A COMPARATIVE STUDY OF VARIOUS APPROACHES TO THE DESIGN OF ON-OFF THRUSTER ATTITUDE CONTROL SYSTEMS

by

David Marc Weisenberg

SUBMITTED TO THE DEPARTMENT OF
MECHANICAL ENGINEERING IN PARTIAL
FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREES OF

BACHELOR OF SCIENCE

AND

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1987

Signature of Author ______________________
Department of Mechanical Engineering
June 1, 1987

Certified by ______
Derek Rowell
Thesis Supervisor

Accepted by ______________________
Ain A. Sonin
Chairman, Graduate Committee

ENGINEERING INTERNSHIP PROGRAM
SCHOOL OF ENGINEERING
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Cambridge, Mass. 02139

THESIS REVIEW LETTER

Attention: John R. Martuccelli, Director, Room 1-211

Subject: Master's Thesis of David Weisenberg
(student name)

The attached thesis, A Comparative Study of Various Approaches to the Design of On-Off (Title) Thruster Attitude Control Systems

has been reviewed by the undersigned Martin Marietta Denver Aerospace
(company name)

representatives and confirmed that it does not contain details objectionable

from the standpoint of Martin Marietta Denver Aerospace.
(company name)

In addition, we understand that the aforementioned thesis report becomes the permanent property of M.I.T., will be placed in the M.I.T. Library within one month of the date of submission and may not be published wholly or in part

except by authorization of the Department of Mechanical Engineering,
(department name)

in which the student is enrolled. (It is understood that authorization is granted to the Company for such limited publication as the Company's prior contractual obligations may require.)

Martin Marietta, Denver		
Company Name	Direct Company Supervisor of Student	Student Name
Approved By (for company)		
Title:		
Date: 4/20/87	Date: 4/18/87	Date: 4/7/87

A COMPARATIVE STUDY OF VARIOUS APPROACHES TO THE DESIGN OF ON-OFF THRUSTER ATTITUDE CONTROL SYSTEMS

by

David Marc Weisenberg

Submitted to the Department of Mechanical Engineering on June 1, 1987 in partial fulfillment of the requirements for the degrees of Bachelor of Science and Master of Science in Mechanical Engineering.

Abstract

Many different systems for mass expulsion (thruster) control of spacecraft attitude have been proposed in the literature. However, there is a lack of any sort of comparative analysis of the relative strengths and weaknesses of these different systems. This comparative analysis is the core of this thesis.

Three different approaches to the control task are possible -- the phase plane approach, in which most of the system complexity is in the phase plane switching lines, the controller approach, in which a simple deadband is used for a phase plane, and the variable pulse width approach, in which no phase plane is used and thruster pulse widths vary according to the current state. A search through the literature was performed, and several representative system designs were chosen in each category. These designs were incorporated into a detailed computer simulation of the Magellan spacecraft which was developed at Martin Marietta Denver Aerospace. Using both a high external disturbance and a low disturbance case, the performance of the various designs was observed and compared.

As system designs were tried and adjusted, some simple design considerations, such as the "approach parabola," the "steady state fuel bump," and the elimination of two-sided deadbands were found to greatly facilitate the optimization of parameter values for a particular system. These considerations were found nowhere in the literature, but would be of great help to an attitude control system designer. An example of a real problem encountered with the Magellan spacecraft is given (Chapter 8) to show the utility of these concepts.

Designs utilizing the phase plane approach were found to be easier to design and optimize because of their simple and intuitive geometric nature. Furthermore, they are easy to tailor with specialized switching lines, thus causing exceptional performance in the low disturbance case and reasonably good performance in the high disturbance case. On the other hand, controller designs were not nearly as easy and intuitive, but pseudo-rate and integral controller terms resulted in excellent performance for high disturbances. Finally, it was difficult to generalize much about the variable pulse width designs. The only obvious generalization is that, because of unavoidable system nonlinearities, they tend to be very jittery in steady state, and thus they cause a high rate of steady state fuel consumption.

Thesis Supervisor:	Derek Rowell
Title:	Associate Professor of Mechanical Engineering

List of Symbols

a	length of the fuel slosh pendulum
A	RLC design parameter. -1/A is the slope of the RLC slanted deadband.
A1,A2,A3,A4	Magellan phase plane attitude ledges
b	1/(pseudo-rate time constant). $1/b_1$ is the charging time constant, while $1/b_2$ is the discharging time constant.
B,C	subscripts indicating body or control coordinates
d	distance from center of propellant tank to spacecraft center of gravity
E	effective torque on the spacecraft resulting from fuel slosh effects
f	limit cycle frequency
f_1,f_2	propellant slosh natural frequencies
F	thrust or force. Also the volume fraction of the fuel tank that is filled with propellant.
F_N	normalized rate of steady state fuel consumption
F_{side}	SRM side force
F_{SRM}	SRM thrust vector
F_{ss}	steady state rate of fuel consumption
G	spacecraft acceleration
G(s)	transfer function used in Galileo controller describing function analysis
H	Hamiltonian
I	spacecraft inertia
I_f	propellant inertia
I_{old}	previous value of Galileo controller integral term
I_{sp}	specific impulse of a thruster

$I_{sp\text{-}SRM}$	SRM specific impulse
J	weighted time-fuel performance index
k	parabola constant in Magellan phase plane
k_6	discrete time analog of k_a
k_7	discrete time analog of k_b
k_{12}, k_{13}	discrete time pseudo-rate gains
k_a	Galileo controller describing function rate gain
k_b	Galileo controller describing function integral gain
k_c	Galileo controller describing function acceleration gain
k_d	Galileo controller actual pseudo-rate gain
k_w	time weighting factor in one particular weighted time-fuel index
L	Lagrangian
L_m	SRM translational misalignment
m	mass of liquid propellant in propellant tank
$\dot{m}$	rate of mass expulsion from the SRM
m_f	mass of liquid propellant consumed since the start of the mission
M	Clark system gain. Also mass of the spacecraft.
N	Galileo controller describing function
N_1, N_2	dummy slosh state variables
p_1, p_2	costate variables
P	propellant tank inlet pressure
P_{old}	previous value of Galileo controller pseudo-rate term
Q_i	thruster thrust level coefficient
r	jet damping moment arm

R	radius of propellant tank
R1,R2,R3	Magellan phase plane rate ledges
R_{mslgn}	misalignment torque lever arm
R_{sf}	SRM side force lever arm
s	Laplace transform operator
S_i	thruster thrust level coefficient
t	time
t_f, t_{final}	final settling time of the system
t_p, t_{pulse}	thruster pulse time
t_{sample}, T	sampling interval
T^*	kinetic coenergy
$T^*_{s/c}$	spacecraft kinetic coenergy
T^*_{fuel}	propellant kinetic coenergy
u	control input to the thrusters. In some formulations, $u = +1$, 0, or -1. In others, $u = +\tau$, 0, or $-\tau$
v_{fuel}	velocity of propellant
V	potential energy
x,y,z	(sometimes 1,2,3 is used) subscript indicating x, y, or z axis
X	amplitude of input sinusoid for Galileo controller describing function analysis
$y(t)$	Galileo controller describing function output
Y_1	first harmonic of $y(t)$
Y_i	thruster thrust level coefficient
z_{cm}	distance from pre-VOI spacecraft center of mass to current spacecraft center of mass
z_{int}	distance from SRM exit plane to spacecraft center of gravity

z_{side}	distance from application point of SRM side forces to pre-VOI spacecraft center of gravity
α	rate signal filter bandwidth
α_s	angle between the spacecraft body y-axis and the SRM side force vector
Γ	constant torque capacity of the thrusters
δ_m	SRM angular misalignment
δW	variational work
ζ	slosh damping coefficient
θ	propellant slosh angle
λ	time weighting factor in a certain weighted time-fuel index
ν_1, ν_2	Pseudo-Open Loop system optimal control coefficients
Ξ	generalized nonconservative force
τ	torque
τ_{JD}	jet damping moment
τ_{psa}	Magellan pseudo-acceleration time constant
τ_{swirl}	SRM swirl torque
ϕ	attitude error
ϕ_0	initial attitude
ϕ_{db}	deadband angle
$\phi_{desired}$	desired attitude (usually 0)
ϕ_{lc-a}	maximum allowable limit cycle attitude
ϕ_{max}	RLC maximum angle
ϕ_{max-a}	maximum allowable attitude error for Normal Attitude Control
ϕ_{max-o}	maximum observed attitude error for Normal Attitude Control
ϕ_N	maximum observed limit cycle attitude

ϕ_r	RLC design parameter
$\dot{\phi}$	rate error
$\dot{\phi}_0$	initial rate
$\dot{\phi}_1$	maximum limit cycle rate
$\dot{\phi}_2$	minimum limit cycle rate
$\dot{\phi}_{in}$	input to the rate filter
$\dot{\phi}_l$	idealized RLC rate ledge
$\dot{\phi}_{lc-a}$	maximum allowable limit cycle rate
$\dot{\phi}_{ledge}$	minimum RLC rate ledge limit
$\dot{\phi}_{lim}$	maximum RLC rate ledge limit
$\dot{\phi}_{max-a}$	maximum allowable rate error for Normal Attitude Control
$\dot{\phi}_{max-o}$	maximum observed rate error for Normal Attitude Control
$\dot{\phi}_N$	normalized limit cycle rate amplitude
$\dot{\phi}_{out}$	output of the rate filter
$\ddot{\phi}$	acceleration
ψ	optimal control terminal constraints
ω	angular velocity. Also sinusoid frequency for Galileo controller describing function analysis

List of Acronyms

AACS	Attitude And Articulation Control System
ACSL	Advanced Continuous Simulations Language
MGN	Magellan
PD	Proportional-Derivative
PID	Proportional-Integral-Derivative
PSA	Pseudo-Acceleration
PSR	Pseudo-Rate
PWPF	Pulse-Width-Pulse-Frequency
RLC	Rate-Ledge Controller
SRM	Solid Rocket Motor
VOI	Venus Orbital Insertion

Acknowledgments

I would like thank everyone who works in the Magellan Attitude and Articulation Control Analysis group at Martin Marietta Denver Aerospace for their help with this thesis. Two of these analysts, Charles Gay and Steven Johnson, deserve special thanks for tolerating my constant questions about FPSIM, the spacecraft simulation program. Finally, I am most indebted to Dr. Whittak Huang, whose advice and assistance made this thesis possible.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Many modern spacecraft have very tight pointing requirements over at least a part of their missions. For example, the Magellan spacecraft, which is to be launched to Venus in 1989, must be designed to maintain its attitude to within about one-twentieth of a degree. Most spacecraft take advantage of the high accuracy obtainable from reaction wheels or control moment gyros to meet these strict requirements. However, these devices generally have very limited torquing capabilities. They function well when the total spacecraft momentum is not increasing rapidly, but the presence of large external disturbances would quickly saturate them. Their limited torques also mean that they are very slow to correct large attitude errors. Finally, these devices operate by exchanging momentum with the spacecraft; when they themselves have built up too much momentum, another device must be used to desaturate them. Thus, these devices alone can not control spacecraft attitude. On-off thrusters are almost universally used to maintain control in the presence of large disturbances, to correct large attitude errors, and to occasionally desaturate reaction wheels or control moment gyros. The system used to determine the behavior of the thrusters is thus one of the most important elements of a spacecraft.

Figures 1-1 and 1-2 depict a generic thruster attitude control system. The physical configuration of the thrusters is shown in Figure 1-1. The thrusters are mounted in opposing pairs such that they deliver torques about three mutually perpendicular axes. Depending on which thruster of a pair is fired, either a positive or a negative torque can be delivered about each axis.[1] Figure 1-2 shows a block diagram of the system used to control

---

[1] Although schemes involving gimballed thrusters have been proposed, they will not be discussed in this thesis.

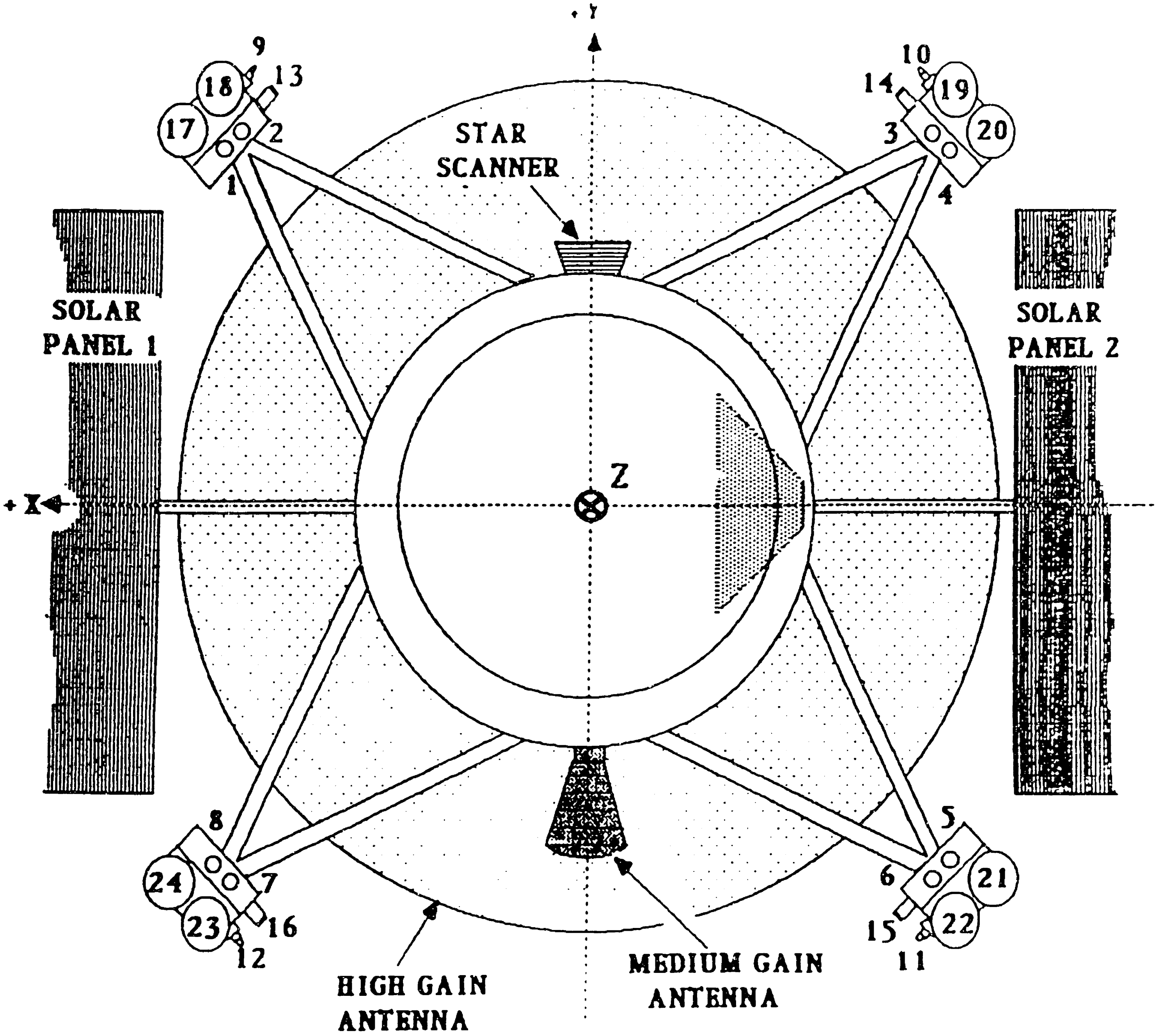


Figure 1-1: Thruster Configuration on the Magellan

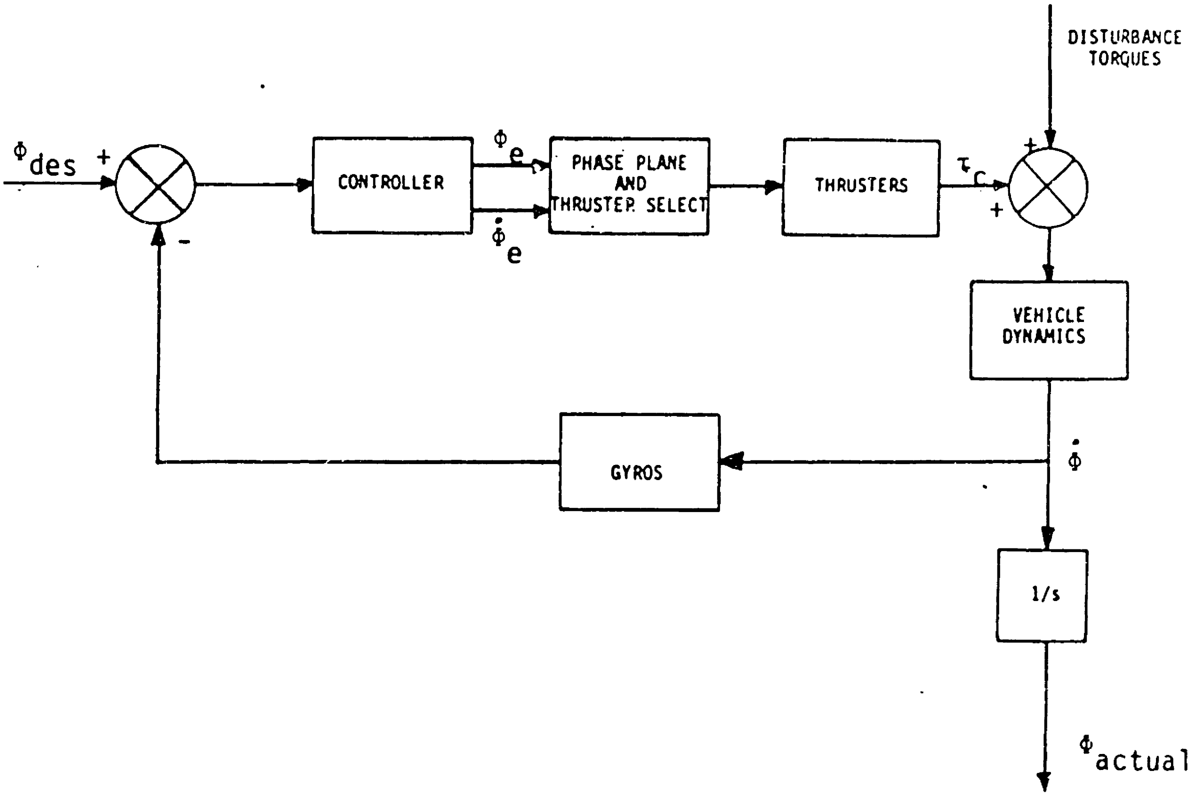
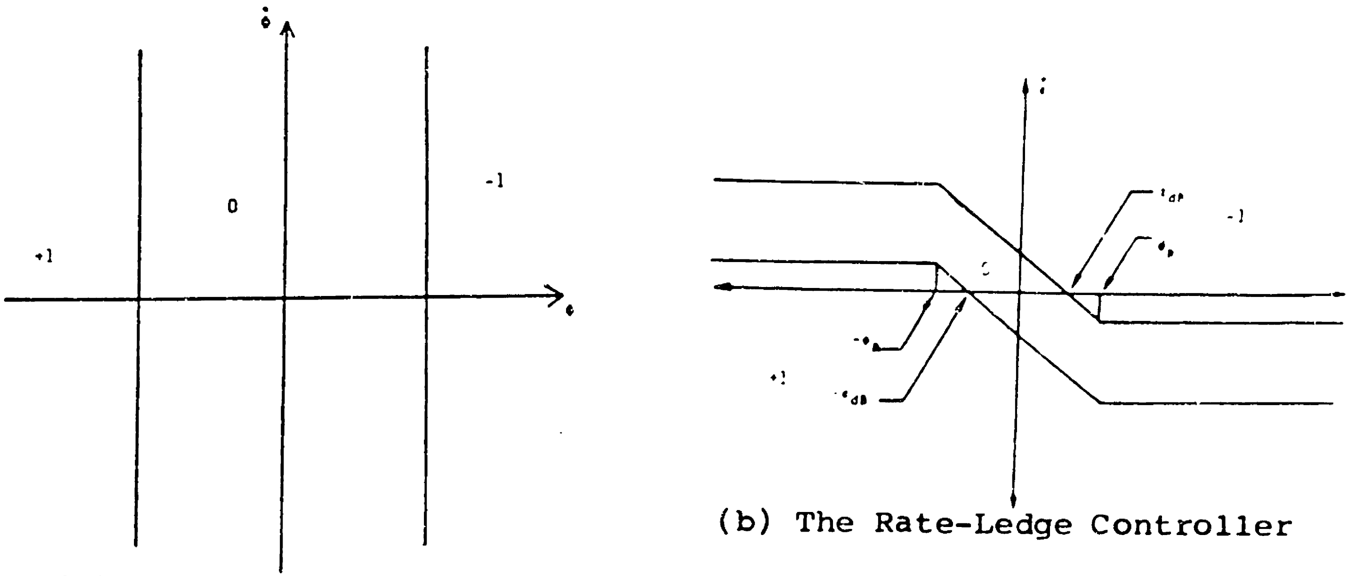


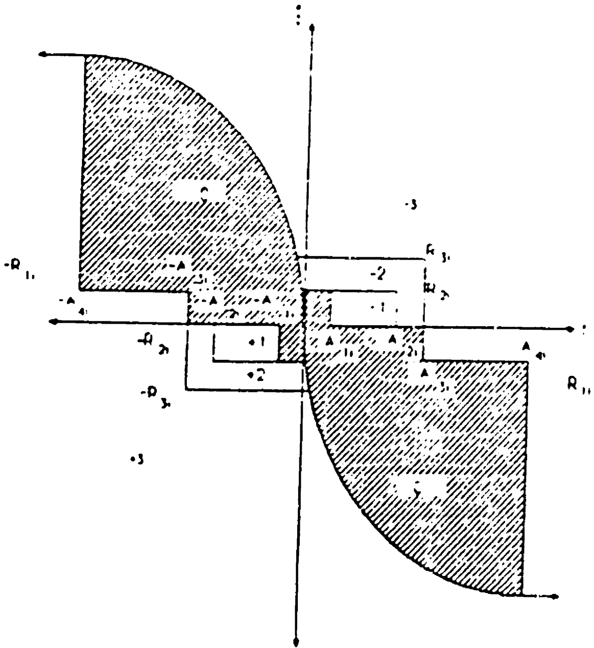
Figure 1-2: A Generic Thruster Attitude Control System

spacecraft attitude about each individual axis. The error signal is fed into a controller, which processes the attitude error and generates an approximate rate error. The controller commonly includes proportional, integral, derivative, pseudo-rate, and path guidance terms, as well as low pass filters to smooth its output signals (see Chapter 3). The outputs from the controller are fed into the phase plane logic. Phase planes can vary from a simple attitude deadband (Figure 1-3a) to a "rate ledge" arrangement (Figure 1-3b) to even more complicated schemes (Figure 1-3c). The phase plane region in which the current state falls determines how the thrusters will be commanded to fire. The phase plane in Figure 1-3a, for example, commands the thrusters to deliver negative torque if the angle output by the controller exceeds a certain deadband value, and positive torque if the angle is less than a certain value. The thrusters follow the commands generated by the phase plane logic, and the vehicle rotates accordingly. Rate-integrating gyros, which function much as their name implies, measure the current angle of the spacecraft and provide feedback.



(a) A Simple Deadband

(b) The Rate-Ledge Controller



(c) The Magellan Phase Plane

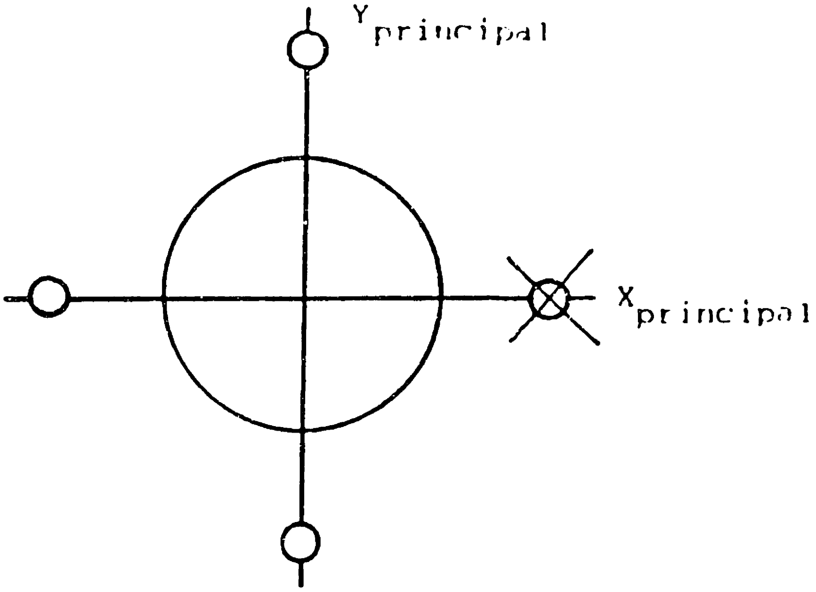
Figure 1-3: Typical Phase Planes for Attitude Control

The design of such a system is very challenging. The bang-bang control provided by the thrusters, various nonlinearities in the system, and time lags in the thrusters make analysis extremely difficult. Still, many authors have proposed solutions to this problem. Some of these solutions are quite original; for example, Quam [18] suggests a control algorithm involving switching lines in a plane plotting the spacecraft rigid-body energy vs. the energy of the first flexible mode, while Liu *et al.* [12] use the steepest descent method to derive an optimal open loop control law. However, most of the control system designs proposed in the literature seem to follow one of three basic approaches. The first approach is to put all the system complexity into the phase plane, while using as simple a controller as possible. Some authors suggest the use of simple deadbands of one sort or another [9]. Others apply optimal control theory to the task, obtaining various more complicated solutions [7, 20, 6, 25, 26]. Some even propose phase planes in which switching lines are not stationary, but move as functions of time or of spacecraft state [5, 15]. The second approach is to put all the complexity in the controller, using as simple a phase plane as possible. Proportional-derivative control is commonly used [2, 13, 14], but pseudo-rate [2, 16, 3] and pulse-width-pulse-frequency [3, 1, 28, 27] control are also widespread. The third approach uses no phase plane at all. Instead, in the variable pulse width approach, a controller outputs a time instead of an angle. The thrusters are commanded to fire for this amount of time. Clark [4] proposes one such scheme, while another is derivable from optimal control theory (see Chapter 3).

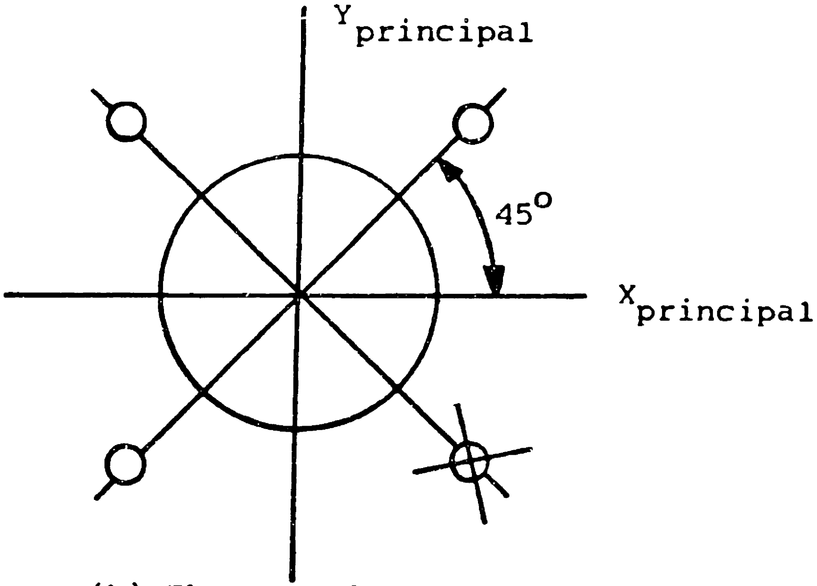
Most of the control schemes described above rely on some drastic simplifying assumptions. The most universal of these is the assumption that thrusters are mounted such that they provide torques about the principal axes of the spacecraft. Thus, torques about one axis have no effect on the other axes, so each axis can be controlled independently. Unfortunately, on a real spacecraft, there is a very good reason not to align thrusters with principal axes. Figure 1-4a shows a set of thrusters that are aligned in this way. As the figure shows, a failure in any single thruster makes control about one of the principal axes impossible. A single point failure can thus cause a complete loss of attitude control. It is much safer to position thrusters at a 45 degree angle from the principal axes, as shown in Figure 1-4b. In the event of the failure of a single thruster, this configuration allows some control torque to be provided about all the principal axes. Yet, because of this thruster configuration, coupling between axes can realistically be expected. Other assumptions often used include the neglecting of time delays. Time delays are usually not negligible. These assumptions are necessary to reduce the complexity of the problem to a point where a design is possible, yet an analyst must wonder -- how well can a system designed with these simplifications control a real vehicle?

To cope with this uncertainty, some designers end up creating very complicated systems. The controller block of the Magellan spacecraft is shown in detail in Figure 1-5 [8]. The Magellan phase plane is shown in Figure 1-3c. As is evident from these figures, both the controller and the phase plane are very complex. Although techniques exist for analyzing the Magellan controller and the Magellan phase plane independently (see Chapter 3), their interaction is far too complicated for any pencil-and-paper analysis. The combination of the two control schemes does result in good system performance, but system parameters are very difficult to choose. The 4 controller parameters and 7 phase plane parameters are, in practice, picked by trial and error, using computer simulations. As one might expect, a large number of very expensive simulation runs are needed to properly choose these parameters. Similar system designs, analyzable without so much reliance upon the computer, would be much less expensive.

What is needed, then, is a simple, easily analyzable design for a thruster attitude control system that works effectively on a real spacecraft. The purpose of this study is to examine several different simple designs to see how well they perform on a real system. Each system considered will be designed analytically, tested, and tuned until performance



(a) Thruster Axes
Aligned with Principal Axes



(b) Thruster Axes
Rotated 45 Degrees

Figure 1-4: Two Possible Thruster Configurations

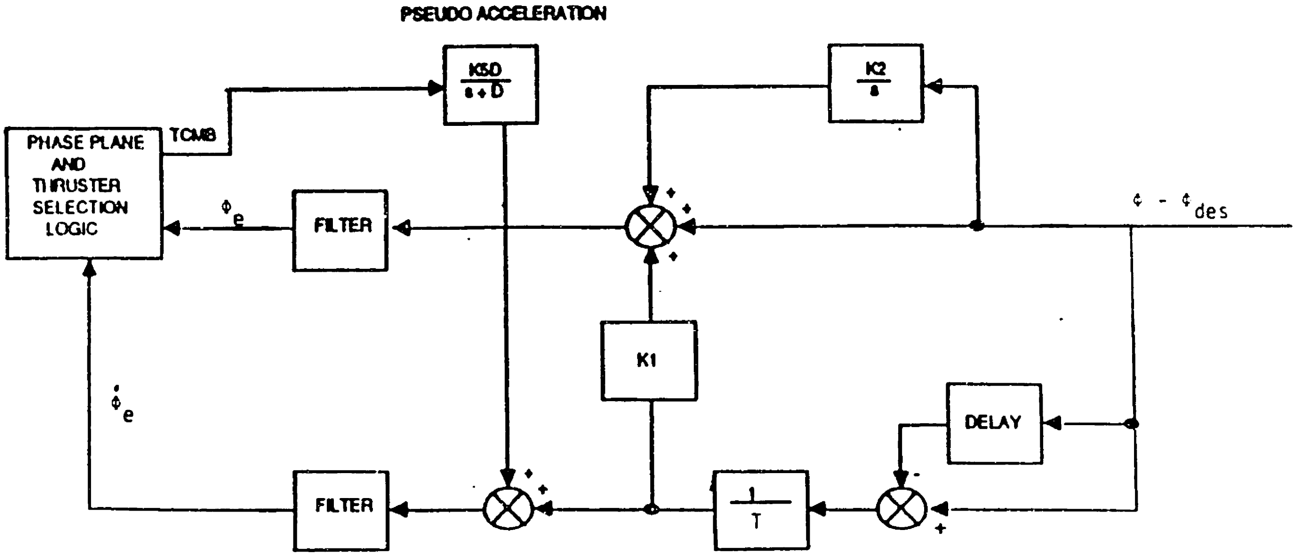


Figure 1-5: The Magellan Controller

appears to be optimized. By comparing the performance of the various systems, one can determine which is more easily analyzable, which performs better in various respects, and the tradeoffs inherent in the design of each of them. And a description of the optimization process might prove to be beneficial to someone who actually has to design one of these systems.

How can these control system designs be tested? The cost of building several different spacecraft and implementing a different attitude control system on each would be prohibitively high. However, the attitude and articulation control group on the Magellan project, working at Martin Marietta Denver Aerospace, have prepared a detailed computer simulation of their craft (see Chapter 2 and Appendix D). This simulation has been developed over a period of several months to mimic the actual design of the vehicle as closely as possible. This provides a unique opportunity to test designs for attitude control systems in as realistic an environment as is practical.

First, the models implemented in the simulator will be examined. The models of the rigid-body vehicle [10], the rate-integrating gyros [10], the thrusters [10], propellant consumption [8], disturbance torques [29, 30], vehicle mass properties [29], and propellant slosh [22, 23] will all be explained in detail in Chapter 2. Next, an overview of the literature will be presented, and likely control schemes will be chosen from it (Chapter 3). In Chapter 4, two important test cases will be discussed. In the first, termed "normal attitude control," the system will be required to respond to a large attitude and rate error (2 degrees and 1 deg/sec, respectively) about each axis (to maximize inter-axis coupling) in the absence of any disturbance torques. In the second case, the Magellan's insertion into Venus orbit (VOI) will be simulated. No initial rate or attitude errors will be assumed, but during VOI disturbance torques can become as high as 80% of the control torque. Chapter 4 will also develop performance criteria for each of these test cases, to facilitate the comparison of control system designs. Each design's parameters will be determined analytically, and the simulator will be used to test them. Then, parameters will be tuned until optimal performance is obtained (Chapters 5-7). The tuning process should give insights into the effectiveness of each system and the tradeoffs inherent in each design approach. Finally, an example of a case in which the findings of this thesis had great practical importance will be presented (Chapter 8), and conclusions, recommendations, and a final comparison of the various attitude control systems will be given (Chapter 9).

Chapter 2

Spacecraft Models

In order to test thruster attitude control systems in a realistic environment, an accurate computer simulation of an actual spacecraft is needed. This study is indebted to the attitude and articulation control system (AACS) group on the Magellan project, which has developed such a simulation. A diagram of the Magellan spacecraft is shown in Figure 2-1. Before discussing control system designs, it is important to understand the pertinent Magellan models that are implemented in the simulator[2].

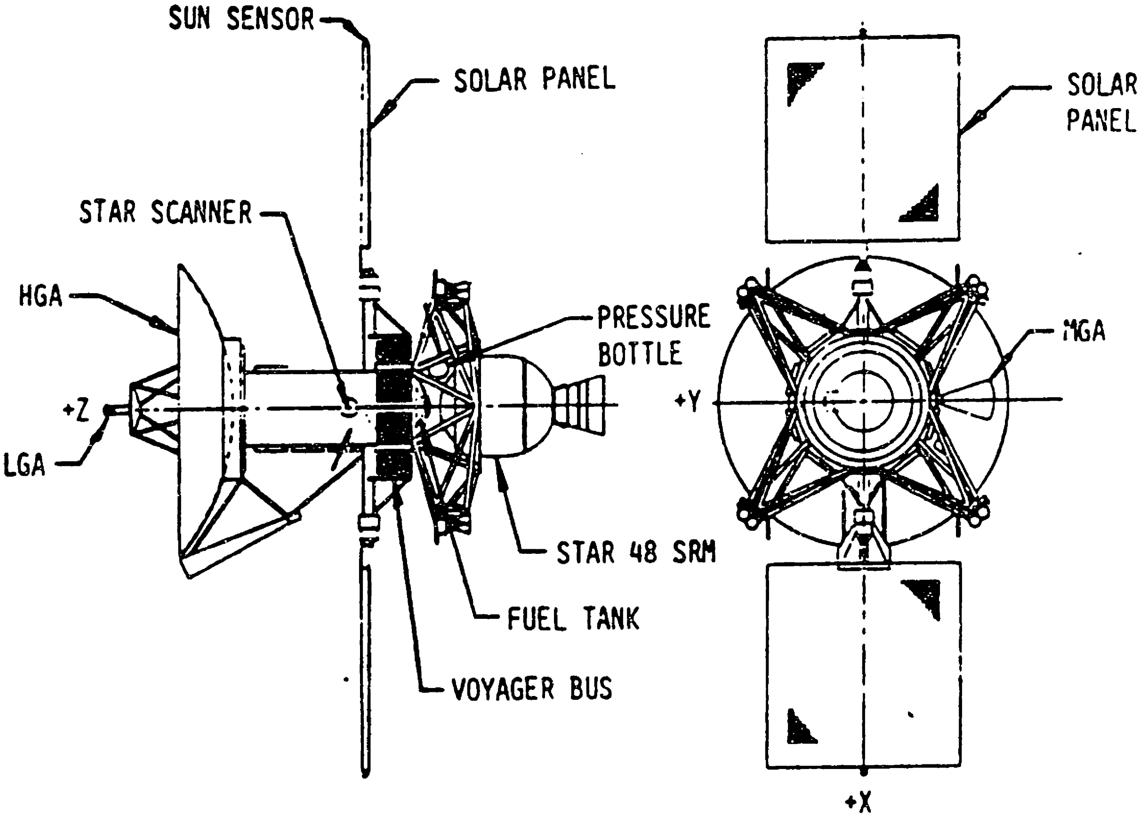


Figure 2-1: The Magellan Spacecraft

---

[2] Except for propellant slosh, all the models discussed in this chapter were developed and implemented by AACS analysts Lyle Cloud, Gerald Francis, Nabil El Gabalawi, Charles Gay, Whittak Huang, Steven Johnson, Thomas Kelecy, Narotham Reddy, and Charles Stockdale. I developed the slosh model myself.

2.1 Normal Attitude Control Models

The following models are adequate to simulate the effects of thruster control on the behavior of the spacecraft in the absence of any external disturbances. They are implemented with a sampling frequency of 7.5 Hz.

2.1.1 Rigid Body Vehicle Model [10]

Although Magellan has two flexible solar arrays, the dynamic complications introduced by these arrays are usually ignored. The effects of flexibility on attitude control system performance are not expected to be very large. Thus, a rigid body vehicle model is used for this thesis.

For rotation about principal axes, spacecraft motion is governed by the equations:

$$\tau_1 = I_1\dot{\omega}_1 + (I_3 - I_2)\omega_2\omega_3 \tag{2.1}$$

$$\tau_2 = I_2\dot{\omega}_2 + (I_1 - I_3)\omega_3\omega_1$$

$$\tau_3 = I_3\dot{\omega}_3 + (I_2 - I_1)\omega_1\omega_2$$

where τ = applied torque, ω = angular velocity, I = inertia, and the subscripts 1, 2, and 3 refer to the body x, y, and z axes respectively. These equations are implemented in the simulation. The inertia values used to test normal attitude control are shown in Table 2-I. Inertia values used in the high disturbance case will be discussed later in this chapter.

Table 2-I: Inertia Values for Normal Attitude Control

body axis (j)	I_{jj} (kg-m^2)
X	1050
Y	2000
Z	1600

The thrusters do not control about principal axes. As shown in Figure 2-2, a set of

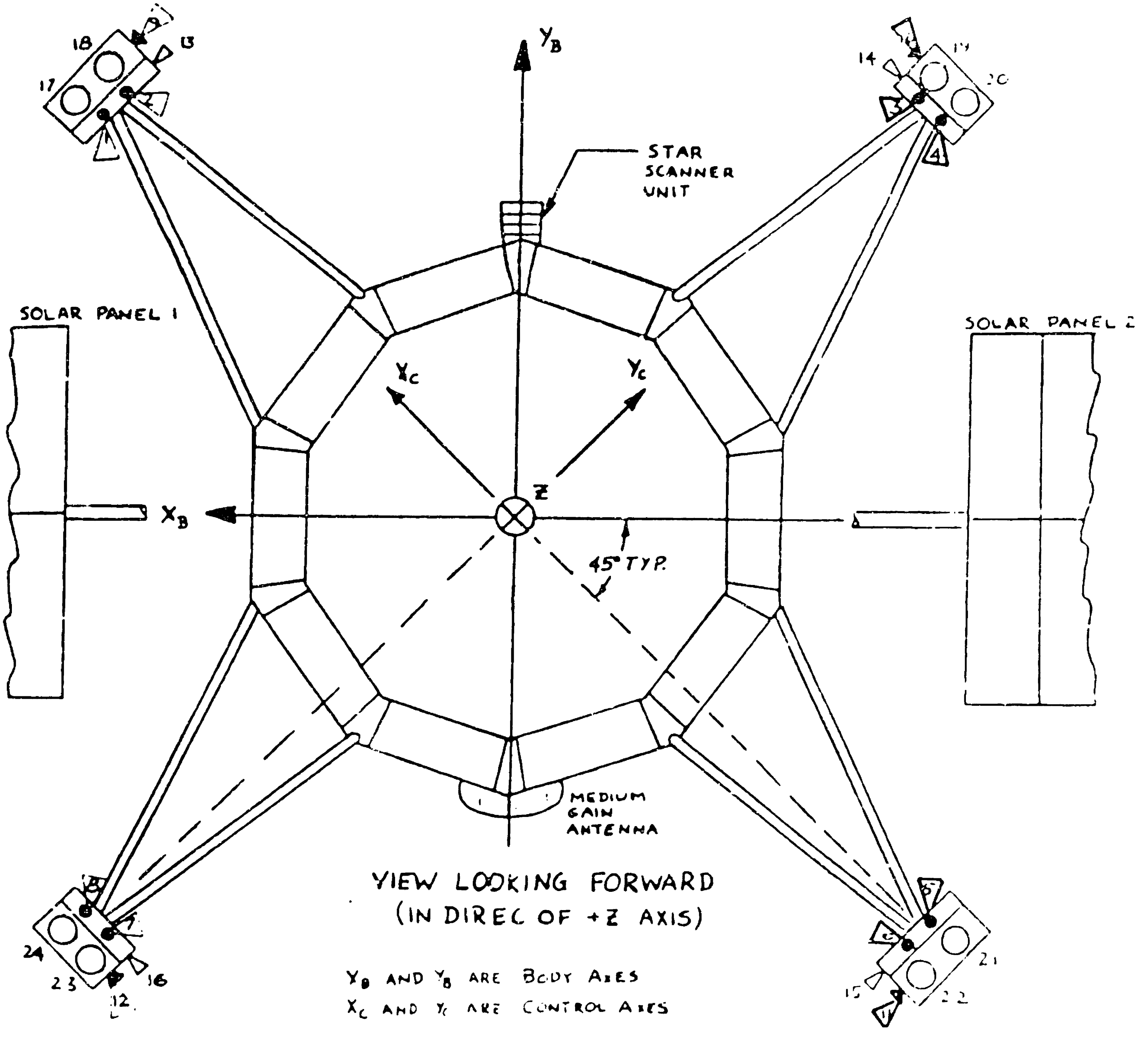


Figure 2-2: Magellan Body and Control Axes

control axes are defined. The x and y control axes are rotated 45 degrees from the principal axes, while the z control axis corresponds to the z body axis. The thrusters can only deliver torques about the control axes. These torques can be resolved into body coordinates by the equations:

$$\tau_{1B}=\frac{\sqrt{2}}{2}(\tau_{1C}-\tau_{2C}) \tag{2.2}$$

$$\tau_{2B}=\frac{\sqrt{2}}{2}(\tau_{1C}+\tau_{2C})$$

$$\tau_{3B}=\tau_{3C}$$

where the C subscript indicates torque about the control axes and B indicates body axes. Torques, angular positions, and angular rates can be transformed back and forth from control to body coordinates by similar equations, *ie*.

$$\phi_{1B}=\frac{\sqrt{2}}{2}(\phi_{1C}-\phi_{2C}) \tag{2.3}$$

$$\phi_{2B}=\frac{\sqrt{2}}{2}(\phi_{1C}+\phi_{2C})$$

$$\phi_{3B}=\phi_{3C}$$

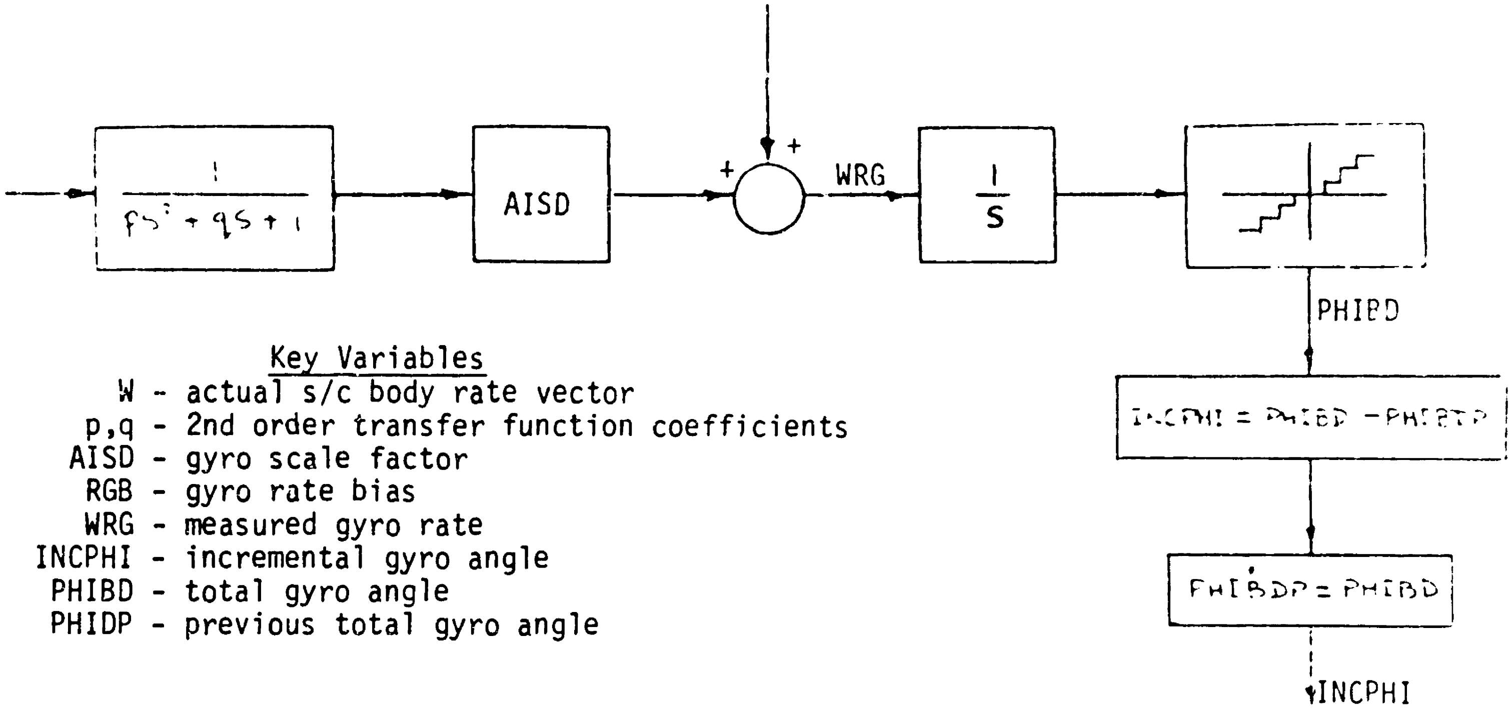
$$\omega_{1B}=\frac{\sqrt{2}}{2}(\omega_{1C}-\omega_{2C})$$

$$\omega_{2B}=\frac{\sqrt{2}}{2}(\omega_{1C}+\omega_{2C})$$

$$\omega_{3B}=\omega_{3C}$$

2.1.2 Gyro and Error Generation Models [10]

A set of rate-integrating gyros measures the angle accumulation about each of the body axes. This can be processed to give the angular position about each axis. The gyro model is shown in Figure 2-3. The angular rate about an axis is effectively shaped by a second order transfer function, integrated, and quantized. The integration gives the incremental angle through which the spacecraft has rotated since the previous sampling instant. The incremental angle can then be added to the known spacecraft position at the end of the last sampling interval to get the current spacecraft position. Parameters for this model are given in Table 2-II.



Key Variables

W - actual s/c body rate vector
p,q - 2nd order transfer function coefficients
AISD - gyro scale factor
RGB - gyro rate bias
WRG - measured gyro rate
INCPHI - incremental gyro angle
PHIBD - total gyro angle
PHIDP - previous total gyro angle

Figure 2-3: Rate-Integrating Gyro Model

Table 2-II: Gyro Model Parameters

parameter	value
p	5.17E-04
q	3.2E-02
AISD	1.0
RGB	0.0
quantization	7.75E-07 rad

The Magellan spacecraft has no way of directly measuring angular rates. The Magellan controller generates rates by simply dividing the incremental angles generated in the gyros by the sampling time interval. The Galileo spacecraft uses a similar scheme. Although this can generate a noisy signal, with filtering it can provide a good estimate of angular velocities.

2.1.3 Thruster Models [10]

The Magellan spacecraft uses three different types of thrusters -- 0.2 lbf thrusters (for normal attitude control), 5 lbf thrusters (for z-axis control during the high disturbances of Venus orbital insertion (VOI)), and 100 lbf thrusters (for x and y-axis VOI control). The behavior of these thrusters is not ideal -- time delays and variations in propellant tank pressure can have a strong effect on performance.

Modeled thruster profiles are shown in Figure 2-4. Time delays T_1, T_2, T_3, and T_4 are defined in the figure. Note that the thrusters do not instantaneously step from zero output to maximum thrust, but rather ramp up to this thrust level in finite time.

The maximum thrust level is not the nominal 0.2, 5, or 100 lbsf. The thruster force magnitude is a function of inlet pressure, which, in turn, is a function of the mass of the propellant in the tank. An experimentally derived plot of inlet pressure vs. expended propellant mass (assuming that, at the beginning of the mission, 133 kg of fuel is present) is shown in Figure 2-5 and the data is tabulated in Table 2-III. A fourth-order polynomial curve fit to the data results in the equation:

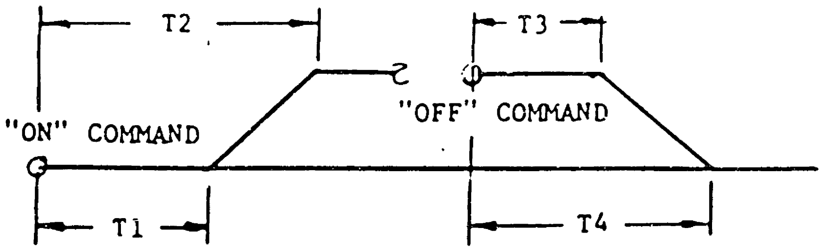
$$P = 577.7445 - 2.7432m + (7.6054\times10^{-2})m^2 - (6.9018\times10^{-6})m^3 - (4.0716\times10^{-9})m^4 \qquad (2.4)$$

where P = inlet pressure (psia) and m = propellant mass (lbm). Thrust level is a function of inlet pressure, and is given by the equation:

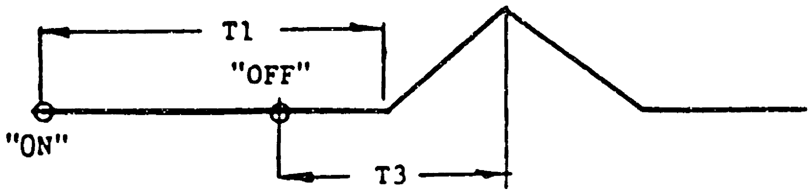
$$\text{Thrust} = Q_iP^2 + S_iP + Y_i \qquad (2.5)$$

```
where i = 1 for the 0.2 lbf thrusters
      i = 2 for the 5 lbf thrusters
      i = 3 for the 100 lbf thrusters
```

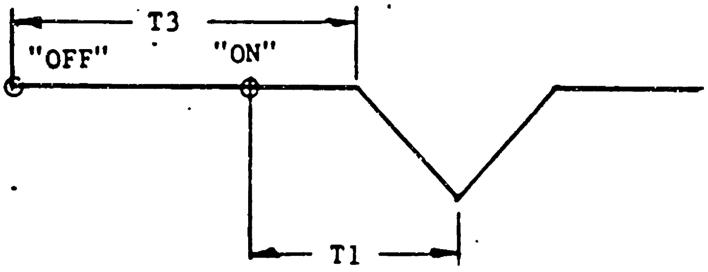
and Q_i, S_i, and P_i are coefficients given in Table 2-IV.



TIME DELAY DEFINITIONS



"OFF" COMMAND SENT PRIOR TO FULL "ON"



"ON" COMMAND SENT PRIOR TO FULL "OFF"

Figure 2-4: Modeled Thruster Thrust Profiles

Values of important thruster model parameters for both normal attitude control and VOI are given in Table 2-V.

2.1.4 Propellant Consumption Model [10]

The propellant consumed by one thruster over a sampling interval of duration t is determined by the equation:

$$\textit{Fuel Consumed} = \int_0^t \frac{F}{I_{sp}} dt \tag{2.6}$$

where F = the thrust magnitude and I_{sp}= the specific impulse of the thruster. I_{sp} values are

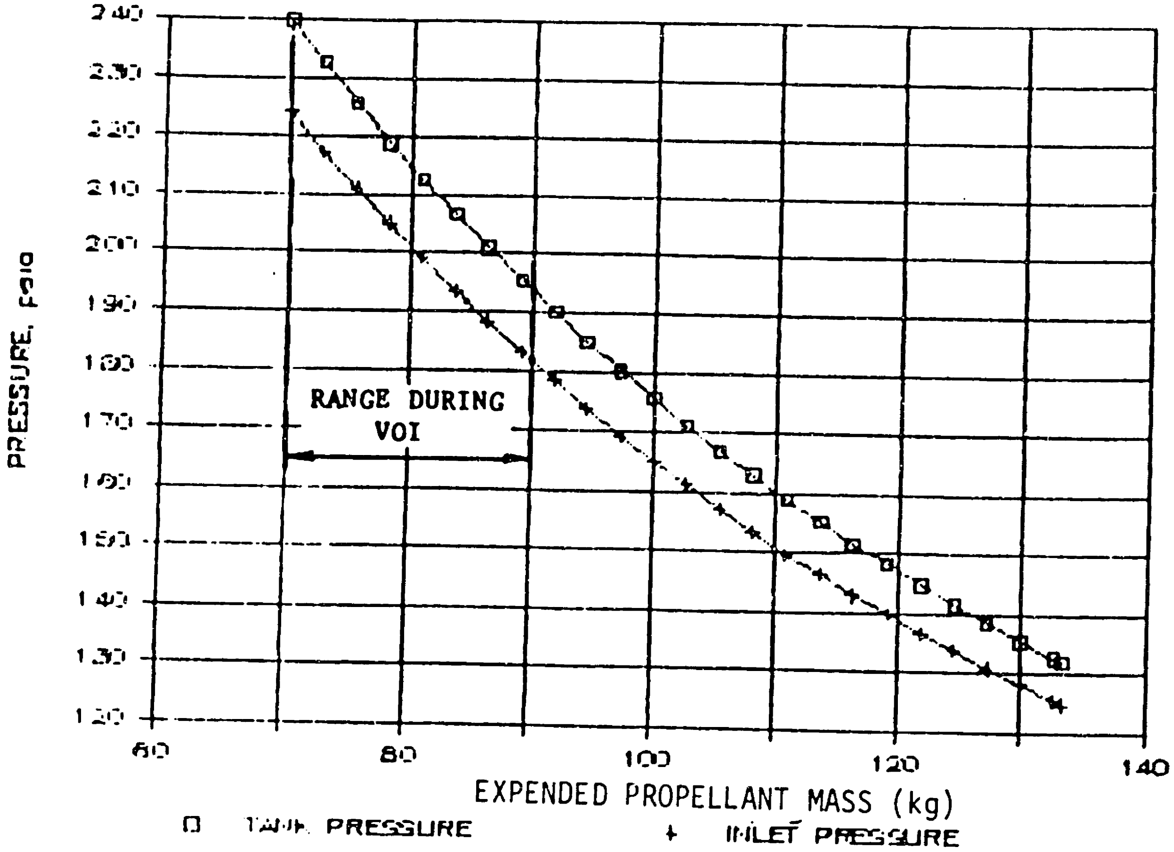


Figure 2-5: Tank Pressure vs. Propellant Mass

given in Table 2-VI. The total fuel consumed is simply the sum of the fuel consumed by each individual thruster.

2.2 VOI Models

The models described above are adequate for simulations of normal attitude control. However, additional models are needed for Venus orbital insertion (VOI). During VOI, the solid rocket motor (SRM) accelerates the spacecraft. Within 80 seconds, this acceleration can reach 7g; a typical acceleration profile is shown in Figure 2-6. Side forces, resulting from a misalignment of the SRM, can cause large disturbance torques. These disturbances must be modeled. In addition, the consumption of large quantities of SRM fuel causes great changes in spacecraft mass. The varying mass properties must be modeled over the 80 seconds of VOI. Finally, sloshing of liquid propellant can affect spacecraft behavior, and thus must also be modeled.

Table 2-III: Tank Pressure and Propellant Mass

PROP MASS kg	TANK PRESS psia	INLET PRESS psia
69.99	239.89	224.08
72.71	232.56	217.37
75.43	225.59	210.99
78.16	218.96	204.92
80.88	212.64	199.13
83.60	206.61	193.60
86.32	200.86	188.32
89.04	195.36	183.27
91.76	190.11	178.48
94.48	185.08	173.85
97.21	180.27	169.42
99.93	175.66	165.18
102.65	171.24	161.10
105.37	167.00	157.19
108.09	162.93	153.43
110.81	159.02	149.82
113.54	155.27	146.34
116.26	151.65	142.99
118.98	148.17	139.77
121.70	144.82	136.67
124.42	141.60	133.68
127.14	138.49	130.80
129.87	135.49	128.02
132.59	132.59	125.33
133.27	131.88	124.67

Table 2-IV: Coefficients for Equation (2.5)

COEFFICIENT	0.2 lb THRUSTER	5.0 lb THRUSTER	100.0 lb ENGINE
QQ(1)	0.000		
QQ(2)		-1.25E-5	
QQ(3)			2.361912E-4
SL(1)	6.1667E-4		
SL(2)		2.375E-2	
SL(3)			3.566581E-1
YI(1)	2.3333E-3		
YI(2)		3.000E-1	
YI(3)			12.23284

Because of the high disturbances, the sampling rate during VOI must be faster than the normal 7.5 Hz. A 30 Hz sampling frequency is currently planned.

Table 2-V: Thruster Model Parameters

variable	0.2 lb thruster	5.0 lb thruster	100.0 lb engine
T1	.015	.011	.022
T2	.038	.041	.029
T3	.023	.052	.017
T4	.050	.061	.061

propellant for normal attitude control = 60.0 kg
propellant at start of VOI = 85.0 kg

(a) Time Parameters and Propellant Masses

0.2 lb Thruster NO.	LTO (inches)		
	XC	YC	ZC
1	76.5	-1.5	-8.5
2	76.5	1.5	-8.5
9	80.2	7.0	10.0
3	1.5	76.5	-8.5
4	-1.5	76.5	-8.5
10	7.0	80.2	10.0
5	-76.5	1.5	-8.5
6	-76.5	-1.5	-8.5
11	-80.2	-7.0	10.0
7	-1.5	-76.5	-8.5
8	1.5	-76.5	-8.5
12	-7.0	-80.2	10.0

(b) Thruster Moment Arms

Table 2-VI: Specific Impulses of Magellan Thrusters

thruster	I_{sp} (N-sec/kg)
0.2 lb	1961
5.0 lb	2314
100.0 lb	2270

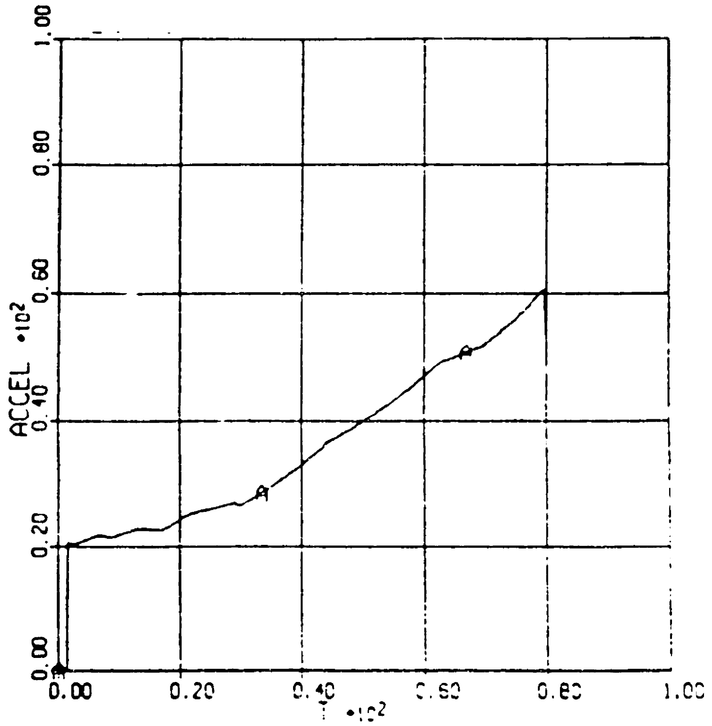


Figure 2-6: Magellan Acceleration During VOI

2.2.1 Disturbance Models [29, 30]

During VOI, the SRM burns fuel and accelerates the spacecraft. SRM thrust during VOI is shown in Table 2-VII and plotted in Figure 2-7 (the 3-σ high thrust indicated in the figure is used in this study as a worst case). Between the data points shown in the table, the simulator interpolates in a linear fashion.

There are four important sources of disturbances during VOI. First, misalignment of the SRM is a problem. Second, even if no misalignment exists, normal operation of the SRM generates heavy side forces. Swirl torques also disturb the spacecraft. Finally, jet damping makes a small contribution to the disturbances.

Misalignment torques occur when the SRM thrust vector does not pass through the spacecraft center of mass. Figure 2-8 shows SRM misalignment. The SRM can be translationally misaligned a distance L_m (modeled as being entirely in the y-body axis) and angularly misplaced an angle δ_m (in the y-z plane). SRM thrust is given by the equations:

Table 2-VII: SRM Thrust During VOI

TIME (seconds)	SRM THRUST (pounds)		
	3 SIGMA HIGH	NOMINAL	3 SIGMA LOW
-10.00	0.00	0.00	
0.00	0.00	0.00	
0.10	15000.00	13800.00	12500.00
0.70	14500.00		
0.80		13400.00	
0.90			12000.00
5.10	15200.00		
5.50		14000.00	
6.10			12700.00
7.30	14700.00		
7.90		13500.00	
8.70			12000.00
12.00	15100.00		
13.00		13900.00	
14.40			12600.00
15.80	14500.00		
17.10		13400.00	
18.90			12100.00
20.20	15500.00		
21.90		14300.00	
24.20			12900.00
27.60	15500.00		
28.20	15100.00		
30.00		14300.00	
30.60		13900.00	
33.20			12900.00
33.80			12600.00
43.10	17800.00		
46.80		16400.00	
51.80			14800.00
60.80	18300.00		
66.00		16900.00	
67.70	17100.00		
73.00			15300.00
73.50		15700.00	
78.30	16400.00		
80.00	1500.00		
80.60	0.00		
81.30			14200.00
85.00		15100.00	
86.80		1400.00	
87.50		0.00	
90.00	0.00	0.00	
94.00			13700.00
95.00	0.00	0.00	
96.00			1300.00

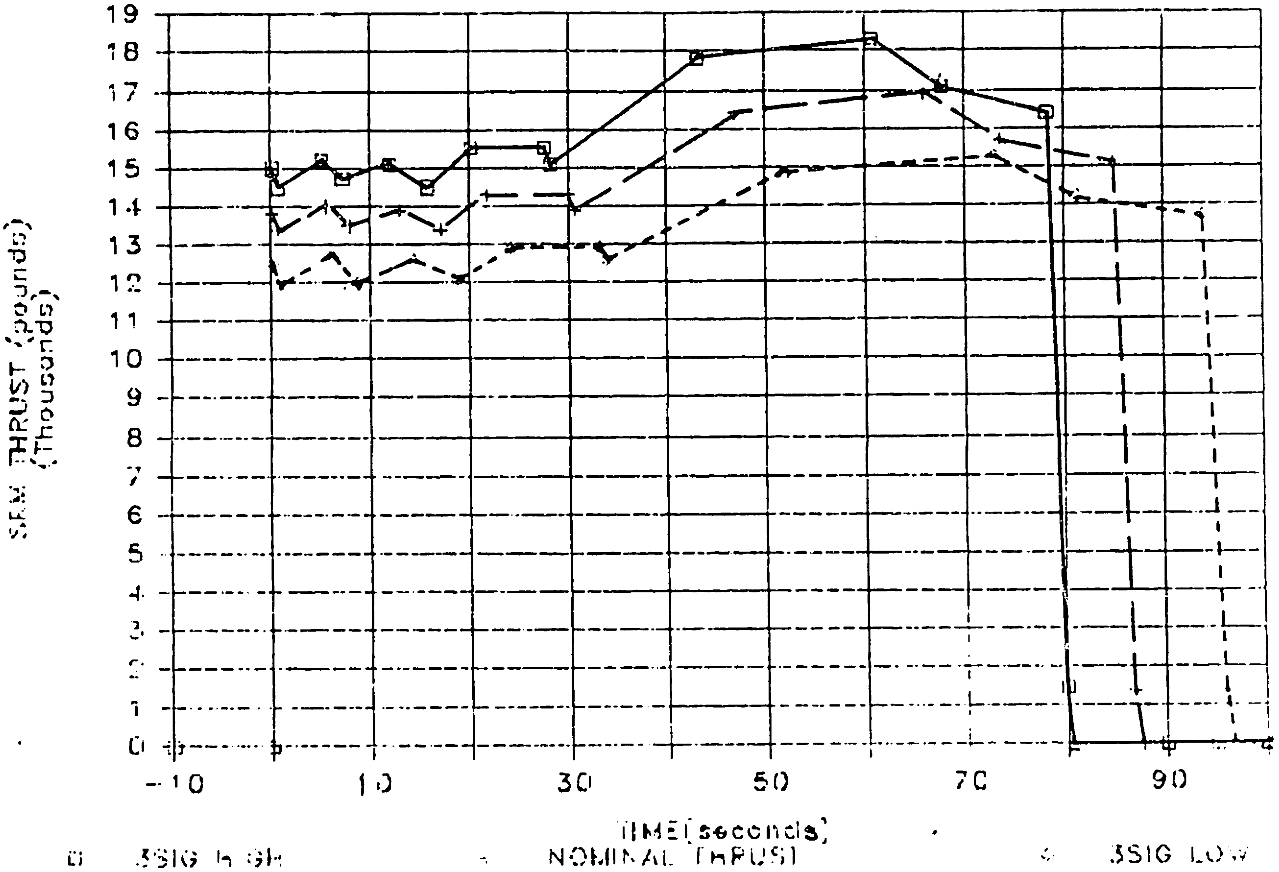


Figure 2-7: Srm Thrust vs. Time in VOI

$$F_{SRM1} = 0 \tag{2.7}$$

$$F_{SRM2} = |F_{SRM}|\sin(\delta_m)$$

$$F_{SRM3} = |F_{SRM}|\cos(\delta_m)$$

where $|F_{SRM}|$ is the thrust taken from Table 2-VII and $F_{SRM1,2,3}$ are the components in the x, y, and z body axes, respectively. The misalignment torque lever arm is:

$$R_{mslgn1} = 0 \tag{2.8}$$

$$R_{mslgn2} = \sqrt{L_m^2 + z_{int}^2 tan^2(\delta_m)}$$

$$R_{mslgn3} = -z_{int}$$

where z_{int} = the distance from the SRM exit plane to the spacecraft center of mass along the z-axis. Finally, misalignment torques are calculated by the equation:

$$\tau_{misalignment} = R_{mslgn} \times F_{SRM} \tag{2.9}$$

Parameter values are listed in Table 2-IX. Misalignment torques calculated from these values are plotted as a function of time in Figure 2-9.

The magnitude of the SRM side force has been empirically determined to be as shown in Table 2-VIII and Figure 2-10 (values indicated in the figure as 3-σ high values are used in this thesis) [21]. This force acts at a distance z_{side} from the pre-VOI center of mass, as shown in Figure 2-11. The side force initially pushes in the +y-body direction, but it rotates about the z-axis at a rate of 1.8 deg/sec. The side force components can be shown to be:

$$F_{side1} = -|F_{side}|\sin(\alpha_s) \tag{2.10}$$

$$F_{side2} = |F_{side}|\cos(\alpha_s)\cos(\delta_m)$$

$$F_{side3} = -|F_{side}|\cos(\alpha_s)\sin(\delta_m)$$

where $|F_{side}|$ is as shown in Figure 2-10 and α_s is the angle between the y-axis and the side force vector. Side force lever arm components are given by the equations:

$$R_{sf1} = 0 \tag{2.11}$$

$$R_{sf2} = R_{mslgn2}$$

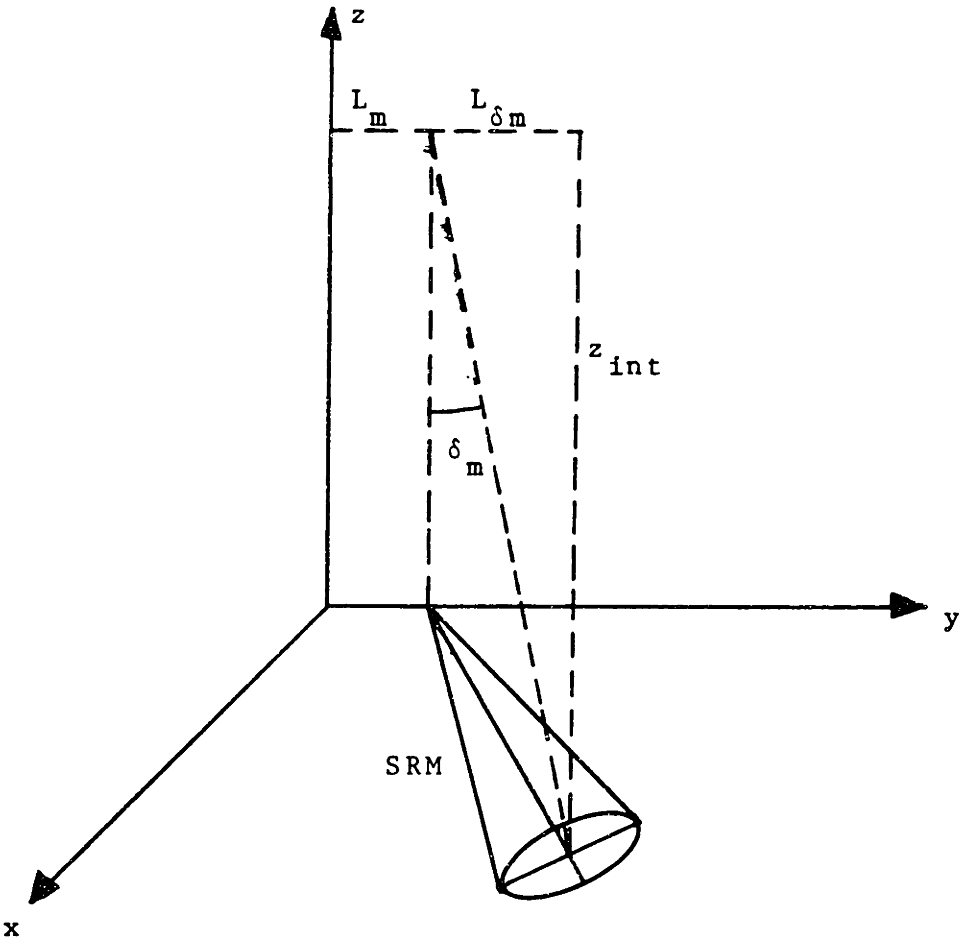


Figure 2-8: SRM Misalignment

$R_{sf3} = -(z_{cm} + z_{side})$

where z_{cm} is the z-axis distance from the pre-VOI center of mass to the current center of mass. Side force disturbances are plotted against time in Figure 2-12.

Swirl torques are caused by exit gases swirling around in the SRM nozzle. This is modeled as a constant torque, τ_{swirl}, about the z-axis.

Jet damping torques are caused by gases flowing out of the rocket nozzle. Thomson and Reiter [19] derive the jet damping moment to be:

$$\tau_{jd} = -r \times \dot{m}(\omega \times r) \tag{2.12}$$

where r = the jet damping moment arm, $\dot{m}$ = the rate of mass expulsion from the SRM (= $F_{SRM}/I_{sp\text{-}SRM}$), and ω = the spacecraft angular rate vector. Jet damping acts at the exit of

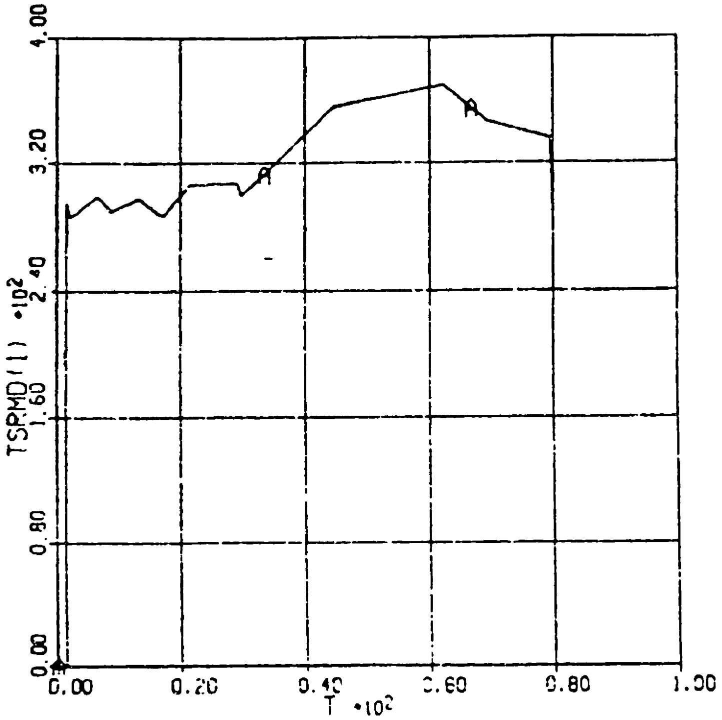


Figure 2-9: Misalignment Torques vs. Time in VOI

Table 2-VIII: SRM Side Force Magnitude During VOI

TIME (seconds)	SRM SIDE FORCE (pounds)	
	3 SIGMA	NOMINAL
-10.00	0.00	0.00
0.00	0.00	0.00
0.10	58.00	14.00
7.80	53.00	16.00
15.70	40.00	15.00
23.50	50.00	17.50
31.30	51.00	20.00
51.00	56.00	22.00
54.80	56.00	25.00
62.60	64.50	26.00
70.50	63.00	26.50
78.30	77.00	30.50
80.00	7.70	3.10
80.60	0.00	0.00
100.00	0.00	0.00

the SRM nozzle, so $r = R_{mslgn}$. Jet damping torques are plotted against time in Figure 2-13. Notice that they are small compared to other disturbance terms.

Figure 2-10: Side Force Magnitude vs. Time in VOI

Disturbance model parameters are given in Table 2-IX. The total disturbance torques about each axis are plotted in Figure 2-14.

2.2.2 Variable Mass Properties Model [29, 30]

As the SRM fires, it burns a significant amount of fuel. SRM fuel consumption is given by the equation:

$$Fuel\,Consumed = \int_0^t \frac{|F_{SRM}|}{I_{sp\text{-}SRM}}\,dt \tag{2.13}$$

where $I_{sp\text{-}SRM}$ is the SRM specific impulse. During VOI, almost 2000 kg of solid propellant is burned. Thus, spacecraft inertias about all three axes change appreciably and the spacecraft center of mass moves in the positive z-direction. Thruster lever arms change as a result of this. The Magellan masses, inertias, center of gravity locations, and thruster

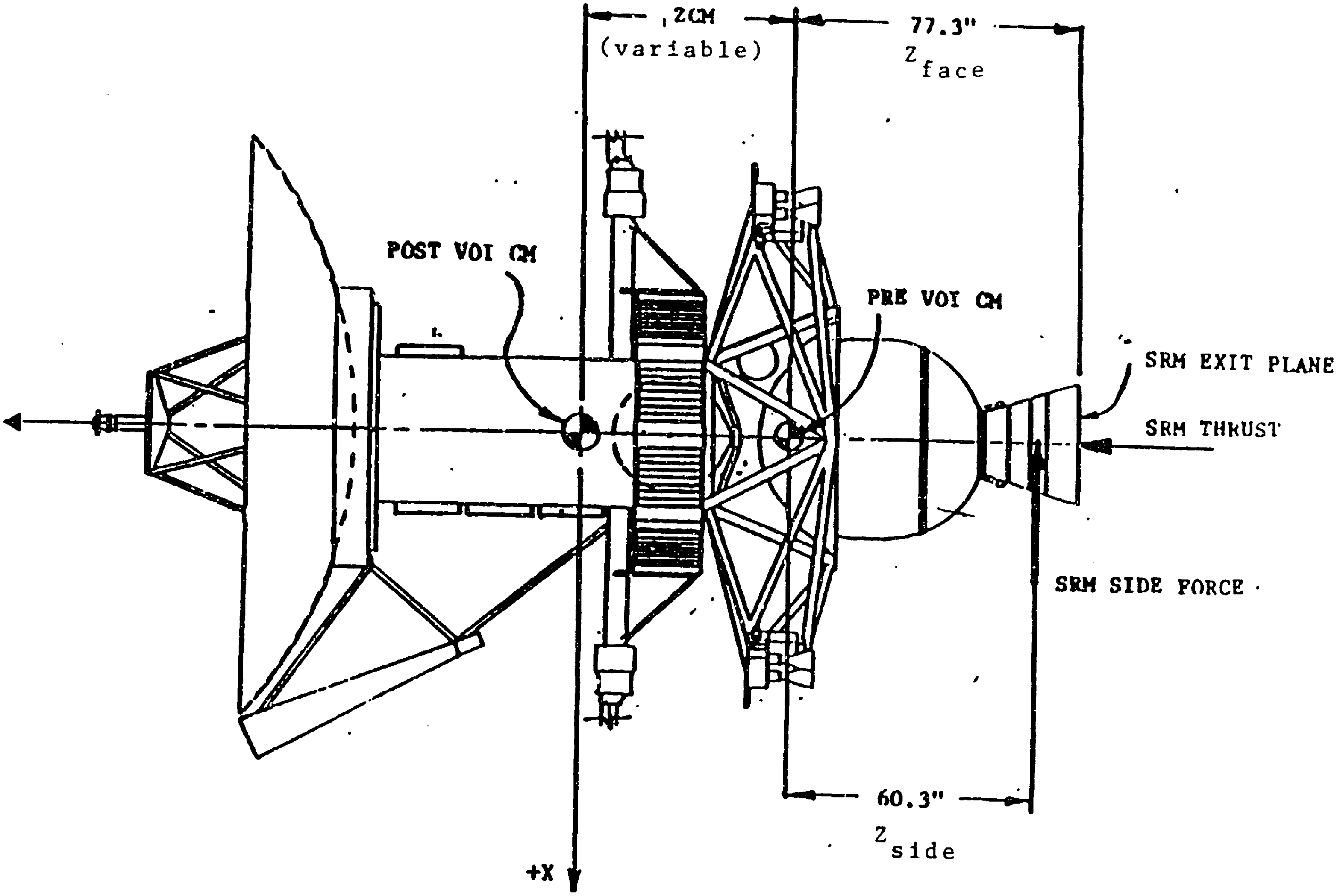


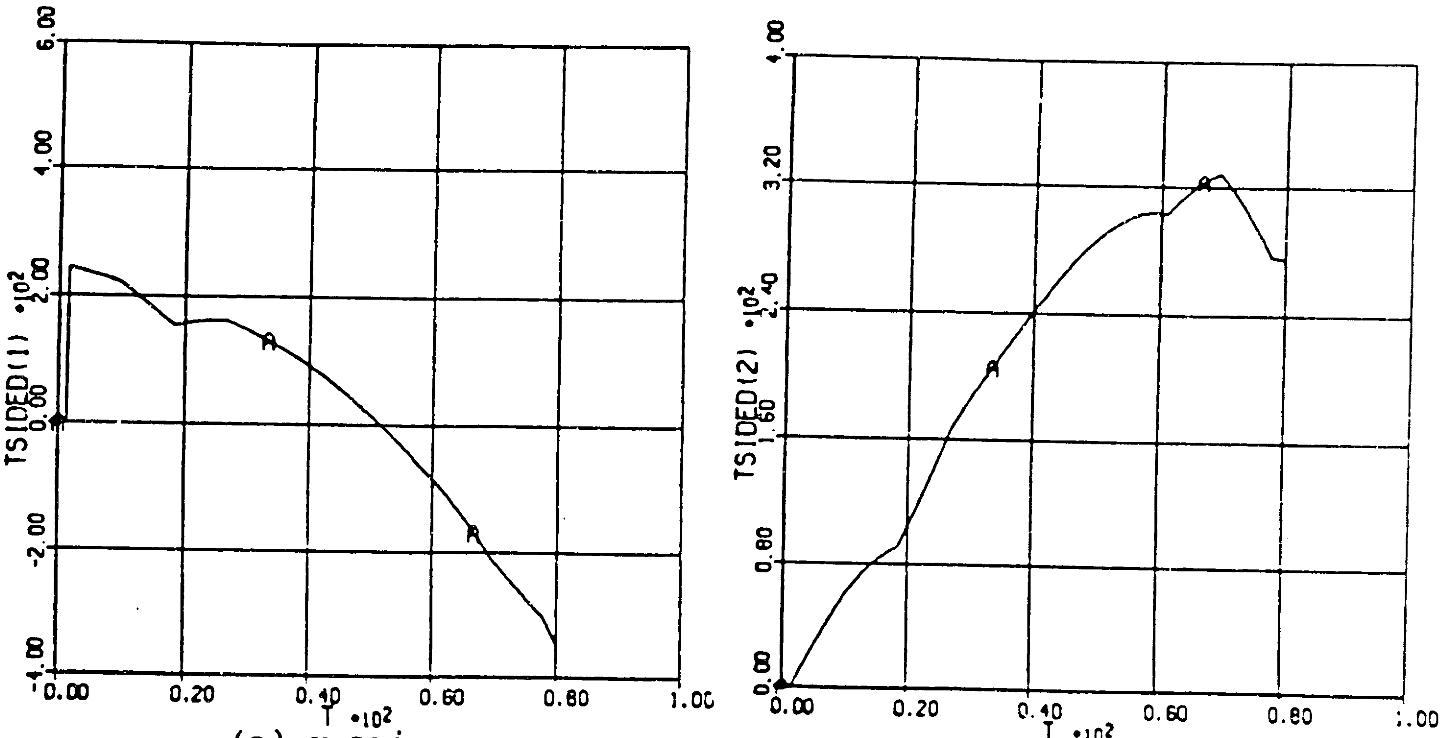
Figure 2-11: Geometry of Magellan During VOI

locations used to simulate VOI are listed in Tables 2-X and 2-XI. Between the data points listed, the simulator interpolates in a linear fashion.

2.2.3 Propellant Slosh [22, 23]

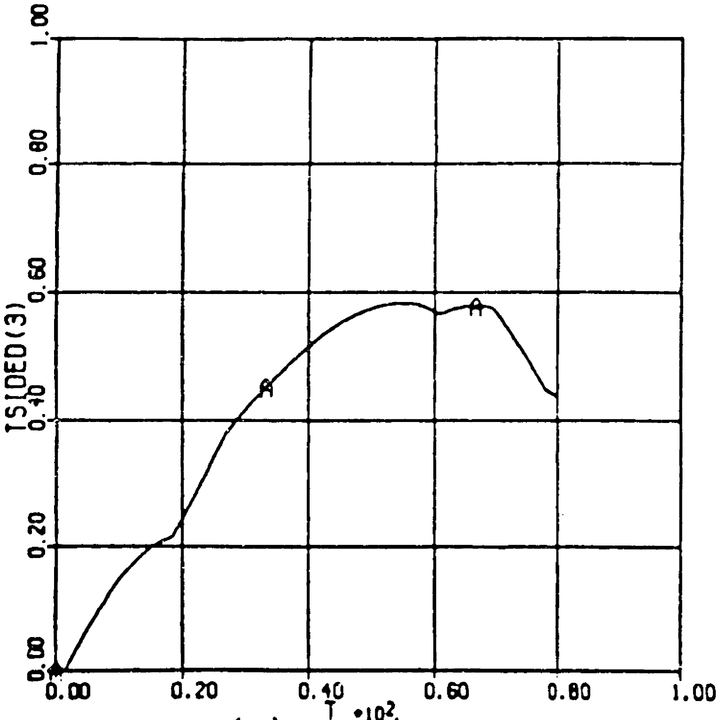
A one-axis model of a spacecraft with sloshing propellant is shown in Figure 2-15. The sloshing fuel is modeled as a swinging pendulum. Important model parameters are defined in the figure.

If E is defined to be the effective torque on the spacecraft, the effect of propellant slosh can be summarized by the equations:



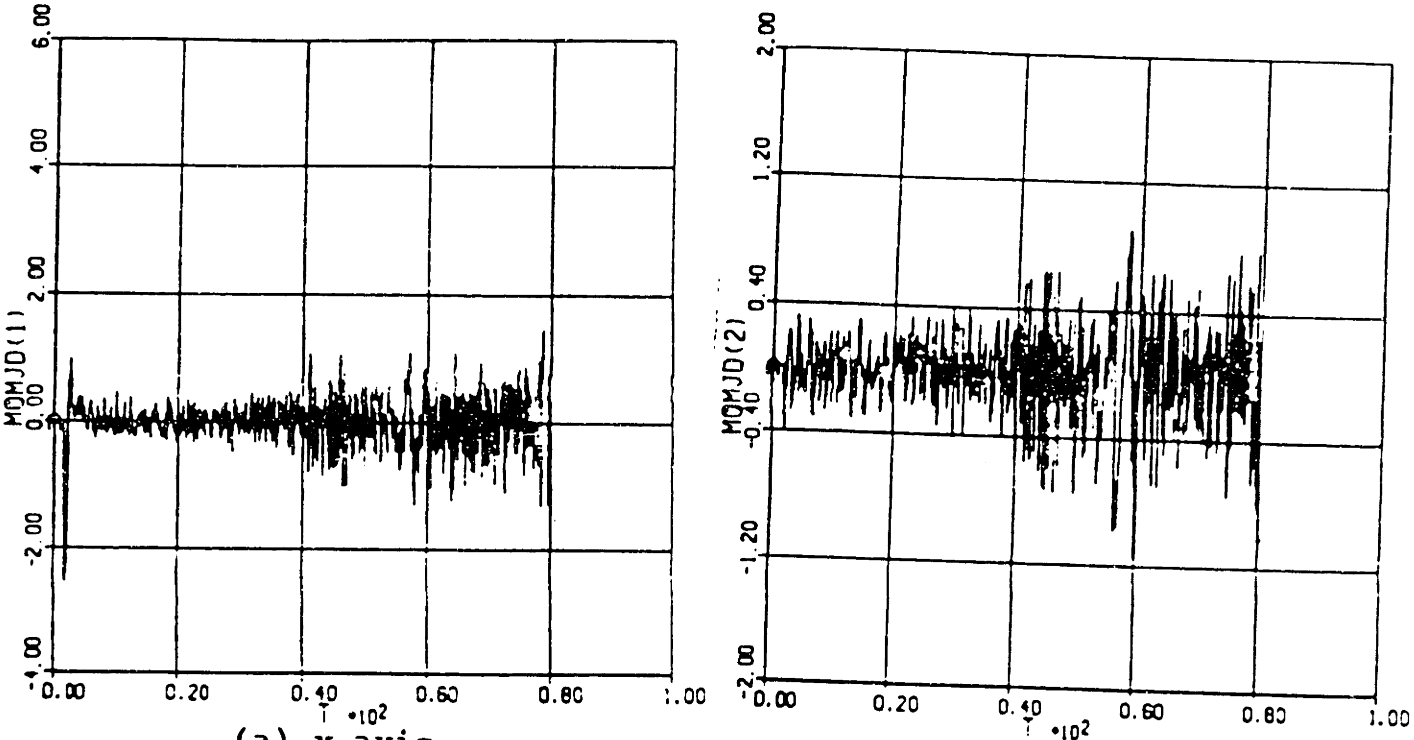
(a) x-axis

(b) y-axis



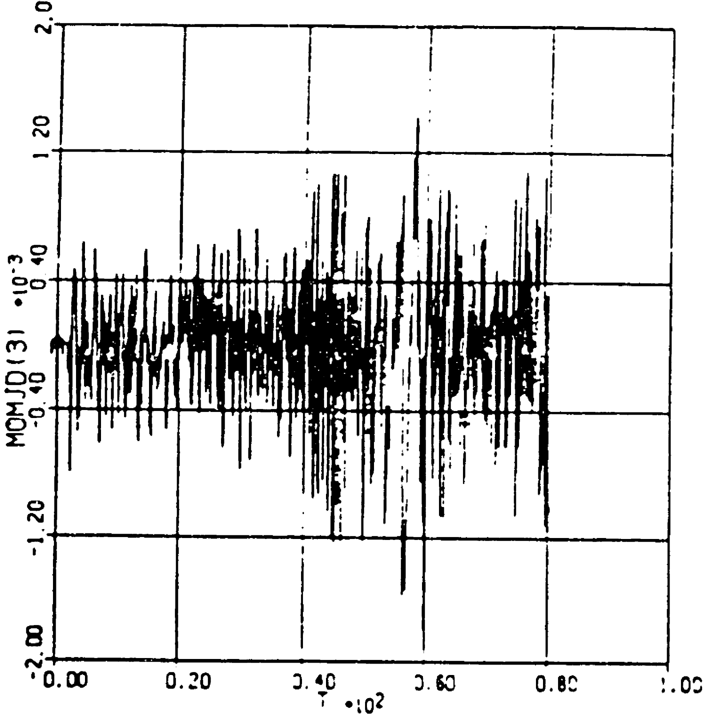
(c) z-axis

Figure 2-12: Side Forces vs. Time in Control Axes



(a) x-axis

(b) y-axis



(c) z-axis

Figure 2-13: Jet Damping Moment vs. Time in Control Axes

Table 2-IX: Disturbance Model Parameters

parameter	simulation value
L_m	0.159 in.
δ_m	0.01 deg.
Z_{face}	77.30 in.
Z_{side}	60.30 in.
I_{sp-SRM}	292.84 lb-sec/lbm
τ_{swirl}	7.36 Nm

$$\begin{bmatrix} \dot{N}_1 \\ \dot{N}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -f_2^2 & -2\zeta f_2 \end{bmatrix} \begin{bmatrix} N_1 \\ N_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \tau \end{bmatrix}$$

$$E=\frac{AI}{B}[(f_1^2-f_2^2)N_1+2\zeta(f_1-f_2)N_2]+\tau \tag{2.14}$$

where

$$A=\frac{I_f+ma^2}{ma}$$

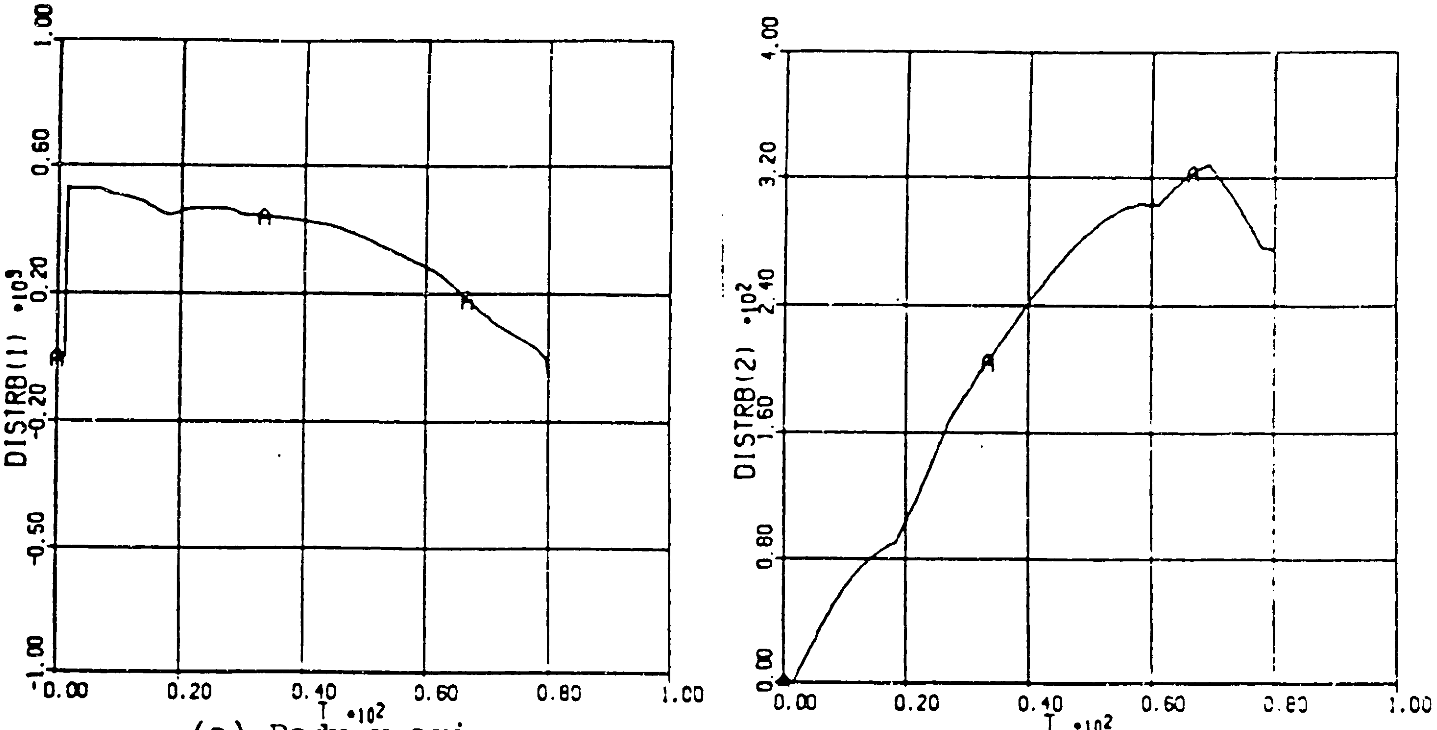
$$B=\frac{II_f+Ima^2+d^2mI_f}{ma}$$

$$f_1^2=\frac{G}{A}$$

$$f_2^2=\frac{G(I+md^2)}{B}$$

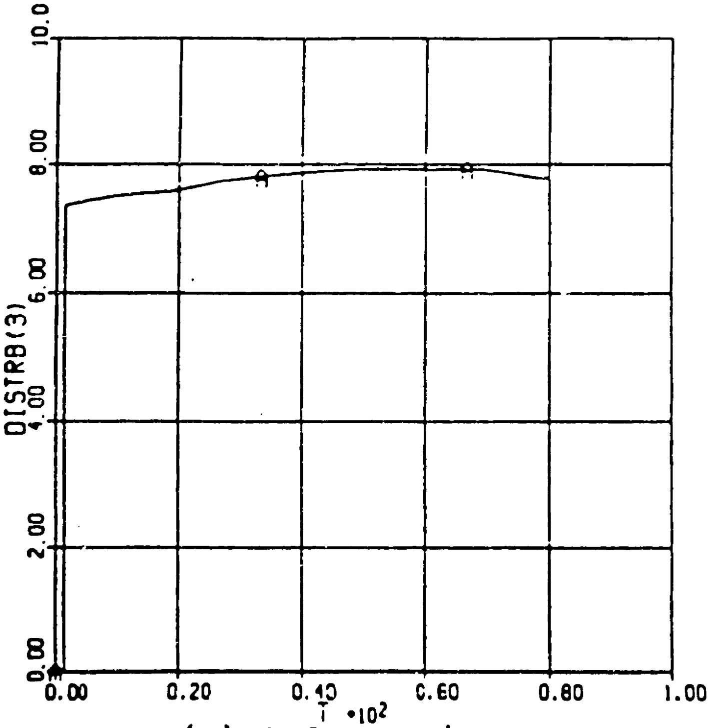
ζ is the slosh damping coefficient, and N_1 and N_2 are dummy variables (see Appendix A.1). These state space equations are implemented in the simulator.

If it is assumed that torques about the x-body axis cause no slosh in the y-axis and vice-versa, the model can be implemented independently for these two axes. Any effects slosh might have on torques about the z-axis are ignored.



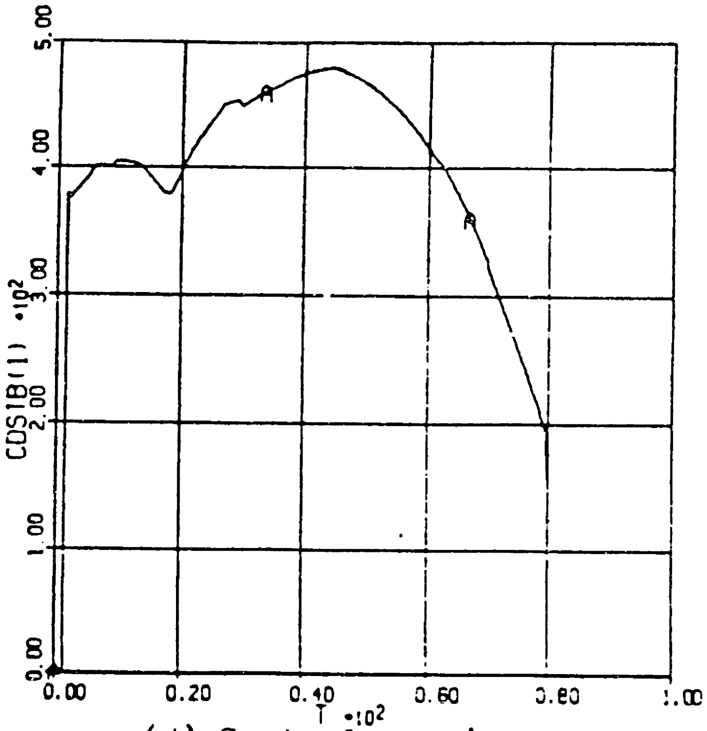
(a) Body x-axis

(b) Body y-axis

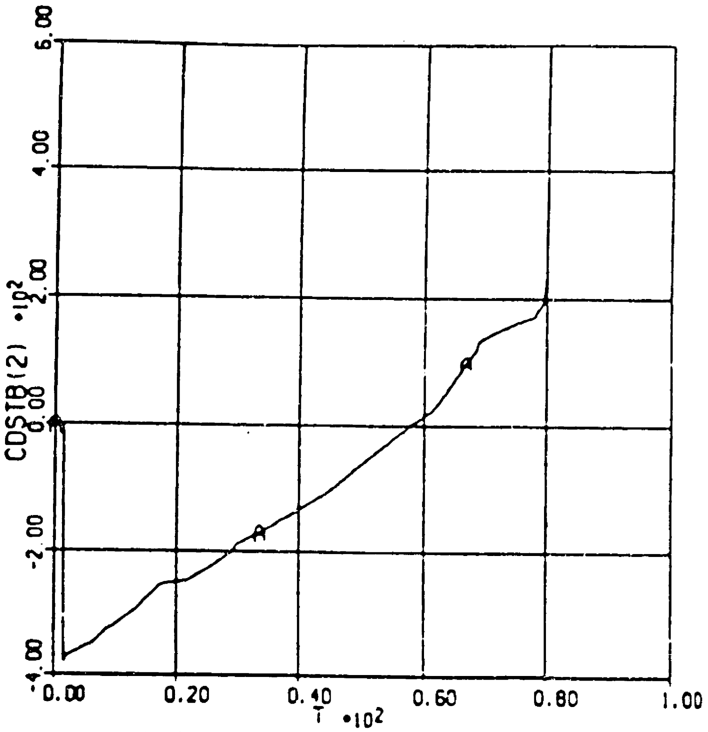


(c) Body z-axis

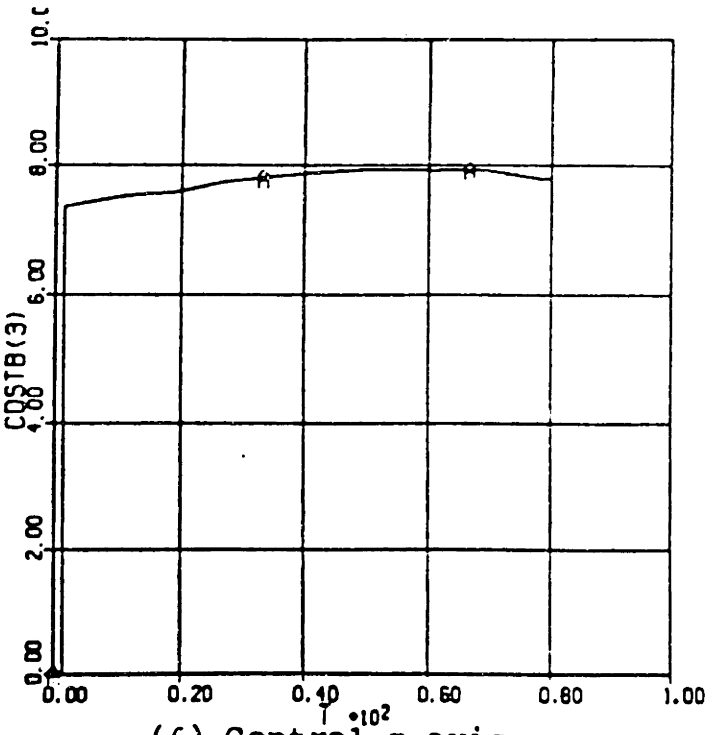
Figure 2-14: Total Disturbance Torques vs. Time



(d) Control x-axis



(e) Control y-axis



(f) Control z-axis

Table 2-X: Variable Magellan Mass Properties During VOI

MISSION PHASE	ZCM inches	MASS slugs	INERTIA (slug-ft**2)		
			IXXT	IYYT	IZZT
END OF LIFE	51.69	68.50	743.55	1420.20	1133.67
BEGIN MAPPING	50.85	70.47	744.51	1432.16	1133.67
PRIOR TO OTM	50.61	71.11	758.66	1435.31	1133.67
POST VOI	40.28	81.74	1208.77	1904.92	1139.97
	35.19	91.60	1627.82	2323.97	1180.60
	32.14	98.40	1732.32	2428.47	1195.64
	29.35	105.30	1873.42	2569.57	1217.74
	26.87	112.10	1983.12	2679.27	1225.14
	22.52	125.80	2135.12	2831.27	1262.04
	18.81	139.50	2248.12	2944.27	1284.14
	15.54	153.20	2383.32	3079.47	1298.94
	12.63	166.60	2466.92	3163.07	1328.44
	9.77	180.70	2565.12	3261.27	1343.14
	7.17	194.40	2659.62	3355.77	1361.64
	4.78	208.10	2717.42	3413.57	1380.04
	3.69	214.90	2757.82	3447.97	1387.44
PRE VOI.	2.94	222.15	2751.92	3448.08	1389.34
PRIOR TO TCM	3.50	226.69	2775.97	3472.12	1389.34
ARRAYS DEPLOYED	0.00	243.89	3233.70	3935.84	1625.90

Most of the variables used in this model have already been defined. They are all quite straightforward except for the fuel inertia, I_f, and the pendulum length, a. The fuel inertia is an effective inertia derived by a complicated finite-element model of fuel slosh implemented in a Martin Marietta computer program named HYDRO. Several fuel inertia data points were generated, and fuel inertia vs. propellant mass was fit with the empirical equation:

$$I_f=3.5\left(1-4\left(\frac{m}{188}-0.5\right)^2\right)-2.45\sin\left(\frac{\pi m}{188}\right) \tag{2.15}$$

(m in kilograms, I_f in kgm^2) This equation is plotted in Figure 2-16[3]. The parameter a, or the distance from the center of the propellant tank to the center of gravity of the fuel, is more complicated. However, it can be shown (Appendix A.2) that a is given by the equation:

---

[3] The HYDRO model could not be used in the Magellan simulator because it is far too complicated and time consuming. The slosh natural frequencies that it predicts, however, agree with those predicted by the pendulum model to within about 2.5%!

Table 2-XI: VOI Thruster Locations

REM NO.	5.0 lb Thruster NO.	LTO (inches)		
		XC	YC	ZC
1	13	78.9	9.5	10.0
2	14	9.5	78.5	10.0
3	15	-78.5	-9.5	10.0
4	16	-9.5	-78.5	10.0

(a) 5.0 lb Thrusters

REM NO.	100.0 lb Engine NO.	LTO (inches)		
		XC	YC	ZC
1	17	81.5	-2.5	-10.6
1	18	81.5	2.5	-10.6
2	19	2.5	81.5	-10.6
2	20	-2.5	81.5	-10.6
3	21	-81.5	2.5	-10.6
3	22	-81.5	-2.5	-10.6
4	23	-2.5	-81.5	-10.6
4	24	2.5	-81.5	-10.6

(b) 100.0 lb Thrusters

$$a = 2R\cos\left(\frac{\alpha}{3} + 240^{o}\right) \tag{2.16}$$

where R = the radius of the tank, F = the volume fraction of the tank filled with propellant, and:

$$\cos(\alpha) = -\sqrt{\frac{(2-2F)^2}{4}} \tag{2.17}$$

Equations (2.14) predict a double pole at a frequency f_2 and a double zero at frequency f_1 (see Appendix A.1 for details). The frequency f_2 is very close to the dominant frequency of the Magellan attitude control system (about 2 Hz), so some resonance problems might be expected. However, the zero frequency f_2 is very close to f_1 (usually within 0.05 rad/sec). Thus, the zeros mask most of the negative effects that the poles might otherwise cause. In practice, slosh is usually observed to have only a very small effect.

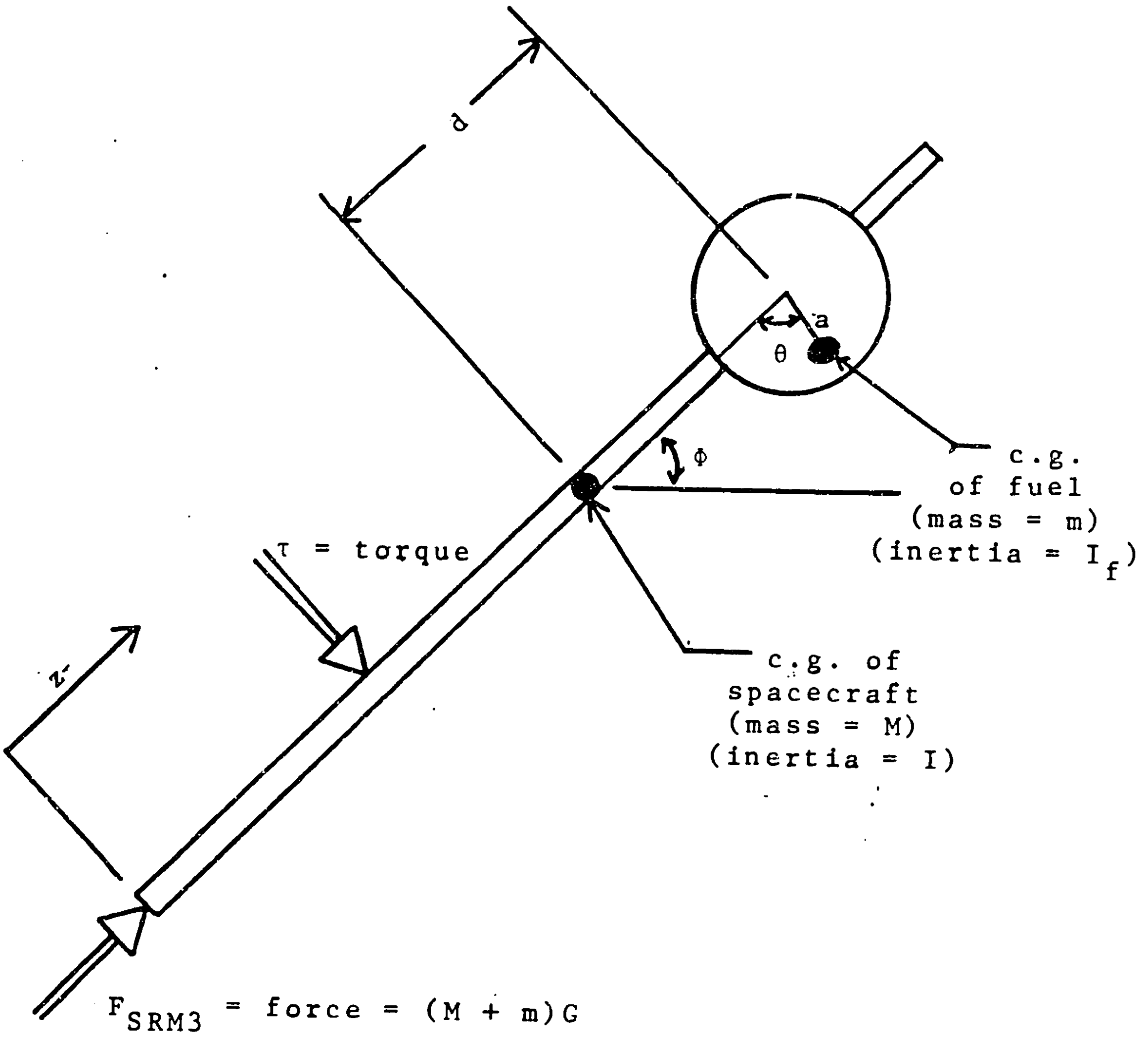


Figure 2-15: One-Axis Propellant Slosh Model

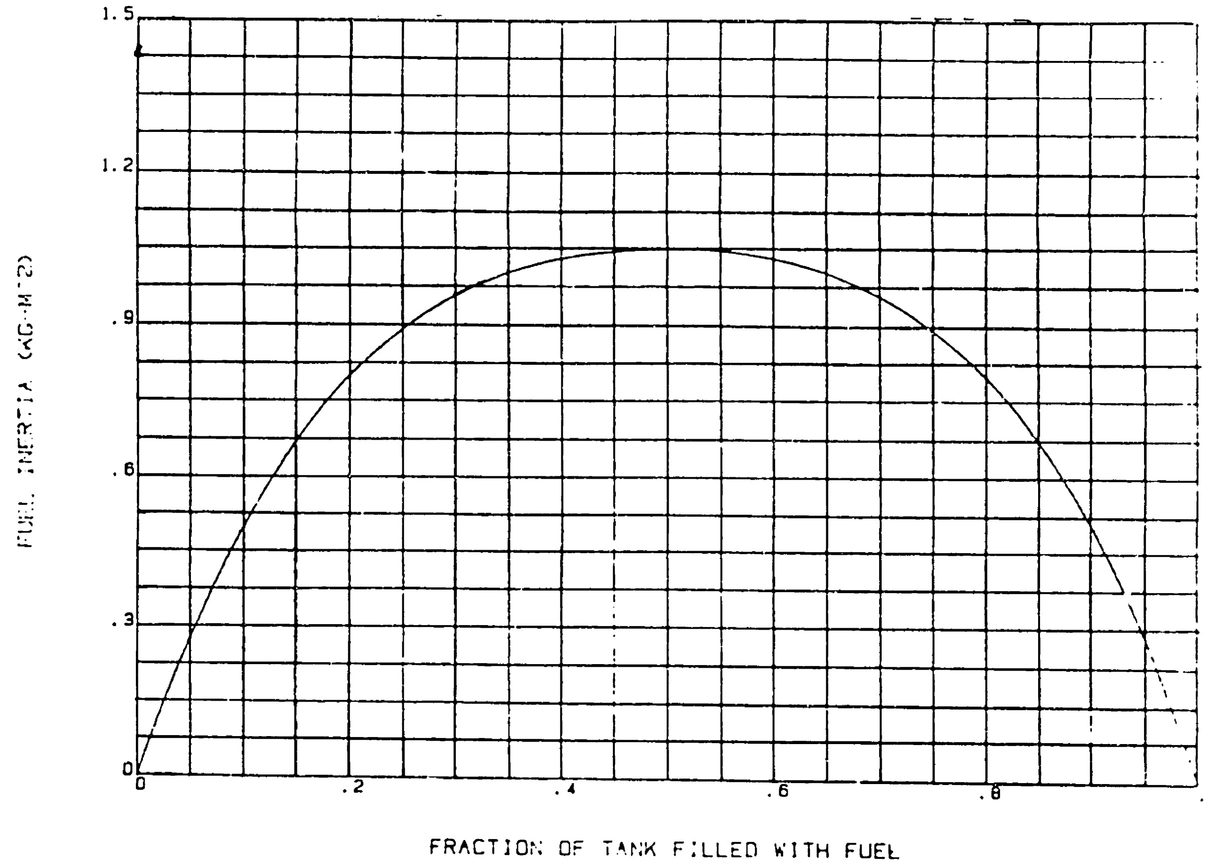


Figure 2-16: Fuel Inertia vs. Fraction of Tank Filled

Chapter 3

The Choice of Attitude Control System Designs

Before comparing the performances of various designs for thruster attitude control systems, one must obviously choose which system designs to compare. A literature search uncovered three main design philosophies. The first, which will be referred to as the "phase plane approach," uses a very simple controller and puts all of its complexity in the phase plane (see Figure 1-2). Indeed, it is possible to use no controller at all and simply feed attitude and rate errors directly into the phase plane block. Second, the "controller approach" puts all the complexity in the controller and uses a very simple phase plane, usually just a deadband. Finally, the "variable pulse width approach" avoids the use of a phase plane altogether. In this approach, the controller output determines the length of the thruster on-pulse. Any of these approaches is simpler than the Magellan system, in which both the controller and the phase plane are complex. Since one goal of this study is to simplify attitude control schemes, only the simpler approaches will be looked at here.

There are a plethora of different phase plane designs and controller designs in the literature, and many variable pulse width designs are possible. In this chapter, those which seem most promising for the comparative study will be chosen.

3.1 Phase Plane Designs

If attention is restricted to those phase plane designs that might perform well on a Magellan-type spacecraft, many of those designs currently found in the literature can be eliminated. Some, for example, require information from sensors that Magellan does not possess [18]. The schemes that look promising all depend upon the calculation of minimum time-fuel optimal switching curves in the phase plane.

Ogata [17] computes the minimum time switching curves. He assumes a control system as in Figure 3-1, where a switching element capable of delivering a torque of ±τ controls a simple inertia. If an attitude hold is commanded, the input to this system is $\phi_{desired}=0$. Thus,

$$\phi=\pm\frac{\tau}{I} \tag{3.1}$$

or

$$\dot{\phi}\frac{d\dot{\phi}}{d\phi}=\pm\frac{\tau}{I} \tag{3.2}$$

which can be integrated to obtain:

$$\dot{\phi}^2=\pm 2\frac{\tau}{I}+C \tag{3.3}$$

This equation describes two families of parabolas in the phase plane. The only two of these parabolas that intersect the origin, though, occur when C = 0 (see Figure 3-2). A trajectory starting on one of these parabolas travels along the parabola. Thus, the only way to reach the origin is to follow either path AO or path BO. The delivered torque ±τ must therefore be chosen such that it tends to force the system toward AO or BO. The resulting phase plane, with a sample trajectory, is shown in Figure 3-3. The switching lines can be shown to generate time optimal trajectories for the system depicted in Figure 3-1.

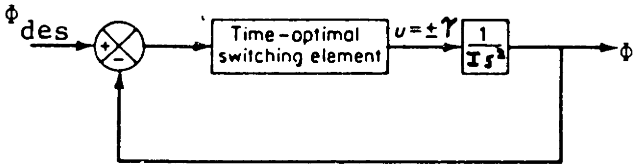


Figure 3-1: Generic Phase Plane Design

Notice, however, that in the regions between the parabola and the ϕ-axis, the spacecraft's velocity already tends to push it toward the parabola, without the need for any control torques (see Figure 3-4). If the phase plane is modified such that the thrusters do not fire in these regions, fuel consumption will be reduced. This may, however, occur at the expense of an increased response time.

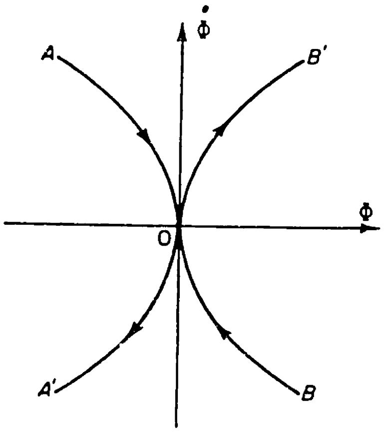


Figure 3-2: Equation (3.3) Curves Intersecting the Origin

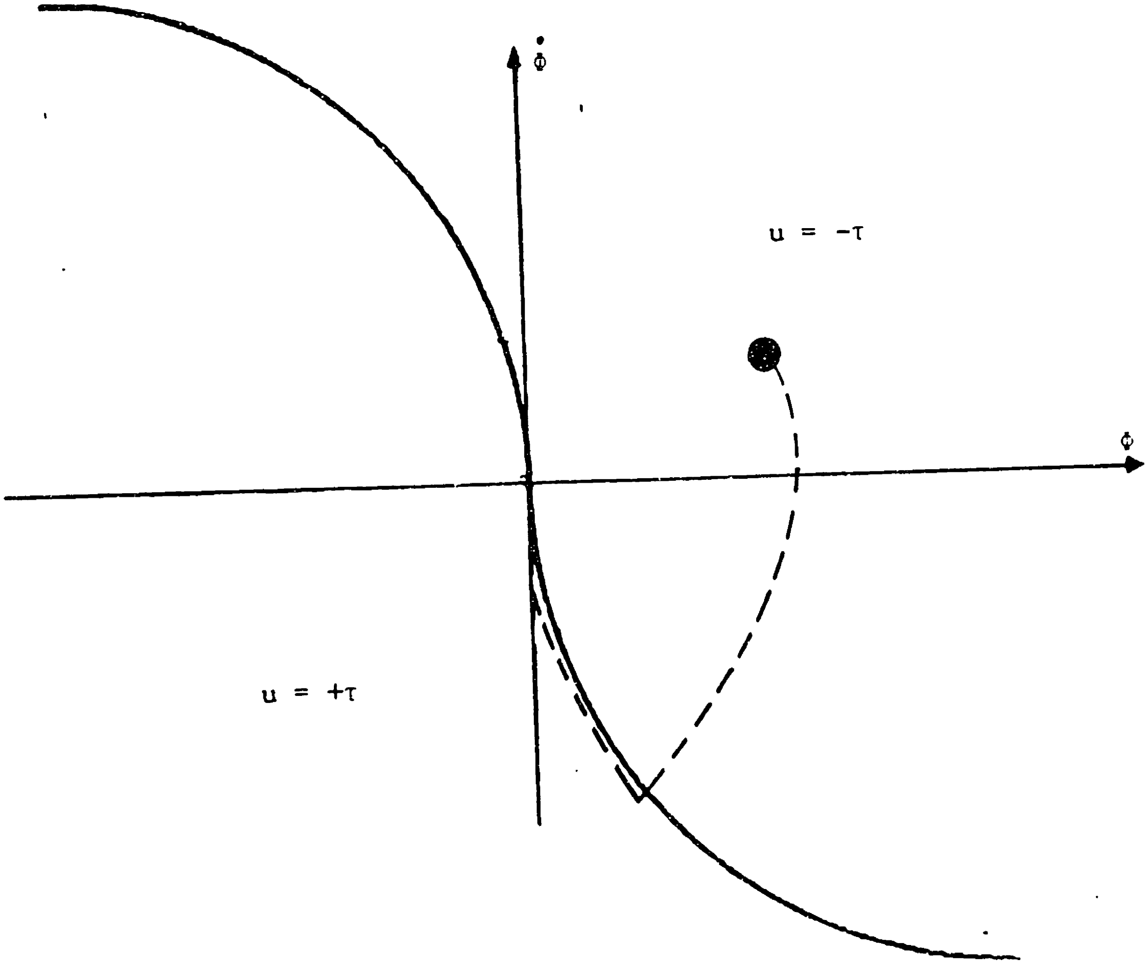


Figure 3-3: Minimum Time Phase Plane Switching Curves

On the Magellan (MGN), the thrusters can be set to fire for on-times less than a full

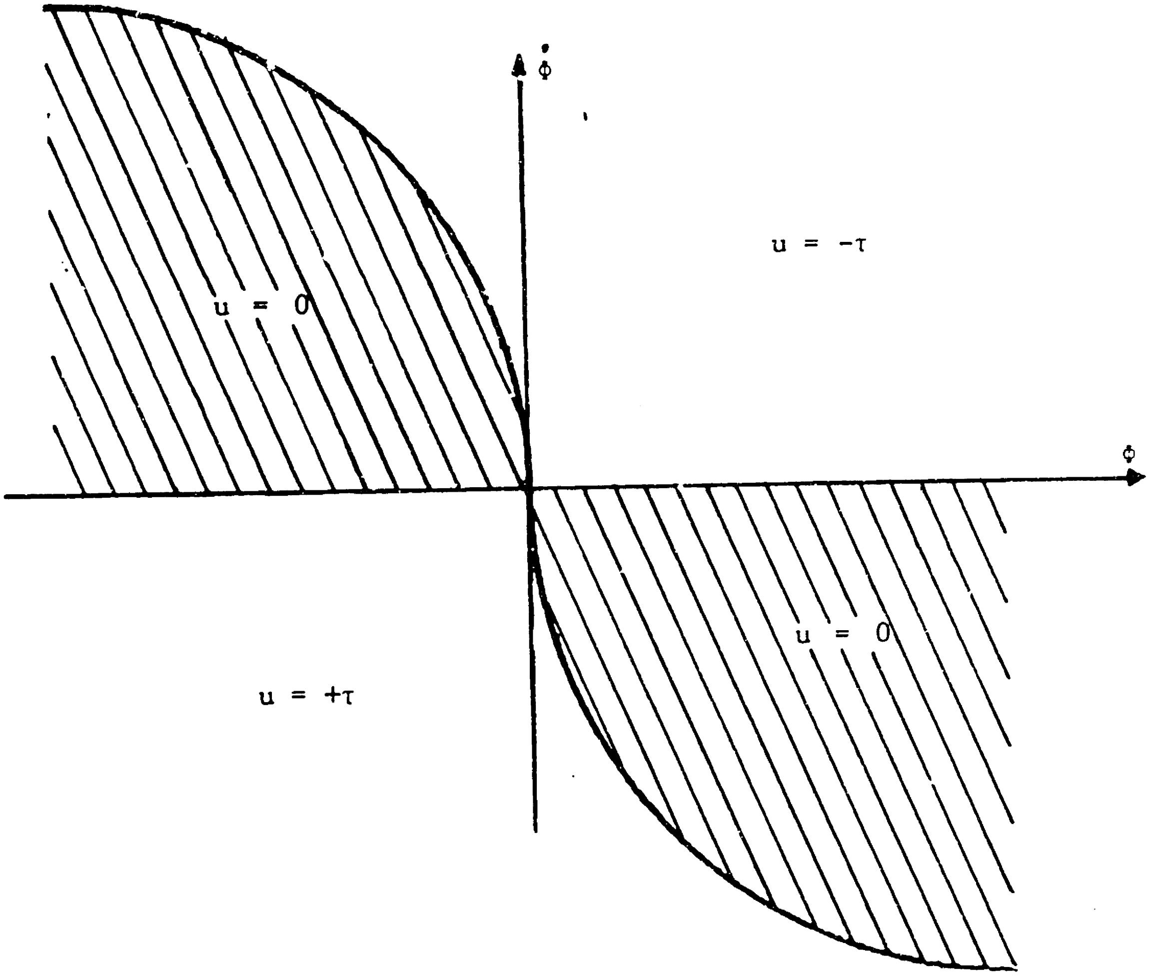


Figure 3-4: In the Shaded Regions, No Control is Needed

sampling interval. Thus, close to the origin, the MGN phase plane adds "short pulse" regions [8], as shown in Figure 3-5. In the normal attitude control case, region 3 indicates a thruster pulse duration equal to an entire sampling interval (133 msec), region 2 indicates a 33 msec pulse, and region 1 indicates an 11 msec pulse. During VOI, region 3 indicates that a pair of thrusters should fire for a full sampling interval (33 msec), region 2 indicates one thruster firing for a full interval, and region 1 indicates a 22 msec pulse for a single thruster. The sign of each region indicates the sign of the torque that is to be delivered. In region 0, the thrusters do not fire at all.

The Magellan phase plane has several other useful features. If the current state is

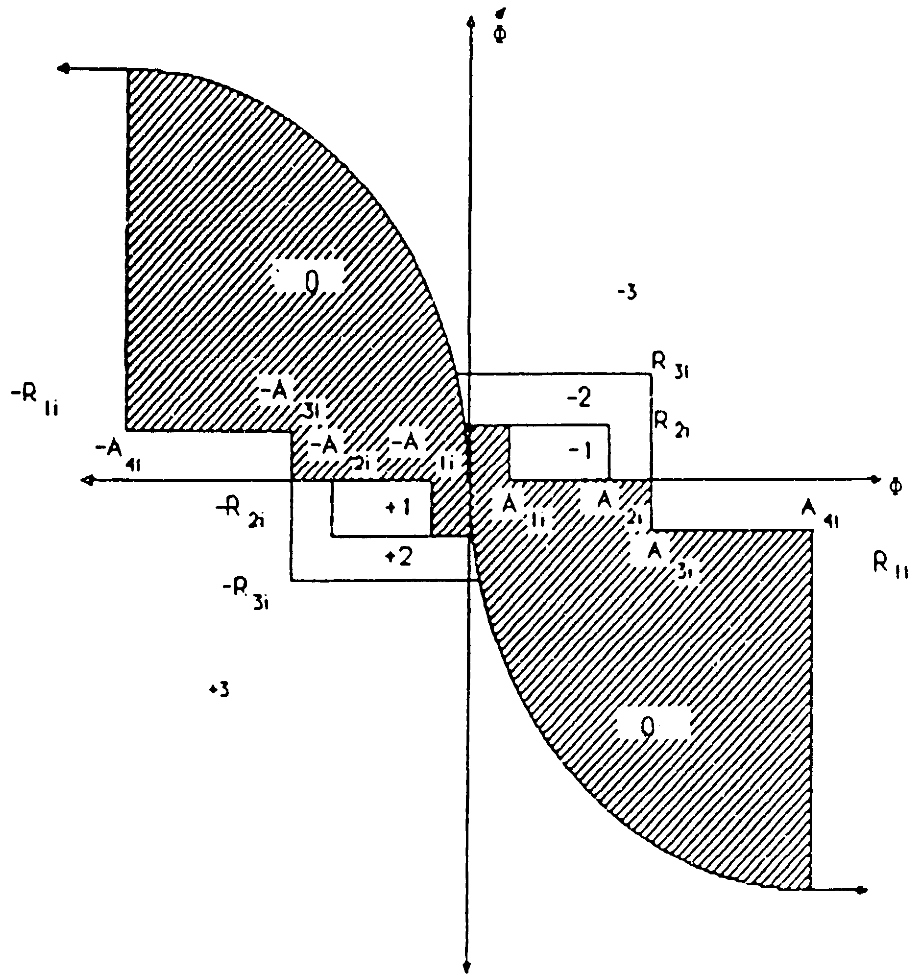


Figure 3-5: The Magellan Phase Plane

very far from the origin, it is desirable for it to zero out quickly regardless of the fuel cost; thus, ledge A4 limits the extent of region 0. And, although a trajectory between the parabola and the ϕ-axis tends toward the parabola without any thruster firings, it may do so more slowly than would be ideal. If this recovery rate is too slow, rate ledge R1 causes the thrusters to fire and accelerate the spacecraft in the proper direction.

This phase plane seems well worth consideration. In many cases, it has been shown to be adequate in controlling the spacecraft all by itself, without the help of any controller. Its complexity allows a great deal of fine-tuning of system response.

Many sources [11, 25, 20, 7] consider the more complicated problem of minimizing a weighted time-fuel index. If the index to be minimized is:

$$J=\int_0^{t_{final}}(\lambda+|u|)\,dt \tag{3.4}$$

where λ is a weighting factor and u = 1.0 x (sign of thruster torque), and if the terminal constraint $\phi(t_{final}) = 0$ is adopted, where ϕ is the spacecraft attitude, then the optimal switching curves can be shown to be as in Figure 3-6 (see Appendix B).

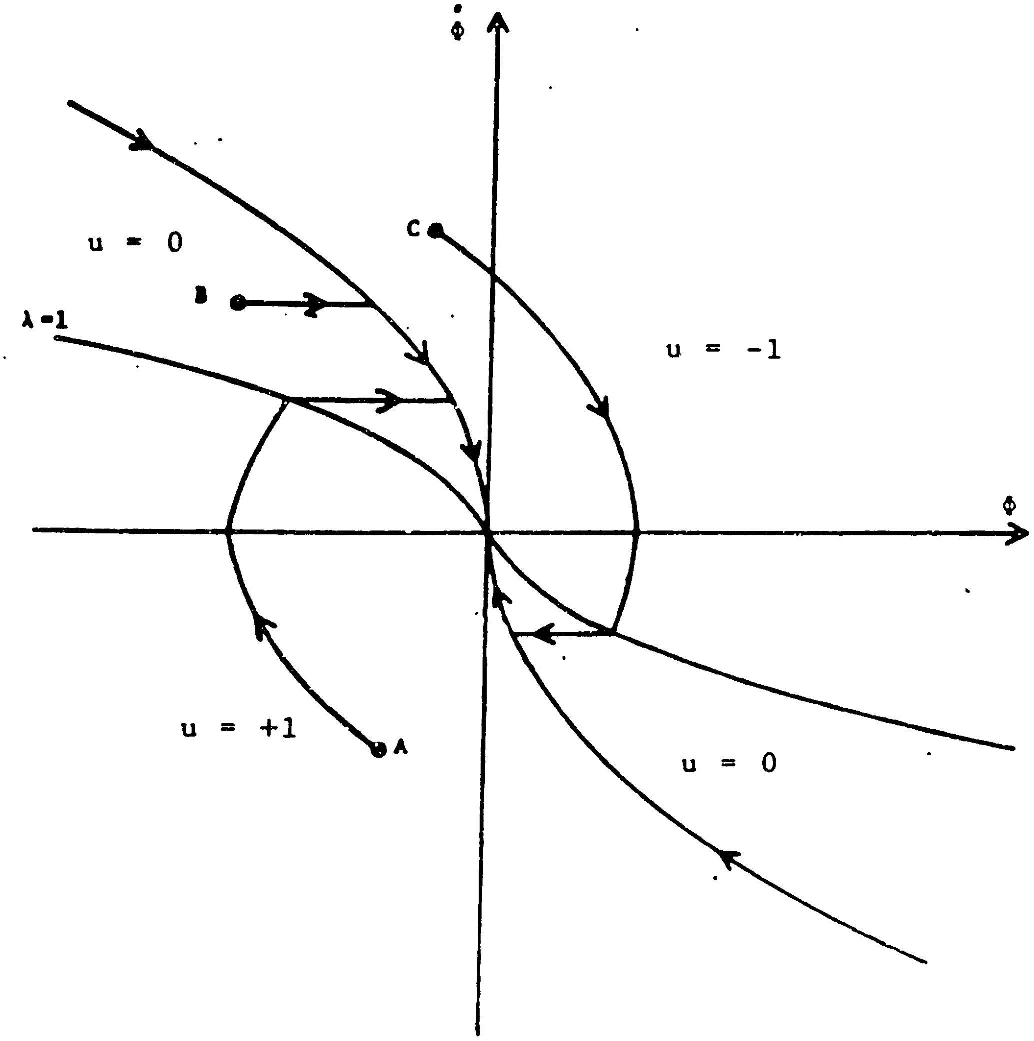


Figure 3-6: Minimum Time-Fuel Phase Plane Switching Lines

Although this phase plane is optimal in theory, Floyd [6] notes some problems with a similar phase plane derived by Wallace Vander Velde [20]. First, the full-on regions close to the origin cause intensive limit-cycling activity that wastes a lot of fuel. Second, and more important, the phase plane is derived in continuous time; however, any computer-controlled system is necessarily in discrete time. The control algorithm turns out to be incredibly sensitive to changes in the sampling time. Floyd [7] presents a discrete time algorithm that results in spacecraft trajectories similar to those that should ideally arise from the Figure 3-6 phase plane. Without going into any detail, Floyd's phase plane is depicted in Figure 3-7.

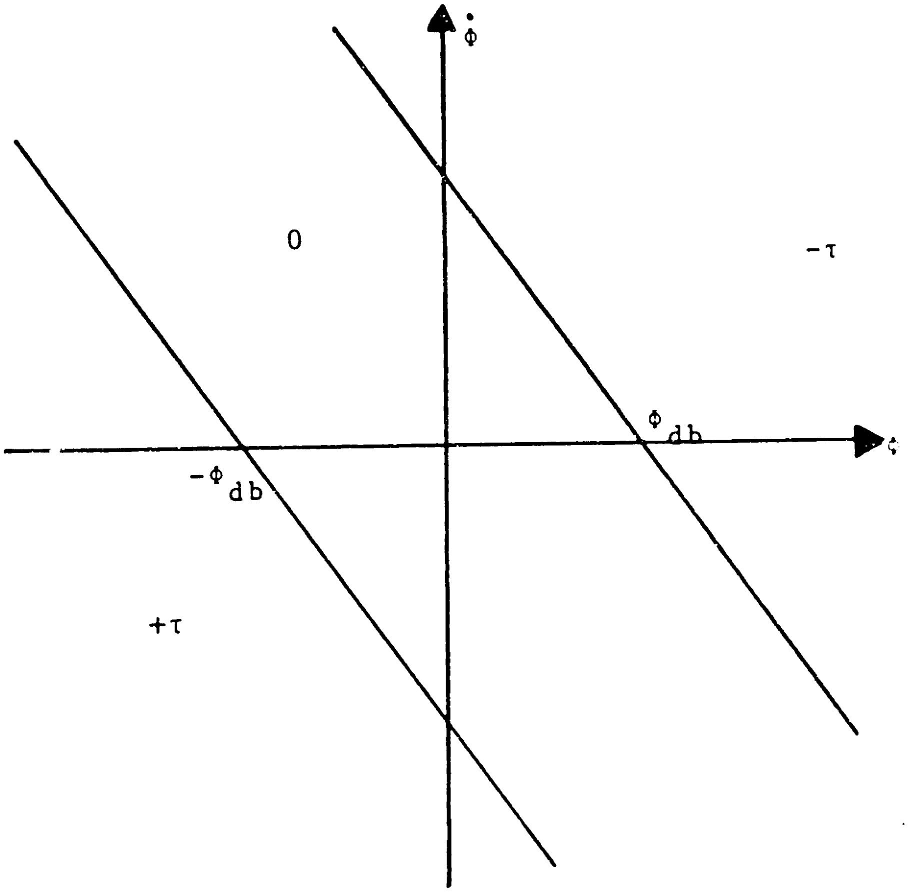


Figure 3-7: Floyd's [7] Phase Plane

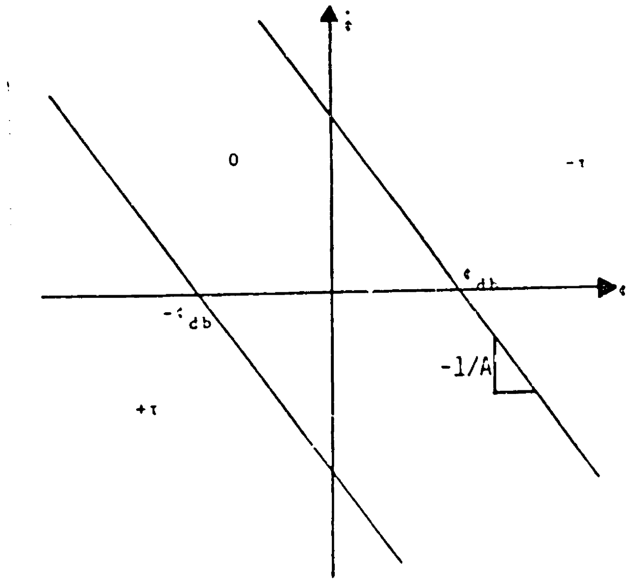
Many other sources recommend similar "slanted deadbands" [9, 13, 14]. By avoiding full-on regions near the origin, limit cycle performance is improved. Also, the system is less sensitive to sampling time than the minimum time-fuel algorithm. However, a "slanted deadband" is equivalent to simple proportional-derivative feedback with a deadband (see Figure 3-8). Both the "slanted deadband" and PD control result in negative thruster firings when $\phi + A\dot{\phi} > \phi_{db}$ and positive firings when $\phi + A\dot{\phi} < -\phi_{db}$. Consideration of the "slanted deadband" will thus be postponed until controller approaches are discussed, because in a controller design, system performance can be enhanced with integral, pseudo-rate, and other terms. These terms can not be added if the slanted deadband is considered to be a phase plane approach.

White, Colburn, and Boland [25, 26] resolve the limit cycle problem in a different way. The constraint that $\phi(t_{final}) = 0$ is not very realistic, they note. Thrusters are usually intended to get ϕ to within a certain deadzone, where a more precise control system can take over. Translating the minimum time-fuel curves to allow for this deadband, they obtain the phase plane depicted in Figure 3-9. They go on to show that this phase plane can be very closely approximated by the Rate-Ledge Controller (RLC) shown in Figure 3-10 (see Appendix B). These authors derive an algorithm which, given a deadband ϕ value and a value for the weighting factor λ, generates an approximately equivalent RLC (Appendix B, once again).

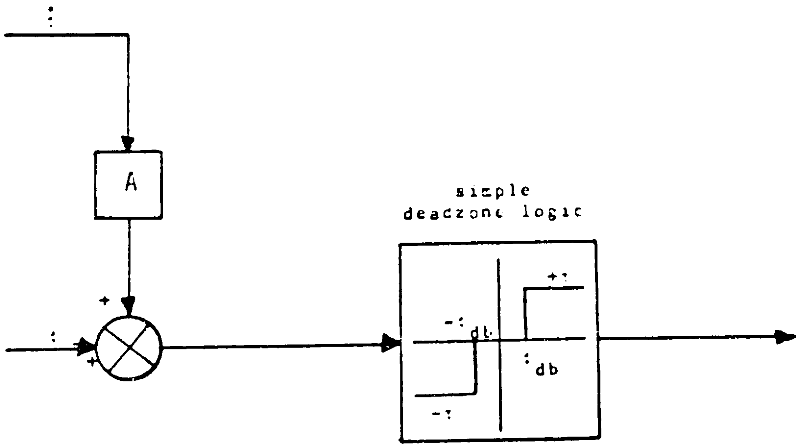
The RLC has been shown to perform well in hundreds of computer simulations of Skylab. Its proven performance and its approximation of optimal control laws make it a good candidate for consideration.

Unlike the MGN phase plane, the RLC has no short pulse regions. If such regions were added, performance might be improved. Thus, the RLC with short-pulse regions, as depicted in Figure 3-11, should also be considered.

In sum, three phase plane approaches to the design of thruster attitude control systems seem worthy of attention. First, the MGN phase plane, a modification of the minimum-time solution to the control problem, will be considered. Second, the RLC, an approximation to a minimum time-fuel solution, also seems promising. In fact, it seems more promising than the minimum time-fuel solution itself, since the latter is formulated in continuous time and does not translate well into discrete time. Third, the RLC with additional short pulse regions appears to be a good design to analyze.



(a) "Slanted Deadband"



(b) PD Control with Deadband

Figure 3-8: Equality of PD Control and "Slanted Deadband"

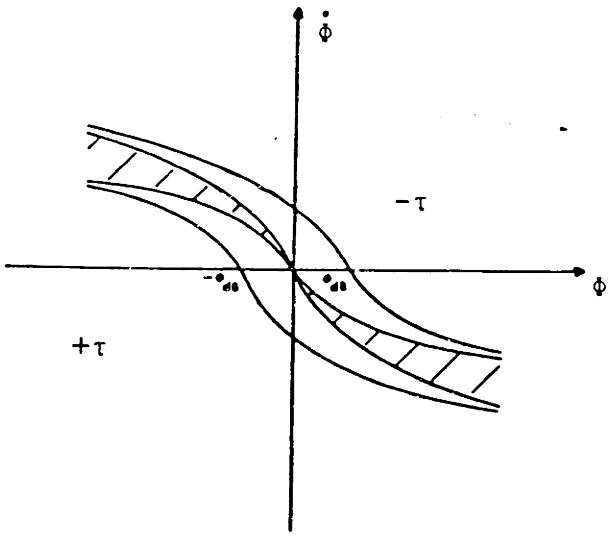


Figure 3-9: Time-Fuel Curves Translated for Deadband

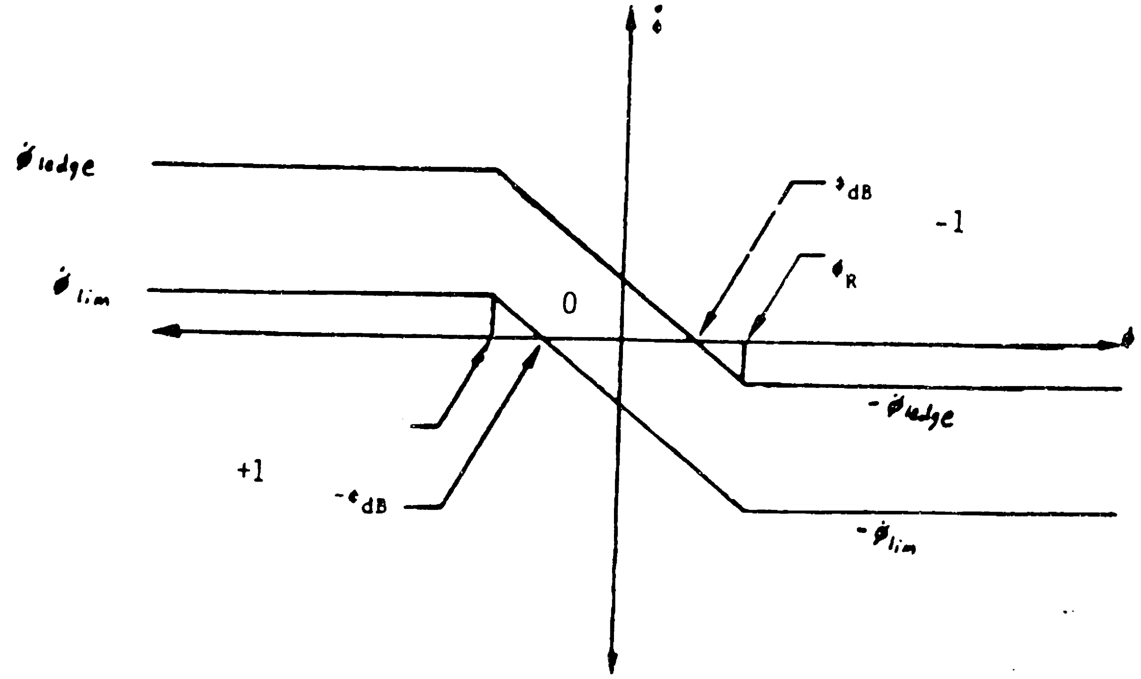


Figure 3-10: The Rate-Ledge Controller

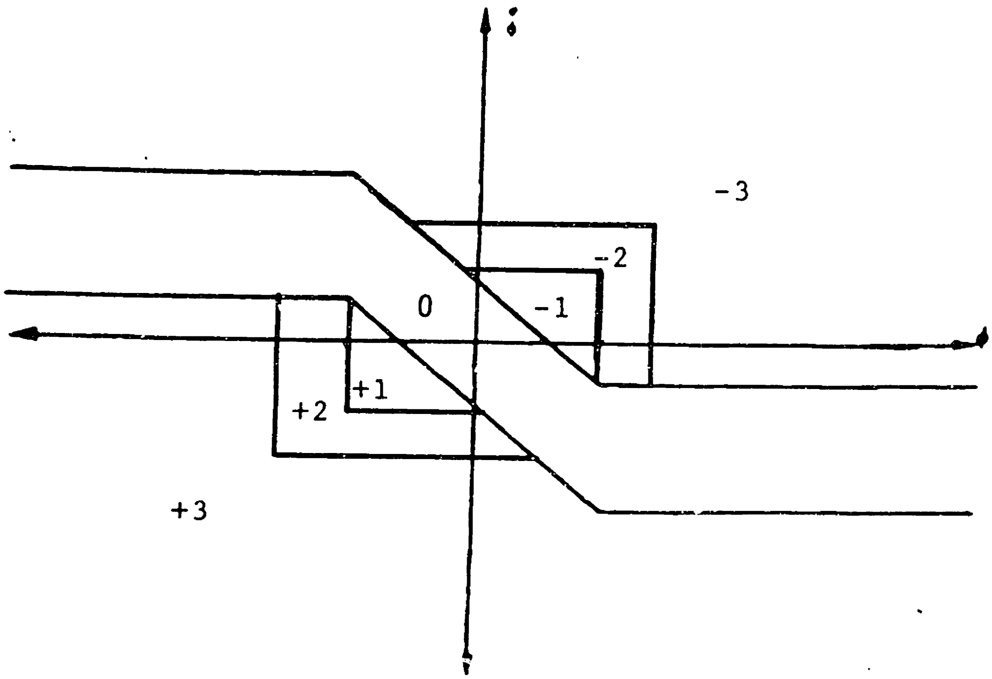


Figure 3-11: The RLC with Short Pulse Regions

3.2 Controller Designs

A "controller approach" design uses a complicated controller block with as simple a phase plane as possible. The simplest possible phase plane is an attitude deadband, depicted in Figure 3-12. Note that with this simple deadband, only one phase plane input is needed. Firing regions are independent of position along the phase plane ordinate. A controller design, then, needs only output one signal.

In the literature, the two major approaches to controller design are pulse-width-pulse-frequency (PWPF)[4] (see Figure 3-13a) and pseudo-rate (PSR) (see Figure 3-13c) control [16, 1, 28, 27]. Bittner, Fischer, and Surauer [3], however, noticed that with some block diagram algebra, these two controllers can be shown to be equivalent (Figure 3-13). Thus, only one of the two needs to be considered. Pseudo-rate control will be chosen.

---

[4] Although its name makes it sound like a variable pulse width approach, most discrete time pulse-width-pulse-frequency schemes cause pulse widths that are integer multiples of the sampling interval. Thus, PWPF is actually a controller approach.

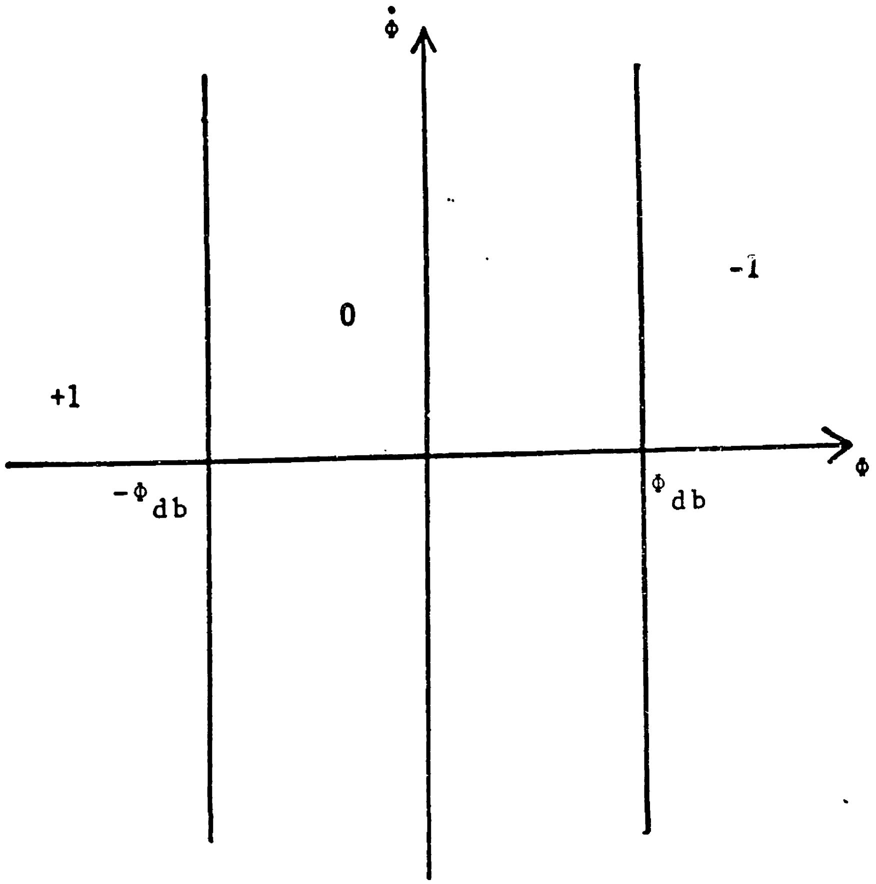


Figure 3-12: The Simplest Phase Plane: A Deadband

Pseudo-rate control is intended to reduce control loop delays. A controller that uses regular feedback must wait for various system time delays before it can respond to a thruster command. By feeding back directly from the phase plane output and not waiting for these time delays, the pseudo-rate term can act immediately to reduce the error signal, thus anticipating the effects of a thruster firing. The PSR transfer function,

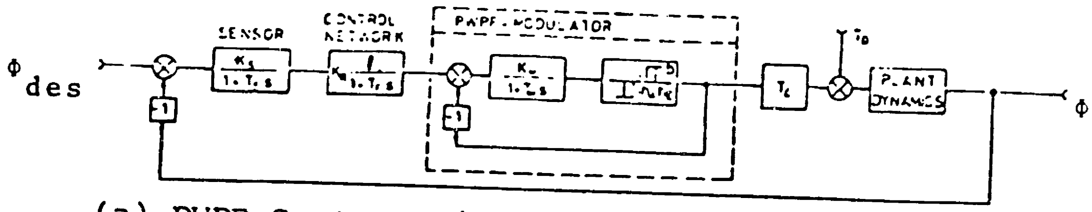
$$H(s)=\frac{kb}{s+b} \tag{3.5}$$

is approximately equal to:

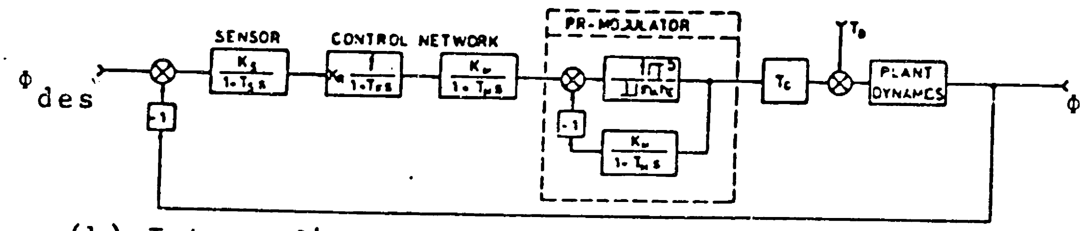
$$H(s) = D + Es \tag{3.6}$$

Thus, PSR control is roughly equivalent to proportional-derivative control. The parameter b in transfer function (3.5) is varied as follows:

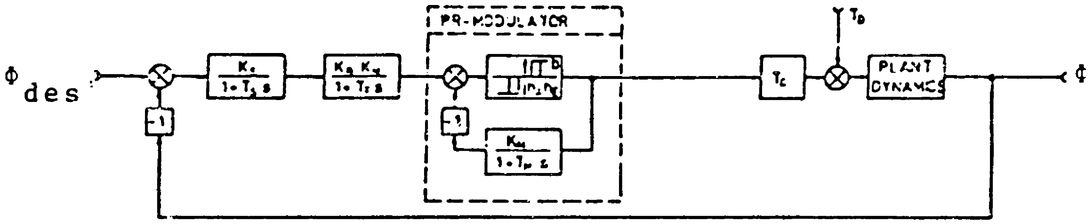
```
b = b1 when the thrusters are commanded on
b = b2 when the thrusters are commanded off.
```



(a) PWPF Control with Lead/Lag Compensator



(b) Intermediary Step Between PWPF and PSR Control



(c) PSR Control with Lag Compensation

Figure 3-13: Equivalence of PWPF and PSR [3]

b_1 can be chosen to allow rapid response to a thruster pulse. After the pulse, the pseudo-rate term should slowly decay as the control system has a chance to catch up to the effects of the pulse. b_2 can be set to allow this slow discharge. $1/b_1$ is usually referred to as the "charging time constant," while $1/b_2$ is the "discharging time constant."

The attitude control system on the Jet Propulsion Laboratory's Galileo spacecraft, which will be launched to Jupiter in the early 1990's, uses a pseudo-rate term. In addition to this term and a low-pass filter, the Galileo utilizes ordinary PID control [2]. The complete Galileo controller is shown in Figure 3-14[5]. This design, which combines almost all of the control terms found in the literature, is definitely worthy of investigation. In fact, since it *does* include all these terms, it is really the only controller needing consideration.

Modem approaches to controller design, such as eigenstructure placement or sliding surface control, all seem very difficult, if not impossible, to apply to this problem. A closed-loop scheme based on optimal control theory would be similarly difficult to design. An open-loop optimal scheme will become useful when variable pulse width approaches are considered.

3.3 Variable Pulse Width Designs

Finally, a variable pulse width design eliminates the phase plane altogether. A variable pulse width design varies thruster pulse lengths continuously (or, more practically, in small but discrete steps). Unlike a phase plane, which determines which thrusters should fire, a variable pulse width design must determine which thrusters should fire and for *how long*.

Not much work seems to have been done in this area. In fact, only one control

---

[5] Actually, the Galileo controller also includes a "path guidance" term which will not be considered here. For a description of this term and an explanation of why it is not to be considered, see section 3.4.

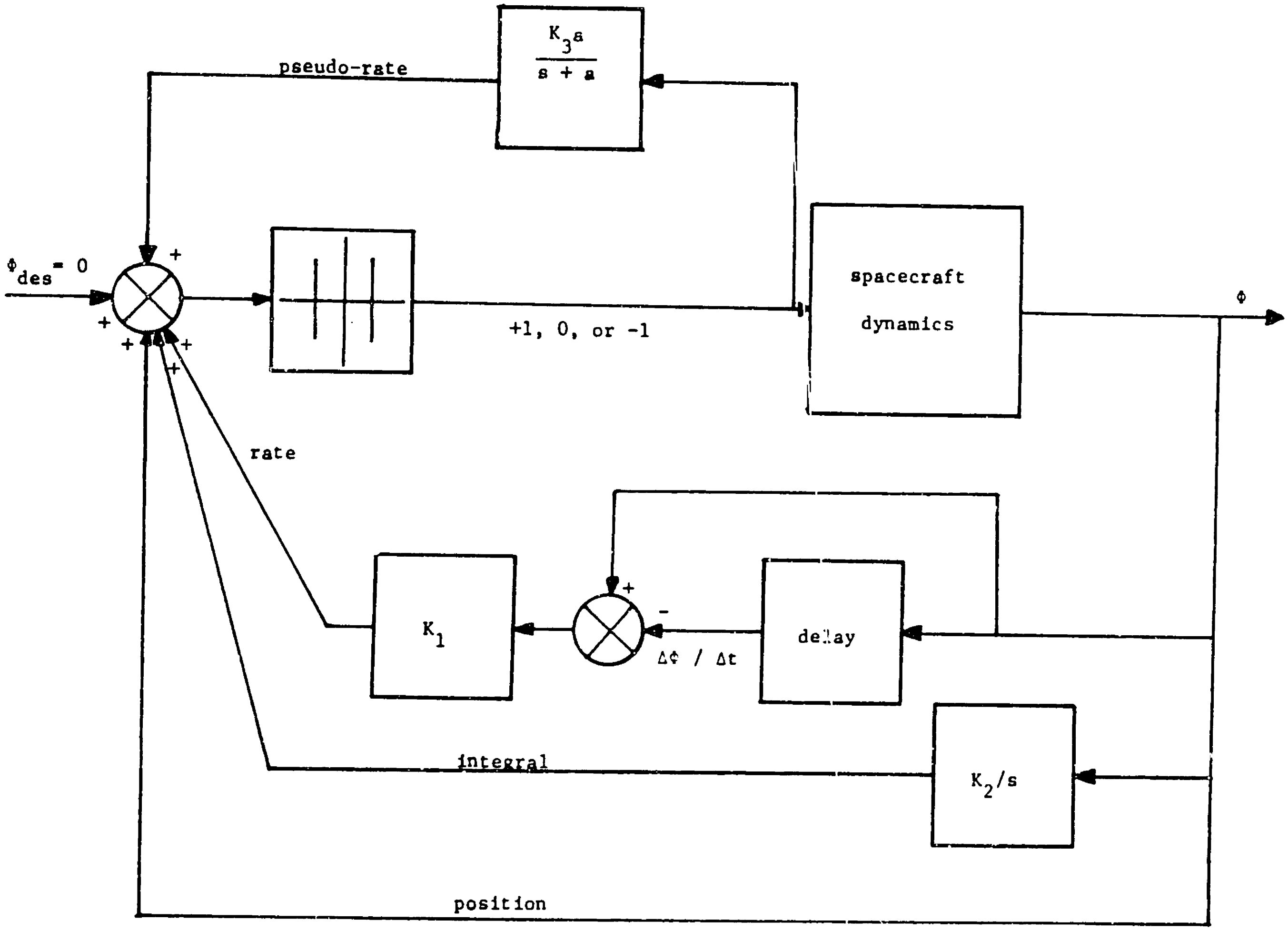


Figure 3-14: The Galileo Controller

algorithm based on this approach was found. Clark, Dumas, and Fosth [4] suggest a very simple and intuitive scheme. This scheme utilizes an ordinary controller, such as, perhaps, the Galileo controller. Thruster pulse width is determined proportional to the controller output signal, as depicted in Figure 3-15. When the signal is below the minimum reliable pulse width deliverable by the thrusters, the thrusters are simply commanded not to fire. The maximum possible pulse width is one sampling interval.

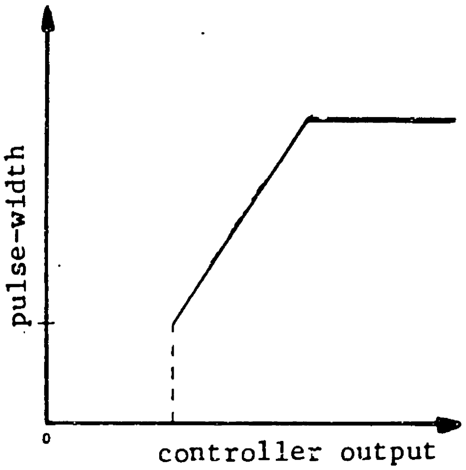


Figure 3-15: Clark Pulse-Width vs Controller Output

Far from the phase plane[6] origin, controller output is large enough that no difference is distinguishable between the response of an ordinary controller approach and that of the Clark scheme. In both cases, the thrusters usually fire for full sampling intervals. Near the origin, however, this variable pulse width scheme has the potential to greatly enhance performance. The Clark, Dumas, and Fosth design is intended to improve steady state performance, but probably will have little effect on transient response. Still, it is well worth consideration.

A design that affects performance of transient, as well as steady state, response can be derived from optimal control theory. Using some convenient time-fuel performance

---

[6] While the phase plane is not needed in the construction of a variable pulse width design, it will be useful to refer to it in this discussion anyway.

index and assuming the simple case of continuously varying control torques, an open-loop optimal control law can be devised. This control law is a function of system parameters and the initial conditions ϕ_0 and $\dot{\phi}_0$. At the beginning of each sampling interval, the current state can be taken as the initial condition, thus introducing a sort of feedback into the open-loop control. The optimal torque profile is found and integrated over the next sampling interval. Then the thrusters can be commanded to fire for a period of time that realizes the same $\int \tau \, dt$ as the optimal control.

The optimal control problem can be formulated as:

$$\textit{Given}\cdot\ \ddot{\phi}=\frac{\tau}{I}\quad (\tau \textit{ assumed continuous}) \tag{3.7}$$

$$\textit{with terminal constraints}\cdot\ \dot{\phi}(t_{final})=\phi(t_{final})=0$$

$$\textit{minimize the index}\cdot\ J=\int_0^{t_{final}}(k_w{}^2+\tau^2)\,dt$$

This particular performance index was chosen because it facilitates solution of the problem. Also, if τ can only take on the constant values $-\tau$, 0, or $+\tau$ and k is carefully scaled, this index is equivalent to the index:

$$J=\int_0^{t_{final}}(\lambda+|u|)\,dt \tag{3.8}$$

which was used in the derivation of many phase plane designs. The solution to this problem can be shown to be (Appendix C):

$$\tau=-\frac{1}{2I}(\nu_1-\nu_2(t-t_{final})) \tag{3.9}$$

where

$$\nu_1=-2Ik_w$$

$$\nu_2=\frac{4I^2\dot{\phi}_0-2\nu_1 t_{final}}{t_{final}{}^2}$$

$$t_{final}=\frac{2I}{\nu_1}(-\dot{\phi}_0 I-\sqrt{\dot{\phi}_0{}^2 I+6k_w I\phi_0})$$

Integrating from 0 to the end of one sampling interval:

$$\int_0^{t_{sample}} \tau\, dt = -\frac{1}{2I}\left(v_1 t_{sample} - v_2\left(\frac{1}{2} t_{sample}{}^2 - t_{sample} t_{final}\right)\right) \tag{3.10}$$

Now, if an on-off torquing capability is imposed, in the form:

$$\int_0^{t_{sample}} \tau\, dt = \Gamma \int_0^{t_{pulse}} dt \tag{3.11}$$

where Γ is the constant torque capacity of the thrusters, it can be shown that the thruster pulse width should be:

$$t_{pulse} = -\frac{1}{2I\Gamma}\left(v_1 t_{sample} - v_2\left(\frac{1}{2} t_{sample}{}^2 - t_{sample} t_{final}\right)\right) \tag{3.12}$$

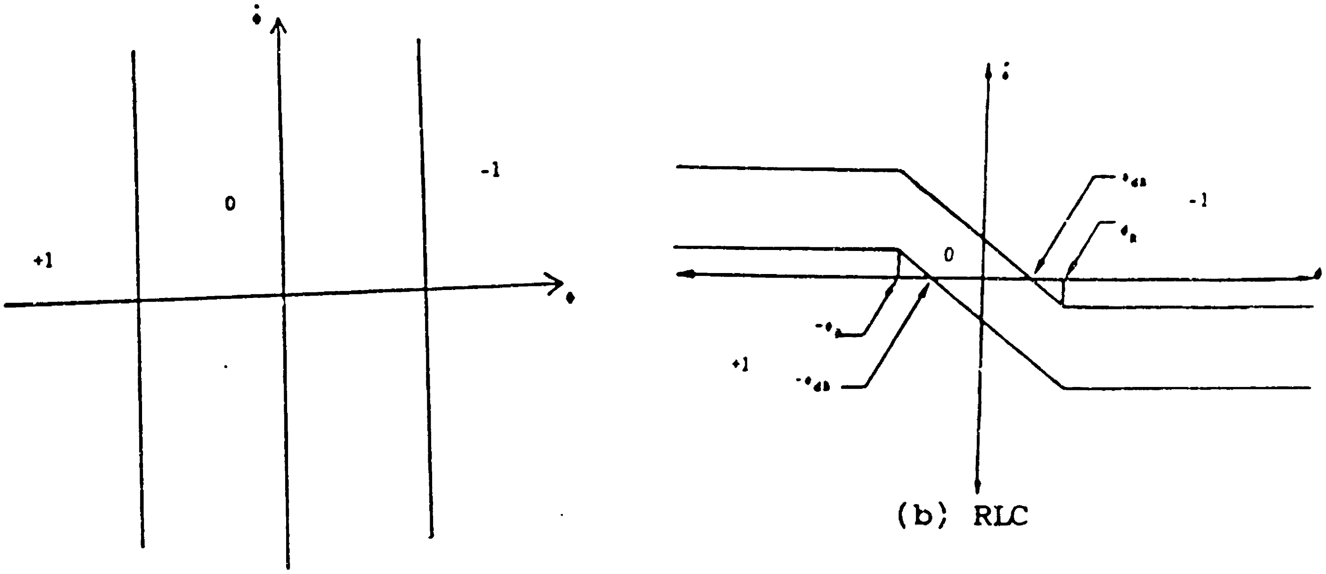
where the sign of t_{pulse} indicates the sign of the commanded thruster torque.

Equation (3.12) is the variable pulse width control law. Only one detail of it needs clarification. The parameter t_{final} is calculated with a square root term, namely:

$$\sqrt{\dot{\phi}_0{}^2 I + 6k_w I \phi_0}$$

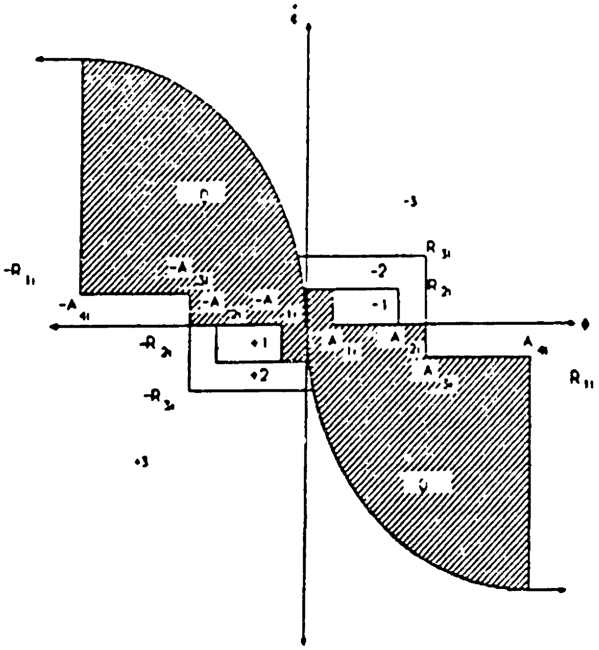
and it must be assured that the value under the square root is non-negative. This is actually quite easy to do. As Figure 3-16 shows, thruster control phase planes are symmetric when rotated 180 degrees about the origin. A point rotated in this way results in similar firings, with opposite sign, to a point at its original location. Thus, if $\dot{\phi}_0{}^2 I^2 + 6kI\phi_0 < 0$, it is valid to rotate 180 degrees by setting $\phi' = -\phi$ and $\dot{\phi}' = \dot{\phi}$. Now, $\dot{\phi}'_0{}^2 I^2 + 6kI\phi'_0 > 0$. If the sign of the calculated t_{pulse} is flipped, the resulting thruster commands will be correct. In practice, this should be done whenever $\phi_0 < 0$, since this is an easier and more restrictive test than $\dot{\phi}_0{}^2 I^2 + 6kI\phi_0 < 0$.

In sum, there are two variable pulse width designs that will be considered. The Clark, Dumas, and Fosth scheme is intended to improve the steady state performance of a controller. The "suboptimal pseudo-open-loop" scheme affects both steady state and transients.



(a) Simple Deadband

(b) RLC



(c) MGN Phase Plane

Figure 3-16: Phase Planes are Symmetric Rotated 180 deg

3.4 Modifications to the Three Basic Approaches

Many thruster attitude control systems employ some interesting modifications to enhance performance. These modifications do not fit into the phase plane, controller, or variable pulse width categories, and thus will not be considered in this thesis. They do, however, deserve to be mentioned. These modifications are not capable of controlling a spacecraft by themselves, but they can be used to improve the performance of many of the control system designs discussed in this chapter.

One modification to an attitude control system that helps performance is known as attitude limiting. If the attitude error becomes too large, an integral term in a controller can grow large as well. The control system reacts to this by very rapidly reducing the attitude error, but it can take a long time for the integral term to reduce itself accordingly. This artificially high term can damage system performance and even cause instability, unless an attitude limit is set. If the attitude error is greater than this limit, the control system is misled into believing that the error is equal to the limit, thus preventing excessive build-up of the integral term. The dangers of instability usually associated with integral control are greatly reduced. Attitude limiting has been found to improve the performance of the Magellan control system.

Two related modifications that are intended to help compensate for system time lags are known as attitude quantization and inhibition. When phase plane or variable pulse width designs command short pulses, the gyros may require some time to catch up to the current attitude. If the attitude error is sampled before the gyros have caught up, the thrusters may fire again, despite the fact that the attitude may actually be well within the no-firing deadband. The inhibitor scheme [4] simply inhibits firing of the thrusters for a certain number of sampling intervals after a short pulse. Attitude quantization requires a comparison of the currently detected attitude error with the error detected at the time the short pulse was commanded. If the difference between these two values is less than a specified constant, the gyros are assumed not to have caught up yet, and thruster firing is inhibited. Attitude quantization has been found useful on the Magellan spacecraft.

Finally, the Galileo controller utilizes a path guidance term [2]. Any attitude control system discussed so far tries to reduce attitude errors to zero. However, during the SRM burn, SRM misalignments and side forces actually make the zero error condition less than ideal. As Figure 3-17 indicates, the spacecraft should ideally be rotated a small angle so that the net SRM thrust vector exactly coincides with the desired flight path. Using estimates of spacecraft disturbances, the path guidance term approximates this angle and causes the control system to control to this angle instead of to a zero attitude. The reason path guidance will not be considered in this thesis is that, for simplicity, all design criteria will be with respect to a zero attitude. Path guidance will only hurt the attainment of a zero attitude. Since the path guidance angle is typically very small, neglecting it should cause no significant problems.

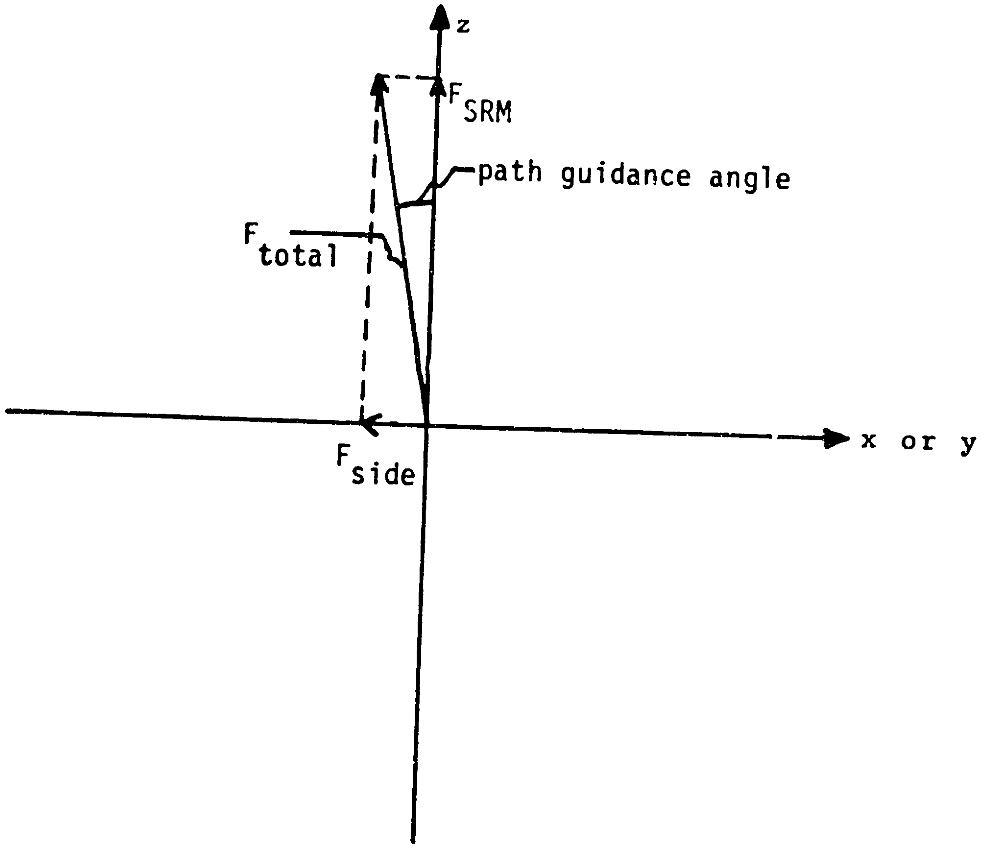


Figure 3-17: Side Forces Make 0 Attitude Undesirable

Chapter 4

Criteria for Design Evaluation

Now that attitude control systems have been chosen, a set of criteria by which these designs can be evaluated must be devised.

4.1 Normal Attitude Control

In the normal attitude control case, there are no disturbance torques. Starting with $\phi_0 = 2$ degrees and $\dot{\phi}_0 = 1$ deg/sec on each control axis, it will be observed how well each control system reduces these errors to zero. The following criteria are desirable:

Transient Response:	**short settling time** **low fuel consumption** **small rate excursions** **small attitude excursions**
Steady State:	**low fuel consumption** **small limit cycle rate amplitude** **small limit cycle attitude amplitude**
Overall:	**ease of assigning parameter values**

In order to compare the performances of different control systems, it would be nice to be able to assign each system a number from 1 to 10 indicating how well it meets these criteria. The following formulae were developed to allow the assignment of a "score" to each system in this manner.

4.1.1 Transient Response Scoring

4.1.1.1 Settling Time

The settling time score is calculated by the equation:

$$score = \frac{minimum\ time}{observed\ time} \times 10 \tag{4.1}$$

Since settling times less than 60 seconds were never observed, this value was used as the "minimum time."

4.1.1.2 Fuel Consumption

From equation (2.6), it is known that:

$$Fuel\ Consumed = \int_0^t \frac{F}{I_{sp}} dt \tag{4.2}$$

or

$$Fuel\ Consumed = \int_0^t \frac{I|\ddot{\phi}|}{rI_{sp}} dt \tag{4.3}$$

assuming $\tau = I\ddot{\phi}$ and r = the thruster moment arm. If $\dot{\phi}(t_{final}) = 0$, the total fuel consumption is:

$$Fuel\ Consumed = \frac{I|\dot{\phi}_0|}{rI_{sp}} \tag{4.4}$$

This was calculated for all three control axes and the results were added. This sum represents the minimum possible fuel consumption, and equals 0.028 kg. The fuel consumption score, then, is simply:

$$score = \frac{minimum\,fuel}{observed\,fuel} \times 10 \tag{4.5}$$

The "observed fuel" is the fuel consumed from t = 0 to t = (the longest settling time among the three axes). Some axes may settle before this time, causing a quantity of steady state fuel to be included in this value. However, steady state fuel consumption is usually much lower than consumption in transient response. Thus, this is still a very good approximation.

4.1.1.3 Rate Excursions

By requiring low rate excursions, it is meant that it is undesirable for the rate error about any axis to get much worse than its initial condition. The rate excursion score is calculated by:

$$score = \frac{\dot{\phi}_{max-a}^{\ 2} - \dot{\phi}_{max-o}^{\ 2}}{\dot{\phi}_{max-a}^{\ 2} - \dot{\phi}_0^{\ 2}} \times 10 \tag{4.6}$$

where $\dot{\phi}_{max-a}$ is the maximum allowable rate error and $\dot{\phi}_{max-o}$ is the maximum observed rate error. On Magellan, rate errors greater than 8 deg/sec cause the gyros to saturate. Thus, this value will be used for $\dot{\phi}_{max-a}$.

4.1.1.4 Attitude Excursions

Similar to the rate excursion score, the attitude excursion score is given by the formula:

$$score = \frac{\phi_{max-a}{}^2 - \phi_{max-o}{}^2}{\phi_{max-a}{}^2 - \phi_0{}^2} \times 10 \tag{4.7}$$

where ϕ_{max-a} is the maximum allowable attitude error and ϕ_{max-o} is the maximum observed attitude error. The value $\phi_{max-a} = 20$ degrees was used for this thesis.

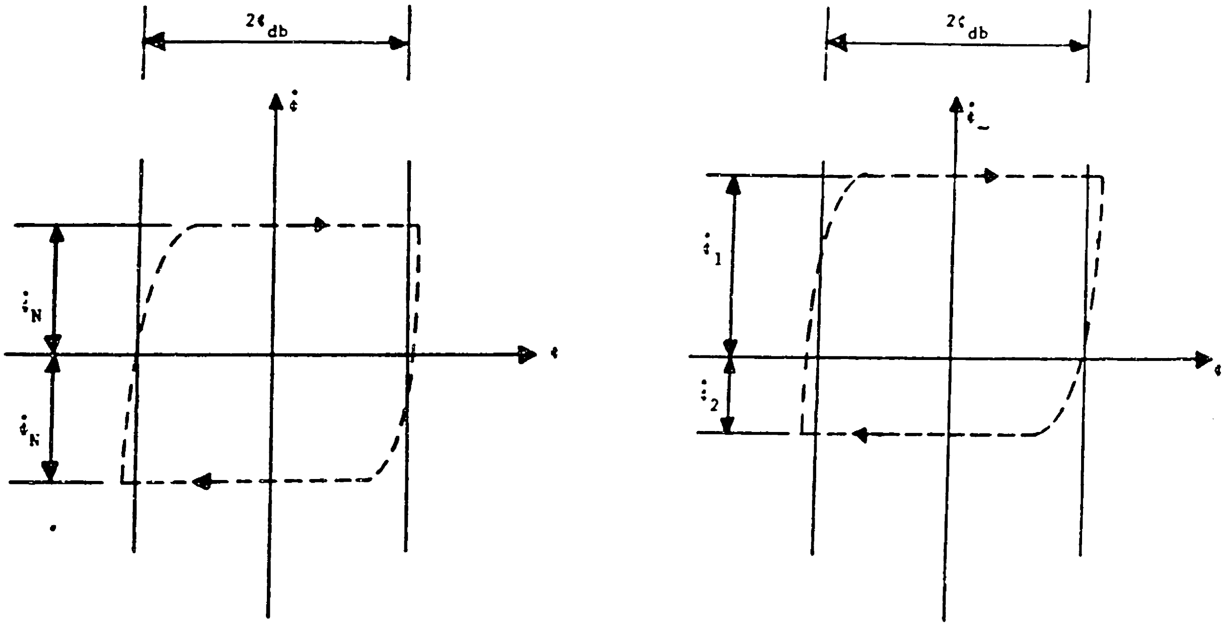
4.1.2 Steady State Scoring

In steady state, comparisons of control system performance are slightly more difficult. Both of the phase plane trajectories shown in Figure 4-1 are possible. Yet the trajectory in Figure 4-1b will be observed to have a higher maximum limit cycle rate, since the limit cycle is not centered on the origin. Also, the 4-1b trajectory has a lower frequency than the 4-1a trajectory, and thus will be observed to use less fuel in a given period of time. To allow fair comparisons, these trajectories must somehow be normalized.

For simplicity, assume the simple deadband phase plane shown in Figure 4-1. A trajectory of the 4-1b type will be normalized into the evenly-spaced 4-1a trajectory. The convention that both $\dot{\phi}_1$ and $\dot{\phi}_2$ are positive is adopted (where these variables are defined in the figure), the frequency is approximately given by the expression:

$$\frac{1}{f} = \frac{2\phi_{db}}{\dot{\phi}_1} + \frac{2\phi_{db}}{\dot{\phi}_2} \tag{4.8}$$

where f is the limit cycle frequency in radians per second. Rearranging this equation and converting f into cycles per second:



(a) Centered on Origin

$\dot{\phi}_1 > \dot{\phi}_2$

(b) Offset from Origin

Figure 4-1: Possible Limit Cycle Trajectories

$$f(Hz)=\frac{\dot{\phi}_1\dot{\phi}_2}{4\pi\phi_{db}(\dot{\phi}_1+\dot{\phi}_2)} \tag{4.9}$$

For the simple limit cycle in Figure 4-1, there are two thruster firings per cycle. Thus:

$$\frac{firings}{sec}=\frac{\dot{\phi}_1\dot{\phi}_2}{2\pi\phi_{db}(\dot{\phi}_1+\dot{\phi}_2)} \tag{4.10}$$

The number of thruster firings should be proportional to fuel consumption. If F_{SS}= the steady state fuel consumed per second,

$$\frac{\dot{\phi}_1\dot{\phi}_2}{2\pi\phi_{db}(\dot{\phi}_1+\dot{\phi}_2)}=cF_{ss} \tag{4.11}$$

where c is a constant. This can be rewritten as:

$$\frac{\dot{\phi}_1\dot{\phi}_2}{2\pi\phi_{db}(\dot{\phi}_1+\dot{\phi}_2)F_{ss}}=c \qquad (4.12)$$

In the normalized case, rates are evenly distributed on either side of the ϕ-axis. Thus, the normalized rate, $\dot{\phi}_N$, is equal to the average of $\dot{\phi}_1$ and $\dot{\phi}_2$. By an analysis similar to the above, it can be shown that:

$$c=\frac{\dot{\phi}_1+\dot{\phi}_2}{8\pi\phi_{db}F_N} \qquad (4.13)$$

where F_N = the normalized fuel consumption. Setting equations (4.12) and (4.13) equal and solving for F_N:

$$F_N=\frac{(\dot{\phi}_1+\dot{\phi}_2)^2}{4\dot{\phi}_1\dot{\phi}_2}F_{ss} \qquad (4.14)$$

The normalized rate, $\dot{\phi}_N$, is given by:

$$\dot{\phi}_N=\frac{\dot{\phi}_1+\dot{\phi}_2}{2} \qquad (4.15)$$

With real trajectories in phase planes more complicated than a simple deadband, equations (4.14) and (4.15) do not really apply. Disturbances, occasional double firings on one side of the deadband, and other factors affect the trajectories. Thus, these equations can only give approximations of normalized fuel and rates. These approximations will be used, but it must be kept in mind that they are very inexact.

Equation (4.15) can be implemented for each axis, but equation (4.14) can not. The simulator is designed to output only the total fuel consumed, and not the fuel consumed by the thruster firings about each individual axis. It was decided not to modify the simulator to output the individual fuel consumptions. Since equation (4.14) is only an approximation anyway, it is much easier (and probably about as accurate) to calculate the scaling factor:

$$\frac{(\dot{\phi}_1+\dot{\phi}_2)^2}{4\dot{\phi}_1\dot{\phi}_2}$$

for each axis, average the three values, and use this as the multiplier of F_{ss} in equation (4.14). This multiplier was usually found to lie between 1.0 and 1.1. Thus, normalized fuel consumption was seldom more than 10% greater than observed fuel consumption.

4.1.2.1 Fuel Consumption

The steady state fuel consumption score is given by the formula:

$$score=\frac{minimal\,fuel}{normalized\,fuel}\text{x}\,10 \tag{4.16}$$

The "minimal fuel consumption" was arbitrarily chosen to be the rate of fuel use that would result in 1 kg of fuel being used for every 4 days spent in steady state, or about $2.9\text{x}10^{-6}$ kg/sec.

4.1.2.2 Limit Cycle Rate Amplitude

The limit cycle rate score is given by the equation:

$$score=(1-\frac{\dot{\phi}_N{}^2}{\dot{\phi}_{lc-a}{}^2})\text{x}\,10 \tag{4.17}$$

where $\dot{\phi}_N$ is the normalized limit cycle rate and $\dot{\phi}_{lc-a}$ is the allowable limit cycle rate. The pointing requirements for the Magellan suggest that $\dot{\phi}_{lc-a}$ should be set to 0.1 deg/sec.

4.1.2.3 Limit Cycle Attitude Amplitude

The score is:

$$score=(1-\frac{\phi_N{}^2}{\phi_{lc-a}{}^2})\text{x}\,10 \tag{4.18}$$

where ϕ_N and ϕ_{lc-a} are defined similarly to the limit cycle rates. ϕ_{lc-a} shall be set to 0.1 degrees.

4.1.3 Ease of Assigning Parameter Values

This score is completely subjective, but will include such factors as the number of variable parameters and how well initial analyses and intuitions about each system help to choose parameter values that work.

While the scoring system outlined above will be useful in comparing attitude control designs, designing to optimize each of these criteria would be a formidable task. Instead, some more modest design goals will be set. Since many control schemes seek to minimize a weighted time-fuel index, this seems like a good approach. The most commonly used index is:

$$J=\int_0^{t_{final}}(\lambda+|u|)\,dt \tag{4.19}$$

For lack of a better choice, the weighting factor λ will be set to $\lambda = 1$. In transient response, the equivalent index:

$$J=\sum_{i=1}^{3} t_{si}+\frac{m_f I_{sp} r}{\tau} \tag{4.20}$$

will be minimized, where t_{si} = the settling time in axis i, m_f = the mass of fuel consumed, r = the thruster moment arm (an average value will be used) and $m_f I_{sp} r/\tau$ = the amount of time during which the thrusters fire (which is obviously proportional to fuel consumption).

In steady state, another design goal must be set. This goal will be for the fuel consumption and the limit cycle attitude scores to be as close together as possible. This is completely arbitrary, but the process of trying to reach this goal should clarify the design tradeoffs inherent in each control system, which is an important purpose of this study.

4.2 VOI

Over its several hundred day mission, the Magellan attitude control system's toughest 80 seconds occurs during VOI. With disturbances as high as 80% of the control torque, the design of a system that results in acceptable performance is very challenging. The attitude control requirements for VOI are much looser than those for normal attitude control; attitude errors about each axis must never exceed 0.806 degrees, rate errors must be no greater than 1.14 deg/sec about each axis, and the total fuel consumption must be less than 24 kg[7]. Despite the looseness of these requirements, Magellan is barely able to meet them.

Similar problems will be encountered as attitude control system designs are considered in this thesis. Most of them will not be able to meet these requirements. Coming up with rating scales analogous to those used in the normal attitude control case seems rather pointless when it will be a struggle to obtain a system that performs acceptably at all.

The design goal for VOI is thus much simpler than those used for the normal attitude control case. A system will be sought which, when started from rest, can meet all the requirements mentioned above. For those designs that can not meet the requirements, a careful look at which requirements are not met and *why* they are not met will give insights into ways the design might be improved, and/or reasons why other designs might be preferable.

---

[7] These requirements are slightly modified from those that have actually been imposed. Still, they are very close to what is expected of Magellan.

Chapter 5

Phase Plane Designs

In this chapter, the three phase plane designs will be considered -- the MGN phase plane, the RLC, and the RLC with short pulse regions.

Before investigating the phase planes themselves, another aspect of the control system must be considered. As described in Chapter 2, the rate error is determined by simply dividing the incremental gyro angle by the sampling interval. This can produce a very noisy signal. One important aspect of any control system is its method of smoothing this noise. Thus, the design of a low pass filter must be the first topic of discussion.

5.1 Filter Design

The first order low pass filter used to filter the rate error signal has the form:

$$\frac{\dot{\phi}_{out}}{\dot{\phi}_{in}} = \frac{\alpha}{s+\alpha} \tag{5.1}$$

This filter has bandwidth α and can be represented by the differential equation:

$$\ddot{\phi}_{out} + \alpha\dot{\phi}_{out} = \alpha\dot{\phi}_{in} \tag{5.2}$$

Since this filter has to be implemented digitally, differential equation (5.2) must be converted into a difference equation. Any differential equation of the form:

$$\dot{x} = Ax + Bu \tag{5.3}$$

can be converted into a difference equation:

$$x_{n+1} = A^{*}x_{n} + B^{*}u_{n} \tag{5.4}$$

which gives approximately equivalent values of x at every sampling instant. This is accomplished with the transformations:

$$A^* = e^{AT} \qquad (5.5)$$

$$B^* = A^{-1}(A^*-I)B \qquad (5.6)$$

where T = the sampling interval. Applying these equations to equation (5.2), one finds the equation of the first order low pass digital filter to be:

$$(\dot{\phi}_{out})_{n+1} = a(\dot{\phi}_{out})_n + (1-a)(\dot{\phi}_{in})_n \qquad (5.7)$$

where

$$a = e^{-\alpha T}$$

To complete the design of the filter, the bandwidth, α, must be chosen.

In the normal attitude control case, a bandwidth of 6.5 rad/sec (slightly more than 1 Hz) was chosen. Since normal attitude control tends to involve low control frequencies, this value was found to work very well at smoothing the rate error signal with a sampling interval, T, of 133 msec.

In VOI, T = 33 msec. Bandwidths of 6.5, 31.4, 50.3, and 81.7 rad/sec were tried and were all found to degrade system performance. In VOI, where everything tends to happen rapidly and violently, the time delay introduced by a filter seems to hurt the system more than the smoothed rate signal helps it. Thus, no filter at all will be used in this case.

To facilitate comparison of different control system designs, the filter values obtained above will be used to filter the rate error of every control system that is to be considered. The first of these control systems is the Magellan phase plane.

5.2 The Magellan Phase Plane

5.2.1 Initial Analysis

The Magellan phase plane is depicted in Figure 3-5. The values of the parameters k, A1, A2, A3, A4, R1, R2, and R3 must be chosen for both the normal attitude control case and VOI. The presentation of the MGN phase plane in Chapter 3 indicates that:

$$k=\sqrt{\frac{2\tau}{I}} \tag{5.8}$$

where τ is the torque delivered about an axis and I is the moment of inertia about that axis. However, this presentation gives no indication of how to choose the various rate and attitude ledges. There is a simple method of doing this. If the spacecraft is in region 1 (pulse time = 11 msec) and more than 3 thruster pulses will be needed to push it into the deadband, it might as well be in region 2 (pulse time = 33 msec). Similarly, if more than 4 pulses are needed to get from a point in region 2 to region 1, the thrusters might as well be commanded to fire as in region 3 (pulse time = 133 msec). This criterion will be used to choose rate and attitude ledges.

If spacecraft motion about one axis is assumed to obey the equation:

$$\ddot{\phi}=-\frac{\tau}{I} \tag{5.9}$$

it can be shown that:

$$\dot{\phi}=\dot{\phi}_0-\frac{\tau}{I}t \tag{5.10}$$

where $\dot{\phi}_0$ is the initial rate. If a thruster firing of duration t_p is to move the spacecraft from rate ledge R_i to ledge R_{i-1}, equation (5.10) becomes:

$$R_i=R_{i-1}+\frac{\tau}{I}t_p \tag{5.11}$$

Equation (5.10) can be integrated to obtain:

$$\phi=\phi_0+\dot{\phi}_0 t-\frac{\tau}{2I}t^2 \tag{5.12}$$

where ϕ_0 is the initial attitude. For simplicity, assume $\dot{\phi}_0=0$. Then, a thruster firing of duration t_p should move the spacecraft state from ledge A_i to ledge A_{i-1}. Thus, in equation (5.12), $\phi_0=A_i$ and $\phi=A_{i-1}$. Substituting, one finds that:

$$A_i=A_{i-1}+\frac{\tau}{2I}t_p^2 \tag{5.13}$$

Keeping in mind that t_p= 33 msec for region 1 and t_p= 133 msec for region 2, equations (5.11) and (5.13) can be used to choose ledges A2, A3, and R3 if A1 and R2 are given. Values of A1 and R2 that have already been empirically shown to perform well on Magellan will be used. Parameter values are shown in Table 5-I.

Table 5-I: Initial Values for MGN Phase Plane Parameters

parameter	units	normal attitude control		VOI	
		x & y axis	z axis	x & y axis	z axis
K	$(rad)^5/sec$	0.0479	0.0565	1.08	0.332
R1	rad/sec	-8.73E-04	-8.73E-04	-0.0852	-0.0257
R2	rad/sec	5.6E-05	7.9E-05	0.02	0.012
R3	rad/sec	1.32E-04	1.84E-04	0.0296	0.0129
A1	rad	4.94E-04	7.0E-04	0.006	0.006
A2	rad	4.9462E-04	7.009E-04	0.0060706	0.0060067
A3	rad	4.9712E-04	7.044E-04	0.0062295	0.0060217
A4	rad	0.1745	0.1745	0.1745	0.1745

No simple technique was found for choosing values of the parameters R1 and A4. A4 will be arbitrarily chosen to equal 10 degrees. Since the purpose of ledge R1 is to allow region 3 firings to speed the spacecraft trajectory toward the phase plane origin if it is moving there too slowly, R1 values must be chosen to represent the minimum rates that seem acceptable.

The equations derived above depend on the spacecraft inertia about a particular axis. In VOI, however, inertias vary dramatically. For the purpose of choosing parameter values, average VOI inertias of $I_{xc} = I_{yc} = 3158$ kgm^2 and $I_{zc} = 1715$ kgm^2 will be used. Initial Magellan phase plane parameters are given in Table 5-I.

5.2.2 Normal Attitude Control

The results of a simulation run with the parameter values determined above are shown in Figure 5-1. Attitude and rate errors converge to the origin. The calculated parameter values result in a stable, reasonably well-behaved system. However, there is much room for improvement. Parameter values were adjusted in an attempt to optimize performance, and a summary of the various simulation runs that were attempted is given in Table 5-II. The phase plane plots in Figure 5-1 indicate one way in which response might be improved. Notice that the trajectory about each axis contains a final "approach parabola" that approaches the origin. However, these parabolas overshoot the origin and then must slowly come back to it. If these parabolas could be forced to intersect the origin, settling times could be greatly reduced, thus improving transient performance.

The key to making the approach parabolas hit the origin is found in the choice of a value for the parameter k. When the thrusters continuously fire in the +3 region, the trajectory about an axis traces a parabola in the phase plane. If k is chosen such that the switching parabola is parallel to this "natural parabola," as depicted in Figure 5-2a, the approach parabola remains parallel to and slightly apart from the switching parabola, thus coming very close to the origin. If, however, k is too small, the trajectory shown in Figure 5-2b results. The approach parabola intersects the switching parabola, causing repeated crossings of the switching boundary. Although this should not have much of an effect on fuel consumption (no extra fuel is consumed in the "off" region), it could cause longer settling times. However, there is an alternative that can have an even worse effect on settling times. If k is chosen too large, the trajectory indicated in Figure 5-2c can result. The origin is overshot, and recovery speeds are slow. This is what is happening in the run shown in Figure 5-1.

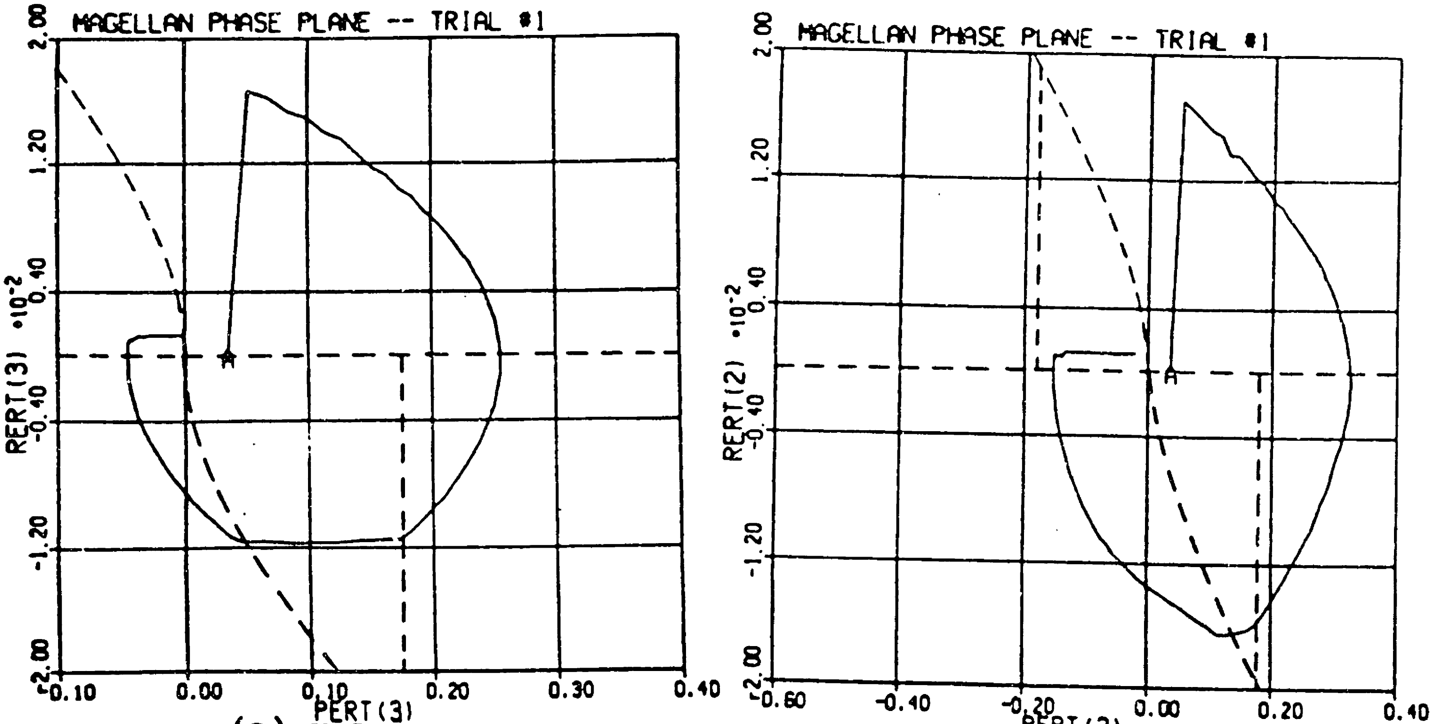
k was calculated by the formula:

$$k=\sqrt{\frac{2\tau}{I}} \tag{5.14}$$

To calculate τ, the nominal 0.2 lbsf delivered by the thrusters was used. However, the actual τ is less than this calculated value, since τ is a function of tank pressure, and pressure is not very high when the simulated propellant mass is only 60 kg. It is not surprising, then, that the initially calculated k value is too high. k must be reduced.

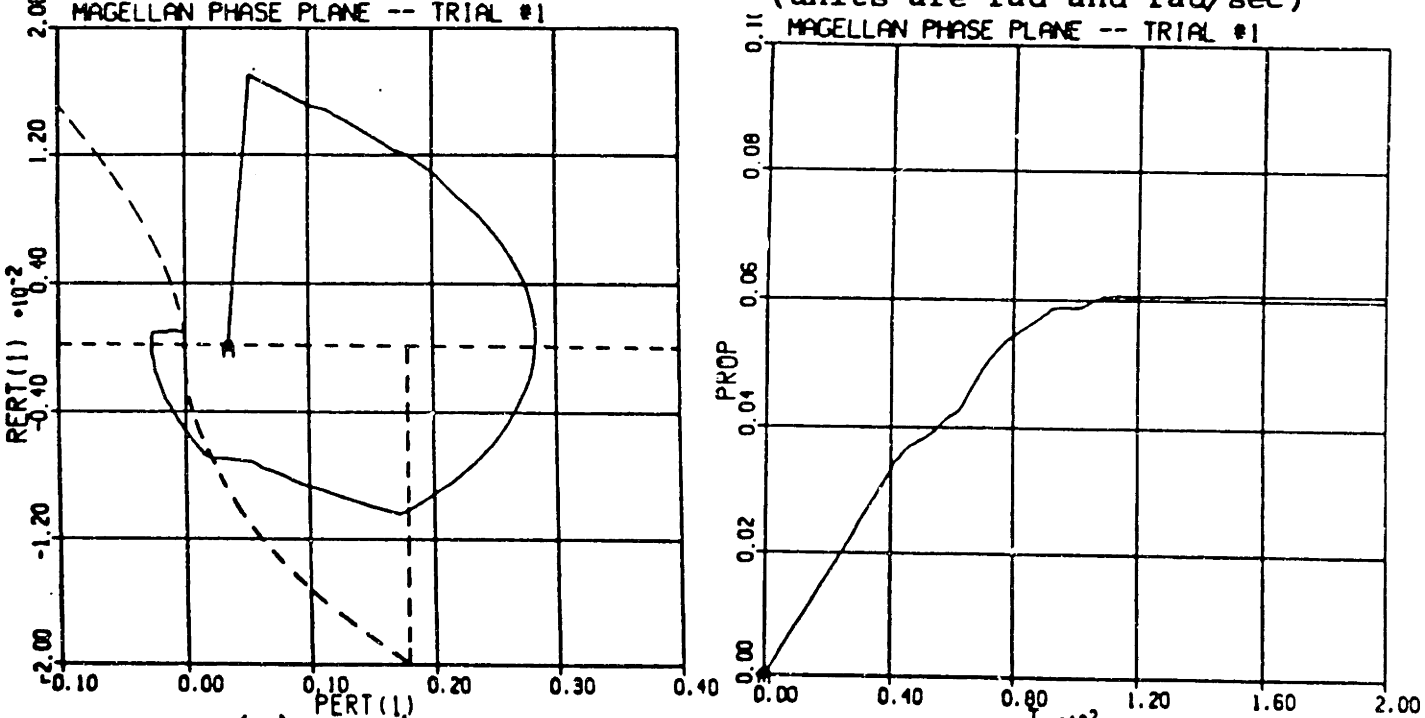
However, some caution is necessary. If k is reduced too far, it will cause the switching parabola to intersect the ledge R1. The maximum permissible value of R1 (*ie.* the value that just causes ledge R1 to touch the switching parabola) is:

$$R1=k\sqrt{A3} \tag{5.15}$$



(a) x-axis
(units are rad and rad/sec)

(b) y-axis
(units are rad and rad/sec)



(c) z-axis
(units are rad and rad/sec)

(d) Propellant Consumed
(kg) vs. time (sec)

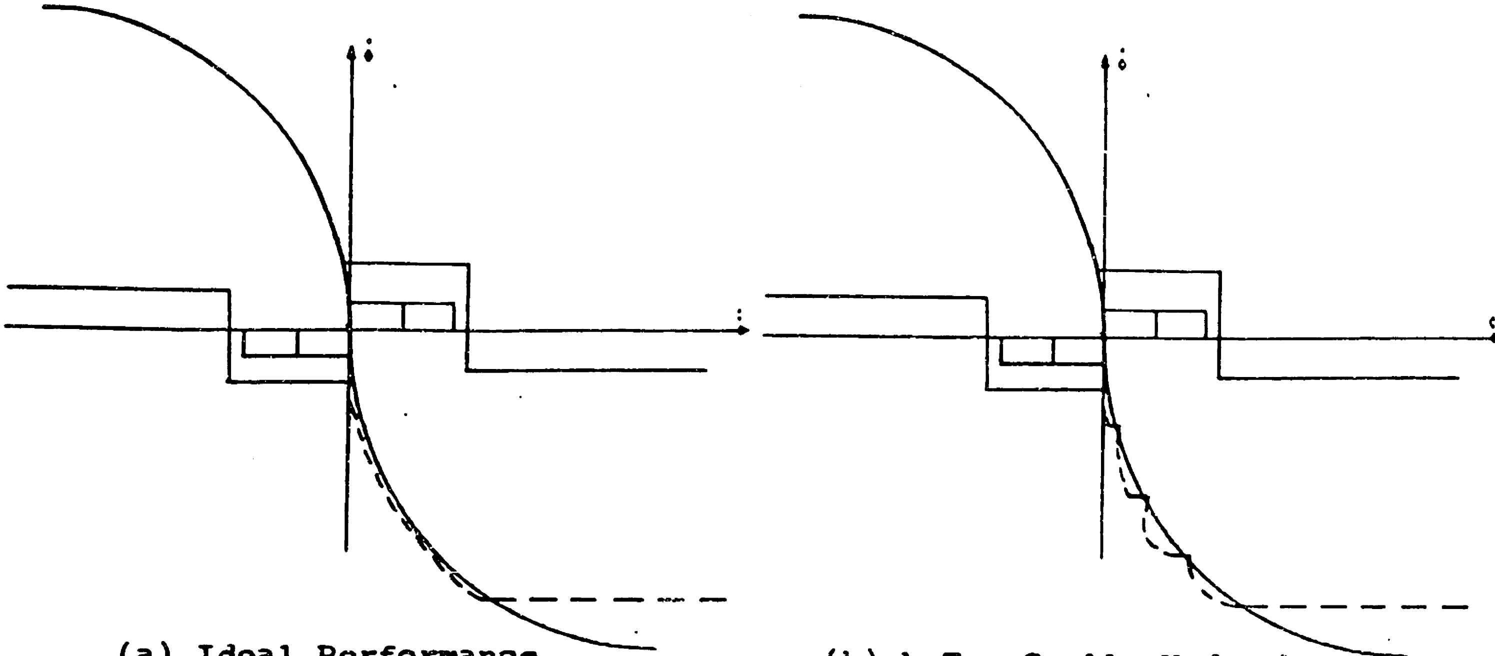
Figure 5-1: Initial MGN Phase Plane Trajectories

Table 5-II: Simulations for MGN Normal Attitude Control

un	comments	settling time (sec)			trans fuel	J	limit cycle attitude			limit cycle rate			ss fuel
		X	Y	Z			X	Y	Z	X	Y	Z	
1	analytically chosen values	---	---	---	---	---	---	---	---	---	---	---	---
2	K = 0.0375, 0.0447	87.0	150.0	87.0	3.16	455.9	9.40	9.49	9.57	9.92	9.69	10.0	0.97
3	K = 0.033, 0.043	70.0	110.0	83.0	3.24	390.0	9.34	9.20	8.40	9.98	9.98	9.91	0.33
4	K = 0.0317, 0.0419 R1 adjusted accordingly	70.0	97.5	77.5	3.26	372.1	9.44	9.64	9.70	10.0	9.98	10.0	10.0
5	K = 0.03, 0.04 R1 adjusted accordingly	70.0	85.0	70.0	3.32	349.7	9.91	9.45	8.36	9.98	10.0	9.91	6.16
6	K_z = 0.038 R1 adjusted accordingly	70.0	85.0	65.0	3.4	341.6	9.23	9.24	8.42	9.98	10.0	9.91	10.0
7	R1 = 0, 0	---	---	---	---	---	---	---	---	---	---	---	---
8	s.s. z axis parameters set to x & y values	70.0	85.0	65.0	3.4	341.7	9.99	9.35	10.0	10.0	10.0	10.0	10.0
9	A4 = 7.5°, 7.5°	70.0	85.0	65.0	3.17	----	----	----	----	----	----	----	----
10	A4 = 12°, 12°	80.0	85.0	73.0	3.62	351.7	----	----	----	----	----	----	----
11	A4 = 10.5°, 10.5°	73.0	85.0	65.0	3.44	342.6	----	----	----	----	----	----	----
12	return to run 8 values A1 = 4E-04, 4E-04	70.0	85.0	65.0	3.40	342.7	9.47	9.46	9.46	9.98	10.0	9.99	8.27
13	A1 = 4.5E-04, 4.5E-04	70.0	85.0	65.0	3.39	342.1	9.33	9.34	9.48	9.98	9.98	9.99	5.91

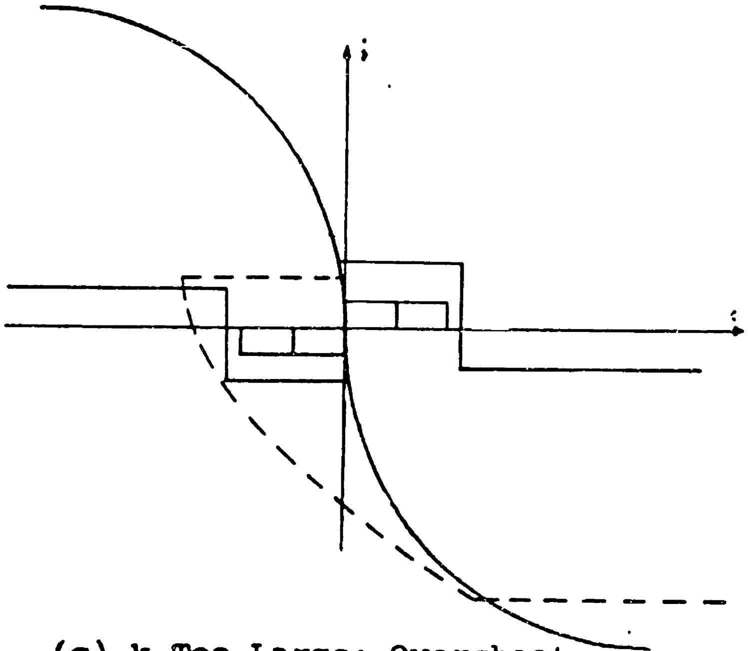
(see Figure 3-5). In runs that reduce k too far, R1 must be adjusted according to this relation.

The parameter k was adjusted until the x-axis was no longer observed to overshoot (Run 3). Phase plane plots from Run 3 are shown in Figure 5-3. Notice that, while the x-axis approach parabola hits the origin, the y-parabola does not. Yet k is a function of τ and I, and the x and y control axes have identical τ and I values. This discrepancy can only arise from one source. Inter-axis coupling causes different torques about the x and y axes, and these torques are affecting the approach parabolas. However, the x and y axis parameters should still not be chosen independently. A different set of initial conditions, after all, might cause the x-axis to overshoot and the y-axis to hit the origin. Yet continuing to reduce the value of k on both axes will simultaneously hurt x-axis response and help



(a) Ideal Performance

(b) k Too Small: Undershoot



(c) k Too Large: Overshoot

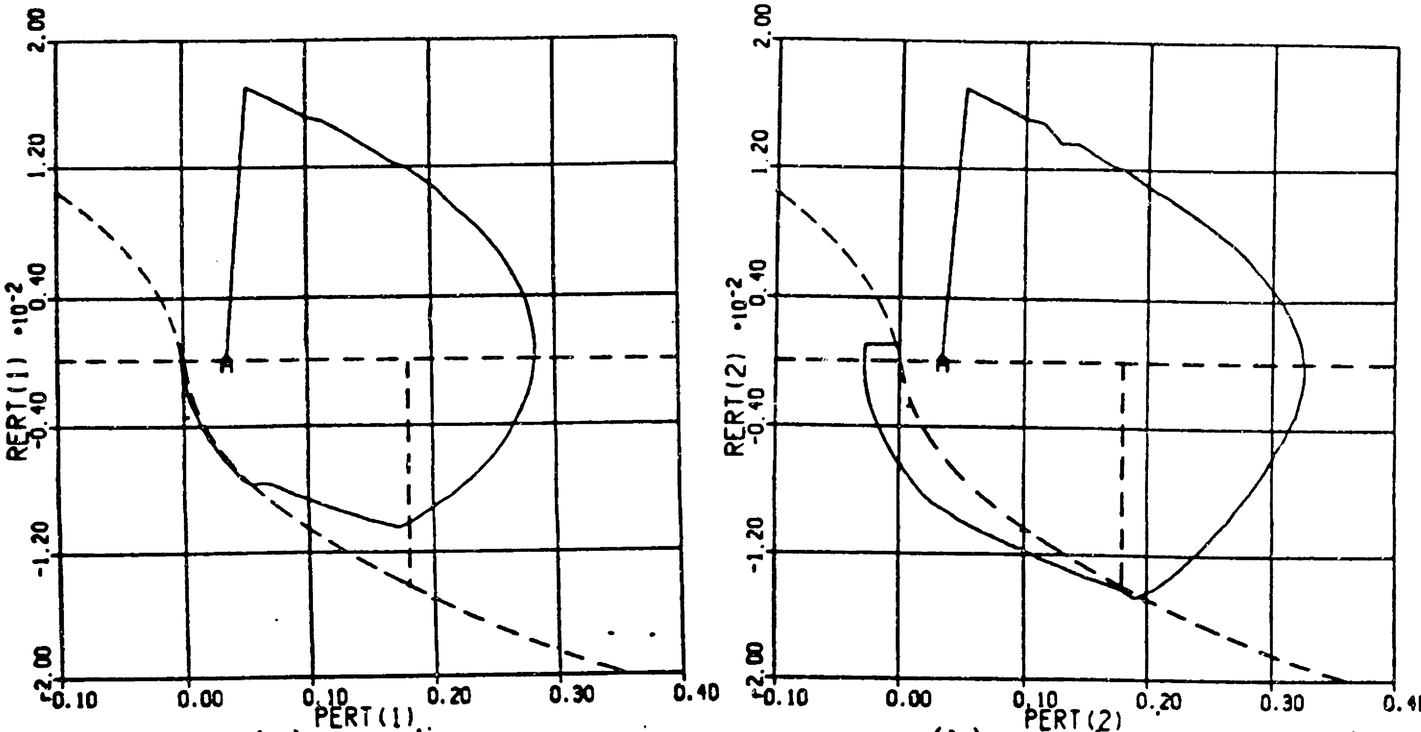
Figure 5-2: Undershoot and Overshoot in the MGN y-axis response. The point in this tradeoff that optimizes the index J (see equation (4.20)) must be located.

By Run 6, all overshoot was eliminated. Phase plane plots from this run are shown in Figure 5-4. Although the simultaneous adjustment of k_x and k_y improved the y-axis settling time by about 25 seconds, x-axis times were increased by less than one second. Clearly, undershoot is much less of a problem than overshoot.

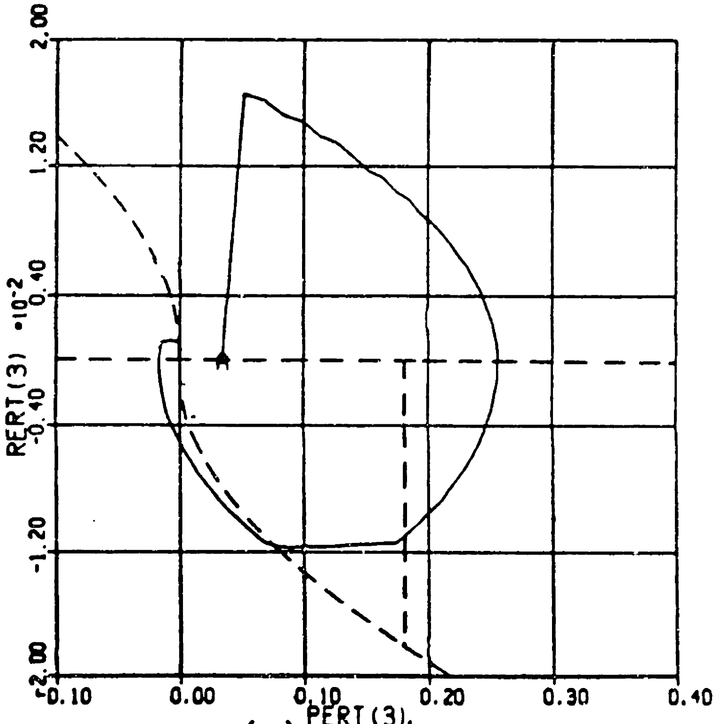
It is easy to decide on a value for the parameter R1. If R1 were set to zero, trajectories like the one sketched in Figure 5-5 would result. Such trajectories converge to the origin very slowly. Also, it may be observed that thus far in the parameter tuning process, a parameter adjustment resulting in large changes in settling times has usually resulted in only small changes in transient fuel consumption. To minimize the equally weighted time-fuel index J (equation (4.20)), then, it is extremely important to reduce settling times, especially because this can probably be done without a high fuel expense. R1 should thus be set as large (in absolute value) as possible without overlapping the switching parabola, thus hurrying the trajectory on its way toward the origin.

The only remaining parameter that can affect transient response is the attitude limit A4. In Runs 9, 10, and 11, A4 was adjusted from its initial 10 degrees. When A4 was less then 10 degrees, fuel consumption rose excessively without any accompanying improvement in settling times. If A4 was greater than 10 degrees, settling times rose. The maximum value of J occurred when A4 = 10 degrees.

As mentioned above, small changes in fuel consumption have accompanied large settling time changes throughout the refinement of transient response. Because of this, the minimum time-fuel behavior that seems to have been settled on (Runs 6 and 8) is also very nearly a minimum time response. This should not be too surprising -- the MGN phase plane is, after all, based on a minimum time solution to the problem (see Chapter 3).

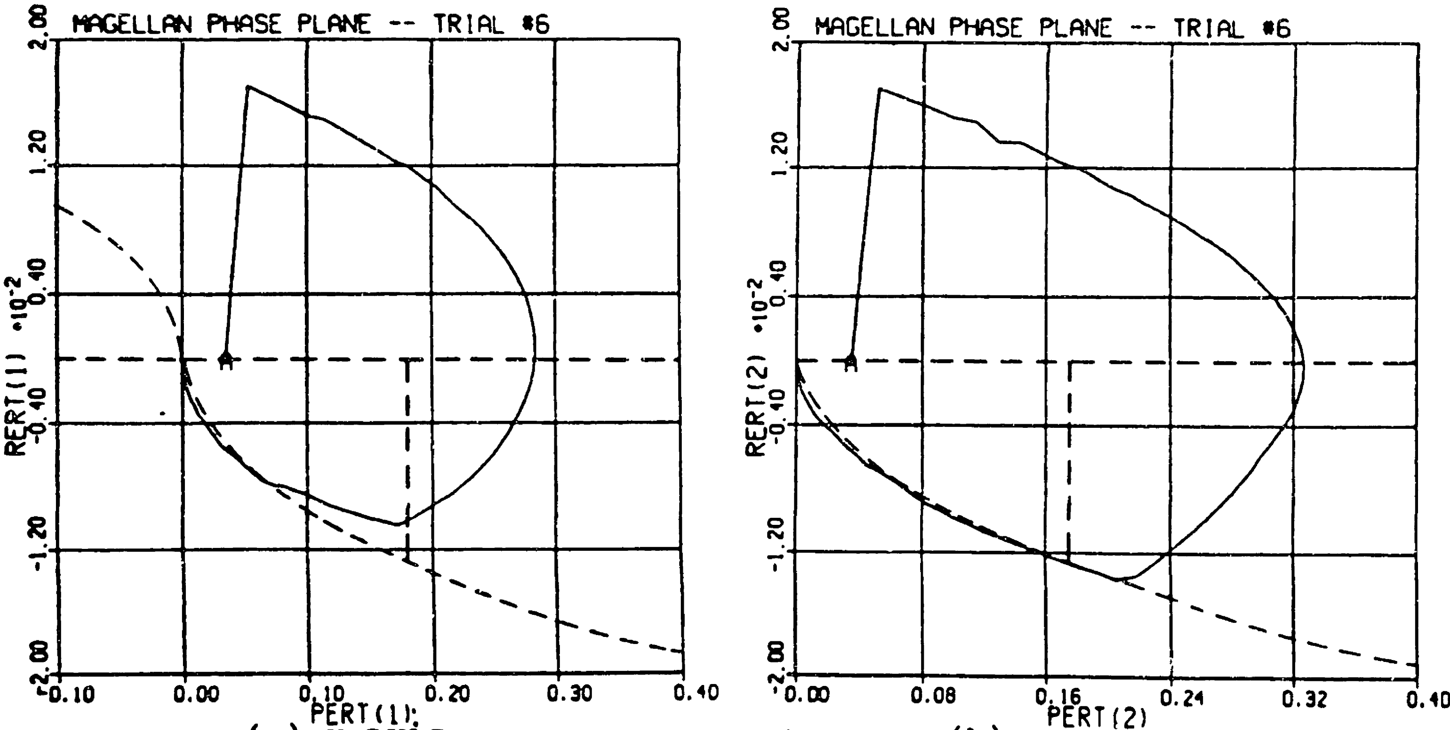


(a) x-axis (b) y-axis

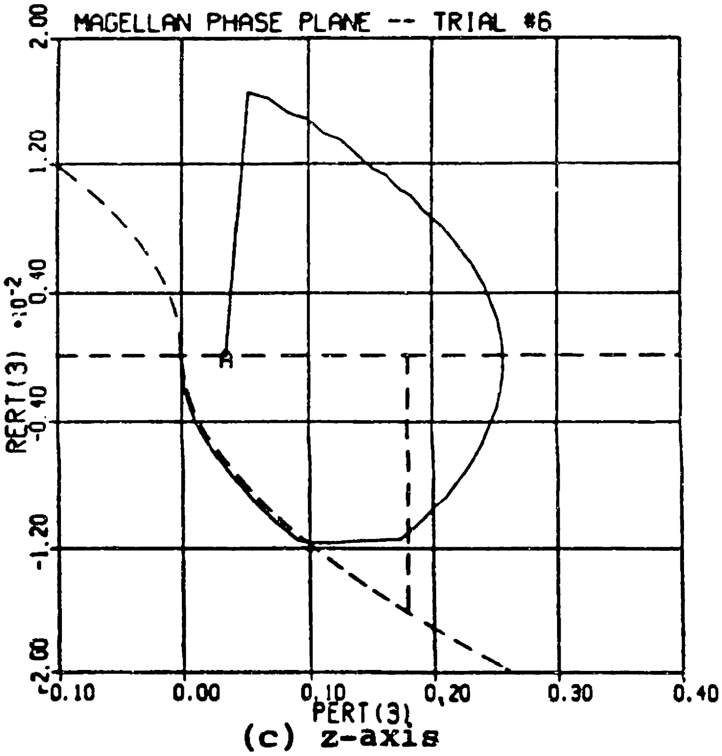


(c) z-axis

Figure 5-3: MGN Phase Plane Run 3 Results



(a) x-axis (b) y-axis



(c) z-axis

Figure 5.4: MGN Trajectories with No Overshoot

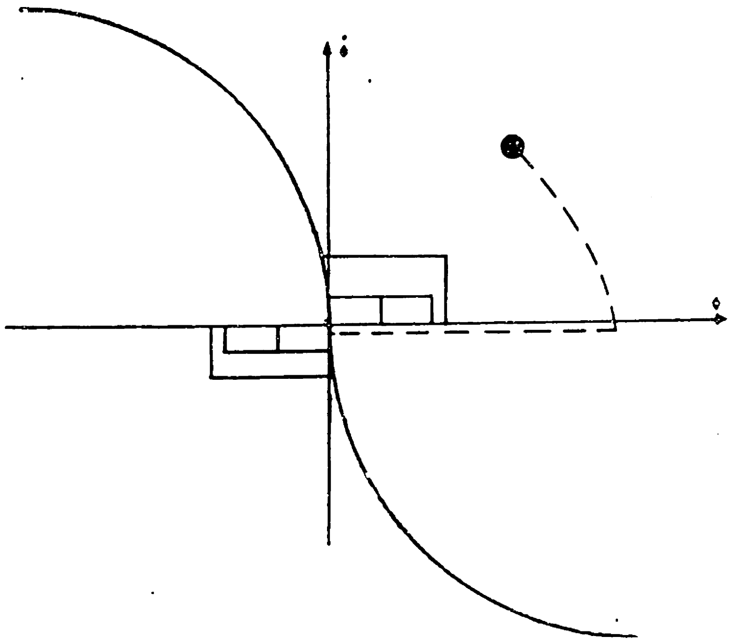
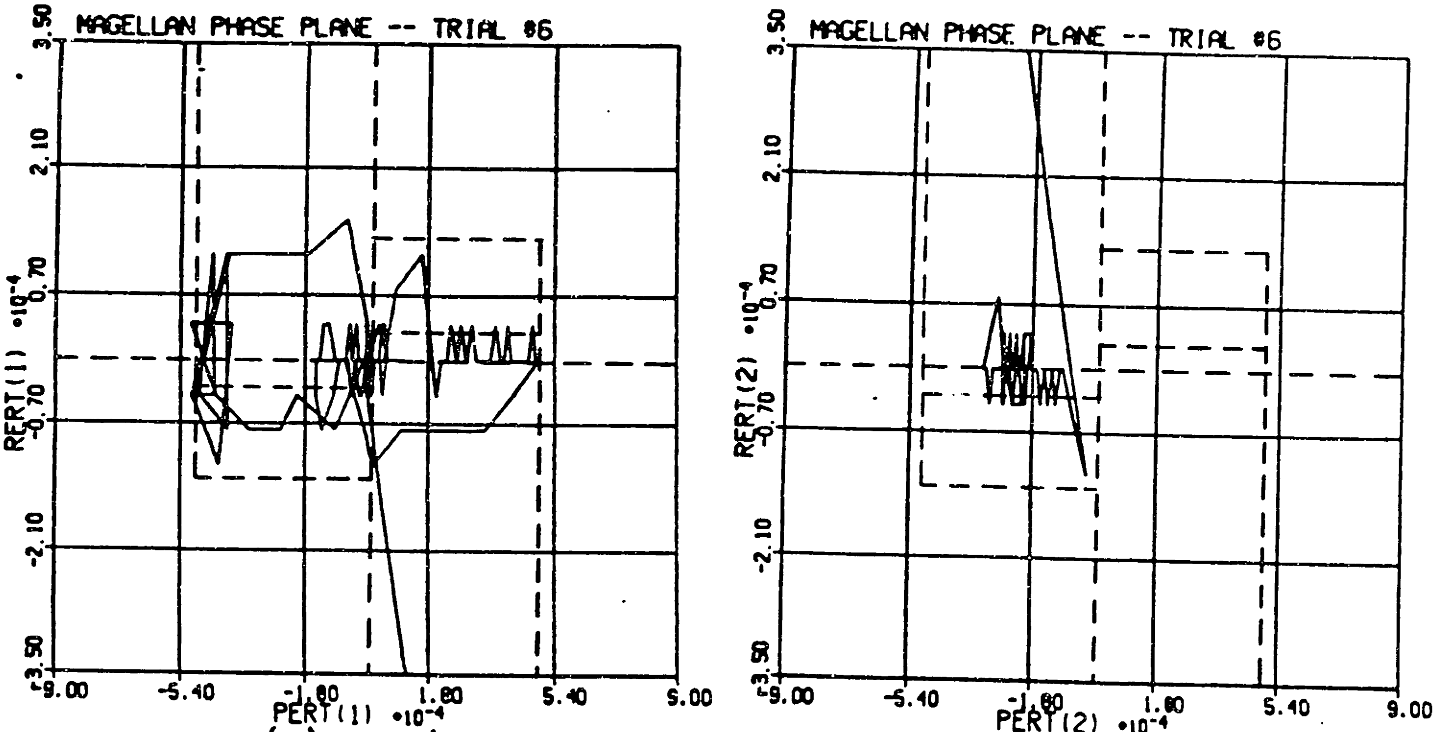


Figure 5-5: Trajectory with R1 Set Too Small

Next, steady state response was considered. Inter-axis coupling exerts a large influence in steady state. For example, in Run 6, the adjustment of k_z affected x and y-axis limit cycle attitudes, and not just z-axis attitudes. Steady state phase plane plots from Run 6 are shown in Figure 5-6. On this small scale, coupling torques and misfirings resulting from rate error noise can be very important. The approach to tuning steady state behavior must therefore be tr[illegible]e first the transient response completely. Then, preceded by a common transient tr[illegible] different values of steady state parameters can be tried.

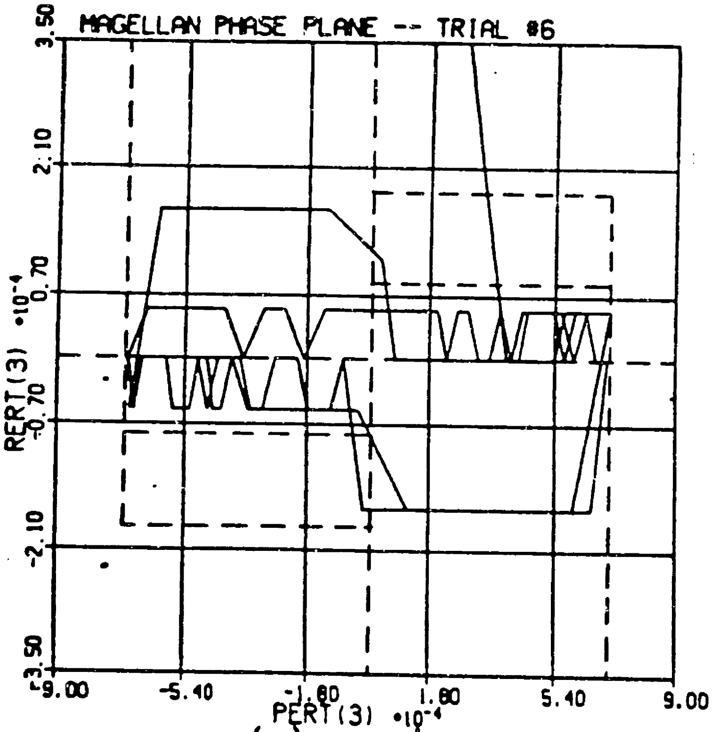
The steady state plots show spacecraft-sensed rate and attitude errors, and not actual system behavior. At this small scale, gyro quantization has a large effect. Quantization can cause a zero rate measurement when rates fall below a certain point. After enough attitude error has built up, however, the rate appears to leap briefly up to a higher value. Spacecraft-sensed error plots will continue to be referred to, since they give better insight into control system performance.

Because of differing inertia values, z-axis steady state parameters do not equal x and y axis values. The first thing that was (somewhat arbitrarily) tried was setting these



(a) x-axis

(b) y-axis



(c) z-axis

Figure 5-6: MGN Steady State Phase Plane Trajectories parameters to the x and y values. This meant tighter attitude and rate limits for the z-axis. As shown in the Table 5-II results for Run 8, this improved performance for all three axes. Limit cycle attitudes and rates were substantially reduced.

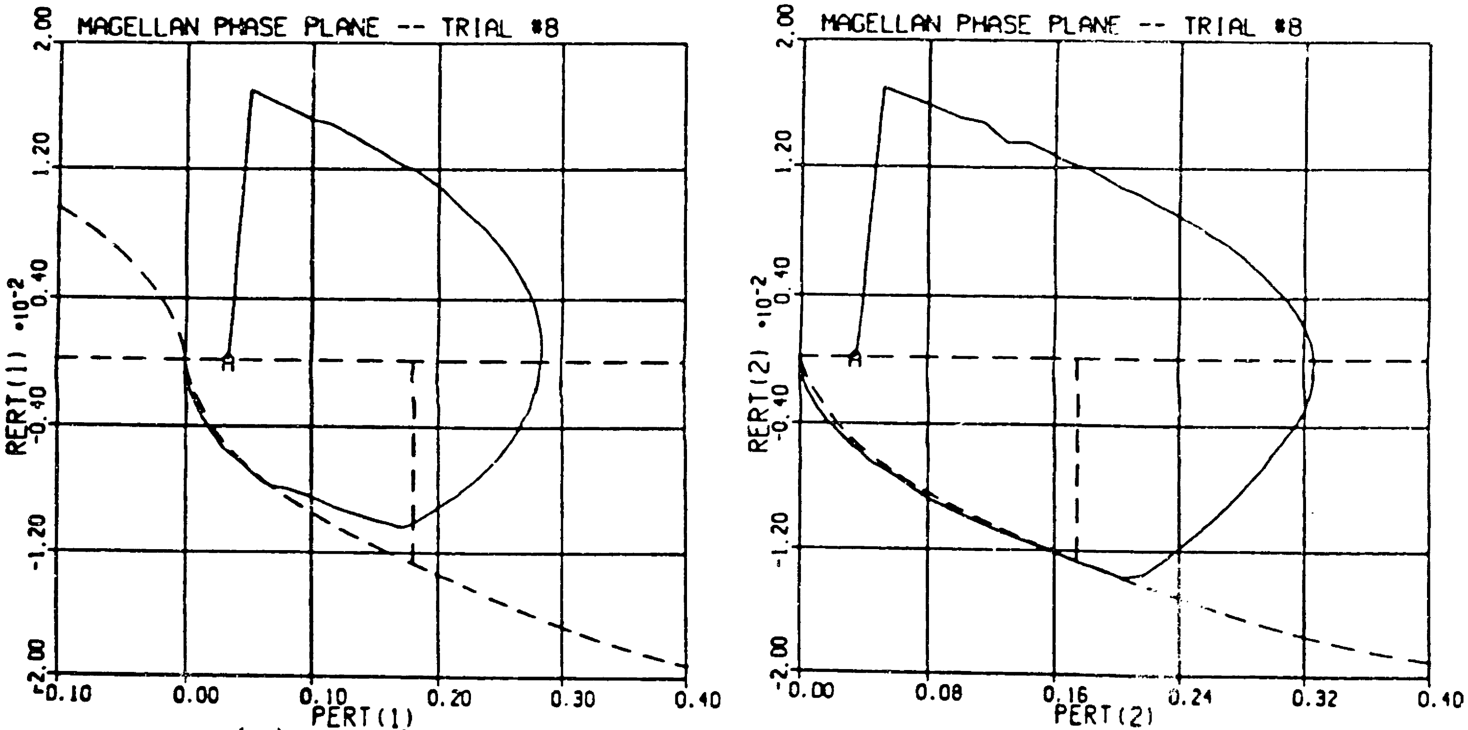
A word is in order here about limit cycle rates. They consistently remain low. It seems easy to reduce rates to a small fraction of the required 0.1 deg/sec. A 0.2 lbf thruster producing an 11 msec pulse can cause some very gentle nudges to the rates (on the order of 10^{-6} rad/sec). Thus, very fine rate control is possible.

In Runs 12 and 13, the innermost attitude deadband, A1, was adjusted. Its original value was found to be best. Simulation runs showed that, in steady state, thruster firings were predominantly in region 1 or region 2, and seldom in region 3. Since this implies that trajectories only occasionally exceeded ledges A2 or R2, these values were not adjusted.

Thus, Run 8 appears to be optimal, with the value of the performance index J (see equation (4.20)) reduced to J = 342. Limit cycle attitudes and rates and steady state fuel consumption are also very good. Plots of the results of the Run 8 simulation are shown in Figure 5-7, and results are tabulated in Table 5-III. A summary of the optimized parameters is given in Table 5-IV.

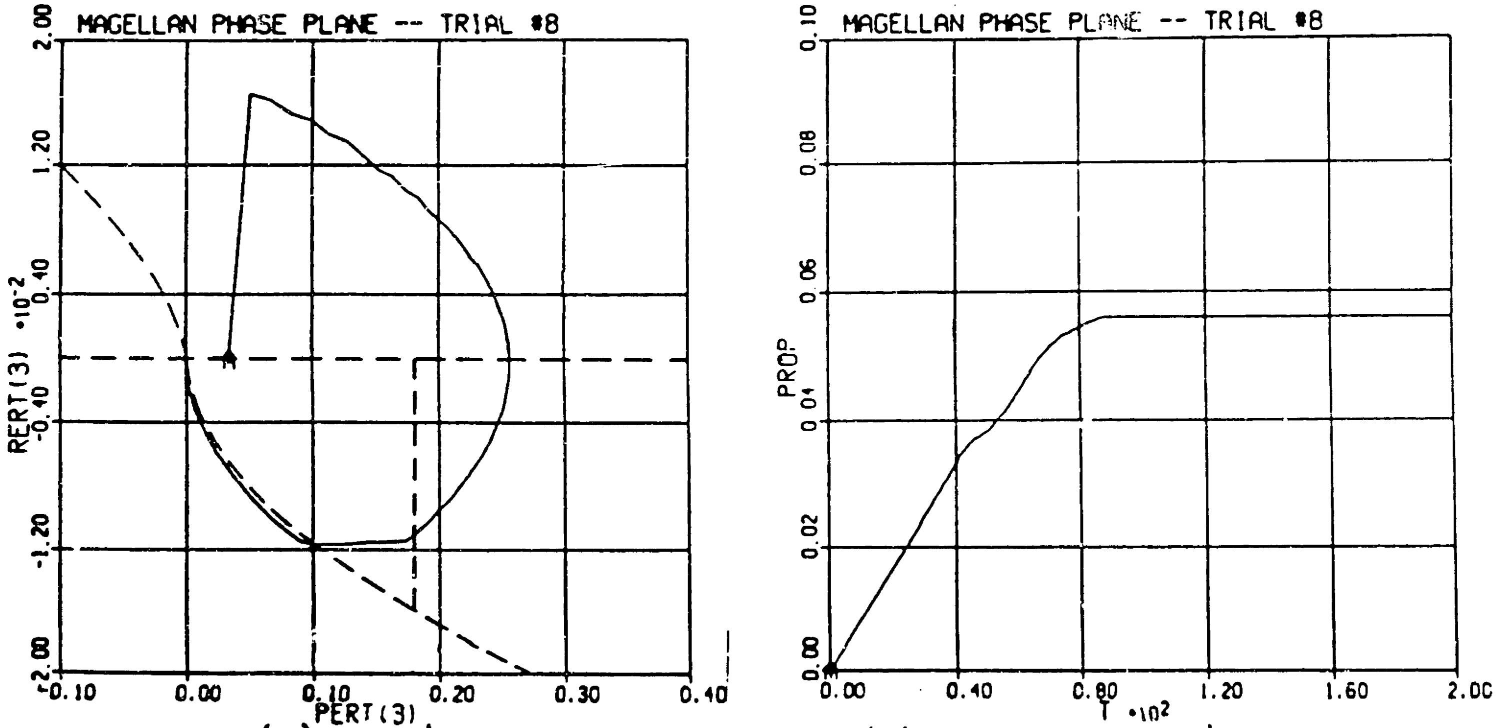
Run 8's rate excursion score is a 10. In none of the runs was a rate excursion in excess of the initial conditions observed. Also, the attitude excursion scores were the same for every run -- x:3.41, y:1.25, and z:4.68. It seems that the thrusters firing full on can damp out the initial rate errors no faster than the time required to reach these attitude excursion scores.

This system initially seems very complex -- there are 8 independent parameters. However, the initial parameter choices performed reasonably well, and the parameter tuning process was quite intuitive. By simply considering the geometry of the situation, the parameters k and R1 were easily adjusted. Thus, the ease of assigning parameter values score for this system (see section 4.1.3), in the normal attitude control case, is a 7.



(a) x-axis

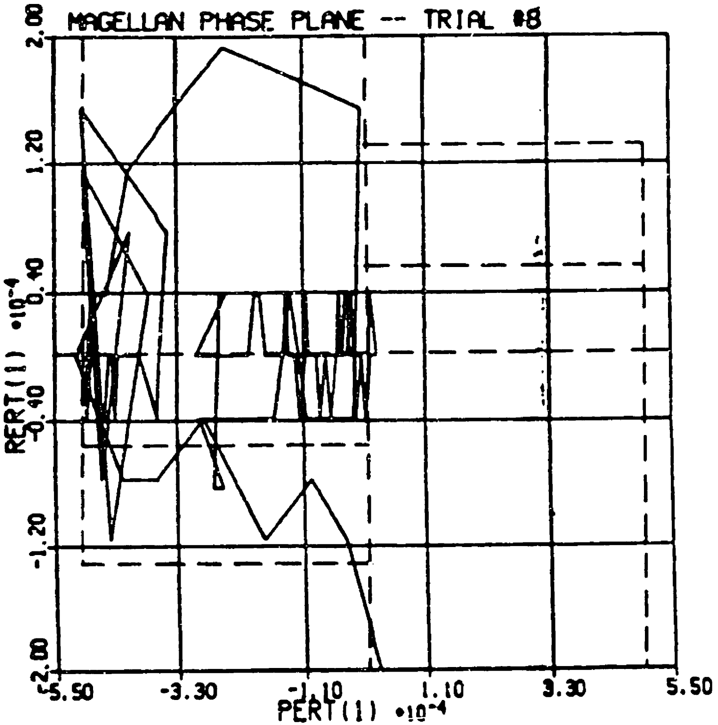
(b) y-axis



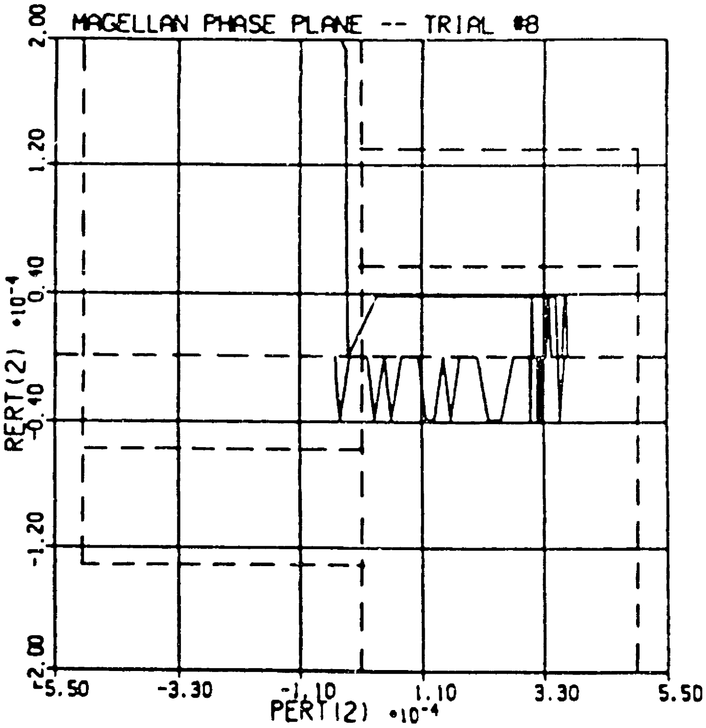
(c) z-axis

(c) Propellant Consumed

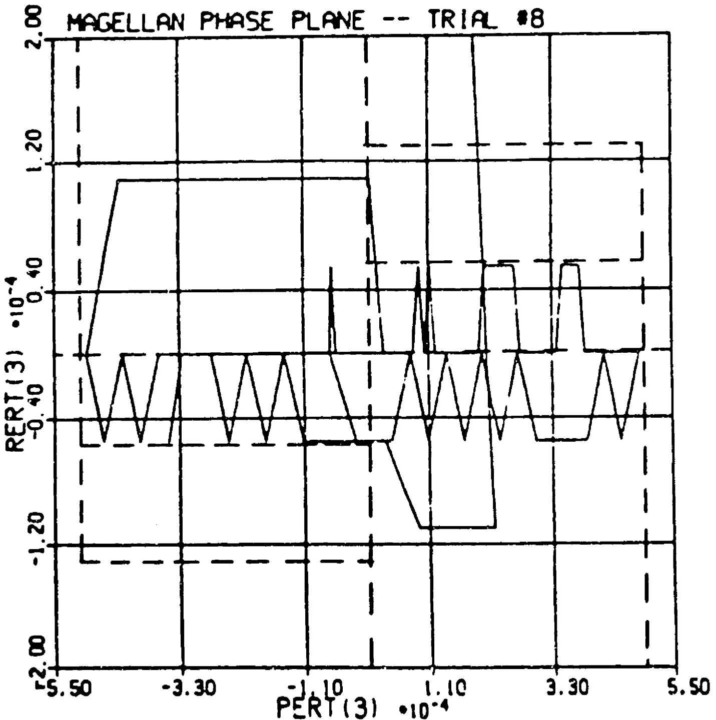
Figure 5-7: Optimal MGN Phase Plane Performance



(d) x-axis: Steady State



(e) y-axis: Steady State



(f) z-axis: Steady State

Table 5-III: MGN Normal Attitude Control Scores

Transient Scores

	X	Y	Z	AVG
settling time	8.57	7.06	9.23	8.29
fuel consumption				3.40
rate excursion	10.0	10.0	10.0	10.0
attitude excursion	3.41	1.25	4.68	3.11

J = 341.7

Steady State Scores

	X	Y	Z	AVG
fuel consumption				10.0
limit cycle rate	10.0	10.0	10.0	10.0
limit cycle attitude	9.99	9.35	10.0	9.78

Overall

	X	Y	Z	AVG
ease of assigning parameter values				7

In sum, the MGN phase plane performs very well. The most important parameter is k -- when this is properly set, J values as low as 342 and excellent steady state behavior can be observed. In transient response, minimizing an equally weighted time-fuel index closely approximates minimizing a simple time index, since large changes in settling times seem to accompany small changes in fuel use. However, both the settling times and the fuel

Table 5-IV: Optimal MGN Phase Plane Parameters

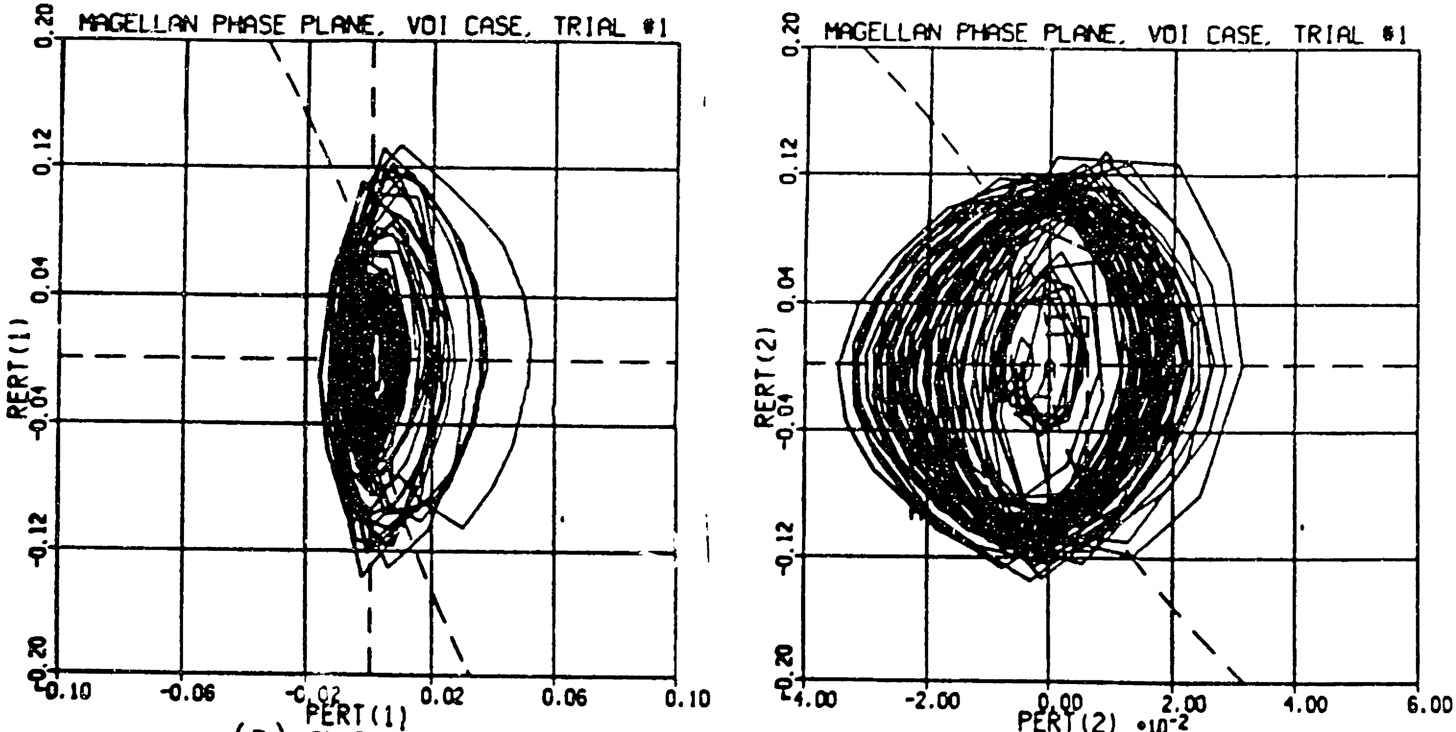
parameter	units	value	
		X & Y	Z
K	$(\mathrm{rad})^{1/2}$/sec	0.030	0.038
A1	rad	4.94E-04	4.94E-04
A2	rad	4.9462E-04	4.9462E-04
A3	rad	4.9712E-04	4.9712E-04
A4	rad	0.1745	0.1745
R1	rad/sec	-6.6889E-04	-8.4726E-04
R2	rad/sec	5.6E-05	5.6E-05
R3	rad/sec	1.3168E-04	1.3168E-04

consumption will compare very favorably with other systems. In steady state, the complexity of all the rate and attitude ledges allowed low fuel consumption *and* excellent steady state attitudes. This system may be complex, but its complexity allows easy tuning and excellent performance.

5.2.3 VOI

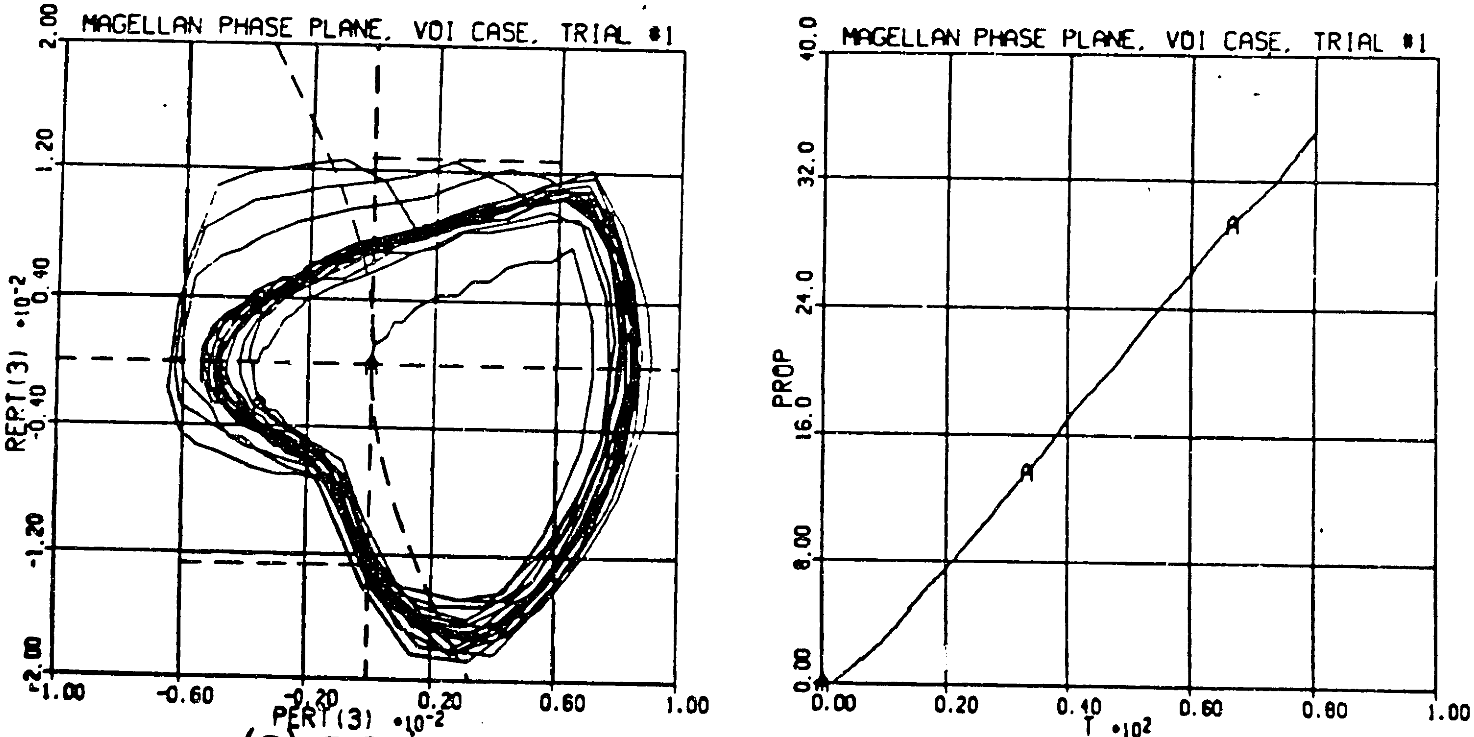
The results of a simulation of VOI using the parameter values calculated above is shown in Figure 5-8. System performance with these values is completely unacceptable. Well-behaved "approach parabolas" do not exist because of the disturbance torques. Rates, attitudes, and propellant usage are all very high. A summary of the simulation runs that were intended to improve this performance is given in Table 5-V.

Why does the control system function so poorly? The reason for this poor performance is the formation of a two-sided deadband. The phase plane plots of Figure 5-8 demonstrate a two-sided deadband -- a thruster firing in a + region exerts enough torque to push the system into a - region, where a - firing then pushes the system back into a + region. Figure 5-9 shows a simplified version of this type of thruster firing. If deadbands



(a) x-axis

(b) y-axis



(c) z-axis

(d) Propellant Consumed

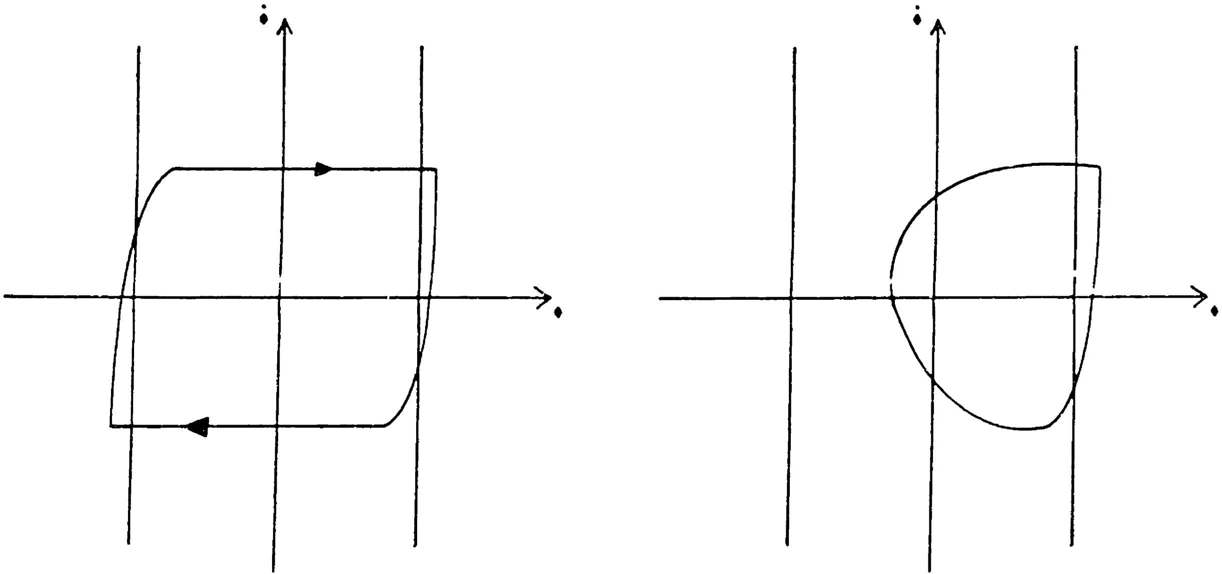
Figure 5-8: MGN VOI Performance with Initial Parameters

Table 5-V: Simulaiions of MGN Phase Plane in VOI

run	comments	maximum attitude (deg)			maximum rate (deg/sec)			fuel (kg)
		X	Y	Z	X	Y	Z	
1	analytically determined values	2.99	1.97	0.52	7.83	7.73	1.08	35.1
2	R3=0.04,0.14/A2= 0.0061412,0.0060134/ A3=0.026459,0.006043	2.12	1.78	0.51	8.94	8.38	1.08	34.5
3	R3=0.08,0.018 $A2_z$=0.007 / $A3_z$=.008	2.30	1.97	0.443	8.75	8.38	0.400	38.4
4	run #1 values k=1.2, 0.4	2.53	2.38	0.517	8.56	8.31	1.13	37.4
5	run #1 values R2=0.02,0.02/R3=0.12 0.03	2.85	2.39	0.775	8.50	8.18	1.58	37.3
6	R3=0.12,0.018/A2=.01 .007/A3=0.03,0.008	1.78	0.574	0.443	2.78	1.70	0.400	9.86
7	$A2_{x,y}$=0.0065	1.76	0.621	0.445	2.39	1.89	0.400	9.85
8	run #6 values $A2_{x,y}$=0.012	0.780	0.531	0.443	2.58	1.71	0.400	9.90
9	R1=0,0	0.762	0.521	0.443	1.80	1.47	0.400	9.75
10	$R2_{x,y}$=0.0175	0.743	0.511	0.444	1.66	1.38	0.400	9.77
11	$R2_{x,y}$=0.015	0.739	0.512	0.444	1.49	1.36	0.400	9.76
12	$R2_{x,y}$=0.0125	0.740	0.494	0.444	1.56	1.26	0.400	9.86
13	run #11 values $k_{x,y}$=0.183	0.839	0.700	0.444	1.84	2.49	0.400	13.7
14	run #11 values $R2_{x,y}$=0.009	0.733	0.476	0.443	1.54	1.18	0.400	9.96
15	$R2_{x,y}$=0.0075	0.748	0.456	0.444	1.71	1.64	0.425	10.8
16	run #14 values A1=0.005,0.005	0.746	0.426	0.380	1.71	1.17	0.350	10.0
17	$A1_{x,y}$=0.007	0.750	0.526	0.380	1.86	1.47	0.350	10.2
18	A1=0.006,0.006 $A2_{x,y}$=0.009	0.743	0.484	0.443	1.75	1.20	0.400	10.1

are widened, as in Figure 5-9b, a one-sided deadband results. Torques exerted by the thrusters are not sufficient to push the spacecraft into another firing region -- before this region is reached, disturbance torques push the spacecraft to the side of the deadband where it started. Such behavior involves half as many thruster firings, and thus lower fuel consumption. Two-sided deadbands can also cause increased rate errors. The two-sided deadband of Figure 5-9a ideally involves one thruster firing on each side of the deadband.

If, however, gyro time lags or other factors cause two thruster firings on one side, higher rates can be expected, and a situation in which there are two firings on *each* side of the deadband will result. From here, the number of firings on each side can continue to grow, steadily worsening rate errors and fuel consumption. A one-sided deadband, by reducing thruster firings, reduces the probability of this occurring.



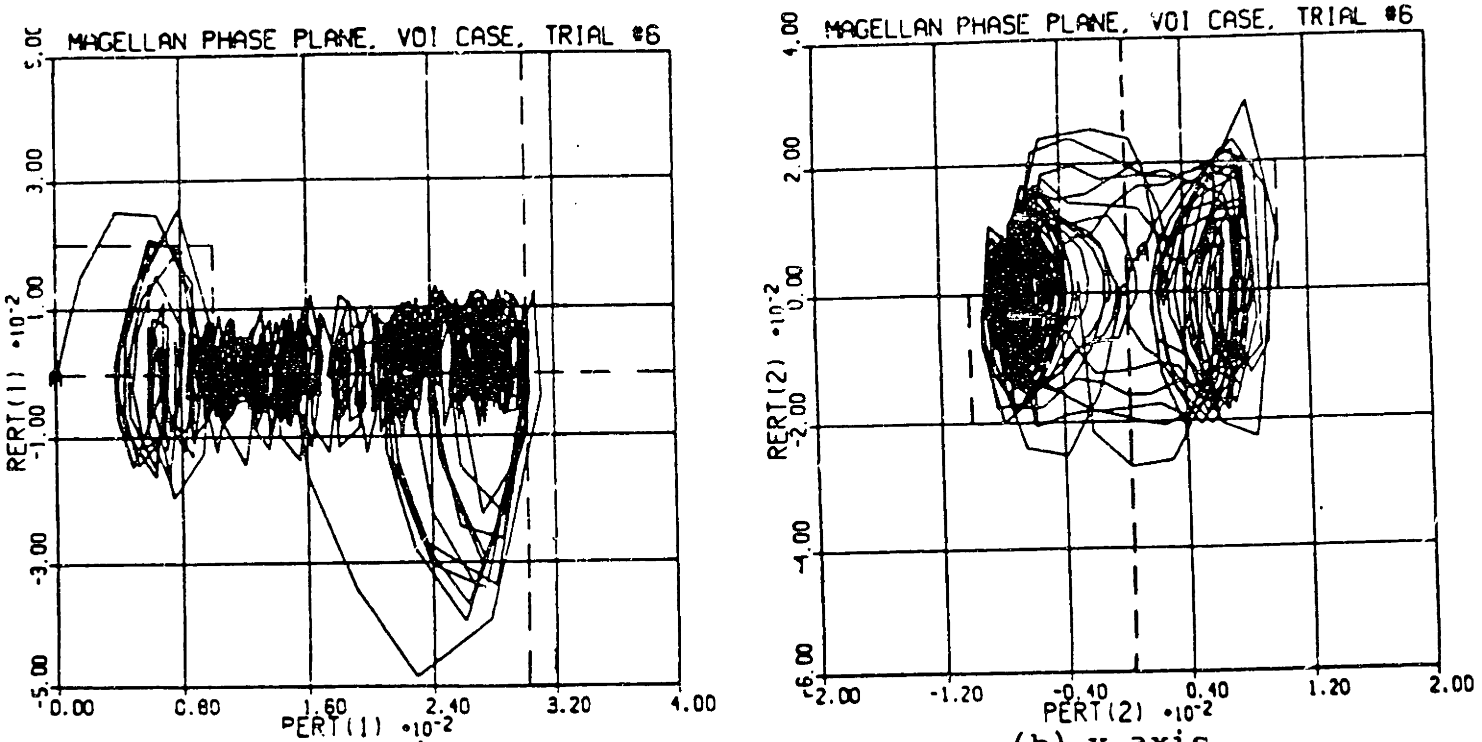
(a) Two-Sided Deadband (b) One-Sided Deadband

Figure 5-9: One and Two-Sided Deadbands in a Phase Plane

To remedy the two-sided deadband problem, deadband limits were widened. Finally, in Run 6, deadbands that were mostly one-sided were achieved. Run 6 phase plane plots are shown in Figure 5-10. Fuel consumption was reduced by almost a factor of 4, and rate errors improved by more than a factor of 3. Obviously, obtaining a one-sided deadband is very important in VOI.

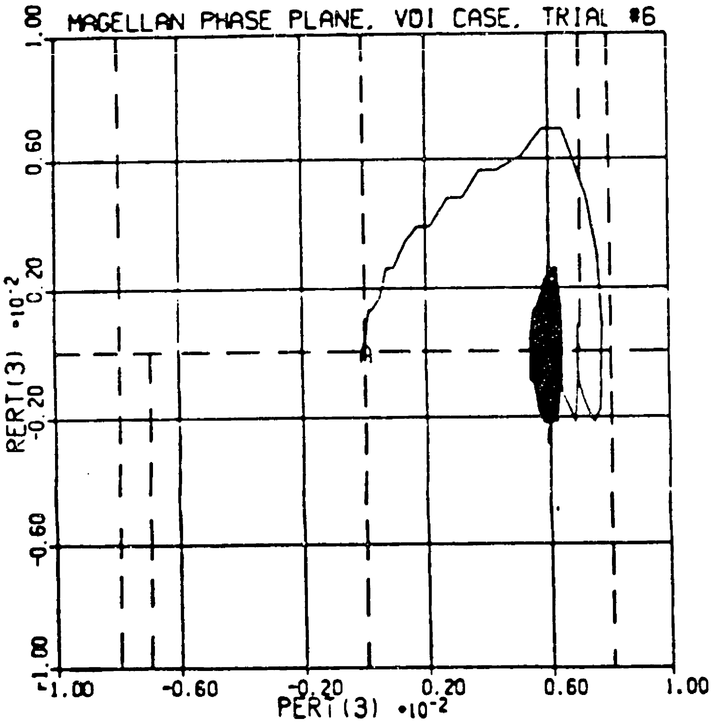
To combat excessive attitude errors, ledges A2 and A3 were adjusted. As should be expected, reducing A3 helped reduce attitude errors by penalizing them with stronger firings. In Run 8, attitude errors were found to be within acceptable limits.

With attitude errors and fuel consumption both staying within reasonable limits, an attempt was made to reduce rate errors. Setting R1 to zero on all three axes should help to



(a) x-axis

(b) y-axis

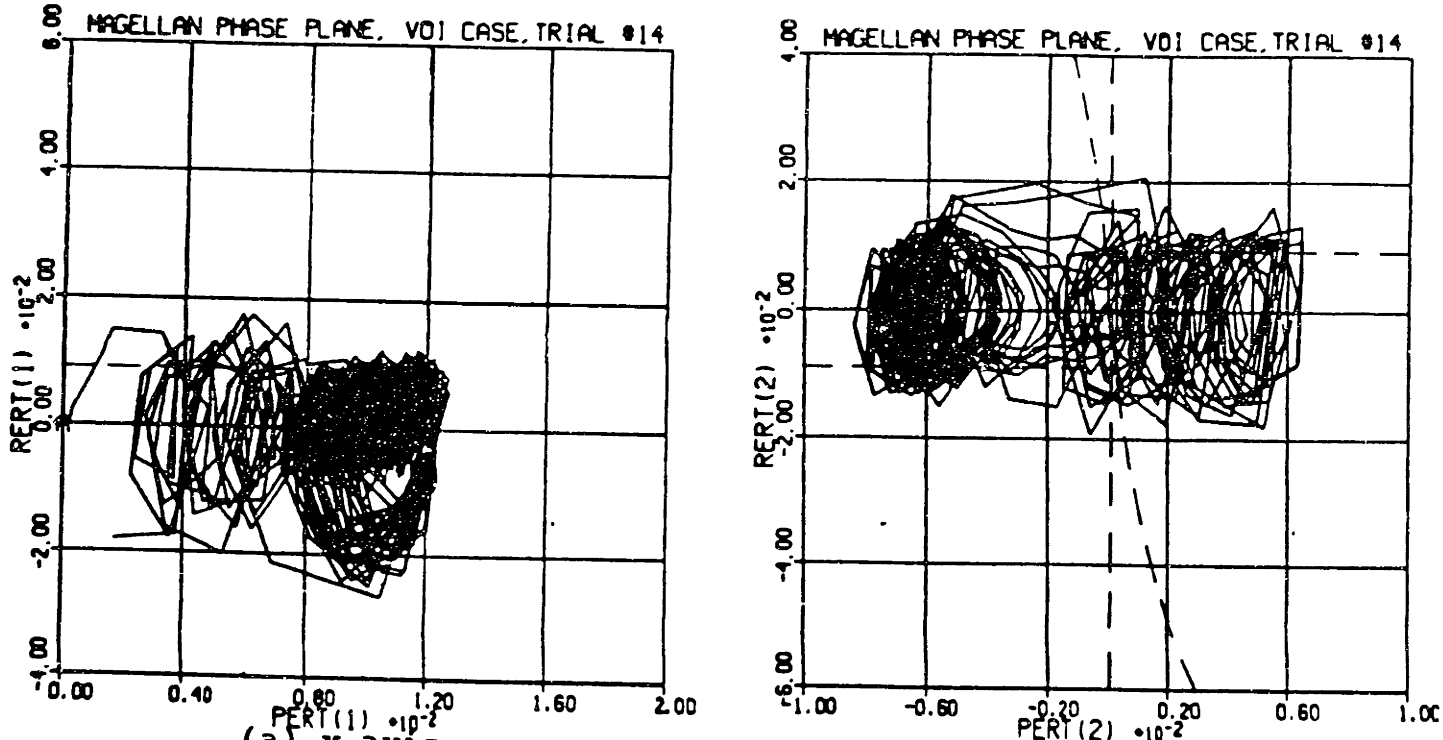


(c) z-axis

Figure 5-10: Mostly One-Sided Deadbands in the MGN attain this goal. In the normal attitude control case, R1 was set as large as possible. The rationale for this was that if the phase plane trajectory was approaching the origin too slowly, R1 would accelerate it. But this rationale is not really applicable to VOI, where disturbance torques already prevent rates from remaining low. Also, the extra region 3 firings caused by the ledge could increase fuel consumption and rate errors. In Run 9, it was found that setting R1 to zero reduced both fuel consumption and rate errors, as expected.

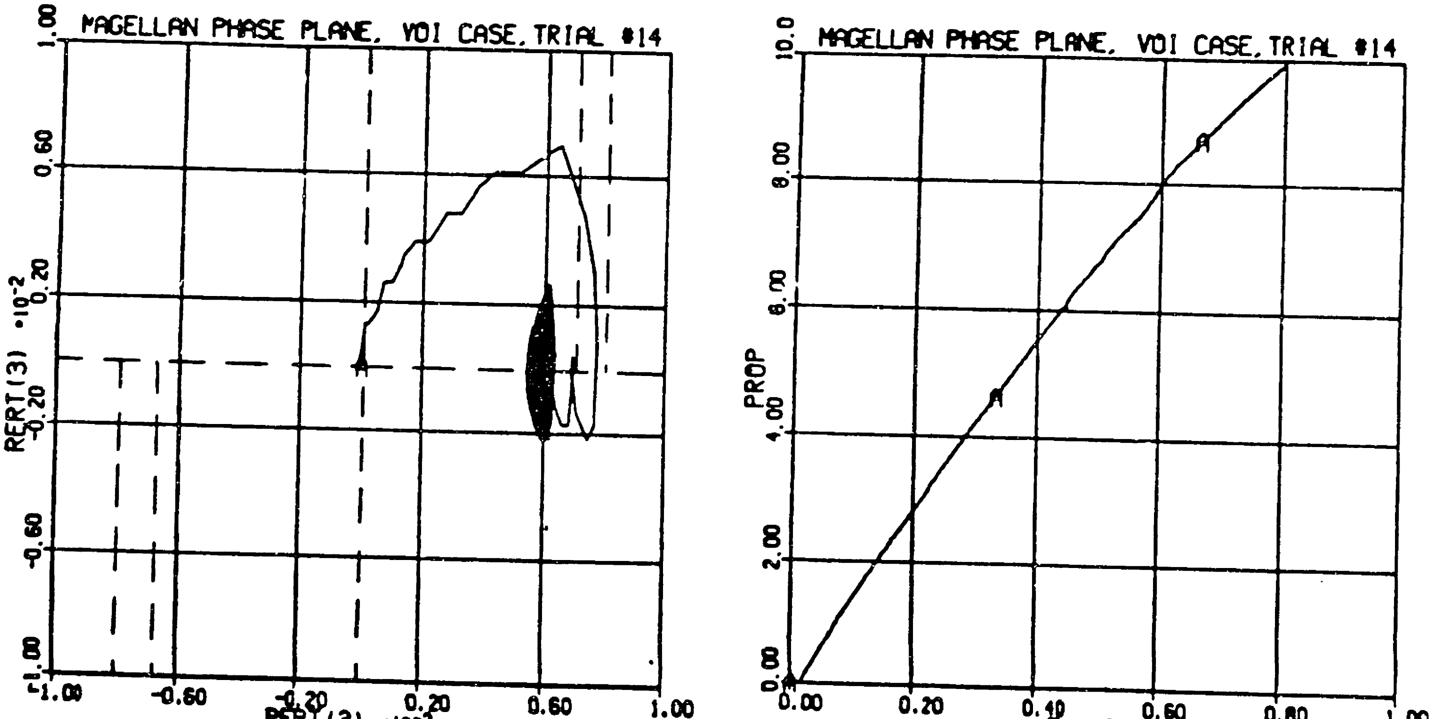
Rate errors still exceeded the desired 1.14 deg/sec about each axis, however (except for the z-axis, which was easy to control because of its low disturbances). Since pulling ledge A3 in toward the origin helped attitude errors, it was decided that pulling in ledge R2 might help rate errors. Pulling in ledge R3 would probably be counterproductive, since this would result in more region 3 firings, which is not desirable. R2 adjustments result in more region 2 firings, which will probably not cause rate or fuel problems. In Runs 10, 11, 12, 14, and 15, R2 was adjusted. Pulling in R2 was found to reduce rate errors, as expected, but only up to a certain point. If R2 was pulled in too far (Run 15), the narrower deadband caused two-sided deadband behavior to increase. The best performance was observed in Run 14. Results of this run are shown in Figure 5-11, and parameter values are listed in Table 5-VI.

In Run 14, y-axis rates are very close to the desired 1.14 deg/sec. In fact, rates are usually less than this value, and only occasionally bounce above it. On the x-axis, rates are also usually acceptable, but excursions above the acceptable limit are higher and more frequent. Figure 5-11a shows the reason for this. As the x-axis attitude limit A3 is reached, region 3 firings cause negative rates that help reduce attitude errors. Usually these negative rates are acceptable. Sometimes, however, the gyros are slow to catch up to the new attitudes caused by these rates, and thus the thrusters, thinking that attitudes are still unacceptable, fire more pulses than is necessary. This causes those unacceptable rate errors that slouch below the ϕ-axis.



(a) x-axis

(b) y-axis



(c) z-axis

(d) Propellant Consumed

Figure 5-11: Optimized MGN VOI Phase Planes

Table 5-VI: Optimized MGN VOI Parameters

parameter	units	value	
		X & Y	Z
K	$(\text{rad})^{1/2}/\text{sec}$	1.08	0.332
A1	rad	0.006	0.006
A2	rad	0.01	0.007
A3	rad	0.012	0.008
A4	rad	0.1745	0.1745
R1	rad/sec	0.0	0.0
R2	rad/sec	0.009	0.02
R3	rad/sec	0.12	0.018

The problem occurs in the fourth quadrant of the phase plane, where no ledges are present except R1. R1 has already been reduced to zero, leaving only the parabola constant, k, to have any effect on that quadrant. In Run 13, k was reduced to try and better control the rates. k was reduced so that the switching parabola would just touch the highest (in absolute value) x-axis fourth quadrant rate errors shown in Figure 5-11a. Unfortunately, doing this involves controlling rates with strong region 3 firings instead of shorter-duration region 1 or 2 firings. As a result, a two-sided deadband developed (see Figure 5-12). Since regions 1 and 2 do not appear in the fourth quadrant, it seems that this rate error problem can not be solved.

In VOI, assigning MGN phase plane parameter values is more difficult than in the normal attitude control case. k is not nearly as critical as it was before, and analytically chosen values performed poorly. However, parameter tuning was still very intuitive; geometric considerations presented good guidelines. The ease of assigning parameter values score for this system in VOI is rated a 5.

Thus, during VOI, the most important considerations in the MGN phase plane are one-sided deadbands and excessive region 3 firings. Yet even when these considerations

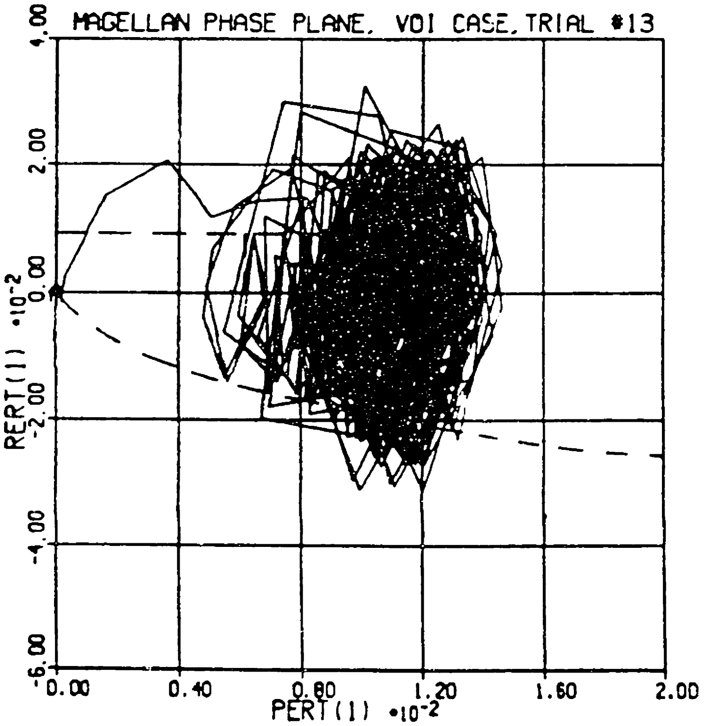


Figure 5-12: Region 3 Firings Cause a Two-Sided Deadband

carefully guide the assignment of phase plane parameters, the geometry of the phase plane prevents the reduction of rate errors to within acceptable limits. Attitude errors and fuel use, however, can be kept within desired limits, and rate errors only occasionally surpass them. MGN phase plane performance is thus reasonable, but certainly not exceptional.

5.3 The Rate Ledge Controller

5.3.1 Initial Analysis

The rate ledge controller (RLC) is shown in Figure 3-10. The recursive algorithm presented in Appendix B can be used to pick parameter values. Given a value for the deadband angle, a maximum angle, and the weighting factor λ in the index:

$$J = \int_0^{t_{final}} (\lambda + |u|)\, dt \qquad (5.16)$$

the algorithm generates the values of the parameters ϕ_r and A which will form an RLC that approximates an optimal weighted time-fuel control law.

First, however, it must be noted that the algorithm assumes a state equation of the form:

$$\ddot{\phi} = u \qquad u = +1, 0, -1 \tag{5.17}$$

This does not apply in any commonly used system of units. However, if an "angle unit" is defined as:

1 angle unit = $(\tau/I) \times \sec^2$

then the state equation will hold. Note that the size of an angle unit differs among axes and varies over the course of VOI. For VOI, an average angle unit will be used. Before using the recursive algorithm, all angles must be converted into angle units.

As inputs into the algorithm, the values of the parameters ϕ_{db}, ϕ_{max}, and λ must be chosen. For normal attitude control, the value $\phi_{db} = 4.94 \times 10^{-4}$ rad will be used, since this value worked well in the MGN phase plane. ϕ_{max} will be set to 20 degrees, the maximum attitude error observed in the MGN system. Finally, since the design evaluation criteria developed in Chapter 4 explicitly sought to minimize an index that weights time and fuel equally, the value $\lambda = 1$ was chosen. For VOI, $\phi_{db} = 0.006$ rad, as in the MGN system. Since the maximum acceptable attitude in this case is 0.806 degrees, this is a good value for ϕ_{max}. Finally, for lack of a better choice, $\lambda = 1$ was chosen.

The parameter values calculated by the recursive algorithm are given in Table 5-VII.

Table 5-VII: Run 1 RLC Normal Attitude Control Parameters

parameter	units	normal attitude control		VOI	
		X&Y	Z	X&Y	Z
ϕ_{db}	rad	4.94E-04	4.94E-04	0.006	0.006
ϕ_r	rad	0.042	0.042	0.0108	0.0108
A	sec	0.146	0.146	0.0916	0.0916

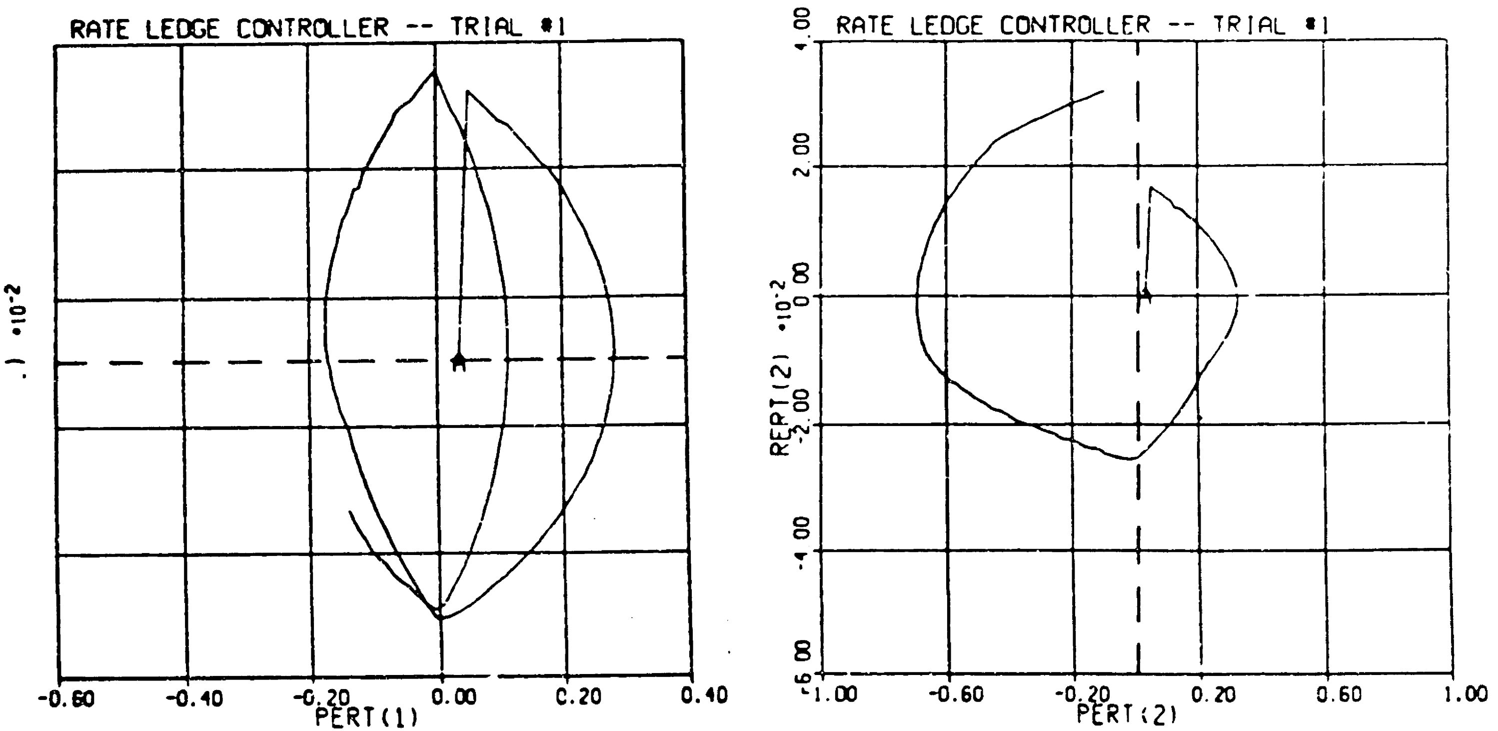
5.3.2 Normal Attitude Control

Results obtained with the parameters derived above for the normal attitude control case are shown in Figure 5-13. System performance is completely unacceptable. The phase plane trajectories never settle down into steady state and the thrusters consume large quantities of fuel. A long series of simulation runs attempting to correct these problems is summarized in Table 5-VIII.

The phase plane plots in Figure 5-13 clarify why the system performance is so bad. First of all, it can be calculated that $\dot{\phi}_{\text{lim}}$ and $\dot{\phi}_{\text{ledge}}$ (see Figure 3-10) are so large that they have no effect on system performance ($\dot{\phi}_{\text{lim}}$=0.292 rad/sec, $\dot{\phi}_{\text{ledge}}$=0.285 rad/sec). Thus, only the "slanted deadband" is having an effect on the system; the rate ledge has no effect. And the deadband is slanted at too steep a slope. As shown in Figure 5-14, when the trajectory hits the slanted deadband, the familiar approach parabola is triggered. If the slope of the slanted deadband is too large, the parabola can drastically overshoot the origin, causing the behavior shown in Figure 5-13.

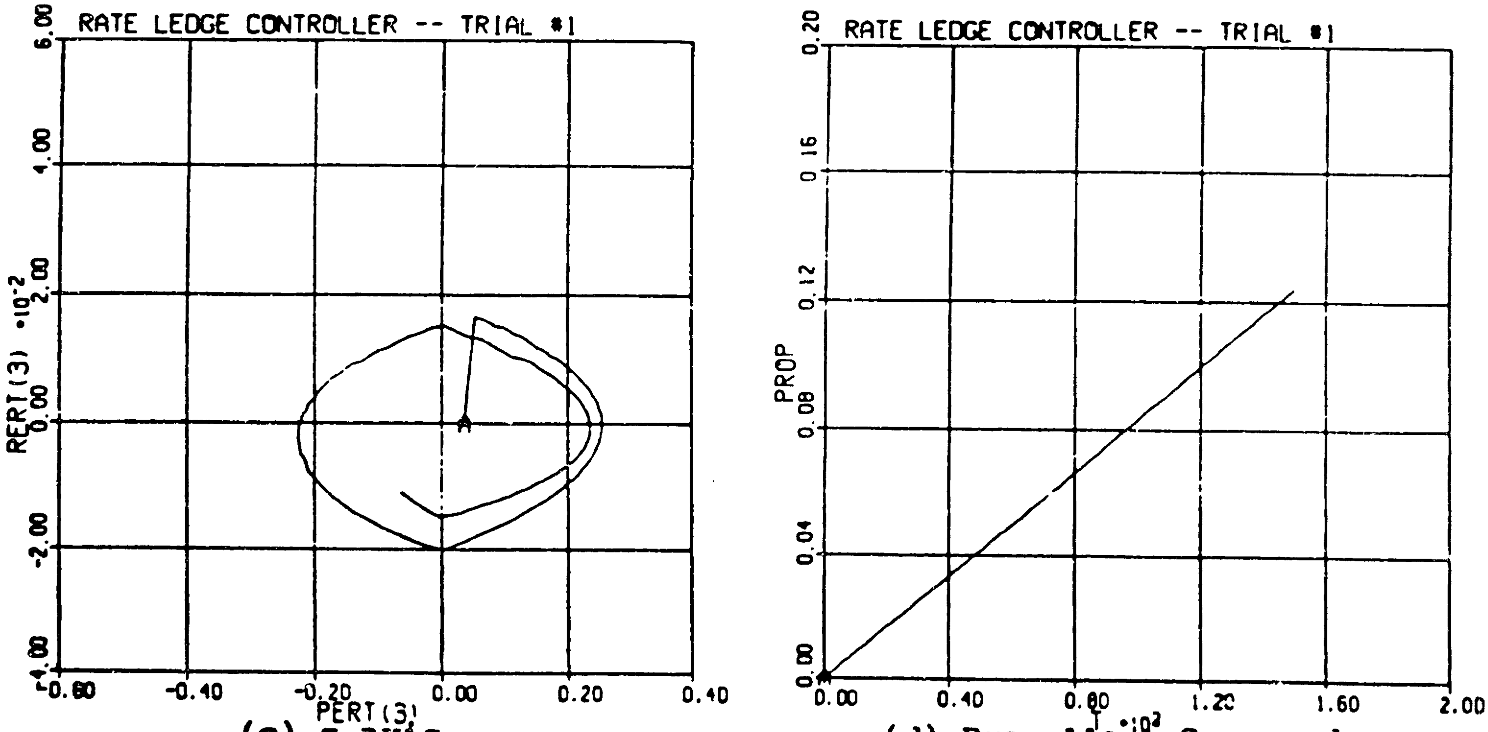
Since this behavior caused very long settling times, a higher value of the weighting factor, λ, was thought to be desirable. But as higher λ values were tried, some problems with the recursive design algorithm became apparent. Trial and error showed that the recursion does not converge for λ greater than about 10.5. White, Colburn, and Boland [26, 25] do not report this problem in their development of the algorithm, but it was discovered that in most cases that would have been of practical importance to this thesis, their algorithm diverged. The authors of the algorithm state that, "...experience has demonstrated rapid...convergence of the algorithm." And, when the recursion converged, it did indeed converge rapidly. The algorithm may function well in a limited environment, but seems useless in this more practical case.

An attempt was made to obtain satisfactory response using this algorithm, but was



(a) x-axis

(b) y-axis



(c) z-axis

(d) Propellant Consumed

Figure 5-13: Initial Results with the RLC

Table 5-VIII: Simulations of RLC Normal Attitude Control

comments	settling time (sec) X	Y	Z	trans fuel	J	limit cycle attitude X	Y	Z	limit cycle rate X	Y	Z	ss fuel
analytically determined parameters	----	----	----	----	----	----	----	----	----	----	----	----
ϕ_r=0.00743 / A=0.0629	----	----	----	----	----	----	----	----	----	----	----	----
ϕ_r=7.0E-04 / A=0.02443	----	----	----	----	----	----	----	----	----	----	----	----
ϕ_r=0.053 / A=11.8	136	156	136	2.59	58[illegible].5	9.98	9.55	9.97	9.92	9.88	9.96	0.07
ϕ_r=0.1335,0.0959 A=7.631,5.499	115	136	115	2.23	5[illegible]1.8	9.92	9.21	9.83	9.96	9.96	9.96	0.96
ϕ_r=0.34027,0.2121 A=19.44,12.12	154	172	114	1.76	674.7	9.92	9.51	9.98	9.92	9.92	9.99	0.05
ϕ_r=0.05645,0.01394 A=7.778,4.849	99	117	80	4.01	399.3	9.97	9.67	9.78	9.96	9.96	9.96	0.40
ϕ_r=0.09,0.05609 A=10.0,6.233	99	123	69	3.24	41[illegible].5	9.95	9.86	9.85	9.88	9.88	9.96	0.16
$\phi_{rx,y}$=0.08429 $A_{x,y}$=9.365	99	118	69	3.37	409.6	9.92	9.92	9.87	9.88	9.92	9.96	0.19
$\phi_{rx,y}$=0.08163 $A_{x,y}$=9.070	99	112	69	3.51	397.9	9.93	9.91	9.86	9.98	9.96	9.96	0.32
$\phi_{rx,y}$=0.07438 $A_{x,y}$=8.2645	91	107	69	3.53	38[illegible].4	9.87	9.85	9.91	9.96	9.96	9.96	0.34
$\phi_{rx,y}$=0.07007 $A_{x,y}$=7.7855	85	104	69	3.57	3[illegible]3.9	9.81	9.81	9.86	9.96	9.96	9.96	0.52
$\phi_{rx,y}$=0.066122 $A_{x,y}$=7.3469	80	102	69	1.68	361.3	9.87	9.80	9.88	9.96	9.96	9.99	0.51
$\phi_{rx,y}$=0.0625 $A_{x,y}$=6.9444	75	96	69	3.69	35[illegible].1	9.84	9.87	9.87	9.96	9.92	9.99	0.80
ϕ_r=0.07716,0.069252 A=7.716,6.9252	----	----	----	----	----	----	----	----	----	----	----	----
ϕ_r=0.06964,0.05609 A=7.330,6.233	----	----	----	----	----	----	----	----	----	----	----	----
$\phi_{rx,y}$=0.075061 $A_{x,y}$=7.5061	70	85	69	3.54	3[illegible]1.0	9.90	9.87	9.92	9.97	9.99	9.98	0.[illegible]
$\phi_{rx,y}$=0.082755 $A_{x,y}$=7.8814	----	----	----	----	----	----	----	----	----	----	----	----
$\phi_{rx,y}$=0.08507 $A_{x,y}$=8.1019	----	----	----	----	----	----	----	----	----	----	----	----
ϕ_r=0.089978,0.0625 A=8.379,6.579	----	----	----	----	----	----	----	----	----	----	----	----
ϕ_r=0.08651,0.05004 A=8.239,5.886	----	----	----	----	----	----	----	----	----	----	----	----
run #17 values ϕ_{db}=8.0E-04	69	93	69	3.56	347.2	9.06	9.23	9.24	9.96	9.98	9.96	5.08
ϕ_{db}=1.0E-03	----	----	----	----	----	----	----	----	----	----	----	----
ϕ_{db}=9.0E-04	69	85	69	3.59	318.4	9.40	9.62	9.25	9.97	9.97	9.98	7.96
ϕ_{db}=9.5E-04	----	----	----	----	----	----	----	----	----	----	----	----
ϕ_{db}=9.25E-04	----	----	----	----	----	----	----	----	----	----	----	----
ϕ_{db}=1.5E-03	----	----	----	----	----	----	----	----	----	----	----	----
ϕ_{db}=1.75E-03	----	----	----	----	----	----	----	----	----	----	----	----

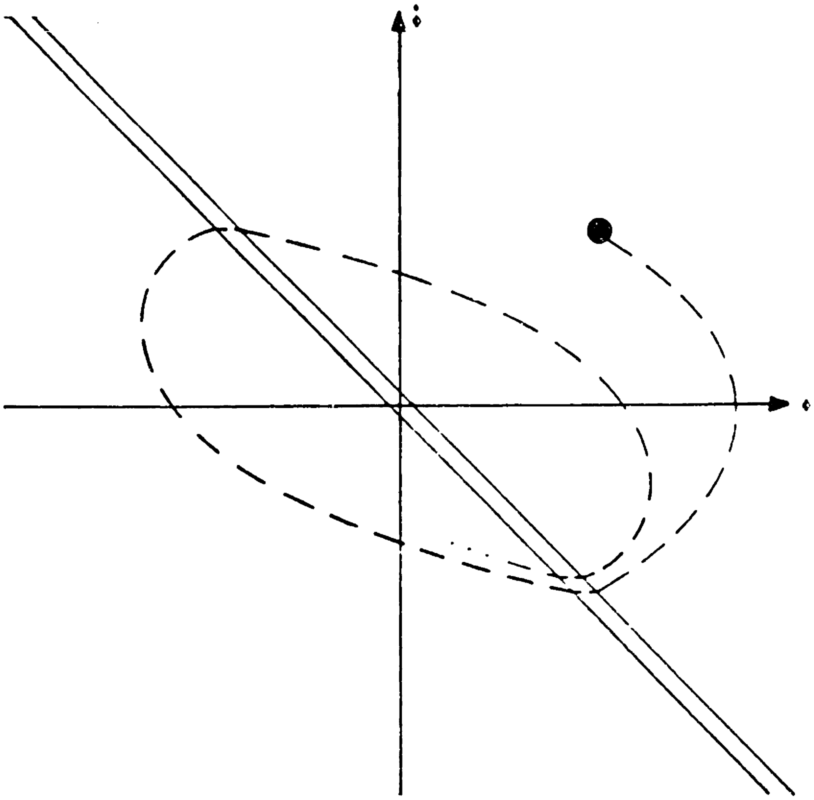


Figure 5-14: Deadband Slanted at Too Steep a Slope

abandoned after only two runs. ϕ_{max} was reduced to 3 degrees (Run 2), and λ and ϕ_{max} were adjusted (Run 3), but without success. Performance was not observed to improve, and almost every time a new value of λ or ϕ_{max} was tried, the recursive algorithm failed to converge.

A new technique, based on the intuitive approach parabolas discussed in conjunction with the MGN phase plane, was developed. Figure 5-15 shows the ideal system behavior, slightly simplified to facilitate analysis (the deadband is assumed to have negligible width). Starting at its initial conditions, the trajectory develops until it is caught by the deadzone between the rate ledges. With no thruster firings, it continues at a constant rate until the ledge ends, where the approach parabola is triggered and it is pushed into the origin. The equation of the parabola is:

$$\dot{\phi}^2=-\frac{2\tau}{I}\phi=-k^2\phi \quad \textit{for the parabola} \tag{5.18}$$

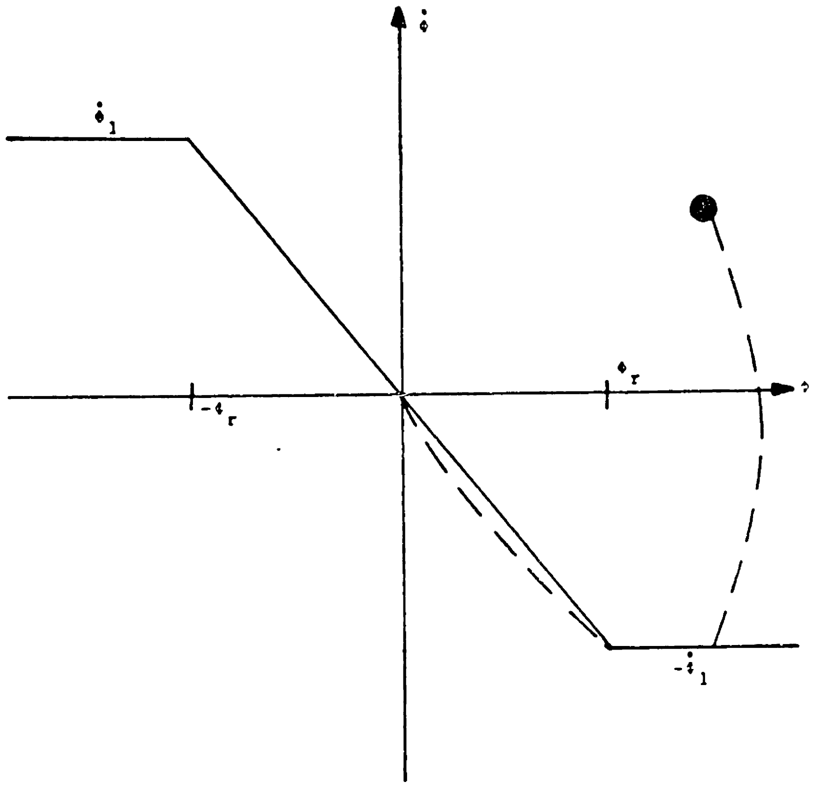


Figure 5-15: Simplified RLC to Aid in Parameter Design

while the equation of the slanted line (which, in this simplified phase plane, exactly intersects the origin) is:

$$\dot{\phi}=-\frac{1}{A}\phi \quad \textit{for the slanted line} \tag{5.19}$$

These curves intersect when $\phi=\phi_r$ and $\dot{\phi} = \dot{\phi}_l$. From this fact and equations (5.18) and (5.19), the relations:

$$A=\frac{\dot{\phi}_l}{k^2} \tag{5.20}$$

$$\phi_r = A\dot{\phi}_l \tag{5.21}$$

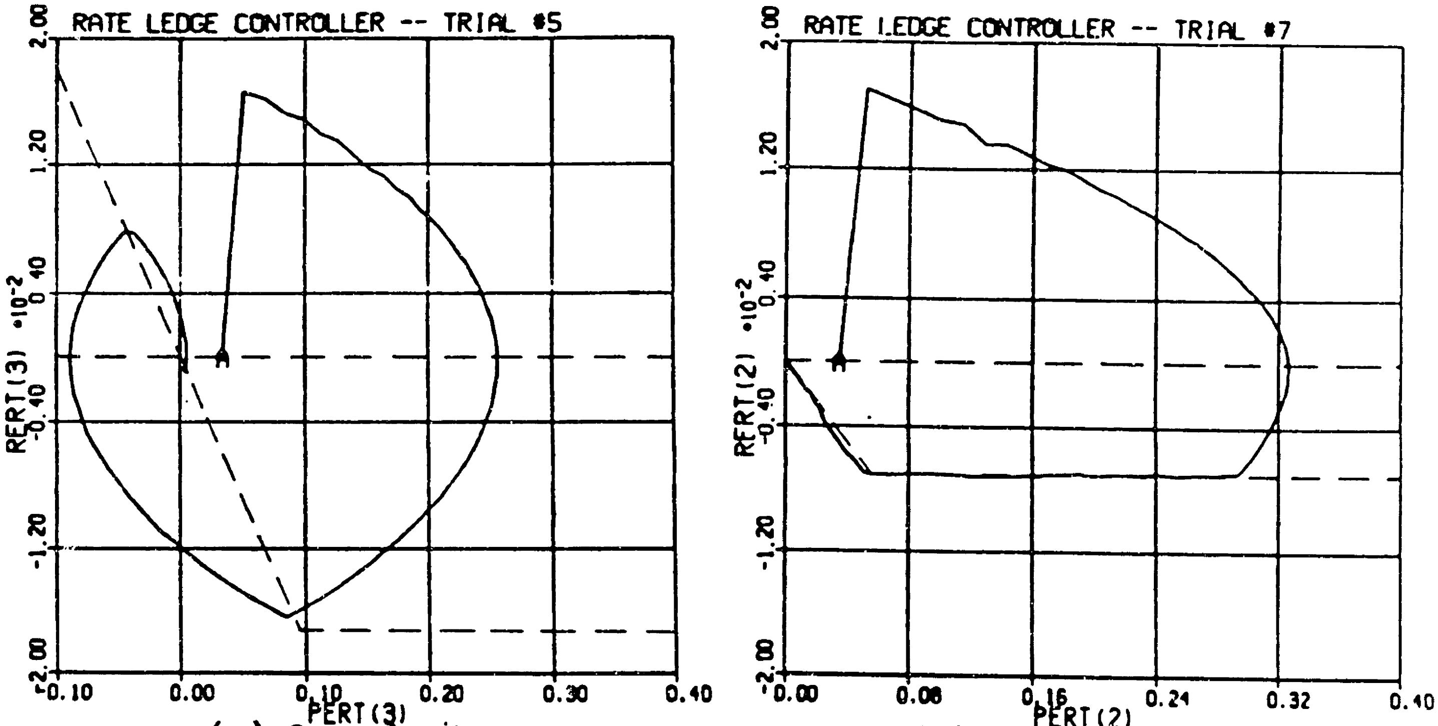
can be derived. Equations (5.20) and (5.21) replace the parameters ϕ_r and A, which are not easy to design intuitively, with the more intuitive parameters k and $\dot{\phi}_l$. The discussion of parabolas in the MGN phase plane makes k easy to understand, and $\dot{\phi}_l$ is simply the rate at which the trajectory levels out. Decreasing $\dot{\phi}_l$ increases times and decreases fuel consumption, while increasing it has just the opposite effect.

However, the value of k that worked in the MGN phase plane should not be expected to work perfectly here. Various simplifying assumptions prevent equations (5.20) and (5.21) from holding precisely. For example, phase plane trajectories will not level out at a rate of precisely $\dot{\phi}_l$; the rate will actually be slightly larger than this, within the limits of the deadzone. Also, the fact that the slanted deadband has a non-zero width will cause the parabola to trigger at an angle slightly less than ϕ_r. Both of these factors will tend to raise the optimal value of the parameter k.

Using a constant $\dot{\phi}_l$ value, k was optimized. Then $\dot{\phi}_l$ was adjusted. Unfortunately, adjusting $\dot{\phi}_l$ was found to destroy the well-behaved parabolas created by the optimized k values. Overshoot (Figure 5-16a) or undershoot (Figure 5-16b) resulted. Because of this, after $\dot{\phi}_l$ was adjusted, k had to be readjusted. In the MGN phase plane the switching line was parabolic, so it could be set parallel to a "natural parabola" (see page 87) and be assured that a trajectory hitting it at any given rate would switch from negative thruster firings to positive ones at the appropriate angle. Now, however, the switching line is linear. It can never be parallel to any natural parabola, so the k value that it represents changes with the rate at which it is reached. The best choice of a k value is the one that results in an appropriate approach parabola when the slanted deadband is hit at the rate $\dot{\phi} = \dot{\phi}_l$.

Because of this, this system should not be expected to perform well for arbitrary initial conditions. When the system is optimized, any set of initial conditions that causes a trajectory that can be captured by the rate ledge will probably result in optimal performance. If, however, this rate ledge is not hit and the slanted deadband is hit at a rate far from $\dot{\phi} = \dot{\phi}_l$, the natural parabola will be inappropriately triggered and overshoot will result. This system will be optimized for its chosen initial conditions, but performance for other initial conditions could be very bad indeed.

The index J was minimized in Run 17, with parameter values of $\dot{\phi}_{lx} = \dot{\phi}_{ly} = 1\text{x}10^{-2}$ rad/sec, $\dot{\phi}_{lz} = 9\text{x}10^{-3}$ rad/sec, $k_x = k_y = 0.0365$ sec^{-1}, and $k_z = 0.0380$ sec^{-1} were used. The



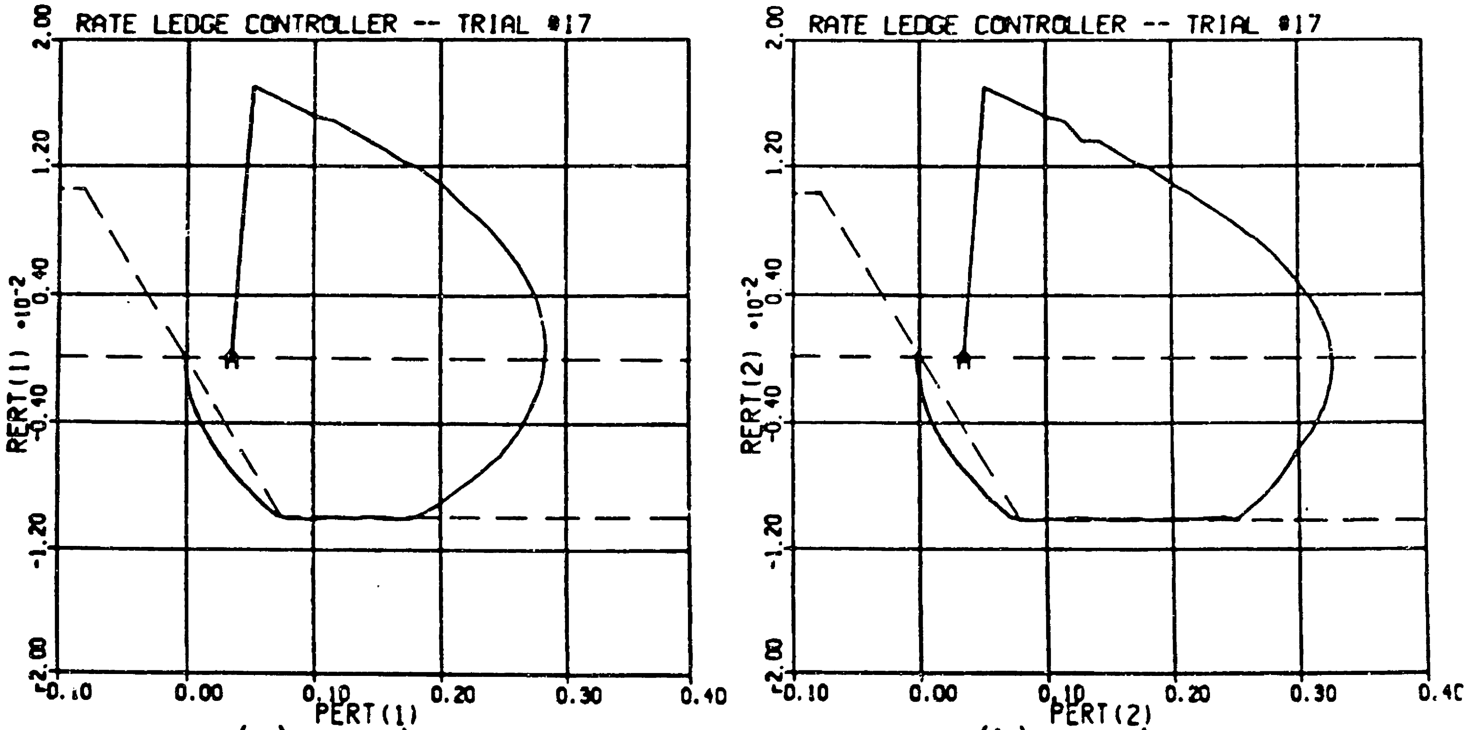
(a) Overshoot

(b) Undershoot

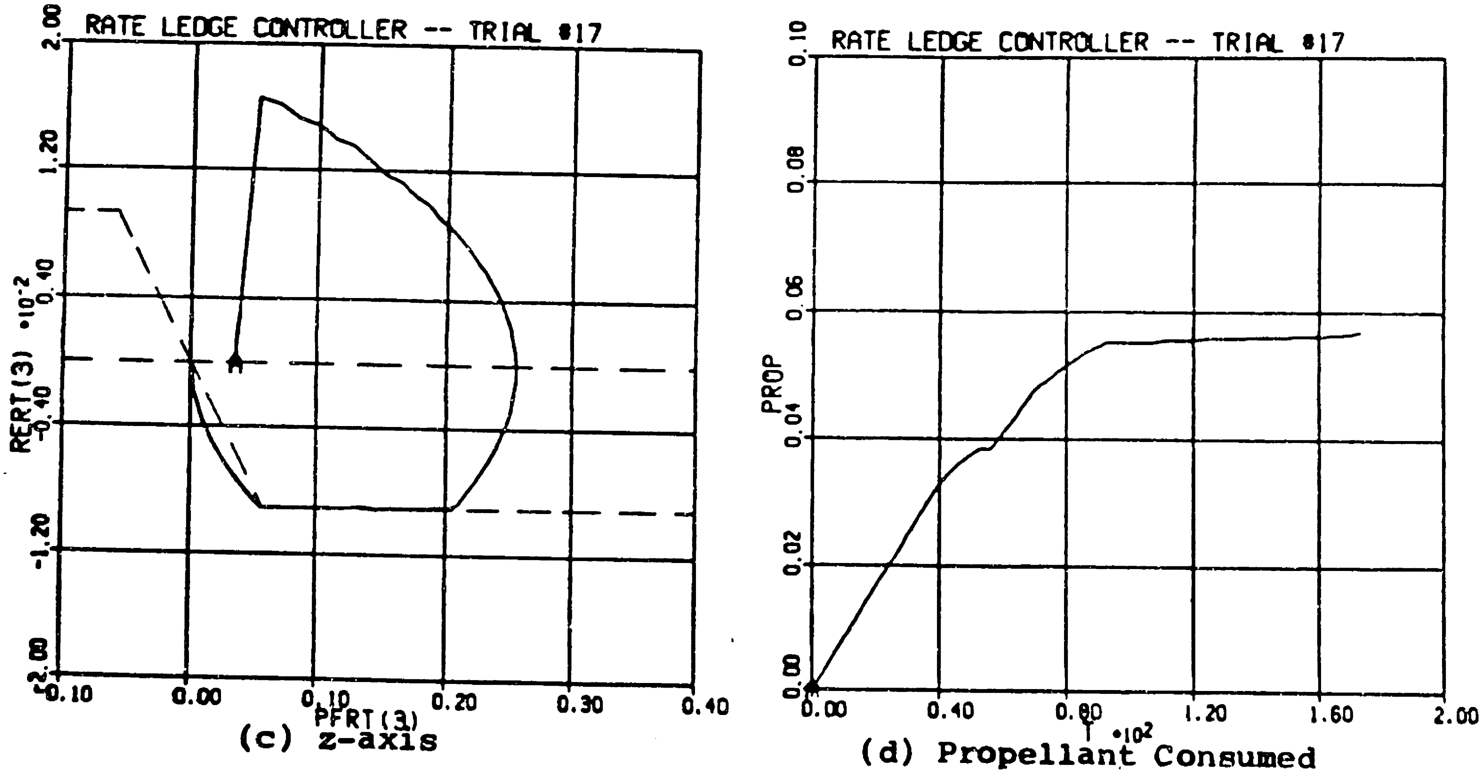
Figure 5-16: Overshoot and Undershoot in the RLC

results of this run are shown in Figure 5-17. Approach parabolas neither undershoot nor overshoot and the performance index J is slightly better than the best value obtained with the MGN phase plane (J_{RLC}=341). Although settling times were slightly worse than in the MGN, fuel consumption was much better. This is not surprising, since the MGN phase plane is based on a minimum time derivation, and varying its parameters was found to have only a small effect on fuel. The RLC is based on the minimization of a time-fuel index, and allows wider variation in fuel consumption.

In Runs 22-29, the steady state parameter ϕ_{db} was varied. One of the first things that was noticed about this was that adjusting ϕ_{db} affected transient response. Sometimes overshoot or undershoot would result. Also, narrower deadbands tended to cause higher transient fuel consumption. This is explained in Figure 5-18. As coupling torques affect a trajectory passing through a wide deadband (5-18a), they occasionally cause the trajectory to hit one side of the deadband, initiating a thruster firing. With a narrower deadband (5-18b), deadband limits are hit more often, thus raising fuel consumption. Unlike the



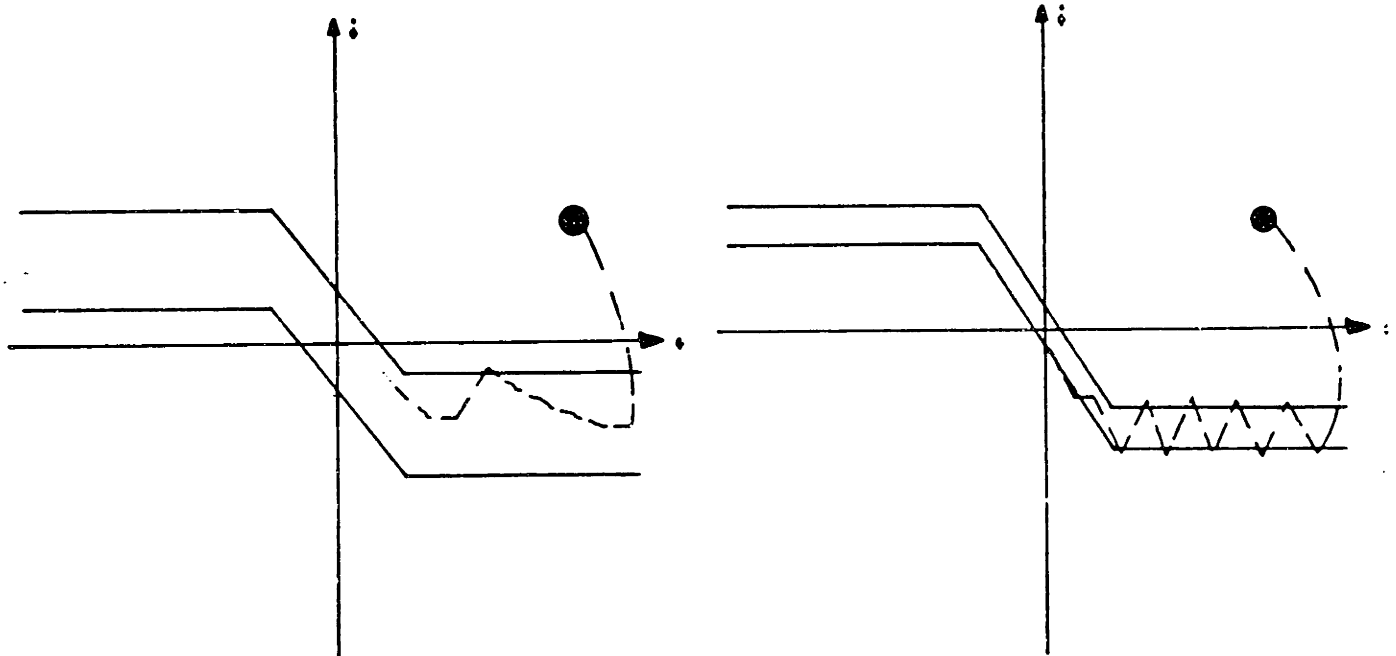
(a) x-axis (b) y-axis



(c) z-axis (d) Propellant Consumed

Figure 5-17: Optimal RLC Transient Performance

MGN phase plane, where transients and steady state could be tuned independently, the two are not independent in the RLC. This is an unfortunate complication.



(a) Wide Deadband: Few Thruster Firings

(b) Narrow Deadband: Many Thruster Firings

Figure 5-18: RLC Deadband Width Affects Fuel Consumption

The RLC contains another unfortunate complication in steady state. Ordinarily, one would expect widening the deadband to cause larger limit cycle attitudes and reduced steady state fuel consumption. The design criterion in steady state is to have the limit cycle attitude score and the steady state fuel score as close together as possible. If the deadband is widened to try and attain this, a problem develops. When the deadband is widened beyond a certain point, steady state fuel consumption starts to *increase*. Fuel consumption is plotted against deadzone width in Figure 5-19. As the deadband is initially widened, the rate of steady state fuel consumption tends asymptotically to zero, as expected. When the deadband reaches about 9×10^{-4} rad, however, the rate of fuel consumption begins to increase. Eventually, consumption resumes its asymptotic approach to the origin. This is very unfortunate, as the "fuel bump" occurs right over the point to which the system should be designed. Deadbands of less than 9×10^{-4} rad have attitude scores much greater than their fuel scores, while deadband values to the right of the "bump" have fuel scores much greater than their attitude scores. The "bump" prevents the system from ever realizing equal attitude and fuel scores. The best that can be done is to set the deadband to the value right at the bottom of the valley that precedes the "bump", or at $\phi_{db} = 9 \times 10^{-4}$ rad. This results in the lowest fuel consumption attainable without drastically raising limit cycle attitudes.

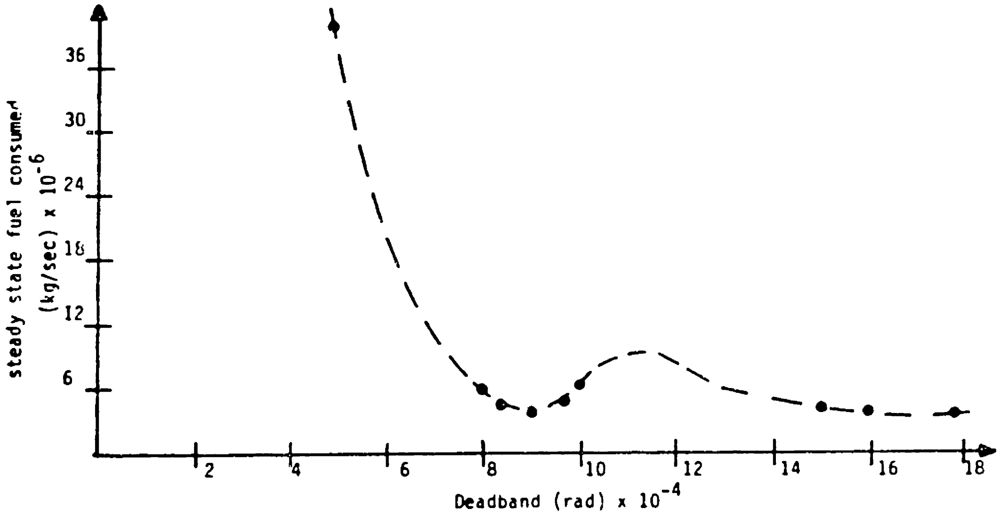


Figure 5-19: RLC Steady State Fuel Use Rate vs Deadband

What causes the "fuel bump"? When ϕ_{db} is small, as in Figure 5-20a, thruster firings are strong enough to push the trajectory rapidly back and forth from one side of the deadband to another. Fuel consumption is thus very high. As the deadband is widened, this bouncing back and forth ceases (Figure 5-20b) and fewer thruster firings occur. There is much more time between deadband crossings. After a point, though, this increased time becomes a problem. In a longer period of time, torques about perpendicular axes have more of a chance to exert an influence. Inter-axis coupling has enough time to accumulate larger changes in rates, so, by the time a trajectory has crossed the deadband, more than a single thruster firing is needed to change the sign of the spacecraft rate error. Figure 5-20c shows a trajectory occurring near the top of the "fuel bump". Notice that, while the trajectory shown in Figure 5-20b strikes the right side of the deadband only once, the trajectory in 5-20c hits it three times. As the deadband is widened still further, steady state fuel use resumes its drop. The increased time between firings begins to outweigh the effects of coupling. In the limit, an infinitely wide deadband would result in no fuel consumption at all.

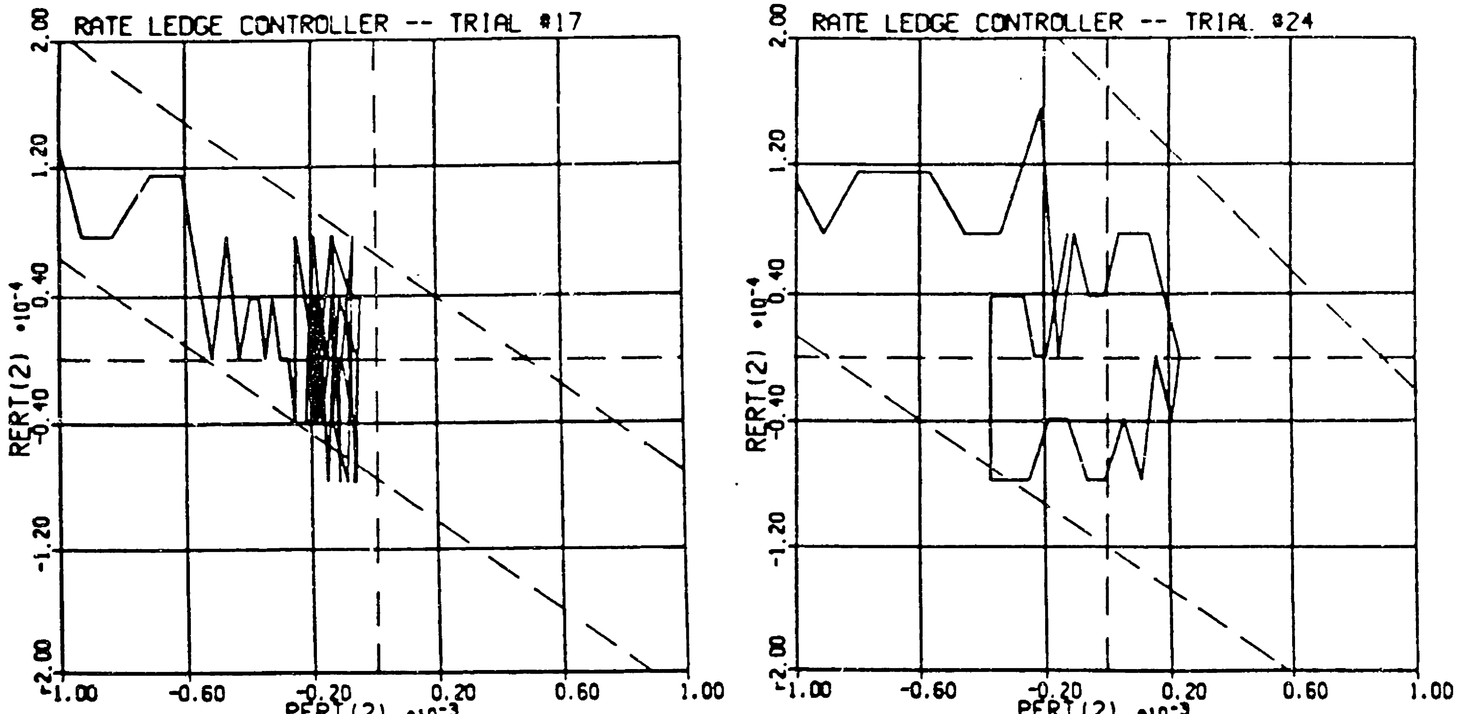
The deadband value that optimizes steady state also reduces the index J. In Run 24, J = 338. Results of this run are plotted in Figure 5-21. The results are summarized in Table 5-IX, and optimized parameter values are listed in Table 5-X.

It is difficult to assign parameter values to the RLC. There are only three independent parameters, but the initial choice of parameter values is very difficult to make. Although parameter adjustment from geometric considerations is still rather intuitive, the parameters can not be adjusted independently, resulting in a long, laborious optimization process. The "ease of assigning parameter values" score for this system is a 3.

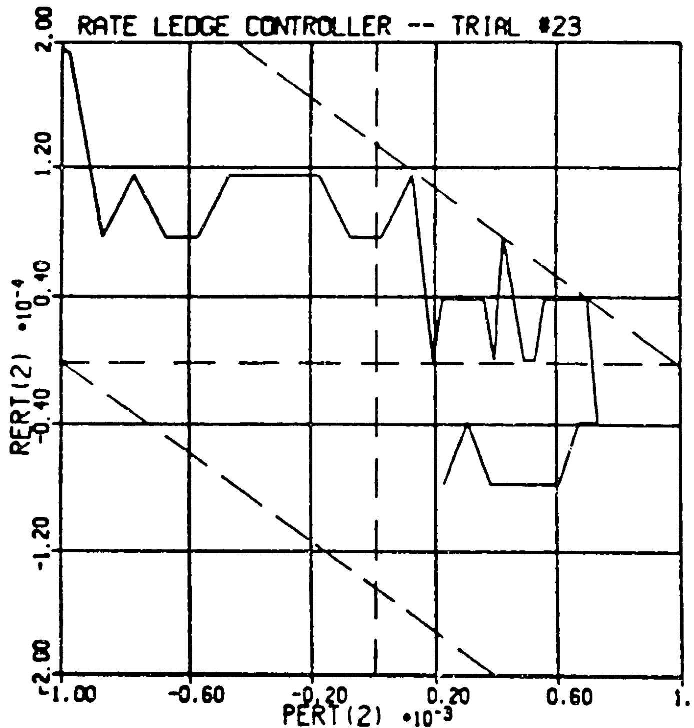
To summarize, the recursive algorithm given in Appendix B does not converge in most practical cases. Using the intuitions gained in designing the MGN phase plane, parameters can be chosen, but the optimization process is long and slow. After optimizing the system, a J value of 338 can be obtained -- better than that achieved by the MGN phase plane. This low J value is misleading, however. A different set of initial conditions from those used here could seriously alter system performance. Also, the RLC is incapable of performing as well as the MGN phase plane in steady state because of the steady state "fuel bump" problem.

5.3.3 VOI

A simulation of VOI run with the parameter values determined in section 5.3.1 does not show very good performance. Results of the simulation are shown in Figure 5-22. Two-sided deadbands cause unacceptably large attitude and rate amplitudes and high fuel consumption. As usual, a series of runs was initiated in an attempt to improve performance. A summary of these runs is presented in Table 5-XI.

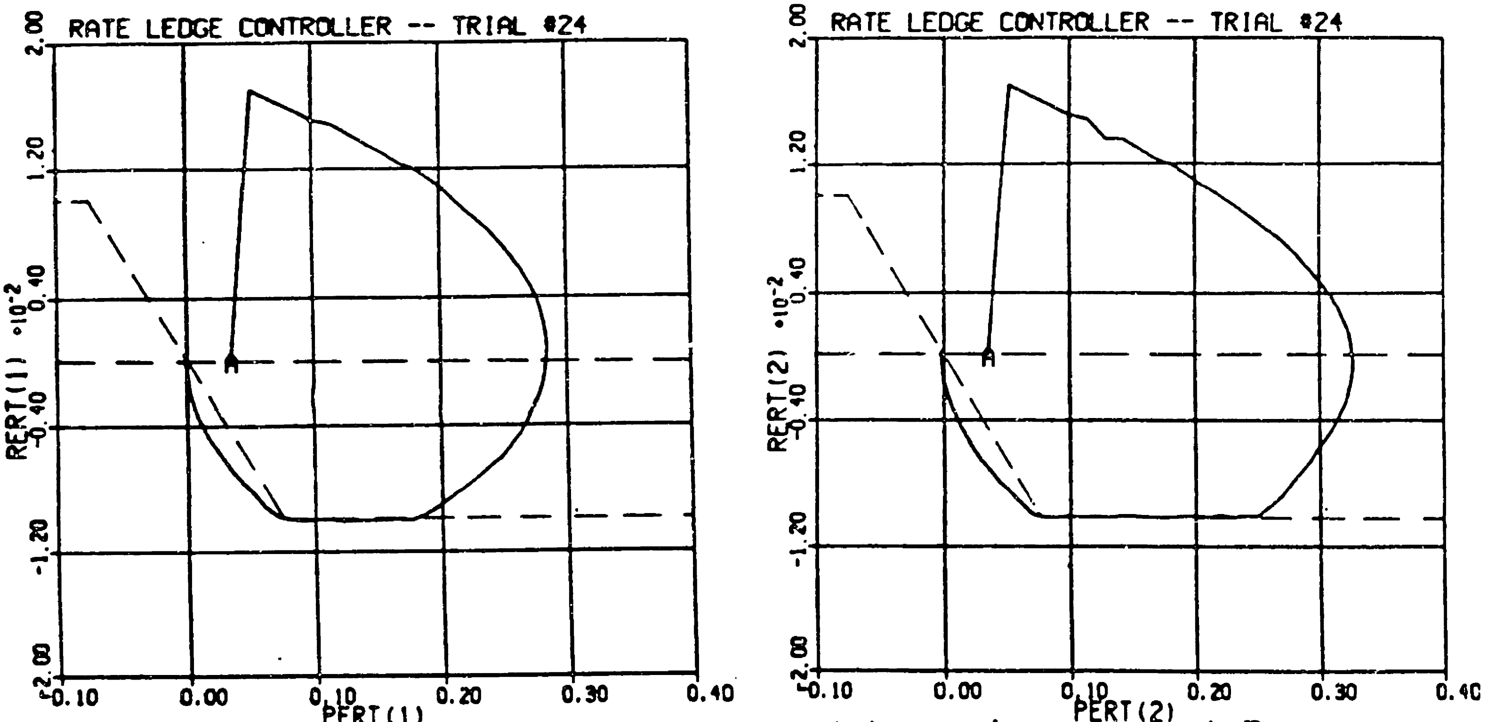


(a) Narrow Deadband (b) Wide Deadband



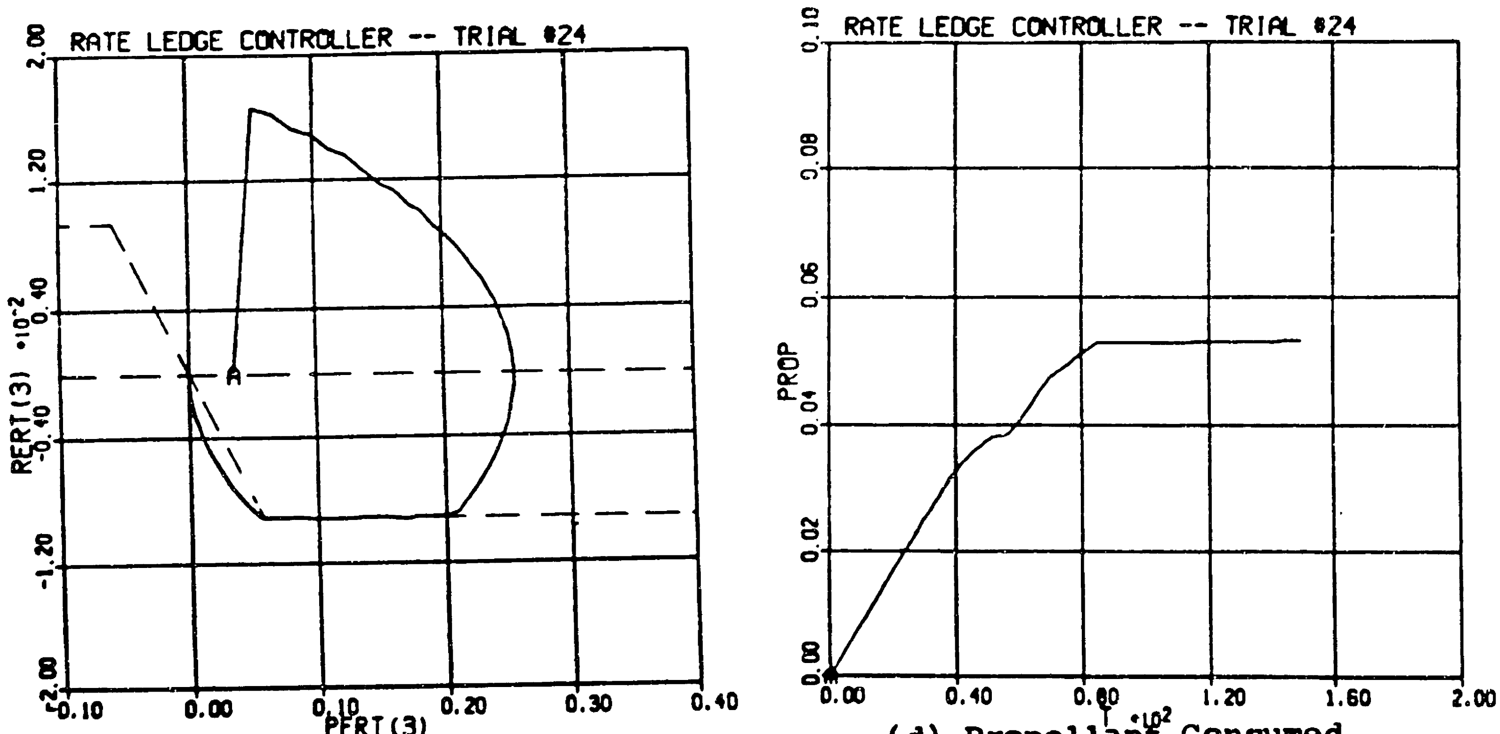
(c) "Fuel Bump Deadband"

Figure 5-20: RLC Steady State with Different Deadbands



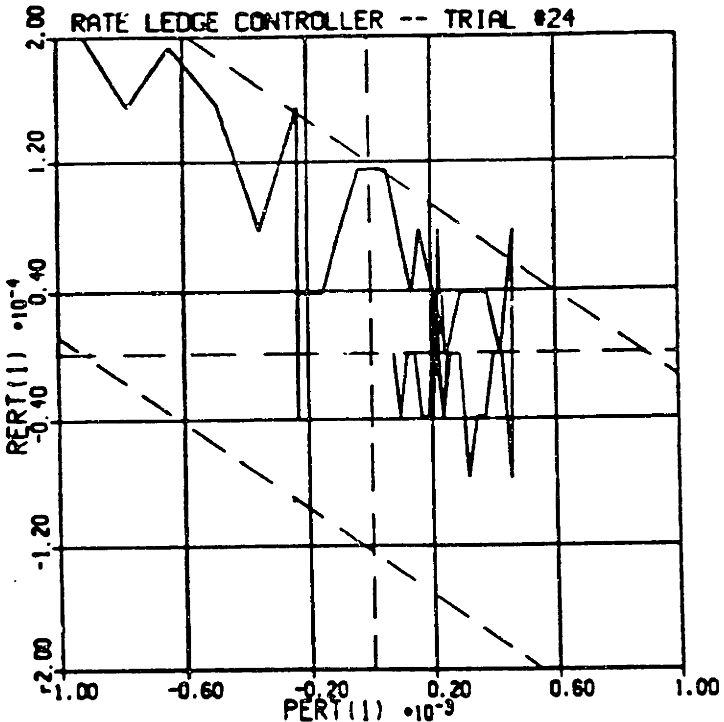
(a) x-axis Transient Response

(b) y-axis Transient Response

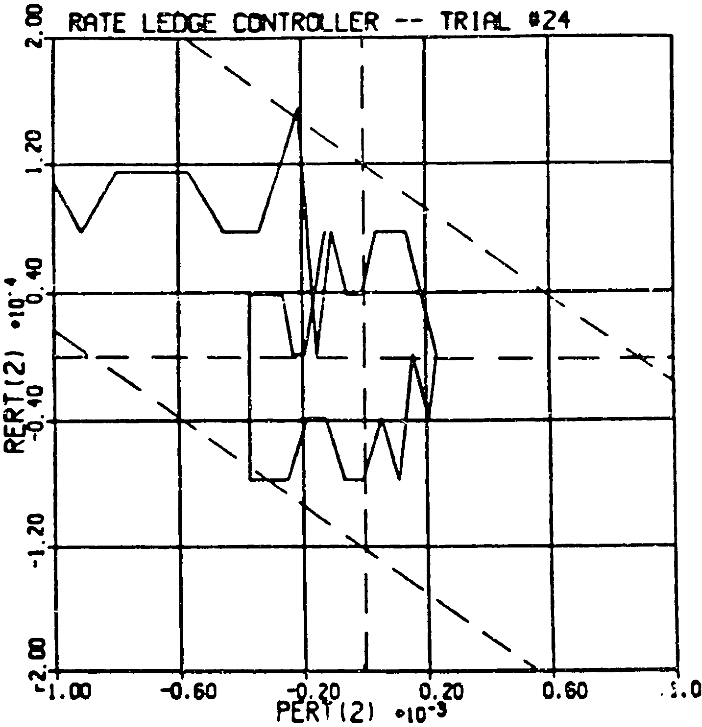


(d) Propellant Consumed

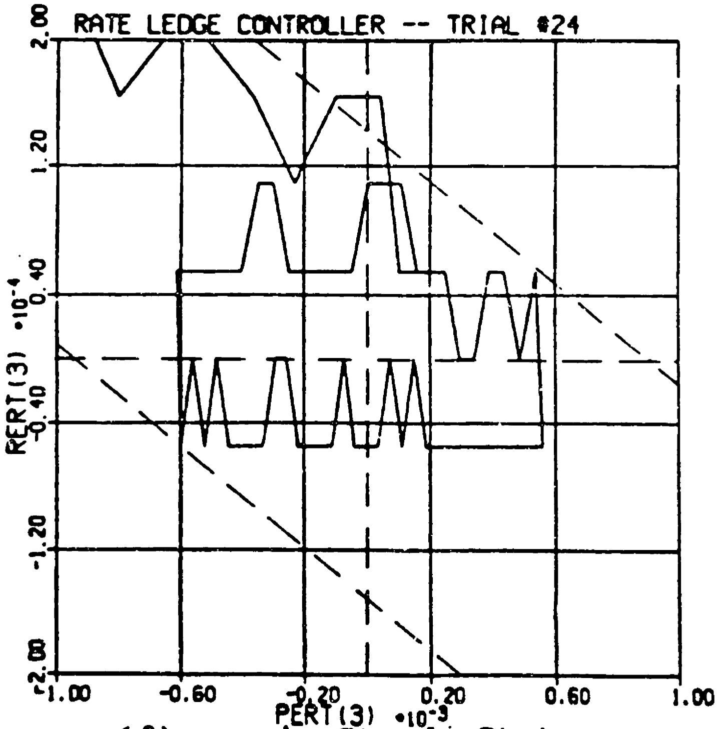
Figure 5-21: Optimal RLC Normal Attitude Control



(d) x-axis Steady State



(e) y-axis Steady State



(f) z-axis Steady State

Table 5-IX: RLC Normal Attitude Control Scores

Transient Scores

	X	Y	Z	AVG
settling time	8.70	7.06	8.70	8.15
fuel consumption				3.59
rate excursion	10.0	10.0	10.0	10.0
attitude excursion	3.41	1.25	4.68	3.11

$J = 338.4$

Steady State Scores

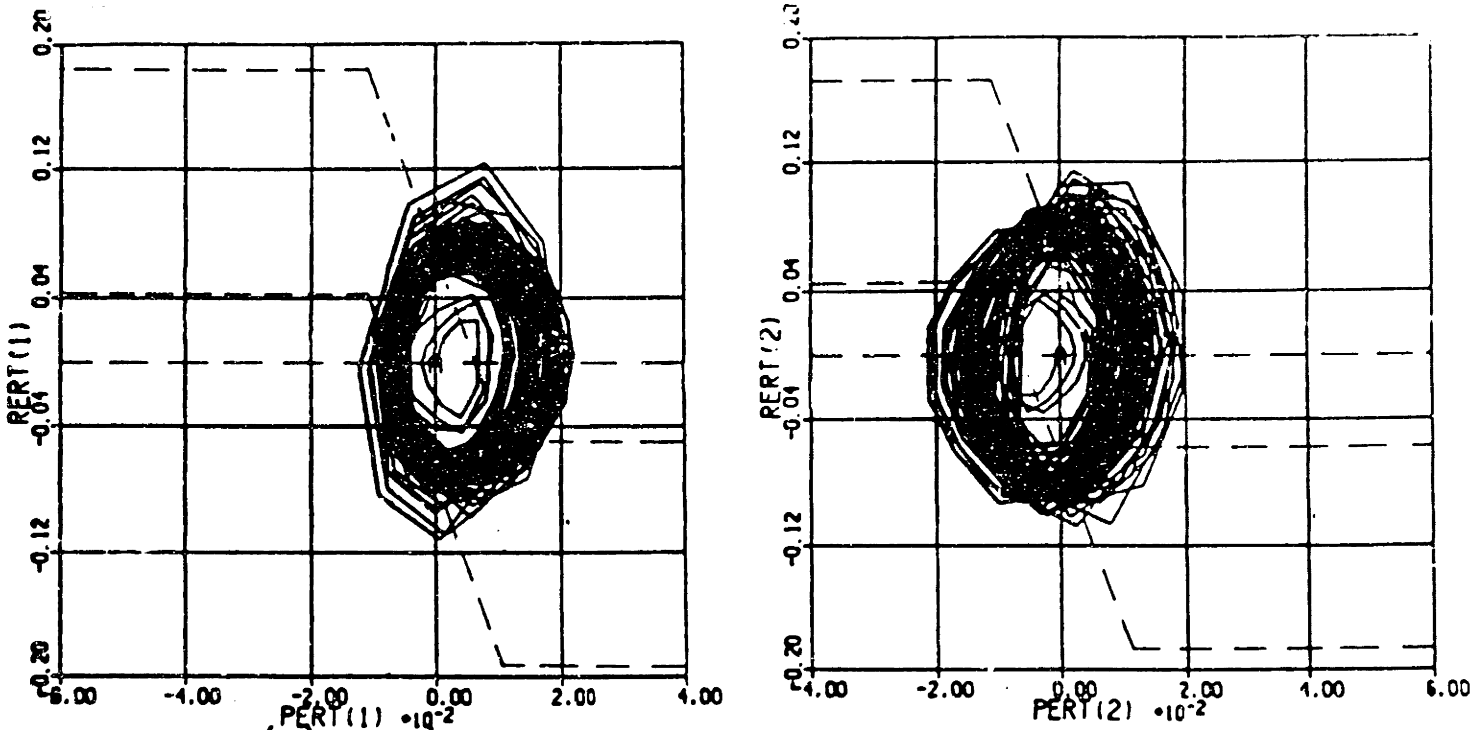
	X	Y	Z	AVG
fuel consumption				7.96
limit cycle rate	9.97	9.97	9.98	9.97
limit cycle attitude	9.40	9.62	9.25	9.42

Overall

	X	Y	Z	AVG
ease of assigning parameter values				3

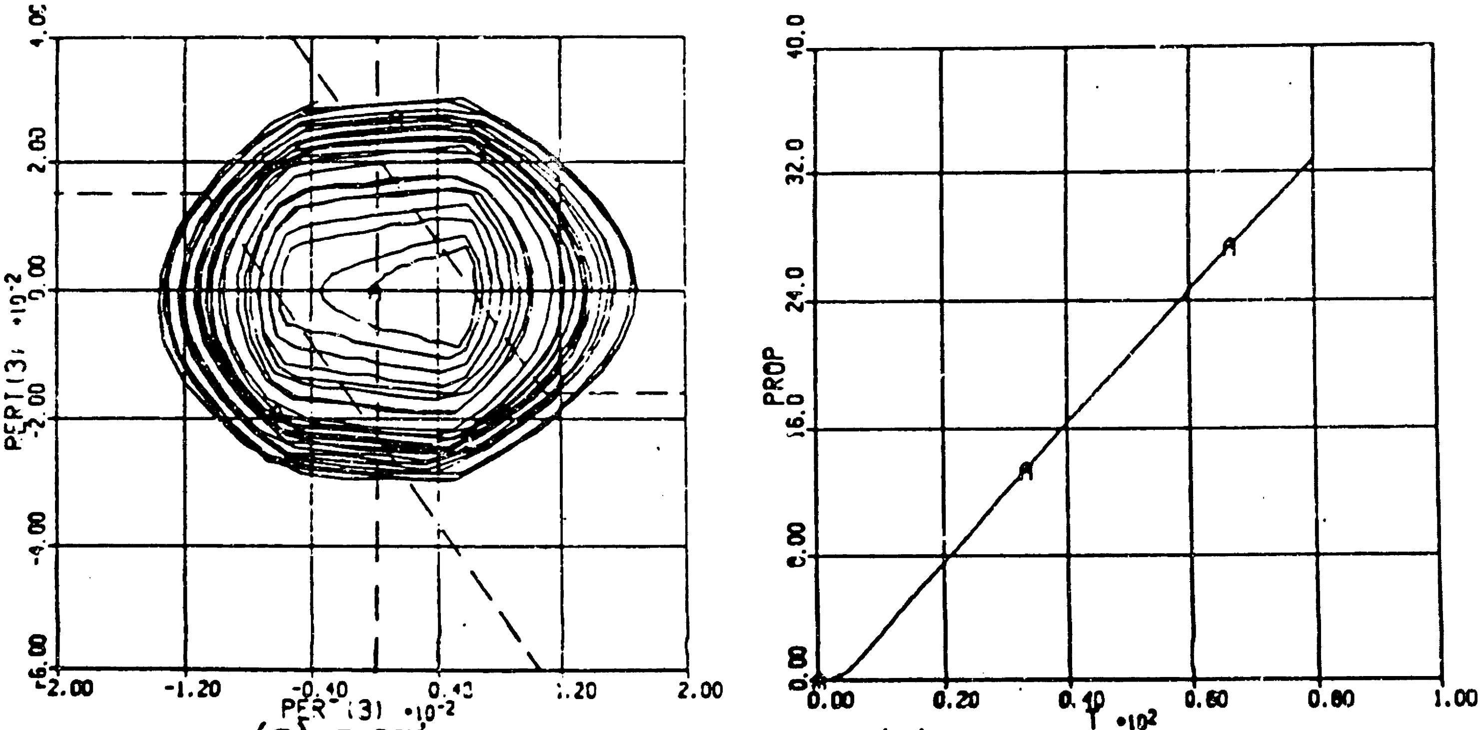
Table 5-X: Final RLC Normal Attitude Control Parameters

parameter	units	value	
		X&Y	Z
Φ_{db}	rad	9.0E-04	9.0E-04
Φ_r	rad	0.075061	0.056094
A	sec	7.5061	6.2327



(a) x-axis

(b) y-axis



(c) z-axis

(d) Propellant Consumed

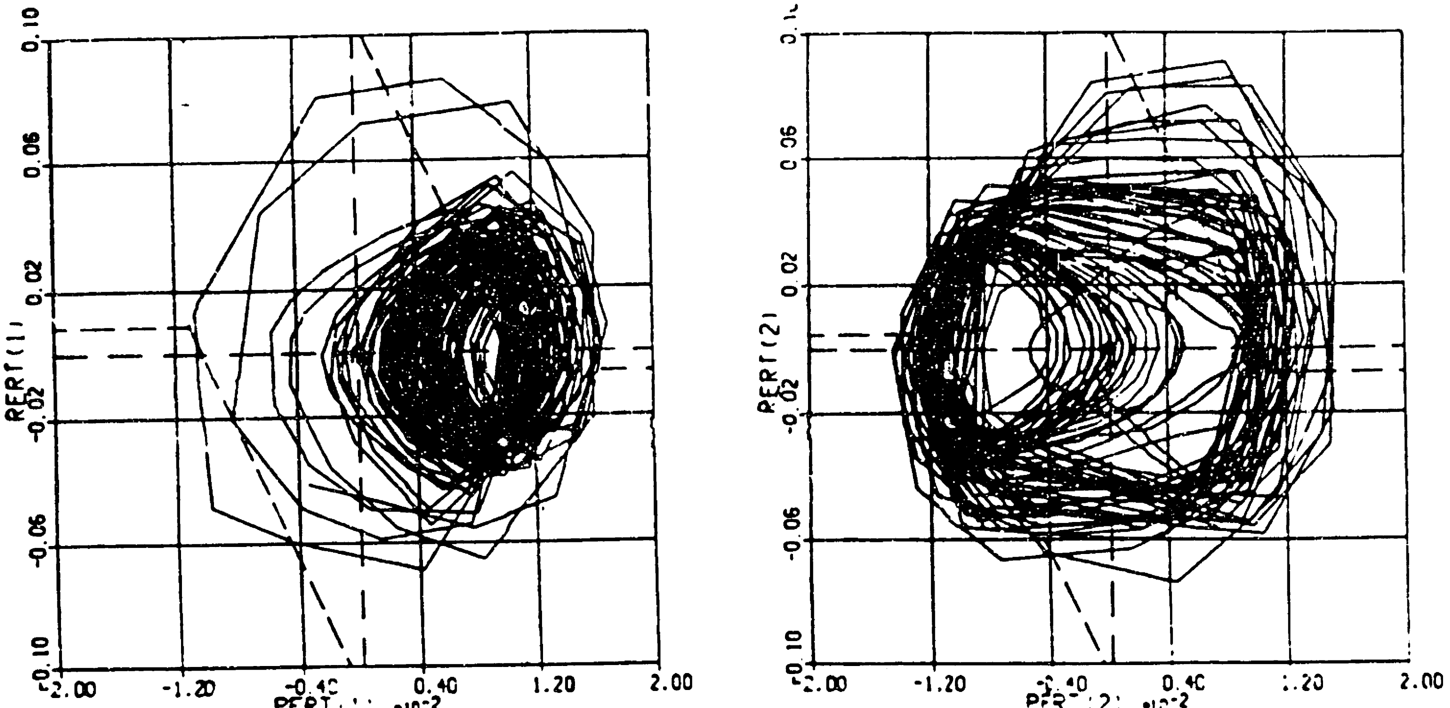
Figure 5-22: RLC VO1 Initial Simulation Run Results

Table 5-XI: Summary of RLC VOI Simulation Runs

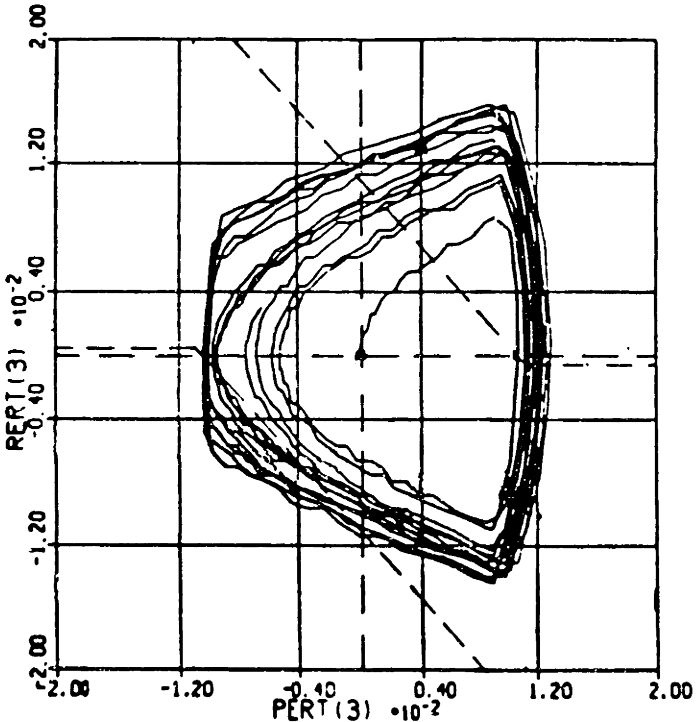
run	comments	maximum attitude (deg)			maximum rate (deg/sec)			fuel (kg)
		X	Y	Z	X	Y	Z	
1	analytically chosen values	1.28	1.21	0.967	7.07	6.59	1.72	32.81
2	ϕ_r=0.0108 / ϕ_{db}=0.01 A=0.0916	0.976	0.887	0.743	4.90	5.20	0.900	13.5
3	A=0.12	0.851	0.920	0.604	3.75	4.61	0.525	13.0
4	ϕ_r=ϕ_{db}=0.01	0.819	0.912	0.600	3.64	4.58	0.500	11.9
5	ϕ_r=ϕ_{db}=0.009	0.848	0.792	0.546	5.43	5.34	0.475	12.9
6	ϕ_r=ϕ_{db}=0.0095	0.779	0.812	0.583	5.34	4.37	0.500	12.3
7	A=0.15	0.693	0.602	0.551	3.15	3.91	0.475	11.8
8	A=0.18	0.656	0.554	0.549	3.45	3.46	0.475	11.5
9	A=0.21	0.633	0.540	0.535	2.90	3.02	0.450	11.1
10	A=0.30	0.539	0.544	0.521	3.62	3.92	0.425	12.0
11	A=0.27	0.559	0.474	0.529	2.90	2.67	0.450	11.4
12	A=0.24	0.566	0.515	0.531	4.19	3.20	0.450	11.4

The first parameter to be adjusted was ϕ_{db}. It was observed that when ϕ_{db} was widened to 0.01 rad, the recursive algorithm failed to converge. The algorithm was thus rapidly abandoned. Using the same values of ϕ_r and A that were used in Run 1, ϕ_{db} was widened to 0.01 rad. It was hoped that a wider deadband would cause one-sided deadband behavior. And the wider deadband *did* help reduce attitude and rate errors. Phase plane trajectories are shown in Figure 5-23. Although these trajectories are not one-sided over their entire histories, they are substantially more one-sided than before.

In the MGN phase plane, it was found that setting R1=0 helped VOI response a good deal. The analogous thing to try in the RLC is setting $\phi_r = \phi_{db}$, resulting in a phase plane as depicted in Figure 5-24. The Figure 5-24 phase plane looks very similar to the MGN phase plane with R1=0. This arrangement was tried in Run 4, with $\phi_r = \phi_{db} = 0.01$ rad. It was found to be of moderate help in improving system performance.



(a) x-axis (b) y-axis



(c) z-axis

Figure 5-23: VOI RLC Deadbands that are Mostly One-Sided

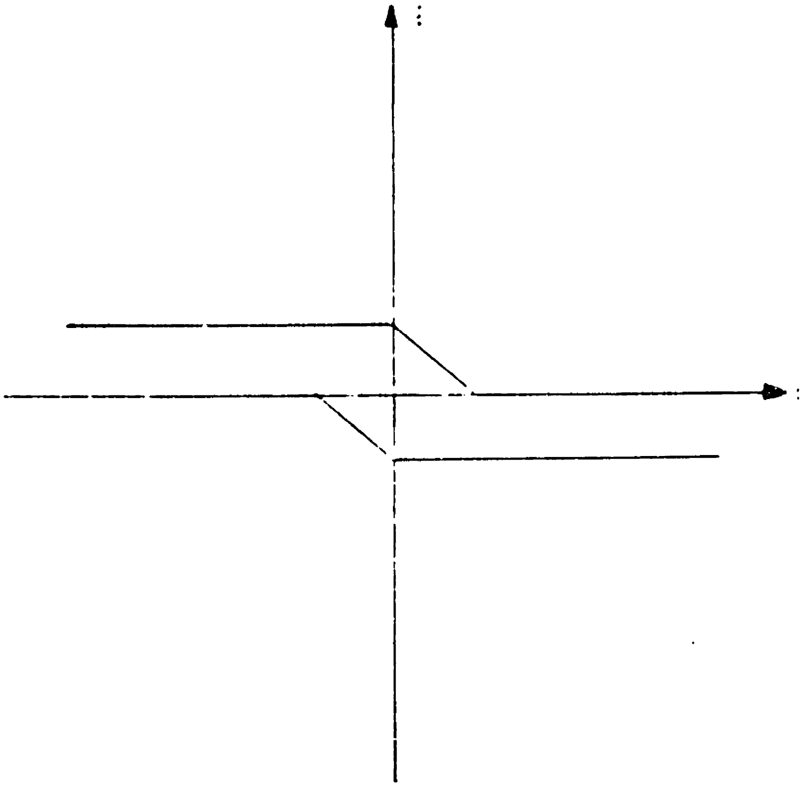


Figure 5-24: The RLC with $\phi_r = \phi_{db}$

ϕ_r and ϕ_{db} were thus adjusted in tandem. For values of these variables that were too high, the system was too permissive with errors. For low values, the system overreacted with excessive thruster firings, thus increasing rate errors. In Run 6, the values of ϕ_r and ϕ_{db} seemed to result in optimal rate errors.

With ϕ_r and ϕ_{db} adjusted appropriately, the only parameter left to tune was A. Although neither the Run 6 attitude errors nor the rate errors met the design criteria, the rate errors were much worse. Recall that, if the rate ledge itself is temporarily ignored, the RLC fires thrusters when $\phi + A\dot{\phi} > \phi_{db}$. A thus functions as a rate gain, increasing system sensitivity to rate errors. To gain better control over these rate errors, one must increase A.

Increasing A *was* found to reduce rate errors substantially, while simultaneously reducing attitude errors. By Run 7, attitude errors were reduced to within the desired limits. Excessively high A values, though, caused oversensitivity to rate errors. The optimal A was observed in Run 11, with A = 0.27 sec. Results of Run 11 are plotted in Figure 5-25, and the optimized parameter values used in this run are given in Table 5-XII.

Run 11 fuel consumption and attitude errors both meet the design criteria. x and y-axis rate errors, however, are unacceptably high -- much higher than in the MGN phase plane. This probably results from the fact that there are no short pulse regions in the RLC. As experience with the MGN showed, excessively strong thruster firings can cause large rate errors. With no short pulse regions, many of these excessive firings can occur.

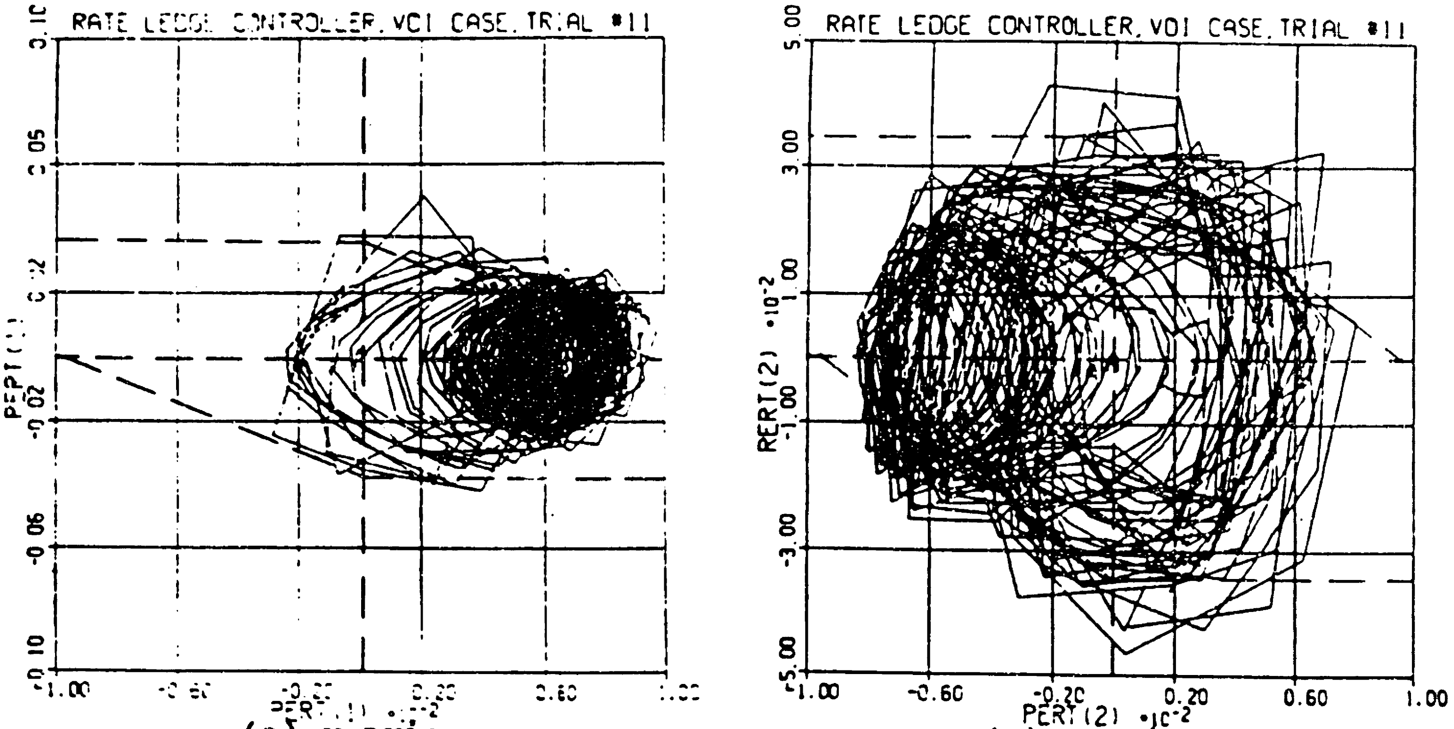
It is easy to choose parameter values for the RLC in VOI. Although the initially chosen values resulted in highly unacceptable performance, the tuning process was easy and geometrically intuitive. There were only three parameters to adjust (two if one considers that ϕ_r should equal ϕ_{db}). Although it was not exhaustively tested, the adjustment of a parameter did not seem to have an undue effect on the optimum values of other parameters. For ease of assigning parameter values, then, this system scores an 8.

Once the recursive RLC parameter design algorithm was abandoned, the RLC was easy to design in the VOI case. Although this system does a good job of reducing attitude errors and fuel consumption, rate errors are very large. This probably results from the lack of short pulse regions, which, by eliminating excessively strong thruster firings, can greatly reduce rate errors. RLC performance can probably be greatly enhanced by the addition of short pulse regions similar to those in the Magellan phase plane.

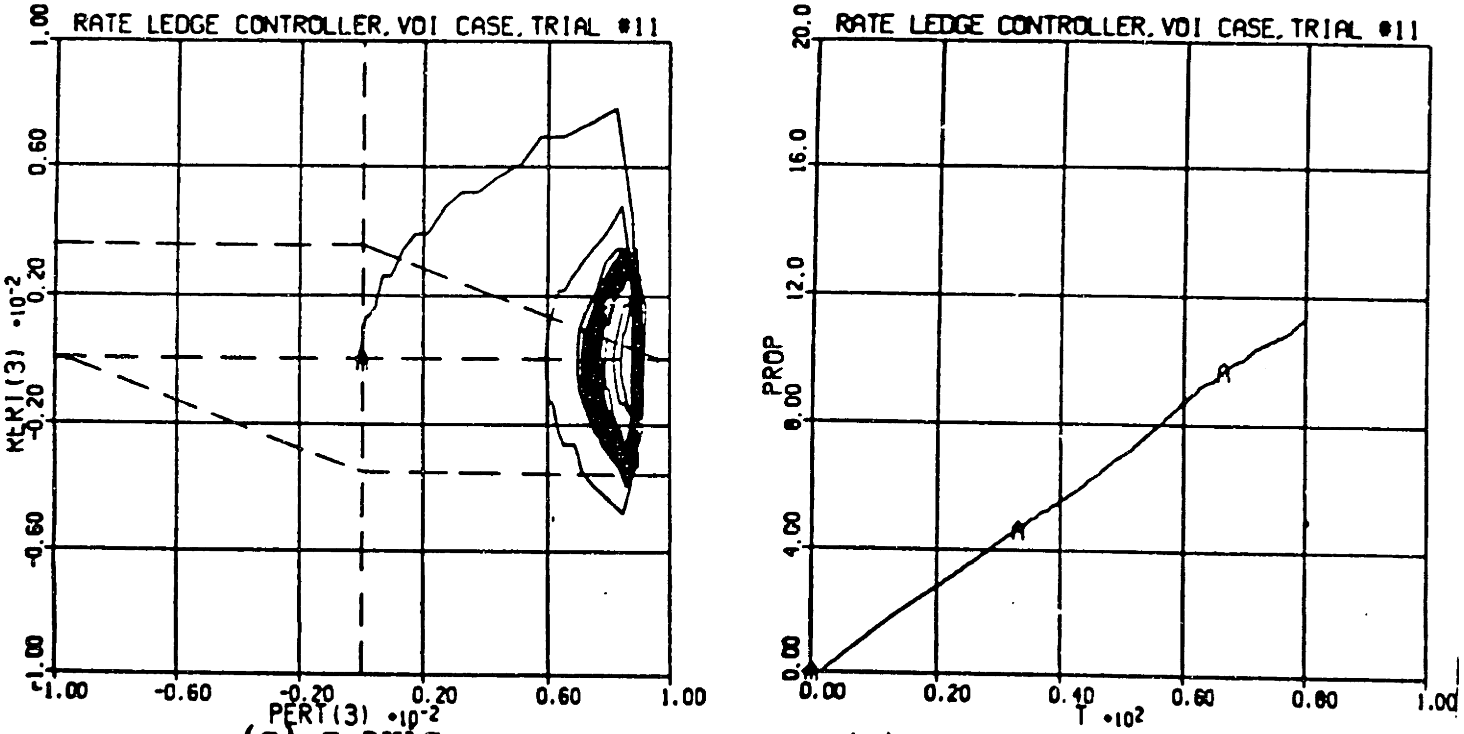
5.4 The RLC with Short Pulse Regions

5.4.1 Initial Analysis

The RLC with MGN-like short pulse regions is shown in Figure 5-26. Since the short pulse regions only exist near the origin of the phase plane, they will probably only affect steady state performance.



(a) x-axis (b) y-axis



(c) z-axis (d) Propellant Consumed

Figure 5-25: Optimized RLC VOI Trajectories

Table 5-XII: Optimized RLC VOI Parameter Values

parameter	units	value X & Y & Z
Φ_r	rad	0.0095
Φ_{db}	rad	0.0095
A	sec	0.27

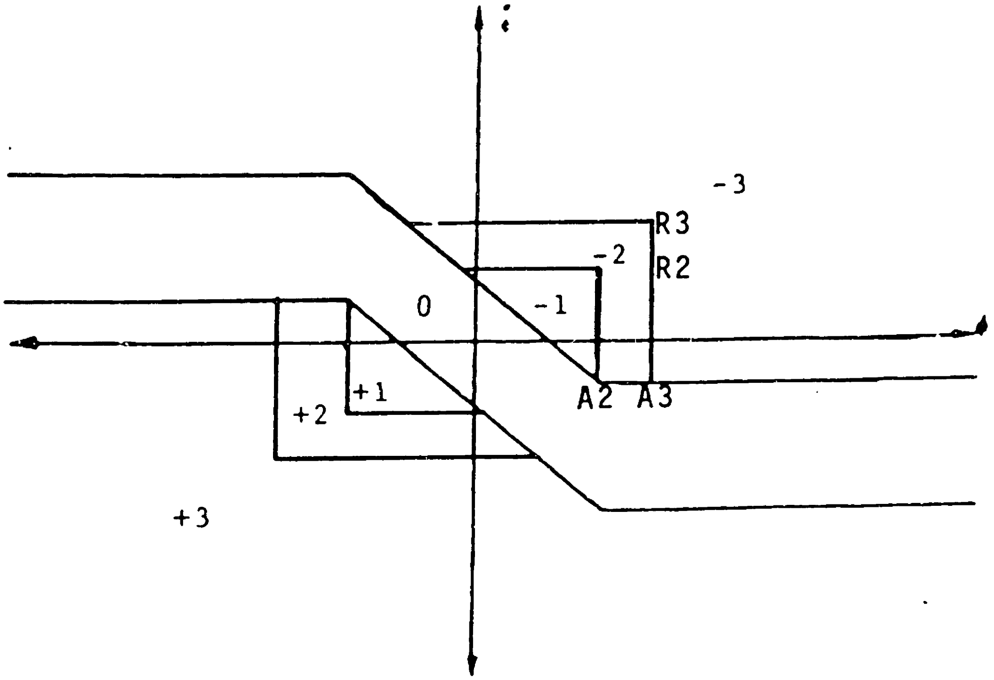


Figure 5-26: The RLC with Short Pulse Regions

The method of choosing parameter values is quite simple. The values of the parameters A and ϕ_r that were found to optimize the RLC itself will be used. ϕ_{db} will be set to the deadband width that optimized MGN phase plane performance (except in VOI, where ϕ_{db} will be set equal to ϕ_r). Rate and attitude ledges R2, R3, A2, and A3 will be set to MGN values. Hopefully, this will add the positive features of MGN steady state to the RLC.

In VOI, different values of the ledges A2, A3, R2, and R3 were used for different axes. But the optimized RLC parameters A, ϕ_r, and ϕ_{db} were the same for all three axes. Thus, for simplicity, the x and y-axis values of the ledges will be used for all three axes. If this results in bad system performance, parameter values will just have to be adjusted.

Initial parameter values are listed in Table 5-XIII.

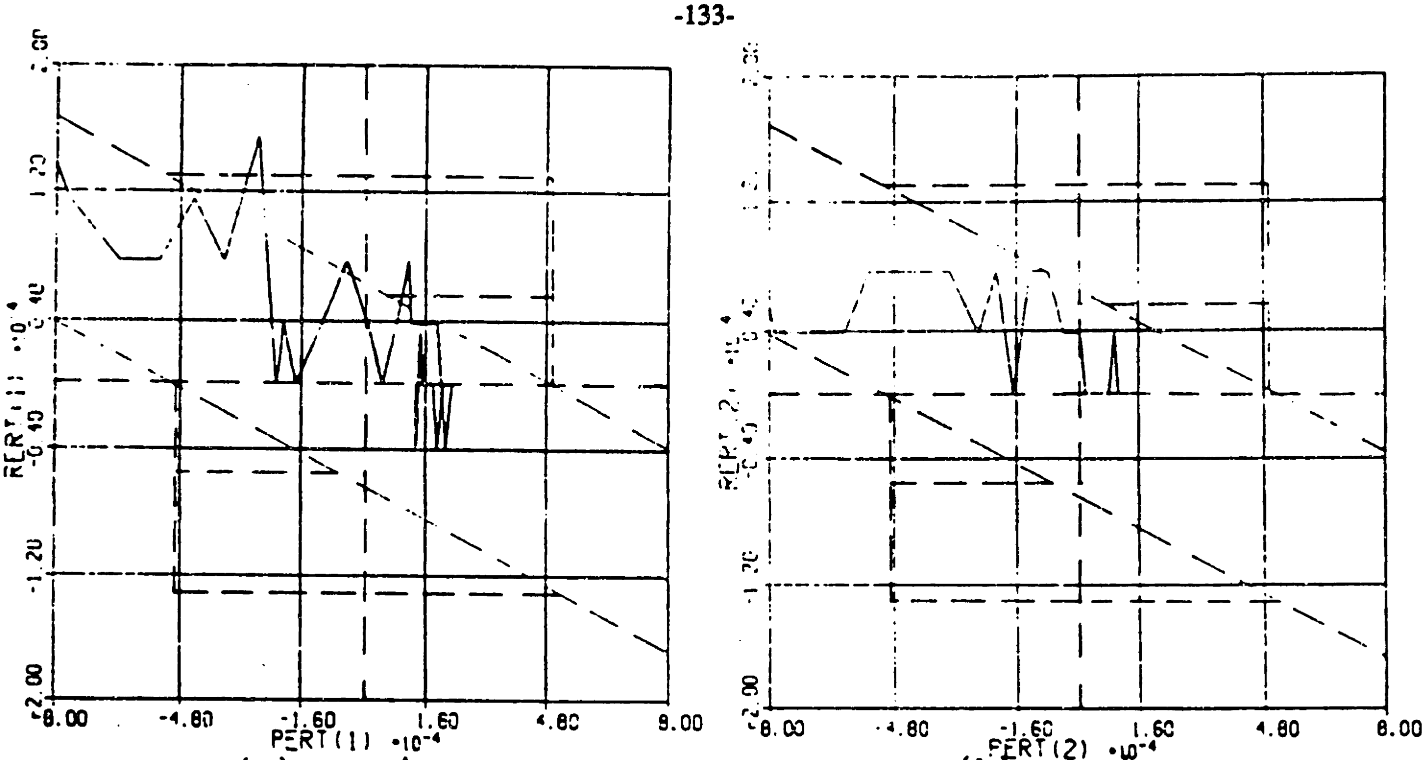
Table 5-XIII: Run 1 Parameters for RLC with Short Pulses

parameter	units	normal attitude control X & Y	normal attitude control Z	VOI X & Y & Z
Φ_r	rad	0.075061	0.056094	0.0095
Φ_{db}	rad	4.94E-04	4.94E-04	0.0095
A2	rad	4.9462E-04	4.9462E-04	0.01
A3	rad	4.9712E-04	4.9712E-04	0.012
R2	rad/sec	5.6E-05	5.6E-05	0.009
R3	rad/sec	1.3168E-04	1.3168E-04	0.12
A	sec	7.5061	6.2327	0.27

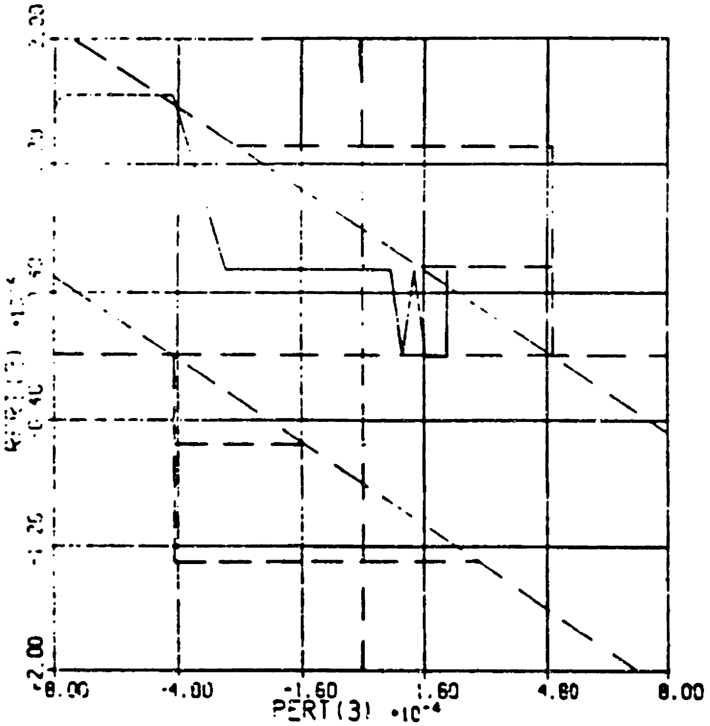
5.4.2 Normal Attitude Control

The parameter values derived above were found to result in exceptional steady state performance. Limit cycle attitude scores were greater than 9.6 and steady state fuel consumption scored a 10. Steady state phase plane plots are shown in Figure 5-27. This exceptional performance is paid for by a slight increase in the value of J.

A few simulations were run in an attempt to improve steady state performance even further. These runs are summarized in Table 5-XIV. By varying deadband widths, it was discovered that the RLC with short pulses is subject to the same "fuel bump" that was discovered in connection with the simple RLC. A plot of steady state fuel vs. deadband width is shown in Figure 5-28. Because this deadband is now flanked by short-duration region 1 thruster pulses, the "bump" occurs at a much narrower deadband than was observed before. Also, region 1 firings consume much less fuel then region 3 firings. Experience has shown that it is best to design the deadband value to rest in the valley before the "bump". The valley occurs when $\phi_{db} = 3\times10^{-4}$ rad (Run 6), and results in extremely good attitudes and fuel consumption. Plots of this case are depicted in Figure 5-29 and results are summarized in Table 5-XV. Table 5-XVI gives the parameter values used in this



(a) x-axis (b) y-axis



(c) z-axis

Figure 5-27: Initial RLC with Short Pulses Steady States run. The only disadvantage of this great performance is that, as discussed above, the narrower deadband causes increased transient fuel consumption, thus raising the value of J. There is a tradeoff between transient performance and steady state performance that makes it difficult to optimize both simultaneously.

Table 5-XIV: Summary of RLC with Short Pulses Runs

comments	settling time (sec) X	 Y	 Z	trans fuel	J	limit cycle attitude X	 Y	 Z	limit cycle rate X	 Y	 Z	ss fuel
analytically chosen values	69	85	72	3.55	342.4	9.84	9.63	9.83	9.98	9.98	9.99	10.0
ϕ_{db}=4.0E-04	69	85	72	3.51	343.9	9.73	9.87	9.88	9.99	9.99	9.99	10.0
ϕ_{db}=3.5E-04	69	85	72	3.47	345.1	9.79	9.95	9.96	9.99	10.0	9.99	10.0
ϕ_{db}=2.5E-04	--	--	--	----	356.0	----	----	----	----	----	----	0.30
ϕ_{db}=3.0E-04	--	--	--	----	-----	----	----	----	----	----	----	----
A=7.6734,6.1354 ϕ_r=0.076734,0.055219	75	85	69	3.39	340.9	9.96	9.92	9.97	10.0	10.0	9.99	10.0

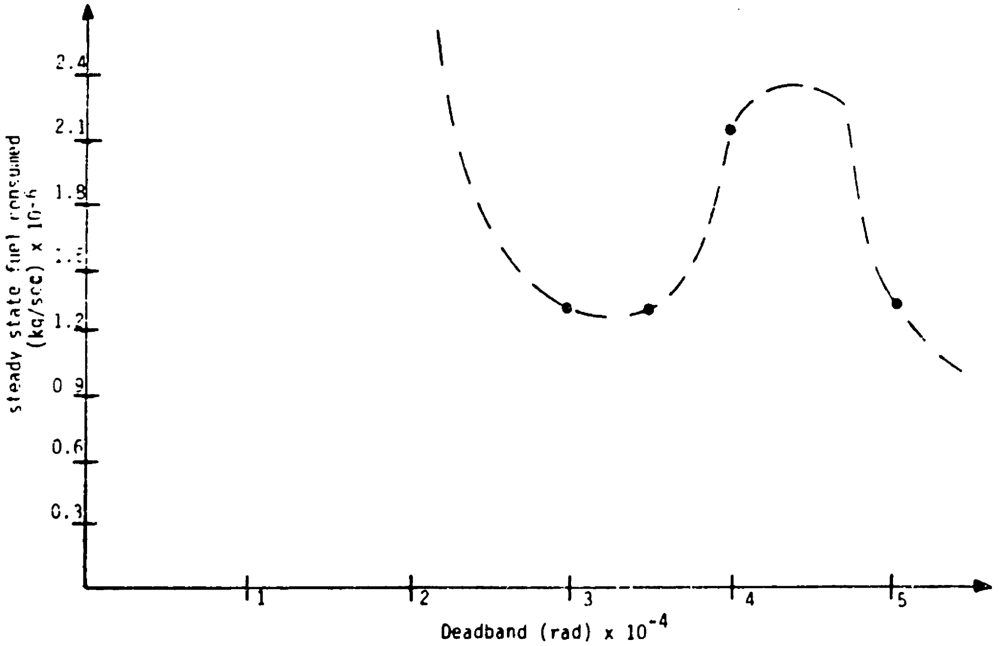
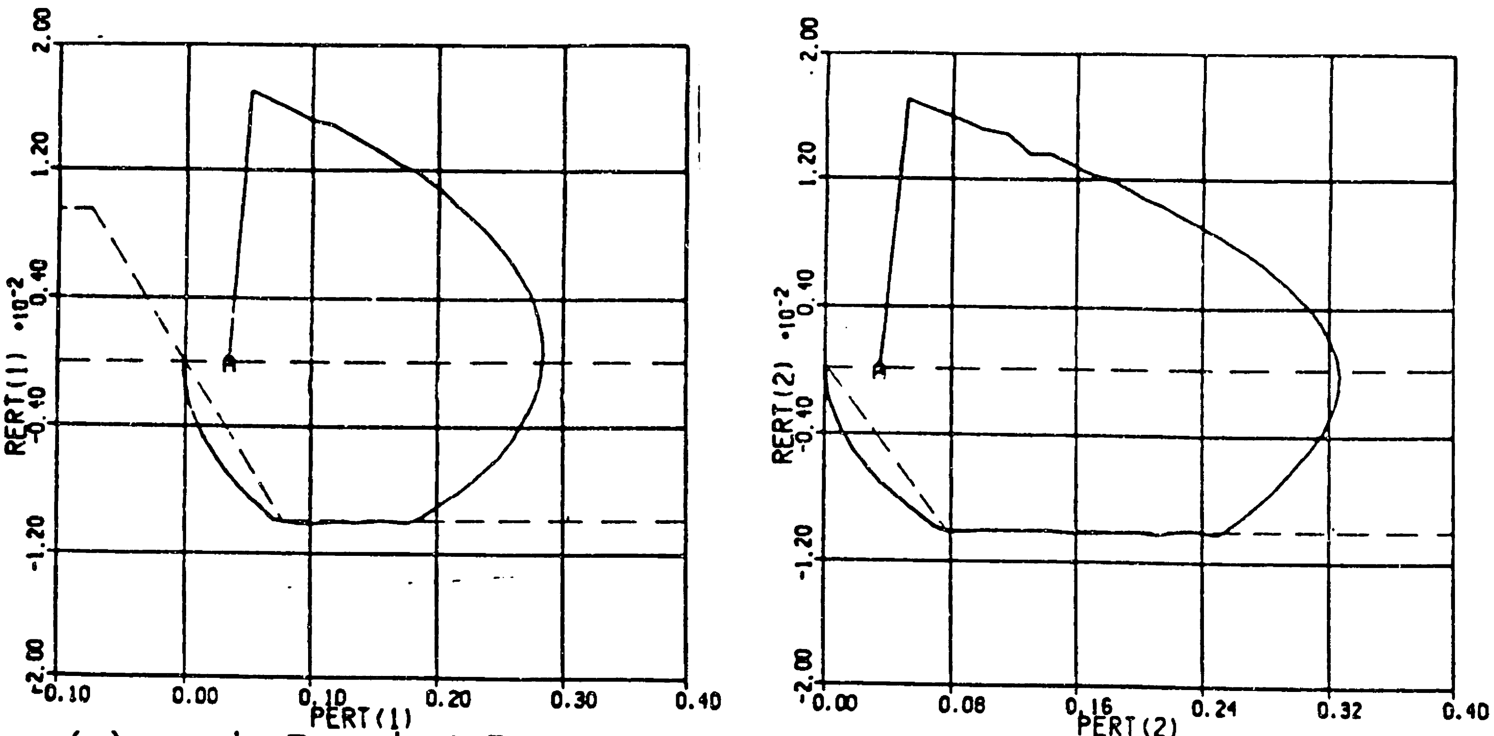


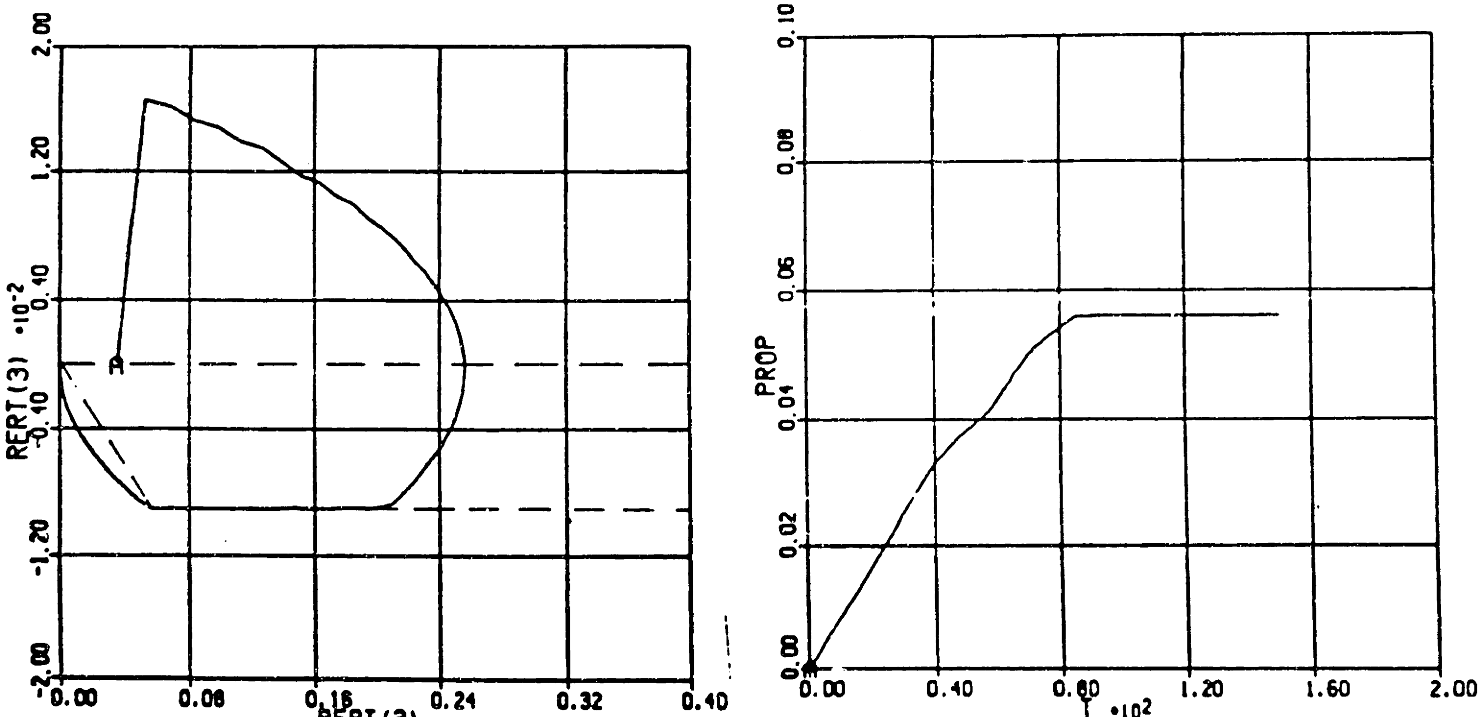
Figure 5-28: RLC with Short Pulses "Fuel Bump"

Given the optimized values of A and ϕ_r that were obtained in the previous section, choosing system parameters was very straightforward. Initially chosen parameters



(a) x-axis Transient Response

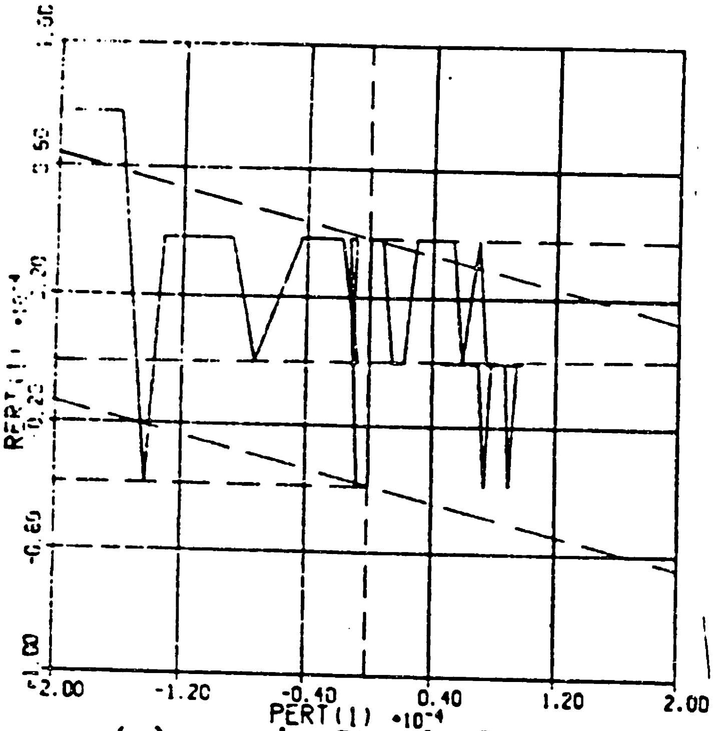
(b) y-axis Transient Response



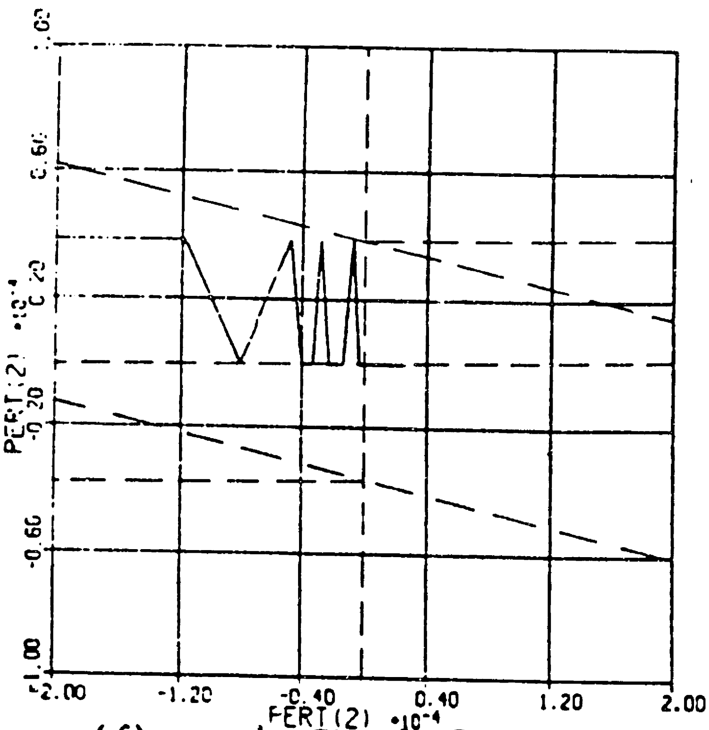
(c) z-axis Transient Response

(d) Propellant Consumed

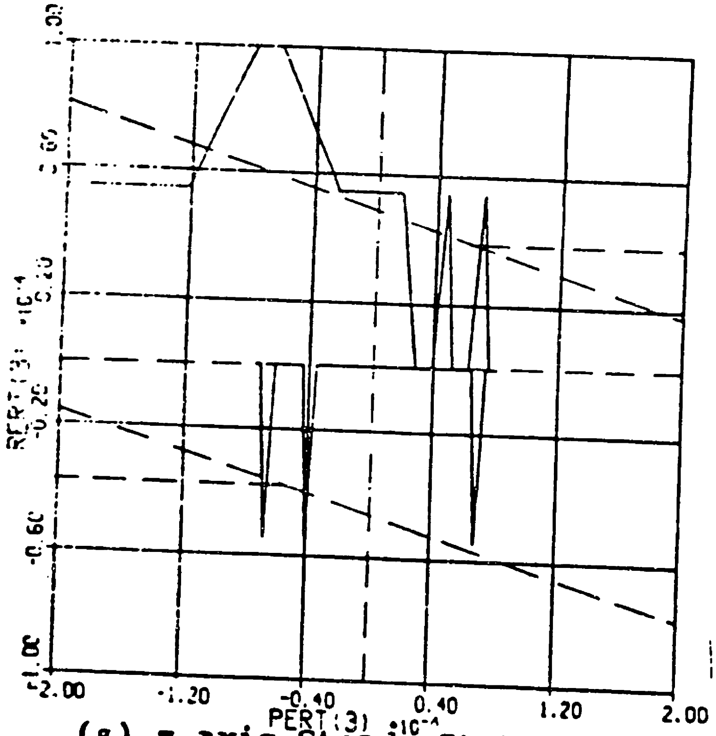
Figure 5-29: Optimized RLC with Short Pulses Phase Planes



(e) x-axis Steady State



(f) y-axis Steady State



(g) z-axis Steady State

Table 5-XV: Optimized RLC + Short Pulses Scores

Transient Scores

	X	Y	Z	AVG
settling time	8.00	7.06	8.70	7.92
fuel consumption				3.39
rate excursion	10.0	10.0	10.0	10.0
attitude excursion	3.41	1.25	4.68	3.11

J = 350.9

Steady State Scores

	X	Y	Z	AVG
fuel consumption				10.0
limit cycle rate	10.0	10.0	9.99	9.99
limit cycle attitude	9.96	9.92	9.97	9.95

Overall

	X	Y	Z	AVG
ease of assigning parameter values				8

improved steady state performance substantially, and the only parameter that needed to be adjusted was ϕ_{db}. One danger, though, is that tuning steady state can affect transient response. In Run 5, an adjustment of ϕ_{db} was found to cause overshoot on the z-axis. When k was adjusted to correct this, very good steady state response was observed. For ease of assigning parameter values , then, this system rates an 8.

Table 5-XVI: RLC + Short Pulses Optimized Parameters

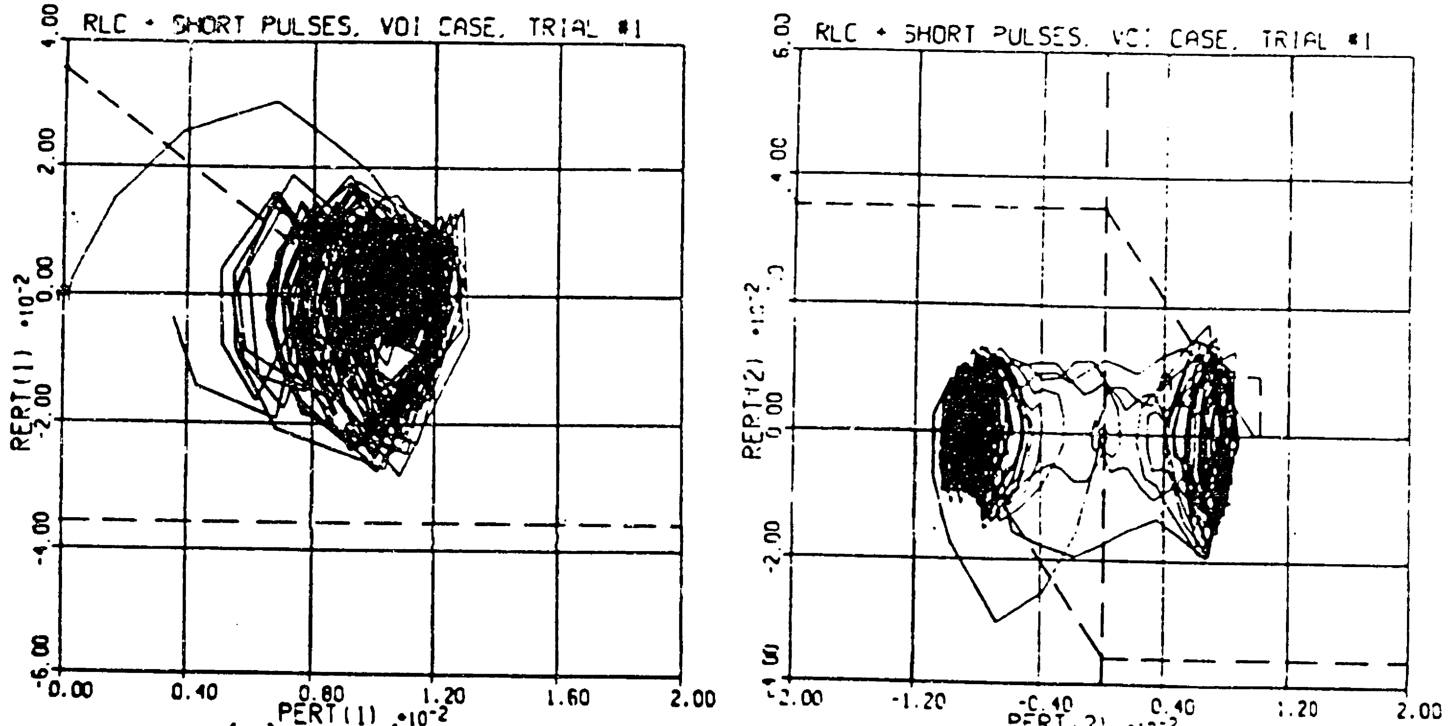
parameter	units	value	
		X & Y	Z
Φ_r	rad	0.076734	0.055219
Φ_{db}	rad	3.0E-04	3.0E-04
A2	rad	4.9462E-04	4.9462E-04
A3	rad	4.9712E-04	4.9712E-04
R2	rad/sec	5.6E-05	5.6E-05
R3	rad/sec	1.3168E-04	1.3168E-04
A	sec	7.6734	6.1354

So adding short pulse regions to the RLC substantially improves steady state performance. Transient response is adversely affected, but this is not that important -- as discussed in the last section, initial conditions far from those for which the system was optimized would be expected to exhibit worsened system performance anyway. The RLC with short pulse regions also scores better for limit cycle rates and attitudes than the MGN phase plane in steady state.

5.4.3 VOI

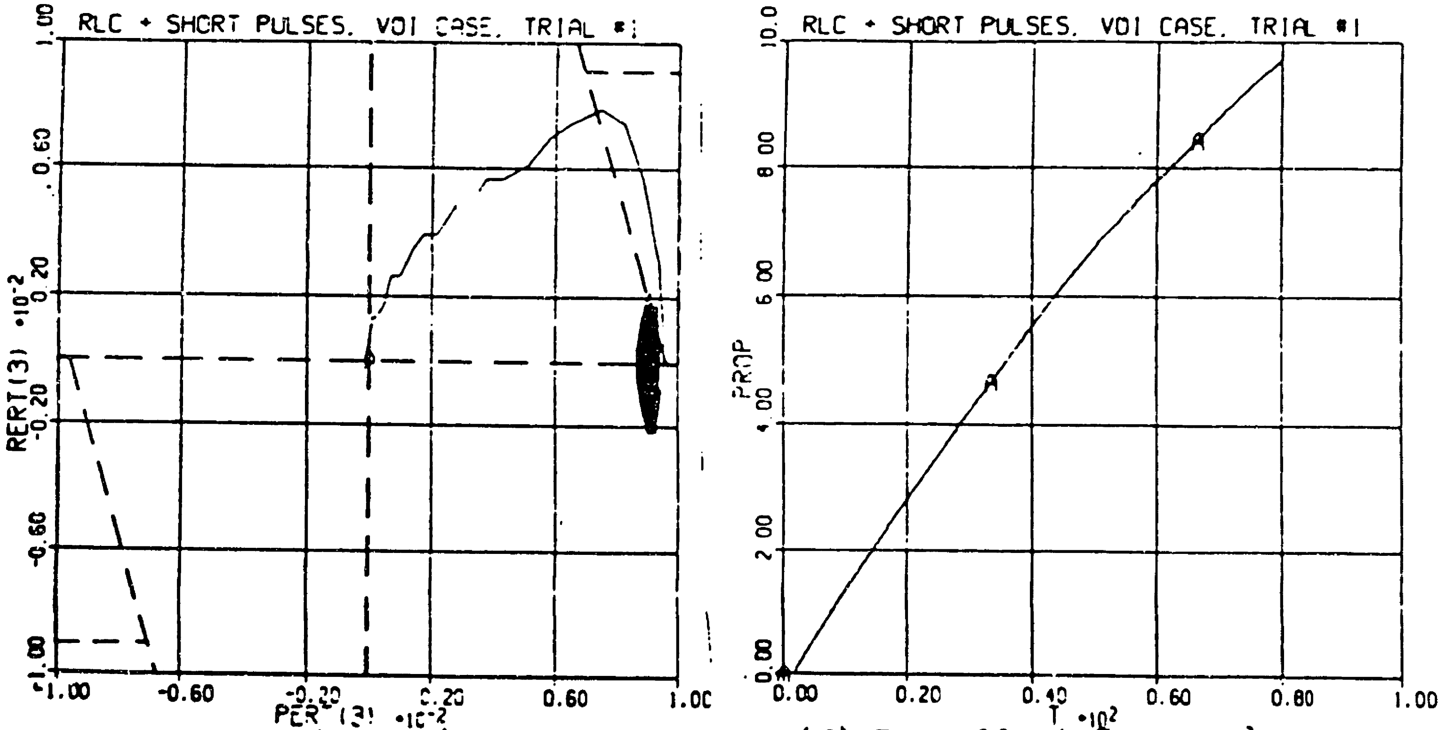
The simulation results obtained with initial parameter choices are shown in Figure 5-30. The addition of short pulse regions to the RLC has slightly worsened attitude errors, but greatly improved rate errors. Weak region 1 firings can cause only very slow rate changes, thus allowing larger attitude errors tc build up. Since the attitude errors are still within acceptable limits, though, the short pulse regions seem to be a positive contribution to the RLC.

The phase plane trajectories in Figure 5-30 are similar to VOI trajectories obtained with the MGN phase plane (see, for example, Figure 5-11). With short pulse regions added to the RLC, the two phase planes look very similar in VOI (see Figure 5-31). Thus, they



(a) x-axis

(b) y-axis



(c) z-axis

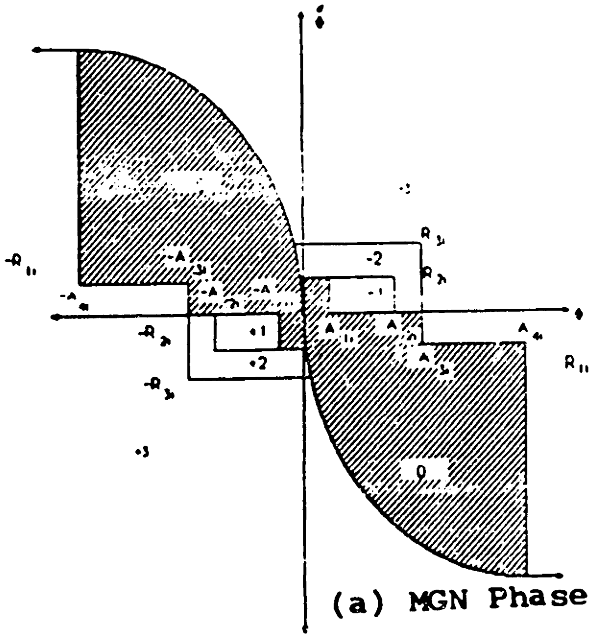
(d) Propellant Consumed

Figure 5-30: RLC + Short Pulses Initial VOI Results *should* behave similarly. But the MGN phase plane was unable to meet rate error requirements. It is thus doubtful that the RLC with short pulses will be able to do so.

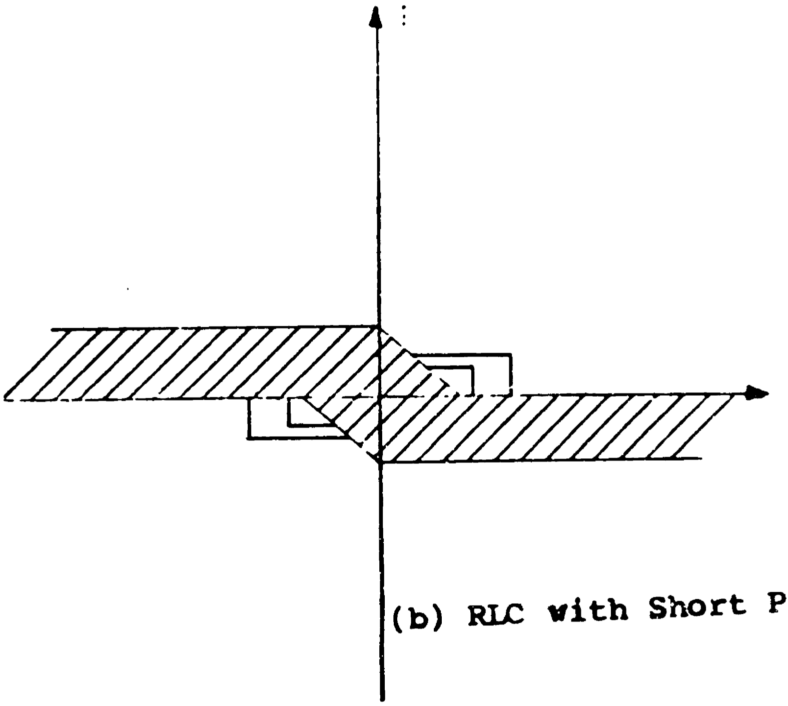
The parameter tuning process is illustrated in Table 5-XVII. It was desirable to keep ϕ_{db} equal to ϕ_r so that excessive region 3 firings could be prevented, and to avoid increasing *A3* in order to keep attitude errors within acceptable bounds. Within these constraints, parameters were adjusted individually. Although rate errors were never reduced to less than the desired maximum of 1.14 deg/sec, Run 2 demonstrated the best performance. In this run, all results met the design criteria except the x and y-axis rates. The maximum x-axis rate was 1.48 deg/sec. The maximum y-axis rate was 1.45 deg/sec, but this was only for the first few seconds of VOI. After this, y-axis rates actually met the criteria. Simulation results are shown in Figure 5-32, and the optimized parameter values are summarized in Table 5-XVIII.

The fact that the second run resulted in optimal performance indicates that the initially chosen values were already close to optimal. Given predesigned values of A and ϕ_r, it took only minimal adjustment to optimize performance. However, there was no way of ensuring that this run had actually been optimal short of trying variations in all the parameters. And there are many parameters to adjust. In the VOI case, the ease of assigning parameter values score for this system is a 7.

Thus, the RLC with short pulse regions is easy to choose parameters for and performs very well in the VOI case. As phase plane trajectories show, it acts very much like the MGN phase plane, because its switching lines *are* very much like those in the MGN phase plane. In the VOI case, where parabolic switching lines are not so important, the two systems give very similar results.



(a) MGN Phase Plane



(b) RLC with Short Pulses

Figure 5-31: MGN and RLC + Short Pulses, Similar in VOI

Table 5-XVII: RLC + Short Pulses VOI Simulations

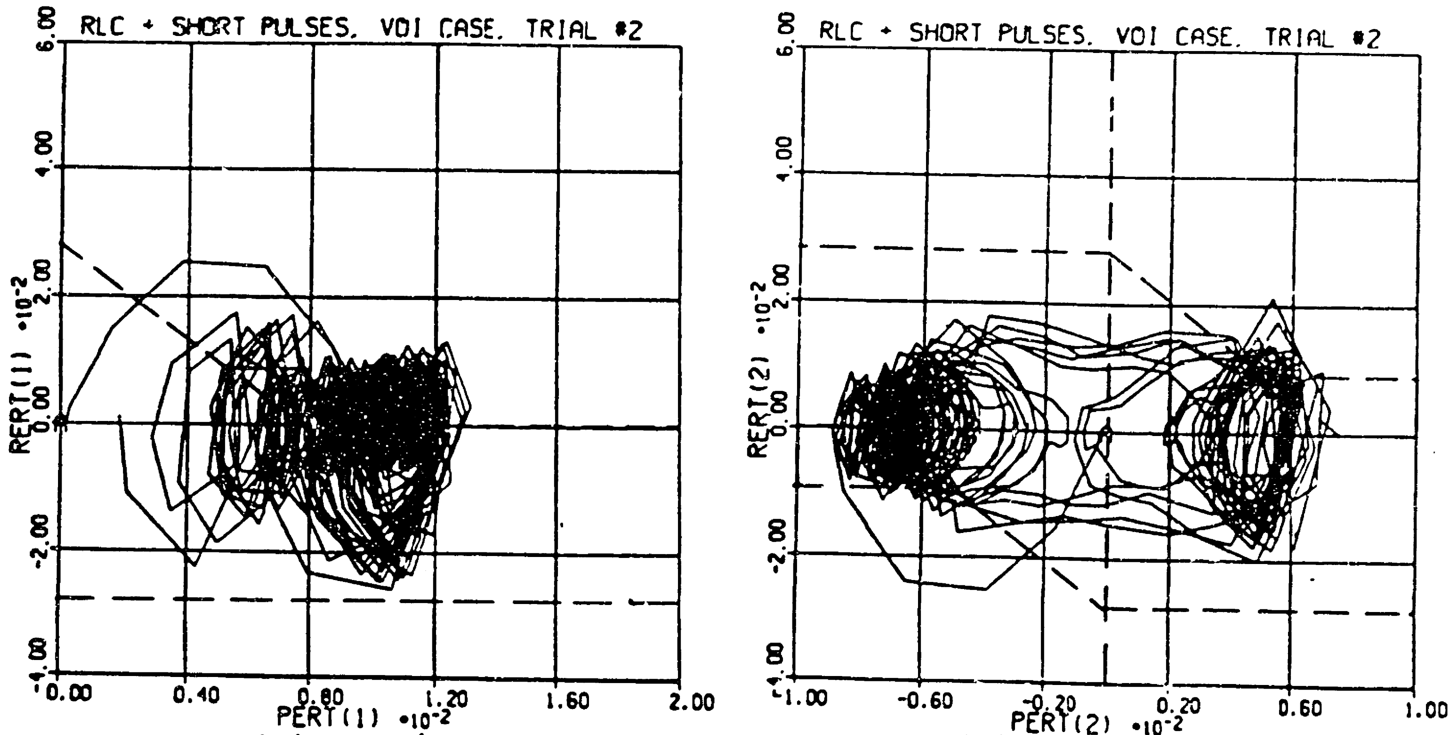
run	comments	maximum attitude (deg) X	Y	Z	maximum rate (deg/sec) X	Y	Z	fuel (kg)
1	analytically chosen values	0.753	0.620	0.547	1.73	1.73	0.450	9.72
2	$\phi_{db}=\phi_r=0.0075$	0.752	0.508	0.428	1.48	1.45	0.400	9.78
3	$\phi_{db}=\phi_r=0.006$	0.748	0.401	0.340	1.54	1.38	0.350	9.88
4	$\phi_{db}=\phi_r=0.0075$ A=0.30	0.754	0.478	0.426	1.64	1.38	0.400	9.78
5	A=0.24	0.737	0.508	0.443	1.52	1.45	0.400	9.76
6	A=0.27 A2=0.009	0.736	0.508	0.444	1.66	1.45	0.400	9.71
7	A2=0.011	0.745	0.508	0.443	1.59	1.45	0.400	9.69
8	A2=0.01 R2=0.0075	0.740	0.508	0.444	1.70	1.45	0.400	9.73
9	R2=0.01	0.752	0.508	0.428	1.52	1.45	0.400	9.74
10	run #2 values A3=0.011	0.694	0.508	0.428	1.71	1.45	0.400	9.77

5.5 Generalizations About Phase Plane Designs

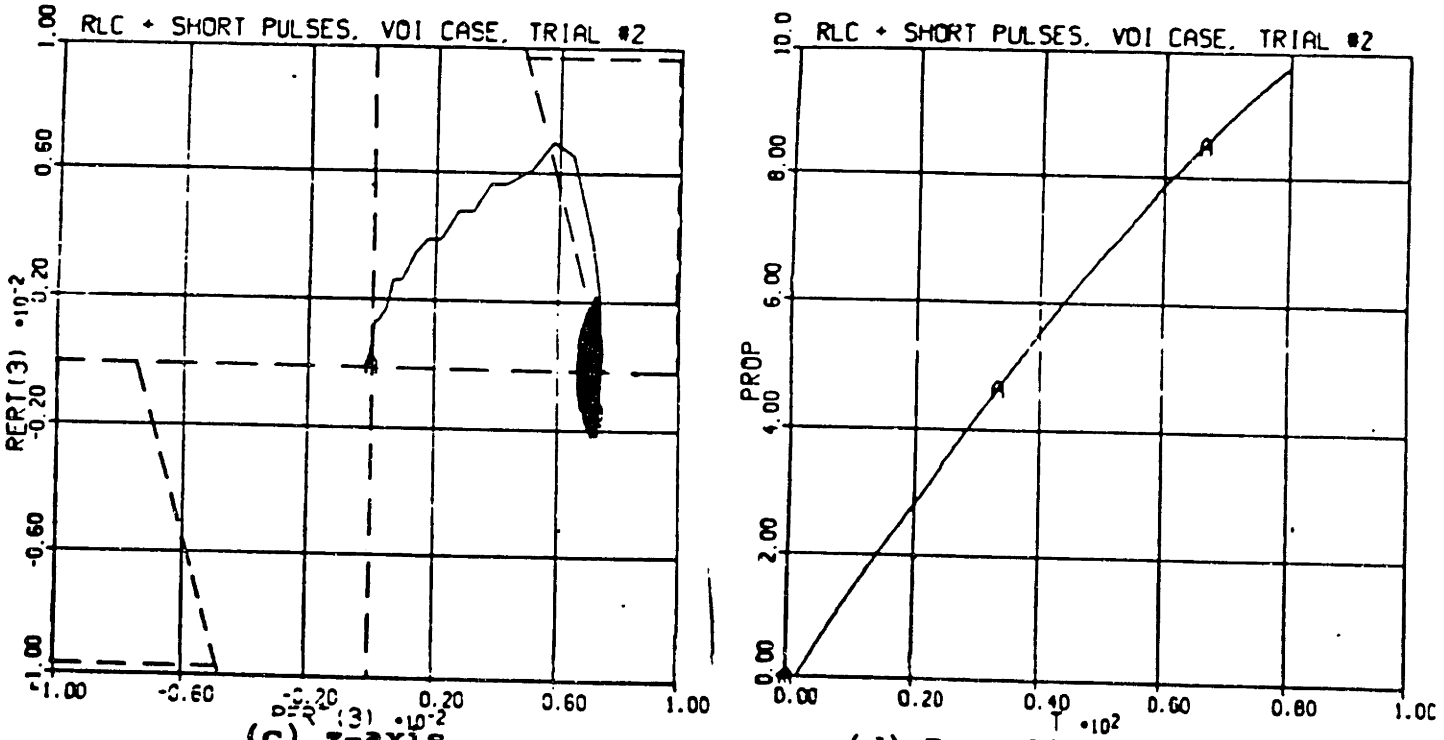
Thruster attitude control system designs based on the phase plane approach seem to perform reasonably well and to be rather easy to choose parameter values for. Some phase plane designs, of course, are better than others, and making sweeping generalizations based only on the three designs considered in this chapter would be dangerous. There are, however, some generalizations that seem safe.

Phase plane designs perform quite well. Using them, one can obtain J values as low as 340, with very high ratings for limit cycle attitudes and steady state fuel consumption. In a phase plane design, it is easy to include lots of rate and attitude ledges and specialized regions to improve performance in specific locations in the phase plane.

Phase planes that take advantage of this, such as the MGN and the RLC with short pulse regions, can perform quite well in both transient response and steady state.



(a) x-axis (b) y-axis



(c) z-axis (d) Propellant Consumed

Figure 5-32: Optimal RLC + Short Pulses VOI Trajectories

Table 5-XVIII: Optimized RLC + Short Pulses Parameters

parameter	units	value X & Y & Z
Φ_r	rad	0.0075
Φ_{db}	rad	0.0075
A2	rad	0.01
A3	rad	0.012
R2	rad/sec	0.009
R3	rad/sec	0.12
A	sec	0.27

Unfortunately, because of all their rate and attitude ledges, these designs also tend to have a large number of parameters -- the MGN and the RLC with short pulses have 8 and 7 respectively. Although this complicates the design process, geometric intuitions about the system greatly facilitate parameter optimization. Since phase plane switchings can be easily sketched, system performance is easy to visualize.

The best phase plane design considered in this chapter is the MGN phase plane. In the normal attitude control case, the importance of an "approach parabola" makes a design with a parabolic switching curve highly desirable. The MGN is only slightly worse than the RLC with short pulses in steady state, and the RLC's transient behavior is unreliable; different initial conditions can cause very different system response. In VOI, disturbances make well-behaved parabolic trajectories unimportant, so the MGN and the RLC with short pulses perform similarly. Overall, the MGN phase plane seems to perform best.

The MGN may compete well with other phase plane designs, but how does it compare to radically different approaches to the design of thruster attitude control systems? This question will be answered as an attitude control system design based on the controller approach is examined.

Chapter 6

A Controller Design: The Galileo Controller

6.1 Initial Analysis

As suggested in Chapter 3, the Galileo controller can be designed using a traditional describing function analysis. A simplified version of the system, neglecting time delays and all other nonlinearities except the phase plane, is shown in Figure 6-1. The pseudo-rate term is treated here as if it were proportional to the vehicle angular acceleration. In fact, a pseudo-rate term *can* be viewed as if it were acceleration passed through a low pass filter. The first step in this analysis must be to find the describing function of the nonlinear block. Then, parameter values must be chosen. Finally, the system must be transformed into the discrete domain.

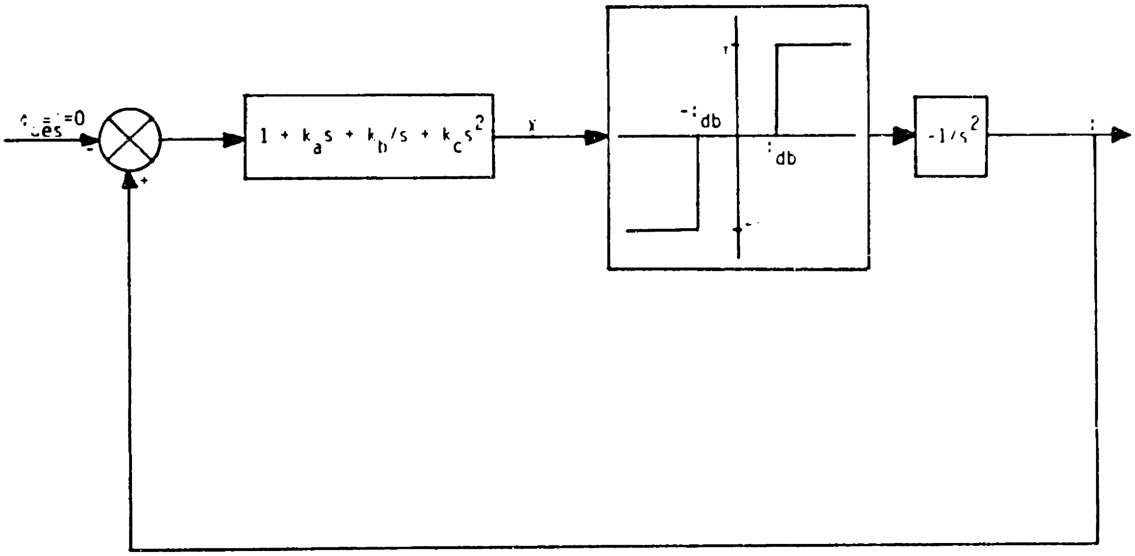
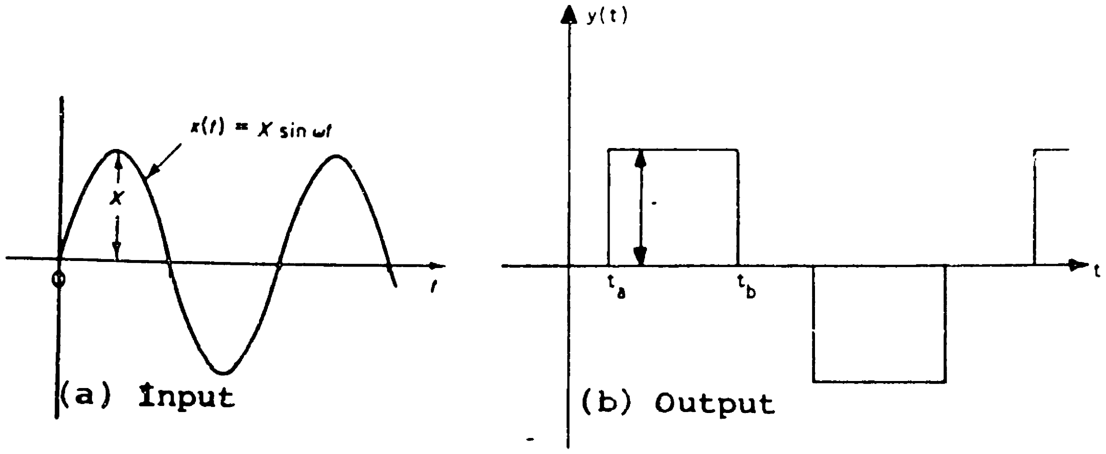


Figure 6-1: Simplified Galileo Controller for Analysis

To find the describing function, an input sinusoid of amplitude X is fed into the nonlinear block, as shown in Figure 6-2a. The output, y(t), of this block is shown in Figure 6-2b. The first harmonic of y(t) is given by the equation:



(a) Input

(b) Output

Figure 6-2: Input and Output of the Deadband Block

$$Y_1 = \frac{2}{\pi}\int_0^{\pi} y(t)\sin(\omega t)\,d(\omega t) \tag{6.1}$$

where Y_1 is the first harmonic. Because of the shape of the curve y(t), this equation can be rewritten as:

$$Y_1 = \frac{2\tau}{\pi}\int_{\omega t_a}^{\omega t_b} \sin(\omega t)\,d(\omega t) \tag{6.2}$$

where t_a and t_b are as shown in Figure 6-2. y(t) transitions from 0 to $+\tau$ when:

$$X\sin\omega t = \phi_{db} \tag{6.3}$$

or

$$\omega t = \omega t_a = \sin^{-1}\frac{\phi_{db}}{X} \tag{6.4}$$

Thus, since y(t)'s first pulse is symmetric about $\omega t = \pi/2$, equation (6.2) becomes:

$$Y_1 = \frac{4\tau}{\pi}\int_{\sin^{-1}(\phi_{db}/X)}^{\pi/2} \sin(\omega t)\,d(\omega t) \tag{6.5}$$

which can be simplified to:

$$Y_1 = \frac{4\tau}{\pi}\sqrt{1-\left[\frac{\phi_{db}}{X}\right]^2} \tag{6.6}$$

The describing function, N, is defined to be Y_1/X. Thus,

$$N = \frac{4\tau}{\pi X}\sqrt{1-\left[\frac{\phi_{db}}{X}\right]^2} \tag{6.7}$$

With the function N, the system shown in Figure 6-1 can be analyzed. Following the procedure outlined in Ogata [17], the transfer function G(s) is defined as:

$$G(s)=-\frac{1}{s^2}(1+k_a s+\frac{k_b}{s}+k_c s^2) \tag{6.8}$$

or

$$G(j\omega)=\left[\frac{1-k_c\omega^2}{\omega^2}\right]+\left[\frac{k_a\omega^2-k_b}{\omega^3}\right]j \tag{6.9}$$

Values of X and ω must be found such that G(jω) = -1/N. These values represent the approximate limit cycle amplitudes and frequencies that will be observed in the system.

Since N is a real-valued function, the value of G(jω) must also be real. Thus, the imaginary term in equation (6.9) must equal 0. Therefore:

$$\frac{k_a\omega^2-k_b}{\omega^3}=0 \tag{6.10}$$

or

$$\omega=\sqrt{\frac{k_b}{k_a}} \tag{6.11}$$

Since N = -1/G(jω), one can find:

$$\frac{4\tau}{\pi X}\sqrt{1-\left[\frac{\phi_{db}}{X}\right]^2}=\frac{\omega^2}{1-k_c\omega^2} \tag{6.12}$$

or, after rearranging:

$$\pi^2X^4-16\tau^2\left[\frac{1-k_c\omega^2}{\omega^2}\right]X+16\tau^2\phi_{db}{}^2\left[\frac{1-k_c\omega^2}{\omega^2}\right]=0 \tag{6.13}$$

A limit cycle amplitude of 0 is desirable. Equation (6.13) shows that this occurs when:

$$k_c=\frac{1}{\omega^2}=\frac{k_a}{k_b} \tag{6.14}$$

Using the derivation performed above, the process of choosing parameter values is straightforward. Start with a value for k_a that is identical to the optimum rate gain, A, that was found in connection with the RLC. Then, with the choice of an arbitrary ω, equation (6.11) gives a value for k_b. Finally, k_c is calculated from equation (6.14). Pseudo-rate charging and discharging time constants (see section 3.2) are ignored in the above derivations, so they will be set to an arbitrary, but small, value.

These parameter values, derived in continuous time, must be converted into discrete time equivalents. These conversions are addressed individually below:

- The simulated discrete time rate gain, k_6, is simply equal to the continuous time rate gain, k_a.
- The integral term is simulated as:

  $$\text{output} = k_7 x\ (\text{input}) + I_{old} \tag{6.15}$$

  where I_{old} is the previous value of the output. If $k_7 = k_b x$ (the sample time), this term will function as an approximate integral term.
- The pseudo-rate term is simulated as:

  $$\text{output} = k_{13} x\ u - k_{12} P_{old} \tag{6.16}$$

  where P_{old} is the previous value of the output and u = +1, 0, or -1. In continuous time, the pseudo-rate term can be written as:

  $$\frac{output}{u} = \frac{k_c a}{s+a} \tag{6.17}$$

  where $a = 1/t_c = 1/(\text{the time constant})$. Using a continuous to discrete conversion similar to the one used to derive filter gains in Chapter 5, it can be shown that:

  $$k_{12} = -e^{-aT} \tag{6.18}$$

  $$k_{13} = k_d(1 - e^{-aT}) \tag{6.19}$$

  where T is the sampling time and $k_d = k_c x\ \tau$, since u = +1, 0, or -1, and not $+\tau$, 0, or $-\tau$

Initially chosen parameter values are listed in Table 6-I. Note that the describing function analysis was not strictly followed in assigning parameters for VOI. A VOI limit cycle frequency of 0 is desirable, implying (equation (6.14)) an infinite k_d value. However, a more intuitive argument was used to set the VOI pseudo-rate gain. The pseudo-rate term provides an estimate of vehicle rates. It does this by anticipating response to a thruster firing. But in VOI, disturbance torques also contribute to body rates, and the pseudo-rate term has no information about these. Thus, this term initially seems to be of little use in VOI, and shall be set to zero.

Finally, a word is in order about the low pass filter used in this case. While studying phase plane designs, it was desirable to filter rate errors in the normal attitude control case. Now, however, the controller has only one output, which represents a combination of rate and attitude errors. This single output will be filtered, using the same filter coefficients derived in the previous chapter.

Table 6-I: Initial Parameters for Galileo System

parameter	units	normal attitude control		VOI
		X & Y	Z	X & Y & Z
ω	rad/sec	8.0E-03	8.0E-03	0.00
$k_a = k_6$	sec	7.7	6.1	0.27
k_b	1/sec	4.93E-04	3.9E-04	0.00
$k_d = \tau/\omega^2$	sec^2	17.917	24.952	0.00
$t_{c\text{-}charging}=1/a_{charging}$	msec	33.0	33.0	0.00
$t_{c\text{-}discharg}=1/a_{discharg}$	msec	33.0	33.0	0.00
k12	----	-0.01832	-0.01832	0.00
k13	----	17.589	24.495	0.00
Φ_{db}	rad	4.94E-04	4.94E-04	0.006

6.2 Normal Attitude Control

The results of a simulation run with the parameters derived above are plotted in Figure 6-3. Performance is probably the worst observed so far. Fuel consumption is very high, and attitude and rate errors are extremely large[8].

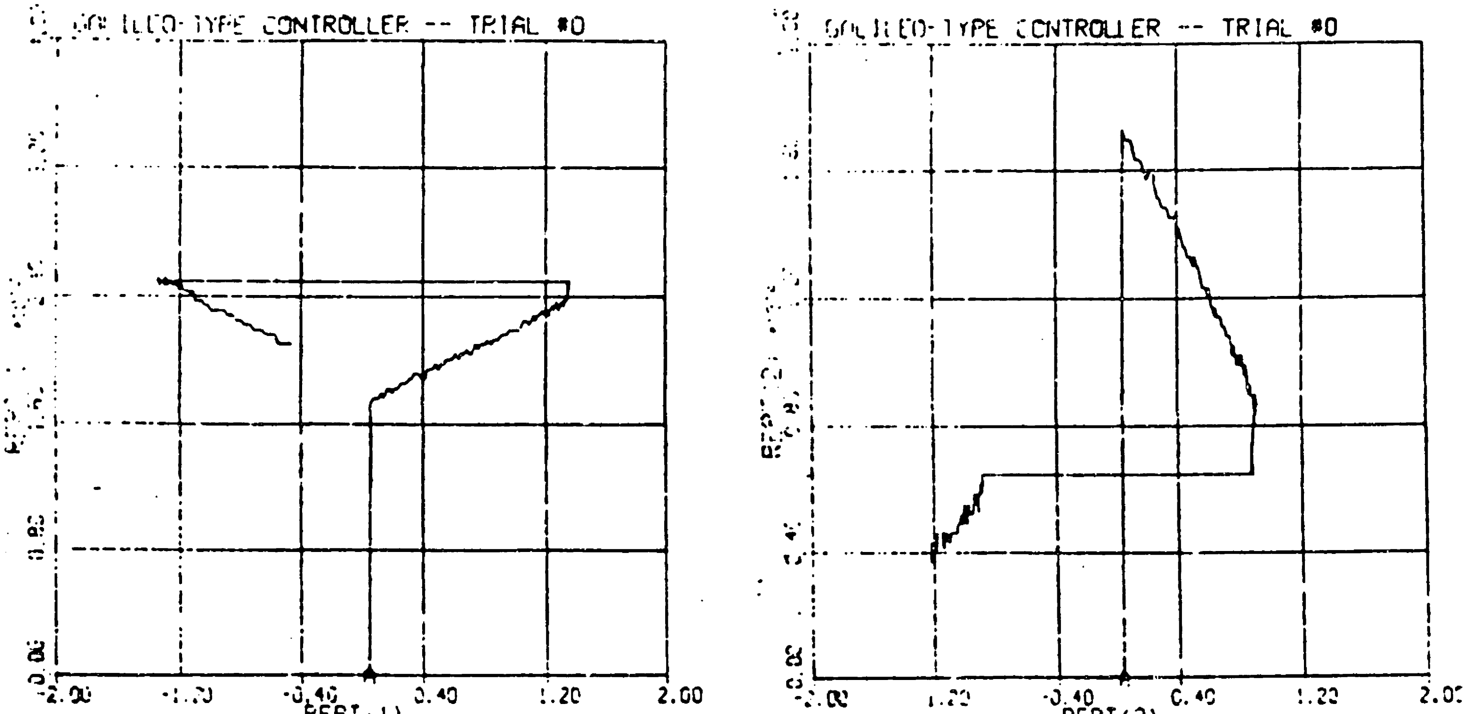
A series of simulations intended to improve performance is outlined in Table 6-II.

First, it was noticed that the pseudo-rate gain seemed to be disproportionately large. A run was attempted in which this gain was set to zero. Phase plane trajectories resulting from this are shown in Figure 6-4. Although these trajectories are far from optimal, they are certainly a substantial improvement over the initial run. Unlike this run, trajectories converge to the origin.

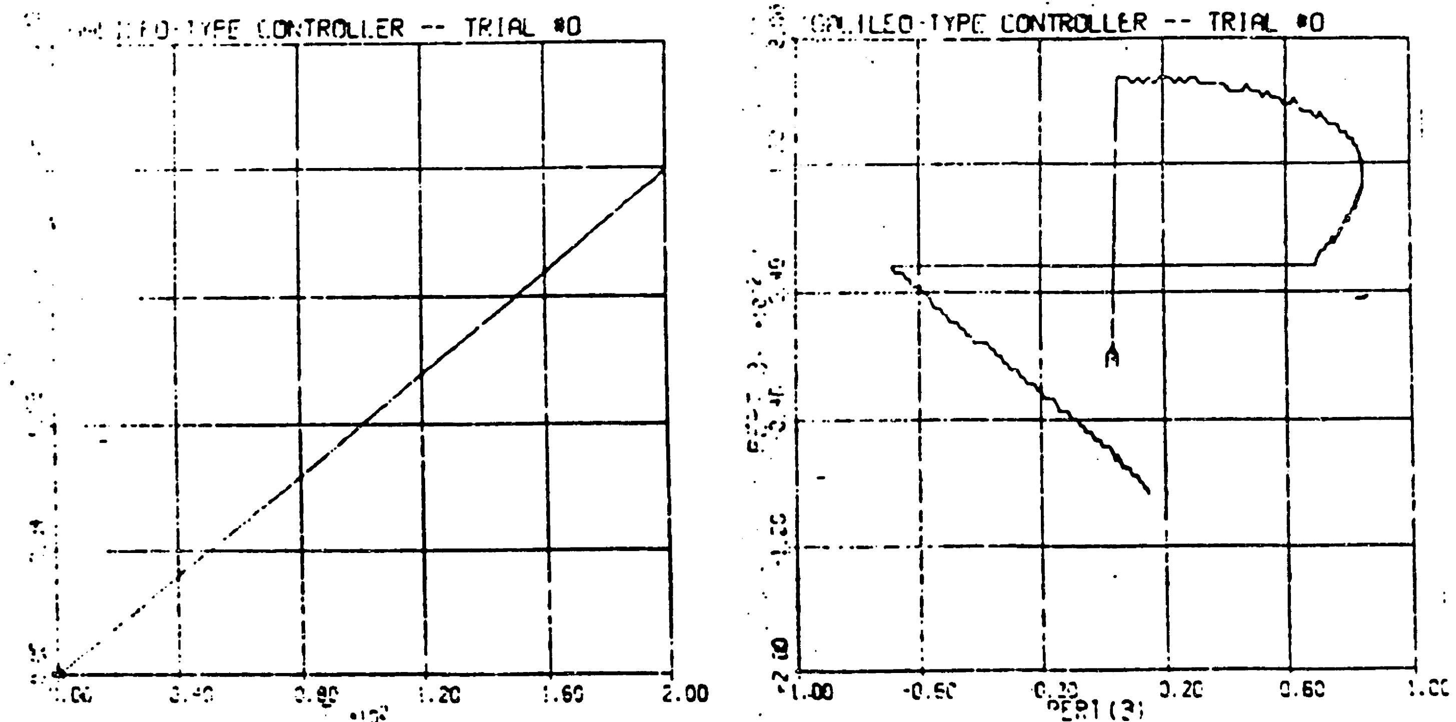
Note, however, that the x and z-body axis trajectories never actually reach the origin.

---

[8] Note that the position errors go through a discontinuity in which they change sign. The reason for this lies in the quaternion representation of attitudes that is used in the simulator, and is too involved to discuss here. For a good explanation of quaternions, see Wertz [24].



(a) x-axis (b) y-axis

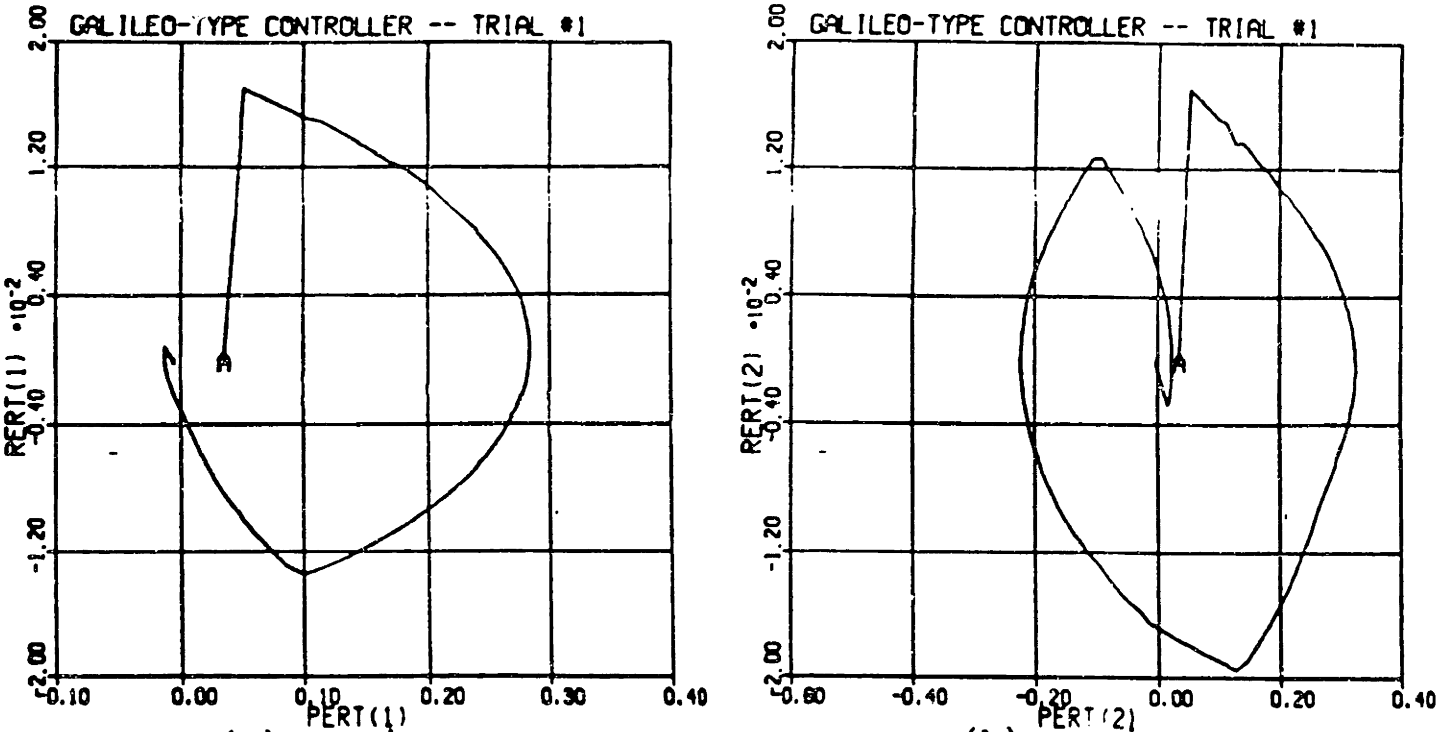


(c) z-axis (d) Propellant Consumed

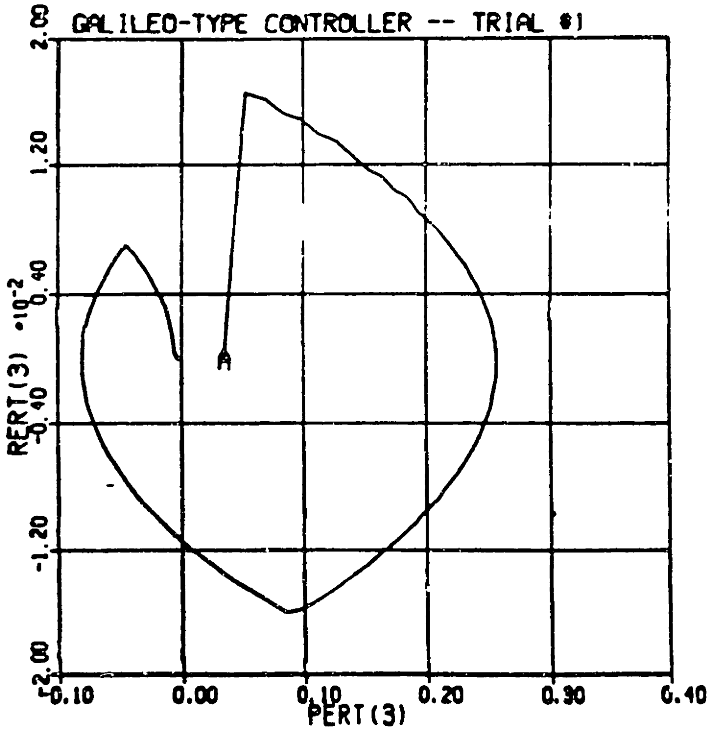
Figure 6-3: Run 1 Galileo Normal Attitude Control Result

Table 6-II: Galileo Normal Attitude Control Simulations

run	comments	settling time (sec)			trans fuel	J	limit cycle attitude			limit cycle rate			ss fuel
		X	Y	Z			X	Y	Z	X	Y	Z	
0	analytically chosen values	----	----	----	----	----	----	----	----	----	----	----	----
1	k_d=0	----	----	----	----	----	----	----	----	----	----	----	----
2	k7=1.8133E-05,1.4933E-05 k6=8.5,7.0	----	----	----	----	----	----	----	----	----	----	----	----
3	k7=1.3333E-06,1.1333E-06 k6=10.0,8.5	94	147	96	2.40	509.6	9.85	9.57	9.69	9.94	9.94	9.98	0.11
4	$k6_{x,y}$=15.625	----	----	----	----	----	----	----	----	----	----	----	----
5	$k6_{x,y}$=11.0	----	----	----	----	----	----	----	----	----	----	----	----
6	$k6_z$=9.5	94	142	83	2.54	481.9	9.99	9.40	9.86	9.97	9.96	9.95	0.23
7	t_{db}=9.0E-04	94	139	99	2.75	482.6	9.34	9.12	9.25	9.99	9.97	9.98	3.61
8	t_{db}=9.5E-04	94	142	80	2.73	467.3	9.45	9.55	9.81	9.98	9.98	9.98	4.56
9	t_{db}=1.0E-03	----	----	----	----	----	----	----	----	----	----	----	----
10	t_{db}=9.5E-04 / t_{c-ch}=33 t_{c-dis}=266 / k_d=3.8E-04	91	139	80	2.75	460.4	9.78	9.90	8.41	9.98	9.97	9.98	2.77
11	k_d=4.275E-04	91	136	80	2.75	457.3	9.63	8.87	8.30	10.0	9.97	9.98	10.0
12	k_d=4.75E-05	----	----	----	----	----	----	----	----	----	----	----	----
13	run #11 values t_{c-ch}=22	91	136	80	2.75	457.3	9.63	8.87	8.30	10.0	9.97	9.98	10.0
14	t_{c-ch}=44	91	136	78	2.75	455.3	9.93	9.18	8.28	9.98	9.98	9.98	4.96
15	t_{c-ch}=77	----	----	----	----	----	----	----	----	----	----	----	----
16	t_{c-ch}=55	----	----	----	----	----	----	----	----	----	----	----	----
17	run #14 values $k7_{x,y}$=2.5E-06	91	142	78	----	----	----	----	----	----	----	----	----
18	$k7_{x,y}$=8.0E-07	----	----	----	----	----	----	----	----	----	----	----	----
19	$k7_z$=2.5E-06	----	----	----	----	----	----	----	----	----	----	----	----
20	$k7_z$=5.0E-07	----	----	----	----	----	----	----	----	----	----	----	----
21	$k7_z$=0.0	----	----	----	----	----	----	----	----	----	----	----	----
22	$k6_{x,y}$=10.25	----	----	----	----	----	----	----	----	----	----	----	----
23	$k6_{x,y}$=9.75	88	142	78	2.76	----	----	----	----	----	----	----	----
24	$k6_z$=9.75	91	136	75	2.78	451.0	9.61	9.34	9.79	9.98	10.0	9.98	4.87
25	t_{c-dis}=200	91	136	78	----	----	----	----	----	----	----	----	----
26	t_{c-dis}	91	136	75	----	----	----	----	----	----	----	----	----



(a) x-axis (b) y-axis



(c) z-axis

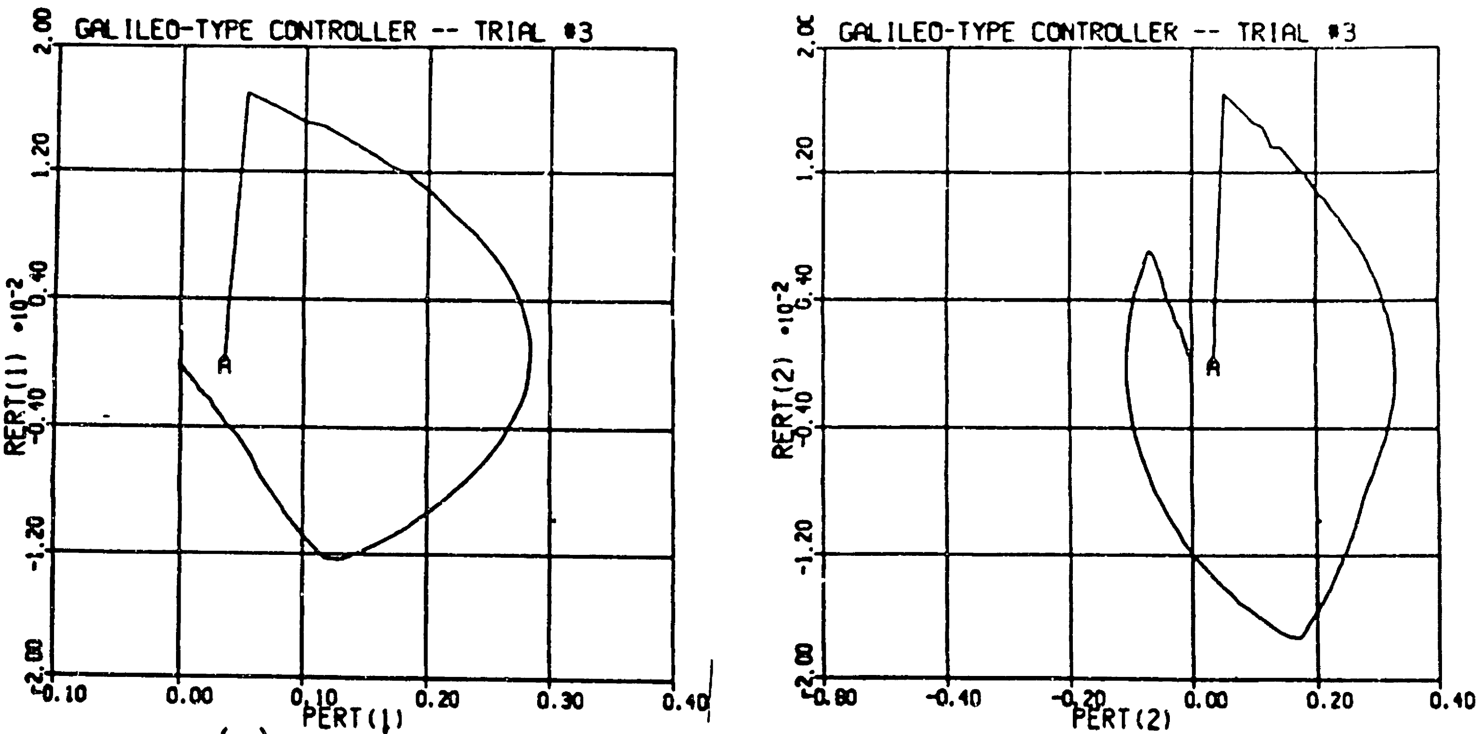
Figure 6-4: Galileo Results with $k_d = 0$

They reach a point near the origin, and then very slowly move in towards it. A likely explanation for this is that the integral gain is too large. When the attitude is positive, the integral term builds up. The brief time during which the attitude is negative is not enough to reduce this term. Thus, the point in the phase plane that zeroes out the error signal is not the origin, but the point at which the attitude errors cancel the integral term. As the integral term slowly decreases, this point slowly moves toward the origin.

To test this, the integral term was reduced in Runs 2 and 3 (keeping the ratio of the integral gain to the rate gain constant). Run 3 phase planes for all 3 axes are shown in Figure 6-5. Trajectories now reach the origin, and the reduction of the integral term has reduced overshoot. Reducing the integral term helped performance.

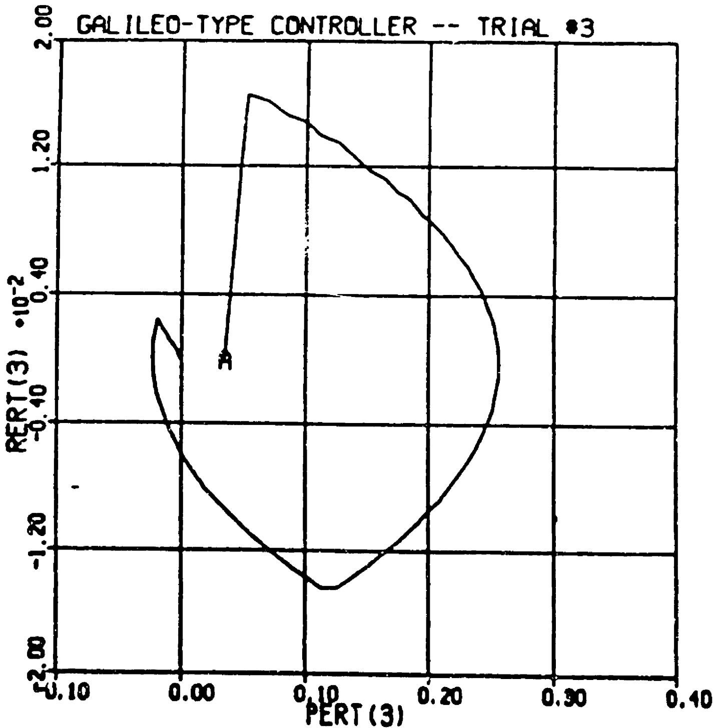
The Figure 6-5 phase planes indicate large undershoot on the x-axis and large overshoot on the y-axis. The dominant term in the Galileo controller seems to be the rate term; the rate gain is many times greater than any other gain. Recall (Figure 3-8) that a rate term can be viewed as a slanted deadband. Thus, this system should perform much like the RLC, with its slanted deadband. There is, however, one important difference. The RLC contains a rate ledge that tends to "capture" trajectories and force them to hit the slanted deadband at a specific rate. This rate can be adjusted so that the approach parabola that is triggered intersects the origin. Trajectories that are not captured by the rate ledge intersect the slanted deadband at rates that vary from the desired rate, thus causing either overshoot or undershoot (see Chapter 5). The Galileo controller does not have a rate ledge. Trajectories are thus no more likely to hit the slanted deadband at one specific rate than at any other. Approach parabolas that intersect the origin are therefore highly unlikely. It would seem, then, that this system must be inferior to the RLC.

If the Galileo controller is similar to the RLC, it might exhibit a similar "fuel bump." If so, the optimization of the deadband width should be simple -- ϕ_{db} should be set to the value in the valley preceding the "bump" (see Chapter 5). In Runs 7-9, this "bump" was



(a) x-axis

(b) y-axis



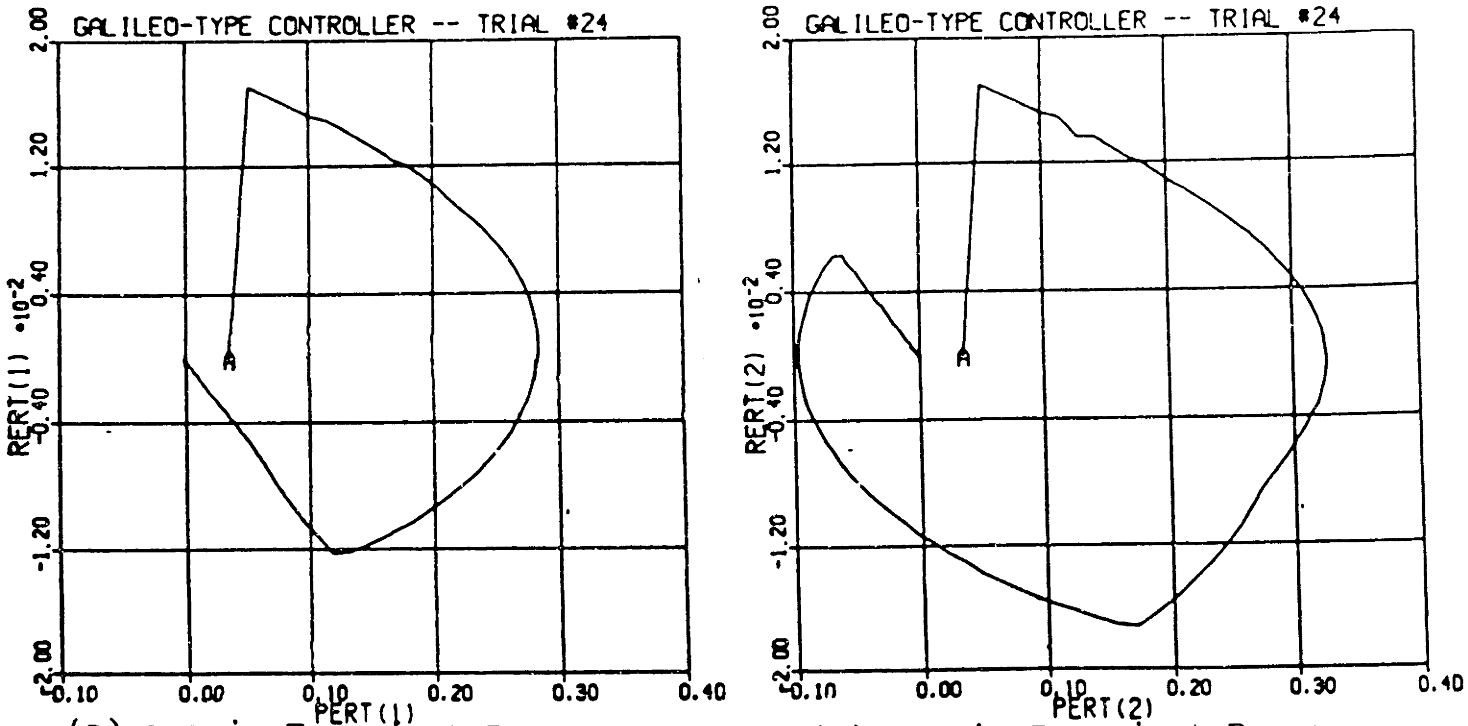
(c) z-axis

Figure 6-5: Galileo Results with Reduced Integral Term found to exist. The valley occurs when $\phi_{db} = 9.5\text{x}10^{-4}$ rad -- very close to where it was in the RLC. This value should optimize steady state performance.

Next, the pseudo-rate term was considered. Describing functions did a poor job of choosing the pseudo-rate gain. This is because the describing function analysis ignored system time delays. Yet these time delays are precisely what the pseudo-rate term is meant to combat (see Chapter 3). If the primary purpose of the term is ignored, one should not *expect* to be able to design it well. The designers of the Galileo reported [2] that their work with this system empirically showed that the pseudo-rate gain, k_d, should be 20% to 60% of the deadband value, the charging time constant should be less than a thruster pulse width, and the discharging time constant should be greater than the sample time. In Runs 11-16, 25, and 26, these parameters were adjusted within these bounds. The designers of the Galileo were proved correct; in Run 14, values in this range resulted in a J value of 455 -- the best the Galileo controller had done to this point. Adjusting the pseudo-rate term also had an effect on steady state, improving fuel consumption substantially while only slightly increasing limit cycle attitudes.

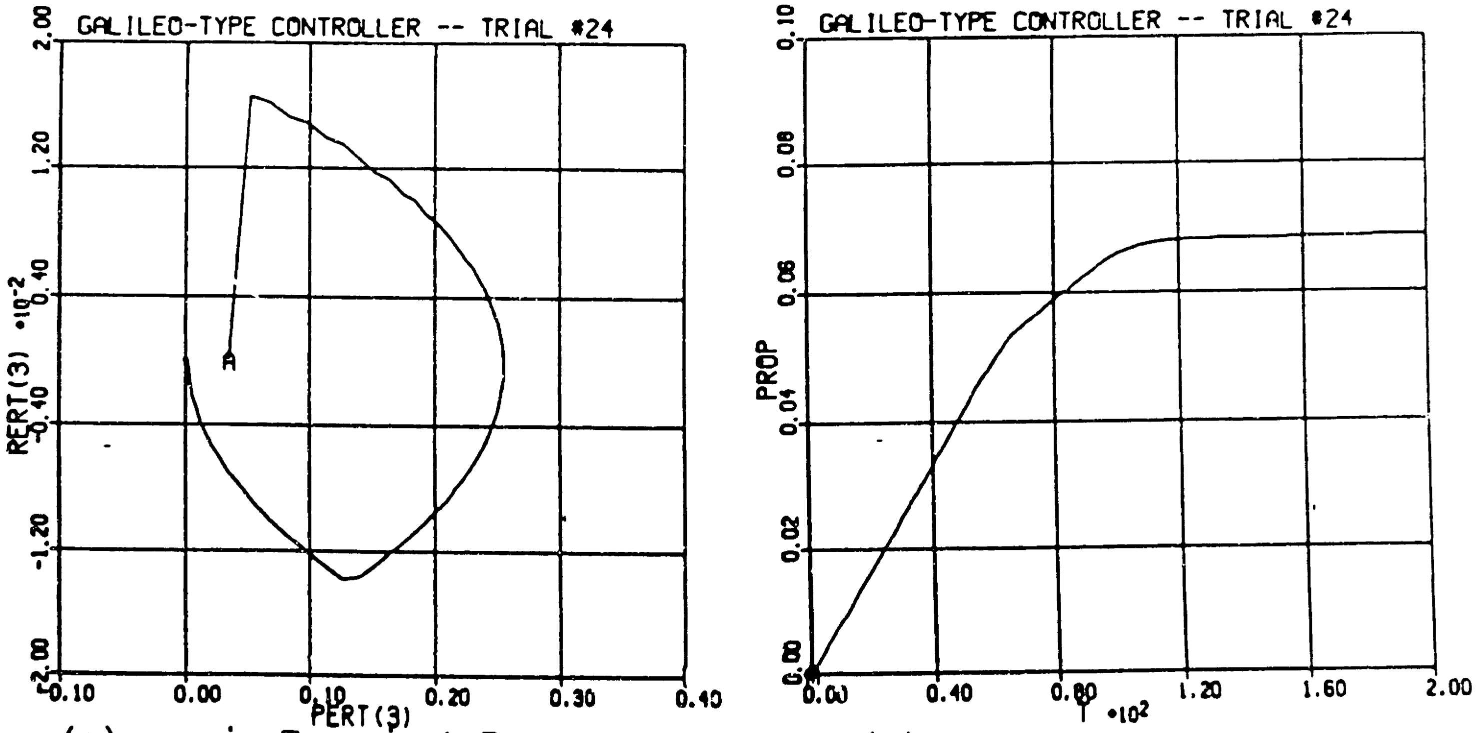
No further intuitive guidelines suggested themselves for finishing the optimization of this system, so k_6 and k_7 were adjusted strictly by trial and error. J could be reduced, but at the expense of steady state fuel consumption. The best value of J was found in Run 24, with J = 451 (steady state was best in Run 13, with J = 457). Run 24 results are plotted in Figure 6-6 and summarized in Table 6-III. Optimized parameter values are given in Table 6-IV.

It was very difficult to choose good parameter values for the Galileo. The initially chosen values performed very poorly. In ignoring time delays, inter-axis coupling, and all other nonlinearities except the deadband-type phase plane, the describing function analysis proved to be inadequate. Physical intuitions about the integral term and the deadband width helped, but no intuitive guidelines presented themselves to aid in the design of other



(a) x-axis Transient Response

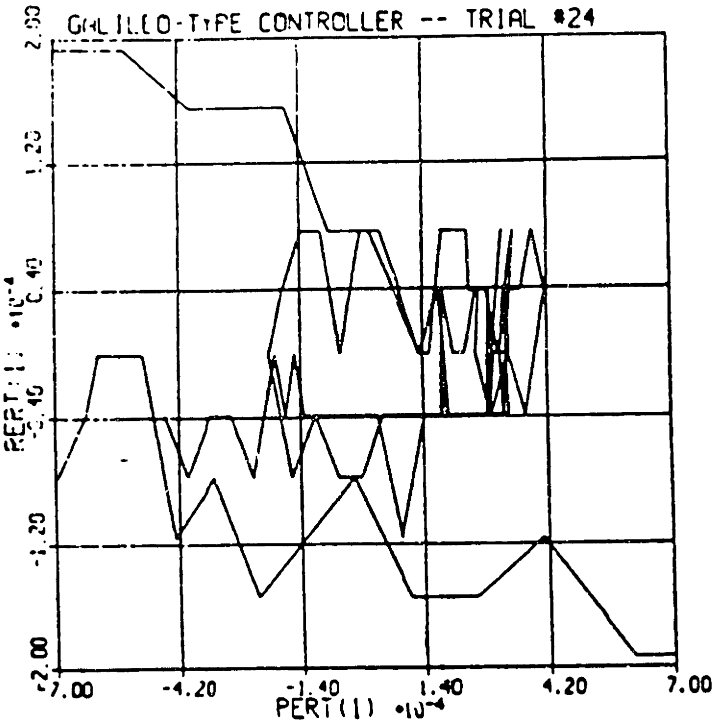
(b) y-axis Transient Response



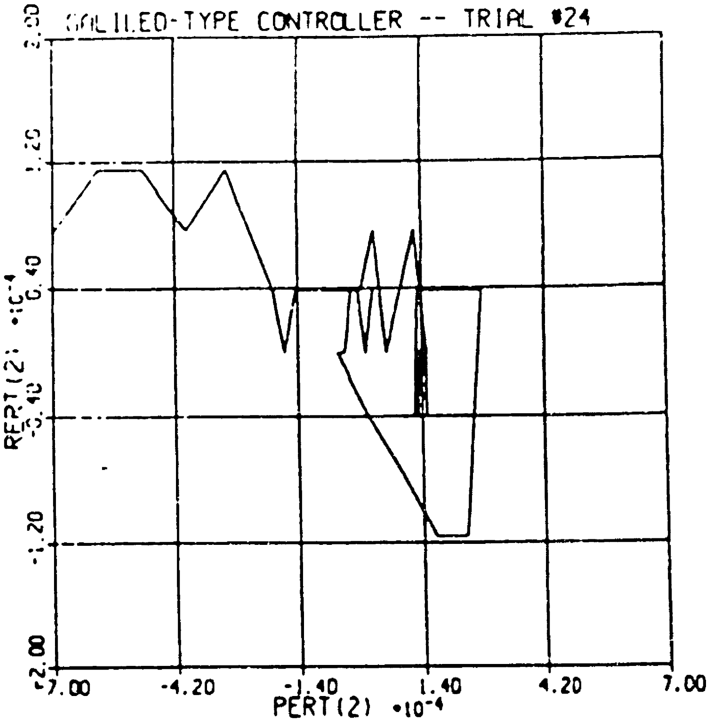
(c) z-axis Transient Response

(d) Propellant Consumed

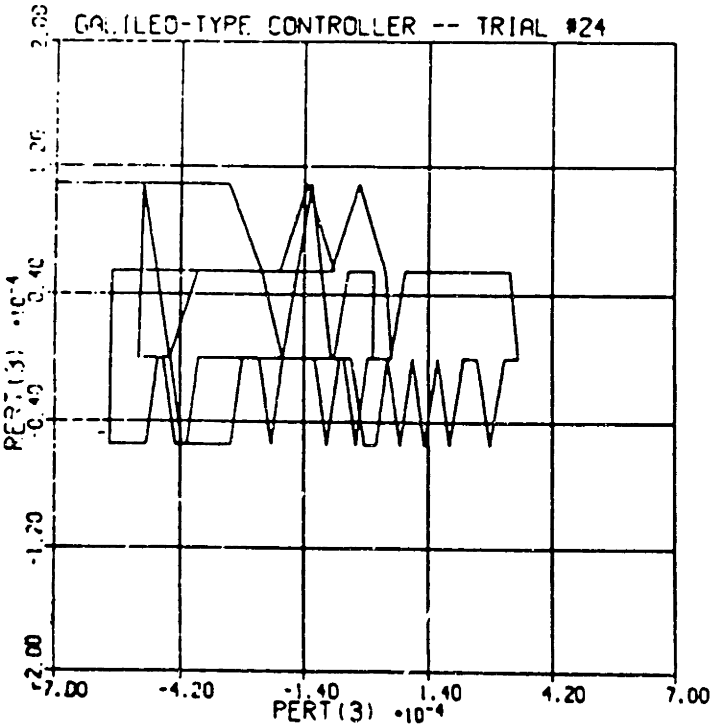
Figure 6-6: Optimized Galileo Results



(e) x-axis Steady State



(f) y-axis Steady State



(g) z-axis Steady State

Table 6-III: Scores for Galileo Normal Attitude Control

Transient Scores

	X	Y	Z	AVG
settling time	6.59	4.41	8.00	6.34
fuel consumption				2.78
rate excursion	10.0	10.0	10.0	10.0
attitude excursion	3.41	1.25	4.68	3.11

J = 451.0

Steady State Scores

	X	Y	Z	AVG
fuel consumption				4.87
limit cycle rate	9.98	10.0	9.98	9.98
limit cycle attitude	9.61	9.34	9.79	9.58

Overall

	X	Y	Z	AVG
ease of assigning parameter values				1

parameters (although the Galileo designers' empirical advice about the pseudo-rate term helped in the adjustment of k_d). Furthermore, pseudo-rate and integral terms had strong effects on both transient response and steady state, thus further complicating the design process. The ease of assigning parameter values score for this control system design is a 1.

In the normal attitude control case, the Galileo controller performs worse than any of

Table 6-IV: Optimized Galileo Parameter Values

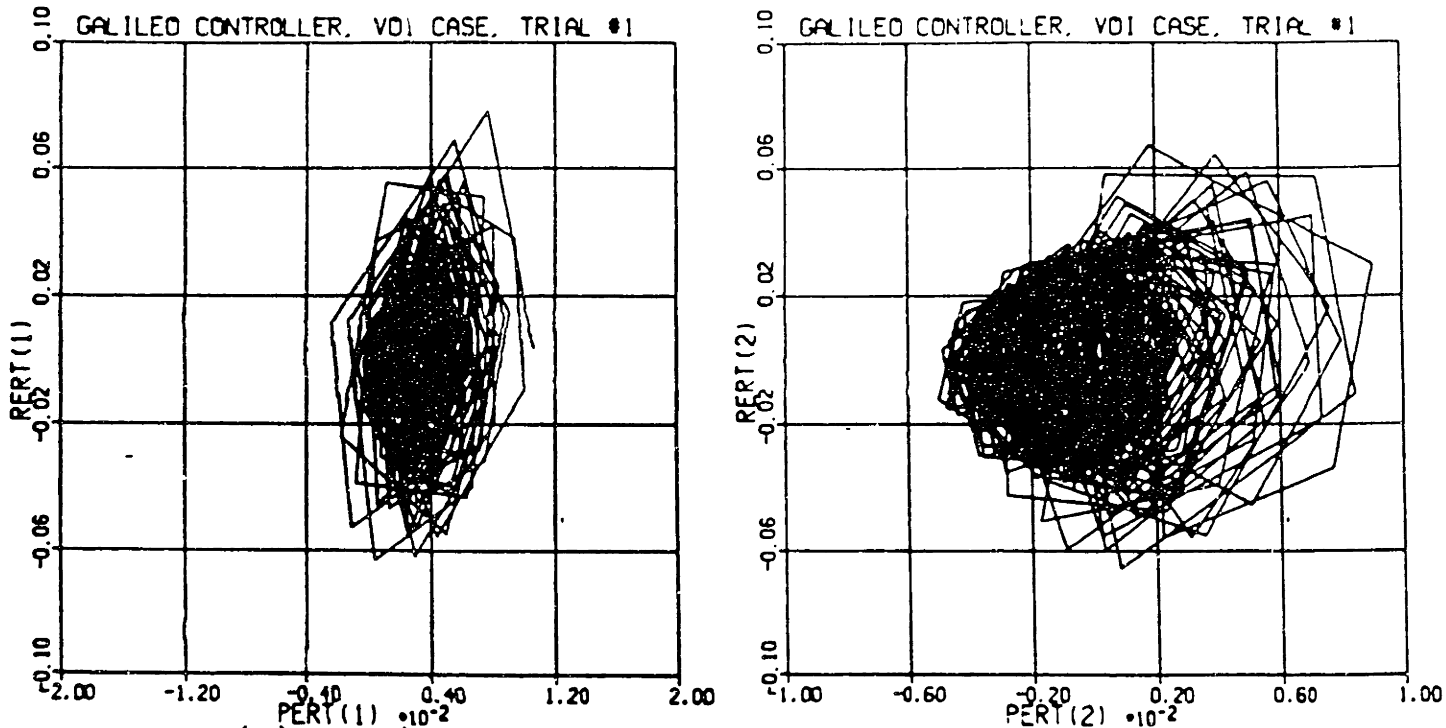
parameter	units	value X & Y	 Z
k6	sec	10.0	9.75
k7	1/sec	1.3333E-06	1.1333E-06
k_d	sec^2	4.275E-04	4.275E-04
Φ_{db}	rad	9.5E-04	9.5E-04
t_{c-ch}	msec	44.0	44.0
t_{c-dis}	msec	266.0	266.0

the phase plane designs that have been considered. With the RLC, overshoot or undershoot could result from a trajectory that hits the slanted deadband at an undesirable rate. In the Galileo controller, there is no rate ledge to capture these trajectories, so undershoot or overshoot should almost always result. Good steady state behavior is possible, but even here the phase plane approaches can do better. In steady state, short pulse regions might prove helpful. This will be tested in the next chapter, when the Clark system is considered.

6.3 VOI

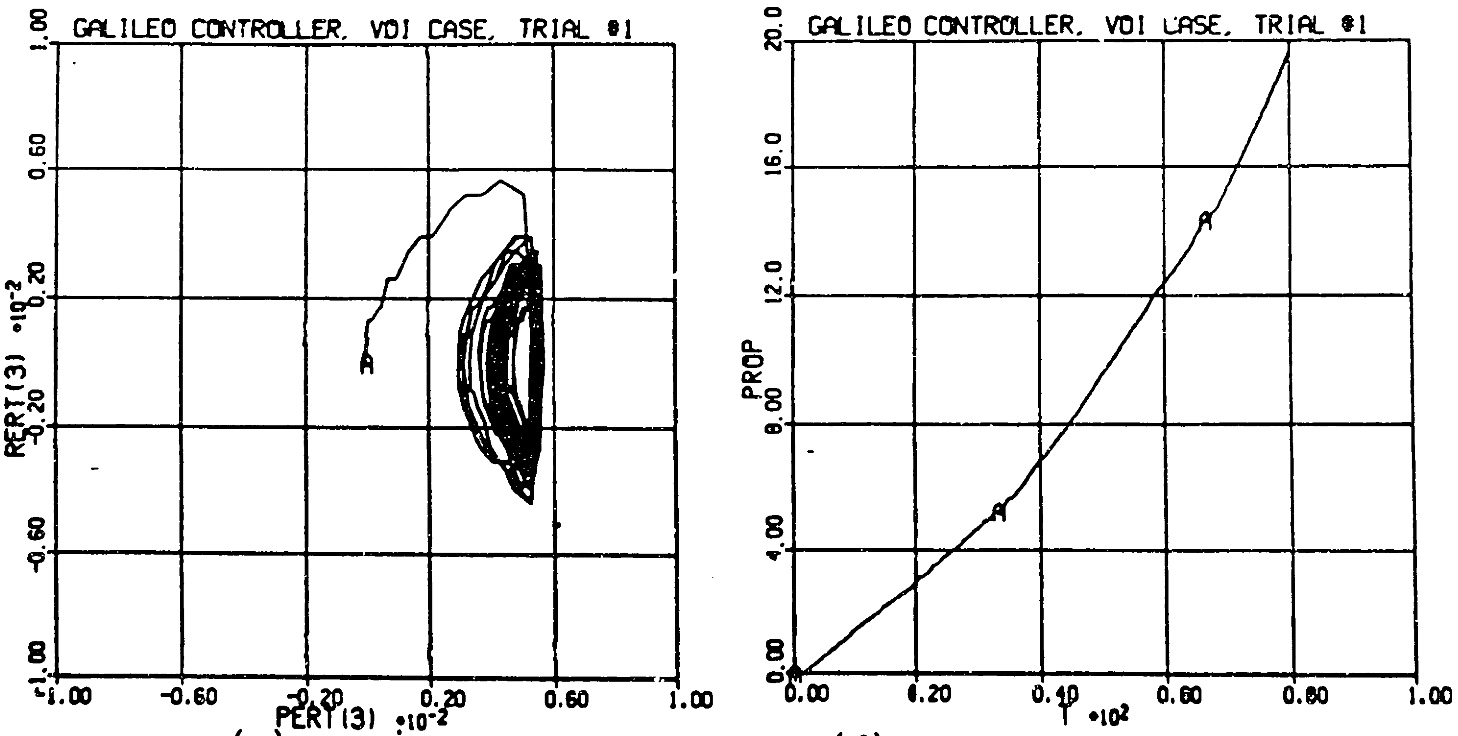
Since the only nonzero parameter values chosen for VOI are ϕ_{db} and the rate gain, k_6, in this case the Galileo controller functions like a simple slanted deadband. Results of a simulation run with these parameters are given in Figure 6-7. The system performs reasonably well. Attitude and fuel criteria are met. X and y-axis rate errors are too high, though (rate errors almost seem unstable), and two-sided deadband behavior predominates.

The parameter tuning process is detailed in Table 6-V. The first parameter to be considered was ϕ_{db}. A wider deadband should help create one-sided deadband behavior. And, in Run 2, such behavior was indeed observed. Phase plane plots of this run are shown in Figure 6-8. Rate errors and fuel consumption were substantially reduced.



(a) x-axis

(b) y-axis



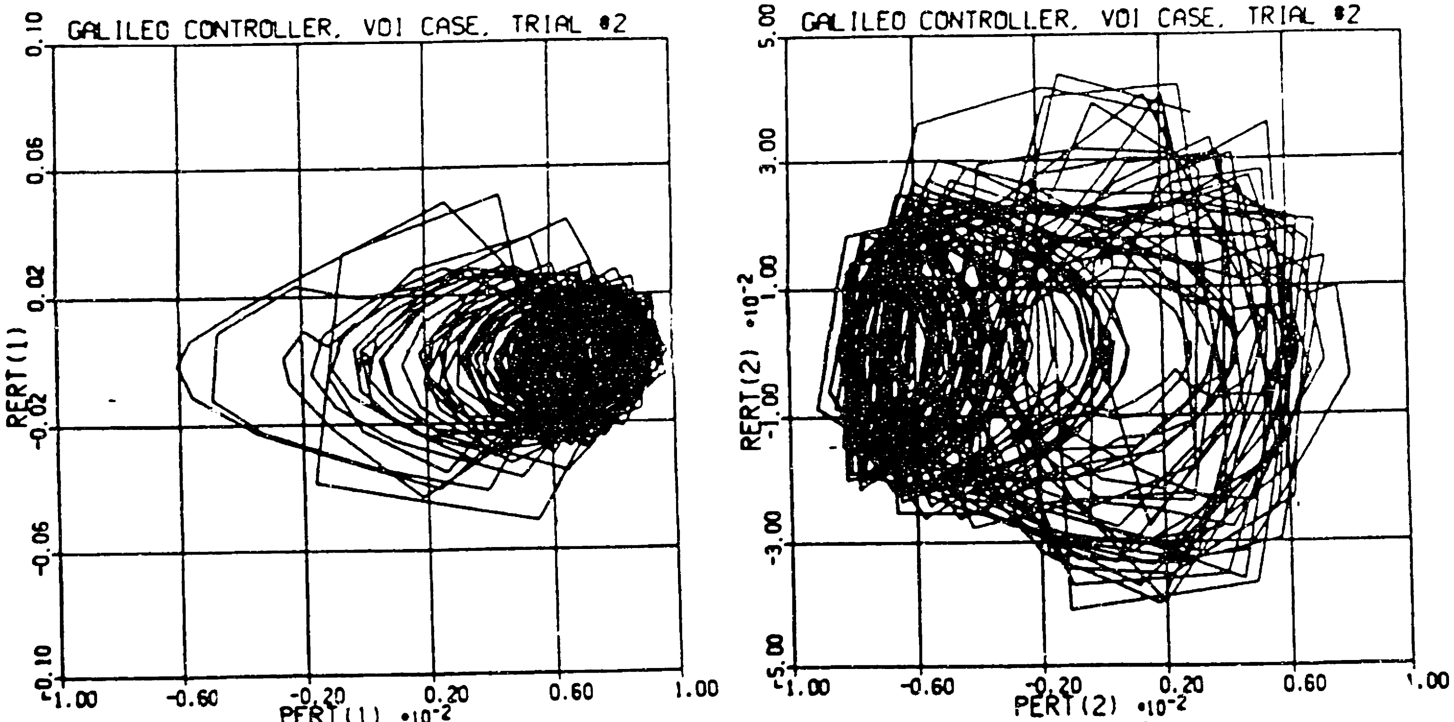
(c) z-axis

(d) Propellant Consumed

Figure 6-7: Initial Galileo VOI Results

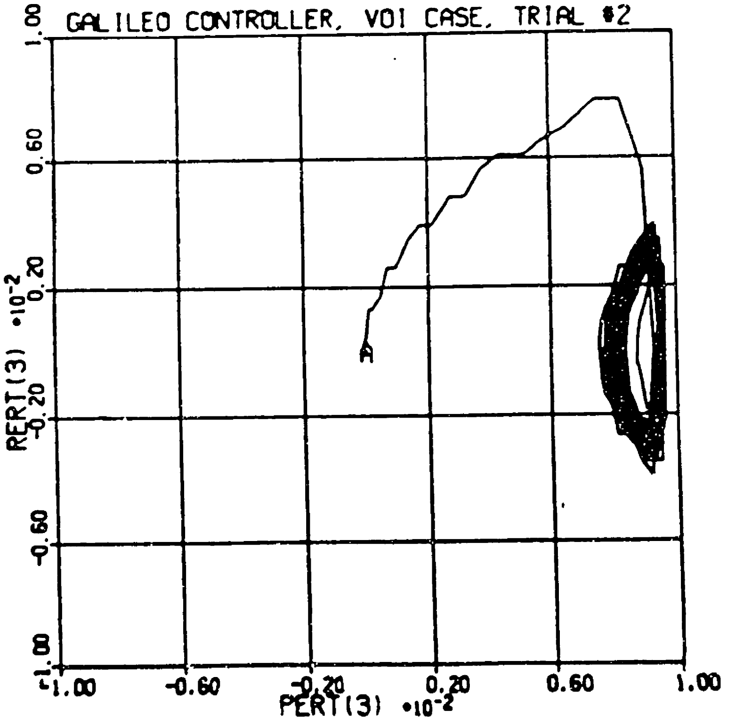
Table 6-V: Summary of Galileo VOI Simulation Runs

run	comments	maximum attitude (deg)			maximum rate (deg/sec)			fuel
		x	y	z	x	y	z	[illegible]
1	initially chosen values	0.612	0.520	0.225	4.42	3.82	0.325	19.65
2	τ_{db}=0.010	0.562	0.522	0.553	2.93	2.49	0.450	10.88
3	k_d=0.04 t_{c-ch}=22 / t_{c-dis}=48	1.36	1.29	1.02	2.14	2.14	0.550	41.94
4	k7=3.3E-04	1.24	1.29	1.02	2.14	2.14	0.550	41.94
5	k_d=0.03	0.853	0.841	0.723	1.66	1.63	0.450	20.46
6	k_d=0.02	0.680	0.711	0.613	1.54	1.50	0.450	9.72
7	k_d=0.01	0.642	0.696	0.595	1.52	1.48	0.450	9.72
8	k_d=0.001	0.546	0.606	0.562	2.62	2.21	0.450	10.38
9	k_d=0.005	0.615	0.737	0.572	1.52	1.56	0.450	9.75
10	k_d=0.01 t_{c-ch}=33	0.626	0.690	0.587	1.52	1.48	0.450	9.71
11	t_{c-ch}=55	0.618	0.703	0.575	1.63	1.73	0.450	9.71
12	t_{c-ch}=11	0.657	0.690	0.596	1.54	1.50	0.450	9.72
13	t_{c-dis}=55	0.654	0.707	0.595	1.94	1.48	0.475	9.73
14	t_{c-dis}=40	0.602	0.684	0.572	1.66	1.75	0.450	9.71
15	run #10 values k6=0.30	0.624	0.690	0.578	1.52	1.48	0.450	9.72
16	k6=0.36	0.625	0.662	0.564	1.86	1.45	0.425	9.73
17	run #10 values k7=3.3E-03	0.600	0.689	0.553	1.52	1.48	0.450	9.71
18	k7=1.65E-02	0.514	0.573	0.462	1.52	1.48	0.425	9.74
19	run #15 values τ_{db}=0.008	0.431	0.445	0.379	1.47	1.75	0.350	9.74
20	k7=3.3E-04	0.516	0.558	0.466	1.64	1.45	0.425	9.74



(a) x-axis

(b) y-axis



(c) z-axis

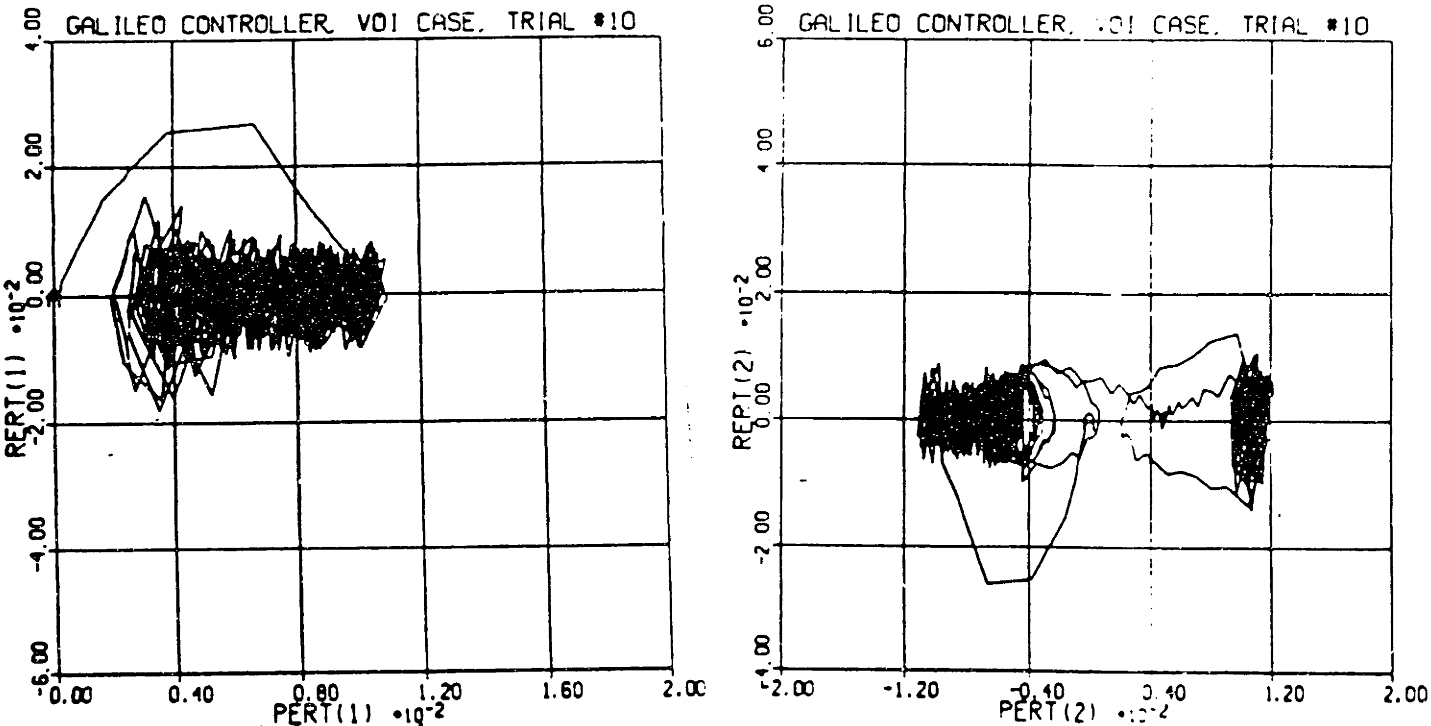
Figure 6-8: One-Sided Deadbands with the Galileo

The pseudo-rate term was expected to be of no help during VOI. The pseudo-rate term estimates vehicle rates, but the only information available to it is the firing of the thrusters. In VOI, disturbance torques also affect the rates. So pseudo-rate control can not provide the information it was intended to give. But when a nonzero pseudo-rate gain was introduced (Run 3), it unexpectedly helped to reduce rate errors. And when pseudo-rate gains and time constants were properly adjusted, rate errors and fuel consumption were reduced significantly. How could this term, which can not function in the way that was intended, actually improve performance?

The pseudo-rate term, in anticipating the response to a thruster pulse, tends to inhibit pulses occurring shortly after another pulse. It has already been observed that avoiding unnecessary thruster firings can help VOI behavior substantially. Pseudo-rate helps to avoid these extraneous firings. A high pseudo-rate gain can increase attitude errors, though (as in Runs 3 and 5). The greater weighting of the pseudo-rate term causes a reduced weighting of the attitude term, thus making the system less sensitive to attitude errors. The system is also made less sensitive to rate errors in this way, but the beneficial effects of reducing excess firings outweighs this factor. k_d must be adjusted such that it is large enough to inhibit extraneous firings, but small enough that attitude errors are kept acceptably small.

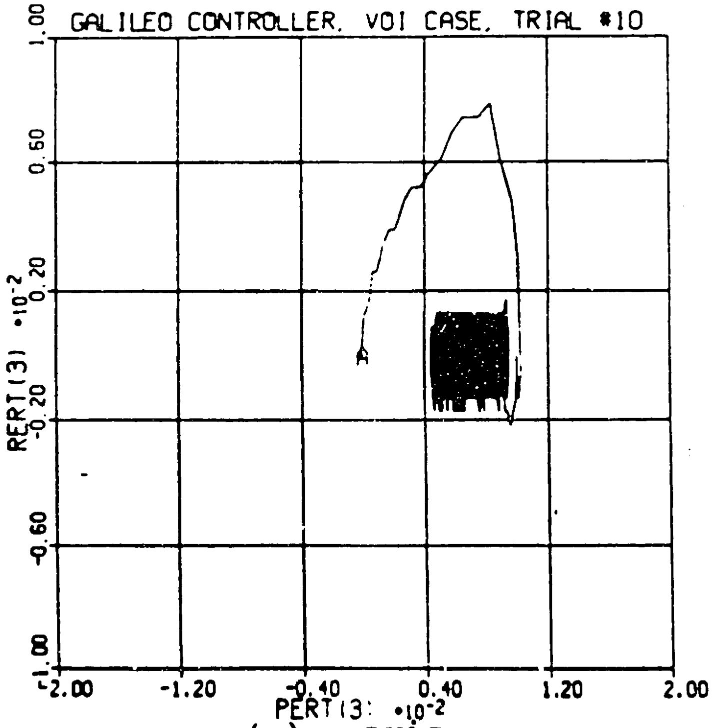
In Run 10, the pseudo-rate term was optimized. Phase plane plots are shown in Figure 6-9. System performance is exceptionally good. Attitude errors are low, fuel consumption is low, and rate errors are low. In fact, except for an initial rate excursion beyond the acceptable limits, rate errors met the design criteria. The pseudo-rate term helped reduce rate errors significantly. If the initial rate excursion could be reduced, system performance would meet the Chapter 4 criteria.

In Runs 15 and 16, k_6 was adjusted in an attempt to reduce these excursions. The attempt was unsuccessful.



(a) x-axis

(b) y-axis



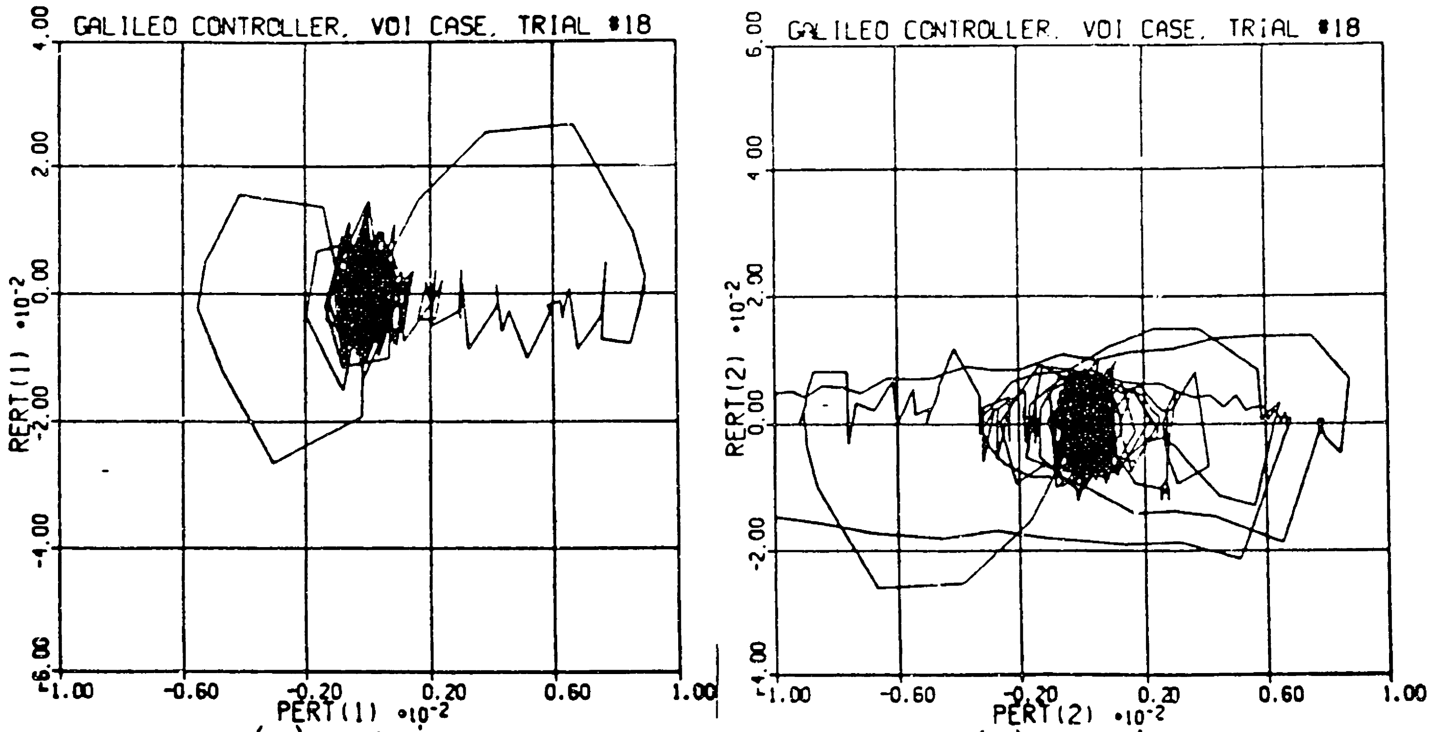
(c) z-axis

Figure 6-9: Galileo Results with the PSR Term Optimized

The only parameter left to adjust was the integral gain, k_7. Adjusting k_7 could not reduce the rate errors. Since these rate errors occur at the very beginning of VOI, the integral term does not have enough time to grow large enough to be of much help. But the integral term had another important effect. Raising k_7 substantially reduced attitude errors. Results of Run 18, in which the integral gain (and the system in general) was optimized are shown in Figure 6-10. Attitude errors reach high values at first, but once the integral term has had time to build up, errors are extremely small. The integral term allows these errors to be centered on the origin, which is something that could never happen in a phase plane design. In such a design, phase plane trajectories tend to pass back and forth across a nonzero deadband limit. But once an integral term is introduced, any nonzero attitude error is penalized, so attitude errors stay small and hover around the origin. There is an upper bound on how high the integral gain can be adjusted, though. If k_7 is too large, rate errors at the end of VOI might increase. At the end of VOI, disturbances are small and a large integral gain can cause the system to overreact to attitude errors, thus introducing instability. The x-axis rate error from Figure 6-9 shows this problem just starting to develop. The integral term was optimized in Run 18. Optimized parameter values are listed in Table 6-VI.

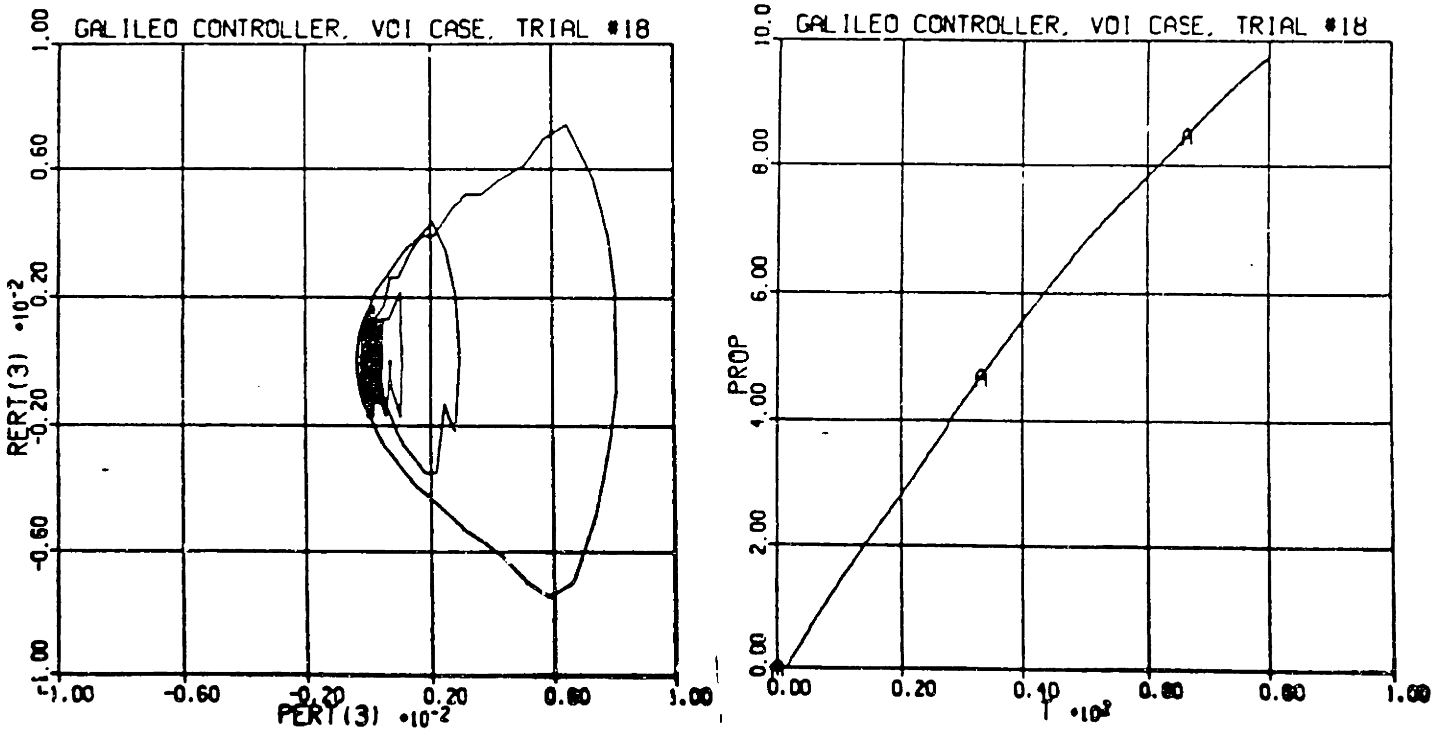
Despite the Galileo controller's exceptional performance, the choice of parameter values was a difficult one. Initially determined parameter values performed well, but tuning these parameters had to be done on an ad-hoc basis. There were few intuitive guidelines, and in fact, the pseudo-rate term behaved in a way that was initially quite counterintuitive. The ease of assigning parameter values score for this system is a 4.

But the Galileo controller's great performance might well make up for the difficulty of assigning parameter values to it. Despite large initial excursions, attitude and rate errors stay very small. And the initial attitude excursion is still within acceptable bounds. The pseudo-rate term really helps reduce rate errors, while the integral term, once it has had



(a) x-axis

(b) y-axis



(c) z-axis

(d) Propellant Consumed

Figure 6-10: Optimized Galileo Controller VOI Performance

Table 6-VI: Optimized Galileo Controller VOI Parameters

parameter	units	value X & Y & Z
Φ_{db}	rad	0.010
k6	sec	0.27
k7	1/sec	1.65E-02
k_d	sec^2	0.010
t_{c-ch}	msec	33.0
t_{c-dis}	msec	48.0

time to build up, reduces attitude errors significantly. The only problem with this system is that the initial rate excursion is unacceptably large (but no larger than the rate errors observed in most phase plane designs). The Galileo controller performs exceptionally well in VOI.

6.4 Generalizations About Controller Designs

It is even more difficult to generalize about controller designs than it was to generalize about phase plane designs. It is impossible to know whether or not additional controller terms that have not been considered here might improve performance dramatically. Still, some generalizations do seem possible.

Although the Galileo controller performed poorly in the normal attitude control case, it performed exceptionally well in VOI. When designing the MGN phase plane and the RLC with short pulse regions, it was observed that gyro time delays caused extraneous thruster firings, thus increasing rate errors in an unavoidable way. The pseudo-rate term helps abolish these extraneous firings, thus performing better than any phase plane *could* perform. And by centering attitudes on the phase plane origin, the integral term greatly reduces attitude errors. It seems reasonable to conclude that controller designs have much potential for use in high disturbance cases, such as VOI.

Controller design parameters are more difficult to choose than phase plane design parameters. It is doubtful that any controller can be designed with the ease of the very simple geometric intuitions that can be used to design a phase plane. Also, any term in a controller must apply over an entire trajectory; therefore, the same terms affect both transient response and steady state. This is not necessarily true of phase plane parameters. And even though controller performance might be improved by the addition of new terms, adding these terms generates new parameters, and thus more complexity.

If thrusters pulses of less than a full sampling interval were made possible, the Galileo controller might perform better. Unfortunately, since the Galileo controller is not a phase plane approach, short pulse regions can not be added to the phase plane. Another way of adding short pulses to the Galileo controller is implemented in the Clark system, which will be discussed in the next chapter.

Chapter 7

Variable Pulse Width Designs

7.1 The Clark System

7.1.1 Initial Analysis

The "Clark" system is a simple variable pulse width approach that is intended to improve steady state performance. The system uses a controller -- such as, for instance, the optimized Galileo controller found in the previous chapter. Instead of sending the controller output to a phase plane, however, the Clark system multiplies the output by a gain, M. The result of this multiplication is a time. The thrusters are commanded to fire (with correct sign) for this length of time. The only parameter that needs to be chosen is the gain M.

Far from the origin, the commanded thruster times will probably exceed the sample time. In this situation, the thrusters will always fire full on. Only near the phase plane origin, *ie.* in steady state, will the controller output generally be small enough to cause the shorter pulse widths. The Clark system is thus intended to improve steady state; transient response should remain largely unaffected.

In the normal attitude control case, hardware limitations prohibit continuously variable pulse times on the Magellan spacecraft. Commanded pulse widths must be in increments of 11 msec. Since it is desirable to test the Clark system in the realistic Magellan environment, this constraint will be adopted for this thesis. Thruster on-times will be rounded down to the nearest 11 msec; on-times greater than a sampling interval will of course be rounded to only one sampling interval. While this does not result in truly continuous pulse width modulation, it is certainly much closer than any other system that has been discussed so far.

The optimal normal attitude control deadband width for the Galileo was $\phi_{db} = 9.5\times10^{-4}$ rad. The gain, M, will thus be chosen such that a controller output of 9.5×10^{-4} rad will result in a thruster pulse width of 132 msec (1 sampling interval). This implies that M = 139 sec/rad.

In VOI, pulse widths must also be in increments of 11 msec, but there are some additional hardware limitations. First, pulses of less than 22 msec can not be realized. Second, whenever a pair of thrusters fire simultaneously, they must fire for the same period of time. Allowable pulse widths are thus 22 msec, 33 msec, 44 msec (2 thrusters firing together for 22 msec, for an effective pulse width of 44 msec), and 66 msec (2 thrusters firing for 33 msec). Although this seems to depart substantially from the ideal of a continuously variable pulse width, it will be adopted as a constraint. It is essential to remain consistent with Magellan hardware limitations.

In the MGN phase plane, optimal VOI performance was obtained with A3 = 0.012 rad. Within this deadband, pulse widths of less than a full sampling interval were delivered. Since this system is being designed for a similar situation, an M value will be chosen such that a controller output of 0.012 rad results in a 66 msec pulse. Thus, M = 5.5 sec/rad.

Initially chosen M values are summarized in Table 7-I.

Table 7-I: Initial Parameter Values for Clark System

parameter	units	normal attitude control all axes	VOI all axes
M	sec/rad	139	5.5

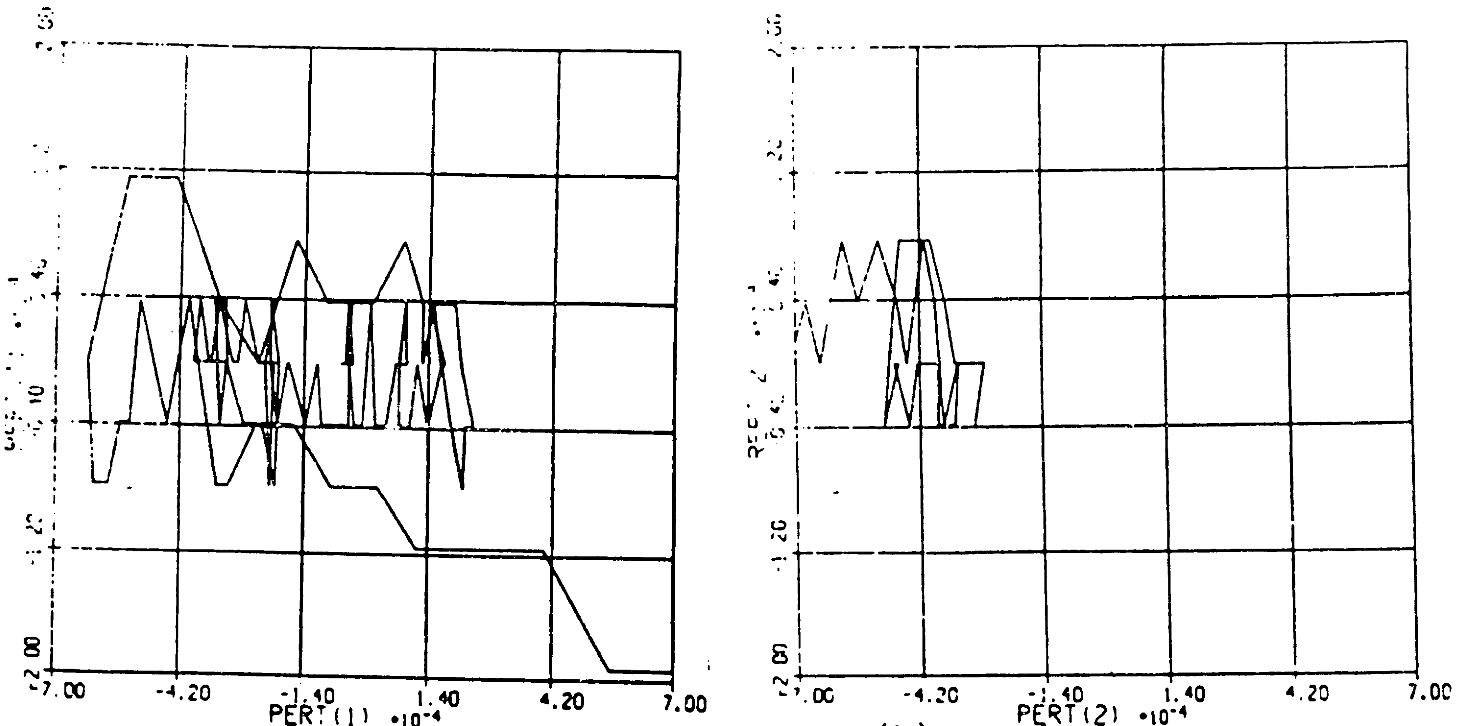
7.1.2 Normal Attitude Control

As expected, the parameter values chosen above resulted in significantly improved limit cycle attitudes. The wide variation of available pulse widths allowed tighter deadbands by more closely tailoring thruster firings to reflect the current state. Transient response was worsened somewhat, but was still virtually indistinguishable from the results of the Galileo controller depicted in Figure 6-6. Steady state fuel consumption, however, deteriorated substantially.

Steady state phase plane plots are shown in Figure 7-1. Phase plane trajectories seem to exhibit a large amount of jitter. And this demonstrates a basic problem with variable pulse width designs. These designs can be expected to cause thruster firings during almost every sampling interval. Even if pulses are perfectly tailored to cause a trajectory to travel from the current state to the origin, nonlinearities and inter-axis coupling will cause the origin to be missed. Thus, the thrusters will fire again; there is no real deadband. This is what causes the high rate of steady state fuel consumption, and probably makes low fuel consumption impossible to obtain.

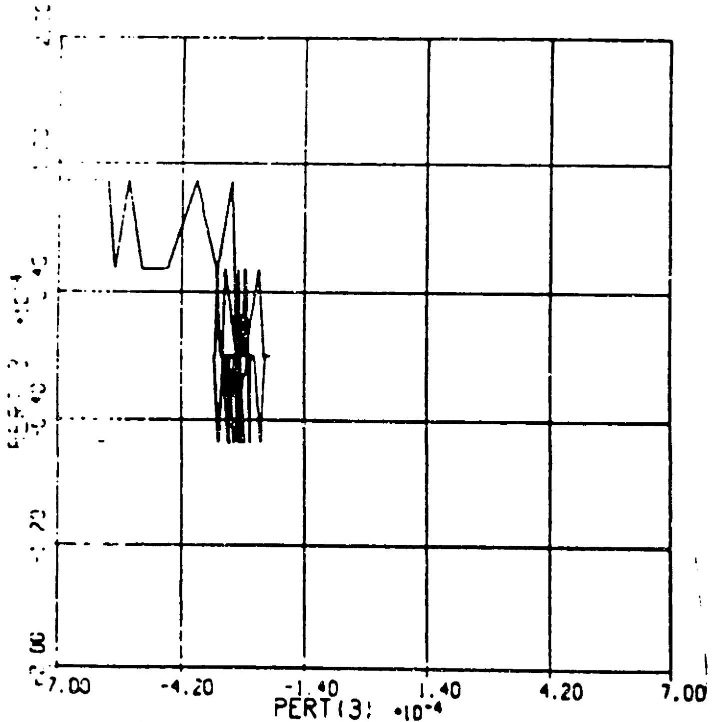
The Clark system *does* contain a narrow deadband, though. A spacecraft state falling in the region in which commanded thruster pulse widths are less than the minimum deliverable pulse width of 11 msec results in no thruster firings. Although this deadband is too narrow to prevent the fuel problems outlined above, it might help in the choice of a value for the parameter M. The Galileo controller contained a "steady state fuel bump". Since the Clark system actually does contain an implicit deadband, it probably has a similar "fuel bump". M should be adjusted in an attempt to find the valley that precedes the "bump". Table 7-II records the search for the optimal value of M.

Effective Clark deadbands are plotted against fuel consumption in Figure 7-2. The "fuel bump" can be located easily. But the Clark system's "fuel bump" is wider than that of the RLC or the Galileo controller. In these two systems, exceeding the deadband value resulted in pulse widths of a predetermined length, which determined the location of the "bump". When short pulse regions were added to the RLC, deadband violations resulted in shorter pulse widths, and the "bump" moved to a new location. In the Clark system,



(a) x-axis Steady State

(b) y-axis Steady State



(c) z-axis Steady State

Figure 7-1: Run 1 Normal Attitude Control Clark Results

Table 7-II: Normal Attitude Control Clark Runs

run	comments	settling time (sec)			trans fuel	J	limit cycle attitude			limit cycle rate			ss fuel
		X	Y	Z			X	Y	Z	X	Y	Z	
1	initially chosen value	91	139	72	2.44	471.4	9.89	9.27	9.93	10.0	9.99	9.99	0.14
2	M = 100	91	139	72	2.56	463.7	9.64	9.47	9.91	9.99	9.98	9.99	0.19
3	M = 50	88	136	75	2.67	454.2	9.49	9.02	9.98	10.0	10.0	9.99	0.57
4	M = 30	----	----	----	----	-----	7.67	8.23	7.73	---	----	----	0.25
5	M = 40	88	136	86	2.64	466.5	9.13	9.50	9.94	9.99	9.98	9.99	0.59
6	M = 45	88	134	75	2.67	451.8	9.65	9.49	9.67	9.98	9.99	9.99	0.84
7	M = 20	----	----	----	----	-----	6.08	6.87	7.44	----	----	----	0.23
8	M = 15	----	----	----	----	----	----	----	----	----	----	----	0.23

deadband violations result in a range of possible pulse widths. This is what causes the "bump" to spread.

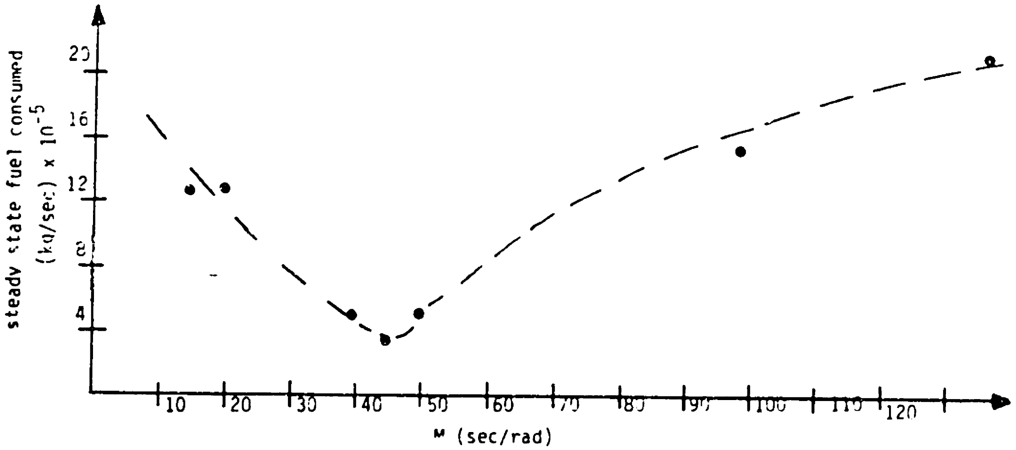
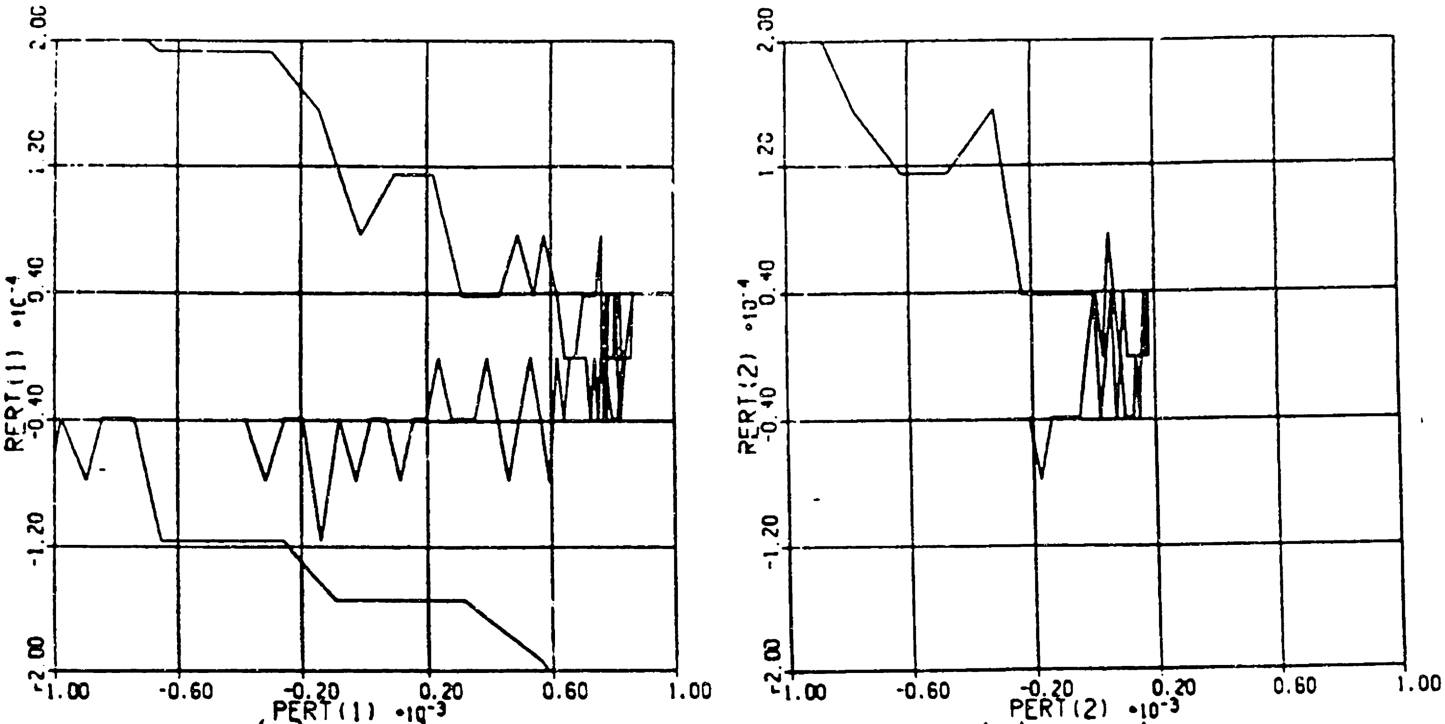


Figure 7-2: The Clark System's "Fuel Bump"

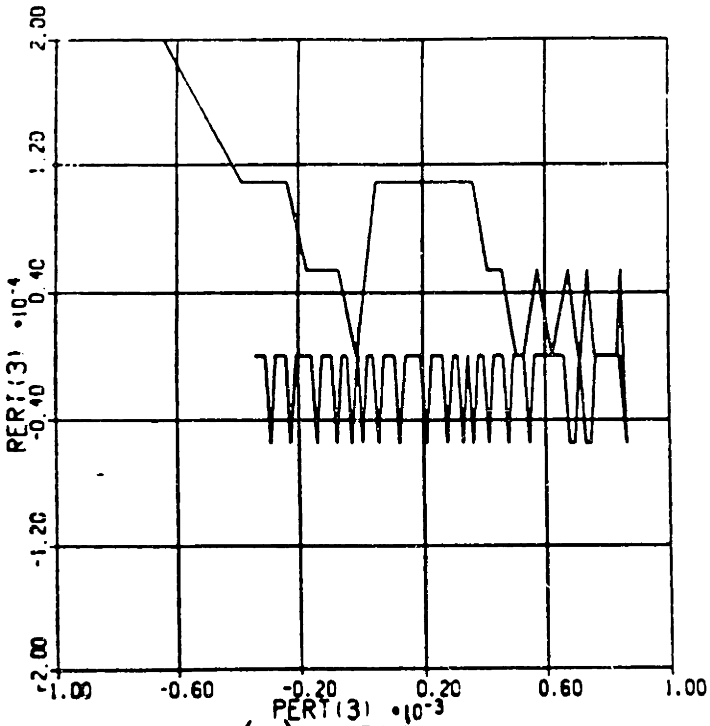
The valley was found in Run 6. Run 6 steady state performance is shown in Figure 7-3 (transient response is still indistinguishable from Figure 6-6), and results are summarized in Table 7-III. Limit cycle attitudes show a significant improvement over the Galileo system, but, as predicted above, steady state fuel consumption is still very high.

The Clark system is quite easy to assign parameter values for. If a predesigned



(a) x-axis

(b) y-axis



(c) z-axis

Figure 7-3: Optimized Clark Steady State Phase Planes

Table 7-III: Summary of Optimized Clark Performance

Transient Scores

	X	Y	Z	AVG
settling time	6.82	4.48	5.22	5.50
fuel consumption				2.67
rate excursion	10.0	10.0	10.0	10.0
attitude excursion	3.41	1.25	4.68	3.11

J = 451.8

Steady State Scores

	X	Y	Z	AVG
fuel consumption				0.84
limit cycle rate	9.98	9.99	9.99	9.99
limit cycle attitude	9.65	9.49	9.67	9.61

Overall

	X	Y	Z	AVG
ease of assigning parameter values				9

controller is assumed, there is only one parameter to tune, and its adjustment is relatively simple if the "steady state fuel bump" phenomenon is considered. The ease of assigning parameter values score for this system is a 9. The only obstacle preventing this score from being a 10 is that the initial guess at an M value was far from the final, optimized value.

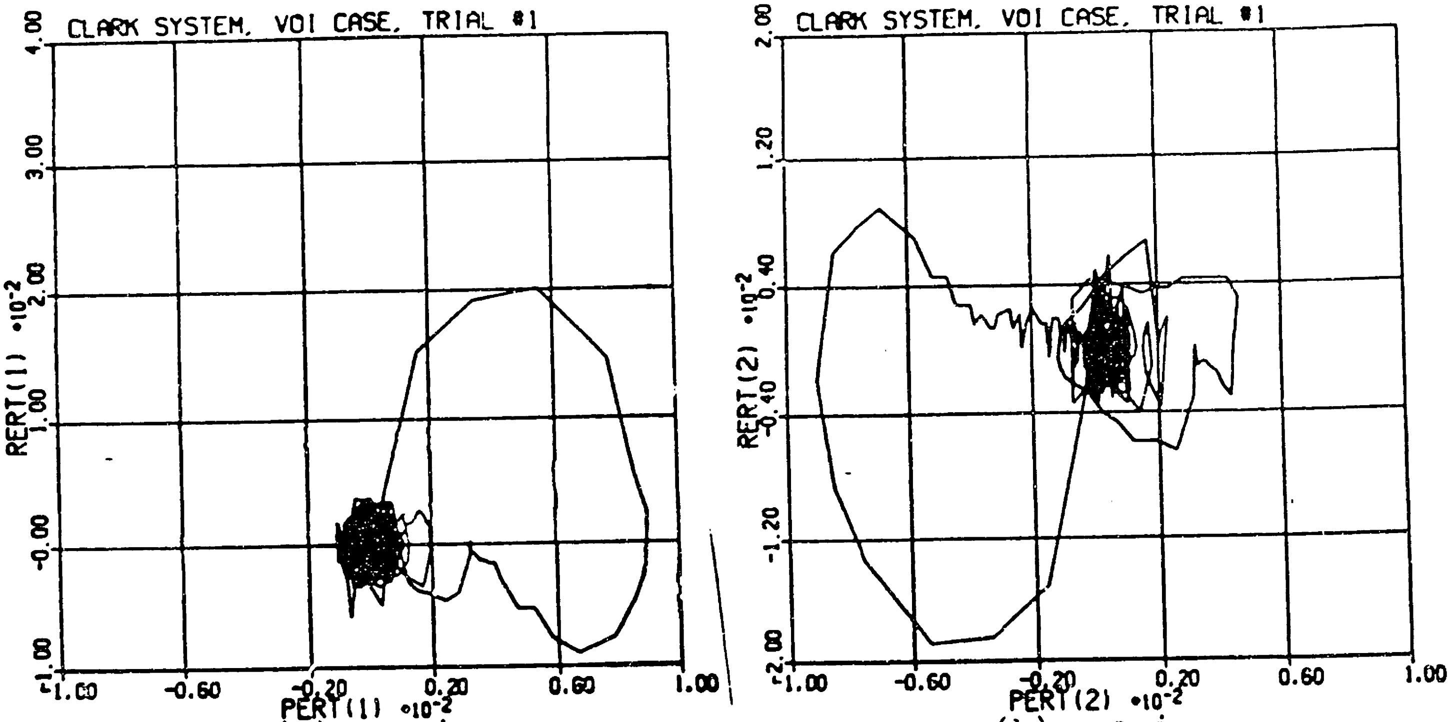
In sum, the Clark control scheme improves the Galileo system's limit cycle attitude amplitudes and is very easy to choose parameter values for. Unfortunately, it seems to have a slight effect on transient response (but only *very* slight in the optimized Run 6). Also, the system causes very high steady state fuel consumption. And even the improved limit cycle attitudes are no better (and are, in fact, slightly worse) than those found in the MGN phase plane or the RLC with short pulses. In the normal attitude control case, the Clark system seems to be of questionable value.

7.1.3 VOI

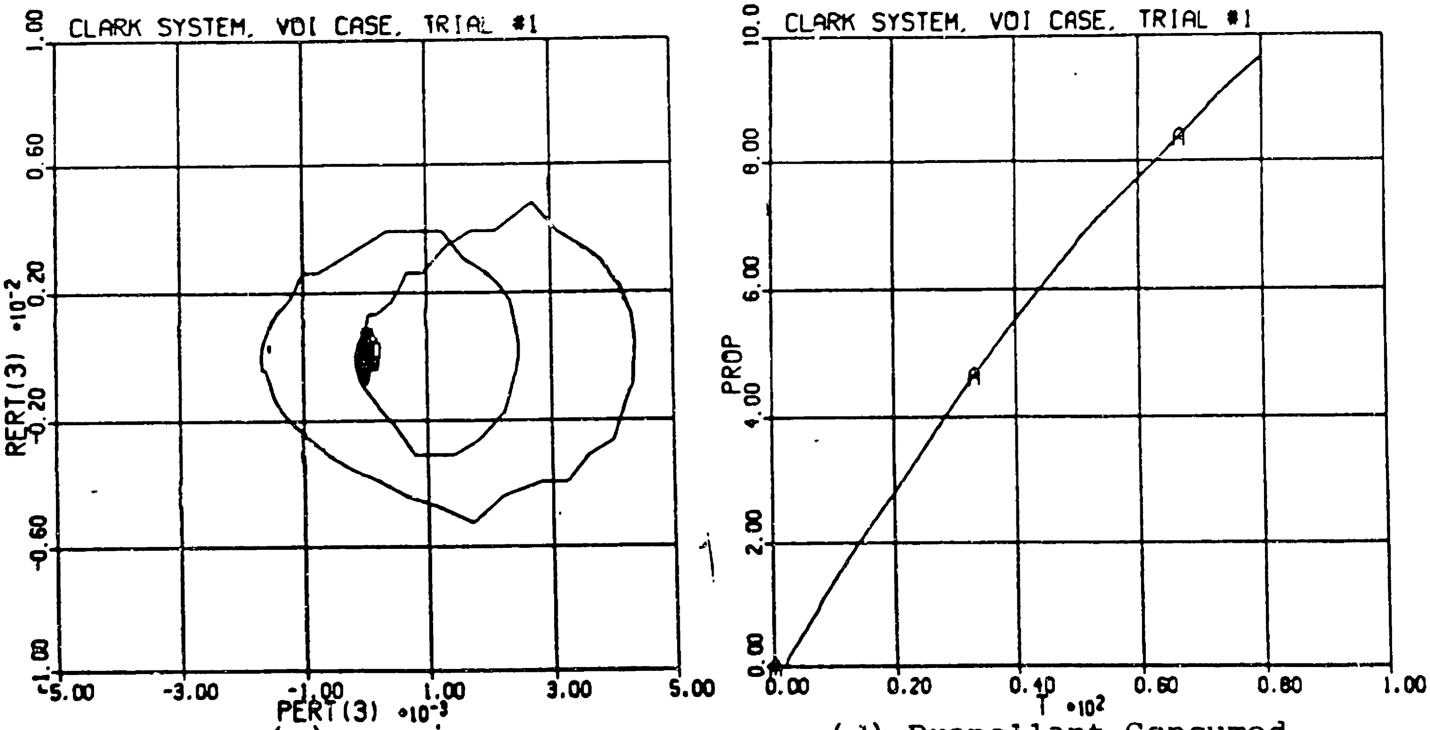
Results of a simulation run with the M value determined in section 7.1.1 are shown in Figure 7-4. System performance is exceptional. After initial excursions, rate and attitude errors become extremely small. Just as the Clark system helped limit cycle attitudes in the normal attitude control case, it reduces attitudes and rates in VOI. Attitude errors and fuel consumption stay within reasonable bounds, and rate errors, which no other system so far has been able to reduce to within acceptable limits, only exceeds these limits on the x-axis, and only by 0.01 deg/sec.

Simulation results show that *no 66 msec pulses were commanded* in this run. The longest commanded pulse width was 44 msec. Evidently, the frequent 66 msec pulses resulting from other system designs are absolutely unnecessary. It has been observed that extraneous thruster firings are detrimental to VOI rate errors. The Clark system really minimizes these extraneous firings. This is responsible for this system's excellent reduction of rate errors.

In the normal attitude control case, the Clark system caused an extremely high rate of steady state fuel consumption. The thrusters delivered pulses in almost every sampling interval. But in VOI, the thrusters fire almost continuously anyway. By reducing the width of these firings, fuel is conserved. Thus, the large steady state fuel consumption does not occur in VOI.



(a) x-axis (b) y-axis



(c) z-axis (d) Propellant Consumed

Figure 7-4: Initial Clark Results in VOI

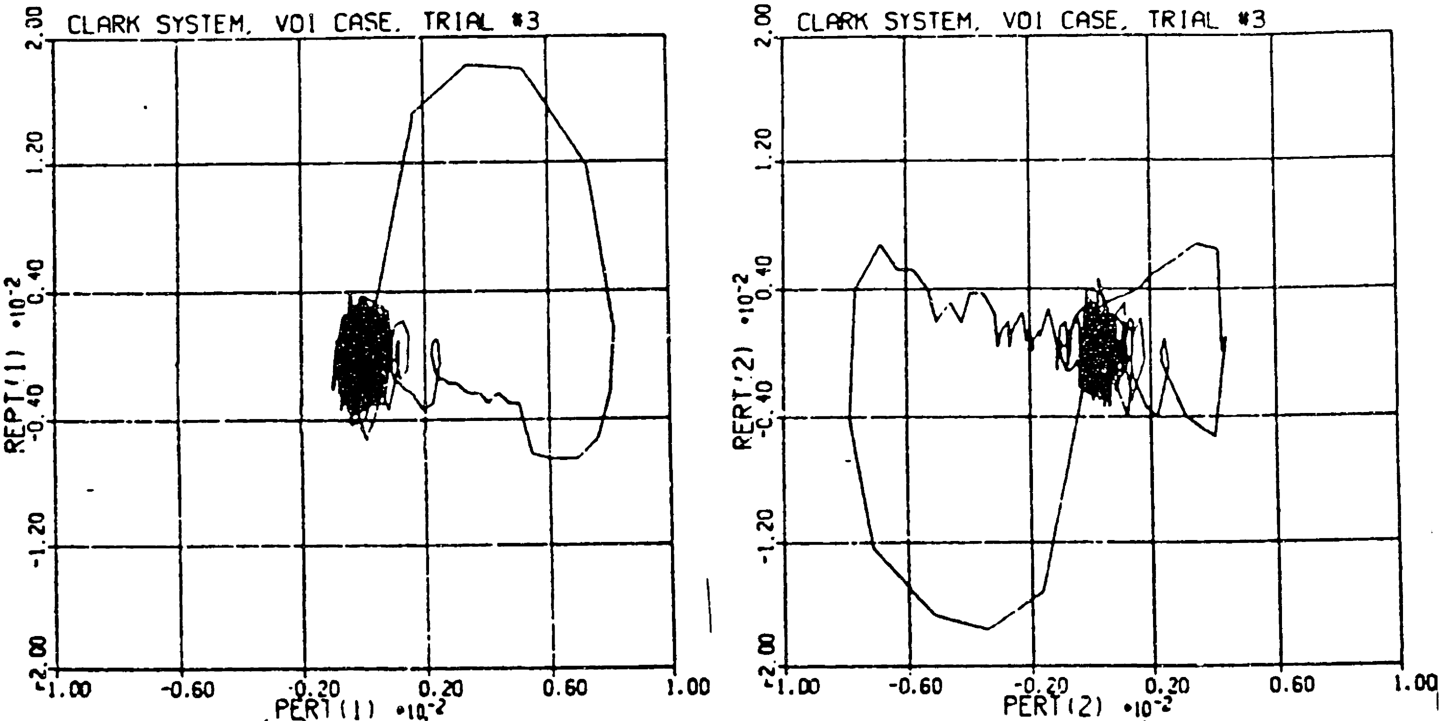
To see if this great performance could be improved any further, the parameter M was adjusted (see Table 7-IV). If M is too low, the system will not have enough control authority near the phase plane origin. If M is too high, there will be too much authority, and extraneous firings will occur. Unfortunately, there is no way of telling in advance which way M should be adjusted. Optimal performance was obtained in Runs 3 and 4 (both were about equally good). Run 3 results are shown in Figure 7-5. Finally, a system has been found that can meet the Chapter 4 design criteria for VOI.

Table 7-IV: Clark VOI Simulation Runs

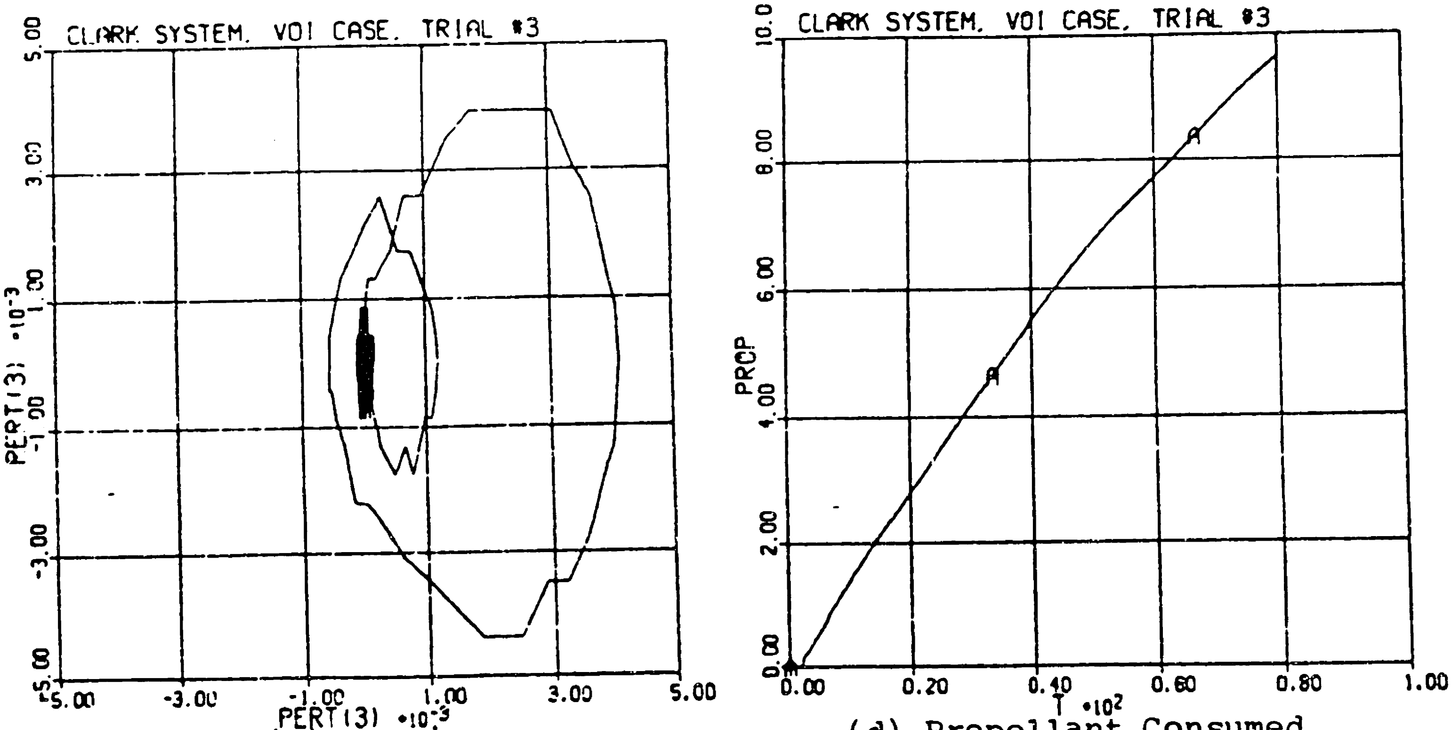
run	comments	maximum attitude (deg)			maximum rate (deg/sec)			fuel (kg)
		x	y	z	x	y	z	
1	analytically determined values	0.511	0.515	0.250	1.15	1.08	0.300	9.65
2	M = 5.0769	0.598	0.535	0.278	1.22	1.18	0.325	9.64
3	M = 6.0	0.467	0.449	0.237	1.04	1.01	0.250	9.64
4	M = 6.6	0.461	0.417	0.216	1.04	1.01	0.250	9.65

It was easy to choose a value for M in VOI. Unfortunately, there was no way of telling whether subsequent tries at a value for M should be increased or decreased from its initial value. But, since M is the only system parameter, only a few runs were necessary to try adjusting it in both directions. The ease of assigning parameter values score for the Clark system in VOI is a 9.

The Clark system is the best observed so far for the VOI case. Attitude errors are low, fuel consumption is low, and, by reducing extraneous firing widths, the Clark system is the only one considered here that has met the Chapter 4 rate error criteria. This system is also easy to design. Despite its questionable value in the normal attitude control case, this system design is exceptional for a VOI-type application.



(a) x-axis (b) y-axis



(c) z-axis (d) Propellant Consumed

Figure 7-5: Optimized Clark VOI Performance

7.2 The Pseudo-Open Loop System

7.2.1 Initial Analysis

The pseudo-open loop system is derived in Appendix C. The only parameter that needs to be chosen is the weighting factor, k_w, in the index:

$$J=\int_0^{t_{final}}(k_w^2+\tau^2)\,dt \tag{7.1}$$

The Chapter 4 design criteria explicitly seek to minimize an equally weighted time-fuel index, implying that k_w should be set equal to τ. Thus, the method of choosing k_w is apparent; the thruster output torque is approximated and k_w is set equal to this value. In VOI, torque differs depending on whether one or two thrusters are firing simultaneously. Initially, a dual thruster firing will be assumed.

Initial k_w values are given in Table 7-V.

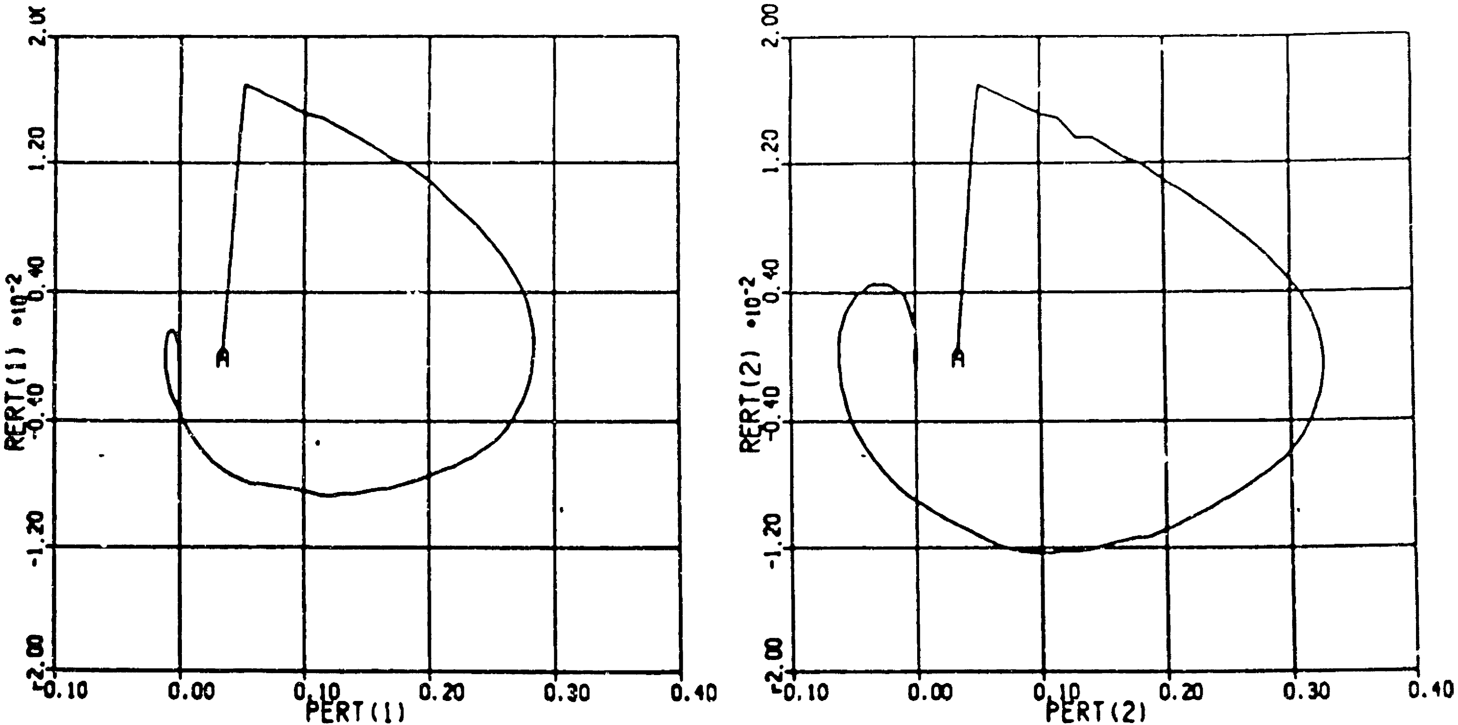
Table 7-V: Initial Pseudo-Open Loop Parameters

parameter	units	normal attitude control X & Y	Z	VOI X & Y	Z
k_w	Nm	1.75	2.56	1843	94.38

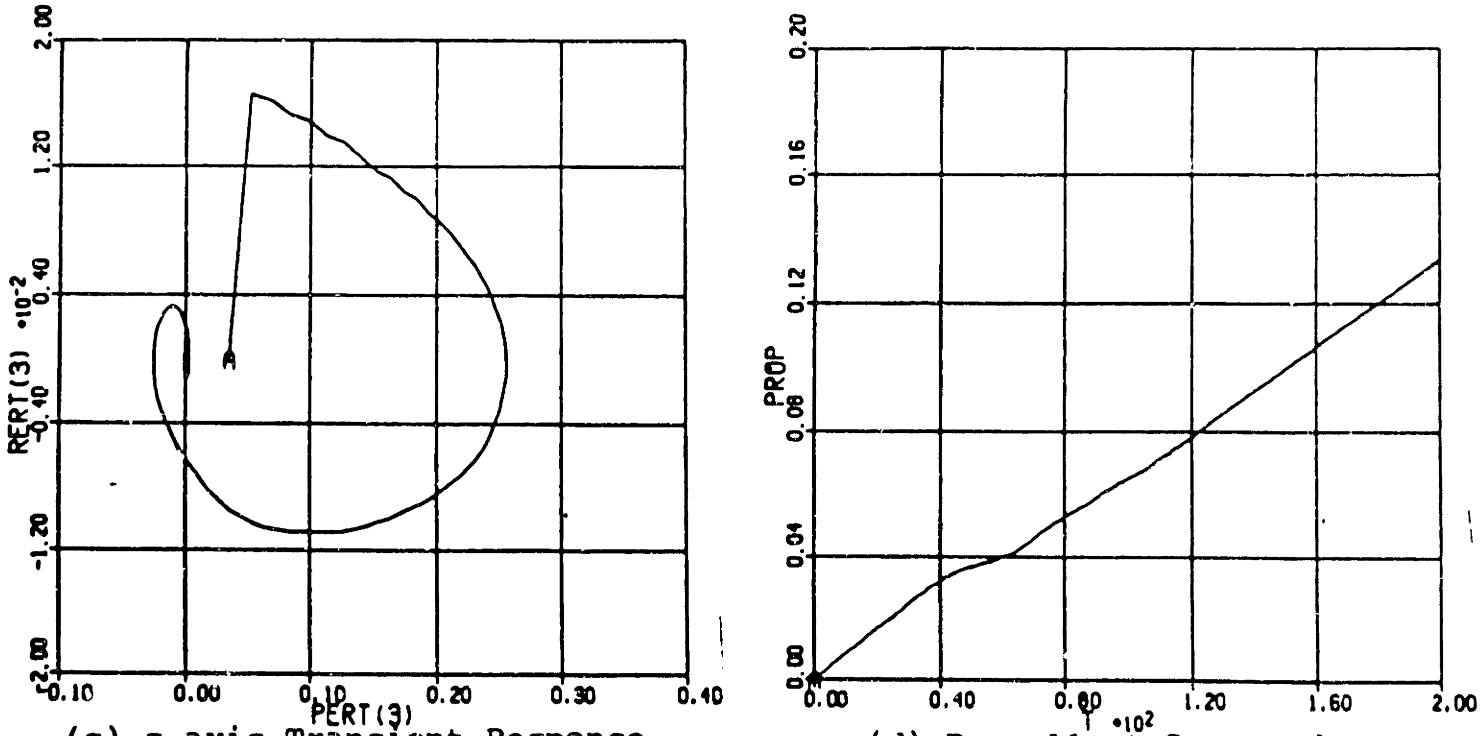
7.2.2 Normal Attitude Control

The k_w value determined above caused reasonably good performance. Results are plotted in Figure 7-6. Limit cycle rates are unusually high, and limit cycle attitudes are unusually low. The major problem in this run is that, as with the Clark system, steady state fuel consumption is very high. Phase planes showing steady state are included in Figure 7-6. This system design appears more jittery in steady state than even the Clark system.

To improve fuel consumption, the time weighting factor should be reduced. A series of runs intended to optimize the weighting factor k_w is shown in Table 7-VI. As this table shows, reduced k_w values resulted in longer settling times and lower transient and steady

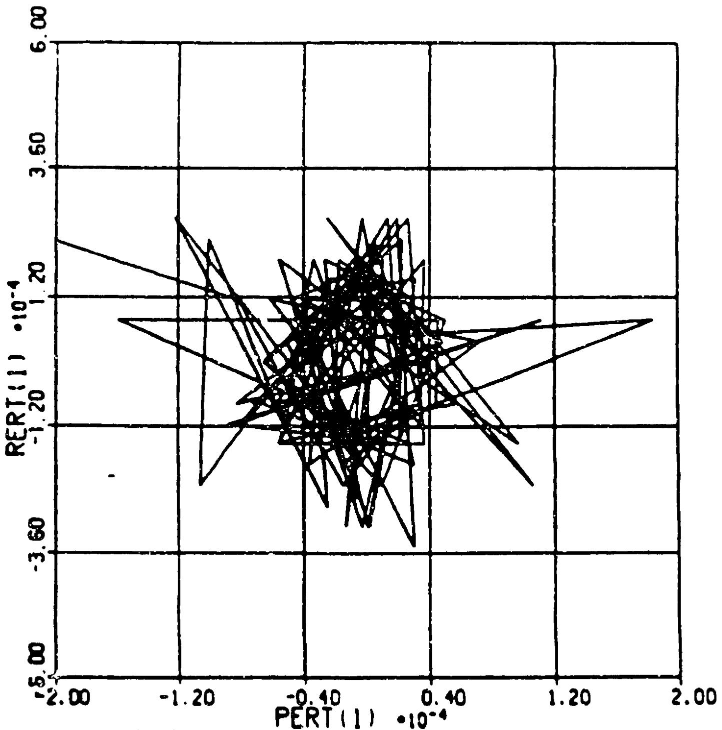


(a) x-axis Transient Response (b) y-axis Transient Response

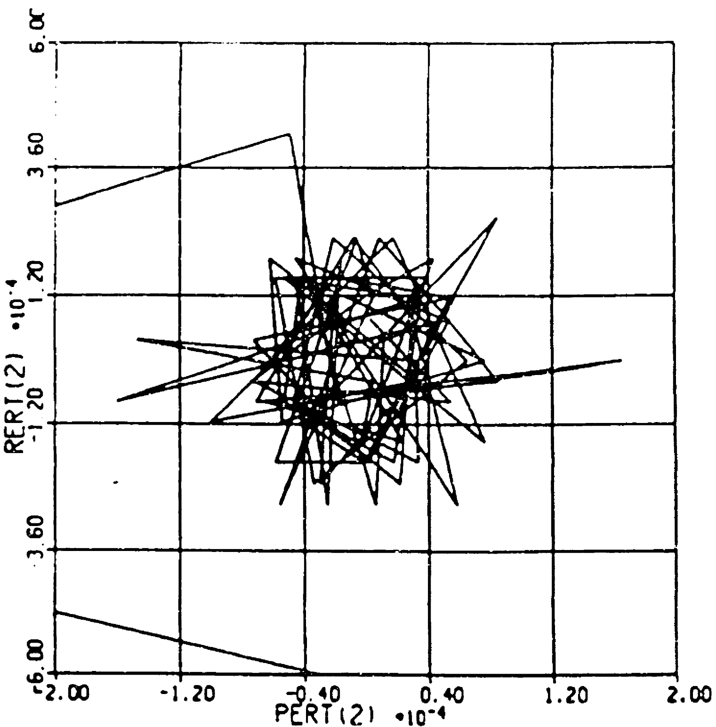


(c) z-axis Transient Response (d) Propellant Consumed

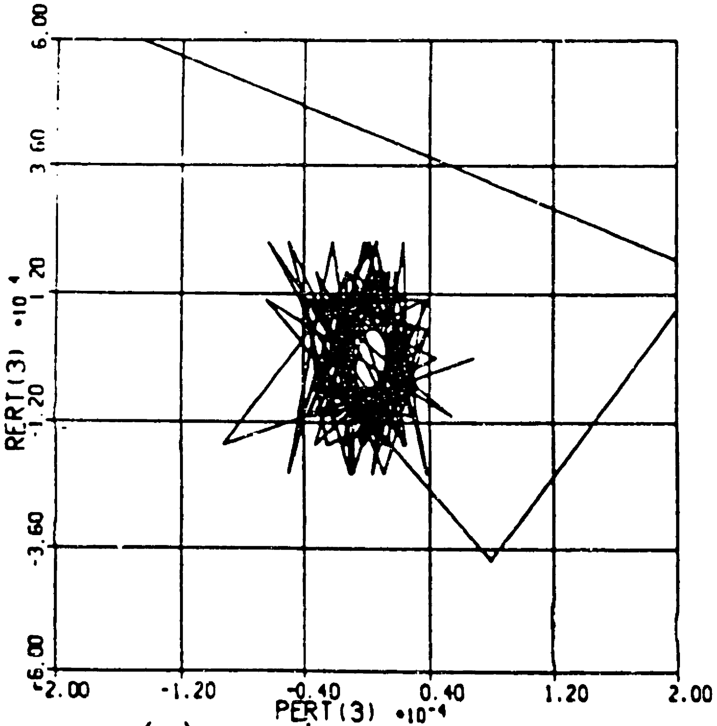
Figure 7-6: Initial Pseudo-Open Loop System Results



(e) x-axis Steady State



(f) y-axis Steady State



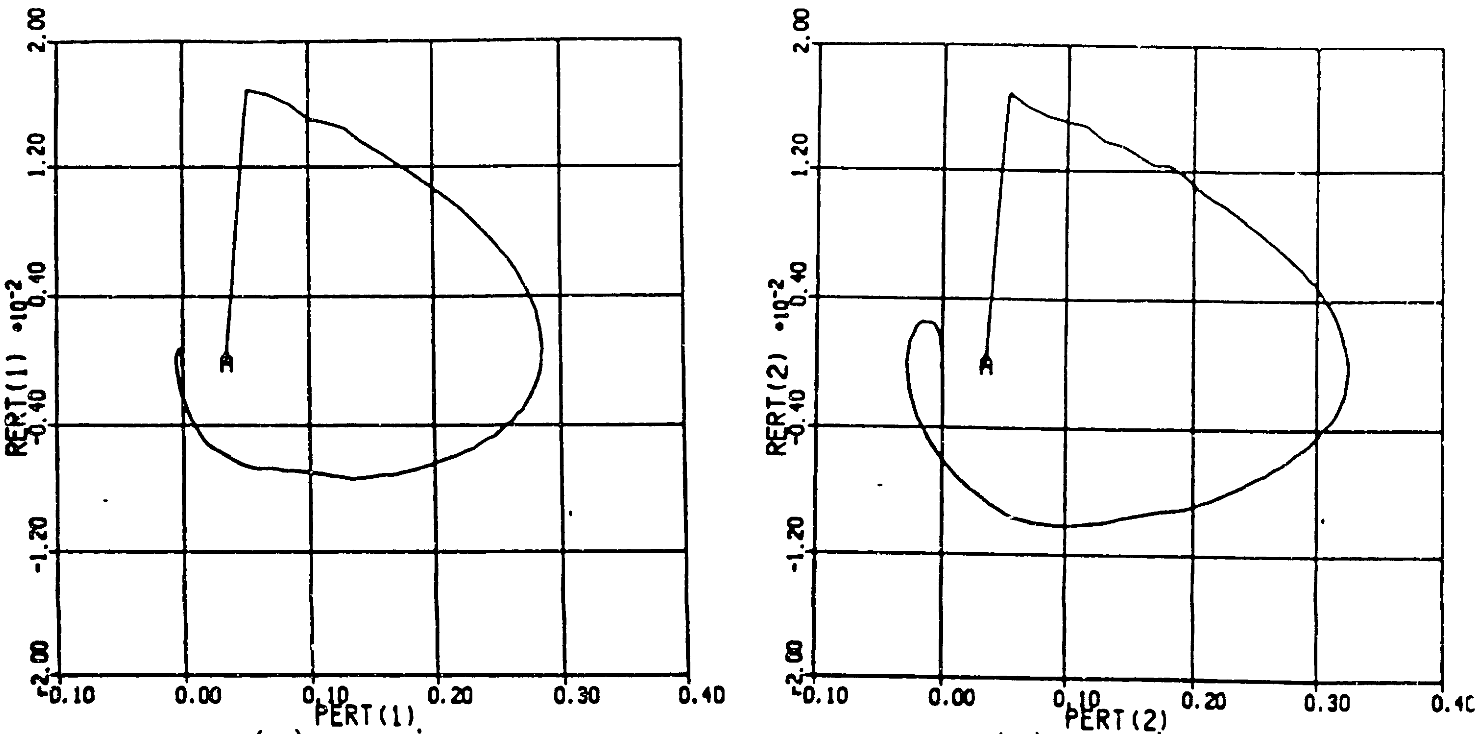
(g) z-axis Steady State

state fuel consumption, just as should be expected. The system reduces transient fuel consumption in two ways, both of which are evident in Run 2 (which, incidentally, is the run in which k_w was optimized). Results of this run are shown in Figure 7-7. As in the MGN phase plane and the RLC, trajectories level out at a certain rate until they are near enough to the origin to trigger an approach parabola. The pseudo-open loop system contains no switching lines to explicitly force this sort of behavior, but the behavior results nonetheless. As the fuel weighting is increased (by decreasing k_w), the level-off rate becomes smaller, thus reducing fuel consumption and increasing settling times. The second way this system conserves fuel is by allowing larger attitude excursions. So far, attitude excursion has not been discussed much. There has never been any real need to do so -- every system discussed so far fired thrusters full on until well after the maximum attitude excursion had been reached. Since these systems all performed identically in this regard, attitude excursions were identical. The pseudo-open loop system, though, exhibits different behavior. In its derivation, the optimal trajectory from an initial state to the origin is used to decide pulse lengths. The system thus foresees the trajectory it will need to follow. If it does not need to initially fire thrusters at their maximum capacity, it does not do so. This causes attitude excursions to increase, but fuel is conserved. Between these two methods of conserving fuel, the system can adjust its performance to obtain a wide variety of transient fuel consumption ratings.

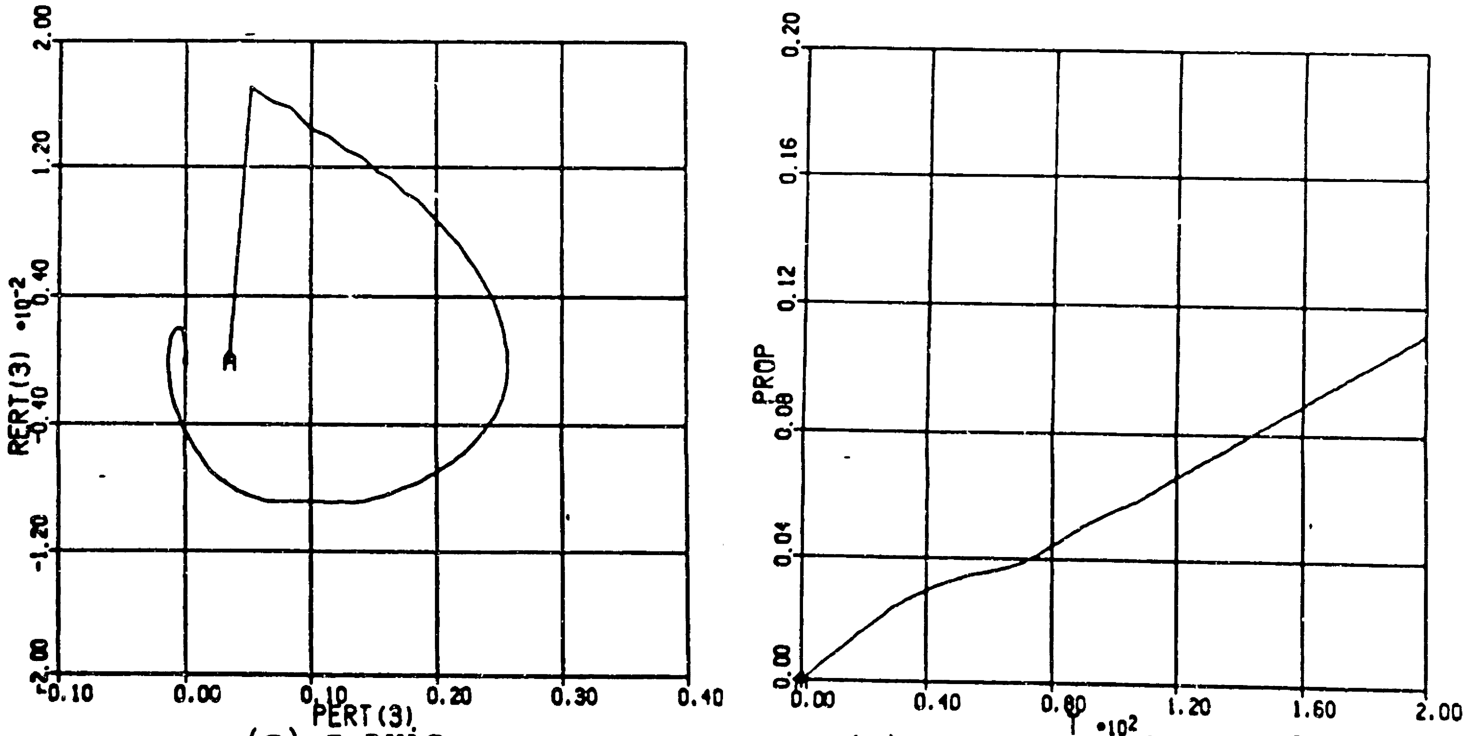
Table 7-VI: Pseudo-Open Loop System Simulation Runs

run	comments	settling time (sec)			trans fuel	J	limit cycle attitude			limit cycle rate			ss fuel
		X	Y	Z			X	Y	Z	X	Y	Z	
1	initially determined values	86	112	86	2.61	442.8	9.97	9.97	9.97	9.69	9.56	9.80	0.04
2	k_w=1.237,1.807	91	110	88	3.13	421.3	9.97	9.99	9.99	9.79	9.82	9.94	0.05
3	k_w=1.010,1.475	99	118	91	3.40	429.8	9.99	9.99	9.99	9.92	9.88	9.91	0.06
4	k_w=0.7139,1.043	139	152	126	3.56	533.3	9.99	10.0	10.0	9.94	9.94	9.94	0.09

Reducing k_w also improved steady state fuel consumption. Unfortunately, the jitter



(a) x-axis (b) y-axis



(c) z-axis (d) Propellant Consumed

Figure 7-7: Optimized Pseudo-Open Loop Performance that is evidently a part of variable pulse width designs allows steady state fuel to improve only very slightly.

k_w was optimized in Run 2 (see Figure 7-7). Not even this optimal run could produce good steady state fuel ratings. In addition, the Figure 7-7 phase planes show that the approach parabola was triggered too late, causing overshoot. This was observed in every run. The only variable parameter is k_w, and adjustment of k_w seems unable to abolish this overshoot. The best J value achieved was J = 421 -- better than that achieved by the Galileo controller, but worse than J's realized by the phase plane designs. This system does, however, have one real advantage over every other system considered so far; the constant steady state thruster firings result in the best steady state limit cycle attitudes observed. Run 2 performance is summarized in Table 7-VII.

Another advantage of this design is that it is extremely easy to assign a value to the parameter k_w. The initial choice of k_w was unusually good, and the system was optimized on the second try. Adjusting k_w is very intuitive -- increasing it causes shorter settling times and higher fuel consumption, while reducing it has the opposite effect. Thus, the ease of assigning parameter values score for this system in the normal attitude control case is a 10.

So the pseudo-open loop system seems difficult to classify. It is easy to design, has a transient response that is neither exceptionally good nor exceptionally bad, causes excellent limit cycle attitudes, but does so at the expense of a very high rate of steady state fuel consumption. In some applications it may be desirable to use this system, but for most, it is probably not the best choice.

7.2.3 VOI

Results of a simulation of VOI run with the k_w values determined above are shown in Figure 7-8. Attitude errors are acceptable (z-axis attitude errors are very small), as is fuel

Table 7-VII: Summary of Pseudo-Open Loop Scores

Transient Scores

	X	Y	Z	AVG
settling time	6.59	5.45	6.82	6.29
fuel consumption				3.13
rate excursion	10.0	10.0	10.0	10.0
attitude excursion	3.35	1.33	4.63	3.11

J = 421.3

Steady State Scores

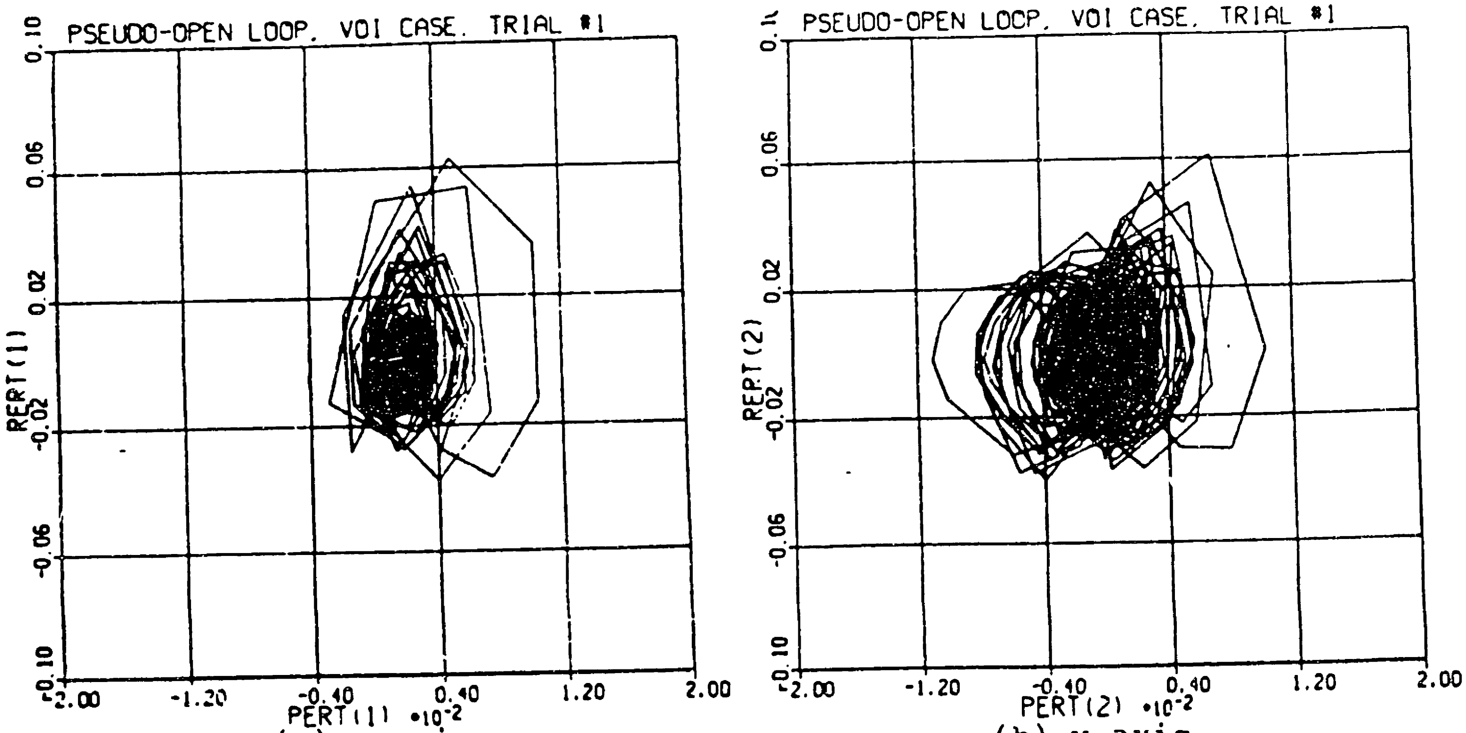
	X	Y	Z	AVG
fuel consumption				0.05
limit cycle rate	9.79	9.82	9.94	9.85
limit cycle attitude	9.97	9.99	9.99	9.98

Overall

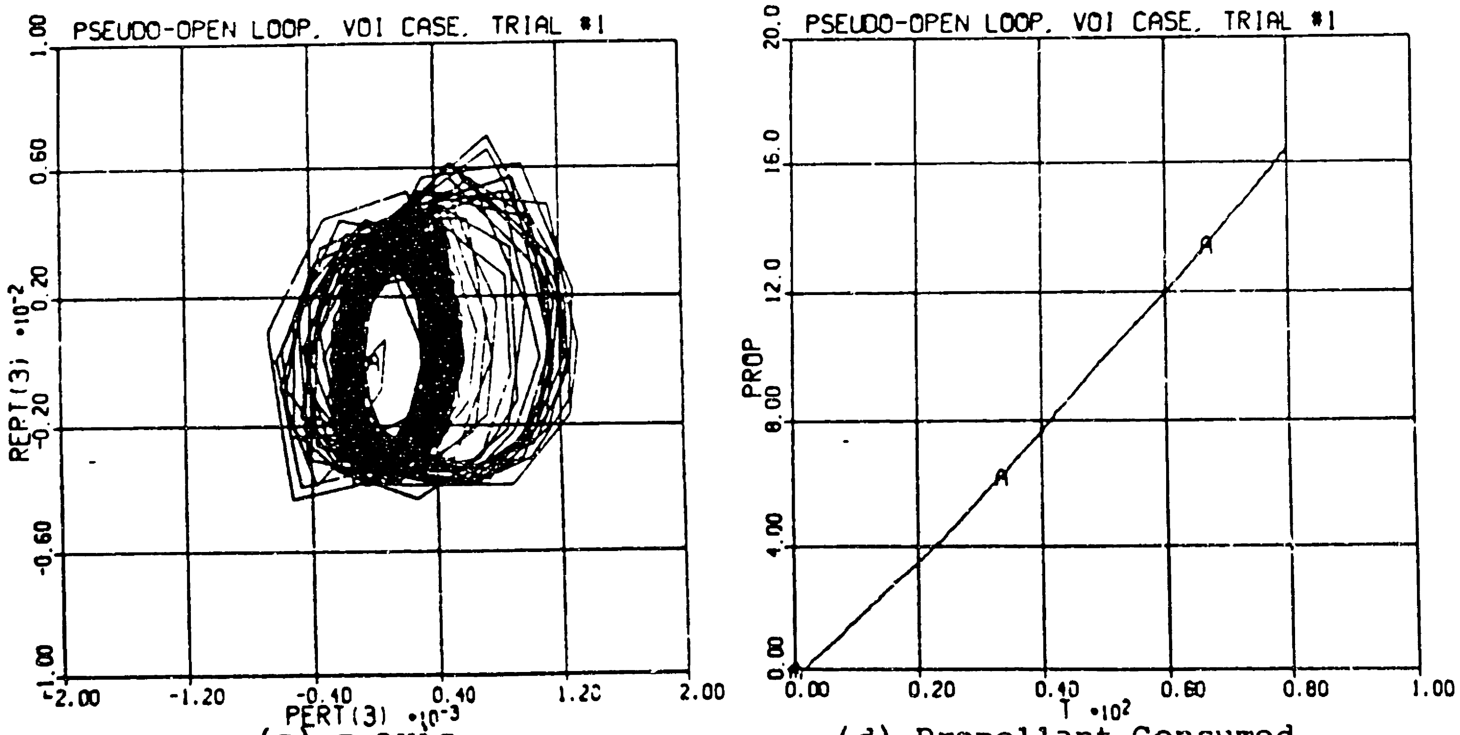
	X	Y	Z	AVG
ease of assigning parameter values				10

consumption. As in the Galileo controller, attitudes are centered on the origin. Rates, though not as high as usual on a first run, are unacceptably large. This may result from the fact that a large number of 66 msec pulses are commanded. It was observed with the Clark system that these long pulses are not really necessary.

The parameter tuning process is outlined in Table 7-VIII. There is a tradeoff



(a) x-axis (b) y-axis

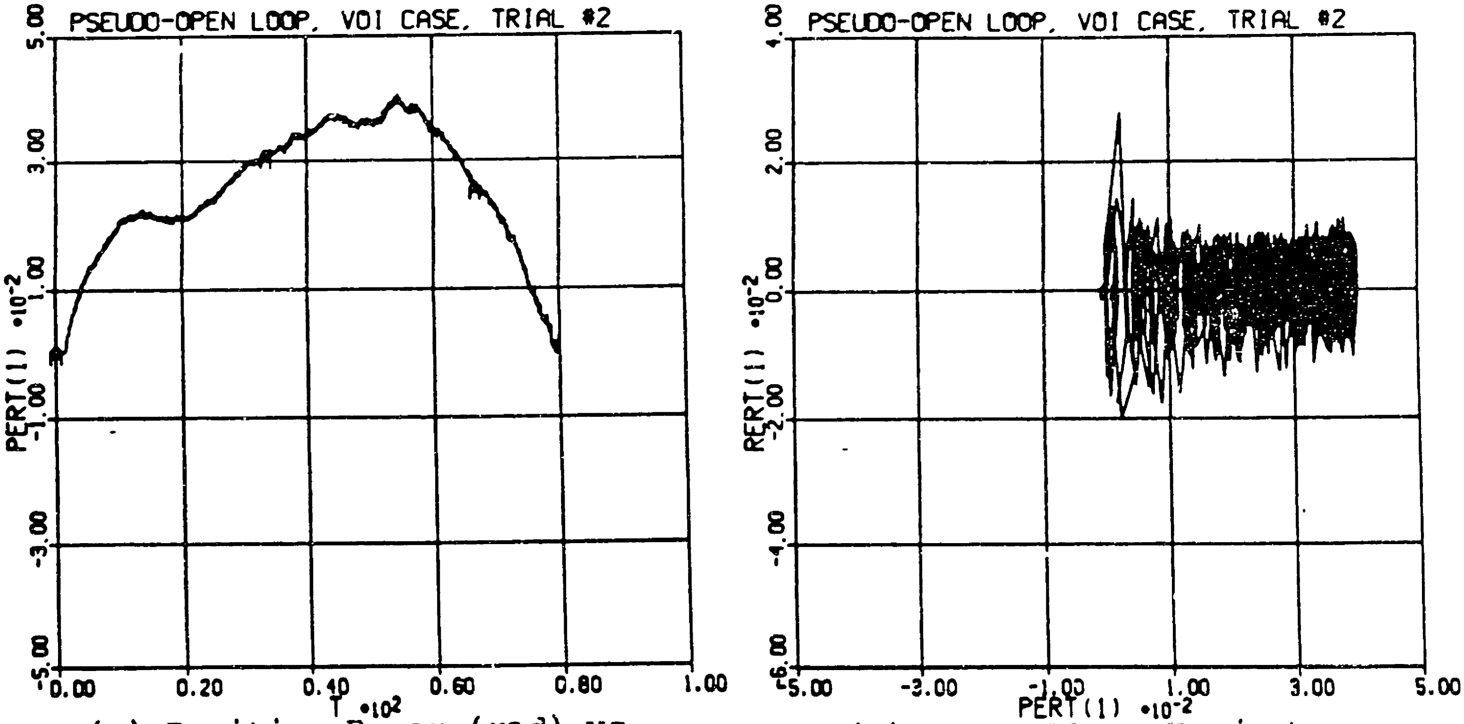


(c) z-axis (d) Propellant Consumed

Figure 7-8: Pseudo-Open Loop Initial VOI Results involved in the optimization of k_w. If k_w is too large, excessive numbers of 66 msec pulses cause large rate errors, as was observed in the first run. To reduce rate errors, k_w should be reduced. But if it is reduced too far, another problem develops. Control authority is reduced in an attempt to save fuel, and disturbances are allowed to cause very large attitude errors. Thus, attitude behavior such as that illustrated in Figure 7-9 results. Control torques do not act long enough to zero out attitude errors, so large attitude errors build up. This explains the 2.32 degree x-axis attitude error observed in Run 2.

Table 7-VIII: Pseudo-Open Loop VOI Simulation Runs

run	comments	maximum attitude (deg)			maximum rate (deg/sec)			fuel (lb)
		x	y	z	x	y	z	
1	initially chosen values	0.600	0.629	0.076	3.61	3.50	0.400	16.49
2	k_w=1303,66.74	2.32	0.710	0.075	1.59	2.00	0.275	11.45
3	k_w=1505,77.06	0.388	0.470	0.101	1.98	2.19	0.350	13.07



(a) Position Error (rad) vs. Time (sec)

(b) Phase Plane Trajectory

Figure 7-9: Pseudo-Open Loop Behavior with k_w too Small

k_w was optimized in Run 3. Results of this run are shown in Figure 7-10. Just as, in the normal attitude control case, the pseudo-open loop system was able to cause extremely low limit cycle attitudes, VOI attitudes were also kept very low. In fact, they are among the lowest observed in this thesis, especially about the z-axis. Unfortunately, rate errors are among the worst optimized rate errors observed. The only design with worse VOI rate errors was the RLC. The tradeoff described above does not permit further reduction of rate errors without substantially penalizing attitude errors.

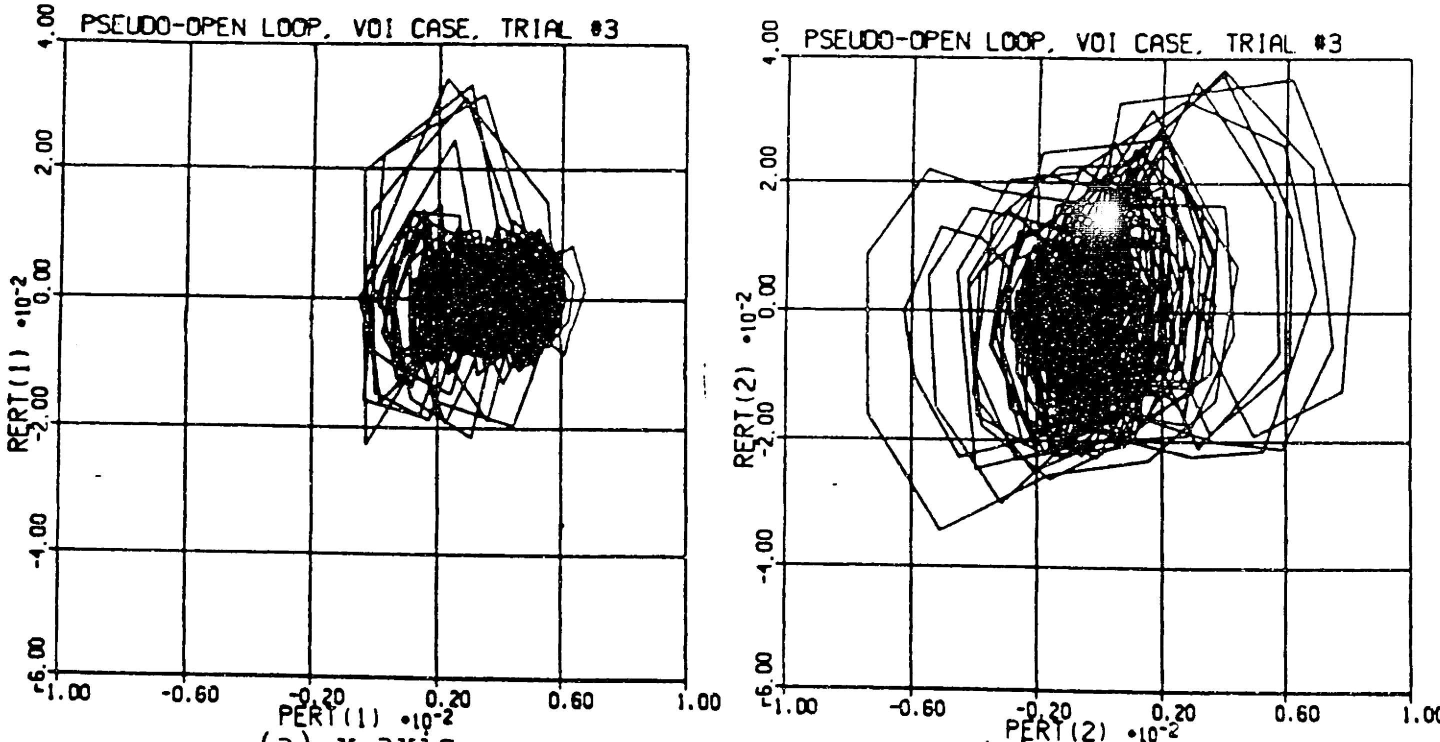
As in the normal attitude control case, this system was very easy to design in VOI. The initial parameter choice performed reasonably well, parameter adjustment was intuitive, and there was only one independent parameter. The ease of assigning parameter values score for this system in VOI is a 10.

The pseudo-open loop system would be excellent in a VOI-like application where rate errors were not very important. Attitude errors are kept quite low and are centered on the origin, and the system is very easy to design. However, rate errors are unacceptable (though not the worst observed in this thesis). The worth of this system would be strongly dependent on the application.

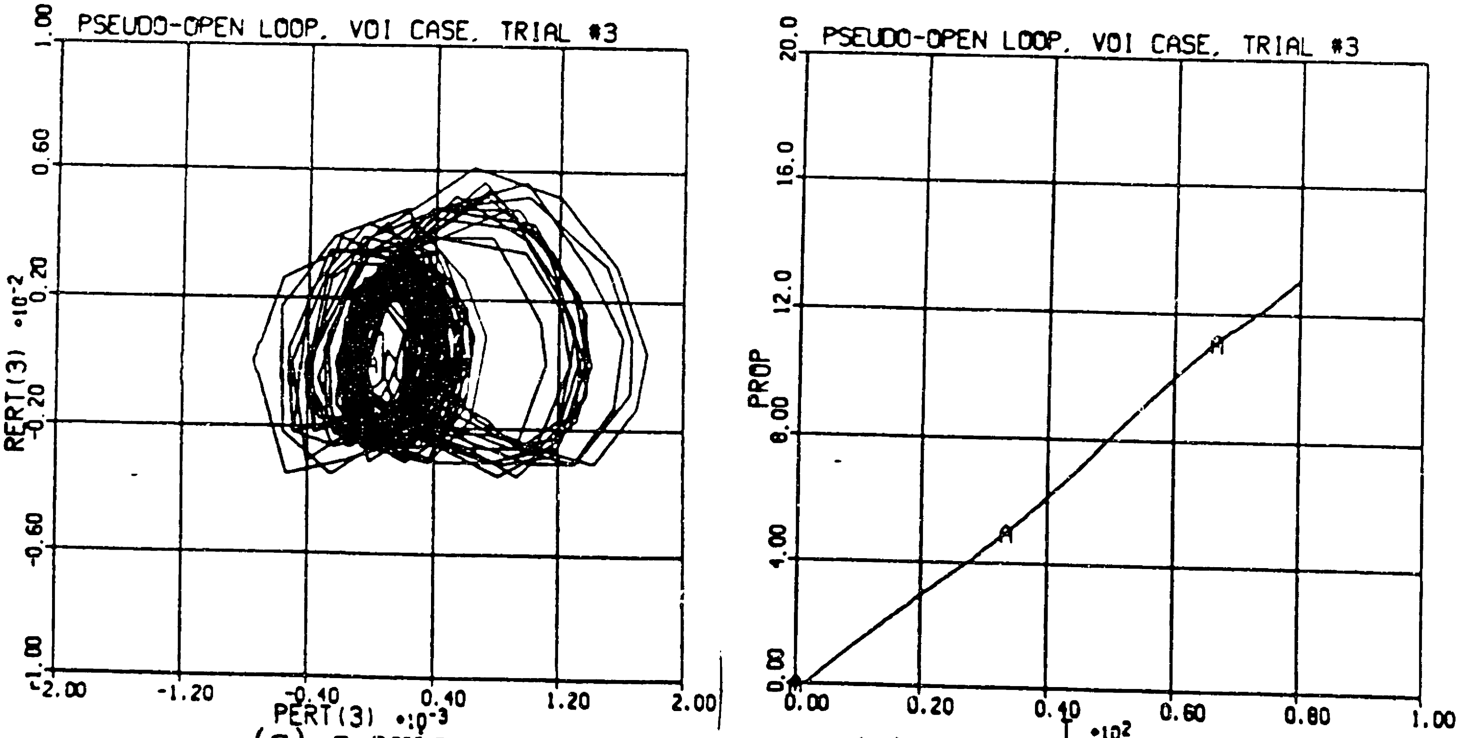
7.3 Generalizations About Variable Pulse Width Designs

Once again, generalizations about this approach to the design of attitude control systems are difficult to make. The Clark system and the pseudo-open loop system were both easy to choose parameter values for, but it is possible to conceive of variable pulse width designs that would be more complicated. Similarly, generalizations about performance are difficult, especially since the Clark system and the pseudo-open loop system performed very differently.

The one way in which these systems performed similarly was that they both caused very high steady state fuel consumption. System nonlinearities and inter-axis coupling



(a) x-axis (b) y-axis



(c) z-axis (d) Propellant Consumed

Figure 7-10: Optimized Pseudo-Open Loop VOI Performance torques probably make this problem unavoidable for any variable pulse width design. Any nonzero state causes a thruster firing, and nonlinearities and coupling make perfect control to a zero state impossible. Variable pulse width designs might be improved by the addition of an explicit deadzone around the origin. Such a deadzone would eliminate this constant thruster firing, and thus would help reduce steady state fuel consumption.

Apart from the fuel problems, though, these designs seem to have some potential. Their performance is not unreasonable, and the performance of the Clark system during VOI is exceptional. Indeed, since variable pulse width designs are unique in their effectiveness at minimizing extraneous thruster firings, they may be the only designs that can reduce VOI rate errors to within acceptable limits.

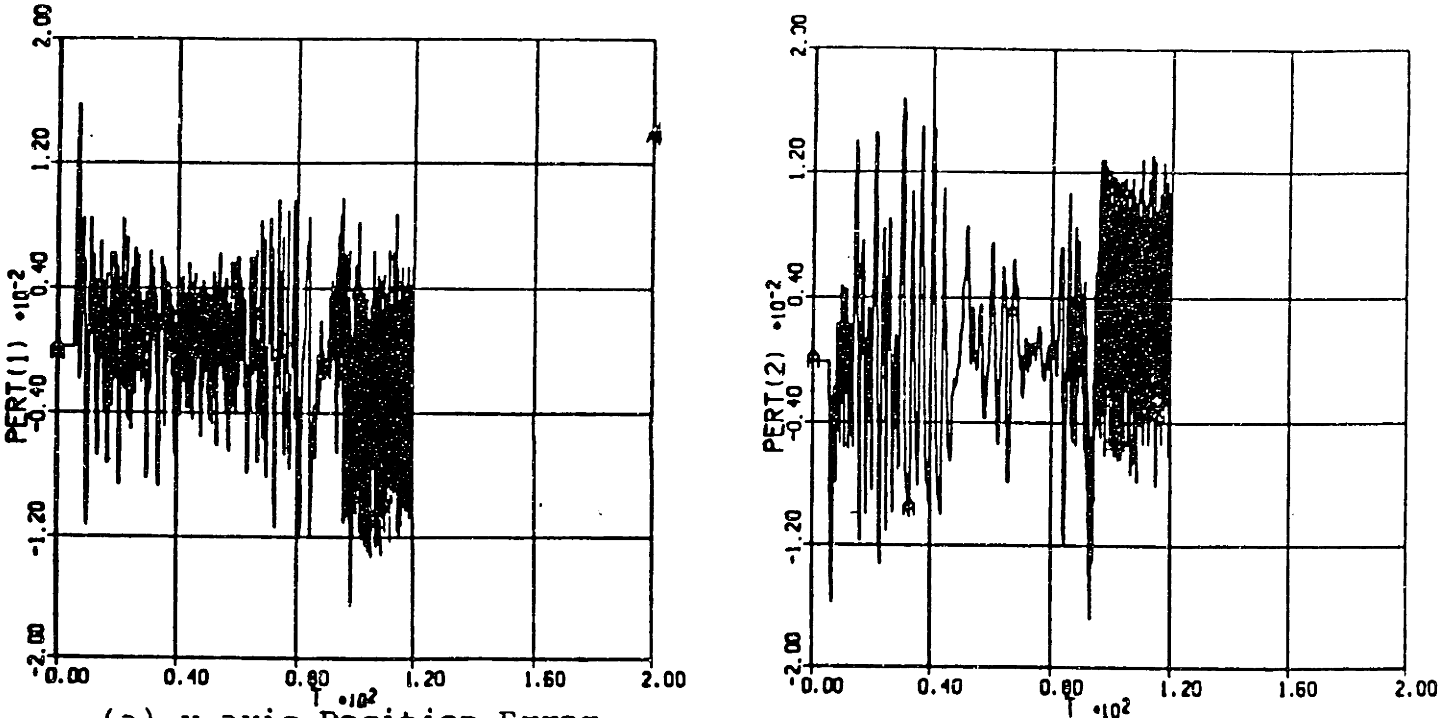
Chapter 8

Practical Applications

The findings of this thesis have already been put to practical use by the analysts working on the Magellan spacecraft. The parameters found to optimize the MGN phase plane in the normal attitude control case (Chapter 5) will actually be used during Venus mapping. Simple concepts like the "approach parabola" now help to optimize the phase plane for other situations. But the findings of this thesis were enormously important in one specific problem that arose with the Magellan mass expulsion control system.

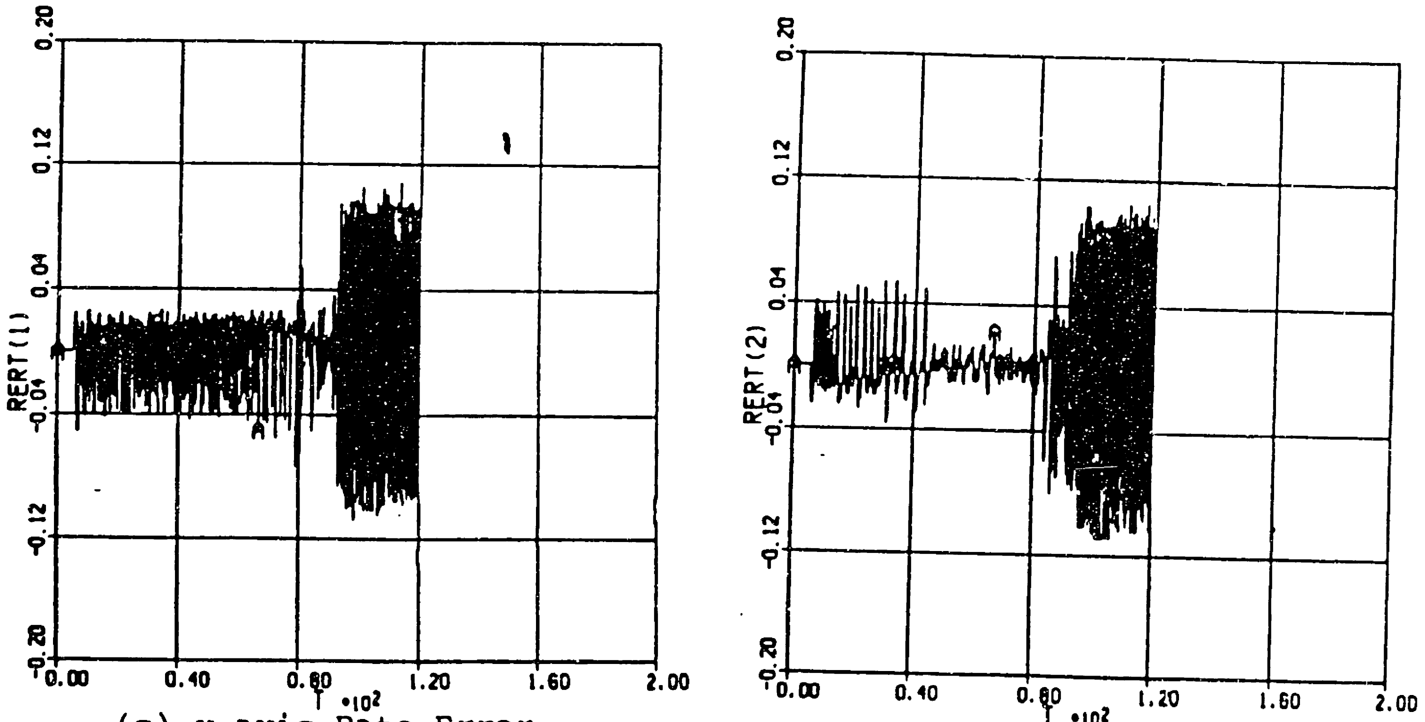
When VOI was considered in previous chapters, simulation runs were ended just as the SRM burn was terminating. This was done for simplicity -- a system that performs well in a high disturbance environment might perform poorly when the burn ends and disturbances vanish. In such a situation, the large control authority necessary at the start of VOI might be excessive. And this was indeed observed to be a problem. Simulations of VOI showed that this excessive control authority caused limit cycles of unacceptably large amplitude and high frequency. This is demonstrated in Figure 8-1. Post SRM-burn angular rates oscillate between about plus and minus 6.25 deg/sec, causing large attitude errors also. As is evident in the figure, this also causes a high rate of hydrazine propellant consumption. The control system needed to be adjusted such that the large amplitude limit cycles did not occur.

The Magellan mass expulsion control system is slightly more complicated than just the MGN phase plane discussed in Chapter 5. A controller processes ϕ and $\dot{\phi}$ for input into the phase plane. This controller is shown in Figure 8-2 (the phase plane itself is shown in Figure 3-16c). Except for the pseudo-acceleration term, all the various controller terms were discussed in connection with the Galileo controller (Chapter 6). The pseudo-



(a) x-axis Position Error

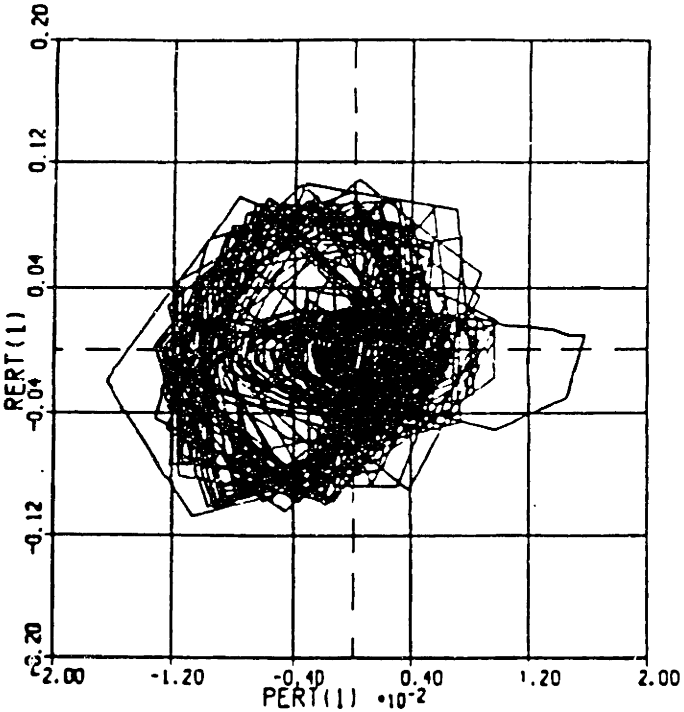
(b) y-axis Position Error



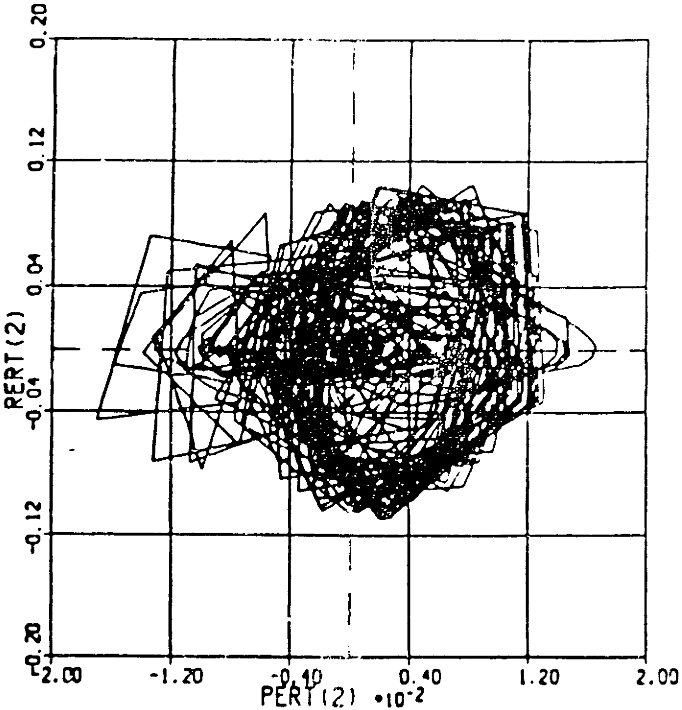
(c) x-axis Rate Error

(d) y-axis Rate Error

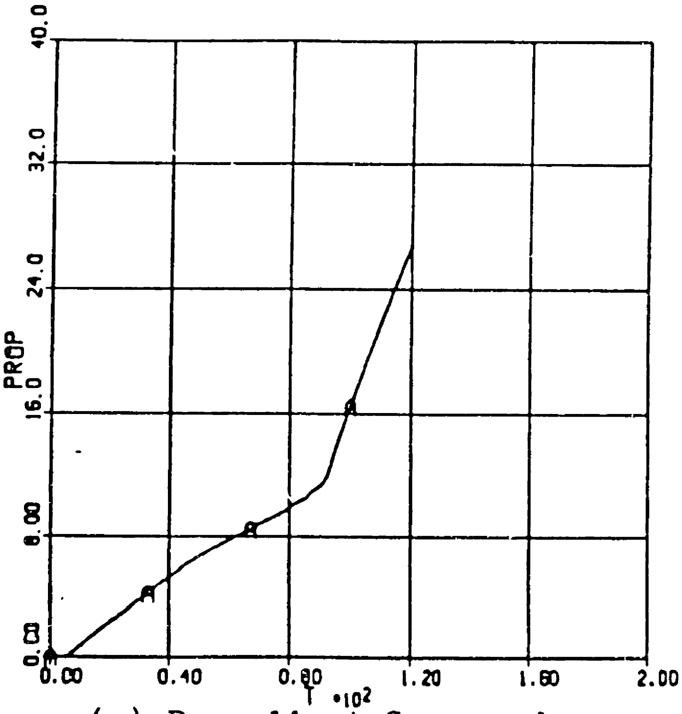
Figure 8-1: Post-VOI Large Amplitude Limit Cycles



(e) x-axis Phase Plane



(f) y-axis Phase Plane



(g) Propellant Consumed

acceleration (PSA) term requires some explanation. It is identical in form to a pseudo-rate term. In the discussion of pseudo-rate in Chapter 6, it was noted that a PSR term can be viewed as simple acceleration passed through a low pass filter. When viewed in this way, the utility of the PSA term becomes apparent. The term provides approximate derivative feedback for the rate signal, and should function similarly to PSR control. However, unlike the PSR term considered earlier, the Magellan PSA term has only one time constant, and not separate ones for charging and discharging.

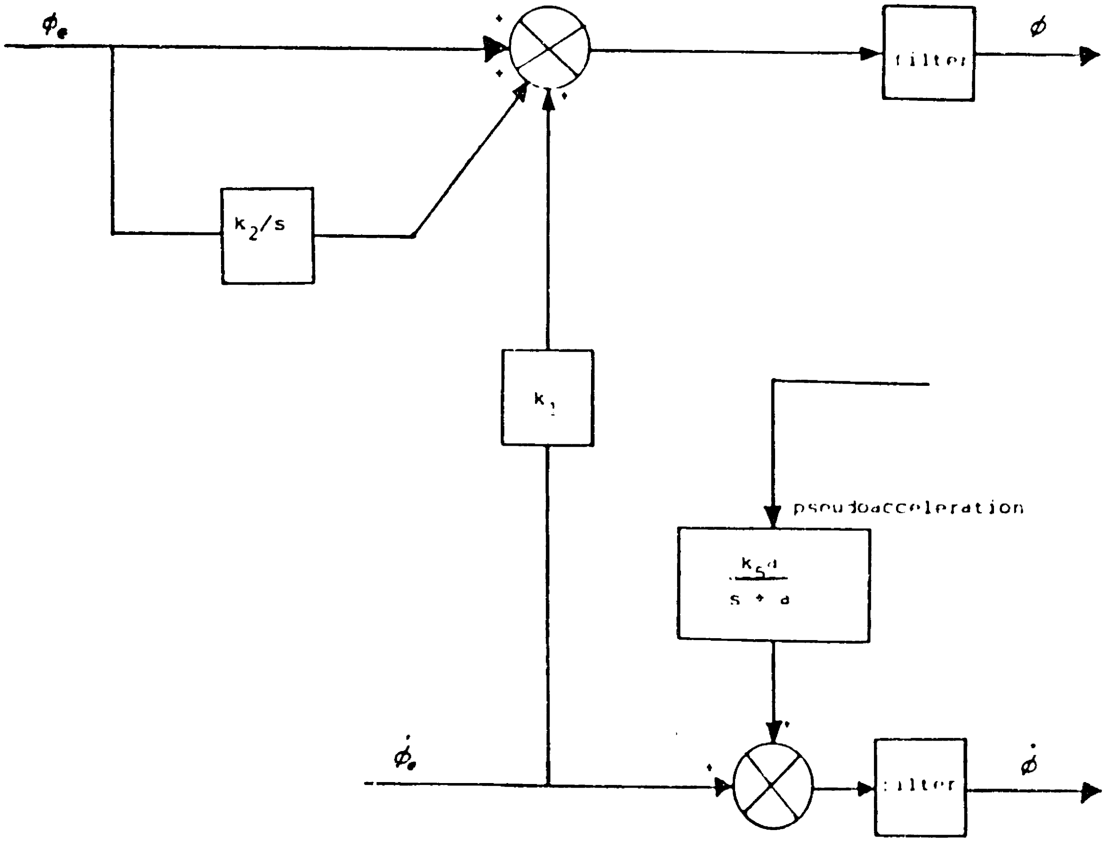


Figure 8-2: The Magellan Controller

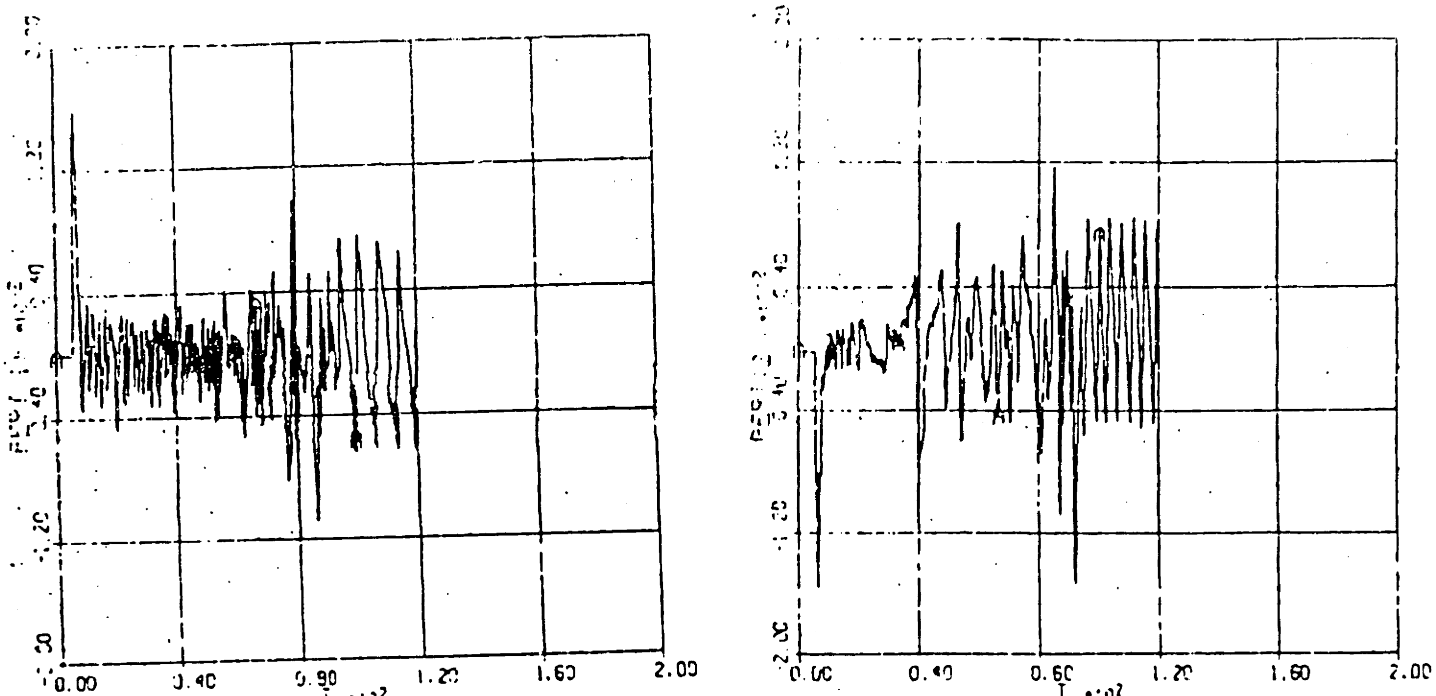
One of the most important considerations during VOI is the elimination of extraneous thruster firings. This consideration is important in this case also. Reducing extraneous region 3 firings will reduce control authority, which is certainly desirable. The large amplitude limit cycle problem, then, is remarkably similar to the VOI design problem that has been considered throughout this thesis.

Three ways of reducing the likelihood of unnecessary region 3 firings in the post-disturbance environment, suggested by the results of Chapters 5 and 6, were found to be successful in computer simulations:

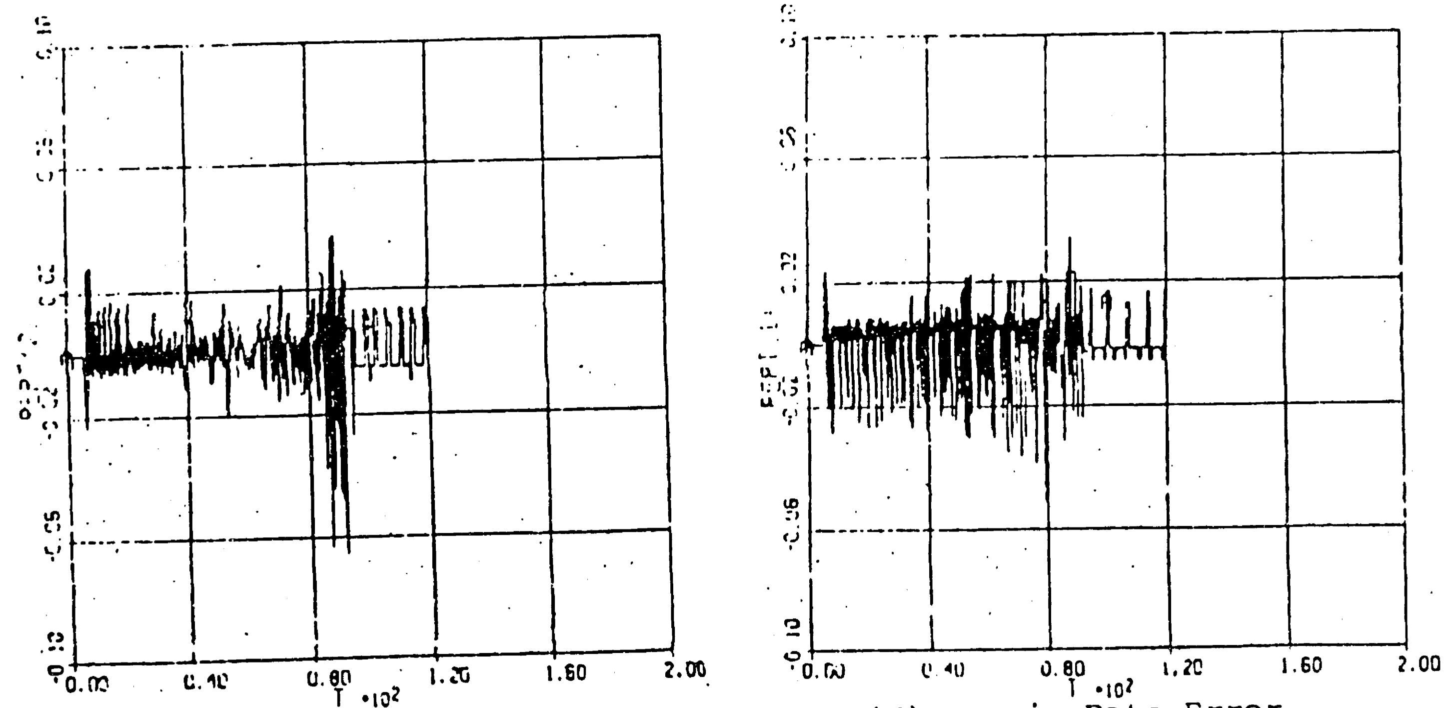
1. In Chapter 5, the optimal value of the rate ledge R1 was found to be zero. By maximizing the separation between the +3 and -3 regions, this R1 value makes it more difficult for thrusters to push a trajectory from one of these regions to another. Thus, extraneous thruster firings are made less likely.
2. Raising the value of the rate ledge R3 (from R3 = 0.04 rad/sec to R3 = 0.08 rad/sec) also helps reduce extraneous region 3 firings, for similar reasons.
3. Once a thruster firing has occurred, a PSA (or PSR) term acts immediately to reduce the phase plane rate error input, thus tending to inhibit another thruster firing. But the Magellan's PSA time constant, τ_{psa}, was originally set to 214 msec, or about 6.5 sampling intervals. Thus, the PSA term built up too slowly to immediately start inhibiting pulses. A time constant more on the order of a single sampling interval would facilitate the immediate inhibition of unnecessary thruster firings. In simulations, optimal performance was observed with τ_{psa}= 44 msec.

Although all of these measures solved the problem individually, it was decided to adopt all three for use aboard the Magellan. The results of simultaneously implementing these measures are shown in Figure 8-3. Compared to Figure 8-1, this simulation shows reduced rate and attitude errors all through VOI. In addition, the large amplitude limit cycles are no longer a problem. After the SRM burn, the rate of propellant consumption no longer increases drastically -- in fact, it is almost zero. Further simulation tests verified that the modified control system performed well. Not only has the limit cycle problem been solved, but overall system performance has improved significantly. A summary of parameter changes that were necessary to accomplish this are given in Table 8-I.

Thus, parts of this thesis have already proven themselves useful. It seems reasonable to believe that this study will continue to be of use to attitude control system designers in the future.

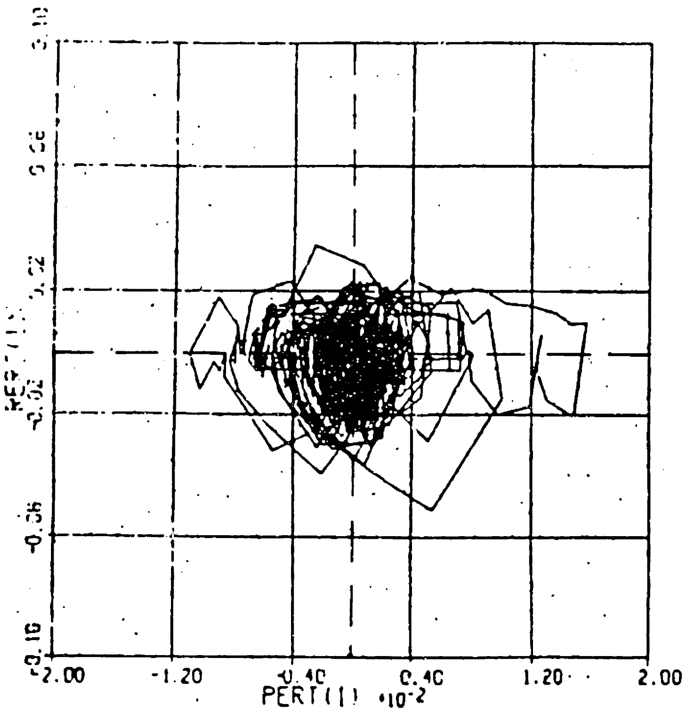


(a) x-axis Position Error (b) y-axis Position Error

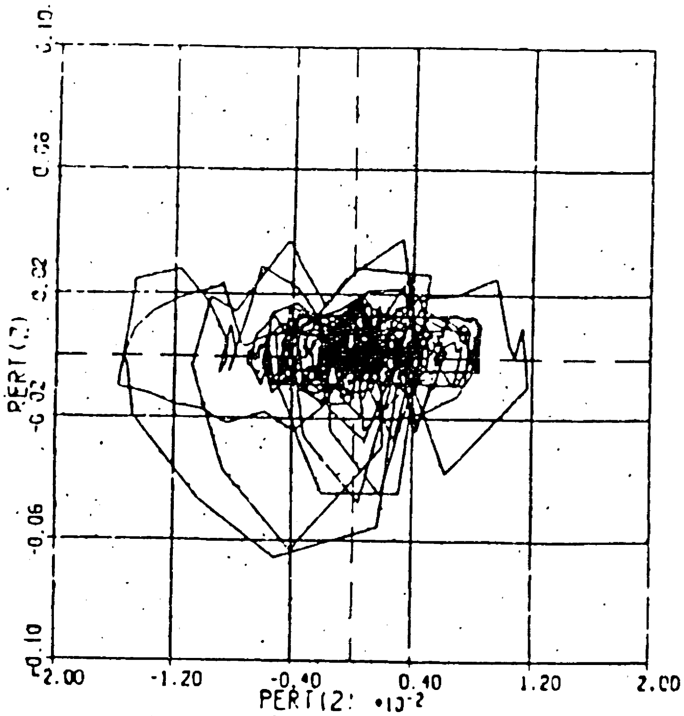


(c) x-axis Rate Error (d) y-axis Rate Error

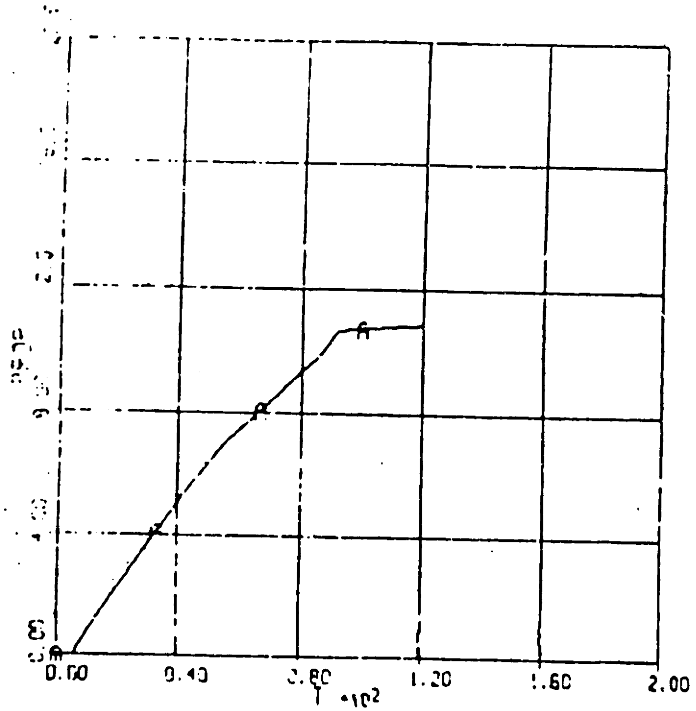
Figure 8-3: MGN Behavior Once the Problem was Solved



(e) x-axis Phase Plane



(f) y-axis Phase Plane



(g) Propellant Consumed

Table 8-I: Parameters Solving the Limit Cycle Problem

parameter	units	old value	new value
R1	rad/sec	-0.025	0.00
R3	rad/sec	0.040	0.080
τ_{psa}	msec	214	44

Chapter 9

Conclusions and Recommendations for Future Work

9.1 Conclusions

Six different mass expulsion attitude control algorithms, representing three different approaches to the control task, have now been considered. By designing and carefully optimizing each algorithm, a comparison of these three approaches in both the normal attitude control case and in VOI has been made possible.

The two most important design considerations in the normal attitude control case are the "steady state fuel bump" and the elimination of undershoot or overshoot of the phase plane origin during transient response. No analytical method of predicting the deadband width that optimizes system performance in the presence of the "fuel bump" phenomenon was discovered. Thus, finding the location of the "bump" should be about equally difficult for every attitude control system. Systems which permit short thruster pulses, however, were observed to have "fuel bumps" at narrower deadband widths, thus allowing smaller maximum limit cycle attitudes in steady state. The variable pulse width designs were therefore found to result in the smallest limit cycle attitudes. Unfortunately, the jittery nature of these algorithms in steady state causes a rate of steady state fuel consumption much greater than the rate of consumption found with other systems. Other system designs allow acceptable performance for both limit cycle attitudes and steady state fuel consumption.

The phase plane designs clearly dominated in normal attitude control transient response. The worst of these designs resulted in optimized J values of less than 345, while the non-phase plane design that scored the best had a J value of about 420. In a phase plane, the designer has explicit control over when and how the "approach parabola" is triggered. Fine-tuning of the approach parabola is very important in the optimization of system response. By the simple geometric consideration of this parabola, a phase plane, especially a phase plane with a parabolic switching line, can be quickly optimized for transient response. This optimization is not so clear cut for other system designs. By viewing a controller's proportional-derivative control as a phase plane's slanted deadband, some of these advantages can be brought to the controller approach. However, parabolic switching lines are superior to linear ones. Phase planes thus have the potential to outperform controllers in this respect. Also, specialized ledges can easily be added to a phase plane to improve its performance in other ways. A phase plane with a parabolic switching line and short pulse regions to improve steady state performance should thus be expected to perform the best in the normal attitude control case. And indeed, the Magellan phase plane was observed to perform exceptionally well.

In VOI, one-sided deadbands and the elimination of extraneous thruster firings were the primary design considerations. The Galileo controller's integral and pseudo-rate terms were found to be most effective at this. In addition, a system including short thruster pulses can further reduce extraneous thruster firings. The Clark system, which couples a controller with a variable pulse width mechanism, was thus observed to perform best.

The conclusions reached above should not be taken too generally. It is possible, for example, that the addition of some extra terms in the Galileo controller might improve that system's normal attitude control performance in some unexpected way. Similarly, a phase plane geometry that performs well in VOI could conceivably be developed. However, the designs considered in this thesis, most of them chosen because they are typical of designs found in the literature, show no sign of such behavior.

So the best system in the normal attitude control case is the MGN phase plane. In VOI, the best is a controller (with at least integral and pseudo-rate control) whose output is fed into a variable pulse width scheme. It is interesting to note that this is the control scheme currently used on the Magellan spacecraft; for normal attitude control, the MGN phase plane is used, while, during VOI, a controller's output is fed into the MGN phase plane, which, with its short pulse regions. acts at least somewhat like a variable pulse width system. It is ironic that this study, which was initially undertaken to improve upon Magellan's attitude control system, has found that the system currently used on Magellan is one of the better systems the spacecraft could have. Initial complaints about the system's complexity are still justified, though. Observations of "fuel bumps", approach parabolas, and one-sided deadbands have made the design task easier, but this combined controller-phase plane system is still difficult to optimize. Thus, a different control scheme might be desired for a spacecraft with less stringent pointing requirements but stricter requirements on an analyst's time, a company's money, or the available room in the spacecraft's on-board computer (because the more complex the system, the more computer time and memory must go to the implementation of it). This thesis should be of use in the choice of a control algorithm should this be the case.

9.2 Recommendations for Future Work

Not unexpectedly, this thesis has raised as many questions as it has answered. Thus, many recommendations for future work are appropriate. These recommendations fall into two categories -- general recommendations, and recommendations concerning the design of a particular control system.

There are two general recommendations. The first concerns the "steady state fuel bump" phenomenon. Although this phenomenon was observed and explained, no obvious analytical technique of approximating the location of the "bump" could be found. Such an analysis is difficult because the "fuel bump" involves inter-axis coupling in Euler's rigid body equations of motion, and is thus nonlinear. But such an analysis could be a great aid to a control system designer.

Second, the equivalence between a proportional-derivative controller and a slanted deadband phase plane was noted. Such analogies should exist for any controller term or phase plane switching line. An integral term, for example, might be analogous to a switching line that moves through the phase plane in a certain way. It would be interesting to see these analogies developed. Controllers and phase planes are equivalent in this sense, and if the equivalences were worked out, a better attitude control system might be discovered.

Finally, many of the control algorithms considered in this thesis could be improved upon. For example, additional terms might be added to the Galileo controller, or the MGN phase plane geometry could be modified in an attempt to improve performance during VOI. Also, the addition of an explicit deadzone to the variable pulse width schemes considered in Chapter 7 might solve the problem of high steady state fuel consumption that was observed with these systems.

This thesis is thus far from the last word on mass expulsion control of rigid spacecraft attitude. There is much work still to be done, and many challenging problems are still to be solved.

Appendix A

Details of the Propellant Slosh Model

A.1 Derivation of the Equations of Motion and Transfer Functions

A spacecraft with sloshing propellant can be modeled as shown in Figure 2-15. This model is holonomic and has three degrees of freedom: z, θ, and ϕ.

First, the kinetic coenergy, T^*, must be found. There are two bodies that can be associated with a kinetic coenergy -- the spacecraft itself and the fuel slosh pendulum. It can be shown that:

$$T_{s/c}^* = \frac{1}{2} M \dot{z}^2 + \frac{1}{2} I \dot{\phi}^2 \qquad (A.1)$$

$$T_{fuel}^* = \frac{1}{2} m v_{fuel}^2 + \frac{1}{2} I_f \dot{\theta}^2 \qquad (A.2)$$

(assuming motion about a principal axis). Figure A-1 shows the factors contributing to v_{fuel}. Using this figure, the equation:

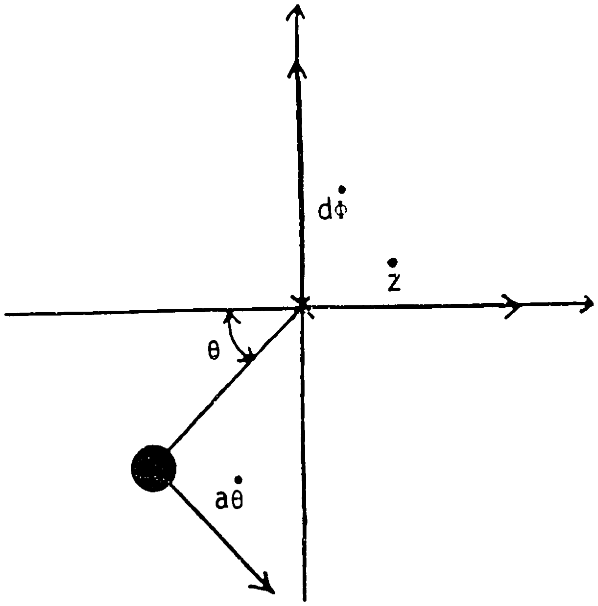


Figure A-1: Contributions to the Slosh Pendulum Velocity

$$(v_{fuel}^2) = (\dot{\phi} d - a\dot{\theta}\cos\theta)^2 + (\dot{z} + a\dot{\theta}\sin\theta)^2 \qquad (A.3)$$

can be found. Thus:

$$T^* = \frac{1}{2}M\dot{z}^2 + \frac{1}{2}I\dot{\phi}^2 + \frac{1}{2}m[(\dot{\phi}d - a\dot{\theta}\cos\theta)^2 + (\dot{z} + a\dot{\theta}\sin\theta)^2] + \frac{1}{2}I_f\dot{\theta}^2 \qquad (A.4)$$

The potential energy, V, of the system equals zero. Thus, the Lagrangian is given by the expression:

$$L = T^* - V = T^* \qquad (A.5)$$

Two generalized nonconservative forces act on the system. The work-displacement relations for these forces are:

$$\delta W_z = F\,\delta z \qquad (A.6)$$

$$\delta W_\phi = \tau\,\delta\phi \qquad (A.7)$$

So the generalized nonconservative forces, Ξ_i, are:

$$\Xi_z = F = (M + m)G \qquad (A.8)$$

$$\Xi_\phi = \tau \qquad (A.9)$$

Substituting all this into Lagrange's equation:

$$\frac{d}{dt}\left[\frac{\partial L}{\partial \xi}\right] - \left[\frac{\partial L}{\partial \xi}\right] = \Xi_\xi \qquad (A.10)$$

gives the equations of motion:

$$(1)\quad \xi_1 = z \qquad (A.11)$$

$$(M+m)\ddot{z} + (ma\sin\theta)\ddot{\theta} + (ma\cos\theta)\dot{\theta}^2 = (M+m)G$$

$$(2)\quad \xi_2 = \phi \qquad (A.12)$$

$$(I+md^2)\ddot{\phi} - (mad\cos\theta)\ddot{\theta} + (mad\sin\theta)\dot{\theta}^2 = \tau$$

$$(3)\quad \xi_3 = \theta \qquad (A.13)$$

$$\left[a + \frac{I_f}{ma}\right]\ddot{\theta} - \ddot{\phi}d\cos\theta + \ddot{z}\sin\theta = 0$$

These equations are very complicated. However, they can be linearized to:

$$\ddot{z} = G \qquad (A.14)$$

$$(I+md^2)\ddot{\phi} - (mad)\ddot{\theta} = \tau \qquad (A.15)$$

$$\left[\frac{I_f}{ma} + a\right]\ddot{\theta} - \ddot{\phi}d + G\theta = 0 \qquad (A.16)$$

If all initial conditions are assumed to be zero, the Laplace transforms of equations (A.15) and (A.16) can be taken:

$$s^2(I+md^2)\phi - s^2(mad)\theta = \tau(s) \qquad (A.17)$$

$$s^2\left[\frac{I_f}{ma}+a\right]\theta-s^2\phi d+G\theta=0 \qquad (A.18)$$

By combining these equations, one can derive the transfer functions:

$$\frac{\theta(s)}{\tau(s)}=\frac{d}{B(s^2+f_2^2)} \qquad (A.19)$$

$$\frac{\phi(s)}{\tau(s)}=\frac{A(s^2+f_1^2)}{Bs^2(s^2+f_2^2)} \qquad (A.20)$$

where

$$A=\frac{I_f+ma^2}{ma}$$

$$B=\frac{II_f+Ima^2+d^2mI_f}{ma}$$

$$f_1^2=\frac{G}{A}$$

$$f_2^2=\frac{G(I+md^2)}{B}$$

If the effective torque on the spacecraft, E, is defined as E = $I\ddot{\phi}$, so that E(s) = $s^2I\phi(s)$, a third transfer function is obtained:

$$\frac{E(s)}{\tau(s)}=\frac{AI(s^2+f_1^2)}{B(s^2+f_2^2)} \qquad (A.21)$$

Or, if damping is included, transfer function (A.21) can be rewritten as:

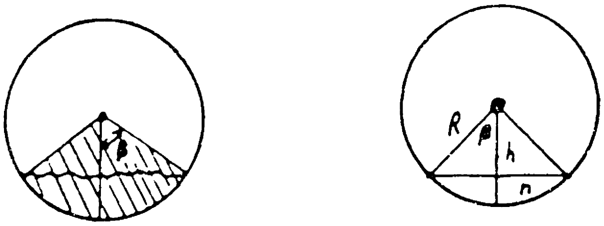
$$\frac{E(s)}{\tau(s)}=\frac{AI(s^2+2\zeta f_1 s+f_1^2)}{B(s^2+2\zeta f_2 s+f_2^2)} \qquad (A.22)$$

In Section 2.2.3, this equation is presented in state-space form. It is the equation used to model the effects of propellant slosh on the torques that affect the Magellan spacecraft.

A.2 The Length of the Fuel Slosh Pendulum

In Figure 2-15, a is shown to be the distance between the geometric center of the propellant tank and the center of mass of the propellant. In this section, this parameter will be calculated.

Figure A-2 shows a spherical tank with fluid in it. The volume of the fluid can be shown to be:



$h = R\cos\beta \qquad n = R\sin\beta$

Figure A-2: Spherical Tank Containing Fluid

$$fluid\ volume = \frac{\pi}{3}R^3[2 - 2\cos\beta - \sin^2\beta\cos\beta] \tag{A.23}$$

where β is as shown in the figure and R is the radius of the tank. If F, the volume fraction of the tank that is filled with fluid, is known, then the volume of the fluid is also given by the expression:

$$fluid\ volume = F\left[\frac{4}{3}\pi R^3\right] \tag{A.24}$$

Assuming uniform fluid density, the center of mass of the fluid occurs in the plane that cuts the fluid volume in half. Or, using equations (A.23) and (A.24), it occurs at the fluid height determined by the angle β, where:

$$\frac{\pi}{3}R^3[2 - 2\cos\beta - \sin^2\beta\cos\beta] = \frac{1}{2}F\left[\frac{4}{3}\pi R^3\right] \tag{A.25}$$

or, equivalently:

$$\cos^3\beta - 3\cos\beta + (2 - 2F) = 0 \tag{A.26}$$

Using the cubic equation, the solution to (A.26) can be shown to be:

$$\cos\beta = 2\cos\left[\frac{\alpha}{3} + 240^o\right] \tag{A.27}$$

where

$$\cos(\alpha) = -\sqrt{\frac{(2-2F)^2}{4}} \tag{A.28}$$

From Figure A-2, a = $R\cos(\beta)$. Thus:

$$a = 2R\cos\left[\frac{\alpha}{3} + 240^o\right] \tag{A.29}$$

The pendulum length, a, is therefore a function of R and F. As propellant is consumed, F decreases, increasing the length of the slosh pendulum.

Appendix B

Approximation of Optimal Switching Lines with the RLC

B.1 Derivation of the Minimum Time-Fuel Control Law

In this section, minimum time-fuel switching curves will be derived[9]. The system equations are assumed to be:

$$\dot{x}_1 = x_2 \qquad (\phi = x_1) \tag{B.1}$$

$$\dot{x}_2 = u \qquad (\dot{\phi} = x_2)$$

where u is restricted to the values -1, 0, and +1. Note that if units are carefully chosen (*ie.* 1 angle unit = (thruster torque capacity/I) x sec^2), then these are the same equations considered elsewhere in this thesis. The performance index to be minimized is:

$$J = \int_0^{t_{final}} (\lambda + |u|)\, dt \tag{B.2}$$

with the terminal constraint:

$$\phi(t_{final}) = 0 \tag{B.3}$$

The Hamiltonian for this problem is:

$$H = \lambda + |u| + p_1\dot{\phi} + p_2 u \tag{B.4}$$

where p_1 and p_2 are the costate variables. The optimal trajectories can be shown to be:

$$\phi = -\frac{1}{2}\dot{\phi}^2 + c_1 \qquad u = -1 \tag{B.5}$$

$$\phi = \frac{1}{2}\dot{\phi}^2 + c_2 \qquad u = +1$$

$$\phi = c_3 t + c_4 \qquad u = 0$$

The optimal switching curves are:

---

[9] The material in this appendix is presented only in broad outline. For details, see references [26, 25].

$$\phi=-\frac{\lambda+4}{2\lambda}\dot{\phi}^2 \quad (\textit{from } u=+1 \textit{ to } u=0) \tag{B.6}$$

$$\phi=\frac{\lambda+4}{2\lambda}\dot{\phi}^2 \quad (\textit{from } u=-1 \textit{ to } u=0)$$

These are the curves shown in Figure 3-6.

However, the constraint that $\phi(t_f) = 0$ is unrealistic. Usually, thrusters are only used to get the state to within a certain deadzone near the origin. Thus, the switching curves can be shifted from the origin to $\phi = \pm\phi_{db}$, where ϕ_{db} is half the allowable deadzone width. This is shown in Figure 3-9.

B.2 Derivation of an RLC that Approximates a Minimum Time-Fuel Control Law [26, 25]

Parameters are defined as shown in Figure B-1. Curves C and H are defined by the equations:

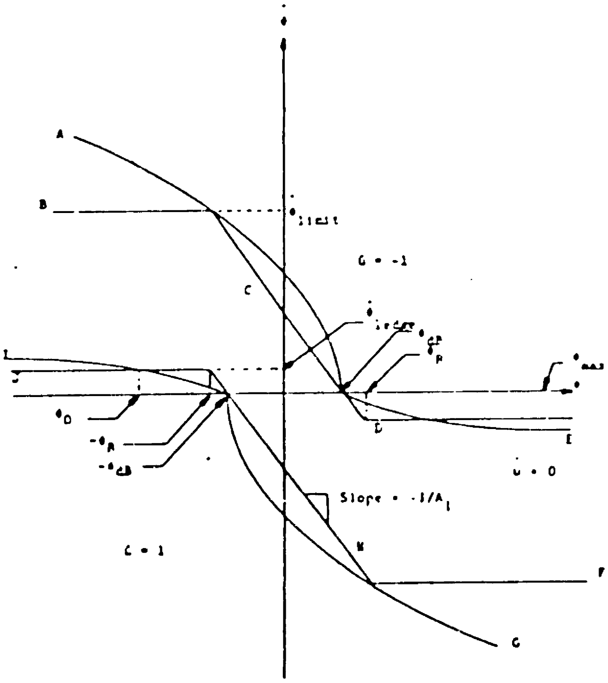


Figure B-1: Analogy Between RLC and Optimal Control Law

C: $\phi + A_1\dot{\phi} = \phi_{db}$ (B.7)

H: $\phi + A_1\dot{\phi} = -\phi_{db}$ (B.8)

The optimal switching curves are:

$$A \cdot \ \phi = -\frac{1}{2}\dot{\phi}^2 + \phi_{db} \qquad (B.9)$$

$$E \cdot \ \phi = \frac{\lambda+4}{2\lambda}\dot{\phi}^2 + \phi_{db} \qquad (B.10)$$

$$G \cdot \ \phi = \frac{1}{2}\dot{\phi}^2 - \phi_{db} \qquad (B.11)$$

$$I \cdot \ \phi = -\frac{\lambda+4}{2\lambda}\dot{\phi}^2 - \phi_{db} \qquad (B.12)$$

From these equations and Figure B-1, it can be shown:

$$A_1 = \sqrt{\frac{\phi_r + \phi_{db}}{2}} \qquad (B.13)$$

$$\dot{\phi}_{limit} = \sqrt{2(\phi_r + \phi_{db})} \qquad (B.14)$$

$$\dot{\phi}_{ledge} = (\phi_r - \phi_{db})\sqrt{\frac{2}{\phi_r + \phi_{db}}} \qquad (B.15)$$

The only parameter needed to complete the definition of the RLC is the angle ϕ_r. The approach taken to get this value is to choose the value of ϕ_r that minimizes the mean-square error between the RLC and the optimal switching curves. If f is defined as:

$$f = (\dot{\phi}_{optimal} - \dot{\phi}_{RLC})^2 \qquad (B.16)$$

ϕ_r is sought such that:

$$\frac{\partial}{\partial \phi_r}\left[\frac{1}{\phi_{max} - \phi_{db}}\int_{-\phi_{db}}^{-\phi_{max}} f \, d\phi\right] = 0 \qquad (B.17)$$

Given values for ϕ_{db}, ϕ_{max}, and λ, the following recursion can be used to estimate this ϕ_r:

Initialization:

$$\phi_0=\frac{-\phi_{db}-\phi_{max}}{2}$$

$$\phi_r=\frac{-a_1+\sqrt{a_1{}^2-4a_0}}{2}$$

$$\text{where } a_1=-2\phi_{db}+\frac{\lambda}{\lambda+4}(\phi_0+\phi_{db})$$

$$a_0=\phi_{db}{}^2+\frac{\lambda}{\lambda+4}(\phi_0+\phi_{db})\,\phi_{db}$$

$$K_2=4\,(\phi_{max}-\phi_{db})^2$$

$$K_{1d}=k_2\sqrt{\frac{\lambda}{\lambda+4}}$$

Recursion:

$$K_1(k+1)=\frac{(16/3)\,(\phi_r(k)-\phi_{db})\,(\phi_{max}-\phi_{db})^{3/2}-(32/15)\,(\phi_r(k)-\phi_{db})^{5/2}}{\sqrt{\phi_r(k)+\phi_{db}}}$$

$$B(k)=\frac{(\phi_r(k)+\phi_{db})^{1/2}-0.5\,(\phi_r(k)+\phi_{db})^{-1/2}(\phi_r(k)-\phi_{db})}{\phi_r(k)+\phi_{db}}$$

$$C(k)=\frac{2.5\,(\phi_r(k)-\phi_{db})^{3/2}(\phi_r(k)+\phi_{db})^{1/2}}{\phi_r(k)+\phi_{db}}$$

$$D(k)=\frac{0.5\,(\phi_r(k)+\phi_{db})^{-1/2}(\phi_r(k)-\phi_{db})^{5/2}}{\phi_r(k)+\phi_{db}}$$

$$K_1'(k)=\frac{16}{3}(\phi_{max}-\phi_{db})^{3/2}B(k)-\frac{32}{15}[C(k)-D(k)]$$

$$\Delta\phi_r(k)=\frac{K_{1d}-K_1(k)}{K_1'(k)}$$

$$\phi_r(k+1)=\phi_r(k)+\Delta\phi_r(k)$$

The ϕ_r value that this iteration converges to can be substituted into equations (B.13), (B.14), and (B.15) to obtain the remaining RLC parameter values.

Appendix C

Derivation of the Pseudo-Open Loop Control Algorithm

In this appendix, the pseudo-open loop control algorithm will be derived. First, the optimal control problem:

$$\text{Given}\cdot\ \ddot{\phi}=\frac{\tau}{I} \tag{C.1}$$

$$\text{with terminal constraints}\cdot\ \psi_1(t_f)=\dot{\phi}(t_f)=0$$

$$\psi_2(t_f)=\phi(t_f)=0$$

$$\text{minimize the index}\cdot\ J=\int_0^{t_f}(k_w^{\ 2}+\tau^2)\,dt$$

must be formulated. For simplicity, τ will be temporarily assumed to be continuous. Later, when the optimal control history has been found, the bang-bang nature of thruster control will be taken into account.

The first-order necessary conditions that any optimal control history must satisfy are:

$\dot{x}=f$ where $x(0)=x_0$

$\dot{p}=-F^Tp-\left[\frac{\partial L}{\partial x}\right]^T$ where $F=\frac{\partial f}{\partial x}$

$J=\int_0^{t_f}L\,dt$

$p(t_f)=\frac{\partial}{\partial x}(\nu^T\psi)$

and $\nu=(\nu_1\ \ \nu_2)^T$ is constant

$\frac{\partial H}{\partial \tau}(t)=0$ where $H=L+p^Tf$

$\Omega(t_f)=0$ where $\Omega=H+\frac{\partial}{\partial t_f}(\nu^T\psi)$

$\psi_i(t_f)=0$

Applying these conditions to the problem formulated above, one obtains the conditions:

$$\ddot{\phi}=\frac{\tau}{I} \quad \phi(0)=\phi_0 \quad \dot{\phi}(0)=\dot{\phi}_0 \tag{C.2}$$

$$\dot{p}_1=-p_2 \quad p_1(t_f)=\nu_1 \tag{C.3}$$

$$\dot{p}_2=0 \quad p_2(t_f)=\nu_2 \tag{C.4}$$

$$2\tau+\frac{p_1}{I}=0 \tag{C.5}$$

$$k_w{}^2+\tau^2(t_f)+p_1(t_f)\frac{\tau(t_f)}{I}=0 \tag{C.6}$$

$$\phi(t_f)=\dot{\phi}(t_f)=0 \tag{C.7}$$

A τ time history satisfying these conditions will minimize the index J.

First, solving equations (C.3) and (C.4) for p_1 and p_2, one finds:

$$p_2=\nu_2 \tag{C.8}$$

$$p_1=\nu_1-\nu_2(t-t_f) \tag{C.9}$$

Rearranging equation (C.5):

$$\tau=-\frac{p_1}{2I} \tag{C.10}$$

Substituting (C.9) into (C.10), the optimal control history is found to be:

$$\tau=-\frac{1}{2I}(\nu_1-\nu_2(t-t_f)) \tag{C.11}$$

Thus, the optimal control history has been determined to within the constants ν_1, ν_2, and t_f. To complete the solution of the problem, these constants must be found.

ν_1 is easy to find. Equation (C.11) gives that, at $t = t_f$:

$$\tau(t_f)=-\frac{\nu_1}{2I} \tag{C.12}$$

Substituting this into (C.6), one obtains:

$$0=k_w{}^2-\frac{\nu_1{}^2}{4I^2} \tag{C.13}$$

or

$$\nu_1=2Ik_w \tag{C.14}$$

To find ν_2 and t_f, the terminal constraints must be used. The optimal time histories of ϕ and $\dot{\phi}$ will be found, and their final values will be set equal to the terminal conditions. This will enable a determination of the desired variables.

First, combining equations (C.2) and (C.11), one obtains:

$$\ddot{\phi}=-\frac{1}{2I^2}(2Ik_w-\nu_2(t-t_f)) \qquad (C.15)$$

which can be integrated, yielding:

$$\dot{\phi}=\dot{\phi}_0-\frac{k_w}{I}t+\frac{\nu_2}{2I^2}(\frac{1}{2}t^2-t_ft) \qquad (C.16)$$

At $t = t_f$, this equation becomes:

$$0=\dot{\phi}_0-\frac{k_w}{I}t+\frac{\nu_2}{2I^2}(-\frac{1}{2}t_f^2) \qquad (C.17)$$

which can be solved for ν_2:

$$\nu_2=\frac{4I^2\dot{\phi}_0-4Ik_wt_f}{t_f^2} \qquad (C.18)$$

Next, (C.16) can be integrated once more, resulting in the equation:

$$\phi=\phi+\dot{\phi}_0t-\frac{k_w}{2I}t^2+\frac{\nu_2}{2I^2}\left[\frac{1}{6}t^3-\frac{t_f}{2}t^2\right] \qquad (C.19)$$

Substituting equation (C.18) for ν_2 and setting $t = t_f$, one finds:

$$0 = k_wt_f^2+ 2\dot{\phi}_0It_f+ 6I\phi_0 \qquad (C.20)$$

or

$$t_f=\frac{2I}{\nu_1}(-\dot{\phi}_0I-\sqrt{\dot{\phi}_0^2I+6k_wI\phi_0}) \qquad (C.21)$$

(the negative root has no physical significance). In order to ensure that t_f is real under arbitrary initial conditions, it must be assured that:

$$k_w\leq\frac{\dot{\phi}_0^2I}{6\phi_0}$$

Since ϕ_0 may be less than 0, k_w must be negative. If, however, the convention that k_w is positive is adopted, equations (C.14), (C.18), and (C.21) must be modified to:

$$\nu_1=-2Ik_w$$

$$\nu_2=\frac{4I^2\dot{\phi}_0-2\nu_1t_f}{t_f^2}$$

$$t_f=\frac{2I}{\nu_1}(-\dot{\phi}_0I-\sqrt{\dot{\phi}_0^2I+6k_wI\phi_0})$$

Equation (C.11) gives the optimal control history under the assumption of a continuous control torque. However, the thrusters can only deliver torques of $-\Gamma$, 0, or $+\Gamma$. Thus, thrusters can not be used to realize (C.11) directly. However, another approach is possible. Using the initial conditions at the beginning of a sampling interval, equation (C.11) can be integrated to calculate the optimal $\int \tau\, dt$ over the interval. This integral can then be set equal to $\int_0^{t_{pulse}} \Gamma\, dt$, so the thrusters, if fired for a time t_{pulse}, will realize the same $\int \tau\, dt$ as the optimal control law.

Integrating (C.11):

$$\int_0^{t_{sample}} \tau\, dt = -\frac{1}{2J}(\nu_1 t_{sample} - \nu_2(\frac{1}{2}t_{sample}{}^2 - t_f t_{sample})) = \Gamma t_{pulse} \qquad (C.22)$$

or

$$t_{pulse} = -\frac{1}{2J\Gamma}(\nu_1 t_{sample} - \nu_2(\frac{1}{2}t_{sample}{}^2 - t_{sample} t_f)) \qquad (C.23)$$

The sign of t_{pulse} indicates the sign of the torque that the thrusters should deliver.

Notice that starting with an open loop optimal control law, a closed loop system has been obtained by updating initial conditions every sampling interval. That is why this scheme has been named pseudo-open loop control.

Appendix D
Simulation Code

The Magellan computer simulation code is far too lengthy to include here in its entirety. However, much of the simulator deals with fault protection, configuration control, and other subjects not directly related to this thesis. Only those portions of the simulator that are relevant are included in this appendix.

The main program is written in ACSL (Advanced Continuous Simulations Language). It implements the rigid-body equations of motion, propellant slosh, disturbance models, and the rate-integrating gyro model, and acts as executive over the various subroutines.

Subroutines are written in FORTRAN. They include attitude control logic, thruster models, models of the MGN thruster attitude control system, the RLC (with and without short pulse regions), the Galileo controller, and the pseudo-open loop control scheme.

```
PROGRAM: ACSL SIM FOR ATTITUDE CONTROL, ARTICULATION CONTROL,           *"
"*         RAM SAFING, AND FAULT PROTECTION ANALYSIS.                   *"
"*         PROGRAM ACSS SIMULATION CONTAINS THE FOLLOWING:              *"
"*                UPDATE: 26 SEP 86 - SBJ                               *"
"*=====================================================================*"
"*       A. RIGID BODY EQUATIONS OF MOTION                              *"
"*       B. X,Y AND Z REA ATTITUDE CONTROL WITH PHASE PLANES            *"
"*       C. FLIGHT ARTICULATION CONTROL ALGORITHM                       *"
"*       D. SSU & SADM MODELS (NOT COUPLED TO EQNS OF MOTION)           *"
"*       E. M-K-S UNITS ARE USED THROUGHOUT                             *"
"*       F. SPERRY REACTION WHEEL MODEL                                 *"
"*       G. POLYNOMIAL APPROXIMATION OF REFERENCE QUATERNIONS           *"
"*       H. SECOND ORDER CLOSED LOOP RW CONTROL                         *"
"*       I. FAULT PROTECTION & CONFIGURATION CONTROL ALGORITHMS         *"
"*       J. ARU & REA MODELS WITH TOTAL SYSTEM LAG                      *"
"*       K. RAM SAFING ALGORITHMS                                       *"
"***********************************************************************"


" "
">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<"
">>         INITIAL SECTION - UNITS ARE M-K-S AND RADIANS             <<"
">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<"
" "
 INITIAL
       IF(ICPAS.NE.1.) GO TO ICP111
       CALL HWINIT(ARUAPW,ARUBPW,GYROON,MIPC,...
                   ARUAEL,ARUBEL,ARUART,ARUBRT,HWSTAT,...
                   ILV1,ILV2,TD1POW,TD2POW,ED1POW,ED2POW)
       CALL CONFIG(6,FLTOCC,CHUSID,NUMCHU,PRGYCH,POWREQ,...
                   MDSCNT,RQRJCT,ARUAPW,ARUBPW,HWSTAT,OFFPWF,...
                   OBGYCP,SWGYRO,COLDST,...
```

```
                   CONREQ,APUDES,ILVCTL,ARUART,ARUBRT,...
                   ARUAEL,APUBEL,GYFOON,MIPC,ILV1,ILV2,ES1,FS2,...
                   MIDSEL,CHSLPT,LRSWEN,GYWUTM,SSUAUS,SSUBUS,...
                   CHREAD,POWIND)

      CALL TPROFILE (T1(1),T2(1),T3(1),T4(1),PDELAY,12,LPUL(1,1),TAVG1A)
      CALL TPROFILE (T1(2),T2(2),T3(2),T4(2),PDELAY,3,LPUL(1,2),TAVG2B)
      CALL TPROFILE (T1(3),T2(3),T3(3),T4(3),PDELAY,3,LPUL(1,2),TAVG3B)
      CALL TPROFILE (T1(2),T2(2),T3(2),T4(2),PDELAY,3,LPUL(1,3),TAVG2C)
      CALL TPROFILE (T1(3),T2(3),T3(3),T4(3),PDELAY,3,LPUL(1,3),TAVG3C)
      CALL TPROFILE (T1(1),T2(1),T3(1),T4(1),PDELAY,12,LPULMM,TAVGMM)
      ICPAS=0.0
       EPSIL=DANT/TOL
       IF (IT(1,1).GT.0.0) GO TO II5
       VARI=.TRUE.
       IT(1,1)=IXXT(MASS)
       IT(2,2)=IYYT(MASS)
       IT(3,3)=IZZT(MASS)
   II5..CONTINUE
       CALL M3NGOD(IT,ITM)
       CALL MATIN3(ITM,ITMI)
       CDELM=COS(DEGTOR*DFLM)
       SDELM=SIN(DEGTOR*DELM)
       TDELM=TAN(DEGTOR*DFLM)
       LDELM=-ZFACE*TDELM                $"SRM OFFSET DUE TO DELM"
       SLSQ=LCGM*LCGM+LAAM*LAAM+LAEM*LAEM $"KSS SRM MISALIGNMENTS"
       RTHR(1)=0.
       RTHR(2)=SQRT(SLSQ+LDFLM*LDELM)
       RTHR(3)=-ZFACE
       RSF(2)=RTHR(2)
       RSF(3)=-(ZCM(MASS)+ZSIDE)
       DO II10 I=1,3
       DAT(I)=THDTB(I)*DEGTOR
   II10..CONTINUE
       SDAT=SIN(DAT(1))
       CDAT=COS(DAT(1))
       CALL SDOT(DEGTOR,ANG,ANGP)
       CALL T313(ANGP,TTB)
       DO II20 J=1,24
       DO II30 I=1,3
       LT(I,J)=LTO(I,J)
   II30..CONTINUE
   II20..CONTINUE
       ALPHA=ALPHA*DEGTOR
       ALDSFR=ALDSFR*DEGTOR
       MASSO=MASS
       PROPMO=PROPM
       TIME = T + TSTART
"======================================================================"
"===      INITIALIZE TIMELINE FLAGS                                 ==="
"======================================================================"
" "

      TMLIN(1)=1272.0
      TMLIN(2)=1622.0
      TMLIN(3)=4146.0
      TMLIN(4)=4496.0
      TMLIN(5)=4896.0
      TMLIN(6)=5246.0
      TMLIN(7)=9536.0
      TMLIN(8)=9888.0
```

```
      TMLIN(9)=0942.0
      SMAP = TMLIN(8)
      EMAP = TMLIN(9)
" "
"===================================================================="
"==      INITIALIZE MODE ACCORDING TO TSTART WRT NOMINAL TIMELINE   =="
"===================================================================="
" "
      IF (.NOT.QDISET) CALL MAPMOD(TSTART,TPER,TS0,PER,0.,TMLIN,...
         SWMODE,DESMAN,MTPTR,RET,CAN,MNVI)
" "
"--------------------------------------------------------------------"
"---      INITIALIZE EARTH/SUN POINTING REFERENCE QUATERNION       ---"
"--------------------------------------------------------------------"
" "
      CALL PLYREF(MPDAY,TE0,PSFE,ASE,8,QDIE)
      CALL VEQL(QDIE,QDIEP,4)
" "
"--------------------------------------------------------------------"
"---      INITIALIZE VENUS MAPPING REFERENCE QUATERNION            ---"
"--------------------------------------------------------------------"
" "
      IF ((TSTART.GT.TMLIN(9)).AND.(TSTART.LT.TMLIN(8))) TMAP=TV0
      IF (TSTART.LE.TMLIN(9)) TMAP=TSTART+(PER-TMLIN(8))
      IF((TSTART.GE.TMLIN(8)).AND.(TSTART.LE.PER))TMAP=TSTART-TMLIN(8)
      CALL PLYREF(TMAP,TV0,PSFV,ASV,8,QDIV)
      CALL VEQL(QDIV,QDIVP,4)
" "
"--------------------------------------------------------------------"
"---      GENERATE ORBIT/TRAJECTORY POSITION OF SPACECRAFT         ---"
"--------------------------------------------------------------------"
" "
      CALL TRJGEN(TSTART,TS0,ECS,AS,GKS,1.,RS,UVS)
      CALL SCOPOS(UVS,RS,TRNSD,NADIR,ALT,SCTA)
      SCTAM=SCTA
" "
"--------------------------------------------------------------------"
"---      INITIALIZE MAPPING RETRACE PARAMETERS                    ---"
"--------------------------------------------------------------------"
" "
      IF (.NOT.RET) GO TO I100
      RTIM=(TMLIN(9)+(PER-TMLIN(8)))*(1.0-(TSTART-TMLIN(9))/TAUM(8))
      IF (RTIM.LT.0.0) RTIM=0.0
      CALL PLYREF(RTIM,TV0,PSFV,ASV,8,QDIR)
      CALL VEQL(QDIR,QDIRP,4)
 I100..CONTINUE
" "
"--------------------------------------------------------------------"
"---      ASSIGN INERTIAL TO DESIRED QUATERNION FOR INITIAL MODE   ---"
"--------------------------------------------------------------------"
" "
     IF(SWMODE.EQ.5 .AND. RET) CALL VEQL(QDIR,QDI,4)
     IF(SWMODE.EQ.5 .AND. .NOT.RET) CALL VEQL(QDIV,QDI,4)
     IF(SWMODE.EQ.1 .AND. CAN.EQ.4) CALL VEQL(QDIE,QDI,4)
" "
"--------------------------------------------------------------------"
"---       INITIALIZE BODY AND TRUTH QUATERNIONS                   ---"
"--------------------------------------------------------------------"
" "
      IF (QRISET) GO TO I120
```

```
      CALL VEQL(QDI,QBI,4)
I120..CONTINUE
      IF (QBBSET) GO TO I130
      CALL VEQL(QBI,QBBI,4)
I130..CONTINUE
      CALL EQTC(DTI,QDI)
      CALL EQTC(BBTI,QBBI)
      CALL EQTC(BTI,QBI)
" "
"-----------------------------------------------------------------"
"---         INITIALIZE REACTION WHEEL SPEED                   ---"
"-----------------------------------------------------------------"
" "
      DO I200 I=1,3
      WSPD(I)=HWIC(I)/RWI
  I200..CONTINUE
" "
"-----------------------------------------------------------------"
"--- INITIALIZE FLIGHT PARAMETERS DEPENDENT UPON MASS PROPERTIES ---"
"-----------------------------------------------------------------"
" "
      DO I250 I=1,3
        SYSRAT(I)=ITPAR(I)/RWI
        ARWL(I)=(WSPMAX*(GR2/GR1))/SYSRAT(I)
      I250..CONTINUE
" "
"================================================================="
"===        INITIALIZE MAXIMUM THRUST MAGNITUDE                ==="
"================================================================="
" "
       IF (PRES.EQ.0.0) PRES=TPRES(FLOAD-PPOPM)
" "
" "
       DO I300 I=1,24
         J=1
         IF (I.GT.12) J=2
         IF (I.GT.16) J=3
         FM=(YI(J)+SL(J)*PRFS+QQ(J)*PRES*PRES)*LBFTON $"LPF TO NEWTONS"
         FT(1,I)=FM*FTUV(1,I)
         FT(2,I)=FM*FTUV(2,I)
         FT(3,I)=FM*FTUV(3,I)
   I300..CONTINUE
" "
 ICP111..CONTINUE
 END$"END OF INITIAL SECTION"
" "
">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<"
">>>        DYNAMIC AND DERIVATIVE SECTIONS                    <<<"
">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<"
" "
 DYNAMIC
 DERIVATIVE
       TIME=T+TSTART
       TPDT=TIME+DIGTIM
       TDAY=MPDAY+(TIME+EARDIG/2.)/CTS
"*****************************************************************"
"***       ANALOG MODELS TO BE INTEGRATED OVER TIME            ***"
"*****************************************************************"
" "
"ASASASASASASASASASASASASASASASASASASASASASASASASASASASASASASASAS"
```

```
"AS       ANALOG SAMPLE LOGIC                                         AS"
"ASASASASASASASASASASASASASASASASASASASASASASASASASASASASASASASASASASAS"
 PROCEDURAL
" "
      ANALOG=.FALSE.
      IF (T.LT.THOLD) GO TO AS100
      ANALOG=.TRUE.
      THOLD=T+DANT
 AS100..CONTINUE
" "
 ENDS" OF ANALOG SAMPLE PROCEDURAL"
 PROCEDURAL
      IF (.NOT.ANALOG) GO TO AS200
"RWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRW"
"RW       REACTION WHEEL AND TACHOMETER MODEL                         RW"
"RWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRWRW"
" "
      CALL RWMOD(DANT,VTCF,OCMVLT,WSPD,TACWS,TWL,HW,WPWR,WFMW,...
           MIPC,WSC,WBIA,WDD,WCDF)
" "
      TQ1=TWL(1)
      TQ2=TWL(2)
      TQ3=TWL(3)
" "
      TW1=CMPXPL(PT,QT,TQ1,IT1,IT2)
      TW2=CMPXPL(PT,QT,TQ2,IT3,IT4)
      TW3=CMPXPL(PT,QT,TQ3,IT5,IT6)
" "
      TWL(1)=TW1
      TWL(2)=TW2
      TWL(3)=TW3
" "
"END OF REACTION WHEEL MODEL PROCEDURAL "
" "
      AS200..CONTINUE
 ENDS"OF ANALOG SAMPLE BLOCK PROCEDURAL"
 PROCEDURAL
"EMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEM"
"E EQUATIONS OF MOTION (EOM) MODEL: EQUATIONS FOR MOMENTS AND FORCE   M"
"E SEE THOMPSON PAGE 234 EQUATION 7.11-13 ,7.11-14 FOR REFERENCE      M"
"E TOTAL APPLIED TORQUE ABOUT THE CENTER OF MASS                      M"
"EMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEMEM"
" "
"---ROTATION----------------------------------------------------------"
" "
       CALL M3V(ITM,W,IW)
       CALL VCRS(W,IW,WXIW)
       CALL VCRS(W,HW,WXHW)
       DO EM30 I=1,3
       IDWD(I)=TDIST(I)+TCB(I)-WXIW(I)-(TWL(I)+WXHW(I))-JDMOM(I)
   EM30..CONTINUE
" "
"---VEHICLE ACCELERATION----------------------------------------------"
" "
       RM=1./MASS
       CP1=COS(PHI(1))
       CP2=COS(PHI(2))
       CALL VADD(FDIST,FCB,FEXT,3)
       RODD(1)=RM*(FEXT(1)*CP2+FEXT(3)*PHI(2))
       RODD(2)=RM*(FEXT(2)*CP1-FEXT(3)*PHI(1))
```

```
      RODD(3)=RM*(FEXT(3)*CP1*CP2-FEXT(1)*PHI(2)+FEXT(2)*PHI(1))
" "
"---FUEL SLOSH MODEL----------------------------------------------------- "
" "
      ACCEL=RODD(3)
      IF (.NOT.SLAG.OR.ACCEL.LE.0.0) GO TO SL10
" "
"   CALCULATE SLOSH MOMENT ARM, FRACT, AND FUEL INERTIA.  FUEL INERTIA"
"      WAS CALCULATED BY A CURVE FIT."
" "
      SLARM=ZPROP-ZFACE-RZIN
      FRACT=PROPM/TMASS
      IFUEL=3.5*(1.0-4.0*(FRACT-0.5)**2)-2.45*SIN(PI*FRACT)
" "
"   CALCULATE PENDULUM LENGTH"
" "
      GUMBY=2.0-(2.0*FRACT)
      POKEY=ACOS(-SQRT(GUMBY**2/4.0))
      RUFUS=2.0*COS((POKEY/3.0)+4.1887902)
      PENLEN=RUFUS*RTANK
" "
"   CALCULATE NATURAL FREQUENCIES, SLOSH ACCELERATION, AND DUMMY"
"      TORQUE VARIABLE ACCELERATION"
" "
      ABRA=(IFUEL+(PROPM*PENLEN*PENLEN))/(PROPM*PENLEN)
      FREQ1=SQRT(ACCEL/ABRA)
      DO SL05 I=1,2
         CDABRA(I)=(ITM(I,I)*IFUEL+ITM(I,I)*PROPM*PENLEN*PENLEN+...
                    SLARM*SLARM*PROPM*IFUEL)/(PROPM*PENLEN)
         FREQ2(I)=SQRT(ACCEL*(ITM(I,I)+PROPM*SLARM*SLARM)/CDABRA(I))
         SLAC(I)=-((FREQ2(I)**2)*SLANG(I))-2.0*SLZETA*FREQ2(I)*...
                 SLRATE(I)+(SLARM*IDWD(I)/CDABRA(I))
         DUMDD(I)=-((FREQ2(I)**2)*DUM(I))-2.0*SLZETA*FREQ2(I)*...
                  DUMD(I)+IDWD(I)
 SL05..CONTINUE
" "
"   INTEGRATE ACCELERATIONS TO GET CURRENT STATE"
" "
     SLRATE=INTVC(SLAC,SLRIC)
     CALL XFERB(SLRD=SLRATE,2)
     SLANG=INTVC(SLRD,SLANIC)
     DUMD=INTVC(DUMDD,DUMDIC)
     CALL XFERB(DUMDO=DUMD,2)
     DUM=INTVC(DUMDO,DUMIC)
" "
"   TORQUE WITH SLOSH IS A LINEAR COMBINATION OF IDWD & DUMMY VARIABLES"
" "
     DO SL07 I=1,2
        TWSLO(I)=(ABRA*ITM(I,I)/CDABRA(I))*((FREQ1**2-FREQ2(I)**2)*...
                 DUM(I)+2.0*SLZETA*(FREQ1-FREQ2(I))*DUMD(I)+IDWD(I))
 SL07..CONTINUE
     TWSLO(3)=IDWD(3)
     GO TO SL20
 SL10..CONTINUE
     CALL VEQL(IDWD,TWSLO,3)
 SL20..CONTINUE
" "
      CALL M3V(ITMI,TWSLO,WD)
"---    VEHICLE ROTATIONAL RATE                                    ---"
      W=INTVC(WD,WIC)
```

```
"---       VEHICLE ROTATIONAL DISPLACEMENT                               ---"
       CALL XFERB(WO=W,3)
       PHI=INTVC(WO,PHIIC)
" "
"IRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIR"
"IR        IRU GYRO MODEL                                                IR"
"IRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIRIR"
" "
      W1=W(1)
      W2=W(2)
      W3=W(3)
      WP1=CMPXPL(P1,Q1,W1,IC1,IC2)
      WP2=CMPXPL(P2,Q2,W2,IC3,IC4)
      WP3=CMPXPL(P3,Q3,W3,IC5,IC6)
" "
      WPP(1)=WP1
      WPP(2)=WP2
      WPP(3)=WP3
" "
"--       GYRO BIAS TERMS                                                 --"
" "
      DO IR10 I=1,3
        WRG(I)=RGB(I)+AISF(I)*WPP(I)
  IR10..CONTINUE
" "
      PHIP1=INTVC(WRG,RGIC)
" "
 ENDS"OF EQUATIONS OF MOTION PROCEDURAL "
"PTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPT"
"PT        POINTING ERROR COMPUTATIONS                                     PT"
"PTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPTPT"
" "
 PROCEDURAL
      IF(.NOT.ANALOG) GO TO PT100
      CALL IRUPRS(PHI,PHIPP,IPHI)
      CALL ATTDET(QBBI,IPHI,BBTI)
      CALL QDBM(QDI,QBBI,QER)
      CALL VUNT(QER,4)
      SGNQE4=SIGN(2.,QER(4))
      PTGER(1)=SGNQE4*QER(1)
      PTGER(2)=SGNQE4*QER(2)
      PTGER(3)=SGNQE4*QER(3)
      PTGERM=PTGER(1)**2+PTGER(2)**2+PTGER(3)**2
      PTGERM=SQRT(PTGERM)
 PT100..CONTINUE
" "
 ENDS" OF POINTING ERROR EVALUATION PROCEDURAL"
 ENDS" OF DERIVATIVE SECTION"
      TERMT((T .GE. TSTOP))
">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<"
">>>       DISCRETE BLOCKS                                                <<<"
">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<"
"DRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDR"
"DR        DESIRED REFERENCE GENERATION                                    DR"
"DRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDRDR"
"----------------------------------------------------------------------------"
"---       GENERATE NEW EARTH/SUN POINTING REFERENCE                      ---"
"----------------------------------------------------------------------------"
 DISCRETE EARREF
 INTERVAL EARDIG=500.
```

```
 PROCEDURAL
" "
      CALL PLYREF(TDAY,TEQ,PSFE,ASE,8,QDIE)
      CALL QDBM(QDIE,QDIEP,QRER)
      WESR(1)=2.*SIGN(1.,QRER(4))*QRER(1)/EARDIG
      WESR(2)=2.*SIGN(1.,QRER(4))*QRER(2)/EARDIG
      WESR(3)=2.*SIGN(1.,QRER(4))*QRER(3)/EARDIG
      CALL VEQL(QDIE,QDIEP,4)
" "
 ENDS
 ENDS" OF EARTH POINTING PROCEDURAL"
" "
 DISCRETE THIRTY $"30 HZ INTERUPT"
 INTERVAL F30W=0.033333 $"(30HZ)"
 PROCEDURAL
"----------------------------------------------------------------------"
"---      INCREMENT RTI COUNTER ON FIRST 30 HZ CYCLE                ---"
"----------------------------------------------------------------------"
" "
      F30 = .NOT.F30
      F75 = .FALSE.
      I30 = I30+1
      I30 = MOD(I30,4)
      IF (I30.EQ.0) F75=.TRUE.
      IF (.NOT.F30) GO TO TH10
        RTI = RTI + 1
        IF (RTI .GE. 10) RTI = 0
        IF (RTI .EQ. 0)  MINFRM=MINFRM+1
        IF (MINFRM .LT. 91) GO TO TH10
          MINFRM=0
          MAJFRM=MAJFRM+1
TH10..CONTINUE
" "
"GAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGA"
"GA      GYRO SAMPLE AND ATTITUDE UPDATE                             GA"
"GAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGAGA"
"----------------------------------------------------------------------"
"---      GYRO SAMPLE                                               ---"
"----------------------------------------------------------------------"
" "
      DO GA10 I=1,3
        PHIP(I)=PHIP1(I)
  GA10..CONTINUE
      DO GA20 I=1,3
        PHIBD(I)=QNTZR(DSTEP,PHIP(I))
  GA20..CONTINUE
" "
"----------------------------------------------------------------------"
"---      ATTITUDE PROPAGATION                                      ---"
"----------------------------------------------------------------------"
" "
      CALL IRUPRS(PHIBD,PHIPDP,INCPHI)
" "
"     CHECK FOR GYRO FAILURE MODES IF ENABLED"
" "
      IF (.NOT.ARUFMW) GO TO GA25
      CALL ADCTRL (INCPHI,CHSLPT,ARLAPT,ARUBRT,GFMW,MIPC,GFBIT,...
       GSBV,ARUAPW,ARUBPW,MTDSEL,MDSCNT,PRGYCH,CHUSID,NUMCHU,FPDS,...
       FLTAB,SWMODE,COLDST,FTAB,DETECT,BODYRT,...
       NETCNT)
```

```
" "
GA25..CALL ATTDET(QBI,INCPHI,BTI)
" "
"----------------------------------------------------------------------"
"---         ATTITUDE CONTROL                                       ---"
"----------------------------------------------------------------------"
" "
"----------------------------------------------------------------------"
"---         ATTITUDE AND RATE CONTROL ERROR GENERATION             ---"
"----------------------------------------------------------------------"
" "
      IF (SWMODE.EQ.6) GO TO AC10 $"RAMSAFE; DONT GENERATE CNTRL ERRORS"
" "
      CALL ATCER(QDI,QBI,PLIM,WDR,INCPHI,PERR,RERR)
" "
"----------------------------------------------------------------------"
"---  PERFORM ATTITUDE AND RATE ERROR FAULT DETECTIONS              ---"
"----------------------------------------------------------------------"
" "
      CALL ACFD (FPDS,FLTAB,DETECT,SWMODE,PERR,RERR,ARHDET,PFINDX)
" "
"----------------------------------------------------------------------"
"---          RAM SAFING                                           ---"
"----------------------------------------------------------------------"
" "
     AC10..CONTINUE
       IF (SWMODE.NE.6) GO TO AC20

      CALL RAMSAF (F75,INTTRS,RSTIME,RSOTIM,RSATTE,RSDBRA,RSSSIN,...
           GS1IIN,GS2IIN,RSFCII,RSSCIN,RSPARI,RSCPCI,RSCPSI,...
           ERININ,RODPII,RESTST,SADMUS,SADMRS,...
           FPDS(10),FPDS(7),FPDS(6),FPDS(5),FPDS(4),FPDS(2),FPDS(1),...
           RSSFPI,FPDS(8),SUNS,ALERT(7),ANTSEL,RSRRAX,RSRPIN,...
           RWCON,MECON,EPHCOB,SUVCAT,SUVCOM,SUVINE,IGSPN,...
           SADENA,DPH,SADMSY,SATIMR,SATIME,...
           DISSPR,ARSCI,INCPHI,SCANCI,NSL,NSU,IGSN,...
           QSX,CATVEC,ASTU,ASB1,ACK31,AGI,VAFC,VAF1,VAF2,...
           SSUUTH,SSUSMA,AUFLAG,BUFLAG,RSAESI,BODYRT,RERR,...
           PERR,RSSSRP,RSACPA,RSSSXL,RSSSZL,...
           GS1SCA,GS2SCA,VTR,CPETHR,QBI,...
           VA,VSUN1,VB,VSUN2,QDIE,QBES,QDC,DCMWDC,VHGA,...
           VANT,QHGA,QANT,VMGA,QMGA,ESA,AHGA,ADP,AMGA,VAF3,...
           EIGATH,RSEPUI,EPTIME,ECRATE,SPICMD,SADMPZ,SADMNZ,...
           CMDPAN,FGSTIM,SGSTIM,ERTIME,...
           RSOTMX,RSSARI,STGS1S,GS1CRA,ROLLDI,STGS2S,GS2CRA,...
           MTOVOI,ALERT(6),VSP,GSABFT,BBTI,...
           ODT,EIGANG,EIGAXS,AEP,QSUN1,QSUN2,VSPXER,VSPYER,...
           C1SUVF,C2SUVF,T)

AC20..CONTINUE
" "
"---------------------------------------------------------------------"
"---                   ATTITUDE CONTROL                            ---"
"---------------------------------------------------------------------"
" "
      CALL RWFAIL(MECON,RWCON,IFW,TTG,PERR,RERR,WRDR,PERW,RERW,...
                PERT,RERT)
" "
      CALL ACEXEC (F75,MECON,SWMODE,TABTRN,PPINDX,TSINDX,BODYRT,...
           DVTIM,VOITIM,CONTHR,TRNTIM,PERT,RERT,ATTLIM,...
```

```
           K6,K7,K9,K12,K13,K14,K15,K16,K17,K18,K19,K20,K21,...
           K22,K23,KA,DB1,DB2,DB3,DB4,RDB1,RDB2,RDB3,QUANT,...
           CAECMD,CATCMD,CAPCMD,CATMM,CAPMM,FPDS,DETECT,...
           DETARM,FLTAB,FIRE,THRMIS,REAACC,ENGCMD,THRCMD,...
           PULDUR,MODREQ,AP,RP,ACDUMP,T,RWCON,PERW,RERW,ARWL,...
           K24,K25,K26,K27,K28,K29,K30,WRDR,ITPAR,GR1,GR2,GR3,...
           VL,K32,WTCD,QCMVLT,TRANST,VTIMER,DTIMER,TMCLIM)
" "
"-----------------------------------------------------------------------"
"--       THRUSTER/ENGINE MODEL                                       --"
"-----------------------------------------------------------------------"
" "
"     PROCESS AT 7.5 HZ IF PHASE PLANE A OR D"
" "
      IF ((PPINDX.EQ.1.OR.PPINDX.EQ.4).AND..NOT.F75) GO TO TM600
" "
      IF (CATMM.EQ.0) GO TO TM50
      CALL CMDMM (CATMM,FIREMM,TAVGMM,THR)
      CATMM=0
      FIREMM(1)=0 $ FIREMM(2)=0 $ FIREMM(3)=0
      GC TO TM60
TM50..CONTINUE
      IF (TSINDX.EQ.1) CALL CMDTHR (THRCMD,FIRE,TAVG1A,THR)
      IF (TSINDX.EQ.2) CALL CMDENG (ENGCMD,FIRE,TAVG2B,TAVG3B,THR)
      IF (TSINDX.EQ.3) CALL CMDENG (ENGCMD,FIRE,TAVG2C,TAVG3C,THR)
TM60..CCNTINUE
" "
"     MODEL THRUSTER FAILUES"
" "
      CALL TFAIL(THR,TFACT,LEAK,DLEAK,PPINDX,MIPC,ILV1,ILV2,...
                 ES1,ES2,HWSTAT)
" "
      DC TM80 J=1,24
        FTC(1,J)=FT(1,J)*THR(J)
        FTC(2,J)=FT(2,J)*THR(J)
        FTC(3,J)=FT(3,J)*THR(J)
        FMC(J)=FTC(1,J)*FTC(1,J)+FTC(2,J)*FTC(2,J)+...
               FTC(3,J)*FTC(3,J)
        FMC(J)=SQRT(FMC(J))
  TM80..CONTINUE
" "
"-----------------------------------------------------------------------"
"---      MASS DEPLETION MODEL                                       ---"
"-----------------------------------------------------------------------"
" "
      PROP=0.0
      DO TM300 J=1,24
        PROT(J)=PROT(J)+THRSAM(PPINDX)*FMC(J)/ISP(J)
        PROP=PROP+PROT(J)
TM300..CONTINUE
        PROPM=PROPM0-PROP
        MASS=MASS0-PROP-DM
" "
"-----------------------------------------------------------------------"
"---      SRM EXPENDED MASS IF VOI                                   ---"
"-----------------------------------------------------------------------"
" "
       IF (VTIMER.LE.0.0) GO TO TM310
        TSRM=VOITIM-VTIMER
        MDOT=THRUST(TSRM)/ISPSRM
```

```
         DM=MDOT*THRSAM(PPINDX) + DM
         MASS=MASSO-PROP-DM
         RZIN=ZCM(MASS)
         ZINT=-(ZFACE+RZIN)
         LDELM=ZINT*TDELM
         RTHR(1)=0.
         RTHR(2)=SQRT(SLSQ+LDELM*LDELM)
         RTHR(3)=ZINT
TM310..CONTINUE
" "
"-----------------------------------------------------------------------"
"---        PRESSURE BLOW DOWN CURVE                                ---"
"-----------------------------------------------------------------------"
" "
        IF (.NOT.VARI) GO TO TM340

        PRES=TPRES(PLOAD-PROPM)
        DO TM320 I=1,24
         J=1
         IF (I.GT.12) J=2
         IF (I.GT.16) J=3
         FM=(YI(J)+SL(J)*PRES+QQ(J)*PRES*PRES)*LBFTON
         FT(1,I)=FM*FTUV(1,I)
         FT(2,I)=FM*FTUV(2,I)
         FT(3,I)=FM*FTUV(3,I)
TM320..CONTINUE
"-----------------------------------------------------------------------"
"---        VARIABLE INERTIA MODEL                                  ---"
"-----------------------------------------------------------------------"
" "
        RZIN=ZCM(MASS)
        IT(1,1)=IXXT(MASS)
        IT(2,2)=IYYT(MASS)
        IT(3,3)=IZZT(MASS)
        CALL M3NGOD(IT,ITM)
        CALL MATIN3(ITM,ITMI)
        DO TM330 J=1,24
         LT(3,J)=LT0(3,J)-RZIN
TM330..CONTINUE
TM340..CONTINUE

"-----------------------------------------------------------------------"
"---        TOTAL CONTROL TORQUES                                   ---"
"-----------------------------------------------------------------------"
" "
       DO TM400 J=1,24
         RXF(1,J)=LT(2,J)*FTC(3,J)-LT(3,J)*FTC(2,J)
         RXF(2,J)=LT(3,J)*FTC(1,J)-LT(1,J)*FTC(3,J)
         RXF(3,J)=LT(1,J)*FTC(2,J)-LT(2,J)*FTC(1,J)
 TM400..CONTINUE
         DO TM500 I=1,3
         TC(I)=0.0
         FC(I)=0.0
         DO TM450 J=1,24
           TC(I)=TC(I)+RXF(I,J)
           FC(I)=FC(I)+FTC(I,J)
 TM450..CONTINUE
 TM500..CONTINUE
       CALL MT3V(TTB,TC,TCB)
       CALL MT3V(TTB,FC,FCB)
```

```
      IF (VTIMER.LE.0.0) GO TO TM550
" "
"---SRM THRUST DISTURBANCES--------------------------------------------"
" "
       TSRM=VOITIM-VTIMER
       TEMP=THRUST(TSRM)
       SRMT(1)=0.0
       SRMT(2)=TEMP*SDELM
       SRMT(3)=TEMP*CDELM
       CALL VCRS(RTHR,SRMT,TSRMD)
" "
"---SRM SIDE FORCE DISTURBANCES----------------------------------------"
" "
       ALPHA=ALPHA+ALDSFR
       CALPHA=COS(ALPHA)
       TEMP=SIDEF(TSRM)
       SIDED(1)=-TEMP*SIN(ALPHA)
       SIDED(2)= TEMP*CDELM*CALPHA
       SIDED(3)=-TEMP*SDELM*CALPHA
       RSF(3)=-(RZIN+ZSIDE)
       CALL VCRS(RSF,SIDED,TSIDED)
       DO TM510 I=1,3
        FDIST(I)=SRMT(I)+SIDED(I)
        TDIST1(I)=TSRMD(I)+TSIDED(I)
TM510..CONTINUE
       TDIST2(3)=TSWIRL
       IF (SRMT(3).LE.4448.) TDIST2(3)=0.
       CALL VADD(TDIST1,TDIST2,TDIST,3)
       IF (RSSM.EQ.0) GO TO TM540
" "
"---RSS MISALIGNMENT TERMS---------------------------------------------"
" "
        RTHR1(3)=RTHR(3)
        RTHR2(2)=RTHR(2)
        CALL VCRS(RTHR1,SRMT,TSRM1)
        CALL VCRS(RTHR2,SRMT,TSRM2)
        TDIST(1)=SQRT(TSRM1(1)**2+TSRM2(1)**2+TSIDED(1)**2+TDIST2(1)**2)
        TDIST(2)=SQRT(TSRM1(2)**2+TSRM2(2)**2+TSIDED(2)**2+TDIST2(2)**2)
        TDIST(3)=SQRT(TSRM1(3)**2+TSRM2(3)**2+TSIDED(3)**2+TDIST2(3)**2)
        FDIST(1)=SQRT(SRMT(1)**2+SIDED(1)**2)
        FDIST(2)=SQRT(SRMT(2)**2+SIDED(2)**2)
        FDIST(3)=SQRT(SRMT(3)**2+SIDED(3)**2)
TM540..CONTINUE
" "
"---JET DAMPING TERM---------------------------------------------------"
" "
       CALL VCRS(W,RTHR,WXREX)
       CALL SDOT(MDOT,WXREX,MDWXR)
       CALL VCRS(RTHR,MDWXR,JDMOM)
       GO TO TM570
TM550..CONTINUE
       DO TM560 I=1,3
        TDIST(I)=0.
        FDIST(I)=0.
        JDMOM(I)=0.
TM560..CONTINUE
TM570..CONTINUE
" "
"---CONTROL/DISTURBANCE RATIOS----------------------------------------"
" "
```

```
       TDT=SQRT(TDIST(1)**2+TDIST(2)**2)
       TCT=SQRT(TCB(1)**2+TCB(2)**2)
       TRATIO=0.0
       IF(ABS(TDT).GT.0.00001) TRATIO=TCT/TDT
       TRATZ=0.0
       IF (ABS(TDIST(3)).GT.0.00001) TRATZ=ABS(TCB(3)/TDIST(3))
TM600..CONTINUE
" "
"----------------------------------------------------------------------"
"---       FAULT PROTECTION                                         ---"
"----------------------------------------------------------------------"
" "
      CALL FRCONTROL (RTI,F30,COLDST,FPDS,FLTAB,FLTOCC,FLTTHT,...
       CHUSID,HWSTAT,ARUART,ARUBRT,DELTIM,DELPER,LRSWEN,LOTHRS,...
       HITHRS,BODYRT,NETCNT,MECON,RWCON,PREMEC,PRERWC,PPINDX,...
       TSINDX,...
       DETECT,CHSLPT,TOTWT,WTSET,DELOLM,DEUPLM,MINWT,IMEDWT,...
       SWMODE,DELTAB,SSUAUS,SSUBUS,FPAS,RSPSEQ,...
       ALERT,MODREQ,CAN,CONREQ,POWREQ,PWRARU,SWGYRO,HRTREQ,...
       SRCCUR,SRCCNT,DRIFT,FPNFLT,ACTCUR,TABTRN,DEHCNT,MIDSEL,...
       ACTCNT,FSTCNT,ACFRI,IGNSEQ,CFR,QDI,QBI,ACFE,THRMIS,...
       CHREAD,ES1,ES2,TRANST,ITPAR,POWIND)
" "
"----------------------------------------------------------------------"
"---       CONFIGURATION CONTROL AT 15 HZ ONLY                      ---"
"----------------------------------------------------------------------"
" "
       IF (.NOT.F30) GO TO TH20

       CALL CONFIG (RTI,FLTOCC,CHUSID,NUMCHU,PRGYCH,POWREQ,...
                    MDSCNT,RORJCT,ARLAPW,ARUBPW,HWSTAT,OFFPWR,...
                    OBGYCP,SWGYRO,COLDST,CONREQ,ARUDES,...
                    ILVCTL,ARUART,ARLBRT,ARUAEL,ARUBEL,...
                    GYROON,MYPC,ILV1,ILV2,ES1,ES2,MIDSEL,...
                    CHSLPT,LRSWEN,GYWUTM,SSUAUS,SSUBUS,CHREAD,...
                    POWIND)

       CALL FPLOG (MAJFRM,MINFRM,RTI,FSTCNT,SRCCNT,ACTCNT,...
                   SRCCUR,ACTCUR,DEHCNT,TOTWT,POWREQ,CONREQ,...
                   FLTOCC,HWSTAT,T,SWMODE,RWCON,MECON,PPINDX,...
                   DELTAB,PRGYCH,CHSLPT)
TH20..CONTINUE
" "
"----------------------------------------------------------------------"
"---       COMMAND PROCESSOR AND SEQUENCING                         ---"
"----------------------------------------------------------------------"
" "
"      TPER = AMOD (TIME-TS0,PER) FIX TS0 SOMETIME CJG"
       TPER = AMOD (TIME,PER)
       CALL CMDSEQ(RTI,F30,TPER,C7ACT,C7BURN,C7DSAT,C7ENG,C7FUNC,...
       C7MAP,C7MMAN,C7MNVR,C7PHAS,C7PROT,C7SAP,C7SMAN,C7STOP,C7SUN,...
         C7VALV,C7VOI,C7DLTA,C7RETP,C7POWR,C7GYRO,FSEL,FSTATE,...
         MMANTM,MMANTP,PSEL,SMANSP,SMANTP,RSEL,SUNTP,MSEL,MNVRTP,...
         POWSEL,GYCMD,PHSREQ,MODREQ,CDI,QBI,...
         SWMODE,PHASE,DESMAN,CAN,ALERT,MTPTR,MNVI,RWCON,MECON,...
         STARS,FTAB,SMAP,EMAP,SUNS,DISINT,SWATH,RET,AUP,DPH,...
         POWREQ,CONREQ,INITRS,COMDI,PSREQ,PSEN1,PSEN2,ACFRI,...
         IGNSEQ,FPDS,WDR,WPDP,ROMVOI)
" "
  END $"PROCEDURAL"
```

```
 END $"OF 30 HZ CYCLE"
 DISCRETE ATTREF $"ATTITUDE REFERENCE"
 INTERVAL DIGTIM=0.133332 $"ACS FREQUENCY (7.5 HZ)"
 PROCEDURAL
" "
"------------------------------------------------------------------"
"---       INITIALIZE MANEUVER WHEN REQUIRED                     ---"
"------------------------------------------------------------------"
" "
      IF (.NOT.MNVI) GO TO DR400
      CALL MANIN(TIME,TAUM,TAUA,0.,TVO,PSFV,ASV,QDI,QREL,QSII,...
           QSFI,QDIE,QBIN,TMREF,KMV1,KMV2,WRM,WRD,EUAN,EUEV,TM,TA,...
           DESMAN,MTPTR,TREL)
      MNVI=.FALSE.
 DR400..CONTINUE
" "
"------------------------------------------------------------------"
"---       GENERATE NEW ORBIT/TRAJECTORY POSITION OF S/C         ---"
"------------------------------------------------------------------"
" "
      SCTA=SCTAM
      CALL TRJGEN(TPDT,TSO,ECS,AS,GKS,1.,RS,UVS)
      CALL SCOPOS(UVS,RS,TRNSD,NADIR,ALT,SCTAM)
" "
"------------------------------------------------------------------"
"---       DESIRED REFERENCE FOR MANEUVERS                       ---"
"------------------------------------------------------------------"
" "
      IF (SWMODE .NE. 2) GO TO DR450
      CALL MANREF(TPDT,TMREF,TM,TA,EUEV,WRM,WRD,KMV1,KMV2,...
                  QBIN,QDTM,WMVR,WRCR,DESMAN,MODREQ,CAN,ACFFI)
 DR450..CONTINUE
" "
"------------------------------------------------------------------"
"---       DESIRED REFERENCE FOR MAPPING RETRACE TURN            ---"
"------------------------------------------------------------------"
" "
      IF (SWMODE.NE. 5) GO TO DR500
      IF (.NOT.RET) GO TO DR460
      RTIM=(TMLIN(9)+(PER-TMLIN(8)))*(1.0-(TPDT-TMLIN(9))/TAUM(8))
      IF (RTIM.LT.0.0) RTIM=0.0
      CALL PLYREF(RTIM,TVO,PSFV,ASV,8,QDIR)
      CALL QDBM(QDIR,QDIRP,QRER)
      WMRR(1)=2.*SIGN(1.,QRER(4))*QRER(1)/(DIGTIM)
      WMRR(2)=2.*SIGN(1.,QRER(4))*QRER(2)/(DIGTIM)
      WMRR(3)=2.*SIGN(1.,QRER(4))*QRER(3)/(DIGTIM)
      CALL VEQL(QDIR,QDIRP,4)
      GO TO DR500
 DR460..CONTINUE
" "
"------------------------------------------------------------------"
"---       DESIRED REFERENCE FOR MAPPING                         ---"
"------------------------------------------------------------------"
" "
      IF (TPDT.LE.TMLIN(9)) TMAP=TPDT+(PER-TMLIN(8))
      IF (TPDT.GE.TMLIN(8)) TMAP=TPDT-TMLIN(8)
      CALL PLYREF(TMAP,TVO,PSFV,ASV,8,QDIV)
      CALL QDBM(QDIV,QDIVP,QRER)
      WMPR(1)=2.*SIGN(1.,QRER(4))*QRER(1)/(DIGTIM)
      WMPR(2)=2.*SIGN(1.,QRER(4))*QRER(2)/(DIGTIM)
```

```
      WMPR(3)=2.*SIGN(1.,QRER(4))*QRER(3)/(DIGTIM)
      CALL VEQL(QDIV,QDIVP,4)
 DR500..CONTINUE
" "
"-------------------------------------------------------------------"
"---       ASSIGN REFERENCE QUATERNION FOR THE CURRENT MODE      ---"
"-------------------------------------------------------------------"
" "
      IF (SWMODE .EQ. 2 ) CALL VEQL(QDIM,QDI,4)
      IF (SWMODE .EQ. 2 ) CALL VEQL(WMVR,WDR,3)
      IF (SWMODE .EQ. 5 .AND.RET) CALL VEQL(QDIR,QDI,4)
      IF (SWMODE .EQ. 5 .AND..NOT.RET) CALL VEQL(QDIV,QDI,4)
      IF (SWMODE .EQ. 1 .AND. CAN .EQ. 4) CALL VEQL(QDIE,QDI,4)
      IF (SWMODE .EQ. 5 .AND.RET) CALL VEQL(WMRR,WDR,3)
      IF (SWMODE .EQ. 5 .AND..NOT.RET) CALL VEQL(WMPR,WDR,3)
      IF (SWMODE .EQ. 1 .AND. CAN .EQ. 4) CALL VEQL(WESR,WDR,3)
      CALL EQTC(DTI,QDI)
"     CALL DUMP(NADIR,DTI)"
" "
"END OF DESIRED REFERENCE GENERATION "
 ENDS
 ENDS"OF ATTITUDE REFERENCE GENERATION DISCRETE BLOCK"
" "
 DISCRETE ARTCON $"ARTICULATION CONTROL"
 INTERVAL ARW=0.533328 $"(1.875 HZ)"
  PROCEDURAL
" "
"-------------------------------------------------------------------"
"---       ARTICULATION CONTROL                                  ---"
"-------------------------------------------------------------------"
" "
"     COMPUTE SUN AND VENUS TARGET VECTORS IN BODY FRAME"
" "
      CALL M3V(BBTI,TRGTS,ACSB)
      CALL M3V(BBTI,NADIR,ACVB)
" "
      CALL ARTIC(SUNS,ALBENA,ALBTHR,DSFTHR,DISTHR,SPX,SWMODE,AUP,...
           TSLAB,REFIND,STOPHI,STOPLO,ASL,ACK31,ACK1,ACK2,ACK3,...
           ACK4,ACK5,PHASF,SEP,SVR,SC1,SC2,SC3,SC4,SATC,SPS,DVAC,...
           DPH,SADENA,ACKSR,FRELIM,CCENTR,DLIMIT,DFMW,SFMW,...
           BTI,TRGTS,ACSB,ACVB,DPOS,MIPC,RS,DFP,SSBV,SFBV,...
           FLTAB,CMDPAN,DETECT,FPDS,ARINIT,BODYRT,ACDT,...
           SSUAUS,SSUBUS,DMWC,STPERR,ASB,DPOT,DISSPR,VSP,DPST,ABRET)

       IF (SWMODE.NE.6) GO TO AR10 $"OTHERWISE, RAMSAF PROCESSING"
         CALL RSSSUP (DISSPR,VSP,SSULTH,SSUSMA,AUFLAG,BUFLAG,...
              RSAESI,ABRET,ASB1,VSPXER,VSPYER)
         CALL RSARTP (SADMPS,SADMSI,SADMUS,SADMPL,SADENA,DPH,...
              ASL,DPST,CMDPAN,PASTPO,BODYRT,ACK31,ALBTHR,...
              ALBENA,ARW,RSPARI)
AR10..CONTINUE
" "
  END $"PROCEDURAL"
 END $"OF ARTICULATION CONTROL DISCRETE"
 DISCRETE MENTUM $"MOMENTUM MANAGEMENT"
 INTERVAL GMTTIM=1.066656 $"(0.9375 HZ)"
  PROCEDURAL
" "
"-------------------------------------------------------------------"
"---       MOMENTUM MANAGEMENT                                   ---"
```

```
"-----------------------------------------------------------------------"
" "
      CALL MOMMAN (BODYRT,COMDI,RWCON,WSPD,...
       FLTAB,FPDS,CAPMM,CATMM,IFW,SYSRAT,...
       WCX,WCY,WCZ,WDX,WDY,WDZ,PEPX,PEPY,PEPZ,...
       PENX,PENY,PENZ,PCPX,PCPY,PCPZ,PCNX,PCNY,...
       PCNZ,TPX,TNX,TPY,TNY,TPZ,TNZ,WPRE,FIREMM)
 END$"PROCEDURAL"
 END$"OF MOMENTUM MANAGEMENT DISCRETE"
" "
 DISCRETE PPTRAN $"PHASE PLANE TRANSITION DETECTION"
 INTERVAL DETTIM=0.66666 $"(1.5 HZ)"
 PROCEDURAL
" "
"-----------------------------------------------------------------------"
"---      PHASE PLANE TRANSITION DETECTION                           ---"
"-----------------------------------------------------------------------"
" "
      CALL PPDET (FPDS,FLTAB,PPINDX)
" "
 END$"PROCEDURAL"
 END$"OF PHASE PLANE TRANSITION DISCRETE"
" "
 DISCRETE DUMY $"ALLOW A 1 HZ BREAKPOINT"
 INTERVAL DUMTIM=1.0
 PROCEDURAL
      CALL DUMMY
 END$"END DUMMY PROCEDURAL"
 END$"OF DUMMY DISCRETE"

 END$"END DYNAMIC SECTION"
 END$"END OF PROGRAM"


      SUBROUTINE ATCER(QDI,QBI,PL,WDR,INCPHI,PE,RE)
C
C     THIS SUBROUTINE COMPUTES THE ATTITUDE AND RATE ERRORS FOR
C     R/W OR ACT CONTROL INPUTS.  SMALL ANGLE ROTATIONS ARE
C     ASSUMED.
C
C     TOM KELECY   5-1-84      ---- MODIFIED DMW, 8-4-86
C
C
C            INPUTS:
C              QDI = CURRENT REFERENCE ATTITUDE QUATERNION
C              QBI = CURRENT BODY ATTITUDE QUATERNION
C              PL = POSITION ERROR LIMIT MAGNITUDE
C              INCPHI = GYRO INCREMENTAL ANGLE
C              WRG - GYRO RATE
C
C            OUTPUTS:
C              PE = POSITION ERROR (RAD)


C              RE = RATE ERROR (RAD/SEC)
C
      DIMENSION QDI(4),QBI(4),QER(4),PE(3),RE(3),PL(3),WDR(3)
      REAL INCPHI(3)
```

```
C
      CALL QDBM(QDI,QBI,QER)
      CALL VUNT(QER,4)
C
      SGNQE4=-SIGN(2.0,QER(4))
C
      DO 100 I=1,3
      PE(I)=SGNQE4*QER(I)
      IF(ABS(PE(I)).GT.PL(I)) PE(I) = SIGN(PL(I),PE(I))
      RE(I)=(INCPHI(I)*30.0)-WDR(I)
  100 CONTINUE
C
      RETURN
      END
      SUBROUTINE IRUPRS(P,PP,DP)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C    INPUT
C        P -  THREE AXES GYRO ANGLE ACCUMULATION
C        PP - SAME AS THE ABOVE FROM PREVIOUS STEP
C        DP - INCREMENTAL ANGLE OF THE GYRO
C
C    OUTPUT
C        DP - INCREMENTAL ANGLE OF THE GYRO
C
C    DESCRIPTION
C                 SIMPLIFIED OBC GYRO PROCESSING
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
          DIMENSION P(3),PP(3),DP(3)
C
          DO 100 I=1,3
          DP(I)=P(I)-PP(I)
          PP(I)=P(I)
  100     CONTINUE
C
        RETURN
C
        END
       SUBROUTINE IRUPRI(P,PP,DP)
C
        DIMENSION P(3),PP(3),DP(3)
         DO 100 I=1,3
         P(I)=0.
  100    CONTINUE
        RETURN
C
        END

        SUBROUTINE ACMMEC (AT,ATTLIM,RT,PPINDX,TCMD,K6,K7,K9,K12,K13,
        1  K14,K15,K16,K17,K18, K19,K20,K21,K22,K23,
        2  AP,RP)
*
****************************************************************************
*
*       MASS EXPULSION CONTROL: PHASE PLANE ERROR GENERATION
*
*       INPUT:
*
*       AT                  ATTITUDE ERROR IN THRUSTER REFERENCE
```

```
*       ATTLIM              ATTITUDE ERROR THRUSTER LIMIT
*       RT                  RATE ERROR IN THRUSTER REFERENCE
*       PPINDX              PHASE PLANE INDEX
*       TCMD                POLARITY OF CURRENT THRUSTER FIRINGS
*       K6                  RATE DAMPING GAIN CONSTANT
*       K7                  INTEGRATOR GAIN CONSTANT
*       K9                  PSEUDO ACCELERATION GAIN CONSTANT
*       K12                 PAST PSEUDO ACCELERATION GAIN CONSTANT
*       K13                 PSEUDO ACCELERATION MAGNITUDE
*       K14..K23            FILTER COEFFICIENTS
*
*       OUTPUT:
*
*       AP                  PHASE PLANE ATTITUDE VECTOR
*       RP                  PHASE PLANE RATE VECTOR
*
************************************************************************
*
        INTEGER PPINDX, TCMD(3)
        REAL AT(3), ATTLIM(3), RT(3)
        REAL K6(3,4), K7(3,4), K9(3,4), K12(3,4), K13(3,4)
        REAL K14,K15,K16,K17,K18,K19,K20,K21,K22,K23
        REAL AP(3), RP(3)
*
*       LOCAL VARIABLES
*
        REAL VI(3), VIL(3), VPA(3), VPAL(3), AR(3), RR(3)
        REAL ARZ1(3), ARZ2(3), APZ1(3), APZ2(3)
        REAL RRZ1(3), RRZ2(3), RPZ1(3), RPZ2(3)
        DATA VIL/3*0.0/, VPAL/3*0.0/
        DATA ARZ1,ARZ2,APZ1,APZ2/12*0.0/
        DATA RRZ1,RRZ2,RPZ1,RPZ2/12*0.0/
*
*       LIMIT ATTITUDE ERROR
*
        DO I=1,3
          IF (ABS(AT(I)) .GT. ATTLIM(I)) AT(I) = SIGN(ATTLIM(I),AT(I))
*
*       CALCULATE INTEGRAL VECTOR
*
          VI(I) = K7(I,PPINDX)*AT(I) + VIL(I)
*
*       CONVERT ATTITUDE ERROR TO PHASE PLANE ATTITUDE VECTOR
*
          AR(I) = AT(I) + VI(I) + K6(I,PPINDX)*RT(I)
*
*       CALCULATE PSEUDO ACCELERATION VECTOR
*
          VPA(I) = TCMD(I)*K9(I,PPINDX)*K13(I,PPINDX)-K12(I,PPINDX)*VPAL(I)
*
*       CONVERT RATE ERROR TO PHASE PLANE RATE VECTOR
*
          RP(I) = RT(I) + VPA(I)
*
*       FILTER PHASE PLANE ATTITUDE VECTOR
*
          AP(I) = K14*AR(I) + K15*ARZ1(I) + K16*ARZ2(I) -
     1            K17*APZ1(I) - K18*APZ2(I)
*
*       FILTER PHASE PLANE RATE VECTOR
```

```
*
          RP(I) = K19*RR(I) + K20*RRZ1(I) + K21*RRZ2(I) -
        1          K22*RPZ1(I) - K23*RPZ2(I)
*
*       BUBBLE PAST VALUES
*
          ARZ2(I)=ARZ1(I)
          ARZ1(I)=AR(I)
          APZ2(I)=APZ1(I)
          APZ1(I)=AP(I)

          RRZ2(I)=RRZ1(I)
          RRZ1(I)=RR(I)
          RPZ2(I)=RPZ1(I)
          RPZ1(I)=RP(I)

          VIL(I)=VI(I)
          VPAL(I)=VPA(I)

        ENDDO

        RETURN
        END

        SUBROUTINE AC+PP+TS (AP,RP,PPINDX,TSINDX,KA,A1,A2,A3,A4,R1,R2,R3,
        1  QUANT,
        2  FIRE,TCMD,ACECMD,ACTCMD,ACPCMD)
*
****************************************************************************
*
*       MASS EXPULSION CONTROL PHASE PLANE
*
*       INPUT:
*
*       AP                FILTERED PHASE PLANE ATTITUDE VECTOR
*       RP                FILTERED PHASE PLANE RATE VECTOR
*       PPINDX            PHASE PLANE INDEX
*       KA                PARABOLIC TIME CONSTANT
*       TSINDX            THRUSTER SELECT INDEX
*       A1..A4            PHASE PLANE ATTITUDE REGIONS
*       R1..R3            PHASE PLANE RATE REGIONS
*       QUANT             PHASE PLANE ATTITUDE ERROR QUANTIZATION
*
*       OUTPUT:
*
*       FIRE              PHASE PLANE REGION
*       TCMD              POLARITY OF ENGINE/THRUSTER FIRING
*       ACECMD            ATTITUDE CONTROL ENGINE COMMAND
*       ACTCMD            ATTITUDE CONTROL THRUSTER COMMAND
*       ACPCMD            ATTITUDE CONTROL PULSE DURATION SELECT
*
****************************************************************************
*
        REAL AP(3), RP(3), A1(3,4), A2(3,4), A3(3,4), A4(3,4)
        REAL R1(3,4), R2(3,4), R3(3,4), R4(3,4), KA(3,4)
        REAL QUANT
        INTEGER PPINDX, TSINDX, FIRE(3), TCMD(3), TSTAB(9,7,3)
        INTEGER ACECMD, ACTCMD, ACPCMD
*    DEFINITION OF THRUSTER SELECT TABLE: TSTAB(TSINDX+AXIS,FIRE,TYPE)
        DATA TSTAB /'00C0'X, '0003'X, '0500'X, '0000'X, '0000'X, ! -3 PP
```

```
     1                 '0000'X, '0000'X, '0000'X, '0000'X,            ! THRCMD
     1                 '00C0'X, '0003'X, '0500'X, '0000'X, '0000'X,   ! -2 PP
     1                 '0000'X, '0000'X, '0000'X, '0000'X,            ! THRCMD
     1                 '00C0'X, '0003'X, '0500'X, '0000'X, '0000'X,   ! -1 PP
     1                 '0000'X, '0000'X, '0000'X, '0000'X,            ! THRCMD
     1                 '00CC'X, '0033'X, '0F00'X, '0000'X, '0000'X,   !  0 PP
     1                 '0000'X, '0000'X, '0000'X, '0000'X,            ! THRCMD
     1                 '000C'X, '0030'X, '0A00'X, '0000'X, '0000'X,   ! +1 PP
     1                 '0000'X, '0000'X, '0000'X, '0000'X,            ! THRCMD
     1                 '000C'X, '0030'X, '0A00'X, '0000'X, '0000'X,   ! +2 PP
     1                 '0000'X, '0000'X, '0000'X, '0000'X,            ! THRCMD
     1                 '000C'X, '0030'X, '0A00'X, '0000'X, '0000'X,   ! +3 PP
     1                 '0000'X, '0000'X, '0000'X, '0000'X,            ! THRCMD
*
     2                 '0000'X, '0000'X, '0000'X, '0C00'X, '0030'X,   ! -3 PP
     2                 '0005'X, '0C00'X, '0030'X, '0005'X,            ! ENGCMD
     2                 '0000'X, '0000'X, '0000'X, '0400'X, '0010'X,   ! -2 PP
     2                 '0001'X, '0C00'X, '0030'X, '0005'X,            ! ENGCMD
     2                 '0000'X, '0000'X, '0000'X, '0400'X, '0010'X,   ! -1 PP
     2                 '0001'X, '0400'X, '0010'X, '0001'X,            ! ENGCMD
     2                 '0000'X, '0000'X, '0000'X, '0CC0'X, '0330'X,   !  0 PP
     2                 '000F'X, '0CC0'X, '0330'X, '000F'X,            ! ENGCMD
     2                 '0000'X, '0000'X, '0000'X, '0040'X, '0100'X,   ! +1 PP
     2                 '0002'X, '0040'X, '0100'X, '0002'X,            ! ENGCMD
     2                 '0000'X, '0000'X, '0000'X, '0040'X, '0100'X,   ! +2 PP
     2                 '0002'X, '00C0'X, '0300'X, '000A'X,            ! ENGCMD
     2                 '0000'X, '0000'X, '0000'X, '00C0'X, '0300'X,   ! +3 PP
     2                 '000A'X, '00C0'X, '0300'X, '000A'X,            ! ENGCMD
*
     3                 '000F'X, '00F0'X, '0F00'X, '0003'X, '0030'X,   ! -3 PP
     3                 '0300'X, '0003'X, '0030'X, '0300'X,            ! PULDUR
     3                 '0003'X, '0030'X, '0300'X, '0003'X, '0030'X,   ! -2 PP
     3                 '0300'X, '0002'X, '0020'X, '0200'X,            ! PULDUR
     3                 '0001'X, '0010'X, '0100'X, '0002'X, '0020'X,   ! -1 PP
     3                 '0200'X, '0002'X, '0020'X, '0200'X,            ! PULDUR
     3               9*'0000'X,                                       !  0 PP
     3                 '0001'X, '0010'X, '0100'X, '0002'X, '0020'X,   ! +1 PP
     3                 '0200'X, '0002'X, '0020'X, '0200'X,            ! PULDUR
     3                 '0003'X, '0030'X, '0300'X, '0003'X, '0030'X,   ! +2 PP
     3                 '0300'X, '0002'X, '0020'X, '0200'X,            ! PULDUR
     3                 '000F'X, '00F0'X, '0F00'X, '0003'X, '0030'X,   ! +3 PP
     3                 '0300'X, '0003'X, '0030'X, '0300'X  /          ! PULDUR
*   LOCAL VARIABLES
      REAL AM(3), SLOPE
      DATA AM/3*0.0/
*
      J=PPINDX
      ACECMD=0
      ACTCMD=0
      ACPCMD=0

      DO I=1,3
      SLOPE=-SIGN(KA(I,J),AP(I))*SQRT(ABS(AP(I)))
        IF ( (AP(I).GT.A4(I,J) .AND.
     1        RP(I).GT.SLOPE) .OR.
     2       (AP(I).GT.A3(I,J) .AND.
     3        RP(I).GT.R1(I,J)) .OR. (RP(I).GT.R3(I,J) .AND.
     4      RP(I).GT.SLOPE)) THEN
          FIRE(I)=-3
```

```
          ELSE IF ((RP(I).GT.R2(I,J) .AND.
        1           RP(I).GT.SLOPE) .OR.
        2          (RP(I).GT.0.0 .AND. AP(I).GT.A2(I,J))) THEN
            FIRE(I)=-2

          ELSE IF (RP(I).GT.0.0 .AND. AP(I).GT.A1(I,J)) THEN
            FIRE(I)=-1

          ELSE IF ((AP(I).LT.-A4(I,J) .AND.
        1           RP(I).LT.SLOPE) .OR.
        2          (AP(I).LT.-A3(I,J) .AND.
        3           RP(I).LT.-R1(I,J) ) .OR.
        4          (RP(I).LT.-R3(I,J).AND.RP(I).LT.SLOPE)) THEN
            FIRE(I)=3

          ELSE IF ((RP(I).LT.-R2(I,J) .AND.
        1           RP(I).LT.SLOPE) .OR.
        2          (RP(I).LT.0.0 .AND. AP(I).LT.-A2(I,J) )) THEN
            FIRE(I)=2

          ELSE IF (RP(I).LT.0.0 .AND. AP(I).LT.-A1(I,J)) THEN
            FIRE(I)=1

          ELSE

            FIRE(I)=0

          ENDIF
*
*       APPLY PHASE PLANE PULSE QUANTUM
*
          IF (ABS(FIRE(I)).EQ.1 .OR. ABS(FIRE(I)).EQ.2) THEN
            IF (ABS(AP(I)-AM(I)).LT.QUANT) THEN
              FIRE(I)=0
            ELSE
              AM(I)=AP(I)
            ENDIF
          ELSE
            AM(I)=0.0
          ENDIF
*
*       DETERMINE POLARITY OF FIRING
*
          TCMD(I)=0
          IF (FIRE(I).GT.0) TCMD(I)=1
          IF (FIRE(I).LT.0) TCMD(I)=-1
*
*       SELECT RCS CONTROL COMMANDS FROM PHASE PLANE REGIONS
*
          ACTCMD=TSTAB(3*(TSINDX-1)+I,FIRE(I)+4,1) .OR. ACTCMD
          ACECMD=TSTAB(3*(TSINDX-1)+I,FIRE(I)+4,2) .OR. ACECMD
          ACPCMD=TSTAB(3*(TSINDX-1)+I,FIRE(I)+4,3) .OR. ACPCMD

        ENDDO

        RETURN
        END


        SUBROUTINE TPROFILE (T1,T2,T3,T4,PDELAY,NP,PULSE,TAUS)
*
*************************************************************************
```

```
*       PURPOSE:  SETS UP THRUSTER/ENGINE THRUST LEVEL TABLES FOR EACH
*                 11 msec INCREMENT OF CONTROL CYCLE
*
*       INPUT:
*                       T2  |<----------->|
*       T1              T1  |             ------------
*       T2                  |<-------->|  /            \
*       T3                  |          | /              \
*       T4                  0++++++0+++/                 \+++++++++
*                          ON     OFF            |  |
*                              T3 |<------------------>|  |
*                              T4 |<---------------------->|
*
*       PDELAY          PROCESSING DELAY
*       NP              NUMBER OF 11 ms INCREMENTS IN CONTROL CYCLE
*       PULSE           VECTOR OF PULSEWIDTHS CORRESPONDING TO PHASE
*                       PLANE REGIONS 0,1,2,3 IN 11 ms INCREMENTS
*
*       OUTPUT:
*
*       TAVG            ARRAY OF THRUSTER/ENGINE FORCE OVER CONTROL CYCLE
*                       INDEXED BY PHASE PLANE REGIONS
*
*****************************************************************************
*
        REAL TAVE(4,4,4,12)
        REAL TAVG(4,4,4)
        REAL T1, T2, T3, T4, PDELAY
        INTEGER NP, PULSE(4)
*
*       LOCAL VARIABLES
*
        REAL TCYCLE
        DATA DT/.0011/
*
*       INITIALIZE RISE AND FALL SLOPE
*
        RISE=1/(T2-T1)
        FALL=1/(T4-T3)
*
*       INITIALIZE CONTROL CYCLE PERIOD
*
        TCYCLE=NP*0.011

        DO 500 IP3=0,3
        DO 400 IP2=0,3
        DO 300 IP1=0,3
*
*       SET UP ACTUAL ENGINE/THRUSTER ON/OFF TIMES FOR EACH PULSE
*
          ON3=PDELAY+T1
          ON2=PDELAY+T1+TCYCLE
          ON1=PDELAY+T1+2*TCYCLE
          OF3=PDELAY+0.011*PULSE(IP3+1)+T4
          OF2=PDELAY+0.011*PULSE(IP2+1)+T4+TCYCLE
          OF1=PDELAY+0.011*PULSE(IP1+1)+T4+2*TCYCLE
*
*       SET UP RISE AND FALL TIMES FOR EACH PULSE
*
          TU3=T2-T1   ! IP3 RISE
```

```
      TD3=T4-T3   ! IP3 FALL
      TU2=TU3     ! IP2 RISE
      TD2=TD3     ! IP2 FALL
      TU1=TU3     ! IP1 RISE
      TD1=TD3     ! IP1 FALL
      TM3=1.0     !
      TM2=1.0     ! MAX THRUST
      TM1=1.0     !
*
*    CHECK FOR PULSE MODE OPERATION
*
      IF (OF3-ON3.LT.TU3+TD3) THEN ! IP3 NEVER REACHES FULL ON (PULSE MOD
        TU3=0.011*PULSE(IP3+1)+T3-T1
        TM3=TU3*RISE                ! MAXIMUM THRUST LEVEL REACHED
        TD3=TM3/FALL
        OF3=ON3+TU3+TD3
      ENDIF

      IF (OF2-ON2.LT.TU2+TD2) THEN ! IP2 NEVER REACHES FULL ON (PULSE MOD
        TU2=0.011*PULSE(IP2+1)+T3-T1
        TM2=TU2*RISE                ! MAXIMUM THRUST LEVEL REACHED
        TD2=TM2/FALL
        OF2=ON2+TU2+TD2
      ENDIF

      IF (OF1-ON1.LT.TU1+TD1) THEN ! IP2 NEVER REACHES FULL ON (PULSE MOD
        TU1=0.011*PULSE(IP1+1)+T3-T1
        TM1=TU1*RISE                ! MAXIMUM THRUST LEVEL REACHED
        TD1=TM1/FALL
        OF1=ON1+TU1+TD1
      ENDIF
*
*    INITIALIZE TIME FOR INTEGRATION AT BEGINNING OF CURRENT CYCLE
*
      T=2*TCYCLE+.5*DT
*
*    LOOP THROUGH NP 11 ms CHUNKS
*
      DO 200 K=1,NP ! 3 OR 11 FOR ENG FOR THRUSTER
        ATHR=0.0     ! ZERO OUT ACCUMULATED THRUST
*
*    INTEGRATE EACH 11 msec CHUNK WITH 1/10 INCREMENT
*
        DO 100 L=1,10

         Y=0.0
*
*    CHECK IP3 PULSE (SECOND PREVIOUS CYCLE)
*
          IF (IP3.NE.0) THEN     ! THERE IS A PULSE; CHECK FOR CONTRIBUTION
           IF (T.GT.ON3) THEN                         ! PULSE HAS STARTED
             IF (T.LT.ON3+TU3) THEN                   ! PULSE ON THE RISE
               Y=(T-ON3)*RISE
             ELSE IF (T.LT.OF3-TD3) THEN              ! PULSE STEADY STATE
               Y=1.0
             ELSE IF (T.LT.OF3) THEN                  ! CHECK FOR OVERLAP
               IF (IP2.GT.0.AND.ON2.LE.OF3) THEN      ! CONTINUOUS ON
                 Y=1.0
               ELSE                                   ! PULSE ON THE FALL
                 Y=TM3-(T-OF3+TD3)*FALL
```

```
                  ENDIF
                ENDIF
              ENDIF
             ENDIF
*
*       CHECK IP2 PULSE (PREVIOUS CYCLE)
*
             IF (IP2.NE.0) THEN
              IF (T.GT.ON2) THEN
                IF (T.LT.ON2+TU2) THEN
                  IF (IP3.GT.0.AND.ON2.LE.OF3) THEN
                    Y=1.0
                  ELSE
                    Y=(T-ON2)*RISE
                  ENDIF
                ELSE IF (T.LT.OF2-TD2) THEN
                  Y=1.0
                ELSE IF (T.LT.OF2) THEN
                  IF (IP1.GT.0.AND.ON1.LE.OF2) THEN
                    Y=1.0
                  ELSE
                    Y=TM2-(T-OF2+TD2)*FALL
                  ENDIF
                ENDIF
              ENDIF
             ENDIF
*
*       CHECK IP1 PULSE (CURRENT CYCLE)
*
             IF (IP1.NE.0) THEN
              IF (T.GT.ON1) THEN
                IF (T.LT.ON1+TU1) THEN
                  IF (IP2.GT.0.AND.ON1.LE.OF2) THEN
                    Y=1.0
                  ELSE
                    Y=(T-ON1)*RISE
                  ENDIF
                ELSE IF (T.LT.OF1-TD1) THEN
                  Y=1.0
                ELSE IF (T.LT.OF1) THEN
                  Y=TM1-(T-OF1+TD1)*FALL
                ENDIF
              ENDIF
             ENDIF
*
*       SUM CONTRIBUTIONS
*
            ATHR=ATHR+Y*DT
*
*       INCREMENT TIME
*
            T=T+DT

100         CONTINUE
*
*       LOAD AVERAGE THRUST ARRAY
*
            TAVE(IP3+1,IP2+1,IP1+1,K)=ATHR/0.011

200       CONTINUE
```

```
300      CONTINUE
400      CONTINUE
500      CONTINUE

         DO IP3=0,3
           DO IP2=0,3
             DO IP1=0,3
               TAVG(IP3+1,IP2+1,IP1+1)=0.0
                DO K=1,NP
                  TAVG(IP3+1,IP2+1,IP1+1)=TAVG(IP3+1,IP2+1,IP1+1)+
     1                                      TAVE(IP3+1,IP2+1,IP1+1,K)
                ENDDO
                TAVG(IP3+1,IP2+1,IP1+1)=TAVG(IP3+1,IP2+1,IP1+1)/NP
             ENDDO
           ENDDO
         ENDDO

2000     FORMAT(1X,3I3,3F12.3)

         RETURN
         END
```

```
      SUBROUTINE CMDTHR (THRCMD,FIRE,TAVG,THR)
*
**********************************************************************
*
*     COMPUTES AVERAGE THRUST OVER CONTROL CYCLE
*
*     INPUT:
*
*     THRCMD          THRUSTER CONTROL COMMAND
*     FIRE            PHASE PLANE REGION OF EACH CONTROL AXIS
*     TAVG            ARRAY OF AVERAGE EFFECTOR FORCES INDEXED BY
*                     CURRENT AND PAST PHASE PLANE REGIONS
*
*     OUTPUT:
*
*     THR             AVERAGE THRUST OVER CONTROL CYCLE PER THRUSTER
*
**********************************************************************
*
      INTEGER THRCMD, FIRE(3)
      REAL TAVG(4,4,4), THR(24)
*
*   LOCAL VARIABLES
*
      INTEGER IPP(24), IPP1(24), IPP2(24)
      DATA IPP/24*0/, IPP1/24*0/, IPP2/24*0/
*
*     DECODE PHASE PLANE REGION FOR EACH EFFECTOR
*
        IPP(1)=IABS(BTEST(THRCMD,0)*FIRE(2))
        IPP(2)=IABS(BTEST(THRCMD,1)*FIRE(2))
        IPP(3)=IABS(BTEST(THRCMD,2)*FIRE(1))
        IPP(4)=IABS(BTEST(THRCMD,3)*FIRE(1))
        IPP(5)=IABS(BTEST(THRCMD,4)*FIRE(2))
        IPP(6)=IABS(BTEST(THRCMD,5)*FIRE(2))
        IPP(7)=IABS(BTEST(THRCMD,6)*FIRE(1))
        IPP(8)=IABS(BTEST(THRCMD,7)*FIRE(1))
        IPP(9)=IABS(BTEST(THRCMD,8)*FIRE(3))
        IPP(10)=IABS(BTEST(THRCMD,9)*FIRE(3))
        IPP(11)=IABS(BTEST(THRCMD,10)*FIRE(3))
        IPP(12)=IABS(BTEST(THRCMD,11)*FIRE(3))

*       IPP(13)=IABS(BTEST(ENGCMD,0)*FIRE(3))
*       IPP(14)=IABS(BTEST(ENGCMD,1)*FIRE(3))
*       IPP(15)=IABS(BTEST(ENGCMD,2)*FIRE(3))
*       IPP(16)=IABS(BTEST(ENGCMD,3)*FIRE(3))
*       IPP(17)=IABS(BTEST(ENGCMD,4)*FIRE(2))
*       IPP(18)=IABS(BTEST(ENGCMD,5)*FIRE(2))
*       IPP(19)=IABS(BTEST(ENGCMD,6)*FIRE(1))
*       IPP(20)=IABS(BTEST(ENGCMD,7)*FIRE(1))
```

```
*           IPP(21)=IABS(BTEST(ENGCMD,8)*FIRE(2))
*           IPP(22)=IABS(BTEST(ENGCMD,9)*FIRE(2))
*           IPP(23)=IABS(BTEST(ENGCMD,10)*FIRE(1))
*           IPP(24)=IABS(BTEST(ENGCMD,11)*FIRE(1))

            DO J=1,12
              THR(J)=TAVG(IPP2(J)+1,IPP1(J)+1,IPP(J)+1)
              IPP2(J)=IPP1(J)
              IPP1(J)=IPP(J)
            ENDDO

            DO J=13,24
              THR(J)=0.0
            ENDDO

          RETURN
          END
          SUBROUTINE CMDENG (ENGCMD,FIRE,TAVG2,TAVG3,THR)
*
***************************************************************************
*
*         COMPUTES AVERAGE THRUST OVER CONTROL CYCLE
*
*         INPUT:
*
*         ENGCMD             ENGINE CONTROL COMMAND
*         FIRE               PHASE PLANE REGION OF EACH CONTROL AXIS
*         TAVG2              ARRAY OF AVERAGE ENGINE FORCES INDEXED BY
*                            CURRENT AND PAST PHASE PLANE REGIONS (5%)
*         TAVG3              ARRAY OF AVERAGE ENIGNE FORCES INDEXED BY
*                            CURRENT AND PAST PHASE PLANE REGIONS (100%)
*
*         OUTPUT:
*
*         THR                AVERAGE THRUST OVER CONTROL CYCLE PER ENGINE
*
***************************************************************************
*
          INTEGER ENGCMD, FIRE(3)
          REAL TAVG2(4,4,4), TAVG3(4,4,4), THR(24)
*
*     LOCAL VARIABLES
*
          INTEGER IPP(24), IPP1(24), IPP2(24)
          DATA IPP/24*0/, IPP1/24*0/, IPP2/24*0/
*
*         DECODE PHASE PLANE REGION FOR EACH EFFECTOR
*
            IPP(13)=IABS(BTEST(ENGCMD,0)*FIRE(3))
            IPP(14)=IABS(BTEST(ENGCMD,1)*FIRE(3))
            IPP(15)=IABS(BTEST(ENGCMD,2)*FIRE(3))
            IPP(16)=IABS(BTEST(ENGCMD,3)*FIRE(3))
            IPP(17)=IABS(BTEST(ENGCMD,4)*FIRE(2))
            IPP(18)=IABS(BTEST(ENGCMD,5)*FIRE(2))
            IPP(19)=IABS(BTEST(ENGCMD,6)*FIRE(1))
            IPP(20)=IABS(BTEST(ENGCMD,7)*FIRE(1))
            IPP(21)=IABS(BTEST(ENGCMD,8)*FIRE(2))
            IPP(22)=IABS(BTEST(ENGCMD,9)*FIRE(2))
            IPP(23)=IABS(BTEST(ENGCMD,10)*FIRE(1))
            IPP(24)=IABS(BTEST(ENGCMD,11)*FIRE(1))
```

```
        DO J=13,24
          IF (J.GT.16) THEN
            THR(J)=TAVG3(IPP2(J)+1,IPP1(J)+1,IPP(J)+1)
          ELSE
            THR(J)=TAVG2(IPP2(J)+1,IPP1(J)+1,IPP(J)+1)
          ENDIF
          IPP2(J)=IPP1(J)
          IPP1(J)=IPP(J)
        ENDDO

        DO J=1,12
          THR(J)=0.0
        ENDDO

      RETURN
      END
```

```
C
      SUBROUTINE RLC (DB1,DB2,DB3,PHIR,ANG,RATE,FIRE,SLOP,RDB2,RDB3)
C
C   THIS SUBROUTINE SIMULATES AN RLC CONTROLLER WITH OPTIONAL
C      SHORT PULSE REGIONS.  FOR DEFINITION OF VARIABLES, SEE
C      WHITE, COLBURN, AND BOLAND, "DESIGN ANALOGY BETWEEN
C      OPTIMAL TIME-FUEL AND RATE-LEDGE RELAY CONTROLLERS",
C      JOURNAL OF SPACECRAFT AND ROCKETS, 1976.
C
       REAL DB1, PHIR, ANG, RATE, SLOP, LEDGE
       REAL LIMAT, RDB2, RDB3, DB2, DB3
       REAL PPANG
       INTEGER FIRE
```

```
C
C    RATE-LEDGE CONTROLLER
C
         IF (ANG.GT.PHIR) THEN
            LIMAT = PHIR
         ELSE IF (ANG.LT.-PHIR) THEN
            LIMAT = -PHIR
         ELSE
            LIMAT = ANG
         ENDIF
         PPANG = -((SLOP * RATE) + LIMAT)
         IF (PPANG.GT.DB1) THEN
            FIRE = 3
         ELSE IF (PPANG.LT.-DB1) THEN
            FIRE = -3
         ELSE
            FIRE = 0
         ENDIF
C
C    SHORT PULSE REGIONS
C
        IF (DB2.NE.0.0) THEN
         IF (FIRE.EQ.-3) THEN
            IF (ANG.LT.DB2.AND.RATE.LT.RDB2) THEN
               FIRE = -1
            ELSE IF (ANG.LT.DB3.AND.RATE.LT.RDB3) THEN
               FIRE = -2
            ENDIF
         ENDIF
         IF (FIRE.EQ.3) THEN
            IF (ANG.GT.-DB2.AND.RATE.GT.-RDB2) THEN
               FIRE = 1
            ELSE IF (ANG.GT.-DB3.AND.RATE.GT.-RDB3) THEN
               FIRE = 2
            ENDIF
         ENDIF
        ENDIF
      RETURN
      END
C
      SUBROUTINE DBAND(INP,FIRE,DB)
C
C    THIS SUBROUTINE IMPLEMENTS A SIMPLE DEADBAND FOR THRUSTER CONTROL.
C
      REAL INP, DB
      INTEGER FIRE
C
      IF (INP.GE.DB) THEN
         FIRE = -3
      ELSE IF (INP.LE.-DB) THEN
         FIRE = 3
      ELSE
         FIRE = 0
      ENDIF
      RETURN
      END
C
      SUBROUTINE OPTIMAL(T,ATT,RAT,K,SAMP,FIRE)
C
C    THIS SUBROUTINE IMPLEMENTS A SINGLE-STEP SUBOPTIMAL CONTROLLER
```

```
C      FOR AN ATTITUDE CONTROL SYSTEM.
C
C   INPUTS:
C      ATT = CURRENT ANGULAR ERROR (RAD)
C      RAT = CURRENT RATE ERROR (RAD/SEC)
C      K = SQUARE ROOT OF TIME-FUEL WEIGHTING FACTOR
C      SAMP = SAMPLING TIME
C      I = INDICATOR OF WHICH CONTROL AXIS WE'RE DEALING WITH
C
C   OUTPUTS:
C      FIRE = PHASE PLANE REGION (-12 TO 12)
C
C   INTERNAL:
C      PRINERT = PRINCIPLE INERTIAS IN CONTROL FRAME
C      NU1,NU2 = PARAMETERS OF THE OPTIMAL CONTROL
C      TERM = OPTIMAL CONTROL HISTORY TERMINAL TIMEME
C      Y = SIGN OF CONTROL TORQUE
C      MULT = INDICATOR OF FLIP ON SIGN OF CONTROL TORQUE
C      GAMMA = TORQUE CAPACITY OF THRUSTERS
C
      REAL ATT, RAT, K, SAMP, PRINERT(3), NU1, NU2, TERM
      REAL Y, MULT, GAMMA(3), QUAD, PULTIME
      INTEGER I, FIRE
      DATA PRINERT /1525.0,1525.0,1600.0/
      DATA GAMMA /1.7487,1.7487,2.5551/
C
C   CALCULATE NU1
C
      NU1 = -2.0 * PRINERT(I) * K
C
C   CALCULATE TERMINAL TIME.  IF ATT < 0, THIS MAY CAUSE PROBLEMS,
C      SO IF SO WE FLIP THE SIGN OF BOTH ATT AND RAT AND SET
C      MULT = -1
C
      IF (ATT.LT.0.0) THEN
         MULT = -1.0
      ELSE
         MULT = 1.0
      ENDIF
      QUAD = SQRT((RAT*PRINERT(I))**2+(6.0*K*PRINERT(I)*MULT*ATT))
      TERM = (MULT*RAT*PRINERT(I)+QUAD) / K
C
C      CALCULATE NU2, Y, AND PULTIME
C
      IF (TERM.GT.0.0) THEN
         NU2=(4.0*PRINERT(I)*PRINERT(I)*MULT*RAT-2.0*NU1*TERM)/TERM**2
         Y = SIGN (1.0,-(NU1*SAMP-(NU2*(0.5*SAMP*SAMP-TERM*SAMP))))
         Y = Y * MULT
         PULTIME = ABS((1.0/(2.0*PRINERT(I)*GAMMA(I)))*
     1                   (NU1*SAMP-NU2*(0.5*SAMP*SAMP-TERM*SAMP)))
      ELSE
         Y = 0.0
         PULTIME = 0.0
      END IF
C
C   CALCULATE FIRE
C
      FIRE = INT(Y*PULTIME/0.011)
      IF (FIRE.GT.12) THEN
         FIRE = 12
```

```
      ELSE IF (FIRE.LT.-12) THEN
         FIRE = -12
      ENDIF
      RETURN
      END
```

References

[1] Alldridge, W.J.
Limit Cycle Stability of Systems Containing a Pulse Ratio Modulator.
Aeronautical Journal, October, 1978.

[2] Bell, C.E.
Injection Module Thrust Vector Control for Galileo and International Solar Polar Missions.
October, 1982.
Jet Propulsion Lab Internal Memorandum.

[3] Bittner, H., Fisher, H.D., and Surauer, M.
Design of Reaction Jet Attitude Control Systems for Flexible Spacecraft.
In *Proceedings of the Ninth Symposium on Automatic Control in Space*. Pergamon Press, 1983.

[4] Clark, R.N., Dumas, P., and Fosth, D.C.
Scheme to Improve Limit Cycle Performance of an Attitude Control System.
Journal of Spacecraft and Rockets, April, 1975.

[5] Dodds, S.J.
A Predicted Signed Switching Time High Precision Satellite Attitude Control Law.
International Journal of Control (5), 1984.

[6] Floyd, Michel A., Brown, Mark E., Turner, James D., and Vander Velde, Wallace E.
Implementation of a Minimum Time and Fuel On/Off Thruster Control System for a Flexible Spacecraft.
In *AAS/AIAA Astrodynamics Specialist Conference*. AAS/AIAA, August, 1983.

[7] Floyd, Michel A.
Single-Step Optimal Control of the RPL Experiment.
In *Rocky Mountain Guidance and Control Conference*. February, 1985.

[8] Francis, Gerald W.
Venus Radar Mapper, VOI Control Analysis.
October, 1983.
Martin Marietta Internal Memorandum.

[9] Hughes, W.G.
Improved Design Procedures for Precision Attitude Control of a Spacecraft Using Pure Jet Control Torquing.
In *Proceedings of the Fourth Symposium on Automatic Control in Space*. Izdavacko Preduzece Tehnika, 1971.

[10] Kelecy, Thomas M.
Venus Radar Mapper Attitude Control Performance Analysis Report.
August, 1985.
Martin Marietta Internal Memorandum.

[11] Larson, Victor.
A Suboptimal Stochastic Controller for an N-Body Spacecraft.
In *AAS/AIAA Astrodynamics Conference.* July, 1973.

[12] Liu, T.C., Thompson, Zack, and Fisher, P.H.
Optimum Propellant Usage for Reaction Jet Systems of Space Vehicles.
In *Proceedings of the Fifteenth Joint Automatic Control Conference.* American Institute of Chemical Engineers, 1974.

[13] Loesch, F.C., and Hecht, C.
Work/Energy Analysis of Bending Limit Cycles in a Deadband Attitude Control System.
Journal of Spacecraft and Rockets , July, 1976.

[14] Loesch, Franklin C.
Deadband Control System Limit Cycles Analyzed by Fourier Series.
Journal of Guidance, Control, and Dynamics , May, 1982.

[15] Martin, M.W., and Kubiak, E.T.
A New Design Approach to Achieve a Minimum Impulse Limit Cycle in the Presence of Significant Measurement Uncertainties.
In *Guidance and Control Conference.* American Institute of Aeronautics and Astronautics, August, 1982.

[16] Millar, R.A., and Vigneron, F.R.
Attitude Stability of a Pseudorate Jet-Controlled Flexible Spacecraft.
Journal of Guidance and Control , March, 1979.

[17] Ogata, Katsuhiko.
Modern Control Engineering.
Prentice-Hall, 1970.

[18] Quam, David Lawrence.
A No-Limit-Cycle Digital Computer Algorithm for Pulsed Attitude Control of a Flexible Spacecraft.
PhD thesis, University of Washington, 1975.

[19] Thomson, W.T., and Reiter, G.S.
Jet Damping of a Solid Rocket: Theory and Flight Results.
AIAA Journal , March, 1965.

[20] Vander Velde, Wallace E., and He, Juan.
Design of Space Structure Control Systems Using On-Off Thrusters.
Journal of Guidance and Control , 1983.

[21] Jet Propulsion Laboratory.
Voyager 1977 Propulsion Module Thrust Vector Control.
Technical Report 618-420, Jet Propulsion Laboratory, August, 1977.

[22] Weisenberg, David.
An Approximation of the Dynamic Effects of Propellant Slosh.
July, 1986.
Martin Marietta Internal Memorandum.

[23] Weisenberg, David.
Analysis of Magellan Propellant Slosh.
November, 1986.
Martin Marietta Internal Memorandum.

[24] Wertz, James R. ed.
Spacecraft Attitude Determination and Control.
D. Reidel Publishing Company, 1985.

[25] White, L.R.
Some Optimal Considerations in Attitude Control Systems.
Master's thesis, Auburn University, 1973.

[26] White, Larry R., Colburn, Bruce K., and Boland, Joseph S.
Design Analogy Between Optimal Time-Fuel and Rate-Ledge Relay Controllers.
Journal of Spacecraft and Rockets , April, 1976.

[27] wie, Bong, and Plescia, Carl T.
Attitude Stabilization of Flexible Spacecraft During Stationkeeping Maneuvers.
In *Guidance and Control Conference.* AIAA, August, 1983.

[28] Wie, Bong, and Barba, Peter M.
Quaternion Feedback for Spacecraft Large Angle Maneuvers.
In *Dynamics Specialists Conference.* AIAA, May, 1984.

[29] Zimbelman, Harold F.
Analysis of the Performance of the Venus Radar Mapper (VRM) Spacecraft During the Venus Orbit Insertion (VOI) Phase of the Mission.
December, 1985.
Martin Marietta Internal Memorandum.

[30] Zimbelman, Harold F.
Analysis of the Performance of the VRM Spacecraft During Venus Orbit Insertion (VOI).
July, 1985.
Martin Marietta Internal Memorandum.