LINEAR POSITION CONTROL IN THE PRESENCE OF NON-LINEAR FRICTION

Frank J. Breau III
University of Rhode Island, frank.breau@navy.mil

Follow this and additional works at: https://digitalcommons.uri.edu/theses

Recommended Citation

This Thesis is brought to you for free and open access by DigitalCommons@URI. It has been accepted for inclusion in Open Access Master's Theses by an authorized administrator of DigitalCommons@URI. For more information, please contact digitalcommons-group@uri.edu.
LINEAR POSITION CONTROL IN THE PRESENCE OF NON-LINEAR FRICTION

BY

FRANK J. BREAU III

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

UNIVERSITY OF RHODE ISLAND

2011
ABSTRACT

This thesis presents the design and simulation of a friction compensation technique for position control systems. Linear techniques are used to implement an “inner loop” velocity tracking system to compensate for friction-induced disturbances. Intended to mimic the effect of an analog tachometer feedback loop on a servo drive motor, inner loops employing proportional feedback, integral feedback, and observer based feedback are explored. Simulation results indicate that the observer-based feedback loop is most effective at friction compensation. Hardware demonstrations substantiate the simulated results.
ACKNOWLEDGMENTS

Completing this thesis marks the conclusion of formal studies and the begin-
ning of a lifetime of discovery and development. It also represents the fulfilment
of a major personal goal. As such, some thanks are in order. None of this work
would have been possible without:

Dr. Richard Vaccaro, my major professor, who has provided countless hours
of guidance and support to this project. His tireless work has made this experience
infinitely better.

Professors in the Electrical Engineering department, who have all gone to
great lengths to offer interesting and relevant classes that accommodated the er-
ratic schedules of a group of working engineers. In particular, Dr. Yang and Dr.
Kumaresan traveled to Newport multiple times a week to offer evening classes.

Collaboration between the University of Rhode Island and the Naval Undersea
Warfare Center, under the supervision of Dr. Bud Vincent, provided the encour-
agement and opportunity to begin graduate studies while working full time.

My friends and family, who have supported me during this process and have
tolerated my absences throughout it; my parents, who have done so much to en-
courage me to succeed; and Heather, who motivates me to live the best life I
possibly can.

Thank you all!
# TABLE OF CONTENTS

ABSTRACT ................................................................. ii

ACKNOWLEDGEMENTS ....................................................... iii

TABLE OF CONTENTS ....................................................... iv

LIST OF FIGURES .......................................................... vi

CHAPTER

1 Introduction ............................................................... 1
  1.1 The Friction Problem ................................................. 1
  1.2 PID Control .......................................................... 6
  1.3 Model-Based Approaches to Friction Compensation ............... 10
  1.4 Summary of Findings ............................................... 13

List of References ....................................................... 13

2 Algorithm Development ................................................. 17
  2.1 Introduction ........................................................ 17
  2.2 Position Controller with no Friction Compensation .............. 18
  2.3 Inner-Loop Velocity Tracking: Proportional Control ............ 25
  2.4 Inner-Loop Tracking System: Integral Control ................. 30
  2.5 An Observer for the Inner Loop ................................... 34
  2.6 Linear Model Development for Overall Tracking System ....    37

List of References ....................................................... 37

3 Simulation Development ................................................. 38
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Hardware Plant Model</td>
<td>38</td>
</tr>
<tr>
<td>3.2</td>
<td>Friction Model Development</td>
<td>40</td>
</tr>
<tr>
<td>3.3</td>
<td>Tracking System Simulations</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>List of References</td>
<td>51</td>
</tr>
<tr>
<td>4</td>
<td>Simulation Examples</td>
<td>54</td>
</tr>
<tr>
<td>4.1</td>
<td>Step Inputs</td>
<td>54</td>
</tr>
<tr>
<td>4.2</td>
<td>Sinusoids</td>
<td>56</td>
</tr>
<tr>
<td>4.3</td>
<td>Discussion</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>List of References</td>
<td>61</td>
</tr>
<tr>
<td>5</td>
<td>Hardware Demonstration</td>
<td>77</td>
</tr>
<tr>
<td></td>
<td><strong>APPENDIX</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bessel Roots</td>
<td>89</td>
</tr>
<tr>
<td></td>
<td>List of References</td>
<td>89</td>
</tr>
<tr>
<td></td>
<td><strong>BIBLIOGRAPHY</strong></td>
<td>90</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Simulink model of a standard state-feedback tracking system</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Position tracking with no friction</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Example linear friction model</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>Position tracking with linear friction</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>An example non-linear friction model</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>Position tracking with non-linear friction</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>Tracking a sinusoid with non-linear friction</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>Position controller without model based friction compensation</td>
<td>7</td>
</tr>
<tr>
<td>9</td>
<td>PID gains in a position tracking system</td>
<td>7</td>
</tr>
<tr>
<td>10</td>
<td>Control loop with model based friction compensation</td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>Standard state-space position tracking system</td>
<td>18</td>
</tr>
<tr>
<td>12</td>
<td>Equivalent depiction of Fig. 11 with state variables indicated</td>
<td>19</td>
</tr>
<tr>
<td>13</td>
<td>Replacing the analog plant with its zero order hold equivalent</td>
<td>22</td>
</tr>
<tr>
<td>14</td>
<td>Simple tachometer feedback inner loop</td>
<td>26</td>
</tr>
<tr>
<td>15</td>
<td>An equivalent depiction of the inner loop in Fig. 14</td>
<td>26</td>
</tr>
<tr>
<td>16</td>
<td>Transfer function for the hardware plant</td>
<td>27</td>
</tr>
<tr>
<td>17</td>
<td>Inner loop from Fig. 15 showing only velocity</td>
<td>27</td>
</tr>
<tr>
<td>18</td>
<td>Outer loop position tracking system for the inner loop in Fig. 15</td>
<td>29</td>
</tr>
<tr>
<td>19</td>
<td>Velocity feedback with integral control inner loop</td>
<td>31</td>
</tr>
<tr>
<td>20</td>
<td>Outer loop position tracking system for inner loop in Fig. 19</td>
<td>34</td>
</tr>
<tr>
<td>21</td>
<td>Inner loop velocity tracking system with observer</td>
<td>35</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>22</td>
<td>Equivalent depiction of Fig. 21</td>
<td>35</td>
</tr>
<tr>
<td>23</td>
<td>Hardware Plant Model</td>
<td>39</td>
</tr>
<tr>
<td>24</td>
<td>Block diagram for $\frac{1}{(s+C)}$</td>
<td>39</td>
</tr>
<tr>
<td>25</td>
<td>Hardware Plant Model: Step 3</td>
<td>40</td>
</tr>
<tr>
<td>26</td>
<td>Hardware Plant Model: Step 4</td>
<td>40</td>
</tr>
<tr>
<td>27</td>
<td>Friction model 1</td>
<td>41</td>
</tr>
<tr>
<td>28</td>
<td>Friction model 2</td>
<td>43</td>
</tr>
<tr>
<td>29</td>
<td>Friction model 3</td>
<td>45</td>
</tr>
<tr>
<td>30</td>
<td>Friction model 4</td>
<td>47</td>
</tr>
<tr>
<td>31</td>
<td>Friction models 1-4 overlaid</td>
<td>48</td>
</tr>
<tr>
<td>32</td>
<td>Friction model Simulink block</td>
<td>49</td>
</tr>
<tr>
<td>33</td>
<td>Second-order hardware plant model</td>
<td>49</td>
</tr>
<tr>
<td>34</td>
<td>Position tracking system without friction compensation</td>
<td>50</td>
</tr>
<tr>
<td>35</td>
<td>Inner loop with tachometer feedback</td>
<td>50</td>
</tr>
<tr>
<td>36</td>
<td>Outer loop for inner loop with tachometer</td>
<td>51</td>
</tr>
<tr>
<td>37</td>
<td>Inner loop with integrator</td>
<td>51</td>
</tr>
<tr>
<td>38</td>
<td>Outer loop for inner loop with integrator</td>
<td>52</td>
</tr>
<tr>
<td>39</td>
<td>Inner loop with full order observer</td>
<td>52</td>
</tr>
<tr>
<td>40</td>
<td>Outer loop for inner loop with observer</td>
<td>53</td>
</tr>
<tr>
<td>41</td>
<td>Tracking 0.2M step inputs</td>
<td>55</td>
</tr>
<tr>
<td>42</td>
<td>Tracking 0.2M step inputs</td>
<td>63</td>
</tr>
<tr>
<td>43</td>
<td>Tracking 0.1m step inputs</td>
<td>64</td>
</tr>
<tr>
<td>44</td>
<td>Tracking 0.1m step inputs</td>
<td>65</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>45</td>
<td>Tracking 0.05m step inputs</td>
<td>66</td>
</tr>
<tr>
<td>46</td>
<td>Tracking 0.05m step inputs</td>
<td>67</td>
</tr>
<tr>
<td>47</td>
<td>Tracking 0.02m step inputs</td>
<td>68</td>
</tr>
<tr>
<td>48</td>
<td>Tracking 0.02m step inputs</td>
<td>69</td>
</tr>
<tr>
<td>49</td>
<td>Proportional control with friction models 1-4</td>
<td>70</td>
</tr>
<tr>
<td>50</td>
<td>Proportional control using various inner loop feedback gains</td>
<td>70</td>
</tr>
<tr>
<td>51</td>
<td>Tracking 0.2m sinusoidal inputs</td>
<td>71</td>
</tr>
<tr>
<td>52</td>
<td>Tracking 0.2m sinusoidal inputs</td>
<td>72</td>
</tr>
<tr>
<td>53</td>
<td>Tracking 0.02m sinusoidal inputs</td>
<td>73</td>
</tr>
<tr>
<td>54</td>
<td>Tracking 0.02m sinusoidal inputs</td>
<td>74</td>
</tr>
<tr>
<td>55</td>
<td>Integral and observer based tracking systems</td>
<td>75</td>
</tr>
<tr>
<td>56</td>
<td>Integral and observer based tracking systems</td>
<td>76</td>
</tr>
<tr>
<td>57</td>
<td>Lab hardware used for control system demonstration</td>
<td>77</td>
</tr>
<tr>
<td>58</td>
<td>Hardware Plant block used for hardware demo</td>
<td>78</td>
</tr>
<tr>
<td>59</td>
<td>Top level Simulink input used for hardware demonstration</td>
<td>79</td>
</tr>
<tr>
<td>60</td>
<td>Tracking a 0.2m sinusoid with no friction compensation</td>
<td>80</td>
</tr>
<tr>
<td>61</td>
<td>Tracking a 0.1m sinusoid with no friction compensation</td>
<td>81</td>
</tr>
<tr>
<td>62</td>
<td>Tracking a 0.05m sinusoid with no friction compensation</td>
<td>82</td>
</tr>
<tr>
<td>63</td>
<td>Tracking a 0.2m sinusoid with tachometer feedback</td>
<td>82</td>
</tr>
<tr>
<td>64</td>
<td>Tracking a 0.1m sinusoid with tachometer feedback</td>
<td>83</td>
</tr>
<tr>
<td>65</td>
<td>Tracking a 0.05m sinusoid with tachometer feedback</td>
<td>83</td>
</tr>
<tr>
<td>66</td>
<td>Tracking a 0.02m sinusoid with tachometer feedback</td>
<td>84</td>
</tr>
<tr>
<td>67</td>
<td>Tracking a 0.1m sinusoid with integral control</td>
<td>84</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>68</td>
<td>Tracking a 0.05m sinusoid with integral control</td>
<td>85</td>
</tr>
<tr>
<td>69</td>
<td>Tracking a 0.02m sinusoid with integral control</td>
<td>85</td>
</tr>
<tr>
<td>70</td>
<td>Tracking a 0.01m sinusoid with integral control</td>
<td>86</td>
</tr>
<tr>
<td>71</td>
<td>Tracking a 0.1m sinusoid with observer based control</td>
<td>86</td>
</tr>
<tr>
<td>72</td>
<td>Tracking a 0.05m sinusoid with observer-based control</td>
<td>87</td>
</tr>
<tr>
<td>73</td>
<td>Tracking a 0.02m sinusoid with observer-based control</td>
<td>87</td>
</tr>
<tr>
<td>74</td>
<td>Tracking a 0.01m sinusoid with observer-based control</td>
<td>88</td>
</tr>
</tbody>
</table>
CHAPTER 1

Introduction

1.1 The Friction Problem

The ability to quickly and accurately track an input command is a primary requirement in the design of position control systems. External disturbances acting on such systems can compromise the accuracy and speed of tracking. One major contributor to performance issues in positioning and tracking systems is the friction force generated between mechanical system components. When uncompensated, friction can introduce highly nonlinear disturbance forces that act against the system output, making standard linear control techniques ineffective. Without an opposing force to negate the effects of friction, these non-linearities will lead to tracking accuracy problems, settling time issues, and other downstream performance problems. Compounding this issue is the fact that friction can vary as the result of many external parameters. Ambient conditions such as temperature and humidity, the lubrication between opposing surfaces, and the varying properties of aging materials can all affect the influence of friction on a system [1].

Friction compensation in electromechanical systems is a common problem that has been approached in many ways with varying degrees of success. In each instance, the friction compensation scheme must be tailored to the specific application for which it is intended.

In the case of a single-axis translational mechanical system (such as a motor driven cart), linear techniques can be used to control the system in the absence of friction or other non-linearities. In this example, the transfer function of the hardware plant is \( \frac{1.3}{s+1.3} \). There is no friction for which to compensate, so a simple linear state-feedback tracking system, shown in Fig. 1, is designed with a settling time of 1 second.
This tracking system is a standard state-space configuration, using a digital integrator and state feedback. The result is comparable to a PID control system [2] (see section 1.2).

When a step signal of height .02\(m\) is fed into the system input, the output responds as expected; reaching a position of .02\(m\) at time \(t = 1\) second.
The simplest friction models describe friction force as linear in proportion to velocity. A linear force acting against the positioning of this system will not significantly degrade its positioning accuracy or speed. Such a force can be modeled as proportional to velocity, and can be included in a linear model for the plant. However, even if the linear friction force is not included in the plant model, the linear tracking system will perform adequately with a linear disturbance acting against it.

![Figure 3. Example linear friction model](image)

The linear friction model may be adequate for high speed applications. The standard state-space tracking system compensates for positioning error, not directly for friction force. This is illustrated in Fig. 4.

However, friction is a highly complicated phenomenon dependent upon the physical properties of opposing surfaces and the surfaces' velocities relative to one another [3], [4]. When a more realistic model of friction is considered, such as the one shown in Fig. 5, problems with linear tracking systems begin to occur. At high velocities, the friction force generated is linearly proportional to velocity. At low velocities near zero, significant non-linearities exist. This model is further detailed.
in a subsequent chapter.

When the same linear techniques are used to control a system under the influence of such a non-linear disturbance, performance is unacceptable. This becomes particularly apparent at low velocities when system hardware operates in the most non-linear portion of the friction model. With increasingly precise input commands, the positioning system must spend more time working against the non-linear section. This comes at the expense of performance.

When the commanded input is bi-directional, friction-related problems are further exacerbated. This is due to velocity reversals imposed by the input command. This becomes particularly noticeable in the case of a sinusoid, which will impose two velocity reversals per cycle on the tracking system.

As can be seen in Fig. 7, friction has become a significant obstacle to satisfactory performance. A friction compensation scheme must be developed. This
thesis will explore the use of nested feedback loops in friction compensation. In particular, the effectiveness of an observer in the nested loop will be explored using standard linear techniques.
1.2 PID Control

The majority of control loops used in industry apply some form of PID (proportional, integral, derivative) control. In PID control, an error signal is generated through the feedback configuration by comparing the current state of the controlled variable to a reference input signal. Errors are compensated for by proportional, integral, and derivative gains. A simplified block diagram for a tracking system without model based friction compensation is given in Fig. 8. A more detailed block diagram for a control system that employs PID control is given in Fig. 9.

PID control is the most prevalent control strategy in current use. It has been estimated that over 90 percent of control loops apply some form of PID [5]. PID’s three-term functionality is well understood and documented, and is sufficient to solve many control problems in industrial applications. Tuning of the three PID gains will address both transient and steady-state response of the control loop.
Figure 8. Position controller without model based friction compensation

Significant research has been devoted to optimizing the PID tuning process in order to get the most from PID control.

The control signal $u$ produced by the PID controller can be described as the sum of the three gains $k_p$ (proportional), $k_i$ (integral), and $k_d$ (derivative) acting on an error signal, $E$. In the Laplace domain,

$$u(s) = (k_i \frac{1}{s} + k_p + k_ds)E(s)$$  \hspace{1cm} (1)$$

Figure 9. PID gains in a position tracking system

The three PID gains provide error compensation for different portions of the reference input signal. The proportional term provides control for error signals across the frequency band. Like its namesake, the proportional gain segment of a PID controller provides a control signal that is in direct proportion to the error signal $E(s)$ at all times. The proportional gain term provides control for the current error signal.
The integral gain term provides control for low frequency error signals. It will not be significant at higher frequencies. In this regard, it reduces the steady-state error of the control system but does not contribute significantly to transient response.

The derivative gain term provides control for high frequency error signals. It will not be significant at lower frequencies. In this regard, it increases the transient response of the control system. It is worth mentioning that the derivative gain is often perceived to have a destabilizing effect on the controller, and is often switched off completely [6]; although with proper tuning this effect can be mitigated.

One of the principle limitations to PID control is the condition that occurs when a large change in the reference signal happens and error is accumulated by the integrator during the change, known as integrator wind-up. Errors associated with integrator wind-up are eventually unwound, (or offset) by error in the opposite direction. The net effect on the control system is a significant overshoot and low frequency oscillation. This can potentially lead to controller instability [6].

Another drawback of PID control is that the derivative term has a large gain at high frequencies. When encountering a discontinuity in the reference signal (such as a step input), an ideal differentiator will generate an infinite control signal. This is generally mitigated by cascading a low pass filter with the differential gain term. A setpoint filter on the reference input can also be employed to reduce sensitivity to rapid input changes by weighting the input signal prior to the derivative term [7].

It is common to eliminate the derivative term altogether due to this destabilizing characteristic. This is known as PI control. In many cases, acceptable controller performance may be achieved with PI control, vice full PID control.

It is also possible to degrade the stability of a PID control system with the
integral term $k_i$. Because an integrator introduces a gain of $\frac{k_i}{jw}$ and phase lag of $\tan^{-1}(-\frac{k_i}{w})$; adding an integral term to a purely proportional controller will reduce its gain margin and stability margins accordingly.

Significant efforts have taken place to optimizing the performance of PID controllers. Getting the most from PID control involves a trade space of controller robustness, controller stability, and output performance [6]. The most prevalent methods are formula-based such as those in [8], [9], [10], and [11], with significantly more related patents being filed since 1971 [12]. Rule-based approaches such as those in [13] and [14] have been less of a focus area. An optimized PID controller will achieve acceptable output performance without sacrificing an unacceptable amount of passband stability and without excessive controller activity. Achieving this degree of optimization in real-world conditions has been the focus of research for some time now.

It is common practice to use only PI action to control plant hardware. This is due largely to the derivative term’s sensitivity to high frequency inputs. In many real-world applications, sensor noise is the main contributor to this band. When the effects of sensor noise can be mitigated, it has been shown that PID controllers are able to employ greater control action than PI controllers [6], giving superior output performance. Additionally, it has been demonstrated that including a low-pass filter in series with the PID controller will allow derivative action to be included in the control system design without high sensitivity to sensor noise [6]. It is demonstrated in [13] and [14] that by treating low, mid, and high frequency properties separately in performance evaluations of PID tuning schemes, high frequency robustness can be maintained while improving closed loop bandwidth and low frequency performance.

Feedback tuning methods such as those presented in [11], [15], [16], [17], [18]
tune PID parameters by using experimental data. This is known as iterative feedback tuning. The disadvantage to this method is that it generally requires a dedicated experimental trial to generate data, making closed-loop tuning difficult. Improvements to this process are proposed in [15] by generating small disturbances in the reference signal. Improvements to the experimental data collection via pre-filtering have been proposed in [16]. The use of predictors to improve the data collection process in large time-delay systems is researched in [17].

Improving the robustness of PID control under dynamic conditions is an area of active research. In [19] and [20], a hybrid fuzzy-PID controller is developed by utilizing fuzzy logic sets. A conventional PID controller is used during most situations, but control is handed off to a fuzzy-PID controller during severe disturbances.

The other major type of friction compensation involves the use of a friction model to anticipate frictional forces and compensate for them.

1.3 Model-Based Approaches to Friction Compensation

A model based friction compensation scheme is one in which a preselected or derived friction map is implemented into the control loop, often as a feed-forward gain summed with the control signal. The model anticipates the friction force based on an input signal (velocity for example) and cancels it out by adding extra force to the control signal. Position tracking systems with model-based friction compensation are generally more precise than those with non model-based friction compensation (or no friction compensation) but are usually less robust to changing conditions.

A simplified block diagram for a tracking system with model based friction compensation is given in Fig. 10. The system illustrated in Fig. 10 will force a hardware plant (which is affected by friction) to track a reference input signal
The friction model block generates an anticipated friction force based on velocity, which is added to the control signal.

Controllers with model-based friction compensation have been developed and demonstrated in numerous publications [21], [22]. At the simplest level, a controller is demonstrated in [23] by experimentally determining friction parameters and applying them in a pre-existing friction model. The characteristics of friction in a one axis motor driven system are determined by measuring output acceleration, motor torque, and minimum torque to begin motion. These parameters are then fed into a model which describes friction behavior, and this model is used in a feed-forward configuration in the final tracking system.

Another prevalent model-based technique is to augment a PID controller with a model-based friction compensator [24], [25], [26], [27]. A PD-type controller is augmented with a model-based “stick-slip” compensator to overcome friction in the pre-sliding regime. The advantage of such a technique is that some of the increased robustness inherent to feedback controllers can be captured in the control system. In [28], a nonlinear feedback compensator augments a PD controller in a similar
fashion. A pre-existing model is required for the development of such controllers.

Increasing the robustness of model-based friction compensators is an active area of research [29], [28], [30]. Recently published findings outline the development of a trajectory following controller which uses iterative data collection steps to predict the value of friction. When compared to controllers that employ a single friction model, this technique proves to be more robust to changing conditions.

The use of feedback loops is also prevalent in current research. One such approach is to augment a PID position controller with an acceleration feedback loop [31], [32]. Acceleration is computed or measured and used as a second feedback loop in the control system. The acceleration loop effectively compensates for static friction, while the PID controller compensates for viscous friction. Using acceleration directly in a feedback loop improves the robustness of the controller [33]. It is also possible to use acceleration indirectly as in input to an observer which improves estimates of other state variables [33]. Another such approach is to augment the PID controller with a position feedback loop [34] based on input/output measurements.

Nested configurations of feedback control loops have also been demonstrated. A typical example is [35], where a proportional controller is implemented for acceleration in an inner loop. Position control is accomplished through a traditional PI controller.

Hybrid approaches to friction compensation combining friction models and feedback loops also exist. In [36], the friction compensation approach uses an estimate of acceleration in conjunction with an estimate of friction. These parameters can be acquired through the use of a reduced-order observer [37], [38].
1.4 Summary of Findings

The position control strategy proposed in Chapter 2 implements a nested feedback control loop around motor velocity. Proportional, integral, and observer-based controllers are utilized in the inner loop. Computer simulations illustrate that the observer-based inner loop yields the best performance in position tracking applications. Results are most apparent when the input signal demands a higher degree of precision from the control system, such as steps of decreasing magnitude and sinusoids of decreasing amplitude. Hardware demonstrations substantiate the simulated results.

List of References


CHAPTER 2

Algorithm Development

2.1 Introduction

The inspiration for the control system design proposed here comes from hardware in the URI Control Systems laboratory. The laboratory’s 1-axis position control system utilizes an advanced servo drive motor and power amplifier that implements an analog control loop around motor velocity which compensates for friction (or other disturbances) on commanded motor velocity. It is possible to control this system with standard linear strategies designed in undergraduate-level controls classes.

In less advanced servo drive systems, the motor and power amplifier do not implement a velocity tracking loop. These systems will suffer from friction-induced effects and behave erratically, particularly at low velocities. When a linear strategy is used to design a control system for such a hardware plant, control is generally unsuccessful as the hardware behaves in a nonlinear fashion. Excessive time will be required for the system to settle at a commanded position, or large gain values will be required for the system to meet the required settling time, leading to windup at the integrators. Some type of friction compensation must be used to adequately control the system.

The control strategy proposed is intended to mimic the effect of an analog feedback loop around motor velocity. A pair of linear tracking systems are implemented in a nested configuration. The inner loop is a tracking system around motor velocity to compensate for friction. This will make the plant hardware behave linearly, as if in the absence of friction. The velocity tracking inner loop is then treated as part of the hardware plant model, and a standard linear control strategy is used to control the position of the hardware.
The first control system described is a simple position controller with no friction compensation. The nested feedback loop strategy is then illustrated with a simple proportional controller in the velocity tracking inner loop. Improvements on this were attempted by using an integral-type controller around motor velocity. The final control strategy utilizes an observer in the inner loop to estimate the value of the inner-loop integrator output and compensate for velocity accordingly.

2.2 Position Controller with no Friction Compensation

The first simulation developed is the case with no friction compensation. In this scenario, a digital tracking system is designed around a hardware actuation system with known properties. Standard linear techniques are used for the development of the tracking system and no consideration is given to friction-induced effects or other disturbances.

![Figure 11. Standard state-space position tracking system](image)

Fig. 11 illustrates the state-space tracking system described in Section 2.1. It is a simple position controller implemented around a second-order hardware plant with known properties. There is no friction compensation used in this tracking system. The input to the control system is a reference position, which is a unit step function. The tracking system consists of two feedback loops; one around the hardware plant feeding back both plant state variables through the gain vector...
\[ G_p(s) = \frac{\beta}{s(s + \alpha)}. \] (2)

The transfer function from input voltage to motor velocity is

\[ G_v(s) = \frac{\beta}{s + \alpha}. \] (3)

This will yield the following depiction of the system.

Figure 12. Equivalent depiction of Fig. 11 with state variables indicated

Fig. 12 shows the three state variables (two for the hardware plant and one for the digital integrator) of the complete system. The interrelationship of the system state variables and calculation of the tracking system coefficients will now be described.

Select Pole Locations

Pole locations for the tracking system are selected based on the order of the system. As this is a third-order system, there will be three system poles. It has been shown that the roots of a normalized Bessel polynomial will produce a
desirable step response for the tracking system [1]. In this case, roots for a third order Bessel polynomial will be used for tracking system poles. A table of Bessel roots is given in the Appendix. Pole locations are referred to as \( \{s_1, ..., s_i\} \). The letter \( s \) is used to signify that the pole locations are in the s-plane, or continuous time domain.

A tracking system settling time is selected. This is referred to as \( T_s \). The system poles must each be scaled to achieve the desired tracking system settling time. A vector of scaled Bessel poles, \( P \), is defined as

\[
P = \frac{1}{T_s} [s_1, ..., s_i].
\]  

(4)

The scaled system poles are now mapped into the discrete time domain (z-plane) using the zero order hold pole mapping formula. For a given continuous time (s-plane) pole \( s_0 \) and sampling interval \( T \), the equivalent z-plane pole \( z_0 \) is

\[
z_0 = e^{s_0 T}.
\]  

(5)

Equation (5) is applied to the vector of system poles to obtain

\[
Z = [e^{s_1 T_s}, ..., e^{s_i T_s}]
\]  

(6)

Now that the tracking system poles have been selected, the state space depiction of the system will be developed.

**State-Space Model for the Plant**

Fig. 12 shows the plant state variables to be

\[
x_1 = \text{motor position}
\]  

(7)

\[
x_2 = \text{motor velocity}.
\]  

(8)

The transfer functions in Fig. 12 correspond to the following state-space model
for the plant
\[ \dot{x}_1 = x_2 \]  
\[ \dot{x}_2 = -\alpha x_2 + \beta u \]
with state vector
\[ x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \]

The previous two equations can be written in standard state-space form as
\[ \dot{x} = Ax + bu \]  
\[ y = cx + du \]
where
\[ A = \begin{bmatrix} 0 & 1 \\ 0 & -\alpha \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ \beta \end{bmatrix}, \quad c = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad d = 0 \]

**Calculate the Zero Order Hold**

In the case of a digital control system for an analog plant, analog to digital and digital to analog converters must be used to actuate the plant hardware as shown in Fig. 12. In order to calculate the gains for a digital tracking system, it is necessary to find a mathematical model for the analog plant with the D/A converter at its input and the A/D converter on each state variable.

The zero order hold (ZOH) equivalent is a discrete-time model that exactly describes an analog system, whose piecewise constant input comes from a D/A converter, at sampling instants. When the zero order hold equivalent is substituted for an analog plant model in a digital control system, it results in a discrete time model for the complete control system. Fig. 13 below is functionally equivalent to Fig. 12. It is important to note that the zero order hold plant model is defined only at sampling instants. The sampling rate must be selected based on the required
Figure 13. Replacing the analog plant with its zero order hold equivalent performance of the control system, and can be limited by the quality of the control system hardware.

For a given continuous-time state-space model in the form of (12) \((A, b, c, d)\), and sampling interval \(T\), the zero order hold discrete-time equivalent is described by \((\Phi, \Gamma, c, d)\).

\[
\Phi = e^{AT} \quad (15)
\]
\[
\Gamma = \int_0^T e^{A\tau}b \, d\tau \quad (16)
\]

When developing simulations, the zero order hold is calculated using a Matlab function, which is included as a digital download.

Now that the zero order hold equivalent for the hardware plant has been calculated, the tracking system feedback gains and digital integrator can be determined.

**Include the Digital Integrator**

A Matlab script called dts (digital tracking system) will calculate the tracking system gains \(L_1\) and \(L_2\). The discrete-time state-space model for the digital integrator block shown in Fig. 12 is referred to as \((\Phi_a, \Gamma_a, L_2)\). This model is defined based on the pole locations of the anticipated reference input. It is important to note that the tracking system will follow any reference input, but it is possible to obtain zero steady-state error to a class of inputs (e.g. step inputs) using a digital integrator [1].
To implement a digital integrator, the matrix $\Phi_a$ must have one eigenvalue at $z = 1$, a discrete time integrator pole. This is easily obtained by defining $\Phi_a = 1$. For this system, the only requirement on $\Gamma_a$ is that it be a nonzero number, which is typically chosen to be $\Gamma_a = 1$.

Cascade Model for Plant and Digital Integrator

A state-space model for the system in Fig. 13 is derived as follows:

$$x[k + 1] = \Phi x[k] + \Gamma u[k] \quad (17)$$

$$x_a[k + 1] = \Phi_a x_a[k] + \Gamma_a (v[k] - C x[k]) \quad (18)$$

$$\begin{bmatrix} x[k + 1] \\ x_a[k + 1] \end{bmatrix} = \begin{bmatrix} \Phi & 0 \\ -\Gamma_a C & \Phi_a \end{bmatrix} \begin{bmatrix} x[k] \\ x_a[k] \end{bmatrix} + \begin{bmatrix} \Gamma \\ 0 \end{bmatrix} u[k] + \begin{bmatrix} 0 \\ \Gamma_a \end{bmatrix} v[k] \quad (19)$$

Where the second array is denoted as $\Phi_d$ (for $\Phi$ design model) and the fourth $\Gamma_d$ (for $\Gamma$ design model). Next, describe the plant input as illustrated in Fig. 13.

$$u[k] = -L_1 x[k] + L_2 x_a[k] = -\begin{bmatrix} L_1 & -L_2 \end{bmatrix} \begin{bmatrix} x[k] \\ x_a[k] \end{bmatrix} \quad (20)$$

$$= -\begin{bmatrix} L_1 & -L_2 \end{bmatrix} = -L_d \quad (21)$$

From (19) and (21),

$$\begin{bmatrix} x[k + 1] \\ x_a[k + 1] \end{bmatrix} = (\Phi_d - \Gamma_d L_d) \begin{bmatrix} x[k] \\ x_a[k] \end{bmatrix} + \begin{bmatrix} 0 \\ \Gamma_a \end{bmatrix} v[k] \quad (22)$$

The poles of the closed-loop system are the eigenvalues of $(\Phi_d - \Gamma_d L_d)$. It is possible to select desired closed-loop poles and then calculate the gain vector $L_d$.

Calculate the Feedback Gains

Calculation of the feedback gains is generally performed in Matlab. A Matlab script that executes this calculation based on a state-space plant model and specified closed-loop pole locations has been used when developing tracking system simulations. However, the calculation of feedback gains is briefly described here.
The characteristic polynomial of $\Phi$ is calculated first. This is also referred to as the characteristic polynomial of $\Phi$. This is the polynomial whose roots are the poles of the zero order hold equivalent model. The `poly` command in Matlab can be used to obtain the coefficients $(a_1, ..., a_n)$ of this polynomial:

$$a(z) = \det(zI - \Phi) = z^n + a_1z^{n-1} + ... + a_n$$ (23)

where $I$ is the identity matrix. The Matlab command `poly(\Phi)` gives

$$poly(\Phi) = [1, a_1, ..., a_n].$$ (24)

The next step is to calculate the polynomial

$$p(z) = z^n + p_1z^{n-1} + ... + p_n$$ (25)

whose roots are the selected pole locations. This calculation can also be done with the Matlab `poly` command. If $zpoles$ is a vector of desired pole locations, then

$$poly(zpoles) = [1, p_1, ..., p_n].$$ (26)

Form the controllability matrix, referred to as $W_c$, for $(\Phi, \Gamma)$.

$$W_c = \begin{bmatrix} \Gamma, \Phi^1\Gamma, ..., \Phi^{n-1}\Gamma \end{bmatrix}$$ (27)

The next step is to construct the controllable canonical form $(\bar{\Phi}, \bar{\Gamma})$. Use the coefficients of $a(z)$ from equation 23 to form $\bar{\Phi}$. A diagonal matrix of ones and zeros augments $\Phi$ to make its dimension square. $\Gamma$ is a column vector of the same dimension as $\Phi$.

$$\bar{\Phi} = \begin{bmatrix} -a_1 & -a_2 & ... & -a_n \\ 1 & 0 & ... & 0 \\ 0 & \ddots & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$ (28)

$$\bar{\Gamma} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$ (29)
The controllability matrix of controllable canonical form is defined in the same way as \( W_c \), using \( \bar{\Phi} \) and \( \bar{\Gamma} \).

\[
\bar{W}_c = \begin{bmatrix}
\bar{\Gamma} & \bar{\Phi}^1\bar{\Gamma} & \bar{\Phi}^2\bar{\Gamma} & \cdots & \bar{\Phi}^{n-1}\bar{\Gamma}
\end{bmatrix}
\]  

(30)

The transformation matrix, referred to at \( T_c \), is calculated. This is the transformation matrix that takes any controllable system to controllable canonical form.

\[
T_c = \bar{W}_cW_c^{-1}
\]  

(31)

For the controllable canonical form described by \( (\bar{\Phi}, \bar{\Gamma}) \), the feedback vector \( \bar{L} \) will position the system poles at the roots of \( p(z) \). These are the pole locations that were previously selected as an input to this process. Define the vector as follows.

\[
\bar{L} = [p_1 - a_1, p_2 - a_2, \ldots, p_n - a_n]
\]  

(32)

Finally, the vector of feedback gains for the closed loop system is calculated using the transformation matrix \( T_c \) and the associated vector of feedback gains \( \bar{L} \).

\[
L = \bar{L}T_c
\]  

(33)

The feedback vector \( L \) is divided into individual feedback gains based on the order of the hardware plant and additional dynamics. The order of the plant is referred to as \( n \), while the order of the additional dynamics is referred to as \( q \). The first \( n \) elements of \( L \) are \( L_1 \), while the remaining \( q \) elements are \( L_2 \).

2.3 Inner-Loop Velocity Tracking: Proportional Control

The next control strategy proposed implements a proportional controller around motor velocity. Velocity feedback is used to generate a velocity error by comparing commanded velocity to actual motor velocity. The resultant error signal is fed through a gain. This signal is used as the control input \( (u) \) to the plant hardware. This inner loop configuration is shown in Fig. 14.
Figure 14. Simple tachometer feedback inner loop

The input to the inner loop is commanded velocity, and the output is a vector consisting of the current position and velocity. Velocity is the only state variable used in feedback for the inner loop. The hardware plant is the same as that in Section 2.2.

Figure 15. An equivalent depiction of the inner loop in Fig. 14

An equivalent depiction of the inner loop illustrated in Fig. 14 is given in Fig. 15. To simulate the tracking system, the zero order hold equivalent $[\Phi_v, \Gamma_v]$ for the inner loop must be known. The calculation of these tracking system coefficients will now be described.

**State-Space Model for the Inner Loop**

To calculate the gain $K_v$, recall that the model from input voltage to output velocity is $\frac{\beta}{(s+\alpha)}$.

This leads to the state-space equation for velocity,

$$\dot{x}_2 = -\alpha x_2 + \beta u$$  \hspace{1cm} (34)
where $-\alpha$ is denoted $A_v$ and $\beta$ is denoted $b_v$.

With D/A and A/D converters, the ZOH equivalent is

$$\begin{bmatrix} \Phi_v, \Gamma_v \end{bmatrix} = \text{zohe}(A_v, b_v, T) \quad (35)$$

where \text{zohe} is a Matlab command used for calculating the ZOH.

Redrawing Fig. 15 so only velocity (state variable $x_2$) is illustrated, the depiction of the inner loop is as follows.

$$x_2[k+1] = (\Phi_v - K_v \Gamma_v)x_2[k] + K_v \Gamma_v v_c[k] \quad (36)$$

The gain $K_v$ can be chosen to place the pole of this first-order system in a desired location. In Matlab, the calculation for this first order system would appear as
follows. Let \( T_{sv} \) be the inner loop tracking system settling time. \( \texttt{Fbg} \) is a Matlab function used to calculate feedback gains.

\[
T_{sv} = \frac{Ts}{f} \quad \text{(for some value of } f)\]

\[
spole = s_{1}/T_{sv}
\]

\[
zpole = \exp(T^{*}spole)
\]

\[
K_{v} = \texttt{Fbg}(\Phi_{v}, \Gamma_{v}, zpole)
\]

The outer loop position tracking system gains illustrated in Fig. 15 can now be calculated. Note that in Fig. 15, only the velocity signal is fed back in the inner loop. Let

\[
x_{2}[k] = c_{v}x[k]
\]

where

\[
c_{v} = \begin{bmatrix}
0 & 1
\end{bmatrix}
\]

Then the inner loop system in Fig. 15 can be written as

\[
x[k + 1] = \Phi x[k] + \Gamma u[k]
\]

where

\[
u[k] = K_{v}(v_{c}[k] - c_{v}x[k])
\]

or

\[
x[k + 1] = (\Phi - K_{v}c_{v})x[k] + \Gamma K_{v}v_{c}[k].
\]

The model described by (41) will be used later as the plant model for the outer loop position tracking system.

When the zero order hold for the inner loop is known, the outer loop position tracking system coefficients can be calculated. A standard digital tracking system is designed around commanded position. It is assumed that the inner loop design effectively compensates for friction-induced disturbances so that linear techniques
can be used to develop the outer loop position tracking system. Fig. 18 illustrates the outer loop. The inner loop velocity tracking system is now treated as the hardware plant was during inner loop design.

**State-Space Model for the Outer Loop**

Full state feedback is used for the control signal around the inner loop velocity tracking system. Position feedback is used for the outer loop position control system.

![Diagram of outer loop position tracking system](image)

Figure 18. Outer loop position tracking system for the inner loop in Fig. 15

In Fig. 18, a step is given as the reference input. In practice, this could be any commanded position or continuous time input such as a sinusoid. The digital integrator block represents the additional dynamics of the tracking system. This block is described by the state space matrices $\Phi_a$ and $\Gamma_a$, and its portion of the feedback gain vector known as $L_{2p}$.

The inner loop block represents the plant hardware with inner loop velocity tracking system.
Select Outer Loop Pole Locations

Pole locations for the outer loop are now selected based on the order of the system. As the inner loop is second order \((n = 2)\) and the additional dynamics added in the outer loop are first order \((q = 1)\), the outer loop is a third order system. As was done in Section 2.2, the roots of a Bessel polynomial will be used. Third order roots are selected from the table of Bessel roots given in the Appendix. The poles listed are in the continuous time domain, or \(s\)-plane.

A settling time for the outer loop is defined as \(T_s = 1\) second. The outer loop tracking system poles are scaled to achieve the desired settling time as done in (4) (as \(Ts = 1\), no scaling is required in this case).

The scaled outer loop poles are now mapped into the discrete time domain using the zero order hold pole mapping formula, (5).

Calculate Outer Loop Feedback Gains

The feedback gains \(L_1p\) and \(L_2p\) are calculated using Matlab, as described in Section 2.2. The inputs to this process are the selected \(z\)-plane pole locations and the zero order hold equivalent for the inner loop, \(\Phi_v\) and \(\Gamma_v\).

Calculate Outer Loop Digital Integrator

The zero order hold equivalent for the digital integrator block in Fig. 18 is referred to as \((\Phi_a, \Gamma_a)\). This is calculated in the same manner as it was in Section 2.2.

2.4 Inner-Loop Tracking System: Integral Control

The next control system proposed attempts to improve upon the simple tachometer feedback design with the addition of an integral-type controller. Instead of a proportional gain, a digital integrator (optimized for tracking step inputs via pole placement) will be used in the inner loop. As was done in the simple
tachometer feedback configuration, the inner loop will attempt to compensate for friction-induced effects in motor velocity by tracking commanded velocity as a reference input.

The outer loop will be designed in a similar manner as the previous configuration, where commanded position is tracked and the inner loop system is treated as a black box representing plant hardware. Again, linear techniques are used for the design of both controllers. The inner loop velocity tracking system is shown in Fig. 19.

![Figure 19. Velocity feedback with integral control inner loop](image)

**State-Space Model for the Inner Loop**

Fig. 19 is a tracking system of the form shown previously, with

\[
\Phi_d = \begin{bmatrix} \Phi_v & 0 \\ -\Gamma_{av} & \Phi_{av} \end{bmatrix}
\]  

(42)

and

\[
\Gamma_d = \begin{bmatrix} \Gamma_v \\ 0 \end{bmatrix}.
\]  

(43)

Desired eigenvalues, (known as zpoles) are chosen. A vector of feedback gains \( L_d \) will be calculated so that the eigenvalues of \((\Phi_d - \Gamma_d L_d)\) equal zpoles. As this is a second order system, \( L_{1v} = L_1(1) \), and \( L_{2v} = -L_d(2) \). In Matlab, this calculation would appear as follows.
\[ Tsv = Ts/f, \text{ (where } f \text{ is a positive number, we use } f=3) \]

\[ spole = s2 / Tsv \]

\[ zpole = \exp(T^*spole) \]

\[ Ld = \text{fbg(Phid, Gammad, zpoles)} \]

The complete inner loop model with position and velocity is obtained by combining the plant ZOH with the plant input signal created by the inner loop shown in Fig. 19. The plant model is

\[ \ddot{x}[k+1] = \Phi \ddot{x}[k] + \Gamma u[k] \]  \hspace{1cm} (44)

and the plant input created by the inner loop in Fig. 19 is

\[ u[k] = L_{2v}x_3[k] - L_{1v}x_2[k]. \]  \hspace{1cm} (45)

The complete inner loop system with position as the output is a third order system with state variables \( x_1, x_2, \) and \( x_3. \) An array of the state variables is defined,

\[ \ddot{z}[k] = \begin{bmatrix} x_1[k] \\ x_2[k] \\ x_3[k] \end{bmatrix}. \]  \hspace{1cm} (46)

The two state variables used in the inner loop can be defined as follows.

\[ x_2[k] = c_2\ddot{z}[k] \]  \hspace{1cm} (47)

\[ x_3[k] = c_3\ddot{z}[k] \]  \hspace{1cm} (48)

Where

\[ c_2 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \]  \hspace{1cm} (49)

and

\[ c_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}. \]  \hspace{1cm} (50)

Therefore,

\[ u[k] = (L_{2v}c_3 - L_{1v}c_2)\ddot{z}[k]. \]  \hspace{1cm} (51)
Referring to Fig. 19,

\[ x_3[k + 1] = \Phi_{av} x_3[k] + \Gamma_{av} (v_c[k] - x_2[k]). \]  

(52)

Substituting (48),

\[ x_3[k + 1] = \Phi_{av} \dot{z}_3[k] + \Gamma_{av} v_c[k] - \Gamma_{av} \dot{z}_3[k] \]  

(53)

which simplifies to

\[ x_3[k + 1] = (\Phi_{av} \xi_3 - \Gamma_{av} \xi_2) \dot{z}[k] + \Gamma_{av} v_c[k]. \]  

(54)

Using (46) and (51), the system can now be described as

\[
\begin{bmatrix}
\dot{z}[k] + 1 \\
\end{bmatrix} = \begin{bmatrix}
\Phi, 0 \\
(\Phi_{av} \xi_3 - \Gamma_{av} \xi_2) \\
\end{bmatrix} \begin{bmatrix}
\dot{z}[k] \\
\end{bmatrix} + \begin{bmatrix}
\Gamma \\
(L_2 v_c - L_1 v_c) \\
\end{bmatrix} \begin{bmatrix}
0 \\
\Gamma_{av} \\
\end{bmatrix} v_c[k].
\]  

(55)

or

\[
\begin{bmatrix}
z[k + 1] \\
\end{bmatrix} = \begin{bmatrix}
\Phi, \Gamma \\
(\Phi_{av} \xi_3 - \Gamma_{av} \xi_2) \\
\end{bmatrix} \begin{bmatrix}
\dot{z}[k] \\
\end{bmatrix} + \begin{bmatrix}
0 \\
\Gamma_{av} \\
\end{bmatrix} v_c[k].
\]  

(56)

where the first array is called \( \Phi_p \) and the second is called \( \Gamma_p \).

**State-Space Model for the Outer Loop**

The third order system described in (56) is now treated as the plant model for outer loop design. The outer loop position tracking system coefficients can now be determined. The outer loop is shown in Fig. 20.

In Fig. 20, a step is given as the reference input. In practice, this could be any commanded position or continuous time input such as a sinusoid. The digital integrator block is described by its zero order hold equivalent state space matrices \( \Phi_{ap} \) and \( \Gamma_{ap} \), and its portion of the feedback gain vector known as \( L_{2p} \).

**Select Outer Loop Pole Locations**

Pole locations for the outer loop are selected based on the order of the system. As the inner loop is third order (\( n = 3 \)) and the additional dynamics are first order...
Figure 20. Outer loop position tracking system for inner loop in Fig. 19

(q = 1), the outer loop is a fourth order system. As was done in Section 2.2, the roots of a Bessel polynomial will be used. Fourth order roots are selected.

A settling time for the outer loop is defined as 1 second. This is referred to as $T_s$. The outer loop poles are now mapped into the discrete time domain using the zero order hold pole mapping formula (5).

Calculate Outer Loop Feedback Gains

Calculate the feedback gains $L_{1p}$ and $L_{2p}$ using Matlab as described in Section 2.2. The inputs to this process are the selected z-plane pole locations and the zero order hold equivalent for the inner loop, $\Phi_p$ and $\Gamma_p$.

Calculate Outer Loop Digital Integrator

The zero order hold equivalent for the digital integrator block in Fig. 20 is referred to as $(\Phi_{ap}, \Gamma_{ap})$. This is calculated in the same manner as it was in Section 2.2.

2.5 An Observer for the Inner Loop

The next approach to friction compensation utilizes a full-order observer in the inner loop velocity tracking system. The observer is used to estimate the value of the integrator state variable in the inner loop, which is then used as one of the
state variables in the outer loop position tracking system. The observer's estimates of this state variable are less affected by friction-induced non-linearities than the integrator output itself, which will produce large signals to counteract friction forces when necessary. This prevents the non-linearity from being included in the feedback of the outer loop position tracking system.

The addition of a full-order observer (w observer in Fig. 21) allows the state vector to be estimated. The observer output is an estimate of $x_3$.

![Figure 21. Inner loop velocity tracking system with observer](image)

An equivalent depiction of the inner loop illustrated in Fig. 21 is given in Fig. 22.

![Figure 22. Equivalent depiction of Fig. 21](image)
The observer’s estimate of the state vector ($\hat{x}$) is described by

$$\hat{x}[k + 1] = \Phi_o \hat{x}[k] + \Gamma_o v_c[k] + K(x_2[k] - c_o \hat{x}[k])$$

where

$$\hat{x}[k] = \begin{bmatrix} \hat{x}_2[k] \\ \hat{x}_3[k] \end{bmatrix}$$

(57)

In the previous equation ($\Phi_o, \Gamma_o, c_o$) is the state-space model for the system from $v_c[k]$ to $x_2[k]$. The observer produces an estimate of $x_3[k], \hat{x}_3[k]$, which is also called $w[k]$. In order to design the observer, it is first necessary to derive expressions for ($\Phi_o, \Gamma_o, c_o$), the system from $v_c[k]$ to $x_2[k]$. From Fig. 22,

$$x_2[k + 1] = \Phi_v x_2[k] + \Gamma_v (-L_{1v} x_2[k] + L_{2v} x_3[k]) = (\Phi_v - \Gamma_v L_{1v}) x_2[k] + \Gamma_v L_{2v} x_3[k]$$

(59)

and

$$x_3[k + 1] = \Phi_{av} x_3[k] + \Gamma_{av} (v_c[k] - x_2[k]) = -\Gamma_{av} x_2[k] + \Phi_{av} x_3[k] + \Gamma_{av} v_c[k].$$

(60)

The complete system is described by

$$\begin{bmatrix} x_2[k + 1] \\ x_3[k + 1] \end{bmatrix} = \begin{bmatrix} \Phi_v - \Gamma_v L_{1v} & \Gamma_v L_{2v} \\ -\Gamma_{av} & \Phi_{av} \end{bmatrix} \begin{bmatrix} x_2[k] \\ x_3[k] \end{bmatrix} + \begin{bmatrix} 0 \\ \Gamma_{av} \end{bmatrix} v_c[k]$$

(61)

where the second array is called $\Phi_o$ and the fourth is called $\Gamma_o$ and

$$x_2[k] = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_2[k] \\ x_3[k] \end{bmatrix}. $$

(62)

In the previous equation, the first array is called $c_o$. This model will be denoted $(\Phi_o, \Gamma_o, c_o)$. A full-order observer for this model is

$$\hat{x}[k + 1] = (\Phi_o - K c_o) \hat{x}[k] + \Gamma_o v_c[k]$$

(63)

$$w[k] = \begin{bmatrix} 0 & 1 \end{bmatrix} \hat{x}[k]$$

(64)

where $K$ is a (2x1) observer gains vector. The settling time of the inner loop is $T_{sv}$ seconds. We choose the observer settling time to be three times faster than
the velocity tracking inner loop: \( T_o = T_{sv}/3 \). Scaled Bessel poles mapped into the 
z-plane will give a vector of observer poles with the desired settling time. The
observer gains vector \( K \) is calculated in Matlab using the \texttt{fbg} script.

\[
K = \text{fbg}(\Phi_o', c_o', \text{observerpoles}').
\] (65)

### 2.6 Linear Model Development for Overall Tracking System

For evaluation purposes, it is desirable to know the ideal characteristics of the
control system. This is accomplished by developing a linear model of the system
from commanded position to output position.

A state space model of the complete system in Fig. 20 is derived as follows.

\[
\hat{z}[k + 1] = \Phi_p \hat{z}[k] + \Gamma_p (L_2 \hat{x}_a[k] - L_1 p \hat{z}[k])
\] (66)

\[
\hat{z}[k + 1] = (\Phi_p - \Gamma_p L_1) \hat{z}[k] + \Gamma_p L_2 \hat{x}_a[k]
\] (67)

Where \( \Phi_p \) and \( \Gamma_p \) are defined in 56. The additional dynamics are

\[
x_a[k + 1] = \Phi_a x_a[k] + \Gamma_a (\Phi a[k] - c_p \hat{z}[k])
\] (68)

Cascading the plant with the additional dynamics, the following state space model
is formed

\[
\begin{bmatrix}
    \hat{z}[k + 1] \\
    x_a[k + 1]
\end{bmatrix} = \begin{bmatrix}
    \Phi_p - \Gamma_p L_1 & \Gamma_p L_2 \\
    -\Gamma_a c_p & \Phi_a
\end{bmatrix} \begin{bmatrix}
    \hat{z}[k] \\
    x_a[k]
\end{bmatrix} + \begin{bmatrix}
    0 \\
    \Gamma_a
\end{bmatrix} \Gamma[k]
\] (69)

\[
y = \begin{bmatrix}
    1 & 0 & 0 & 0
\end{bmatrix} \begin{bmatrix}
    z \\
    x_a
\end{bmatrix}
\] (70)

### List of References

CHAPTER 3

Simulation Development

3.1 Hardware Plant Model

The hardware plant model is based on Palm’s model [1] of a DC servo drive motor that propels a cart sliding along a fixed track in one dimension. The plant model that will be used in simulation uses commanded velocity as an input and cart velocity as an output. The idealized plant model for such a hardware configuration is as follows:

\[
\frac{\beta}{s + \alpha}. \tag{71}\]

Additional parameters are given to the hardware plant model to more accurately simulate real-life conditions. These additional parameters, back EMF and friction, are modeled as forces acting against the hardware plant output. Because the cart is driven by a DC motor, the plant is subject to back electromotive force (EMF). This is the voltage generated by the motion of the motor armature through the external magnetic field. This voltage acts against the current which generates it [2]. The equation describing back EMF, referred to as \(e\), follows.

\[
e = k_E w \tag{72}\]

where \(k_E\) is the motor’s electric constant and \(w\) is the rotational velocity.

Beginning with Palm’s model [1] and setting the inductance parameter \(L\) to 0, figure 23 is realized. In the figure, \(K_T\) is the motor torque constant and \(I\) is the armature current.

The transfer function of this model will appear as follows.

\[
\frac{K_T}{R} \frac{1}{(Is + C)} \frac{1}{1 + \frac{K_T}{R} \frac{K_E}{(Is + C)}} \tag{73}\]
Focusing on the transfer function block, dividing the numerator and denominator by $I$ yields the following equivalent for the block.

The transfer function (73) can be simplified by multiplying by $\frac{R(I_s + C)}{R(I_s + C)}$ to obtain

$$\frac{K_T}{R(I_s + C) + KTK_E}$$

and again by $\frac{1}{IR}$ to obtain

$$\frac{K_T}{s + \frac{C}{I} + \frac{KTK_E}{IR}}.\quad (75)$$

Using (75), the following variables are defined.

$$\beta = \frac{K_T}{IR}\quad (76)$$

$$\alpha = \frac{C}{I} + \frac{KTK_E}{IR} = \frac{C}{I} + \beta K_E\quad (77)$$

This brings the system transfer function into the standard form defined by (71). Figure 24 is substituted into the Hardware Plant Model block diagram (Fig-
ure 23) to obtain Figure 25. In this case, $\beta = 1.3$ and $\beta K_E + \frac{C}{I} = 1.43$.

![Figure 25. Hardware Plant Model: Step 3](image)

As the friction models defined later in this chapter account for total friction, the viscous friction coefficient of the hardware plant model is replaced by a block containing the predefined friction models.

![Figure 26. Hardware Plant Model: Step 4](image)

3.2 Friction Model Development

The hardware plant being simulated will behave under the influence of one of several friction models. These models encompass major components of sliding
friction in a 1-axis positioning system: Viscous Friction, Coulomb Friction, and Stribeck Friction \([3,4]\). A fourth model is proposed which also incorporates a significant boundary zone of high friction at low velocities before the Stribeck effect decreases friction force \([3]\). These friction models are described in further detail here.

**Friction Model 1: Viscous Friction**

The friction experienced by two rigid bodies moving relative to one another at high speed is relatively linear \([3]\). This is known as the viscous, or kinetic, friction. The first friction model simulates this effect alone, and is the simplest friction model used. The viscous friction coefficient is user-defined in the simulation by parameter \(f_3\). Here it is equal to 1.3. It is the slope of the friction force vs. velocity plot at higher velocities. The Simulink block for this model executes the following logic:

\[
\text{Friction Force (N)} = f_3 \cdot \text{velocity} \tag{78}
\]

Figure 27 illustrates the friction force developed vs. velocity in friction model 1.

![Friction Model 1: Linear](image)

Figure 27. Friction model 1
**Friction Model 2: Stribeck**

Friction Model 2 begins to describe the friction force near zero velocity. This incorporates the static friction between two contacting bodies that must be overcome by an outside force before motion can occur [4]. Near zero velocity, friction force is highly nonlinear and is often viewed as a discontinuity [3]. In simulation, this is modeled as a steeply sloped line \( \text{slope} = 50 \). The peak static friction is defined in simulation as \( f_1 \).

The rapid decrease in friction force immediately after motion has occurred is known as the Stribeck Effect [3]. This is modeled in simulation with a parabola. Friction model 2 also incorporates the viscous friction coefficient from Friction Model 1 (known as \( f_3 \)) at higher velocities. The Simulink block for friction model 2 executes the following logic. Velocity is represented by \( v \).

\[
\begin{align*}
\text{IF } |v| \leq \frac{f_1}{50}, & \text{ THEN friction force } = 50v \\
\text{IF } \frac{f_1}{50} < |v| < \frac{f_1}{50} + \frac{f_1}{10} + \frac{f_1}{50}, & \text{ THEN friction force } = av^2 + bv + c \\
\text{IF } |v| \geq \frac{f_1}{50} + \frac{f_1}{10} + \frac{f_1}{50}, & \text{ THEN friction force } = f_3 \cdot v
\end{align*}
\]

Parameters \( a, b, \) and \( c \) are calculated by solving three simultaneous equations:

\[
\begin{align*}
f_1 = a f_1^2 + b f_1 + c & \quad (79) \\
0.14 f_1 \cdot f_3 = 0.14 \cdot a \cdot f_1^2 + 0.14 \cdot b \cdot f_1 + c & \quad (80) \\
f_3 = 0.28 \cdot a \cdot f_1 + b & \quad (81)
\end{align*}
\]

Equations 79 and 80 define the friction force at the parabola’s starting and ending points; velocity \( \frac{f_1}{50} \) and \( \left( \frac{f_1}{50} + \frac{f_1}{10} + \frac{f_1}{50} \right) \), respectively. Equation 81 is the slope of the parabola at its ending point. Friction Model 2 is illustrated in figure 28.
Friction Model 3: Stribeck Effect with Offset

Friction Model 3 builds on Model 2 with the addition of a Stribeck Curve. This model further contributes to nonlinearities near zero velocity, as friction force rapidly drops after motion has occurred. This is also known as the Stribeck Effect [3]. The Stribeck curve is based on the user-defined parameters $f_1$ (static friction) and $f_3$ (viscous friction), and is simulated as a second-order parabola.

At velocity $f_1 + f_3$, friction force is equal to $f_1$.

$$F_{fric} = 0.02a \cdot f_1^2 + 0.02b \cdot f_1 + c \quad (82)$$

The slope of the curve is zero at velocity $f_1 + f_3$.

$$0.24 \cdot a \cdot f_1 + b = 0 \quad (83)$$

$$2a = \frac{-b}{0.12f_1} \quad (84)$$

The slope of the curve is $f_3$ at velocity $(f_1 + f_3 + f_5)$. This is the velocity at which viscous friction becomes the dominant force.

$$0.28 \cdot a \cdot f_1 + b = f_3 \quad (85)$$
Substitute equation 84 into equation 85.

\[ b_1 - \frac{0.14f_1}{0.12f_1} = f_3 \]  

(86)

Solve for parameters \(a\) and \(b\) from equation 86, then solve equation 82 for parameter \(c\).

\[ b = \frac{f_3}{1 - \frac{0.14f_1}{0.12f_1}} \]  

(87)

\[ a = \frac{-b}{0.24f_1} \]  

(88)

\[ c = f_1 - 0.02b \cdot f_1 - 0.02a \cdot f_1^2 \]  

(89)

Calculate \(f_2\), the friction force at velocity \(\frac{f_1}{50} + \frac{f_1}{10} + \frac{f_1}{50}\).

\[ f_2 = 0.14a \cdot f_1^2 + 0.14b \cdot f_1 + c \]  

(90)

The Simulink block for friction model 3 uses the defined parameters \(f_1\) and \(f_3\), and the calculated parameters \(a\), \(b\), and \(c\), to execute the following logic, illustrated in figure 29:

**Friction Model 4: Stribeck with Offset and Boundary Zone**

Friction model 4 builds on friction model 3 with the addition of a “boundary zone,” which is a region of high friction force immediately after motion has occurred. This has been measured in [3], but is mainly intended to illustrate the effectiveness of the control techniques used in later chapters. It is modeled as
a flat spot in the friction force vs. velocity curve. The parameter $\delta$ defines the length of the flat spot, or the velocity range for which the boundary zone is in effect. The Stribeck curve is placed immediately thereafter, and is defined in much the same way as it was for model 3.

At velocity $\frac{f_1}{50} + \delta$, friction force is equal to $f_1$.

$$F(v) = (0.02a \cdot f_1 + \delta)^2 + (0.02b \cdot f_1 + \delta) + c$$  \hspace{1cm} (91)

The slope of the curve is zero at velocity $(\frac{f_1}{50} + \delta + \frac{f_1}{10})$.

$$(0.24a \cdot f_1 + \delta) + b = 0$$  \hspace{1cm} (92)

$$2a = \frac{-b}{0.12f_1 + \delta}$$  \hspace{1cm} (93)

The slope of the curve is $f_3$ at velocity $\frac{f_1}{50} + \delta + \frac{f_1}{10} + \frac{f_1}{50}$.

$$2a \cdot (0.14f_1 + \delta) + b = f_3$$  \hspace{1cm} (94)

Substitute equation 93 into equation 94.

$$b \cdot (1 - \frac{0.14f_1 + \delta}{0.12f_1 + \delta}) = f_3$$  \hspace{1cm} (95)
Solve for parameters $a$ and $b$ from equation 95, and then solve equation 91 for parameter $c$.

\[
b = \frac{f_3}{0.14f_1 + \delta} \quad (96)
\]

\[
a = \frac{-b}{0.24f_1 + \delta} \quad (97)
\]

\[
c = f_1 - b \cdot (0.02f_1 + \delta) - a \cdot (0.02f_1 + \delta)^2 \quad (98)
\]

Calculate $f_2$, the friction force at velocity $\frac{f_1}{50} + \delta + \frac{f_1}{10} + \frac{f_1}{50}$.

\[
f_2 = a \cdot (0.14f_1 + \delta)^2 + b \cdot (0.14f_1 + \delta) + c \quad (99)
\]

The Simulink block for friction model 4 uses the defined parameters $f_1$, $f_3$, and $\delta$, and the calculated parameters $a$, $b$, and $c$, to execute the following logic, illustrated in figure 30:

IF $|v| \leq \frac{f_1}{50}$, THEN friction force = 50$v$

IF $\frac{f_1}{50} < |v| \leq \delta$, THEN friction force = $f_1$

IF $\delta < |v| < \frac{f_1}{50} + \delta + \frac{f_1}{10} + \frac{f_1}{50}$, THEN friction force = $av^2 + bv + c$

IF $|v| \geq \frac{f_1}{50} + \delta \frac{f_1}{10} + \frac{f_1}{50}$, THEN friction force = $f_2 + f_3 \cdot (u - 0.14f_1 - \delta)$

Friction model 4 is illustrated in figure 30. Friction models 1-4 can be seen relative to one another in figure 31.

### 3.3 Tracking System Simulations

The first simulation developed is the case with no friction compensation. The simulation will be developed by specifying the friction models, then the tracking system coefficients used in the feedback loop.

#### Specify the Friction Models

First, the parameters for the four friction models will be calculated in Matlab. Friction models are implemented in a Simulink block containing four functions, one
for each friction model. This block is referred to as friction model and appears in
the hardware plant block for each tracking system developed.

For friction model 1, the coefficient of viscous friction, $f_3$, is defined. The
friction model is a function block that executes (78) to calculate the friction force.
The value chosen for $f_3$ is 1.3. Equation 78 is entered into the function block
as described in Section 3.2. For all input values, the value of the function is
proportional to parameter $f_3$.

For friction model 2, Matlab is used to calculate parameters $a$, $b$, and $c$ as
described in Section 3.2. The Stribeck curve is defined by $a$, $b$, and $c$. Input values
to this calculation are $f_1 = .6$, and $f_3 = 1.3$. The function block is programmed
to execute the logic described in Section 3.2. This is done using a conditional
statement based on the absolute value of the input variable, $u$. For example, the
statement ($u > 2$) will compute as 1 for all input values greater than 2, and will
compute as 0 for all input values less than 2. The function block expression is then
entered in this manner.

$$(u \leq 0.02f_1) \cdot 50u + (u \leq 0.02f_1) \ldots$$
For friction model 3, Matlab is used to calculate parameters $b$, $a$, and then $c$ as defined by equations 87 through 89. Input values to the calculation are $f_1 = 0.6$ and $f_3 = 1.3$. Next, equation 15 is applied to solve for $f_2$. The function block is programmed to execute the logic described in Section 3.2 using a conditional statement as done for friction model 2.

For friction model 4, Matlab is used to calculate parameter $b$ according to equation 21. Input values to this calculation are $f_1 = 0.6$ and $f_3 = 1.3$. Next, parameter $a$ is calculated according to equation 22, parameter $c$ according to equation 23, and parameter $f_2$ according to equation 24. Program the function block to execute the logical statements in Section 3.2 as done for friction model 2.

**Specify the Hardware Plant Models**

Next the hardware plant model is specified. The plant model used for the simple position controller is slightly different than that used in the other models. The plant is second order, as we are interested in position as well as velocity.
Figure 32. Friction model Simulink block

Figure 33. Second-order hardware plant model

Figure 33 illustrates the hardware plant block used in simulation. Note the second integrator used to model position as the integral of velocity. These two signals are multiplexed on the $pv$ bus and used in feedback. The hardware plant model block is specified in Simulink as illustrated in figure 33, no additional calculations are necessary.

Specify the Tracking Systems

Figure 34 illustrates the top level simulation block diagram for the position tracking system without friction compensation. Note that the hardware plant
block is illustrated in figure 33, and the friction model block is illustrated in figure 32.

The inner loop velocity tracking system with tachometer feedback is illustrated in figure 35. It is specified in Simulink as illustrated.

The outer loop position tracking system for inner loop with feedback is illustrated in figure 36. It is specified in simulink directly. The inner loop block appears in figure 35.

The inner loop velocity tracking system with integral control is illustrated in figure 37. It is specified in simulink directly.

The outer loop position tracking system for the inner loop in figure 37 is illustrated in figure 38.

The inner loop velocity tracking system with full-order observer is illustrated in figure 39. It is specified in simulink directly.
The outer loop position tracking system for the inner loop in figure 39 is illustrated in figure 40.

The blocks in the tracking system models are defined in Simulink using the coefficients calculated in Chapter 2.

**List of References**


Figure 38. Outer loop for inner loop with integrator

Figure 39. Inner loop with full order observer

Figure 40. Outer loop for inner loop with observer
Simulations have been conducted to compare the effectiveness of the various friction compensation techniques with progressively more difficult friction models acting against them. The simulations developed require the tracking systems to follow steps and sinusoids of various amplitudes. Smaller position inputs are more significantly influenced by the friction model, as the near-zero velocity portion of the friction model becomes increasingly significant.

4.1 Step Inputs

The first simulation is illustrated in Fig. 41 and Fig. 42. This compares the four friction compensation techniques tracking a 0.2m step input under the influence of friction model 1, model 2, model 3, and model 4. In Fig. 43 and 44, the input command is a 0.1m step. Other parameters remain the same. The reference input step size is reduced to 0.5m in Fig. 45 and Fig. 46. It is reduced to 0.2m in Fig. 47 and Fig. 48.

It can be determined from the figures that within the same size input command, the influence of friction models 1-4 becomes increasingly challenging for the tracking system to overcome. This is further illustrated in Fig. 49, where the proportional control system tracks the same step input under the influence of all four friction models.

It can also be determined from the step input simulations that the proportional control system, as developed, performs very similarly to the standard tracking system. This is due to the calculation of $K_v$ for a settling time of $\frac{1}{3}$ second. If $K_v$ is increased, simulation performance will also increase. This is illustrated in Fig. 50. Throughout the simulations, it can be noted that the standard tracking
and proportional control systems have similar position outputs, but significantly different hardware plant inputs.

Increasing the inner loop feedback gain $K_v$ is equivalent to calculating the
inner loop gains with a faster and faster settling time. When performing trials on positioning system hardware, it was discovered that manually increasing $K_v$ will excite a resonance in the system drive belt. For the purposes of this research, calculation of the inner loop gains is left as it is described in Chapter 2. All the systems containing inner loops use a settling time of $\frac{1}{3}$ second.

### 4.2 Sinusoids

The four tracking system simulations are now given a sinusoidal reference input. This is a more challenging signal to track, as there are repeated instances of velocity reversal being commanded. These velocity reversals force the tracking system to spend more time compensating for the discontinuities in the friction model at near-zero velocity.

The first sinusoidal simulation is illustrated in Fig. 51 and Fig. 52. This compares the four tracking systems following a 0.2m sinusoidal input with a period of $2\pi s$. Simulation results for friction models 1, 2, 3, and 4 are illustrated.

The next sinusoidal simulation is illustrated in Fig. 53 and Fig. 54. This is similar to Fig. 51, but with a smaller input signal.

Fig. 55 and Fig. 56 illustrate the integral control and observer based tracking systems tracking sinusoidal inputs of 0.01m, 0.005m, 0.002m, and 0.001m. The simulations are run with the most difficult friction model, number 4. The other two tracking systems are not pictured, as they are not able to adequately perform under these conditions.

### 4.3 Discussion

The simulation results shown in this chapter illustrate the effectiveness of the tracking systems under increasingly difficult conditions. This has been done to show approximately where each system begins to break down relative to the
others. Beginning with a 0.2m step input and friction model 1 (linear friction), all tracking systems perform adequately. This is due to the linear nature of each tracking system- the inner and outer loops are all linear control systems. This can be seen in all of the step input simulations (Fig. 41, 42 43, 44, 45, 46, 47, and 48) and sinusoidal simulations (Fig. 51, 52, 53, and 54).

When the simulation is switched to friction model 2, the same 0.2m step input begins to show decreased tracking performance across the systems. Friction models 3 and 4 produce worsening overshoot and oscillation about the reference position. This illustrated in Fig. 41 and Fig. 42. All of the tracking systems manage to produce an acceptable result, producing the desired position at the specified settling time of 1 second with minimal overshoot and oscillation.

Decreasing the step height begins to illustrate the limitations of the tracking systems. In Fig. 44, significant overshoot begins to appear in the standard tracking and proportional control systems with friction models 3 and 4. This trend continues in Fig. 45 and Fig. 46, when the observer based system begins to show some oscillation about the commanded position. This occurs with friction models 2, 3, and 4. The observer based system’s performance is better than the standard tracking and proportional control systems, but the integral control system shows less oscillation. The integral control system does not pass through the commanded position at the desired settling time, however. The integral control system shows greater positioning error at larger step heights. When the step height is decreased, the standard tracking and proportional control systems show similar (and larger) positioning errors.

This effect is magnified in Fig. 45, Fig. 46, Fig. 47 and Fig. 48. The standard tracking system and proportional control system have significant overshoot and oscillation, while the observer based system has comparatively little. The in-
Integral control system has the least overshoot and the least oscillation, but does not produce the commanded position as fast as the other three systems and has more position error.

Similar step responses to those developed in this chapter are demonstrated in [1] with 0.02m step input yielding moderate overshoot and oscillation about the setpoint in a PI controller. A second simulation in [1] yields significantly more overshoot and oscillation from a P controller. However, the settling time demonstrated is faster than any of the simulations developed in this chapter.

In [2], 1m step inputs are simulated on PID controllers using plant hardware of varying orders. Compared to the second order plant, all the simulations developed in this chapter achieve faster settling time and lower overshoot.

A tracking system with model-based friction compensation is demonstrated in [3] which appears to give greater precision than the integral control or observer-based systems developed in this chapter. The system uses the Generalized Maxwell Slip friction model in a feedforward configuration to successfully track 0.005m step inputs with very little overshoot or oscillation. The feedback based simulations developed in this chapter do not use such a friction map.

A similar tracking system using a derived friction model (based on experimental data from the hardware plant) is developed in [4]. This model-based system tracks sinusoidal signals of .0002m amplitude, showing improved precision over the feedback-based systems developed in this chapter. Model-based systems such as this appear to yield greater tracking performance when high precision is required. The performance of such systems is dependent on the accuracy of the friction model, however, and will generally not be as robust to changing conditions as feedback approaches.

In [5], a hardware demonstration is conducted using a hardware plant produc-
ing a rotational output (i.e. output position is in degrees). The hardware plant tracks step inputs over a significant portion of its travel using nonlinear PID control techniques. When qualitatively compared to a larger step input simulation (using friction model 4) on the translational (1-axis linear) hardware plant simulated in this chapter, the integral and observer-based tracking systems produce better settling time, overshoot, and oscillation. A similar observation can be made for [6]. With large magnitude step inputs, both systems will spend less time working against the friction-induced nonlinearity at low velocity. Comparison to other trials using rotational positioning hardware (such as [7], [8], [9], and [10]) are not useful, as the step input magnitude is very small.

When sinusoidal inputs are considered, the standard tracking and proportional control systems adequately track larger amplitude signals, but fall short as the amplitude decreases. This is illustrated in Fig. 51, Fig. 52, Fig. 53, and Fig. 54, where there is significant overshoot at the signal peaks.

The integral control and observer based tracking systems are able to effectively track sinusoidal inputs of much smaller amplitudes. In Fig. 56, both tracking systems follow sinusoidal inputs of decreasing amplitude, down to 0.001M. At this point, the observer based system produces the best result, with the integral control system not producing the full amplitude commanded. The position error of the integral control signal is generally higher than the observer based system.

In [11], a P controller with observer-based friction estimator is used to track a sinusoidal position command with 1s period and approximately 0.03m amplitude. This produces an output position waveform that appears somewhat shifted and flattened. The most applicable simulation developed in this chapter (0.02m amplitude, 1 rad period, friction model 4) demonstrates improved results. Both the integral control and observer based systems produce output waveforms with
minimal phase shift and amplitude reduction. This is illustrated in Fig. 54.

In [12], sinusoidal position tracking using model-based friction compensation is demonstrated. The Generalized Maxwell Slip model is applied in feedforward to track position commands. A 0.2m sinusoid is tracked successfully. Referencing Fig. 52, with friction model 4, the integral control and observer-based tracking systems proposed here achieve a faster settling time and similar tracking performance.

Sinusoidal input tracking using a rotational hardware plant is demonstrated in [5]. A large amplitude (60 degree) input of varying frequency is tracked by a PID controller. Qualitative comparison to a large amplitude sinusoid (0.2m, friction model 4) of similar frequency indicate the integral and observer-based system yield have potential to yield superior performance. A low frequency input sinusoid will force the controller to spend more time compensating for the nonlinear portion of the friction model, so the comparison becomes increasingly dependent on the friction model being used.

A circular input command is tracked by the position control system in [13]. While such x-y positioning will involve two actuators, useful observations can be made about tracking system performance. When comparing a 0.02m radius circle position input to the closest applicable simulation developed in this chapter (0.02m sinusoid, friction model 4), it can be noted that the integral control and observer based systems have less overshoot at velocity reversals than the x-y position tracking system. While a static friction model in a feed-forward configuration improves upon this this somewhat, significant error is still present. Adding an observer to the feed-forward portion of the compensator yields position tracking error similar to that demonstrated by the integral control and observer-based simulations demonstrated in this chapter.
List of References


Figure 42. Tracking 0.2M step inputs
Figure 43. Tracking 0.1m step inputs
Figure 44. Tracking 0.1m step inputs
Figure 45. Tracking 0.05m step inputs
Figure 46. Tracking 0.05m step inputs
Figure 47. Tracking 0.02m step inputs
Figure 48. Tracking 0.02m step inputs
Figure 49. Proportional control with friction models 1-4

Figure 50. Proportional control using various inner loop feedback gains
Figure 51. Tracking 0.2m sinusoidal inputs
Figure 52. Tracking 0.2m sinusoidal inputs
Figure 53. Tracking 0.02m sinusoidal inputs
Figure 54. Tracking 0.02m sinusoidal inputs
Figure 55. Integral and observer based tracking systems
Figure 56. Integral and observer based tracking systems
CHAPTER 5

Hardware Demonstration

The friction compensation techniques developed in the previous four chapters are now demonstrated in the laboratory to substantiate simulated results. The system chosen for the demonstration is a simple 1-axis translational mechanical system, similar to that which was simulated. The system moves a cart along a track by pulling it with a drive belt.

![Lab hardware used for control system demonstration](image)

Figure 57. Lab hardware used for control system demonstration

Fig. 57 depicts the hardware system used. This consists of a Feedback USA model 33-005 Digital Pendulum Control System and PC workstation running Simulink R13. The friction properties of the cart and track are unknown, and there is no type of friction compensation built into the hardware.

Some adaptations to the Simulink models developed in Chapter 3 were required to apply the control system to the hardware. These changes all occur in the Hardware Plant block of the Simulink models.
Fig. 58 illustrates the Hardware Plant block used for all hardware demonstrations. This differs from the block used in simulation, as there is no transfer function to simulate the plant’s behavior, and no friction model block to simulate the effects of friction on the hardware plant. There is also a transfer function applied to the cart’s output position that is specific to the hardware setup. This is not part of control system development, but is used to differentiate and filter the position signal to obtain an estimate of the velocity signal. Because Simulink is run with a fixed sampling interval, the second (analog) transfer function is implemented by numerical integration.

$$\frac{z - 1}{T \begin{bmatrix} 1 & 0 \end{bmatrix} z} = \frac{83.3 \begin{bmatrix} 1 & 120 \end{bmatrix}}{s^2 + 70.7s + 10^4} \quad (100)$$

A top level Simulink block is illustrated in Fig. 59. This is largely unchanged from the one developed Chapter 3. Data recorded are the control signal generated as the input to the hardware plant, the ideal output position, and the actual output position.

With the hardware up and running, the standard tracking system is demonstrated first. Only sinusoidal position inputs are illustrated, as they better illustrate the tracking system’s ability to follow an input command.
Fig. 60 illustrates the standard tracking system tracking a 0.2m input sinusoid with a $2\pi$-second period. Performance appears acceptable, with very minor overshoot at the signal peaks and minimal positioning error.

Fig. 61 illustrates the standard tracking system tracking an input command with half the amplitude and the same period as that in Fig. 60. Overshoot is more noticeable at the commanded signal's peaks, but performance is most likely still acceptable. The overshoot can be better seen in the positioning error plot.

In Fig. 62, the amplitude of the input position command is reduced to 0.05m. Significant overshoot can now be seen at the position peaks. In the plant input portion of the plot, it can be noted that the control signal generated spikes at velocity reversal. This is the tracking system attempting to overcome the nonlinear friction force at low velocity. Performance is no longer acceptable.

Hardware demonstration results are consistent with the simulations in Chapter 4. It can be seen that some of the friction models employed by tracking system simulations appear to be harsher than the friction present in the hardware system. This is expected, as friction model 4 was intended to be difficult. Additionally, the hardware’s friction map is unknown- demonstrations on an alternate system
Figure 60. Tracking a 0.2m sinusoid with no friction compensation would yield different results.

Similar tracking results are noted when the tachometer feedback system is employed. This is illustrated in Fig. 63, Fig. 64, Fig. 65, and Fig. 66. As the input sinusoid’s amplitude is reduced from 0.2m to 0.02m, performance degrades with noticeable overshoot at signal peaks. This is consistent with simulation results.

Fig. 67 illustrates the integral control system tracking an input sinusoid with 0.1m amplitude. Performance is improved over the previous systems. The amplitude of the input sinusoid is decreased progressively to 0.01m in Fig. 68, Fig. 69, and Fig. 70. The integral control system is able to successfully track smaller signals than the previous systems, but some oscillation about the input command becomes noticeable as the amplitude decreases.

When the observer based system is employed, similar performance to the integral control system is demonstrated. In Fig. 71, the input sinusoid is 0.1m in amplitude.
Figure 61. Tracking a 0.1m sinusoid with no friction compensation

As the amplitude of the input sinusoid is decreased, the observer based tracking system maintains adequate performance with less oscillation about the input position command. This is illustrated in Fig. 72, Fig. 73, and Fig. 74. The positioning error of the observer based system is comparatively low.

Throughout the simulations and hardware trials, the observer based system has produced the best tracking performance with consistently lower positioning error than the standard tracking system, proportional control system, or integral control systems. Both the integral control and observer based systems are able to effectively track lower amplitude sinusoids than the standard tracking or proportional control systems; but the observer based system provides the best overall performance.
Figure 62. Tracking a 0.05m sinusoid with no friction compensation

Figure 63. Tracking a 0.2m sinusoid with tachometer feedback
Figure 64. Tracking a 0.1m sinusoid with tachometer feedback

Figure 65. Tracking a 0.05m sinusoid with tachometer feedback
Figure 66. Tracking a 0.02m sinusoid with tachometer feedback

Figure 67. Tracking a 0.1m sinusoid with integral control
Figure 68. Tracking a 0.05m sinusoid with integral control

Figure 69. Tracking a 0.02m sinusoid with integral control
Figure 70. Tracking a 0.01m sinusoid with integral control

Figure 71. Tracking a 0.1m sinusoid with observer based control
Figure 72. Tracking a 0.05m sinusoid with observer-based control

Figure 73. Tracking a 0.02m sinusoid with observer-based control
Figure 74. Tracking a 0.01m sinusoid with observer-based control
APPENDIX

Bessel Roots

Roots of a normalized Bessel polynomial corresponding to a 1 second settling time, as published in [1]. For a settling time of $T$ seconds, divide the pole location (real and imaginary parts) by $T$.

<table>
<thead>
<tr>
<th>Order</th>
<th>Pole Locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-4.6200</td>
</tr>
<tr>
<td>2</td>
<td>-4.0530 ± j2.3400</td>
</tr>
<tr>
<td>3</td>
<td>-5.0093, -3.9668 ± j3.7845</td>
</tr>
<tr>
<td>4</td>
<td>-4.0156 ± j5.0723, -5.5281 ± j1.6553</td>
</tr>
<tr>
<td>5</td>
<td>-6.4480, -4.1104 ± j6.3142, -5.9268 ± j3.0813</td>
</tr>
<tr>
<td>6</td>
<td>-4.2169 ± j7.5300, -6.2613 ± j4.4018, -7.1205 ± j1.4540</td>
</tr>
<tr>
<td>7</td>
<td>-8.0271, -4.3361 ± j8.7519, -6.5714 ± j5.6786, -7.6824 ± j2.8081</td>
</tr>
<tr>
<td>8 (cont)</td>
<td>-8.7693 ± j1.3616</td>
</tr>
<tr>
<td>9 (cont)</td>
<td>-9.4013 ± j2.6655</td>
</tr>
<tr>
<td>10 (cont)</td>
<td>-9.9657 ± j3.9342, -10.4278 ± j1.3071</td>
</tr>
</tbody>
</table>

List of References

BIBLIOGRAPHY


