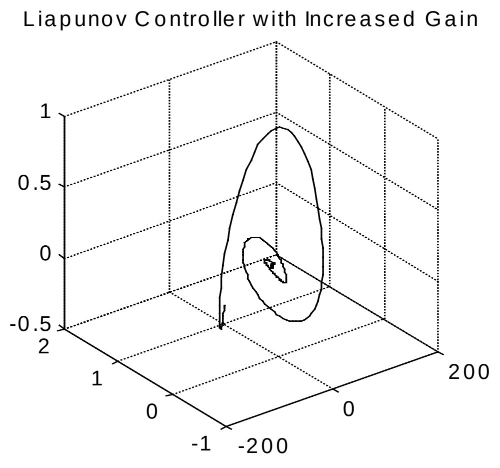
Figure 5-12. Control effort (u_c) for the three principle directions and their norm for (a) Liapunov; (b) Fuzzy

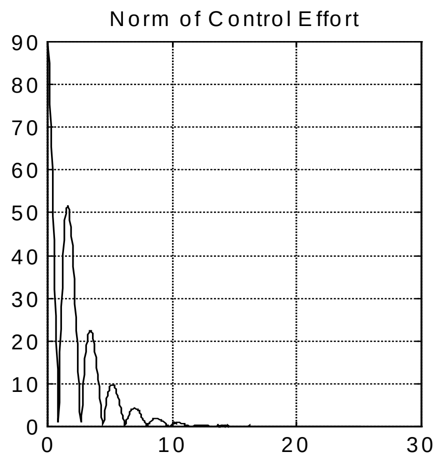




(a) (b)

Figure 5-13. Error phase portiates for (a) Liapunov controller; (b) Fuzzy controller





(a) (b)

Figure 5-14. Liapunov controller when the proportional gain in increased by a factor of 10.
(a) the phase plot; (b) the control effort

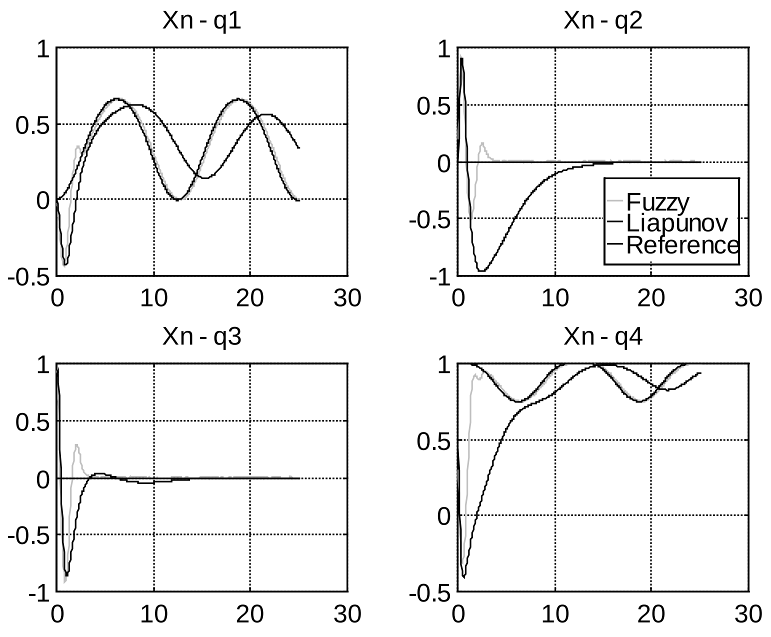
Tracking

Tracking is a more complex maneuver than pointing. This is because the desired quaternions are constantly moving (like a ramp input). Tracking is also difficult because of the build up of momentum that is produced as a satellite changes directions during the maneuver. In space, an uncontrollable build up of momentum can turn a multi-million-dollar satellite into very expensive space junk. Thus, a good controller can execute a complex maneuver, while gaining and dumping momentum as needed. If the controller is unable to perform these operations successfully, then momentum will pull the satellite off the desired trajectory or too much momentum will build up resulting in disaster.

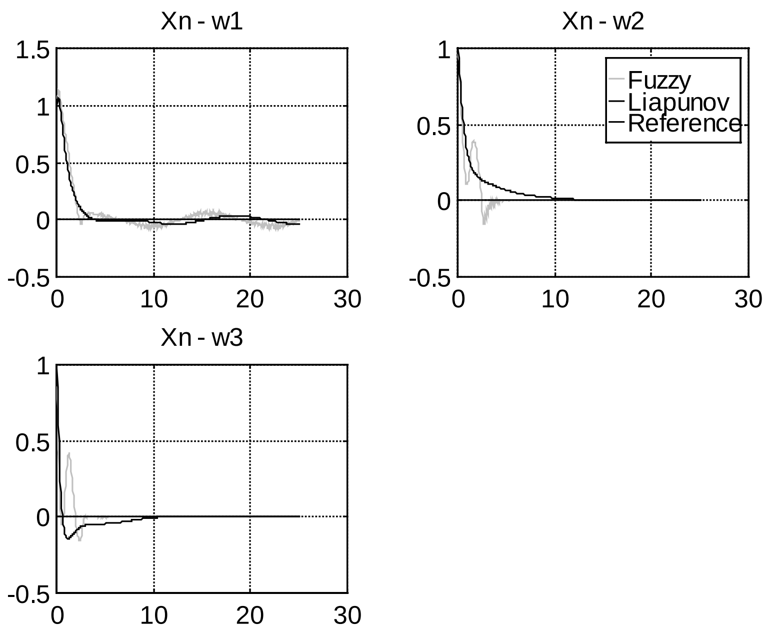
The final test case on the SAMPEX satellite was tracking a moving desired quaternion. The path that was chosen for SAMPEX to follow was just arbitrarily created. The q1 followed the sinusoidal path $q_{desired1} + .01 \cdot \sin\left(\frac{1}{250} \cdot ii \cdot 2\pi\right)$, where ii was the current loop iteration. This simulation was conducted with no noise or disturbances. The results are shown in Figure 5-15.

The Liapunov controller was unable to track sinusiodal change in the q1 and q4 while the fuzzy controller produced a reasonable tracking responce. Both the fuzzy and the Liapunov were capable of tracing q2 while both exibited a very small sinusoidal response in q3.

To demonstrate the abilities of the fuzzy controller in tracking further, sinusoids were added to both q1 and q2. The sinusoids were of the same form as was used in the previous tracking example. The results for both the fuzzy controller and the Liapunov controller are shown in Figure 5-16. Once again, the fuzzy controller was capable of performing the maneuver while the Liapunov controller was incapable.

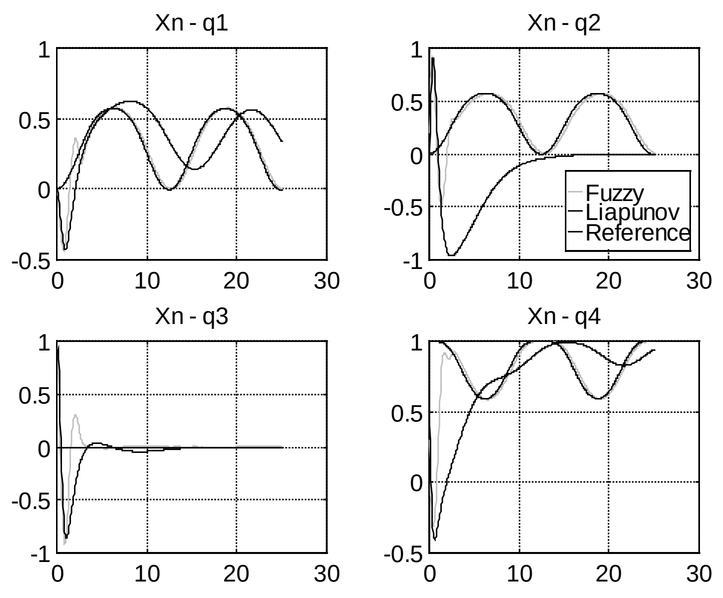


(a)

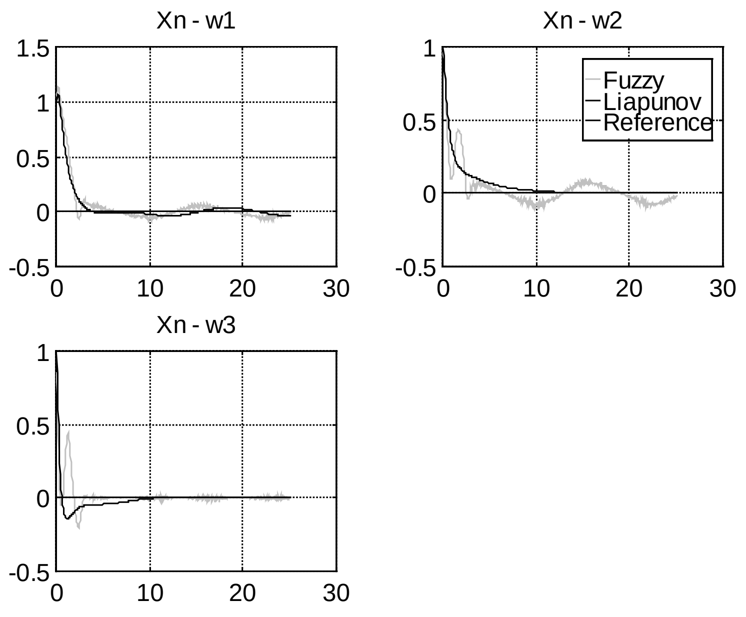


(b)

Figure 5-15. Controller's tracking a moving target in q1 where (a) are the quaternions and (b) are the angular velocities and initial condidtions were: q=[0;0;2;1] L=[0;0;0]



(a)



(b)

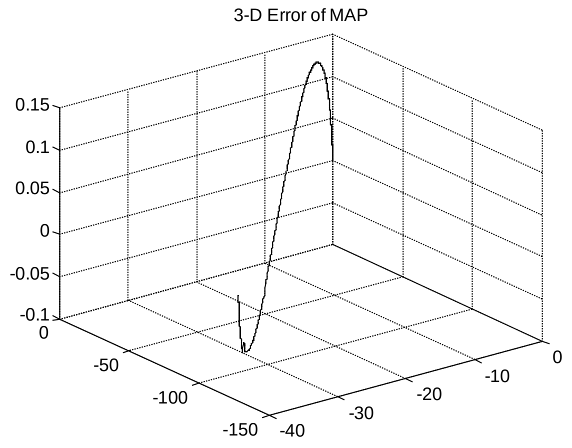
Figure 5-16. Tracking of two sinusoids in q1 and q2.
(a) quaternions; (b) angular velocities

The Liapunov controller again was unable to track this difficult maneuver again, while the fuzzy was able to perform with reasonable success.

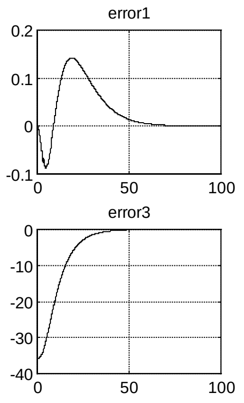
MAP Results

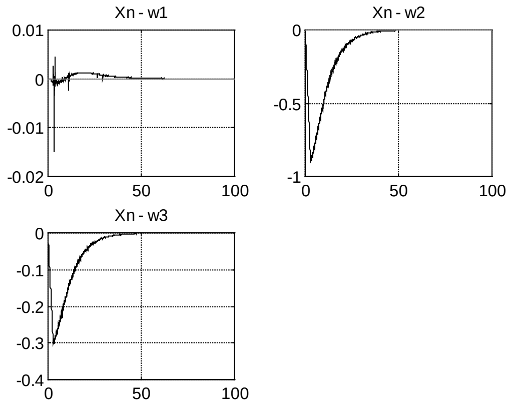
The final simulation illustraites the generic capabilities of the fuzzy controller. This is achieved by applying it to a different satellite. The only difference between MAP and SAMPEX is that the output membership function range was reduced by a factor of ten. Once again, the satellite was told to orient to q = [0, 0, 0, 1] and L = [0,0,0]. The fuzzy controller was able to satisfactorly accomplish this manuver, shown in Figure 5-17. A comparison was also made on the portability between fuzzy and the Liapunov controller. Figure 5-18 shows the results from applying the SAMPEX controller to the MAP satellite.

The outcome of this trial was intuitively obvious, meaning there was little to no chance that the unmodified Liapunov controller would achieve the desired response. Infact the Liapunov controller was so incompatable with MAP, the satellite went unstable. Thus this confirms that since Liapunov controllers use system dynamics to obtain stable responces, they can not be swapped between different systems with different dynamics without a significant amount of modification or redesign.



(a)





(b) (c)

Figure 5-17. (a)Error seen by the the fuzzy controller on the MAP satellite, (b) quaternions going to the desired [0, 0, 0, 1] values, and (c) angluar velocities going to the desred [0, 0, 0] values.

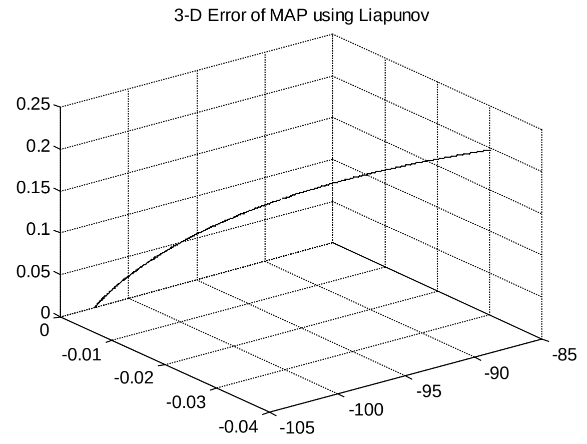


Figure 5-18. The MAP satellite with the sampex Liapunov controller

CHAPTER 6
CONCLUSIONS

Observations and Comparisons

Throughout the experimentation, the fuzzy controller produced better results than did the Liapunov controller. During simple pointing maneuvers, the fuzzy controller produced faster settling times than the Liapunov controller. Although the fuzzy controller performed better, there is currently no way to design for performance. The Liapunov controller is capable of being designed for specific performance criteria (settling time, percent overshoot, etc.); however, this process is very difficult.

The fuzzy controller produced better results in the presence of noise during pointing. The fuzzy controller also had better disturbance rejection characteristics then the Liapunov controller. The Liapunov controller could be tuned to perform better in the presence of noise and reject disturbances better, but this would constitute more design time and produce a more proprietary controller to that specific system and it's dynamics.

During complex tracking maneuvers, the fuzzy controller was able to follow the changing quaternions much better than did the Liapunov controller. This Liapunov controller would only be able to follow a simple, slowly moving path.

The fuzzy controller produced higher control efforts than did the Liapunov controller in all of the simulations. The Liapunov's proportional control gain was increased by a factor of ten so that its control effort range was similar to the fuzzy's control effort range. The resulting increased Liapunov gain controller produced oscillatory responses when during the pointing maneuvers. The performance (i.e. settling time, steady state error, etc.) of the Liapunov controller remained the same at the cost of higher control efforts rather than improve with the increased gain.

Finally, the controller was applied to the MAP satellite. Here the fuzzy controller was capable of controlling the new system while the Liapunov controller was unable too. This generic capability would enable the controller to be switched from satellite to satellite with only minor tuning to the output MF's. This controller would carry with it the resistance to noise, disturbance rejection capabilities and superior tracking characteristics. Thus design time required to construct control systems for pointing and tracking would be greatly reduced while maintaining performance.

Also, the generic capabilities would increase survivability of the spacecraft greatly. If any collision occurred (e.g. when a supply ship recently crashed into Mir and the whole station was almost lost) in which a significant amount of mass was broken off (assuming the control equipment was in tact and just some mission specific equipment was lost/damaged) this would only amount to a change in the inertia matrix. Which has already been shown not to effect the fuzzy controller much. Thus the spacecraft would still be partially operational and could possibly complete part of its mission rather than having been a complete loss.

Recommendations

This work provides the basis for generic fuzzy MIMO control of a satellite. Presented here were derivations showing stability and robustness of the controller.

Simulations demonstrated the fuzzy controller’s ability to over come noise, disturbances, and track complex maneuvers that its predecessor could not handle. The next steps that need to be taken are to incorporate learning into the controller. Currently an engineer must adjust input and output ranges of the MF’s. Ideally, the controller would automatically be able to sense the level of inputs it sees and adjust the MF’s for input/output automatically. This would not only simplify swapping the controller between satellites, but also aid in survivability of catastrophic events (as already mentioned).

REFERENCES

[1] Mason, P., and Walchko, K., “Fuzzy Gain Scheduling for Satellite Attitude Control,” to be published in the *Proceedings of the Flight Mechanics/Estimation Theory Symposium*, Goddard Space Flight Center, Greenbelt, MD, 1999.

[2] Walchko, K., Petroff, N., Mason, P.A.C., and Fitz-Coy, N., "Development of a Generic Hybrid Fuzzy Controller," *Intelligent Engineering Systems Through Artificial Neural Networks,* Vol. 8, 1998, pp. 214-218, St. Louis, MS.

[3] Brehm, T., and Kuldip, S. R., “Hybrid Fuzzy Logic PID Controller,” IEEE International Conference on Fuzzy Systems, Vol. 3, 1994, pp. 1682-1687.

[4] Brehm, T., and Kuldip, S. R., “Proportional Fuzzy Logic Controller: Classical Proportional-Plus-Derivative Like Response,” IEEE International Conference on Systems, Man, and Cybernetics, Vol. 3, 1995, pp. 2029-2033, Piscataway, NJ.

[5] Gonsalves, P. G., and Caglayan, A. K., “Fuzzy Logic PID Controller for Missile Terminal Guidance,” IEEE International Symposium on Intelligent Control - Proceedings, Vol. 1, 1995, pp. 377-382.

[6] Misir, D., Malki, H. A., and Chen, G., “Design and Analysis of a Fuzzy Proportional-Integral-Derivative Controller,” Fuzzy Sets and Systems, Vol. 79, 1996, pp. 297-314.

[7] Wie, B., and Barba, P., “Quaternion Feedback for Spacecraft Large Angle Maneuvers,” Journal of Guidance, Control and Dynamics, Vol. 8, No. 3, May-June 1985, pp. 360-365.

[8] Crassidis, J.L., and Markley, F.L., “Predictive Filtering for Attitude Estimation Without Rate Sensors,” Journal of Guidance, Control and Dynamics, Vol. 20, No. 3, May-June 1997, pp. 522-527.

[9] Pena, J. D., Crassidis, J. L., McPartland, M. D., Meyer, T. J., and Mook, D. J., “MME-Based Attitude Dynamics Identification and Estimation for SAMPEX,” Proceedings of the Flight Mechanics/Estimation Theory Symposium, Goddard Space Flight Center, Greenbelt MD., 1994, pp.497-511.

[10] Woodard, M., “Fuzzy Open-Loop Attitude Control for the FAST Spacecraft,” http://fdd.gsfc.nasa.gov/mwoodard/aiaa_96/aiaa_96.html.

[11] Buijtenen, W. M., Schram, G., Babuska, R., and Verbruggen, H. B., “Adaptive Fuzzy Control of Satellite Attitude by Reinforcement Learning,” IEEE Transactions on Fuzzy Systems, Vol. 6, No. 2, May 1998, pp. 185-194.

[12] Dorf, R., Bishop, R., Modern Control Systems, 8th Ed. Menlo Park, CA: Addison Wesley, 1998.

[13] Ogata, K., Modern Control Engineering, 3rd Ed. Upper Saddle River, NJ: Prentice-Hall, 1997.

[14] Kosko, Bart. Fuzzy Engineering. New Jersey: Prentice Hall, 1997.

[15] Petroff, N., Walchko, K., Mason, P.A.C., Reisinger, K.D., "Numerical Stability Analysis of a Fuzzy Controller", *Intelligent Engineering Systems Through Artificial Neural Networks,* Vol. 8, 1998, pp. 219-224, St. Louis, MS.

[16] Gulley, N., and Roger Jang, J. S. Fuzzy Logic Toolbox for use with Matlab®. Nortick, MA: The MathWorks, Inc., 1995.

[17] Wertz, James R., Spacecraft Attitude Determination and Control, Boston: Kluwer Academic Publishers, 1995.

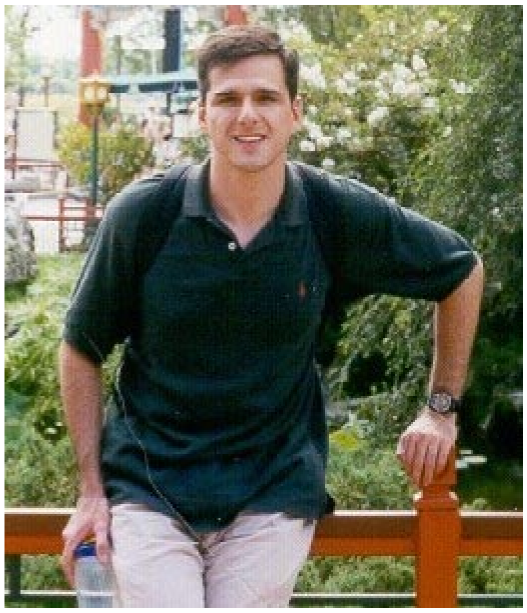
[18] Crane, C., and Duffy, J., Kinematic Analysis of Robot Manipulators, Cambridge University Press, 1998.

[19] Baker, D. N., Mason, G. M., Figueroa, O., Colon, G., Watzin, J. G., and Aleman, R. M.. “An Overview of the Solar, Anomalous, and Magnetospheric Particle Explorer (SAMPEX) Mission,” IEEE Transactions on Geoscience and Remote Sensing. Vol. 31 No. 3, May 1993.

[20] Flatley, Thomas W., Forden, Josephine K., Henretty, Debra A., Lightsey, E. Glenn, Markley, F. Landis, “MME-Based Attitude Dynamics Identification and Estimation for SAMPEX,” Proceedings of the Flight Mechanics/Estimation Theory Symposium, Goddard Space Flight Center, Greenbelt, MD, 1990, pp.497-511

[21] Culp, R. D., “Orbital Debris,” *14th Annual AAS Guidance and Control Conference*, Keystone, Colorado, February 2-6, 1991.

BIOGRAPHICAL SKETCH



Kevin J. Walchko was born on 1 Nov. 1972 in Sarasota, FL. He attended the University of Florida from 1991-1997 and received a B.S. degree in mechanical engineering. Most of the research Kevin has done was in the areas of controls and dynamics. Specifically, Kevin has focused on intelligent controls such as fuzzy logic and neural networks. Kevin will receive his master's degree in Spring of 1999, and will stay at the University of Florida to complete a Ph.D. degree in mechanical engineering. Kevin is also currently working concurrently on a masters degree in electrical engineering.