RICE UNIVERSITY

Web-based Control of the Rice SPENDULAP:
A Web-enabled Telemechatronic System

by

Joshua McCune

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE
MASTER OF SCIENCE

APPROVED, THESIS COMMITTEE:

Fathi H. Ghorbel, Chair
Associate Professor of Mechanical Engineering

Marcia K. O’Malley
Assistant Professor of Mechanical Engineering

John W. Clark
Professor of Electrical Engineering

James B. Dabney
University of Houston - Clear Lake
Assistant Professor of Systems Engineering

Houston, Texas
January, 2003
ABSTRACT

Web-based Control of the Rice SPENDULAP: A Web-enabled Telemechatronic System

by

Joshua McCune

The recent development of sophisticated Internet architectures has led to the proliferation of telerobotic operated systems, ranging from education driven, resource sharing robots to life saving, telemedical robots, to risk avoidance robots employed by the military to reduce human peril. Currently, these systems are controlled and maintained via complex software downloaded to the client’s computer. The web-architecture presented within this thesis demonstrates advances in telemechatronic control that allows control and analysis of a system by a client-side computer without the necessity of external software. The Rice Spherical Pendulum Apparatus (SPENDULAP) is the mechatronic system utilized by the architecture to show how dynamics and controls experiments can be safely performed and monitored by anybody, anywhere, with a simple computer and internet browser.
Acknowledgments

First and foremost I would like to thank Fathi Ghorbel and James Dabney for making this opportunity possible. Their guidance, support and wisdom made all of this possible. Secondly, I would like to thank Joe Gesenhues – without his machining and wiring skills the SPENDULAP would not have been created. Also, a very big thanks goes to Kevin Bowen, Benoit Brion, Jason White, Paul Leu, Remzi Artar, Wayde Shipman and Igor Karpov. Without their physical and design talents, the SPENDULAP project would not have come to fruition in such a short period of time.

I want to thank my labmates, Kun Lu, Sushant Dutta, and Zhiyong Wang for suffering the noise and disturbances created by the SPENDULAP and always maintaining a positive humor even when sometimes I did not. Thanks guys.

I would also like to give special thanks to Marcia O’Malley and John Clark for taking time out of their busy schedules to be a part of my defense committee.

Finally, I want to thank the light of my life, Christina, for suffering me outside of the lab. Her stress was twice again as mine, but she always managed to keep me pointed in the right direction even when her own path was difficult to steer.
Contents

Abstract ii
Acknowledgments iii
List of Tables vii
List of Figures ix

1 Introduction 1

1.1 Motivation 2
1.2 Terminology 3
1.3 Historical Background 7
1.4 System Description 9
1.5 Outline of Thesis 12

2 Dynamics Formulation and Analysis 14

2.1 Kinematic Analysis 14
2.2 Dynamics 17
   2.2.1 Ignorable Coordinates 20
2.3 Dynamics Analysis 21
3 Control Algorithms and Issues

3.1 Pole Placement .................................................. 30
3.2 Input State Linearization ..................................... 33
3.3 Sliding Mode ...................................................... 36
  3.3.1 Continuous Sliding Mode ................................. 40
3.4 Summary ......................................................... 41

4 Web Operational Architecture ................................. 44

4.1 Operational Scenarios .......................................... 45
  4.1.1 Constant Frame Angular Velocity Experiments .... 46
  4.1.2 Feedback Control Experiments ......................... 47
4.2 Web System Architecture ...................................... 48
  4.2.1 Web-based Commander Scenario ....................... 50
  4.2.2 Web-based Spectator Scenario ........................ 52
  4.2.3 Example .................................................... 54
4.3 LabVIEW ......................................................... 54
5 Conclusions

References

A BLDCM Dynamics

B Friction Analysis

C LabVIEW Software
List of Tables

1.1 SPENDULAP control and dynamics parameters .................. 4
1.2 Greek symbols used in SPENDULAP equations .................. 5
1.3 BLDCM variables .................................................. 5
1.4 Acronym List ....................................................... 6
2.1 Moving reference frames [1] ....................................... 16
4.1 Experiment data .................................................... 54
A.1 Motor dynamics variable definitions ............................. 66
A.2 DDR motor parameters .............................................. 68
C.1 Main VIs .............................................................. 73
C.2 SPENDULAP logic and parameter VIs ............................ 73
C.3 Pendulum VIs ......................................................... 74
C.4 Torque calculation and conversion VIs ............................ 74
C.5 Sliding mode VIs ..................................................... 74
C.6 DDR VIs .............................................................. 75
C.7 Data formatting VIs .................................................. 75
C.8 Miscellaneous VIs ........................................ 75
List of Figures

1.1 System communication physical architecture .............................................. 8

1.2 Rice SPENDULAP layout ................................................................. 10

1.3 Encoder, clutch, and brake mounted on the pendulum swivel pin ... 10

1.4 SPENDULAP initial positioning mechanism, consisting of lift motor,
gears, pulley, clutch, and brake (brake not shown) ................................. 11

1.5 Brushless direct-drive DC motor and slipring ..................................... 12

2.1 SPENDULAP kinematic diagram ....................................................... 15

2.2 SPENDULAP notational diagram .................................................... 19

2.3 Equilibrium pendulum deflection .................................................... 23

2.4 Supercritical pitchfork bifurcation plot .......................................... 29

3.1 Pole Placement: \( \theta_{init} = 85^o, \theta_{des} = 30^o, Poles = [-4, -5, -9] \) .... 34

3.2 Input State Linearization: \( \theta_{init} = 85^o, \theta_{des} = 30^o, K_p = 100, K_d =
20, K_\phi = 25 \) ........................................................................... 36

3.3 Discrete Sliding Mode: \( \theta_{init} = 85^o, \theta_{des} = 30^o, K_p = 100, K_d = 20, K_b =
200 \) .................................................................................. 39

3.4 Discrete and continuous sliding mode graphical definition ............. 40
3.5 Comparison of DSM versus CSM: \( H = 0.7 \)  ........................................... 41

3.6 Continuous Sliding Mode: \( \theta_{init} = 85^\circ, \theta_{des} = 30^\circ, K_p = 100, K_d = 20, K_b = 200, H = 0.2 \) ........................................... 42

4.1 SPENDULAP telemechatronic communication diagram ............................. 44

4.2 Data and control flow software architecture ........................................... 49

4.3 Commander’s user interface display ....................................................... 52

4.4 Spectator user interface display ............................................................... 53

4.5 Commander display after data entry ....................................................... 55

4.6 Commander display after experiment ..................................................... 56

4.7 Front Side: Real Time LabVIEW VI that controls inputs and outputs of the SPENDULAP .............................................................. 57

4.8 Front Side: LabVIEW VI that defines inputs and output torque to SPENDULAP .............................................................. 57

A.1 Comparison of simulation versus experiment commanded velocity for the SPENDULAP ............................... 68

B.1 SPENDULAP viscous friction: simulation versus experiment ............... 72
Chapter 1
Introduction

Today’s ever-expanding world of telecommunications gives us the ability to control mechatronic systems from a thousand miles away. The Internet provides the potential for the implementation of a variety of telemechatronic systems, ranging from the everyday application such as turning off lights in your home from the office, to the cutting-edge, whether it be a remotely operated military spy plane, or an intricate surgery performed using robotic hands in New York by a doctor in Singapore.

The ability to witness and control events telemechatronically is of ever increasing importance as more systems become linked to and controlled via the Internet. The advent of this technology has two enormous effects. With the Internet, remote monitoring and unattended operation of infeasible or dangerous systems (e.g., hazardous chemical systems, oil-wells established in foreign countries, precarious military operations, etc.) has become a reality. Secondarily, the resource sharing provided by the Internet architecture facilitates dissemination of knowledge. Previously, an expensive mechatronic system would be experimented on by a few scientists within physical travelling range of the system. With telemechatronic systems, researchers and students are afforded the ability to explore an ever-expanding universe of dynamic systems and controls by giving them access to mechatronic systems they otherwise wouldn’t be able to experiment upon.
1.1 Motivation

It is apparent that telemechatronic systems are not just important in today's world, but necessary. The study and development of improved processes for telerobotic operation is, thus, of significant import. Currently, several examples of telemechatronic systems exist [2, 3, 4]. These systems require the Internet user, or hereafter called the web-engineer, to either import or already have non-standard software on their computers to interface with the systems.

Several challenges face the developers of telemechatronic systems. The first challenge is to provide safe, reliable real-time control in spite of the unpredictable transport lags exhibited by the Internet. The second is the provision of a high-quality, robust graphical user interface (GUI) that does not require custom software on the client side. (Custom software presents severe configuration management issues and hardware and operating system compatibility problems.) Finally, the architecture must allow for multiple users, ensure that active control is exercised by only one remote user at a time, and manage access security.

This thesis presents an architecture that solves all these issues. Real-time control is exercised by a dedicated real-time control processor installed in a host personal computer. This controller features easily implementable safety features that prevents unsafe operation.

The second and third challenges are addressed through the use of a Java-based client GUI that runs on standard web browsers. Thus, all configuration management
is handled through the Java server facility and remote client software is machine-independent.

Finally, the architecture manages user interaction by allowing for a specified number of spectators and a single commander at any time. The server side software handles authentication and manages the allocation of commander rights for specified time intervals.

The architecture is demonstrated using the Rice Spherical Pendulum Laboratory Apparatus (SPENDULAP). This is a telemechatronic system configured for continuous unattended operation. The SPENDULAP is an ideal testbed for this architecture due to its interesting dynamics and wide range of compatible control algorithms.

1.2 Terminology

Throughout this thesis several variables and acronyms will be used in defining the parameters of the SPENDULAP. These terms are listed alphabetically for reference and are found in Tables 1.1, 1.2, 1.3 and 1.4. Also given are the common values of the variables or where they can be found.*

---

*Many of the variables are defined to coincide with variable terminology used originally in [1, 5, 6]
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Applicable Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>Matrix used in the state equation $\dot{x} = Ax + Bu$</td>
<td>System Dependent</td>
</tr>
<tr>
<td>$B$</td>
<td>Matrix used in the state equation $\dot{x} = Ax + Bu$</td>
<td>System Dependent</td>
</tr>
<tr>
<td>$g$</td>
<td>Gravitational constant</td>
<td>9.81 m/s$^2$</td>
</tr>
<tr>
<td>$I_x$</td>
<td>Inertia about the x axis</td>
<td>0.00845 kg-m$^2$</td>
</tr>
<tr>
<td>$I_{Fz}$</td>
<td>Inertia about the z (i.e., rotational) axis</td>
<td>0.01384 kg-m$^2$</td>
</tr>
<tr>
<td>$K$</td>
<td>Gain matrix defined for the control $u = -Kx$</td>
<td>Control Dependent</td>
</tr>
<tr>
<td>$K_\phi$</td>
<td>Gain for $\dot{\phi}$ in input state linearization control</td>
<td>User Defined</td>
</tr>
<tr>
<td>$K_h$</td>
<td>Gain for $\dot{\phi}$ in sliding mode control</td>
<td>125 - 175</td>
</tr>
<tr>
<td>$K_d$</td>
<td>Gain for $\dot{\theta}$ in nonlinear control</td>
<td>User Defined</td>
</tr>
<tr>
<td>$K_p$</td>
<td>Gain for $\theta$ in nonlinear control</td>
<td>User Defined</td>
</tr>
<tr>
<td>$L$</td>
<td>Lagrangian value</td>
<td>$T - V$</td>
</tr>
<tr>
<td>$l_{bob}$</td>
<td>Length to the bob's center of mass from the swing shaft</td>
<td>0.254 m</td>
</tr>
<tr>
<td>$l_{rod}$</td>
<td>Length of the pendulum shaft</td>
<td>0.235 m</td>
</tr>
<tr>
<td>$m_{bob}$</td>
<td>Mass of the bob located at the end of the pendulum shaft</td>
<td>0.117 kg</td>
</tr>
<tr>
<td>$m_{rod}$</td>
<td>Mass of the pendulum shaft</td>
<td>0.044 kg</td>
</tr>
<tr>
<td>$Q$</td>
<td>Generalized coordinate vector</td>
<td>See Dynamics</td>
</tr>
<tr>
<td>$q$</td>
<td>Rearranged state vector for control</td>
<td>See Control</td>
</tr>
<tr>
<td>$T$</td>
<td>Kinetic energy</td>
<td>$T_T + T_P$</td>
</tr>
<tr>
<td>$T_F$</td>
<td>Frame kinetic energy</td>
<td>See Dynamics</td>
</tr>
<tr>
<td>$T_P$</td>
<td>Pendulum kinetic energy</td>
<td>See Dynamics</td>
</tr>
<tr>
<td>$V$</td>
<td>Potential energy</td>
<td>$V_T + V_P$</td>
</tr>
<tr>
<td>$V_F$</td>
<td>Frame potential energy</td>
<td>0</td>
</tr>
<tr>
<td>$V_P$</td>
<td>Pendulum potential energy</td>
<td>See Dynamics</td>
</tr>
<tr>
<td>$W$</td>
<td>Work done by nonconservative forces</td>
<td>See Dynamics</td>
</tr>
<tr>
<td>$x$</td>
<td>Vector containing the states of the SPENDULAP</td>
<td>See Control</td>
</tr>
</tbody>
</table>

Table 1.1 SPENDULAP control and dynamics parameters
<table>
<thead>
<tr>
<th>Variable:</th>
<th>Description:</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Omega$</td>
<td>Constant rotational rate of frame</td>
</tr>
<tr>
<td>$\phi$</td>
<td>Rotation angle of frame</td>
</tr>
<tr>
<td>$\dot{\phi}$</td>
<td>Rotational rate of frame</td>
</tr>
<tr>
<td>$\theta$</td>
<td>Pendulum angle</td>
</tr>
<tr>
<td>$\dot{\theta}$</td>
<td>Pendulum angular rate</td>
</tr>
<tr>
<td>$\tau$</td>
<td>Torque delivered to SPENDULAP</td>
</tr>
</tbody>
</table>

Table 1.2  Greek symbols used in SPENDULAP equations

<table>
<thead>
<tr>
<th>Variable:</th>
<th>Description:</th>
<th>Applicable Value:</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d$</td>
<td>Direct axis coordinate</td>
<td>N/A</td>
</tr>
<tr>
<td>$K_d$</td>
<td>Damping coefficient</td>
<td>0.0112 N-m/s</td>
</tr>
<tr>
<td>$k_e$</td>
<td>Motor constant</td>
<td>$\sqrt{3/2}k_t$</td>
</tr>
<tr>
<td>$k_t$</td>
<td>Motor constant</td>
<td>268 V/kRPM</td>
</tr>
<tr>
<td>$I_i$</td>
<td>Motor current</td>
<td>N/A</td>
</tr>
<tr>
<td>$J$</td>
<td>System inertia</td>
<td>N/A</td>
</tr>
<tr>
<td>$L$</td>
<td>Motor inductance</td>
<td>18.6 mH</td>
</tr>
<tr>
<td>$M$</td>
<td>Torque to current multiplier</td>
<td>-5, -7.2</td>
</tr>
<tr>
<td>$n$</td>
<td>Number of magnetic pole pairs</td>
<td>8</td>
</tr>
<tr>
<td>$q$</td>
<td>Quadrature axis coordinate</td>
<td>N/A</td>
</tr>
<tr>
<td>$R$</td>
<td>Armature resistance</td>
<td>4.83 $\Omega$</td>
</tr>
<tr>
<td>$V,v$</td>
<td>Voltage input to motor</td>
<td>N/A</td>
</tr>
<tr>
<td>$\Omega$</td>
<td>Constant rotational velocity</td>
<td>N/A</td>
</tr>
<tr>
<td>$\omega$</td>
<td>BLDCM rotational velocity</td>
<td>N/A</td>
</tr>
<tr>
<td>$\tau$</td>
<td>Electromagnetic torque</td>
<td>See BLDCM Dynamics</td>
</tr>
<tr>
<td>$\tau_f$</td>
<td>Frictional torque</td>
<td>0.61 N-m</td>
</tr>
<tr>
<td>$\tau_l$</td>
<td>Load torque</td>
<td>See BLDCM Dynamics</td>
</tr>
<tr>
<td>$\theta$</td>
<td>BLDCM rotation angle</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 1.3  BLDCM variables
<table>
<thead>
<tr>
<th>Acronym:</th>
<th>Definition:</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLDCM</td>
<td>BrushLess DC Motor</td>
</tr>
<tr>
<td>CSM</td>
<td>Continuous Sliding Mode</td>
</tr>
<tr>
<td>DAC</td>
<td>Data Acquisition Card</td>
</tr>
<tr>
<td>DDBLDCM</td>
<td>Direct Drive BLDCM</td>
</tr>
<tr>
<td>DSM</td>
<td>Discrete Sliding Mode</td>
</tr>
<tr>
<td>EI</td>
<td>Experiment Information</td>
</tr>
<tr>
<td>EOM</td>
<td>Equations Of Motion</td>
</tr>
<tr>
<td>FBD</td>
<td>Free Body Diagram</td>
</tr>
<tr>
<td>IRF</td>
<td>Inertial Reference Frame</td>
</tr>
<tr>
<td>ISL</td>
<td>Input State Linearization</td>
</tr>
<tr>
<td>LM</td>
<td>Lagrangian Method</td>
</tr>
<tr>
<td>LRTP</td>
<td>LabVIEW Real Time Program</td>
</tr>
<tr>
<td>LTI</td>
<td>Linear Time Invariant</td>
</tr>
<tr>
<td>MRF</td>
<td>Moving Reference Frame</td>
</tr>
<tr>
<td>NM</td>
<td>Newtonian Method</td>
</tr>
<tr>
<td>OS</td>
<td>Operational Scenario</td>
</tr>
<tr>
<td>PP</td>
<td>Pole Placement</td>
</tr>
<tr>
<td>RT</td>
<td>Real Time</td>
</tr>
<tr>
<td>SHC</td>
<td>Spendulap Host Computer</td>
</tr>
<tr>
<td>SPENDULAP</td>
<td>Spherical PENDULum APparatus</td>
</tr>
<tr>
<td>VI</td>
<td>Visual Instrument</td>
</tr>
<tr>
<td>DAC</td>
<td>Data Acquisition Card</td>
</tr>
</tbody>
</table>

Table 1.4  Acronym List
1.3 Historical Background

In 1994, Professor Fathi Ghorbel developed the idea of the SPENDULAP as a homework assignment for his Kinematics, Dynamics and Controls course. The next year, the SPENDULAP was transformed into a class project. The students were responsible for deriving and simulating the dynamics and control of the SPENDULAP using the basic principles of control [7]. The SPENDULAP consisted, in theory, of a stiff pendulum that started at some initial angle and, as it was released, a motor provided a torque function that rotated the pendulum around its central axis in such a way that a desired angle was reached. The project called on the understanding of several topics in kinematics, dynamics, and control, including underactuation, rotating reference frames, trajectory control and other interesting properties.

In 1996, a group of senior undergraduate mechanical engineers at Rice University, under the leadership of Professor Ghorbel, created the first prototype of the SPENDULAP. However, it turned out that the initial positioning mechanism, a solenoid-actuated disk brake, was weak and not autonomous. Another problem was found in the motor that rotated the SPENDULAP. It provided insufficient power and thus could not perform the desired control. More information on the initial project can be found in [8].

To facilitate the dissemination of the SPENDULAP’s educational potential, the original design was modified [9] to allow for unattended Internet operation, affording a user anywhere in the world the same options and results as an experimenter in
the SPENDULAP laboratory. Over the next few years, several undergraduate and graduate students redesigned the SPENDULAP to account for the flaws in the prototype and the requirements for Internet compatibility. These design improvements included the installation of an electromechanical initial-positioning mechanism to replace the solenoid and a powerful direct drive brushless DC motor (DDBLDCM) to replace the impotent geared motor *. The designs for these improvements can be found in [9, 10, 11].

By 2001, the SPENDULAP had been reconstructed according to the suggested design enhancements. The next stage in the realization of the fully operational system involved implementation of electronic control circuitry and the custom developed control software that interfaced with the hardware. The telemechatronic system was fully defined in 2002 when the communication architecture was linked to the control software. The overall system architecture is shown in Figure 1.1.

**Figure 1.1** System communication physical architecture

*Analysis of the BLDCM used for the SPENDULAP can be found in Appendix A*
1.4 System Description

The layout of the Rice SPENDULAP is shown in Figure 1.2. The pendulum is mounted in a rotating frame that is supported at the top and bottom by bearings mounted in a cylindrical housing. The pendulum consists of an aluminum tube that is threaded on both ends and attached to a cylindrical aluminum bob at the bottom. At the top, the pendulum is attached to a tee that swings on a stainless steel swivel pin. The swivel pin is mounted in bearings to the rotating frame.

The housing consists of an acrylic cylinder, which also serves as a safety shield. The top of the cylinder is an acrylic disk to which the top frame bearing is attached. The frame is supported at the bottom by an aluminum base that encloses the motor and drive mechanism.

The rotating frame is constructed of rectangular cross-section aluminum bar. There is a bearing shaft mounted to the top of the frame and a drive shaft mounted to the bottom of the frame. The drive shaft is mounted directly on the drive motor shaft.

An optical encoder driven by the swivel pin is mounted outside the frame. The encoder has an accuracy of 5000 cts/rev. Inside the frame, adjacent to the encoder, is an electromagnetic brake. The brake, when energized, prevents the pendulum from swinging while the system is rotating.

An electromagnetic clutch is mounted inside the frame opposite the brake via a custom housing. The clutch, when activated, connects the swivel pin to a shaft which is fitted with a sprocket. Figure 1.3 shows the relation of the encoder, brake, and
Figure 1.2  Rice SPENDULAP layout

clutch to the swivel pin.

Figure 1.3  Encoder, clutch, and brake mounted on the pendulum swivel pin

At the bottom of the rotating frame, a small geared DC motor (the lift motor) is fitted with another sprocket which in turn drives the upper sprocket via a cogged belt. The subsystem consisting of the lift motor, the cogged belt and sprockets, the
clutch, and the brake form the autonomous initial positioning mechanism. This mechanism allows the pendulum to be locked in any desired angular orientation while the frame is stationary, and to remain in that orientation once frame rotation begins, until the brake is released. With both the clutch and brake released, the pendulum swings freely. Figure 1.4 shows the relation of the belt, gears, lift motor and clutch.

![Figure 1.4 SPENDULAP initial positioning mechanism, consisting of lift motor, gears, pulley, clutch, and brake (brake not shown)](image)

The frame is driven by a DDBLDCM. The motor is encased and bolted to an aluminum frame and is powered by a programmable motor driver. The angular rotation is measured by an internal encoder connected to the motor driver. The motor encoder signal is also available as an output from the motor driver. The electrical components mounted on the frame (i.e., encoder, brake, clutch, lift motor) are connected to an 18-channel slipring mounted directly on the DDBLDCM motor shaft. A detailed photograph of the motor section and slipring is shown in Figure 1.5.

A real-time video camera is mounted on a tripod and is connected to the host PC
via a Universal Serial Bus interface. The camera is used for remote monitoring of the SPENDULAP.

Custom electronic hardware conditions signals from the swivel pin encoder and drives the initial positioning mechanism motor, brake, and clutch. The host PC contains a Real-Time (RT) PXI-6030E LabVIEW data acquisition board\textsuperscript{*} that accepts inputs from the encoders and produces control signals for the motors, brake, and clutch. The signals to and from the system are handled by custom designed electronic controls and an intricate LabVIEW software interface developed by the author.

1.5 Outline of Thesis

The outline of this thesis is as follows: Chapter 1 describes the SPENDULAP and the overall telemechatronic hardware environment. Chapter 2 presents briefly the dynamics of the SPENDULAP and discusses various dynamics characteristics that can be illustrated via web-based experiments. Chapter 3 describes three setpoint control strategies that have been implemented and discusses relevant issues and limitations

\textsuperscript{*}the PXI-6030E board is a 16 bit board that communicates at a maximum rate of 100 KHz
of the current architecture. Chapter 4 presents the web-based architecture in detail and discusses the relevant operational scenarios. Finally, Chapter 5 summarizes the contributions of this thesis and points a finger toward the telemechatronic future.
Chapter 2

Dynamics Formulation and Analysis

An understanding of the dynamics of the telemechatronic system is necessary for both implementation and analysis of the web-based experiments and determination of telerobotic fidelity. This chapter formulates the dynamic equations and properties of the the SPENDULAP and investigates some associated phenomena that are of potential interest to a web-engineer; particular emphasis is placed on the nonlinear behavior of the system.

A brief analysis of the SPENDULAP kinematics is presented in Section 2.1. The SPENDULAP dynamics are subsequently formulated Section 2.2. Dynamic analysis, focused on the nonlinear behavior of the SPENDULAP, namely bifurcation, is presented in Section 2.3.

2.1 Kinematic Analysis

The purpose of kinematic analysis is to study the motion of the system without regard to the forces. Kinematic analysis, prior to experimentation, supplies the web-engineer with a better understanding of experiment behavior.

The relevant kinematic data vectors in any system are position $\vec{r}(t)$, velocity $\vec{v}(t)$, and acceleration $\vec{a}(t)$ of a point, $P$. Several different scenarios can be used to determine the kinematics of a system; scenarios in which the moving reference frame (MRF) can translate and/or rotate w.r.t. the inertial reference frame (IRF). More
information on the kinematics can be found in [1].

For the SPENDULAP, the kinematics can be derived by allowing the MRF to rotate relative to the IRF about the $X$ axis with angular velocity $\dot{\theta}$ and the $Z$ axis with angular velocity $\dot{\phi}$. Figure 2.1 and Table 2.1 visually demonstrates this setup and formulation.

![SPENDULAP kinematic diagram](image)

**Figure 2.1** SPENDULAP kinematic diagram

Referring to Figure 2.1 and Table 2.1 and [1], the SPENDULAP angular velocity is

$$\omega = \dot{\theta} \hat{i} + \dot{\phi}(\sin \theta \hat{j} + \cos \theta \hat{k}).$$

The pendulum angular acceleration is

$$\ddot{\omega} = \frac{d\omega}{dt} = \ddot{\theta} \hat{i} + (\dot{\phi} \cos \theta + \ddot{\phi} \sin \theta) \hat{j} + (-\dot{\phi} \dot{\phi} \sin \theta + \ddot{\phi} \cos \theta) \hat{k}.$$
The resulting velocity and acceleration of point P are

\[
\vec{v} = -\dot{\phi}_{cm} \sin \theta \vec{i} + \dot{\theta} l_{cm} \vec{k}
\]

and

\[
\vec{a} = (-2\dot{\theta} \dot{\phi}_{cm} \cos \theta - \dot{\phi}_{cm} \sin \theta) \vec{i} + (\ddot{\theta} l_{cm} - \dot{\phi}_{cm} \cos \theta \sin \theta) \vec{j} + (\dot{\phi}_{cm}^2 + \dot{\phi}_{cm}^2 \sin^2 \theta) \vec{k}.
\]

where \( l_{cm} \) is the length to the center of mass.

Now that the kinematics of the system are understood, the next step is to explore the system dynamics.
2.2 Dynamics

An inherent understanding of the system dynamics are necessary for two reasons. First, dynamic analysis provides the web-engineer with insight into expected experiment results and consequentially the robustness of the implemented web-architecture. Second, comprehension of the dynamics of a system is required to derive the control algorithms for that system.

Before analysis of the system dynamics can take place, the equations of motion (EOM) must be derived. For the SPENDULAP, the EOM are readily obtained using either the Newtonian Method (NM) or the Lagrangian Method (LM).

The NM provides the user with an understanding of the reaction forces acting on the system. Detailed analysis of the NM for the SPENDULAP can be found in [1]. The LM, however, is the simpler of the two methods for deriving the SPENDULAP EOM.

The Lagrangian is the difference in kinetic energy and potential energy, namely,

\[ L = T - V, \tag{2.1} \]

where \( T \) is the developed kinetic energy of the system and \( V \) is the potential energy.

The work due to nonconservative forces during a small displacement in the direction of a particular generalized coordinate \( q_k \) is defined as

\[ \delta W = Q_k \delta q_k, \tag{2.2} \]
where
\[
Q_k = \frac{d}{dt}\frac{\partial L}{\partial \dot{q}_k} - \frac{\partial L}{\partial q_k}.
\] (2.3)

The overall kinetic energy of the SPENDULAP is
\[
T = T_F + T_P,
\]
where
\[
T_F = \frac{1}{2}\dot{\phi}^2 I_{Fz}
\]
and
\[
T_P = \frac{1}{2} [I_x \dot{\theta}^2 + I_y \dot{\phi}^2 \sin^2 \theta].
\]

The overall potential energy of the SPENDULAP is
\[
V = V_F + V_P,
\]
where
\[
V_F = 0
\]
and
\[
V_P = \left( \frac{l_{rod}m_{rod}}{2} + l_{bob}m_{bob} \right) g (1 - \cos \theta).
\]

The coordinates \( x, y, \) and \( z \) are illustrated in Figure 2.2. Defining the vector
\[
\vec{q}_k = \begin{bmatrix} \theta \\ \phi \end{bmatrix},
\]
Equation (2.1) becomes
\[
L = \frac{1}{2} [\dot{\phi}^2 (I_{Fz} + I_z \sin^2 \theta) + I_x \dot{\theta}^2] - \left( \frac{l_{rod}m_{rod}}{2} - l_{bob}m_{bob} \right) g (1 - \cos \theta).
\] (2.4)
Note that $I_x$ has replaced $I_y$ due to the symmetry of the pendulum. Ignoring friction, the only nonconservative force in the system is the input torque $\tau$ due to the drive motor. Thus, Equation (2.2) becomes

$$\delta W = \tau \delta \phi.$$ \hspace{1cm} (2.5)

Applying Equation (2.3) and simplifying into convenient matrix form produces the following result

$$\begin{bmatrix} I_x & 0 \\ 0 & I_F + I_x \sin^2 \theta \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} \frac{I_x \omega_{\text{max}}}{2} + I_{\text{bob}} \omega_{\text{bob}} \sin \theta - I_x \dot{\phi}^2 \sin \theta \cos \theta \\ 2I_x \dot{\phi} \dot{\phi} \sin \theta \cos \theta \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tau. \hspace{1cm} (2.6)$$

![Figure 2.2 SPENDULAP notational diagram](image)

Figure 2.2 SPENDULAP notational diagram
2.2.1 Ignorable Coordinates

The SPENDULAP was chosen as the representative mechatronic system because of its interesting dynamic properties. The first of these, readily associated with the previous dynamic formulation, is the fact that the SPENDULAP dynamics have an ignorable coordinate. That is, in the formulation of Equation (2.6) the state variable $\phi$ is not evident. Since the term $\phi$ does not appear in Equation (2.4), Equation (2.3) w.r.t. $\phi$ is

$$\frac{\partial L}{\partial \phi} = 0;$$

hence, $\phi$ is an ignorable coordinate. Thus, the Lagrangian is invariant to rotation by the coordinate $\phi$ [12]. The significance of the invariance of $\phi$ is that the $\dot{\phi}$ term, in the absence of external forces, can be expressed solely in terms of the variable $\theta$. That is, if there are no external forces on the system (i.e., the right hand side of Equation (2.3) is 0) the conjugate momentum is conserved [13]; i.e.,

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\phi}} = \frac{\partial L}{\partial \phi} = 0. \quad (2.7)$$

Integrating Equation (2.7) w.r.t. $t$,

$$\frac{\partial L}{\partial \phi} = \beta. \quad (2.8)$$

Applying Equation (2.8) to Equation (2.4) leads to the conclusion that

$$\dot{\phi}(I_{Fz} + I_x \sin^2 \theta) = \beta. \quad (2.9)$$
Thus,

$$\dot{\phi}(0)(I_{Fz} + I_x \sin^2 \theta(0)) = \beta$$

Rearranging Equation (2.9), the relationship between $\dot{\phi}$ and $\theta$ is derived.

$$\dot{\phi} = \frac{\beta}{I_{Fz} + I_x \sin^2 \theta} = \dot{\Omega}(\theta)$$

Assuming there are no external forces acting on the system, Equation (2.6) could be rewritten as

$$\begin{bmatrix} I_z & 0 \\ 0 & I_{Fz} + I_x \sin^2 \theta \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \dot{\Omega} \end{bmatrix} + \frac{(I_{tot} \omega \cos \phi + I_{bob} \omega \phi) \sin \theta - I_z (I_{Fz} + I_x \sin^2 \theta)^2 \sin \theta \cos \theta}{2 I_x \dot{\theta} \Omega \cos \theta} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (2.10)$$

Equation (2.9) indicates that $\dot{\phi} = \dot{\phi}(\theta)$. Thus, as apparent in Equation (2.10), the $\phi$-dynamics of the system can be reduced to a first order equation and handled accordingly. Analysis in the following section could be performed using the above formulation, but to remain consistent, the terminology of Equation (2.6) is employed.

### 2.3 Dynamics Analysis

Most mechatronic systems will have interesting dynamic properties that must be fleshed out so that the web-engineer can understand dynamic events in the system experiment. This section explores two interesting properties of the SPENDULAP, equilibrium behavior and bifurcation. These two cases are studied because in the current system setup, a constant velocity input, $\Omega$, causes the system to oscillate
about an equilibrium point and based on the system parameters, bifurcation could result; evidence of both scenarios can be seen in web-based experiments.

2.3.1 Equilibrium Behavior

The equilibrium behavior of the SPENDULAP is important to study for two reasons. First, for a constant velocity input, the SPENDULAP oscillates around and slowly approaches an equilibrium corresponding solely to the system speed, demonstrating the underactuation evident in Equation (2.6) and the friction dynamics of the system. Second, the control discussed in Chapter 3 relies on driving the system toward a setpoint based on the equilibrium.

Examining Equation (2.6) it is apparent that there are some quite complex equilibria for the SPENDULAP. To simplify the situation, only the special case when $\dot{\phi}$ equals a constant value of $\Omega$ is considered. In this case, equilibrium is achieved by setting $\ddot{\theta}$ and $\dot{\theta} = 0$. Equation (2.6) becomes

$$\begin{bmatrix}
\left(\frac{I_{x_{\text{mcut}}} + l_{\text{bob}} m_{\text{bob}}}{2}\right) g \sin \theta - I_x \phi^2 \sin \theta \cos \theta \\
0
\end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tau \quad (2.11)
$$

Solving for $\theta$ in Equation (2.11), the equilibrium is defined as

$$\theta_{eq} = \pm \cos^{-1}\left(\frac{\left(I_{x_{\text{mcut}}} + l_{\text{bob}} m_{\text{bob}}\right) g}{\Omega^2 I_x}\right). \quad (2.12)$$

A plot of the equilibrium dependence on $\Omega$ is shown in Figure 2.3. Only the positive equilibrium values are shown. The graph demonstrates the critical value above which a non-zero equilibrium is possible (approximately $60^\circ$). This critical
value is important in determining the point of bifurcation, discussed in the Section 2.3.

![Graph of Positive Equilibrium Angles vs. Rotational Speed](image)

**Figure 2.3** Equilibrium pendulum deflection

The equilibria defined by Equation (2.12) can be analyzed by linearizing the EOM. The behavior of the SPENDULAP can be approximated in some small region in the vicinity of the equilibrium with a linear model.

Reexamining Equation (2.6) and letting \( \dot{\phi} = \Omega \), define a state vector \( \vec{q} \) that origi-
nates at the equilibrium

\[
\tilde{q} = \begin{bmatrix}
q_1 \\
q_2 \\
q_3
\end{bmatrix} = \begin{bmatrix}
\theta - \theta_{eq} \\
\dot{\theta} \\
\dot{\phi} - \Omega
\end{bmatrix}.
\]

With no external torque, the \( q \) dynamics are

\[
\ddot{q} = \begin{bmatrix}
q_2 \\
-\frac{I_x}{I_x} \left( \frac{I_{\text{rod mass}}}{2} + I_{\text{bob mass}} \right) g \sin(q_1 + \theta_{eq}) + I_x (\sin(q_1 + \theta_{eq}) + \cos(q_1 + \theta_{eq}) + \Omega) \right) + I_x \left( q_2 + \Omega \right) \sin(q_1 + \theta_{eq}) \cos(q_1 + \theta_{eq}) \\
-2I_x q_2 (q_3 + \Omega) \sin(q_1 + \theta_{eq}) \cos(q_1 + \theta_{eq}) \\
I_x + \sin^2(q_1 + \theta_{eq})
\end{bmatrix}.
\]

(2.13)

Thus, the linearized system is \( \tilde{q} = \tilde{A} \tilde{q} \), where

\[
\tilde{A} = \begin{bmatrix}
\frac{\partial q_1}{\partial q_1} & \frac{\partial q_1}{\partial q_2} & \frac{\partial q_1}{\partial q_3} \\
\frac{\partial q_2}{\partial q_1} & \frac{\partial q_2}{\partial q_2} & \frac{\partial q_2}{\partial q_3} \\
\frac{\partial q_3}{\partial q_1} & \frac{\partial q_3}{\partial q_2} & \frac{\partial q_3}{\partial q_3}
\end{bmatrix}
\]

\[
\begin{array}{c}
q_1 = 0 \\
q_2 = 0 \\
q_3 = 0
\end{array}
\]

(2.14)

From Equations (2.13) and (2.14)

\[
\tilde{A} = \begin{bmatrix}
0 & 1 & 0 \\
A_{21} & 0 & A_{23} \\
0 & A_{32} & 0
\end{bmatrix}
\]

where

\[
A_{21} = -\frac{\left( \frac{I_{\text{rod mass}}}{2} + I_{\text{bob mass}} \right) g \sin^2 \theta_{eq}}{I_x \cos \theta_{eq}},
\]

\[
A_{23} = 2 \sin \theta_{eq} \cos \theta_{eq} \sqrt{\frac{\left( \frac{I_{\text{rod mass}}}{2} + I_{\text{bob mass}} \right) g}{I_x \cos \theta_{eq}}},
\]
\[ A_{32} = \frac{2I_z \sin \theta_{eq} \cos \theta_{eq} \sqrt{\frac{I_{bob} m_{bob}}{2} + I_{bob} m_{bob}} g}{I_x + I_z \sin^2 \theta_{eq}}, \]

for example, if \( \theta_{eq} = 30 \), the \( \tilde{A} \) matrix becomes

\[
\tilde{A} = \begin{bmatrix}
0 & 1 & 0 \\
-11.692 & 0 & 5.9225 \\
0 & -3.2642 & 0
\end{bmatrix}
\]

The eigenvalues, \( \lambda \) for \( (\theta, \dot{\theta}) \) are \( \pm 5.5019i \). This corresponds to a frequency of oscillation of approximately 5.5 rad/sec in the neighborhood of the equilibrium.

Since \( Re(\lambda) = 0 \) the system stability cannot be determined. Adding viscous friction to the EOM, Equation (2.5) becomes

\[ \delta W = \tau \delta \phi + C \delta \dot{\phi} \] (2.15)

and Equation (2.6) becomes

\[
\begin{bmatrix}
I_x & 0 \\
0 & I_{Fz} + I_z \sin^2 \theta
\end{bmatrix}
\begin{bmatrix}
\dot{\phi} \\
\dot{\theta}
\end{bmatrix}
+ \begin{bmatrix}
\frac{I_{bob} m_{bob}}{2} g \sin \theta - I_z \phi^2 \sin \theta \cos \theta + C \dot{\theta} \\
2I_z \phi \sin \theta \cos \theta
\end{bmatrix} \begin{bmatrix}
0 \\
1
\end{bmatrix} \tau,
\]

where \( c \) is the viscous damping term to be determined experimentally. Equation (2.13) becomes

\[
\tilde{q} = \begin{bmatrix}
q_2 \\
-\frac{I_{bob} m_{bob}}{2} g \sin(q_1 + \theta_{eq}) + I_z (q_3 + \Omega)^2 \sin(q_1 + \theta_{eq}) \cos(q_1 + \theta_{eq}) \\
-I_z \phi^2 \sin(q_1 + \theta_{eq}) \\
-2I_z q_2 (q_3 + \Omega) \sin(q_1 + \theta_{eq}) \cos(q_1 + \theta_{eq})
\end{bmatrix}
\]

and the Jacobian matrix \( \tilde{A} \) is

\[
\tilde{A} = \begin{bmatrix}
0 & 1 & 0 \\
\frac{-C}{I_z} & 0 & 0 \\
A_{21} & \frac{-C}{I_z} & A_{23} \\
0 & A_{32} & 0
\end{bmatrix}
\]
As an example, take a low value of damping, say $C = 0.001$, and setting $\theta_{eq} = 30$, the resulting eigenvalues $\lambda$ for $(\theta, \dot{\theta})$ are $(-0.0592 \pm 5.5016i)$. It is apparent that the small amount of friction does not greatly affect the frequency of oscillation (compare 5.5016 versus 5.5019) but it does put the poles of the system in the left-half plane. Therefore, accounting for viscous friction the system is stable and will approach one of the equilibria defined by Equation (2.12). That is, $\lim_{t \to \infty} \theta = \theta_{eq}$, if the model accounts for viscous damping. As can be seen in Appendix B, the viscous damping model is a suitable model for the SPENDULAP. Web-experiments can be used to confirm the results.

2.3.2 Bifurcation

Bifurcation is important to study in the SPENDULAP because of the possibility of multiple equilibria defined by Equation (2.12). In a constant rotation experiment the web-engineer can readily see the bifurcation phenomenon by manipulating certain conditions in the system as is seen in the following analysis.

Examining Equation (2.6) it is obvious that the EOM of the SPENDUALAP are nonlinear in both $\theta$ and $\dot{\theta}$. In Equation (2.12), the equilibrium around which the system oscillates and settles to is either a positive angle, $\theta_1$, defined inherently by the rotational speed $\Omega$ of the system, or is the negative angle, $-\theta_1$, defined by the same rotational speed, or $\theta = 0$. The possibility of multiple equilibria, dependent on the same value, namely $\Omega$, indicates the potential for the system dynamics to bifurcate [13, 14, 15].
The simplest method to determine if a system bifurcates is to look at the behavior of the attractors. An attractor is defined as an asymptotically stable non-wandering set, such as a limit cycle, or a fixed point. If the number of attractors in a system changes when a system parameter is manipulated, the system exhibits bifurcation. The bifurcation point is the value of the parameter where the number of attractors changes, coincident with a change in stability [16]. In the case of the SPENDULAP, one such parameter is $\Omega$ and the bifurcation point is $\Omega_c$, defined by Equation (2.12).

Evaluating for $\theta_{eq} = 0$,

$$
\Omega^2_c = \frac{\left(\frac{I_{ext, fixed}}{2} + I_{bob,m_{bob}}\right)}{I_x} g
$$

The SPENDULAP exhibits classical co-dimension-one bifurcation, and based on stability analysis, the specific type of bifurcation (e.g., pitchfork, Hopf), can be determined.

In Equation (2.6), if $\dot{\phi}$ is held constant at some $\Omega$, the angular momentum of the system is conserved and thus Equation (2.6) may be reduced to a single degree of freedom nonlinear oscillator of the form [17]

$$
\ddot{\theta} + \Omega_c^2 \sin \theta + \Omega^2 \sin \theta \cos \theta = -\frac{\nu \dot{\theta}}{I_x},
$$

where $\nu = C/I_x$ is the viscous damping term. In state space form,

$$
\ddot{x} = \tilde{f}(\vec{x}, \Omega^2) = \begin{bmatrix}
x_2 \\
\Omega^2 \sin x_1 \cos x_1 - \Omega_c^2 \sin x_1 - \nu x_2
\end{bmatrix},
$$

where $\vec{x} = [x_1, x_2]^T$. Defining $\vec{x}$ such that $\tilde{f}(\vec{x}, \Omega^2) = 0$ such that

$$
0 = \vec{x}_2 = \Omega^2 \sin x_1 \cos x_1 - \Omega_c^2 \sin x_1 - \nu x_2.
$$
When $\Omega^2 < \Omega_c^2$ the system reduces to a simple pendulum with equilibria at $\bar{x} = (0,0)$ and $\bar{x} = (\pi,0)$. If $\Omega^2 > \Omega_c^2$ the equilibrium $\bar{x} = (0,0)$ splits in a pitchfork pattern. The new equilibria of the system are at $\bar{x} = (0,0)$, or $\bar{x} = (\pm \arccos(\frac{\Omega_c^2}{\Omega^2}),0)$.

The system can be linearized and the Jacobian analyzed to determine stability. The Jacobian is

\[
J = \begin{bmatrix}
0 & 1 \\
\Omega^2(\cos^2 \bar{x}_1 - \sin^2 \bar{x}_1) - \Omega_c^2 \cos \bar{x}_1 & -\nu
\end{bmatrix}
\] 

(2.16)

For $\bar{x} = (\pi,0)$, the determinant of Equation (2.16) is negative; i.e., this is an unstable equilibrium. The $\bar{x} = (0,0)$ is a stable focus or node (dependent on amount of damping) when $\Omega^2 < \Omega_c^2$ but is an unstable saddle if $\Omega^2 > \Omega_c^2$. The $\bar{x} = (\pm \arccos(\frac{\Omega_c^2}{\Omega^2}),0)$ equilibria are both stable nodes or focuses (dependent on amount of damping) and exist only when $\Omega^2 > \Omega_c^2$.

As seen by the above analysis, as $\Omega$ increases past the critical speed $\Omega_c$, two events occur. First, the number of attractors changes from one to two. Second, the system, and the stability of the original attractor at $\bar{x} = (0,0)$ becomes unstable as two new attractors branch out as the stable equilibria. These two combined events define a supercritical pitchfork bifurcation [16, 18], graphically shown in Figure 2.4 and evident in constant rotation web-experiments.
Figure 2.4  Supercritical pitchfork bifurcation plot
Chapter 3
Control Algorithms and Issues

Most web-engineers are not going to be satisfied by just monitoring the dynamics of a system. They will want to control the system dynamics. The purpose of this chapter is to discuss the control algorithms available to the web-engineer for the SPENDULAP and to evaluate their ability to drive the system to an arbitrary equilibrium angle, \( \theta \). Currently, three different set-point control algorithms, selected from [1, 5, 6, 19], have been implemented to demonstrate the variety and limitations of controls for the current system setup. These algorithms are formulated and discussed in:

1. Section 3.1 - Pole Placement
2. Section 3.2 - Input State Linearization
3. Section 3.3 - Sliding Mode
   (a) Section 3.3.1 - Continuous Sliding Mode

Each section details the formulation of the respective algorithms and present graphical analysis comparing experiments to simulations.

3.1 Pole Placement

Pole placement is a commonly used method to design the feedback gains in a full state feedback system. The pole placement method allows the user to define the rate
of decay toward the steady state in a linear system. State feedback control utilizing pole placement is also one of the simplest closed-loop control algorithms and is thus a good introduction to feedback control.

The pole placement algorithm is only suitable for a linear time invariant (LTI) system. Since the SPENDULAP state equations contain trigonometric parameters, the system must be linearized. To effectively remove the nonlinearities from the system, the equations of motion must be linearized about an equilibrium; i.e., \( \vec{q} = \vec{0} \).

The corresponding equilibrium torque is \( \tau_{eq} = 0 \).

Define \( \vec{q} \) with its origin at the setpoint, or

\[
\vec{q} = \begin{bmatrix}
q_1 \\
q_2 \\
q_3
\end{bmatrix} = \begin{bmatrix}
\theta - \theta_{eq} \\
\dot{\theta} \\
\dot{\phi} - \Omega
\end{bmatrix}.
\]

Restating Equation (2.6) in terms of \( \vec{q} \),

\[
f = \vec{\dot{q}} = \begin{bmatrix}
q_2 \\
-\left(\frac{l_{bob} m_{bob}}{g} \sin(q_1 + \theta_{eq}) + I_x(q_3 + \Omega)^2 \sin(q_1 + \theta_{eq}) \cos(q_1 + \theta_{eq})\right) \\
\frac{\tau - 2I_x q_2(q_3 + \Omega) \sin(q_1 + \theta_{eq}) \cos(q_1 + \theta_{eq})}{I_F + I_x \sin^2(q_1 + \theta_{eq})}
\end{bmatrix}.
\]

The advantage of the vector \( \vec{q} \) is that the control problem reduces to designing a control algorithm such that

\[
\lim_{t \to \infty} \vec{q} = \vec{0}.
\]

Thus, if the system is perturbed from the equilibrium by some \( \delta \vec{q} \), the control
algorithm forces it back toward the equilibrium. To move the system back toward the equilibrium, some \( \delta \tau \) must also be developed. Arranging the state vector in terms of \( \delta \dot{q} \) (since \( \ddot{q} = q_{eq} + \delta \dot{q} = \delta \dot{q} \)), we get

\[
\delta \dot{q} = \bar{A} \delta q + \bar{B} \delta \tau + H.O.T.
\]

where

\[
\bar{A} = \begin{bmatrix}
\frac{\partial f_1}{\partial q_1} & \frac{\partial f_1}{\partial q_2} & \frac{\partial f_1}{\partial q_3} \\
\frac{\partial f_2}{\partial q_1} & \frac{\partial f_2}{\partial q_2} & \frac{\partial f_2}{\partial q_3} \\
\frac{\partial f_3}{\partial q_1} & \frac{\partial f_3}{\partial q_2} & \frac{\partial f_3}{\partial q_3}
\end{bmatrix}
\]

and

\[
\bar{B} = \begin{bmatrix}
\frac{\partial f_1}{\partial u_1} \\
\frac{\partial f_2}{\partial u_1} \\
\frac{\partial f_3}{\partial u_1}
\end{bmatrix}
\]

Both \( \bar{A} \) and \( \bar{B} \) are evaluated around \( \ddot{q} = q_{eq} = 0 \) and \( u \) represents the input torque. Assuming the H.O.T. (higher order terms) are small, the system is now linear and the pole placement algorithm can be applied.

Evaluating the partial derivatives in \( \bar{A} \) and \( \bar{B} \),

\[
\bar{A} = \begin{bmatrix}
0 & 1 & 0 \\
A_{21} & 0 & A_{23} \\
0 & A_{32} & 0
\end{bmatrix}
\]

where

\[
A_{21} = \frac{-(l_{\text{rad}} m_{\text{rad}} + l_{bob} m_{bob}) \ g \sin^2 \theta_{eq}}{I_x \cos \theta_{eq}},
\]
\[ A_{23} = 2 \sin \theta_{eq} \cos \theta_{eq} \sqrt{\frac{\left(I_{bob} + m_{bob} \frac{I_{m,rad}}{2}\right)g}{I_x \cos \theta_{eq}}}, \]
\[ A_{32} = \frac{2I_x \sin \theta_{eq} \cos \theta_{eq} \sqrt{\frac{\left(I_{bob} + m_{bob} \frac{I_{m,rad}}{2}\right)g}{I_x \cos \theta_{eq}}}}{I_{Fz} + I_x \sin^2 \theta_{eq}}, \]

and
\[
\tilde{B} = \begin{bmatrix}
0 \\
0 \\
\frac{1}{I_{Fz} + I_x \sin^2 \theta_{eq}}
\end{bmatrix}.
\]

Using the control law \( \delta \tau = -\tilde{K} \delta \tilde{q} \) ensures that the closed loop system corresponding to \( \delta \tilde{q} = [\tilde{A} - \tilde{B} \tilde{K}] \delta \tilde{q} \) will have the prescribed poles as defined by the gain matrix \( \tilde{K} \). The mathematical procedure for deriving \( \tilde{K} \) can be found in [20].

Figure 3.1 demonstrates the ability of the pole placement method to drive the system toward the desired equilibrium. The figure demonstrate the ability to drive the system from an arbitrary initial angle to user-defined desired angle smoothly and quickly.

### 3.2 Input State Linearization

The SPENDULAP can be controlled without linearizing the equations of motion but still with simplicity by employing the technique known as input state linearization discussed in [21]. This is a more complicated algorithm than pole placement, but is a nice simple nonlinear algorithm that can be implemented in many mechatronic systems.
Figure 3.1  Pole Placement: $\theta_{\text{init}} = 85^\circ, \theta_{\text{des}} = 30^\circ, Poles = [-4,-5,-9]$

To simplify the derivation, the vector $\tilde{q}$ is redefined such that

$$
\tilde{q} = \begin{bmatrix}
\theta \\
\dot{\theta} \\
\dot{\phi}
\end{bmatrix}
$$

Thus,

$$
\ddot{q} = \begin{bmatrix}
q_2 \\
\sin q_1 u \cos q_1 - (\frac{l_{\text{free}}}{2} + l_{\text{bob}} m_{\text{bob}} \phi) g \sin q_1 / I_z \\
\dot{q}_3
\end{bmatrix}
$$

(3.1)

where $u = \dot{\phi}^2$. 
As implied by the name, the input state must be linearized. Therefore, define auxiliary control \( u \) such that

\[
u = \frac{(l_{\text{end}}m_{\text{end}} + l_{\text{bob}}m_{\text{bob}}) g \sin q_1 I_x + v}{\sin q_1 \cos q_1}.
\]  

(3.2)

As seen in Equations (3.1) and (3.2) the \( \vec{q} \) vector was chosen to simplify the definition of \( u \) and to create a clear error feedback system for \( v \). So now \( \dot{q}_2 = v \).

A simple PD Controller is used to control \( v \) such that

\[
v = K_p(\theta_{\text{desired}} - q_1) + K_dq_2.
\]

Using proportional control in cascade, the input torque is defined as

\[
\tau = \dot{\phi}_{\text{desired}}[I_F + I_z \sin^2(q_1 + \theta_{\text{desired}})] + 2I_zq_3q_3 \sin(q_1 + \theta_{\text{desired}}) \cos(q_1 + \theta_{\text{desired}})
\]

where

\[
\dot{\phi}_{\text{desired}} = \sqrt{u}
\]

and

\[
\ddot{\phi}_{\text{desired}} = K_\phi(\dot{\phi}_{\text{desired}} - \dot{\phi}).
\]

Figure 3.2 demonstrates the input state linearization algorithm's ability to drive the system toward the desired equilibrium. Of the three algorithms investigated, the input state linearization provided the best results for the SPENDULAP.
Figure 3.2  Input State Linearization: $\theta_{\text{init}} = 85^\circ, \theta_{\text{des}} = 30^\circ, K_p = 100, K_d = 20, \phi = 25$

3.3 Sliding Mode

Another common nonlinear control technique employed by the SPENDULAP is sliding mode control which operates by switching control torques around the sliding surface defined by equilibrium parameters. This is the most complicated control algorithm of the three discussed.
The state vector $\vec{q}$ is the same as in the input state linearization case, i.e.,

$$
\vec{q} = \begin{bmatrix} 
\theta \\
\dot{\theta} \\
\dot{\phi} 
\end{bmatrix}.
$$

Thus,

$$
\vec{q} = \begin{bmatrix}
q_2 \\
q_3^2 \sin q_1 \cos q_1 - \left( \frac{l_{rod,mod}}{2} + l_{bob,mod} \right) g \sin q_1 / I_x \\
-2g \dot{q}_3 \sin q_1 \cos q_1 \\
\frac{1}{I_{Fz} + I_x \sin^2 q_1}
\end{bmatrix} + \begin{bmatrix}
0 \\
0 \\
\tau
\end{bmatrix} \tau. \quad (3.3)
$$

The reduced order system is chosen such that $\vec{\eta} = [q_1 \ q_2]^T$ and $\xi = q_3$.

Equation (3.3) becomes

$$
\vec{\eta} = \begin{bmatrix} 
\eta_2 \\
\xi^2 \sin \eta_1 \cos \eta_1 - \left( \frac{l_{rod,mod}}{2} + l_{bob,mod} \right) g \sin \eta_1 / I_x 
\end{bmatrix}
$$

and

$$
\dot{\xi} = f(\vec{\eta}, \xi) + G(\vec{\eta}, \xi) \tau
$$

where

$$
f(\vec{\eta}, \xi) = f = \frac{-2I_x \eta_2 \xi \sin \eta_1 \cos \eta_1}{I_{Fz} + I_x \sin^2 \eta_1}
$$

and

$$
G(\vec{\eta}, \xi) = G = \frac{1}{I_{Fz} + I_x \sin^2 \eta_1}.
$$
Utilizing the previously discussed input state linearization technique, the function $\gamma(\eta)$, similar to $\phi_{desired}$ in the previous section, is formulated as

$$
\gamma(\eta) = \sqrt{\left(\frac{l_{load}m_{load} + l_{bob}m_{bob}}{2}g \sin \eta_1/I_x + K_p(\theta_{desired} - \eta_1) - K_d \eta_2\right) \over \sin \eta_1 \cos \eta_1}.
$$

The sliding surface $z$ is defined as the difference between the actual velocity and the desired velocity, or

$$z = \xi - \gamma(\eta).$$

The input torque provided by the BLDCM is given by the equation

$$\tau = \tau_{eq} + G^{-1}\nu$$

where

$$\tau_{eq} = G^{-1}(-f + \partial \gamma(\eta) / \partial \eta)$$

and $\nu$, the subsidiary control, is defined as

$$\nu = -K_p \text{sgn}(z).$$

Although sliding mode control, in theory, seems like a good control for any system due to its switching parameters, it does not provide the desired control for the SPENDULAP due to the system parameters. An experimental run is shown in Figure 3.3.

Figure 3.3 shows the ineffectiveness of the discrete sliding mode algorithm. It is important to note that sliding mode experiments were run with a sampling rate of
Figure 3.3  Discrete Sliding Mode: $\theta_{\text{init}} = 85^\circ, \theta_{\text{des}} = 30^\circ, K_p = 100, K_d = 20, K_b = 200$ approximately 125 Hz. If the sampling rate could be increased, the chatter seen in Figure 3.3 would vanish. However, with the maximum sampling rate of 125 Hz no set of gains could be found that produced the desired smooth curve seen in the input state experiment.

Faced with this problem, there were two possible solutions. One, was to increase the sampling rate. This involved acquiring new hardware and was deemed infeasible. The second solution is discussed next.
3.3.1 Continuous Sliding Mode

The above discussed methodology is based on discrete sliding mode (DSM) control. Continuous sliding mode (CSM) [19] was examined and compared against the discrete cases. In CSM the control equations formulation is the same except the $\nu$ term is no longer defined by the signum function; i.e., the sliding mode surface does not switch instantly. Instead,

$$\nu = -K_v \text{sat}(z)$$

where sat is the saturation function. Graphically, the difference can be seen in Figure 3.4, where $H$ is the slope of the line. As $H \to \infty$, the continuous sliding mode approaches the discrete case.

![Diagram of sgn function and sat function](image)

**Figure 3.4** Discrete and continuous sliding mode graphical definition

Figure 3.5 compares the continuous versus discrete cases. As expected, as the slope increases, the CSM results approach the DSM results. The acceptable range for
smooth operation was experimentally found to be when $0.15 < H < 1.05$. Figure 3.5 demonstrates an acceptable slope, $H$, for a CSM simulation compared against a DSM simulation.

![Graphs showing angular velocity and control torque](image)

**Figure 3.5** Comparison of DSM versus CSM: $H = 0.7$

Figure 3.6 demonstrates an experimental test case for the continuous sliding mode. $H$ was set to 0.2. As evident from the figure, the CSM case approaches the desired equilibrium more smoothly and quickly than the DSM experiment shown in Figure 3.3.

### 3.4 Summary

The pole placement and input state linearization algorithms work well in the current SPENDULAP framework, driving the system to the desired equilibrium. Though
Figure 3.6 Continuous Sliding Mode: $\theta_{init} = 85^\circ, \theta_{des} = 30^\circ, K_p = 100, K_d = 20, K_b = 200, H = 0.2$

the discrete sliding mode drives the system toward the equilibrium, it chatters excessively and occasionally will lose control due to the low sampling time available to the SPENDULAP. The continuous sliding mode algorithm circumvents the sampling time problem effectively.

The control algorithms discussed, along with many more, can be effectively used in other telemechatronic systems under the correct conditions (e.g., for the DSM, the sampling rate must be sufficiently high). The algorithms developed for the SPENDULAP were used to demonstrate the potential variety available to a telemechatronic
system and to examine limitations associated with implementing controls into the system (e.g., available sampling time) and possible solutions (e.g., CSM for DSM).
Chapter 4
Web Operational Architecture

The communication architecture of the SPENDULAP is shown in Figure 4.1. This chapter discusses the software and hardware architecture that allows the system to be controlled from a standard Internet browser without any external clients or downloadable software as often required by telemechatronic systems [3, 4, 22, 23, 24].

![Diagram of SPENDULAP telemechatronic communication](image)

**Figure 4.1** SPENDULAP telemechatronic communication diagram

Section 4.1 discusses the basic operational behavior the SPENDULAP performs during a desired experiment. There are two operational scenarios (OS). The first OS allows the web-engineer to drive the SPENDULAP at a constant velocity and thus explores the various dynamics of the constant velocity system discussed in Chapter 2. The second OS allows the web-engineer to implement one of the three control algorithms discussed in Chapter 3 and test its validity and robustness.
The web-architecture that allows the system to be controlled over the Internet is discussed in Section 4.2 with emphasis on the communication flow between the Internet and the SPENDULAP system. Included in this section are detailed explanations of the directional communication of the system and layered software architecture. Hardware and software interaction, dependent upon whether the web-engineer is commanding the experiment or just spectating via the web, is included. The end of the section presents an example of a controlled experiment run over the Internet.

Finally, Section 4.3 presents a brief discussion of the primary programming language used for software implementation, LabVIEW. The LabVIEW software threads the entire system together from the physical SPENDULAP interface to the Java web interface.

4.1 Operational Scenarios

There are two possible operational scenarios, the sequence of steps that the software causes the system to perform. There are two classes of experiments: constant frame angular velocity and feedback control.

For the constant frame angular velocity case, the user specifies a frame angular velocity and initial pendulum angular deflection about the swivel pin. For the feedback control case the user chooses from a set of control algorithms, specifies the initial pendulum deflection angle and setpoint deflection angle, and a set of gains corresponding to the control algorithm. For all cases, the user also specifies the experiment's duration.
For both operational scenarios, the initialization sequence is as follows:

- Release clutch and brake.
- Verify that the pendulum is not swinging.
- Initialize encoder.
- Engage clutch.
- Raise pendulum to specified deflection angle.
- Lock brake.
- Release clutch.

At this point in the sequence, the frame angular velocity is zero and the pendulum is locked at the user-specified initial deflection angle.

4.1.1 Constant Frame Angular Velocity Experiments

The simplest SPENDULAP experiment entails rotating the frame at a constant angular velocity and observing the pendulum deflection trajectory based on the analysis provided in Chapter 2.

Based on the equations from the Chapter 2 there is a constant frame angular velocity associated with any $\theta_{eq}$ between $-90^\circ$ and $90^\circ$. If, while the frame is rotating at a constant angular velocity $\Omega$, the pendulum is released from an initial deflection of $\theta_{eq}$, the pendulum will remain deflected at $\theta_{eq}$. If the initial pendulum angular
deflection is different from $\theta_{eq}$, the pendulum will oscillate around the equilibrium angle.

The constant frame angular velocity case provides an excellent opportunity to apply a number of nonlinear analysis and modelling techniques and to examine such phenomena as bifurcation. More information on this can be found in [25].

The operational sequence for the constant angular velocity experiment is as follows:

- Execute the initialization sequence.
- Initiate frame rotation at the specified angular velocity.
- Release brake.
- The pendulum oscillates in a pattern determined by the frame angular velocity and initial pendulum deflection angle. The oscillations slowly decay due to friction in the swivel pin bearings.
- Terminate frame rotation at the expiration of the experiment time.

4.1.2 Feedback Control Experiments

The SPENDULAP is also ideal to test the feedback control experiments examined in Chapter 3. The setpoint control experiments implemented in the web user interface allow the use of both linear and nonlinear control laws. In the setpoint control experiments, the goal is to adjust frame drive motor torque so as to cause the pendulum to rapidly stabilize at a preselected $\theta_{eq}$. The experiment begins with the
system at rest and the pendulum deflected at some angle different from $\theta_{eq}$. Then, the pendulum brake is released and simultaneously feedback control of drive motor torque is initiated. The experiment ends after a user-specified time interval.

The operational sequence for the feedback control experiments is as follows:

- Execute the initialization sequence.
- Release brake.
- Activate the feedback control law. The controller computes the motor torque according to the selected control law, causing the pendulum to stabilize at the specified $\theta_{eq}$.
- Terminate frame rotation at the expiration of the experiment time.

4.2 Web System Architecture

The software architecture consists of four main elements: a Java client running on the user's web browser, a Java host program running on the lab computer, a LabVIEW host interfacing with the Java host, and a LabVIEW control program running on the LabVIEW Real Time (RT) Board. The Java client applet contains a real-time video window, a user input window to select from a variety of control laws and set controller parameters, and an output window that displays experimental results. The host application manages the user session and serves as a trusted intermediary between the client and the LabVIEW application; i.e., the dedicated control software, LabVIEW, runs solely on the host computer and has no direct interface with the
Internet. The LabVIEW host converts the Java input via TCP/IP protocol into a compatible format and sends the data to the RT program. The selected control law is implemented, the experiment is performed, and the results are sent back in reverse order to the user.

The simple two-way data and control flow and software architecture are depicted in Figure 4.2. Referring to Figure 4.1, there are two types of users: Commander and Spectator. The communication and control process for each type of user is discussed next.

Figure 4.2  Data and control flow software architecture
4.2.1 **Web-based Commander Scenario**

The Java host computer program allows one user at a time to serve as commander. Commander rights are allocated to individual users via a control file maintained by the system administrator. The commander communication scenario is as follows:

1. Commander logs on to the web site and presses the Connect button. Commander enters an assigned user name and password and is connected to the Java server running on the SPENDULAP computer. The commander window (See Figure 4.3) is displayed on the commander’s PC.

   (a) Commander chooses experiment from a menu and sets parameter values. Commander presses Start button to activate the experiment. The **Experiment Information** (EI) is transmitted across the web to the Java server running on the **SPENDULAP Host Computer** (SHC).

2. The SHC Java server communicates with the LabVIEW server **Visual Instrument** (VI) via TCP/IP protocol and translates the EI into suitable LabVIEW form.

3. EI is passed from the LabVIEW server to a Host VI. The Host VI validates the data, and if no errors are found, transmits the data to the **LabVIEW RT Program** (LRTP).

4. LRTP receives the data from the Host in a sequential manner and does not activate until all of the EI has been received. This process is ensured by running two nested control loops.
(a) The outer loop receives data from and sends data to the Host VI. The outer loop takes the data from the inner loop and transmits it back to the Host VI.

(b) The inner loop is activated on receipt of all parameters. The inner loop executes the experiment sequence and then resets to an inactive state.

5. Data from the inner loop is physically output by the LabVIEW Data Acquisition Card (DAC)* that sends the desired voltage into separate control circuits for each electrical device. The control circuits translate the voltage into the necessary values (e.g., on/off, currents) to perform the experiment.

6. SPENDULAP begins experiment as defined by the Operational Scenarios.

7. Data from the SPENDULAP encoders are passed back to the LabVIEW RT Engine through the DAC. The LabVIEW RT Board samples the data at \( \approx 100 \) Hz.†

8. Data from the experiment are stored in the inner loop. Once the experiment has finished the inner loop passes this information to the outer loop and the outer loop transmits the data to the Host VI.

9. The host VI receives results and translates them into a form suitable for the Java server. The LabVIEW server transmits results via TCP/IP protocol to the Java server.

*The LabVIEW DAC is connected directly to the LabVIEW RT Board
†The LabVIEW RT Board can sample data up to 100 KHz, but due to the complex algorithms involved for SPENDULAP, sampling occurs at 100 Hz
10. Results are transmitted to the Java server.

11. The Java server sends the result data across the web to the commander’s browser.

12. The commander’s browser receives the data and the results are plotted.

Figure 4.3  Commander’s user interface display

4.2.2 Web-based Spectator Scenario

While there can be only one commander at any time, the SPENDULAP system allows for an unlimited number of spectators. The spectator communication scenario is as follows:
1. Spectator logs on to web site and presses the Connect button. Spectator signs on using the login name: guest. The spectator window (See Figure 4.4) is displayed on the user’s PC.

(a) The experiment is initiated by the commander.

2. Internal communication process among Java server and LabVIEW components is the same as in the commander’s situation.

3. Results are plotted on spectator’s browser. Spectator has control over viewing his own plots.

Figure 4.4 Spectator user interface display
4.2.3 Example

This section presents details of a typical feedback control experiment using input-state linearization. Table 4.1 lists the user inputs.

<table>
<thead>
<tr>
<th>Control type</th>
<th>input state linearization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial angle</td>
<td>85°</td>
</tr>
<tr>
<td>Desired angle</td>
<td>30°</td>
</tr>
<tr>
<td>Kp</td>
<td>100</td>
</tr>
<tr>
<td>Kd</td>
<td>20</td>
</tr>
<tr>
<td>Kphi</td>
<td>25</td>
</tr>
<tr>
<td>Time(ms)</td>
<td>5000</td>
</tr>
</tbody>
</table>

**Table 4.1** Experiment data

Figure 4.5 shows the commander’s browser display after all input data are entered. After entering the data, the commander clicks the Start button, and the experimental sequence executes automatically. The video window shows the experiment window in real time. Once the experiment terminates, trajectory parameters ($\theta$, $\dot{\theta}$, $\dot{\phi}$, $\tau$) are transmitted to the commander and to any spectators currently logged-on. The browser display, post-experiment, is shown in Figure 4.6. The commander (and spectators) can also save trajectory data on their local storage for subsequent processing and analysis.

4.3 LabVIEW

The primary programming tool that controls the operation of the SPENDULAP is LabVIEW, a visual programming language that requires the creation of VI’s, or Visual Instruments.
LabVIEW affords the programmer the same abilities as many text-based programs, like C++, Fortran, or MATLAB (and can incorporate these languages into its own architecture). It lends the added advantage of allowing the user to create the program in the background while allowing a secondary user, not familiar with the program architecture, to perform the experiment. This ease of use is due to the fact that LabVIEW allows the programmer to create a front end panel that has simple controls and indicators that require no programming knowledge and little explanation.

The LabVIEW architecture runs solely on the host computer without any direct interface to the Internet. The LabVIEW programs written by the author communi-
Figure 4.6  Commander display after experiment

cate with the Java host to receive control commands and transmit results and thus the web-engineer requires no knowledge of either LabVIEW or the software implementation to perform an experiment.

Figures 4.7 and 4.8 show the front panels of some of the primary LabVIEW programs used by the SPENDULAP. These two figures are the two primary RT programs. Figure 4.7 is the front panel for the outer loop program discussed in Section 4.2.1. Figure 4.8 shows the front panel for the inner loop, the output VI to the SPENDULAP.

The programs shown in Figures 4.7 and 4.8 are only a couple of the many VI’s used
Figure 4.7  Front Side: Real Time LabVIEW VI that controls inputs and outputs of the SPENDULAP

Figure 4.8  Front Side: LabVIEW VI that defines inputs and output torque to SPENDULAP
by the SPENDULAP. All of the VI’s used by the SPENDULAP are listed in Tables C.1 - C.8 in the Appendix C. Corresponding LabVIEW icons and brief descriptions are included.
Chapter 5
Conclusions

The advent of telemechatronics promises a future of remotely operated homes, unattended military devices, resource sharing among scientists and universities, and thousands of other ordinary to extraordinary achievements due to teleoperated mechatronic systems. The advancement toward this future will be facilitated by simplifying the web-architecture employed by current mechatronic systems. This thesis demonstrates research that takes a step toward that future.

The developed communication architecture demonstrates the online capability for control and dynamic analysis of a mechatronic system. The Rice SPENDULAP, used to exhibit this architecture, is one of a few remote-access physical systems available as an engineering resource, and of those, it is the only one known to the author that requires no external clients or software by the web-engineer. This stand-alone quality, achieved by employing a Java-based client GUI, alleviates the configuration management and operational compatibility issues evident in previous systems.

The host-driven architecture allows for reliable real-time control of the system dynamics, thus circumventing the chaotic nature of communication transfer of the internet. The developed architecture also provides a straightforward method for control coordination, allotting control privileges to one web-engineer for a specified time slot while allowing multiple spectators.
The SPENDULAP is a simple example of a mechatronic system that can be controlled by the developed architecture. The SPENDULAP was chosen for its relatively inexpensive cost, interesting dynamic behavior and suitability to demonstrate a variety of control algorithms exposed to physical system constraints (e.g., sampling time). The following operations can be performed by a web-engineer on the SPENDULAP to demonstrate the web-architecture developed by the author:

1. Constant Rotation Experiments

2. Control Experiments
   (a) Pole Placement
   (b) Input State Linearization
   (c) Sliding Mode (Discrete and Continuous)

The constant rotation experiments offer insight into the kinematics and dynamics of the SPENDULAP. The equilibrium behavior of the system and the potential for bifurcation provide a good opportunity to explore the various phenomena associated with a mechatronic system.

The setpoint control experiments show the SPENDULAP’s capacity as a testbed for a wide variety of control algorithms. The pole placement and input state linearizations algorithm provide the best control. The discrete sliding mode model has a small range of control, and within that range, the system experiences severe
chattering due to the finite switching time of the system. Implementing continuous sliding mode control solves this problem.

The SPENDULAP elucidates the several advantages of the next generation telemechatronic system: it provides greater availability than traditional systems, providing twenty-four hour a day experiment capability with continual multiple-spectator access and commander time allotment; it can be remotely operated by anybody with access to a standard internet browser, independent of software or hardware; and it provides greater functionality than the traditional systems due to the straightforward communication architecture, allowing for implementation of alternate control algorithms into the system, for both lab and internet use.

Future work on the system will include adding more control algorithms to the SPENDULAP, both setpoint and tracking. Future work on the web interface will include developing a more aesthetic webpage, along with greater functionality, including a 3-D simulation of the system that allows the spectators to perform a desired experiment via simulation. Eventually, a functionality will be implemented that will allow web-engineers to implement their own algorithms into the system, thereby giving the web-engineer the ability to fully define the system. Finally, a haptic interface will be developed based on the work in this thesis and will be another step toward the telemechatronic future.
References


Appendix A
BLDCM Dynamics

The actuating motor that creates the drive torque, $\tau$, in Equation (2.6) is a three phase (3-φ) BLDCM. The dynamics of the BLDCM, dependent on precise manipulation of the 3-φ signal, can be derived using techniques found in [26, 27], but are usually derived using the classical AC method of decomposing the system into direct ($d$) and quadrature ($q$) axes terms. The derivation involves nonlinear differential equations [28]. Using the geometry of the 3-φ system and assuming the BLDCM is a smooth air-gap motor operating at a constant velocity, the resulting dynamic equations can be simplified to [26]:

$$V = \frac{M(K_d \omega + \tau_f \frac{\omega}{|\omega|} + \tau_i)}{K_t K_i}, \quad (A.1)$$

where

$$K_i = \sum_{j=1}^{3} \sin\left(\frac{n \theta - 2\pi(j - 1)}{3}\right) \sin\left(\frac{n \theta - 2\pi(j - 1) - \beta}{3}\right).$$

Table A.1 defines the variables used in this section.

The coefficient $M$, experimentally found to be $-5$ or $-7.2$, is a constant coefficient that correlates the input data acquisition card (DAC) voltage to the motor input current; it is negative because the actual torque of Equation (A.1) relates the torques of the system (left hand side) to the negative values of the output voltage. The coefficient is -5 for the sliding mode case and -7.2 for other cases. The coefficient is a significantly high value to compensate for two factors:
<table>
<thead>
<tr>
<th>$K_d$</th>
<th>Motor damping coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_i$</td>
<td>3-φ current multiplier</td>
</tr>
<tr>
<td>$K_t$</td>
<td>Motor constant</td>
</tr>
<tr>
<td>$I$</td>
<td>Input current from SERVOSTAR CD to motor</td>
</tr>
<tr>
<td>$L$</td>
<td>Overall motor inductance</td>
</tr>
<tr>
<td>$M$</td>
<td>System multiplier to define input voltage</td>
</tr>
<tr>
<td>$n$</td>
<td>Number of magnetic pole pairs in motor</td>
</tr>
<tr>
<td>$R$</td>
<td>Motor coil resistance</td>
</tr>
<tr>
<td>$V$</td>
<td>Input voltage for the DAC</td>
</tr>
<tr>
<td>$\beta$</td>
<td>$\beta = \arctan \frac{wL}{R}$</td>
</tr>
<tr>
<td>$\tau_f$</td>
<td>Rotation friction torque</td>
</tr>
<tr>
<td>$\tau_i$</td>
<td>Torque created by the SPENDULAP</td>
</tr>
<tr>
<td>$\omega$</td>
<td>Rotational velocity of the motor</td>
</tr>
</tbody>
</table>

**Table A.1** Motor dynamics variable definitions

1. The torque due to acceleration of the system, i.e., the $\frac{d\omega}{dt}$ term, was omitted. This term was ignored in the formulation of Equation (A.1) because the motor chatters and acceleration readings are imprecise at the available sampling rate of 125 Hz.

2. The second reason $M$ is a high value is to account for the pre-defined relationship of the current to torque as defined by the SERVOSTAR CD. The current to torque relationship can be changed using the SERVOSTAR CD control box.

It is important to note that the difference in $M$ between the sliding mode case and the other cases is due to the omission of the acceleration term in the formulation of Equation (A.1). The nature of the sliding mode algorithm is more greatly affected by the omitted acceleration term and hence the discrepancy in values for $M$. 
The actual current sent to the motor is

\[ I = 0.18 \times V. \]

The overall relationship between the input voltage from the DAC to the output torque of the motor, \( \tau_l \) in Equation (A.1), is

\[ \tau_l = \frac{K_t K_i V}{M} - \tau_f \frac{\omega}{|\omega|} - K_d \omega. \]

This relationship between the output voltage from Equation (A.1) sent to the DAC and converted to a current in the SERVOSTAR CD can be altered by changing the motor control variables DIPEAK and ISCALE using SERVOSTAR MOTIONLINK. The specific parameters for the motor used by the SPENDULAP in Equation (A), a Kollmorgen DDR Servomotor (D063A), are listed in Table A.2 [29].

Figure A.1 shows the nature of the BLDCM motor, namely that the system, commanded to operate at a constant \( \Omega \), fluctuates about the commanded velocity. The chattering nature of the BLDCM along with the available sampling rate of 125 Hz, led to the decision to use Equation (A.1) as the torque model for the control software.
Figure A.1  Comparison of simulation versus experiment commanded velocity for the SPENDULAP

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum $\omega$</td>
<td>500 RPM</td>
</tr>
<tr>
<td>Continuous Torque</td>
<td>19.7 N-m</td>
</tr>
<tr>
<td>Peak Torque</td>
<td>51.3 N-m</td>
</tr>
<tr>
<td>Continuous Current</td>
<td>4.7 Amps</td>
</tr>
<tr>
<td>Peak Current</td>
<td>14.5 Amps</td>
</tr>
<tr>
<td>$K_t$</td>
<td>268 V/kRPM</td>
</tr>
<tr>
<td>$R$</td>
<td>4.83 $\Omega$</td>
</tr>
<tr>
<td>$L$</td>
<td>18.6 mH</td>
</tr>
<tr>
<td>Number of Poles</td>
<td>16</td>
</tr>
<tr>
<td>$\tau_f$</td>
<td>0.61 N-m</td>
</tr>
<tr>
<td>Viscous Damping Coefficient</td>
<td>1.17 N-m/kRPM</td>
</tr>
</tbody>
</table>

Table A.2  DDR motor parameters
Appendix B
Friction Analysis

The SPENDULAP, like any physical system, has some amount of friction in it. There is both linear and nonlinear friction evident in the SPENDULAP. The linear model developed in this section is based solely on viscous damping in the swivel pin since that is the dominant linear factor. There is little coulomb friction evident due to the ball bearings in the system and hence it is ignored. A nonlinear model is hard to identify and is unnecessary as will be seen in the proceeding analysis.

A MATLAB program was written to measure the logarithmic decrement to find the linear viscous damping factor $\zeta$ in the SPENDULAP. To improve the accuracy of the approximation thirty-four experiments were performed with the DDBLDCM speed set to 0. The pendulum was raised to values between 5 - 85 degrees at five degree intervals and each initial angle experiment was performed twice.

The equation used to find the damping, based on the logarithmic decrement [7], is

$$\zeta = \frac{\delta}{(2\pi)^2 + \delta^2)^{1/2}}$$

where $\delta$, the logarithmic decrement, is the natural logarithm of the ratio of two consecutive peak values - i.e., $\delta = \ln \frac{x(t)}{x(t + T)}$.

The period of the system, $T$, can be found from graphical analysis of an ideal SIMULINK model or from analytical analysis of the equations of motion. When the
speed is 0, the period found using the simple pendulum equation [30], is

\[ T = 2\pi \sqrt{\frac{I_x}{MLg}} \]  

(B.1)

where \( M \) and \( L \) in Equation (B.1) are the overall mass and length to the center of mass of the pendulum bob and shaft respectively. The overall mass, \( M \), is the combined masses of the bob and rod and is 161 grams. The center of mass length, \( L \), is found from the following equation, derived in [31],

\[ \frac{m_{bob}l_{bob} + \frac{m_{rod}l_{rod}}{2}}{M} = 0.2167m. \]  

(B.2)

The physical values in Equations (B.1) and (B.2) can be found in 1.1. These are repeated below for convenience.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( g )</td>
<td>Gravitational constant</td>
<td>9.81 m/s(^2)</td>
</tr>
<tr>
<td>( I_x )</td>
<td>Inertia about the x axis</td>
<td>0.00845 kg-m(^2)</td>
</tr>
<tr>
<td>( l_{bob} )</td>
<td>Length to the bob’s center of mass from the swing shaft</td>
<td>0.254 m</td>
</tr>
<tr>
<td>( l_{rod} )</td>
<td>Length of the pendulum shaft</td>
<td>0.235 m</td>
</tr>
<tr>
<td>( m_{bob} )</td>
<td>Mass of the bob located at the end of the pendulum shaft</td>
<td>0.117 kg</td>
</tr>
<tr>
<td>( m_{rod} )</td>
<td>Mass of the pendulum shaft</td>
<td>0.044 kg</td>
</tr>
</tbody>
</table>

The resulting period at a 0 rotational velocity is thus \( T = 0.987s \).

When no rotational velocity was present, the median value for the damping factor, \( \zeta \), was found to be between 0.0002 and 0.068 with a mean value of 0.0025 and a median value of 0.0155.
To examine the system’s periodic motion at a constant, nonzero speed of \( \dot{\phi} = \Omega \), the system is linearized around \( \theta_{eq} \) and \( \dot{\theta} = 0 \). Based on the values for \( \zeta \) and knowing that \( C = 2\zeta I_x \omega_n \), where \( \omega_n = 6.364 \text{rad/sec} \), the range of values for damping coefficient \( C \) used in constant rotation analysis was set between 0.001 and 0.0075. This correlates to a damping coefficient range of 0.009 - 0.07. Though the true friction of the SPENDULAP is nonlinear, viscous damping is sufficient because inertia dynamics dominates cases of current interest.

Figure B.1 demonstrate the effects of friction on the dynamics of the system. The graph indicates the precision required to correlate simulation to experimental evidence. A constant correlation is impossible to derive since the friction constant \( C \) varies from experiment to experiment. Thus it was deemed impractical to include a friction model in the control software. Simulations show negligible discrepancy between cases when friction is and isn’t modelled in the control algorithms.
Figure B.1  SPENDULAP viscous friction: simulation versus experiment
Appendix C
LabVIEW Software

LabVIEW programs were developed to control the SPENDULAP. Some of the primary LabVIEW programs were discussed in Section 4.6. However, those programs are comprised of several smaller VI's. All the VI's used by the SPENDULAP are presented in Tables C.1 - C.8

<table>
<thead>
<tr>
<th>VI Name</th>
<th>Icon</th>
<th>Description</th>
<th>Called By</th>
<th>Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>SpendyServer Control</td>
<td></td>
<td>Main program that connects Internet to SDP</td>
<td>None</td>
<td>SC</td>
</tr>
<tr>
<td>Spendulap Control RT</td>
<td></td>
<td>Real Time program interface between server and SDP</td>
<td>None</td>
<td>SC</td>
</tr>
<tr>
<td>Spendulap Control Host</td>
<td></td>
<td>Inputs info from server to the RT engines</td>
<td>SpendyServer Control</td>
<td>SC</td>
</tr>
<tr>
<td>Calculate_DDR_Torque</td>
<td></td>
<td>Derives torque to DDR &amp; collects data</td>
<td>Spendulap Control RT</td>
<td>SC</td>
</tr>
</tbody>
</table>

Table C.1 Main VIs

<table>
<thead>
<tr>
<th>VI Name</th>
<th>Icon</th>
<th>Description</th>
<th>Called By</th>
<th>Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>spendulap.logic</td>
<td></td>
<td>Controls the on/off operation of SDP</td>
<td>Calculate_DDR_Torque</td>
<td>root</td>
</tr>
<tr>
<td>lift.motor.control</td>
<td></td>
<td>Controls the on/off operation of lift motor</td>
<td>spendulap.logic</td>
<td>root</td>
</tr>
<tr>
<td>motor.logic</td>
<td></td>
<td>Controls the on/off operation of brake/clutch</td>
<td>spendulap.logic</td>
<td>root</td>
</tr>
<tr>
<td>spendulap info &amp; gain</td>
<td></td>
<td>Has SDP properties &amp; computes gains for PP</td>
<td>Calculate_DDR_Torque</td>
<td>root</td>
</tr>
<tr>
<td>spendulap.info</td>
<td></td>
<td>Contains SDP's inertial parameters</td>
<td>spendulap info &amp; gain, SM_Calc_torque</td>
<td>root</td>
</tr>
</tbody>
</table>

Table C.2 SPENDULAP logic and parameter VIs

*SDP = SPENDULAP
†The root directory is: C:/Program Files/National Instruments/LabVIEW/Spendulap/, SC = SpendulapControl/, SM = Sliding Mode/
## Table C.3  Pendulum VIs

<table>
<thead>
<tr>
<th>VI Name</th>
<th>Icon</th>
<th>Description</th>
<th>Called By</th>
<th>Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal Rescaler</td>
<td><img src="image1.png" alt="Icon" /></td>
<td>Ensures pendulum encoder count doesn't overrun</td>
<td>Calculate_DDR_Torque, spendFun_logic</td>
<td>root</td>
</tr>
<tr>
<td>convert_to_degrees</td>
<td><img src="image2.png" alt="Icon" /></td>
<td>Converts pendulum encoder count into degrees</td>
<td>Calculate_DDR_Torque</td>
<td>root</td>
</tr>
<tr>
<td>rod and bob inertial term</td>
<td><img src="image3.png" alt="Icon" /></td>
<td>inertia parameters for pendulum</td>
<td>calc_torque ISIL, Calc_y, Calc_dynamics</td>
<td>root</td>
</tr>
</tbody>
</table>

## Table C.4  Torque calculation and conversion VIs

<table>
<thead>
<tr>
<th>VI Name</th>
<th>Icon</th>
<th>Description</th>
<th>Called By</th>
<th>Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gain Matrix Calc</td>
<td><img src="image4.png" alt="Icon" /></td>
<td>Calculates pole placement gains</td>
<td>spendFun_info &amp; gain</td>
<td>root</td>
</tr>
<tr>
<td>pole place matrix values</td>
<td><img src="image5.png" alt="Icon" /></td>
<td>Calculates A &amp; B matrices for PP</td>
<td>Gain Matrix Calc</td>
<td>root</td>
</tr>
<tr>
<td>sinusoid_torque</td>
<td><img src="image6.png" alt="Icon" /></td>
<td>Converts control torque to current value for DDR</td>
<td>Calculate_DDR_Torque</td>
<td>root</td>
</tr>
<tr>
<td>calculate states &amp; torque</td>
<td><img src="image7.png" alt="Icon" /></td>
<td>Calculates states and necessary torques for SDP</td>
<td>Calculate_DDR_Torque</td>
<td>root</td>
</tr>
<tr>
<td>calc_states</td>
<td><img src="image8.png" alt="Icon" /></td>
<td>Converts angles/rates into state form</td>
<td>calculate_states &amp; torque</td>
<td>root</td>
</tr>
<tr>
<td>calculate torque</td>
<td><img src="image9.png" alt="Icon" /></td>
<td>Calculates required torque based on chosen control</td>
<td>calculate_states &amp; torque</td>
<td>root</td>
</tr>
<tr>
<td>calc_torque ISIL</td>
<td><img src="image10.png" alt="Icon" /></td>
<td>Calculates torque for input state linearization control</td>
<td>calc_torque ISIL</td>
<td>root</td>
</tr>
<tr>
<td>SM_Calc_torque</td>
<td><img src="image11.png" alt="Icon" /></td>
<td>Calculates torque for sliding mode control</td>
<td>calc_torque ISIL</td>
<td>root</td>
</tr>
</tbody>
</table>

## Table C.5  Sliding mode VIs

<table>
<thead>
<tr>
<th>VI Name</th>
<th>Icon</th>
<th>Description</th>
<th>Called By</th>
<th>Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calc_y</td>
<td><img src="image12.png" alt="Icon" /></td>
<td>Calculates $\gamma$ term for sliding mode control</td>
<td>SM_Calc_torque</td>
<td>root</td>
</tr>
<tr>
<td>Calc_Ga</td>
<td><img src="image13.png" alt="Icon" /></td>
<td>Calculates $G(y, \xi)$ term for sliding mode control</td>
<td>SM_Calc_torque</td>
<td>root</td>
</tr>
<tr>
<td>Calc_fn</td>
<td><img src="image14.png" alt="Icon" /></td>
<td>Calculates $f(y, \xi)$ term for sliding mode control</td>
<td>SM_Calc_torque</td>
<td>root</td>
</tr>
<tr>
<td>Calc_dynamics</td>
<td><img src="image15.png" alt="Icon" /></td>
<td>Calculates $\frac{\partial f_2}{\partial y}$ term for sliding mode control</td>
<td>SM_Calc_torque</td>
<td>root</td>
</tr>
<tr>
<td>VI Name</td>
<td>Icon</td>
<td>Description</td>
<td>Called By</td>
<td>Directory</td>
</tr>
<tr>
<td>------------------</td>
<td>--------</td>
<td>------------------------------</td>
<td>-------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>Op_mode</td>
<td>![Op_mode Icon]</td>
<td>Changes the operational mode of the DDR</td>
<td>Spendlap Control Host</td>
<td>root</td>
</tr>
<tr>
<td>Internal Rescaler DDR</td>
<td>![Internal Rescaler DDR Icon]</td>
<td>Ensures DDR encoder count doesn't overrun</td>
<td>Calculate_DDR_Torque, spendlap_logic</td>
<td>root</td>
</tr>
<tr>
<td>DDR Count to Degrees</td>
<td>![DDR Count to Degrees Icon]</td>
<td>Converts DDR encoder count into degrees</td>
<td>Calculate_DDR_Torque</td>
<td>root</td>
</tr>
<tr>
<td>DDR Speed Info2</td>
<td>![DDR Speed Info2 Icon]</td>
<td>Calculates rotational speed of SDP</td>
<td>Calculate_DDR_Torque</td>
<td>root</td>
</tr>
<tr>
<td>DDR Parameters</td>
<td>![DDR Parameters Icon]</td>
<td>Contains DDR electrical parameters</td>
<td>Calculate_DDR_Torque</td>
<td>root</td>
</tr>
<tr>
<td>DDR,logic</td>
<td>![DDR,logic Icon]</td>
<td>Used for constant speed applications</td>
<td>Calculate_DDR_Torque</td>
<td>root</td>
</tr>
</tbody>
</table>

Table C.6  DDR VIs

<table>
<thead>
<tr>
<th>VI Name</th>
<th>Icon</th>
<th>Description</th>
<th>Called By</th>
<th>Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>contract_array</td>
<td>![contract_array Icon]</td>
<td>Removes all repeated data</td>
<td>Spendlap Control Host</td>
<td>root</td>
</tr>
<tr>
<td>contract_array.index</td>
<td>![contract_array.index Icon]</td>
<td>Gives the indices of data from contract_array</td>
<td>Spendlap Control Host</td>
<td>root</td>
</tr>
<tr>
<td>Eliminate Large Data</td>
<td>![Eliminate Large Data Icon]</td>
<td>Removes large/erroneous data from data set</td>
<td>Calculate_DDR_Torque</td>
<td>root</td>
</tr>
</tbody>
</table>

Table C.7  Data formatting VIs

<table>
<thead>
<tr>
<th>VI Name</th>
<th>Icon</th>
<th>Description</th>
<th>Called By</th>
<th>Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>timer</td>
<td>![timer Icon]</td>
<td>Calculates elapsed time of experiment</td>
<td>Calculate_DDR_Torque</td>
<td>root</td>
</tr>
<tr>
<td>unit_converter</td>
<td>![unit_converter Icon]</td>
<td>Converts units from rad to degrees, rad/s to rpm, etc.</td>
<td>Several locations</td>
<td>root</td>
</tr>
</tbody>
</table>

Table C.8  Miscellaneous VIs