

region on the plate using a quantized set of language-based commands. The size of this containment region was analytically determined as a function of the communication constraints and other control system parameters. The effectiveness of the proposed control law was evaluated in experiments and mathematical simulations.

SYMBOL-BASED CONTROL OF A BALL-ON-PLATE MECHANICAL SYSTEM

by

Phillip Yip

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2004

Advisory Committee:

Assistant Professor Dimitrios Hristu-Varsakelis, Chair/Advisor
Professor Balakumar Balachandran
Professor Amr Baz

©Copyright by

Phillip T. Yip

2004

DEDICATION:

To my family

ACKNOWLEDGEMENTS:

I would like to give thanks to my advisor, Dr. Hristu, for his guidance, to my labmates for their help along the way, and, of course, to my friends and family for their constant and invaluable support.

Contents

1	Introduction	1
1.1	Objectives and Contributions	3
1.2	Outline	4
2	Related Work	5
2.1	Stability Without Attention	5
2.2	Hybrid System Stability	6
2.3	Limited Communication Control	8
2.4	Quantization	10
2.5	Language-Based control	11
3	A Mechanical “Ball-On-Plate” System	14
3.1	Plate Kinematics	14
3.2	Ball and Plate Dynamics	16
3.3	Restriction to One Dimension	22
3.4	Linearization	23
4	Experimental Setup	25
4.1	Hardware	25
4.1.1	Mechanical Construction	25
4.1.2	PC-Based Controller	28

4.1.3	Computer Vision	28
4.1.4	Language-Driven Motors	29
4.2	Software	32
4.2.1	Operating System	32
4.2.2	Overall Control Program Layout	34
4.2.3	Image Acquisition	36
4.2.4	Ball Location Detection	37
4.2.5	Serial Communications	39
5	Control Strategy and Experiments	40
5.1	Stability for Unconstrained Switching	40
5.2	Communication Strategy and Control System Layout	42
5.3	Open-loop control	44
5.4	Proposed Control Algorithm	45
5.4.1	Controllability	47
5.4.2	Switching Regions	50
5.5	Simulation Results	53
5.5.1	Instantaneous Switching with Minimum Dwell Time	53
5.5.2	Effects of Delayed Switching	56
5.5.3	Non-instantaneous Switching with Dwell Time	57
5.5.4	Implementation of Predictive Switching and Experimental Results	60
6	Conclusions and Future Work	67
A	Mechanical Linkage Kinematics	69
	Bibliography	73

List of Figures

3.1.1	Diagram of ball on plate with $\phi_1 = \phi_2 = 0$	14
3.1.2	Diagram of ball on plate with one rotation of $-\phi_1$ in the \hat{i} direction as shown in light blue. In configuration of the plate for $\phi_1 = \phi_2 = 0$ is in yellow.	15
3.1.3	Diagram of ball on plate with a rotation of $-\phi_1$ in the \hat{i} direction and then another of ϕ_2 in the \hat{e}_2 direction	16
3.3.1	Diagram of ball on beam	22
4.1.1	Photograph of tilt-plate hardware assembly	26
4.1.2	Annotated photograph of a tilt-plate assembly with an inner plate and an outer platform. Double arrows indicate axes of rotation.	27
4.1.3	Photograph of tilt-plate with aluminum mounting blocks	28
4.1.4	Photograph of braces on the underside of the tilt-plate	29
4.1.5	Photograph of the outer platform supports and of the chain drive	30
4.1.6	Photograph of an aluminum support block with shaft and bearing	30
4.1.7	Photograph of an aluminum support block connected to a mechanical linkage and of the outer platform support	31
4.1.8	Photograph of yoke and clevis joint with shaft and coupler	31
4.1.9	Photograph motor with right angle coupler	32
4.1.10	Photograph of overhead camera	33

4.1.11	Photograph of one SilverMax motor	34
4.1.12	Photograph of chain drive	35
4.1.13	Photograph of RS-232 to RS-485 interface in use	36
4.2.1	Diagram of image thresholding. Top: Acutal camera image. Bottom: thresholded image	37
4.2.2	Ball location algorithm boxes	38
5.2.1	Control system block diagram. The inner loop consists of the motor and tilt plate assembly and K_i a variable (indicated by an arrow) feed- back controller integrated into the motor that receives language-based commands from the PC. The outer loop is what a casual observer would see if they were to view the system. The configuration of the mo- tor/plate assembly drives the ball dynamics. The camera functions as a sensor to retrieve the location of the ball. The PC interprets this data and issues a language-based command to the variable controller.	43
5.3.1	The general shape of the potential function with respect to r . A local minimum indicates a stable point.	45
5.4.1	Trajectories of the phase portrait for the one-dimensional ball on beam problem. Dashed lines are trajectories for $\gamma > 0$ and travel in the $-\dot{r}$ direction. Dashed-dot lines are trajectories for $\gamma < 0$ and travel in the $+\dot{r}$ direction.	51
5.4.2	Phase portrait switching curves for the one dimensional ball on beam problem	52

5.5.1	Simulated phase portrait of ball trajectory in the x direction with stability region under the first switching assumption: instantaneous switch followed by a finite time hold	54
5.5.2	Simulated phase portrait of ball trajectory in the y direction with stability region under the assumption: instantaneous switch followed by a minimum dwell time	55
5.5.3	Phase portrait of ball trajectory in the x direction with stability region superimposed under the assumption of time-delayed actuation	57
5.5.4	Phase portrait of ball trajectory in the y direction with stability region superimposed under the assumption of time-delayed actuation	58
5.5.5	Simulated phase portrait of ball trajectory in the x direction with stability region under assumption of non-instantaneous switching with minimum dwell time	60
5.5.6	Simulated phase portrait of ball trajectory in the y direction with stability region under assumption of non-instantaneous switching with minimum dwell time	61
5.5.7	Overhead x-y trajectory plot of the ball with and without predictive switching.	62
5.5.8	Phase portrait of ball trajectory in x direction with and without predictive switching	63
5.5.9	Phase portrait of ball trajectory in y direction with and without predictive switching	64
5.5.10	Position of ball in x direction with respect to time with and without predictive switching	64

5.5.11	Position of ball in y direction with respect to time with and without predictive switching	65
5.5.12	Velocity of ball in x direction with respect to time with and without predictive switching	65
5.5.13	Velocity of ball in y direction with respect to time with and without predictive switching	66
A.0.1	Diagram of inner linkage of plate assembly	69
A.0.2	Diagram of inner linkage of plate assembly superimposed over a photograph of the experimental assembly	70
A.0.3	Three-dimensional depiction of linkage annotated with angles θ , ϕ_1 , ϕ_2 , and α_2 . The dotted line between the spheres indicates the possible path of travel of point B.	72

Chapter 1

Introduction

Network-based components are quickly becoming the de facto standard for use in complex, distributed electromechanical systems. Modeling these complex systems requires the use of both continuous and discrete dynamics, since some components may trigger discrete “switches” in the behavior of other components. The overall system is thus a hybrid or switched system and cannot be studied effectively in the domain of classical controls.

A defining characteristic of a networked control system is that continuous control signal paths between components do not always exist. Signals flow between components through a shared medium. Controllers must thus carefully allocate their “attention”, or the portion of resources dedicated to communicate with other subsystems, to keep the system’s performance within an acceptable bound. It follows that the attention that a component receives is an important design parameter.

For example, consider a control system consisting of a person who is trying to juggle three balls. Components of this control system include sensing elements (eyes), a computing element (brain), and actuators (arms). A “low attention” control network has very little communication between its components. For this example, “low attention”

is akin to forcing the person to try to juggle while blindfolded. There is a very small chance that this person can successfully juggle, but if the person knows exactly how to throw each ball and how to catch each ball, it may be possible. However, if there are any disturbances in their throw then they will be unable to compensate for them since sensory data is not available. In a “full attention” control network, components have continuous access to one another. There is no “full attention” solution to this problem because the person lacks the sensing capabilities to track the location of all three balls at once. If they focus their eyes on the location of only one of the balls, they will be unable to ascertain the location of the other two balls. The only feasible solution in this case is a limited attention configuration where the person divides their attention between each of the balls and attends to each when they are about to catch it. When their eyes are not focused on a ball, the brain estimates its position based on the knowledge available the last time the person looked at that ball. Though the person is not able to throw and catch each ball with the same precision that the “full attention” configuration offers, the person can now juggle instead of just throw one ball and catch it.

There are two main approaches to dealing with attention constraints in a networked control system. The designer can choose to schedule attention between components in a way to allow the system to perform acceptably. If the communication network of a control system simply consists of a multi-poled switch that connects different wires at different times, the designer can schedule when each connection is allowed to be made. The attention between components does not have to be divided uniformly. Some components may be more demanding or susceptible to disturbances than others and therefore will need to receive more attention.

Alternately, the designer could implement a language-based control scheme which is akin to a coding policy that allows actuator signals to be communicated in short se-

quences of bits. The use of language-based control can allow lengthy commands to be shortened and frequently used sets of commands to be condensed to short symbols. This motion description language may also be universal so that commands sent in one networked control system will produce similar results in another. For example, any person can be given three balls and asked to “juggle”. Issuing this command is much simpler than issuing the series of commands required for one to juggle successfully. The linguistic term, “juggle” must be interpreted by controller to generate appropriate actuator signals.

In this thesis we will describe an electromechanical system that relies on a network to close its feedback loop and cannot be controlled effectively without taking into account its communication constraints. We seek to understand how the performance of the system is affected by the presence of communication constraints.

1.1 Objectives and Contributions

This aim of this thesis is to study the dynamics of a ball that rolls without slipping on a plane with controllable orientation. This system cannot be asymptotically stabilized due to its structural and communication constraints. This “ball-on-plate” system is an extension of the classical “ball-on-beam” experiment that is often used to study various types of control and stability problems. The objective of the problem remains the same: to balance a rolling ball at the center of the surface the plate surface. As we shall see, this system is not stabilizable with traditional feedback control techniques. We accomplish this by sampling feedback data and by using language-based controllers that draw commands from a finite set.

This thesis describes the mechanics and dynamics of a novel ball-on-plate system

and a control strategy for containing the ball within a small region on the plate. We showed that our proposed control scheme is “efficient” in the sense that balancing the ball cannot be accomplished with “full attention” or “no attention”. We calculated our system’s performance as a function of the communication rate of the system and parameters of the switching model.

1.2 Outline

The remainder of this thesis is arranged as follows: Chapter 2 consists of a review of previous research on related topics including the control of hybrid and switched systems, limited communication control, and motion description languages. Chapter 3 describes the kinematics and equations of motion of the ball-on-plate system. Chapter 4 is a discussion of the experimental apparatus used and the communication constraints that govern its operation. Chapter 5 proposes a control algorithm, presents a set of containability experiments using this algorithm and compares the results with theoretical predictions. Chapter 6, presents the conclusions of the paper and ideas for future work.

Chapter 2

Related Work

This section contains a brief summary of the research literature related to the problem of stabilization with limited attention. We survey works from the areas of open loop control, hybrid and switched systems, limited communication control, quantization and language-based control.

2.1 Stability Without Attention

Perhaps the simplest way to stabilize a system involves “no attention”, or the use of open-loop inputs without any feedback whatsoever. This principle was introduced in [38]. A classical unstable system, the inverted pendulum, is shown to be stabilizable at its upper unstable equilibrium point if sufficiently high frequency oscillations are introduced at its pivot point. A properly chosen range of pivot excitation amplitudes and frequencies force the pendulum to move upright because the average value of the torque over one period is not zero. Of course, not all systems are vibrationally controllable and in [38], [3] and [4], conditions for vibrational stability and controllability are established.

The work on the open-loop stabilization of the inverted pendulum was furthered in [15] and [11]. The last work used the potential energy function of the pendulum as a tool for determining the stability regions of the system. If a system is vibrationally controllable then this effective potential function can be shaped to have local minima at normally (unexcited) unstable points.

2.2 Hybrid System Stability

In a system with limited attention, components do not have access to one another all of the time. The dynamics of the system “switch” when connections between components are made or lost. The study of hybrid systems, or more specifically, switched systems, describe this type of discontinuous dynamics. Switched systems are a special class of hybrid systems where the discrete dynamics are simply changes in the continuous equations of motion.

For example:

$$\dot{x} = A_{\sigma(t)}x \tag{2.2.1}$$

where $\sigma(t) : R \rightarrow \{1, \dots, N\}$ and $A_1, \dots, A_N \in R^{n \times n}$.

It may be difficult to asymptotically stabilize a hybrid system due to switches in its dynamics or by discrete changes in the value of the state as a result of a switch. These systems can switch between different sets of dynamics with or without supervisory intervention. For example, a bouncing ball switches between sets of dynamics (free fall or contact) and can do so without intervention. A model of a ball that is being dribbled, however, is a switched system with supervisory and non-supervisory switches. It was shown in [7] that such a switched system is stable if switching points are chosen such that a Lyapunov function decreases at the system’s switching points. This total Lya-

Lyapunov function is composed of other Lyapunov functions that are active for different sets of dynamics.

Rather than pursue asymptotic stability, [40] explored the existence of periodic orbits and limit cycles in hybrid systems that converge to a closed, periodic trajectory. This is not asymptotic stability, however the size of the orbit may be small enough that the system's performance is acceptable. Conversely, some stable systems may have undesirable transient properties under switching, so that the boundedness of a system (the convergence to a fixed region after finite time), becomes a more useful criterion than stability.

The notion of practical stability was introduced in [41]. Practical stability implies that trajectories starting in an arbitrarily large neighborhood of the origin end up in an arbitrarily small neighborhood of the origin.

An alternate definition of practical stability was introduced in [35], where a system is practically stable if the state initially starts within a certain bound and stays within a certain larger bound as it evolves. More formally,

A system, $\dot{x} = f(t, x), x(t_0) = x_0, t_0 \geq 0$ is practically stable if, given (λ, A) with $0 < \lambda < A$, $|x_0| < \lambda$ implies $|x(t)| < A, t \geq t_0$ for some $t_0 \in R_+$. A system is uniformly practically stable if the above holds for all $t_0 \in R_+$.

In [48], the notion of containability is described. A system is containable on R^n if for any sphere N centered at the origin there exists an open neighborhood of the origin M and control and feedback laws such that any trajectory started in M remains in N for all time. This requires that M be a subset of N .

Sufficient conditions for practical stability for switched systems where the switches are not dependent on the state are developed in [50] and [39]. The authors show practical stability for systems where certain bounds exist for an auxiliary function of the state that

is similar to a Lyapunov function.

The authors of [40] introduce two notions of controllability for hybrid systems. The first is state-controllability and has the same goal as classical controllability. The second notion is mode-controllability, which is the having the ability to drive the system to a certain mode (set of continuous dynamics) or to force the system to pass through a sequence of modes. Conditions for state-controllability for hybrid systems in general have not been developed but have for restricted classes of hybrid systems, as in [25] and [45]. These two works limit their analysis to piecewise-linear hybrid systems. Numerical tests to determine controllability for piecewise affine hybrid systems are given in [5].

2.3 Limited Communication Control

The limited communication constraints in a control system are often the cause for a system to appear hybrid or switched. The control of any device over a network is subject to limited communication constraints that may arise due to bandwidth limitations of a shared communication channel. Additionally, communications and control using wireless network protocols are sensitive to these problems as often there may be physical obstructions that block communication, or problems associated with the distance of transmission and signal attenuation. Packets sent over a TCP/IP network are never guaranteed to arrive at their destination and may be lost if there are any computer problems along their route. The addition of random and expected delays obviously affect system performance.

If communication between a controller and other components is periodic, this introduces periodicity in the closed loop dynamics. In [1] and [32] studied systems with

periodic state feedback and found a practical method for assigning eigenvalues of the closed-loop transformation matrix using the period of the state feedback as a parameter. In such problems, it is often useful to create a Kalman filter that estimates the value of the state while relying on periodic updates from the feedback sensor to reduce the variance of the estimate of the state. This is done in [6], where additional necessary and sufficient conditions for asymptotic stability are given.

In [8], the effects of a limited communication constraint on a networked system with a central controller were studied. The authors studied how performance is affected by the choice of switching sequences between components. The problem of scheduling such sequences in networked control systems was researched in [47]. The authors implemented a dynamic scheduler that established network connections while the system ran and then determined its effects.

A set of feedback control systems subject to the constraint that only some subsystems have access to their controller at any given time is examined in [27]. A condition for Lyapunov stability of a switching sequence is developed that is a function of the characteristics of the controller subsystems and the number of allowed simultaneous network connections. It is also shown how stabilizing sequences can be chosen to minimize network congestion.

In [51], the authors examine a networked control system and establish relationships between stability regions, the sampling rate of the network and the time delay associated with data transmission in the network. The authors discuss methods of compensating for these communication problems by using an estimator. A Lyapunov measure was also developed to check for the stability of systems with data loss.

2.4 Quantization

While a limited communication control system has constraints in its communication network, quantization in controls applies constraints to the input or output space of a component. Quantization can reduce the communication bandwidth of a system by partitioning the input and output spaces of a control system into (possibly unbounded) cells and assigning a control value. The study of the quantization of system feedback in [14] showed that traditional feedback laws applied to open-loop unstable control systems do not always asymptotically stabilize the system and can result in chaotic behavior. In [18], the authors determine when a system with feedback quantization is stable by determining the coarsest stabilizing quantizer.

Another way to analyze a quantized control system is to approach it as an estimation problem. In [12], a certain amount of estimation error is assumed to be induced by the quantizer as “round-off error” and the true value of the state is estimated given knowledge of the nature of the quantization “noise” and feedback measurements.

An innovative approach to stabilizing a quantized linear system was proposed in [10]. That method involved dynamically adjusting the resolution of the quantizer as the system evolved. The approach was found to yield global asymptotic stability for feedback-stabilizable linear time-invariant systems but required that the system be treated as a hybrid system. In [19], the authors examine the effects of digital rounding in the feedback loop of a closed-loop system, which is a form of quantization. They develop an expression for the coarsest possible quantization of the feedback loop that would still guarantee stability.

2.5 Language-Based control

The use of language-based commands requires the formation of a command language but can reduce the required communication bandwidth of a system. As stated earlier, one simple command could substitute for a series of frequently performed commands in a networked control utilizing a language-based control structure. These commands can also be “universal” in that one could swap sensors and actuators with other compatible devices and expect a similar level of performance from the system. For example, with a PC as a supervisory controller, some printers are configured with a set of user-selectable fonts defined in read-only memory. When printing, the user can expect the same printed output regardless of which printer model is used.

Recent technological advances in system integration have allowed for control systems and related devices which are language-driven. The motors used to drive the mechanical system described in this thesis are examples of such technology. Current research topics include the establishment of an architecture for such systems and the development of appropriate “control languages”.

The foundations of language-based control were put forth by [46] and then [30] in the field of coding theory. For binary communication channels of finite capacity, [46] proposed arranging the set of possible commands in order of their likelihood of transmission and associating with each probability a binary number. The command most likely to be sent is assigned “0” and for each less likely command, the binary number symbol is incremented until it reaches the total number of possible commands. Thus, commands that are more frequently sent are encoded as smaller binary numbers to reduce the time it would take to transmit the command. This coding procedure is not optimal but reaches optimality as the number of commands transmitted approaches

infinity.

A similar encoding principle mentioned in [46] and [30] is used in language-based control. Instead of transmitting a short binary number in place of a potentially long command, in language-based control, a series of commands are represented by a single language-based command. This has the same benefit of freeing up attention by reducing the load on the communication channel.

A component in a networked control system may abruptly change its behavior upon receipt of a command. This is done in [24] by quantizing the set of system dynamics, implying a finite-length command set. The authors develop a control architecture for such systems and investigate its use in motion planning. Rather than searching for trajectories in an infinite-dimensional space, a “maneuver” library is constructed from a set of “primitives”, which are then used to compose more complicated trajectories. In their analysis, the authors seek to capture relevant characteristics of vehicle dynamics and to examine the behaviors resulting from combinations of control primitives as well as the required size of the set of these behaviors.

In [16], the authors attempted to define a metric for the command set of a language-based control system. For a robot that moves in a complex, cluttered environment, the authors established a metric for the number of commands that need to be issued to do perform a task. They also developed a method for calculating the benefits of the resulting reduced computational and communication overhead. The work in [17] found that the use of feedback in the specification language-based commands could reduce the length of the set of these commands.

The work in [8] describes MDL, a general motion description language for use in computer controlled devices that interact with their environment. The author frames issues involved in device-independent motion planning and attempts to formalize robot

programming by using kinematic and dynamic models of their motion and by constructing “behaviors” from control primitives. Language parameters such as sampling rate and instruction length are examined as well as methods of command translation.

MDL was later extended and as MDLe, or “extended MDL” in [36] [37], and [31]. The work in [29] combined these previous efforts in motion description languages into a more formal language definition of MDLe. The implementation of MDLe in stochastic environments was studied in [2] and the authors derived optimal control policies for solving navigation problems using MDLe sequences.

Chapter 3

A Mechanical “Ball-On-Plate” System

In this section we develop the kinematics and the equations of motion for a two degree of freedom ball-on-plate system.

3.1 Plate Kinematics

Consider the rigid plate in Figure 3.1.1 that rotates along one space-fixed axis and then again along a body fixed axis.

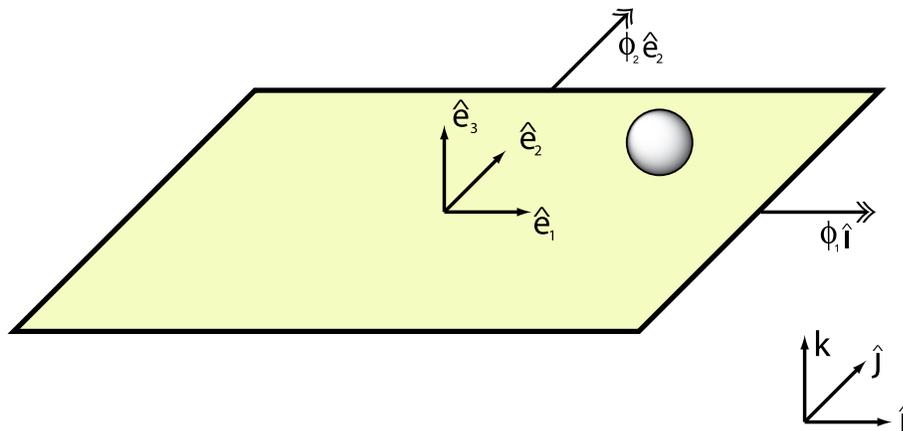


Figure 3.1.1: Diagram of ball on plate with $\phi_1 = \phi_2 = 0$

Let \hat{i} , \hat{j} and \hat{k} define an inertial reference frame with \hat{k} pointing in the vertical direction. Let \hat{e}_1 , \hat{e}_2 and \hat{e}_3 be a coordinate system fixed on the plate with \hat{e}_3 in the direction normal to the plate, and $\hat{e}_1 = \hat{i}$, $\hat{e}_2 = \hat{j}$ when the plate is level.

When the plate undergoes the rotation $\phi_2 \hat{e}_2$ followed by $\phi_1 \hat{i}$, the space-fixed coordinates of the body-fixed frame are:

$$\begin{pmatrix} \hat{e}_1 \\ \hat{e}_2 \\ \hat{e}_3 \end{pmatrix} = \begin{pmatrix} \cos\phi_2 & \sin\phi_1 \sin\phi_2 & -\cos\phi_1 \sin\phi_2 \\ 0 & \cos\phi_1 & \sin\phi_1 \\ \sin\phi_2 & -\sin\phi_1 \cos\phi_2 & \cos\phi_1 \cos\phi_2 \end{pmatrix} \begin{pmatrix} \hat{i} \\ \hat{j} \\ \hat{k} \end{pmatrix} \quad (3.1.1)$$

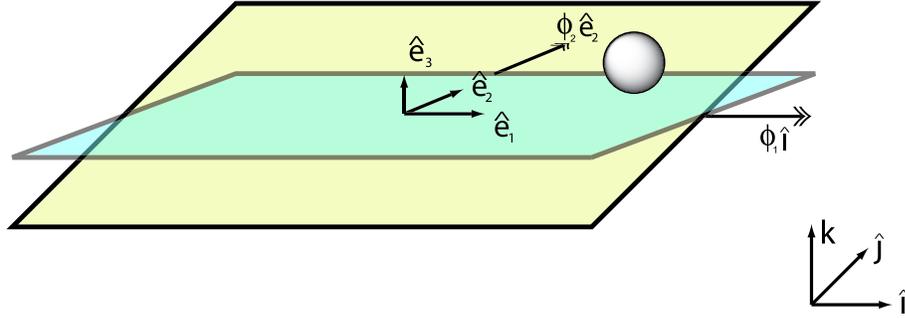


Figure 3.1.2: Diagram of ball on plate with one rotation of $-\phi_1$ in the \hat{i} direction as shown in light blue. In configuration of the plate for $\phi_1 = \phi_2 = 0$ is in yellow.

The inertial coordinates relative to the plate-fixed coordinates are:

$$\begin{pmatrix} \hat{i} \\ \hat{j} \\ \hat{k} \end{pmatrix} = \begin{pmatrix} \cos\phi_2 & \sin\phi_1 \sin\phi_2 & -\cos\phi_1 \sin\phi_2 \\ 0 & \cos\phi_1 & \sin\phi_1 \\ \sin\phi_2 & -\sin\phi_1 \cos\phi_2 & \cos\phi_1 \cos\phi_2 \end{pmatrix}^{-1} \begin{pmatrix} \hat{e}_1 \\ \hat{e}_2 \\ \hat{e}_3 \end{pmatrix} \quad (3.1.2)$$

In terms of rotation matrices, the transformation from $\{\hat{e}_1, \hat{e}_2, \hat{e}_3\}$ coordinates to

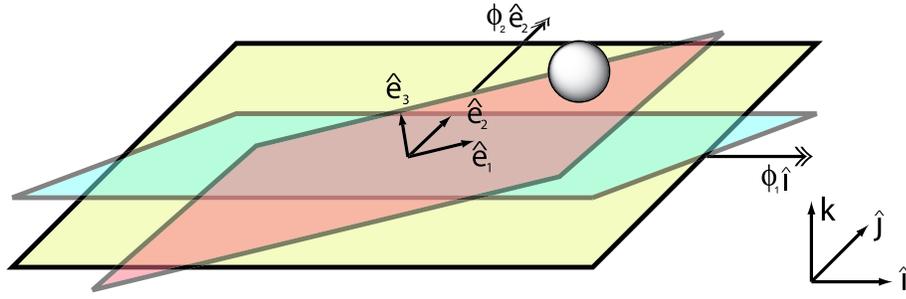


Figure 3.1.3: Diagram of ball on plate with a rotation of $-\phi_1$ in the \hat{i} direction and then another of ϕ_2 in the \hat{e}_2 direction

$\{\hat{i}, \hat{j}, \hat{k}\}$ is given by the orthogonal matrix:

$$\Theta = \begin{pmatrix} \cos\phi_2 & 0 & \sin\phi_2 \\ \sin\phi_1\sin\phi_2 & \cos\phi_1 & -\cos\phi_2\sin\phi_1 \\ -\cos\phi_1\sin\phi_2 & \sin\phi_1 & \cos\phi_1\cos\phi_2 \end{pmatrix} \quad (3.1.3)$$

If the position of a ball on the plate is, in terms of the body-fixed coordinates with basis $\{\hat{e}_1, \hat{e}_2, \hat{e}_3\}$:

$$\rho = \begin{pmatrix} r_1 \\ r_2 \\ R \end{pmatrix} \quad (3.1.4)$$

Then its position in the spaced-fixed coordinates with basis $\{\hat{i}, \hat{j}, \hat{k}\}$ is $\Theta \cdot \rho$.

3.2 Ball and Plate Dynamics

The position of the mass center of the ball at any time is:

$$\rho(t) = r_1(t)\hat{e}_1 + r_2(t)\hat{e}_2 + R\hat{e}_3 \quad (3.2.1)$$

where R is the radius of the ball.

The angular velocity of the plate can be calculated by computing the matrices:

$$\Omega_1 = \dot{\Theta}\Theta^T \quad (3.2.2)$$

$$\Omega_2 = \Theta^T\dot{\Theta} \quad (3.2.3)$$

Where Ω_1 and Ω_2 contain the components of the angular velocity in the space-fixed and body-fixed frames, respectively. They are:

$$\Omega_i = \begin{pmatrix} 0 & -\omega_a & \omega_b \\ \omega_c & 0 & -\omega_a \\ -\omega_b & \omega_a & 0 \end{pmatrix} \quad (3.2.4)$$

We can compute the absolute angular velocity of the plate as:

$$\omega_p = \dot{\phi}_1\hat{i} + \cos\phi_1\dot{\phi}_2\hat{j} + \sin\phi_1\dot{\phi}_2 \quad (3.2.5)$$

$$\omega_p = \cos\phi_2\dot{\phi}_1\hat{e}_1 + \dot{\phi}_2\hat{e}_2 + \sin\phi_2\dot{\phi}_1\hat{e}_3 \quad (3.2.6)$$

Taking the time derivative of the position of the ball and expressing its velocity relative to the inertial reference frame:

$$\dot{\rho} = \dot{r}_1\hat{e}_1 + \dot{r}_2\hat{e}_2 + \omega_p \times \rho \quad (3.2.7)$$

$$\begin{aligned} \dot{\rho} = & \left(\dot{\phi}_2 R + \dot{r}_1 - \dot{\phi}_1 r_2 \sin\phi_2 \right) \hat{e}_1 + \\ & \left(\dot{r}_2 + \dot{\phi}_1 (-R \cos\phi_2 + r_1 \sin\phi_2) \right) \hat{e}_2 + \\ & \left(-\dot{\phi}_2 r_1 + \cos\phi_2 r_1 \dot{\phi}_1 \right) \hat{e}_3 \end{aligned} \quad (3.2.8)$$

Alternatively, $\dot{\rho}$ could be calculated by:

$$\dot{\rho} = \frac{d}{dt}(\Theta\rho_e) \quad (3.2.9)$$

where ρ_e is the ball's location in the $\{\hat{e}_1, \hat{e}_2, \hat{e}_3\}$ coordinate system.

The rolling without sliding constraints for the ball are:

$$-R\omega_1 = \dot{r}_2 \quad (3.2.10)$$

$$R\omega_2 = \dot{r}_1 \quad (3.2.11)$$

where ω_1 and ω_2 are the angular velocities of the ball in the \hat{e}_1 and \hat{e}_2 directions, respectively.

The angular velocity of the ball relative to the space-fixed frame is equal to the sum of the angular velocity of the ball relative to the plate and the angular velocity of the plate.

Therefore,

$$\omega_b = \omega_1 \hat{e}_1 + \omega_2 \hat{e}_2 + \omega_3 \hat{e}_3 + \omega_p \quad (3.2.12)$$

Using the Lagrangian formulation with no external forces nor constraints:

$$L = K - V \quad (3.2.13)$$

$$\left(\frac{d}{dt}D_2L(q, v) - D_1L(q, v)\right) \cdot u = \alpha_e \cdot u \quad (3.2.14)$$

where K is the kinetic energy, V is the potential energy, $\alpha_e = 0$ are the external forces, and the test vector u is $u = (u_1, u_2, \mu_1, \mu_2)^T$.

The kinetic energy of the ball is given by:

$$K = \frac{m}{2}(\dot{\rho} \cdot \dot{\rho}) + \frac{1}{2}\omega_b \cdot J_b \cdot \omega_b \quad (3.2.15)$$

where

$$J_b = \begin{pmatrix} \frac{2mR^2}{5} & 0 & 0 \\ 0 & \frac{2mR^2}{5} & 0 \\ 0 & 0 & \frac{2mR^2}{5} \end{pmatrix} \quad (3.2.16)$$

Substituting,

$$\begin{aligned}
K = & \frac{m}{10} [2(\omega_1 + \cos\phi_2\dot{\phi}_1)^2 R^2 + 2(\omega_2 + \dot{\phi}_2)^2 R^2 + 2(\omega_3 + \sin\phi_2\dot{\phi}_1)R^2 + \\
& 5[(\cos\phi_2 r_2 \dot{\phi}_1 - r_1 \dot{\phi}_2)^2 + (-r_2 \sin\phi_2 \dot{\phi}_1 + R\dot{\phi}_2 + \dot{r}_1)^2 + \\
& ((-R\cos\phi_2 + r_1 \sin\phi_2)\dot{\phi}_1 + \dot{r}_2)^2]
\end{aligned} \tag{3.2.17}$$

The potential energy of the ball is given by:

$$V = mg[r_1 \cos\phi_1 \sin\phi_2 + r_2 \sin\phi_1 + R \cos\phi_1 \cos\phi_2] \tag{3.2.18}$$

Substituting known values, the Lagrangian is:

$$\begin{aligned}
L = & \frac{m}{10} [2(\omega_1 + \cos\phi_2\dot{\phi}_1)^2 R^2 + 2(\omega_2 + \dot{\phi}_2)^2 R^2 + 2(\omega_3 + \sin\phi_2\dot{\phi}_1)R^2 + \\
& 5[(\cos\phi_2 r_2 \dot{\phi}_1 - r_1 \dot{\phi}_2)^2 + (-r_2 \sin\phi_2 \dot{\phi}_1 + R\dot{\phi}_2 + \dot{r}_1)^2 + \\
& ((-R\cos\phi_2 + r_1 \sin\phi_2)\dot{\phi}_1 + \dot{r}_2)^2] - \\
& mg[r_1 \cos\phi_1 \sin\phi_2 + r_2 \sin\phi_1 + R \cos\phi_1 \cos\phi_2]
\end{aligned} \tag{3.2.19}$$

Evaluating appropriate derivatives:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{r}_1} = m[-\cos\phi_2 r_2 \dot{\phi}_1 \dot{\phi}_2 - \sin\phi_2 \dot{\phi}_1 \dot{r}_2 - r_2 \sin\phi_2 \ddot{\phi}_1 + R\ddot{\phi}_2 + \ddot{r}_1] \quad (3.2.20)$$

$$\begin{aligned} \frac{d}{dt} \frac{\partial L}{\partial \dot{r}_2} &= m[\dot{\phi}_1(\cos\phi_2 r_1 \dot{\phi}_2 + \sin\phi_2(R\dot{\phi}_2 + r_1)) + \\ &\quad (-R\cos\phi_2 + r_1 \sin\phi_2)\ddot{\phi}_1 + \ddot{r}_2] \end{aligned} \quad (3.2.21)$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\omega}_1} = \frac{2}{5} m R^2 (\dot{\omega}_1 - \sin\phi_2 \dot{\phi}_1 \dot{\phi}_2 + \cos\phi_2 \ddot{\phi}_1) \quad (3.2.22)$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\omega}_2} = \frac{2}{5} m R^2 (\dot{\omega}_2 + \ddot{\phi}_2) \quad (3.2.23)$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\omega}_3} = \frac{2}{5} m R^2 (\dot{\omega}_3 + \cos\phi_2 \dot{\phi}_1 \dot{\phi}_2 + \sin\phi_2 \ddot{\phi}_1) \quad (3.2.24)$$

$$\begin{aligned} \frac{\partial L}{\partial r_1} &= m[g\cos\phi_1 \sin\phi_2 + \sin\phi_2(-R\cos\phi_1 + r_1 \sin\phi_2 \dot{\phi}_1^2 + r_1 \dot{\phi}_2^2 + \\ &\quad \dot{\phi}_1(-\cos\phi_2 r_2 \dot{\phi}_2 + \sin\phi_2 r_2)] \end{aligned} \quad (3.2.25)$$

$$\begin{aligned} \frac{\partial L}{\partial r_2} &= -m[g\sin\phi_1 - r_2 \dot{\phi}_1^2 + \cos\phi_2 r_1 \dot{\phi}_1 \dot{\phi}_2 + R\sin\phi_2 \dot{\phi}_1 \dot{\phi}_2 + \\ &\quad \sin\phi_2 \dot{\phi}_1 \dot{r}_1] \end{aligned} \quad (3.2.26)$$

To find the equations of motion,

$$\left(\frac{d}{dt} D_2 L(q, v) - D_1 L(q, v)\right) \cdot u = \Lambda_{u_1} u_1 + \Lambda_{u_2} u_2 + \Lambda_{\mu_1} \mu_1 + \Lambda_{\mu_2} \mu_2 + \Lambda_{\mu_3} \mu_3 \quad (3.2.27)$$

where

$$\begin{aligned} \Lambda_{u_1} &= \frac{d}{dt} \frac{\partial L}{\partial \dot{r}_1} - \frac{\partial L}{\partial r_1} = -m[g\cos\phi_1 \sin\phi_2 + \sin\phi_2(-R\cos\phi_2 + r_1 \sin\phi_2) \dot{\phi}_1^2 + \\ &\quad r_1 \dot{\phi}_2^2 + 2\sin\phi_2 \dot{\phi}_1 \dot{r}_2 + r_2 \sin\phi_2 \ddot{\phi}_1 - R\ddot{\phi}_2 - \ddot{r}_1] \end{aligned} \quad (3.2.28)$$

$$\begin{aligned} \Lambda_{u_2} &= \frac{d}{dt} \frac{\partial L}{\partial \dot{r}_2} - \frac{\partial L}{\partial r_2} = m[g\sin\phi_1 - r_2 \dot{\phi}_1^2 + 2R\sin\phi_2 \dot{\phi}_1 \dot{\phi}_2 + 2\sin\phi_2 \dot{\phi}_1 \dot{r}_1 - \\ &\quad R\cos\phi_2 \ddot{\phi}_1 + r_1(2\cos\phi_2 \dot{\phi}_1 \dot{\phi}_2 + \sin\phi_2 \ddot{\phi}_1) + \ddot{r}_2] \end{aligned} \quad (3.2.29)$$

$$\Lambda_{\mu_1} = \frac{d}{dt} \frac{\partial L}{\partial \dot{\omega}_1} = \frac{2}{5} m R^2 [\dot{\omega}_1 - \sin\phi_1 \dot{\phi}_1 \dot{\phi}_2 + \cos\phi_2 \ddot{\phi}_1] \quad (3.2.30)$$

$$\Lambda_{\mu_2} = \frac{d}{dt} \frac{\partial L}{\partial \dot{\omega}_2} = \frac{2}{5} m R^2 [\dot{\omega}_2 + \ddot{\phi}_2] \quad (3.2.31)$$

$$\Lambda_{\mu_3} = \frac{d}{dt} \frac{\partial L}{\partial \dot{\omega}_3} = \frac{2}{5} m R^2 [\dot{\omega}_3 + \cos\phi_2 \dot{\phi}_1 \dot{\phi}_2 + \sin\phi_2 \ddot{\phi}_1] \quad (3.2.32)$$

The rolling without slipping constraint is applied to the test vector. If the rolling without slipping constraint were to be applied to the variables that become a part of the Lagrangian, the resulting equations would be incorrect due to the nonholonomic nature of the constraints.

Applying the rolling constraint to the test vector and simplifying,

$$\mu_1 = \frac{-u_2}{R} \quad (3.2.33)$$

$$\mu_2 = \frac{u_1}{R} \quad (3.2.34)$$

The time derivative of the rolling constraint yields two more equations for substitution:

$$\dot{r}_1 = R\dot{\omega}_2 \quad (3.2.35)$$

$$\dot{r}_2 = -R\dot{\omega}_1 \quad (3.2.36)$$

After these substitutions:

$$\begin{aligned} \Lambda_{u_1} + \frac{\Lambda_{\mu_2}}{R} &= \frac{2}{5}mR[\dot{\omega}_2 + \ddot{\phi}_2 - m(g\cos\phi_1\sin\phi_2 + \\ &\sin\phi_2(-R\cos\phi_2 + r_1\sin\phi_2)\dot{\phi}_1^2 + r_1\dot{\phi}_2^2 + 2\sin\phi_2\dot{\phi}_1\dot{r}_2 + \\ &r_2\sin\phi_2\ddot{\phi}_1 - R\ddot{\phi}_2 - \ddot{r}_1] \end{aligned} \quad (3.2.37)$$

$$\begin{aligned} \Lambda_{u_2} - \frac{\Lambda_{\mu_1}}{R} &= \frac{m}{5}[5g\sin\phi_1 - 5r_2\dot{\phi}_1^2 + 12R\sin\phi_2\dot{\phi}_1\dot{\phi}_2 + 10\sin\phi_2\dot{\phi}_1\dot{r}_1 - \\ &7R\cos\phi_2\ddot{\phi}_1 + 5r_1(2\cos\phi_2\dot{\phi}_1\dot{\phi}_2 + \sin\phi_2\ddot{\phi}_1) + 7\ddot{r}_2] \end{aligned} \quad (3.2.38)$$

Solving for \ddot{r}_1 and \ddot{r}_2 ,

$$\begin{aligned} \ddot{r}_1 &= \frac{1}{7}[5g\cos\phi_1\sin\phi_2 + 5\sin\phi_2(-R\cos\phi_2 + r_1\sin\phi_2)\dot{\phi}_1^2 + 5r_1\dot{\phi}_2^2 + \\ &10\sin\phi_2\dot{\phi}_1\dot{r}_2 + 5r_2\sin\phi_2\ddot{\phi}_1 - 7R\ddot{\phi}_2] \end{aligned} \quad (3.2.39)$$

$$\begin{aligned} \ddot{r}_2 &= \frac{1}{7}[-5g\sin\phi_1 + 5r_2\dot{\phi}_2^2 - 12R\sin\phi_2\dot{\phi}_1\dot{\phi}_2 - 10\sin\phi_2\dot{\phi}_1\dot{r}_1 + 7R\cos\phi_2\ddot{\phi}_1 - \\ &5r_1(2\cos\phi_2\dot{\phi}_1\dot{\phi}_2 + \sin\phi_2\ddot{\phi}_1)] \end{aligned} \quad (3.2.40)$$

Assume that the mass of the ball is much smaller than the mass of the beam. The moment of inertia of the plate is assumed to be J_{plate} and the same in the r_1 and r_2 directions. Linearizing about the origin:

$$\ddot{r}_1 \approx \frac{5}{7}g\phi_2 - 7R\frac{u_2}{J_{plate}} \quad (3.2.41)$$

$$\ddot{r}_2 \approx -\frac{5}{7}g\phi_1 + 7R\frac{u_1}{J_{plate}} \quad (3.2.42)$$

where u_1 and u_2 are torque inputs. We have chosen to linearize this system about the origin. Not that if $\phi_1 = \phi_2 = 0$ then there are an infinite number of equilibrium points for the system as long as the appropriate external torques are applied. In the special case of $r_1 = r_2 = 0$, no external torques are needed to maintain the system's equilibrium.

3.3 Restriction to One Dimension

When the dynamics of the ball on a tilt-plate are restricted to the one dimensional beam, the resulting equations of motion are similar to those found in the literature.

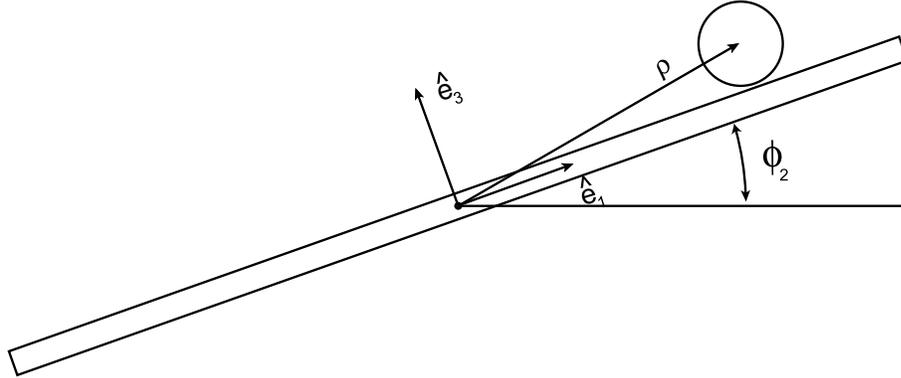


Figure 3.3.1: Diagram of ball on beam

Let ρ describe the location a ball of radius R on a one-dimensional beam at a distance of r from the center of the beam. The beam is rotated by an angle, ϕ_2 . Let \hat{e}_1 and \hat{e}_3 be

the basis vectors of a right-handed coordinate system fixed on the beam, with \hat{e}_1 oriented along the axis of the beam and \hat{e}_3 normal to the beam.

Setting $\ddot{\phi}_1 = \dot{\phi}_1 = \phi_1 = \dot{r}_2 = 0$ in Equation 3.2.39:

$$\ddot{r}_1 = \frac{1}{7} \left(5g \sin \phi_2 + 5r_1 \dot{\phi}_2^2 - 7R \ddot{\phi}_2 \right) \quad (3.3.1)$$

Linearizing about the origin yields 3.2.41. It should be noted that the $7R \frac{u_2}{J_{plate}}$ term in 3.2.41 is not included in discussions of the ball-on-beam problem in [33] and [43] for example.

3.4 Linearization

The following are the linearized equations of the entire system about the equilibrium point:

$$\begin{pmatrix} \dot{r}_1 \\ \dot{r}_2 \\ \ddot{r}_1 \\ \ddot{r}_2 \\ \dot{\phi}_1 \\ \dot{\phi}_2 \\ \ddot{\phi}_1 \\ \ddot{\phi}_2 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{5}{7}g & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{5}{7}g & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ \dot{r}_1 \\ \dot{r}_2 \\ \phi_1 \\ \phi_2 \\ \dot{\phi}_1 \\ \dot{\phi}_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -7R \frac{u_2}{J_2} \\ 7R \frac{u_1}{J_1} \\ 0 \\ 0 \\ \frac{u_1}{J_1} \\ \frac{u_2}{J_2} \end{pmatrix} \quad (3.4.1)$$

If the tilt angles of the plate are considered to be the inputs to the control system, the equations of motion of the system can be represented as:

$$\begin{pmatrix} \dot{r}_1 \\ \dot{r}_2 \\ \ddot{r}_1 \\ \ddot{r}_2 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ \dot{r}_1 \\ \dot{r}_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \frac{5}{7}g\phi_2 \\ -\frac{5}{7}g\phi_1 \end{pmatrix} \quad (3.4.2)$$

Defining

$$\gamma_1 \triangleq \frac{5}{7}g\phi_2 \quad (3.4.3)$$

$$\gamma_2 \triangleq \frac{5}{7}g\phi_1 \quad (3.4.4)$$

reduces Equation 3.4.2 to

$$\begin{pmatrix} \dot{r}_1 \\ \dot{r}_2 \\ \ddot{r}_1 \\ \ddot{r}_2 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} r_1 \\ r_2 \\ \dot{r}_1 \\ \dot{r}_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \gamma_2 \\ -\gamma_1 \end{pmatrix} \quad (3.4.5)$$

Chapter 4

Experimental Setup

This chapter describes the details of a mechanical ball-on-plate system including a description of its mechanical design and software that is used to control the system.

4.1 Hardware

We chose to model our system after the assembly of Figure 4.1.2 that includes an inner, gimbaled plate and an outer platform. This configuration allowed our tilt-plate to reach the angles specified in the kinematics section.

The rest of the hardware used in the experiment consists of an Intel-based PC running RTLinux, an overhead camera, a frame grabber card, smart motors, and the plate assembly. The experimental setup is shown in Figure 4.1.1.

4.1.1 Mechanical Construction

Aluminum braces for the plate were fabricated from hand drawings. The plate itself is 1/8" thick Lexan. Lexan was one of many possible materials that were chosen for their stiffness to weight ratio. Aluminum braces were placed on the underside of the tilt-plate

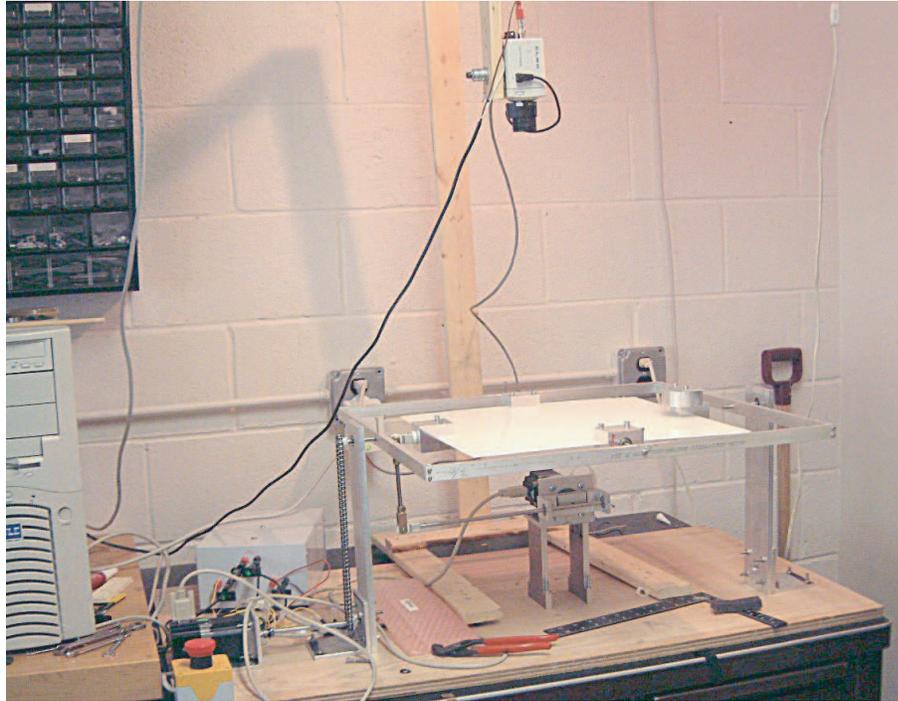


Figure 4.1.1: Photograph of tilt-plate hardware assembly

to prevent possible warping and nonlinearities arising from a plate that is not perfectly flat. The tilt-plate is shown in Figure 4.1.3 and the braces are shown in Figure 4.1.4. The plate is square and 16 inches in length.

The entire plate assembly consists of an inner and outer platform. As shown in Figure 4.1.2, this inner plate is gimbaled by the outer platform, which is actuated by a smart motor via a chain drive (see Figures 4.1.5 and 4.1.12). The plate is supported by four aluminum blocks that served to hold the aluminum braces against the underside of the platform, to gimbal the inner plate to the outer platform, and to connect the plate with the mechanical linkage that is used to adjust its angle. Two short shafts with radial bearings at each end connect to two of the aluminum blocks to the outer platform (see Figure 4.1.6).

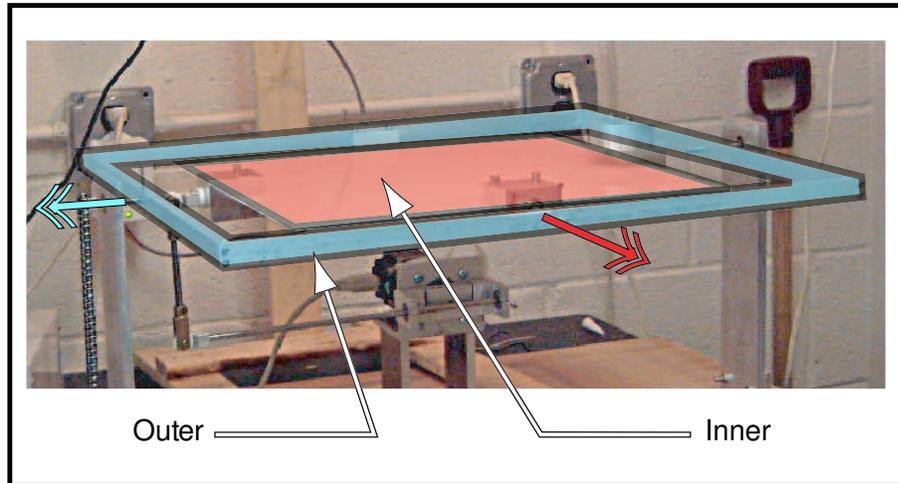


Figure 4.1.2: Annotated photograph of a tilt-plate assembly with an inner plate and an outer platform. Double arrows indicate axes of rotation.

One of the aluminum blocks is directly connected to the mechanical linkage (see Figure 4.1.7). A bearing fitted inside of the block is connected by a short shaft and an adapter to a yoke and clevis joint to approximate a ball joint. This configuration was selected over a traditional ball joint due to decreased friction and increased range of motion. Another aluminum block supports the aluminum braces and serves as a counterweight to balance the platform.

The outer platform of the assembly is supported on two ends by two short shafts that are free to rotate by use of bearings. One of these supports is shown in Figure 4.1.7.

The mechanical linkage consists of two yoke and clevis joints and the required shafts and couplers. The yokes were threaded onto one of the shafts for adjustability (see Figure 4.1.8). The linkage was coupled to the motor shaft via a right angle coupler designed to transmit motion with a minimum amount of play (see Figure 4.1.9).

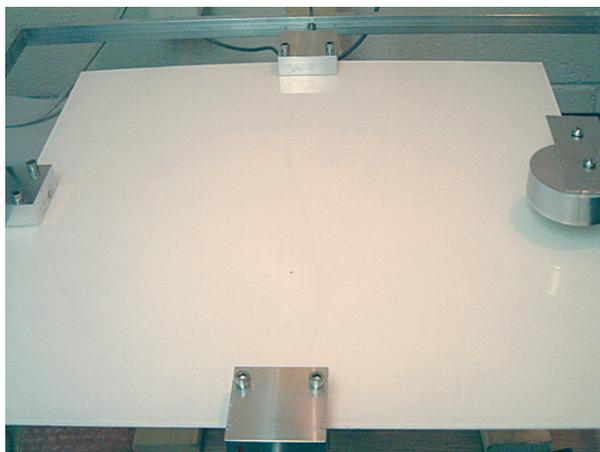


Figure 4.1.3: Photograph of tilt-plate with aluminum mounting blocks

4.1.2 PC-Based Controller

The computer used in the experiment is an Intel Pentium 4 1.6GHz computer with 1024MB of RAM. A patched Real-Time Linux operating system is used to ensure hard real-time capabilities. The use of RTLinux ensures that processes designated as real-time will not be superceded by other non-real-time processes despite software configuration changes or load increases. The PC serves at the central controller for the experiment, accessing both a vision system and two motors, described next.

4.1.3 Computer Vision

The camera used in the experiment as seen in Figure 4.1.10, is an Elmo TEB-4404 Black and White CCD camera with 570x350 resolution. The camera is configured to use the NTSC standard, providing interlaced data at 60Hz, alternating between odd-numbered and even-numbered horizontal image lines. The camera is fitted with a 3.5-8mm lens that has manual focus and zoom control with an automatic iris to regulate the amount of



Figure 4.1.4: Photograph of braces on the underside of the tilt-plate

light the sensor receives. A cable with a BNC plug at the camera end is connected to the frame grabber via an RCA plug. The camera provides feedback for the control system and is accessible by the PC via a frame grabber.

An Arvoo PCI-2SQ frame grabber is used for image acquisition. A number of other frame grabbers were considered, but the Arvoo frame grabber was selected for its RT Linux compatibility and support. The frame grabber supports multiple video formats (NTSC, PAL, and SECAM, with composite CVBS or Y/C S-video connectors) and accepts two digital inputs which were unused in the experiment. It can accept up to four inputs (multiplexed) however, in this case, only one was needed.

4.1.4 Language-Driven Motors

The tilt-plate was actuated using two QuickSilver Controls SilverMax 23-5 motors. One is shown in Figure 4.1.11. These motors were chosen for their high torque (255 oz-in peak), integrated design, and digital input capability. The motors have built-in encoders and controllers that can be tuned to suit the application. String-based commands, rather

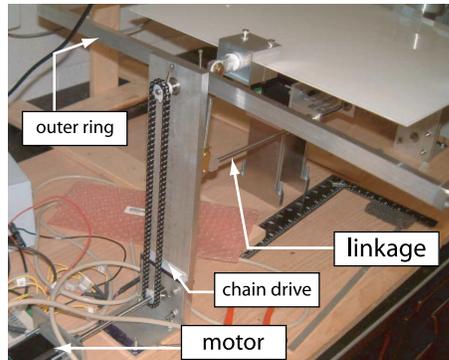


Figure 4.1.5: Photograph of the outer platform supports and of the chain drive

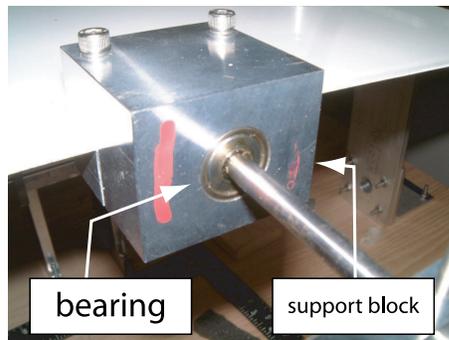


Figure 4.1.6: Photograph of an aluminum support block with shaft and bearing

than voltages and currents, are sent to the motors. The command set includes parameters for tuning the internal controllers. The motors' internal controllers run a frequency of 8.33kHz.

The mounts of the motors adhere to the NEMA 23 standard. The motors require a 36V DC power supply for motion and 5V DC power supply for logic purposes. One motor actuates the plate via a mechanical linkage and the other motor actuates the outer platform using a 1:1 chain drive as shown in Figure 4.1.12.

A RS-232 to RS-485 interface allows the two motors to be accessed by one serial port (see Figure 4.1.13). An example of a command to a motor is:

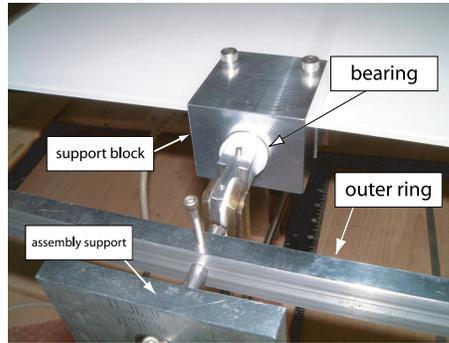


Figure 4.1.7: Photograph of an aluminum support block connected to a mechanical linkage and of the outer platform support

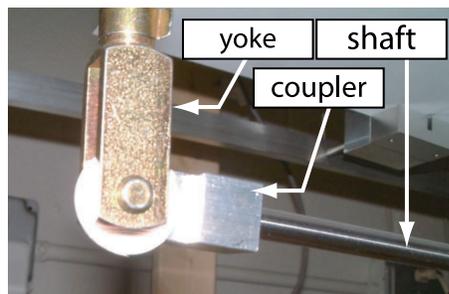


Figure 4.1.8: Photograph of yoke and clevis joint with shaft and coupler

@17 176 200 100 1000 0 0

The “@” symbol signifies the start of a command. Each motor can be identified by an 8-bit number. In this case, the motor was configured for the number 17. The number 176 is shorthand for the motor command “Move Absolute - Time Based” which tells the motor to accept a position command with parameters specified by the numbers following the 176. The parameters for the move specify a rotation of 200 counts (4000 counts are in a revolution) with an acceleration time of 100 ticks (120 μ s per tick) and with 1000 ticks being the total time of the motion. The last two numbers of the command are for interrupts and are not used in the experiment.

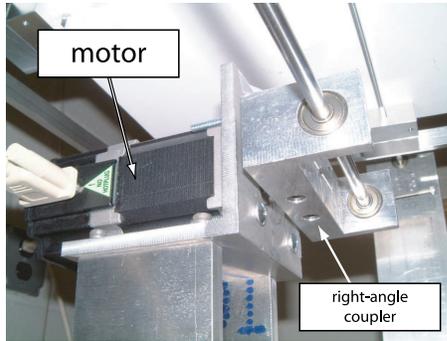


Figure 4.1.9: Photograph motor with right angle coupler

The motors have built-in controllers that accept digital position and velocity commands over an 8-bit serial connection. The PC locates the ball based on images obtained from the frame grabber and issues a command to the motors as necessary. The control loop runs at 14Hz (limited by the speed of image acquisition) and alternates between sending commands to each of the motors.

4.2 Software

The software used in this project was developed using the C language on a Real-Time Linux operating system.

4.2.1 Operating System

Most multi-tasking operating systems attempt to fairly balance the CPU load between numerous separate processes. RT Linux enables the user to write programs that have priority over all other processes. This enables hard real-time capabilities as timing can be specified to within 12 nanoseconds in some cases. The hard real-time nature of RT Linux removes most of the timing unpredictability that sometimes comes from having a



Figure 4.1.10: Photograph of overhead camera

computer-controlled system.

The Linux operating system was chosen due to the fact that it is open source and that the software used in this experiment was available for use, free of charge. The use of Linux also allows for more flexibility in configuring the computer. RT Linux allows for both real-time and non-real-time programs to run together, though the real-time component receives priority. Memory buffers can be shared between both real-time and non-real-time components for data access.

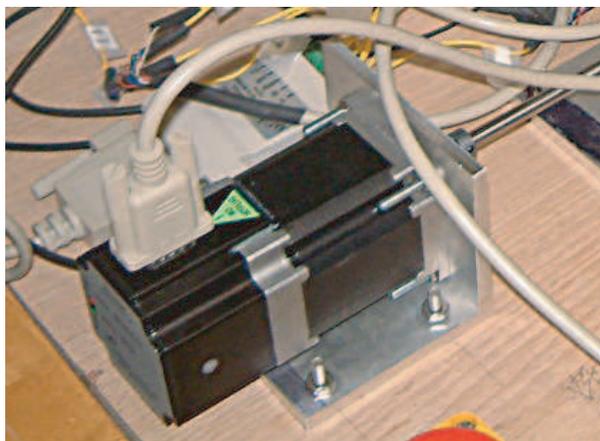


Figure 4.1.11: Photograph of one SilverMax motor

4.2.2 Overall Control Program Layout

The control loop can be summarized as follows:

1. Load necessary drivers (serial communications, video card, RT Linux).
2. Load real-time program component (Starts loop when non-real-time component sends “start” command). This component runs in hard real-time and thus its timing is much more precise than the non real-time program component.
3. Start non real-time component which will eventually trigger the real-time program component.
 - (a) Allocate memory buffers for data storage.
 - (b) Send “start” command to real-time component.
4. Real time component loop
 - (a) Acquire 8-bit bitmap image pixel data in the form of a long array.

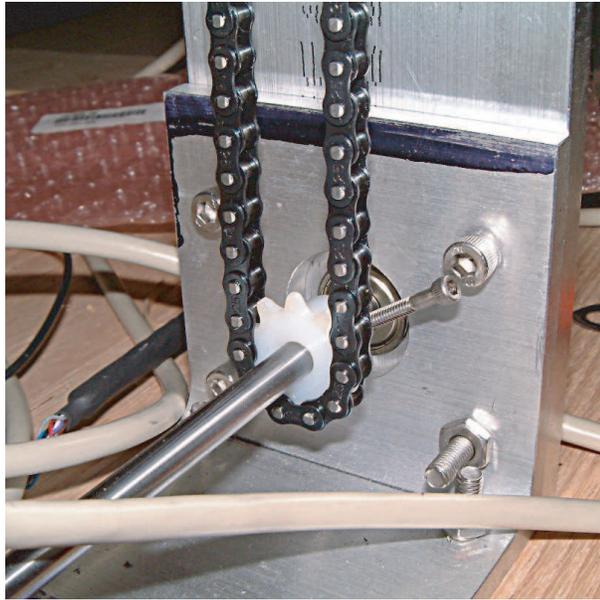


Figure 4.1.12: Photograph of chain drive

- (b) Apply a threshold filter to the image. Any pixel darker than a certain value is set to the color black. Anything lighter is set to be white.
- (c) Determine the image location of the ball using ball location detection algorithm described below.
- (d) Compute the actual location of the ball, accounting for plate tilt. Since the camera sees a two-dimensional projection of the plate and ball, changes in the plate angle could result in the apparent motion of the ball to the camera, even if the ball is held fixed.
- (e) Estimate the velocity of the ball based on its position at the last cycle and the time elapsed since the last cycle.
- (f) Calculate motor command parameters.
- (g) Alternate between each motor and send command to motor to move, if nec-

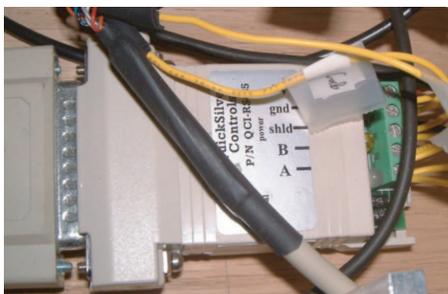


Figure 4.1.13: Photograph of RS-232 to RS-485 interface in use
essary.

- (h) Receive and interpret motor response (check for errors, etc.).
- (i) Output relevant data to the console for troubleshooting purposes.
- (j) Go back to the beginning of the loop.

The loop runs at about 14Hz, limited mainly by the time it takes to transfer an image buffer from the frame grabber to the PC's memory.

4.2.3 Image Acquisition

Due to the interlaced nature of the NTSC standard, the frame grabber can produce an updated, full 640 pixel by 480 image at the rate of 30Hz. Half of the image (alternating between odd and even horizontal lines) is delivered at the rate of 60Hz. To speed up the control loop, each odd or even frame is examined rather than the entire 640 by 480 frame. This reduces the effective resolution of the camera to 640 by 240 but can potentially double the speed of the control loop. A border around the image was ignored so that the braces for the plate did not interfere with the ball location detection algorithm. This yielded a final effective resolution of 500 by 200 pixels. The image covers a 10

inch by 10 inch area of the plate and therefore yields a 50 pixel/inch resolution in one direction and a 20 pixel/inch resolution in the other.

The image received via the frame grabber starts as an 8-bit greyscale bitmap image. The color data for all pixels is extracted into a one-dimensional array consisting of 8-bit elements.

4.2.4 Ball Location Detection

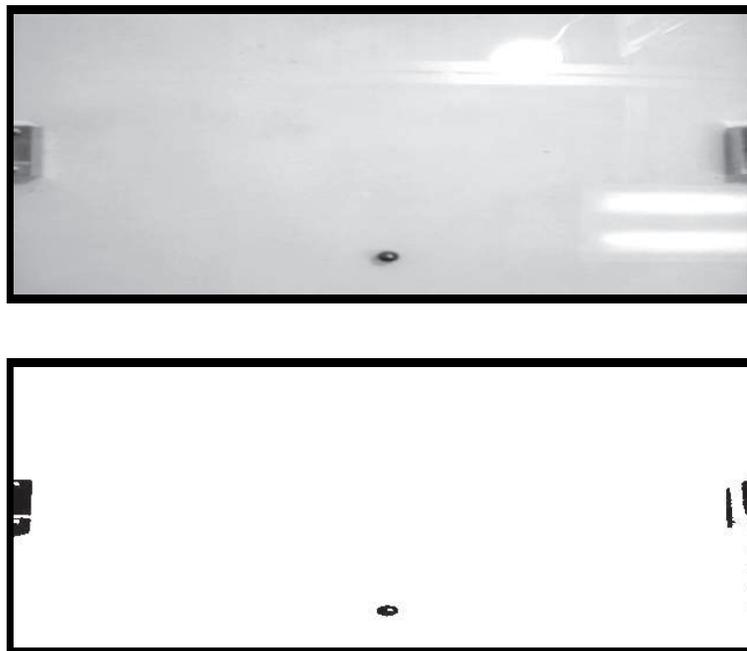


Figure 4.2.1: Diagram of image thresholding. Top: Actual camera image. Bottom: thresholded image

A threshold filter is applied at a preselected level to simplify the pixel data, with the goal of making the ball black and everything else white.

The ball location detection algorithm starts by looking for clusters of 3 black pixels. If the ball had been successfully located during the previous cycle, only a portion of the

image is scanned to search for the ball. The 3-pixel clusters are noted and then, with knowledge of the size of the ball in a camera image, the number of black pixels in a surrounding 10 by 10 pixel area is counted. This step is repeated for a 20 by 20 pixel area and a 40 by 40 pixel area.

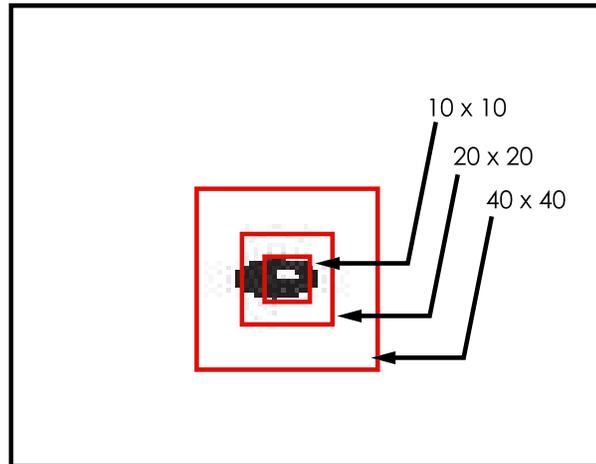


Figure 4.2.2: Ball location algorithm boxes

If the blackened area is too large (compared to the size of the ball), the pixel count for that area will be too high. These results are discarded. The potential guess for the location of the ball is also discarded if the black pixel count for an area is too low. The center of the ball is designated as the center of the 10 by 10 region that qualifies. If the ball is located, the next search is conducted by scanning only 50% of the total image area to avoid excess computation. This reduced image area is centered at a predicted location of the ball using position and velocity data from the previous cycle. If the ball is not located and if only a portion of the image was scanned, the search is reset, this time with the algorithm scanning the entire image.

4.2.5 Serial Communications

The **rt.com** software module is used in the experiment as a serial port interface. This allows data to be read and written to the serial port in the same way that data is read or written to a file. With the aid of the RS-232 to RS-485 converter, the host PC is able to communicate with both addressable SilverMax motors. Serial communications occurs at speeds up to 57600 bits per second. At 10 bits per character, commands consisting of 25 characters can be sent at a rate of 230.4Hz. Since the clock speed of the processor in each motor runs at 8.33kHz, the highest frequency at which commands can be sent and executed is 224Hz for 25-character commands.

Chapter 5

Control Strategy and Experiments

We go on to explore the possibility of open-loop (“no attention”) stabilization of the ball-on-plate system, as well as a feedback control law that uses inputs drawn from a finite set. We chose to limit the control system inputs to reduce the network’s communication bandwidth. First, we show that the system can be asymptotically stabilized if its angles are set to $\pm\gamma_1$ and $\pm\gamma_2$, where γ_1 and γ_2 are arbitrary. By quantizing the space of plate angles, we can ease the communication requirements of the system. Our choice of angles does not allow ϕ_1 or ϕ_2 to be equal to zero. With our choice of angles, there are no stable equilibrium points for the ball on the plate.

5.1 Stability for Unconstrained Switching

The ball-on-plate system can be made asymptotically stable if as few as four position inputs are used, and if we assume that switches can be made instantaneously and without any dwell time.

Let the time varying Lyapunov function, $V = f(r(t), \dot{r}(t))$ be given by:

$$V(t) = r^2(t) + \dot{r}^2(t) \tag{5.1.1}$$

where $r(t)$ is restricted to one dimensional ball-on-beam problem from Equation 3.4.5. Since the equations for $r_1(t)$ and $r_2(t)$ are not coupled, each direction can be treated independently.

Substituting time equations,

$$V(t) = \gamma^2 t^2 + 2\dot{r}(0)\gamma t + \dot{r}^2(0) + \frac{\gamma^2 t^2}{4} + \frac{\gamma t^3 \dot{r}(0)}{2} + \frac{\gamma t^2 r(0)}{2} + \dot{r}^2(0)t^2 + \dot{r}(0)r(0)t + r^2(0) \quad (5.1.2)$$

and

$$V(0) = \dot{r}^2(0) + r^2(0) \quad (5.1.3)$$

$V(t) - V(0)$ is the change in the value of the Lyapunov function after t seconds.

Substituting,

$$V(t) - V(0) = (\gamma^2 t^2 + \frac{\gamma^2 t^2}{4} + \dot{r}^2(0)t^2 + \dot{r}(0)r(0)t) + (2\gamma\dot{r}(0)t + \frac{\gamma t^3 \dot{r}(0)}{2} + \frac{\gamma t^2 r(0)}{2}) \quad (5.1.4)$$

$$V(t) - V(0) = \frac{\gamma^2 t^4}{4} + \frac{\gamma \dot{r}(0)t^3}{2} + (\gamma^2 + \frac{\gamma r(0)}{2} + \dot{r}^2(0))t^2 + (2\gamma\dot{r}(0) + \dot{r}(0)r(0))t \quad (5.1.5)$$

Let $\Delta V(t) = V(t) - V(0)$. If $\Delta V(t)$ is negative for some $t > 0$ then we can select a switching time, t , that would decrease the value of the Lyapunov function.

$$\frac{\partial \Delta V}{\partial t} = \frac{1}{4}t(\gamma t + 2\dot{r}(0))(\gamma(4 + t^2) + 4r(0) + 2t\dot{r}(0)) \quad (5.1.6)$$

The roots of $\frac{\partial \Delta V}{\partial t} = 0$ are:

$$t_{1,2} = \frac{-\dot{r}(0) \mp \sqrt{-2\gamma^2 - 2\gamma r(0) + \dot{r}^2(0)}}{\gamma} \quad (5.1.7)$$

and

$$t_3 = \frac{-\dot{r}(0)}{\gamma} \quad (5.1.8)$$

$t_3 > 0$ as long as the proper γ is chosen. Calculating $\Delta V(t_3)$ yields:

$$\Delta V(t_3) = -(\gamma + r(0))^2 \quad (5.1.9)$$

which is always less than or equal to zero. This shows that there always exists a $t > 0$ and γ such that the Lyapunov function will not increase after time t . Of course, this system is not physically possible as it will take a finite amount of time to switch γ . Introducing finite-time switching or a dwell time into the simulation model will more closely approximate the reality of the ball-on-plate problem.

Since we must choose γ_1 and γ_2 in Equation 3.4.2 to be greater than zero, the ball will roll away from the origin if the initial condition is at the origin. Thus, without instantaneous switching, the ball can never be forced to stay at the origin. In our control algorithm, we shall seek to aim for containability by driving the ball towards the origin after every switch has been completed.

5.2 Communication Strategy and Control System Layout

Of course, instantaneous switching is impossible and therefore we can not expect asymptotic stability. Constructing an observer or using full-attention feedback control via the PC is infeasible due to the low speed (14Hz) of its control loop, which is slowed while interpreting sensor information. The PC would not have been able to effectively send torque commands to the motors while interpreting information from the overhead camera. Though the high-frequency (8.33kHz) internal motor controllers are capable of issuing torque commands, they are incapable of communicating with the overhead camera.

Despite these constraints, the use of integrated language-driven motors enables us to use the idea of backstepping to contain the ball. Backstepping is, generally, the idea of letting some states function as controls for other states. We consider the position of the motor shafts, rather than the torque that the motors apply, as the control inputs for the ball-on-plate problem. This assumption can be made due to the fact that the integrated motor controllers operate at a much higher frequency than the decision-making PC. The PC controls the motors via a set four of language-based position commands (see Figure 5.2.1).

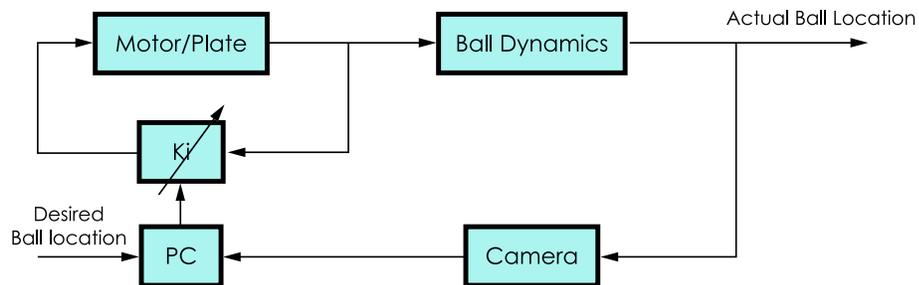


Figure 5.2.1: Control system block diagram. The inner loop consists of the motor and tilt plate assembly and K_i a variable (indicated by an arrow) feedback controller integrated into the motor that receives language-based commands from the PC. The outer loop is what a casual observer would see if they were to view the system. The configuration of the motor/plate assembly drives the ball dynamics. The camera functions as a sensor to retrieve the location of the ball. The PC interprets this data and issues a language-based command to the variable controller.

5.3 Open-loop control

The complexity of the control system for the ball-on-plate problem would be minimized if open-loop stabilization were possible. The effective potential method (see [15] and [11]) of open-loop oscillatory control was explored to see if the ball-on-plate system could be stabilized in this fashion. If the effective potential function can be shown to contain a local minima based on acceptable input parameters, then the system can be stabilized using high frequency oscillations without state feedback. The calculations leading to an expression for the effective potential energy function of the one-dimensional ball on beam setup follow.

We illustrate this by restricting the system to the familiar ball-on-beam problem. The equations of motion for the ball are given by Equation 3.2.41.

The potential function can be found by $U = -\frac{dF_i}{dr}(r)$ where $F_i = m\ddot{r}$ is the inertial force applied to the ball.

$$U = -\frac{5m}{7}(R\ddot{\phi}_2 r + \frac{r^2 \dot{\phi}_2^2}{2} - gr \sin \phi_2) \quad (5.3.1)$$

$$\frac{dU}{dr} = -\frac{5m}{7}(R\ddot{\phi}_2 + r\dot{\phi}_2^2 - g \sin \phi_2) \quad (5.3.2)$$

$$\frac{d^2U}{dr^2} = -\frac{5m}{7}\dot{\phi}_2^2 \quad (5.3.3)$$

The potential function has an extremum at

$$r = \frac{-R\ddot{\phi}_2 - g \sin \phi_2}{\dot{\phi}_2^2} \quad (5.3.4)$$

with a second derivative that is always decreasing. This implies that the only extreme for the potential function is a maximum. Even if ϕ_2 is sinusoidal in nature, the system cannot be stabilized using this method of open-loop control. Figure 5.3.1 shows the general shape of the potential function.

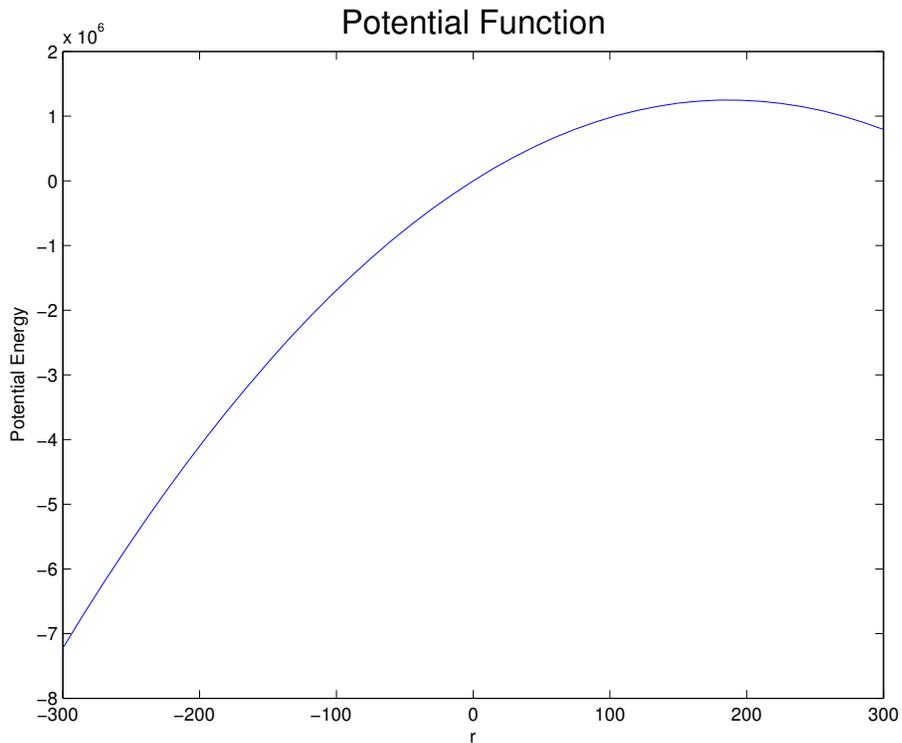


Figure 5.3.1: The general shape of the potential function with respect to r . A local minimum indicates a stable point.

5.4 Proposed Control Algorithm

The control algorithm used to contain the ball in a small area on the plate assumes that backstepping is used and that the tilt-plate is controlled by position commands. Note that the dynamics of each degree of freedom of the ball (r_1 and r_2 in 3.2.41 and 3.2.42) are decoupled, and therefore we can address the control of each direction independently. For each direction, we seek to steer the ball to the origin in minimum-time in a method similar to bang-bang control. Since the ball will overshoot the origin due to our switching and communication constraints, we will repeat this process *ad infinitum*.

Theorem 5.4.1. *Given:*

i. *An upper triangular system of the form:*

$$\dot{x} = \begin{pmatrix} A_1 & A_2 \\ 0 & A_3 \end{pmatrix} x + Bu \quad (5.4.1)$$

with

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in R^n \quad \text{and} \quad u \in R^m \quad (5.4.2)$$

ii. *The upper system, x_1 is controllable with $x_2 \in \{v_1, \dots, v_p\}$*

iii. *The transient of x_1 is bounded: $\|x_1(t) - x_{1f}\| \leq K \|x_{10} - x_{1f}\| \forall x_{10}, x_{1f} \in R^n$
such that \exists a time T and a sequence of v_i that steer $x_1(0) = x_{10}$ to $x_1(T) = x_{1f}$.*

iv. *There exists a control $u(t)$ that steers the lower system, x_2 , from any x_a to any x_b
with bounded error: $\|e(t)\|_2 < \alpha$ and $e_2(t) \rightarrow 0$ faster than $\max \operatorname{Re}(\lambda(A^T A))$*

*Then there exists a constant β and a $u(t)$ such that, we can steer the upper system
from any initial state to any final state in finite time, given the error function $e_2(t)$,
 $K \|x_o - x_f\| \rightarrow K \|x_o - x_f\| + \beta$ and $x_f \rightarrow x_f + \beta$.*

Proof.

$$x(t) = e^{At} x_0 + \int_0^t e^{A(t-\sigma)} B u(\sigma) d\sigma + \int_0^t e^{A(t-\sigma)} B e(\sigma) d\sigma \quad (5.4.3)$$

$$= x_*(t) + E(t) \quad (5.4.4)$$

where $x_*(t)$ is the value of the state in the absence of the error term.

$$\|E(t)\| \leq \int_0^t \|e^{A(t-\sigma)}\| \cdot \|B\| \cdot \|e_2(\sigma)\| d\sigma \leq \quad (5.4.5)$$

$$\int_0^t \|e^{\eta(t-\sigma)}\| \cdot \|B\| \cdot \|e_2(\sigma)\| d\sigma \leq \quad (5.4.6)$$

$$\beta \quad (5.4.7)$$

where $\eta = \max \lambda(A^T A)$. The integral converges for finite $t = T$ or $\|e_2(t)\| \leq \kappa e^{\gamma t}$ with $\gamma < -\eta$ and some constant κ .

Thus,

$$\|x_0 - x_f\| \leq (K + \beta) \|x_o - x_f\| + \beta \quad (5.4.8)$$

$$x_f = x_{f*} + \beta \quad (5.4.9)$$

□

Equation 3.4.2 shows that the ball-on-plate system in this experiment satisfies assumption *i* of the theorem. Assumption *iii* can be confirmed by measurement of the performance of the motors and the error of assumption *v* can be internally configured as the motors' "anti-hunt" tolerance. The controllability requirement of assumption *ii* is covered in the next section and the state boundedness requirement of assumption *iv* can be seen in the following section on switching regions.

5.4.1 Controllability

In this section, the controllability of the ball-on-plate system with a finite number of control inputs is established. It is easier to establish the controllability of the one-dimensional case and then extend it to the two-dimensional case. Equation 3.4.5 reduced to the one-dimension is:

$$\begin{pmatrix} \dot{r}_1 \\ \ddot{r}_1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} r_1 \\ \dot{r}_1 \end{pmatrix} + \begin{pmatrix} 0 \\ \gamma_2 \end{pmatrix} \quad (5.4.10)$$

Suppose that the magnitude of γ_2 is fixed and that switches occur that limit γ_2 to be only $\pm \frac{5}{7}g\phi_2$. Also suppose that no switches occur in the system for time, T_i . Equation 5.4.10 can be rearranged as a discrete-time system:

Let

$$\hat{r}_k = \begin{pmatrix} r_k \\ \dot{r}_k \end{pmatrix} \quad (5.4.11)$$

$$\begin{pmatrix} r_k \\ \dot{r}_k \end{pmatrix} = f_i(\gamma_2, r_{k-1}, T_i) = \begin{pmatrix} 1 & T_i \\ 0 & 1 \end{pmatrix} \begin{pmatrix} r_{k-1} \\ \dot{r}_{k-1} \end{pmatrix} + \begin{pmatrix} \gamma_2 \frac{T_i^2}{2} \\ \gamma_2 T_i \end{pmatrix} \quad (5.4.12)$$

for $k = \{1, 2, 3, \dots\}$.

Suppose for this system, $\gamma_2 > 0$ for time T_1 and the angle is switched afterwards for time T_2 . then

$$\hat{r}_k = f_2(-\gamma_2, r_{k-1}, T_2) \circ f_1(\gamma_2, r_{k-1}, T_1) \quad (5.4.13)$$

If the order of the switches is reversed,

$$\hat{r}_k = f_1(\gamma_2, r_{k-1}, T_2) \circ f_2(-\gamma_2, r_{k-1}, T_1) \quad (5.4.14)$$

Theorem 5.4.2. *Given the system 5.4.12 and switching order of 5.4.13 or 5.4.14, the system is controllable for the choices of either 5.4.13 or 5.4.14 and T_1 and T_2 subject to:*

$$T_2 \geq r_{k-1} + \sqrt{2\gamma_2 r_{k-1} + r_{k-1}^2 + 2\gamma_2 r_T} \quad (5.4.15)$$

or

$$T_2 \leq r_{k-1} - \sqrt{2\gamma_2 r_{k-1} + r_{k-1}^2 + 2\gamma_2 r_T} \quad (5.4.16)$$

and

$$T_1 > -r_{k-1} + r_T + \gamma_2 T_2 \quad (5.4.17)$$

for $T_1, T_2 \geq 0$.

Proof. Let

$$f_{1,2} = f_2 \circ f_1 \quad (5.4.18)$$

then,

$$f_{1,2} = \begin{pmatrix} r_{k-1} + r_{k-1}T_1 + \gamma_2 \frac{T_1^2}{2} + r_{k-1}T_2 + \gamma_2 T_1 T_2 - \gamma_2 \frac{T_2^2}{2} \\ r_{k-1} + \gamma_2 T_1 - \gamma_2 T_2 \end{pmatrix} \quad (5.4.19)$$

The system is controllable if 5.4.19 can be driven to any arbitrary value in R^2 . Solving 5.4.19 for the arbitrary value (r_T, r_T) yields the following constraints on T_1 and T_2 :

$$T_2 \geq \frac{r_{k-1} + \sqrt{2\gamma_2 r_{k-1} + r_{k-1}^2 + 2\gamma_2 r_T}}{\gamma_2} \quad (5.4.20)$$

or

$$T_2 \leq \frac{r_{k-1} - \sqrt{2\gamma_2 r_{k-1} + r_{k-1}^2 + 2\gamma_2 r_T}}{\gamma_2} \quad (5.4.21)$$

and

$$T_1 > \frac{-r_{k-1} + r_T + \gamma_2 T_2}{\gamma_2} \quad (5.4.22)$$

for $T_1, T_2 \geq 0$.

Since $\gamma_2 > 0$, Equations 5.4.20 through 5.4.22 can be reduced to Equations 5.4.15 through 5.4.17. This process may be repeated for the case of $f_{2,1}$ to complete the proof.

□

5.4.2 Switching Regions

Given a one dimensional plate, the phase portrait can be divided into certain switching regions based on the current position γ , the magnitude of γ , and the dynamics of the system. The switching planes (if the trajectory of the ball crosses a switching plane, it is best for the ball to switch) are found by solving for the trajectories that intersect with the origin, a minimum-time solution.

3.4.5 can also be integrated to yield the position of the ball as a function of time:

$$\dot{r}(t) = \gamma t + \dot{r}(0) \quad (5.4.23)$$

$$r(t) = \frac{\gamma t^2}{2} + \dot{r}(0)t + r(0) \quad (5.4.24)$$

To find solutions that pass through the origin, set $\dot{r}(t) = 0$ and $r(t) = 0$. This gives the equation:

$$\frac{\dot{r}^2(t)}{2\gamma} = r(t) \quad (5.4.25)$$

This equation tells us that if at time t its left and right sides agree with configuration γ then the ball will come to rest at the origin.

Setting 5.4.23 and 5.4.24 equal to zero and substituting yields:

$$r(t) = \frac{-\dot{r}^2(t)}{2\gamma} \quad (5.4.26)$$

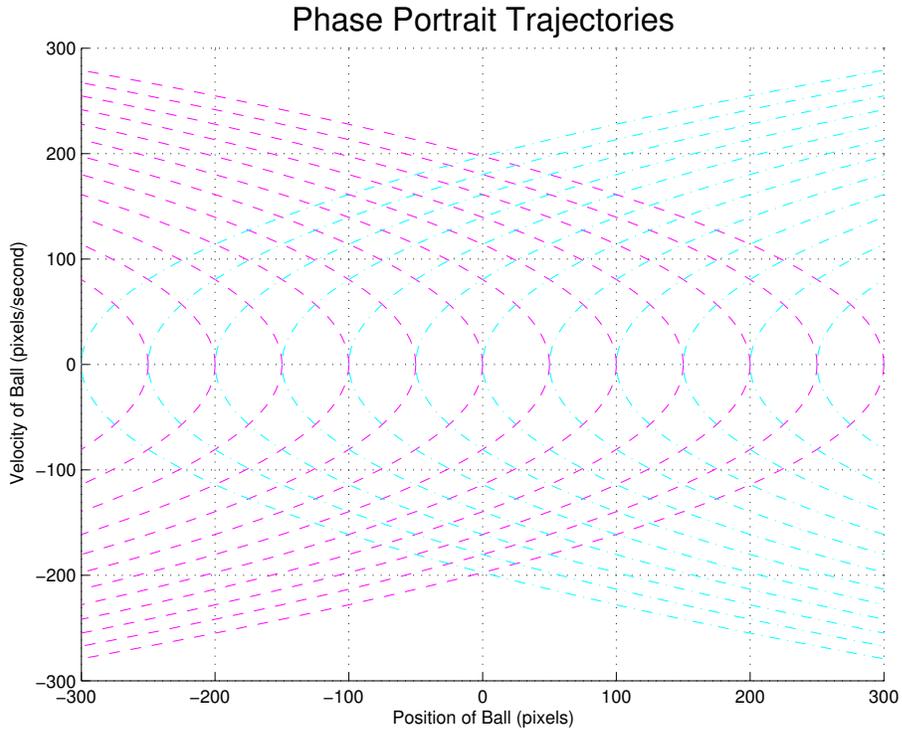


Figure 5.4.1: Trajectories of the phase portrait for the one-dimensional ball on beam problem. Dashed lines are trajectories for $\gamma > 0$ and travel in the $-\dot{r}$ direction. Dashed-dot lines are trajectories for $\gamma < 0$ and travel in the $+\dot{r}$ direction.

In general, and as shown in Figure 5.4.1, the trajectories of the ball on the phase portrait are:

$$r(t) = \frac{-\dot{r}^2(t)}{2\gamma} + c \quad (5.4.27)$$

where

$$c = \frac{-\dot{r}^2(0)}{2\gamma} + r(0) \quad (5.4.28)$$

The switching plane is

$$r(t) = \begin{cases} \frac{-\dot{r}^2(t)}{2|\gamma|} & \dot{r} > 0 \\ \frac{\dot{r}^2(t)}{2|\gamma|} & \dot{r} < 0 \end{cases} \quad (5.4.29)$$

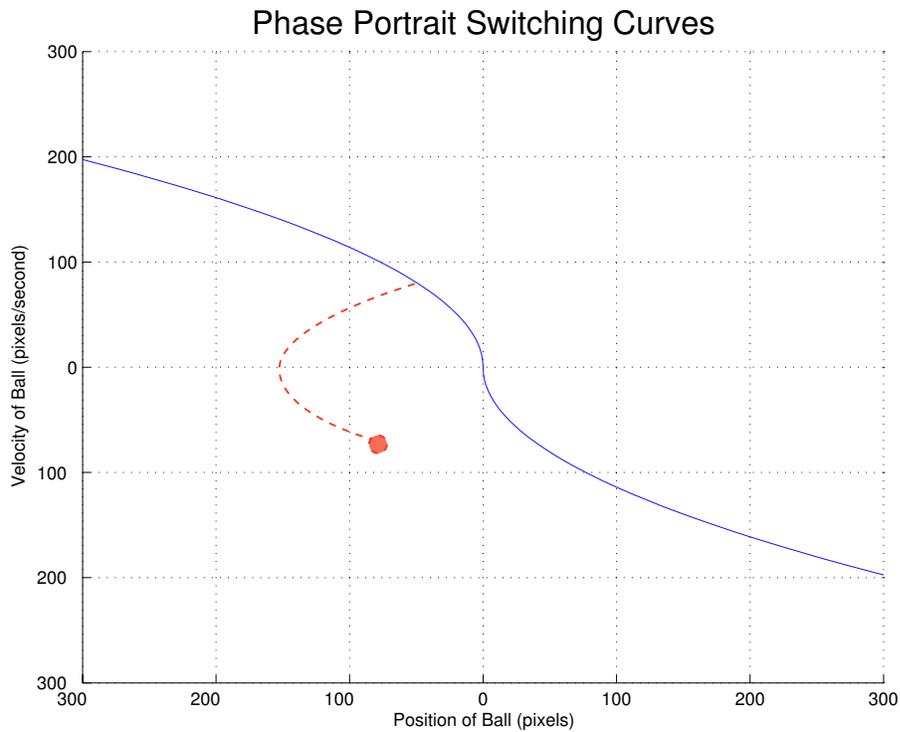


Figure 5.4.2: Phase portrait switching curves for the one dimensional ball on beam problem

The switching curves are shown in Figure 5.4.2. When the trajectory crosses one of these curves, the system should switch to reach the origin.

For $\gamma > 0$, it is best to switch to $\gamma < 0$ whenever the location of the ball on the phase portrait is to the left of the switching plane. For $\gamma < 0$, it is best to switch to $\gamma > 0$ whenever the location of the ball on the phase portrait is to the right of the switching plane.

If $\gamma > 0$ then there should be a switch when:

$$r(t) < \begin{cases} \frac{\dot{r}^2(t)}{2|\gamma|} & r(t) > 0 \\ \frac{-\dot{r}^2(t)}{2|\gamma|} & r(t) < 0 \end{cases} \quad (5.4.30)$$

and if $\gamma < 0$ then there should be a switch when:

$$r(t) > \begin{cases} \frac{\dot{r}^2(t)}{2|\gamma|} & r(t) > 0 \\ \frac{-\dot{r}^2(t)}{2|\gamma|} & r(t) < 0 \end{cases} \quad (5.4.31)$$

5.5 Simulation Results

5.5.1 Instantaneous Switching with Minimum Dwell Time

We constructed a simulation based on the assumption that switches for the angles ϕ_1 and ϕ_2 happen instantaneously but the tilt-plate is subject to a minimum dwell time after a switch takes place. Thus, if ϕ_1 is switched at t_0 , the angle ϕ_1 is fixed for the dwell time of T seconds, during which ϕ_2 is free to switch if ϕ_2 has not been switched in the last T seconds.

We found that the ball could be kept within a bounded area whose size and shape are dependent on the assumed delay time, T , and the plate angles, $\pm\gamma$. A point (r_s, \dot{r}_s) is in the region if the following hold:

$$\frac{\dot{r}_s^2}{2|\gamma|} - |\gamma|T^2 < r_s < \frac{-\dot{r}_s^2}{2|\gamma|} + |\gamma|T^2 \quad (5.5.1)$$

$$\begin{aligned} -2\gamma T + \sqrt{2\gamma(r + \gamma T^2)} < \dot{r} < 2\gamma T - \sqrt{2\gamma(r + \gamma T^2)} & \text{for } r < 0 \\ -2\gamma T + \sqrt{2\gamma(-r + \gamma T^2)} < \dot{r} < 2\gamma T - \sqrt{2\gamma(-r + \gamma T^2)} & \text{for } r > 0 \end{aligned} \quad (5.5.2)$$

The region enclosing the ball, as shown in Figures 5.5.1 and 5.5.2 was determined by examining the worst case scenario that would drive the ball the furthest from the origin. The worst possible time for a switch is when the ball is at the origin of its phase

portrait, because the ball cannot be steered any closer to the origin. The ball is forced to roll away from the origin for T seconds under our switching assumption.

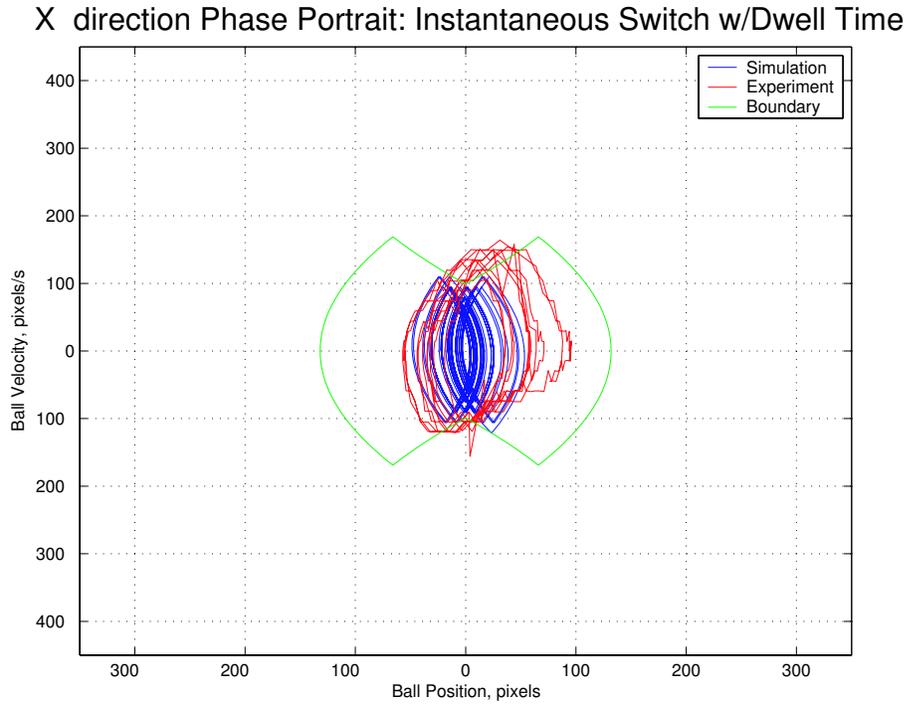


Figure 5.5.1: Simulated phase portrait of ball trajectory in the x direction with stability region under the first switching assumption: instantaneous switch followed by a finite time hold

If the switching strategy is followed correctly, the ball should come to rest at the origin in one switch. The reason for this is that there was no previous switch and therefore there is no time delay T to prevent a switch from being made. This switching strategy does not allow a switch to be made “early” if the ball is headed towards the origin, since an “early” switch will always push the ball farther from the origin after T seconds. The only possible time for a switch is a “late” switch that is made after the ball passes through the origin. This “late” switch occurs when the angle of the plate is held due to

Y direction Phase Portrait: Instantaneous Switch w/Dwell Time

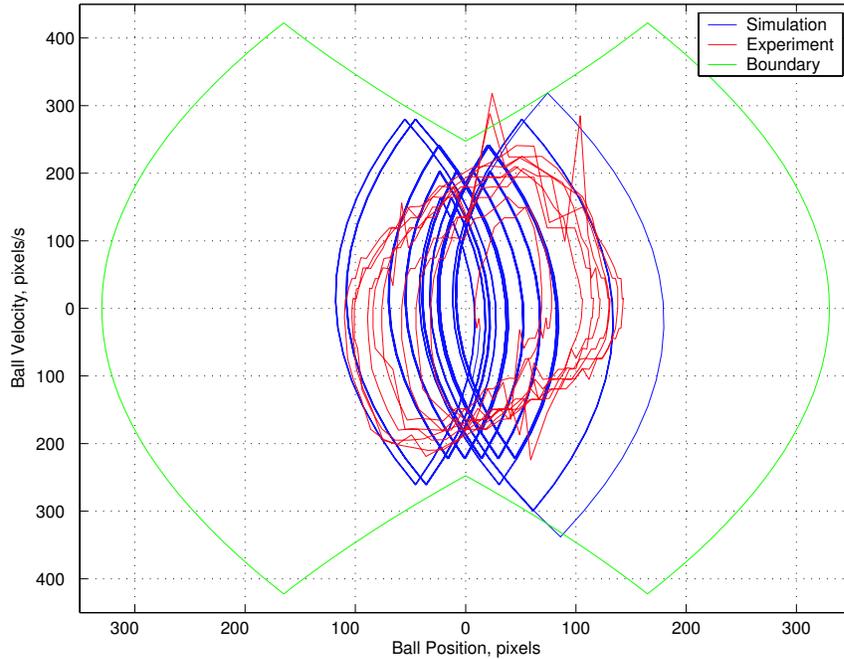


Figure 5.5.2: Simulated phase portrait of ball trajectory in the y direction with stability region under the assumption: instantaneous switch followed by a minimum dwell time the system’s dwell time.

Most of the bounding region in Figures 5.5.1 and Figures 5.5.2 was created by a switch at the origin. Following a switch at the origin, the ball will reach either the upper-right cusp or the lower-left cusp of the boundary region as the plate’s dwell time expires. The location of these cusps are given by $(\frac{\pm\gamma T^2}{2}, \frac{\pm\gamma T}{2})$. Another switch then occurs, tracing the parabola on the right or the left side of the boundary region until. Under our control algorithm, the plate switches again, $3T$ seconds after the original switch at the origin, and $2T$ seconds after the previous switch. We computed the remainder of the boundary region by analyzing the results of “late” switches. Four unstable limit cycles form under this switching assumption. Two of these cycles form when a switch is made

$\frac{T}{2}$ seconds before and after the ball crosses the origin. The other two form when a switch is made T seconds before and after the ball crosses the origin.

We superimposed a simulation of the trajectory of the ball with dynamics adhering to our control algorithm and actual experimental results in Figures 5.5.1 and 5.5.2. A random initial location was chosen for the experiment and we attempted to match this location with the initial conditions in our simulation. The results of the experiment appear to correlate well with the simulated results and stay within the boundary region for the most part.

5.5.2 Effects of Delayed Switching

In an attempt to refine our model, we made an alternate assumption: The decision to switch an angle of the plate could be made at any time but the actual switch could not take place until T seconds after the decision was made.

The shape of the limit cycle appeared to be the intersection of those trajectories that intersect the \dot{r} axis T seconds after the switching curve is crossed. We confirmed this analytically. The intercepts on the \dot{r} axis of the limit cycle are $\pm 2\gamma T + \sqrt{2}\gamma T$ and the r axis intercepts of the limit cycle are $\gamma T^2(3 + 2\sqrt{2})$. The orbit is symmetric along the \dot{r} axis and the r axis, and appears to be stable. A diagram that illustrates this limit cycle is shown in Figures 5.5.3 and 5.5.4.

The simulated results under this delayed switching assumption showed a much larger containment region than that observed under our previous instantaneous switching assumption and did not correlate well with experimental results. The delayed switching assumption is much too conservative.

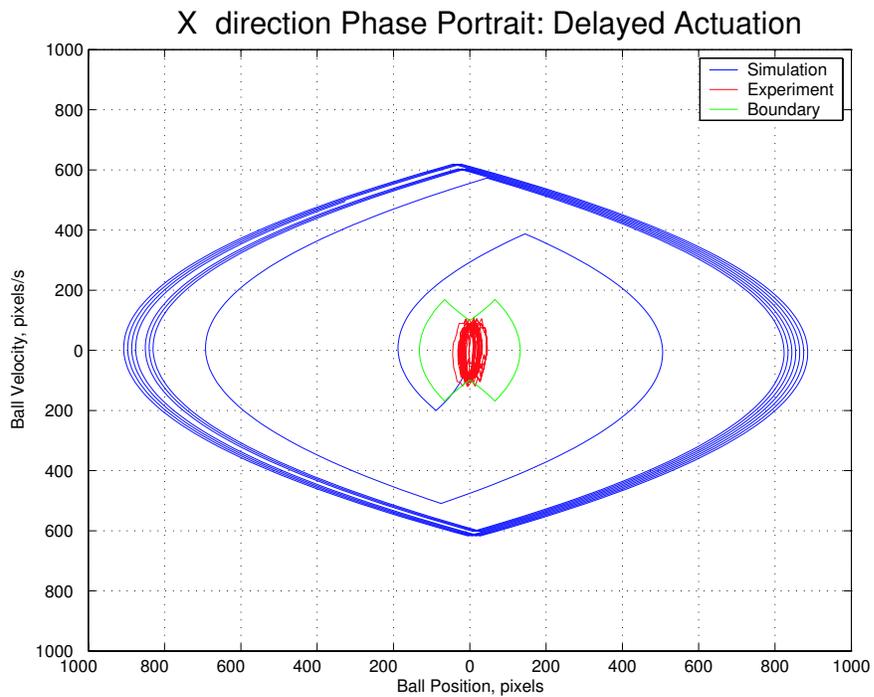


Figure 5.5.3: Phase portrait of ball trajectory in the x direction with stability region superimposed under the assumption of time-delayed actuation

5.5.3 Non-instantaneous Switching with Dwell Time

A third more refined switching model was postulated. As in the previous case, the decision to make a switch may occur at any time, but a switch cannot occur until T seconds after the decision is made. This time, however, a switch is made when the state at $t + T$ crosses the desired switching curve. The prediction of the state at time $t + T$ is made by assuming a constant plate velocity while the switch is being made, though in the simulation, it is still assumed that the switch is made instantaneously T seconds after the decision to switch is made.

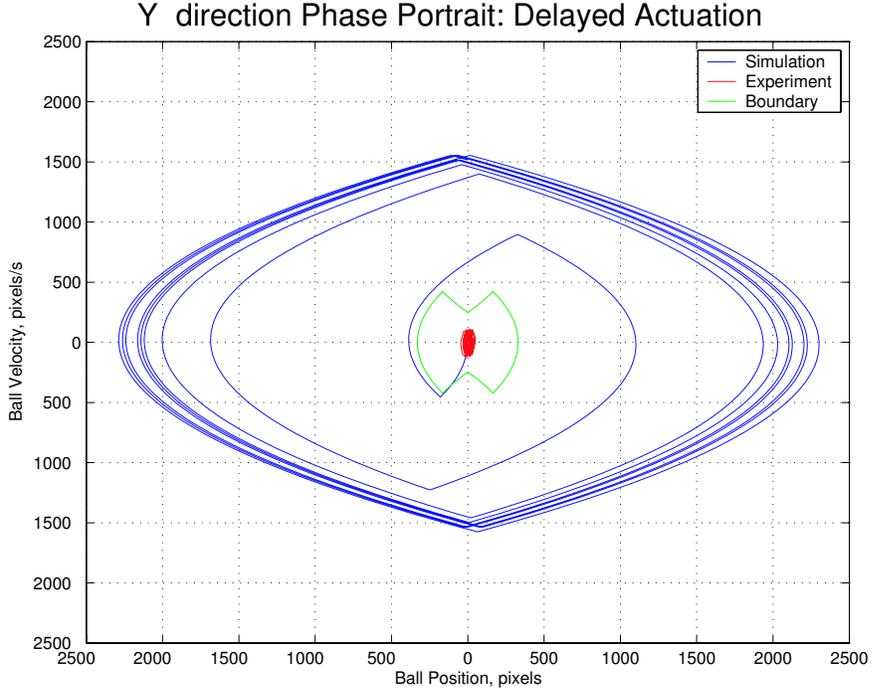


Figure 5.5.4: Phase portrait of ball trajectory in the y direction with stability region superimposed under the assumption of time-delayed actuation

The equations of motion during the switch are:

$$\dot{r}(t) = \frac{5gT \cos(\alpha_s - \frac{2\alpha_s t}{T})}{14\alpha_s} - \frac{5gT \cos \alpha_s}{14\alpha_s} + \dot{r}(0) \quad (5.5.3)$$

$$r(t) = \frac{-5gT^2 \sin(\alpha_s - \frac{2\alpha_s t}{T})}{28\alpha_s^2} + \frac{5gT^2 \sin \alpha_s}{28\alpha_s^2} + \frac{5gT \cos \alpha_s}{14\alpha_s} + \dot{r}(0)t + r(0) \quad (5.5.4)$$

where α_s is the angle the plate achieves upon completion of the switch, and $\dot{r}(0), r(0)$ are the values of the state right before the switch. To find the location of the state after the switch, set $t = T$:

$$\dot{r}(T) = 0 \quad (5.5.5)$$

$$r(T) = \frac{5gT^2 \sin \alpha_s}{14\alpha_s^2} - \frac{5gT^2 \cos \alpha_s}{14\alpha_s} + \dot{r}(0)T \quad (5.5.6)$$

The switching algorithm is thus modified so that the decision to switch is based on when this $r(T)$ crosses the switching plane. When this switch is done correctly, the ball will be on its way to the origin of the phase portrait. Eventually, the system will settle so that a switch will be made at the origin. When this is the case, the bounds of the phase portrait are:

$$r(0) = r(T) = \pm \frac{5gT^2}{14\alpha_s} \left(\frac{\sin\alpha_s}{\alpha_s} - \cos\alpha_s \right) \quad (5.5.7)$$

$$\dot{r}(T/2) = \pm \frac{5gT}{14\alpha_s} (1 - \cos\alpha_s) \quad (5.5.8)$$

From the equation $\dot{r}(t) = -\frac{5}{7}g\sin(\frac{2\alpha_s t}{T} - \alpha_s)$, one can see that the maximum of \dot{r} is at $t = \frac{T}{2}$. The phase portrait settles here because $r(t) = 0$ is the only point where $-\dot{r}(t) = \dot{r}(t)$ (the evolution of \dot{r} after a switch) and the switching planes intersect. If the initial conditions are $\dot{r}(0) = 0$ and $r(0) = 0$ then after a switch, the ball will end up at $\dot{r} = 0$ and $r = \frac{5gT^2}{14\alpha_s} \left(\frac{\sin\alpha_s}{\alpha_s} - \cos\alpha_s \right)$. If these coordinates are, in turn, used as initial conditions, the ball will end up back at the origin, according to Equations 3.2.42. This forms a limit cycle.

There is no chance that the system will “switch early” because the switching algorithm does not allow for it. If, however, due to a lengthy switching delay, the system cannot follow the switching curve to the origin, then the trajectory of the ball on the phase portrait will be characterized by a series of orbits offset by $\dot{r}(0)T$. These orbits will continue to drift by this amount until a switch will bring the ball back to the origin. If this is the case, the orbits will slightly exceed the aforementioned bounds in the \dot{r} direction. The bounds in the r direction should continue to hold.

The resulting Matlab simulations are shown in Figures 5.5.5 and 5.5.6. The bounding region developed from the instantaneous switching with minimum dwell time assumption are superimposed for comparison.

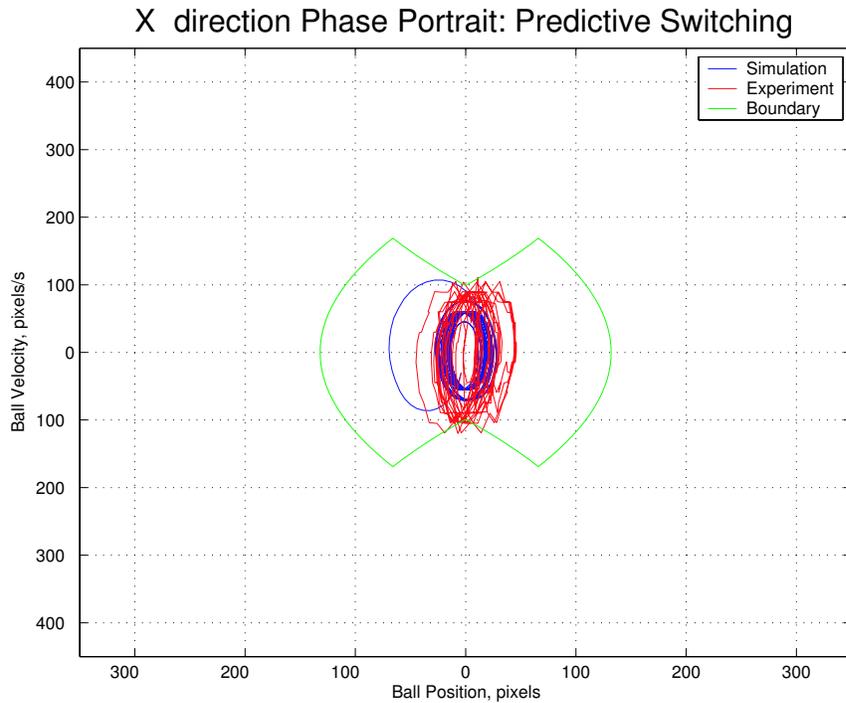


Figure 5.5.5: Simulated phase portrait of ball trajectory in the x direction with stability region under assumption of non-instantaneous switching with minimum dwell time

5.5.4 Implementation of Predictive Switching and Experimental Results

After observing the results of the predictive switching algorithm employed in the theoretical simulation, we applied the same algorithm to the experiment. For the experiment, the magnitudes of the angles of the platform were chosen to be 25 counts. At 4000 counts per revolution, a 25 count angle is equivalent to a 2.25 degree angle. The amount of time that it took to switch the plate was assumed to be 0.781 seconds. This was computed by averaging measured switching times in the experimental data.

In Figure 5.5.7 one can clearly see that predictive switching contained the ball to

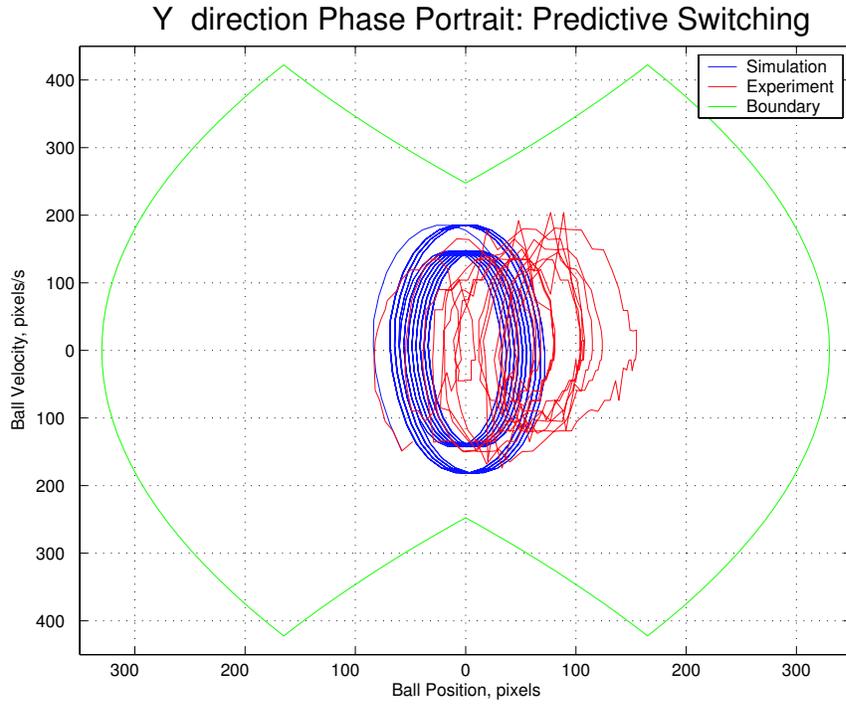


Figure 5.5.6: Simulated phase portrait of ball trajectory in the y direction with stability region under assumption of non-instantaneous switching with minimum dwell time a smaller region near the origin. Similarly, the phase portraits, position time response and velocity time response plots also show a smaller containment region consistent with what was expected from simulation results (see Figures 5.5.8 through 5.5.13).

For the experimental data, the velocity and acceleration were calculated simply:

$$Velocity = \frac{\Delta Position}{\Delta Time} \quad (5.5.9)$$

$$Acceleration = \frac{\Delta Velocity}{\Delta Time} \quad (5.5.10)$$

Sensor noise and or slight errors in the computation of the location of the ball were thus magnified in the velocity and acceleration data (see Figures 5.5.12, 5.5.13, 5.5.10, and 5.5.11). Equation 3.4.5 shows that for $\phi_1 = \pm\alpha_1$ and $\phi_2 = \pm\alpha_2$ for some fixed,

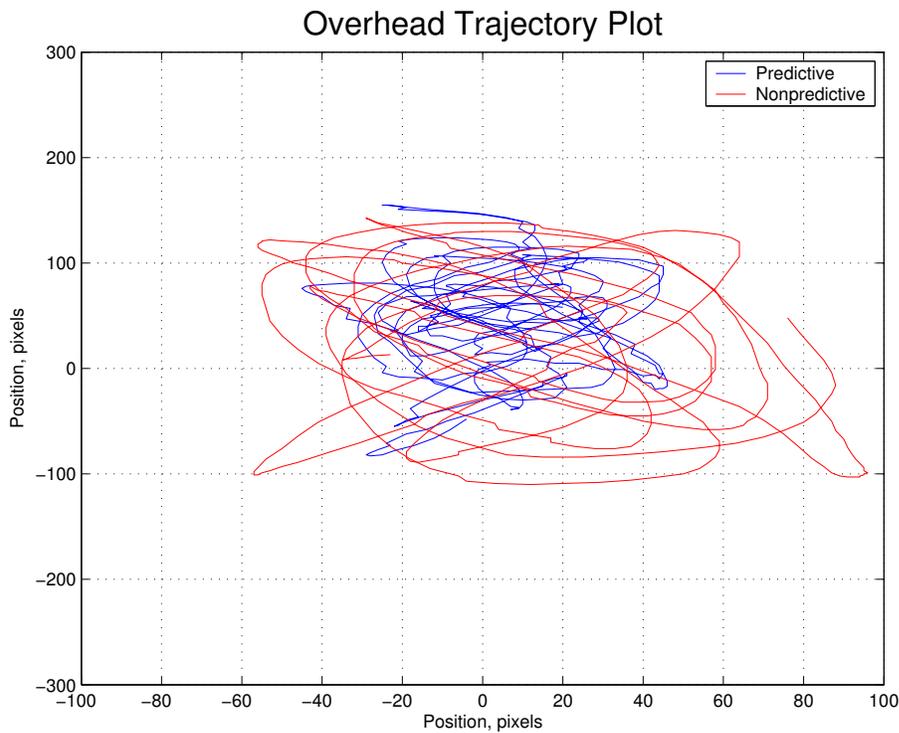


Figure 5.5.7: Overhead x-y trajectory plot of the ball with and without predictive switching.

arbitrary α_1 and α_2 , the acceleration plots should be square in shape, the velocity plots should be triangular, and the position plots should be parabolic. The curves of the phase portrait also appear parabolic, as predicted in Equation 5.4.28. These predictions appear to hold true although the noise in the velocity and acceleration data makes this harder to see.

Since the only differences between the experimental results for the predictive switching and non-predictive switching plots are the switching times, the magnitudes of the acceleration plots should be similar. The slopes of the velocity plots should also be similar. These predictions also appear to hold true.

The acceleration of gravity in pixels/s² was calculated to be 7717 pixels/s² in the

x direction and 19291 pixels/s² in the y-direction, based on the size of the plate and the effective resolution of the camera. In Equation 3.4.5, based on these values for the acceleration of gravity, $\gamma_1 = \pm 540 \text{ pixels/s}^2$ and $\gamma_2 = \pm 216 \text{ pixels/s}^2$.

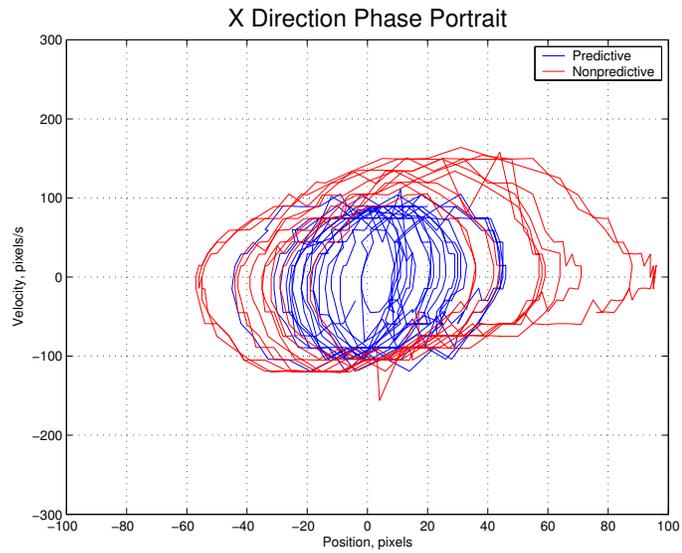


Figure 5.5.8: Phase portrait of ball trajectory in x direction with and without predictive switching

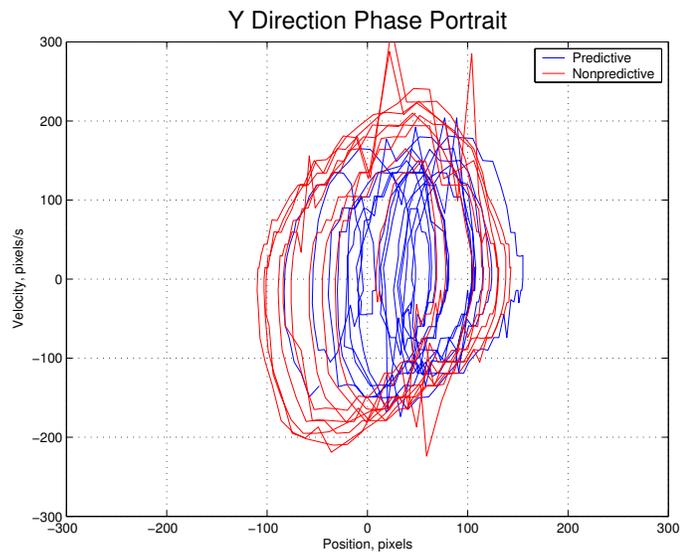


Figure 5.5.9: Phase portrait of ball trajectory in y direction with and without predictive switching

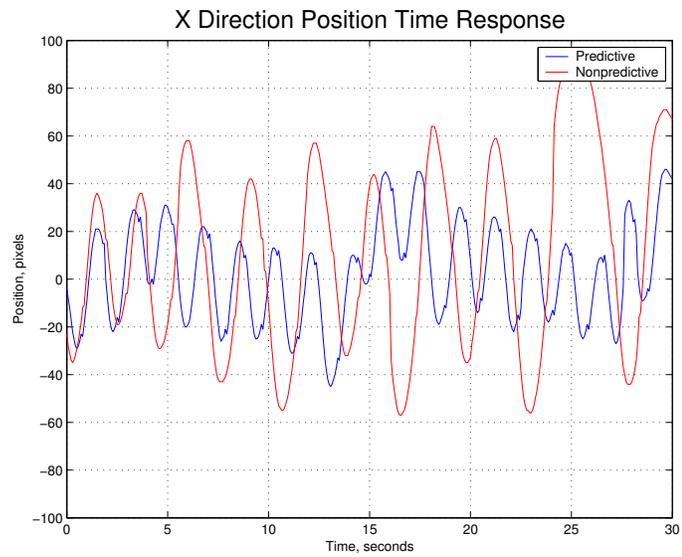


Figure 5.5.10: Position of ball in x direction with respect to time with and without predictive switching

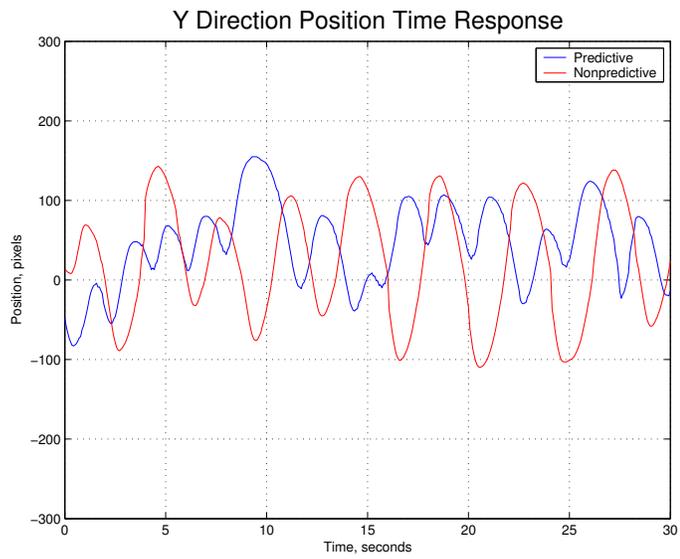


Figure 5.5.11: Position of ball in y direction with respect to time with and without predictive switching

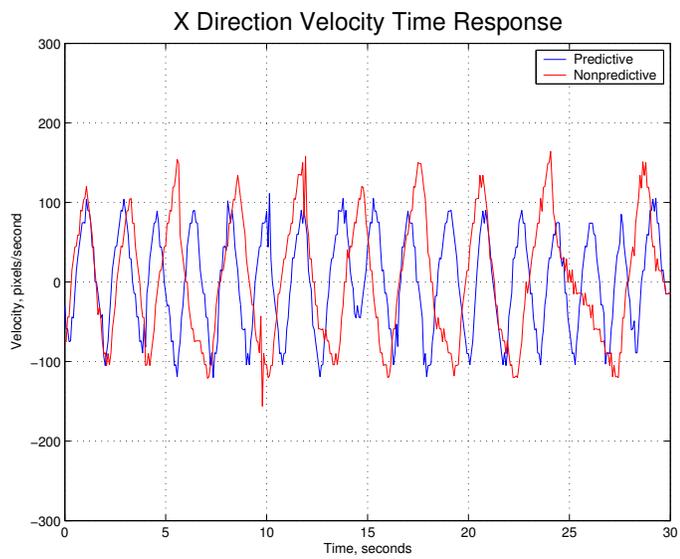


Figure 5.5.12: Velocity of ball in x direction with respect to time with and without predictive switching

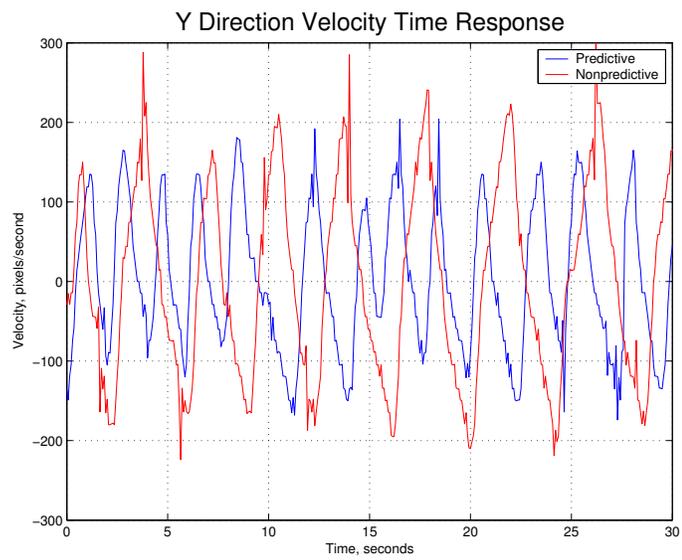


Figure 5.5.13: Velocity of ball in y direction with respect to time with and without predictive switching

Chapter 6

Conclusions and Future Work

We described the design and construction of a two degree of freedom ball-on-plate system. The task of balancing the ball on the rotating plate was accomplished using a language-based controller that transmits controls rather than continuous or discrete time actuator signals. The controller relied on an overhead camera to monitor the state of the ball and chose controls from a small finite set. This had the result of limiting the plate to a choice of two fixed angles for each of its degrees of freedom. The outward simplicity of the controller was a compromise that resulted in lower precision for the position of the ball on the plate. The use of language-based control implied a type of controller quantization that did not allow the system to be asymptotically stable but did allow us to contain the state of the ball within a small region on the plate. This region depends on: *i*) the magnitude of the rotation angles of the plate and *ii*) the amount of time that it took to switch between plate angles. We showed that the ball could not be stabilized using open-loop oscillatory inputs or traditional observer-based or feedback-based methods, and therefore a limited attention scheme was the only reasonable alternative. We presented simulations of the resulting closed loop system under a variety of assumptions for the process of switching between controller commands, and compared the results

with the experimental data.

It would be interesting to generalize the proposed control strategy to a broader class of dynamical systems and explore other language-based control strategies. We will seek to expand our result on containment and find what sufficient conditions a linear system must satisfy if it is to also satisfy the assumptions of our main theorem. Other important, but open as of yet questions involve optimizing the set of language-based commands used for control and expanding the idea of language-based control to include estimation problems.

Appendix A

Mechanical Linkage Kinematics

One of the axes of rotation of the plate is connected to a motor via a mechanical linkage, while the other is connected to a second motor via a chain drive. The mechanical linkage is unique in that its members are not confined to a single plane. Two of the joints, A and B (see Figure A.0.1), have two degrees of freedom. These added degrees of freedom add significant complexity to the kinematics of the linkage. This diagram is superimposed over the actual linkage in the experiment in Figure A.0.2.

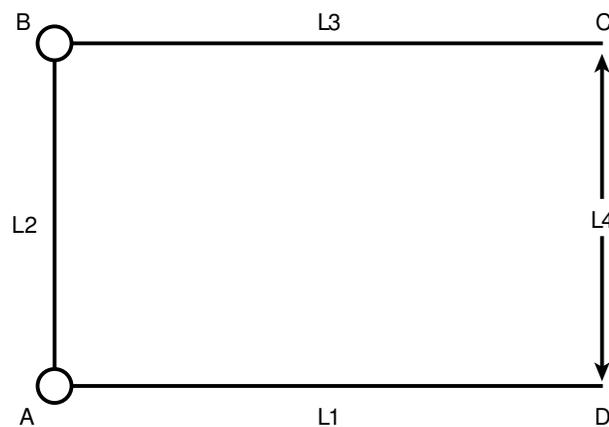


Figure A.0.1: Diagram of inner linkage of plate assembly

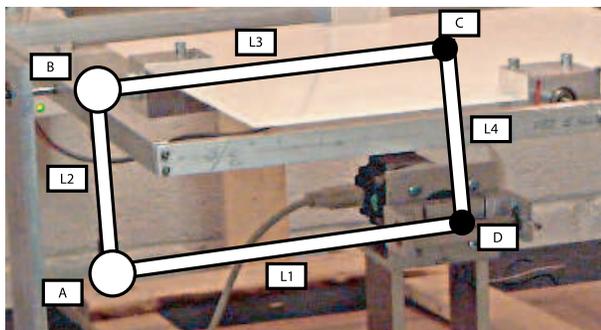


Figure A.0.2: Diagram of inner linkage of plate assembly superimposed over a photograph of the experimental assembly

Let a side view of the linkage be described by a rectangle, ABCD, as shown in Figure A.0.1.

Point C is the center of the plate on which the ball rolls. Point D is the location of the motor shaft, directly below point C. Point B where the linkage meets the edge of the plate, and point A is an intermediate joint between arms of the linkage. The lengths of the members are $l_1 = DA$, $l_2 = AB$, $l_3 = BD$ and $l_4 = CD$.

Let $\{\hat{i}, \hat{j}, \hat{k}\}$ be the unit vectors of an orthonormal, space-fixed coordinate frame whose origin is D, with x pointing to the right, y pointing into the page, and z pointing upwards. The coordinate system is shown in Figure A.0.3. Let (x_a, y_a, z_a) denote the coordinates of A, and so on for B, C and D. Points C and D are fixed, and the link DA can only move within the XZ plane. Point B is thus confined to a set of points which can be described by the intersection of two spheres centered at A and C as seen in Figure A.0.3. The equations for the spheres are:

$$(x_b - x_a)^2 + (y_b - y_a)^2 + (z_b - z_a)^2 = l_2^2 \quad (\text{A.0.1})$$

$$(x_b - x_c)^2 + (y_b - y_c)^2 + (z_b - z_c)^2 = l_3^2 \quad (\text{A.0.2})$$

where $x_c = 0$, $y_c = 0$, $z_c = l_4$, $y_a = 0$, $l_1 = l_3$, $l_2 = l_4$, and $y_a = 0$ so that

$$(x_b - x_a)^2 + y_b^2 + (z_b - z_a)^2 = l_2^2 \quad (\text{A.0.3})$$

$$x_b^2 + y_b^2 + (z_b - l_2)^2 = l_3^2 \quad (\text{A.0.4})$$

The locations of C and D are known and because link DA is directly connected to the motor underneath the plate, the location of A can be readily computed. Let θ and ϕ_2 be the motor input angles and let ϕ_1 and ϕ_2 be the angles of rotation about the \hat{j} and the \hat{i} axes, respectively. Let α_2 be the angle between the link AB and the \hat{k} axis (see Figure A.0.3). Then we can write:

$$\sin\phi_1 = \frac{z_b - l_2}{l_1} \quad (\text{A.0.5})$$

$$\sin\alpha_2 = \frac{x_a - x_b}{l_2} \quad (\text{A.0.6})$$

$$x_a = -l_1 \cos\theta \quad (\text{A.0.7})$$

$$z_a = l_1 \sin\theta \quad (\text{A.0.8})$$

$$x_b = -l_2 \sin\alpha_2 + x_a \quad (\text{A.0.9})$$

$$y_b = l_2 \cos\alpha_2 \sin\phi_2 \quad (\text{A.0.10})$$

$$z_b = l_2 \cos\alpha_2 \cos\phi_2 + z_a \quad (\text{A.0.11})$$

Solving Equations A.0.3 through A.0.8 and A.0.10 simultaneously, we have:

$$\begin{aligned} \phi_1(\theta, \phi_2) = & \sin^{-1}(2\cos\theta\sqrt{\eta} + l_2(l_2 - l_1\sin\theta)^2(-l_1(3 + \cos 2\theta) + 4l_2\cos^2\phi_2\sin\theta - \\ & 2l_1\cos 2\phi_2\sin^2\theta) \end{aligned} \quad (\text{A.0.12})$$

where

$$\begin{aligned} \eta = & l_2^4 \cos^2\phi_2 (l_2 - l_1 \sin\theta)^2 (l_1^2 - 2l_2^2 + (l_1^2 + 2l_2^2) \cos 2\phi_2 - l_1^2 (-3 + \cos 2\phi_2) \cos 2\theta + \\ & 8l_1 l_2 \sin^2\phi_2 \sin\theta \end{aligned} \quad (\text{A.0.13})$$

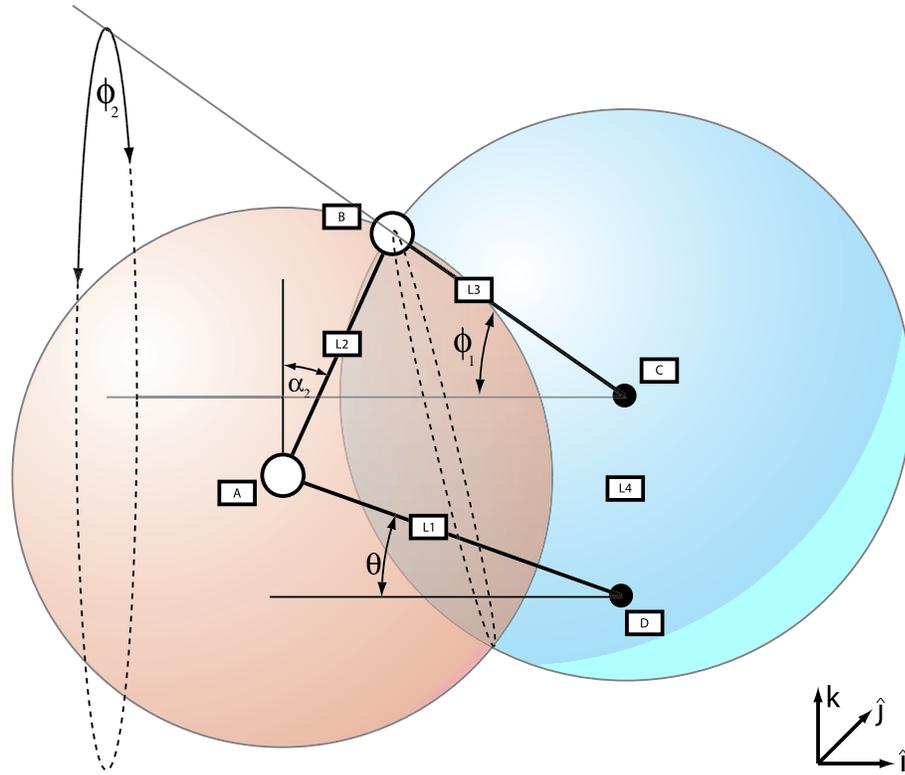


Figure A.0.3: Three-dimensional depiction of linkage annotated with angles θ , ϕ_1 , ϕ_2 , and α_2 . The dotted line between the spheres indicates the possible path of travel of point B.

Thus, given motor angles ϕ_2 and θ , we have the angles of the plate, ϕ_1 and ϕ_2 . If Equation A.0.12 is linearized about $\phi_2 = \theta = 0$:

$$\phi_1 \approx \theta \tag{A.0.14}$$

which justifies our assumption.

Bibliography

- [1] Al-Rahmani, Hamed and Gene F. Franklin, "Linear Periodic Systems: Eigenvalue Assignment Using Discrete Periodic Feedback," *IEEE Transactions on Automatic Control*, vol. 34, no. 1, IEEE, Piscataway, NJ, 1989, pp. 99-103.
- [2] Andersson, S and D. Hristu. "Stochastic Language-Based Motion Control," *Proceedings of the 42nd IEEE Conference on Decision and Control*, vol. 4, IEEE, Piscataway, NJ, 2003, pp. 3313-3318.
- [3] Bellman, R., J. Bentsman and S. Meerkov. "Vibrational Control of Nonlinear Systems: Vibrational Controllability and Transient Behavior," *IEEE Transactions on Automatic Control*, vol. 31, no. 8, IEEE, Piscataway, NJ, 1986, pp. 717-724.
- [4] Bellman, R., J. Bentsman and S. Meerkov. "Vibrational Control of Nonlinear Systems: Vibrational Stability," *IEEE Transactions on Automatic Control*, vol. 31, no. 8, IEEE, Piscataway, NJ, 1986, pp. 710-716.
- [5] Bemporad, A, G. Ferrari-Trecate and R. Morari. "Observability and Controllability of Piecewise Affine and Hybrid Systems," *IEEE Transactions on Automatic Control*, vol. 45, no. 10, IEEE, Piscataway, NJ, 2000, pp. 1864-1876.

- [6] Boutayeb, M. and M. Darouach. "Observers for Discrete-time Systems with Multiple Delays," *IEEE Transactions on Automatic Control*, vol. 46, no. 5, IEEE, Piscataway, NJ, 2001, pp. 746-750.
- [7] Branicky, M.S. "Multiple Lyapunov Functions and Other Analysis Tools for Switched and Hybrid Systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, IEEE, Piscataway, NJ, 1998, pp. 475-482.
- [8] Brockett, R.W. "On the Computer Control of Movement," *IEEE International Conference on Robotics and Automation*, vol 1, IEEE, Piscataway, NJ, 1988, pp. 534-540.
- [9] Brockett, R.W. "Stabilization of Motor Networks," *Proceedings of the 1995 IEEE 34th Conference on Decision and Control*, IEEE, Piscataway, NJ, 1995, pp. 1484-1488.
- [10] Brockett, R.W. and D. Liberzon "Quantized Feedback Stabilization of Linear Systems," *IEEE Transactions on Automatic Control*, vol. 45, no. 7, IEEE, Piscataway, NJ, 2000, pp. 1279-1289.
- [11] Butikov, Eugene I. "On the Dynamic Stabilization of an Inverted Pendulum", *Am. J. Phys.*, vol. 69, no. 7, 2001, pp. 755-768.
- [12] Clements, K., and R. Haddad. "Approximate Estimation for Systems with Quantized Data," *IEEE Transactions on Automatic Control*, vol. 17, no. 2, IEEE, Piscataway, NJ, 1972, pp. 235-239.
- [13] Colaneri, P., R. Scattolini and N. Schiavoni. "Stabilization of Multirate Sampled-data Linear Systems," *Automatica*, vol. 26, no. 2, pp. 377-380.

- [14] Delchamps, D. F. "Stabilizing a Linear System with Quantized State Feedback," *IEEE Transactions on Automatic Control*, vo. 35, no. 8, IEEE, Piscataway, NJ, 1990, pp. 916-924.
- [15] Dimeo, Robert M. and Stelios C. A. Thomopoulos. "Novel Control of an Inverted Pendulum," *Proceedings of the American Control Conference*, 1994.
- [16] Egerstedt, M. "Linguistic Control of Mobile Robots," *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, 2001, pp. 877-882.
- [17] Egerstedt, M and R. Brockett. "Feedback can Reduce the Specification Complexity of Motor Programs," *Proceedings of the 40th IEEE Conference on Decision and Control*, vol. 2, IEEE, Piscataway, NJ, 2001, pp. 1651-1656.
- [18] Elia, Nicola and Sanjoy K. Mitter. "Quantization of Linear Systems," *Proceedings of the 1999 IEEE Conference on Decision and Control*, vol. 4, IEEE, Piscataway, NJ, 1999, pp. 3428-3422.
- [19] Elia, Nicola and Sanjoy K Mitter. "Stabilization of Linear Systems with Limited Information," *IEEE Transactions on Automatic Control*, vol. 46, no. 9, IEEE, Piscataway, NJ, 2001, pp. 1384-1400.
- [20] Ezzine, J. and A.H. Haddad. "Controllability and Observability of Hybrid Systems," *International Journal of Control*, vol. 49, no. 6, 1999, pp. 2045-2055.
- [21] Fagnani, F. and S. Zampieri. "Stability Analysis and Synthesis for Scalar Linear Systems with a Quantized Feedback," *Proceedings of the 40th IEEE Conference on Decision and Control*, vol. 3, IEEE, Piscataway, NJ, 2001, pp. 2204-2209.

- [22] Feng, Zhao-Shu, Feng, Yong-Qing Liu and Feng-Wei Guo. "Mean-square Practical Stability of Stochastic Large-scale Dynamical Systems," *Proceedings of the 1991 IEEE International Conference on Decision Aiding for Complex Systems*, vol. 1, IEEE, Piscataway, NJ, 1991, pp. 517-520.
- [23] Fenn, Julia G., Derek A. Bayne, and Bruce D. Sinclair. "Experimental Investigation of the 'Effective Potential' of an Inverted Pendulum," *Am. J. Phys.*, vol. 66, no. 11, 1998, pp. 981-984.
- [24] Frazzoli, E., M.A. Dahleh, and E. Feron. "Robust Hybrid Control for Autonomous Vehicle Motion Planning," *Proceedings of the 39th IEEE Conference on Decision and Control*, vol. 1, IEEE, Piscataway, NJ, 2000, pp. 821-826.
- [25] Habets, L.C.G.J.M. and J.H. van Schuppen. "A Controllability Result of Piecewise-Linear Hybrid Systems," *Proceedings of the European Control Conference*, 2001.
- [26] Hauser, J., S. Sastry, and P. Kokotovic. "Nonlinear Control via Approximate Input-output Linearization: the Ball and Beam Example," *IEEE Transactions on Automatic Control*, vol. 37, no. 3, IEEE, Piscataway, NJ, 1992, pp. 392-398.
- [27] Hristu, D. "Feedback Control Systems as Users of a Shared Network: Communication Sequences that Guarantee Stability," *Proceedings of the 2001 IEEE Conference on Decision and Control*, vol. 4, IEEE, Piscataway, NJ, 2001, pp. 3631-3636.
- [28] Hristu, D. "Generalized Inverses for Finite Horizon Tracking," *Proceedings of the 1999 IEEE Conference on Decision and Control*, vol. 2, IEEE, Piscataway, NJ, 1999, pp. 1397-1402.

- [29] Hristu, D., M. Egerstedt and P.S. Krishnaprasad. "On the Structural Complexity of the Motion Description Language MDLe," *Proceedings of the 42nd IEEE Conference on Decision and Control*, vol. 4, IEEE, Piscataway, NJ, 2003, pp. 3360-3365.
- [30] Huffman, D. A. "A Method for the Construction of Minimum-Redundancy Codes," *Proceedings of the Institute of Radio Engineering*, vol. 40, no. 9, 1952, pp. 1098-1101.
- [31] Krishnaprasad, P.S., F. Zhang, S. Andersson, L. D'Anna and P. Sodre. "The MDLe Engine: A Software Tool for Hybrid Motion Control," ISR Technical Report TR2000-54, University of Maryland, 2000.
- [32] Kabamba, P.T. "Monodromy Eigenvalue Assignment in Linear Periodic Systems," *IEEE Transactions on Automatic Control*, vol. 31, no. 10, IEEE, Piscataway, NJ, 1986, pp. 950-952.
- [33] Kokotovic, Petar K. "The Joy of Feedback: Nonlinear and Adaptive," *Control Systems Magazine*, vol. 12, no. 3, IEEE, Piscataway, NJ, 1992, pp. 7-17.
- [34] Kolmanovsky, I. and N.H. McClamroch. "Developments in nonholonomic control problems," *Control Systems Magazine*, vol. 15, no. 6, IEEE, Piscataway, NJ, 1995, pp. 20-36.
- [35] Lakshmikantham, V, V.M. Mastrosov, and S. Sivasundaram. *Vector Lyapunov Functions and Stability Analysis of Nonlinear Systems*, Kluwer Academic Publishers: Dordrecht, 1991.
- [36] Manikonda, V., P.S. Krishnaprasad and J. Hendler. "A Motion Description Language and a Hybrid Architecture for Motion Planning With Nonholonomic

- Robots,” *Proceedings of the 1995 IEEE International Conference on Robotics and Automation*, vol. 2, pp. 2021-2028, IEEE, Piscataway, NJ, 1995.
- [37] Manikonda, V., P.S. Krishnaprasad and J. Hendler. “Languages, Behaviors, Hybrid Architectures, and Motion Control,” *Mathematical Control Theory*, Baillieul and Willems eds., Springer-Verlag, New York, 1998.
- [38] Meerkov, S. “Principle of Vibrational Control: Theory and Applications,” *IEEE Transactions on Automatic Control*, vol. 25, no. 4, IEEE, Piscataway, NJ, 1980.
- [39] Michel, A.N. “Quantitative Analysis of Simple and Interconnected Systems: Stability, Boundedness and Trajectory Behavior,” *IEEE Transactions on Circuit Theory*, vol. 17, no. 3, IEEE, Piscataway, NJ, 1970, pp. 292-301.
- [40] Mohrenschildt, M.V. “Hybrid systems: Solutions, Stability, Control,” *Proceedings of the 2000 American Control Conference*, vol. 1, no. 6, 2000, pp. 692-698.
- [41] Moreau, L. and D. Aeyels. “Practical Stability for Systems Depending on a Small Parameter,” *Proceedings of the 37th IEEE Conference on Decision and Control*, IEEE, Piscataway, NJ, 1998, pp. 1428-1433.
- [42] Nerode, A and W. Kohn. “Models for Hybrid Systems: Automata Topologies, Controllability, Observability,” *Hybrid Systems*, Berlin, Germany, 1991, pp. 10-12.
- [43] Olfati-Saber, R. and A. Megretski. “Controller Design for the Beam-and-Ball System,” *Proceedings of the 37th IEEE conference on Decision and Control*, IEEE, Piscataway, NJ, 1998, pp. 4555-4560.

- [44] Phelps, F.M. and J.H. Hunter, Jr. "An analytical solution of the inverted pendulum," *Am. J. Phys.*, vol. 33, 1965, pp. 285-295.
- [45] Schuppen, J.H. "A Sufficient Condition for Controllability of a Class of Hybrid Systems," *The First International Workshop on Hybrid Systems: Computation and Control*, Berkeley, CA, 1998.
- [46] Shannon, S.E. "A Mathematica Theory of Communication," *Bell Systems Technical Journal*, vol. 27, 1948, pp. 379-423, 623-656.
- [47] Walsh, Gregory C. and Hong Ye. "Scheduling of Networked Control Systems," *IEEE Control Systems Magazine*, vol. 21, no. 1, IEEE, Piscataway, NJ, 2001, pp. 57-65.
- [48] Wong, Wing Shing and R.W. Brockett. "Systems with Finite Communication Bandwidth Constraints - Part II: Stabilization with Limited Information Feedback," *IEEE Transactions on Automatic Control*, vol. 44, no. 5, IEEE, Piscataway, NJ, 1999, pp. 1049-1053.
- [49] Wong, Wing Shing and R.W. Brockett. "Systems with Finite Communication Bandwidth Constraints - Part I: State Estimation Problems," *IEEE Transactions on Automatic Control*, vol. 42, no. 9, IEEE, Piscataway, NJ, 1997, pp. 1294-1299.
- [50] Zhai, Guisheng and A.N. Michel. "On Practical Stability of Switched Systems," *Proceedings of the 41st IEEE Conference on Decision and Control*, vol. 3, IEEE, Piscataway, NJ, 2002, pp. 3488-3493.
- [51] Zhang, Wei, M.S. Branicky and S.M. Phillips. "Stability of Networked Control Systems," *IEEE Control Systems Magazine*, vol. 21, no. 1, IEEE, Piscataway, NJ, 2001, pp. 84-99.