



# **ATTITUDE AND ORBIT CONTROL USING THE SPACECRAFT CONTROL TOOLBOX v4.6**

**by the Staff of Princeton Satellite Systems, Inc.**

---

The software described in this document is furnished under a license agreement. The software may be used, copied or translated into other languages only under the terms of the license agreement.

Attitude and Orbit Control Using the Spacecraft Control Toolbox v4.6 (December 2000)

© Copyright 1996-2000 by Princeton Satellite Systems, Inc. All rights reserved.

Any provision of Princeton Satellite System Software to the U.S. Government is with "Restricted Rights" as follows: Use, duplication, or disclosure by the Government is subject to restrictions set forth in subparagraphs (a) through (d) of the Commercial Computer Restricted Rights clause at FAR 52.227-19 when applicable, or in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, and in similar clause in the NASA FAR Supplement. Any provision of Princeton Satellite Systems documentation to the U.S. Government is with Limited Rights. The contractor/manufacturer is Princeton Satellite Systems, Inc., 33 Witherspoon Street, Princeton, New Jersey 08542.

MATLAB is a registered trademark of The MathWorks, Inc.

Macintosh is a registered trademark of Apple Computer, Inc.

Silicon Graphics is a registered trademark of Silicon Graphics

Adobe, Acrobat and FrameMaker are trademarks of Adobe Systems Incorporated.

All other trademarks are the property of their respective holders.

All other brand or product names are trademarks or registered trademarks of their respective companies or organizations.

### **Printing History**

December 2000

First Printing

### **Princeton Satellite Systems, Inc.**

33 Witherspoon Street

Princeton, New Jersey 08542

Technical Support/Sales/Info: <http://www.psatellite.com>

*For Marilyn and Eric*



---

# *Table of Contents*

## *Introduction 21*

### Spacecraft Control 21

*Introduction 21*

*Mnemonics 22*

*Control System Design 22*

*Dynamics and Modeling of Systems 22*

*Software Design 23*

*User Interface Design 24*

*Spacecraft Operations 24*

*The Spacecraft Control Engineer's Job 24*

### This Book 24

### ACS Terms 24

*Pointing Accuracy 25*

*Pointing Knowledge 25*

*Pointing Stability 25*

*Jitter 25*

### What A Spacecraft Control Engineer Needs To Know 26

## *Design Process 29*

### Introduction 29

### Design Issues 29

*Introduction 29*

*Requirements Evaluation and Interpretation 29*

*Databases 30*

*Configuration Management of Software and Documentation 31*

*Test Plans 31*

### Design Process 31

*Steps in the Process 31*

*General Comments 33*

## *Preliminary Designs 35*

### Introduction 35

### Requirements Analysis 36

*Direct Requirements 36*

---

<i>Indirect (or Derived) Requirements</i>	37
<i>Control System Requirements</i>	37
<b>Satellite Design</b>	<b>38</b>
<i>Selecting a Satellite Configuration</i>	38
<i>Selecting Actuators</i>	39
<i>Selecting Sensors</i>	41
<i>Cost</i>	42

## ***Mathematics*** 45

<b>Introduction</b>	<b>45</b>
<b>Vectors and Matrices</b>	<b>45</b>
<i>Notation</i>	45
<i>Vector and Matrix Representations of Operations</i>	47
<i>Matrix Operations</i>	47
<i>Special Matrices</i>	48
<i>The Matrix Inverse</i>	48
<i>Useful Matrix-Vector Identities</i>	49
<i>Floating Point Operations Required for Matrix Arithmetic</i>	50
<b>Spherical Geometry</b>	<b>50</b>
<b>Differential Equations</b>	<b>51</b>
<i>Introduction</i>	51
<i>Numerical Integration</i>	52
<i>Discontinuities</i>	53
<i>Stiff Equations</i>	54
<b>Probability</b>	<b>55</b>

## ***Time*** 57

<b>Introduction</b>	<b>57</b>
<b>Time Scales</b>	<b>57</b>

## ***Coordinate Transformations*** 59

<b>Coordinate Frames</b>	<b>59</b>
<b>Transformation Matrices</b>	<b>62</b>
<b>Quaternions</b>	<b>64</b>
<i>Introduction</i>	64
<i>Fundamental Properties of the Quaternion</i>	65
<i>Quaternion Nomenclature</i>	66
<i>Quaternion Operations</i>	67
<i>Quaternion Transformations</i>	67

---

*Quaternion Derivative* 68  
*Linearization of the Quaternion* 69  
Small Angles 73  
Physical Interpretation of the Quaternion 73

## *Control Design* 75

Introduction 75  
The General Control System 75  
Fundamental Relationships 76  
Tracking Errors 79  
State Space Closed Loop Equations 81  
Approaches to Robust Control 82  
    *Introduction* 82  
    *Modeling Uncertainty* 83  
    *Control Structure Design* 86  
    *Nyquist Like Techniques* 86  
    *Quantitative Feedback Theory* 87  
    *LQG Methods* 88  
     *$H_\infty$  and  $m$  synthesis* 89  
Single-Input Single-Output Control Design 91  
    *Introduction* 91  
    *Elementary Loop Compensation* 91  
Digital Control 98  
    *Introduction* 98  
    *Modified Continuous Design* 99  
Flexible Spacecraft Control 114  
    *Introduction* 114  
    *Two Coupled Inertias* 114  
    *Lead Compensation of the Minimum Phase System* 118  
    *Non-Collocated Sensor and Actuator* 120

## *Command Distribution* 125

Introduction 125  
The Optimal Torque Distribution Problem 125  
Examples 127  
    *Reaction Wheels* 127  
    *Gimbaled Thrusters* 128  
Simplex 129

---

## *Attitude Dynamics* 131

- Introduction 131
- Rigid Body 131
- Multibody 133
- Pivoted Momentum Wheel 135
- Modeling Flexible Structures 139

## *Orbits* 143

- Representations of Orbits 143
- Propagating Orbits 144

## *Budgets* 147

- Introduction 147
- Pointing Budgets 147
  - Methodology* 147
  - Beam Pointing Accuracy* 148
  - Example Pointing Budget* 148
- Propellant Budgets 149
  - Introduction* 149
  - Example Propellant Budget* 150

## *Sensors* 153

- Introduction 153
- Types of Sensors 153
- Optical Sensors 154
  - Star Trackers and Cameras* 154
  - Horizon Sensors* 155
  - Earth Sensors* 156
  - Analog Sun Sensors* 157
  - Digital Sun Sensors* 158
- Gyros 158
- Other Sensors 159
  - Magnetometers* 159
  - Accelerometers* 159

## *Actuators* 161

- Introduction 161

---

Types of Actuators	<b>161</b>
Stepping Motor Model	<b>162</b>
<i>Introduction</i>	<i>162</i>
<i>Electromechanical Dynamics</i>	<i>162</i>
Reaction Wheel Model	<b>164</b>
<i>Introduction</i>	<i>164</i>
<i>Motor Model</i>	<i>164</i>
<i>Reaction Wheel State Equations</i>	<i>165</i>
<i>Tachometer</i>	<i>166</i>
Control Moment Gyro	<b>166</b>
<i>Introduction</i>	<i>166</i>
<i>Modeling</i>	<i>167</i>
<i>Torque Distribution</i>	<i>167</i>
Thrusters	<b>168</b>
<i>Introduction</i>	<i>168</i>
<i>Physics of Thrusters</i>	<i>168</i>
<i>Pulsewidth Modulation</i>	<i>168</i>
<i>Minimum Impulse Bit</i>	<i>169</i>
<i>Time Constants</i>	<i>169</i>
<i>Fuel System</i>	<i>170</i>
Magnetic Torquers	<b>170</b>
Solenoids	<b>171</b>
<i>Introduction</i>	<i>171</i>
<i>Derivation of the equations of motion for a dual coil solenoid</i>	<i>172</i>
<i>Derivation of the equations of motion for a single coil solenoid</i>	<i>176</i>

## *Simulation 181*

Introduction	<b>181</b>
Linear	<b>182</b>
<i>Creating a State Space System</i>	<i>182</i>
<i>Zero Order Hold</i>	<i>183</i>
<i>First Order Hold</i>	<i>183</i>
Nonlinear	<b>184</b>

## *Disturbances 185*

Introduction	<b>185</b>
External Disturbances	<b>185</b>
<i>Introduction</i>	<i>185</i>
<i>Gravity Gradient</i>	<i>186</i>
<i>Residual Dipole</i>	<i>187</i>
<i>RF</i>	<i>187</i>
<i>Solar Pressure</i>	<i>187</i>
<i>Thruster Plumes</i>	<i>188</i>

---

Internal Disturbances **189**  
*Introduction* **189**  
Fourier Series Representation **189**

## *Attitude Estimation* **191**

Introduction **191**  
Introduction to Estimation Theory **191**  
Gyro Model **193**  
Conversion of Continuous Time Data to Discrete **195**  
Simulation of Stochastic Processes **195**  
The Kalman Filter Algorithm **196**  
Batch Methods **197**  
Vector Measurements **199**

## *Geosynchronous Satellite Control* **201**

Introduction **201**  
Requirements **201**  
The Design Process **202**  
Transfer Orbit **202**  
Mission Orbit Design **203**  
The Geometry **205**  
Acquisition **205**  
Summary **207**

## *Geosynchronous Spacecraft Control Design* **209**

A Mission Architecture **209**  
Design Steps **211**  
Spacecraft Overview **211**  
Spinning Transfer Orbit **213**  
*Dynamics* **213**  
*Actuators and Sensors* **214**  
*Changing the Spin Rate* **216**  
*Attitude Determination* **217**  
*Delta V Engine Firing* **218**  
*The Transfer Orbit Generator* **218**  
Acquisition Using The Dual Spin Turn **219**  
*Dynamics* **219**  
*Actuators and Sensors* **219**

---

<i>Initialization</i>	219
<i>Simulation</i>	219
<i>Pitch Acquisition</i>	220
Disturbances	220
Dynamics	221
<i>Introduction</i>	221
<i>Normal Operations</i>	221
<i>Dual Spin Stability</i>	223
<i>Stationkeeping Operations</i>	223
<i>Actuators and Sensors</i>	225
<i>Control System Organization</i>	225
<i>Modes</i>	226
<i>Earth Sensor</i>	226
<i>Gyros</i>	227
<i>Noise Filtering</i>	227
<i>Momentum Wheel Pitch and Tachometer Loops</i>	227
<i>Low Bandwidth Roll/Yaw Control</i>	228
<i>Thruster Control</i>	231
<i>High Bandwidth Roll/Yaw and Pitch Control</i>	231
<i>Magnetic Torquer Control</i>	232
<i>Thruster Control</i>	232
<i>Actuator Saturation</i>	233
<i>Thruster Resolution</i>	233
Summary	234

## *Sun Nadir Pointing Control* 235

Introduction	235
Coordinate Frames	236
Sun Nadir Pointing	237
Components	239
<i>Sensors</i>	239
<i>Actuators</i>	241
Attitude Determination	242
<i>Roll</i>	242
<i>Pitch</i>	243
<i>Sun Sensor Eye Pre-Processing</i>	243
<i>Solar Array Pitch</i>	243
<i>Yaw</i>	244
Control	245
<i>Reaction Wheel Loop</i>	245
<i>Attitude Loop</i>	246
<i>Solar Array Control</i>	246
<i>Momentum Control</i>	246

---

## *ACS Testing* 249

- Introduction 249
- Industry Examples 249
  - Introduction* 249
  - Telecom 1* 250
  - MOS-1* 251
  - IRAS* 253
  - INSAT I* 254
  - Intelsat V* 257
  - Space Telescope* 258
  - MBB* 258
  - MSTI and DC-X* 261
  - Summary* 262
- A Testing Methodology 263
  - Introduction* 263
  - Requirements Flow and Testing* 263
  - Testing Life-Cycle for the ACS Flight Software* 264
  - Flight Vehicle Control System Testing* 266
  - Test Levels (Preflight)* 268
  - Test Levels (Flight)* 269

## *Fault Detection* 271

- Introduction 271
- Failures 271
- Failure Analysis 273
- Fault Detection Isolation and Reconfiguration 274
  - Introduction* 274
  - Approaches to Redundancy Management* 276
  - The Fault Detection and Isolation System* 278
  - An Example Using Detection Filters* 281

## *References* 285

- Introduction 285
- Dynamics 285
- Attitude Estimation 286
- Control 286
- Spacecraft Control 286
- Astrodynamics 286
- Propulsion 287
- Real-Time Systems 287

---

Devices **287**  
Software **287**  
Ephemeris **287**  
Heat Transfer **287**  
Testing **288**  
Coordinates **288**  
Failures **289**

*Glossary* **291**

*Index* **295**



# List of Figures

FIGURE 1.1	PSS spacecraft simulation facility	23
FIGURE 1.2	Jitter	26
FIGURE 3.1	Friction Torque Diagram	40
FIGURE 3.2	Pivoted Wheel	41
FIGURE 6.1	ECI and LVLH Frames	59
FIGURE 6.2	Parallax Angle Diagram	60
FIGURE 6.3	Dual Frame Vector Diagram	61
FIGURE 6.4	Euler Angle Diagram	62
FIGURE 6.5	Quaternion Diagram	64
FIGURE 9.1	Rigid body	131
FIGURE 9.2	Multibody spacecraft model	133
FIGURE 9.3	A Topological Tree	134
FIGURE 9.4	Pivoted momentum wheel [X,Y,Z] are LVLH axes. The orbit rate is aligned with the -Y axis.	136
FIGURE 9.5	Flexible structure	140
FIGURE 10.1	Orbital Elements	143
FIGURE 12.1	CCD Element.	154
FIGURE 12.2	Horizon Sensor Measurement Geometry	155
FIGURE 12.3	Horizon pulse	156
FIGURE 12.4	Earth sensor geometry diagram	157
FIGURE 12.5	Sensor geometry	158
FIGURE 12.6	Proof-mass accelerometer	159
FIGURE 13.1	Reaction Wheel	164
FIGURE 13.2	DC Motor Model	164
FIGURE 13.3	Tank Geometry	170
FIGURE 13.4	Dual Coil Solenoid	172
FIGURE 13.5	Contours for integrating the magnetic field	173
FIGURE 13.6	Single Coil Solenoid	176
FIGURE 13.7	Contours for integrating the magnetic field	177
FIGURE 13.8	Valve driver	179
FIGURE 14.1	Coulomb Friction	183
FIGURE 15.1	Plume Geometry Diagram	188
FIGURE 16.1	Small Angles	199
FIGURE 17.1	Orbital Geometry	205
FIGURE 18.1	Transfer Orbit Configuration Diagram	210
FIGURE 18.2	Deployment Configuration Diagram	210
FIGURE 18.3	Mission Orbit Configuration Diagram	210

FIGURE 18.4	Close-up of spacecraft	212
FIGURE 18.5	Thruster Layout Diagram	213
FIGURE 18.6	Spin-Axis Attitude Determination Diagram	215
FIGURE 18.7	Great Circle Precession Diagram	216
FIGURE 18.8	Rhumb Line Precession Diagram	217
FIGURE 18.9	Attitude Determination Geometry Diagram	218
FIGURE 18.10	Dual Spin Turn	220
FIGURE 18.11	Stationkeeping Geometry	224
FIGURE 19.1	Orbit Geometry Diagram	236
FIGURE 19.2	Solar Array Frame	237
FIGURE 19.3	Sun-nadir yaw trajectory for a GPS orbit	238
FIGURE 19.4	Sensor geometry	240
FIGURE 19.5	Earth sensor geometry diagram	241
FIGURE 21.1	The FDI architecture	281
FIGURE 21.2	The Detection Filter	281
FIGURE 21.1	Spinning spacecraft loop failure and detection	284

---

# *List of Tables*

TABLE 1.1	Attitude control system mnemonics	22
TABLE 2.1	Software development status	31
TABLE 2.2	ACS Design Process	31
TABLE 3.1	Recurring Costs	42
TABLE 3.2	Nonrecurring costs	43
TABLE 1.	Matrix Arithmetic Floating Point Counts	50
TABLE 7.1	Relationships Between PK and T and S	78
TABLE 7.2	What Should be Small to Compensate for Large Inputs	79
TABLE 7.3	Inputs	80
TABLE 2.	Control System Performance Summary	95
TABLE 3.	Control System Specification	95
TABLE 4.	Z-Plane Representation of a filtered Differentiator	112
TABLE 5.	Transfer function summary	118
TABLE 11.1	Normal mode pointing budget	148
TABLE 11.2	Example Propellant Budget	150
TABLE 12.1	Classes of sensors	153
TABLE 13.1	Classes of actuators	161
TABLE 15.1	Disturbances	185
TABLE 15.2	Disturbances	189
TABLE 17.1	Possible Configurations	204
TABLE 17.2	Control System Summary	207
TABLE 20-1	IRAS Test Program	253
TABLE 20-2	INSAT I AOCS Testing	257
TABLE 20-3	Trade-off Between Open and Closed Loop Testing	259
TABLE 20-4	Problems Discovered During Closed-Loop Testing	260
TABLE 20.8	Flight Vehicle Software Testing	266
TABLE 20.9	Simulation Types	267
TABLE 20.10	Test Levels	268
TABLE 21.1	Spacecraft Failures	272
TABLE 21.2	Spinner simplified failure model	282
TABLE 23.1	Acronyms	291



# *List of Code*

Coulomb Friction Code 183



---

## *1.1 Spacecraft Control*

### **1.1.1 Introduction**

Spacecraft control is usually synonymous with “Attitude Control,” the engineering discipline of keeping a satellite or spacecraft pointed in the right direction.

As an engineering discipline, spacecraft control embodies five distinct areas:

1. Control system design
2. Dynamics and modeling of systems
3. Software design
4. User interface design
5. Spacecraft operations

The last is rarely considered, but is of critical importance in satellite control system design. The five areas together make up the System in Attitude Control System (ACS). Usually the system aspects of an spacecraft control system are more important than the control laws themselves and often much more difficult to realize.

This book discusses all aspects of spacecraft control. It is meant to be used in conjunction with demos from the Spacecraft Control Toolbox and consequently does not give many simulation results in the text.

### 1.1.2 Mnemonics

The following table gives some mnemonics used to denote spacecraft control systems.

**TABLE 1.1 Attitude control system mnemonics**

Mnemonic	Meaning	Where used
ACS	Attitude Control System	General
AOCS	Attitude and Orbit Control System	Inmarsat 3
ADS	Attitude Determination System	Used on GPS IIR
ADACS	Attitude Determination and Control System	
ADCS	Attitude Determination and Control System	

#### 1.1.1 Control System Design

Control system design can be further decomposed into:

1. Attitude determination
2. Attitude control
3. Control distribution

The first involves sensor measurements, often using a combination of different sensors, to get an estimate of the spacecraft pointing. The second involves the design and implementation of the control loops. The last involves taking control demand and converting them into demands on the actuators. For example, a thruster may be pulsewidth modulated so torque or angular acceleration demand must be converted into pulsewidths for the thrusters. In many cases, control distribution and control are combined into one step. For many simple systems, all three can be realized in a single controller. Early control systems were analog and the sensors were designed, in some cases, to produce outputs that could be used directly. For example, a scanning earth sensor produces roll and pitch measurements and often autonomously makes adjustments for sun and moon interference so that the sensor always gives valid measurements without the need for complex decision making logic.

#### 1.1.2 Dynamics and Modeling of Systems

The second category can be decomposed into:

1. Modeling
2. Simulation

This section involves the implementation of the models into numerical simulations. Since it is rarely practical to test control systems on real satellites, the designer must rely on simulation to validate his or her designs. Simulations can

range from all software running on the same platform, to software models with the control system running on a flight or equivalent board or one in which actuator and sensor hardware are integrated into the simulation. An example of the second type of simulation is shown in the following figure.

**FIGURE 1.1** PSS spacecraft simulation facility

---



The multi spacecraft simulation, written in C++, runs on the Apple G4 machine. Three spacecraft are simulated and each has a separate Force 6750 board in the chassis on the left. The Dell machine is used for software development and hosts all of the developer tools. The simulation communicates with the boards using TCP/IP.

### 1.1.3 Software Design

The third area is the implementation of both control systems and simulations in software. This is of critical importance. Ultimately, spacecraft control engineers must also be software engineers, whether they write in C++ or use a block diagram language. Most of a spacecraft control system has little to do with control theory, but rather is related to how the satellite will operate. Aside from controlling the satellite, the software must:

1. Implement the user interface (command and telemetry)
2. Switch operational modes
3. Provide fault detection
4. Provide redundancy management
5. Plan maneuvers, operations, etc.

### 1.1.4 User Interface Design

The fourth category is the most important. Most satellites that are lost are the result of operator error; operator error is often due to user interface problems. A user interface problem can be a wrong command, or one whose effect is not completely understood.

In the early days of satellite design, the user interface was never “designed”; rather, it was implemented in an ad hoc fashion. Today some satellite manufacturers (by no means the majority) have recognized that this is an important component and must be considered early in the design phase.

### 1.1.5 Spacecraft Operations

The last area is flying the spacecraft, which many designers do once their designs are launched. This involves planning maneuvers, monitoring the spacecraft during maneuvers and investigating anomalies. Anomaly investigation ranges from post-mortems on lost spacecraft to fine-tuning the control system performance.

### 1.1.6 The Spacecraft Control Engineer’s Job

How much time does a satellite control system designer spend on these tasks? Roughly 5% of a spacecraft control engineer’s time is in actual control loop design. About 10% of his or her time will be spent modeling the hardware. The other 85% will be spent writing software. This latter number includes writing analysis tools and simulations.

---

## 1.2 *This Book*

This book gives the reader everything he or she needs to know to design control systems. It is intended for use with the Spacecraft Control Toolbox, but is written to be independent of the software. The reader will find that this book does not approach control system design for spacecraft in the traditional way. You will not see discussions of spinning spacecraft, minimum energy states, etc. Rather, it is a practical guide for engineers who are designing satellite control systems that will run on digital computers on the spacecraft.

The rest of this chapter will discuss some attitude control terminology that is used frequently but rarely explained. The remaining chapters discuss various aspects of spacecraft control design including several complete, or at least comprehensive, point designs.

---

## 1.3 *ACS Terms*

Some attitude control terminology is not defined consistently throughout the aerospace industry. Textbooks and companies use different terms for the same concept. This section explains a few confusing terms, and gives the definitions that are used by Princeton Satellite Systems.

### 1.3.1 Pointing Accuracy

Also referred to as the “attitude control accuracy”, this generally refers to how well the attitude of the vehicle in question can be controlled with respect to a commanded direction. When given as a requirement, it is an absolute bound on the error allowed in the spacecraft’s orientation with respect to the commanded orientation.

### 1.3.2 Pointing Knowledge

Also referred to as the “attitude determination accuracy”, this refers to how well the orientation of the spacecraft is known with respect to an absolute reference. This term can be used for either real-time or after the fact orientation knowledge. When given as a requirement, it is an absolute bound on the error allowed in the knowledge of the spacecraft’s orientation with respect to an absolute reference.

Sensor accuracy is the sum of the ultimate accuracy of the measurement, determined by the object used by the sensors, and all the errors in the measurement such as mounting tolerances and thermal effects. Pointing knowledge is the overall accuracy of attitude determination sensor suite. Stars are the most accurate source for a measurement followed by the sun and earth.

Pointing knowledge can sometimes be improved after the fact, such as by monitoring the earth’s magnetic field and later post-processing the output of a magnetometer for better knowledge of a telescope’s pointing direction for a particular photograph.

The pointing knowledge of a spacecraft is usually better than its pointing accuracy, especially if it is not required to be available in real-time. For example, control systems which use thrusters as the sole actuators generally have a pointing accuracy limited to half the thrusters’ pulsewidth plus the attitude determination accuracy. Systems which incorporate wheels into the attitude control system can usually operate close to the attitude determination accuracy so that the pointing accuracy and the pointing knowledge are very close to each other.

### 1.3.3 Pointing Stability

When given as a requirement, this is the maximum rate of change of angular orientation allowed.

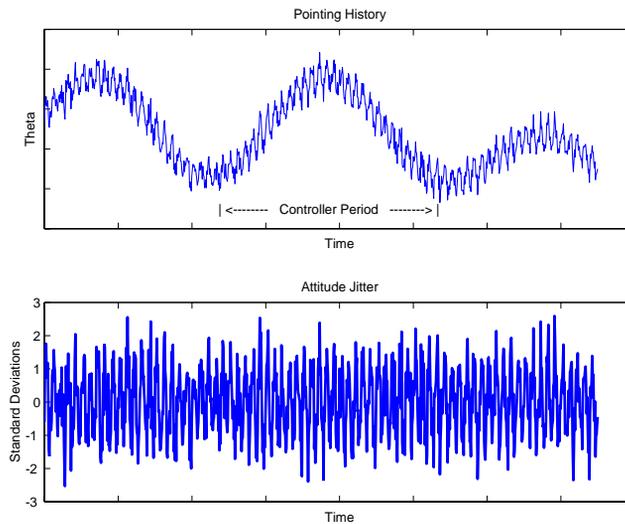
### 1.3.4 Jitter

Jitter refers to the errors in attitude of a frequency too high to be controlled by the attitude control system. When given as an attitude control performance requirement, it is a specified angle bound or angular rate limit on short-term, high-frequency motion of the spacecraft.

Each spacecraft has an inherent controller period required to sense an attitude error and implement a correction. This period determines the controller’s bandwidth by the relationship  $1/t$ . Disturbances of a frequency above the bandwidth are not attenuated by the attitude control system. Sources of such high-frequency disturbances could include the internal vibration of a sensor.

The figure below shows a spacecraft pointing history with jitter. The jitter can be seen separately by filtering the data and subtracting the smoothed curve from the original.

**FIGURE 1.2 Jitter**



---

## 1.4 What A Spacecraft Control Engineer Needs To Know

The following is an outline of topics with which an attitude control designer needs to be familiar.

1. *Dynamics*
  - Rigid body
  - Basic orbit dynamics
  - Multibody dynamics
  - Electromechanical systems
2. *Disturbances*
  - Solar pressure
  - Gravity gradient
  - Radio Frequency (RF)
  - Thermal (radiators)
  - Aerodynamic (drag and lift)
  - Outgassing
3. *Kinematics*
  - Coordinate frames
  - Transformation matrices
  - Quaternions

Euler angles  
Small angles from a nominal earth pointing frame

4. *Control*

Mostly Single-Input-Single-Output (SISO)  
Some Multi-Input-Multi-Output (MIMO)

5. *Attitude Sensors*

Earth sensors  
Sun sensors  
Star sensors  
Gyros  
Magnetometers

6. *Actuators*

Reaction wheels  
Magnetic torquers  
ACS thrusters  
Pivoted wheels  
Control Moment Gyros  
Stepping motors  
DC motors  
Solenoids  
Lead screws  
Dampers (ball in tube, etc.)

7. *Attitude estimation*

Spherical geometry  
Fixed gain filters  
Variable gain filters  
Kalman Filters

8. *Math*

Differential equations  
Linearization of nonlinear models  
Numerical methods  
Probability and statistics

This book covers most of these topics. In many cases, the topics are explored as part of extended examples.



---

## *2.1 Introduction*

This chapter describes the design process and the steps that need to be taken when designing an attitude control system. The process is independent of the type of design.

---

## *2.2 Design Issues*

### **2.2.1 Introduction**

Major issues for an attitude control system design process are:

- Requirements Evaluation and Interpretation
- Databases
- Configuration Management of Software and Documentation
- Test Plans

Each is discussed in the following sections.

### **2.2.2 Requirements Evaluation and Interpretation**

This process involves taking written customer requirements and converting them into specifications on the control system. There are three types of requirements:

1. Direct
2. Derived
3. Implied

The first type includes those requirements the customer will give you, such as an allowable pointing error and jitter. The second type are those derived from direct requirements. For example, a payload may have a requirement that it not see the sun. If you were designing a maneuvering spacecraft, you would have to make certain that no maneuver ever caused the payload to be illuminated by the sun. Another derived requirement might be a time limit on the duration of maneuvers that cause the solar panels to lose illumination. The disturbances on the spacecraft are usually a derived requirement because they are derived from the configuration which is based on other system requirements, but rarely directly on ACS requirements. (An exception is having a balanced configuration to minimize disturbances.)

Responses to failures are a combination of direct and derived requirements. A customer might state:

- The spacecraft will survive a single failure
- The spacecraft will operate through a single failure
- The spacecraft performance will not be impacted by a single failure

These are the direct requirements. The derived requirements are based on a determination of which failures are credible and how those failures might affect performance.

Implied requirements can be thought of as “good design” practices. For example:

- The control system should be easy to operate
- The purpose of each command should be obvious
- Only telemetry points that are absolutely necessary are included
- Transient responses should be reasonably well damped
- No command should ever lead to the loss of the spacecraft (in the absence of a failure that happens AFTER the command)
- The control system should be easy to implement in software or hardware
- Maneuver rates should be reasonable from an operational point-of-view
- The design should have reasonable margins for changes late in the program

Requirement evaluation is something that will need to be done throughout the design of a spacecraft. Often the customer cannot exactly specify his or her requirements at the beginning of the program. In some cases the requirements may not even be known when the design process begins. This is why the last implied requirement is important.

### 2.2.3 Databases

ACS designers should work from a common database of parameters. Design and simulation software should never have hardwired parameters.

## 2.2.4 Configuration Management of Software and Documentation

All ACS design software, simulations and databases should be configuration managed. A version control system should be used. Since nearly all ACS work is embodied in software, it is recommended that commercial designations be used as shown in the following table. The first number is always the version number.

**TABLE 2.1 Software development status**

Designation	Letter	Example	Explanation
delta		1.0d1, 4.4d77	The software has many bugs and is missing significant features.
alpha		1.0a1, 2.1a3	The software has all major features but is buggy.
beta		1.0b1	All features are included and the software is ready for user testing.
release		1.0, 1.2	The software is ready for non-developers to use.

This system applies to ACS design functions and scripts, flight software and simulations. Automated configuration management systems can simplify the logistics of managing the software. Using commercial products (such as the Spacecraft Control Toolbox) can also simplify configuration management.

Part of a configuration management system is having a bug reporting system. This is particularly important once the software is released to a customer, usually in beta.

### 2.2.1 Test Plans

All software, including the ACS design tools, must be formally tested. Testing is discussed elsewhere in this document. When instituting a test program, it is important to have a written test plan. All tests, once performed are collected for automated regression testing which is done periodically or whenever significant changes have been made in the software base.

---

## 2.3 Design Process

### 2.3.1 Steps in the Process

The following table gives steps in the design process. Steps that may be done in parallel are in the same box.

**TABLE 2.2 ACS Design Process**

Step	Tasks	Explanation
1	Evaluate requirements Create a disturbance model	
2	Select actuators and sensors	

**TABLE 2.2 ACS Design Process**

Step	Tasks	Explanation
3	Size actuators	Use maneuver requirements and disturbances.
4	Create ACS algorithms Build the ACS simulation in Matlab	Algorithms are in Matlab. Use the simplest algorithms that meet requirements! The simulation will be used in testing throughout the program.
5	Test ACS in simulation to validate functionality	This is done in Matlab.
6	Test ACS in simulation to validate performance	Performance must meet or exceed requirements. This includes implied requirements too!
7	Test ACS to verify failure response	This should be done early.
8	Create flight software Build ACS simulation in C/C++	Should be a copy of the Matlab simulation.
9	Test ACS in simulation to validate functionality	All software simulations.
10	Test ACS in simulation to validate performance	
11	Test ACS to verify failure response	
12	Port ACS to flight hardware or equivalent	Need not be the flight boards and actuators and sensor may be modeled in software
13	Test ACS in simulation to validate functionality	Flight code and processor.
14	Test ACS in simulation to validate performance	
15	Test ACS to verify failure response	
16	Connect actuators and sensors	
17	Test ACS interfaces	
18	Load flight software on spacecraft	
19	Test ACS interfaces	
20	Flight	
21	Verify ACS performance	
22	Validate all ACS models	This is done to validate the entire design process.

If requirements change, the designer may have to return to steps 1 through 4 many times. Performance tests are only done using software actuator and sensor models. The purpose of the hardware testing is to verify interfaces. The last step in the process has two purposes. First, it validates the models for later use during anomaly investigations. Second, it paves the way for future programs whether they be variations on the current spacecraft or completely new designs.

### 2.3.1 General Comments

Failure accommodation should be evaluated early in a program. ACS designers like to focus on estimators and controllers, but failure accommodation and detection is usually the most difficult task in an ACS design and sufficient resources must be allocated early. Simulations should include the modeling of spacecraft hardware failures from the very beginning.

Usability testing should be done as early as possible. Many attitude control systems have modes, such as stationkeeping and sun pointing. Transitions between modes are usually a source of frustration, not unlike trying to use an early model VCR. Problems can only be found if the ACS simulations include all moding code from the very beginning. A bad habit is to build “simplified” simulations to test algorithms and then worry about moding once the C/C++ coding begins.

Another factor in usability testing involves commands. It should be easy for the user to understand the purpose of every command that can be sent to the ACS. The effects of a command should be localized to one module. A command that needs to change parameters in several modules should be changed to a mode command and the mode change should cause the module parameter changes.

Finally, one last comment on configuration management. It will often be necessary to repeat steps in the design process. If software and documentation is not properly managed, the software and documentation base can be a source of errors. As a consequence all ACS memos, reports and the software used to generate the results should be configuration managed. This should include the design and analysis packages, such as Matlab. It should be possible for another engineer to take an early memo and regenerate the results for that memo.



---

### 3.1 Introduction

This chapter describes how to develop a preliminary control system design. It describes how to interpret customer requirements, how to select actuators and sensors, and how to organize your design.

Designing an attitude control system from scratch presents the control systems engineer with a bewildering set of choices. The engineer must select:

- Spacecraft type (dual-spin, momentum-bias, zero-momentum, sun-nadir)
- Control strategy (active or passive)
- Actuators (wheels, magnetic torquers, thrusters)
- Sensors (earth, star, sun, planet, magnetic field, gyros)
- Processors (1750A, 80386, 1802, 8086, RAD750)
- Delta-V engine (AKM, LAE)
- Stationkeeping thrusters (REA, EHTs, Arc jets, Ion)
- Interfaces (A/D and D/A converters, databuses)

Within each of these categories there are many choices. For example, star sensors can be trackers or mappers and may have built in star identification or may pass a pixel map to the spacecraft computer. Sun sensors can be digital or analog, one or two axis and may be high or low accuracy. All of these items must be chosen in conjunction with the other subsystem engineers.

This section gives a qualitative discussion of attitude control system design. Its purpose is to outline an approach to designing a spacecraft control system and provide the reader with an introduction to some of the options.

---

## 3.2 *Requirements Analysis*

### 3.2.1 **Direct Requirements**

Each mission will have a set of requirements. Requirements are either direct or derived. Typical direct mission requirements are for:

- Pointing accuracy
- Pointing knowledge
- Jitter
- Lifetime
- Reliability

Pointing accuracy is how tightly the spacecraft actually points at its target and knowledge is how well you know the attitude error. For example, a passively stabilized spacecraft with a star sensor might have very high attitude knowledge but relatively poor pointing accuracy. Jitter is measured in a variety of ways but ultimately it is a measure of how much the attitude changes in a given period of time. For example, if you use noisy sensors and low resolution D/A converters, you are likely to have a lot of jitter.

The pointing numbers may be given for the spacecraft as a whole or for individual payloads. If the latter is the case, you have the additional option of putting some of the payloads on articulated platforms with their own sensors. For example, on a dual spin spacecraft only the despun platform has a pointing requirement. Pointing requirements only apply to the payload. It is not unusual to collocate the payload and attitude control sensors on the same composite structure to minimize errors due to structural vibration and thermal distortion.

The lifetime requirement stipulates that the spacecraft must have a certain probability of lasting for the desired lifespan. Two things can happen to terminate a mission. The spacecraft can run out of consumables, (usually fuel but sometimes coolant), or the components can fail or wear out.

Redundancy requirements are often given as how long it takes the spacecraft to return to normal operation after a failure. The assumption is that no one failure will impact the mission, i.e. cause it not to meet its requirements, except for a permitted transient period after the failure. Different failures may have different requirements. Commercial customers usually require the quickest recovery from a failure since they lose money if the satellite is out of operation, not to mention irritating millions of customers whose Super Bowl broadcast is suddenly interrupted. Scientific customers are usually more lenient—the Hubble Space Telescope was operating out of specification for a long time before it was fixed.

The customer does not always specify the requirements carefully and may change them without notice as the design evolves. Thus, aside from meeting the baseline requirements, it is always necessary to make the design robust enough to tolerate requirements changes—the most unpredictable type of changes.

Another problem with requirements is that the customer may specify requirements in a way that needlessly constrains the design. For example, a geosynchronous satellite customer may specify roll, pitch, and yaw when she really means to specify beam pointing (circular error) or even better, power distributed over the coverage area. If you are given the third type of requirement you could meet it by upping the beam power. If you are given the second you could trade roll accuracy against pitch accuracy and ignore yaw altogether (with the caveat mentioned above).

### 3.2.2 Indirect (or Derived) Requirements

Derived requirements are levied on the ACS subsystem by other subsystems. If the power over the coverage area is the direct requirement then attitude accuracy becomes an indirect requirement. Power consumption of your components is always an indirect requirement. Some indirect requirements are:

- Power
- Thermal
- Radiation hardness
- Radio frequency emission sensitivity
- Induced noise (such as from an unbalanced momentum wheel)
- Computational loading (i.e. how many floating point operations you can use per control cycle)

and so forth.

### 3.2.3 Control System Requirements

Traditional control system requirements, such as gain and phase margins, are rarely specified. They must be derived from the pointing and jitter requirements. For example, most satellites have a special thruster control mode. When you turn on the delta V thrusters, there will be an attitude transient. This transient must not exceed the pointing requirements for the spacecraft hence this puts a limit on the peak overshoot permissible. It is usually desirable not to have to change gains over the mission life, hence this places a requirement on the tolerance of the control system to mass properties variations. Sensor scale factors may drift due to accumulated radiation damage, leading to a requirement on the tolerance to scale factor variations.

While some, such as sensor and actuator scale factors and inertia variations, can be lumped into an aggregate gain variation it is a good idea to perform a sensitivity analysis on the closed loop control system for all identifiable parameters that are known to vary.

### 3.3 *Satellite Design*

#### 3.3.1 **Selecting a Satellite Configuration**

Satellites generally have two major phases in their life. The first phase is transfer orbit during which the satellite is placed in its operational orbit. The second phase is the mission orbit where the satellite performs its nominal mission. Low earth orbiting satellites are often placed directly into their mission orbits and do not have a transfer orbit phase.

The first step is choosing the mission orbit configuration, followed by choosing the transfer orbit configuration.

A few of the more popular satellite configurations are:

- Gravity-gradient stabilized
- Spinner
- Dual-Spin
- Bias Momentum
- Zero Momentum

Gravity gradient stabilization is passive and at most requires a libration damper. Libration dampers are dampers designed to damp spacecraft rigid body oscillations due to gravity gradient. If there is sufficient internal damping (due to heat pipes, fuel slosh), even the damper may not be required. Gravity gradient stabilization is suitable only for spacecraft with loose pointing requirements.

A spinner can be used if one axis of the spacecraft is required to remain inertially fixed and the payload is not affected by spinning, for example a satellite with a long wire antenna along the spin-axis. If the satellite spins about its major axis a passive damping system will suffice. If it has to spin about its minor axis an active nutation control system is needed to keep it out of a flat spin (i.e. a steady spin about any other axis). The satellite will also need a control system to keep the spin-axis in the same inertial direction if there are significant inertially fixed torques, for example due to a center-of-solar-pressure offset from the center-of-mass along the spin-axis.

Dual spin spacecraft have two rotating parts. Usually one is a platform that rotates at orbit rate (and consequently is nominally pointing at the earth.) The other rotates at high speed and gives the spacecraft gyroscopic stiffness.

A bias momentum spacecraft is conceptually the same as a dual-spin spacecraft, except that the part that is rotating at high speed is a momentum wheel and rotates at very high rates.

A zero momentum design has low net inertial momentum. The spacecraft may be controlled by reaction wheels, magnetic torquers or by thrusters. In the former case the wheels may each have significant momentum but ideally the vector sum of their momentum is zero. It is relatively easy to rotate a spacecraft with zero momentum because there is no gyroscopic stiffness to resist rotation. However, if the control system fails, there is also no gyroscopic stiffness to keep the spacecraft from rotating.

### 3.3.2 Selecting Actuators

Typical satellite actuators are:

- Wheels or spheres (momentum or reaction)
- Pivoted wheels (Control moment gyros, for example)
- Thrusters
- Gimballed thrusters
- Magnetic torquers
- Reflective surfaces (either to solar or aerodynamic drag)

#### Thrusters

If the satellite has a requirement to maintain its orbit it will require thrusters and a complete propulsion system. It is almost always necessary to use some thrusters for attitude control during orbit adjustment maneuvers. This may be done by firing other thrusters to provide control torques to compensate for the disturbance torques generated by the delta V thrusters, or by gimbaling the delta-V thrusters. For example, the Space Shuttle Orbiter does both. It gimbals the Orbital Maneuvering System (OMS) engines and also fires bipropellant thrusters. Gimbaling is generally not very popular because the gimbal mechanisms tend to be heavy, complex and expensive. A gimbal was used on NASA's Deep Space 1 to point the ion engine through the center-of-mass effectively providing 2-axis control.

Thrusters can be throttled, fired with a fixed length pulse, or pulsewidth modulated. Pulsewidth modulation means that the thruster is on for only part of the control period so that the average torque equals the commanded torque. This technique can be extended to have a variable firing period (known as pulse width-pulse frequency modulation.) Throttled thrusters tend to be expensive and complex and are rarely used. Pulsewidth modulation is frequently used because it lends itself to linear control systems. Fixed pulsewidth control systems are derived from optimal control theory and minimize fuel consumption. They were used on the Shuttle and the Apollo spacecraft. However, they do not lend themselves to spacecraft that cannot be modeled as rigid bodies.

#### Wheels

A momentum wheel nominally has a fixed (large) momentum. A reaction wheel nominally has zero momentum. A reaction wheel (or reaction sphere) works by providing a reaction to the torque that the motor produces. The torque is applied equally to the spacecraft and to the wheel. Wheels present an interesting control problem. As they spin up they create a momentum bias on the spacecraft, thus changing the plant. This perturbation is nonlinear because the torque on the spacecraft is the product of the spacecraft angular velocity and the wheel momentum. If the disturbances are small, and mostly cyclic, and the spacecraft has a relatively large inertia, the plant perturbation is second order. Many algorithms have been written to take advantage of this during maneuvers. The algorithm will try to pick a path in velocity space that minimizes time while staying within the momentum limits of the wheels.

Since there will always be an external bias torque on the spacecraft due to external disturbances, wheels must be unloaded. To unload a wheel is to reduce the speed back to its nominal value. This can be done with magnetic torquers, reflective surfaces, aerodynamic surfaces or thrusters. An alternative is to partition the control effort so that the

inertially fixed disturbance compensation is performed by thrusters or torquers, and only the cyclic disturbances are controlled by the wheels. This can be done with a complementary filter pair—a matched low and high pass filter pair such as

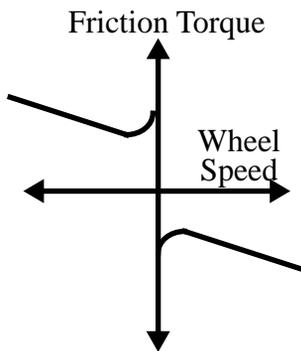
$$\frac{u_{\text{wheel}}}{u} = \frac{s}{s+a} \quad \frac{u_{\text{thruster}}}{u} = \frac{a}{s+a} \quad (3.1)$$

where  $a$  is the break frequency.

Reaction wheels must pass through zero wheel speed; this presents a problem because the friction torques have a discontinuity, shown in Figure 3.1.

**FIGURE 3.1 Friction Torque Diagram**

---



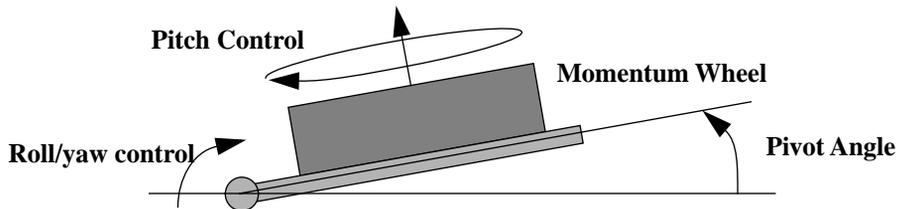
This is highly nonlinear and not amenable to linearization about the zero wheel speed! The wheel can also get stuck at zero speed. One standard approach is to use triangle wave dither in the voltage input to the wheel motor. In this technique a triangle voltage wave is added to the desired voltage command. Another is to embed the wheel in a tachometer loop where the outer loop commands wheel speed changes and the inner loop tries to force the wheel speed to track the desired speed changes.

### Pivoted Wheels and CMGs

Pivoted wheels use pivots (either one or two) to generate transverse axis torques. They also use the wheel like a reaction wheel. Control Moment Gyros (CMGs) use only the gimbal motors to provide torque and the wheel is maintained at a constant speed. CMGs come in double and single gimballed types. The former require complex slip rings to get power to the inner gimbal. Since one CMG only provides two axes of torque at least two are required. Generally more are needed and it is necessary to distribute the control activity among the gimbals in an optimal way (usually to keep the CMG platforms at small angles relative to the spacecraft). CMGs are popular for maneuvering spacecraft of intermediate size or for really large spacecraft (like Skylab.)

A pivoted wheel is shown below.

**FIGURE 3.2 Pivoted Wheel**



Pivoted wheels are used on Lockheed Martin communications satellites and on the Hughes HS601 spacecraft. The former uses a single pivot about roll while the latter employs a double pivot.

Pivoted wheels and CMGs require momentum unloading unless the complementary filter approach is used. In either case, an actuator that can apply an external torque to the spacecraft, such as a thruster or magnetic torquer, is required. Reaction wheels, pivoted wheels and CMGs can redistribute momentum within a spacecraft; they cannot change the total inertial momentum of the spacecraft or change its vector in the inertial frame.

### 3.3.3 Selecting Sensors

Typical satellite sensors are:

- Earth sensors (scanning or static)
- Star sensors (mappers or trackers)
- Sun sensors (one or two axis)
- Horizon sensors (need a spinning spacecraft)
- Gyros (many types)
- Magnetometers
- Potentiometers
- Angle encoders

Many satellites are earth pointing and need to point a payload at a target on the earth. Some are inertially pointing (such as the space telescope) and point to targets away from the earth. Star sensors and sun sensors give an inertial reference directly and can, through means of an ephemeris, give an earth reference. Earth sensors, magnetometers and horizon sensors give an earth-fixed reference, but are much less accurate than star and sun sensors.

Gyros measure inertial rates. Rate integrating gyros give the integral of the body rates. Since each gyro integrates about a body axis gyros cannot give an inertial reference directly and are always used in conjunction with star, sun or earth sensors.

Angle encoders and potentiometers are used to measure relative orientation of components on the spacecraft.

Selection of sensors is one of trading cost against sensor life, accuracy and processing software complexity. An important factor in selecting sensors is their radiation hardness. Radiation will degrade many analog circuits and most CCD or CID arrays thus limiting the life of optical sensors. Other considerations include sensor placement and interference with the payload.

### 3.3.4 Cost

Most manufacturers wish to design a satellite that meets all requirements and costs as little as possible. The total cost of a spacecraft is composed of recurring and nonrecurring costs. The former are the costs associated with building each spacecraft. The latter are costs associated with designing a spacecraft or customizing an existing design. Each type of cost is composed of the cost of personnel and the cost of purchased products, facilities and other non-personnel costs.

Recurring costs are discussed in the following table.

**TABLE 3.1 Recurring Costs**

<b>Cost</b>	<b>Description</b>
Direct Cost of Components	Costs of components generally rise with the required lifetime of the component and the devices performance. The cost of integrating a component into a design must be added to this cost. Sometimes a more expensive component is much less expensive to integrate.
Direct Cost of the Consumables	The cost of consumables (such a fuel and cryogenics) must include the cost of loading the consumable and the cost of storing the consumable on-board the spacecraft.
Cost to Lift the System into Orbit	Weight is almost always at a premium on any spacecraft. The transportation cost must always be considered. The greater the required delta-V for the mission the higher the cost. This is more complicated than it seems at first since there are thresholds of weight where you must go to the next size up launch vehicle (like from a Delta to an Ariane) which then has a different price schedule.
Cost of Insurance	New designs cost more to insure than proven designs. The cost of insurance also varies with the launch vehicle.
Cost of Maintenance	Technical support for a customer can be very expensive, especially if it requires onsite support. In addition to the direct cost of the engineering support, the cost of losing the services of the engineers must be considered.

Nonrecurring costs are described in the following table.

**TABLE 3.2** Nonrecurring costs

Cost	Description
Cost to Design the Algorithms	This includes the design costs and the cost of all tools needed to generate the design.
Cost to implement in software	Includes software requirements generation, software design, coding and test.
Cost to document the system	Includes memos, reports and user's guides.



---

## 4.1 Introduction

This chapter provides an introduction to the mathematics used in spacecraft simulation and control.

---

## 4.2 Vectors and Matrices

### 4.2.1 Notation

In this book, vectors may represent either ordered collections of values, or a quantity with a magnitude and direction in a 3-space. The state vector for a system is an example of the former; the position vector of a thruster is an example of the latter.

This book uses a combination matrix and vector notation to represent vectors and matrices. Matrix notation implies that each vector is connected with a particular coordinate frame. Thus the position vector

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{4.1}$$

cannot be assigned a value unless the reference frame for which  $x$ ,  $y$  and  $z$  are defined is given. Usually the reference frame will be obvious from the context. For example, if the transformation matrix  $B$  transforms from frame  $b$  to  $a$ , then

$$y = Bu \tag{4.2}$$

implies that  $u$  is in frame  $b$  and  $y$  in frame  $a$ . In some cases, mostly when presenting other authors work, vector notation will be used and will be clear from the context. A vector, in contrast to a “ $3 \times 1$  matrix” exists independently of any coordinate frame conceptually, but when numbers are to be used the coordinate frame must be specified.

Subscripts will be used to denote the elements of a vector. For example, the vector  $v$  is

$$v = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \tag{4.3}$$

and the matrix  $m$  might be

$$m = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \tag{4.4}$$

Sometimes, however, subscripts are used to differentiate one vector from another. In those cases, the second element of the subscript will denote the vector component.

$$v_1(x) = \begin{bmatrix} v_{1x}(x) \\ v_{1y}(x) \\ v_{1z}(x) \end{bmatrix} \tag{4.5}$$

In this case the vector is a function of  $x$ , the  $x$  does not denote element  $x$ .

All vectors, such as position or velocity vectors, are column vectors. Row vectors are obtained by taking the transpose of a column vector

$$a = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \tag{4.6}$$

The row vector is the transpose of  $a$  or

$$a^T = [a_x \ a_y \ a_z] \tag{4.7}$$

This differs from other books where the  $a$  vector may be a row or column vector depending on the context.

A matrix may be composed of other matrices. For example, a matrix composed of three vectors is

$$\begin{bmatrix} u_x & u_y & u_z \end{bmatrix} = \begin{bmatrix} u_{xx} & u_{yx} & u_{zx} \\ u_{xy} & u_{yy} & u_{zy} \\ u_{xz} & u_{yz} & u_{zz} \end{bmatrix} \quad (4.8)$$

Each column is a vector. The first index refers to the vector and the second to the component of the vector.

## 4.2.2 Vector and Matrix Representations of Operations

A few common vector operations will be described here, along with their corresponding matrix representation. In this section, vectors will be denoted with an arrow.

The dot product (or inner product) of a vector is equivalent to the transpose of a 3×1 matrix times a 3×1 matrix.

$$\vec{a} \bullet \vec{b} = a^T b \quad (4.9)$$

The cross product of two vectors is equivalent to the product of the skew-symmetric form of the column matrix times the second vector.

$$\vec{a} \times \vec{b} = a^\times b \quad (4.10)$$

where

$$a^\times = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \quad (4.11)$$

The last vector operation is the outer product.

$$\vec{a}\vec{b} = ab^T = \begin{bmatrix} a_x b_x & a_x b_y & a_x b_z \\ a_y b_x & a_y b_y & a_y b_z \\ a_z b_x & a_z b_y & a_z b_z \end{bmatrix} \quad (4.12)$$

Throughout this book the identity matrix will be denoted by the symbol  $E$ , instead of  $I$  so that it is not confused with the inertia matrix. The arrow notation denotes the quantity as a vector that is not tied to a particular coordinate frame.

## 4.2.3 Matrix Operations

Matrix addition and subtraction are defined for matrices of the same size

$$A = B + C \quad (4.13)$$

Matrix multiplication is defined for matrices as follows

$$A(n, p) = B(n, m)C(m, p) \quad (4.14)$$

Matrix multiplication is commutative

$$A(BC) = (AB)C \quad (4.15)$$

and associative

$$A(B + C) = AB + BC \quad (4.16)$$

Generally, however

$$AB \neq BA \quad (4.17)$$

The transpose of a matrix is

$$A = B^T, a_{ij} = b_{ji} \quad (4.18)$$

and

$$(AB)^T = B^T A^T \quad (4.19)$$

#### **4.2.4 Special Matrices**

A matrix is symmetric if

$$A = A^T \quad (4.20)$$

A matrix is skew-symmetric if

$$A^T = -A \quad (4.21)$$

The matrix form of the vector cross product is a skew-symmetric matrix, but in general the diagonal does not have to be zero. .

#### **4.2.5 The Matrix Inverse**

The matrix inverse frequently has to be computed. The inverse is defined as

$$AA^{-1} = E \quad (4.22)$$

where  $E$  is the identify matrix, or

$$A^{-1}A = E \quad (4.23)$$

The inverse of a product is defined

$$(AB)^{-1} = B^{-1}A^{-1} \quad (4.24)$$

and the transpose of an inverse is the inverse of the transpose

$$(A^T)^{-1} = (A^{-1})^T \quad (4.25)$$

The inverse of a 2×2 matrix is

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-1} = \frac{\begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}}{a_{11}a_{22} - a_{12}a_{21}} \quad (4.26)$$

The inverse is not defined if the denominator on the right-hand-side is zero. The denominator is also referred to as the determinant.

#### 4.2.6 Useful Matrix-Vector Identities

The following is a list of useful matrix-vector identities given without proof. In the following equations, **C** is an orthonormal matrix. **A** and **B** are real matrices, and **a** and **b** are vectors.

$$-a^\times b = b^\times a \quad (4.27)$$

$$(Ca)^\times = Ca^\times C^T \quad (4.28)$$

$$a^\times b^\times b^\times a = b^\times a^\times a^\times b \quad (4.29)$$

$$(A + B)^T = A^T + B^T \quad (4.30)$$

$$(ab)^T = b^T a^T \quad (4.31)$$

$$(a^\times)^T = -a^\times \quad (4.32)$$

$$(A^T)^{-1} = (A^{-1})^T \quad (4.33)$$

$$(AB)^{-1} = B^{-1}A^{-1} \quad (4.34)$$

$$C^{-1} = C^T \quad (4.35)$$

$$|AB| = |A||B| \quad (4.36)$$

$$A(BD) = (AB)D \quad (4.37)$$

$$|a| = \sqrt{a^T a} \quad (4.38)$$

$$Unit(a) = \frac{a}{|a|} \quad (4.39)$$

$$\cos \theta = \frac{a^T b}{|a||b|} \quad (4.40)$$

### 4.2.7 Floating Point Operations Required for Matrix Arithmetic

The following table lists important matrix-vector elementary operations and the required number of floating point operations.

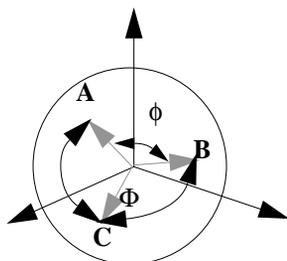
**TABLE 1. Matrix Arithmetic Floating Point Counts**

Operation	Matrix Type	Adds	Multiplies	Total
$A(m,n)B(n,p)$	Real, general	$mp(n-1)$	$mpn$	$mp(2n-1)$
$A(m,n)+B(m,n)$	Real, general	$mn$	0	$mn$
$A(m,m)B(m,m)$	Real, Symmetric	$m(m^2+1)/2$	$m^2(m+1)/2$	$m(2m^2+m+1)/2$
$A(m,m)+B(m,m)$	Real, Symmetric	$m(1+m)/2$	0	$m(1+m)/2$
Trace( $A(m,m)$ )	Real, Any	$m$	0	$m$

---

## 4.3 Spherical Geometry

Spherical geometry is an alternative way of looking at unit vectors. The following figure shows the unit sphere and three unit vectors.

**Figure 4-1 Unit sphere and unit vectors**


There are two angles that are important in spherical geometry. The first is the angle between the vectors which is found from the relation

$$A^T B = \cos(\phi) \quad (4.41)$$

which is the dot product. The second angle is the angle between the arcs, for example angle  $\Phi$  between arcs AC and AB. Spherical trigonometry is useful when using vectors to stars or planets for measurements. The two most important equations are the law of sines and law of cosines for spherical angles. The law of sines is

$$\frac{\sin(\theta)}{\sin(\Theta)} = \frac{\sin(\lambda)}{\sin(\Lambda)} = \frac{\sin(\phi)}{\sin(\Phi)} \quad (4.42)$$

The law of cosines for sides is

$$\cos(\lambda) = \cos(\theta)\cos(\phi) + \sin(\theta)\sin(\phi)\cos(\Lambda) \quad (4.43)$$

and the law of cosines for angles is

$$\cos(\Lambda) = -\cos(\Theta)\cos(\Phi) + \sin(\Theta)\sin(\Phi)\cos(\lambda) \quad (4.44)$$

The small Greek letter denotes the angle from the dot product. The capital Greek letter denotes the angle between the arcs on the surface of the sphere that is opposite the corresponding small letter. The three arcs form a spherical triangle

---

## 4.4 Differential Equations

### 4.4.1 Introduction

Spacecraft dynamics and control deals primarily with the initial value problem.

$$\begin{aligned}
 \dot{x} &= f(x, u, t) \\
 y &= g(x, t) \\
 x(t_0) &= x_0
 \end{aligned}
 \tag{4.45}$$

These equations include:

- orbital dynamics and kinematics
- attitude dynamics and kinematics
- structural dynamics and kinematics
- fluid dynamics and kinematics
- actuator dynamics and kinematics
- sensor dynamics and kinematics
- environmental forces and torques
- external actuator forces and torques

Internal forces and torques, generated by moving parts on the spacecraft, are part of the equations. The input vector will depend on what subsystems the equations include. Usually they are the commands sent to the actuators. These commands are usually data words passed to an actuator using a serial or parallel bus. D/A converters, or other devices, then produce an analog voltage which drives the actuator. In simplified models, the input vector might be the actual desired analog voltages.

#### 4.4.2 Numerical Integration

The most general way to solve the equations is through numerical integration. Runge-Kutta methods are used almost universally. Runge-Kutta methods require multiple computations of the right-hand-sides ( $f$ ) every integration time step. This permits the changing of the step size every integration time step which is often advantageous (especially in simulations that need to be synchronized with real-time). The standard fourth order Runge-Kutta formula is:

$$\begin{aligned}
 k_1 &= f(x, u(t), t) \\
 k_2 &= f\left(x + \frac{h}{2}k_1, u\left(t + \frac{h}{2}\right), t + \frac{h}{2}\right) \\
 k_3 &= f\left(x + \frac{h}{2}k_2, u\left(t + \frac{h}{2}\right), t + \frac{h}{2}\right) \\
 k_4 &= f(x + hk_3, u(t + h - \epsilon), t + h) \\
 x &= x + \frac{h}{6}(k_1 + 2(k_2 + k_3) + k_4) + O(h^4)
 \end{aligned}
 \tag{4.46}$$

The right-hand-sides are computed at four different points, twice at  $h/2$  and once at  $t$  and  $t + h$ . At  $t+h$  the input term is given at  $t + h - \epsilon$  where  $\epsilon$  is an infinitesimal number. This means that if  $u$  is changing at every integration time step (as a digital controller might) then the previously held value is used, not the new value.

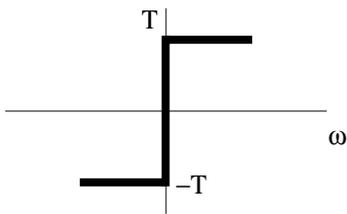
This method does not provide an error estimate. However, it is possible to run a 4th and 5th order (or in general  $n$  and  $n+1$  order) method with only one more calculation of  $f$ . The difference between the solutions can be used to adjust the step size. Since the step size is a scalar some scalar comparison must be made. For spacecraft problems, angular momentum is the best quantity to use for this purpose. In general, any quantity that is conserved is a good choice for choosing the step size.

Some solvers can also vary the order of the method. With Runge-Kutta this can be done every step.

### 4.4.3 Discontinuities

Many mechanical systems have nonlinearities that are modeled using discontinuous functions. Coulomb friction is an example. Coulomb friction is shown in the following figure.

**Figure 4-2 Coulomb Friction**



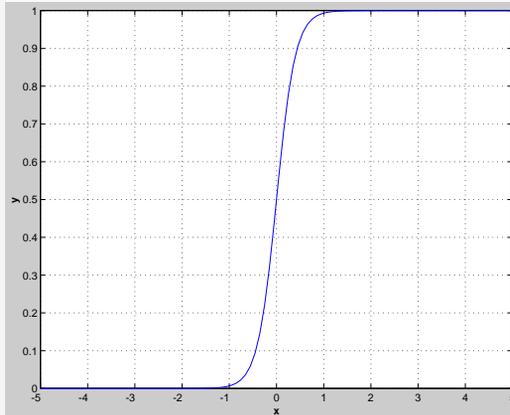
When numerically integrating a function with Coulomb friction, the solution will limit cycle. This would imply that a mechanism with Coulomb friction would oscillate forever. Rather than using this kind of function, the discontinuous function should be replaced with a smooth function. For example the function

$$y = \frac{1}{1 + e^{-5x}} \quad (4.47)$$

produces the following example.

**Example 4-1 Smooth Friction**

```
x = linspace(-5,5);
Plot2D(x,1./(1 + exp(-5*x)));
PrintFig(0,1,7,'SmoothFriction')
```



#### 4.4.4 Stiff Equations

Stiff differential equations are equations in which there is a wide range of dynamic ranges embodied in the equations. This makes it difficult to choose a time step. An example of stiff equation is

$$\frac{dy}{dt} = 5(y - t^2) \tag{4.48}$$

The exact solution is

$$y = Ae^{5t} + t^2 + 0.4t + 0.08 \tag{4.49}$$

where the first solution is the homogeneous solution. If  $y(0) = 0.08$ ,  $A = 0$ . However, the slightest numerical inaccuracy will cause the homogeneous solution to explode.

**Example 4-1 Stiff equation**

```

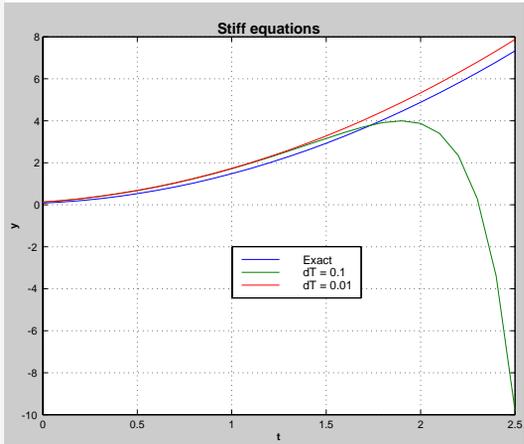
t = linspace(0,2.5,26);
y = t.^2 + 0.4*t + 0.08;
n = 10;
dT1 = t(2); dT2 = dT1/n;
y1 = 0.08; y2 = 0.08;
t1 = 0; t2 = 0;

for k = 1:length(t)
    y1 = RK4( 'FStiff', y1, dT1, t1 );
    for j = 1:n
        y2 = RK4( 'FStiff', y2, dT2, t2 );
        t2 = t2 + dT2;
    end
    y1Plot(k) = y1;
    y2Plot(k) = y2;
    t1 = t1 + dT1;
end

Plot2D(t,[y;y1Plot;y2Plot],'t','y','Stiff equations');
legend('Exact','dT = 0.1','dT = 0.01');

function yDot = FStiff( y, t )
yDot = 5*(y-t^2);

```



Decreasing the integration step size puts off the time at which the solution will explode. For this kind of problem stiff equation solvers are needed. Stiff solvers can prevent the growth of these kind of parasitic solutions.

---

4.5 Probability



---

## 5.1 Introduction

This section introduces time scales that are used in astronomical work. Time scales are important when the spacecraft must measure its orientation or position with respect to the earth, sun, planets or stars.

---

## 5.2 Time Scales

UT1 (Universal Time 1)—A time scale based on the rotation of the earth. UT1 is defined with respect to Sidereal Time.

TDT (Terrestrial Dynamical Time)—A time scale that would be kept by an ideal clock at sea level.

TDB (Barycentric Dynamical Time)—A time scale that would be kept by an ideal clock moving with the solar system barycenter. It is always within 2 milliseconds of TDT and the difference is caused by relativistic effects.

TAI (International Atomic Time)—A time scale kept by atomic clocks on the earth.

UTC (Universal Time Coordinated)—The basis for international time keeping. Its rate is the same as TAI, but its epoch is adjusted in one-second steps to keep it within 0.9s of UT1, which is measured by the earth's rotation against the stars.

Sidereal Time—Measured by the rotation of the earth against the stars.

These time scales are related by the formulas

$$UT1 = UTC + DUT1$$

$$TDT = TAI + 32.184s$$

DUT1 is published weekly (and transmitted with the UTC signals). Thus there are two ways of measuring time, by looking at the fixed stars or by measuring oscillations of atoms. The above formulas relate the two.

Julian date gives a continuous count of dates since January 1, 4713 BC at noon. Integer Julian Dates always refer to noon. The Julian date for the standard epoch J2000.0, January 1, 2000 12h is JD 2451545.0 TDB.

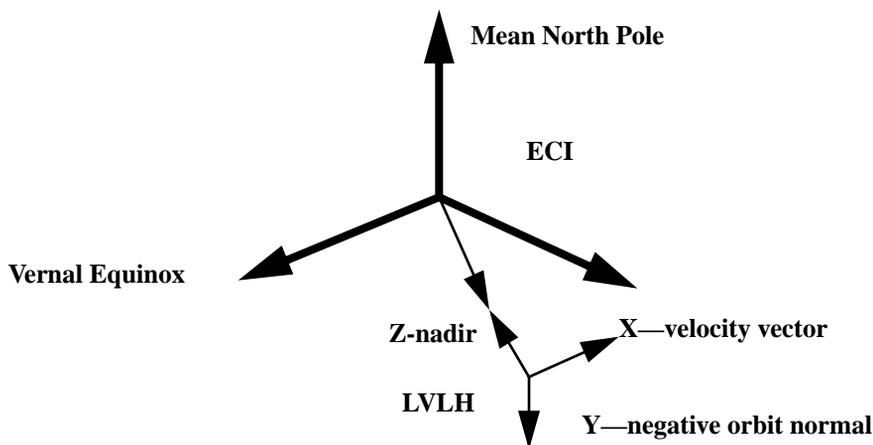
# Coordinate Transformations

## 6.1 Coordinate Frames

A right-handed reference frame is defined by a set of three orthogonal axes in a right-handed set and an origin. In this toolbox the origin will be either the center of the earth or the center-of-mass of the spacecraft.

The inertial reference frame used throughout the toolbox is the J2000.0 frame. J2000.0 defines the date January 1, 2000 at 12<sup>h</sup> Universal Time. This frame has xy-plane parallel to the mean Earth equator at epoch J2000.0 and its z-axis pointing towards the mean north celestial pole of J2000.0. The x-axis points toward the mean equinox of J2000.0. This frame is also sometimes called the Earth-Centered Inertial Frame (ECI).

FIGURE 6.1 ECI and LVLH Frames



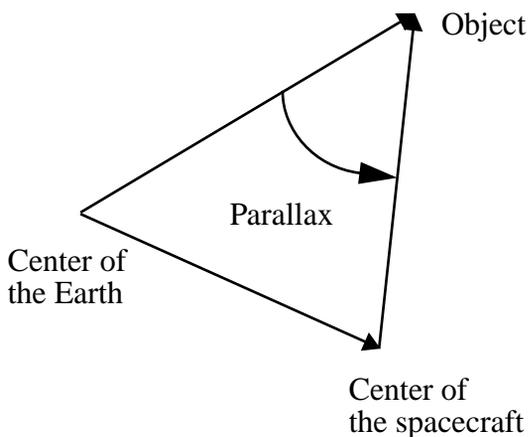
A second common reference frame is the earth-pointing frame with +z pointing at the earth (nadir), -y along the orbit normal and +x completing the right-hand set. “Mean” coordinates for the Earth neglect the earth’s nutational motion. If Earth nutation is include, the coordinates are known as “true”.

A few other definitions are in order. Nadir refers to the vector from the center of the spacecraft to the center of the earth. Zenith points away from the earth. In the earth-pointing frame nadir is the +z axis and zenith is -z. The nadir vector would always be parallel to the spacecraft position vector if the Earth were spherically symmetric.

Celestial objects are often referred to the earth-centered frame. When viewed from the spacecraft centered frame, they will appear to be at a slightly different angle. This is known as parallax. It is defined below.

**FIGURE 6.2 Parallax Angle Diagram**

---



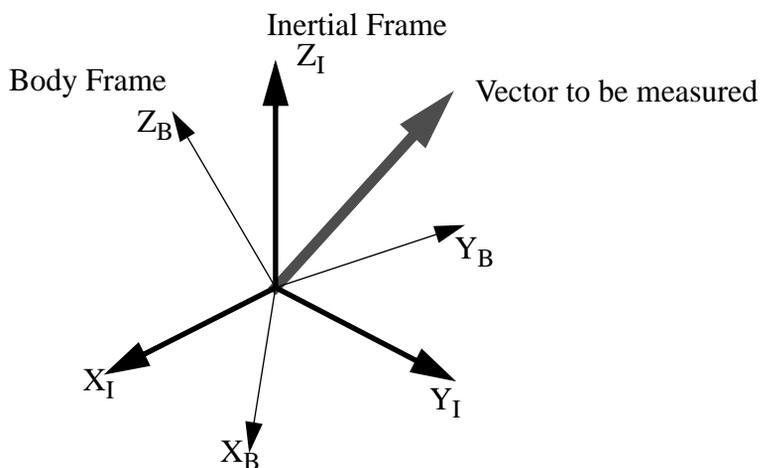
Parallax is not important for distant objects such as stars, but should be accounted for when measuring sun angles, and definitely for when measuring lunar angles.

## Coordinate Transformations

Given two frames, the orientation of one frame with respect to the other can be defined in a number of ways. Three that are widely used are quaternions, Euler angles and transformation matrices.

A transformation matrix is an orthonormal matrix that transforms a vector measured in one frame to another frame. An orthonormal matrix is a matrix whose inverse equals its transpose and each column or row is a unit vector. The following illustration shows a vector that could be measured in two frames.

**FIGURE 6.3 Dual Frame Vector Diagram**



The vector orientation and length do not change, but its x, y and z components do depending on which frame is used to measure it.

If  $u_b$  is the vector in the body frame, and  $u_I$  is the vector as measured in the inertial frame, the two are related by

$$u_I = M_{BToI} u_b \quad (6.1)$$

Note that

$$M_{IToB} = M_{BToI}^T \quad (6.2)$$

is the MATLAB notation for transpose. The transformation matrix is defined as

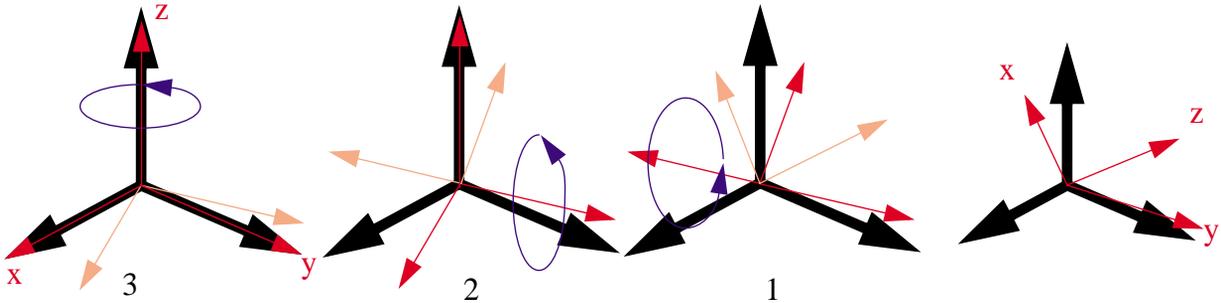
$$M_{BToI} = \begin{bmatrix} x_{BinI}^T \\ y_{BinI}^T \\ z_{BinI}^T \end{bmatrix} \quad (6.3)$$

where each row is the unit vector of the axis of the body frame as measured in the I-frame.

### Euler Angles

Euler angles are commonly used to represent attitude.

**FIGURE 6.4 Euler Angle Diagram**



The first rotation is about the 3-axis of both the inertial and body frames. The second is about the 2-axis of the body frame and the third is about the 1-axis of the body frame.

Each rotation can be used to define a transformation matrix and the product of the three gives the overall transformation matrix.

$$M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_1 & \sin\theta_1 \\ 0 & -\sin\theta_1 & \cos\theta_1 \end{bmatrix} \begin{bmatrix} \cos\theta_2 & 0 & -\sin\theta_2 \\ 0 & 1 & 0 \\ \sin\theta_2 & 0 & \cos\theta_2 \end{bmatrix} \begin{bmatrix} \cos\theta_3 & \sin\theta_3 & 0 \\ -\sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.4)$$

There are twelve possible sets of Euler angles. Each set of three rotations is capable of pointing a unit vector in any orientation.

Alternately,  $M$  could be obtained by a single rotation about a unit vector. The combination of rotation angle and unit vector gives the quaternion.

---

## 6.2 Transformation Matrices

Figure 6-2 shows two coordinate frames. Each of the axes is represented by a unit vector such that

$$\begin{aligned} Z \times X &= Y \\ X \times Y &= Z \\ z \times x &= y \\ x \times y &= z \end{aligned} \quad (6.5)$$

This implies that the coordinate frames are orthogonal, every axis makes a right angle with every other axis. The vector  $a$  can be parameterized using coordinates from either set of axes. If  $x$ ,  $y$  and  $z$  are parameterized as

$$\begin{aligned} x(X, Y, Z) \\ y(X, Y, Z) \\ z(X, Y, Z) \end{aligned} \tag{6.6}$$

then

$$a(x, y, z) = \begin{bmatrix} x(X, Y, Z) \\ y(X, Y, Z) \\ z(X, Y, Z) \end{bmatrix} a(X, Y, Z) \tag{6.7}$$

The  $x$  component of  $a$  in the  $(x, y, z)$  system is just the dot product of the  $x$  axis,  $x(X, Y, Z)$  with  $a(X, Y, Z)$ . Because the axes are orthogonal and unit vectors the following identity holds

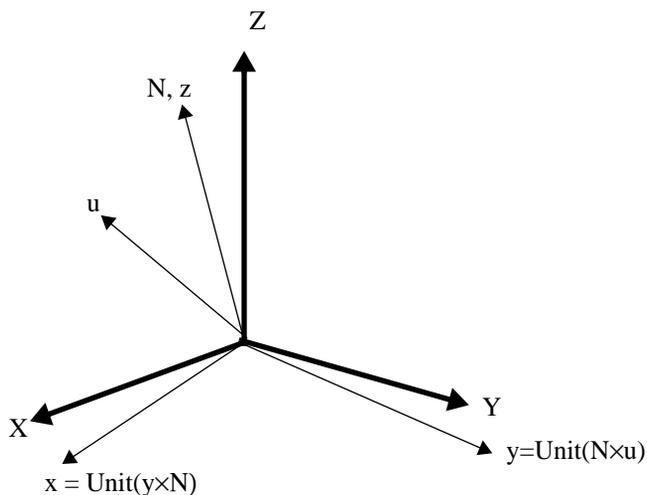
$$E = CC^T \tag{6.8}$$

where  $C$  is the transformation matrix. The transformation matrix from  $(x, y, z)$  to  $(X, Y, Z)$  is just  $C^T$ .

Often it is necessary to create a transformation matrix from two vectors. Figure 6-2 gives an example. Two vectors are

**Figure 6-1 Transformation matrix from two vectors**

---



parameterized in  $(X, Y, Z)$ ,  $N$  and  $u$ . Define  $N$  to be the  $z$ -axis. Then the axes are

$$\begin{aligned}
 z &= \text{Unit}(N) \\
 y &= \text{Unit}(N \times u) \\
 x &= \text{Unit}(y \times z)
 \end{aligned}
 \tag{6.9}$$

$x, y, z$  are orthonormal.

---

## 6.3 Quaternions

### 6.3.1 Introduction

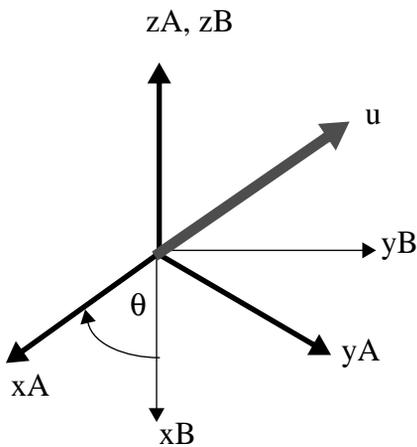
Propagation of the orientation of an object is most efficiently done with quaternions. This is because it has only 4 elements instead of 9 in the transformation matrix, and does not have a singularity like Euler angles.

Euler angles are useful for visualizing the attitude. Quaternions should be used for numerical integration. If more than three vectors are to be transformed from one frame to another using the same quaternion, it is faster to first convert the quaternion to a rotation matrix and then do the matrix-vector multiplies to get the transformations. Transforming a vector directly with a quaternion takes 30 floating point operations; it takes 45 operations to convert from a quaternion to a vector. A matrix transformation takes 15 operations.

A quaternion is a more convenient way of representing the orientation of one frame with respect to another.

**FIGURE 6.5 Quaternion Diagram**

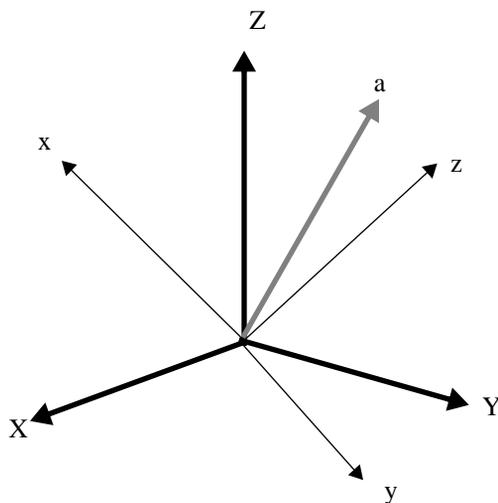
---



Assume that we have two reference frames that have the same z-axis and a vector  $u$  that can be represented in either frame. The length of  $u$  is the same in each frame. In this case the z component of  $u$  will be the same in both frame A and B but the x and y components will have different values. It is important to be precise about the definition of a quaternion. Specifically, in this toolbox, given two frames A and B that are misaligned, then  $q_{A \rightarrow B}$  transforms a vector represented in the A frame coordinates into a vector represented by the B frame coordinates. For example, suppose that we have a spacecraft frame (B) aligned with the inertial frame (I). We then rotate it in a sequence of three rotations. The body frame of the spacecraft is now misaligned with respect to the inertial frame.

**Figure 6-2 Coordinate Frames**

---



### 6.3.2 Fundamental Properties of the Quaternion

A quaternion is a four parameter set used to describe the orientation of one reference frame with respect to a second reference frame. Although only three parameters are needed to uniquely specify the relative orientation, all three parameter sets have singularities which make them unsuitable for numerical simulations, or integration in flight software.

The quaternion is represented by a four row-vector

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} s \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} \cos \frac{\phi}{2} \\ a_1 \sin \frac{\phi}{2} \\ a_2 \sin \frac{\phi}{2} \\ a_3 \sin \frac{\phi}{2} \end{bmatrix} = \mathbf{q} \quad (6-1)$$

The first notation is the standard used in software. No distinction is made between the four elements of the quaternion since numerically there is none. Occasionally authors will make the first element  $q_0$  and the remainder  $[q_1 \ q_2 \ q_3]$  to correspond to the scalar and vector notation. The second notation breaks the quaternion into a scalar and 3-vector part. Letting the first term be the “scalar” is arbitrary.

The third form relates to the theorem of Euler that any arbitrary rotation about any number of axes can be reduced to a single rotation about a fixed axis. If the rotation angle is defined as  $\phi$  and the axis of rotation is  $\mathbf{a}$ ., then the sign of the first value is arbitrary since any rotation can be represented by two quaternions of opposite sign. In this text, the first element will always be positive. This would imply that  $\phi$  will always be between  $\pm 180^\circ$ . The transpose of a quaternion is defined as

$$\begin{bmatrix} q_1 \\ -q_2 \\ -q_3 \\ -q_4 \end{bmatrix} = \begin{bmatrix} s \\ -v_1 \\ -v_2 \\ -v_3 \end{bmatrix} = \begin{bmatrix} \cos \frac{\phi}{2} \\ -a_1 \sin \frac{\phi}{2} \\ -a_2 \sin \frac{\phi}{2} \\ -a_3 \sin \frac{\phi}{2} \end{bmatrix} = \mathbf{q}^* \quad (6.10)$$

with the property that

$$\mathbf{q}\mathbf{q}^T = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (6.11)$$

### 6.3.3 Quaternion Nomenclature

Quaternions will be defined as

$$q_{ab} \quad (6-1)$$

which is defined as the quaternion that rotates a vector from frame b to frame a (right to left in the order of the coefficients). Thus

$$\mathbf{u}_a = q_{ab}\mathbf{u}_b \quad (6-2)$$

and quaternion multiplication is defined as

$$q_{ac} = q_{ab}q_{bc} \quad (6-3)$$

The “inner” frames cancel in the multiplication. Quaternion multiplication will be discussed in the next section.

### 6.3.4 Quaternion Operations

Quaternion multiplication is defined as

$$q_1q_2 = \begin{bmatrix} s_1s_2 - \hat{v}_1\hat{v}_2 \\ s_1\hat{v}_1 + s_2\hat{v}_2 + \hat{v}_1 \times \hat{v}_2 \end{bmatrix} \quad (6.12)$$

This can be expressed in matrix form in two ways

$$q_1q_2 = \begin{bmatrix} q_1(1) & -q_1(2) & -q_1(3) & -q_1(4) \\ q_1(2) & q_1(1) & -q_1(4) & q_1(3) \\ q_1(3) & q_1(4) & q_1(1) & -q_1(2) \\ q_1(4) & -q_1(3) & q_1(2) & q_1(1) \end{bmatrix} q_2 \quad (6.13)$$

or

$$q_1q_2 = \begin{bmatrix} q_2(1) & -q_2(2) & -q_2(3) & -q_2(4) \\ q_2(2) & q_2(1) & q_2(4) & -q_2(3) \\ q_2(3) & -q_2(4) & q_2(1) & q_2(2) \\ q_2(4) & q_2(3) & -q_2(2) & q_2(1) \end{bmatrix} q_1 \quad (6.14)$$

The only difference between Equation (6.10) and Equation (6.11) is that the sign is reversed on the skew symmetric 3×3 matrix in the lower right-hand-corner.

### 6.3.5 Quaternion Transformations

Quaternions transform vectors by means of the following operation

$$\mathbf{x}_b = q_{ba}\mathbf{x}_a q_{ba}^* \quad (6.15)$$

using quaternion multiplication with the vectors defined as quaternions with a scalar part equal to zero or

$$x_a = \begin{bmatrix} 0 \\ x_a(1) \\ x_a(2) \\ x_a(3) \end{bmatrix} \quad (6.16)$$

### 6.3.6 Quaternion Derivative

The derivative of a quaternion defined as

$$\dot{q} = \lim_{\Delta t \rightarrow 0} \frac{\delta q}{\Delta t} = \lim_{\Delta t \rightarrow 0} \frac{q(t + \Delta t) - q(t)}{\Delta t} \quad (6.17)$$

The derivative of the quaternion will be related to the angular velocity of one frame with respect to the other. There are four possible formulations for the derivative of a quaternion. They are

$$\begin{aligned} \{q_{ab} \omega_a\} \quad \{q_{ba} \omega_b\} \\ \{q_{ab} \omega_a\} \quad \{q_{ba} \omega_b\} \end{aligned} \quad (6.18)$$

The quaternion can transform in either direction between a and b and the angular rate can be expressed either in frame a or b. Define

$$q(t + \Delta t) = \Delta q q \quad (6.19)$$

The new quaternion is just the old multiplied by a quaternion that transforms the first into the second. Define

$$\omega = a \omega \quad (6.20)$$

where a is the unit vector for  $\omega$ .

The derivations for each of the four sets are straightforward. Start with set  $\{q_{ba} \omega_b\}$ .  $\Delta q$  becomes, when expressed in the scalar-vector form of the quaternion

$$\Delta q = \begin{bmatrix} \cos \frac{\omega \Delta t}{2} \\ a \sin \frac{\omega \Delta t}{2} \end{bmatrix} \quad (6.21)$$

and Equation (6.17) becomes

$$\dot{q} = \lim_{\Delta t \rightarrow 0} \frac{\delta q}{\Delta t} = \lim_{\Delta t \rightarrow 0} \frac{(\Delta q - e)q}{\Delta t} \quad (6.22)$$

$\Delta q$  is the state transition matrix for the quaternion

$$\Delta q = \begin{bmatrix} \cos\left(\frac{\omega\Delta t}{2}\right) & -\mathbf{a}^T \sin\left(\frac{\omega\Delta t}{2}\right) \\ \mathbf{a} \sin\left(\frac{\omega\Delta t}{2}\right) & \mathbf{E}(3)\cos\left(\frac{\omega\Delta t}{2}\right) - \tilde{\mathbf{a}} \sin\left(\frac{\omega\Delta t}{2}\right) \end{bmatrix} \quad (6.23)$$

Equation (6.22) becomes

$$\dot{q} = \lim_{\Delta t \rightarrow 0} \frac{q}{\Delta t} \begin{bmatrix} \cos\left(\frac{\omega\Delta t}{2}\right) - 1 \\ \mathbf{a} \sin\frac{\omega\Delta t}{2} \end{bmatrix} \quad (6.24)$$

Switching the order of the multiplication and putting  $\Delta q$  into matrix form gives

$$\dot{q} = \lim_{\Delta t \rightarrow 0} \left( \frac{1}{\Delta t} \begin{bmatrix} \cos\left(\frac{\omega\Delta t}{2}\right) - 1 & -\mathbf{a}^T \sin\left(\frac{\omega\Delta t}{2}\right) \\ \mathbf{a} \sin\left(\frac{\omega\Delta t}{2}\right) & \mathbf{E}(3)\left(\cos\left(\frac{\omega\Delta t}{2}\right) - 1\right) - \tilde{\mathbf{a}} \sin\left(\frac{\omega\Delta t}{2}\right) \end{bmatrix} q \right) \quad (6.25)$$

In the limit the cosine terms become 1 and the sine terms become their arguments.

$$\dot{q} = \left(\frac{\omega}{2}\right) \begin{bmatrix} 0 & -\mathbf{a}^T \\ \mathbf{a} & -\tilde{\mathbf{a}} \end{bmatrix} q \quad (6.26)$$

or expressing the angular rate as a vector

$$\dot{q} = \frac{1}{2} \begin{bmatrix} 0 & -\omega^T \\ \omega & -\tilde{\omega} \end{bmatrix} q. \quad (6.27)$$

### 6.3.7 Linearization of the Quaternion

The differential equation for a quaternion that transforms from frame A to B when frame B is rotating relative to frame A with rate  $w$  as measured in frame B is

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -q_2 & -q_3 & -q_4 \\ q_1 & q_4 & q_3 \\ -q_4 & q_1 & -q_2 \\ -q_3 & q_2 & q_1 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \quad (6.28)$$

These differential equations are bilinear because all of the terms on the right-hand-side are products of the inputs and the states.

Often it is required to look at small deviations in attitude for a (nearly) steadily rotating spacecraft. The body rate can be expressed as

$$\omega_{ca} = \omega_{cb} + C_{cb}\omega_{ba} \quad (6.29)$$

where  $\omega_{cb}$  is the body rate with respect to the rotating frame,  $C_{cb}$  is the transformation matrix from the rotating frame to the body frame,  $\omega_{ba}$  is the constant rate vector of the rotating frame with respect to the reference frame, and  $\omega_{ca}$  is the body rate with respect to the reference frame as measured in the body frame.

Assume a steady rotation about the x-axis at a rate of  $n$  radian/sec. The nominal quaternion from the rotating frame to the body is  $[1 \ 0 \ 0 \ 0]'$ , therefore for the quaternion representation

$$C_{cb} = \begin{bmatrix} 1 & 2q_4 & -2q_3 \\ -2q_4 & 1 & 2q_2 \\ 2q_3 & -2q_2 & 1 \end{bmatrix} \quad (6.30)$$

and the quaternion rate equation becomes

$$\begin{bmatrix} \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & n \\ 0 & -n & 0 \end{bmatrix} \begin{bmatrix} q_2 \\ q_3 \\ q_4 \end{bmatrix} + \frac{1}{2} \left( \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} - \begin{bmatrix} n \\ 0 \\ 0 \end{bmatrix} \right) \quad (6.31)$$

The nominal angle vector is  $[0 \ 0 \ 0]'$ , therefore the same equation for rotation angles is

$$\begin{bmatrix} \dot{\theta}_x \\ \dot{\theta}_y \\ \dot{\theta}_z \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & n \\ 0 & -n & 0 \end{bmatrix} \begin{bmatrix} \theta_x \\ \theta_y \\ \theta_z \end{bmatrix} + \frac{1}{2} \left( \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} - \begin{bmatrix} n \\ 0 \\ 0 \end{bmatrix} \right) \quad (6.32)$$

This leads to the equations

$$\begin{aligned}
 \dot{\theta}_x &= \omega_x \\
 \ddot{\theta}_y + n^2 \theta_y &= \dot{\omega}_x + \omega_z \\
 \ddot{\theta}_z + n^2 \theta_z &= -\omega_y + \dot{\omega}_z
 \end{aligned} \tag{6.33}$$

and the eigenvalues for the kinematics

$$[0 \pm n] \tag{6.34}$$

For the non-spinning case the two sets of equations become

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{6.35}$$

and

$$\begin{bmatrix} \dot{\theta}_x \\ \dot{\theta}_y \\ \dot{\theta}_z \end{bmatrix} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{6.36}$$

$q_1$  is always one and the remainder differ from the corresponding angle only by a scale factor of 2.

One might also have linearized (1) about  $[1 \ 0 \ 0 \ 0]'$ ,  $[n \ 0 \ 0]'$ . This would lead to the equations

$$\begin{aligned}
 \dot{q}_1 &= \frac{1}{2}(-nq_2 + \omega_x) \\
 \dot{q}_2 &= \frac{1}{2}(nq_1 + \omega_x) \\
 \dot{q}_3 &= \frac{1}{2}(nq_4 + \omega_y) \\
 \dot{q}_4 &= \frac{1}{2}(-nq_2 + \omega_z)
 \end{aligned} \tag{6.37}$$

which leads to the repeated eigenvalues  $(\pm n/2)$ . This quaternion is no longer with respect to the rotating frame, it is with respect to the reference frame. The quaternion from the reference frame to the rotating frame is

$$q_{ba} = \begin{bmatrix} \cos\left(\frac{nt}{2}\right) \\ \sin\left(\frac{nt}{2}\right) \\ 0 \\ 0 \end{bmatrix} \quad (6.38)$$

A solution to the quaternion differential equation from the rotating frame to the body frame is

$$q_{cb} = \begin{bmatrix} 1 \\ 0 \\ \cos nt \\ -\sin nt \end{bmatrix} \quad (6.39)$$

Multiplying the two quaternions gives

$$q_{ca} = \begin{bmatrix} 1 & 0 & -\cos nt & -\sin nt \\ 0 & 1 & -\sin nt & \cos nt \\ \cos nt & \sin nt & 1 & 0 \\ \sin nt & -\cos nt & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\left(\frac{nt}{2}\right) \\ \sin\left(\frac{nt}{2}\right) \\ 0 \\ 0 \end{bmatrix} \quad (6.40)$$

or

$$q_{ca} = \begin{bmatrix} \cos\left(\frac{nt}{2}\right) \\ \sin\left(\frac{nt}{2}\right) \\ \cos\left(\frac{nt}{2}\right) \\ \sin\left(\frac{nt}{2}\right) \end{bmatrix} \quad (6.41)$$

which has frequencies of  $n/2$ .

(6.42)

---

## 6.4 *Small Angles*

Often spacecraft control systems deal with small deviations from a nominal attitude. In these cases all attitude representations are the same and the attitude can be treated as a vector.

---

## 6.5 *Physical Interpretation of the Quaternion*

Since the quaternion on first appearance does not appear to impart a physical feeling for what attitude it represents, we will go through a few examples to show that with a little effort, a quaternion is as easy to visualize as Euler angles.

### **Single axis rotation**

The quaternion  $[0.7071 \ 0.7071 \ 0.0 \ 0.0]'$  represents a pure rotation about the x-axis. The first argument is 0.7071 and equals the  $\cos(90^\circ/2)$ . We cannot tell the quadrant from the first argument. The second argument is the 1 component of the unit vector, (which in this case is  $[1 \ 0 \ 0]'$ ) times the argument  $\sin(90^\circ/2)$ . Since the sign is positive, the rotation must be a positive  $90^\circ$  rotation.

Large multi-axis rotations are somewhat more difficult to interpret.

### **Multiple axis rotation**

The quaternion  $[0.5 \ 0.5 \ 0.5 \ 0.5]'$  represents a rotation. The first argument is 0.5 and equals the  $\cos(120^\circ/2)$ . The  $\sin(120^\circ/2)$  is 0.866. Therefore the unit vector,  $\mathbf{a}$ , is  $[0.5773 \ 0.5773 \ 0.5773]$ . So this quaternion represents a  $+60^\circ$  rotation about  $\mathbf{a}$  or a  $-60^\circ$  about  $-\mathbf{a}$ .

If the rotation is sufficiently small, the last three components of the quaternion represent half the equivalent angles.

### **Small rotation**

The quaternion  $[0.999975 \ 0.004 \ 0.003 \ -0.005]'$  represents a rotation. The first argument equals the  $\cos(0.8^\circ/2)$  which is small. Therefore the last three angles can be interpreted directly as angles (in radians). Therefore, this quaternion represents  $\approx$  the rotation vector  $[0.460^\circ \ 0.344^\circ \ 0.572^\circ]$ .

## 6.6 **Linearization of the Quaternion**



---

## 7.1 Introduction

This chapter describes control system design. The fundamental approach used throughout is loop shaping. This is applied both for continuous controllers and discrete-time control systems.

---

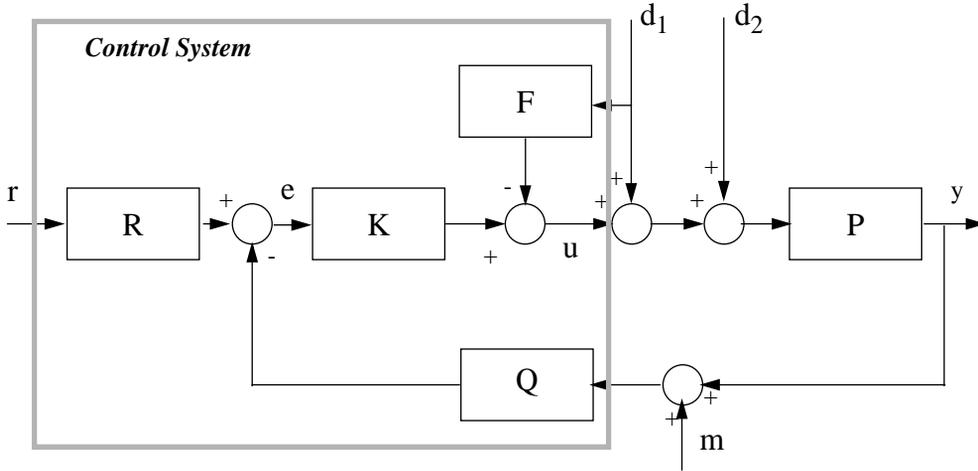
## 7.2 The General Control System

Figure 1 shows a general block diagram of a control system.  $r$  is the command input to the system.  $m$  is the measurement noise input.  $y$  is the system measured output.  $d_1$  is the disturbance that is measured and used for feedforward purposes.  $d_2$  is the unmeasured disturbance.  $R$  is the command prefiltering block,  $K$  is the control block,  $Q$  is the feedback block,  $F$  is the feedforward block and  $P$  is the plant.  $P$  includes all plant dynamics including those of the actuators and sensors. If the system is implemented digitally,  $K$  and  $F$  include samplers on the inputs and sample and holds, or pulsewidth modulators, etc. on the outputs. Assume  $E$  is the identity matrix.

Often unity feedback configurations are chosen in which  $Q = I$ . If  $r$  is always 0 then the system is a regulator and  $R$  is zero. Many control systems do not have disturbance feedback and  $F = 0$ . Sometimes the disturbance inputs are put after the plant and the relevant plant dynamics are included in the disturbances.

This block diagram applies equally well to systems with multiple inputs and outputs. It is important to remember that when dealing with multi-output system (which all real systems are), that not all outputs are measured by control sensors, but may still be important to the performance.

Figure 1. General Control system block diagram



### 7.3 Fundamental Relationships

In this section we will derive algebraic relationships between the inputs and outputs of Figure 1. The error  $e$  is

$$e = Rr - Q(m + y) \tag{7.1}$$

the output,  $y$  is

$$y = P((E - F)d + u) \tag{7.2}$$

and the control output is

$$u = Ke \tag{7.3}$$

The system inputs are  $m$ ,  $r$  and  $d$ .  $u$  and  $e$  are intermediate quantities. Solving for  $y$  in terms of the inputs

$$(E + PKQ)y = P(E - F)d + PK(Rr - Qm) \tag{7.4}$$

The feedforward matrix attenuates the disturbance  $d$  and may modify its dynamics. For example, if  $d$  were a step input and  $F$  were a low-pass filter,  $d$  would then look like the product of a step and an exponential decay. Since  $F$  cannot equal  $E$  without drawing infinite power, some of  $d$  will always pass through. Therefore, no generality is lost if  $F$  is dropped from further discussion since the magnitude and dynamics of  $d$  are not significant in this discussion.

$Q$  modifies the characteristics of the feedback path. If we set

$$K = KQ^{-1} \tag{7.5}$$

and

$$\mathbf{R} = \mathbf{Q}\mathbf{R} \quad (7.6)$$

Q disappears. Q may give some design flexibility when mechanizing the control system, but from a strictly algebraic point-of-view can be eliminated.

Eliminating Q and F and multiplying both sides by E+PK gives

$$y = (\mathbf{E} + \mathbf{PK})^{-1}(\mathbf{P}\mathbf{d} + \mathbf{PK}(\mathbf{R}\mathbf{r} - \mathbf{m})) \quad (7.7)$$

where

$$\mathbf{o} = \mathbf{E} + \mathbf{PK} \quad (7.8)$$

is known as the output return difference. This is because if the feedback loop were broken at the output of the plant,  $\mathbf{P}$ , then  $\mathbf{F}_o$  would be  $\mathbf{E}$ , the identity matrix. The return difference shows the effect of feedback on the output of the plant. In Equation (7.7) note that r and m are indistinguishable to the output, except that r is multiplied by the prefiltering matrix  $\mathbf{R}$ .

The inverse of Equation (7.8) is known as the sensitivity,  $\mathbf{S}$ . It shows how sensitive the output is to the product of the plant dynamics and the disturbances. If the sensitivity is small, the output will be insensitive to disturbances. If it is one, the product of the plant dynamics and the disturbances will be passed unattenuated to the output. The sensitivity is one when the loop is opened so the sensitivity (at any given frequency) shows how close the system is to being open-loop at that frequency. Equation (7.7) can be rewritten as

$$y = \mathbf{S}\mathbf{P}\mathbf{d} + \mathbf{G}_c\mathbf{r} - \mathbf{T}\mathbf{m} \quad (7.9)$$

where  $\mathbf{G}_c$  is the transmission function and equals

$$\mathbf{G}_c = \mathbf{T}\mathbf{R} \quad (7.10)$$

and determines how well y will track r. Ideally,  $\mathbf{G}_c$  would equal E, by letting

$$\mathbf{R} = \mathbf{T}^{-1} \quad (7.11)$$

but again that would require infinite power and is not possible. T is the closed loop transfer function

$$\mathbf{T} = \mathbf{S}\mathbf{P}\mathbf{K} \quad (7.12)$$

and determines how well the noise is attenuated. Ideally it should equal zero. It is sometimes called the complementary sensitivity because

$$\mathbf{S} + \mathbf{T} = \mathbf{E} \quad (7.13)$$

This leads to the fundamental trade-off in control design. Ideally **S** and **T** would be equal to zero everywhere but since their sum is the identity matrix, this is not possible. At best, and what control design is really all about, **S** can be made small where **Pd** is large and **T** small where **m** is large. This works in practice since **Pd** is usually large at low frequencies and small at high frequencies while sensor noise is generally high at high frequencies, and low at low frequencies except for a component near dc (known as a bias) which varies slowly with time. Equation (7.1) summarizes the relationship between **PK**, **T** and **S**.

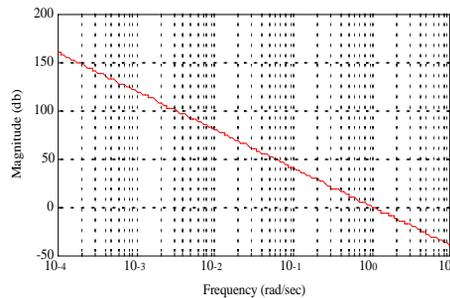
Figure 2 shows a double integrator plant, which represents a single axis of a satellite in its simplest form. At  $\omega=0$  it

**TABLE 7.1 Relationships Between PK and T and S**

	PK small	PK large
T	0	E
S	E	0

has infinite gain and the gain rolls off as  $1/\omega^2$ . It is clear how important **P** is when evaluating the effect of disturbances. In this case any steady-state disturbance will eventually cause the output to grow very large. Many practical actuators, including all momentum exchange devices, have high-pass filter characteristics because they cannot store infinite momentum. Consequently, an additional form of actuation (such as magnetic torquers or thrusters) is necessary when controlling a spacecraft

**Figure 2. Double Integrator**



It is also important to keep **u** small, since that determines how much power is consumed, how large the actuators must be, etc. The control output is

$$\mathbf{u} = (\mathbf{E} + \mathbf{KP})^{-1} \mathbf{K}(\mathbf{Rr} - \mathbf{m} - \mathbf{Pd}) \tag{7.1}$$

The factor  $\mathbf{E} + \mathbf{KP}$  is called the input return difference (denoted by  $\mathbf{F}_i$ ) since it shows the effect of feedback on the input to the plant from the control. Notice that **K** and **P** are transposed from the output return difference function. If the plant is single-input-single-output **K** and **P** are scalars so the order doesn't matter. For multi-input and multi-output plants, it does make a difference.

If  $KP$  is  $\gg E$  Equation (7.1) becomes

$$u = P^{-1}(Rr - m) - d \quad (7.2)$$

This says that if the control is fast enough, it will exactly cancel the disturbances,  $d$ . The reference and the noise inputs will be amplified by the inverse of the plant. Thus, in the double integrator example, high frequency references and noise will require more command energy than low frequency.

Finally, look at the tracking error which is defined as

$$e = r - y \quad (7.3)$$

Substituting,

$$e = (E - TR)r - (E - T)Pd + Tm \quad (7.4)$$

$T$  should equal  $E$  to eliminate tracking errors due to disturbances, but that would cause all noise to pass through.  $TR$  should equal  $E$  to eliminate tracking errors due to the reference input. Of course, for  $TR$  to equal  $E$ ,  $R$  would have to be the inverse of  $T$ , the closed loop transfer function, which is not physically possible. Nonetheless, these goals can be achieved over different parts of the frequency spectrum. Thus, at frequencies where the reference is likely to have energy  $TR$  should equal  $E$ . At disturbance frequencies  $T$  should equal  $E$  and everywhere else  $T$  should equal zero to minimize noise.

**TABLE 7.2** What Should be Small to Compensate for Large Inputs

	<b>m large</b>	<b>Pd large</b>	<b>r large</b>
y	T	S	N/A
e	T	<b>E-T</b>	E-TR
u	$F_i K$	<b><math>F_i K</math></b>	$F_i K R$

---

## 7.4 Tracking Errors

In Equation (7.4) if  $R = E$  then the transfer function between the reference input and the error is just the sensitivity  $S$ .  $Rr$  is really just a modification of the input signal dynamics, therefore the sensitivity will determine the tracking error to the output of  $r$ .

Disturbances and reference signals can be characterized in the frequency domain just as plants and controllers can. Typical inputs are given with their transfer functions in Equation (7.3)

**TABLE 7.3 Inputs**

Input	Time Domain	Frequency Domain
Impulse	$\delta$	1
Step	A	1/s
Ramp	At	1/s <sup>2</sup>
Sinusoid	Asin $\omega t$	$\omega/(s^2 + \omega^2)$

**R** can be used to modify the dynamics of the inputs. This can make it easier for the closed loop system to follow its input.

For example, given the simple single-input-single output plant with a gain of  $k$ , a  $1/s$  plant, an input transfer function of **R** and a ramp input, the error is

$$\frac{e}{A} = \frac{R}{s^2} \left( \frac{s}{s+k} \right) \quad (7.1)$$

where the quantity in parenthesis is the sensitivity. The steady-state tracking error can be found using the final value theorem which states that

$$\lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} sf(s) \quad (7.2)$$

**R** must be of order  $s$  for the tracking error to go to zero in steady-state. A suitable transfer function would be

$$R = \frac{s}{\tau s + 1} \quad (7.3)$$

An alternative is to make the controller a proportional-integral controller.

$$K = k \left( 1 + \frac{1}{\tau s} \right) \quad (7.4)$$

Then the sensitivity would become

$$S = \frac{\tau s^2}{\tau s^2 + k\tau s + 1} \quad (7.5)$$

which has the same desired effect. In general, for the tracking error to approach zero asymptotically, the sensitivity must be of the same order as the input.

---

## 7.5 State Space Closed Loop Equations

Figure 1 shows the block diagram in the unity feedback configuration without the command prefiltering. 1

The control block and the plant can both be represented by sets of first order differential equations

$$\dot{\mathbf{x}}_p = \mathbf{A}_p \mathbf{x}_p + \mathbf{B}_p u_p \quad (7.6)$$

$$\mathbf{y}_p = \mathbf{C}_p \mathbf{x}_p + \mathbf{D}_p u_p \quad (7.7)$$

$$\dot{\mathbf{x}}_c = \mathbf{A}_c \mathbf{x}_c + \mathbf{B}_c u_c \quad (7.8)$$

$$\mathbf{y}_c = \mathbf{C}_c \mathbf{x}_c + \mathbf{D}_c u_c \quad (7.9)$$

$$u_c = \mathbf{r} - \mathbf{m} - \mathbf{y}_p \quad (7.10)$$

$$u_p = \mathbf{d} + \mathbf{K} \mathbf{y}_c \quad (7.11)$$

The state equations for the closed loop plant are

$$\dot{\mathbf{x}} = \mathbf{A} \mathbf{x} + \mathbf{B}_d \mathbf{d} + \mathbf{B}_m (\mathbf{r} - \mathbf{m}) \quad (7.12)$$

and the measurement equation is

$$\mathbf{y} = \mathbf{C} \mathbf{x} + \mathbf{D}_d \mathbf{d} + \mathbf{D}_m (\mathbf{r} - \mathbf{m}) \quad (7.13)$$

where

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_p \\ \mathbf{x}_c \end{bmatrix} \quad (7.14)$$

$$= \begin{bmatrix} \mathbf{A}_p - \mathbf{B}_p \mathbf{K} \mathbf{D}_c \mathbf{F} \mathbf{C}_p & \mathbf{B}_p \mathbf{K} (\mathbf{C}_c - \mathbf{D}_c \mathbf{F} \mathbf{D}_p \mathbf{K} \mathbf{D}_c) \\ -\mathbf{B}_c \mathbf{F} \mathbf{C}_p & \mathbf{A}_c - \mathbf{B}_c \mathbf{F} \mathbf{D}_p \mathbf{K} \mathbf{C}_c \end{bmatrix} \quad (7.15)$$

$$\mathbf{B}_d = \begin{bmatrix} \mathbf{B}_p (\mathbf{E} - \mathbf{K} \mathbf{D}_c \mathbf{F} \mathbf{D}_p) \\ -\mathbf{B}_c \mathbf{F} \mathbf{D}_p \end{bmatrix} \quad (7.16)$$

$$\mathbf{B}_m = \begin{bmatrix} \mathbf{B}_p \mathbf{K} \mathbf{D}_c (\mathbf{F} \mathbf{D}_p \mathbf{K} \mathbf{D}_c - \mathbf{E}) \\ \mathbf{B}_c (\mathbf{E} - \mathbf{B}_p \mathbf{K} \mathbf{D}_c \mathbf{F} \mathbf{D}_p) \end{bmatrix} \quad (7.17)$$

$$= \begin{bmatrix} \mathbf{F} \mathbf{C}_p \\ \mathbf{F} \mathbf{D}_p \mathbf{K} \mathbf{C}_c \end{bmatrix} \quad (7.18)$$

$$\mathbf{D}_d = \mathbf{F} \mathbf{D}_p \quad (7.19)$$

$$\mathbf{m} = -\mathbf{F} \mathbf{D}_p \mathbf{K} \mathbf{D}_c \quad (7.20)$$

$$\mathbf{F} = (\mathbf{E} + \mathbf{D}_p \mathbf{K} \mathbf{D}_c)^{-1} \quad (7.21)$$

Adding the prefilter plant is straightforward.  $\mathbf{r}$  is replaced by  $\mathbf{y}_r$  and an additional set of state equations

$$\dot{\mathbf{x}}_r = \mathbf{A}_r \mathbf{x}_r + \mathbf{B}_r r \quad (7.22)$$

$$\mathbf{y}_r = \mathbf{C}_r \mathbf{x}_r + \mathbf{D}_r r \quad (7.23)$$

$\mathbf{B}_m \mathbf{C}_r$  is added to the state equations for  $\mathbf{x}_p$  and  $\mathbf{x}_c$ .  $\mathbf{B}_m \mathbf{D}_r$  becomes a new input matrix with the third block row equal to  $\mathbf{B}_r$ .  $\mathbf{y}$  is unchanged. The final state equation becomes

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{A}_{pp} & \mathbf{A}_{pc} & \mathbf{B}_{mp} \mathbf{A}_r \\ \mathbf{A}_{cp} & \mathbf{A}_{cc} & \mathbf{B}_{mc} \mathbf{A}_r \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_r \end{bmatrix} \begin{bmatrix} \mathbf{x}_p \\ \mathbf{x}_c \\ \mathbf{x}_r \end{bmatrix} + \begin{bmatrix} \mathbf{B}_{pm} & \mathbf{B}_{pd} & \mathbf{B}_{mp} \mathbf{D}_r \\ \mathbf{B}_{cm} & \mathbf{B}_{cd} & \mathbf{B}_{mc} \mathbf{D}_r \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_r \end{bmatrix} \begin{bmatrix} \mathbf{m} \\ \mathbf{d} \\ r \end{bmatrix} \quad (7.24)$$

---

## 7.6 Approaches to Robust Control

### 7.6.1 Introduction

This section outlines several approaches to robust control. Robustness is the property of a closed loop system by which it remains stable and delivers acceptable performance in the face of unmodeled perturbations to the plant, disturbances, and noise. These perturbations may be as simple as unmodeled gain variations in an actuator, or as complex as variations in the structure of the plant model. The two forms of robustness are known as stability and performance robustness. When discussed in the context of multivariable control systems, an additional important property, sometimes known as the integrity of the system, is its stability in the face of loop failures.

Since the field of robust control is very active, it is not possible to describe all methodologies. Noticeably absent in this proposal are detailed discussions of adaptive control and variable structure control. Adaptive control has been used with a fair degree of success in the process control industry, but does not, at the present time, appear suitable for fully autonomous control systems. Variable structure control has been proposed for control systems, and has been successfully applied to induction motor controllers, among other applications, but since it is a nonlinear control approach it requires extensive simulations to validate, making it costly to implement and impractical given the schedule constraints on most flight software.

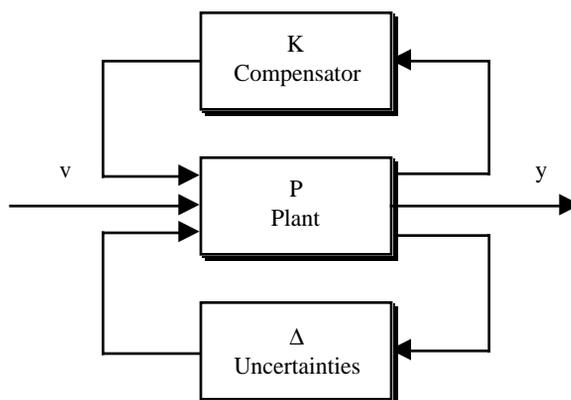
### 7.6.2 Modeling Uncertainty

Robust control techniques require models of the uncertain dynamics. The form of the model will directly determine the conservativeness of the resulting control system. The more conservative a design is, the poorer its performance will be with the nominal dynamics; this can lead to a need for better, and more expensive, actuators and sensors.

In classical control design, uncertainty is often represented by loop gain uncertainty at the phase cross-over frequencies and phase uncertainty at the gain cross-over frequencies. These translate into the gain and phase margins. Complex systems, such as those with bending or nutation modes near or within the control system bandwidth, have multiple crossovers. For example, a designer of a single-input single-output (SISO) system might decide that the frequency of the first bending mode frequency was so uncertain that the loop crossover frequency had to be significantly below the nominal first bending mode frequency so that the bending mode was gain stabilized. Gain stabilization implies that the mode peak is below 0 dB and no amount of phase shift can cause it to destabilize the closed-loop system. Naturally, this leads to a lower performance since the bandwidth is restricted and lower than the bending mode frequency. If the modal frequency were better known, a notch filter might be employed or the mode might be phase stabilized. These solutions would lead to better performance.

A standard representation of uncertainty is illustrated below.

**Figure 3. Standard representation of uncertainty**



All uncertainty in the plant is pulled out into the transfer matrix  $\Delta$ . The simplest approach is to permit  $\Delta$  to be unstructured and to only restrict its 'size' which is measured by the matrix norm

$$\|\Delta(s)\|_{\infty} \quad (7.25)$$

This expression is the maximum value of the norm of  $\Delta(s)$ . For a single-input single-output transfer function, it would be the peak magnitude on a Bode plot.

The size can also be made frequency dependent by pre and post multiplication by frequency dependent weighting functions. The uncertainty may be treated as an additive perturbation to the plant or a multiplicative uncertainty. In the latter case, which is often more useful since the size of  $\Delta$  can be specified as a percentage variation in the plant dynamics, the uncertainty must be specified as either an input or output multiplicative uncertainty since matrix multiplication does not commute.

When a robust design is based on the size of  $\Delta$ , the system will remain stable regardless of the variations within  $\Delta$  as long as the size remains within the prespecified limits. This can lead to a conservative design. For example, even though it may be known that  $\Delta$  is diagonal, a given size of  $\Delta$  can also accommodate off-diagonal terms. Therefore, a design based on the size will tend to be conservative.

A less-conservative approach is to account for the structure of  $\Delta$ . Structured uncertainty methods take advantage of knowing which elements of  $\Delta$  are non-zero.  $\Delta$  takes on a block-structure with submatrices representing unstructured uncertainty. The uncertainty model might be

$$\Delta = \begin{bmatrix} \delta_r & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \Delta_3 \end{bmatrix} \quad (7.26)$$

where  $\Delta_3$  represents the plant uncertainty and  $\delta_r$  is the roll channel gain uncertainty. The pitch channel uncertainty is zero. This model, while better than the unstructured model, is still conservative because the elements of  $\Delta$  may be known to be pure real, yet any general bound on  $\Delta$  will permit complex elements. As a consequence, a design employing this form of structured uncertainty will accommodate phase uncertainty that may not be physically possible.

A more accurate representation of uncertainty is to compute the set of possible frequency responses of the plant at each frequency. For example, for a double integrator with a flexible mode and collocated sensor and actuator, the plant is

$$\frac{\theta(s)}{T(s)} = \frac{s^2 + 2\zeta\omega s + \omega^2}{Is^2(s^2 + \gamma(2\zeta\omega s + \omega^2))} \quad (7.27)$$

where  $\gamma$  is the modal coupling factor

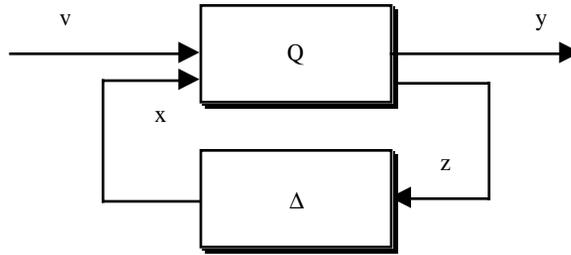
$$\gamma = 1 + \frac{I_{\mu}}{I} \quad (7.28)$$

where  $I_\mu$  is the modal inertia and  $\gamma$  is  $\geq 1$ . The uncertainty template would be found by computing all values of the transfer function for the specified range of the parameters. This permits a direct representation of the uncertainty with all of the gain and phase information.

Equation (7.27) is particularly relevant to aircraft since the uncertain dynamics are often in that form. Generally,  $I$  and  $\omega$  are known accurately, but  $\gamma$  and  $\zeta$  are not. Nonetheless, the transfer function varies in a deterministic way with these parameters and it is easy to find a set of uncertain plants for realistic ranges of  $\gamma$  and  $\zeta$ . There is nothing to prevent the use of a nonlinear relationship between the gain, phase and plant parameters.

The second form of robustness, performance robustness can be related to the structured uncertainty described above.

**Figure 4. Compensated plant**



Given the model in Figure 4, the transfer function is

$$\begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \begin{bmatrix} v \\ z \end{bmatrix} \quad (7.29)$$

and performance robustness can be defined as

$$\begin{aligned} & \|Q_{11} + Q_{12}\Delta(I - Q_{22}\Delta)^{-1}Q_{21}\|_\infty < 1 \\ & \|Q_{22}\|_\mu < 1 \\ & \Delta = \text{diag}\{\Delta_1, \dots, \Delta_n\} \\ & \|\Delta_i\| \leq 1 \end{aligned} \quad (7.30)$$

where

$$\|Q_{22}\|_\mu = \sup \mu(Q_{22}(j\omega)) \quad (7.31)$$

and  $\mu(Q)$  is the structured singular value of  $Q$  for a block diagonal  $\Delta$ . Structured uncertainty can be used both to design for performance robustness and to make designs less conservative when designing for stability robustness.

These uncertainty models do not accommodate loop failures and the resulting designs may be unstable when loops, or combinations of loops are broken. For a single-input single-output system, the open-loop performance and stability is just that of the open-loop plant. For a multi-input multi-output design, the performance will be that of the remainder of the closed-loop plant. With three loops in a 3-axis controlled aircraft, there are three possible systems with one loop broken, and six if two loops break.

### 7.6.3 Control Structure Design

The pairing of inputs and outputs in many systems, such as aircraft and turbofan engines, is of critical importance in designing a low-cost robust system with good performance. Since aircraft controls affect many modes of the system effective pairing can reduce controller complexity.

### 7.6.4 Nyquist Like Techniques

Nyquist-like techniques are extensions of single-input single-output design for multivariable systems. Three common techniques are sequential loop closing, the characteristic-loci method, and the Nyquist array approach. These approaches all replace the multivariable problem with a sequence of SISO problems. In each method, the first stage is to decouple the system across a range of frequencies with either a pure gain, or dynamic compensator. If the latter is used, there is always the danger of cancelling lightly-damped poles—which often exist in aircraft problems. A simple example of a decoupling stage is in the case of a satellite with significant off-diagonal inertia terms. The state equations for the system are

$$\dot{x} = \begin{bmatrix} 0 & E \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ I^{-1} \end{bmatrix} u \quad (7.32)$$

where  $E$  is a  $3 \times 3$  identity matrix. If the feedback structure is chosen to be of the form

$$u = IKx \quad (7.33)$$

the state equations become

$$\dot{x} = \begin{bmatrix} 0 & E \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ E \end{bmatrix} Kx \quad (7.34)$$

and  $K$  can be designed for the 3 decoupled loops.

In sequential loop closing, the fastest loops (such as a reaction wheel tachometer loop) are closed first. If the separation of loop bandwidths is sufficient, this can lead in a straightforward fashion to the final design. When this is not the case, the design usually proceeds in an ad-hoc manner and design decisions made when closing the first loop may have deleterious effects later. The only means available for reducing interaction between the loops is to apply high loop gains.

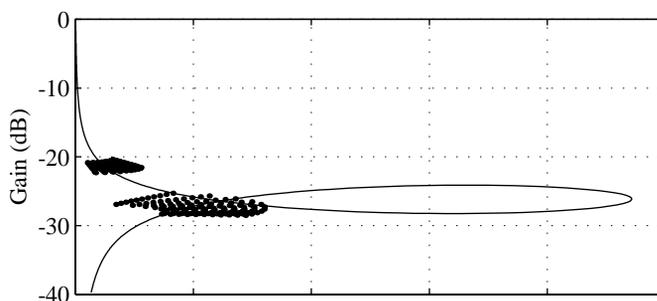
A second method is the characteristic-locus method which treats the loci of a transfer function matrix much as one would treat the locus of a single-input single output system. With this method, a single-input single-output controller is designed for each locus.

In the Nyquist-array method, the first step is to introduce compensator in series with the plant that makes the product of the two diagonally dominant over some range of frequencies. When this is accomplished, the system will behave like a set of SISO loops and the design can proceed.

### 7.6.5 Quantitative Feedback Theory

Quantitative Feedback Theory (QFT) is also considered a Nyquist-like technique in that it is based around a SISO design methodology. However, QFT permits a more precise application of the design specifications and uncertainty models than do the other Nyquist-like techniques. Quantitative feedback theory is based on the idea that the main purpose of feedback is to combat uncertainty, in the form of unpredictable noise, disturbances or plant variations. If uncertainty models exist, as well as a precise specification to the performance in the face of the modeled uncertainty, then we can proceed systematically.

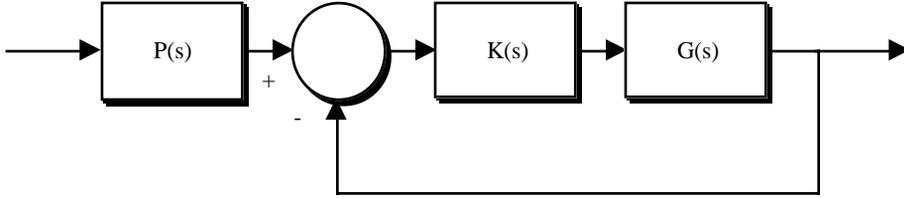
**Figure 5. Uncertainty templates for the single axis plant superimposed over the nominal transfer function.  $1.1 \leq \gamma \leq 1.2$  and  $0.05 \leq \zeta \leq 0.25$ .**



With QFT one starts with uncertainty templates such as the one for the single-axis plant with flexibility shown in Figure 5, and a bound on the magnitude of the sensitivity to external disturbances or the transmissivity of noise. Closed loop time responses, the usual form of system specifications, can be related to frequency domain bounds.

QFT uses a feedback configuration with two degrees of freedom as shown in Figure 6.

**Figure 6. Two degree of freedom configuration**



Given a design specification of the form

$$a(\omega_1) \leq |T(j\omega_1)| \leq b(\omega_1)$$

$$T(s) = G(s)[1 + G(s)K(s)]^{-1}P(s)$$

the compensation  $K(j\omega_1)$  is determined that limits the range of  $T(j\omega_1)/P(j\omega_1)$  (given the  $G(j\omega_1)$  template) such that it would not intersect any pair of M-circles whose values differed by more than  $|b(\omega_1)|/|a(\omega_1)|$ . This is done for each frequency. This generates the compensator. Design trade-offs may be made to simplify the form of the compensator.

The absolute value of  $T(j\omega_1)$  can then be set using P. This design method can be extended to multivariable systems. The simplest multivariable form has only diagonal elements in the compensator and reserves the off-diagonal terms for P since off-diagonal terms within K are particularly sensitive to plant variations.

As is evident from this brief description, QFT permits a systematic approach to feedback design and allows direct trade-offs between compensator complexity and performance in a straightforward fashion.

A major advantage of QFT is that its uncertainty models can handle nonlinearities directly. This makes it an ideal complement to feedback linearization which will be discussed later.

### 7.6.6 LQG Methods

Linear Quadratic Gaussian methods (LQG) are fundamentally state-space methods. The form of an LQG controller is

$$u = -K\hat{x} \tag{7.35}$$

and the estimated state is obtained from

$$\dot{\hat{x}} = (A - BK)\hat{x} + Bu + L(y - C\hat{x}) \tag{7.36}$$

given the state space model

$$\begin{aligned} \dot{x} &= Ax + Bu + \Gamma w \\ y &= Cx + v \end{aligned} \tag{7.37}$$

where  $w$  and  $v$  are Gaussian White noise sources for which

$$\begin{aligned} E\{ww^T\} &= W \\ E\{vv^T\} &= V \\ E\{wv^T\} &= N \end{aligned} \tag{7.38}$$

The plant matrix  $A$  may be of the dynamic system alone, or may include additional dynamics, such as disturbance models, integrators, filters, etc. to add frequency dependence to Equation (7.35). The gain matrix is one that minimizes the function

$$J = \lim_{T \rightarrow \infty} E \left\{ \int_0^T (z^T Q z + u^T R u) dt \right\} \tag{7.39}$$

where

$$z = Mx \tag{7.40}$$

$Q$  and  $R$  can also be functions of frequency.

An LQG controller can exhibit arbitrarily poor robustness properties. Loop-Transfer-Recovery (LTR) was developed to recover the robustness properties of the full state feedback controller (i.e. one with all states measured.) The LTR procedure places some of the estimator eigenvalues at the zeros of the plant, and the remainder are allowed to be arbitrarily fast. This only works if the plant does not have any right-hand-plane zeros near the desired bandwidth and if the plant is square, that is the number of inputs and outputs are equal. If it has unequal numbers of inputs and outputs, LTR can still be applied by squaring the plant. For example, if there are more inputs than outputs, linear combinations of outputs can be added to make the number of ‘outputs’ equal the number of inputs. If this is done, recovery can only be applied at the output.

The LQR methodology does not address uncertainty in the plant model directly. Since uncertainty models are not part of the design procedure, it is necessary to test the system over the range of parameters expected in the plant, including higher-frequency dynamics that were neglected in the design. In addition, choosing the weighting matrices  $Q$  and  $R$  to get desired performance is not easy, although methodologies for doing so have been suggested, making LQR labor intensive and expensive to implement.

### 7.6.7 $H_\infty$ and $\mu$ synthesis

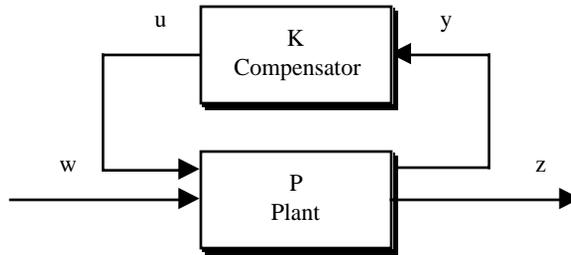
The  $H_\infty$  (pronounced H-infinity where the H stands for Hardy space) optimal controller is defined as follows. The standard plant representation is shown below. If  $P$  is partitioned into four parts such that

$$\begin{bmatrix} z \\ y \end{bmatrix} = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix} \quad (7.41)$$

the input/output relationship is

$$z = [P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}]w \quad (7.42)$$

**Figure 7. Standard Plant Model**



Define

$$F_1(P, K) = [P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}] \quad (7.43)$$

The  $H_\infty$  problem is to minimize

$$\|F_1(P, K)\|_\infty \quad (7.44)$$

over all realizable controllers  $K(s)$  which stabilize the closed loop system. A consistent term for the LQG problem is the  $H_2$  problem since it involves minimizing

$$\|F_1(P, K)\|_2 \quad (7.45)$$

Many control design problems such as sensitivity minimization and robustness to perturbations can be formulated as  $H_\infty$  problems. However, performance robustness in the face of unmodeled perturbations cannot be formulated as an  $H_\infty$  problem. As noted in the above section, the performance robustness is related to the structured singular value of  $Q$ . This relationship is used as a basis for the  $\mu$ -synthesis approach to robust compensator design. The structured singular value cannot be computed directly, and it is necessary to resort to numerical algorithms.

## 7.7 Single-Input Single-Output Control Design

### 7.7.1 Introduction

This chapter describes control system design. The fundamental approach used throughout is the frequency domain approach to loop shaping. This is applied both for continuous controllers and discrete-time control systems. All techniques are applied to single-input single-output control systems in this chapter.

### 7.7.2 Elementary Loop Compensation

#### First Order Compensators

Control compensators can be created in a number of ways. In this section we will use first order building blocks to build control systems. The first block of interest is the first order transfer function

$$K(s) = K \frac{\tau_1 s + 1}{\tau_2 s + 1} \quad (7.46)$$

This is sometimes known as a lead or lag compensator. At low frequencies it has a gain of  $K$  and at high frequencies its gain is constant and is

$$K(s) = K \frac{\tau_1}{\tau_2} \quad (7.47)$$

It is useful to convert Equation (7.46) into its equivalent phase and magnitude as a function of frequency, where  $s = j\omega$ . The first step is to make the denominator pure real by multiplying the numerator and denominator by the complex conjugate of the denominator.

$$K \left( \frac{\tau_1 s + 1}{\tau_2 s + 1} \right) \left( \frac{1 - \tau_2 s}{1 - \tau_2 s} \right) = K \frac{(1 + \tau_1 \tau_2 \omega^2) + j(\tau_1 - \tau_2)\omega}{1 + (\tau_2 \omega)^2} \quad (7.48)$$

The phase as a function of frequency is

$$\theta = \text{atan} \left( \frac{(\tau_1 - \tau_2)\omega}{1 + \tau_1 \tau_2 \omega^2} \right) \quad (7.49)$$

with a peak magnitude of

$$\theta_{peak} = \text{atan}\left(\frac{\tau_1 - \tau_2}{2\sqrt{\tau_1\tau_2}}\right) \quad (7.50)$$

at

$$\omega_{peak} = \frac{1}{\sqrt{\tau_1\tau_2}} \quad (7.51)$$

This compensation element is most useful to add or subtract phase at a desired frequency. Therefore, we want to know what the time constants should be. Solving Equation (7.50) and Equation (7.51) for the time constants gives

$$\tau_2 = \sqrt{\frac{1 + \frac{1}{4}\tan^2\theta_{peak}}{\omega_{peak}^2}} + \frac{\tan\theta_{peak}}{2\omega_{peak}} \quad (7.52)$$

and

$$\tau_1 = \tau_2 + \frac{\tan\theta_{peak}}{\omega_{peak}} \quad (7.53)$$

If the peak phase is negative  $\tau_1 < \tau_2$ , the high-frequency gain will be lower than the low-frequency gain. This configuration is often called a low-pass filter or lag compensator. If the  $\tau_2 < \tau_1$  the peak phase will be positive and the high-frequency gain will be higher than the low-frequency gain. This is known as a high-pass filter or lead compensator.

## Generalized Integrator

A generalized integrator is

$$K(s) = \frac{\tau s + 1}{s + \sigma} \quad (7.54)$$

If  $\sigma$  and  $\tau$  are zero this becomes a pure integrator. If only  $\sigma$  is zero it is an integrator with high-frequency phase compensation. If only  $\tau$  is zero, it is a low pass filter with a low-frequency gain of  $1/\sigma$ .

$\sigma$  is non-zero when infinite gain at  $\omega=0$  is not necessary. Given a specification that the dc gain equal  $k$ , and at frequency  $\omega$  the phase to have the value  $\theta_o$  (between  $-90^\circ$  and  $0^\circ$ ) the constants are

$$\tau = \frac{\frac{\sigma \tan\theta_o}{\omega} + 1}{\sigma - \omega \tan\theta_o} \quad (7.55)$$

$$\sigma = \frac{1}{k} \tag{7.56}$$

### Gain Compensation of a Double Integrator Plant

The simplest model of a spacecraft is the double integrator which represents one axis of a rigid body spacecraft.

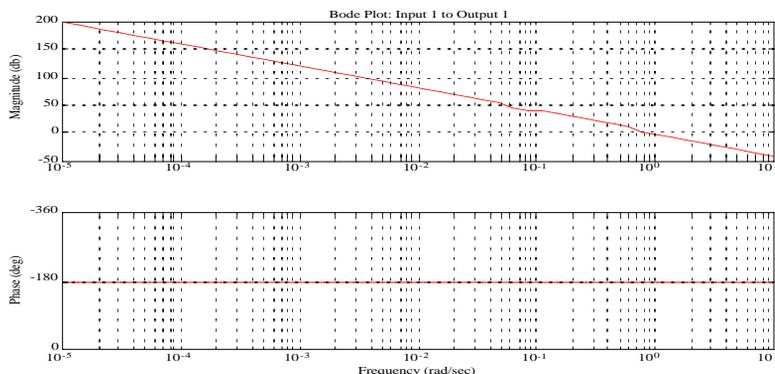
$$\ddot{\theta} = T \tag{7.57}$$

The output is the attitude and the input is in torque. The Laplace Transform of Equation (7.57) is

$$\frac{\theta(s)}{T(s)} = \frac{1}{I s^2} \tag{7.58}$$

The double integrator plant is only truly valid if the spacecraft is a rigid sphere. Nonetheless, the principles used to compensate it are valid for more complex systems. First, look at the Bode plot of the plant, with an inertia of 1, in Figure 8.

**Figure 8. The double integrator plant**



The gain is infinite at  $\omega = 0$ . Therefore, any steady input will cause the angle to go to infinity eventually. The response to inputs at other frequencies drops as  $\omega^2$ . The phase is  $-180^\circ$  at the gain crossover at  $\omega = 1$  rad/sec. As a consequence, if we were to close the loop with negative feedback, the plant, if perturbed, would oscillate with a frequency equal to the crossover frequency. Since the phase is  $-180^\circ$  everywhere, changing the feedback gain would just shift the frequency of the oscillation. This is clear from adding a  $-K\theta$  term to the right-hand-side of Equation (7.57)

$$\ddot{\theta} + \frac{K}{I}\theta = \frac{T}{I} \tag{1}$$

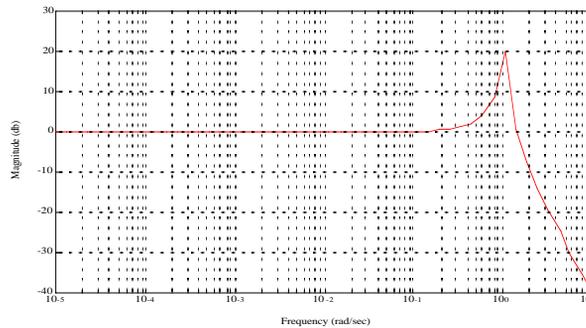
which is the equation for an oscillator with an oscillation frequency of

$$\omega = \sqrt{\frac{K}{I}} \quad (2)$$

The first question is why bother using feedback at all? Assume that the design requirements state that the only input is a 0.01 Nm torque at 0.1 rad/sec and that the peak attitude error must be less than 0.001 rad. If the inertia is greater than 1000 kg-m<sup>2</sup>, the open-loop plant meets specifications and no further work is necessary. If the inertia were too small, we could meet the specification by adding inertia.

The primary reason for using feedback is to deal with uncertainty, or errors, in the models of the plant, the disturbances, and in the specifications. The latter should not be easily dismissed. It is not unusual for specifications to change after the design is done. In this simple example, that might mean a tighter pointing requirement after the spacecraft had been built, and increasing its inertia no longer a viable option. To see how feedback reduces sensitivity to uncertainty, add unity feedback (K=1) to the above spacecraft and look at the magnitude plot of the closed loop system.

Figure 9. Torque transmission plot



From  $\omega=0$  to  $\omega = 0.2$  the angle is the same for all frequency inputs. Thus for disturbances at frequencies lower than the resonant frequency, we need not be concerned about their frequency! This is in sharp contrast to the open loop case where the angle went up as the frequency went down. Thus if the specification were that the response to any single disturbance between 0.001 and 0.4 rad/sec was to be less than 0.01 rad, it would be much easier to meet the specification with feedback. At frequencies greater than 2 rad/sec, the transfer function between angle and torque is nearly the same as the open loop case. This can be seen from the closed loop transfer function

$$\frac{\theta(s)}{T(s)} = \frac{1}{Is^2 + K} \quad (7.59)$$

where the first term in the denominator dominates at large frequencies. The range of frequencies for which the control is effective, in this case from 0 to 2 rad/sec, is called the bandwidth of the control system.

The system is now also much less sensitive to variations in inertia. For all frequencies for which  $K \gg Is^2$ , the transfer function is

$$\frac{\theta(s)}{T(s)} = \frac{1}{K} \tag{7.60}$$

Thus, the angle response to a disturbance torque is independent of the inertia, and depends only on K, the feedback gain.

Unfortunately, in this design we have greatly increased our sensitivity to disturbances (or noise) near 1 rad/sec due to our control system. Table 2, on page 95 summarizes the performance of the closed loop system.

**TABLE 2. Control System Performance Summary**

Frequency Range	Torque Transmission	Characteristic
$0.01 \leq \omega \leq 0.2$	$\frac{\theta(\omega)}{T(\omega)} = \frac{1}{K}$	Closed Loop
$0.2 < \omega < 2$	$\frac{\theta(\omega)}{T(\omega)} = \left  \frac{1}{K - I\omega^2} \right $	Closed Loop
$\omega \geq 2$	$\frac{\dot{\theta}(\omega)}{T(\omega)} = \frac{1}{I\omega^2}$	Open Loop

In summary, this control system has minimal effect on frequencies greater than the bandwidth, causes the torque to angle transmission to be dominated by the feedback gain at frequencies lower than a decade below the bandwidth, and causes the response to be very sensitive to disturbances or noise at frequencies near the resonant frequency of the system.

In this system we applied a gain K as the feedback elements. This implies that any input, no matter how high its frequency, is amplified by a factor K. In practice, it is not possible to design a controller with flat response to infinity, but the bandwidth of the sensors and actuators usually exceeds that of the control system, which is usually sufficient.

### More Complex Compensation of a Double Integrator Plant

Table 3, on page 95 gives a more interesting specification for our control system in terms of both noise and disturbance.

**TABLE 3. Control System Specification**

Specification	Frequency Range	Input at any one Frequency	Maximum Angle	Gain
Noise	$\omega \geq 1.0$	0.01 rad	0.0001 rad	-40 db

**TABLE 3. Control System Specification**

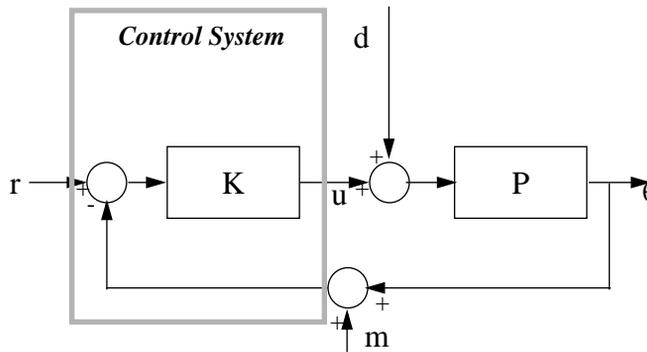
Specification	Frequency Range	Input at any one Frequency	Maximum Angle	Gain
Disturbance	$\omega = 0$	0.0001 Nm	0.0000 rad	$\infty$
Disturbance	$0 < \omega \leq 0.001$	0.0001 Nm	0.0001 rad	+120 db

Figure 1 gives the block diagram for this system. This is a unity feedback configuration without any disturbance feedforward and no prefiltering. The reference is assumed to be 0, and we are only interested in the control system as a regulator.

The relationship between the output and the noise and disturbance inputs is

$$\theta = \frac{P}{1+PK}d - \frac{PK}{1+PK}m \tag{7.61}$$

**Figure 10. Control system block diagram**



Where  $PK$  is large, the equation becomes

$$\theta = \frac{1}{K}d - m \tag{7.62}$$

Thus, noise is passed into the output with a gain of 1 and the disturbance is attenuated by a factor of  $K$ . Where  $PK$  is small it becomes

$$\theta = Pd \tag{7.63}$$

which is the open-loop equation.

Equation (7.62) is a fundamental problem with feedback control. If one applies feedback to attenuate disturbances at a given frequency, any noise at that frequency will be passed into the output.

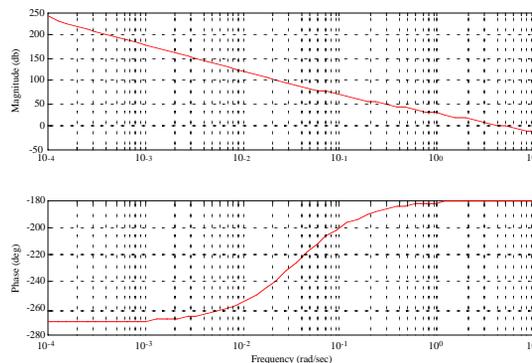
It is also worth noting that the product of PK is important, not just K alone. Thus, the roll-off at high frequencies exhibited by all physical systems means that at high frequencies, PK will go to zero.

The gain specifications are found by substituting in the values for the ratios of the output to either the disturbance or noise and computing the product PK. In the case of the disturbance, we must also know P at the frequency of interest. The specifications require -100 db of roll-off in three decades. For stability purposes, the roll-off must be less than  $1/s^2$ , or 40 db/decade at the cross-over since for a minimum phase system a  $1/s^2$  slope; this means the phase shift is  $-180^\circ$ .

The design of this control system starts by adding an integrator to meet the steady-state offset requirement. This gives the system a gain slope of -60 db/decade at low frequencies. This must transition to greater than -40 db/decade at the gain cross-over. The numerator of the integrator

At 0.01 rad/sec, the integrator only contributes  $-10^\circ$  of phase shift to the system.

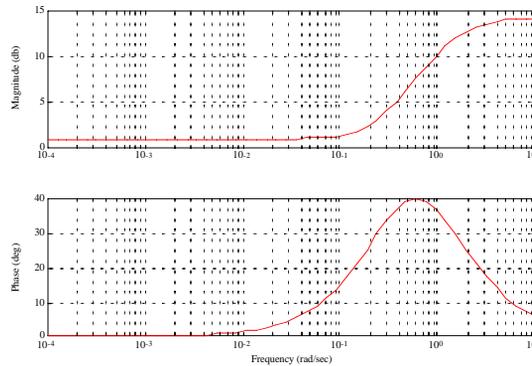
**Figure 11. Plant plus integrator**



The system remains unstable because at the crossover (to the right of the graph), the phase will be slightly lower than  $-180^\circ$  due to the pole at the origin introduced by the integrator. Our specification calls for a gain of xxx db at 0.001 rad/sec and a gain of -40db at 1 rad/sec. The difference between those gains is 150 db. However, the plant is unstable.

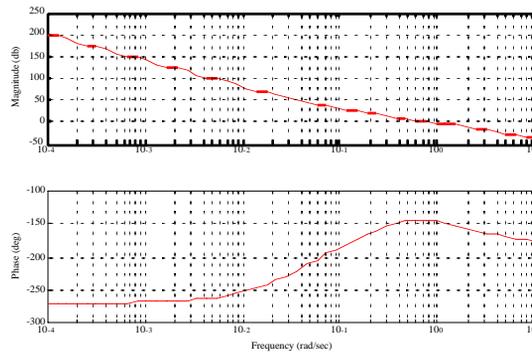
This system is still unstable.

**Figure 12. Lead compensator**



To correct this, we would like to adjust the phase at the crossover without changing the gain much. This can be done with lead compensation. We need to add about  $+60^\circ$  of phase shift at 0.4 rad/sec.

**Figure 13. After Gain Adjustment and Addition of Lead Compensation**




---

## 7.8 Digital Control

### 7.8.1 Introduction

Spacecraft control systems are usually implemented on digital computers. This chapter discusses digital control system design techniques.

## 7.8.2 Modified Continuous Design

### Introduction

Most control engineers are first taught the control of continuous systems in which the plant to be controlled is described by differential equations

$$\dot{x} = f(x, u, t) \quad (7.64)$$

and the measurements are described by the nonlinear relationship

$$y = h(x, u, t) \quad (7.65)$$

The linearized version of these equations are

$$\dot{x} = Ax + Bu \quad (7.66)$$

and

$$y = Cx + Du \quad (7.67)$$

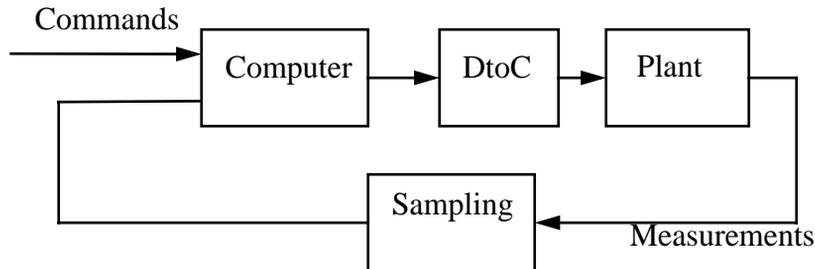
Corresponding to these equations are the frequency domain representation of these equations

$$y(s) = (C(sE - A)^{-1}B + D)u(s) \quad (7.68)$$

Figure 1 shows a block diagram of a control system and plant. The only discrete part of the system is the computer and its software, the remainder of the system is continuous. Measurements are periodically entered into the computer by means of a sampling mechanism and numbers representing desired actuation are converted to continuous operations by means of the discrete to continuous conversion block.

The discrete to continuous conversion block may take several forms. The most common is the sample and hold. During every control period, an external register is set to the number representing the desired actuation. This value is maintained for the entire control period. For example, a register that is an input to a digital to analog converter that determines the input voltage to a reaction wheel might be set to the desired value. A single bit that drives a thruster valve relay might be set and held high for the duration of the command period.

Figure 14. Computer controlled system



Some magnetic torquers and most thrusters are examples of actuators that have only two states; these are sometimes driven by pulsewidth modulation. The output of the control system, which is a torque level for that output, is converted into a pulsewidth. If the desired torque level is  $u$ , and the control period is  $T$ , then the pulsewidth is

$$\tau = \frac{uT}{u_c} \quad (7.69)$$

where  $u_c$  is the torque available from the actuator. Thus, the average torque over the control period equals  $u$ . The pulsewidth will be limited by the resolution of the timing mechanism, and the speed of response of the actuator.

Since the output sampling rate does not need to be fixed, it is also possible to vary the actuation period on the output. This is known as pulsewidth-pulsefrequency modulation.

The modified frequency continuous design approach represents the conversion of the measurements to digital form, and the conversion of the commands to continuous from as a continuous functions. This permits the use of frequency domain design techniques and the nearly direct conversion of the resulting algorithms into discrete form, with minimal effort. Since many modern control design techniques are continuous time techniques, this permits a wider variety of design approaches.

In this section we will look at the key elements that give a continuous time system the characteristics of a discrete time system. These are the sampler, the delay, and the discrete to continuous conversion function. For the latter, we will look at the sample and hold and pulsewidth modulation.

## The Sampler

The sampler is represented in the continuous time domain as a gain of  $1/T$ , where  $T$  is the sample period.

## The Delay

All digital control systems have true delays, as opposed to lags. A delay means that a signal, regardless of frequency, is always the delay time late. The Laplace Transform of a delay is

$$e^{-s\Delta} \tag{7.70}$$

where  $\Delta$  is the delay in seconds and  $s$  is  $j\omega$ . This is a nonminimum phase function. Nonminimum phase means that the phase angle is greater than the minimum possible for a function with the same magnitude. Equation (7.70) can be expanded as

$$e^{-s\Delta} = \cos \omega\Delta - j\sin \omega\Delta \tag{7.71}$$

The magnitude of this function is 1 and the phase is  $-\omega\Delta$ . Thus, the phase decreases linearly with frequency. This is also known as an all-pass function. Clearly, the minimum phase for a constant magnitude function is 0. Therefore, this function is nonminimum phase. All-pass functions are always nonminimum phase.

Any uncertainty in a delay can wreak havoc on control approaches that require precise phase knowledge at high frequencies since the phase error will be proportional to the frequency. For example, if one has a  $\pm 20\%$  delay uncertainty, in a 0.3 second delay, then the phase uncertainty at 10 rad/sec, for example, will be  $\pm 34^\circ$ .

Equation (7.70) is difficult to use directly in s-plane analysis. Instead, series expansions are employed to represent it. The Taylor series expansion for an exponential is

$$e^{sa} = 1 + sa + \frac{(sa)^2}{2} + \frac{(sa)^3}{6} + \frac{(sa)^4}{24} + \frac{(sa)^5}{120} + \dots + \frac{(sa)^n}{n!} \tag{7.72}$$

The transfer function of the exponential should be proper, that is to say the denominator order must be greater than or equal to the order of the numerator. The general approach to expanding Equation (7.70) is to write it as fraction

$$e^{-s\Delta} = \frac{e^{-sf\Delta}}{e^{s(1-f)\Delta}} \tag{7.73}$$

where  $f$  is less than 1. If  $f$  is zero, for example, the first order expansion is

$$e^{-s\Delta} = \frac{1}{1 + s\Delta} \tag{7.74}$$

This function will represent the exponential at very low frequencies, but does not capture its minimum phase nature. A widely used representation is with  $f = 1/2$ . To second order this is

$$e^{-s\Delta} = \frac{1 - \frac{s\Delta}{2} + \frac{(s\Delta)^2}{8}}{1 + \frac{s\Delta}{2} + \frac{(s\Delta)^2}{8}} \quad (7.75)$$

Notice that this function has a pair of zeros in the right-half plane and is all-pass. If  $f$  and  $1-f$  are not equal the expansions in the numerator and denominator should be carried out so that their relative error is equal. For example, if  $f = 1/3$  the appropriate function is

$$e^{-s\Delta} = \frac{1 - \frac{s\Delta}{3}}{1 + \frac{2s\Delta}{3} + \frac{(s\Delta)^2}{6}} \quad (7.76)$$

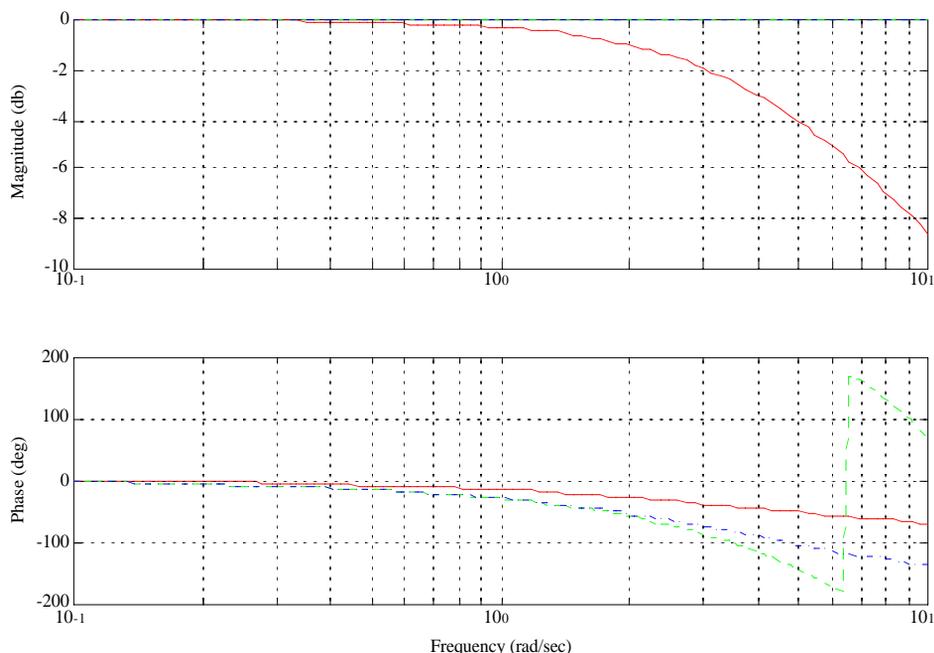
and if  $f = 2/5$

$$e^{-s\Delta} = \frac{1 - \frac{2s\Delta}{5} + \frac{(s\Delta)^2}{20}}{1 + \frac{3s\Delta}{5} + \frac{3(s\Delta)^2}{20} + \frac{(s\Delta)^3}{60}} \quad (7.77)$$

Equation (7.75), Equation (7.76) and Equation (7.77) are known as Padé Approximants.

Figure 15 shows Equation (7.74) (the solid line) and Equation (7.75) (the dot-dash) expanded only to first order with  $\Delta = 0$  and compared with the exponential (the dashed line). Equation (7.74) rolls off at high frequency and does not match the phase well above 0.2 rad/sec. The nonminimum phase transfer function matches the magnitude exactly and the phase to about 2 rad/sec. For frequencies less than 0.1/D, the delay has minimal effect and can be ignored.

Figure 15. Bode plots of the delay and two series representations



Assume that we are designing a rate control system whose plant is a single integrator with a delay. Represent the delay by Equation (7.75) expanded only to second order. A gain is a perfectly adequate controller for a single integrator. The plant is

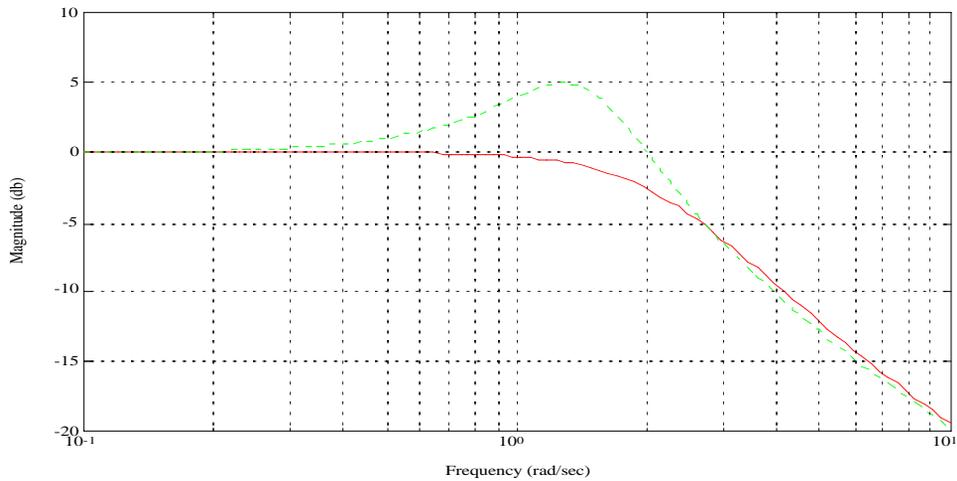
$$P(s) = \frac{1 - \frac{s\Delta}{2}}{s\left(1 + \frac{s\Delta}{2}\right)} \quad (7.78)$$

The controller is  $K$  and the closed loop transfer function is

$$\frac{y}{r} = \frac{K\left(1 - \frac{s\Delta}{2}\right)}{\frac{\Delta}{2}s^2 + \left(1 - K\frac{\Delta}{2}\right)s + K} \quad (7.79)$$

If  $\Delta = 0$  this reduces to the standard equation for rate feedback. The system will become unstable if  $K$  exceeds  $2/\Delta$ . Thus, the delay limits the system gain, a natural consequence of a right-half-plane zero. Figure 16 shows the Bode magnitude plot with  $K = 1$  and  $\Delta = 1$ .

**Figure 16. Bode Magnitude Plot of the Rate Control Loop with a Delays of 0.5 and 1 second**



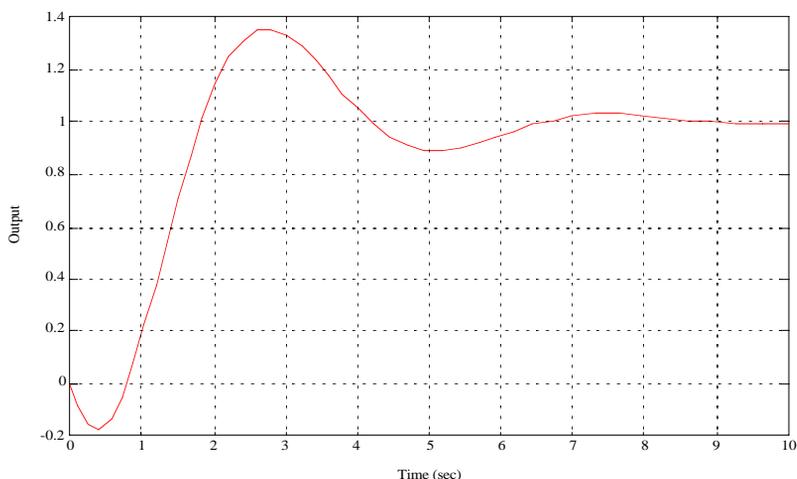
The solid line shows the magnitude response with  $\Delta = 0.5$  and the dashed line shows it with  $\Delta = 1.0$ . The damping decreases as the delay increases. The shape of the magnitude curve is also somewhat different than what you would expect for a second order system with the same damping ratio. At high frequencies it appears to be concave instead of convex.

The zero has an interesting effect on the transient response. At low frequencies, the transfer function is 1. At high frequencies, the transfer function is

$$\frac{y}{r} = -\frac{K}{s} \quad (7.80)$$

The sign is reversed! Therefore, any input (such as a step) with sufficient high frequency content will cause the output to go in the opposite direction of the input at first. This is illustrated below.

**Figure 17. Response to a Unit Step of the Integrator with a Time Delay of 1 second.**



This effect can be reduced by shortening the delay or low-pass filtering the command input.

### The Zero Order Hold

The zero-order hold is the standard method of driving analog actuators with signals from computers. The zero order hold is given by

$$u(t) = u_{-1}(t) - u_{-1}(t - T) \tag{7.81}$$

This is just the difference of a step and a delayed step. The Laplace Transform of Equation (7.81) is

$$u(s) = \frac{1 - e^{-sT}}{s} \tag{7.82}$$

When multiplied by the gain of the sampler,  $1/T$ , it becomes

$$u(s) = \frac{1 - e^{-sT}}{sT} \tag{7.83}$$

Equation (7.83), when included in the continuous-time system, represents the effects of sampling on input and the zero-order hold on output. This allows us to include the effects of the digital system while designing in the continuous time domain. Equation (7.83) is most easily handled using its series expansion. This is done by substituting Equation (7.76) or Equation (7.77) for the delay in the numerator. The result is

$$u(s) = \frac{1 - \frac{sT}{6}}{1 + \frac{sT}{3}} \quad (7.84)$$

for Equation (7.76) and

$$u(s) = \frac{1 - \frac{sT}{10} + \frac{(sT)^2}{60}}{1 + \frac{2sT}{5} + \frac{(sT)^2}{20}} \quad (7.85)$$

for Equation (7.77). Substituting Equation (7.75) would not have provided the right-half-plane zeros that give the zero-order hold its nonminimum phase character.

### 7.8.3 Pulsewidth Modulation

Pulsewidth modulation involves varying the pulsewidth from 0 to  $T$ . While this cannot be directly represented, as can the zero-order hold, the effect can be analyzed by assuming that the pulsewidth is fixed at  $\tau$ , a fraction of  $T$ . The pulsewidth modulator is

$$u(t) = u_{-1}(t - T + \tau) - u_{-1}(t - T) \quad (7.1)$$

The Laplace transform (including the sampler gain) is

$$u(s) = \frac{e^{-s(T-\tau)} - e^{-sT}}{sT} \quad (7.2)$$

or, after some manipulation

$$u(s) = \left( \frac{e^{s\tau} - 1}{e^{sT} - 1} \right) \left( \frac{1 - e^{-sT}}{sT} \right) \quad (7.3)$$

The first term in parenthesis is an additional nonminimum phase function. Using

$$e^{sT} - 1 = -\frac{1 + \frac{sT}{10} + \frac{(sT)^2}{60}}{1 - \frac{2sT}{5} + \frac{(sT)^2}{20}} \quad (7.4)$$

and substituting gives

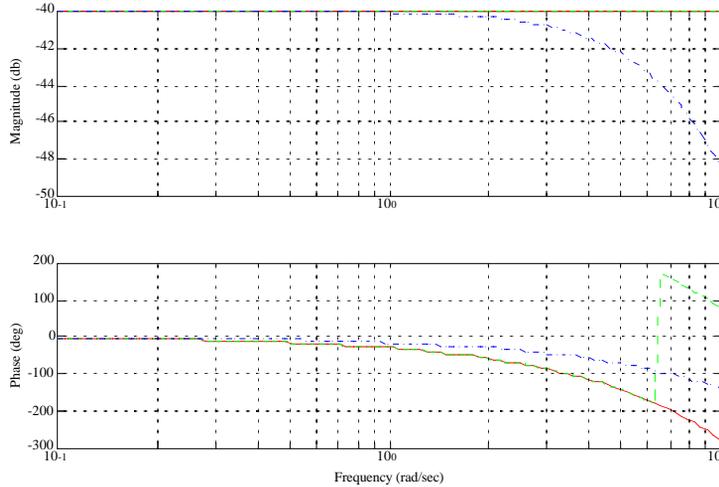
$$\frac{e^{s\tau} - 1}{e^{sT} - 1} = \frac{\tau}{T} \left( \frac{1 + \frac{s\tau}{10} + \frac{s\tau^2}{60}}{1 - \frac{2s\tau}{5} + \frac{s\tau^2}{20}} \right) \left( \frac{1 - \frac{2sT}{5} + \frac{sT^2}{20}}{1 + \frac{2sT}{10} + \frac{sT^2}{60}} \right) \quad (7.5)$$

The finite pulsewidth introduces a very high-frequency zero-pair and an unstable pole pair. For frequencies below the sampling frequency, determined by T, this transfer function does not appreciably affect the dynamics. Therefore, the transfer function for the pulsewidth modulator is

$$u(s) = \frac{\tau}{T} \left( \frac{1 - \frac{2sT}{5} + \frac{sT^2}{20}}{1 + \frac{2sT}{10} + \frac{sT^2}{60}} \right) \left( \frac{1 - \frac{sT}{10} + \frac{(sT)^2}{60}}{1 + \frac{2sT}{5} + \frac{(sT)^2}{20}} \right) \quad (7.6)$$

shows the Bode plot for the exact and series expansion, as well as the zero-order hold, for T = 0.5 and τ = 0.005

**Figure 18. Comparison of Pulsewidth Modulator and Zero Order Hold**



The solid line is the series expansion for the pulsewidth modulator, the dashed line is the exact representation, and the dot-dashed line is the zero-order hold. The gain on the zero-order hold was made identical to that of the pulsewidth modulator for comparison purposes only.

The pulsewidth modulator has greater phase shift than the zero-order hold and no high-frequency roll-off. It will tend to interact more with high-frequency dynamics.

## 7.9 Continuous to Discrete Transformations

### 7.9.1 The Difference Equation

Once we have a continuous design, it is necessary to convert it into a set of difference equations for implementation in the computer. The controller will be of the form

$$x_i = \sum_{j=i-1}^{i-n} a_j x_j + \sum_{j=i}^{i-m} b_j u_j \quad (7.1)$$

where  $u_j$  is the  $j$ th input and  $x_j$  is the  $j$ th output. The time between inputs and outputs is the sample period  $T$ . If we define  $z$  as a one sample period delay, we can transform the difference equation into

$$x(z) = \sum_{j=1}^n a_j z^j x(z) + \sum_{j=0}^m b_j z^j u(z) \quad (7.2)$$

For example, the difference equation for a Trapezoidal rule integrator is

$$x_i = x_{i-1} + \frac{T}{2}(u_i + u_{i-1}) \quad (7.3)$$

has  $n = 1$  and  $m = 1$  and transformed into the  $z$  domain is

$$\frac{x(z)}{u(z)} = \frac{T(z+1)}{2(z-1)} \quad (7.4)$$

### 7.9.2 Transforming From the S Plane to the Z Plane

The delay  $z$  in the  $s$  plane is

$$z = e^{sT} \quad (7.1)$$

and

$$s = \frac{\ln(z)}{T} \quad (7.2)$$

Equation (7.1) and Equation (7.2) allow us to move between the  $s$  and  $z$  planes. Given a continuous time transfer function, the equivalent  $s$  domain function can be found by substituting Equation (7.2) for  $s$ . Since the natural logarithm does not easily convert to difference equations, we expand it in a series.

$$s = \frac{2}{T} \left\{ \left( \frac{z-1}{z+1} \right) + \frac{1}{3} \left( \frac{z-1}{z+1} \right)^3 + \dots + \frac{1}{n} \left( \frac{z-1}{z+1} \right)^n \right\} \quad (7.3)$$

where  $n$  is odd. The first order expansion is known as the Tustin Transformation and is

$$s = \frac{2}{T} \left( \frac{z-1}{z+1} \right) \quad (7.4)$$

The second order expansion is

$$s = \frac{8}{3T} \frac{z^3 - 1}{(z+1)^3} \quad (7.5)$$

Since  $s$  is the derivative operator, one could also replace it with a first difference

$$s = \frac{1}{T}(z-1) \quad (7.6)$$

or a second difference

$$s = \frac{1}{2T}(3z^2 - 4z + 1) \quad (7.7)$$

### 7.9.3 Transformation of a Differentiator

A differentiator with roll-off at high frequencies is

$$\frac{s}{\tau s + 1} \quad (7.1)$$

The response to a ramp at  $t = \infty$  can be found by the final value theorem

$$\lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} s \left( \frac{s}{\tau s + 1} \right) \frac{1}{s^2} = 1 \quad (7.2)$$

This can be implemented with a first difference. The result is

$$\frac{1}{\tau} \left( \frac{z-1}{z - \left(1 - \frac{T}{\tau}\right)} \right) \quad (7.3)$$

where  $\tau > T$ . The final value theorem can be applied to this function

$$\lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} s \left( \frac{1}{\tau} \left( \frac{e^{sT} - 1}{e^{sT} - \left(1 - \frac{T}{\tau}\right)} \right) \right) \frac{1}{s^2} = \frac{1}{T} \frac{e^{sT} - 1}{s} = \frac{sT}{sT} = 1 \quad (7.4)$$

The matched pole zero method is also easy to apply. Transforming the zero and pole gives

$$K \frac{z - 1}{z - e^{T/\tau}} \quad (7.5)$$

The gain is found using the final value theorem

$$K \lim_{s \rightarrow 0} \left( \frac{1}{e^{sT} - e^{T/\tau}} \right) \frac{e^{sT} - 1}{s} = 1 \quad (7.6)$$

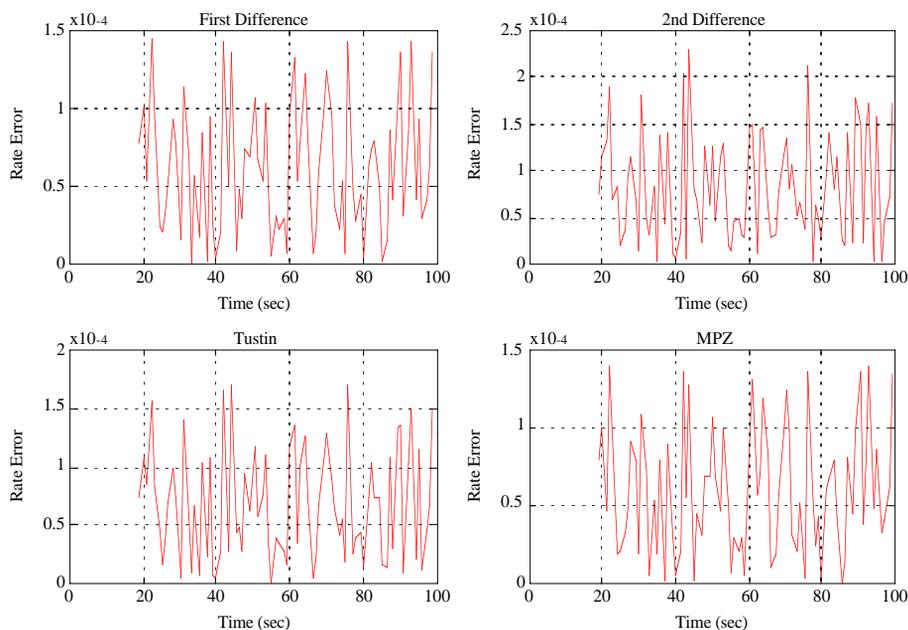
or

$$K = \frac{1 - e^{T/\tau}}{T} \quad (7.7)$$

These results, along with the Tustin and 2nd difference are summarized in Table 4, on page 112.

All four rate estimator errors are shown below.

**Figure 19. Comparison of the Four Rate Estimators**



### 7.9.4 State Estimator

An alternative is to use a state estimator. The discrete time state equations for a double integrator plant are

$$\begin{bmatrix} x_1 \\ x_x \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_x \end{bmatrix}_k + \begin{bmatrix} T^2 \\ T \end{bmatrix} u + \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} \left( y - \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_x \end{bmatrix}_k \right) \quad (7.1)$$

The first state is the estimated angle and the second is the estimated rate.  $y$  is the measured angle. The transfer function from the measurement to the rate is

$$\frac{x_x(z)}{y(z)} = k_1 \frac{z-1}{z^2 + (k_1-2)z + Tk_2 - k_1 + 1} \quad (7.2)$$

which has a second order denominator. The gains can be chosen in many ways. One is just to map desired s-plane

**TABLE 4. Z-Plane Representation of a filtered Differentiator**

Form	Transfer Function
First Difference	$\frac{1}{\tau} \left( \frac{z-1}{z - \left(1 - \frac{T}{\tau}\right)} \right)$
Second Difference	$\frac{1}{\tau} \left( \frac{3z^2 - 4z + 1}{3z^2 - 4z + 1 + \frac{2T}{\tau}} \right)$
Tustin	$\frac{2/\tau}{1 + \frac{T}{2\tau}} \frac{z-1}{z - \left( \frac{1 - \frac{T}{2\tau}}{1 + \frac{T}{2\tau}} \right)}$
Matched Pole Zero	$\frac{1 - e^{-T/\tau}}{T} \left( \frac{z-1}{z - e^{-T/\tau}} \right)$
State Estimator	$k_1 \frac{z-1}{z^2 + (k_1 - 2)z + Tk_2 - k_1 + 1}$

poles into the z-plane using the relationship

$$z = e^{sT} \tag{7.3}$$

where

$$s = -\zeta\omega \pm j\omega\sqrt{1 - \zeta^2} \tag{7.4}$$

This gives

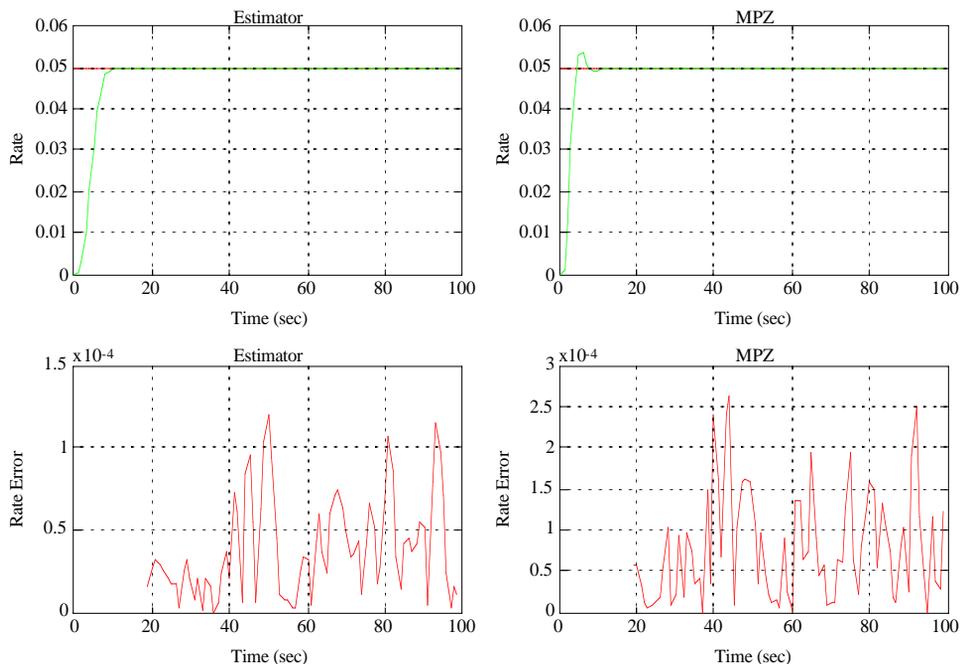
$$\begin{aligned} \theta &= \omega T \sqrt{1 - \zeta^2} \\ k_1 &= 2(e^{-\zeta\omega T} \cos\theta + 1) \\ k_2 &= \frac{e^{-\zeta\omega T} (1 + 2\cos\theta) + 1}{T} \end{aligned} \tag{7.5}$$

Note that

$$j^\theta = \cos\theta + j\sin\theta \tag{7.6}$$

Figure 20 shows a comparison of the state estimator and matched pole zero response.

**Figure 20. Comparison of the Matched Pole Zero with the Estimator**



The s-plane parameters for the estimator are  $\zeta = 0.18$  and  $\omega = 2.9$ .

The Tustin and First difference are just low-order expansions of the Matched Pole Zero. The second difference has zeros at 1 and 1/3 and poles at

$$\frac{4 \pm \sqrt{4 - \frac{24T}{\tau}}}{6} \tag{7.7}$$

## 7.10 Flexible Spacecraft Control

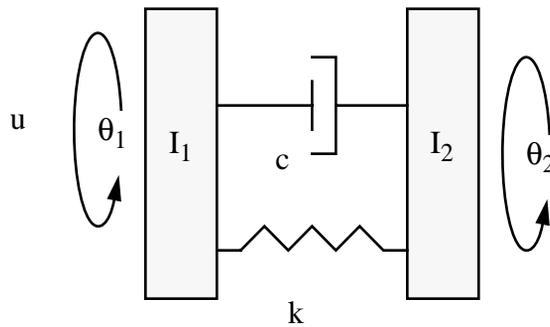
### 7.10.1 Introduction

Most spacecraft cannot be considered as rigid bodies when their control systems are being designed. Generally, a flexible body is one in which the structure deforms and those deformations are much smaller than the rigid body motion. However, many other types of dynamics can be included in this category. Attached booms, Astromasts, solar arrays or fluid slosh all serve to make the spacecraft non-rigid. For many types of solar arrays, the first few bending modes are due to rotation at the hinges, rather than actual deformations of the solar panels.

### 7.10.2 Two Coupled Inertias

The simplest possible flexible body problem is that of two coupled inertias, illustrated in

**Figure 21. Two inertias coupled by a spring and a damper.**



The inertias are connected by a torsional spring and damper. A control torque can be applied to body 1. Disturbances may be applied to either body. The linear differential equations for the coupled set are

$$\begin{aligned} I_1 \ddot{\theta}_1 + c(\dot{\theta}_1 - \dot{\theta}_2) + k(\theta_1 - \theta_2) &= u + d_1 \\ I_2 \ddot{\theta}_2 + c(\dot{\theta}_2 - \dot{\theta}_1) + k(\theta_2 - \theta_1) &= d_2 \end{aligned} \quad (7.8)$$

Taking the Laplace Transform of the equations gives

$$\begin{bmatrix} I_1 s^2 + cs + k & -cs - k \\ -cs - k & I_2 s^2 + cs + k \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} u + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \quad (7.9)$$

and solving for  $\theta_1$  and  $\theta_2$  gives

$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \frac{\begin{bmatrix} I_2 s^2 + cs + k & cs + k \\ cs + k & I_1 s^2 + cs + k \end{bmatrix}}{I_1 I_2 s^2 \left( s^2 + c \frac{\gamma}{I_2} s + k \frac{\gamma}{I_2} \right)} \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \quad (7.10)$$

$\gamma$  is the modal coupling factor and is

$$\gamma = \frac{I_1 + I_2}{I_1} \quad (7.11)$$

$\gamma$  can range from 1 to  $\infty$ .

Two cases are of interest. The first is the collocated sensor and actuator case. The transfer function from  $u$  to  $\theta_1$  is

$$\frac{\theta_1}{u} = \frac{s^2 + \frac{c}{I_2} s + \frac{k}{I_2}}{I_1 s^2 \left( s^2 + c \frac{\gamma}{I_2} s + k \frac{\gamma}{I_2} \right)} \quad (7.12)$$

When  $I_1 \gg I_2$  the modal coupling factor is 1 and the quadratic factor in the denominator equals the quadratic factor in the numerator and they drop out. The resulting transfer function is for a rigid body with inertia  $I_1$ .

At frequencies much lower than the roots of the denominator quadratic factor, the first two terms in the quadratic factor are negligible and the transfer function reduces to a rigid body with inertia  $I_1 + I_2$ .

At frequencies much higher than the complex poles, the transfer function reduces to a rigid body with inertia  $I_1$ .

Since  $\gamma$  is always greater than 1, the pole pair determined by the quadratic term in the denominator, will always be at a higher frequency than the zeros. Therefore, the phase which starts at  $-180^\circ$  due to the double pole at the origin, will flip to  $0^\circ$  and then return to  $-180^\circ$ . No matter how many flexible body modes there are, if the sensor and actuator are collocated, the phase is never less than  $-180^\circ$ .

The damping ratio for the fixed base mode is

$$\zeta = \frac{c}{2\sqrt{kI_2}} \quad (7.13)$$

and the damping ratio when it is measured attached to the spacecraft is

$$\zeta = \frac{c}{2\sqrt{\gamma k I_2}} \quad (7.14)$$

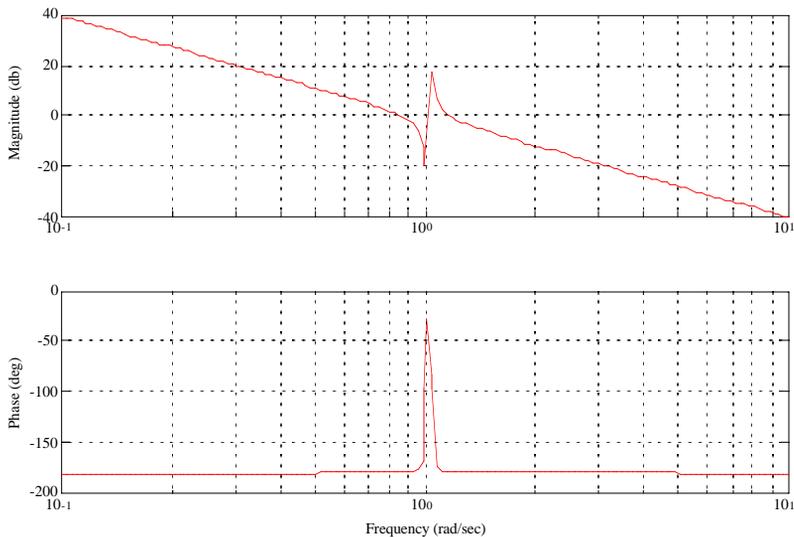
Since  $\gamma$  is always greater than one, the damping ratio will always be less than that measured on the ground.

The poles and zeros of the quadratic terms are always complex since flexible modes (if they are of concern at all) are lightly damped.

The numerator zeros are the fixed base frequencies of the flexible part of the system (represented by  $I_2$  and its spring and damper.) They are the frequencies that would be measured if the flexible part were cantilevered to a fixed base. It is easy to go from the fixed base vibrational frequencies to the frequencies of the spacecraft if one knows the inertias.

The Bode plot for this transfer function for  $I_1 = 1$ ,  $I_2 = 0.1$ , fixed base  $\zeta = 0.005$  and fixed base frequency equal to 1 rad/sec is illustrated in Figure 22.

**Figure 22. Bode plot for the collocated sensor actuator transfer function.**



Note that this system has multiple crossovers. This is a characteristic of flexible body systems that is not seen in many other control problems.

Suppose that instead of measuring  $\theta_1$ , we measure  $\theta_2$ .  $\theta_2$  is separated from the actuator by the flexible coupling. The transfer function is

$$\frac{\theta_2}{u} = \frac{\frac{c}{I_2}s + \frac{k}{I_2}}{I_1s^2 \left( s^2 + c\frac{\gamma}{I_2}s + k\frac{\gamma}{I_2} \right)} \quad (7.15)$$

The numerator is a single real zero at frequency  $k/c$ . In terms of damping ratio this is

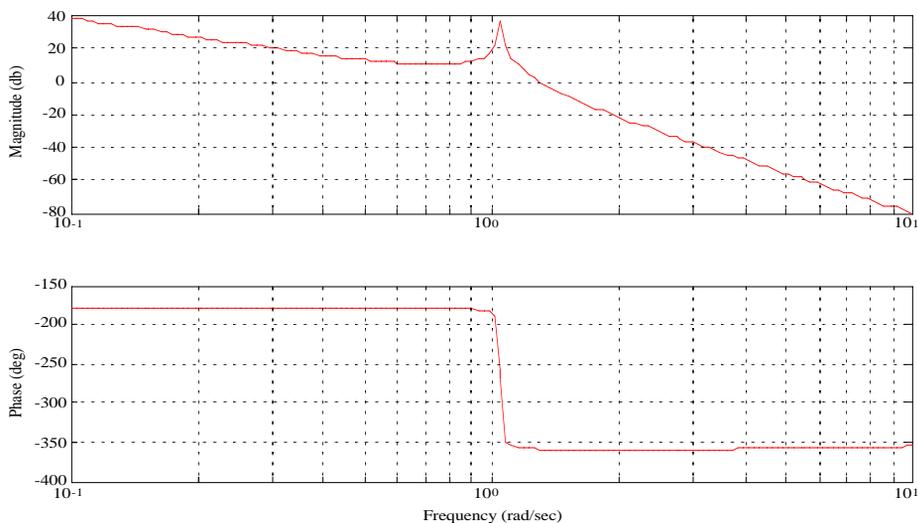
$$s = \frac{\omega}{2\zeta} \quad (7.16)$$

If  $\zeta$  is small, this is much higher than the denominator frequency. Therefore, in the noncollocated system, the phase will decrease to  $-360^\circ$  after the pole pair and will not return to  $-180^\circ$  until a much higher frequency.

Note that the system behavior at very high and very low frequencies (relative to the complex) pole is not changed. It is only the behavior at intermediate frequencies that is different.

The Bode plot for the noncollocated transfer function is illustrated in Figure 23. The transfer functions for the different frequency domains for both collocated and noncollocated transfer functions are

**Figure 23. Bode plot for the non-collocated sensor and actuator transfer function.**



summarized in Table 5, on page 118.

**TABLE 5. Transfer function summary**

Frequency	Collocated	Noncollocated
Low	$\frac{1}{(I_1 + I_2)s^2}$	$\frac{1}{(I_1 + I_2)s^2}$
Intermediate	$\frac{s^2 + \frac{c}{I_2}s + \frac{k}{I_2}}{I_1s^2\left(s^2 + c\frac{\gamma}{I_2}s + k\frac{\gamma}{I_2}\right)}$	$\frac{1}{I_1s^2\left(s^2 + c\frac{\gamma}{I_2}s + k\frac{\gamma}{I_2}\right)}$
High	$\frac{1}{I_1s^2}$	$\frac{1}{I_1s^2}$

At low frequencies both bodies move together as if they were rigid. At intermediate frequencies the motion is governed by the flexible body modes. At very high frequencies only the core body moves.

### 7.10.3 Lead Compensation of the Minimum Phase System

The minimum phase system can be compensated with a lead compensator of the form

$$C(s) = K \frac{\tau_1 s + 1}{\tau_2 s + 1} \tag{7.17}$$

where  $\tau_1 > \tau_2$ . That is, the zero is at a lower frequency than the pole. This can also be thought of as a proportional derivative controller cascaded with a low-pass filter. The advantage of using the pole/zero combination in Equation (7.17) is that the phase lag effects of the pole are automatically accounted for in the design.

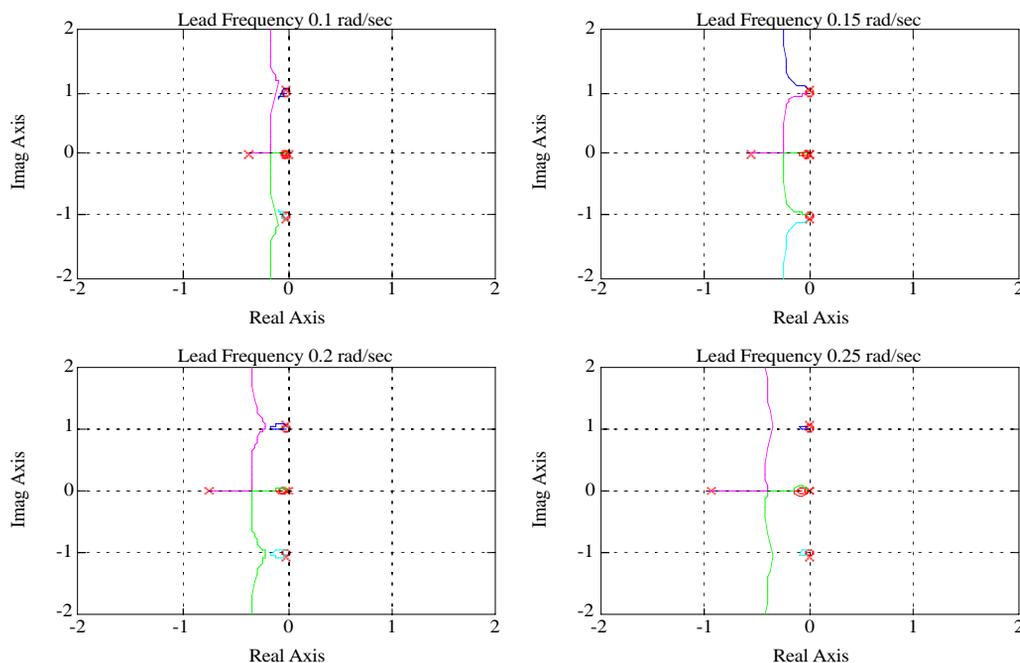
Our goal is to provide for a well damped response to a input command. Therefore, it is required to add damping to the double pole at the origin, and to add some damping to the flexible mode. We have three degrees of freedom in the lead compensator. We can adjust the gain, we can choose the maximum amount of phase lead it can add, and we can select the frequency where the phase lead is at its maximum.

If we were not concerned with damping the flex mode, we would put our maximum phase shift at the gain crossover. However, if we want to damp the flex modes, we want to add some phase lead where they have their gain crossovers. Note that in Figure 23 there are three gain crossovers, one at a frequency lower than the resonance and two associated with the resonance pole.

To begin, look at the root-locus plots for lead compensators with different maximum phase lead frequencies. Four root-locus plots are given in Figure 24. In each plot, one of the poles at the origin is pulled into the lead zero. The

other pole at the origin and the pole from the compensator coalesce on the real-axis then breakaway. In three of the plots, they then go to infinity as the gain increases.

**Figure 24. Root-Locus for Lead Compensators**



In the plot with the lead frequency equal to 0.15 rad/sec they are pulled into the flex mode zeros and the flex mode poles go to infinity. As the lead frequency increases, less and less damping is added to the resonance, but its effect on the closed loop system becomes less pronounced as the flex poles are pushed onto the flex zeros so they disappear from the transfer function. At even higher lead frequencies, the breakaway locus becomes straighter.

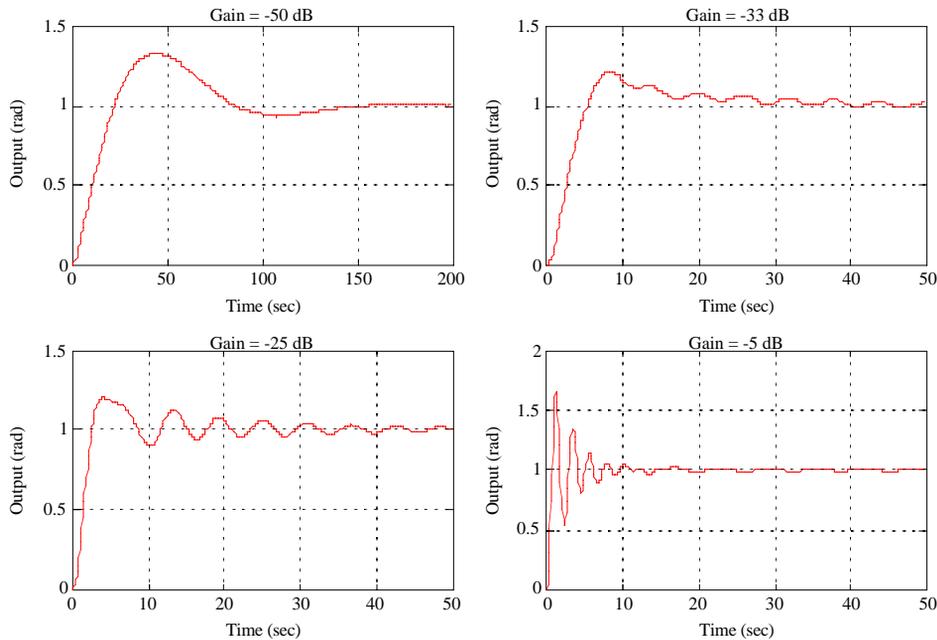
Thus, if our goal were just to get a good closed loop response, we would choose a high lead frequency and ignore the resonance. However, the resonance has not disappeared from the system; it just cannot be seen in the response of the body. In fact, it may be more lightly damped than before and this control strategy could lead to a failure in the structure.

Looking at the root-locus plots in Figure 24, a good starting point is the lead compensator with a peak lead frequency of 0.2 rad/sec. For a gain of -33 dB we get a damping ratio of 0.6861 for the rigid-body/controller poles and a damping ratio of 0.0133 for the flex mode. We have nearly critically damped the rigid-body motion and increased the damping ratio on the flex mode by a factor of three. If we increase the gain to -24 dB we can get a damping ratio of 0.0411 on the flex modes and 0.3829 on the rigid-body/controller poles.

Figure 25 shows the step responses for four different gains for the lead controller with a peak phase frequency of 0.2 rad/sec. The -50 dB shows no flex response but the response is very sluggish. The -5 dB case has a very oscillatory

transient response, but this dies out quickly and the very lightly damped flex mode response is barely visible. In the -33 dB and -25 dB cases the oscillation is due to the flex mode. While the latter has more damping on the flex mode the overall response is more oscillatory. With this particular compensator the -25 dB gain seems the best.

**Figure 25. Step Responses for the Compensator with a Peak Phase Frequency of 0.25 rad/sec**



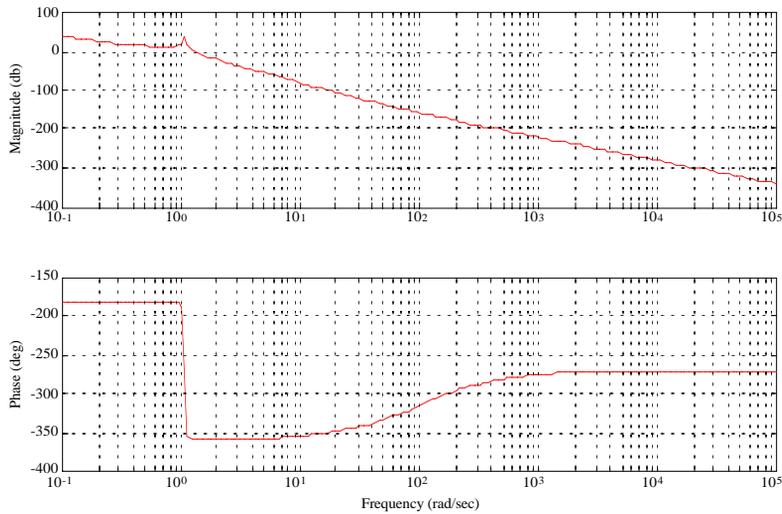
This simple compensator is not the best compensator that could be designed for this system. However, it serves to demonstrate what can be achieved with a very simple design approach. Even with this compensator, the four transient responses do not necessarily show the optimal response. One would not leave the maximum phase lead frequency fixed as the gain, and crossover were increased. Rather, one would push the lead frequency up. This would eliminate the oscillatory transient response, but would not increase the damping on the flex mode. To push the bandwidth up and increase the damping on the flex mode, a different compensation approach would be required.

#### 7.10.4 Non-Collocated Sensor and Actuator

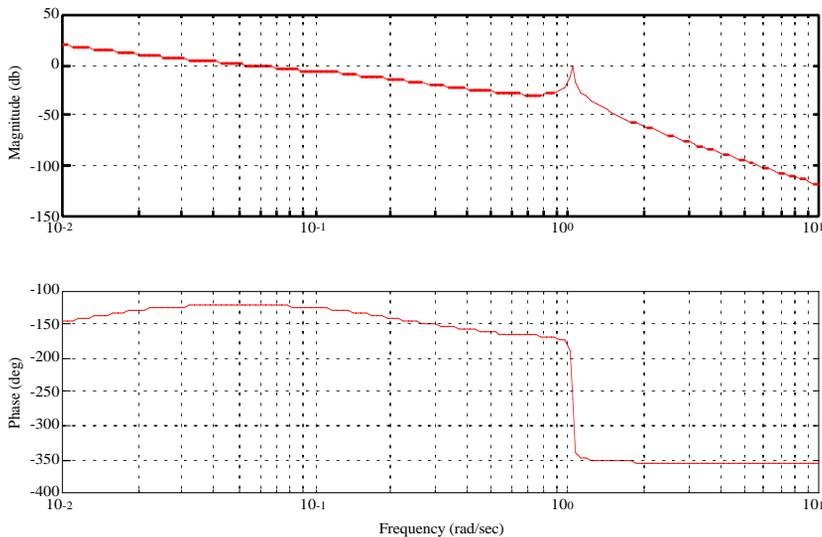
Figure 26 shows the Bode plot for a non-collocated sensor and actuator. Figure 27 shows the frequency response over a wider range to show the effect of the high frequency zero. As expected, it reduces the phase lag to  $-270^\circ$  at high frequencies. Clearly, at frequencies below the resonance, the control problem is the same as for the collocated sensor and actuator. To show this, use a phase lead controller with a cross-over at 0.05 rad/sec. The Bode plot is shown in Figure 27. -62 dB was chosen for the forward gain. The lead introduced by the controller stabilizes both the control crossover at 0.05 rad/sec and the first resonance crossover. The second crossover is stable because the gain is less than 0 dB. This controller has about 16 dB of margin.

Figure 28 shows the root locus plot. The system has two zeros at infinity, a high frequency plant zero and the controller zero. The two rigid body poles at the origin go around the controller zero, coalesce on the real axis then one goes to the plant zero and the other goes to the controller zero. The two flex poles go to the zeros at infinity by means of the right half plane. They cross into the right-half plane when the gain is about -46 dB.

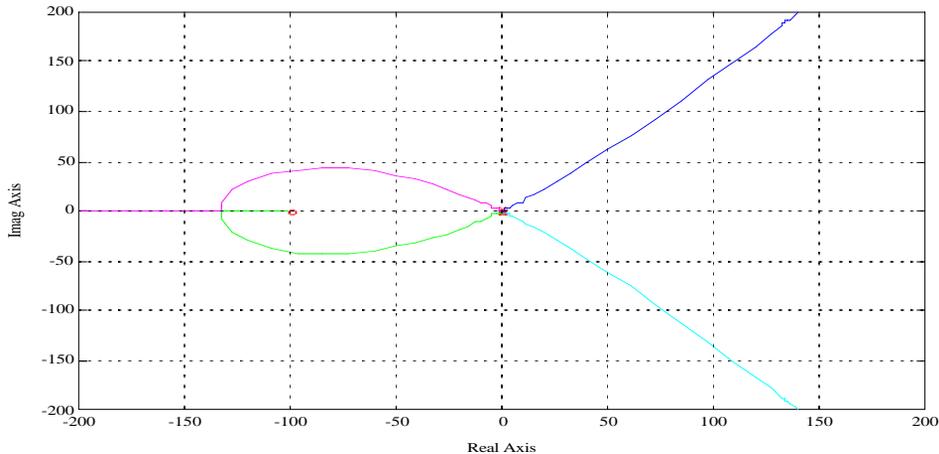
**Figure 26. Bode plot for the non-collocated sensor and actuator transfer function.**



**Figure 27. Bode plot for the non-collocated sensor and actuator transfer function with a phase-lead controller and a crossover at 0.05 rad/sec.**



**Figure 28. Root Locus plot for the non-collocated sensor and actuator transfer function with a phase lead controller.**



The controller, even at low gains, does not do anything to improve the damping on the flex mode. At best, it ignores it. Unlike the collocated case, the flex poles do not conveniently approach the open-loop zeros. Instead, they go unstable.

Ignoring the flex mode may be all that is needed to meet specifications. If that is the case the simple controller will do. Suppose, however, that you want your crossover to be above 1 rad/sec? The simple lead compensator will not work since it is only supplying a maximum of  $60^\circ$  of phase lead.

To compensate this plant, we need a pair of zeros to attract the flex poles. However, every pair of zeros requires another pair of poles, otherwise the controller would not be proper. In this plant just adding a pair of zeros (in addition to the lead compensator) would mean that at high frequencies the plant roll off like  $1/s$ , which is inadequate when unmodeled dynamics are considered.

The flex compensator will have the form

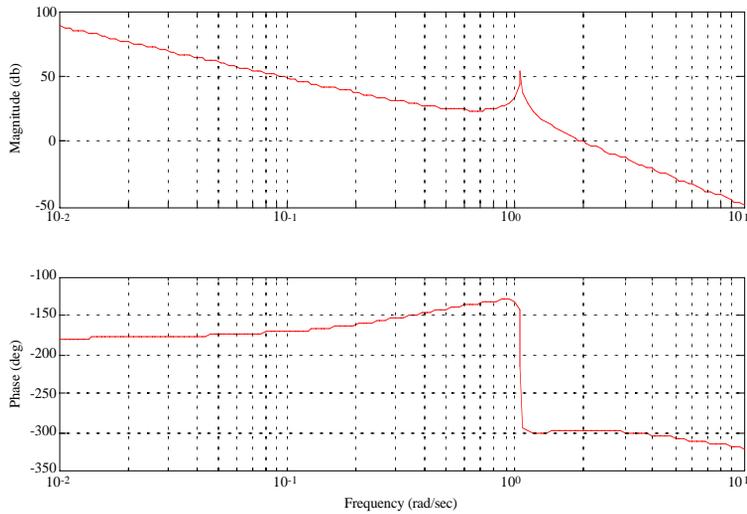
$$\frac{s^2 + 2\zeta_1\omega_1s + \omega_1^2}{s^2 + 2\zeta_2\omega_2s + \omega_2^2} \quad (7.18)$$

Start by putting a lead compensator with a crossover at 2 rad/sec. As expected, the resonance is now unstable since at the phase crossover the gain is more than 1. We need to advance the phase nearly  $180^\circ$  in the vicinity of the resonance, and the controller crossover.

To get the proper phase advance requires that the numerator of Equation (7.18) be at a lower frequency than the denominator. The zeros should also be at a lower frequency than the resonance to get the full effect of the zeros

before hitting the resonant poles. The appropriate pole and zero frequencies can be achieved with poles or zeros in either the right-half-plane or the left half plane.

**Figure 29. Bode plot with a high frequency phase lead controller**

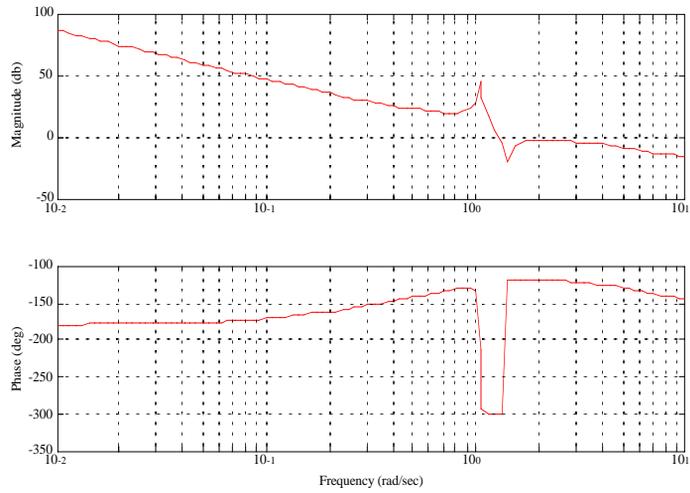


.The question of where to put the zeros has to do with our knowledge of the system. Try two different flex compensators of the form

$$\frac{s^2 + (1 \pm 0.4)^2}{s^2 + 0.2 \cdot 30 + 30^2} \quad (7.19)$$

The damping in the denominator is necessary to stabilize the high frequency pole.

**Figure 30. Bode plot with a flex compensator zero at 1.4 rad/sec.**



---

## 8.1 Introduction

Often a 3-axis torque command, or a 6-axis force and torque command must be distributed to multiple actuators. One approach is to design a multi-output control algorithm with one output for each actuator. This becomes complicated when fault-tolerance and redundancy management require that the system operate with subsets of the actuators or when the number of actuators is very large (the Shuttle Orbiter has forty-four thrusters.) A more practical approach is to decouple the torque distribution from the control.

When distributing torque in this manner, it is important that the actuators have similar dynamics, and that these dynamics are accounted for in the control design. One must be careful when trying to treat a DC motor like a pulse-width modulated thruster!

The following sections discuss the use of optimization to distribute torque to a set of actuators.

---

## 8.2 The Optimal Torque Distribution Problem

Given  $n$  actuators, all with similar dynamics as discussed above, three equality constraints, and  $n$  inequality constraints on the magnitude of the actuation, the general cost function to be minimized can be written as

$$L = C(u) + \lambda'(T - Bu) + \mu'u \quad (8.1)$$

where  $u$  is the control vector of order  $n$ ,  $C(u)$  is the scalar cost function to be applied to the control,  $\lambda$  is the 3-vector of Lagrange multipliers used to adjoin the equality constraints,  $\mathbf{T}$  is the desired torque vector,  $\mathbf{B}$  is the torque distribution matrix, and  $\mu$  is the vector of Lagrange multipliers used to adjoin inequality constraints on the control.

Inequality constraints come in a number of forms. For a scalar actuator, one in which the torque is the product of a scalar controlled by the controller and a fixed unit vector, the effective scalar actuation is

$$u_{eff} = \begin{cases} f(u) & u_{min} \leq u \leq u_{max} \\ 0 & otherwise \end{cases} \quad (8.2)$$

This would apply to a fixed orientation thruster (such as the Shuttle Orbiters RCS thrusters, but not is OMS thrusters) a reaction wheel, a magnetic torquer, a CMG gimbal actuator, etc. Generally, the effective output is a nonlinear function of the input (throttle setting, voltage, pulse duration, etc.). Unfortunately trying to minimize Equation (8.1) with a function like Equation (8.2) leads us to nonlinear optimization which, at present, is impractical in a real-time system.

In many cases, the actuator can be modeled as

$$u_{eff} = \begin{cases} u & u \geq 0 \\ 0 & otherwise \end{cases} \quad (8.3)$$

for a unidirectional actuator and

$$u_{eff} = u \quad (8.4)$$

For a bidirectional actuator such as a reaction wheel. If Equation (8.4) is used,  $m = 0$  and drops out of the cost function. If Equation (8.3) is used, we still have the simple constraint on  $u$ .

The next step is to select an appropriate cost functional. For fixed orientation thrusters, fuel is to be minimized which means

$$C = \sum_{i=1}^n c_i u_i \quad (8.5)$$

For a motor energy is usually minimized

$$C = \sum_{i=1}^n c_i u_i^2 \quad (8.6)$$

If Equation (8.6) is the cost functional, and Equation (8.4) is the control, the optimal command vector can be found in a straightforward manner. Combine Equation (8.1), Equation (8.4) and Equation (8.6) where Equation (8.6) has been replaced by the equivalent vector notation and  $c$  is a diagonal matrix whose components are the individual costs

$$L = \frac{1}{2}u'cu + \lambda'(T - Bu) \quad (8.7)$$

The partial derivative of L with respect to u is the vector equation

$$\frac{\partial L}{\partial u} = 0 = cu - B'\lambda \quad (8.8)$$

Taking the derivative of Equation (8.7) with respect to l gives the equality constraint  $T - Bu = 0$ . Substituting Equation (8.8) into the equality constraint gives the equation

$$T = Bc^{-1}B'\lambda \quad (8.9)$$

Solving for l and replacing l in Equation (8.8) with the result gives

$$u = c^{-1}B'(Bc^{-1}B')^{-1}T \quad (8.10)$$

which is the familiar formula for the pseudo-inverse. This solution minimizes the square of the command, so it is in some sense a minimum energy solution. It also penalizes large values of u most strongly, so it will tend to keep the actuators away from their limits. Since c and B are constant, the pseudo-inverse need not be computed on-board.

---

## 8.3 Examples

### 8.3.1 Reaction Wheels

Many earth-pointing spacecraft use four reaction wheels in a pyramid configuration about one of the axes. That axis is usually the one about which the spacecraft rotates as it goes around its orbit. Assume that four reaction wheels are in a pyramid and each is canted  $45^\circ$  from the z-axis. The control matrix is  $3 \times 4$  and in this case is

$$B = \begin{bmatrix} 0.7071 & -0.7071 & 0 & 0 \\ 0 & 0 & 0.7071 & -0.7071 \\ 0.7071 & 0.7071 & 0.7071 & 0.7071 \end{bmatrix} \quad (0.1)$$

If all four wheels have equal costs, c is the identity matrix and the torque distribution matrix is the  $4 \times 3$  matrix

$$B'(BB')^{-1} = \begin{bmatrix} 0.7071 & 0 & 0.3536 \\ -0.7071 & 0 & 0.3536 \\ 0 & 0.7071 & 0.3536 \\ 0 & -0.7071 & 0.3536 \end{bmatrix} \quad (0.2)$$

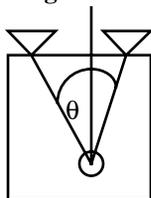
All four wheels will contribute equally to any z-axis torque demand. Wheel one will respond to a positive x-axis torque demand with a positive command and wheel 2 will respond with a negative command.

Another interesting example is for gimbaled thrusters.

### 8.3.2 Gimbaled Thrusters

Two gimbaled thrusters are mounted on the north face of a geosynchronous spacecraft and are to be used for north-south stationkeeping.

**Figure 31. Gimbaled Thrusters**



It is desirable to keep them as closely aligned with the z-axis as possible to minimize east-west coupling and cosine losses for the north-south burn. Since a squared cost functional will do just that we use (10) as our torque distribution law. Each thruster has two gimbals and the control system has been sized so that the gimbal deflections are always small. Assume that the double-gimbaled thruster first rotates about x and then about the y-axis

Gimbaled Thrusters

Each thruster has two gimbal input commands, x and y. The thrust unit vector is

$$u = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix} \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \sin\theta_y \\ -\cos\theta_y \sin\theta_x \\ \cos\theta_x \cos\theta_y \end{bmatrix} \quad (0.3)$$

The torque distribution matrix, after linearization, is

$$T = \begin{bmatrix} 0 & z_1 & 0 & z_2 \\ z_1 & 0 & z_2 & 0 \\ -y_1 & -x_1 & -y_2 & -x_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \quad (0.4)$$

where

$$u = \begin{bmatrix} \theta_{y1} \\ \theta_{x1} \\ \theta_{y2} \\ \theta_{x2} \end{bmatrix} \quad (0.5)$$

The disturbance torque is

$$T_d = \begin{bmatrix} y_1 + y_2 \\ -x_1 - x_2 \\ 0 \end{bmatrix} \quad (0.6)$$

So the thrusters on the north face should be located on opposite sides of the center-of-mass. Assume that the thrusters are 30 inches from the center of mass in z and located on the edge of the box on the corners so that the x and y locations are equal and opposite and each component is 30 inches. The command distribution matrix is

$$B = 30 \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ -1 & -1 & 1 & 1 \end{bmatrix} \quad (0.7)$$

and the command distribution matrix is

$$D = \begin{bmatrix} 0.0111 & 0 & 0.0250 \\ 0 & -0.0111 & -0.0250 \\ 0.0111 & 0 & -0.0250 \\ 0 & -0.0111 & 0.0250 \end{bmatrix} \quad (0.8)$$

z-axis torques are obtained by differentially rotating the thrusters. x and y axis torques are generated by single axis rotations. In practice, since most geosynchronous spacecraft have solar arrays on the north face of the spacecraft, the torque distribution matrix would be computed by linearizing the output of a plume interaction program which would give torque as a function of the gimbal angles, including plume interaction.

---

## 8.4 Simplex

Given the problem

$$\begin{aligned} x &= Au \\ \min(c^T u) \\ u &\geq 0 \end{aligned} \quad (8.11)$$

where the length of u is > length of x, the solution can be found using the simplex algorithm. It is also possible to handle a magnitude constraint on u without adding additional constraint equations.



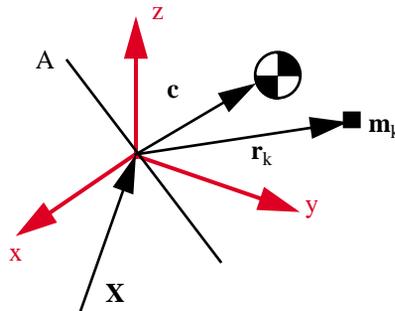
## 9.1 Introduction

This chapter provides an introduction to spacecraft attitude dynamics.

## 9.2 Rigid Body

Define a rigid body as a collection of point masses. The spacecraft is shown in Figure 9.1 on page 131.

**FIGURE 9.1** Rigid body



The position vector of each mass with respect to the inertial frame is

$$R_k = X + Ar_k \quad (9.1)$$

Assume that the spacecraft is rotating with respect to the inertial frame with angular velocity  $\omega$ . Then the derivative of  $A$  is

$$\dot{A} = A\omega^\times \quad (9.2)$$

We want to find the equations of motion about the center-of-mass. The position vector with respect to the center-of-mass is

$$D_k = A(r_k - c) \quad (9.3)$$

The total angular momentum with respect to the center-of-mass is

$$H = -\sum_k m_k D_k^\times \dot{D}_k \quad (9.4)$$

Where

$$\dot{D}_k = A\omega^\times(r_k - c) \quad (9.5)$$

and

$$\ddot{D}_k = A\omega^\times\omega^\times(r_k - c) - A(r_k - c)^\times\dot{\omega} \quad (9.6)$$

The derivative of the angular momentum in the body frame is

$$T = -A^T \sum_k m_k D_k^\times \ddot{D}_k \quad (9.7)$$

Expanding

$$= \sum_k m_k (r_k - c)^\times (\omega^\times \omega^\times (r_k - c) - (r_k - c)^\times c \dot{\omega}) \quad (9.8)$$

which is

$$\begin{aligned} T &= I\dot{\omega} + \omega^\times I\omega \\ I &= -\sum_k m_k (r_k - c)^\times (r_k - c)^\times \end{aligned} \quad (9.9)$$

This gives the rotational dynamics. The translational dynamics are

$$F = \sum_k m_k (\ddot{C} + \ddot{X}) \quad (9.10)$$

where  $C = Ac$ .

or

$$F = A \sum_k m_k (\omega^\times \omega^\times (r_k - c) - (r_k - c)^\times \dot{\omega}) \quad (9.11)$$

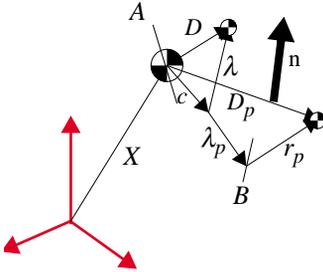
---

### 9.3 Multibody

The model of the multibody spacecraft is shown below.

**FIGURE 9.2 Multibody spacecraft model**

---



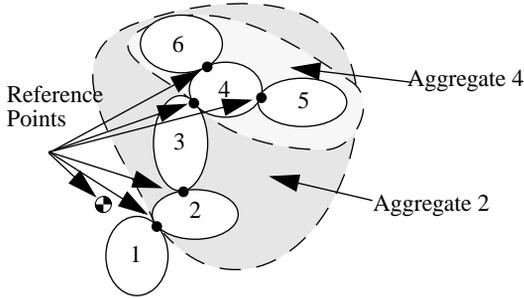
The multibody equations for a topological tree are found by writing the equations of motion for each aggregate body. The angular momentum for the  $k$ th aggregate is

$$H_k = \sum_{j=1}^{N_k} (B_j I_j \omega_j + m_j E_j^\times \dot{D}_j) \quad (9.12)$$

where  $B_j$  transforms from the  $j$ th body fixed frame to the inertial frame,  $E_j$  is the vector from the reference point for the  $k$ th aggregate to the center of mass of body  $j$ ,  $D_j$  is the vector from the system center of mass to the center of mass of the  $j$ th body,  $m_j$  is the mass of the  $j$ th body,  $\omega_j$  is the angular rate of the  $j$ th body in the  $j$ th body fixed frame and  $I_j$  is the inertia of the  $j$ th body in the  $j$ th body fixed frame. The reference point for the core aggregate (which includes every body in the spacecraft) is the system center of mass. Consequently  $E_1 = D_1$ . for the core frame  $B_1 = A$  with  $A$  defined as above. The reference point for all other aggregates is the hinge that leads to the core. No closed loops are

allowed. The position of the center-of-mass of each body is at position  $r_k$  in the body frame where  $r_k$  is defined as the

FIGURE 9.3 A Topological Tree



vector from the hinge that is nearest the core.  $\lambda_k$  is the vector from the reference point of body 1 to the hinge point that leads to body  $k$ . In this diagram, hinge  $k$  connects body 1 and body  $k$  and hinge  $m$  connects body  $m$  and body  $k$ .

The equations of motion are the  $N_k$  equations

$$T_k = B_k^T \sum_{j=1}^{N_k} (B_j I_j \dot{\omega}_j + \dot{B}_j I_j \omega_j + m_j (\dot{E}_j^\times \dot{D}_j + E_j^\times \ddot{D}_j)) \quad (9.13)$$

where  $I_j$  is assumed constant in the  $j$ th body frame and  $T_k$  is defined in the  $k$ th frame. For  $k = 1$  (the core body) this simplifies to

$$T_1 = A^T \sum_{j=1}^{N_k} (B_j I_j \dot{\omega}_j + \dot{B}_j I_j \omega_j + m_j D_j^\times \ddot{D}_j) \quad (9.14)$$

$E_j$  is measured from a fixed reference point while  $D_j$  is measured from the system center of mass. The center of mass is defined in the core frame by

$$\begin{aligned}
 0 &= \sum_{k=1}^{N_1} m_k (e_k + c) \\
 c &= -\frac{\sum_{k=1}^{N_1} m_k e_k}{\sum_{k=1}^{N_1} m_k} \\
 c &= -\sum_{k=1}^{N_1} \mu_k e_k
 \end{aligned} \tag{9.15}$$

Therefore

$$\begin{aligned}
 d_k &= e_k + c \\
 d_k &= e_k - \sum_{j=1}^{N_1} \mu_j e_j \\
 d_k &= \sum_{j=1}^{N_1} \mu_{kj} e_j
 \end{aligned} \tag{9.16}$$

where

$$\mu_{kj} = \begin{cases} -\mu_j & j \neq k \\ 1 - \mu_j & j = k \end{cases} \tag{9.17}$$

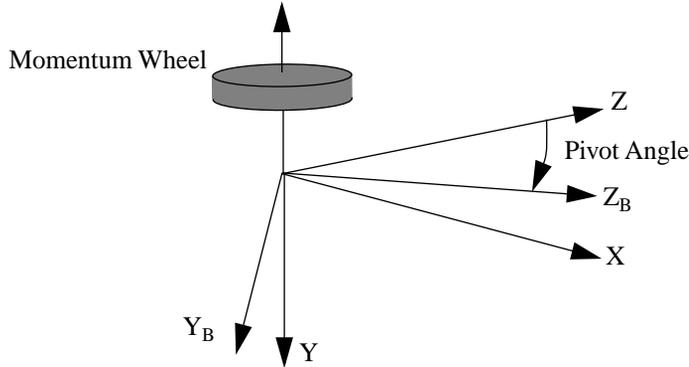
---

## 9.4 Pivoted Momentum Wheel

An interesting multibody problem is that of a pivoted momentum wheel. This configuration is used in RCA and OSC bias momentum satellites.

The pivot axis is the body x-axis. When the pivot rotates, the momentum wheel stays inertially fixed and the spacecraft rotates. This way roll pointing can be controlled. Since the pivot torque is internal, the inertial angular momentum magnitude and direction do not change.

FIGURE 9.4 Pivoted momentum wheel [X,Y,Z] are LVLH axes. The orbit rate is aligned with the -Y axis.



Designate the core spacecraft as 1, the pivot platform as 2, and the momentum wheel as 3. The transformation matrix from the core frame to the inertial frame is  $b_1$ , the transformation matrix from the pivot frame to the core frame is  $b_2$  and the transformation matrix from the momentum wheel frame to the pivot frame is  $b_3$ . The angular rate of frame  $k$  with respect to the previous frame is  $\Omega_k$ .  $b_3$  transforms from the wheel base frame to the pivot frame. The wheel is assumed to be symmetric and rotate only about its axis of symmetry. Consequently, the time derivative of  $b_3$  is zero.

The angular rates of the three components in their own frames are

$$\begin{aligned}\omega_1 &= \Omega_1 \\ \omega_2 &= \Omega_2 + b_2^T \omega_1 \\ \omega_3 &= \Omega_3 + b_3^T \omega_2\end{aligned}\tag{9.18}$$

The angular momentum of the three components is

$$\begin{aligned}h_3 &= I_3 \omega_3 \\ h_2 &= I_2 \omega_2 + b_3 h_3 \\ h_1 &= I_1 \omega_1 + b_2 h_2\end{aligned}\tag{9.19}$$

Notice the recursive nature of the two sets of equations.

To find the equations of motion, we must take derivatives of each equation in the inertial frame, transform the result back into each body's frame and set it equal to the torque. In the case of the momentum wheel and pivot the torques are scalar along the pivot x-axis and momentum wheel y-axis. The equations of motion are

$$\begin{aligned}
 T_1 &= b_1^T \frac{d}{dt} (b_1 h_1) \\
 T_2 &= u(1) b_2^T b_1^T \frac{d}{dt} (b_1 b_2 h_2) \\
 T_3 &= u(2) b_3^T b_2^T b_1^T \frac{d}{dt} (b_1 b_2 b_3 h_3)
 \end{aligned} \tag{9.20}$$

where  $u(k)$  selects the  $k$ th component of the equation.

Expanding

$$\begin{aligned}
 T_1 &= b_1^T (\dot{b}_1 h_1 + b_1 \dot{h}_1) \\
 T_2 &= u(1) b_2^T b_1^T (\dot{b}_1 b_2 h_2 + b_1 \dot{b}_2 h_2 + b_1 b_2 \dot{h}_2) \\
 T_3 &= u(2) b_3^T b_2^T b_1^T (\dot{b}_1 b_2 b_3 h_3 + b_1 \dot{b}_2 b_3 h_3 + b_1 b_2 \dot{b}_3 h_3)
 \end{aligned} \tag{9.21}$$

The derivatives of the transformation matrices are

$$\begin{aligned}
 \dot{b}_1 &= b_1 \Omega_1^\times \\
 \dot{b}_2 &= b_2 \Omega_2^\times
 \end{aligned} \tag{9.22}$$

Substituting these into the above equations

$$\begin{aligned}
 T_1 &= \Omega_1^\times h_1 + \dot{h}_1 \\
 T_2 &= u(1) (b_2^T \Omega_1^\times b_2 h_2 + \Omega_2^\times h_2 + \dot{h}_2) \\
 T_3 &= u(2) (b_3^T b_2^T \Omega_1^\times b_2 b_3 h_3 + b_3^T \Omega_2^\times b_3 h_3 + \dot{h}_3)
 \end{aligned} \tag{9.23}$$

$$\begin{aligned}
 \dot{h}_3 &= I_3 \dot{\omega}_3 \\
 \dot{h}_2 &= I_2 \dot{\omega}_2 + b_3 \dot{h}_3 \\
 \dot{h}_1 &= I_1 \dot{\omega}_1 + b_2 \Omega_2^\times h_2 + b_2 \dot{h}_2
 \end{aligned} \tag{9.24}$$

Substituting

$$\begin{aligned}
 T_1 &= \Omega_1^\times h_1 + I_1 \dot{\omega}_1 + b_2 \Omega_2^\times h_2 + b_2 (I_2 \dot{\omega}_2 + b_3 I_3 \dot{\omega}_3) \\
 T_2 &= u(1) (b_2^T \Omega_1^\times b_2 h_2 + \Omega_2^\times h_2 + I_2 \dot{\omega}_2 + b_3 I_3 \dot{\omega}_3) \\
 T_3 &= u(2) (b_3^T b_2^T \Omega_1^\times b_2 b_3 h_3 + b_3^T \Omega_2^\times b_3 h_3 + I_3 \dot{\omega}_3)
 \end{aligned} \tag{9.25}$$

Define

$$\begin{aligned}
 s_1 &= \Omega_1^\times \\
 s_2 &= b_2^T \Omega_1^\times b_2 + \Omega_2^\times \\
 s_3 &= b_3^T s_3 b_3
 \end{aligned} \tag{9.26}$$

therefore

$$\begin{aligned}
 T_1 &= s_1 h_1 + I_1 \dot{\omega}_1 + \dot{b}_2 h_2 + b_2 (I_2 \dot{\omega}_2 + b_3 I_3 \dot{\omega}_3) \\
 T_2 &= u(1)(s_2 h_2 + I_2 \dot{\omega}_2 + b_3 I_3 \dot{\omega}_3) \\
 T_3 &= u(2)(s_3 h_3 + I_3 \dot{\omega}_3)
 \end{aligned} \tag{9.27}$$

where

$$\begin{aligned}
 \dot{\omega}_1 &= \dot{\Omega}_1 \\
 \dot{\omega}_2 &= \dot{\Omega}_2 + b_2^T \dot{\omega}_1 + \dot{b}_2^T \omega_1 \\
 \dot{\omega}_3 &= \dot{\Omega}_3 + b_3^T \dot{\omega}_2
 \end{aligned} \tag{9.28}$$

If we write the final equations as

$$I \dot{\Omega} = T - p \tag{9.29}$$

we get

$$p = \begin{bmatrix} s_1 h_1 + \dot{h}_1^P \\ u(1)(s_2 h_2 + \dot{h}_2^P) \\ u(2)(s_3 h_3 + \dot{h}_3^P) \end{bmatrix} \tag{9.30}$$

where

$$\begin{aligned}
 \dot{h}_3^P &= I_3 b_3^T \dot{b}_2^T \omega_1 \\
 \dot{h}_2^P &= I_2 \dot{b}_2^T \omega_1 + b_3 \dot{h}_3^P \\
 \dot{h}_1^P &= \dot{b}_2 h_2 + b_2 \dot{h}_2^P
 \end{aligned} \tag{9.31}$$

which are the momentum derivative terms that do not contain rate derivatives and

$$T = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} \quad (9.32)$$

The remaining terms

$$\begin{aligned} & I_1 \dot{\omega}_1 + b_2(I_2 \dot{\omega}_2 + b_3 I_3 \dot{\omega}_3) \\ & u(1)(I_2 \dot{\omega}_2 + b_3 I_3 \dot{\omega}_3) \\ & u(2)(I_3 \dot{\omega}_3) \end{aligned} \quad (9.33)$$

form the basis for the inertia matrix

$$I = \begin{bmatrix} I_1 + b_2(I_2 + b_3 I_3 b_3^T) b_2^T & b_2(I_2 + b_3 I_3 b_3^T) & b_2 b_3 I_3 \\ (I_2 + b_3 I_3 b_3^T) b_2^T & I_2 + b_3 I_3 b_3^T & b_3 I_3 \\ I_3 b_3^T b_2^T & I_3 b_3^T & I_3 \end{bmatrix} \quad (9.34)$$

If we define primed inertias as the total inertia of a body, including everything further up the chain, we get

$$I = \begin{bmatrix} I_1' & b_2 I_2' & b_2 b_3 I_3' \\ I_2' b_2^T & I_2' & b_3 I_3' \\ I_3' b_3^T b_2^T & I_3' b_3^T & I_3' \end{bmatrix} \quad (9.35)$$

where

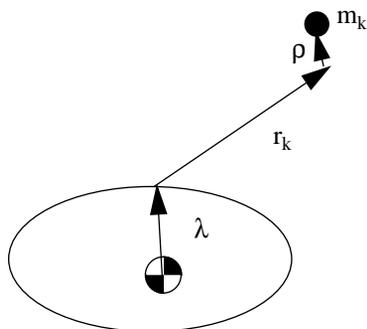
$$\begin{aligned} I_1' &= I_1 + b_2 I_2' b_2^T \\ I_2' &= I_2 + b_3 I_3' b_3^T \\ I_3' &= I_3 \end{aligned} \quad (9.36)$$

---

## 9.5 Modeling Flexible Structures

The simplest model of a flexible structure is shown in the following figure.

FIGURE 9.5 Flexible structure



The spacecraft is assumed to be composed of a rigid core to which a flexible structure is attached. The mass of the core is much larger than the mass of the structure, and the movement of the structure relative to the core is assumed to be small. Define

$$D = A(\lambda + r + \rho) \quad (9.37)$$

where the subscripts on  $r$  and  $\rho$  have been dropped. Taking derivatives

$$\begin{aligned} \dot{D} &= A\omega^\times(\lambda + r + \rho) + A\dot{\rho} \\ \ddot{D} &= A(\omega^\times\omega^\times(\lambda + r + \rho) + 2\omega^\times\dot{\rho} - (\lambda + r + \rho)^\times\dot{\omega} + \ddot{\rho}) \end{aligned} \quad (9.38)$$

The total angular momentum of the system is

$$H = AI\omega + mD^\times\dot{D} \quad (9.39)$$

where  $A$  transforms from the body to the inertial frame. The equations of motion of the entire spacecraft are

$$T = I\dot{\omega} + \omega^\times I\omega + m(\lambda + r + \rho)^\times(\omega^\times\omega^\times(\lambda + r + \rho) + 2\omega^\times\dot{\rho} - (\lambda + r + \rho)^\times\dot{\omega} + \ddot{\rho}) \quad (9.40)$$

If  $\rho$  is much smaller than  $r$

$$T = I\dot{\omega} + \omega^\times I\omega + m(\lambda + r)^\times(\omega^\times\omega^\times(\lambda + r) + 2\omega^\times\dot{\rho} - (\lambda + r)^\times\dot{\omega} + \ddot{\rho}) \quad (9.41)$$

Collecting terms

$$T = (I - m(\lambda + r)^\times(\lambda + r)^\times)\dot{\omega} + \omega^\times(I - m(\lambda + r)^\times(\lambda + r)^\times)\omega + 2m(\lambda + r)^\times(\omega^\times\dot{\rho}) + m\ddot{\rho} \quad (9.42)$$

or

$$\begin{aligned} T &= I_T \dot{\omega} + \omega^\times I_T \omega + 2m(\lambda + r)^\times (\omega^\times \dot{\rho}) + m(\lambda + r)^\times \ddot{\rho} \\ I_T &= I - m(\lambda + r)^\times (\lambda + r)^\times \end{aligned} \quad (9.43)$$

These equations are nonlinear in both body angular rate,  $\omega$  and mass displacement,  $\rho$ . The inertia matrix is constant because the flex deformation is assumed small.

The equation for the mass is

$$f = m(\omega^\times \omega^\times (\lambda + r) + 2\omega^\times \dot{\rho} - (\lambda + r)^\times \dot{\omega} + \ddot{\rho}) \quad (9.44)$$

where  $F$  are all of the forces acting on the mass. The coupled equations are

$$\begin{bmatrix} I_T & m(\lambda + r)^\times \\ -m(\lambda + r)^\times & m \end{bmatrix} \begin{bmatrix} \dot{\omega} \\ \dot{\rho} \end{bmatrix} = - \begin{bmatrix} \omega^\times I_T \omega + 2m(\lambda + r)^\times (\omega^\times \dot{\rho}) \\ m(\omega^\times \omega^\times (\lambda + r) + 2\omega^\times \dot{\rho}) \end{bmatrix} + \begin{bmatrix} T \\ f \end{bmatrix} \quad (9.45)$$

If  $m$  is small the equations decouple

$$\begin{bmatrix} I_T & 0 \\ -m(\lambda + r)^\times & m \end{bmatrix} \begin{bmatrix} \dot{\omega} \\ \dot{\rho} \end{bmatrix} = - \begin{bmatrix} \omega^\times I_T \omega \\ m(\omega^\times \omega^\times (\lambda + r) + 2\omega^\times \dot{\rho}) \end{bmatrix} + \begin{bmatrix} T \\ f \end{bmatrix} \quad (9.46)$$

But the mass is still driven by the core rates.

Further linearization of these equations must be done with care. If both  $\omega$  and  $d\rho/dt$  are small then they become

$$\begin{bmatrix} I_T & m(\lambda + r)^\times \\ -m(\lambda + r)^\times & m \end{bmatrix} \begin{bmatrix} \dot{\omega} \\ \dot{\rho} \end{bmatrix} = \begin{bmatrix} T \\ f \end{bmatrix} \quad (9.47)$$

The coupling is entirely through the inertia matrix. If  $I_T$  is large and the rates are small the equations become

$$\begin{bmatrix} I_T & m(\lambda + r)^\times \\ -m(\lambda + r)^\times & m \end{bmatrix} \begin{bmatrix} \dot{\omega} \\ \dot{\rho} \end{bmatrix} = - \begin{bmatrix} \omega^\times I_T \omega \\ m\omega^\times \omega^\times (\lambda + r) \end{bmatrix} + \begin{bmatrix} T \\ f \end{bmatrix} \quad (9.48)$$

The mass still has a  $\omega^2$  term driving it but  $\rho$  only appears on the left-hand-side.

This derivation can be extended to multiple bodies. First rewrite the equations in the form

$$\begin{aligned}
 T &= I\dot{\omega} + \omega^\times I\omega - \sum_k (\lambda + r_k)^\times f_k \\
 f_k &= m(\omega^\times \omega^\times (\lambda + r_k) - (\lambda + r_k)^\times \dot{\omega} + \ddot{\rho}_k) \\
 f_k &= \sum_j k_{kj}(\rho_j - \rho_k)
 \end{aligned} \tag{9.49}$$

Notice that  $I$  is now just the core inertia. Just writing the equations for the masses

$$M\ddot{\rho} + K\rho = F \tag{9.50}$$

where  $K$  is symmetric and full. If we define the following variables

$$\begin{aligned}
 \rho &= \Phi\eta \\
 \Phi^T K \Phi &= \Omega^2 \\
 \Phi^T M \Phi &= E
 \end{aligned} \tag{9.51}$$

where  $E$  is the identity matrix. Since  $\rho$  is a vector quantity,  $K$  is  $3 \times n$  by  $3 \times n$ . If the number of modes is  $m$ ,  $\Phi$  is  $3 \times n$  by  $m$ . The mass equations become

$$\ddot{\eta} + \Omega^2 \eta = \Phi^T F \tag{9.52}$$

where  $F$  contains the terms dependent on  $\omega$ .

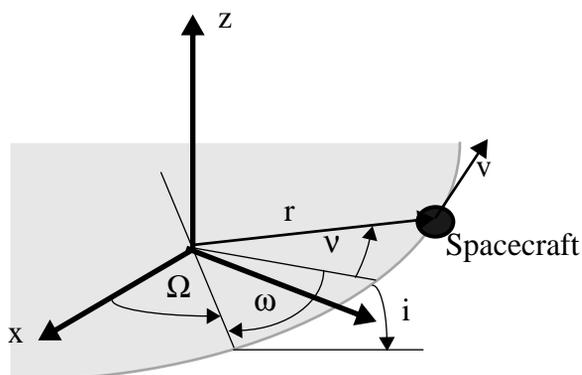
---

## 10.1 Representations of Orbits

Seven parameters are necessary to define an orbit of a spacecraft about a spherically symmetric body. One is the gravitational parameter, generally denoted by the symbol  $\mu$ . There are many ways of representing the other six elements. The most popular are position and velocity ( $r$  and  $v$ ) vector and Keplerian elements, both shown in Figure 10.1 below.

**FIGURE 10.1** Orbital Elements

---



The Keplerian elements are defined as follows:  $\Omega$  is the longitude of the ascending node, or the angle from the +X axis of the reference frame to the line where the orbit plane intersects the xy-plane;  $\omega$  is the argument of perigee and is the angle in the orbit plane between the ascending node line and perigee (where the orbit is closest to the center of

the Earth);  $\nu$  is the true anomaly and is the angle between perigee and the spacecraft;  $i$  is the inclination and is the angle between the  $xy$ -plane and the orbit plane; the “size” of the orbit is determined by the semi major axis  $a$ , which is the average of the perigee radius and apogee radius; and the final parameter is  $e$ , the eccentricity, which determines the shape of the orbit. If  $e = 0$  the orbit is circular, if  $e = 1$  it is parabolic. Any eccentricity in between results in an ellipse. If  $e > 1$ , then the orbit is a hyperbola.

Other types of elements commonly used are the equinoctial elements. The equinoctial elements are given by

$$\left[ a \ P_1 \ P_2 \ Q_1 \ Q_2 \ l \right] \quad (10.1)$$

where  $a$  is the semi major axis and

$$\begin{aligned} P_1 &= e \sin \varpi \\ P_2 &= e \cos \varpi \\ Q_1 &= \tan \frac{i}{2} \sin \Omega \\ Q_2 &= \tan \frac{i}{2} \cos \Omega \\ l &= \varpi + M \\ \varpi &= \omega + \Omega \end{aligned} \quad (10.2)$$

where  $M$  is the mean anomaly,  $e$  is eccentricity,  $i$  is inclination,  $\omega$  is the argument of perigee and  $\Omega$  is the right ascension of the ascending node. The right ascension of the ascending node and inclination do not appear independently in these elements since for small inclinations they are poorly defined. Eccentricity and argument of perigee are also combined. These are useful when the orbit is only slightly inclined and has small eccentricity.

---

## 10.2 Propagating Orbits

There are three general methods for propagating an orbit. One is simply *Kepler propagation*. For example, if there are no perturbing accelerations on the spacecraft (such as thruster firings, solar pressure or the gravitational pull of other planets) it is possible to just propagate the elements directly. More complex propagators can account for some kinds of perturbations.

The second method is an *analytical theory*. This is used for planets and their moons. Very complex nonlinear expressions can be used to predict the position of a planet or moon with high accuracy over long periods of time. Some of these analytical theories are hundreds of pages long. While rarely used for attitude or orbit control work, they can be useful onboard flight computers if it is necessary to predict the position of an astronomical body.

The third method is *numerical integration*, which is used for most practical orbit modeling. It allows the engineer to incorporate all perturbations in a straightforward fashion and permits easy modification of the models for additional disturbance sources.

## Kepler Propagation

If perturbing accelerations are small, it is possible to propagate the orbital elements directly. Kepler propagation is very useful when you just need to test a control system as the spacecraft orbits the planet and you are not concerned with what is happening to the orbit as a result of thruster firings, etc. To determine the orbit it is necessary to know the eccentric anomaly. The equation that must be solved for obtaining the eccentric anomaly from elliptical orbits is

$$M = E - e \sin E \quad (10.3)$$

which is known as Kepler's equation. It is necessary to solve for  $E$  iteratively but this can be done in a straightforward fashion.

Impulse changes in velocity can be accommodated easily. To change the orbit with an impulse change in velocity in which the velocity changes while the position stays fixed just

1. Convert the orbital elements to  $r$  and  $v$  vectors.
2. Add the  $\Delta v$  to  $v$ .
3. Convert  $v + \Delta v$  and  $r$  back to orbital elements

This procedure is useful for modeling solid rocket burns or any delta-V which happens in a time much less than the orbit period.

## Numerical Integration

The state equations for orbit propagation are

$$\begin{aligned} \frac{dr}{dt} &= v \\ m \frac{dv}{dt} &= -\frac{m\mu r}{(r^T r)^{3/2}} + mg(r) + f(\theta) + h \end{aligned} \quad (10.4)$$

The terms on the right-hand-side of the velocity derivative equation are the spherical gravity force, additional forces that are functions of position, forces that are functions of angles and forces that are independent of orientation or position (such as thruster firings.) Orientation dependent forces include aerodynamic drag and solar pressure forces. The orientation may just be the orientation of the spacecraft with respect to inertial space, or may include the relative orientation of components such as solar arrays. If the orientation forces are small, then the orbit equations may be decoupled from the attitude equations and propagated separately. Otherwise, they must be integrated together. For example, the drag on the space shuttle in low earth orbit is dependent on orientation so the attitude and orbit equations cannot be decoupled.

For precise orbit predictions it is necessary to use integrators with error correction. The most widely used today are the Runge Kutta  $n/n+1$  integrators in which coefficients of the equations are chosen so that with minimal extra computation a  $n$ th order and  $n+1$ th order solution are computed. The results can be compared and the step size adjusted. Some integrators will also change order to minimize the solution error.



---

## *11.1 Introduction*

This chapter describes how to prepare pointing and propellant budgets—the primary means by which you convey to your customer that you have met his or her requirements.

---

## *11.2 Pointing Budgets*

### **11.2.1 Methodology**

There are a number of competing methodologies for doing pointing budgets for satellites. The most popular is to divide the error contributions into four temporal categories:

- Bias (e.g. manufacturing misalignment)
- Long-Term (e.g. seasonal variation in the yaw body fixed torque on a momentum bias spacecraft without yaw sensing)
- Diurnal (e.g. thermally induced rotation)
- Short Term (e.g. random noise)

and treat all contributions within each temporal category as if they were normally distributed, uncorrelated random variables. The contributions within each category are then summed (by taking the square root of the sum of their squares). The totals for each temporary category are then added to get the final per axis numbers.

Other methodologies examine each contribution and determine its nature, separating out deterministic and other contributions from random. For example, orbit errors should be separated from alignment errors. This provides somewhat more conservative but statistically more justifiable results.

### 11.2.2 Beam Pointing Accuracy

Once roll, pitch, and yaw error values have been computed, they must be combined to form the beam pointing accuracy.

The beam pointing accuracy is

$$\begin{aligned}\phi_{AZ} &= \theta_{\text{Roll}} + \left(\frac{\delta_{AZ}}{180/\pi}\right)\theta_{\text{Yaw}} \\ \phi_{EL} &= \theta_{\text{Pitch}} - \left(\frac{\delta_{EL}}{180/\pi}\right)\theta_{\text{Yaw}}\end{aligned}\tag{11.1}$$

where  $\delta_{EL}$  and  $\delta_{AZ}$  are the beam center offsets.

### 11.2.3 Example Pointing Budget

An example pointing budget is shown below.

**TABLE 11.1 Normal mode pointing budget**

Pointing budget 22-Jun-1999

	Item	Roll	Pitch	Yaw	
<b>Bias</b>					
1	ESA temperature effects	0.0150	0.0150	0.0000	deg
2	ESA optics w.r.t. cube	0.0250	0.0250	0.0000	deg
3	ESA cube to MRC	0.0250	0.0250	0.0000	deg
4	Antenna B.S. w.r.t. antenna RC	0.0180	0.0180	0.0000	deg
5	Antenna RC w.r.t. MRC	0.0150	0.0150	0.0000	deg
6	MWA spin axis w.r.t. mirror	0.0060	0.0060	0.0000	deg
7	MWA mirror w.r.t. MRC	0.0170	0.0170	0.0000	deg
8	Attitude sensing (earth radiance)	0.0012	0.0012	0.0000	deg
9	S/C disturbance torque	0.0031	0.0076	0.0000	deg
	<b>Subtotal</b>	<b>0.0486</b>	<b>0.0491</b>	<b>0.0000</b>	<b>deg</b>
<b>Diurnal</b>					
10	East/west position	0.0100	0.0100	0.0000	deg

---

## Propellant Budgets

---

**TABLE 11.1 Normal mode pointing budget**

11	Attitude sensing (earth radiance)	0.0100	0.0100	0.0000	deg
12	Orbit inclination	0.0100	0.0000	0.0100	deg
13	S/C disturbance torque	0.0080	0.0000	0.1470	deg
14	Antenna thermal distortion w.r.t. ESA RC	0.0100	0.0100	0.0000	deg
15	Thermal distortion MWA w.r.t. ESA RC	0.0200	0.0500	0.0000	deg
	<b>Subtotal</b>	<b>0.0294</b>	<b>0.0529</b>	<b>0.1473</b>	<b>deg</b>
<b>Short Term</b>					
16	Attitude sensor & actuator noise/resolution	0.0200	0.0200	0.0000	deg
17	SA stepping transients	0.0000	0.0500	0.0000	deg
18	S/C disturbance torques	0.0000	0.0000	0.0000	deg
	<b>Subtotal</b>	<b>0.0200</b>	<b>0.0539</b>	<b>0.0000</b>	<b>deg</b>
	<b>Total</b>	<b>0.0980</b>	<b>0.1558</b>	<b>0.1473</b>	<b>deg</b>

CEP = 0.1707 deg

This budget was generated for a momentum bias spacecraft that uses an earth sensor (ESA) to measure roll and pitch. The quantity of interest is the beam alignment (antenna boresight or B.S.) with respect to a target on the earth. The temporal quantities are: bias, assumed fixed for the life of the spacecraft; diurnal, which vary with the orbit period; and short term, which change at rates faster than orbit rate. Yaw is not sensed and is controlled solely by the spacecraft momentum. The largest component, in this case, is the orbit rate disturbance so the only entries for yaw are orbit inclination and this disturbance related error. In roll and pitch there are many errors due to misalignments. Alignments are measured using an optical system. MRC means master reference cube, an optical target that is the master reference for the spacecraft. ESA RC is the reference cube for the earth sensor.

ESA errors appear in all three temporal categories. These are taken from the manufacturers specifications. All alignment errors are determined by measuring post vibration errors. The spacecraft is aligned and then placed on a shaker to simulate launch. The change in alignments are considered the alignment uncertainties.

---

## 11.3 Propellant Budgets

### 11.3.1 Introduction

Propellant budgets are detailed accountings of the propellant needed for all spacecraft operations. They include fuel needed for all orbit change operations and for attitude control operations which may include pointing control, momentum unloading, reorientations, etc.

Attitude control fuel consumption can be divided into components that are proportional to the orbit change fuel consumption and those that are not. The former are due to misalignments in the delta-V engines or asymmetries of the

delta-V engines with respect to the center-of-mass. The attitude control system must cancel the disturbance torques due to the delta-V engines.

The second component includes firings that are not proportional to the magnitude of the delta-V. Both components must be determined through analysis and simulation.

Orbit operations which involve changing the velocity of the spacecraft include:

1. Station acquisition
2. Stationkeeping
3. Station changes
4. Deorbit

Stationkeeping includes in-plane and out-of-planet maneuvers. Station changes are generally within the plane of the orbit. Deorbit maneuvers include forcing an early reentry of the satellite for low-earth orbit satellites and maneuvers to push geosynchronous satellites out of a geosynchronous orbit.

### 11.3.2 Example Propellant Budget

Table 11.2 below shows an example propellant budget. This shows a budget for a hypothetical geosynchronous spacecraft through its second year of life. There are three thruster systems used. The mission orbit system is monopropellant hydrazine and is divided into two half systems that are pressurized independently. The third system is the solid rocket motor.

**TABLE 11.2 Example Propellant Budget**

Item	Description	Total	ACS Isp	DV Isp	Fuel Remaining	Fuel Remaining	Fuel Remaining	Fuel Used	Fuel Used	Fuel Used
1	Initial State	2055.92	0.00	0.00	117.89	117.89	865.00	0.000	0.000	0.000
2	SPM NON to AKM	2055.82	100.50	0.00	117.84	117.84	865.00	0.051	0.051	0.000
3	Spin Up 5 to 60 RPM	2051.14	118.33	0.00	115.50	115.50	865.00	2.341	2.341	0.000
4	SPM Trim	2051.13	100.10	0.00	115.50	115.50	865.00	0.001	0.001	0.000
5	AKM Firing	1188.56	0.00	265.00	115.50	115.50	2.43	0.000	0.000	862.574
6	Despin to 5 RPM	1183.88	118.33	0.00	113.02	113.30	2.43	2.481	2.202	0.000
7	SPM AKM to PON	1183.43	102.51	0.00	112.79	113.07	2.43	0.225	0.226	0.000
8	Despin to 2.53 RPM	1183.22	118.33	0.00	112.68	112.97	2.43	0.111	0.099	0.000
9	Despin from 1.1 RPM to orbit rate	1183.15	118.33	0.00	112.64	112.94	2.43	0.037	0.033	0.000
10	Drift orbit station acquisition	1174.15	100.00	210.00	108.14	108.44	2.43	4.500	4.500	0.000
11	Drift orbit inclination correction	1163.94	100.02	360.00	103.05	103.32	2.43	5.093	5.115	0.000

---

**Propellant Budgets**

---

**TABLE 11.2 Example Propellant Budget**

<b>Item</b>	<b>Description</b>	<b>Total</b>	<b>ACS Isp</b>	<b>DV Isp</b>	<b>Fuel Remai ning</b>	<b>Fuel Remai ning</b>	<b>Fuel Remai ning</b>	<b>Fuel Used</b>	<b>Fuel Used</b>	<b>Fuel Used</b>
12	East/West 1997	1163.89	100.00	143.01	103.03	103.30	2.43	0.022	0.022	0.000
13	East/West 1998	1163.85	100.00	143.02	103.00	103.28	2.43	0.022	0.022	0.000
14	North/South 1998	1159.97	100.02	360.00	101.07	101.33	2.43	1.936	1.943	0.000
15	East/West 1998	1159.92	100.00	144.29	101.05	101.31	2.43	0.022	0.022	0.000
16	East/West 1998	1159.88	100.00	144.30	101.02	101.29	2.43	0.022	0.022	0.000
17	East/West 1998	1159.84	100.00	144.31	101.00	101.27	2.43	0.022	0.022	0.000
18	North/South 1998	1155.97	100.02	360.00	99.07	99.33	2.43	1.930	1.936	0.000
19	East/West 1998	1155.93	100.00	145.57	99.05	99.31	2.43	0.022	0.022	0.000
20	East/West 1998	1155.88	100.00	145.58	99.03	99.29	2.43	0.022	0.022	0.000
21	East/West 1998	1155.84	100.00	145.60	99.01	99.27	2.43	0.022	0.022	0.000

Maneuvers 1 through 4 prepare for the AKM firing. They include a spin precession maneuver (SPM) from negative orbit normal (NON) to the AKM burn attitude, a spin-up to the AKM spin rate (needed to stabilize the vehicle) and a small SPM trim. All use of the solid occurs during the AKM burn. This is followed by more attitude maneuvers to prepare for acquisition. Maneuvers 10 and 11 are drift orbit maneuvers designed to get the spacecraft to station. Usually more than two are required. The remaining maneuvers are to correct east/west drift and inclination drift.

The inclination maneuvers use electrothermal hydrazine thrusters (EHT) so the specific impulses for delta-V are much higher. In all cases, the budget accounts for the different specific impulses for the attitude control thrusters and the delta-V thrusters.

---

## Budgets

---

---

## *12.1 Introduction*

The performance of a spacecraft control system is limited by the performance of its sensors and actuators. This chapter discusses the types of sensors that are used in spacecraft control systems. Many of the sensors are used for determining the attitude or attitude rates of the spacecraft. Others are used for determining the relative orientation or position of components on a spacecraft.

---

## *12.2 Types of Sensors*

The following table lists the classes of sensors used in spacecraft. Within each class, there may be many types of sensors.

**TABLE 12.1** Classes of sensors

Sensor	Description	Application
Accelerometer	Measures acceleration on an element.	Inertial acceleration measurements, vibration sensing, nutation measurements.
Angle encoder	Measures either absolute or relative angle of a shaft.	Angle measurements or angular rate measurements.
Earth sensor	Measures the location of the sensor boresight relative to the nadir vector. Does not require the spacecraft to rotate.	Relative earth orientation.

**TABLE 12.1** Classes of sensors

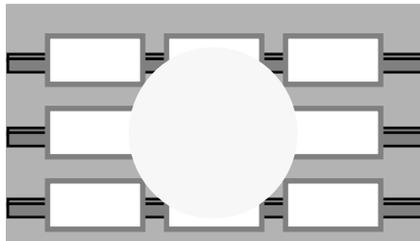
Sensor	Description	Application
Gyro	Any device that measures angular rates.	Rotation angle and rate measurements.
Hall effect	Measures the change in magnetic field.	Used to determine rotor position in a motor.
Horizon sensor	Detects the earth horizon. Requires spacecraft rotation.	Attitude determination for spinning spacecraft.
Magnetometer	Measures the magnitude of a magnetic field.	Attitude determination.
Potentiometer	A variable resistance device in which the resistance varies as it is rotated.	Measurement of angles between rotating spacecraft components.
Rangefinder	Determines the distance between two objects that are not connected.	Docking, orbit maneuvers
Star mapper	Measures the times of the crossing of a star across a detector.	Inertial attitude determination
Star tracker	Measures the position of a star image in the focal plane of the sensor	Inertial attitude determination.
Sun sensor	Either a single axis or two axis measurement of the angle of the sun with respect to the bore-sight of the sensor.	Attitude determination particularly yaw attitude determination. Also used for sun safe modes.

## 12.3 Optical Sensors

### 12.3.1 Star Trackers and Cameras

Modern star trackers and cameras are solid-state video cameras. The tracker takes a picture of the stars and this is converted into a pixel map for use by the flight computer. The star image is slightly defocused so that each star image extends over a block of several pixels. This makes it easier to determine the centroid. A CCD array with rectangular pixels is shown with a defocused star image.

**FIGURE 12.1** CCD Element.



CCD arrays may have square or rectangular pixels. Contrast the above picture with one in which the star was just a point within one pixel. If the above image moves the output of several pixels will change. If the point moves, the output doesn't change until it leaves one pixel and enters another.

Two other types of sensing elements are CID (charge injection device) and APS (advanced pixel sensor.) The first type allows the user to access parts of the array. In addition it does not suffer from blooming problem like a CCD. Blooming is the phenomena in which the charge from one CCD element overflows into others. This will happen when the sun or other bright object appears in the field-of-view of the sensor. The APS sensor is based on CMOS technology, much like a CMOS memory array. It also allows direct access to each element and does not suffer from blooming problems. However, its sensitivity is less than of CCDs.

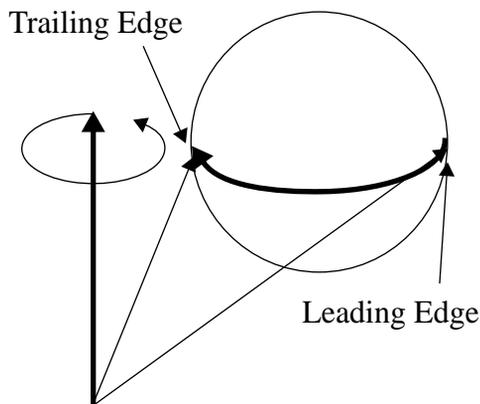
A tracker has a computer in the sensor that computes the attitude. A star camera sends the pixel map to the flight computer.

### 12.3.2 Horizon Sensors

A horizon sensor uses the spacecraft to scan the horizon of a celestial body. The measurement geometry for the earth is shown below.

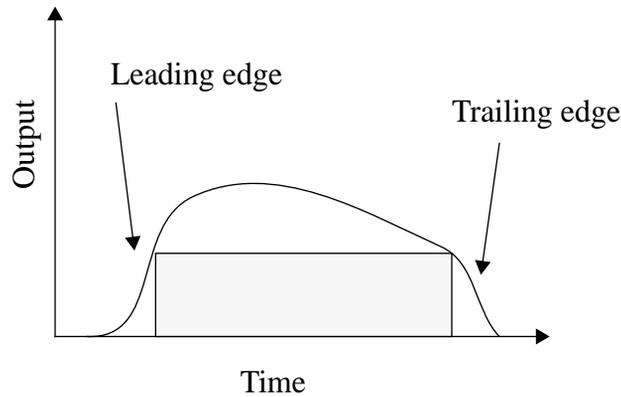
**FIGURE 12.2** Horizon Sensor Measurement Geometry

---



The horizon sensor has a very narrow field of view and responds to the scanning of the earth as shown below. The box shows the ideal "earth pulse".

FIGURE 12.3 Horizon pulse



Several problems can arise. If a bright object is near the main object, the scan will be extended causing the sensor to think it is closer to the equator of the body than it is. A second problem can occur if there is a dim spot on the scanned body. The sensor may then decide the edge of the body has been detected. The shortened chord makes it think the scan is nearer the pole.

Generally, the main problem is detecting the edge. The earth's CO<sub>2</sub> layer is about 40 km above the surface of the earth. The layer is not constant and varies with season of the part of the earth being scanned. For "precise" attitude determination using this kind of sensor, it is necessary to maintain correction factors for seasonal variations in the edge.

### 12.3.3 Earth Sensors

Earth sensors are sensors that detect the edge of the earth without requiring the spacecraft to spin. There are three types. The first use an oscillating mirror to scan the earth. The latter are similar to a horizon sensor. These sensors used the chordwidths to determine the roll attitude and the time from scan center to scan edge to determine pitch.

Scanning sensors measure roll by measuring the earth chord at a specific cant angle and difference the measurements to get an approximation to roll. The measured roll angle for a scanning sensor is

$$\theta_{meas} = 2 \left[ \arccos \left( \frac{\cos(\rho) - \sin(\delta) \sin(\theta)}{\cos(\delta) \cos(\theta)} \right) - \arccos \left( \frac{\cos(\rho) + \sin(\delta) \sin(\theta)}{\cos(\delta) \cos(\theta)} \right) \right] \quad (12.1)$$

where  $\delta$  is the angle of the scan as measured from the plane normal to the sensor scan axis positive about the roll axis,  $\rho$  is the angular radius of the earth, and  $\theta$  is roll.

A major advantage of these sensors is that they give roll and pitch outputs directly. The static earth sensor has sets of thermopiles arranged in a circle about the boresight. Typically each set has three. One looks at the earth all of the

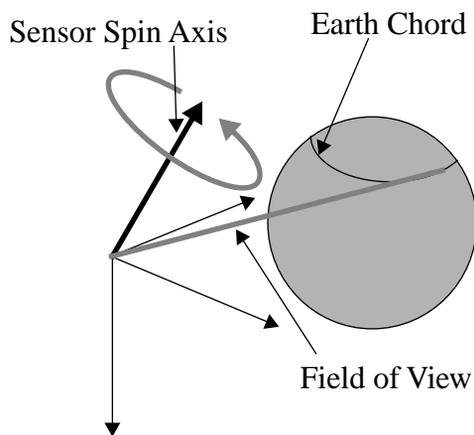
time, the second straddles the earth and the third looks at “cold” space. The former and latter are used to calibrate the straddling sensor. The calibrated temperatures of the straddling sensors are used to compute roll and pitch.

The second detects the edges with a fixed set of sensors in a circular array. This type can use solid sensing elements that ride on the edge of the earth or use an array to determine the edge.

The conical scanning earth sensor has a motor that rotates a mirror or prism that sweeps the field of view of the sensing element across the earth.

**FIGURE 12.4 Earth sensor geometry diagram**

---



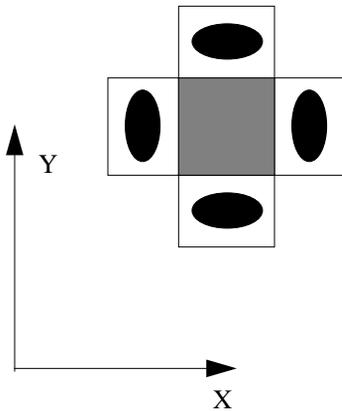
Pitch is found by measuring the center of the chord with respect to an internal reference and roll is found from the length of the chord. If the scan angle is set properly the scan will traverse the northern or southern hemisphere of the earth resulting in a one to one relationship between chordwidth and attitude. This will not be the case for large attitude errors.

### 12.3.4 Analog Sun Sensors

Analog sun sensors are used in pairs to get attitude information. The sensor geometry is illustrated below.

FIGURE 12.5 Sensor geometry

---



Two sensors' boresights are in the  $yz$ -array plane the other two are in the  $xz$ -array plane. The first is canted 45 deg from  $y$  and the other 45 deg from  $+z$ . The difference between the sensors outputs is an indication of the angle. Each sensor measures, to first order, the cosine of the angle between the sun vector and the sensing element.

### 12.3.5 Digital Sun Sensors

Digital sun sensors measure the position of the sun image in the sensor plane. These may use special patterns of thermal detectors in the sensor plane, or may use CCD type elements.

---

## 12.4 Gyros

Gyros are used for two purposes on spacecraft

- Rate measurement
- Integrating attitude between sun or star sensor updates

A gyro output can either be rate, or the integral of rate. In both cases, the output is about the body axes, but the measurement is with respect to the inertial frame. By themselves, gyros cannot give an attitude measurement and are used to supplement other devices.

A simple model that applies to many single-degree-of-freedom gyros is

$$\begin{aligned}\dot{\theta} &= \omega + b + \eta_{\theta} \\ \dot{b} &= \eta_b\end{aligned}\tag{12.2}$$

$\theta$  is the integrated gyro angle;  $\omega$  is the true inertial rate in body axes;  $b$  is the gyro bias;  $\eta_{\theta}$  causes the integrated attitude to random walk;  $\eta_b$  causes the bias to drift. In mechanical gyros  $\eta_{\theta}$  is small and  $\eta_b$  is large. In optical gyros the reverse is true.

---

## 12.5 Other Sensors

### 12.5.1 Magnetometers

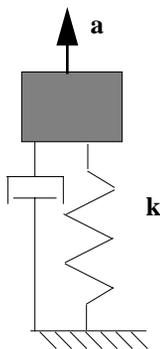
Magnetometers are used for attitude determination and for measuring the magnetic field for adjusting the gain of magnetic torquer systems. Magnetometers measure the total magnetic field and must be kept away from sources of magnetic fields.

### 12.5.2 Accelerometers

Accelerometers measure the acceleration of a mass relative to their base. A simple proof-mass accelerometer is shown in the following figure.

**FIGURE 12.6 Proof-mass accelerometer**

---



The mass is attached to the base via a spring and damper. The transfer function is

$$\frac{s^2}{s^2 + 2\zeta\omega s + \omega^2}\tag{12.3}$$

At low frequencies, the accelerometer has no response. At high frequencies, it has a transfer function of unity.

---

### *13.1 Introduction*

The performance of a spacecraft control system is limited by the performance of its sensors and actuators. This chapter discusses the types of actuators that are used in spacecraft control systems. Many of the actuators are used just for attitude control. Others are used for orbit adjustment and stationkeeping as well as attitude control.

---

### *13.2 Types of Actuators*

The following table lists some actuators used in spacecraft.

**TABLE 13.1** Classes of actuators

Type		Description	Application
Magnetic torquer	Air Coil	Wire wrapped around the frame of the spacecraft to form a planar loop.	Momentum unloading, two-axis control.
	Torquer Bar	Wire wrapped around a magnetic steel core.	Momentum unloading, two-axis control.

**TABLE 13.1**    **Classes of actuators**

Type		Description	Application
Thruster	Hydrazine Monopropellant	Hydrazine is passed through a catalyst bed which causes an exothermic reaction	Attitude and orbit control.
	Bipropellant	An oxidizer and fuel are mixed to produce a chemical reaction	Attitude and orbit control.
	Cold gas	Pressurized gas is expanded through a nozzle.	Attitude control.
	Electrothermal Hydrazine Thrusters (EHT)	Same as hydrazine except that it has a heating coil	Attitude control.
	Arc jet	Same as hydrazine except that it has an electric arc.	Orbit changes.
	Pulsed Plasma Thruster (PPT)	Ionized gas is accelerated in a magnetic field	Attitude and orbit control.
Reaction Wheel		A wheel is attached to the spacecraft via an electric motor. When torques are applied to the spacecraft it reacts.	Attitude control.
Pivoted Wheel		Reaction wheel on a single or two-degree-of-freedom platform	Attitude control.
Control Momentum Gyro		A spinning wheel is attached to the spacecraft through two single-degree-of-freedom hinges. The spin rate is maintained and torques are generated only about the pivot axes.	Slew maneuvers.
Solar paddles		Solar pressure torques are generated by moving the paddles.	Attitude control and momentum unloading.

---

### 13.3 Stepping Motor Model

#### 13.3.1 Introduction

Stepping motors can be of many different types including variable reluctance, permanent magnet, hybrid (a combination of variable reluctance and permanent magnet), claw tooth permanent magnet, among others. A permanent magnetic motor was chosen as an example to demonstrate how the dynamics of the motor may be handled in models.

#### 13.3.2 Electromechanical Dynamics

The equations of motion for a two phase permanent magnet stepping motor are

$$\begin{aligned}
 \frac{di_a}{dt} &= [v_a - Ri_a + K_m \omega \sin(N_r \theta)] / L \\
 \frac{di_b}{dt} &= [v_b + Ri_b + K_m \omega \cos(N_r \theta)] / L \\
 J \frac{d\omega}{dt} &= K_m (i_b \cos(N_r \theta) - i_a \sin(N_r \theta)) - B\omega \\
 \frac{d\theta}{dt} &= \omega
 \end{aligned} \tag{13.1}$$

where  $J$  is the motor inertia,  $v_a$ ,  $v_b$ ,  $i_a$  and  $i_b$  are the phase voltages and currents;  $\theta$  is the shaft angular position;  $\omega$  is the shaft angular velocity;  $B$  is the viscous friction coefficient;  $L$  is the phase winding inductance;  $N_r$  is the number of rotor teeth;  $K_m$  is the motor torque constant and  $R$  is the phase winding resistance. This model neglects magnetic coupling between the phases, the small change in inductance as a function of rotor position, and the variation in inductance with magnetic saturation. It also ignores the detente torque. The last two equations are for a single degree of freedom system which is only possible if the system is rotationally symmetric. The overall motor torque is

$$T = K_m (i_b \cos(N_r \theta) - i_a \sin(N_r \theta)) - B\omega \tag{13.2}$$

which includes both electrical and mechanical torques and is oriented along the shaft axis.

This model has mechanical and electrical time constants. If we assume that the currents instantaneously reach the level specified by the input voltages, then the equations become

$$\begin{aligned}
 J \frac{d\omega}{dt} &= \frac{K_m}{R} ((v_b - K_m \omega \cos(N_r \theta)) \cos(N_r \theta) - (K_m \omega \sin(N_r \theta) - v_a) \sin(N_r \theta)) - B\omega \\
 \frac{d\theta}{dt} &= \omega
 \end{aligned} \tag{13.3}$$

or

$$\begin{aligned}
 J \frac{d\omega}{dt} &= \frac{K_m}{R} (v_b \cos(N_r \theta) + v_a \sin(N_r \theta) - K_m \omega) - B\omega \\
 \frac{d\theta}{dt} &= \omega
 \end{aligned} \tag{13.4}$$

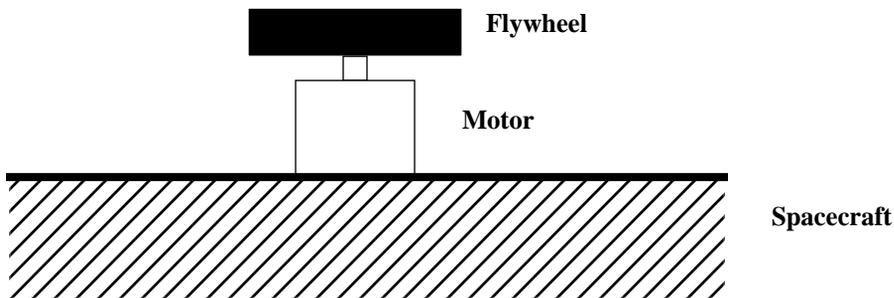
The electrical torque is a nonlinear function of shaft angle and the input voltages. By setting the  $a$  and  $b$  voltages to discrete values in the appropriate sequence, the motor can be made to step. The motor torque constant provides a rate damping torque as well as a drive torque to the system.

## 13.4 Reaction Wheel Model

### 13.4.1 Introduction

Reaction wheels are momentum exchange devices. A motor is fixed to the spacecraft and the shaft of the motor is attached to a flywheel. When a voltage is applied to the motor, the motor generates a torque spinning the wheel in one direction and the spacecraft in the other. Hence the term reaction wheel. Since the torque is internal to the spacecraft, the reaction wheel cannot change the inertial angular momentum of the spacecraft. Rather, it moves momentum from the spacecraft to the flywheel. If the external torque on the spacecraft is cyclic with respect to the inertial frame, then the reaction wheel can completely control the spacecraft. However, if there is a constant inertial torque, the wheel will spin up and eventually saturate.

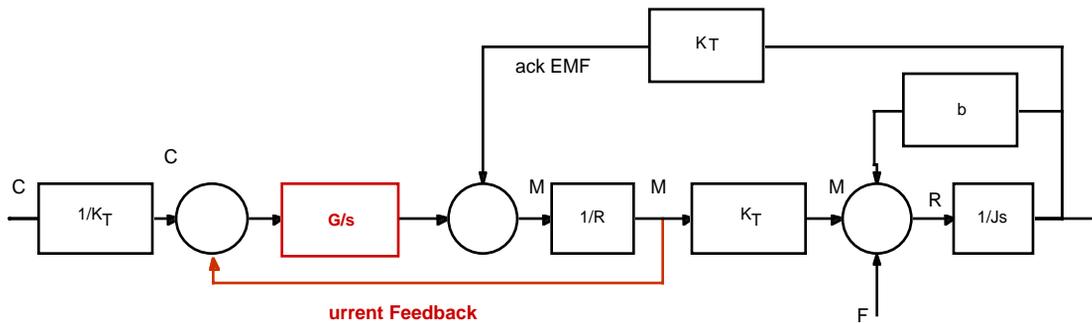
FIGURE 13.1 Reaction Wheel



### 13.4.2 Motor Model

Most reaction wheels are DC motors. A simplified model is shown in the following figure.

FIGURE 13.2 DC Motor Model



The incoming torque command is converted into a current by the first block. The summing junction subtracts the motor current. This current feedback loop compensates for the back electromotive force which is a function of speed. This permits the device to be used as a torque actuator.

The difference between the commanded current and feedback current is integrated and multiplied by a gain. The back emf voltage is subtracted resulting in the motor voltage. The motor voltage is limited by the spacecraft bus voltage. The motor voltage is divided by the motor resistance to get the motor current which is then multiplied by the motor gain to produce the motor torque. The frictional torques are subtracted from this to get the reaction torque. When using the wheel as a torque actuator, the transfer function from commanded torque to reaction torque is of interest. The current loop compensates for the back-emf. To compensate for the friction torques, another loop would be needed that fed back rate. This is often done in robotics where precise torque control is necessary.

The transfer function from commanded torque to reaction torque is

$$\frac{T_R}{T_C} = \frac{G}{R} \frac{s}{s^2 + \left( \frac{b}{J} + \frac{k_T^2}{RJ} + \frac{G}{R} \right) s + \frac{bG}{JR}} \quad (13.5)$$

where  $J$  is the reaction wheel inertia,  $b$  is the viscous damping coefficient,  $R$  is the motor resistance,  $k_T$  is the current to motor torque gain and  $G$  is the forward gain. There is not DC response since when a constant command is supplied the wheel spins up and the viscous friction torque eventually cancels the motor torque. At high frequencies, the RWA gain rolls off as  $1/s$ . At intermediate frequencies, if  $G$  is large, the gain is nearly 1.

### 13.4.3 Reaction Wheel State Equations

There are two states in this model, the wheel speed (a mechanical state) and the integral of the current difference (an electrical state). The state equations are derived as follows. The reaction torque is

$$T_R = k_T i_M - b\omega - T_F \quad (13.6)$$

where  $T_F$  is all nonviscous friction torques. The motor current is

$$i_M = \frac{1}{R}(V_M - k_T\omega) \quad (13.7)$$

The motor torque is

$$V_M = G \int (i_M - i_C) \quad (13.8)$$

The electrical state is

$$\dot{z} = i_M - i_C \quad (13.9)$$

The mechanical state is

$$\dot{\omega} = \frac{T_R}{J} \quad (13.10)$$

The state equations are

$$\begin{bmatrix} \dot{\omega} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -\frac{k_T^2}{RJ} - \frac{b}{J} & \frac{k_T G}{RJ} \\ \frac{k_T}{R} & -\frac{G}{R} \end{bmatrix} \begin{bmatrix} \omega \\ z \end{bmatrix} + \begin{bmatrix} 0 & \frac{1}{J} \\ \frac{1}{k_T} & 0 \end{bmatrix} \begin{bmatrix} T_C \\ T_F \end{bmatrix} \quad (13.11)$$

The electrical dynamics are high frequency and are much higher frequency than the mechanical frequencies. They can be removed from the equations by setting the derivative of  $z$  equal to zero and finding the steady-state value of  $z$  which is

$$z = \frac{G}{R} \left( \frac{T_C}{k_T} + \frac{k_T}{R} \omega \right) \quad (13.12)$$

The RWA can then be modeled with a single state equation. This is the equivalent of saying that  $i_M = i_C$ .

The state equations are linear and do not include voltage and current limits. When implemented in code, this must be included.

### 13.4.4 Tachometer

Most spacecraft use permanent magnet motors. The rotor has poles made up of rare earth magnets. The stator currents are switched based on the angle of the magnets with respect to the stator. The angle is determined by Hall effect sensors that give a pulse when the magnet pole passes over the switch. The time between Hall sensor pulses can be used to determine the wheel rate. Motors can have many poles; 18 sensors are not unusual. Time is computed by running a counter and dumping the count when a Hall sensor signals. If the motor is running fast, one count per cycle is sufficient. If it is running slowly, the count between each Hall sensor is output. If the motor is running sufficiently slow, there may not be an update available between samples. This presents problems for low speed operation. One solution is to use current measurements to estimate rate at low speed. Another is to put an angle encoder on the shaft.

---

## 13.5 Control Moment Gyro

### 13.5.1 Introduction

Reaction wheels are limited in torque capability and momentum storage capability. When bigger torques are needed, for example on a reconnaissance satellite that must track a target on the earth, control-moment gyros (CMGs) are often used. A CMG is a spinning wheel on a gimballed platform. CMGs may have one or two gimbals. The wheel

spin rate is kept constant with a tachometer loop. This degree-of-freedom is not used for control. The gimbal torquer motors are used for control.

### 13.5.2 Modeling

The CMG adds another body to the spacecraft with 2 or more degrees-of-freedom. Generally, the inner gimbal is limited to  $\pm 90$  degrees while the outer gimbal can rotate 360 degrees. Power is supplied to the inner gimbal via slip rings. Any spacecraft with CMGs is a multi-body spacecraft. For control design purposes, however, it is useful to model the CMG drive as a high bandwidth drive in which the gimbal rates achieve a commanded rate instantaneously.

The output torque is then

$$T = Bh \begin{bmatrix} \cos(\alpha) & 0 \\ -\cos(\beta)\sin(\alpha) & -\sin(\beta)\cos(\alpha) \\ -\sin(\beta)\sin(\alpha) & \cos(\beta)\cos(\alpha) \end{bmatrix} \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \end{bmatrix} \quad (13.13)$$

where  $B$  is the transformation matrix from the CMG base to the spacecraft frame. When  $a$  and  $b$  are zero, the CMG momentum, in the CMG frame, is

$$H = \begin{bmatrix} 0 \\ h \\ 0 \end{bmatrix} \quad (13.14)$$

### 13.5.3 Torque Distribution

Each CMG has two control degrees of freedom. At least two CMGs are need for 3-axis control, but even two will have four degrees of freedom. One way to distribute the torque demand is through a pseudo-inverse law which penalizes the squared magnitude of the gimbal rates. If

$$T = A \begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \end{bmatrix} \quad (13.15)$$

then

$$T = \begin{bmatrix} A_1 & A_2 \end{bmatrix} \begin{bmatrix} \dot{\alpha}_1 \\ \dot{\beta}_1 \\ \dot{\alpha}_2 \\ \dot{\beta}_2 \end{bmatrix} \quad (13.16)$$

The gimbal rates are then

$$\begin{bmatrix} \dot{\alpha}_1 \\ \dot{\beta}_1 \\ \dot{\alpha}_2 \\ \dot{\beta}_2 \end{bmatrix} = [A_1 \ A_2]^T \left( [A_1 \ A_2] [A_1 \ A_2]^T \right)^{-1} T \quad (13.17)$$

This torque distribution law only requires the inverse of a 3-by-3 matrix. However, while it attempts to minimize the gimbal rates it does not try and keep the CMGs away from gimbal lock, a condition in which control is lost along one degree-of-freedom. If large CMG motion is expected, a more sophisticated approach will be required.

---

## 13.6 Thrusters

### 13.6.1 Introduction

Small monopropellant and bipropellant thrusters are used for attitude control on many satellites. They are used for attitude control during orbit change maneuvers or for momentum unloading, as an alternative to magnetic torquers. Most small thrusters are on-off thrusters. That is, their valves have only two positions, on and off.

### 13.6.2 Physics of Thrusters

The thrust of a thruster is

$$T = \dot{m} u_e \quad (13.18)$$

where  $T$  is thrust,  $\dot{m}$  is the mass flow rate and  $u_e$  is the exhaust velocity. This equation shows that for a given exhaust velocity, which is fixed by the propellant properties, the mass flow rate must be changed to change the thrust. This would mean a thruster valve with selectable orifice areas. For most applications, this is too complex.

### 13.6.3 Pulsewidth Modulation

An alternative is to control the pulse length of the thruster. If the control period is 1 second, for example, the average thrust can be halved if the pulsewidth is 0.5 sec. The thrust does not change, but the average thrust over the period of control is halved. This is done at the expense of adding high frequency control activity. For example, if a constant thrust of 1/2 the thruster thrust is desired, the actual thrust will be a square wave with a 50% duty cycle and a cycle period equal to the control period. Thus, the actual output of the thruster will have significant high frequency harmonics and might excite vibratory modes in the system.

### 13.6.4 Minimum Impulse Bit

A major problem is that due to the dynamics of the valve, all thrusters have minimum impulse bits. Generally, 16 ms is the minimum on-time for a thruster. If the control period is 1 second, that gives an effective throttle ratio of 62.5:1. That is the equivalent of a 6 bit data word! In contrast, an audio CD uses 16 bit data words! This can be improved by using a multi-rate control system. The digital controller runs with a period of 1 second, but the output is sampled less frequently. For example, the output might be sampled every 8 control periods. This improves the control resolution to 9 bits.

When designing a control system it is important to include the effects of undersampling. The easiest way to do this is to add a zero-order hold with a duration of the minimum impulse bit, followed by a delay of length equal to the period between output samples.

### 13.6.5 Time Constants

Thrusters with solenoid valves can be modeled as first order systems with separate rise and fall time constants. When fuel is flowing into the thruster the model is

$$\dot{u} + \frac{u}{\tau_r} = 1 \quad (13-1)$$

when no fuel is flowing the model is

$$\dot{u} + \frac{u}{\tau_f} = 0 \quad (13-2)$$

The thrust is

$$T = ku \quad (13-3)$$

where  $k$  is the maximum thrust.  $k$  is

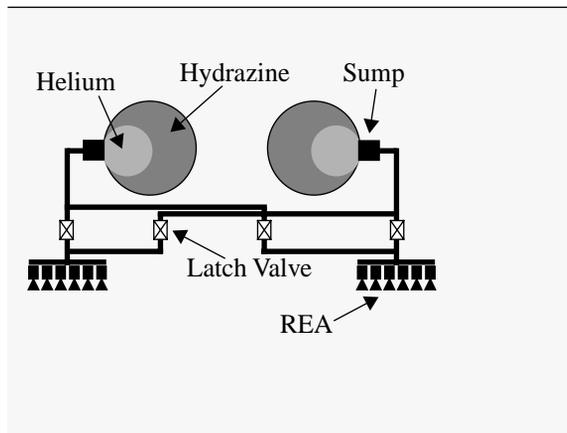
$$k = C_F A^* P \quad (13-4)$$

where  $P$  is the pressure in the REA combustion chamber,  $C_F$  is the thrust coefficient and  $A^*$  is the throat area.

### 13.6.6 Fuel System

A typical fuel system is shown below . The system has two fuel volumes (that may consist of several tanks) with

**FIGURE 13.3 Tank Geometry**



Hydrazine fuel pressurized by Helium. The systems are cross strapped with a total of four latch valves. Each half system is attached to six REAs.

If the system is unregulated, the pressure in each system is

$$P = \frac{m_{\text{He}} R_{\text{He}} T}{V - \frac{m_{\text{Hydrazine}}}{\rho_{\text{Hydrazine}}}} \quad (13.19)$$

where  $m_{\text{He}}$  is the mass of Helium,  $R_{\text{He}}$  is the gas constant of Helium

$$R_{\text{He}} = \frac{8.3}{0.004} \quad (13.20)$$

$T$  is the gas temperature,  $V$  is the system volume (including any auxiliary pressurant tanks),  $m_{\text{Hydrazine}}$  is the mass of the fuel and  $\rho_{\text{Hydrazine}}$  is the density of the fuel

## 13.7 Magnetic Torquers

Magnetic torquers are either air coil or torquer bar. Torquers generate torque through interaction with the earth's magnetic field

$$T = M \times B \quad (13.21)$$

where  $B$  is the magnetic field and  $M$  is the dipole of the torquer. For an air coil, the dipole is

$$M = NIA \quad (13.22)$$

where  $N$  is the number of turns of wire,  $I$  is the current in the wire and  $A$  is the area of the coil. Typically, the wire is wrapped around the corners of the spacecraft structure.

Torquer bars use magnetic steel to concentrate the flux, getting the same dipole in a smaller volume. However, the magnetic steel can saturate, limiting the dipole. In addition, the magnetic steel will always have some small residual dipole when off.

---

## *13.8 Solenoids*

### **13.8.1 Introduction**

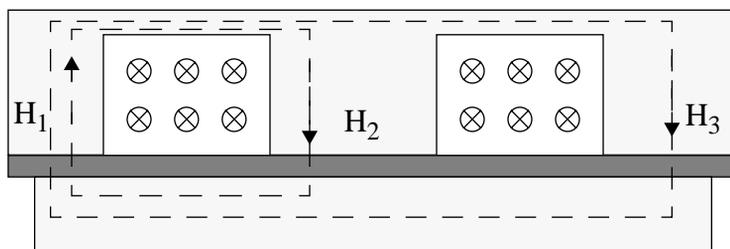
Solenoids are electromechanical devices that are used in many types of spacecraft systems. This section derives the equations of motion for



the third is to the right of coil 3. Only two paths are required to relate the fields. These are shown in Figure 13.4. The

**FIGURE 13.5 Contours for integrating the magnetic field**

---



two integrals are

$$\begin{aligned} gH_1 + gH_2 &= Ni_1 \\ gH_1 + gH_3 &= N(i_1 + i_2) \end{aligned} \quad (13.24)$$

A third relationship is needed to solve for the fields. The surface integral of the magnetic flux over region in which there are no sources must be zero, or

$$\oint_S \mathbf{B} \cdot \mathbf{n} da = 0 \quad (13.25)$$

This surface integral is over a volume, unlike the one in Equation [13.23]. This results in

$$\mu_o c((d-x)H_1 - 2dH_2 - (d+x)H_3) = 0 \quad (13.26)$$

Substituting for  $H_2$  and  $H_3$  gives

$$\begin{aligned} H_1 &= \frac{N}{4g} \left( 2i_1 + \left( 1 + \frac{x}{d} \right) (i_1 + i_2) \right) \\ H_3 &= \frac{N}{4g} \left( 2i_2 + \left( 1 - \frac{x}{d} \right) (i_1 + i_2) \right) \end{aligned} \quad (13.27)$$

To find the flux linked by each of the coils, it is necessary to find the flux through the volume within each of the coils. For each coil this is

$$\begin{aligned} \phi_1 &= \mu_o cd \left( 1 - \frac{x}{d} \right) H_1 \\ \phi_2 &= \mu_o cd \left( 1 + \frac{x}{d} \right) H_3 \end{aligned} \quad (13.28)$$

Each flux is linked  $N$  times by each coil, so that total flux linked becomes

$$\begin{aligned}\lambda_1 &= \frac{\mu_o N^2 cd}{4g} \left( \left( 3 - 2\frac{x}{d} - \frac{x^2}{d^2} \right) i_1 + \left( 1 - \frac{x^2}{d^2} \right) i_2 \right) \\ \lambda_2 &= \frac{\mu_o N^2 cd}{4g} \left( \left( 1 - \frac{x^2}{d^2} \right) i_1 + \left( 1 + 2\frac{x}{d} - \frac{x^2}{d^2} \right) i_2 \right)\end{aligned}\tag{13.29}$$

The inductances are

$$\begin{aligned}L_{11} &= L_o \left( 3 - 2\frac{x}{d} - \frac{x^2}{d^2} \right) \\ L_{22} &= L_o \left( 1 + 2\frac{x}{d} - \frac{x^2}{d^2} \right) \\ L_{12} &= L_o \left( 1 - \frac{x^2}{d^2} \right) \\ L_o &= \frac{\mu_o N^2 cd}{4g}\end{aligned}\tag{13.30}$$

The terminal voltages are

$$v = iR + \frac{d\lambda}{dt}\tag{13.31}$$

or

$$\begin{aligned}v_1 &= i_1 R + L_{11} \frac{di_1}{dt} + L_{12} \frac{di_2}{dt} - \frac{2L_o}{d} \left( \left( 1 + \frac{x}{d} \right) i_1 + \left( \frac{x}{d} \right) i_2 \right) \frac{dx}{dt} \\ v_2 &= i_2 R + L_{12} \frac{di_1}{dt} + L_{22} \frac{di_2}{dt} - \frac{2L_o}{d} \left( \left( \frac{x}{d} \right) i_1 + \left( \frac{x}{d} - 1 \right) i_2 \right) \frac{dx}{dt}\end{aligned}\tag{13.32}$$

The electrical force is

$$f_e = \frac{\partial}{\partial x} W_m'\tag{13.33}$$

where  $W_m'$  is the coenergy. For this example

$$W_m' = \int_0^{i_1} \lambda_1 di_1 + \int_0^{i_2} \lambda_2 di_2\tag{13.34}$$

This becomes

$$W_m' = \frac{1}{2}L_{11}i_1^2 + 2L_{12}i_1i_2 + \frac{1}{2}L_{22}i_2^2 \quad (13.35)$$

and the force on the plunger is

$$f_e = \frac{1}{2}\frac{\partial L_{11}}{\partial x}i_1^2 + 2\frac{\partial L_{12}}{\partial x}i_1i_2 + \frac{1}{2}\frac{\partial L_{22}}{\partial x}i_2^2 \quad (13.36)$$

or

$$f_e = \frac{L_o}{d}\left(\left(1 - \frac{x}{d}\right)i_2^2 - 4\left(\frac{x}{d}\right)i_1i_2 - \left(1 + \frac{x}{d}\right)i_1^2\right) \quad (13.37)$$

$$f_e = \frac{L_o}{d}\left(i_2^2 - i_1^2 - \left(\frac{x}{d}\right)(i_2^2 + i_1^2 + 4i_1i_2)\right)$$

The equation of motion for the mechanical part of the system is

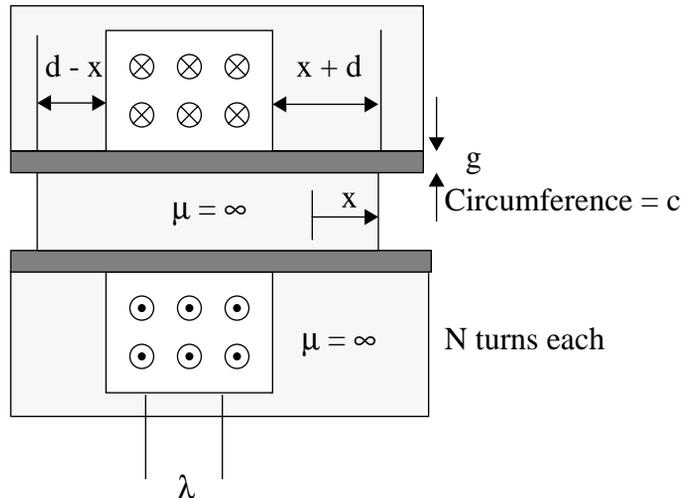
$$M\frac{d^2x}{dt^2} + c\frac{dx}{dt} + kx + b\frac{dx/dt}{|dx/dt|} = \frac{L_o}{d}\left(i_2^2 - i_1^2 - \left(\frac{x}{d}\right)(i_2^2 + i_1^2 + 4i_1i_2)\right) \quad (13.38)$$

The first term is the inertial acceleration, the second is viscous damping, the third is linear spring stiffness and the fourth is Coulomb friction. If the currents are balanced and constant the last term in the coupling term provides stiffness to the system. The mechanical equations are linear in  $x$  and the electrical equations are linear in  $i$ , but the coupled equations are nonlinear in both  $x$  and  $i$ .

### 13.8.3 Derivation of the equations of motion for a single coil solenoid

The solenoid is illustrated below in Figure 13.4.

FIGURE 13.6 Single Coil Solenoid



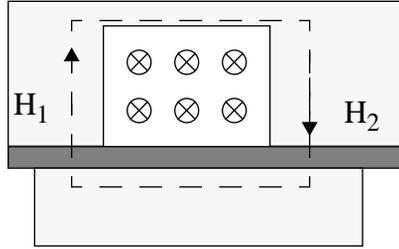
The first step is to find the H fields. The H fields are significant only in the sleeve where the permeability is that of free space. In both the plunger and the enclosure, the permeability is assumed to be infinite hence the fields are nearly zero. The magnetic fields are found from the equation

$$\oint_C \mathbf{H} \cdot d\mathbf{l} = \int_S \mathbf{J} \cdot n d\mathbf{a} \quad (13.39)$$

This equation means that if you integrate the magnetic fields on a closed path around a surface S, the integral will equal the sum of all of the currents that flow normal to the surface and are within the path. In the above drawing, there are three possible paths. One surrounds just coil 1, one surrounds just coil 2, and one surrounds both coils. There are also three regions in which the magnetic field can be found. One is to the left of coil 1, one is between coil

1 and 2 and the third is to the right of coil 3. Only two paths are required to relate the fields. These are shown in Figure 13.4. The two integrals are

**FIGURE 13.7** Contours for integrating the magnetic field



$$gH_1 + gH_2 = Ni \quad (13.40)$$

A third relationship is needed to solve for the fields. The surface integral of the magnetic flux over region in which there are no sources must be zero, or

$$\oint_S \mathbf{B} \cdot \mathbf{n} da = 0 \quad (13.41)$$

This surface integral is over a volume. This results in

$$\mu_o c((d-x)H_1 - (d+x)H_2) = 0 \quad (13.42)$$

Substituting for  $H_2$  gives

$$H_2 = \frac{N}{g} \left(1 - \frac{x}{d}\right) i \quad (13.43)$$

To find the flux linked by the coil, it is necessary to find the flux through the volume within the coil. This is

$$\phi = \mu_o c d \left(1 + \frac{x}{d}\right) H_2 \quad (13.44)$$

Each flux is linked  $N$  times by the coil so the that total flux linked becomes

$$\lambda = \frac{\mu_o c d N^2}{g} \left(1 - \frac{x^2}{d^2}\right) i \quad (13.45)$$

The inductances are

$$L = L_o \left( 1 - \frac{x^2}{d^2} \right) \tag{13.46}$$

$$L_o = \frac{\mu_o N^2 c d}{g}$$

The terminal voltage is

$$v = iR + \frac{d\lambda}{dt} \tag{13.47}$$

or

$$v = iR + L \frac{di}{dt} - \frac{2L_o}{d} \left( \frac{x}{d} \right) i \frac{dx}{dt} \tag{13.48}$$

The electrical force is

$$f_e = \frac{\partial}{\partial x} W_m' \tag{13.49}$$

where  $W_m'$  is the coenergy. For this example

$$W_m' = \int_o^i \lambda di \tag{13.50}$$

This becomes

$$W_m' = \frac{1}{2} Li^2 \tag{13.51}$$

and the force on the plunger is

$$f_e = \frac{1}{2} \frac{\partial Li^2}{\partial x} \tag{13.52}$$

or

$$f_e = -\frac{L_o}{d} \left( \frac{x}{d} \right) i^2 \tag{13.53}$$

The equations of motion for the system are

$$M \frac{d^2 x}{dt^2} + k(x + x_o) + b \frac{dx/dt}{|dx/dt|} = -\frac{L_o}{d} \left(\frac{x}{d}\right) i^2 \quad (13.54)$$

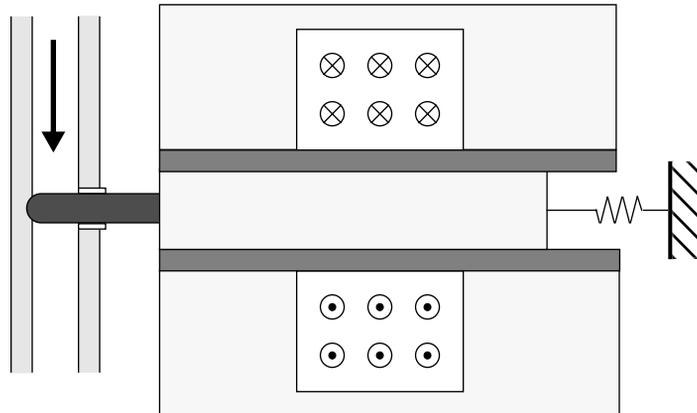
$$v = iR + L \frac{di}{dt} - \frac{2L_o}{d} \left(\frac{x}{d}\right) i \frac{dx}{dt}$$

The first term is the inertial acceleration, the second is linear spring stiffness, and the third is Coulomb friction. When the current is off and motion has damped the damper will sit at  $-x_o$ . When current is introduced the equation of motion will be

$$M \frac{d^2 x}{dt^2} = \frac{L_o}{d} \left(\frac{x_o}{d}\right) i^2 \quad (13.55)$$

which will push the solenoid in the  $+x$  direction. This setup will work well as a valve driver.

**FIGURE 13.8** Valve driver





---

### *14.1 Introduction*

Spacecraft designers rely on simulation to verify their designs prior to flight. Unlike other areas of engineering, it is rarely possible to test a prototype in a realistic environment. Consequently, numerical simulation is of primary importance.

When we talk about simulation, it is convenient to break it into two categories, linear and nonlinear. Spacecraft dynamics are inherently nonlinear, and most satellite actuators and sensors are nonlinear. Nonetheless, it is usually possible to linearize the dynamics and devices about some operating point where, in a sufficiently restricted region, the system behaves linearly. This is the basis for the linear control laws developed in this toolbox. Occasionally, for example during the dual-spin-turn, we take advantage of nonlinearities, but for the most part we avoid them.

When dealing with the linearized system, we always work with a state-space description of the form:

$$\begin{aligned}\dot{x} &= ax + bu \\ y &= cx + du\end{aligned}\tag{14.1}$$

$a$  is the plant matrix that relates the state  $x$  to its derivative;  $b$  is the input matrix that relates the inputs, such as external disturbances and control inputs, to the state derivative;  $y$  is the measured output of the system;  $c$  relates the state to the measurements and  $d$  provides for direct feedthrough of inputs into the measurements. Often  $d$  is omitted in textbooks, but actually it appears frequently in practice. For example the state space description of the system

$$\dot{y} = y + \phi\tag{14.2}$$

is

$$\begin{aligned}\dot{x} &= x + u \\ y &= x + u \\ u &= \phi\end{aligned}\tag{14.3}$$

where  $d = I$ . Sometimes the state is completely measured and  $c$  is the diagonal matrix. In that case, the measurement equation is often omitted completely.

Some control design tools cannot handle systems with a non-zero  $d$  matrix.

---

## 14.2 Linear

### 14.2.1 Creating a State Space System

Given a system of the form

$$\begin{aligned}\dot{x} &= f(x, u, t) \\ y &= g(x, u, t)\end{aligned}\tag{14.4}$$

it can be linearized about  $(x_0, u_0, t_0)$  by expanding the function in a Taylor Series to first order

$$\begin{aligned}\dot{x} &= \left. \frac{d}{dx} f(x, u, t) \right|_{(x_0, u_0, t_0)} x + \left. \frac{d}{du} f(x, u, t) \right|_{(x_0, u_0, t_0)} u + f(x_0, u_0, t_0) \\ y &= \left. \frac{d}{dx} g(x, u, t) \right|_{(x_0, u_0, t_0)} x + \left. \frac{d}{du} g(x, u, t) \right|_{(x_0, u_0, t_0)} u + g(x_0, u_0, t_0)\end{aligned}\tag{14.5}$$

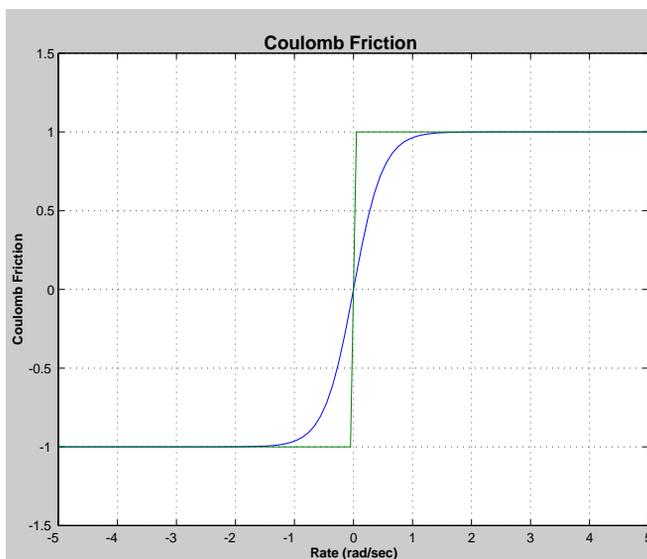
or

$$\begin{aligned}\dot{x} &= ax + bu + f_0 \\ \delta y &= cx + du\end{aligned}\tag{14.6}$$

If the equations are linearized about an equilibrium state, then  $f_0$  is zero.

A major concern is that  $f$  and  $g$  be smooth around the linearization point. An example where this is not true (and is often encountered) is when the system includes a reaction wheel with Coulomb friction. Coulomb friction changes from  $-T_c$  to  $+T_c$  instantaneously at zero rate. This is shown in the following plot.

**FIGURE 14.1 Coulomb Friction**



**LISTING 14.1 Coulomb Friction Code**

```
w      = linspace(-5,5);
fS     = 2./(1 + exp(-4*w)) - 1;
fC     = [-ones(1,50)ones(1,50)];
```

For this reason, it is generally better to approximate Coulomb friction as a function that passes through zero with a finite slope.

### 14.2.2 Zero Order Hold

The simplest way to simulate a continuous time system is to discretize it using the zero order hold. One is the standard zero order hold. The second is the delta form of the zero order hold. These approximations assume that the input is held constant over the interval  $T$ .

This approximation is satisfactory for most spacecraft control design problems.

### 14.2.3 First Order Hold

An alternative approach is to discretize the system using a first order hold. This approximation assumes that the input varies linearly from step  $k$  to step  $k + 1$ .

---

### 14.3 Nonlinear

Nonlinear systems require numerical integration. This usually means by some type of Runge-Kutta integrator. The system is formulated as

$$\begin{aligned} \dot{x} &= f(x, u, t) \\ y &= g(x, u, t) \end{aligned} \tag{14.7}$$

and  $f$  is called by the integrator. All continuous models must be included in  $f$ . Discrete models, for example the digital control system, are outside of the numerical integration routine.

---

## *15.1 Introduction*

This chapter discusses torque and force disturbances on a spacecraft. Disturbances may be internal or external. External disturbances must be countered by actuators and are used to size actuators. The ultimate performance of the spacecraft will depend on the magnitude and direction of the disturbances.

---

## *15.2 External Disturbances*

### **15.2.1 Introduction**

The following table lists major external disturbances on spacecraft.

**TABLE 15.1 Disturbances**

Type	Source	Characteristics
Aerodynamic	Planetary atmosphere	Drag and lift on the spacecraft. A strong function of altitude and position relative to the sun.
Albedo Pressure	Sun reflection from the earth	Average is 0.34 solar flux for the earth. Varies depending on the latitude and longitude of the spacecraft, season, etc.

**TABLE 15.1 Disturbances**

Type	Source	Characteristics
Gravity gradient	Inertia	Due to the distributed nature of a satellite. If the satellite has off-diagonal inertia terms it will produce a body fixed torque for a planet pointing spacecraft. The diagonal terms can cause a gravity-gradient modal frequency.
Leaks	Onboard gas and liquid supplies	Acts like a cold gas thruster
Outgassing	Moisture embedded in the structure.	Caused by the heating of surfaces and resulting emission of gas. This often happens with composite solar arrays and can produce large forces.
Thruster Plumes	Interaction of rocket exhaust plumes and the structure	
Radiation Pressure	Temperature of the planet	Black body radiation. 400 W for the earth.
Residual Dipole	Residual dipole on the spacecraft	This torque is caused by the interaction of an external magnetic field and internal dipoles due to current loops, etc.
RF	Transmit antennas	Radiation pressure. Divide the wattage by the speed of light to get pressure.
Solar Pressure	Sun	The disturbance torques and forces are at harmonics of orbit rate for a planet pointing satellite. The solar flux is 1358 W at the earth orbit.
Thermal Pressure	Radiators	Heat radiates diffusely from all radiators. This produces a thrust proportional to the heat flux.

External disturbances can change the angular momentum of the spacecraft. Some of these are discussed in the following sections.

### 15.2.1 Gravity Gradient

The gravity gradient disturbance torque arises from the off-diagonal terms in the inertia matrix. The vector torque is

$$T = 3\omega_o^2 \begin{bmatrix} -I_{yz} \\ I_{xz} \\ 0 \end{bmatrix}$$

The off-diagonal terms will include contributions from the core and the solar arrays. Ideally, the solar arrays would have very small off-diagonal terms being essentially flat plates, but thermal distortion will warp them, introducing asymmetries. The main diagonal inertia terms produce small restoring torques that are functions of attitude.

### 15.2.2 Residual Dipole

The residual dipole disturbances comes from the interaction of the magnetic fields generated by current loops on the spacecraft and solar arrays with the earth's magnetic field. The solar arrays and core spacecraft must be accounted for separately. The torque due to residual dipole is

$$T = (M_C + C_N M_N + C_S M_S)^\times B \quad (15.1)$$

where  $B$  is the earth's magnetic field resolved into the body frame,  $C_N$  is the transformation matrix that transforms from the north array frame to the core frame,  $C_S$  is the south transformation matrix, and  $M_i$  is the dipole for the  $i$ th body. Since  $C_N$  and  $C_S$  rotate their respective dipole at the solar day rate, the residual dipole disturbance will have a bias and a solar day rate term. Typically, each of the dipoles is  $5 \text{ ATM}^2$ , and the earth's field magnitude is about  $90 \text{ nT}$ , and so each term contributes a torque of  $0.45 \text{ }\mu\text{Nm}$ .

### 15.2.3 RF

Transmitting antennas will produce torques. The torque on the spacecraft will be

$$T = -\frac{P}{c} r^\times u \quad (15.2)$$

where  $u$  is in the direction opposite the antenna boresight,  $r$  is the vector to the antenna boresight from the spacecraft center-of-mass, and  $c$  is the speed of light ( $3 \times 10^8 \text{ m/sec}$ ).

Internal RF transmissions, such as between the feeds and reflectors or in waveguides, do not produce a net torque on the spacecraft. Generally, RF poses a problem with high power satellites with offset reflectors. For example, a transmitting antenna with an output of 600 watts offset 3 m from the spacecraft center of mass will generate a  $6 \text{ }\mu\text{Nm}$  torque!

### 15.2.4 Solar Pressure

Solar pressure is the dominant disturbance on a spacecraft in geosynchronous orbit. Solar pressure is due to the force of photons on the surfaces of the spacecraft. A photon striking a surface can do one of four things: it can be absorbed, it can reflect specularly (meaning at the same angle with respect to the surface normal that it hit), it can reflect diffusely (meaning at any angle), or it can pass through the surface. Photons that are absorbed must either be transferred somewhere else (through heat conduction or as electricity in the case of a solar array) or be reemitted locally. If the latter happens, the photon must be lumped in with the diffusely reemitted photons. In terms of fractions of the incoming photons, the following is true

$$1 = \rho_a + \rho_s + \rho_d + \rho_t \quad (15.3)$$

where  $\rho$  stands for the fraction of photons that are absorbed, specularly reflected, diffusely reflected or transmitted.

The solar pressure force is

$$F = -SA_s^T n (2(\rho_s s^T n + \rho_d/3)n + (\rho_a + \rho_d)s) \quad (15.4)$$

where  $s$  is the sun vector,  $n$  is the unit normal to the surface,  $A$  is the area of the surface and  $S$  is the solar flux in  $\text{N/m}^2$ . If the solar array tracks the sun this becomes

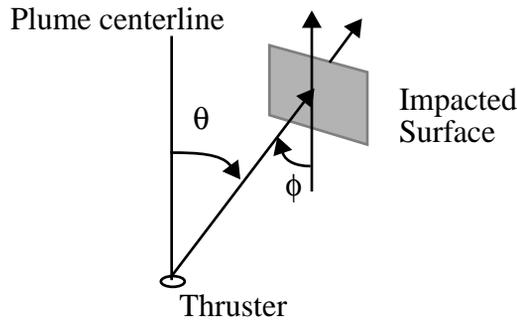
$$F = -SA \left( \rho_a + \frac{4}{3}\rho_d + 2\rho_s \right) s \quad (15.5)$$

The specular component produces the biggest force, the diffuse the second largest and the absorbed produces the smallest force contribution.

### 15.2.5 Thruster Plumes

Thruster plumes cause disturbance torques and forces on the spacecraft when they impact spacecraft surfaces. Plume models are of two types, near-field and far-field. The former apply near the thruster nozzle when the exhaust gas can still be modeled as a fluid. The latter model is applicable when the gas is best modeled as individual particles. The plume geometry is illustrated below.

**FIGURE 15.1 Plume Geometry Diagram**



## 15.3 Internal Disturbances

### 15.3.1 Introduction

The following table lists major internal disturbances on spacecraft. Internal disturbances cause jitter.

**TABLE 15.2 Disturbances**

Type	Source	Characteristics
Dynamic imbalance	Rotating component with off-diagonal inertia terms.	Periodic with the rotation rate.
Friction	Rotating or sliding parts	Has three major components: stiction, or starting friction; Coulomb friction which is independent of rate; and viscous which is proportional to rate. Turbulent damping is proportional to rate squared.
Motor torques and forces	Motors	Articulating components.
Moving parts	Dynamics of internal moving parts.	Torques are proportional to the square of relative rates.
Rotors	Momentum of rotors	Gyroscopic torques.
Rotor imbalances	Center-of-mass offset of a rotating component.	Periodic with the rotation rate.
Slosh	Liquids	Movement of liquids within the spacecraft. Important for liquid fueled upper stages.
Thermal snap	Sudden structural deformation	Caused by rapid heating or cooling.

---

## 15.4 Fourier Series Representation

Disturbances can often be put in the form

$$T = a_0 + \sum_k a_k \cos(knt + \beta) + b_k \sin(knt + \beta) \quad (15.1)$$

where  $n$  is the orbit period. This is convenient for long duration studies, particularly of momentum control systems.



---

## 16.1 Introduction

This chapter describes how to build attitude estimators.

---

## 16.2 Introduction to Estimation Theory

Given a physical system that can be modeled in the form

$$\dot{x} = f(x, u, t) \tag{16.1}$$

We would like to get the best possible estimate of  $x$ .

Let's look at a very simple physical system in continuous time

$$\begin{aligned} \dot{x} &= -ax + u \\ y &= x \end{aligned} \tag{16.2}$$

Only one parameter,  $a$ , and the equations are first order and linear. The input is  $u$  and the state is  $x$ . The measurement is  $y$ , which in this case exactly equals the state.

A simple estimator is

$$\dot{x}_E = ax_E + u_E + L(y_M - y_E) \quad (16.3)$$

The subscript  $E$  means estimate.  $L$  is a gain. In effect,  $L$  times the difference between the estimated measurement and measurement produces an input, added to  $u_E$  that pushes the estimate,  $x_E$  towards  $x$ .

The transfer function between the estimated  $x$  and the measurement and estimated input is

$$x_E = \frac{Ly_M}{s + (L + a)} + \frac{u_E}{s + (L + a)} \quad (16.4)$$

Or a first order low pass filter with a DC gain of  $L/(L+a)$ . The filter has the same dynamics as the plant. If  $L = 0$  just propagate the estimated inputs. If  $L$  is large ignore the inputs and just use the measurements.

The estimator assumes that little valid information about  $x$  is available above  $L + a$  since the state does not respond to signals above  $a$ . How do we get  $L$ ?

If we know what  $u$  to expect we would pick  $L$  so that frequencies beyond those contained in  $u$  would be attenuated. Given a knowledge of the measurement noise and the uncertainty in the plant we can optimally pick  $L$  using quadratic estimator theory. The optimal gain is found from the equation

$$\begin{aligned} \dot{p} &= -2ap + q - \frac{p^2}{r} \\ L &= \frac{p}{r} \\ E[v(t)v^T(t')] &= r\delta(t-t') \\ E[w(t)w^T(t')] &= q\delta(t-t') \end{aligned} \quad (16.5)$$

If we assume the model

$$\begin{aligned} \dot{x} &= -ax + u + w \\ y &= x + v \end{aligned} \quad (16.6)$$

where  $w$  and  $v$  are white noise processes;  $\delta$  has units of seconds;  $q$  represents uncertainty in the plant model and in the inputs  $u$ ;  $r$  represents measurement noise.

In steady state (as  $t$  approaches  $\infty$ )

$$L = \sqrt{a^2 + q/r} - a \quad (16.7)$$

In the limit

$$L = \begin{pmatrix} 0 & r/q \rightarrow \infty \\ \sqrt{q/r} & q/r \rightarrow \infty \end{pmatrix} \quad (16.8)$$

For very noisy signals or for perfect knowledge of the plant, set  $L$  to zero.

This can also implement as a time varying filter. The time varying gain is

$$L(t) = \beta - a + \frac{2\beta}{\left(\frac{p(0) + r(\beta + a)}{p(0) - r(\beta - a)}\right)e^{2\beta t} - 1} \quad (16.9)$$

$$\beta = \sqrt{a^2 + q/r}$$

$L(0) = \frac{p(0)}{r}$ , the bigger the uncertainty in  $x$  initially, the bigger the gain. This gain is independent of the measurements or the state. The steady-state value is independent of  $p(0)$ . There given  $a$ ,  $q$  and  $r$ ; it will always converge to the same value. This will be true about all time varying filters derived this way including Extended Kalman Filters.

The gain is dependent on the ratio between the plant and measurement uncertainty and the plant parameters. If a time-varying filter is used, the gain will always vary in the same way, given the same initial covariance. The white noise approximation for the plant noise is usually not very good. It is often necessary to augment the state equations with additional equations modeling the plant and disturbance uncertainty. The ultimate form of the estimator is a first low pass filter, except that an estimate of the disturbance is added. If this is unavailable the estimator is a first order low pass filter.

---

### 16.3 Gyro Model

A simple model that applies to many single-degree-of-freedom gyros is

$$\begin{aligned} \dot{\theta} &= \omega + b + \eta_{\theta} \\ \dot{b} &= \eta_b \end{aligned} \quad (16.10)$$

$\theta$  is the integrated gyro angle.  $\omega$  is the true inertial rate in body axes.  $b$  is the gyro bias.  $\eta_{\theta}$  causes the integrated attitude to random walk.  $\eta_b$  causes the bias to drift. In mechanical gyros,  $\eta_{\theta}$  is small and  $\eta_b$  is large. In optical gyros, the reverse is true. This is a continuous model. Assume that you have a measurement of  $\theta$  such that

$$y = \theta + \eta_y \quad (16.11)$$

You can then use a Kalman Filter to come up with an optimal estimate of  $b$  and  $\theta$ . While it is not normally feasible to implement a continuous Kalman Filter in practice, it is instructive to look at the covariance matrix.

The Riccati equation is

$$\dot{P} = FP + PF^T - PH^TR^{-1}HP + Q \quad (16.12)$$

where  $F$  is the state matrix

$$F = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad (16.13)$$

and  $H$  is the measurement equation

$$H = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad (16.14)$$

$Q$  is the spectral density matrix for the plant and  $R$  is the spectral density matrix for the measurement. The units of  $R$  are  $\text{rad}^2\text{-sec}$  and the units for  $Q$  are for (1,1)  $\text{rad}^2/\text{sec}$  and  $\text{rad}^2/\text{sec}^3$  for (2,2). In this case,  $R$  is a scalar. Expanding the covariance equation with the derivative set to zero gives

$$\begin{bmatrix} p_{12} & p_{22} \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} p_{12} & 0 \\ p_{22} & 0 \end{bmatrix} - \frac{1}{r} \begin{bmatrix} p_{11}^2 & p_{11}p_{12} \\ p_{11}p_{12} & p_{12}^2 \end{bmatrix} + \begin{bmatrix} q_{11} & 0 \\ 0 & q_{22} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (16.15)$$

Writing out the scalar equations gives

$$\begin{aligned} 2p_{12} + q_{11} - \frac{p_{11}^2}{r} &= 0 \\ p_{22} &= \frac{p_{11}p_{12}}{r} \\ p_{12} &= \sqrt{q_{22}} \end{aligned} \quad (16.16)$$

Solving

$$\begin{aligned} p_{11} &= \sqrt{r(q_{11} + 2\sqrt{q_{22}})} \\ p_{22} &= \sqrt{\frac{q_{22}(q_{11} + 2\sqrt{q_{22}})}{r}} \\ p_{12} &= \sqrt{q_{22}} \end{aligned} \quad (16.17)$$

If there were no plant noise, i.e. if  $q_{11} = q_{22} = 0$ , the steady state covariance matrix would go to zero.

Usually, data for gyros is provided as spectral densities. Thus, these equations can be used directly to determine the covariances of the estimates in steady-state. They provide a useful check for the discrete time Kalman Filter Equations.

---

### 16.4 Conversion of Continuous Time Data to Discrete

The continuous time data assumes continuous measurements. In practice, the sensors are sampled. Converting the continuous spectral densities to covariances is done by

$$\begin{aligned} Q_k &= Q\tau \\ R_k &= \frac{R}{\tau} \end{aligned} \tag{16.18}$$

where  $\tau$  is the sampling time. The units of  $Q_k$  and  $R_k$ , in this case become  $\text{rad}^2$  and  $(\text{rad}/\text{sec})^2$ .

---

### 16.5 Simulation of Stochastic Processes

Simulations are done almost exclusively on digital computers, and require discrete representations of continuous random processes. The gyro bias dynamics, given in Eq. (16.10) and repeated here, will be used as an example in this section.

$$\dot{b} = \eta_b \tag{16.19}$$

If  $\eta_b$  has spectral density  $Q$ , then Eq. (16.19) can be implemented discretely as

$$b_k = b_{k-1} + w_k \tag{16.20}$$

where the time step from  $b_{k-1}$  to  $b_k$  is  $\tau$ , and  $w_k$  is a random number with a variance of  $Q\tau$  (and standard deviation  $\sqrt{Q\tau}$ ).

Sometimes, there may be nonlinear dynamics in addition to noise driving a state, and numerical integration will be necessary. The random quantities used in the numerical integration must be scaled appropriately in order represent the correct continuous process. An example using fourth order Runge-Kutta integration is shown below; for simplicity, the bias equation given in Eq. (16.19) is used even though it is more practically implemented using Eq. (16.20).

Applying fourth order Runge-Kutta, the update equation becomes

$$b_k = b_{k-1} + \tau \frac{(w_{k1} + 2w_{k2} + 2w_{k3} + w_{k4})}{6} \tag{16.21}$$

where  $w_{k1} - w_{k4}$  are random numbers with variance  $\sigma_k^2$ . Of course, the last term in Eq. (16.21) is itself a random number,

$$w_{k(RK)} = \tau \frac{(w_{k1} + 2w_{k2} + 2w_{k3} + w_{k4})}{6} \quad (16.22)$$

and for Eqs. (16.20) and (16.21) to be equivalent,  $w_k$  and  $w_{k(RK)}$  must have the same variance.

$$\begin{aligned} Q\tau &= E(w_{k(RK)}^2) \\ &= \tau^2 \frac{(\sigma_k^2 + (2\sigma_k)^2 + (2\sigma_k)^2 + \sigma_k^2)}{36} \end{aligned} \quad (16.23)$$

Therefore,

$$\sigma_k^2 = \frac{36}{10\tau} Q \quad (16.24)$$

---

## 16.6 The Kalman Filter Algorithm

The Kalman Filter algorithm is

$$\begin{aligned} k &= ph^T(hph^T + r)^{-1} \\ x &= x + k(z - hx) \\ p &= (E - kh)p(E - kh)^T + krk^T \\ p &= \frac{1}{2}(p + p^T) \\ x &= \Phi x + \Upsilon u \\ p &= \Phi p \Phi' + \Gamma q \Gamma^T \\ p &= \frac{1}{2}(p + p^T) \end{aligned} \quad (1)$$

where  $z$  is the measurement vector,  $h$  is the measurement matrix,  $p$  is the state covariance matrix,  $x$  is the state vector,  $\Gamma$  is the noise input matrix,  $\Upsilon$  is the deterministic input matrix,  $q$  is the plant noise covariance matrix, and  $E$  is the identity matrix. Averaging the covariance matrix with its transpose stabilizes the covariance propagation and eliminates the largest error term due to roundoff.  $q$  encompasses both plant uncertainty and unmodeled disturbance torques.

This implementation assumes that the state transition matrix  $\Phi$  and measurement matrix  $h$  are constant. The output you want to use a step  $k$  is the output from the second equation, which is the measurement update. The next steps are predictions that are used to initiate the following step in the filter.

If the measurement matrix is nonlinear, the measurement equation can be replaced by the iterated extended Kalman Filter

The recursive least squares method is implemented with an iterated extended Kalman Filter. The equation is

$$\begin{aligned} x_{k,i+1} &= x_{k,0} + K_{k,i}[y - h(x_{k,i}) - H(x_{k,i})(x_{k,0} - x_{k,i})] \\ K_{k,i} &= P_{k,0}H'(x_{k,i})[H(x_{k,i})P_{k,0}H'(x_{k,i}) + R_k]^{-1} \\ P_{k,i+1} &= [1 - K_{k,i}H(x_{k,i})]P_{k,0} \end{aligned} \quad (16.25)$$

where  $i$  denotes the iteration. Note that on the right hand side  $P$  and  $R$  are always from step  $k$  and are not updated during the iteration.

---

## 16.7 Batch Methods

Batch methods of attitude determination take a set of measurements and attempt to fit a solution.

There are four types of batch least squares estimators. They are least squares, weighted least squares, maximum likelihood and Bayesian. Given the measurement equation

$$z = \mathbf{H}\mathbf{x} + \mathbf{v} \quad (16.26)$$

where  $\mathbf{v}$  is unbiased white noise, the least squares estimator is

$$\hat{\mathbf{x}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{z} \quad (16.27)$$

$\mathbf{z}$  may be a vector with measurements from different types of sensors taken at different times. The assumption is that the state being measured has not changed much during the interval. This process treats all measurements equally.

The next type of estimator is the weight least squares with a weighting matrix  $\mathbf{R}$

$$\hat{\mathbf{x}} = (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{z} \quad (16.28)$$

$\mathbf{R}$  may be chosen by any means desired. For example, if there are too many measurements from one sensor, rather than removing them completely one might deweight them.

The third kind of estimator is the Maximum Likelihood Estimator. It is

$$\hat{\mathbf{x}} = (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{z} \quad (16.29)$$

which is identical in form to the weighted least squares. The difference is that  $\mathbf{R}$  is the covariance matrix for  $\mathbf{z}$ . Thus, the selection of the weighting matrix is put on firm statistical footing.

The final type of estimator is the Bayesian Estimator which is

$$\hat{x} = (P_0^{-1} + H^T R^{-1} H)^{-1} H^T R^{-1} z \quad (16.30)$$

where  $P_0$  is the a priori covariance matrix for  $x$ . This assumes that we know something about  $x$  prior to taking any measurements.

It is instructive to look at the case of three sequential measurements. In this case the measurement equation is

$$z = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} x \quad (16.31)$$

The least squares estimator becomes

$$\hat{x} = \frac{1}{3}(z_1 + z_2 + z_3) \quad (16.32)$$

which is just the average of the three measurements. The weighted least squares (and Maximum Likelihood) estimators become

$$\hat{x} = \frac{\left(\frac{z_1}{r_1} + \frac{z_2}{r_2} + \frac{z_3}{r_3}\right)}{\frac{1}{r_1} + \frac{1}{r_2} + \frac{1}{r_3}} \quad (16.33)$$

If the measurements happen to be identical, the weights have no affect on the estimate. Otherwise, the measurement with the smallest weight will be favored. The Bayesian estimator is

$$\hat{x} = \frac{\left(\frac{z_1}{r_1} + \frac{z_2}{r_2} + \frac{z_3}{r_3}\right)}{\frac{1}{r_1} + \frac{1}{r_2} + \frac{1}{r_3} + \frac{1}{p_0}} \quad (16.34)$$

In the limit, as  $p_0$  goes to infinity, the estimate goes to zero.

The Differential Corrector algorithm is

$$x_{k+1} = x_k + [S_0 + H_k' W H_k]^{-1} [H_k' W (y - h_k) + S_0 (x_0 - x_k)] \quad (16.35)$$

where  $W$  is the measurement weighting matrix and  $S_0$  is the a priori covariance matrix of vector  $x_0$ . For the purposes of evaluating the quality of the result the condition number of the inverse should be evaluated.  $S_0$  tends to improve the conditioning.

The conjugate gradient method finds the solution to the loss function directly. The loss function is

$$2J = (y - h_k)'W(y - h_k) + (x_0 - x_k)'S_0(x_0 - x_k) \quad (16.36)$$

Conjugate gradient and steepest descent both require the derivative of  $J$ , which requires  $H$ . Both find  $x_k$  that minimizes  $J$ .

---

## 16.8 Vector Measurements

For small angles the transformation matrix from the reference frame to the body frame is given by

$$M = E - \theta^\times \quad (16.37)$$

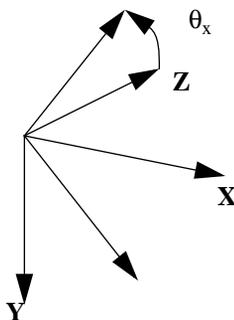
where  $E$  is a 3-by-3 identity matrix and  $\theta^\times$  is the skew-symmetric matrix

$$\theta^\times = \begin{bmatrix} 0 & -\theta_z & \theta_y \\ \theta_z & 0 & -\theta_x \\ -\theta_y & \theta_x & 0 \end{bmatrix} \quad (16.38)$$

This can be seen from the following diagram.

**FIGURE 16.1 Small Angles**

---



Assume that the transformation matrix from the body frame to the sensor frame is  $m_s$ . Then, a vector in the reference frame becomes

$$\begin{aligned}u_M &= m_S(E - \theta^\times)u \\u_M &= m_S u - m_S \theta^\times u \\u_M - m_S u &= m_S \theta^\times u \\z &= h\theta\end{aligned}\tag{16.39}$$

This is the form of the standard linear measurement equation for a Kalman Filter formulation.  $h$  is a 3-by-3 matrix however the inverse of  $h$  is singular, hence it is not possible to estimate three components of  $\theta$  from any set of measurements,  $z$ , which are derived from a single vector unless  $h$  (and therefore  $u$ ) is time varying. If multiple, independent vectors measurements are available then the three components of  $q$  can be estimated.

Vector measurements include star measurements, sun measurements, planet measurements and magnetic field measurements. All can be combined using this formulation.

---

# *Geosynchronous Satellite Control*

---

## *17.1 Introduction*

This chapter describes how to create a preliminary control system design for a geosynchronous satellite. It describes how to interpret customer requirements, select actuators and sensors, and organize your design.

---

## *17.2 Requirements*

Typical requirements are:

- The beam center shall not deviate from nominal by more than 0.2 deg.
- The spacecraft shall stay within 0.1 deg of station for its entire life.
- The spacecraft life shall be greater than 10 years.
- No single point failure shall cause a pointing error of greater than 0.2 deg.
- The payload consists of one global beam pointing at nadir.

These are the minimum requirements and give you a lot of flexibility. However, you are usually competing against other manufacturers so you must strive to meet these requirements with the lowest cost design.

The requirements only talk about mission orbit performance. You also have to get the satellite to its mission orbit. If you can ride on a Proton or IUS, you may not have to worry about transfer orbit. Otherwise, you have three distinct modes:

- Transfer Orbit
- Acquisition
- Mission Orbit

Your transfer orbit and acquisition requirements, besides getting the satellite safely to its mission orbit, are to minimize the fuel consumption since fuel is usually the life limiting factor in a satellite design.

---

### *17.3 The Design Process*

You can generally run the following in parallel:

- Simulation design
- Transfer orbit control design
- Mission orbit control system design

It is important to start the simulation design right away since simulations take a long time to test and debug. If you don't have any simulation at all you should start by building the simplest possible—rigid body with generic sensors and actuators using the simplest possible models. An object-oriented approach can help here. You can always add complexity later.

---

### *17.4 Transfer Orbit*

There are several choices delta V engines:

Solid motor (big thrust—1000's of N)  
Liquid fuel motor (moderate thrust —100's of N)  
Electric thrusters (low thrust— N)

High thrust engines usually require that the spacecraft be spun for stability during the burn. Spacecraft with moderate thrust levels can be spun or three axis stabilized. The cheapest way for small satellites is to use a solid. This design will use a solid motor.

When it comes time to fire the solid, you will need to be pointing in the right direction. For that, you need attitude information and thrusters to adjust the orientation. At a minimum, you need it to know your inertial orientation. If you had perfect gyros you could initialize them on the ground and get your attitude from them. A more practical alternative is to use celestial objects as your reference. The sun is a convenient reference and so is the earth. Since your spacecraft is spinning, you can use the spacecraft spin as your scan mechanism. This leads you to the cheapest sensors, single axis sun sensors and horizon sensors.

Since you need thrusters for stationkeeping you need a set of thrusters that can provide three-axis control. It is undesirable to add additional thrusters for transfer orbit, therefore it is necessary to make certain that the thrusters are usable (i.e. are not blocked by undeployed solar arrays and antennas) for controlling the spacecraft spin rate or changing the spin-axis orientation.

Since we have chosen a spinner, the final consideration is whether the spacecraft is spinning about its major or minor axis. If the former is true we are done with the preliminary design. Otherwise, we will need a control system to damp nutation and to prevent the spacecraft from ending up spinning about its major axis.

---

### *17.5 Mission Orbit Design*

We have left acquisition for last. Acquisition is the transition from the spinning phase to the mission orbit phase.

You have one firm requirement that dictates a particular choice of actuators—you must perform stationkeeping— so you need thrusters for delta V. If you pick either bipropellant or monopropellant unaugmented thrusters you will also need additional thrusters for attitude control during stationkeeping. If you pick electric thrusters this may not be the case.

For this design, electric heater augmented hydrazine thrusters will be used for inclination control. These thrusters are cost-effective because inclination control requires a lot of propellant. These thrusters have thrusts in the 0.1 lbf range and will require attitude control thrusters.

This leads to a decision on our architecture. There are three choices:

- Spin stabilized
- Momentum bias
- Three-axis

The first two are identical except that in the first the spinning part has a bigger inertia than the despun part. In the second, the opposite is true. The major advantage of a high momentum design is that it provides passive attitude control. Although inertially fixed torques will precess the spin-axis, the momentum in the system will resist those torques and may be sufficient to compensate entirely for the cyclic torques. In addition, the momentum in the spacecraft makes it resistant to the consequences of thruster failures.

Another advantage is that since the momentum makes it resistant to disturbances, it may be possible to forgo a yaw sensor. It is easy to sense roll and pitch with an earth sensor. Yaw is more difficult. It can be sensed intermittently with a sun sensor, or continuously with a star sensor or sun sensor/gyro combination. Forgoing a yaw sensor can save a considerable amount of money and weight.

For control we can use thrusters, magnetic torquers, solar pressure or wheel pivoting.

A three-axis design would use pivoted or fixed reaction wheels or thrusters for control. Reaction wheels require some mechanism for unloading momentum from the system. This can be done with thrusters, magnetic torquers or solar

pressure. Yaw sensing is required if there is no bias momentum since a body-fixed yaw disturbance cannot be sensed in roll and without yaw sensing this would cause the attitude to diverge.

On this spacecraft, the only yaw requirement is levied by the spot beam.

A yaw error will translate into

$$\begin{aligned}\Delta A_z &= 0.035\Delta y_{\text{yaw}} \\ \Delta E_l &= 0.035\Delta y_{\text{yaw}}\end{aligned}\tag{17.1}$$

Thus, we can tolerate much larger yaw errors than roll or pitch errors. This is because this mission does not have a direct yaw requirement, the beam is circular and the yaw error has a gain of 0.035 when applied to azimuth or elevation. This means that we can probably do without yaw sensing if we use a spinner or momentum bias design. This might not be the case if we had cross-links to other satellites.

At this point we have a number of trades to make. Some of the possible configurations are shown in the following table. You can probably come up with more configurations.

**TABLE 17.1 Possible Configurations**

<b>Design</b>	<b>Roll/Yaw Control</b>	<b>Pitch Control</b>	<b>Roll/Yaw Unloading</b>	<b>Pitch Unloading</b>
Spinner	Thrusters	Spin motor	N/A	Thrusters
Momentum Bias	Thrusters	MWA Motor	N/A	Thrusters
Momentum Bias	Magnetic Torquers	MWA Motor	N/A	Thrusters
Momentum Bias	Solar Pressure	MWA Motor	N/A	Solar Pressure
Momentum Bias	Pivots	MWA Motor	Magnetic Torquers	Thrusters
Momentum Bias	Pivots	MWA Motor	Thrusters	Thrusters
3-Axis	Reaction Wheels	Reaction wheel	Thrusters	Thrusters
3-Axis	Reaction Wheels	Reaction wheel	Magnetic Torquers	Thrusters

As discussed above, a careful cost trade-off should be made before selecting the configuration. For the sake of this tutorial, assume that this has been done and the momentum bias with thruster roll/yaw control has been chosen. No yaw sensor will be used. The roll and pitch axes will be sensed using an earth sensor. Earth sensors can be either static, consisting of a ring of thermopiles in the focal plane of the sensor, or scanning which have an oscillating mirror that sweeps the image of the earth across pyroelectric detectors or bolometers. The choice is one largely of cost.

A stationkeeping control system will also be needed. Thrusters will be used as actuators but the choice of sensors is still open. Since the disturbances will be larger (tenths of inlbs versus hundreds of micro inlbs) a faster acting control system will be needed. Earth sensors tend to be noisy thus it is difficult to differentiate the signal to get rate for a high bandwidth control system. As a consequence, it is a good idea to use gyros to produce rate information for sta-

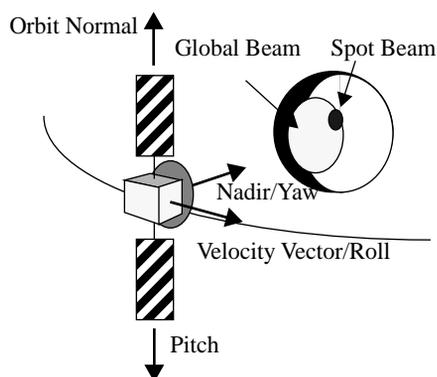
tionkeeping. Since the gyros will only be used during stationkeeping, they need not be long-life gyros. While the earth sensor can still be used for angle information in pitch and roll, the yaw gyro output must be integrated for yaw. This cannot give an absolute yaw measurement, only one relative to the original yaw estimate when the gyro was initialized.

---

## 17.6 The Geometry

The orbital geometry of a spacecraft with its solar arrays deployed is shown below.

**FIGURE 17.1 Orbital Geometry**



The solar arrays are deployed from the north and south faces. When stowed the array panels will lie flat on these surfaces. The east and west faces require space for thrusters to perform east/west stationkeeping. Consequently, the apogee kick motor (AKM) has to be on the anti-earth face. The spin axis will run through this face and point out the front of the nozzle. The earth sensors will be on the nadir face. The antenna may get in the way of the sensors and cutouts in the dish (which should not pose much of an antenna problem if they are much smaller than the wavelength) may be necessary. During the AKM burn, the spin axis will point at an angle from the earth equatorial plan roughly equal to the negative of the transfer orbit inclination. After the AKM burn, it will be necessary to get this vector pointing at the earth and rotating about the pitch axis at orbit rate.

---

## 17.7 Acquisition

Acquisition is the transition between transfer orbit and mission orbit. In transfer orbit the spacecraft is spinning about its solid motor axis at the attitude needed for the delta V burn to put the satellite in the geosynchronous orbit, or a nearby orbit from which the satellite can drift to station. With this design we have several options. If we add a sun sensor we can

- Acquire the sun

- Rotate about the sun line (a safe attitude)
- Acquire the earth
- Lock on to the earth
- Spin up the MWA

This is known as a sun/earth acquisition. This is desirable because the earth is a small target at geosynchronous altitudes. Even without a sun sensor, we can use the solar arrays as a sun sensor of sorts.

Another option is to

- Precess the spin axis into the orbit plane
- Rotate in yaw until the desired orientation is achieved
- Spin up the MWA
- Lock on to the earth

A third option is to

- Precess the spin-axis to orbit normal (leaving the MWA axis in the orbit plane)
- Spin up the MWA. The spacecraft will flip over since the inertial momentum vector cannot be moved by internal torques
- Lock on to the earth

This last option is known as the dual spin turn and was pioneered by RCA Astro Electronics. This last choice does not require any inertial reference after the precession to orbit normal is complete. Measuring the spin-axis attitude prior to the dual-spin-turn is done using horizon sensors and a single-axis sun sensor.

When choosing an acquisition scheme one must pay close consideration to power. During parts of the acquisition, the solar arrays may not get much sun and if the acquisition takes too long it could leave the spacecraft in a perilous state.

Since the spacecraft is a major-axis spinner, and the last option does not require any sensing, we will choose it for our design.

## 17.8 Summary

The control system is summarized in the table below.

**TABLE 17.2 Control System Summary**

Mode	Actuator	Sensors
Transfer Orbit	Thrusters	Sun sensor Horizon sensors
Acquisition	MWA motor	None
Mission Orbit Normal	MWA motor Thrusters	Earth sensor
Mission Orbit Stationkeeping	Thrusters	Gyros Earth sensor



# *Geosynchronous Spacecraft Control Design*

---

This chapter walks you through a detailed spacecraft control system. The goals are a ten year life and overall beam pointing of 0.2 deg.

---

## *18.1 A Mission Architecture*

The requirements: form a geosynchronous spacecraft with a mission life of 10 years. It will have a global beam which must point to nadir to within 0.2 deg. The goal is to design the least expensive spacecraft that can achieve these requirements and make some money doing it!

There are three phases to this mission:

- Transfer Orbit
- Acquisition
- Mission Orbit

During transfer orbit, the spacecraft is spinning about its major axis. This makes it passively stable. Reorientations are done using thrusters that precess the spin axis.

During the mission orbit, the satellite is stabilized using a momentum wheel. The momentum in the wheel is sufficiently high so that the spacecraft is dual-spin stable. Control may be accomplished using either magnetic torquers and the momentum wheel motor, or with thrusters. Stationkeeping maneuvers are performed using thrusters.

The transition between the spinning and momentum bias states is done with the dual spin turn. During the dual spin turn, nutation can be damped using the momentum wheel motor.

The transfer orbit, deployment and mission orbit configurations are shown below

**FIGURE 18.1 Transfer Orbit Configuration Diagram**

---



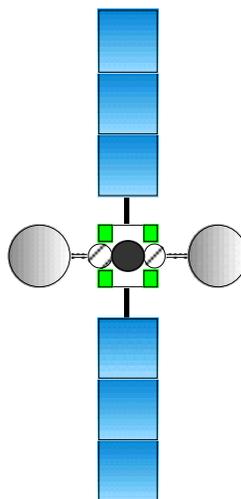
**FIGURE 18.2 Deployment Configuration Diagram**

---



**FIGURE 18.3 Mission Orbit Configuration Diagram**

---



---

## 18.2 Design Steps

Although as a controls engineer you will be concerned about such things as control system stability margins, transient response, etc. your customer is only concerned whether you meet the requirements for beam pointing and spacecraft life. Spacecraft life is determined by the life of the components you choose and by the amount of fuel the spacecraft carries. You address component life issues by carrying redundant components as necessary. The pointing budget and propellant budgets must be computed and updated regularly as the spacecraft design evolves.

---

## 18.3 Spacecraft Overview

The spacecraft is illustrated in the following figure. Some of the features are listed below

**Power system**—two three panel silicon solar arrays, Nickel Hydrogen batteries, radiators on the north and south panels. Heat pipes and resistance heaters for thermal control. The outer solar panels are exposed to sunlight during transfer orbit.

**Payload**—two 1.6 m receive/transmit dishes. Maximum power for transmitting a signal is 500 watts.

**Communications**—two omni antennas, one parallel to the Y-axis and one along the Z-axis on the nadir panel.

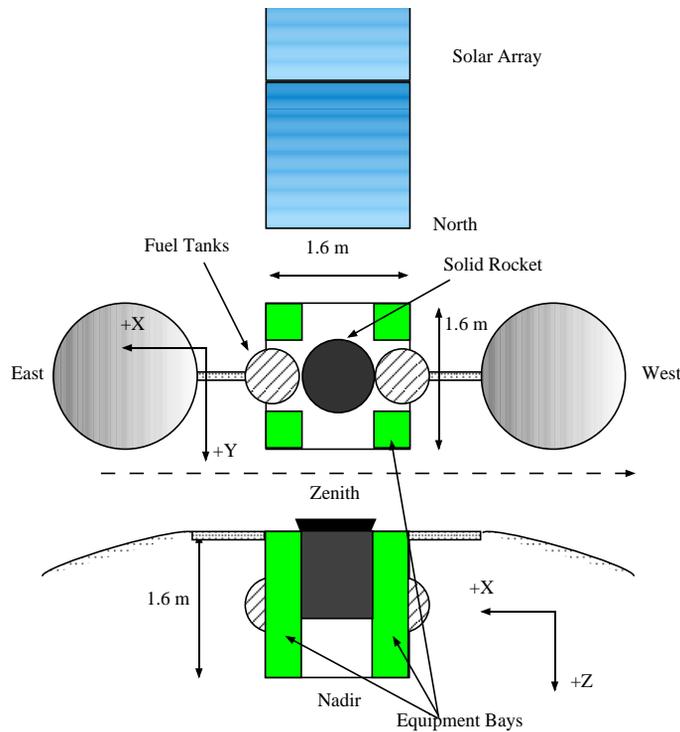
**Structure**—composite truss structure with aluminum panels. Composite solar array structure. Most of the flexibility of the array will come from the hinges between the solar panels and where the array is attached to the core.

**Transfer Orbit Attitude Control**—two horizon sensors and two single axis digital sun sensors for attitude determination. The eight east and west face thrusters are used for precession maneuvers and spin changes during transfer orbit.

**Orbit Control**—four electrothermal hydrazine thrusters on the north face for inclination control. A solid motor along the Z-axis for orbit insertion. Eight hydrazine monopropellant engines on the east and west face for east/west maneuvers.

**Mission Orbit Attitude Control**—a fixed momentum wheel with positive momentum along the -Y axis. Two magnetic torquers along the +Y and +Z axes. Twelve hydrazine thrusters for attitude control and momentum management. A scanning earth sensor for sensing roll and pitch. No yaw sensing is available during mission orbit. Three rate integrating gyros are used for rate sensing during stationkeeping maneuvers.

FIGURE 18.4 Close-up of spacecraft



The spacecraft layout was designed so that spacecraft would be a major axis spinner during transfer orbit and be nearly symmetrical about the Z-axis. In other words, the X-axis inertia and Y-axis inertia are nearly the same during transfer orbit. This was accomplished by splitting the electronics into four boxes and locating them as illustrated above. Laying out a spacecraft so that it has good thermal properties, is easy to assemble and is balanced is very difficult and expensive. A satellite that was three axis stabilized in transfer orbit and didn't spin at a high rate would not have to be so carefully balanced.

The antenna beams point at nadir, nearly +Z in this picture. Their angle is dependent on the location of the feeds which are on the nadir panel.

The fuel supply is split into two half systems for redundancy. Two tanks are located on the east and west faces.

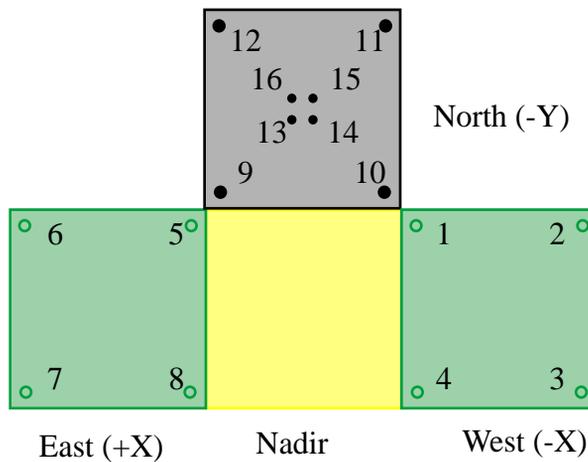
The thrusters are located at the corners of the box on the east, west and north faces. The electrothermal hydrazine thrusters are clustered around the Y-axis on the north face. Their exact location and cant angles is set to minimize losses due to plume drag. The plumbing is designed so that if one half system fails the other half system can perform all required maneuvers.

Fixed momentum wheels were chosen for simplicity. Pivoted wheels (either single or double) are more flexible, but heavier and more expensive.

The propulsion system is a blowdown system. You fill the tanks with fuel and pressurize them with helium. The mass and volume of helium determine the blowdown ratio, that is the ratio of initial pressure to empty pressure. For this spacecraft, we choose an initial pressure of  $2,413,250 \text{ N/m}^2$  (350 psia) and a final pressure of  $689,500 \text{ N/m}^2$  (100 psia).

The thruster layout is illustrated below. Thrusters 1 through 12 are monopropellant hydrazine thrusters with 5 N maximum thrust at a tank pressure of  $2,413,250 \text{ N/m}^2$ .

**FIGURE 18.5 Thruster Layout Diagram**



Thrusters 12 through 16 are electrothermal hydrazine thrusters with a maximum thrust of 2 N at a tank pressure of  $2,413,250 \text{ N/m}^2$ .

## 18.4 Spinning Transfer Orbit

### 18.4.1 Dynamics

A typical spinning spacecraft in transfer is composed of a fairly rigid (especially compared to the deployed configuration) core spacecraft with a significant fraction of its mass in the form of liquid propellant. Fuel motion will cause energy dissipation in the spacecraft. If the spacecraft is a major axis spinner this generally does not pose a problem since any energy dissipation will cause the spacecraft to return to its major axis spin state. However, if the spacecraft is a minor axis spinner its attitude will diverge and it will attempt to spin about its major axis. Consequently, minor

axis spinners require nutation control systems. In this tutorial, the spacecraft will be assumed to be a major axis spinner, and nutation control will not be discussed further.

The equations of motion for a rigid spinning spacecraft are

$$I\dot{\omega} + \omega \times I\omega = T \quad (18.1)$$

where  $I$  is the spacecraft inertia, which assumed to be constant, and  $\omega$  is the spacecraft angular velocity. Analytically, the transfer functions are

$$\begin{aligned} \begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix} &= \frac{\begin{bmatrix} s & k_x \\ -k_y & s \end{bmatrix}}{s^2 + k_x k_y} \begin{bmatrix} T_x/I_x \\ T_y/I_y \end{bmatrix} \\ k_x &= \frac{\omega_o(I_z - I_y)}{I_x} \\ k_y &= \frac{\omega_o(I_z - I_x)}{I_y} \end{aligned} \quad (18.2)$$

These reduce to a pair of single integrators if  $\omega_o$ , the z-axis spin rate, is zero.

The inertial angular momentum vector is

$$H = AI\omega \quad (18.3)$$

where  $A$  transforms a vector from the body frame to the inertial frame and the energy is

$$E = \frac{1}{2}\omega'I\omega \quad (18.4)$$

$H$  is fixed in magnitude and direction in the absence of external torques. Internal torques can cause  $E$  to change even though  $H$  does not. For example, when a minor axis spinner has energy dissipation it will reorient itself so that its energy is a minimum. Even though the spin axis changes, the angular momentum does not and it redistributes itself in the body frame so that it remains fixed in direction and magnitude in the inertial frame.

The rotation of the spin-axis about the momentum vector is often known as precession. In this tutorial precession will mean motion of the momentum vector with respect to the inertial frame. Since momentum is conserved, only an external torque, such as the torque produced by a thruster, can precess the momentum vector.

### 18.4.2 Actuators and Sensors

Thrusters are used to control the orientation and spin rate of the spacecraft during transfer orbit. Fixed pulsewidth, throttleable or pulsewidth modulated thrusters can be used. Fixed pulsewidth are the least flexible. Both pulsewidth modulation and throttling permit the use of linear control laws, but both have threshold and saturation nonlinearities

associated with them. Keep in mind that the propellant pressurization system will have an effect on the use of thrusters. This spacecraft employs pulsewidth modulation.

Sun sensor and horizon sensors are used for attitude determination. The sun sensor has two outputs. One provides a pulse every time the sun passes through the sun sensor boresight/spin-axis plane. This provides a convenient timing reference for computing spin rate. The other outputs the sun angle in the spin-axis/boresight plane.

Gyros are available, but the spin rates during transfer orbit are so high that their output would be saturated.

The horizon sensors detect the presence of the earth. The horizon sensors view the earth in the CO<sub>2</sub> band, around 14  $\mu\text{m}$ . The Earth's atmosphere is relatively stable at this frequency. When the earth is in the sensor field-of-view the output of the sensor jumps. The horizon sensors have circuitry that detects the transition from cold space to warm earth. There are many ways to do this. The simplest outputs a pulse whenever the output of the detector reaches a fixed threshold. Another outputs a pulse when the output reaches a fixed percentage of the peak. The latter tends to be less sensitive to Earth atmospheric radiance variations.

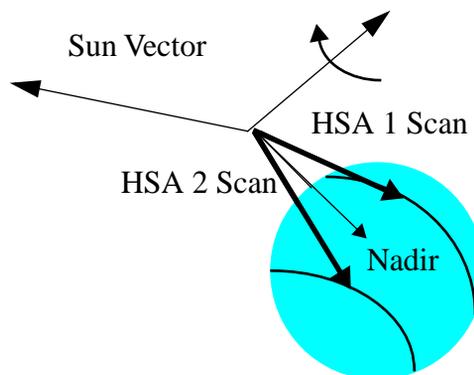
Horizon sensor measurements have many error sources. These include:

- Earth seasonal and daily radiance variations
- Sun and moon interference
- Sensor dynamics
- Misalignments
- Irregularities in the Earth's figure
- and so on

The geometry of spin-axis attitude determination is illustrated below

**FIGURE 18.6 Spin-Axis Attitude Determination Diagram**

---



### 18.4.3 Changing the Spin Rate

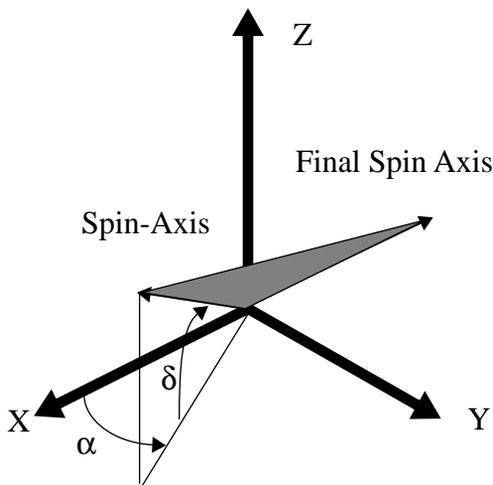
Two ways are available to change the spin rate. One is by manually firing the appropriate thruster(s). The second is to use the automatic spin rate control function. This function measures spin rate using the sun sensor pulses and a spin rate estimator structure, and controls the spin rate with a simple proportional controller. The estimator accounts for spin rate changes due to firing thrusters.

Reorientations of a spinning spacecraft are performed using what is known as a spin precession maneuver. The idea is to reorient the momentum vector without changing the spin rate. Because only the orientation of the spin-axis is of interest, we need only concern ourselves with two angles, right ascension and declination.

If we have continuous information about the spin-axis attitude then it is possible to perform a great-circle precession. This is illustrated below and the spin-axis moves in the plane represented by the shaded region.

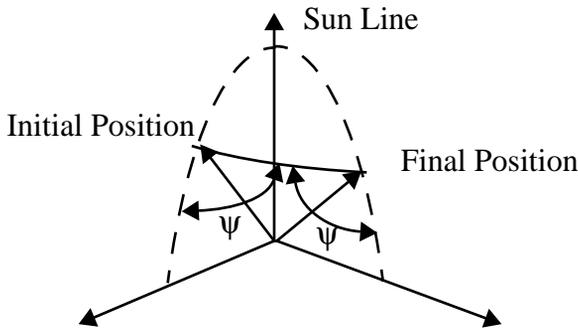
**FIGURE 18.7 Great Circle Precession Diagram**

---



This could be performed if the satellite had three-axis gyros or if it updated its attitude from horizon and sun sensor measurements using a Kalman Filter.

**FIGURE 18.8 Rhumb Line Precession Diagram**



This satellite uses a simpler approach which is known as a rhumb line precession. The geometry is illustrated below.

The coordinate frame is such that the sun vector is the +Z axis. The initial and final spin-axis vectors are shown as the path connecting them. The azimuth angle,  $\psi$ , which is the angle between the trajectory and the sun/spin-axis plane, is a constant during the maneuver.  $\psi$  is constant because a fixed delay between the sun crossing the spin-axis sun-sensor plane and the thruster firing is used. The following generates an rhumb line spin-precession maneuver that moves 10 deg in declination. The maneuver is done at equinox so the sun is along the +X axis.

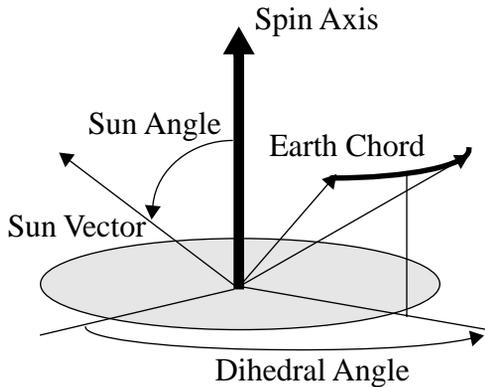
#### 18.4.4 Attitude Determination

Attitude determination is performed using three measurements, the sun angle, and the times when the horizon sensor scan crosses the edge of the earth measured relative to the time at which the sun crosses the sun sensor boresight/spin-axis plane. The three measurements are resolved into the sun angle, the dihedral angle and the chordwidth. The chordwidth is the difference between the trailing and leading edge times divided by the spin rate; the dihedral angle is the sum of the leading and trailing edge times divided by twice the spin-rate. If the noise statistics for the leading and trailing edge times are identical, then the dihedral angle and chordwidth are uncorrelated.

Besides the measurements, attitude determination also requires the orbital position of the satellite. Consequently, accurate attitude determination requires accurate orbit determination. The attitude determination geometry is illustrated below.

FIGURE 18.9 Attitude Determination Geometry Diagram

---



The three angles give an unambiguous attitude reference. Usually many measurements spaced over a period around apogee are taken and a least squares algorithm is used to estimate both attitude and any biases in the data.

#### 18.4.5 Delta V Engine Firing

The delta-V engine is fired near apogee to put the spacecraft into the geosynchronous orbit. Since the spacecraft is rarely at its station at the time of the firing, the maneuver is planned to leave the spacecraft in a slightly eccentric orbit so that it will drift to its final station. The firing of the AKM causes nutation and a slight spin up of the spacecraft.

#### 18.4.6 The Transfer Orbit Generator

Transfer orbit operations for most geosynchronous satellites are sequences of operations initiated from the ground. These are:

- Attitude data collection
- Spin rate change maneuvers
- Spin precession maneuvers
- Apogee motor firings

## 18.5 Acquisition Using The Dual Spin Turn

### 18.5.1 Dynamics

The dual spin turn is based on conservation of angular momentum. Inertial angular momentum is conserved, therefore the momentum vector will remain fixed in the inertial frame unless and external torque is applied. The spacecraft is first precessed so that the +z axis (the spin axis) is aligned with positive orbit normal. The momentum wheel that is normal to the +z axis is turned on. As it spins up momentum is transferred in the body frame to the -y axis. Since the angular momentum vector is inertially fixed, the -y-axis aligns itself with positive orbit normal with some residual spin rate about the -y-axis.

For the dual spin turn to work, the initial spin rate about the +z-axis must be within the limits specified by the equations

$$\frac{h_W}{I_{Yaw}} \leq \Omega \leq h_W \frac{(1-f)}{f I_{Yaw}} \quad (18.5)$$

$$f = \left| 1 - \frac{I_{Pitch}}{\max(I_{Yaw}, I_{Roll})} \right|$$

where  $I_{Pitch}$  is the inertia of the axis with which the wheel is aligned,  $I_{Yaw}$  is the inertia of the axis about which the spacecraft is initially spinning, and  $I_{Roll}$  is the other axis inertia.

### 18.5.2 Actuators and Sensors

The momentum wheel motor is the only actuator used during the dual spin turn. No sensing is required during the dual spin turn.

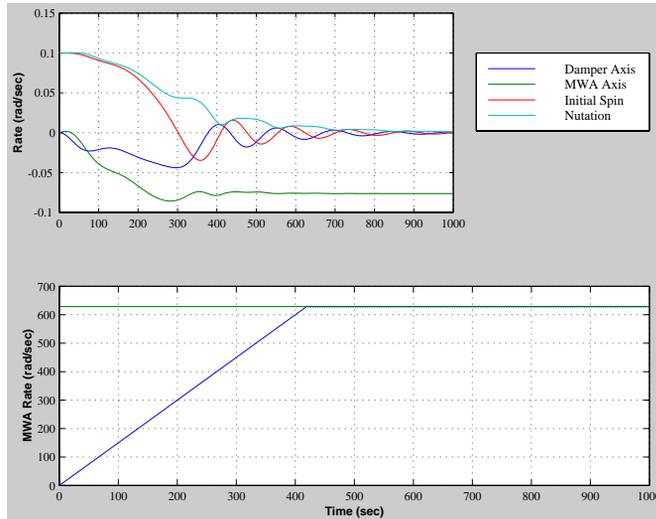
### 18.5.3 Initialization

The dual spin turn is initialized by despinning the spacecraft to the desired spin rate and then commanding the momentum wheel to the desired post dual spin turn rate.

### 18.5.4 Simulation

A simulated dual spin turn is shown below. the rate about z goes to zero and the momentum is absorbed in the momentum wheel and in the y-axis rate. nutation damps to zero by the end of the simulation.

FIGURE 18.10 Dual Spin Turn



In the second plot the upper line is the commanded momentum.

### 18.5.5 Pitch Acquisition

Once the spacecraft is spinning about its y-axis and nutation has damped, the pitch control system is activated and thrusters are selected. The earth sensor processing will automatically estimate pitch rate and pitch even though the earth is only passing through the sensor field-of-view.

## 18.6 Disturbances

The major disturbance sources on the spacecraft are solar pressure, including thermal effects, gravity gradient, residual dipole, and during stationkeeping thruster plume interaction. Each will be discussed in the following sections.

## 18.7 Dynamics

### 18.7.1 Introduction

During the mission orbit a momentum bias spacecraft is modeled as a rigid core with a rotor and flexible solar arrays. During normal operation, the solar array flexibility does not impact the control system performance. Consequently, it is convenient to discuss the dynamics of the two operational modes separately.

### 18.7.2 Normal Operations

The roll/yaw and pitch dynamics can be decoupled for analysis of normal operations. The pitch dynamics are a double integrator. The roll dynamics are a coupled fourth order plant. There are two pure imaginary pole pairs. The first at orbit rate is due to the kinematics, the second is the nutation mode and is due to the bias momentum. The main diagonal channels each have a pure imaginary zero pair. This zero pair is located between the orbit rate and nutation mode providing 180 degrees of phase shift and making the control problem easier.

Disturbances at orbit rate, or at the nutation frequency, will cause uncontrolled growth in the attitude errors. Since many of the disturbance sources are at orbit rate, it is necessary to add an automatic control system. The nutation pole causes large oscillatory responses whenever a sudden torque is applied to the spacecraft.

The high and low frequency limits are of interest. At high frequencies as  $s$  approaches infinity the system becomes a double integration. At low frequencies, the orbit rate pole pair dominates.

The attitude kinematics come from the small angle approximation

$$\omega = \dot{\theta} + (1 - \theta^{\times})v \quad (18.6)$$

where  $v$  is the orbit rate vector. The dynamical equations are

$$I\dot{\omega} + \omega^{\times}(I\omega + h_W) + \dot{h}_W = T \quad (18.7)$$

For the purpose of analyzing the roll/yaw dynamics assume that  $h_W$  is constant. The orbit rate vector is  $[0 \ -\omega_o \ 0]^T$ . Assume also that the inertia matrix is diagonal. Linearizing the dynamics, and coupling with the kinematics gives the transfer function matrix

$$\begin{bmatrix} \theta_x(s) \\ \theta_z(s) \end{bmatrix} = \frac{\begin{bmatrix} I_z s^2 + \omega_o h_z & (\omega_o I_x - h_x) s \\ -(\omega_o I_z - h_z) s & I_x s^2 + \omega_o h_x \end{bmatrix}}{I_x I_z (s^2 + \omega_o^2) \left( s^2 + \frac{h_x h_z}{I_x I_z} \right)} \begin{bmatrix} T_x(s) \\ T_z(s) \end{bmatrix}, \quad (18.8)$$

$$h_x = h_w + \omega_o (I_y - I_z)$$

$$h_z = h_w + \omega_o (I_y - I_x)$$

The roll/yaw equations and the equation of the damper wheel are

$$\begin{aligned} I_x \dot{\omega}_x + \omega_y \omega_z (I_z - I_y) - \omega_z h_w + J(\dot{\Omega} + \dot{\omega}_x) &= T_x \\ I_z \dot{\omega}_z + \omega_y \omega_x (I_y - I_x) - \omega_y J(\Omega + \omega) + \omega_x h_w &= T_z \\ J(\dot{\Omega} + \dot{\omega}_x) &= -D\Omega \end{aligned} \quad (18.9)$$

where  $J$  is the inertia of the damper wheel. If we lump  $J$  with  $I_x$  we can simplify the above to

$$\begin{aligned} \dot{\omega}_x + k_x \omega_z + \varepsilon \dot{\Omega} &= 0 \\ \dot{\omega}_z + k_z \omega_x - \gamma \dot{\Omega} &= 0 \\ \dot{\Omega} + \dot{\omega}_x &= -\sigma \Omega \\ k_x &= \frac{\omega_y (I_z - I_y) - h_w}{I_x} \\ k_z &= \frac{\omega_y (I_y - I_x) + h_w}{I_z} \\ \gamma &= \frac{\omega_y J}{I_z} \quad \sigma = \frac{D}{J} \quad \varepsilon = \frac{J}{I_x} \end{aligned} \quad (18.10)$$

The characteristic equation can be written in Evans form

$$\frac{s \left( s^2 - \frac{k_x (k_z + \gamma)}{1 - \varepsilon} \right)}{s^s - k_x k_z} = -\frac{\sigma}{1 - \varepsilon} \quad (18.11)$$

When  $\sigma = 0$  the numerator polynomial gives the poles of the system. When  $\sigma = \infty$  the denominator gives the poles of the system. Hence the open loop zeros are given by the denominator and the open loop poles by the numerator. The  $s = 0$  pole is the damper wheel pole and the quadratic terms give the roll/yaw poles. For stability, the open-loop zeros must be between the closed loop poles. These equations are only valid for  $J \neq 0$

$$\begin{bmatrix} \theta_x(s) \\ \theta_z(s) \end{bmatrix} = \frac{\begin{bmatrix} \frac{1}{I_x} & 0 \\ 0 & \frac{1}{I_z} \end{bmatrix}}{s^2} \begin{bmatrix} T_x(s) \\ T_z(s) \end{bmatrix}, \quad (18.12)$$

As is evident from the transfer function matrix, high bandwidth controllers can ignore the orbit rate and nutational dynamics.

At low frequencies the plant becomes

$$\begin{bmatrix} \theta_x(s) \\ \theta_z(s) \end{bmatrix} = \frac{\begin{bmatrix} \omega_o & (\omega_o I_x / h_x - 1)s \\ (1 - \omega_o I_z / h_z)s & \omega_o \end{bmatrix}}{s^2 + \omega_o^2} \begin{bmatrix} T_x(s) / h_x \\ T_z(s) / h_z \end{bmatrix}, \quad (18.13)$$

The diagonal terms are small and the dynamics are dominated by cross-coupling torques. Controllers concerned with attenuating low frequency (near orbit rate) can ignore the nutation mode.

### 18.7.3 Dual Spin Stability

It is desirable for a dual-spin spacecraft to be passively stable as long as the momentum wheel is at its nominal speed. This can be demonstrated by assuming that energy dissipation is provided by a damper wheel.

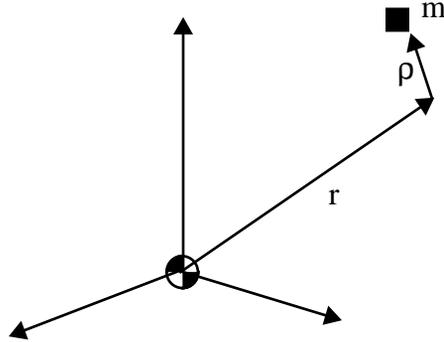
### 18.7.4 Stationkeeping Operations

It is necessary to include the dynamics of flexible appendages, mainly the solar arrays, for stationkeeping operations. There are two reasons for this. The first is that stationkeeping control systems have to respond to large disturbances caused by the delta V thrusters. Since it is desirable to minimize the average pointing error during the burn the stationkeeping system should respond quickly to these disturbances. This leads to a high bandwidth system which may interact with flexible modes of the spacecraft. The second reason is that all thrusters are nonlinear. Even throttleable thrusters have limited throttle range and a significant minimum impulse bit. The nonlinearities give rise to control spillover beyond the control system bandwidth. Consequently, it is necessary to consider higher frequency modes.

The simplest way to add flex modes is to assume that the spacecraft center-of-mass does not move when the spacecraft bends. In addition, assume that the flexible part of the spacecraft is not rotating relative to the core of the spacecraft. Finally, assume that the displacements due to bending are small and that nonlinear terms, that include the bending displacement and rate, are insignificant.

The geometry is illustrated below.

FIGURE 18.11 Stationkeeping Geometry



Without any loss of generality the flexible body model will deal only with point masses as opposed to rigid bodies.  $r$ , the vector from the center of mass to the point mass is not necessarily small.  $\rho$  is the displacement of the mass from its nominal position.

We will assume that the inertia matrix of the spacecraft includes the contributions due to the mass  $m$  and the moment arm  $r$ . Thus  $I$  is the inertia of the undeformed spacecraft. The equation for the core spacecraft angular acceleration is

$$I\dot{\omega} + \omega^\times(I\omega + h_w) + \dot{h}_w + \sum m_i r_i^\times \ddot{\rho}_i = T \quad (18.14)$$

The summation is over the physical coordinates. The acceleration of each point mass causes a torque on the spacecraft. The equation for the point mass acceleration is

$$\ddot{\rho}_i - r_i^\times \dot{\omega} + \omega^\times \omega^\times r_i = \frac{f_i}{m_i} - \frac{F}{m_i} \quad (18.15)$$

These equations are quite general and can apply to any flexible appendage of any shape or size. Since the center-of-mass does not move, there is no inertial coupling between the flexible degrees of freedom. Coupling may be introduced through the force term on the right-hand-side, however.

For simple systems these equations can be solved directly. For more complex systems with many flexible degrees of freedom it is convenient to transform the flexible degrees of freedom into modal coordinates. Neglect the nonlinear terms and write

$$m_i \ddot{\rho}_i - m_i r_i^\times \dot{\omega} = -k_i \rho_i - F \quad (18.16)$$

Substitute  $\rho = \Phi \eta$  and premultiply by  $\Phi'$

$$\Phi' m_i \Phi \ddot{\eta}_i - m_i r_i^\times \dot{\omega} = -\Phi' k_i \Phi \eta_i - F \quad (18.17)$$

Choose  $\Phi$  so that

$$\begin{aligned}\Phi^T m_i \Phi &= E \\ \Phi^T k_i \Phi &= \text{diag}(\sigma_i^2)\end{aligned}\tag{18.18}$$

This decouples the flexible degrees of freedom. It also makes it easier to reduce the order of the model because it can be done on the basis of both the frequency of the mode and its modal participation.

### 18.7.5 Actuators and Sensors

There are two sensors used during the mission orbit. One is the earth sensor that measures roll and pitch. The second is the rate integrating gyro that gives integrated rates as the output. No measurement of absolute yaw attitude is available. The earth sensor is used to measure roll and pitch during all mission orbit modes. During stationkeeping, the yaw gyro is used to measure yaw. This permits a higher bandwidth roll loop for more precise control.

Thrusters, magnetic torquers and the momentum wheel motor are used for control. Alternatively, a single or double pivoted momentum wheel could be used and the pivot motors used for control.

Magnetic torquers depend on the earth's magnetic field vector for their control torques. The magnetic field at geosynchronous altitudes is influenced primarily by the sun. As a consequence, it varies diurnally and with the solar cycle. During periods of intense solar activity the field can even reverse direction.

The main advantage of magnetic torquers is that they do not change the satellite's velocity and they do not consume any propellant. Of course, since they do weigh something (and one must include their impact on the power system, telemetry and command subsystem, etc.) there is a clear trade-off with thrusters which are required for stationkeeping.

The momentum wheel motor is used to control the pitch axis. If there are bias pitch torques, the momentum wheel will eventually saturate and the momentum wheel motor will be unusable until momentum is unloaded with thrusters.

Thrusters are used to change the velocity of the spacecraft and are used when either the magnetic torquers or momentum wheel motor cannot be used. This can happen during magnetic field disturbances or when the errors are too large for the torquers or motor to control. The disadvantage of using thrusters is that they consume fuel and that typically their control threshold is quite large, making precise control difficult.

### 18.7.6 Control System Organization

The flow of the control system is

- Sensor interface processing
- Error computation
- Control torque computation
- Control torque distribution

- Actuator interface processing

The interface processing is the low-level conversion to and from hardware data. The error computation takes the raw sensor data (now in physically meaningful form) and subtracts biases, misalignments, etc.) to produce control system errors. These are processed by the control laws to produce desired torques. The control torques are converted to pulsewidths, in the case of thrusters and magnetic torquers, and voltage, in the case of the momentum wheel motor, by the control torque distribution algorithms. The actuator interface processing then converts the results into values that the hardware can read.

### 18.7.7 Modes

The mission orbit control system has two modes

- Low bandwidth control
- High bandwidth control

Either may be implemented with the momentum wheel motor and magnetic torquers or with thrusters. The former mode uses only the earth sensor for sensing while the latter mode requires the yaw gyro. When thrusters are used, the momentum wheel automatically goes to a preset speed (which can be its current speed.)

The high bandwidth control is normally used for stationkeeping. The low bandwidth mode is used with thrusters to transition from stationkeeping to torquer/momentum wheel motor control, or when the magnetic field is perturbed.

### 18.7.8 Earth Sensor

Earth sensors are of three types, conical scanning, scanning and static. The latter two are the most popular on geosynchronous satellites.

Scanning sensors measure roll by measuring the earth chord at a specific cant angle and difference the measurements to get an approximation to roll. The measured roll angle for a scanning sensor is

$$\theta_{meas} = 2 \left[ \text{acos} \left( \frac{\cos(\rho) - \sin(\delta) \sin(\theta)}{\cos(\delta) \cos(\theta)} \right) - \text{acos} \left( \frac{\cos(\rho) + \sin(\delta) \sin(\theta)}{\cos(\delta) \cos(\theta)} \right) \right] \quad (18.19)$$

where  $\delta$  is the angle of the scan as measured from the plane normal to the sensor scan axis positive about the roll axis,  $\rho$  is the angular radius of the earth, and  $\theta$  is roll.

A major advantage of these sensors is that they give roll and pitch outputs directly. The static earth sensor has sets of thermopiles arranged in a circle about the boresight. Typically each set has three. One looks at the earth all of the time, the second straddles the earth and the third looks at “cold” space. The former and latter are used to calibrate the straddling sensor. The calibrated temperatures of the straddling sensors are used to compute roll and pitch.

### 18.7.9 Gyros

Each gyro has two outputs. When it senses a zero rate it generates a 44 kHz pulse train on each line. The two lines are attached to a counter, one to the upcount and one to the down count. When a 1 LSB positive rate is sensed the upcount line pulse count increases by one pulse per cycle and the downcount line pulse count decreases by one pulse per cycle. Hence the threshold for rate detection is equal to two pulses. The updown counter is read at 4 Hz and the difference between the current and previous values of the counter gives the change in attitude over the sampling interval. Counter overflow must be accounted for in the computation. The integrated change in angle is then passed through a noise filter and used for yaw rate and attitude. The gyro must be initialized by passing it through a low-pass filter during the gyro initialization phase.

### 18.7.10 Noise Filtering

All sensors have significant amounts of random noise that should be filtered prior to use by the control system. This filtering is independent of any filtering that is done by the control loops themselves.

The sensors are sampled at 4 Hz. To get the maximum filtering, the noise filters are designed for a 4 Hz sampling frequency. To minimize the impact on the control system no more than 10 deg of phase shift is acceptable at 0.1 rad/sec.

### 18.7.11 Momentum Wheel Pitch and Tachometer Loops

The basis for the pitch control is a tachometer loop that maintains the speed of the momentum wheel. It is desirable to keep the wheel speed within a range of the nominal speed, normally  $\pm 10\%$  to keep the spacecraft gyroscopically stiff. A DC motor has the transfer function

$$\frac{\omega}{T} = \frac{1}{Js + \beta} \quad (18.20)$$

where  $\beta$  is due to the back emf and viscous friction in the motor. A simple control scheme is to multiply the difference between the desired speed and the measured speed by a gain  $K$  and to filter the measured speed by a first order filter. The resulting closed loop system is

$$\Omega = \frac{(s + \omega_T)(K\Omega_C + T_C)}{Js^2 + (J\omega_T + \beta)s + \omega_T(K + \beta)} \quad (18.21)$$

where  $\omega_T$  is the filter cutoff.  $\omega_T$  is chosen to provide adequate damping and  $K$  is made sufficiently large to provide good disturbance rejection and command tracking.

The input to the tach loop is the wheel speed demand. The equation for the pitch loop is

$$I\ddot{\theta} + J\dot{\Omega} = T \quad (18.22)$$

Since we want a pitch equation of the form

$$I\ddot{\theta} + c\dot{\theta} + k\theta = T \quad (18.23)$$

It follows that the wheel speed demand will be

$$\Omega = \frac{c\theta + k\int\theta}{J} \quad (18.24)$$

which is a proportional-integral controller.

### 18.7.12 Low Bandwidth Roll/Yaw Control

There are two aspects of the roll yaw control system that must be considered. The first is that it must attenuate the external disturbances on the spacecraft. These are at harmonics of the orbit rate and generally have no significant components above twice orbit rate. The second aspect is nutation damping. Thruster firings will tend to excite nutation. The control system must damp the nutation. Since the nutation frequency is much higher than orbit rate we can break the design process into two parts, one to attenuate low frequency disturbances and the second to damp nutation.

The controller will be designed in two parts. The first part takes the low frequency approximation to the open loop system and selects a pair of gains to meet the pointing requirements. This approach, a purely proportional control, is the simplest. An alternative is to estimate yaw and use full state feedback. This system is one input and two output but the two outputs are always in a fixed ratio to each other.

The second part is the nutation damper. We will insert a second order compensator to stabilize

The first step is to reduce the roll/yaw equations to their low-frequency approximation. The roll yaw equations are

$$\begin{aligned} \frac{T_z}{h_w} &= \dot{\theta}_x - \omega_o \theta_z \\ \frac{T_x}{h_w} &= \dot{\theta}_z + \omega_o \theta_x \end{aligned} \quad (18.25)$$

Remember that

$$h = \begin{bmatrix} 0 \\ -h_w \\ 0 \end{bmatrix} \quad v = \begin{bmatrix} 0 \\ -\omega_o \\ 0 \end{bmatrix} \quad (18.26)$$

The torque command is

$$\begin{bmatrix} T_x \\ T_z \end{bmatrix} = h_w \begin{bmatrix} K_{xx} \\ K_{zx} \end{bmatrix} \theta_x \quad (18.27)$$

since only roll is measured. This can be implemented with either a skew dipole or separate torquers since the ratio of the torques is fixed. Alternatively, we could have used a dynamic compensator by having

$$\begin{bmatrix} T_x \\ T_z \end{bmatrix} = h_w \begin{bmatrix} K_{xx} & K_{xz} \\ K_{zx} & K_{zz} \end{bmatrix} \begin{bmatrix} \theta_x \\ \theta_z \end{bmatrix} \quad (18.28)$$

and estimating yaw. The latter approach requires two separate torquers.

The transfer function for the first approach is quite simple and is

$$\begin{bmatrix} \theta_x \\ \theta_z \end{bmatrix} = \frac{1}{h_w s^2 + K_{zx} s + \omega_o^2 - \omega_o K_{xx}} \begin{bmatrix} s & \omega_o \\ -\omega_o + K_{xx} & s + K_{zx} \end{bmatrix} \begin{bmatrix} -T_z \\ T_x \end{bmatrix} \quad (18.29)$$

The disturbance torques are scaled by the bias momentum, therefore large bias momentum will reduce the attitude errors.  $K_{zx}$  provides damping.  $K_{xx}$  must be negative if it is larger in magnitude than  $\omega_o$ , which will almost always be the case. The major disturbance sources will be at DC and orbit rate. The transfer function matrices are

$$\begin{bmatrix} \theta_x \\ \theta_z \end{bmatrix} = \frac{1}{h_w} \begin{bmatrix} 0 & \omega_o \\ -\omega_o + K_{xx} & K_{zx} \end{bmatrix} \begin{bmatrix} -T_z \\ T_x \end{bmatrix} \quad (18.30)$$

The magnitude of the errors for large gains is

$$\begin{bmatrix} \theta_x \\ \theta_z \end{bmatrix} = \frac{T}{\omega_o h_w \sqrt{K_{xx}^2 + K_{zx}^2}} \begin{bmatrix} 2\omega_o \\ |K_{xx}| + K_{zx} \end{bmatrix} \quad (18.31)$$

Increasing either gain will decrease the roll errors. However, the two gains must be carefully balanced to attenuate yaw errors.

The estimator approach is to use full state feedback and to estimate yaw. The estimator is

$$\begin{bmatrix} \dot{\theta}_x \\ \dot{\theta}_z \end{bmatrix} = \begin{bmatrix} 0 & \omega_o \\ -\omega_o & 0 \end{bmatrix} \begin{bmatrix} \theta_x \\ \theta_z \end{bmatrix} + \begin{bmatrix} L_x \\ L_z \end{bmatrix} (z - \theta_x) \quad (18.32)$$

The transfer functions are

$$\begin{aligned}\frac{\theta_x}{z} &= \frac{sL_x + \omega_o L_z}{s^2 + sL_x + \omega_o(\omega_o + L_z)} \\ \frac{\theta_z}{z} &= \frac{sL_z - \omega_o L_x}{s^2 + sL_x + \omega_o(\omega_o + L_z)}\end{aligned}\tag{18.33}$$

at  $s = j\omega_o$  these become

$$\begin{aligned}\frac{\theta_x}{z} &= 1 \\ \frac{\theta_z}{z} &= j\end{aligned}\tag{18.34}$$

At orbit rate the estimator always passes the roll measurement unfiltered to the roll estimate and phase shifts the roll measurement by 90 deg to get the yaw estimate. Notice that the estimator rolls off as  $1/s$  making it sensitive to unmodeled dynamics. The roll to yaw transfer function is nonminimum phase. This limits the gain of any controller using this estimator.

the nutation mode. The next step is to compensate for nutation. First, turn the one-input two output system into a SISO system by appending the gain calculated above.

Several different approaches can be used to stabilize nutation. One is to use a nonminimum phase transfer function of the form

$$\frac{s - a}{s + a}\tag{18.35}$$

The magnitude of this transfer function is always one and the phase is

$$\phi = \text{atan}\left(\frac{-2\omega}{\omega^2 - a^2}\right)\tag{18.36}$$

at large  $\omega$  the phase is  $\text{atan}(-0)$  or -180 deg. Since the magnitude of the phase shift is greater than 90 deg, this is known as a nonminimum phase transfer function. Interestingly enough, the first order Padé approximant to a delay of period  $T$  is

$$\frac{s - \frac{2}{T}}{s + \frac{2}{T}}\tag{18.37}$$

Thus this nonminimum phase transfer function could be implemented as a delay.

The approach presented here is to introduce a complex zero just before the nutation pole pair using the transfer function

$$\frac{s^2 + \omega_z^2}{s^2 + 2\zeta\omega_p + \omega_p^2} \quad (18.38)$$

and a damped pole after the nutation mode. As long as the zero is at a frequency less than that of the nutation pole pair the system will be stable.

### 18.7.13 Thruster Control

Thrusters are used both during stationkeeping and as a backup to the magnetic torquer/momentum wheel control system. For backup purposes, the thruster roll/yaw control uses the same compensator as the magnetic roll/yaw control. However, the pitch control uses a proportional-derivative control.

### 18.7.14 High Bandwidth Roll/Yaw and Pitch Control

In high-bandwidth mode, each axis is controlled independently with a proportional-integral-differential controller. The yaw measurement is taken from the yaw gyro. The yaw gyro does not provide an absolute attitude reference and the control system assumes that the initial yaw attitude is zero. This does not pose a problem as long as the mode is not run for long periods of

The rigid body equations are

$$T = I\dot{\omega} + \omega^\times I\omega \quad (18.39)$$

If rates are small and the angles are small this reduces to

$$T = I\ddot{\theta} \quad (18.40)$$

If we define

$$T = Iu \quad (18.41)$$

Then the axes are decoupled and the control problem is reduced to

$$u = \ddot{\theta} \quad (18.42)$$

or three decoupled double integrators. These equations are valid if the flex modes are at much higher frequencies than the control bandwidth and, in our case, the bandwidth is much higher than the nutation mode. Once the control is computed the control torques are formed by the operation

$$Iu \quad (18.43)$$

which is a 3-by-3 matrix times a 3-by-1 vector multiply. We can now design each axis' control loop independently.

The PID controller is of the form

$$T = K_P + K_R \frac{\omega_R s}{s + \omega_R} + \frac{K_I}{s} \quad (18.44)$$

The derivative term is divided multiplied by a first order filter to prevent differentiation at high frequencies. Rearranging gives

$$\frac{T}{\theta} = \frac{(K_P + K_R \omega_R) s^2 + (K_P \omega_R + K_I) s + (K_P + K_I \omega_R)}{s(s + \omega_R)} \quad (18.45)$$

If  $\omega_R$  is set to  $\infty$  this becomes

$$\frac{T}{\theta} = \frac{K_R s^2 + K_P s + K_I}{s} \quad (18.46)$$

This transfer function is not proper, i.e. it goes to  $\infty$  as  $s$  goes to  $\infty$ . The latter cannot be implemented in state-space form for this reason. In practice this is not a problem if noise filtering is included in the loop.

In this system the plant is not a set of pure double integrators, instead it has the undamped nutation pole. If the bandwidth is set high enough, the nutation pole will be unobservable by the control system. This does not mean it has disappeared and it will be excited by control activity and the disturbance torques. This would pose a problem when the PID was turned off. The normal mode thruster control will damp the nutation and so this is not a problem.

### **18.7.15 Magnetic Torquer Control**

The magnetic torquers are controlled with a timer. Each control period the timer is passed the duration of the magnetic pulse. The pulse always starts immediately.

### **18.7.16 Thruster Control**

The thrusters are controlled with timers. Each control period the timer is passed the duration of each thruster pulse and a delay from the issuing of the command. This permits off-pulse modulation of the thrusters and execution of spin precession maneuvers.

The control algorithms produce a three axis torque demand. This must be translated into pulsewidth demands for the selected thrusters. The thrusters generate unidirectional torques and it is generally not possible to divide the thrusters into bidirectional pairs.

The simplest approach is to use a linear programming approach to torque distribution. This is done using the simplex algorithm. This algorithm uses matrix methods to speed the computation and is optimized for three constraint equations.

### 18.7.17 Actuator Saturation

All actuators have a saturation limit. If the compensator has integrators this can lead to sluggish behavior. The solution is to add anti-windup compensation. This is straightforward to do

Implement the digital controller as

$$\begin{aligned} y_k &= \text{sat}(Cx_k + Du_k) \\ x_{k+1} &= (A - LC)x_k + (B - LD)u_k + Ly_k \end{aligned} \quad (18.47)$$

If  $y$  is not saturated this reduces to the equation

$$\begin{aligned} y_k &= Cx_k + Du_k \\ x_{k+1} &= Ax_k + Bu_k \end{aligned} \quad (18.48)$$

If it is saturated the dynamics of the compensator are governed by the plant  $A - LD$ .  $L$  must be chosen so that the saturated compensator is stable. Making  $A - LD$  have all of its poles at zero is a convenient choice.

### 18.7.18 Thruster Resolution

The pulsewidth is related to the torque demand by the equation

$$\tau = \frac{u}{u_{max}} T \quad (18.49)$$

where  $T$  is the pulsing period,  $u$  is the torque demand and  $u_{max}$  is the maximum torque. The larger  $T$  is the larger  $\tau$  will be for a given torque demand, thus increasing the pulsing period can increase the resolution. This does not mean that the control period must be increased, rather one fires a pulse every  $T$  seconds.  $T$  should be an integral multiple of the control period.

If  $u$  is much less than  $u_{max}$  this will not normally have a major impact on the phase margins for the system. It will have an impact and must be considered when compensating the loop.

An alternative is to add triangle wave dither to the pulse demand. If the dither frequency is high enough, and the plant has low pass filter characteristics, dither can reduce the effect of the minimum pulsewidth. Unfortunately, in a digital control system, the sampling rate puts a limit on the dither frequency hence the dither will appear as a high frequency oscillation. This may not pose a problem but must be accounted for in the jitter budget.

This leads to the problem of the minimum pulsewidth resolution which has a major impact on pointing performance since it leads to a large deadband around zero.

---

## 18.8 Summary

The control system design for *ComStar* is complete, albeit at a preliminary stage. All control loops have been designed and with thirteen years life and a pointing CEP of 0.17 deg the design meets specifications.

Our task was made easier by a relatively stiff array and absence of a yaw pointing requirement. If the array were more flexible we would have had to add compensation to the stationkeeping loops to stabilize the low frequency flex modes. If yaw were important we would have had to design a more sophisticated low frequency yaw controller or add yaw sensing.

The next step in the design is to add realistic actuator and sensor models and nonlinear dynamics models and verify that the design still meets requirements.

Once your design is verified you will have to update it as mass properties change. Periodic updates to both budgets and the disturbance model must be done.

Other things that must still be done are:

- Add in the actuator and sensor dynamics models
- Determine what commands are needed for each controller.
- Determine how to do mode transitions (from stationkeeping to normal mode, for example.)
- Determine what telemetry is needed to monitor the control system.
- Translate the Matlab into flight software.
- Write operational procedures.
- Train the spacecraft operators.

Nonetheless you are well on your way to a complete design!

---

# *Sun Nadir Pointing Control*

---

## *19.1 Introduction*

This chapter provides the theory behind the sun nadir pointing control system embodied in this module. It contains the following sections:

- Introduction
- Coordinate Frames
- Sun Nadir Pointing
- Components
- Attitude Determination
- Control
- Momentum Management
- The Sun Nadir Control Script

A sun nadir pointing control system is designed to point one axis of the spacecraft at the earth and keep the solar array cell face normal aligned with the sun vector. This maximizes power input while keeping earth pointing sensors on target. Sun nadir pointing control systems are generally used on spacecraft in highly inclined orbits where the changes in the sun angle with respect to the orbit plane may be large. A characteristic of sun nadir pointing spacecraft is that its yaw angle changes over the orbit. When the sun angle is near the orbit plane the spacecraft flips nearly 180 degrees at spacecraft noon and midnight. Since the spacecraft must rotate in pitch and yaw, this precludes the use of a momentum bias or spinning design.

This module deals only with sun nadir pointing control. Stationkeeping and acquisition modes are not covered. In addition, the accompanying script does not include logic for reinitializing of the control loops, error handling and other necessary functions that must be included in flight control software.

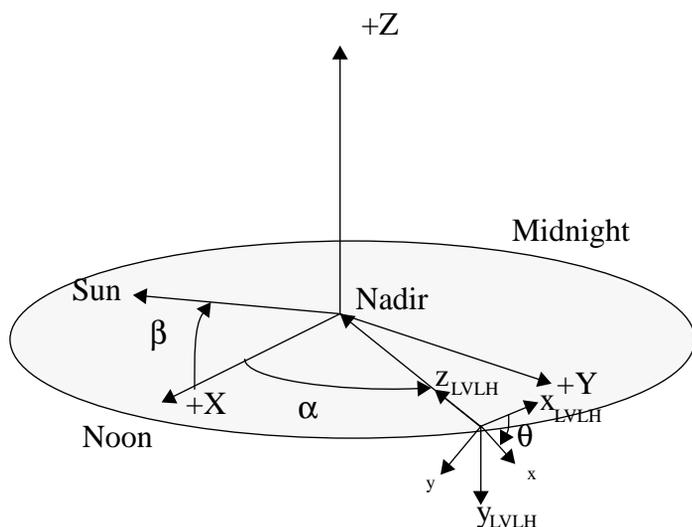
---

## 19.2 Coordinate Frames

The orbit geometry is illustrated below.

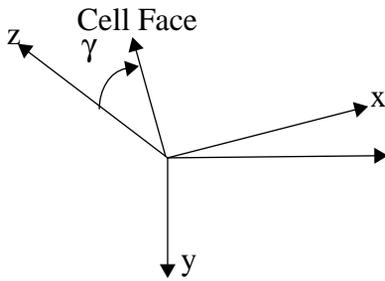
**FIGURE 19.1 Orbit Geometry Diagram**

---



The sun angle is defined as positive above the orbit plane and its projection in the orbit plane is along the  $+X$  axis. The spacecraft vector in the orbit plane is an angle  $\alpha$  away from the sun projection. The Local Vertical Local Horizontal (LVLH) frame is defined with  $+z$  along the nadir vector,  $+y$  along the orbit anti-normal and  $+x$  completing the right-hand set. The spacecraft is yawed an angle  $\theta$  with respect to the LVLH frame. Finally, the solar arrays are oriented at an angle  $\gamma$  with respect to the body frames. When  $\gamma = 0$  the  $+z$  array axis (the cell face) is aligned with the  $+z$  body axis. In this module, the inertial frame will be defined with the  $XY$  axes in the orbit plane, the  $+Z$  along the orbit normal and the  $+X$  axis aligned with the sun projection. This will not normally coincide with standard inertial frames such as the ECI or J2000.0 frames. The solar array frame is illustrated below.

**FIGURE 19.2 Solar Array Frame**




---

### 19.3 Sun Nadir Pointing

The sun vector is

$$\beta = \begin{bmatrix} -\sin\alpha\cos\beta \\ -\sin\beta \\ -\cos\alpha\cos\beta \end{bmatrix} \quad (19.1)$$

The transformation matrix from the array to the core frame is

$$\begin{bmatrix} \cos\gamma & 0 & \sin\gamma \\ 0 & 1 & 0 \\ -\sin\gamma & 0 & \cos\gamma \end{bmatrix} \quad (19.2)$$

The +z unit vector in the array frame is, in the core frame,

$$u = \begin{bmatrix} \sin\gamma \\ 0 \\ \cos\gamma \end{bmatrix} \quad (19.3)$$

The transformation matrix from the core frame to the LVLH frame is

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (19.4)$$

Transforming into the LVLH frame, assuming a yaw rotation of  $\theta$ , and equating to the sun vector gives

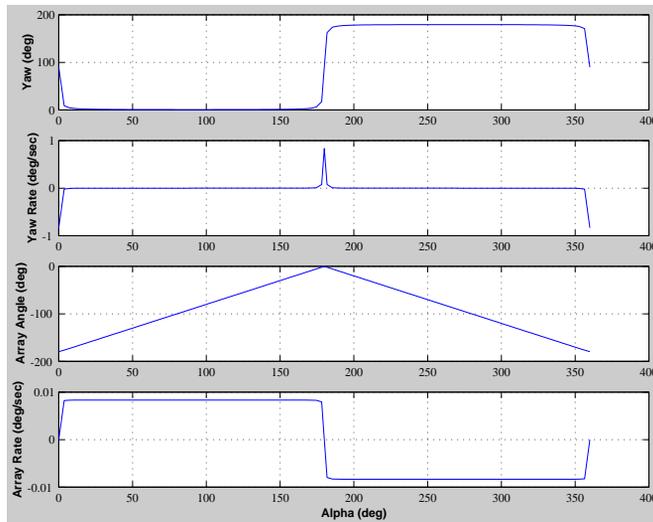
$$\begin{bmatrix} -\sin\alpha\cos\beta \\ -\sin\beta \\ -\cos\alpha\cos\beta \end{bmatrix} = \begin{bmatrix} \cos\theta\sin\gamma \\ \sin\theta\sin\gamma \\ \cos\gamma \end{bmatrix} \quad (19.5)$$

The yaw and solar array angles are

$$\begin{aligned} \theta &= \text{atan2}(\sin\beta, \sin\alpha\cos\beta) \\ \gamma &= \text{atan2}(-\sqrt{(\sin\alpha\cos\beta)^2 + (\sin\beta)^2}, -\cos\alpha\cos\beta) \end{aligned} \quad (19.6)$$

The solar array and yaw angles can be found by equating the sun vector as measured in the LVLH frame, with the solar array normal also measured in the LVLH frame.

**FIGURE 19.3 Sun-nadir yaw trajectory for a GPS orbit**



The plots show trajectories ranging from  $\beta = 0$  deg to  $\beta = 70$  deg. When  $\beta = 0$ , the yaw angle is always zero and the array rotates 360 deg every orbit. For  $\beta = \pm\epsilon$ , the yaw angle changes 180 deg at spacecraft noon and midnight. If reaction wheels are used to perform the noon-midnight turns, the peak yaw rate and yaw inertia will largely determine the momentum requirements of the system.

Because of limitations in the momentum storage (or control authority if using thrusters), it will not be possible to track the ideal yaw trajectory exactly.

Assuming that the solar arrays can always be pointed so that the only error is in the  $yz$ -plane of the solar arrays, the dot product of the sun vector in the solar array plane and the solar array normal will be

$$\begin{aligned}
 \sqrt{1 - u_y^2} &= \cos \varepsilon \\
 1 - u_y^2 &= (\cos \varepsilon)^2 \\
 u_y &= \sin \varepsilon
 \end{aligned} \tag{19.7}$$

$u_y$  is found by transforming the sun vector in the LVLH frame to the core frame; therefore

$$\begin{aligned}
 \sin \theta \sin \alpha \cos \beta - \cos \theta \sin \beta &= \sin \varepsilon \\
 \theta - \theta_{\text{Ideal}} &= \text{asin} \left( \frac{\varepsilon}{\sqrt{(\sin \alpha \cos \beta)^2 + (\sin \beta)^2}} \right)
 \end{aligned} \tag{19.8}$$

The second equation is found using the relationship

$$A \sin(a + b) = A \sin a \cos b + A \cos a \sin b \tag{19.9}$$

and gives us the relationship between angular error and deviation from the ideal trajectory. This can be implemented by feeding an offset as a function of  $\alpha$  and  $\beta$  into the yaw error measurement. When  $\beta$  is small this becomes

$$\theta - \theta_{\text{Ideal}} = \text{asin} \left( \frac{\varepsilon}{\sin \alpha} \right) \tag{19.10}$$

or

$$\varepsilon = \sin \alpha \sin(\theta - \theta_{\text{Ideal}}) \tag{19.11}$$

which implies that even large errors at noon and midnight cause only small errors in the solar array pointing.

---

## 19.4 Components

### 19.4.1 Sensors

Three axis sensing is required for a sun nadir pointing spacecraft. One configuration is to measure roll and pitch with an earth sensor and yaw with a sun sensor. If the sun sensor is array mounted three axis information will be available except in a narrow region about spacecraft noon and midnight. The sensing gaps can be bridged with gyros or (since as discussed above the solar array pointing error becomes less sensitive to yaw error as the spacecraft approaches the noon/midnight regions) rely on ephemeris information to estimate yaw.

Another possibility is to use a star sensor for all three axes and rely on ephemeris information to determine the orientation of the nadir vector. For very precise applications, the star sensor could be used in conjunction with gyros. For

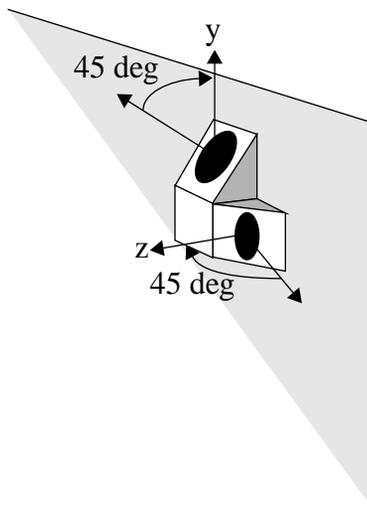
less precise pointing, the star sensor could be used by itself. If gyros are used, a bright star sensor may be appropriate.

In this module, the combination of a conical scanning earth sensor and array mounted sun sensors will be used. This is a relatively low cost option and should provide adequate pointing for a GPS satellite type application (0.1 deg in roll and pitch and 3 deg solar array pointing error.)

Four sun sensor heads are mounted on the arrays, two on each array. The sensor geometry is illustrated below.

**FIGURE 19.4** Sensor geometry

---

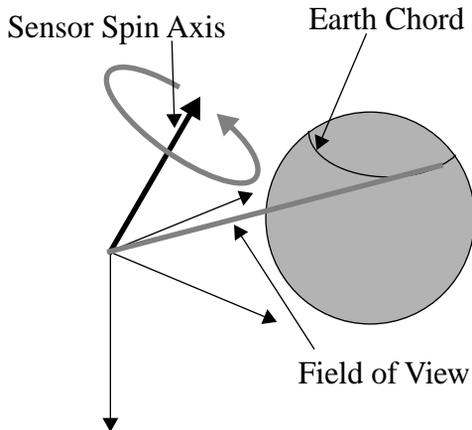


One sensor's boresight is in the  $yz$ -array plane the other is in the  $xz$ -array plane. The first is canted 45 deg from  $y$  and the other 45 deg from  $+z$ . The elements on the other array are canted in the opposite direction so that the  $yz$  sensor points down (in the picture) and the  $xz$  sensor points to the left.

The earth sensor has a motor that rotates a mirror or prism that sweeps the field of view of the sensing element across the earth.

**FIGURE 19.5 Earth sensor geometry diagram**

---



Pitch is found by measuring the center of the chord with respect to an internal reference and roll is found from the length of the chord. If the scan angle is set properly the scan will traverse the northern or southern hemisphere of the earth resulting in a one to one relationship between chordwidth and attitude. This will not be the case for large attitude errors.

### 19.4.2 Actuators

This design uses four reaction wheels mounted in a pyramid about the yaw axis. All four wheels run at the same time but only three are needed for three axis control. Reaction wheels make sense in this application because the spacecraft must maneuver about both the pitch and yaw axes, and since the maneuvers are cyclic, the momentum can be transferred back and forth between the core and the reaction wheels without the use of any consumables.

Thrusters will be required to perform orbit adjust maneuvers. Since misalignments and thrust mismatches in the orbit adjust thrusters may be significant. Thrusters will also be needed for attitude control during maneuvers.

Momentum unloading can be performed by thrusters or by magnetic torquers.

The torque produced by a torquer of dipole  $m$  is

$$T = m \times B \tag{19.12}$$

where

$$m^\times = \begin{bmatrix} 0 & -m_z & m_y \\ m_z & 0 & -m_x \\ -m_y & m_x & 0 \end{bmatrix} \quad (19.13)$$

The above matrix cannot be inverted; consequently, it is not possible to realize a three axis torque demand with magnetic torquers. However, if  $B$  changes it will be possible on average to satisfy an average three axis demand. Assume that we want to drive the momentum vector  $h$  to zero over an interval  $T$ . Write the Hamiltonian

$$H = \frac{1}{2} m^T m - \lambda^T B^\times m \quad (19.14)$$

The control law is found by taking the derivative of  $H$  with respect to  $m$  and setting the result to zero

$$m = -B^\times \lambda \quad (19.15)$$

$\lambda$  is a constant vector. Therefore, substituting the above into the constraint equation gives

$$\begin{aligned} h &= \int_0^T B^\times m dt \\ h &= - \left[ \int_0^T B^\times B^\times dt \right] \lambda \\ \lambda &= - \left[ \int_0^T B^\times B^\times dt \right]^{-1} h \end{aligned} \quad (19.16)$$

If the integral is invertible, it will be possible to drive  $h$  to zero over the interval  $T$ . It is necessary to know  $B$  over the time interval  $T$  which requires an onboard ephemeris.

Magnetic torquers produce torque by interacting with the earth's magnetic field.

---

## 19.5 Attitude Determination

### 19.5.1 Roll

A conical scanning earth sensor returns the earth chordwidth as the measurement. The chordwidth is related to the nadir angle, the angle between the spin axis of the sensor and the nadir vector, by the relationship

$$\cos \rho = \sin \eta \sin \gamma \cos \frac{\Omega}{2} + \cos \gamma \cos \eta \quad (19.17)$$

where  $\rho$  is the earth angular radius,  $\eta$  is the nadir vector,  $\Omega$  is the chordwidth and  $\gamma$  is the cant angle between the spin axis of the sensor and the boresight. Using the sum of angles trigonometric identity this becomes

$$\cos \left( \eta + \operatorname{atan} \left( \tan \gamma \cos \frac{\Omega}{2} \right) \right) = \frac{\cos \rho}{\sqrt{\left( \sin \gamma \cos \frac{\Omega}{2} \right)^2 + (\cos \gamma)^2}} \quad (19.18)$$

The nadir angle is related to roll through the spherical triangle identity

$$\cos \left( \theta_x + \delta - \frac{\pi}{2} \right) = \cos \theta_y \cos \eta \quad (19.19)$$

where  $\delta$  is the offset between the spin axis and the spacecraft pitch axis.  $\theta_y$  is the spacecraft pitch angle.

## 19.5.2 Pitch

The pitch measurement is just the average of the earth rise and fall times with respect to a fixed reference in the sensor.

### 19.5.3 Sun Sensor Eye Pre-Processing

The output from each sun sensor eye is an analog voltage. We need to recover the value of the dot product between the sun unit vector and the eye boresight. The response of the sensor can be written as

$$v = \sum_{k=0}^N a_k x^k \quad (19.20)$$

where  $x = \cos \theta$ . If

$$a_k \ll a_1 \quad k > 1 \quad (19.21)$$

then  $x$  can be extracted from  $v$  using a Newton-Raphson algorithm.

### 19.5.4 Solar Array Pitch

The solar array pitch measurement is computed from the sun sensor eyes with components in the  $\pm x$  directions. The algorithm is

$$\theta_{\text{SAY}} = \text{asin}\left(\frac{\Delta m}{\sqrt{2}}\right) \quad (19.22)$$

where  $\Delta m$  is the difference between the +x and -x eye measurements.

This is

$$u_y = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & -\theta_y \\ 0 & 1 & \theta_x \\ \theta_y & -\theta_x & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x \\ s_y \\ s_z \end{bmatrix} \quad (19.23)$$

### 19.5.5 Yaw

The measurement from the  $\pm y$  eyes is of the y-component of the sun vector in the array frame. Solving for yaw in terms of  $u_y$  and  $s$  gives

$$u_y = s_y \cos\theta - s_x \sin\theta + s_z \theta_x \quad (19.24)$$

Using the sum of angles trigonometric identity

$$\theta = \text{asin}\left(\frac{u_y - s_z \theta_x}{\sqrt{s_x^2 + s_y^2}}\right) - \text{atan}\left(\frac{s_y}{-s_x}\right) \quad (19.25)$$

The  $u_y$  component is just

$$u_y = \frac{\Delta m}{\sqrt{2}} \quad (19.26)$$

where  $\Delta m$  is the difference between the +y and -y eyes. This equation assumes that  $\theta_x$  is small, which it will be during all normal operations.

The equation in Table 61 shows the sensitivity of the yaw measurement to geometry. When  $s_x$  and  $s_y = 0$  the sun lies along the z-axis and the sun sensor measurement is completely insensitive to yaw. When the x and y components are small, the measurement will be sensitive to yaw but noise will be amplified due to the geometry. Consequently, there will be some region around spacecraft noon and midnight where it will be difficult to get good yaw measurements from the array mounted sun sensors. In these regions, gyros may be used or yaw estimated from roll and spacecraft momentum.

## 19.6 Control

### 19.6.1 Reaction Wheel Loop

Reaction wheels are a motor with a flywheel attached. The motor base is attached to the spacecraft. A torque applied to the spacecraft by the motor causes the wheel to rotate in the other direction. The reaction wheel will experience friction due to bearings, windage and other sources. The friction can be modeled as

$$\begin{aligned} T_F &= -\operatorname{sgn}(\omega)T_C - c\Omega & |\Omega| \geq \Omega_L \\ T_F &= \left(-\left(\frac{T_C + c\Omega_L}{\Omega_L}\right)\right)\Omega & |\Omega| < \Omega_L \end{aligned} \quad (19.27)$$

where  $\operatorname{sgn}$  gives the sign of the argument.

A pure damping model is used at low speeds to insure that the wheel speed damps to zero in a simulation. Otherwise, an unrealistic limit cycle will be caused by finite size numerical integration. The reaction torque on the spacecraft is

$$T_R = T_M - T_F \quad (19.28)$$

Consequently, without knowledge of the friction there will always be a wheel speed dependent error. The vector angular acceleration demand on the wheels is

$$\dot{\Omega} = \frac{K_p\theta + K_D\dot{\theta} + K_I\int\theta}{J} \quad (19.29)$$

This must be distributed among the wheels. If three orthogonal wheels are used then each component of  $\dot{\Omega}$  is sent to one of the three wheels. However, if more than three wheels are used it will be necessary to distribute the three acceleration components among more than three wheels. An additional relationship must be introduced. It is required that

$$\dot{\Omega} = U\dot{\Omega}_W \quad (19.30)$$

where  $U$  is a 3-by- $n$  matrix where  $n \geq 3$  of the unit vectors of the reaction wheel spin axes.

We want  $\dot{\Omega}_W$  chosen so that

$$J = \frac{1}{2}\dot{\Omega}_W^T\dot{\Omega}_W + \lambda^T(\dot{\Omega} - U\dot{\Omega}_W) \quad (19.31)$$

is minimized.  $\lambda$  is a 3-vector of Lagrange multipliers that adjoin the constraint equations to the scalar cost equation.

Taking derivatives of this equation with respect to  $\dot{\Omega}_W$  and  $\lambda$  gives the two vector equations

$$\begin{aligned}\dot{\Omega} &= U\dot{\Omega}_W \\ \dot{\Omega}_W &= U^T\lambda\end{aligned}\tag{19.32}$$

An alternative approach is to command wheel speed changes instead of torques. This can be done because the equations of motion (neglecting Euler coupling) are

$$I\ddot{\theta} + J(\dot{\Omega} + \dot{\omega}) = T\tag{19.33}$$

If we make

$$J\dot{\Omega} = K_p\theta + K_D\dot{\theta} + K_I\int\theta\tag{19.34}$$

then we can use tach loops for each wheel to control the spacecraft.

### 19.6.2 Attitude Loop

Since there will be steady external disturbances on the spacecraft, the attitude loops must contain integral action.

A general form for a single-axis proportional-integral-differential (PID) controller is

$$T = K\left(1 + \tau_R\left(\frac{\omega_{R^s}}{s + \omega_R}\right) + \frac{1}{\tau_I s}\right)\tag{19.35}$$

This is generally implemented in a digital control system. If multi-rate sampling is available, it may be advantageous to run the rate filter at a higher rate than the rest of the controller.

### 19.6.3 Solar Array Control

The solar array pitch measurements are translated into solar array steps, and the step demand is fed to the solar array stepping motor. The stepping motor acts like a moderately damped second order system.

### 19.6.4 Momentum Control

The spacecraft is nominally zero momentum in the absence of external disturbances. Of course, external disturbances will cause the momentum to grow and this momentum growth must be controlled. The reaction wheels can only exchange momentum with the body hence some other type of actuator is required to remove momentum. The obvious choices are either thrusters or magnetic torquers. Magnetic torquers have the advantage that they do not produce a force on the spacecraft and therefore do not perturb the orbit, a major advantage if the satellite is used for navigation purposes. Thrusters can remove much more momentum in a given period of time. In addition, magnetic torquers can only remove momentum about two axes at one time although on average, if the magnetic field direction in the body frame changes, they can remove momentum in all three axes.

In this module, we will not be concerned with how the momentum is removed. Instead, we will look at the momentum control law and assume that there is a three axis means for removing the excess momentum. The momentum in the system is

$$h = (I + J) \begin{bmatrix} 0 \\ -\omega_o \\ \omega_z \end{bmatrix} + JU \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix} \quad (19.36)$$

and the momentum control law is

$$T = -Kh \quad (19.37)$$

The gain is set so that it balances the disturbance terms that cause the momentum to build up. Generally, the torque command will be very small.



---

## *20.1 Introduction*

This chapter discusses a critical aspect of ACS design testing. It is important to have a testing program that starts with the first algorithms or code written and ends with the decommissioning of the spacecraft. A properly crafted testing program can dramatically reduce program costs by reducing the number of errors that get transmitted to the next stage of the design where fixes are more costly and ultimately, when problems are discovered on-orbit, can and have led to the loss of a mission.

This chapter has two parts. The first gives industry experience in testing and different approaches to testing. The second section discusses a test approach that has been used successfully at Princeton Satellite Systems for a number of years.

---

## *20.2 Industry Examples*

### **20.2.1 Introduction**

This section discusses the experiences of satellite manufacturers in testing the attitude control systems for satellites. This section provides a context for the discussion in the first section.

Many of the papers cited were written in the early 1980's when satellite manufacturers were beginning to see the value of building interactive digital satellite simulation. This had been done in the 1960's at the Draper Laboratory for the Apollo and Space Shuttle Programs but it was only with the introduction of 16 bit and 32 bit minicomputers, such as the PDP-11 and VAX series, that it became practical to build closed-loop interactive simulations.

### **20.2.2 Telecom 1**

The Telecom 1 spacecraft is a bias momentum geosynchronous communications satellite built by Matra. The spacecraft spins in transfer orbit and uses a solid rocket motor for injection into the geosynchronous orbit. It is a bias momentum 3-axis stabilized in its mission orbit. It employs an infrared earth sensor (IRES) for roll and pitch measurements and a two-axis digital sun sensor for yaw measurements.

Matra developed a hardware-in-the-loop simulation to test the Attitude and Orbit Control System (AOCS). The test goals were

- check the electrical compatibility between all AOCS units (interface—harness—grounding diagram),
- provide beginning-of-life (BOL) AOCS performance validation including the power budgets for the AOCS,
- check the on-board software integration in the subsystem,
- elaborate and check all telecommand procedures, and telemetry allocations used for the operational requirements handbook,
- test the redundancy management system,
- provide for operators training,
- permit reconstitution of events observed on-orbit.

The simulator included the flight processor box, wiring harness, all hardware interfaces and an IRES on a rotating table with an earth simulator. The spacecraft simulation ran on a digital computer. This system was instrumental in finding a number of serious errors including

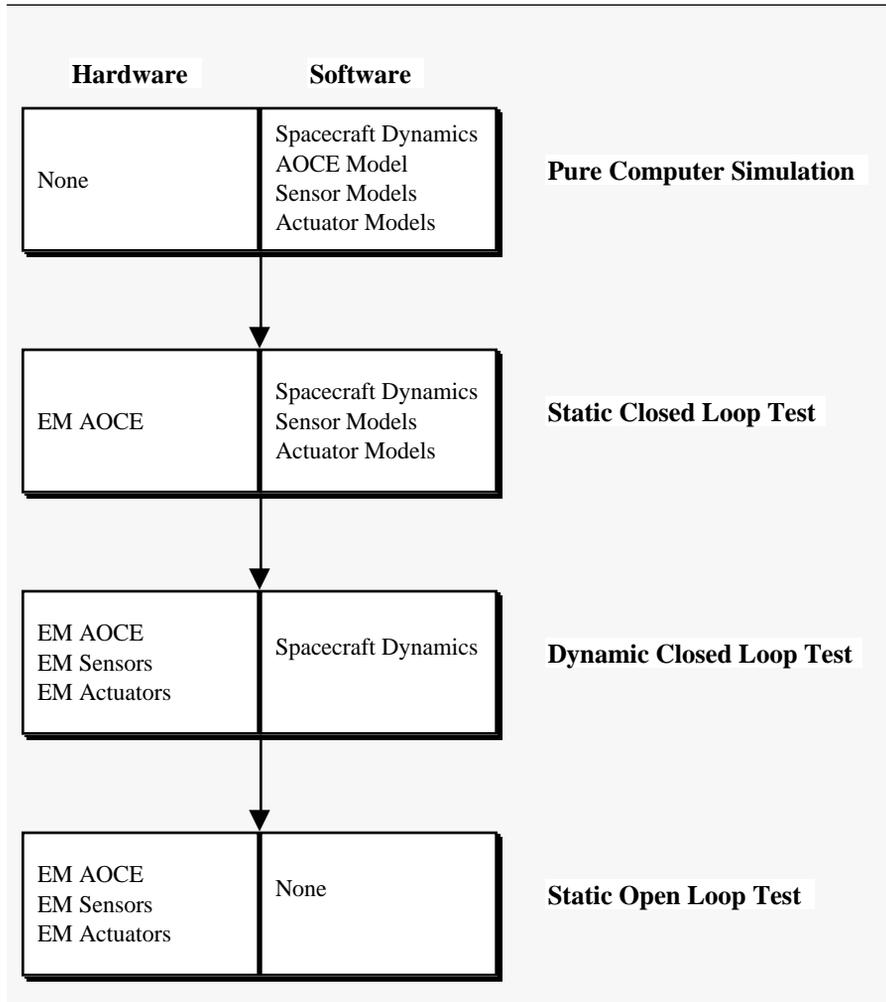
- four incompatible hardware interfaces,
- AOCS sensitivity to common mode due to large transient currents when the unit was switched on—required two modifications in the grounding diagram,
- electronic saturation in the wheel control that occurred in a failure case that was not predicted in studies.

Although software problems were found, this simulation was not designed to be a flight software testbed.

### 20.2.3 MOS-1

MOS-1 is a sun-synchronous earth observation satellite. The AOCS sensing hardware includes sun sensors, rate

**Figure 20-1 MOS-1 Test Flow**

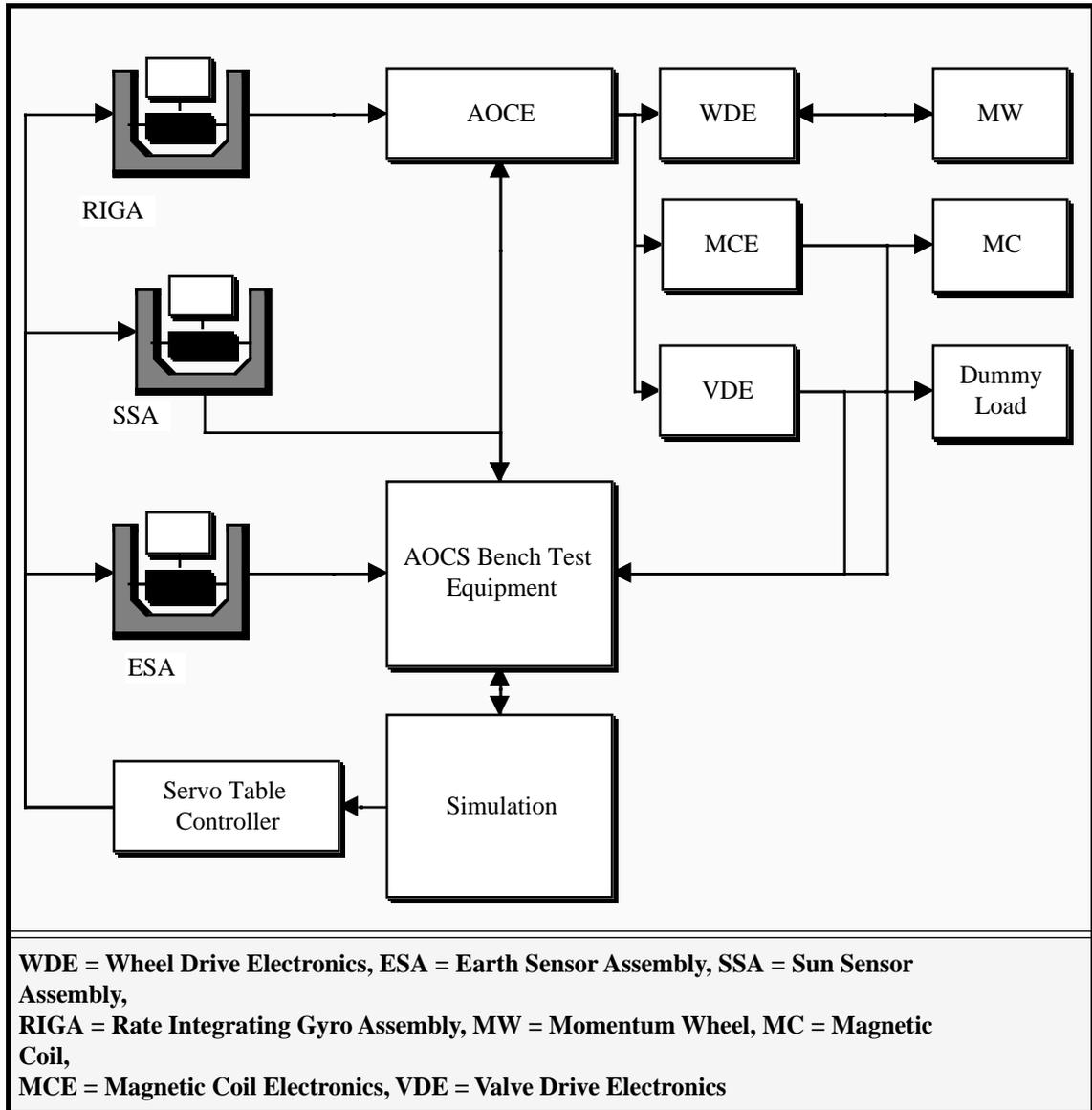


gyros and earth sensors. Actuators include magnetic torquers, thrusters and momentum wheels.

The AOCS subsystem test followed the plan shown in Figure 20-1. Initial tests were with all-software simulations and final tests were static open loop tests with the engineering model hardware. The subsystem hardware test results were verified by comparing them with the all-software simulation results.

The hardware set up is illustrated in Figure 20-2. The AOCE, interfaces, and AOCS sensors are integrated through the

**Figure 20-2 MOS-1 Test Setup**



AOCS Bench Test Equipment. The all-software simulator can be run either by itself or integrated with the hardware.

A critical item was the software configuration control. Both models and the data were maintained under configuration management. The system and AOCS parameters were stored in a common database. The initial conditions for each test were stored with the test results. The version management approach employed a manual documentation sys-

tem, and at times the system broke down leading to confusion about which test data went with which version. Future configuration management systems be enhanced.

### 20.2.4 IRAS

The Infrared Astronomy Satellite (IRAS) is an inertially pointing scientific satellite built by Fokker. The attitude control system is very complex with 15 different ACS modes, all software controlled. Reprogramming is available. The actuator and sensor complement includes reaction wheels, magnetic torquers, horizon sensors, coarse and fine sun sensors, magnetometers and a star sensor. Because of its complexity, a multi-phase test approach was developed.

**TABLE 20-1 IRAS Test Program**

Test	Description	Objective	Errors
ACM Test	<ul style="list-style-type: none"> <li>• One axis air bearing test</li> <li>• Only ACS units, excluding star sensor and OBC</li> <li>• Sensors and on-board software adapted to one-axis configuration</li> <li>• Test facility determined test computer</li> </ul>	Verify ACS performance	41 Anomalies 3% OBC 36% OBS 24% TE 37% ACS Units
EM	<ul style="list-style-type: none"> <li>• ACS EM units integrated into spacecraft system</li> <li>• Tests controlled by GCE containing S/C dynamics model and stimuli controls</li> <li>• No optical part of FSS and HSE</li> <li>• Most unit stimuli analog</li> <li>• Three test categories</li> <li>• Open loop unit level</li> <li>• Closed loop ROM functions</li> <li>• Open loop RAM functions</li> </ul>	Verify ACS performance	100 Anomalies 5% OBC 62% OBS 14% TE 19% ACS Units
<p><b>OBS = on board software, OBC = on board computer, TE = test equipment, S/S = star sensor, FM= flight model, EM = engineering model, GCE = ground computer equipment, HSE = horizon sensor electronics, S/C = spacecraft</b></p>			

**TABLE 20-1 IRAS Test Program**

Test	Description	Objective	Errors
FM	<ul style="list-style-type: none"> <li>• ACS FM Units integrated in S/C</li> <li>• One axis air bearing test</li> <li>• Coarse physical stimuli for all units</li> <li>• Test controlled by GCE, containing Stimuli commands</li> <li>• ACS FM units integrated into S/C</li> <li>• ACS EM Test, open loop unit level</li> </ul>	Verify Correct Polarity of ACS function Verify correct electrical function of S/S Monitor Consistency Detect degradation	148 Anomalies 12% OBC 23% OBS 47% TE 18% ACS Units
Flight			19 Anomalies 58% OBS 21% Ground S/W 21% ACS Units
<b>OBS = on board software, OBC = on board computer, TE = test equipment, S/S = star sensor, FM= flight model, EM = engineering model, GCE = ground computer equipment, HSE = horizon sensor electronics, S/C = spacecraft</b>			

Table 20-1, on page 253 lists the tests and the phases. In this table ACS units refers to any ACS component except the OBC.

Overall, the on-board computer and on-board software were the major cause of anomalies. The second largest cause was the test equipment and the test software. The test software was the major cause of the test equipment anomalies. Because of this, a special effort was required for software testing, documentation and configuration control.

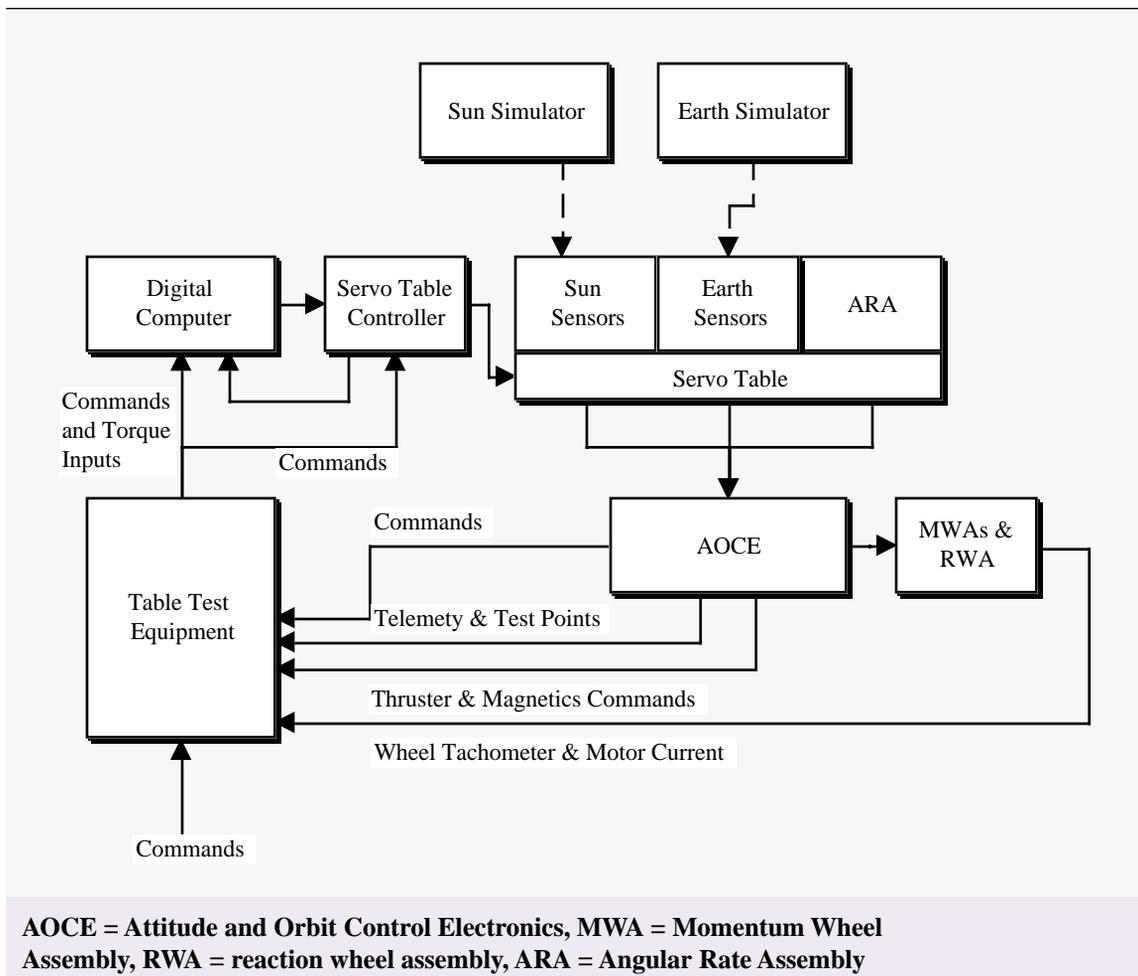
The number of in-flight anomalies was quite small considering the complexity of the control system. However, of the eleven referenced, five were caused by inadequate testing. Three were due to a total lack of testing of the function, and the remaining two were only tested by the designers of the algorithms.

### 20.2.5 INSAT I

The Indian National Satellite I (INSAT I) is a three-axis stabilized general purpose satellite. It employs a microprocessor based control system. Three reaction wheels, one along yaw and two in a V configuration about pitch, are used for primary control. A bipropellant propulsion system is used for both attitude control and apogee boost. Large angle coarse sun sensors are used for acquisition. Infrared earth sensors and a digital sun sensor are used for normal mode control. Rate gyros are used for despin and attitude control during stationkeeping.

INSAT I was tested using a servo table in a closed-loop simulation facility. Sun and Earth simulators were employed

**Figure 20-3 INSAT I Test Facility Block Diagram**



as sources for the sun and earth sensors. The Table Test Equipment (TTE) provided the interfaces between the AOCE and the simulation computer. The TTE provided a simulation of the spacecraft TT&C command and telemetry units. It also provided buffering of all signals for inputs into the digital computer, in this case a PDP-11/70, and a means for monitoring the outputs of the AOCS including thruster drive signals, wheel torque commands and magnetic torquer control signals. The TTE also provided strip chart and magnetic tape recording capabilities.

The earth simulator was an 8.3 inch diameter aperture backed by a hotplate with a peak radiance in the 14 to 16  $\mu\text{m}$  band. The sun simulator was a 2.5 kW Xenon lamp modified to provide a collimated beam.

The three-axis motion simulator was an electrically driven servo table allowing continuous rotation about all three axes. It stands on a concrete slab anchored to the floor of the building. The sun simulator is mounted on the same

slab. The table provides 100 slip rings for instrumentation and custom designed hot/cold temperature chambers can be mounted on the inner gimbal platform for environmental testing.

The purpose of the AOCS test program was to

- validate the design of the Attitude and Orbit Control Subsystem
- verify the closed loop functioning of the AOCS engineering model hardware in the various configurations,
- demonstrate conformance of the subsystem characteristics with the overall performance specifications,
- provide data to validate the computer simulations run during the control loop design phase,
- provide operator training,
- help develop contingency plans.

This facility was used to verify the AOCS capabilities listed in Table 20-2, on page 257. Bench testing of the AOCS

**TABLE 20-2 INSAT I AOCS Testing**

<b>Capability</b>	<b>Performance Measurements</b>	<b>Test Purpose</b>
Acquisition	Sun tracking ability. Earth pointing performance. Thruster firing rates.	To demonstrate the satellite capability to acquire the sun and the earth from a specified set of initial dynamics conditions.
Apogee Boost	Thruster vector pointing accuracy. Thruster firing rates. Transient response at motor start and stop.	Demonstrate the satellite's ability to hold the proper attitude in roll, pitch and yaw.
Stationkeeping	Pointing accuracy. Gain and phase margins. Transient response at the beginning and end of maneuvers.	Satellite stability.
On-Orbit	Pointing accuracy. Wheel unloading. Short and long term stability.	Pointing accuracy and stability.
Antenna Bias Command	Attitude bias range.	Requirements verification.
Sun/Moon Interference	Subsystem response.	Effects of interference.
Failure Recovery	Mode transitions.	Demonstrate redman and reacquisition.
Interface	Telemetry output. Command input. Power consumption. Propulsion interface.	Verify subsystem interfaces.

component testing was done prior to the servo-table tests.

One problem with this facility was the large lag in the servo table. This tended to reduce the phase margins in the control loops. This had to be accounted for in all control verification simulations.

### **20.2.6 Intelsat V**

Intelsat V has a real-time closed loop simulator that interfaces with the operator consoles for operator training. The simulator includes an engineering model Attitude Determination and Control Electronics (ADCE) and the Fine Digital Sun Sensor Electronics (FDSSE). These are connected to both the Command and Telemetry Processing (CTP),

and to a general purpose computer that hosts the simulation. The simulation has moderate to high fidelity models of the spacecraft including detailed sensor and actuator models, and dynamics including the lowest frequency flexible modes of the solar arrays. The models include failure modes.

The primary purpose of the simulation is for operator training, exploring failure modes, and in the definition of operational procedures. It does not appear to have any significant capability as a software development station and appears to lack any significant software test or development tools.

### **20.2.7 Space Telescope**

The original Space Telescope Simulator was used with analog and discrete digital models of the spacecraft and attitude control hardware. The development testing was done on this system, including both static tests and dynamic tests on a three-axis airbearing table. The spacecraft model, implemented in analog form, was extremely simple and included only reaction wheel assembly (RWA) and rigid body effects. A key feature of this effort was an extensive effort to calibrate and measure the input/output response of the actuators and sensors to validate the models used in the simulation.

It soon became apparent that the number of changes needed to support the software validation and test phase required a transition to a hybrid simulation where the spacecraft, and its attitude control hardware, would be simulated on a digital computer. A PDP-11/44 was chosen and the simulation was written in FORTRAN 77. Although this was a major improvement, the extremely limited computational throughput of the PDP-11, still placed a severe limitation on the fidelity of the spacecraft models. For example, it would not be possible to include a detailed model of the solar array panels and such disturbances as thermal snap. As a consequence, this system, while adequate for software testing, cannot be used for performance verification.

### **20.2.8 MBB**

MBB performs both open and closed loop tests on their AOCS systems. Both tests involve hardware in the loop. This includes the actuators, sensors and the interfaces to the flight computer. The advantages and disadvantages of each

are listed in Table 20-3, on page 259. The MBB closed loop setup is similar to that for INSAT I. Closed loop testing,

**TABLE 20-3 Trade-off Between Open and Closed Loop Testing**

	Open Loop	Closed Loop
Advantages	<ul style="list-style-type: none"> <li>• High sensitivity and accuracy</li> <li>• Easy localization of (primarily static) errors</li> </ul>	<ul style="list-style-type: none"> <li>• Component operation takes place under realistic signal conditions</li> <li>• Signal sign and proper arrangement is easily verified</li> <li>• Testing of interface compatibility between subsystem components is covered</li> <li>• Operational subsystem performance evaluation as compared to requirements is easy to perform</li> <li>• Test results represent reference recordings for in-flight measurements</li> <li>• Comparison with theoretical analyses and simulation results is possible</li> <li>• Verification of mathematical and simulation models is achieved</li> <li>• Validity of sensitivity analyses is check</li> <li>• Testing of closed loop system with open loop sensitivity is possible using model reference techniques</li> </ul>
Disadvantages	Interface incompatibilities with other equipment and unforeseen dynamic phenomena are difficult to detect.	<ul style="list-style-type: none"> <li>• Higher complexity</li> <li>• Localization of error sources is difficult due to error propagation along the loop.</li> </ul>

including tests done in thermal vacuum have led to discovery of a number of significant problems in satellites. The most notable are listed in Table 20-4, on page 260. As it turned out, some of these errors were temperature dependent

and would not have been discovered if closed-loop thermal vacuum testing had not been performed. This shows the

**TABLE 20-4 Problems Discovered During Closed-Loop Testing**

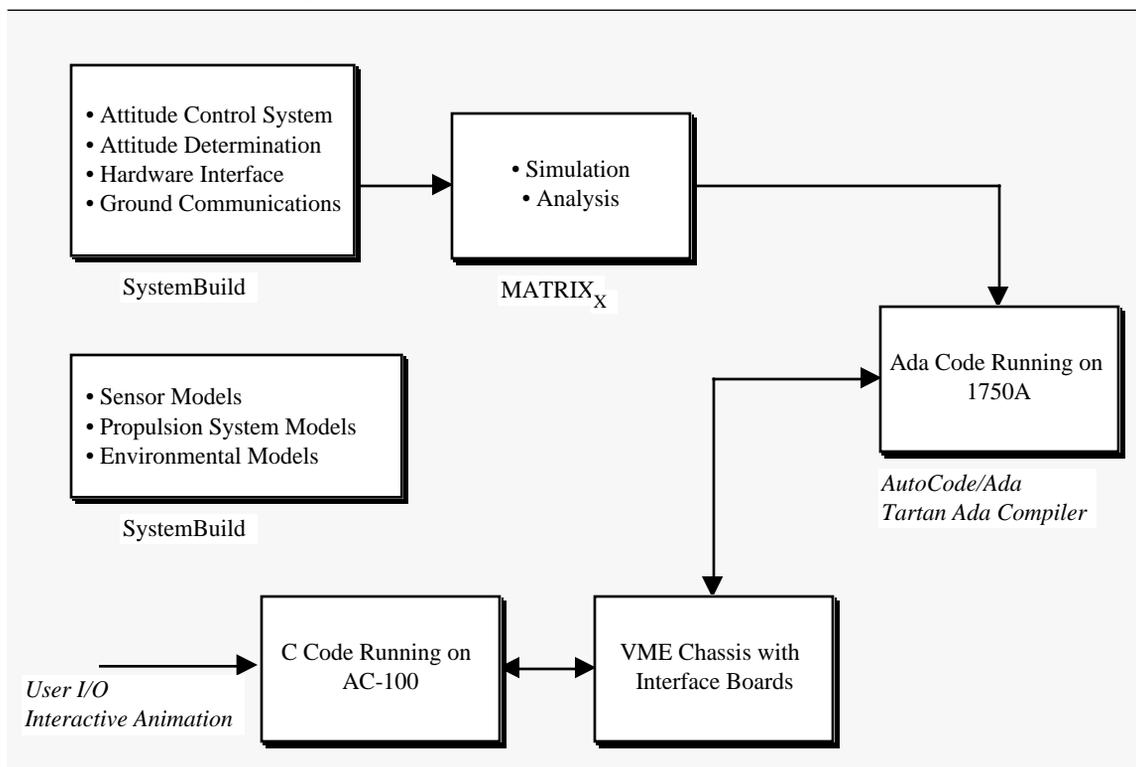
Satellite	Problems
Symphonie	<ul style="list-style-type: none"> <li>● Incorrect modeling of IR-sensor noise (in particular the low frequency range.)</li> <li>● Coupling between roll and yaw in the sun sensor which could have led to instability.</li> </ul>
Intelsat V	<ul style="list-style-type: none"> <li>● GEO IRS susceptible to acoustic noise generated by the wheels and thrusters. Redesign of the brackets was required.</li> <li>● In stationkeeping mode, panel instability was detected that caused an amplitude dependent phase shift of up to 40° in the FDSS. The sampling frequency had to be changed by a factor of 30.</li> <li>● Temperature drift in the GEO-IRS that could cause up to 0.3° error in pitch. This led to a design change in the suspension.</li> <li>● Error in the eclipse logic caused a spike in the output signal when north or south scan was inhibited by ground command. The switching logic had to be changed.</li> <li>● Cross-coupling in the harness caused disturbances in the wheel drive electronics during wheel spin-up. This caused unwanted wheel unloading.</li> </ul>
TV-SAT	<ul style="list-style-type: none"> <li>● Sign error in the earth capture mode</li> <li>● Various overflows due to scaling errors</li> <li>● Errors in switching matrix.</li> </ul>

need for complete simulations of the space environment.

### 20.2.9 MSTI and DC-X

The flight code for the Miniature Sensor Technology Integration (MSTI) satellite was done by Integrated Systems (ISI) using the MATRIX<sub>X</sub><sup>®</sup> Product Family. The design flow is illustrated in Figure 20-4. The target languages were

**Figure 20-4 MSTI Design Flow**



Ada and assembly language. The flight processor was based on a 1750A architecture. The control system algorithms were implemented using the MATRIX<sub>X</sub> block diagram language and converted to Ada using ISI's AutoCode™/Ada package. Supporting functions, including command and data handling and telemetry processing, were hand coded. The spacecraft models were also designed in the block diagram language and translated into C for the AC-100 using ISI's AutoCode™/C package.

The initial design employed only the processor in the loop with the sensor and actuator interface boards hooked into the AC-100. In the second phase of testing the actual flight sensors and thruster valve relays were connected to the system.

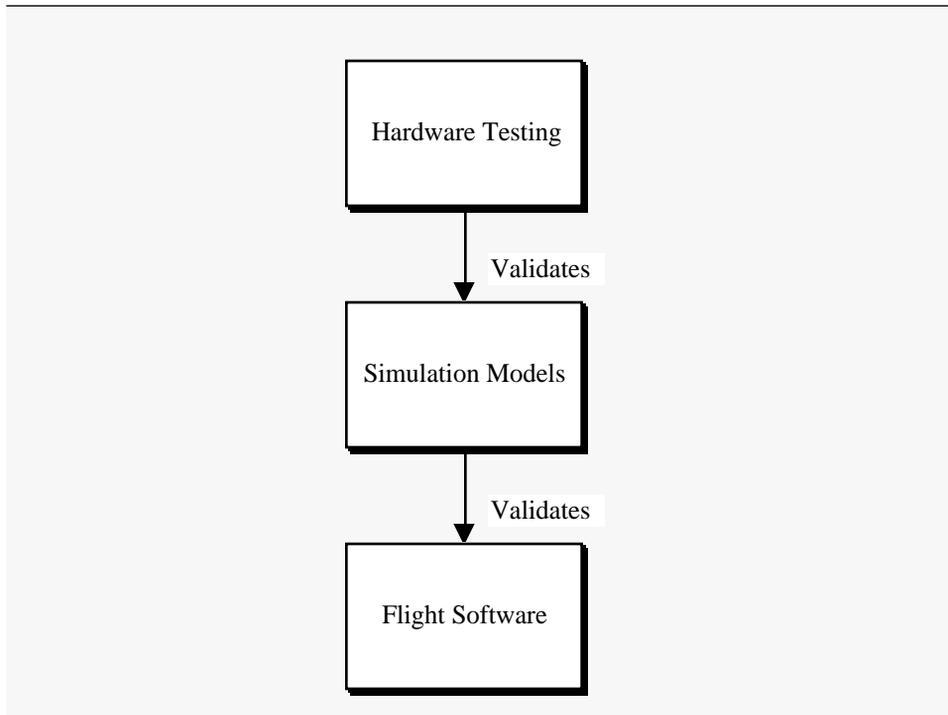
A similar process was used for the DC-X single stage to orbit test vehicle. McDonnell Douglas was able to develop the flight control software in ten months in a program with an overall 18 month schedule. McDonnell Douglas was very pleased with the results and plans to use this methodology on future programs. The DC-X used an Intel 80960 RISC processor in the flight computer.

In both cases, however, the use of the block-diagram design tool was limited to the GN&C portions of the code. The remainder had to be coded by hand. In addition, many of the blocks used in the block-diagram tool were also hand-coded. Using block diagrams is actually just using a different higher-order language. Research has shown that every language has bugs that are specific to the language and that while using a different language may reduce certain types of problems, it will introduce new problems specific to that language.

### 20.2.10 Summary

This chapter describes test approaches used on a number of spacecraft ranging from geosynchronous communications satellites to precision astronomy satellites. While the exact testing methodology was different in each case, fundamentally successful testing followed the flow in Figure 20-5. Validating flight software prior to operation requires numerical simulations of the spacecraft and its components. The numerical simulations need valid models of the

**Figure 20-5 Test Flow**



spacecraft and its hardware. These can only be obtained through testing of the hardware and careful analysis of the mechanisms, optics, etc. that comprise the components.

The second major conclusion is that closed loop testing with the actual flight computer and interfaces is required. This should be performed both at the bench test level and in thermal vacuum conditions. At the very least, the spacecraft dynamics and kinematics models must be realized in a digital simulation. However, even if the ultimate intent is to use hardware actuators and sensors in the testbed, it is still a good idea to develop numerical models for early test-

ing prior to delivery of the components, and as a substitute as needed in later testing. The MOS-1 test flow is a good example of this process.

The need for automatic configuration management of software and of data is emphasized in all of these examples. Automatic version control and use of centralized databases, also under configuration management is deemed essential for a flight software test program.

---

## 20.3 A Testing Methodology

### 20.3.1 Introduction

This section provides the background for the software testing of an Attitude Control System (ACS) and recommends a test program necessary to ensure that the probability of failure of the ADACS software due to hidden bugs is sufficiently low. This section also covers the testing of the flight software test-bed and supporting tools.

### 20.3.2 Requirements Flow and Testing

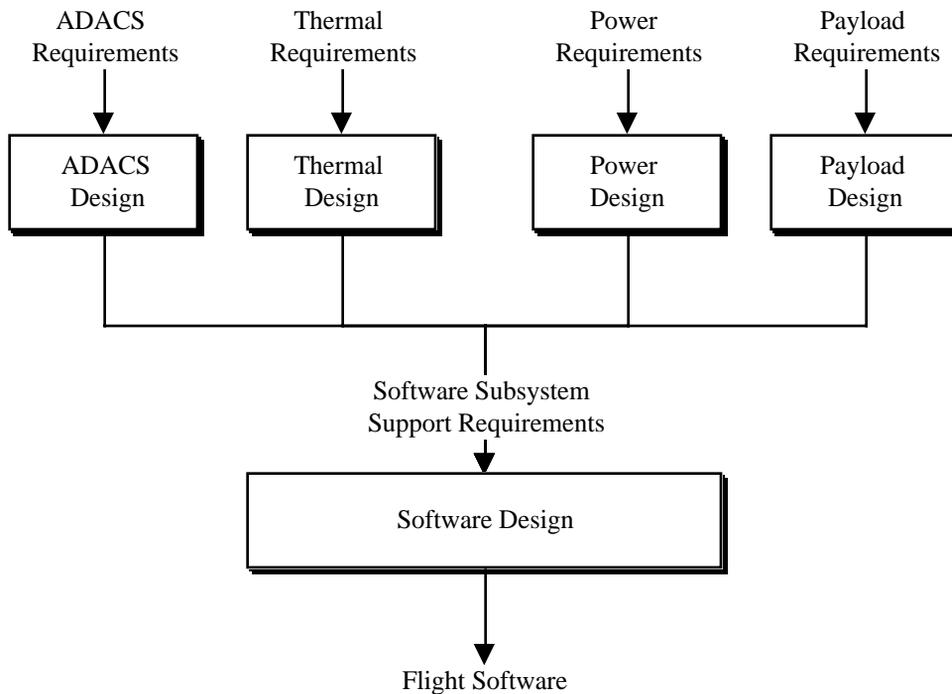
Figure 20-6 shows the requirements flow for flight software. The figure shows only the requirements levied by other subsystems that the flight software must support, such as the Attitude Determination and Control Subsystem (ADACS), power subsystem and thermal subsystem. Requirements levied directly on the flight software, are not shown.

Each subsystem has performance requirements that can only be met through operation of the flight software. This includes closed-loop attitude control, power management, etc. During the design cycle, each subsystem performs analyses and simulations to demonstrate that the subsystem design meets its requirements. The subsystem then generates software requirements and these are realized in the flight software. Flight software testing verifies that it meets the requirements levied on it by the subsystems, but not necessarily that the subsystem design, as implemented in flight software, meets the original subsystem requirements. In practice, this usually happens indirectly since flight software is often verified by comparing simulation results with the flight software to simulation results generated by subsystem simulations that have been shown to meet the performance requirement. The validity of this indirect approach is based on the assumption that the test platform for the flight software has sufficient fidelity to make the comparisons (in terms of performance) meaningful. In addition, the subset of tests run on the flight software test platform are assumed to be representative of the subsystem tests originally used to demonstrate that the subsystem met its requirements. These assumptions may not be valid if the flight software test platform simulation is of insufficient fidelity.

A careful distinction must be made among the types of requirements. Each subsystem has requirements it must fulfill and many fulfill their requirements through flight software. They do this by levying requirements on the flight software. In theory, if the flight software meets these requirements levied upon it by the subsystems, the subsystems will also meet their requirements. Unfortunately, this may not be the case and it is usually necessary to exercise the actual flight software and a subset of the subsystem hardware to demonstrate that the subsystem requirements have been

met. Furthermore, some iteration between the flight software design phase and the subsystem design phase is almost always necessary since subsystem requirements may not be implementable as first specified.

**Figure 20-6 Typical Requirements Flow**



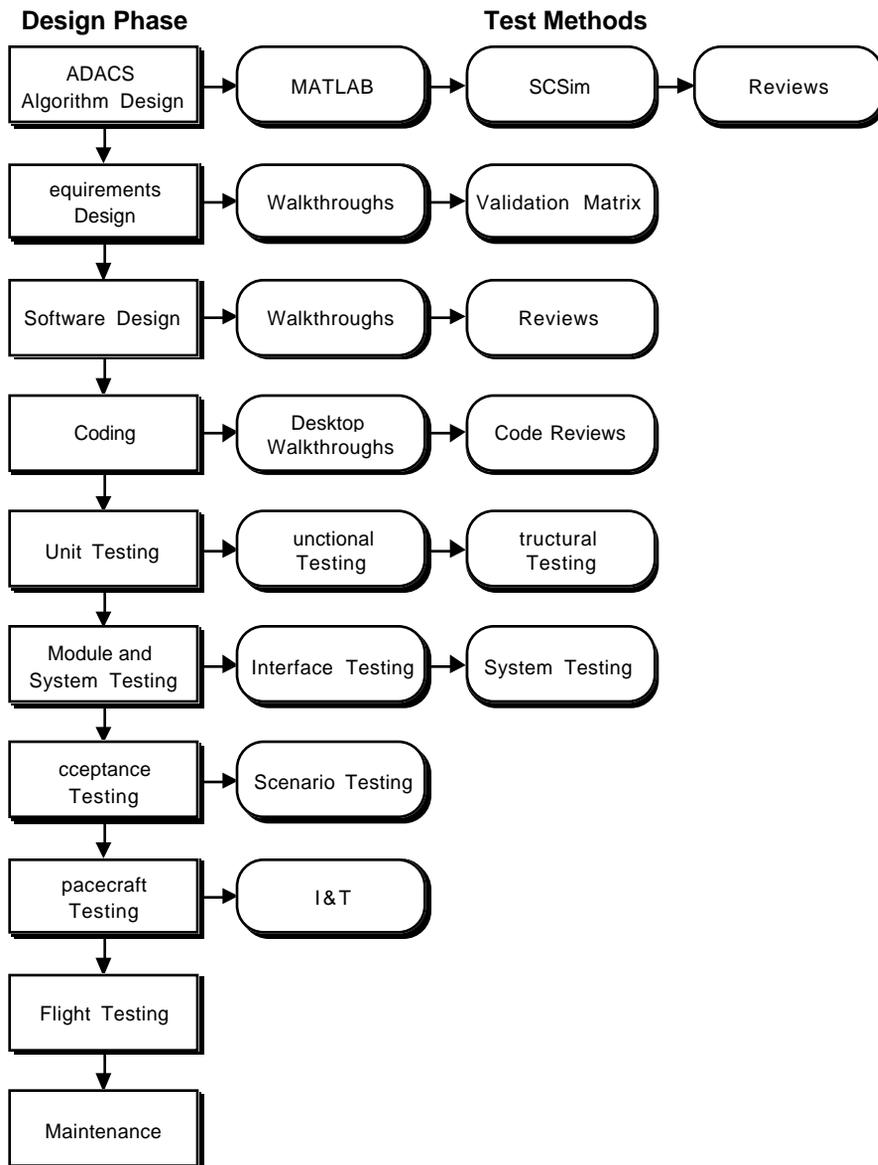
### 20.3.3 Testing Life-Cycle for the ACS Flight Software

An example of a complete testing life-cycle for a flight software product is given in Figure 20-7. The rectangular symbols represent the design phases and the boxes with the rounded corner represent test methods or phases that are used to validate the results of a design phase before proceeding to the next phase. The first two phases in the design life-cycle involve the design of the ADACS algorithms, as opposed to the software that implements them in the flight computer. This phase is driven by the ADACS requirements. In the first phase, the ADACS algorithms are designed and tested in a control analysis package, usually MATLAB. The design is then validated using a high fidelity spacecraft simulation such as the Spacecraft Simulator (SCSim.) In this case, the sensor, actuator, environment and dynamics model are high fidelity. Flight computer arithmetic and interfaces are not modeled. The output of this phase is the ADACS specification and requirements documents along with supporting analysis memos.

The flight software ADACS support requirements are derived from the results of the ADACS analysis. The remaining steps follow a standard software design cycle. The test tools and platforms shown in the figure will be discussed in more detail later in this chapter.

The purpose of this report is to evaluate the testing of the ADACS subsystem during all phases of development, and to present an independent analysis of the testing in these phases and recommend additional tests to insure full coverage of all modules, modes of operations and ADACS functions. All phases, from MATLAB testing through flight software hardware testing will be discussed.

Figure 20-7 Testing Life Cycle



### 20.3.4 Flight Vehicle Control System Testing

Flight vehicle control system testing goes through two stages. These are defined in Table 20.8, on page 266. The

**TABLE 20.8 Flight Vehicle Software Testing**

Stage	Description
Preflight	All functions are tested either through closed-loop simulation or with open-loop test data (usually generated by another closed-loop simulation.) Simulations may include some, none or all of the ADACS hardware. ADACS algorithms may be implemented in an abstract form (such as MATLAB), in another language (such as FORTRAN) in the target language, C, but on a different processor, and in the target language on the flight processor.
Flight	Transfer orbit and acquisition ACS functions are tested as each is introduced in the mission sequence. During a normal transfer orbit it is possible to fire the apogee motor on any one of several orbits. In addition, since the spacecraft is a major axis spinner, its attitude is stable and the satellite can be left alone for extended periods of time, the only constraint being power. As a consequence, there is usually adequate time to solve any problems that might arise. Problems are usually resolved through procedural changes and work arounds. If necessary, the flight software can be patched. Mission orbit operations, such as stationkeeping, are tested by performing the shortest possible version of the operation. Problems are usually resolved through procedural changes and work around. If necessary the flight software can be patched.

actual flight vehicle is rarely available for test purposes until late in a program. Because of scheduling problems, it is rarely possible to perform extensive closed loop testing on the spacecraft when it is in integration and test. Testing on the complete spacecraft is limited by the lack of internal visibility into the software. The only access to the flight software is usually through the command and telemetry interface. It is difficult to gain access to internal variables in the flight software making debugging difficult. Sensor inputs must be generated either by external targets, that provide little flexibility, or by inserting inputs into the lines running from the sensors to the flight computer. Measurements of the outputs is limited by whatever measurement channels are available on the actuators. For example, if the only speed measurement on a reaction wheel is the tachometer, it will not be possible to know the true reaction wheel state.

Simulating the complete space environment, including thermal, vacuum and dynamics effects, is impractical with a complete spacecraft. In addition, it is difficult to simulate off nominal or worst-case conditions involving the actuators and sensors.

The most cost-effective solution is to run a simulation of the spacecraft dynamics and environment on another computer and use taps on the sensors and actuators to close the loop. This provides the most complete picture of how the flight software will perform, but must be used in conjunction with a high-visibility tool, such as a development station or processor emulator, to help trace problems that arise.

Flight software designers must rely on simulations of the vehicle and the operational environment to validate the flight software. Simulations are of five types as given in Table 20.9, on page 267.

**TABLE 20.9 Simulation Types**

Simulation	Description and Use	Indostar Facility
All software	<p>The vehicle, flight computer and interfaces are modeled in software. This makes it a very flexible and portable tool. The fidelity of the simulation is limited by many factors including:</p> <ul style="list-style-type: none"> <li>• Available component data</li> <li>• Host computer throughput</li> <li>• Time available to design, build and test numerical models</li> </ul> <p>It is assumed that the fidelity of the simulation is sufficient to verify the performance of the algorithms in the flight software. Since it can usually run many times faster than real-time, it is possible to test the long-duration behavior of the algorithms.</p>	SCSim
All software with flight computer emulation.	<p>The flight computer is emulated. This provides more accurate timing information at the expense of execution speed. It also gives excellent visibility into the workings of the software. Emulation can uncover compiler bugs or instruction set problems.</p>	
All software with a flight computer in the loop.	<p>This permits real-time operation. If dual-port memory is available the inner workings of the flight software can be monitored. However, if flight prototype components are used, the visibility into the flight software is usually limited by the telemetry interface.</p>	
Software dynamics, all ADACS hardware in the loop but fixed base.	<p>All interfaces are faithfully represented. The closed loop simulation generates the stimuli for the sensors so sensor dynamics are not tested. This is sometimes done with a hardware breadboard ACS system. It can also be done on the actual spacecraft.</p>	I&T (No independent high-fidelity closed loop simulation)
Software dynamics, all ACS hardware in the loop with a moving base.	<p>All of the sensors are stimulated by sources representative of actual operation. For spacecraft with gyros this requires a multi-axis rate table. Earth and sun targets are needed for the earth and sun sensors. Orientation of the rate table and sun/earth targets must be synchronized. This provides the highest possible fidelity, short of using the flight vehicle in operation. The cost is usually prohibitive for all but the most expensive satellites.</p>	None

Each of the simulations may include a complete closed-loop dynamics model of the spacecraft or may be used open loop with test data from another source—usually from another simulation. Industry experience with different ADACS simulation testbeds is given in the next section.

### 20.3.1 Test Levels (Preflight)

This report uses the three levels of testing for the flight software as defined in Table 20.10, on page 268. Unit testing is

**TABLE 20.10 Test Levels**

Level	Scope	Tests
Unit	Individual procedures	Range of valid inputs. All paths and all code.
Module	Individual functions	All mission operations relevant to that module.
System	Entire ADACS	All mission sequences.

usually done by the designer of the procedure. It is meant to be exhaustive and guarantees that the procedure will work correctly for all valid inputs. Since it is rarely possible to check every valid input, the inputs must at least encompass the full range of valid inputs. Boundary values must always be included in these tests. A combination of boundary values, random inputs (within the valid ranges) and “best guesses” of inputs that may cause problems, should be employed. Random inputs should be normally distributed about the mean input values. In addition, all logic paths must be exercised. The former type of testing is called black box testing, the latter in which test inputs are chosen to follow all logic paths, is known as white box testing. Whenever possible, the procedure is tested against analytical results or, if that is not possible, against already verified, higher-fidelity models. Unit testing must be exhaustive because it is not feasible to fully test a procedure at the higher testing levels since it is difficult to guarantee, or prove, that a given routine has been fully exercised. Unit testing is usually open-loop, although closed-loop test drivers may be employed.

Module testing tests functions of the flight software. For example, the backup ephemeris is a function and is implemented by means of several different procedures and/or packages in the flight software. Module testing demonstrates that independent functions of the software work properly and that all of the procedures that comprise a module interact correctly. Module testing is usually open loop, although closed-loop test drivers may be used.

Module testing is the first step in verifying the interfaces between procedures. This is also accomplished during system testing. For some modules, such as thruster commanding, module testing would include testing of the hardware interfaces.

Most of the lower-level flight software requirements will be validated using either unit or module level testing, if only because the requirements are partitioned by function, and sometimes by module. Module testing should demonstrate that the module will meet its requirements over the full range of operating conditions. While unit testing should include all valid inputs, module testing need only include valid inputs within the operational range.

The highest level of testing, system testing, demonstrates that all of the modules work together correctly and that the flight software can perform all required mission operations. System testing should include realistic command inputs through the appropriate hardware.

All tests should employ a coverage monitor tool. A coverage monitor tool keeps track of what code is exercised in each test. For a given module, it is essential that all code be exercised in its module test. At the system level, all code should be exercised in at least one of the system level tests. Studies have shown that testing done without coverage tools typically exercises only 55 percent of the code.

### **20.3.1 Test Levels (Flight)**

The first flight tests of the ADACS software will happen during actual mission operations in transfer orbit. This does not present a major problem since the spacecraft does not rely on the flight software for closed loop attitude or orbit control during this phase. Each function can be tested when it is first used. Whenever possible, the duration of use of the function should be limited. Spin precession maneuvers are sufficiently slow that the first maneuver to the Apogee Kick Motor (AKM) firing attitude can be done in its entirety and aborted should the maneuver path (as indicated by the sun angle measurements) diverge. A similar approach can be used with all other functions.



---

## *21.1 Introduction*

This chapter discusses a usually overlooked aspect of ACS design, failures. All spacecraft control systems must be designed to be fault tolerant. This chapter first discusses failures that have occurred on spacecraft. It then discusses failure analysis. Finally, it talks about methods for building fault-tolerant software. In this chapter, we deal primarily with software designed to catch failures in hardware, not in the software itself.

This chapter has two parts. The first gives industry experience in testing and different approaches to testing. The second section discusses a test approach that has been used successfully at Princeton Satellite Systems for a number of years.

---

## *21.2 Failures*

Failures can be grouped into two categories

- Those that required ground intervention
- Those that didn't

Each category of failure can be further divided into two groups

- Those that caused a degradation in mission performance
- Those that didn't

The following table lists some real spacecraft failures. In addition to the above categories, it lists other relevant data.

**TABLE 21.1 Spacecraft Failures**

<b>Failure</b>	<b>Ground Intervention?</b>	<b>Mission Degradation?</b>	<b>Operator Error?</b>	<b>Loss of Mission?</b>	<b>Soft or Hard?</b>
Single access sun sensor assembly bit failure. At a particular sun angle the output would jump 2 degrees.	Yes	No	No	No	Soft
Momentum wheel electronics failure caused the total loss of momentum wheel functioning. It was necessary to fly using thrusters.	Yes	Yes	No	No	Hard
A periodic disturbance during stationkeeping was traced to helium pressurant being passed the thrusters. When this happened, no thrust was produced for 2 seconds. The attitude control system compensated for this problem.	No	No	No	No	Soft
The high gain antenna on a spacecraft stuck partially open. Advances in signal processing technology mitigated the problem.	Yes	Yes	No	No	Hard
Operators dumped the fuel from a spacecraft twice.		Yes	Yes	Yes	Hard
A spinning spacecraft's attitude control system went into a hard limit cycle during an apogee motor burn. This was due to an unmodeled slosh disturbance.	Yes	No	No	No	Soft
A momentum bias spacecraft experienced a growing oscillation at nutation period due to a incorrectly set parameter.	Yes	No	Yes	No	Soft
A spacecraft began to oscillate at the control crossover frequency during a stationkeeping burn due to an unmodeled sensor delay.	Yes	No	No	No	Soft
A geosynchronous spacecraft flipped over during a solid rocket firing in transfer orbit. This was due to a large offset in the center-of-mass. The remainder of its delta-V was achieved by turning it on its side and using its electrothermal hydrazine thrusters in a series of 180 maneuvers. This turned out to be the first geosynchronous transfer done with electric thrusters.	Yes	Yes	No	No	Hard
During a stationkeeping burn a spacecraft lost earth lock. This was due to a parameter that was not set properly in a pitch control system.	Yes	No	Yes	No	Soft

**TABLE 21.1   Spacecraft Failures**

Failure	Ground Intervention?	Mission Degradation?	Operator Error?	Loss of Mission?	Soft or Hard?
A momentum bias spacecraft had its nutation increased because ground operators misinterpreted the roll error from the earth sensor.	Yes	No	Yes	No	Soft
Fine sun sensor spikes caused attitude degradation. New software was uploaded.	Yes	No	No	No	Soft
Thrusters failed on causing high rotation rates and a switch into sun safe mode. Problem was traced to a read error between the star tracker and attitude control electronics.	Yes	Yes	No	No	Hard

This list is not comprehensive, but it does show a surprisingly large number of errors due to operator error.

---

### *21.3 Failure Analysis*

Failure analysis is quite simple. There are two types of failures, hard and soft. Hard failures mean that the component stops working completely. Soft failures are instances where a component still works, but in a degraded condition. An example of the former is a thruster being stuck off. An example of the latter is when the bearing friction increases on a reaction wheel.

First it is necessary to look at all spacecraft components (including software) and determine each component's failure mode(s). The second step is to look at the probability of that failure mode happening. A failure mode may be independent of time or the probability of failure may increase or decrease with time. For example, a reaction wheel bearing failure becomes more likely as time progresses. On the other hand, electronics failures are greatest at the beginning of operation and drop rapidly after the initial "burn-in" period.

Once the probabilities are established, it is necessary to determine the probability of the spacecraft meeting its mission requirements. To do this, it is necessary to tie the mission requirements to the possible failures. For example, if a spacecraft has only three reaction wheels and two magnetic torquers, if one of the wheels fails it may not be able to meet three axis pointing requirements. Since the spacecraft still has two wheels and two torquers, it may be possible to point in three axes but the pointing accuracy is likely to be degraded. If the designer wants to use this "mode" of operation, then he or she must do a performance evaluation of this mode.

---

## 21.4 *Fault Detection Isolation and Reconfiguration*

### 21.4.1 Introduction

Substantial gains in system performance have been realized through continued implementation of automatic control systems in a wide variety of vehicles, devices, and commercial products. This enhanced functionality has not come without its own cost, however. As more system functions are relegated to automatic control, the increased complexity of highly automated systems threatens a corresponding decrease in utility due to the increased likelihood of component or sub-system failures that may render the device unusable. This problem becomes particularly acute in aerospace systems, as the compromises inherent in the design of airborne and space systems (ultimately driven by cost constraints) make these systems particularly susceptible to catastrophic failures due to faults in sub-systems or critical components.

One means of addressing this problem, beyond the use of basic design approaches that attempt to minimize the impact of component failures on overall performance, is through the use of a fault detection and isolation (FDI) system. FDI systems are designed to identify when a failure occurs in a dynamic system and to indicate what component or subsystem has experienced the failure. The typical trade-off facing the FDI system designer is to properly balance the FDI logic to respond rapidly to failures while minimizing the number of “false alarms” that may be generated by noise present within or external to the system being monitored. In general, the use of FDI systems is a means to extend the useful life of a system without the (possibly great) increases in cost associated with use of additional or more reliable sensors, actuators, or system components. Designers of FDI systems have accepted that the component lifetime for a system is finite, and wish to incorporate knowledge of how various sub-systems may malfunction into a total scheme for performance enhancement (or extension).

Interest in FDI as a means of extracting more utility from vehicles and systems has steadily increased with attendant increases in the use of feedback control on these same systems. Willsky (1976) and Isermann (1984) provide excellent surveys on the state of design approaches for FDI. Methods range from hardware redundancy and inter-component voting to analytical redundancy, where use is made of the interrelationships between the various sensors and the dynamics of the “plant”, to “intelligent” FDI that incorporates learning through the use of neural networks and expert systems. Of paramount importance in the design of practical FDI systems, however, is that proper attention has been paid to the trade-off of complexity of the FDI system vs. the increased performance it provides to the overall system.

The proposed scheme for FDI described in this proposal extends a promising (but until now, somewhat neglected) method for fault detection and isolation for linear, time-invariant systems to a nonlinear, time-varying spacecraft system. The method consists of an innovative combination of nonlinear filtering techniques, eigenstructure assignment methodology, analytic linearization, and robust control concepts using frequency-shaped cost functionals. The proposed approach represents to this author the first known application of these techniques to spacecraft FDI systems, and promises to provide increased autonomy for both individual satellites and satellite constellations. Appreciation of the proposed concept for FDI is best gained after a brief survey of the current schemes used for satellite fault detection.

## Satellite Fault Detection and Isolation

Satellite fault detection and isolation systems are designed to identify problems with spacecraft components and to take appropriate action to minimize the influence of the failure on the mission. If the component that is causing the off-nominal condition can be identified, the first course of action is to switch in a new unit. If a new unit is unavailable the function of that unit may be “synthesized” using other components. For example, if a reaction wheel fails on a spacecraft, thrusters might be employed to replace its function. If no replacement is possible, or the system cannot identify the error, the redundancy management system will put the spacecraft into a safe mode until the ground operators can fix the problem.

One of the challenges to designing a fault detection system is to get it to correctly identify problems. A greater challenge is to make it insensitive to modeling errors, noise, etc. so that the system does not falsely trigger. Unfortunately, the latter is rarely accomplished and satellite operators generally turn off the fault detection system, unless they are certain that a problem is likely to occur.

Failures are broadly grouped into two categories, hard and soft failures. A hard failure is when a device either fails to operate, or begins to behave in an extremely off-nominal way. A soft failure is when the device still operates, but its performance no longer meets its specification. While it may be acceptable to allow a device that has experienced a soft failure to continue operating, it is never acceptable to use a device with a hard failure.

All fault detection systems are based on a model of the system behavior. These may be implicit or explicit. In the former case, the designer uses his or her knowledge of the system to describe all possible failure modes of all critical components. The next step is to determine the failure signature of each failure. The failure signature is the expected observable result of the failure. To be observable, it must be detectable by one or more of the spacecraft’s sensors.

It may turn out that some failures are difficult, if not impossible to detect with the existing sensor suite. It may be necessary to add sensors to improve failure detectability. Once the failure signatures of all known failures have been determined, the designers must decide which failures will be subject to automatic monitoring. This subset forms the core of the fault detection system. At this point, the designer can choose to use implicit or explicit models. In addition, each type of model can be learning or static. A learning model will modify itself based on the observed behavior of the system. A static model is based only on a-priori information about the system and must be changed manually.

Implicit models are usually found in boolean and fuzzy logic fault detection and isolation systems. The designer “knows” that if the earth sensor assembly (ESA) output (for example) is constant for more than 10 counts, it has probably failed off. The on-board logic therefore consists of a counter and a maximum number of counts. The model is implicit in the maximum number of counts.

Explicit models are numerical models of the system behavior. These are typically linear and model the input-output behavior of the system.

Learning systems can adjust their parameters. For the implicit model described above, a learning algorithm might note that the ESA counter often reaches 8. It might then adjust the counts required to declare a failure to 15 since 10 leaves too little margin. In the case of the explicit model, the system might perform system identification to adjust the model parameters to better fit the observed behavior. Learning systems can account for soft failures through these kinds of adjustments.

The degree of memory inherent in a fault detection system is an important characteristic. Slow changes in a device's performance may be indication of degradation due to the on-orbit environment. However, if the performance changes over a very short period of time, it is more indicative of a failure. A learning system can accommodate these kinds of changes, a static system cannot.

Given the model, and its outputs, the designer must then determine what constitutes a failure. It is not always possible to isolate all failure signatures. Some systems, such as the fault detection system of Galileo, will look for the set of failures that are indicated by the signature and group them as most likely and most dangerous. The logic then decides which it is and responds accordingly.

## 21.4.2 Approaches to Redundancy Management

### Discrete Logic

Discrete logic is used frequently in redundancy management systems. The thresholds indicating a failure are precisely defined and are based on an implicit failure model. Some failures, such as constant output failures of sensors, are easy to detect with implicit models. Failures that involve evaluating the dynamical response of the system to determine if a failure has happened are very difficult to mechanize with boolean logic.

A major problem with the boolean logic approach is that deeply nested logic structures can be very difficult to debug. One approach is to replace much of the logic with tables. This reduces the amount of code that must be tested, but it is still necessary to exercise all of the data in the tables.

Fuzzy logic [Terano et al. (1992)] is an alternative approach to FDI. Fuzzy logic works within the framework of set theory and is better at dealing with ambiguities. For example, three sets might be defined for a sensor: hard failure, soft failure and no failure. The three sets might overlap and at any given time the sensor may have a degree of membership in each set. The degree of membership in each set can be used to determine what action to take.

A major problem with discrete logic systems is the cost of testing. During software testing, all logic paths must be exercised. While white box testing is feasible, system level tests that exercise all of the paths is rarely practical. As a consequence, logic-based systems usually address only the easy-to-detect failure cases.

### Reference Models

Explicit models are used in many fault detection schemes. A general reference model is of the form

$$\begin{aligned}\dot{x} &= f(x, u, t) \\ y &= g(x, u, t)\end{aligned}\tag{21.1}$$

where  $x$  is the state vector of the system,  $u$  is the vector of known inputs and  $y$  is the vector of measured outputs. The residuals are the difference between the measured output and the expected output

$$r = y_m - y\tag{21.2}$$

Equation (21.1) rarely includes all of the dynamics that are visible in  $y_m$ , thus it is usually a reduced order model. An important consideration when modeling the plant is whether the closed or open loop dynamics are to be included in the model. If the open loop dynamics are used then Equation (21.1) becomes

$$\begin{aligned}\dot{x} &= f(x, u_c + u_d, t) \\ y &= g(x, u_c + u_d, t)\end{aligned}\tag{21.3}$$

where  $u_d$  are known disturbances and  $u_c$  are the control inputs. If the closed loop model is used Equation (21.1) becomes

$$\begin{aligned}\dot{x} &= f(x, u_d, t) \\ y &= g(x, u_d, t)\end{aligned}\tag{21.4}$$

One of the purposes of feedback is to combat uncertainty, consequently a closed-loop model will tend to be easier to track than an open-loop model. However, when a closed-loop model is used, failures of control actuators and sensors are seen as plant perturbations and not as input-output errors.

Generally, linear reference models of the form

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx\end{aligned}\tag{21.5}$$

are used. The estimator is therefore

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x})\tag{21.6}$$

The residuals are

$$r = y - C\hat{x}\tag{21.7}$$

and are used for failure detection. The estimator may already be a part of the control system. Threshold or statistical tests are applied to the residuals to determine if there is a failure. The former may employ boolean logic, fuzzy logic or an expert system.

Explicit models have the advantage that they allow incorporation of a priori information about both the system and certain likely failure modes. Direct inclusion of bias states and scale factor changes are possible in this formulation, but for complex systems, such direct inclusion of “failure states” leads to an overwhelming computational burden for the system estimation task. In addition, failures may manifest themselves in the measurement residual in ways that are not easily (or reliably) detected using standard statistical metrics. However, a modification to this approach, described in a later section, allows the designer to circumvent these limitations and produce a very capable FDI system.

## Voting

The voting approach, which works with sensors and flight processors, is to install multiple copies of each and compare their outputs. At least three of each sensor or processor is required. The outputs from two that agree the closest are used. If the third consistently gives different results it is declared to be failed. This approach is completely impractical for all but man-rated spacecraft, due to the associated cost and weigh penalties of the redundant hardware requirements.

## Expert Systems

Expert systems are a variation on the boolean logic approach. Generally, expert systems are database driven and their speed of operation depends on the speed of memory (RAM) access. The major advantage of expert systems is that they minimize the amount of coding required thus reducing software development costs. However, their major drawback is the teaching required of the expert system, and the possibly prohibitive amount of simulation required to validate the inference engine and database.

### 21.4.3 The Fault Detection and Isolation System

#### Introduction

The proposed FDI system described here represents a combination of analytical redundancy using a detection filter, nonlinear filtering techniques, analytical linearization, and frequency-weighted cost functions to design a fault detection and isolation system that: (1) is sensitive to failures, (2) provides information as to the type and magnitude of failure, (3) accommodates slowly time-varying and nonlinear functions within the system dynamics, and (4) is easy to design and implement. The FDI system is an extension of that first proposed by Beard (1971) and Jones (1973), and later extended by Meserole (1981) to nonlinear systems. Interestingly, Patton (1992) recently rediscovered this approach. The main feature of the method consists of constructing a state estimator for the dynamic system that selects a feedback gain on the measurement residual to make the overall estimator sensitive to pre-specified types of system failures. Through careful selection of the gain matrix, one may force the measurement residual to remain fixed in a particular direction or a specific plane for a given failure mode. A simple example for a linear time-invariant system from Beard (1971) serves to illustrate the approach. For a system described by

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (21.8)$$

with measurements related to the state variable according to

$$z(t) = Cx(t) \quad (21.9)$$

a detection filter is designed of the form

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Dr(t) + Bu(t) \quad (21.10)$$

with  $r(t) = [z(t) - C\hat{x}(t)]$  the measurement residual, and with  $D$  chosen to accentuate any failures that may occur in the system. For example, if a bias of magnitude  $v$  suddenly appears on the  $i$ -th actuator, the system becomes

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B(u(t) + ve_i) & \text{or} \\ \dot{x}(t) &= Ax(t) + Bu(t) + vb_i \end{aligned} \quad (21.11)$$

where  $b_i$  represents the  $i$ -th column of matrix  $B$ . For the case of full state measurement,  $C = I$  (the identity matrix), and thus the equation governing the measurement residual becomes

$$\dot{r}(t) = [A - D]r(t) + vb_i \quad (21.12)$$

so if  $D$  is chosen  $D = [\sigma I + A]$ , the residual dynamics become

$$\begin{aligned} \dot{r}(t) &= -\sigma r(t) + vb_i & \text{or} \\ r(t) &= \exp\{-\sigma(t - t_o)\}r(t_o) + v[1 - \exp\{-\sigma t\}](b_i/\sigma) \end{aligned} \quad (21.13)$$

Thus, after the initial transient dies out, the steady residual maintains a fixed direction aligned with vector  $b_i$ . Note that its magnitude decreases as the bandwidth of the detection filter increases, thus trading off speed of response with sensitivity to noise in the system.

The above technique has been extended to handle a wide variety of failure types that may be present in the actuation systems, system dynamics, or sensors, for systems that do not possess full state feedback. For systems that have observation matrices  $C$  of rank less than the number of system states, some limitations may exist on the capability to isolate all potential system failures. This limitation is somewhat akin to the observability of the system, and typically poses no problem, as the sensors have been installed to provide sufficient information for achieving good automatic control by design. The primary limitation of the above technique, then, is that the selection of feedback matrix  $D$  is deterministic, and that the system representation is both linear and time-invariant.

Meserole (1981) applied the above technique to detect faults in the sensors and control system for an F100 aircraft gas turbine engine, and included nonlinear effects in the engine performance map through re-linearization of the system about a new operating point. He found that the success of the detection filter did not strongly depend upon the accuracy of the linearized representation of the rather nonlinear character of the engine response. The method used to select feedback gains for the residuals included a combination of transformation techniques and least-squares methods for eigenvalue/eigenvector assignment. Eigenstructure control over the dynamics representing the measurement residuals assures that each failure mode exhibits a unique and identifiable deviation in the measurement residual whose magnitude may change with time but whose shape (direction) does not.

Eigenstructure assignment techniques are well established in the context of feedback control design, and may be performed directly using least-squares methods (Andry, et. al, 1983) or through asymptotic properties in specification of the cost function to be optimized (Stein, 1979). The least-squares methods (sometimes called the “direct approach”), do not result in guaranteed stability margins for the resulting feedback controllers. Along with providing a feedback structure with stabilizing gain values, the asymptotic approach lends itself to other methods of gain adjustment in order to enhance robustness of the resulting feedback system. In particular, use of frequency-weighted cost functions (Gupta, 1980) can significantly improve the robustness of both control systems and recursive filters to unmodeled (or poorly modeled) high-frequency dynamic components. This becomes particularly attractive in spacecraft configurations that are limited in their allowable sample rates for the sensors used for attitude control. For this reason, the pro-

posed fault detection methodology will investigate application of frequency shaping in the determination of filter feedback gains for enhanced fault detection using directed residuals.

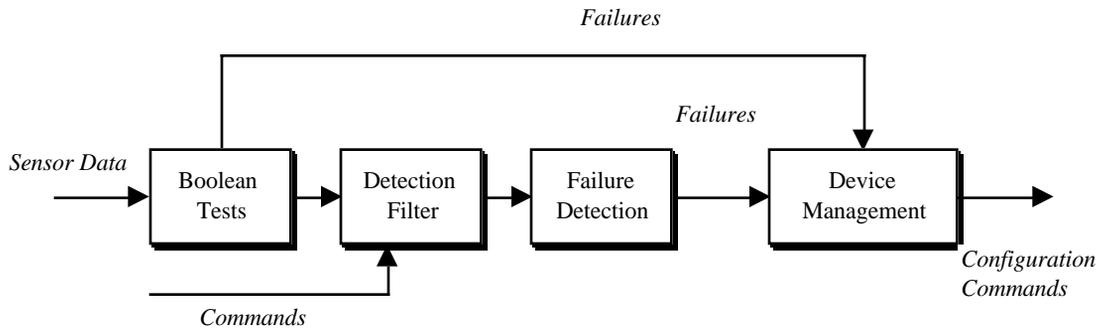
For systems that are time-invariant, the filter gains would form constants in the “D” matrix above, but if the technique is to be applied to a nonlinear, time-varying system, then the filter structure would need to track these variations. Use of an Extended Kalman Filter formulation (Gelb, 1969) allows these variations to be represented in the filter, by using the full nonlinear and time-varying form of the system dynamics in the state propagation and measurement equations, but a linearized representation of the dynamics and observation matrix in the state estimation error covariance equations. The linearized system and measurement matrix are formed by using the current state estimate (and time) to compute the matrix elements of “A” and “C” above. This may be done through numerical linearization of the system, where small perturbations are introduced to the system model and the resulting outputs used to compute the parametric sensitivity of each matrix component, but such techniques rapidly become impractical for any system of moderate dimension. Instead, it is proposed that the system to be studied be analytically linearized about a completely general operating point, in order to provide an efficient computational resource for rapidly generating these linearized system matrices as the state estimate changes. Such an approach has been used in the past for system identification applications to rotorcraft flight dynamics problems with considerable success (McKillip, et. al., 1988). The fault detection methodology proposed here will also implement a similar analytical framework for determination of a locally linearized system for use in the nonlinear filtering scheme.

The proposed methodology, in summary, represents a means of designing a failure detection and identification system that is a novel combination of: (1) failure detection and identification through analytical redundancy, incorporating filter feedback structures that constrain the measurement residuals so as to uniquely identify particular failure modes; (2) eigenstructure assignment for computation of the filter feedback gains incorporating frequency-weighted cost functionals, in order to maintain the desired residual directivity under both system parameter variations and nonlinear dynamics effects in the plant, actuators, or sensors, while also reducing sensitivity to unmodeled or high-frequency plant dynamics; and (3) use of an analytically linearized representation of the spacecraft dynamics, for rapid incorporation of the localized linear approximation of the system in the covariance (and hence gain computation) equations. A discussion of the form of the FDI system, and a simplified example, is given in the following sections.

## **System Architecture**

The overall system architecture for the detection filter system is illustrated in

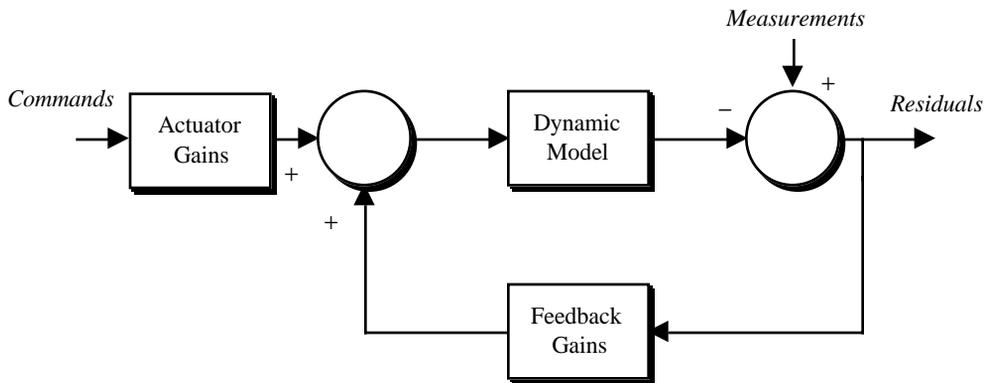
FIGURE 21.1 The FDI architecture



The first block has simple boolean or fuzzy logic tests to detect failed-off or failed-constant sensors.

The detection filter block is shown in Figure 21.2 on page 281.

FIGURE 21.2 The Detection Filter



The block diagram is similar to that of any Kalman Filter. The failure detection logic evaluates the residuals from the filter and decides if a failure has occurred. The device management logic then reconfigures the system.

### 21.4.4 An Example Using Detection Filters

#### Introduction

This section gives an example of the use of detection filters to determine control system failures in a spinning spacecraft. The spacecraft is symmetric and spins about its major axis. Only the transverse axes are included in this example. A damping controller using thrusters is in operation. The controller has two independent loops that add damping

to each transverse axis. Each controller is a simple proportional control system and it is assumed that rate measurements are available. The purpose of the failure detection system is to insure that the control loops are working properly, and if one has failed, determine where the failure lies.

### Dynamics Model

The closed loop state equations for the transverse axes of a spinning spacecraft are

$$\begin{bmatrix} \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} -c & k \\ -k & -c \end{bmatrix} \begin{bmatrix} \omega_y \\ \omega_z \end{bmatrix} + \begin{bmatrix} u_y \\ u_z \end{bmatrix} \tag{21.14}$$

$$k = \frac{\omega_o(I_x - I_{y,z})}{I_{y,z}}$$

where  $u$  represents external disturbances. The closed loop control is embodied in the constant  $c$ .

### Failure Model

Table 21.2, on page 282 shows a simplified failure model. Either a thruster or a gyro can fail and the failures can man-

**TABLE 21.2 Spinner simplified failure model**

Gyro	Thruster	Result
Fail off	Fail off	Control loop is broken
Soft failure	Soft failure	Closed loop plant dynamics change
Fail bias	Fail on	Control loop is broken and the spacecraft experiences a steady transverse torque

ifest themselves in different ways. The table groups the failures according to their affect on the system. For example, if either a gyro or a thruster fails off then the control loop will be broken. It is easy to detect if a sensor has failed off, but without additional sensing (such as temperature sensors) a thruster failure can only be detected through its affect on the plant dynamics. The category of soft failures includes increased noise, periodic outages of either gyro or thruster, scale factor drifts, etc. Any of these problems will cause a change in the closed loop dynamics.

The most serious failure is one in which either a thruster fails on or a gyro fails with a bias. The latter is not easy to detect with boolean logic since the gyro output is changing. The dynamics effect of a sudden failure of this type is that the nutation will jump suddenly, and the spin-axis will be offset, but since the spacecraft is spinning (and the other loop is still working) nutation will damp and the inertial orientation of the spin-axis will not change further.

## Detection Filter Design

The two failure models to be considered in this design are the failure, or degradation of either of the control loops. This leads to two failure models. If the y loop degrades, the value of c in the first row of Equation (21.14) will change. As a consequence the failure model is

$$\dot{\omega} = A\omega + Bu + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \omega_y(t) \quad (21.1)$$

The failure model for the other loop is

$$\dot{\omega} = A\omega + Bu + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \omega_z(t) \quad (21.2)$$

$\omega_y$  and  $\omega_z$  are measured directly, hence the measurement matrix is the identity matrix. The detection filter is

$$\dot{\hat{\omega}} = A\hat{\omega} + D(\omega - \hat{\omega}) \quad (21.3)$$

For unidirectional failure signatures, the eigenvectors of A-D must be

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (21.4)$$

The matrix D that satisfies this constraint is

$$D = \begin{bmatrix} -c - \lambda_1 & -k \\ k & -c - \lambda_2 \end{bmatrix} \quad (21.5)$$

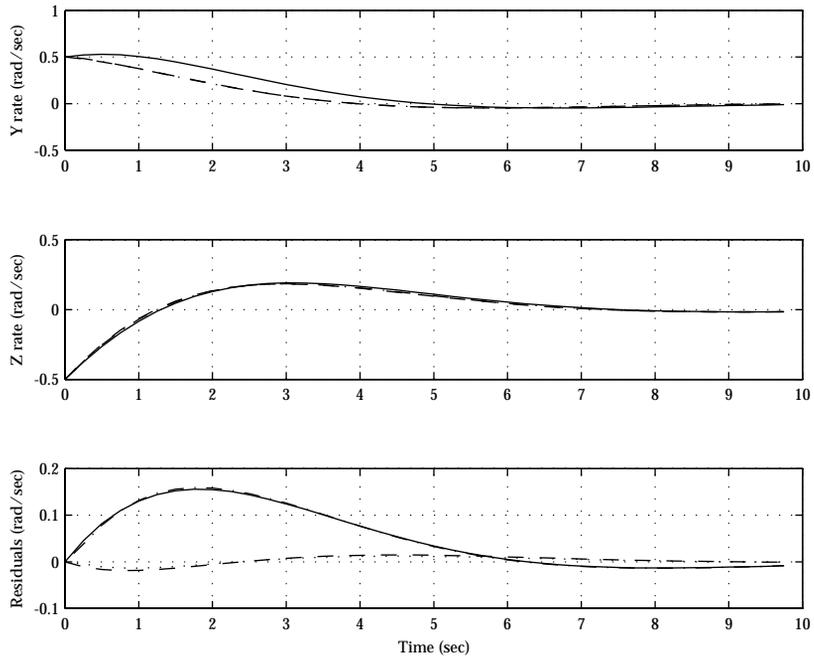
where  $\lambda_1$  and  $\lambda_2$  are the eigenvalues of A-D. For this example, the eigenvalues are chosen to be somewhat more negative than the plant eigenvalues.

## Simulation Results

Figure 21.1 on page 284 shows the detection filter performance when the gain on the y loop decreases by 50%. The simulation shows the performance of the filter when implemented using a zero order hold as in Meserole (1981) and using a first order hold which gives better unidirectionality. The first plot shows the estimated and true y-axis rates. The second plot shows the estimated and true z-axis rates and the last plot shows the measurement residuals. The zero-order-hold implementation is the dashed line and the first-order-hold is the dotted line. The effect on the estimated rate is barely discernible, yet the direction of the residual vector is nearly unidirectional and clearly indicates that the y-loop has a problem. The residual is not exactly unidirectional because of the conversion to a discrete time estimator. The control system eventually damps the rate and the residuals converge. Since there is no further excitation, the y residual also goes to zero since there is no longer a basis for detecting a failure.

FIGURE 21.1 Spinning spacecraft loop failure and detection

---



---

### *22.1 Introduction*

- 22.1. Fortescue, Peter and Stark, John, *Spacecraft Systems Engineering*, John Wiley and Sons, New York, NY, 1995.
- 22.2. French, James R. and Griffin, Michael D., *Space Vehicle Design*, American Institute of Aeronautics and Astronautics, Inc., Washington, DC, 1991.
- 22.3. Larson and Wertz, *Space Mission Analysis and Design*, Microcosm, Inc., Torrance, CA and Kluwer Academic Publishers, Boston, MA, 1991.
- 22.4. Agrawal, B. N., *Design of Geosynchronous Spacecraft*, Prentice-Hall, Englewood Cliffs, NJ, 1986.

---

### *22.2 Dynamics*

- 22.5. Thomson, W. T., *Introduction to Space Dynamics*, Dover, New York, 1986.
- 22.6. Bate, R. R., D. D. Mueller, J. E. White, *Fundamentals of Astrodynamics*, New York, Dover, 1971.
- 22.7. Hughes, P. C., *Spacecraft Attitude Dynamics*, John Wiley & Sons, New York, 1986.
- 22.8. Shabana, A. A., *Dynamics of Multibody Systems*, John Wiley & Sons, New York, 1989.
- 22.9. Weaver, W. Jr., P. R. Johnston, *Structural Dynamics by Finite Elements*, Prentice-Hall, Englewood Cliffs, NJ, 1987.
- 22.10. Kaplan, M. H., *Modern Spacecraft Dynamics & Control*, John Wiley & Sons, New York, 1976.

---

### 22.3 Attitude Estimation

- 22.11. Bryson, A. E., Jr., and Y. Ho, *Applied Optimal Control*, Hemisphere Publishing Corporation, Washington, D.C., p. 367.
- 22.12. Brown, R. G., *Introduction to Random Signal Analysis and Kalman Filtering*, John Wiley & Sons, New York, 1983.
- 22.13. Bierman, G. J., *Factorization Methods for Discrete Sequential Estimation*, Academic Press, Orlando, FL, 1977.
- 22.14. Gelb, A., *Applied Optimal Estimation*, The M.I.T. Press, Cambridge, MA, 1974.

---

### 22.4 Control

- 22.15. Franklin, G. F., J. D. Powell, M. L. Workman, *Digital Control of Dynamic Systems*, 2nd Edition, Addison-Wesley, Reading, MA, 1990.
- 22.16. Bryson, A.E. Jr., Y. Ho, *Applied Optimal Control*, Hemisphere, Washington, D.C., 1975.
- 22.17. Doyle, J. C., B. A. Francis, A. R. Tannenbaum, *Feedback Control Theory*, Macmillan, New York, 1992.
- 22.18. Stevens, B. L., F. L. Lewis, *Aircraft Control and Simulation*, John Wiley & Sons, New York, 1992.
- 22.19. Maciejowski, J. M., *Multivariable Feedback Design*, Addison-Wesley, Reading, MA, 1989.
- 22.20. Åström, K. J., B. Wittenmark, *Computer Controlled Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1984.
- 22.21. Ogata, K., *Modern Control Engineering*, Prentice-Hall, Englewood Cliffs, NJ, 1970.
- 22.22. D’Azzo, J. J., C. H. Houpis, *Linear Control System Analysis & Design*, McGraw-Hill, New York, 1988.
- 22.23. Houpis, C. H., G. B. Lamont, *Digital Control Systems*, McGraw-Hill, New York, 1992.

---

### 22.5 Spacecraft Control

- 22.24. Wertz, J. R., *Spacecraft Attitude Dynamics and Control*, Kluwer, Dordrecht, The Netherlands, 1978.

---

### 22.6 Astrodynamics

- 22.25. Battin, R. H., *An Introduction to the Mathematics and Methods of Astrodynamics*, AIAA, New York, 1987.
- 22.26. Vallado, D. A. and W. D. McClain. (1997.) *Fundamentals of Astrodynamics and Applications*. McGraw-Hill.

---

## 22.7 Propulsion

- 22.27.** Sutton, G. P., D. M. Ross, *Rocket Propulsion Elements*, John Wiley & Sons, New York, 1976.
- 22.28.** Hill, P. G., Peterson, C. R., *Mechanics and Thermodynamics of Propulsion*, Addison-Wesley, Reading, MA, 1970.

---

## 22.8 Real-Time Systems

- 22.29.** Laplante, P. A., *Real-Time Systems Design and Analysis*, IEEE Press, Piscataway, NJ, 1993.

---

## 22.9 Devices

- 22.30.** de Silva, C. W., *Control Sensors and Actuators*, Prentice-Hall, Englewood Cliffs, NJ, 1989.
- 22.31.** Karim, M. A., *Electro-Optical Devices and Systems*, PWS-Kent Publishing Company, Boston, MA, 1990.

---

## 22.10 Software

- 22.32.** Barnes, J. G. P., *Programming in Ada*, Addison-Wesley, Reading, MA, 1991.
- 22.33.** Bennatan, E. M., *On Time Within Budget Software Project Management Practices and Techniques*, QED Publishing Group, Boston, MA, 1992.
- 22.34.** Stroustrup, B., *The C++ Programming Language Second Edition*, Addison-Wesley, Reading, MA, 1991.
- 22.35.** Forsythe, G. E., M. A. Malcolm, C. B. Moler, *Computer Methods for Mathematical Computations*, Prentice-Hall, Englewood Cliffs, NJ, 1977.

---

## 22.11 Ephemeris

- 22.36.** *The Astronomical Almanac*, U.S. Government Printing Office, Washington, D.C., 1993.

---

## 22.12 Heat Transfer

- 22.37.** Thomas, L. C., *Heat Transfer*, Prentice-Hall, Englewood Cliffs, NJ, 1992

---

### 22.13 Testing

- 22.38. Pascault, G. "Telecom 1 Attitude and Orbit Control Subsystem (AOCS) Tests and Results," *IFAC Symposium on Spacecraft Testing*, The Charles Stark Draper Laboratory, August 24, 1983.
- 22.39. Wakasugi, N., T. Suzuki, S. Ichikawa, "MOS-1 ACOS Subsystem Test," *IFAC Symposium on Spacecraft Testing*, The Charles Stark Draper Laboratory, August 24, 1983.
- 22.40. Hamann, R. J., "Concept for a Generalized AOCS Test Strategy," *IFAC Symposium on Spacecraft Testing*, The Charles Stark Draper Laboratory, August 24, 1983.
- 22.41. Teule, F., "IRAS ACS, In Flight Validation," *IFAC Symposium on Spacecraft Testing*, The Charles Stark Draper Laboratory, August 24, 1983.
- 22.42. Bénet, C., D. Gamble, "Testing of a Three-Axis Stabilized Satellite Attitude Control System," *IFAC Symposium on Spacecraft Testing*, The Charles Stark Draper Laboratory, August 24, 1983.
- 22.43. Virdee, L, E. Main, T. Smagala, "Intelsat V Attitude Determination and Control Subsystem Flight Simulator," *IFAC Symposium on Spacecraft Testing*, The Charles Stark Draper Laboratory, August 24, 1983.
- 22.44. Dougherty, H., et. al, "Space Telescope Pointing Control System Simulator," *IFAC Workshop*, Stanford, CA, August 22-26, 1983.
- 22.45. Dougherty, H., et al, "Space Telescope Pointing Control System Software," AAS 82-027, 1982.
- 22.46. Dougherty, H., et al, "Space Telescope Pointing Control," *IFAC Automatic Control in Space*, 1982.
- 22.47. Gottzein, E., H. Bittner, "Closed-Loop Testing of Attitude Control Systems of Geostationary Satellites," *IFAC Symposium on Spacecraft Testing*, The Charles Stark Draper Laboratory, August 24, 1983.
- 22.48. "Rapid prototyping speeds DC-X delivery," *Military & Aerospace Electronics*, June 14, 1993.
- 22.49. Beizer, B., *Software Testing Techniques*, Second Edition, Van Nostrand Reinhold, New York, 1990., p. 26.
- 22.50. Hetzel, W., *The Complete Guide to Software Testing*, QED Information Sciences, Inc., Wellesley, MA, 1984.
- 22.51. IEEE, *IEEE Standard for Binary Floating-Point Arithmetic*, ANSI/IEEE Std. 754-1985, IEEE, New York, August 12, 1985.
- 22.52. USAF, *Military Standard Sixteen-Bit Computer Instruction Set Architecture*, MIL-STD-1750A(USAF), May 21, 1982.
- 22.53. McConnell, S., *Code Complete*, Microsoft Press, Redmond, WA, 1993, p. 615.
- 22.54. Laplante, P. A., *Real-Time Systems Design and Analysis: An Engineer's Handbook*, IEEE Press, Piscataway, NJ, 1993, p. 100.

---

### 22.14 Coordinates

- 22.55. Carroll, J.V., "The Notation and Use of Quaternions for Shuttle Ascent Steering," SSV Memo 10C-75-47, The Charles Stark Draper Laboratory, Inc., November 21, 1975.

---

## 22.15 Failures

- 22.56. McLean, D., and Benkhedda, H. (1994.) An analytical redundancy scheme for an automatic flight control system. *Aerospace Vehicle Dynamics and Control*, Oxford University Press, Oxford, 145-160.
- 22.57. Patton, R.J. et al. (1987). Parameter insensitive techniques for fault-detection. *J. Guidance and Control*, 10(4), 359-367.
- 22.58. Freidland, B. (1982). Maximum likelihood failure detection of aircraft flight control sensors. *J. Guidance and Control*, 5(5), 498-503.
- 22.59. Merrill, W.C. (1985). Sensor failure detection for jet engines using analytical redundancy. *J. Guidance and Control*, 8(6), 673-682.
- 22.60. Handleman, D.A. and Stengel, R.F. (1989). Combining expert system and analytical redundancy concepts for fault-tolerant flight control. *J. Guidance and Control*, 12(1), 39-45.
- 22.61. Gertler, J.J. (1988). Survey of Model-Based Failure Detection and Isolation in Complex Plants. *IEEE Control Systems Magazine*, 8(6), 3-11.
- 22.62. Naidu, S.R., et al. (1990). Use of Neural Networks for Sensor Failure Detection in a Control System. *IEEE Control Systems Magazine*, 10(3), 49-55.
- 22.63. Stengel, R.F. (1991). Intelligent Failure-Tolerant Control. *IEEE Control Systems Magazine*, 11(4), 14-23.
- 22.64. Farrell, J. et al. (1993). Using Learning Techniques to Accommodate Unanticipated Faults. *IEEE Control Systems*, 13(3), 40-49.
- 22.65. Willsky, A. S., (1976) A Survey of Design Methods for Failure Detection in Dynamic Systems. *Automatica*, 12, 601-611.
- 22.66. Isermann, R., (1984) Process Fault Detection Based on Modeling and Estimation Methods - A Survey, *Automatica*, 20, 387-404.
- 22.67. Beard, R. V. (1971), "Failure Accommodation in Linear Systems Through Self-Reorganization," Report. MVT-71-1, *Man Vehicle Laboratory*, M.I.T., Cambridge, MA.
- 22.68. Jones, H. L. (1973), "Failure Detection in Linear Systems," Ph.D. Thesis, Dept. of Aeronautics and Astronautics, M.I.T., Cambridge, MA.
- 22.69. Meserole, J. S., Jr. (1981), "Detection Filters for Fault-Tolerant Control of Turbofan Engines," Ph. D. Thesis, Dept. of Aeronautics and Astronautics, M.I.T., Cambridge, MA.
- 22.70. Andry, A. N., Shapiro, E. Y. and Chung, J. C. (1983). Eigenstructure Assignment for Linear Systems, *IEEE Transactions on Aerospace Electronic Systems*, AES-19, 711-729.
- 22.71. Stein, G. (1979), "Generalized Quadratic Weights for Asymptotic Regulator Properties," *IEEE Transactions on Automatic Control*, 24, 559-566.
- 22.72. Gupta, N. K. (1980), "Frequency-Shaped Cost Functionals: Extensions of Linear-Quadratic-Gaussian Design Methods," *Journal of Guidance and Control*, 3, 529-535.
- 22.73. Gelb, A., ed. (1974). *Applied Optimal Estimation*. Cambridge, Mass.: M.I.T. Press.
- 22.74. Patton, R.J. and Chen, J. (1992). Robust Fault Detection of Jet Engine Sensor Systems Using Eigenstructure Assignment. *J. Guidance, Control and Dynamics*, 15(6), 1491-1497.

---

## References

---

- 22.75. McKillip, R. M., Jr., Curtiss, H. C., Jr., and Xiao, Z. (1988), Studies in Interactive System Identification of Helicopter Rotor/Body Dynamics Using an Analytically Based Linear Model. *Proc. International Conference on Helicopter handling Qualities and Control*, London, U.K., November.
- 22.76. Terano, T., Asai, K., and Sugeno, M. (1992). *Fuzzy Systems Theory and its Applications*. Academic Press, Boston, MA.
- 22.77. Teule, F. (1983). IRAS ACS, in flight validation. *IFAC Workshop on Simulation and Validation Techniques to Establish Spacecraft Control Performance*, Stanford, CA.
- 22.78. Parkes, A.G. (1984). Reconfigurable Spacecraft Systems, Autonomous and Non-autonomous. *IFAC Workshop on Reconfigurable Spacecraft Systems, Autonomous and Non-Autonomous*, Cambridge, MA.
- 22.79. Ianculescu, G.D. and Bramble, M.A. (1984). Software design and implementation of the attitude control fault protection for Galileo. *IFAC Workshop on Reconfigurable Spacecraft Systems, Autonomous and Non-Autonomous*, Cambridge, MA.
- 22.80. Cikanek, H.A., III (1986). Space Shuttle Main Engine Failure Detection. *IEEE Control Systems Magazine*, 6(3), 13-18.
- 22.81. Murugesen, S. and Goel, P.S. (1989). A Scheme for Fault Tolerance in Earth Sensors. *IEEE Transactions on Aerospace and Electronic Systems*, 25(1), 21-30.
- 22.82. Dougherty, H., et al. (1984). Space Telescope Safemode System for Spacecraft Reconfiguration. *IFAC Workshop on Reconfigurable Spacecraft Systems, Autonomous and Non-Autonomous*, Cambridge, MA.

TABLE 23.1 Acronyms

Acronym	Meaning
A/D	Analog to Digital
ADCS / ADACS	<u>A</u> ttitude <u>D</u> etermination and <u>C</u> ontrol <u>S</u> ystem
AKM	Apogee Kick Motor
BOL	<u>B</u> eginning <u>O</u> f <u>L</u> ife
CAD	<u>C</u> omputer <u>A</u> ided <u>D</u> esign
CMG	<u>C</u> ontrol <u>M</u> oment <u>G</u> yro
D/A	Digital to Analog
ECI	<u>E</u> arth <u>C</u> entered <u>I</u> ntertial (coordinate frame)
EF	Earth-Fixed (coordinate frame)
EHT	<u>E</u> lectrothermal <u>H</u> ydrazine <u>T</u> hruster

**TABLE 23.1 Acronyms**

<b>Acronym</b>	<b>Meaning</b>
<b>EPSF</b>	<u>E</u> ncapsulated <u>P</u> ostscript <u>F</u> ile
<b>ESA</b>	<u>E</u> arth <u>S</u> ensor <u>A</u> ssembly
<b>FOG</b>	<u>F</u> iber <u>O</u> ptic <u>G</u> yro
<b>HST</b>	<u>H</u> ubble <u>S</u> pace <u>T</u> elescope
<b>I&amp;T</b>	<u>I</u> ntegration and <u>T</u> esting
<b>IEEE</b>	<u>I</u> nstitute of <u>E</u> lectrical and <u>E</u> lectronics <u>E</u> ngineers
<b>J2000</b>	Mean of Aries 2000 Reference
<b>JD</b>	<u>J</u> ulian <u>D</u> ate
<b>LAE</b>	<u>L</u> iquid <u>A</u> pogee <u>E</u> ngine
<b>LEO</b>	<u>L</u> ow <u>E</u> arth <u>O</u> rbital
<b>LQ</b>	<u>L</u> inear <u>Q</u> uadratic
<b>LQG</b>	<u>L</u> inear <u>Q</u> uadratic <u>G</u> aussian
<b>LQE</b>	<u>L</u> inear <u>Q</u> uadratic <u>E</u> stimator
<b>LTR</b>	<u>L</u> oop <u>T</u> ransfer <u>R</u> ecovery
<b>LVLH</b>	<u>L</u> ocal <u>V</u> ertical <u>L</u> ocal <u>H</u> orizontal (coordinate frame)
<b>MIMO</b>	<u>M</u> ulti- <u>I</u> nterface <u>M</u> ulti- <u>O</u> utput
<b>MWA</b>	<u>M</u> omentum <u>W</u> heel <u>A</u> ssembly
<b>OMS</b>	<u>O</u> rbital <u>M</u> aneuvering <u>S</u> ystem
<b>PDF</b>	<u>P</u> ortable <u>D</u> ocument <u>F</u> ormat
<b>PPT</b>	<u>P</u> ulsed <u>P</u> lasma <u>T</u> hruster
<b>PSS</b>	<u>P</u> inceton <u>S</u> atellite <u>S</u> ystems
<b>REA</b>	<u>R</u> ocket <u>E</u> ngine <u>A</u> ssembly

---

**TABLE 23.1 Acronyms**

<b>Acronym</b>	<b>Meaning</b>
<b>RF</b>	<u>R</u> adio <u>F</u> requency
<b>RMA</b>	<u>R</u> ate <u>M</u> easuring <u>A</u> ssembly
<b>RPY</b>	<u>R</u> oll <u>P</u> itch and <u>Y</u> aw
<b>RWA</b>	<u>R</u> eaction <u>W</u> heel <u>A</u> ssembly
<b>SISO</b>	<u>S</u> ingle- <u>I</u> nput <u>S</u> ingle- <u>O</u> utput



---



# *Index*

---

## **Numerics**

1750A 35  
80386 35

## **A**

AC-100 261  
acquisition 202, 203  
acquisition mode 236  
actuator 39  
adaptive control 83  
Aerodynamic 185  
AKM 151  
Albedo Pressure 185  
analytical result 268  
angle encoder 41  
angular momentum 219  
antenna 187  
AOCS Bench Test Equipment 252  
apogee 218  
Apogee Kick Motor 269  
apogee kick motor 205  
Apollo 249  
Ariane 42  
articulated platform 36  
atmospheric radiance 215  
attitude control 203  
attitude control system 35  
Attitude Determination and Control Electronics 257  
azimuth angle 217

## B

$\beta$  238  
barycenter 57  
beam center 201  
beam center offset 148  
beam pointing accuracy 148  
best guess 268  
bias momentum 38, 221, 229  
bipropellant thruster 39  
black box testing 268  
boresight 243  
boundary value 268  
bright star sensor 240

## C

CCD 154  
chordwidth 217, 242  
classical control 83  
closed-loop dynamics 268  
CMG 166  
coarse sun sensor 253, 254  
command and telemetry 266  
communications satellite 250  
compiler bugs 267  
complementary filter 40  
conical scanning earth sensor 240, 242  
conjugate gradient 199  
control moment gyros 40  
control spillover 223  
control system 266  
control system requirements 37  
coverage monitor tool 269  
coverage tool 269  
cross links 204

## D

damper wheel 222  
DC motor 227  
defocused star image 154  
Delta 42  
delta V 218, 223  
digital control 246  
dihedral angle 217  
Draper Laboratory 249  
drift orbit maneuver 151  
dual spin 38  
dual spin spacecraft 36  
dual spin turn 181, 210, 219  
DUT1 58  
dynamic compensator 229  
Dynamic imbalance 189

## E

earth 155

---

Earth Centered Inertial Frame 59  
earth sensor 41, 157, 220, 225, 226, 239, 240  
earth simulator 255  
eccentricity 144  
ECI 236  
electrothermal hydrazine thruster 151  
energy dissipation 213, 214  
ephemeris 239  
equinoctial elements 144  
estimator 229  
Euler angle 60, 64  
Euler coupling 246  
Evans form 222  
external disturbance 246

## **F**

failure 36  
far-field 188  
Fine Digital Sun Sensor Electronics 257  
fine sun sensor 253  
first order hold 183  
flexible appendage 223  
flight vehicle 266  
Fokker 253  
FORTRAN 266  
Friction 189  
    Coulomb 189  
    stiction 189  
    turbulent 189  
    viscous 189  
friction 245

## **G**

gas constant 170  
geosynchronous 206, 250  
geosynchronous orbit 218  
geosynchronous satellite 37  
geosynchronous spacecraft 150  
GPS 240  
gravitational parameter 143  
Gravity gradient 186  
gravity gradient 220  
gravity-gradient 38  
great circle precession 216  
gyro 35, 41, 215, 216, 225, 226, 227, 231, 239

## **H**

half system 170  
Hamiltonian 242  
Helium 170  
hinge 133  
horizon sensor 41, 155, 202, 215, 253  
HS601 41  
Hydrazine 170

**I**

inclination 144  
inertial frame 61  
Infrared Astronomy Satellite 253  
INSAT I 254  
insurance 42  
Integrated Systems 261  
integration and test 266  
Intel 80960 261  
Intelsat V 257  
IRES 250  
IUS 201

**J**

J2000.0 59, 236  
jitter 36

**K**

Kalman Filter 197, 200, 216  
Keplerian elements 143

**L**

Lagrange multiplier 245  
Leaks 186  
libration damper 38  
linear control 214  
linear control system 39  
linear programming 232  
liquid fuel motor 202  
liquid propellant 213  
Local Vertical Local Horizontal Frame 236  
longitude of the ascending node 143

**M**

magnetic field 187, 225, 246  
magnetic torquer 35, 39, 203, 209, 225, 226, 232, 241, 246, 251  
magnetometer 41, 253  
maintenance 42  
major axis spinner 213  
Matlab 266  
Matra 250  
MATRIX 261  
matrix  
    addition 47  
    multiplication 48  
    skew-symmetric 48  
    symmetric 48  
MBB 258  
midnight 239  
minor axis spinner 213  
mirror 157, 240  
mission orbit 38, 202, 205, 209  
modal coordinates 224  
Module testing 268

---

momentum 223, 238, 246, 247  
momentum bias 35, 204, 210, 221, 235  
momentum wheel 37, 39, 209, 225, 226, 227, 251  
    torque 135  
momentum wheel motor 209, 219, 225  
Motor torques and forces 189  
Moving parts 189  
MSTI 261  
multibody equations 133

## **N**

nadir 60  
nadir angle 242  
nadir vector 242  
navigation 246  
near-field 188  
noise 227  
nonminimum phase 230  
noon 239  
noon-midnight turn 238  
numerical integration 245  
numerical simulation 181  
nutation 220, 223, 230, 232  
nutation frequency 221

## **O**

off pulse modulation 232  
operator training 258  
orbit  
    inclination 151  
orbit geometry 236  
orbit plane 236  
Orbital Maneuvering System 39  
orthogonal wheels 245  
orthonormal matrix 60  
Outgassing 186

## **P**

Padé approximant 230  
parallax 60  
payload 201  
perigee 143  
photon 187  
pitch 204  
pivoted wheel 39, 40  
pixel 155  
pixel map 154  
Pointing Budget  
    Temporal Category  
        Bias 147  
        Diurnal 147  
        Long Term 147  
        short term 147

pointing budget 147  
Pointing Budgets  
  Temporal Category 147  
potentiometer 41  
power consumption 37  
power subsystem 263  
precession 214  
pressurant tank 170  
propellant pressurization 215  
proportional integral controller 228  
proportional integral differential controller 231, 232  
proportional-integral-differential 246  
Proton booster 201  
pulsewidth modulation 39, 214

## Q

quaternion 60, 62, 64  
  derivative 68  
  multiplication 67

## R

radiation hardness 37  
Radiation Pressure 186  
random input 268  
random variable 147  
rate gyro 254  
RCA Astro Electronics 206  
Reaction wheel 164  
reaction wheel 40, 203, 238, 241, 245, 246, 253, 254  
reaction wheel assembly 258  
recursive least squares 197  
redundancy 36  
redundancy management 250  
reference frame 59  
reorientation 216  
requirements 36, 201, 268  
residual dipole 220  
RF 186  
RF feed 187  
rhumb line precession 217  
roll 204  
Rotor imbalances 189  
Rotors 189  
RWA 166

## S

saturation 214  
scale factor 37  
SCSim 267  
semi major axis 144  
servo table 255  
Sidereal Time 57  
simplex 232  
simulation 202

---

single-input single-output 83  
skew dipole 229  
Slosh 189  
solar array 186, 205, 236, 238, 239  
solar array flexibility 221  
solar array pitch 243  
Solar Pressure 186  
solar pressure 187, 203, 220  
solid motor 202  
solid rocket 150  
Space Shuttle 249  
Space Shuttle Orbiter 39  
Space Telescope 36, 258  
spacecraft dynamics 181  
spacecraft life 201  
spacecraft midnight 235, 239  
spacecraft noon 235, 239  
spin axis attitude determination 215  
spin precession maneuver 151, 216, 232  
spinner 38, 206  
    major axis 203  
    minor axis 203  
spinning spacecraft 213  
spot beam 204  
star image 154  
star sensor 35, 36, 41, 239  
state space 181  
static earth sensor 156, 226  
stationkeeping 35, 203, 204, 209, 236, 254, 266  
steepest descent 199  
stepping motor 246  
    claw tooth permanent magnet 162  
    hybrid 162  
    permanent magnet 162  
    variable reluctance 162  
sun nadir pointing 235, 239  
sun sensor 35, 41, 202, 205, 215, 239  
sun sensor eye 243  
sun simulator 255  
sun vector 235, 237, 238  
sun/earth acquisition. 206  
System testing 269

## T

Table Test Equipment 255  
tach loop 246  
tachometer loop 40, 227  
TAI 57  
TDB 57  
TDT 57  
Telecom 1 250  
thermal 220, 266  
Thermal Pressure 186  
Thermal snap 189

thermal subsystem 263  
three-axis motion simulator 255  
throttleable thruster 223  
thruste 35  
thruster 39, 168, 203, 214, 225, 232, 241, 251  
    bipropellant 203  
    electrothermal 203  
    monopropellant 203  
thruster plume 188, 220  
topological tree 133  
transfer orbit 38, 202, 205, 209  
transformation matrix 237  
true anomaly 144

## **U**

Unit testing 268  
universal time 59  
UTI 57  
UTC 57

## **V**

vacuum 266  
video camera 154

## **W**

waveguide 187  
wheel speed demand 228  
white box testing 268  
windage 245

## **Y**

yaw 204, 235, 238

## **Z**

zenith 60  
zero momentum 35, 38  
zero order hold 183