

Motion Control Theory Needed in the Implementation of Practical Robotic Systems

James Mentz

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical Engineering

Hugh F. VanLandingham, Chair
Pushkin Kachroo
Richard W. Conners

April 4, 2000
Blacksburg, Virginia

Keywords: Motion Control, Robotics, Obstacle Avoidance, Navigation
Copyright 2000, James Mentz

Motion Control Theory Needed in the Implementation of Practical Robotic Systems

James Mentz

(Abstract)

Two areas of expertise required in the production of industrial and commercial robotics are motor control and obstacle navigation algorithms. This is especially true in the field of autonomous robotic vehicles, and this application will be the focus of this work. This work is divided into two parts. Part I describes the motor types and feedback devices available and the appropriate choice for a given robotics application. This is followed by a description of the control strategies available and appropriate for a variety of situations. Part II describes the vision hardware and navigation software necessary for an autonomous robotic vehicle. The conclusion discusses how the two parts are coming together in the emerging field of electric smart car technology.

The content is aimed at the robotic vehicle designer. Both parts present a contribution to the field but also survey the required background material for a researcher to enter into development. The material has been made succinct and graphical wherever appropriate.

(Grant Information)

This early part of this work done during the 1999-2000 academic year was conducted under a grant from Motion Control Systems Inc. (MCS) of New River, Virginia.

Acknowledgments

I would like to thank the folks at MCS for supporting the early part of this research and for letting me build and go right-hand-plane with the inverted pendulum system of Chapter 5. A one meter pendulum on a one kilowatt motor looked pretty harmless in simulation. Thanks to Jason Lewis for helping with that project and the dynamics.

I would also like to thanks the teachers who have influenced me for the better throughout my years: my parents, Mrs. Geringer, Mrs. Blymire, Mr. Koba, and Dr. Bay. I also learned a lot from my colleagues on the Autonomous Vehicle Team, who know who they are. Special thanks to Dave Mayhew, Dean Haynie, Chris Telfer, and Tim Judkins for their help with the many incarnations of the Mexican Hat Technique.

*To my family:
Anne, Bob, Karl, and Karen*

Table of Contents

(ABSTRACT)	ii
(GRANT INFORMATION)	ii
ACKNOWLEDGMENTS	iii
TABLE OF FIGURES	vii
INDEX OF TABLES	viii
CHAPTER 1. INTRODUCTION	1
PART I. MOTION CONTROL	2
CHAPTER 2. CHOOSING A MOTION CONTROL TECHNOLOGY	2
<i>Field-Wound versus Permanent Magnet DC Motors</i>	5
<i>Brush or Brushless</i>	6
<i>Other Technology Choices</i>	6
CHAPTER 3. THE STATE OF THE MOTION CONTROL INDUSTRY	8
<i>Velocity Controllers</i>	12
<i>Position Controllers</i>	15
<i>S-curves</i>	17
<i>The No S-curve</i>	21
<i>The Partial S-curve</i>	22
<i>The Full S-curve</i>	24
<i>Results of S-curves</i>	24
CHAPTER 4. THE STATE OF MOTION CONTROL ACADEMIA	26
<i>Motor Modeling, Reference Frames, and State Space</i>	26
<i>Control Methodologies</i>	31
<i>Design of a Sliding Mode Velocity Controller</i>	33
<i>Design of a Sliding Mode Torque Observer</i>	34
<i>A High Gain Observer without Sliding Mode</i>	36
<i>Conclusion</i>	42
CHAPTER 5. SOFT COMPUTING	45
<i>A Novel System and the Proposed Controller</i>	45
<i>The Fuzzy Controller</i>	48
<i>Results and Conclusion</i>	52

CHAPTER 6. A PRACTICAL IMPLEMENTATION	57
<i>Purchasing Considerations</i>	57
<i>Motion Control Chips</i>	59
<i>Other Considerations</i>	61
CHAPTER 7. A CONCLUSION WITH AN EXAMPLE	63
<i>Conclusion</i>	63
<i>ZAPWORLD.COM</i>	63
PART II. AUTOMATED NAVIGATION.....	66
CHAPTER 8. INTRODUCTION TO NAVIGATION SYSTEMS	66
CHAPTER 9. IMAGE PROCESSING TECHNIQUES.....	69
CHAPTER 10. A NOVEL NAVIGATION TECHNIQUE	71
CHAPTER 11. CONCLUSION	77
VITA.....	78
BIBLIOGRAPHY	79
<i>References for Part I</i>	79
<i>References for Part II</i>	82

Table of Figures

Figure 2.1. A typical robotic vehicle drive system.	2
Figure 2.2a. DC Brush Motor System.....	4
Figure 2.2b. DC Brushless Motor System.....	4
Figure 2.3a. Field-Wound DC Brush Motor. 2.3b. Torque-Speed Curves.	5
Figure 3.1. Common representations of the standard DC motor model.	8
Figure 3.2. A torque-speed plotting program.....	10
Figure 3.3. Bode Diagram of a motor with a PI current controller.	10
Figure 3.4. A typical commercial PID velocity controller.....	12
Figure 3.5a. A step change in velocity. 3.5b. The best response	14
Figure 3.6a. A popular position compensator	16
Figure 3.6b. A popular position compensator in wide industrial use.....	16
Figure 3.6c. A popular position compensator	16
Figure 3.7. Two different points of view of ideal velocity response.	18
Figure 3.8. S-curves profiles resulting in the same velocity	19
Figure 3.9. S-curve profiles that reach the same velocity and return to rest.....	20
Figure 3.10. S-curve profiles that reach the same position.....	25
Figure 4.1. The stationary and the rotating reference frame	28
Figure 4.2. Three models of friction.. ..	30
Figure 4.3. Block diagram of system to be observer and better controlled.....	32
Figure 4.4. Comparison of High Gain and Sliding Mode Observers.....	37
Figure 4.5. Block diagram of a system with a sliding mode observer and feedforward current compensation.....	38
Figure 4.6. Comparison of three control strategies ($J=1$ p.u.).....	39
Figure 4.7. Comparison of three control strategies ($J=2$ p.u.).....	41
Figure 4.8. Comparison of three control strategies ($J=10$ p.u.).....	41
Figure 5.1. An inverted pendulum of a disk.....	45
Figure 5.2. Inverted Pendulum on a disk and its control system.	48
Figure 5.3. Input and Output Membership Functions	50
Figure 5.4. This surface maps the input/output behavior of the controller	50
Figure 5.5. The final shape used to calculate the output and its centroid	52
Figure 5.6. The pendulum and disk response to a 10° disturbance	54
Figure 5.7. The pendulum and disk response to a 25° disturbance.....	55
Figure 5.8. The pendulum and disk response to a 45° disturbance.....	56
Figure 6.1. Voltage captures during two quick motor stall current surges	61
Figure 7.1. The ZAP Electricruizer (left) and Lectra Motorbike (right).....	64
Figure 8.1. A typical autonomous vehicle system	66
Figure 10.1. The Mexican Hat	71
Figure 10.2. The Shark Fin	72
Figure 10.3. A map of obstacles and line segments	73
Figure 10.4. The potential field created by Mexican Hat Navigation.....	73
Figure 10.5. The path of least resistance through the potential field	74
Figure 10.6. The resulting path through the course.....	74

Index of Tables

TABLE 3.2. FEEDBACK PARAMETERS TYPICALLY AVAILABLE FROM MOTOR CONTROLLERS AND THEIR SOURCES	11
TABLE 4.1. TRANSFORMATIONS BETWEEN DIFFERENT DOMAINS ARE POSSIBLE	28
TABLE 5.1. WEIGHT GIVEN TO PID CONTROLLERS TORQUE COMMAND	49
TABLE 5.2. WEIGHT GIVEN TO PID CONTROLLERS TORQUE COMMAND	51
TABLE 6.1. MOTION CONTROL CHIPS AND PRICES	59
TABLE 6.2. TOP 10 TIME CONSUMING TASKS IN THE DESIGN OF AUTONOMOUS ELECTRIC VEHICLES	62

Chapter 1. Introduction

Most research in robotics centers on the control and equations of motion for multiple link and multiple degree-of-freedom armed, legged, or propelled systems. A great amount of effort is expended to plot exacting paths for systems built from commercially available motors and motor controllers. Deficiencies in component and subsystem performance are often undetected until the device is well past the initial design stage.

Another popular area of research is navigation through a world of known objects to a specified goal. An often overlooked research area is the navigation through an area without a goal, such as local obstacles avoidance on the way to a global goal. The exception is smart highway systems, where there is a lot of research in lane and line tracking. However, more general applications such as off-road and marine navigation usually rely on less reliable methods such as potential field navigation.

Part I presents the research necessary for the robotics designer to select the motor control component and develop the control system that will work for each actuator. It follows the path the robot developer must follow. Hardware and performance constraints will dictate the selection of the motor type. With this understanding environmental and load uncertainty will determine the appropriate control scheme. After the limitations of the available control schemes are understood the hardware choices must be revisited and two compromises must be made: feedback quality v system cost and response v power budget.

Part II presents the research necessary to develop a practical navigation system for an autonomous robotic vehicle. The most popular sensors and hardware are surveyed so that a designer can choose the appropriate information to gather from the world. The usual navigation strategies are discussed and a robust novel obstacle detection scheme based on the Laplacian of Gaussians is suggested as robust obstacle avoidance system. Designers must take this new knowledge of navigation strategies and once again return to the choice of hardware until they converge upon an acceptable system design.

Part I. Motion Control

Chapter 2. Choosing a Motion Control Technology

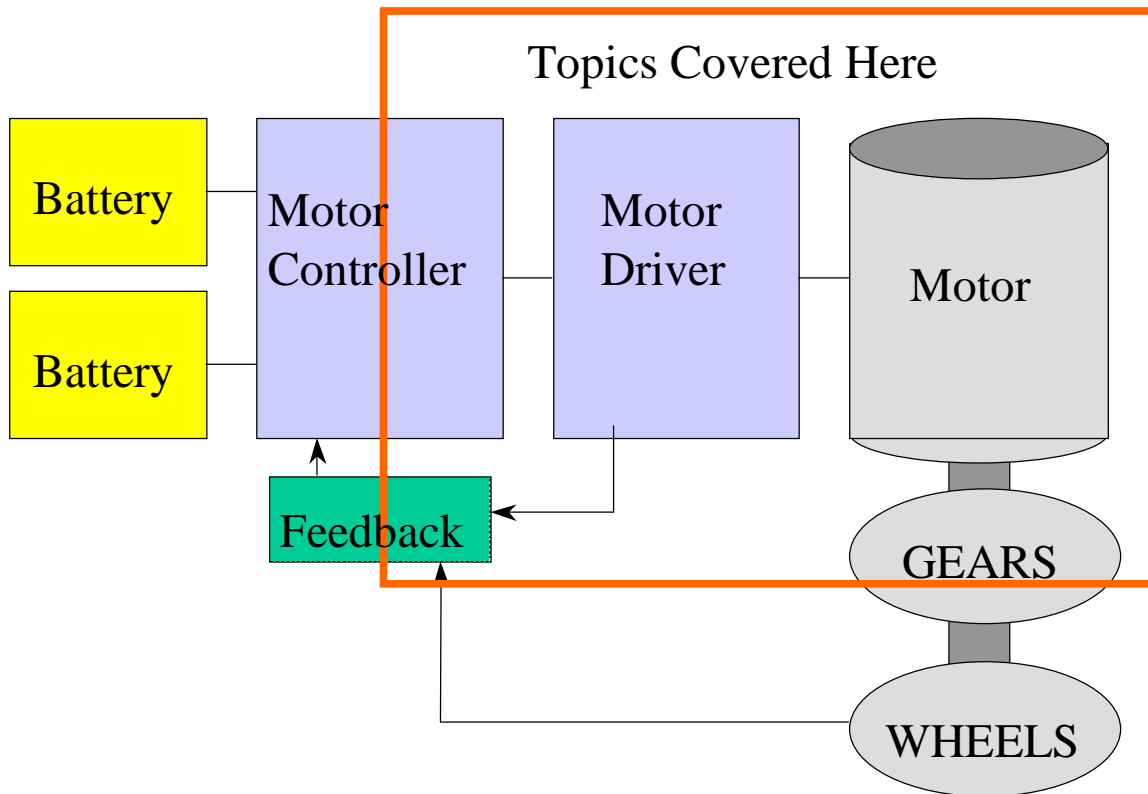


Figure 2.1. A typical robotic vehicle drive system showing the parts discussed here.

Many robots are built and operated only in simulation. Regardless of how painstakingly these simulations are designed it is rare that a device can be constructed with behavior exactly matching the simulation. The construction experience is necessary to be assured of a practical and robust mechanical and electrical design. With an advanced or completed prototype the mechanical designer can provide all the drawings, inertias, frictions and losses to create an accurate simulation. Ideally, the choice of motor, motor controller, feedback devices and interface is made and developed concurrently with the system design. This chapter serves a guide to the appropriate technology.

Table 2.1 presents each of the popular motor types and their most important characteristics for the purpose of constructing robotic vehicles. An important factor that has been left out of the table is cost. There are some good reasons for doing this:

- Competition has made the cost for a given performance specification relatively invariant across the available appropriate technologies.
- The cost of powering, controlling, and physically designing in the motion system with the rest of the robot is greatly reduced by choosing the appropriate motor.

Table 2.1. Common motor types and their characteristics

Motor Type	Power at Motor Leads	Typical Efficiency ⁽¹⁾	Coupling	Controller
DC Brush	DC	< 50%	Direct or Reducer	Simple to Complex
DC Brushless	Variable Freq. 3 Phase AC	> 90%	Direct or Reducer	Complex
AC Induction	3 Phase AC	< 90%	Reducer	Simple
AC Synchronous	Variable Freq. 3 Phase AC	> 90%	Direct or Reducer	Simple to Complex
Stepper	Digital Pulse	< 5%	Direct or Reducer	Simple

(1) Efficiencies are for motors below 3.7 kW. By necessity, motor efficiency increases with size for all types and is over 90% for almost all motors in the tens of kilowatts.

The first consideration in choosing a motor type is the input power available. Large stationary robots used in automation and manufacturing can assume a 3 Phase AC supply, but robotic vehicles are often all-electric and operate off DC busses or hybrid electric and convert power to a common DC bus. Figure 2.2 illustrates how DC motors are named “DC” based on the input power to the controller, not the shape of the voltage or current on the motor leads.

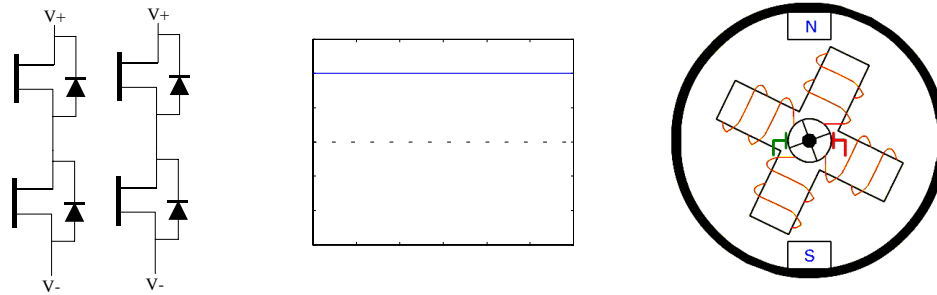


Figure 2.2a. DC Brush Motor System with inverter (left), DC on motor leads (center), and brush motor.

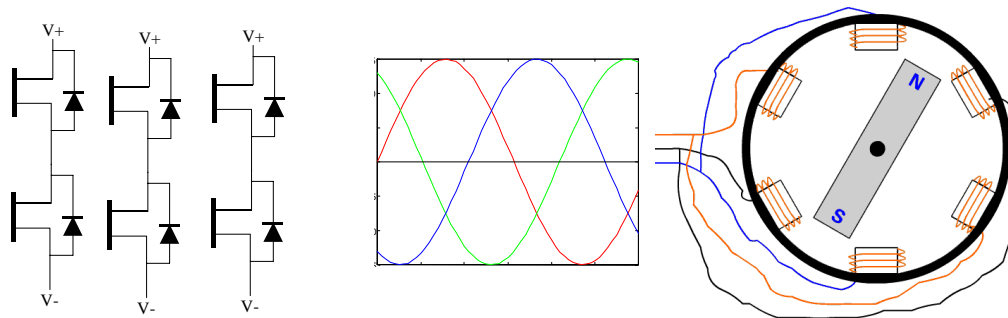


Figure 2.2b. DC Brushless Motor System with inverter (left), AC on motor leads (center), and brushless motor.

The remainder of this thesis will concentrate on DC motors as they are the most common choice for electrically powered robotic vehicles. However, it is noteworthy that for large vehicles and power levels over about 5 kW, an inverter controlled AC machine may be a better choice because of its availability in larger size ranges and the greater control over the motor's torque-speed characteristics gained by using windings to generate all the fluxes instead of relying on permanent magnets. Luttrell *et. al.* [1] used a synchronous motor that is inverter-fed off a DC bus in the award-winning Virginia Tech 1999 Hybrid Electric FutureCar.

AC Induction motors are rarely used in propulsion because they slip, and therefore lose efficiency, whenever they are under load and also have very poor performance at low speed, again where slip is high. However, AC Induction motors are the general work-horse of industry because of relatively high starting torque and high general reliability. There are several attempts to encourage the research and industry-wide adoption of high-efficiency induction motors, such as the specifications of Pyrhönen *et. al.* in [4].

Stepper motors are built to “step” from one position to the next through a fixed angle of rotation every time they receive a digital pulse. The common fixed angles sold by Oriental Motor in [2] are 0.72° and 1.8° , or 500 and 200 steps per revolution. Stepper motors are appealing in many applications where easy control and smooth velocity and position changes are not required. A common example of an easy to control and low cost application is a stepper motor used to turn the helical snack dispensing screw in a vending machine. Sometimes the discrete motion of a stepper motor is advantageous, as when a stepper motor and belt drive is used to step a horizontal document scanner vertically down a document. Robots and electric vehicles are often covered with sensors and parts that are best moved with stepper motors, but their jerky motion and low efficiency make them a poor choice for vehicle propulsion.

Field-Wound versus Permanent Magnet DC Motors

DC Brush motors all use brushes to transfer power to the rotor. However, the field may be created by permanent magnets or by another set of windings. When another set of windings is used De La Ree [3] shows how the two sets of motor leads can be connected in different arrangements to produce different torque-speed curves, as shown in Figure 2.3b.

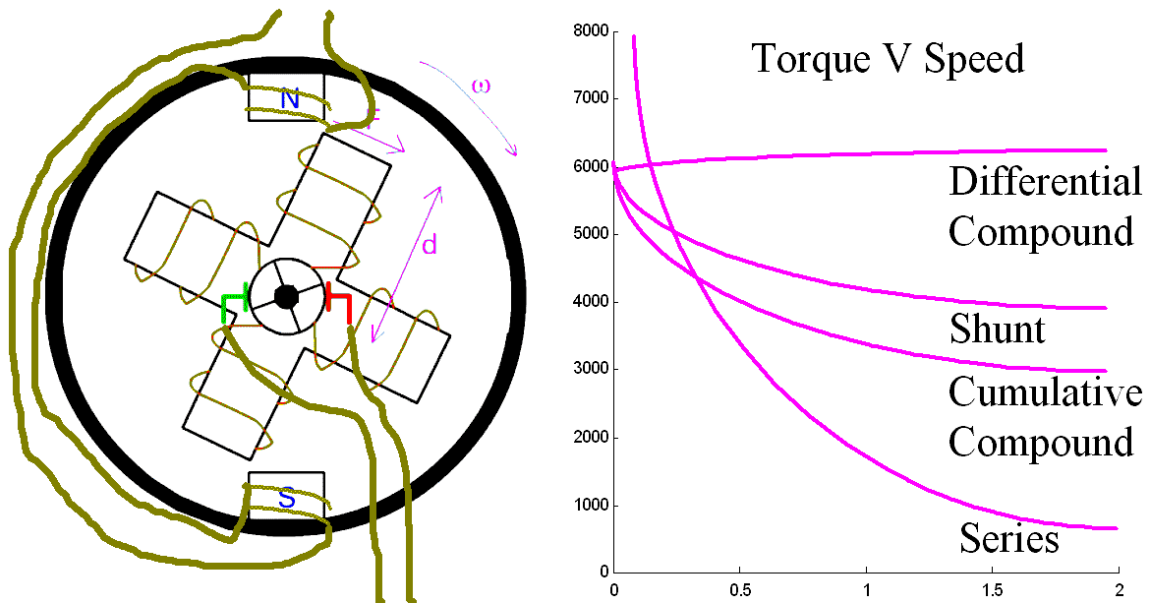


Figure 2.3a (left). Field-Wound DC Brush Motor. 2.3b. Torque-Speed Curves for various configurations.

In general wound field DC motors are bigger, bulkier, and less efficient than permanent magnet DC machines. Their use in electric vehicles should be compared to the use of AC synchronous machines. The following chapters will further limit discussion to permanent magnet DC brush motors. DC brushless motors always use windings in the stator and permanent magnets on the rotor to remove the need for brushes.

Brush or Brushless

Brush motors are older and more broadly used. They have difficulty at high speed when brush currents start arcing from pad to pad. They have problems with torque ripple at low speed when high amounts of current and flux switch from one winding to the next. Brushes create sparks that may need to be contained and the brushes will eventually wear. However, brush motors are easy to control, and the motor leads can be connected directly to a DC current source.

Brushless motors overcome all the problems of brush motors. They work at very high speeds -- even speeds where air or magnetic bearings are required because ball bearing liquefy. They can be designed to work at low speed with very high torque and low torque ripple. The trade-off comes in the complexity of the controller. The brushless controller needs to modulate three sinusoidal signals in-phase with the electrical or mechanical angle of the machine. The deciding factor that makes the choice of brushless motors worthwhile is if designs allow for direct drive. Brushless motors are more likely to be available with torque-speed characteristics that allow them to be directly coupled to the load, avoiding the cost, size, and loss of a reducer like a gearbox.

Other Technology Choices

Brush and Brushless motors are both available framed -- the typical motor with bearings in a housing with shaft and wire leads coming out -- and frameless -- the rotor, stator, and slip-ring or brush assembly (if a brush motor) come as loose pieces and are

build-in around the larger system's (potentially very large) shaft. If a reducer is needed, spur or planetary gearheads will often be sold as part of the system. When manually measuring reduction ratios the curious engineer needs to be aware that to minimize wear patterns gearheads are often made with non-integer reduction ratios. Torque tubes are a form of reducer also popularly used in robotics.

The feedback device will greatly affect the performance and price of the system. The popular feedback devices are resolvers, encoders, and hall-effect sensors. Resolvers are rotating transforms that modulate a high frequency carrier signal as the transformer core, which is coupled to the shaft, rotates. Resolvers actually produce two sinusoidally modulated signals that are 90° out of phase. Resolvers work well and are relatively inexpensive, but the electronics to interpret high resolution velocity and position data from the sinusoidal signals can be complex and expensive. Hall-effect sensors are used mostly to measure the rotor angle for electrical commutation.

Encoders detect the flashes of light that come shining through a slotted disk attached to the rotating shaft. Many low-cost, low-resolution encoders are available that easily interface to control electronics. Higher priced encoders use the varying intensity interference pattern caused by light shining through adjacent slits to produce sinusoidal signals like resolvers. In [4] Canon USA describes the most accurate encoder the author could locate, with 230 million pulses per revolution, an accumulative accuracy of 1 arc/second or less and 0.005625 arc-second resolution.

Finally, the choice of controller greatly affects system performance. If performance, size, and weight specifications are well known in advance, the motor, controller, and all necessary interface and feedback devices can be purchased as a system. Controllers contain an interface, a control loop, and a current amplifier. The interface can be any communications standard such as ethernet, RS-232, or analog +/-10V values, and one that works with the rest of the system should be available. The current stage can be a switching amplifier (the current on the motor leads is controlled through PWM of the voltage) or a more expensive linear amplifier (the voltage to the motor is smooth, as in a giant audio amplifier). The contents of the control loop is the subject of the remaining chapters of Part I.

Chapter 3. The State of the Motor Control Industry

The standard model for a DC motor is shown in Figure 3.1. This model applies to the Brush DC motor viewed from the motor leads. Also, when an entire Brushless DC motor system has its three-leg inverter switched so that the voltage on the motor leads peaks at the peak voltage of the DC link stage (see Figure 2.2 to help visualize this) the DC Brushless motor will have the same behavior as the DC Brush motor for modeling purposes. Being able to use the same model for Brush DC and Brushless DC motors is extremely convenient for both writing simulations and using motor sizing software. Krause [6] and others imply that this identical behavior is the real reason behind the name of the DC Brushless motor.

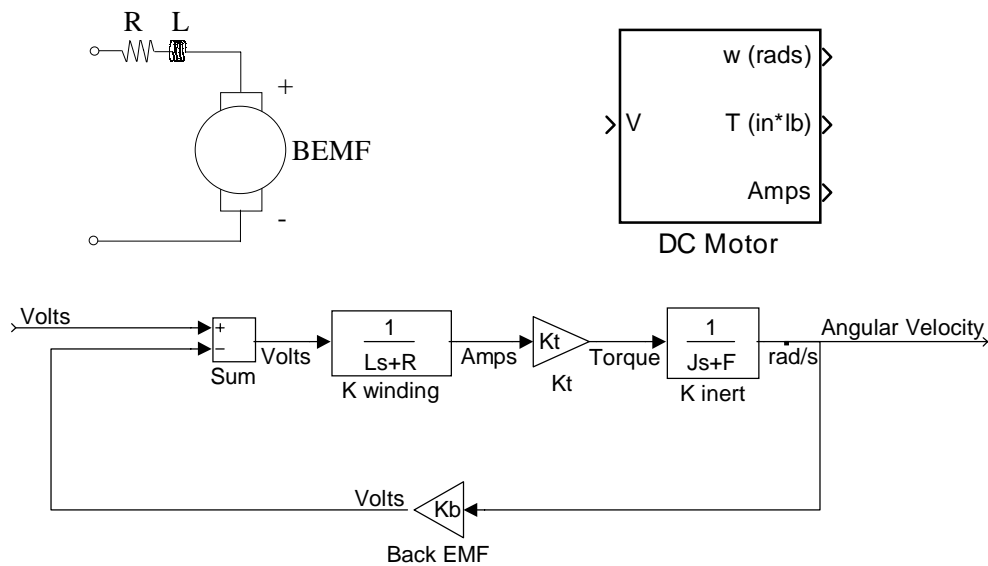


Figure 3.1. Common representations of the standard DC motor model. 3.1a. (upper left) as a circuit schematic. 3.1b (upper right) as an input/output block. 3.1c. as a block diagram.

The values in Figure 3.1c are:

L = induction of windings

F = rotary friction of motor

R = resistance of windings

K_t = torque constant

J = inertia of motor and load

K_b = back EMF

This model produces the Velocity/Volts transfer function:

$$\frac{\omega}{V}(s) = \frac{Kt/LJ}{s^2 + \left(\frac{F}{J} + \frac{R}{L}\right)s + \frac{RF + Kt Kb}{LJ}} \quad (3.1)$$

The parameters L and R are usually given in standard (metric) units. The parameters J and F are usually easily convertible to standard values. However, Kt and Kb can present difficulties. When all parameters have been converted to standard units as Ramu [6] does, Kt and Kb are have the same value and can be represented with one parameter. When motor manufacturers supply Kt and Kb value, they are usually used for motor testing and not for modeling, and are therefore in a convenient unit for testing such as Volts / 1000 RPM. This would still not be a difficulty if not for the Brushless motors: the standard units for Ke use voltage per phase, but Ke is often printed using line-line voltage; the standard units of Kt are per pole pare, but Kt is usually printed in total torque for the entire motor.

The solution to the units confusion is to ask each manufacturer; most companies use units that are consistent across their literature. A more common solution is to bypass modeling parameters and provide torque-speed curves for each motor. In [8] the author provides the torque-speed curve generating program shown in Figure 3.2. This program is useful for both generating the torque-speed curve for a given set of parameters and manually adjusting parameters to find possible values for a desired level of performance.

Most motor manufacturers will provide either torque-speed curves or tables of critical points along the torque-speed curve in their catalog. Some manufacturers will provide complete motor and system sizing software packages such as Kollomorgen's *MOTIONEERING* [9] and Galil's Motion Component Selector [10]. These programs collect information about the load, reducers, available power, and system interface and may suggest a complete system instead of just a motor. They usually contain large motor databases and can provide all the motor modeling parameters required in (3.1).

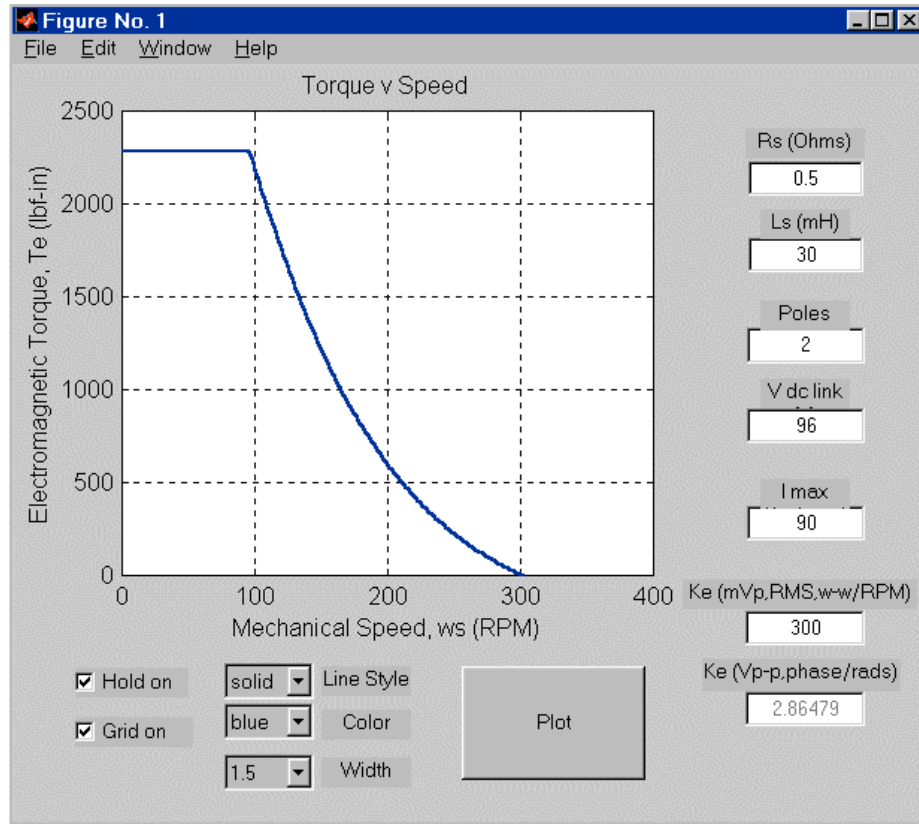


Figure 3.2. A torque-speed plotting program.

Compensator auto-tuning software is disappointingly less advanced than motor selection software. The main reason is that PID-style control loops work well enough for many applications; when the industry moved from analog control to DSP-based control new features like adjusting gains through a serial port took precedence over new control schemes that differed from the three PID knobs that engineers and operators knew how to tune. Tuning is still based on simple linear design techniques as shown in Figure 3.3.

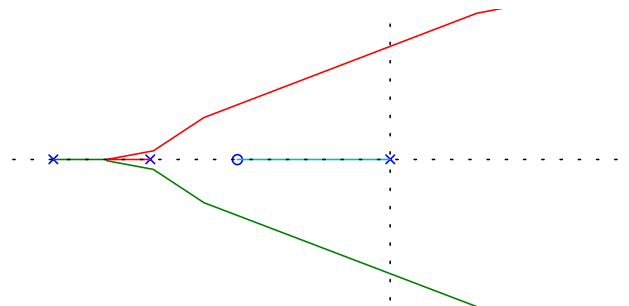


Figure 3.3. Bode Diagram of a motor with a PI current controller.

The industry has devised several interesting variations and refinements on the PID compensators in motor controllers. The first piece of the motor controller to be examined is the current stage. Table 3.2 shows the feedback typically available from a motor controllers and their sources.

Table 3.2. Feedback parameters typically available from motor controllers and their sources.

Feedback Parameter	Source
Back EMF	scaled ADC measurement, calculation from PWM signal, calculation from speed measurement
Current	Hall-Effect sensor
Acceleration	Encoder, Resolver, or acceleration-specific sensor [11] (rare)
Velocity	Encoder, Resolver, or Tachometer
Position	Encoder, Resolver, Potentiometer, or positioning device [12]

Usually voltage is manipulated to control current. The fastest changing feedback parameter is current. Change in current is impeded mostly by the inductance of the windings, and to a much smaller degree by the Back EMF, which is proportional to motor speed. All other controlled parameters, acceleration, velocity, and position, are damped in their rate of change by the inductance of the windings and the inertia of the moving system. All systems will have positive inertia, so reversing the current will always happen faster than the mechanical system can change acceleration, velocity, or position.

In practice, current can be changed more than ten times faster than the other parameters. This make it acceptable to model the entire power system, current amplifier and motor, as an ideal block that provides the requested current. Because K_t , torque per unit current, is a constant when modeling, the entire power system is usually treated as a block that provides the request torque -- especially when modeling velocity or position control system. This also has the effect of adding a layer of abstraction to the motor control system; the torque providing block may contain a Brush or Brushless motor but will have the same behavior. For the discussion that follows, the torque block may be any type of motor and torque controller.

Velocity Controllers

A typical commercial PID velocity controller as can be found in the Kollmorgen BDS-5 [13] or Delta-Tau PMAC [14] is shown in Figure 3.4. Nise [15] has a good discussion of adjusting the PID gains, K_P , K_I , and K_D . Acceleration and velocity feed forward gains and other common features beyond the basic gains are discussed below.

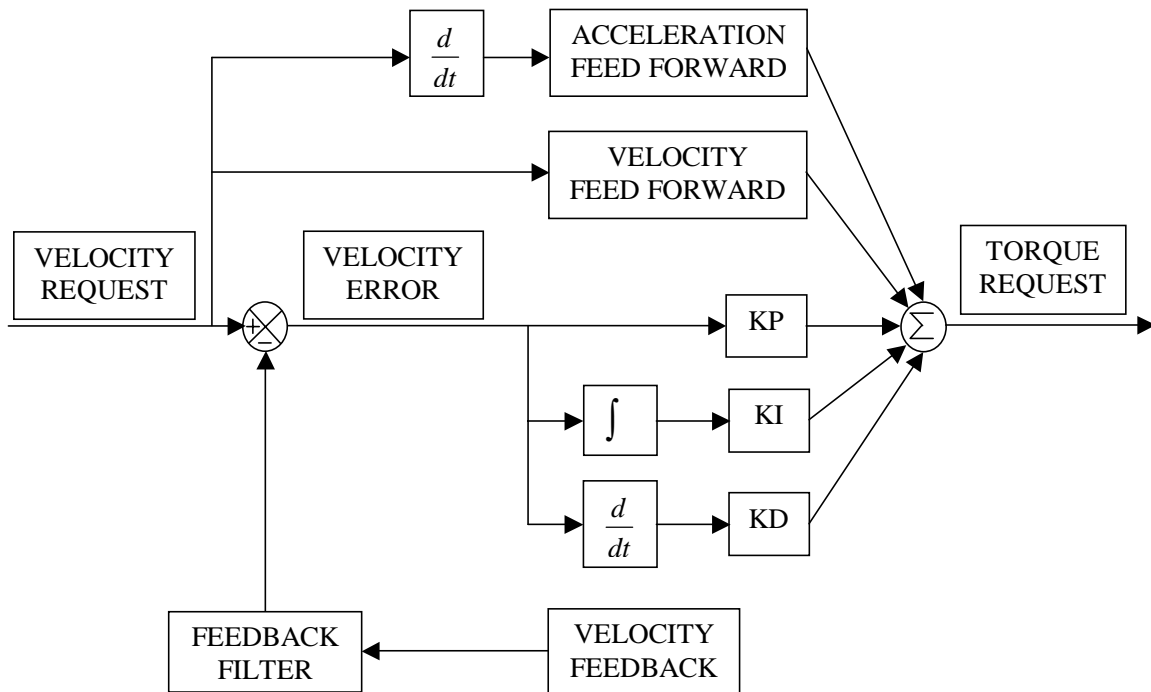


Figure 3.4. A typical commercial PID velocity controller.

Velocity feed forward gain. The basic motor model of (3.1) uses F , the rotary friction of the motor. This is a coefficient of friction modeled as linearly proportional to speed. Velocity feed forward gain can be tuned to cancel frictional forces so that no integrator windup is required to maintain constant speed. One problem with using velocity feed forward gain is that friction usually does not continue to increase linearly as speed increases. The velocity feed forward gain that is correct at one speed will be too large at a higher speed. Any excessive velocity feed forward gain can quickly become destabilizing, so velocity feed forward gain should be tuned to the correct value for the

maximum allowed speed of the system. At lower speeds integral gain will be required to maintain the correct speed.

Acceleration feed forward gain. Newton's Second Law, Force = mass * acceleration, has the rotational form, $T = J\dot{\omega}$, or Torque = Inertia * angular acceleration. For purely inertial systems or systems with very low friction, acceleration feed forward gain will work as this law suggests and give excellent results. However, it has a problem very similar to feed forward gain. Acceleration feed forward gain must be tuned for speeds around the maximum operating speed of the system. If tuned at lower speeds its value will probably be made too large to cancel out the effects of friction that are incompletely cancelled by feed forward gain in that speed region. Acceleration feed forward gain requires taking a numerical derivative of the velocity request signal, so it will amplify any noise present in the signal. Acceleration feed forward, like all feed forward gains, will cause instability if tuned slightly above its nominal value so conservative tuning is recommended.

Integral Windup Limits. Most controllers provide some adjustable parameter to limit integral windup. The most commonly used and widely available, even on more expensive controllers, is the integral windup limit. The product of error and integral gain is limited to a range within some windup value. At a maximum this product should not be allowed to accumulate beyond the value that results in the maximum possible torque request. The integral windup is often even expressed as a percentage of torque request. Any values below one hundred percent has the desired effect of limiting overshoot, but this same limit will allow a steady-state error when more than the windup limit worth of torque is required to maintain the given speed.

The second most popular form of integral windup limiting is integration delay. When there is a setpoint change in the velocity request the integrated error is cleared and held clear for a fixed amount of time. The premise is that during the transient the other gains of the system, mostly proportional and acceleration feed forward, will bring the velocity to the new setpoint and the integrator will just wind up and cause overshoot. This delay works if the system only has a few setpoints to operate around and if the transient times between each setpoint are roughly equal. There are many simple and complex

schemes that could calculate a variable length delay and greatly improve upon this method.

The best method of integrator windup limiting is to limit the slew rate of the velocity request to an acceleration that the mechanical system can achieve. This is illustrated in Figure 3.5. Figure 3.5a is a step change in velocity request. The motor, having an inertial load whose speed cannot be changed instantaneously and a finite torque limited by the current available, can not be expected to produce a velocity change that looks better than Figure 3.5b. During the transient the error is large and the integrator is collecting the large windup value that will cause overshoot. If the velocity request of Figure 3.5a can change with a slew rate equal to the maximum achievable acceleration of the system, the slope of the transient in Figure 3.5b, the error will be small during the entire transient and excessive integrator windup will not accumulate. Most commercial velocity loops have programmable accelerations limits so that an external device may still send the signal of Figure 3.5a and the controller will automatically create an internal velocity request with the desired acceleration limit.

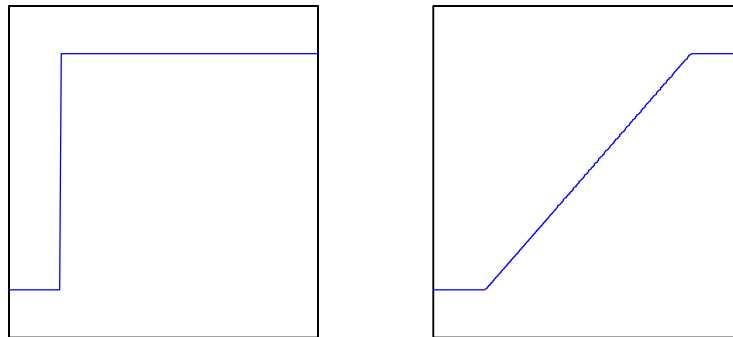


Figure 3.5a (left). A step change in velocity. 3.5b. The best possible response of the system.

In addition to a programmable acceleration limit, many commercial controllers allow separate acceleration and deceleration limits, or different acceleration limits in each direction. Either these limits must be conservative limits or the acceleration and deceleration in each direction must be invariant, requiring an invariant load. The problem of control with a changing torque load or inertial load will be discussed in Chapter 4.

Position Controllers

Figure 3.6 shows the block diagrams of three popular position loop configurations. Figure 3.6a shows the typical academic method of nesting faster loops within slower loops. The current loop is still being treated as an ideal block that provides the requested torque. This configuration treats the velocity loop as much faster than the position loop and assumes that the velocity changes very quickly to match the compensated position error. Academically, this is the preferred control loop configuration. This is a type II system, the integrators in the position and velocity loops can act together to provide zero error during a ramp change in position. This configuration is unpopular in industry because it requires tuning a velocity loop and then repeating the tuning process for the position loop. It is also unpopular because there is a tendency to tune the velocity loop to provide the quickest looking transient response regardless of overshoot; the ideal velocity response for position control is critical damping.

The assumption that the velocity of a motor control system changes much faster than position is based on the state-variable point of view that velocity is the derivative of position. Acceleration, which is proportional to torque, is the derivative of velocity and acceleration and torque definitely change much faster than velocity or position. However, when tuning systems where small position changes are required, the system with the compensator of Figure 3.6b, which forgoes the velocity loop altogether, often outperforms the system using the compensator of Figure 3.6a. Small position changes are defined as changes where the motor never reaches the maximum velocity allowed by the system. Most motor control systems are tuned to utilize the nonlinear effects discussed in the following sections, and when position moves are always of the same length these nonlinear effects make the results of tuning a system with either the compensator of Figure 3.6a or the compensator of Figure 3.6b look identical.

Figure 3.6c is a compensator that provides both a single set of gains and an inner velocity loop. This type of compensator is popular on older controllers. The compensator of Figure 3.6a can be reduced to the compensator of Figure 3.6c by adjusting K_P to unity and all other gains to zero in the velocity compensator.

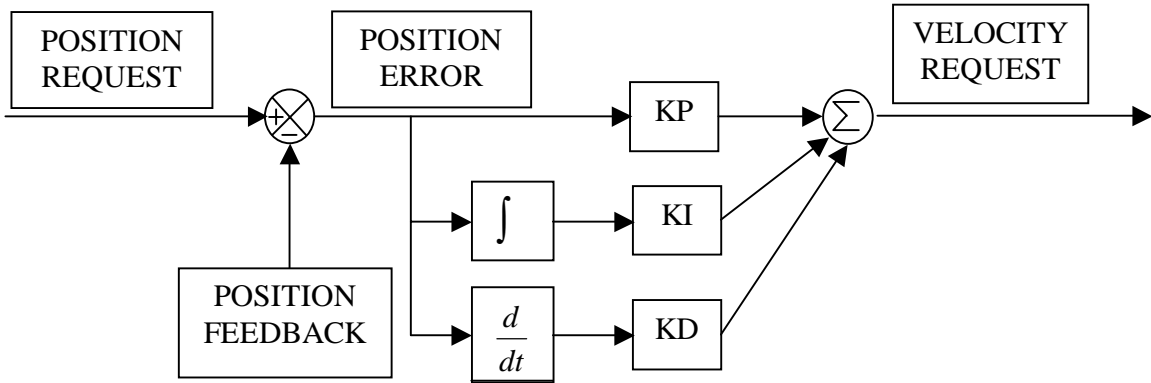


Figure 3.6a. A popular position compensator.
The velocity request becomes the input to the compensator shown in Figure 3.4.

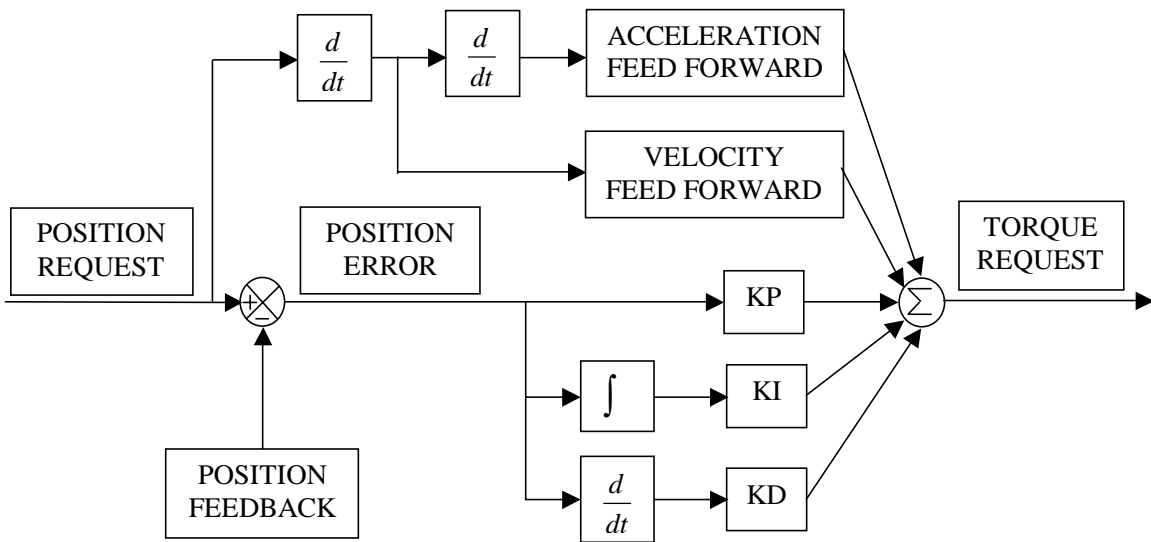


Figure 3.6b. A popular position compensator in wide industrial use.

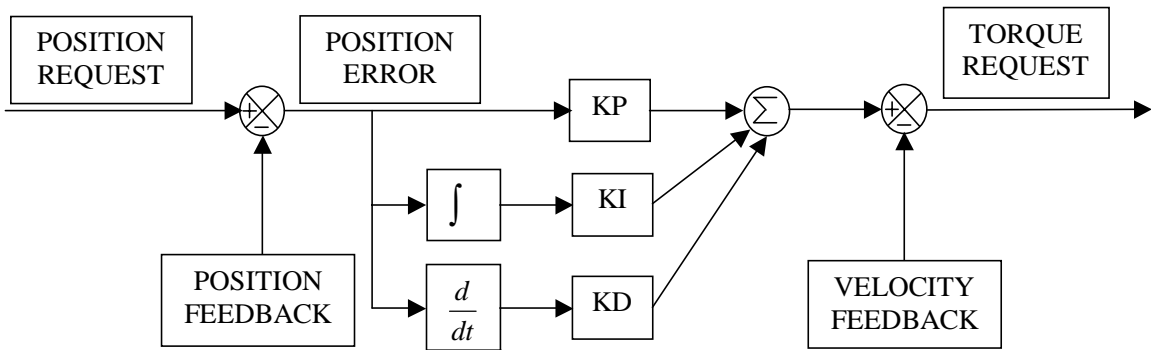


Figure 3.6c. A popular position compensator before the compensator of Figure 3.6b.

S-curves

Many of the familiar concepts of position and velocity control are based on the assumption of linear compensators and motors. An unignorable nonlinearity of motor control systems is their limited velocity and limited available torque. In a linear model a change in velocity can be made arbitrarily fast by increasing the compensator gains indefinitely. In an actual system the current will quickly reach a saturation point. A system can be tuned to operate in its linear region most of the time and display a linear response. However, the goal of the servo system designer is often to minimize transient times, and transient times are often minimized by sending a fully saturated torque request and using all the torque available.

These two viewpoints are illustrated in Figure 3.7. The same change in a velocity setpoint has been sent to the velocity request of two motor control systems, one tuned to operate in the linear region and one tuned to utilize saturation effects. From the linear viewpoint, the ideal response is the critically damped response on the left. This response is produced by the smoothly decaying torque below. From the non-linear viewpoint the ideal response on the right has used the full current available for the entire transient and reached the new setpoint in a finite time.

The velocity responses of both systems in Figure 3.7 have the same initial slope, corresponding to an identical maximum acceleration. If the gains of the linear system are increased the torque curve will start to saturate and the velocity response will have constant acceleration for longer and longer parts of the move. However, the gains will have to be increased indefinitely to approach the response of the nonlinear system. The problems with very high gains and alternative methods of achieving the same response will be discussed later.

Though both system have the same maximum acceleration they do not have the same jerk. Jerk is the change in acceleration and is used as a measure of passenger comfort in a moving machine. Jerk is controlled by derivative gain in linear systems and by S-curves in nonlinear positioning systems. In any motor control application, it is important to choose the right jerk for the job.

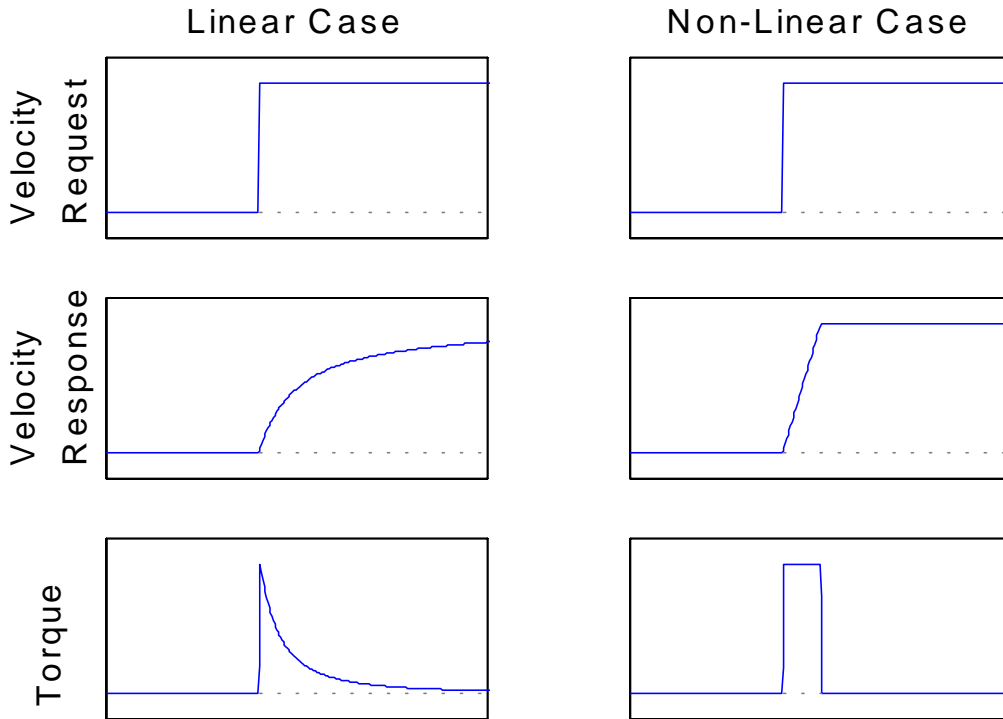


Figure 3.7. Two different points of view of the ideal velocity response of a purely inertial system.

The best way to minimize jerk is to tune the system in its linear range with smooth inputs. This will create a smooth position or velocity profile, and this profile is usually continuously differentiable until smooth acceleration and jerk profiles are obtained. Jerk is also effectively controlled by setting acceleration limits: the maximum possible jerk is a change from maximum acceleration to maximum deceleration.

Acceleration limits are the preferred method of controlling the velocity profile of a motor control system set up as a velocity regulator. However, in a positioning system, or servo system, motors are tuned to operate in the popular nonlinear case on the right hand side of Figure 3.7 and are given nonlinear inputs. In servo systems jerk is usually controlled by choosing an S-curve. The typical S-curves, known as None, Partial, and Full for their limits on acceleration, are shown in Figure 3.8 along with their resulting velocity and jerk. With no S-curve the maximum available acceleration is used to complete a velocity change or move in the minimum possible time. With a partial S-

curve acceleration and deceleration are set at a constant that can be the maximum available and a velocity limit imposed. With the full S-curve the acceleration and deceleration are adjusted so that a maximum velocity is reached at a single moment during the move.

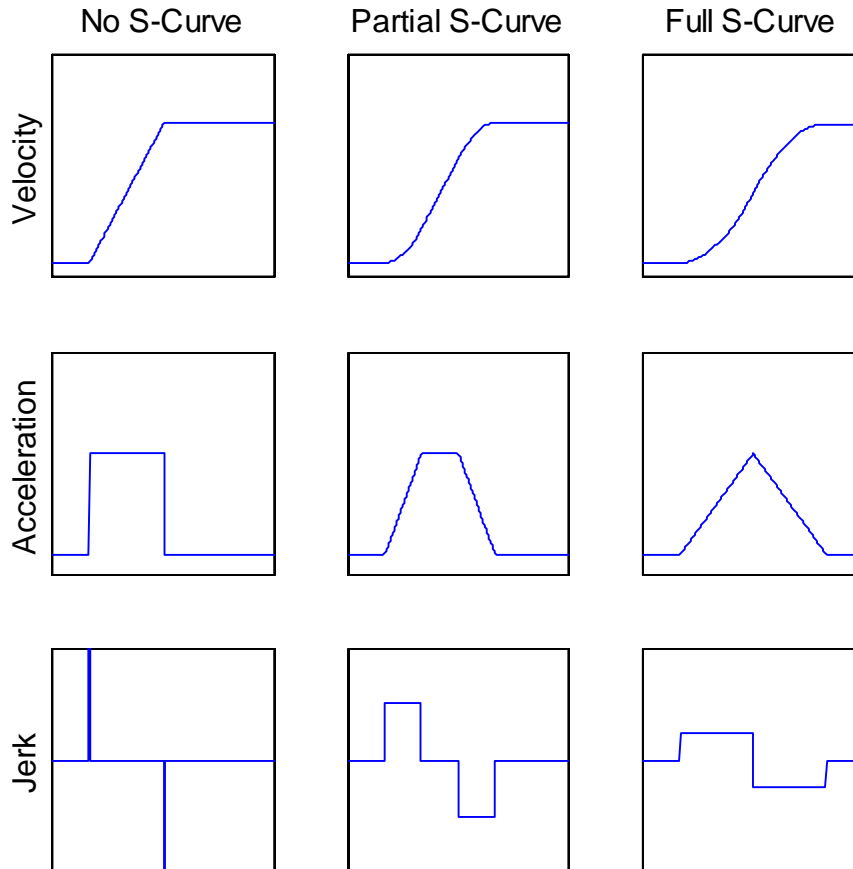


Figure 3.8. S-curves profiles resulting in the same velocity.

The advantages of using S-curves over acceleration and deceleration limits are not fully apparent from the velocity profiles in Figure 3.8. The full advantage is shown in Figure 3.9 where position curves are added to the graphs. Using S-curves the start and end of the transient look opposite but identical, avoiding the overshoot problems present in a linear compensator system that exhibits second order response. In practice a linear compensator as a positioning system can only be critically damped for a position move of a fixed distance. Shorter moves will have overdamped response and longer moves will have overshoot. Overshoot is particularly unacceptable in systems such as CNC milling machines where the result is cutting into a part, so the use of S-curves is imperative.

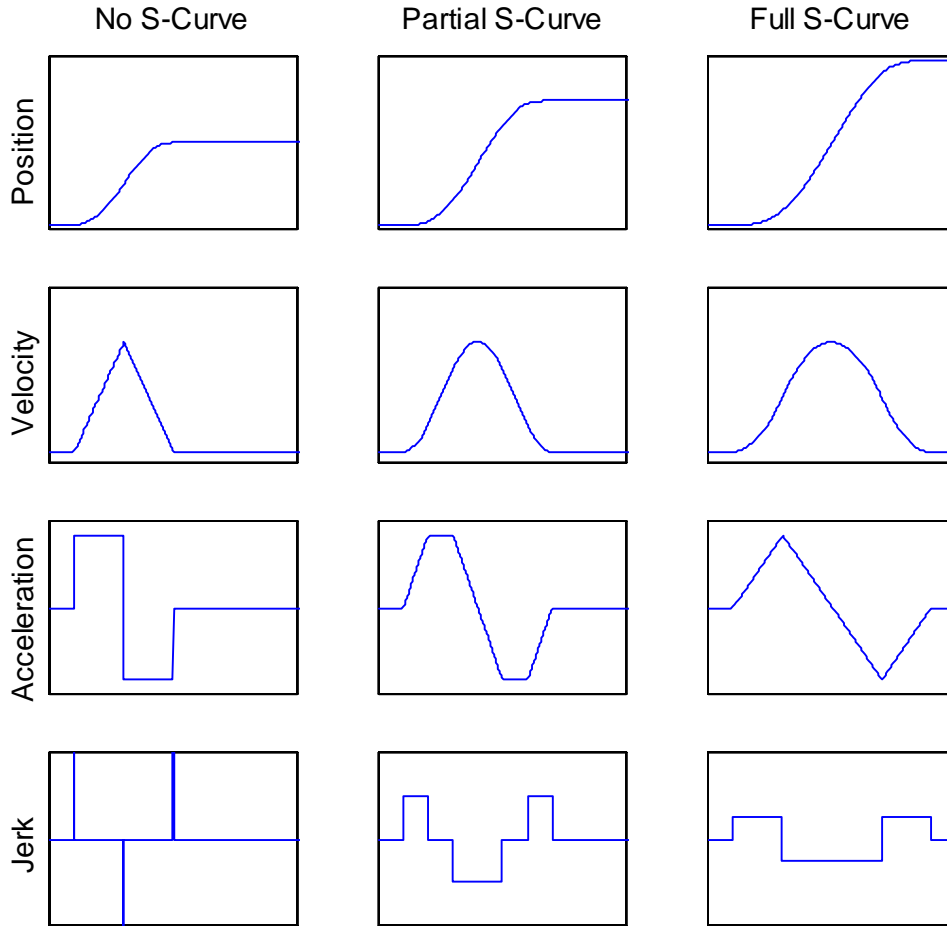


Figure 3.9. S-curve profiles that reach the same velocity and return to rest.

Control using S-curves is significantly more involved than control using linear compensators because there are many position errors that correspond to the same request to accelerate to the maximum velocity. A system of nested control loops does not inherently contain the knowledge of when to start decelerating to reach the final position just as velocity reaches zero. In classic controls this is referred to as the problem of the double integrator. This problem is still present in the S-curve system; it is illustrated in Figure 3.9 by the fact that the three acceleration graphs, each of which brings the velocity from zero to the same maximum value and then back to zero, do not all result in the same change in position. The industry solution is that the entire velocity profile must be calculated before the move begins.

The No S-curve

A velocity profile can be computed using the four basic kinematic equations for constant acceleration as found in Cutnell and Johnson [16]:

$$v = v_o + at \quad (3.2)$$

$$x = x_o + \frac{1}{2}(v_o + v)t \quad (3.3)$$

$$x = x_o + v_o t + \frac{1}{2}at^2 \quad (3.4)$$

$$v^2 = v_o^2 + 2ax \quad (3.5)$$

The parameters used above or in the following equations are:

x = position	d = deceleration
v = velocity	t_1 = acceleration time
a = acceleration	t_2 = deceleration time

Also, the following subscripts will be used: o for initial, d for the point of maximum velocity, and f for the final. Three examples of S-curve calculations are presented below. They are included because examples of straightforward S-curve calculations are otherwise scarce in the relevant literature.

The No S-curve is named for having no velocity limit. The load accelerates for as long as possible and decelerates in time to stop at the desired position. Even using the same power to accelerate and decelerate these values may not be the same. A piece being fed into a cutting blade may decelerate much quicker than it can be accelerated. The initial and final velocities are zero, so the initial and final conditions are:

$$x_o = 0 \quad v_o = 0 \quad v_f = 0 \quad (3.6)$$

substituting the conditions of (3.6) into (3.3)

$$x_d = \frac{1}{2}at_1^2$$

$$x_f = x_d + v_d t_2 + \frac{1}{2}dt_2^2 \quad (3.7)$$

from (3.2)

$$v_d = at_1 \quad (3.8)$$

substituting (3.8) into (3.7) gives

$$x_f = \frac{1}{2}at_1^2 + at_1t_2 + \frac{1}{2}dt_2^2 \quad (3.9)$$

an equation with two unknowns, t_1 and t_2 . The relationship between these can be found from (3.2) with

$$v_f = v_d + dt_2 = 0$$

$$v_d = at_1$$

$$0 = at_1 + dt_2$$

$$t_2 = -\frac{at_1}{d} \quad (3.10)$$

substituting (3.10) into (3.9) and simplifying:

$$x_f = t_1^2 \left(\frac{1}{2}a - \frac{1}{2} \frac{a^2}{d} \right) \quad (3.11)$$

For a given distance x_f the profile accelerates with acceleration a for time t_1 found from (3.11) and the decelerated at rate d to a stop at point x_f . It is easy to see that this is true for the case where $a=-d$ and (3.11) reduces to $x_f = at_1^2$. In this case the final distance moved is twice the distance moved during the acceleration, as shown in the No S-curve of Figure 3.9.

The Partial S-curve

The Partial S-curve is more practical than the No S-curve because it utilizes a velocity limit. Most systems have a safe velocity limit whether it has been specified or

not, so the No S-curve becomes increasingly dangerous on larger and larger moves where the maximum velocity increases.

Equations (3.2) through (3.5) are not sufficient for calculating the Partial S-curve because they assume constant acceleration with zero jerk. They are derived by taking $a = dv/dt$ and integrating twice with respect to time. Starting with $j = da/dt$ and integrating thrice with respect to time yield:

$$a = a_o + jt \quad (3.12)$$

$$v = v_o + a_o t + \frac{1}{2} jt^2 \quad (3.13)$$

$$x = x_o + v_o t + \frac{1}{2} a_o t^2 + \frac{1}{6} jt^3 \quad (3.14)$$

Where the new parameter j is jerk.

The first step is to find the velocity after completing a jerk to maximum acceleration using (3.12) and then (3.13). If this velocity is greater than the maximum velocity, the Full S-curve case should be used for the move. In this example the system is not yet at full velocity after a jerk to full acceleration.

The final conditions for acceleration, velocity, position, and time become the initial conditions for the next leg of the acceleration profile, the part at full positive acceleration. The same relative change in velocity will occur during the negative jerk to zero acceleration as did during the positive jerk to full acceleration, which is now known. Taking the difference between the full velocity and twice the velocity change during the positive jerk yields the velocity change required during the period of full acceleration. From this the duration of the maximum acceleration segment of the acceleration profile can be obtained.

The final conditions after the steps above again produce the initial conditions for the next step. All the parameters should be recalculated after the acceleration is jerked back down to zero. The system is now at maximum velocity and no acceleration. To reverse the acceleration profile at this point in time and bring the system back to a stop requires exactly as much distance as already traveled. If the distance traveled is already more than half the total distance change requested, the Partial S-curve will have to be

recalculated by finding the maximum velocity that is actually reached before reversing the profile and bringing the system to a stop. In this example the distance traveled at this point is less than half the total distance requested.

The distance that must be added to the profile is the difference between the total distance requested and twice the distance traveled to this point. The system currently has maximum velocity and zero acceleration, so $t = d/v$ is all that is required to find the additional time at full velocity. After this time the initial jerk profile is inverted and repeated to bring the system to a stop at the requested position.

The Full S-curve

The Full S-curve minimizes the maximum jerk by spreading out the jerk over the entire length of the move. In this example the jerk is assumed to have equal magnitude in both directions. As shown by the symmetry of the Full S-curves in Figure 3.9, it is only necessary to compute when jerk and acceleration profile to the point of maximum velocity; the deceleration half of the profiles are symmetrical. One new equation that is useful here is the constant jerk analogue of (3.4):

$$x = x_o + v_o t + \frac{1}{2}(\bar{a})t^2 \quad (3.15)$$

Where \bar{a} is the average acceleration $\bar{a} = \frac{1}{2}(a_o + a_f)$. This is applicable for the same reason that (3.4) is applicable in the constant acceleration case.

Results of S-curves

For the velocity profiling of Figure 3.8 S-curves may be applied by simply limiting the jerk. Figure 3.9 shows that S-curves that produce the same maximum velocity do not all result in the same position change. The result of applying the example calculation to compute profiles with the same position change are shown in Figure 3.10.

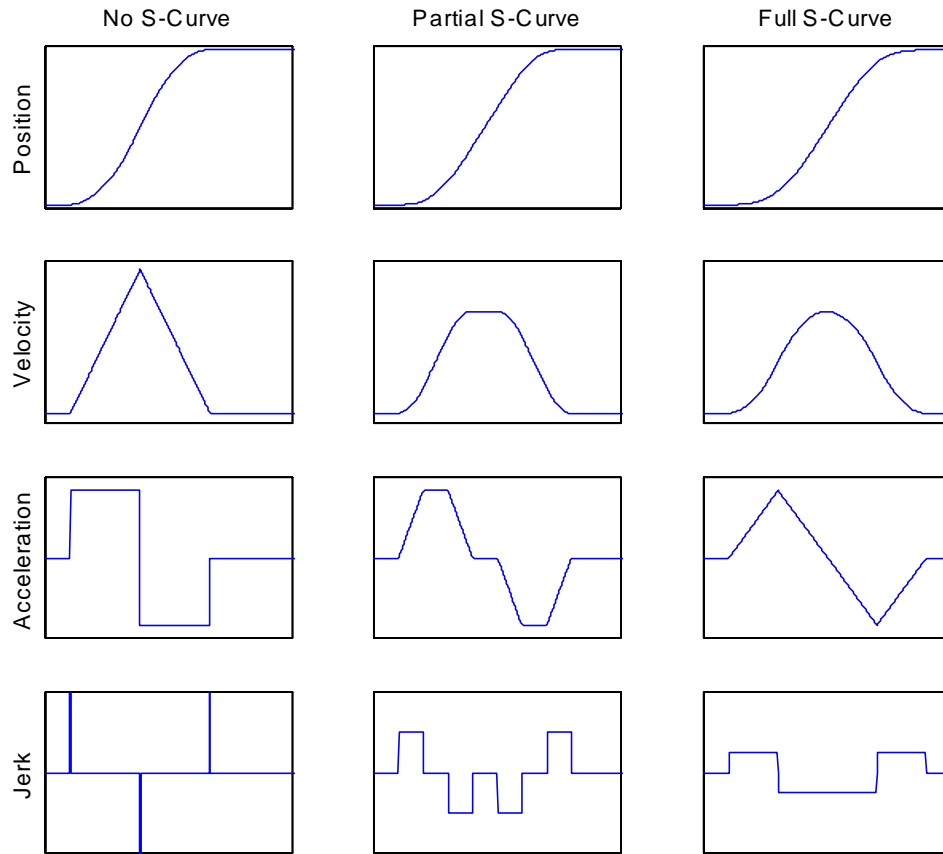


Figure 3.10. S-curve profiles that reach the same position.

S-curves rely on knowledge of the maximum possible acceleration and deceleration of the system. These values are found experimentally and assumed to be invariant after tuning. Most commercial systems rely on the linear velocity loops discussed above to produce the velocity requested by the profile. The best way to deal with large disturbances is to recalculate the profile in real-time taking the measured feedback as the initial conditions of the new profile. A better profile could be plotted if the controller could observe the new acceleration and deceleration limits of the system. These factors are affected by the inertia and torque of the load, and a method of observing these parameters would increase system performance.

Chapter 4. The State of Motor Control Academia

Motor Modeling, Reference Frames, and State Space

The Velocity/Volts transfer function (3.1) describing the motor control block diagram of Figure 3.1c is insufficient for modeling the nonlinearities and disturbances of interest in a system. State space modeling will be required. In frequency domain notation the impedance of an inductor is $Z=Ls$. The differential equation for an inductor is $v(t) = L di/dt$. In state space notation the function dx/dt is written in shorthand as \dot{x} or in code as x_dot for all x . The state equation for an inductor is then $\dot{x} = 1/L u$ where x is the state (here $x = \text{current}$) and u is the input (here $u = \text{voltage}$). See Bay [17] for a complete discussion of state space.

The state space equations for a brush DC motor are

$$\begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{J} \left(F + \frac{Kt \cdot Kb}{R} \right) \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{Kt}{R \cdot J} \end{bmatrix} v \quad (4.1)$$

$$y = [0 \quad 1] \begin{bmatrix} \theta \\ \omega \end{bmatrix} \quad (4.2)$$

Where the new parameters are:

$\theta = \text{electrical angle (rad)}$

$y = \text{desired output } \omega$

$\omega = \text{electrical velocity (rad/s)}$

These equations have the standard form:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \quad (4.3)$$

$$y = \mathbf{C}\mathbf{x} \quad (4.4)$$

The state space equations can be expanded out into state equation of the form:

$$\begin{aligned}\dot{\theta} &= 0 \cdot \theta + 1 \cdot \omega + 0 \cdot v \\ \dot{\omega} &= 0 \cdot \theta + -\frac{1}{J} \left(F + \frac{Kt \cdot Kb}{R} \right) \cdot \omega + \frac{Kt}{R \cdot J} v \\ y &= 0 \cdot \theta + 1 \cdot \omega\end{aligned}\tag{4.5}$$

In state equation form nonlinearities can be added. In (4.6) below a voltage limit has been added by adding a min operator to choose the lesser of two absolute values and a sign operator has been added to return the absolute value to its original sign. The same limit can be implemented with a set of if-then rules.

$$\begin{aligned}\dot{\theta} &= 0 \cdot \theta + 1 \cdot \omega + 0 \cdot v \\ \dot{\omega} &= 0 \cdot \theta + -\frac{1}{J} \left(F + \frac{Kt \cdot Kb}{R} \right) \cdot \omega + \frac{Kt}{R \cdot J} \cdot \min(\text{abs}(v), v_{\max}) \cdot \text{sign}(v) \\ y &= 0 \cdot \theta + 1 \cdot \omega\end{aligned}\tag{4.6}$$

Transformations such as the bilinear transformation can change state equations from the continuous domain to the discrete domain. The numerical values in the equations will change based on the sampling time and the meaning of \dot{x} will change based on the domain used. The latter differences are shown in Table 4.1. The equations in this chapter are developed in the continuous domain but were simulated and implemented with discrete time simulators and digital signal processors. The actual plant in question, a motor and load, exists in an analog world. The choice of continuous or discrete domain and the appropriate transform is a subject of continuing academic work.

Table 4.1. Transformations between different domains are possible.

Symbol	\dot{x}
Meaning in continuous domain	$\frac{dx}{dt}$
Meaning in discrete domain	$x(n) - x(n-1)$

DC brushless motors are driven by 3-phase AC power and are synchronous machines; their velocity is proportional to their input frequency. The standard model of a synchronous machine is constructed in the dq, or direct/quadrature, reference frame, as shown in Figure 4.1. In this frame the “direct” current is that which produces force directly out from the magnet in the radial direction. Such force holds the rotor in the center of the motor and is considered wasted; it is almost immediately converted into heat. The quadrature current pushes each magnet of the rotor perpendicular (thus the term quadrature) to the direct force, producing the electromagnetic torque of the motor. The abc reference frame looks at the signal on the motor leads. The dq reference frame rotates with the motor.

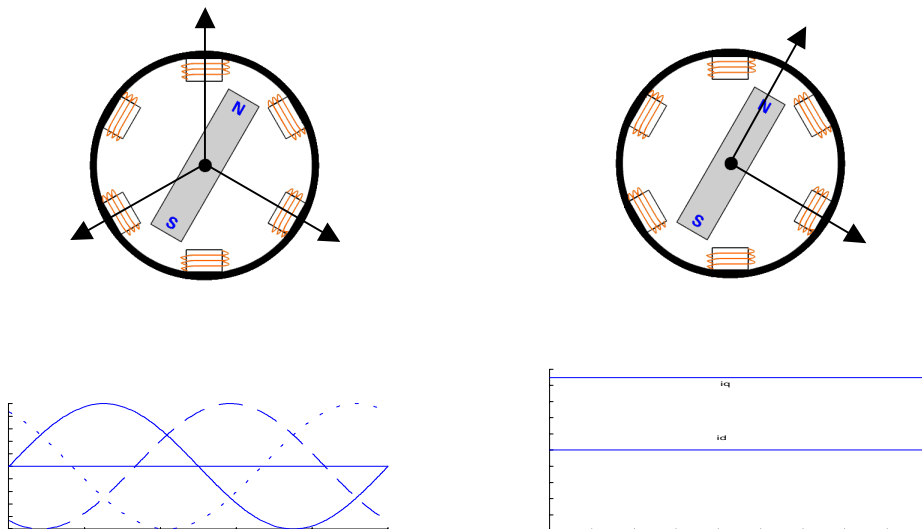


Figure 4.1. The stationary abc reference frame (left) and the rotating dq reference frame.

Values can be converted from the three phase (abc) reference frame to the dq reference frame with the Park-Clarke [18] transform. The same transform applied whether the values are voltage, current, or flux. The transform is:

$$\begin{bmatrix} q \\ d \\ o \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos(p\theta) & \cos(p\theta - 2\pi/3) & \cos(p\theta + 2\pi/3) \\ \sin(p\theta) & \sin(p\theta - 2\pi/3) & \sin(p\theta + 2\pi/3) \\ 1 & 1 & 1 \end{bmatrix} * \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad (4.7)$$

The inverse Park-Clarke transform can be performed by

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \cos(p\theta) & \sin(p\theta) & 1 \\ \cos(p\theta - 2\pi/3) & \sin(p\theta - 2\pi/3) & 1 \\ \cos(p\theta + 2\pi/3) & \sin(p\theta + 2\pi/3) & 1 \end{bmatrix} * \begin{bmatrix} d \\ q \\ o \end{bmatrix} \quad (4.8)$$

Where θ is the rotor mechanical angle and p is the number of pole pairs. The phase o is provided to make the transformation matrix square and is assumed to be zero for the balanced load cases considered here.

The model for a synchronous machine is then as given by Leonard [19]:

$$\frac{d}{dt} i_d = -\frac{R}{L} i_d + p\omega i_q + \frac{1}{L} v_d \quad (4.9)$$

$$\frac{d}{dt} i_q = -\frac{R}{L} i_q - p\omega i_d + \frac{1}{L} v_q - \omega \frac{1}{L} \frac{2}{3} Kt \quad (4.10)$$

$$\frac{d}{dt} \omega = \frac{Kt i_q - F\omega - Tl}{J + Jl} \quad (4.11)$$

$$\frac{d}{dt} \theta = \omega \quad (4.12)$$

R and L are the stator winding resistance and inductance, Kt is the torque sensitivity, J is the rotor inertia, F is the friction factor, and p is the number of pole pairs. The states i_d and i_q are the currents, ω is the mechanical angular velocity, and θ is the mechanical angle. The voltages v_d and v_q and the torque and inertial loads Tl and Jl are the inputs.

The simulated model using these equations was compared to an actual motor with both being given the same current input to create changes in the velocity setpoint. The motor and model had near-perfect agreement at moderate and high speeds but at low speeds the model predicted up to ten percent more energy in the final spinning load than possessed by the actual system. This variation is attributed to an imperfect model of friction. Three common models of friction are shown in Figure 4.2.

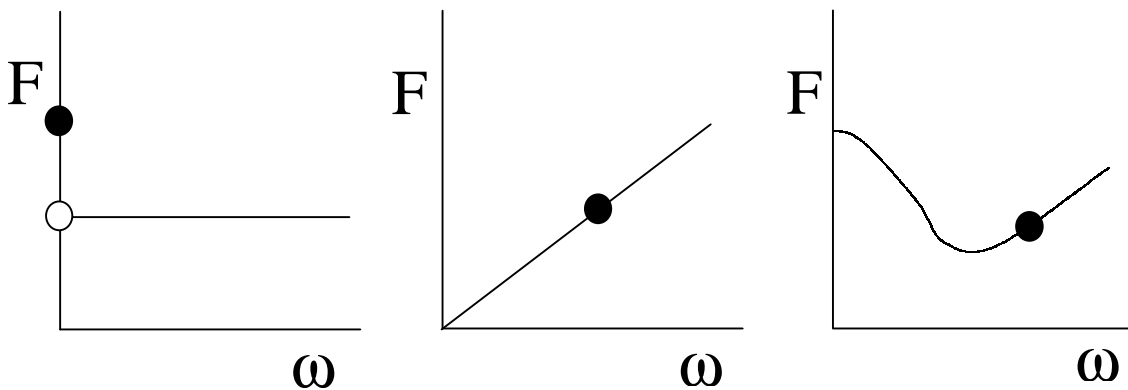


Figure 4.2. Three models of friction. 4.2a (left). Static and sliding friction. 4.2b (center) Friction as a linear function of velocity. 4.2c. Friction as a complex function of velocity.

Figure 4.2a shows the model of friction used in physics classes in which there is one static coefficient and one sliding or rotating coefficient. Figure 4.2b is the model used here where friction is a linear function of velocity. Figure 4.2c shows a likely actual model for friction as a function of velocity. The model from Figure 4.2b used in simulation is adjusted to agree with the actual friction in the system at moderate and high speeds. Friction is therefore underestimated at low speeds, accounting for the extra energy in the simulation in this region. Real control systems must be robust enough to

account for this incongruity between the simulation and reality. The robustness of various control systems will be discussed throughout the remainder of Part I.

Control Methodologies

The voltage applied to the motor is the controlling input to the motor and load plant. In high quality motors the parameters do not drift far from their nominal values. The torque load and inertial load may vary from nothing to the limits of what the motor can move. In [20] Chung et al. demonstrate that a changing inertial load can be treated as a transient torque load. This is visualizable by considering inertia as an extra “push” that only has to be given to change the speed of the load. Their torque observer assumes a low inertia and observes an increase in torque load every time a speed change occurs.

A low value for the modeled inertia will result in this observed torque load and possible suboptimal performance, but an overestimated inertia will quickly result in instability as the system overreacts to a nonexistent inertia. The other modeling error that can cause instability is excessive feedforward gain. Both of these problems are easy to visualize from the Bode plot of the linear system but hold true under analysis of the nonlinear system.

In most industrial and test systems, including those considered here, a current stage is already available with a feedback system designed to deliver a requested current in i_q and drive i_d toward zero. This system will be taken as:

$$\frac{d}{dt}\omega = \frac{Kt i_q - F\omega - Tl}{J + Jl} \quad (4.13)$$

with i_q is the input and ω is the only state of this first order system and the output to be tracked. This follows Chung et. al.’s development in [20]. In [22] Lee et. al. use a similar technique to provide position control, thus repeating the exercise for a second order

system. The block diagram of the system to be controlled is shown in Figure 4.3. This is the basis of the sample output shown in the rest of this chapter.

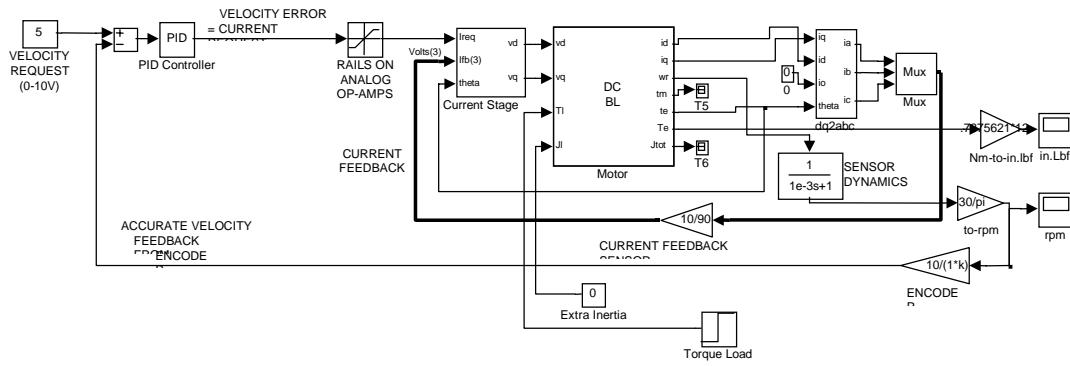


Figure 4.3. Block diagram of system to be observer and better controlled.

The design of a sliding mode controller will follow the method of Slotine and Li [25] for the simpler case of a first order system. For now the higher order dynamics have been ignored, specifically (4.9) and (4.10). Two other phenomenon are present in the simulated model that will be ignored in designing a controller. First, the current i_q , which is proportional to the electromagnetic torque by Kt , cannot be directly measured in the present implementation but the total current i can be measured. Though $i=i_q$ in the steady state, this is not true during varying current loads. This is equivalent to the synchronous machine slipping, though by far less than 90 electrical degrees. The second phenomenon is that the modeled friction is imperfect, as previously discussed. In the results to follow the effect of friction is not visible.

First a sliding mode controller will be designed to provide velocity control and it will be graphically shown why it is impractical. Then a sliding mode observer will be constructed to observe the unknown torque load. This sliding mode observer will be replaced by a high gain observer. Finally, a simple feedforward scheme will demonstrate that torque load information can be used to design a better compensator.

Design of a Sliding Mode Velocity Controller

Design begins by restating the system model in (4.13).

$$\frac{d}{dt}\omega = \frac{Kt i_q - F\omega - Tl}{J + Jl} \quad (4.13)$$

Taking $u=i_q$, $a=-F/(J+Jl)$, $b=Kt/(J+Jl)$, and $d=-l/(J+Jl)$ where $J+Jl$ is a known constant,

(4.13) may be more conveniently restated as:

$$\frac{d}{dt}\omega = a\omega + bu + d Tl \quad (4.14)$$

Let ω_{ref} be the reference velocity and choose the new state $x=\omega-\omega_{ref}$. Assuming that

$\frac{d}{dt}\omega_{ref} = 0$ leads to $\frac{d}{dt}x = \frac{d}{dt}\omega$. Substituting x and $\frac{d}{dt}x$ into (4.14):

$$\frac{d}{dt}x = ax + bu + a\omega_{ref} + d Tl \quad (4.15)$$

This has reduced the tracking control problem to a disturbance rejection problem.

Choosing the conventional sliding surface

$$s = \left(\frac{d}{dt} + \lambda\right)^{n-1} x \quad (4.16)$$

for $n=1$ gives $s=x$ as verified by a similar development by Ünsal and Kachroo in [23]. An argument for stability can be made by showing that $s \rightarrow 0 \Rightarrow x \rightarrow 0$ and

$$\frac{1}{2} \frac{d}{dt} s^2 \leq -\eta |s| \Rightarrow s\dot{s} \leq -\eta |s| \quad (4.17)$$

so for $s(0) > 0$, $\dot{s} \leq -\eta$

and for $s(0) < 0$, $\dot{s} \leq \eta$

This shows stability and gives us the control law:

$$u = c - k \cdot \text{sign}(s) \quad (4.18)$$

Figures 4.6, 4.7, and 4.8 placed just before the conclusion of this chapter show the results of the sliding mode control law compared to the other control laws about to be developed. This controller is impractical as implemented for two main reasons. First, the chattering around the reference velocity puts unacceptable mechanical wear on the actual system. Second, the maximum current (either forward or reverse) is being requested at every moment; this can actually cause some current amplifiers to fail explosively.

Design of a Sliding Mode Torque Observer

Though many solutions to the chattering problem exist, Young et. al. [24] suggests that a superb solution in practice is to construct a sliding mode observer and use the resulting observed value to perform better control. This solution was chosen because a measure of the torque load is also a desirable display on a motor control system. A sliding mode observer is any observer where high gain is implemented with switching.

Both sliding mode and adaptive controller design start by assuming the observer is faster than the relevant dynamics in Tl so that one can also assume $d/dt Tl = 0$. Regarding Tl as an extra state variable, the augmented equations can be written as:

$$\frac{d}{dt} \begin{bmatrix} \hat{\omega} \\ \hat{Tl} \end{bmatrix} = \begin{bmatrix} a & d \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{\omega} \\ \hat{Tl} \end{bmatrix} + \begin{bmatrix} bo \\ 0 \end{bmatrix} + \begin{bmatrix} k \\ -lk \end{bmatrix} \text{sign}(\sigma) \quad (4.19)$$

where the states are the measurements of their true values and σ is the sliding surface

$$\sigma = \omega - \hat{\omega} \quad (4.20)$$

The errors in the states are given a tilde and are

$$\begin{aligned} \tilde{\omega} &= \omega - \hat{\omega} \\ \tilde{Tl} &= Tl - \hat{Tl} \end{aligned} \quad (4.21)$$

Taking the derivatives of (4.21) and substituting in (4.19) yields

$$\begin{aligned} \dot{\tilde{\omega}} &= a\tilde{\omega} + d\tilde{Tl} + lk \cdot \text{sign}(\sigma) \\ \dot{\tilde{Tl}} &= lk \cdot \text{sign}(\sigma) \end{aligned} \quad (4.22)$$

so, in the sliding mode

$$\dot{\tilde{Tl}} = dl \tilde{Tl} \quad (4.23)$$

and l is chosen using pole placement methods. The author chose $l = 10,000$ as “fast enough” in the five times the second order speed sense and achieved good results in the simulation.

The augmented system is a second order system. The stability condition (4.17) requires

$$\begin{aligned} \sigma(a\tilde{\omega} + d \cdot \Delta Tl - k \cdot \text{sign}(\sigma)) &< 0 \\ \Rightarrow k &> \max \|a\tilde{\omega} + d \cdot \Delta T\| \end{aligned} \quad (4.24)$$

yielding a $k > 4111$ for the chosen sample parameters.

The final output $\hat{T}l$ is low-pass filtered at a frequency below the chattering frequency but sufficiently high to show the desired response. This simulation made feedback measurements at 100 kHz, well above the desired change in torque load and allowing for a low order filter. The performance of the sliding mode torque observer is shown in Figure 4.4 in the next section.

A High Gain Observer without Sliding Mode

A modified version of (4.19) that still works by modeling $\hat{T}l$ as an extra state variable can be used to implement a high gain observer without using a switching law for the high gain, as in

$$\frac{d}{dt} \begin{bmatrix} \hat{\omega} \\ \hat{T}l \end{bmatrix} = \begin{bmatrix} a & d \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{\omega} \\ \hat{T}l \end{bmatrix} + \begin{bmatrix} bo \\ 0 \end{bmatrix} + \begin{bmatrix} k \\ -l \end{bmatrix} \sigma \quad (4.25)$$

A step change in torque load and the resulting torque load observed by sliding mode, filtered sliding mode, and high gain observer schemes is shown in Figure 4.4. The high gain observer output does not require the filtering of the sliding mode observer output. For the high gain observer, k is directly proportional to the maximum allowable slope of ω and l is proportional to the maximum allowable slope of Tl ; these parameters can be adjusted to provide the desired filtering effect. The end result is that both observers have roughly equivalent performance.

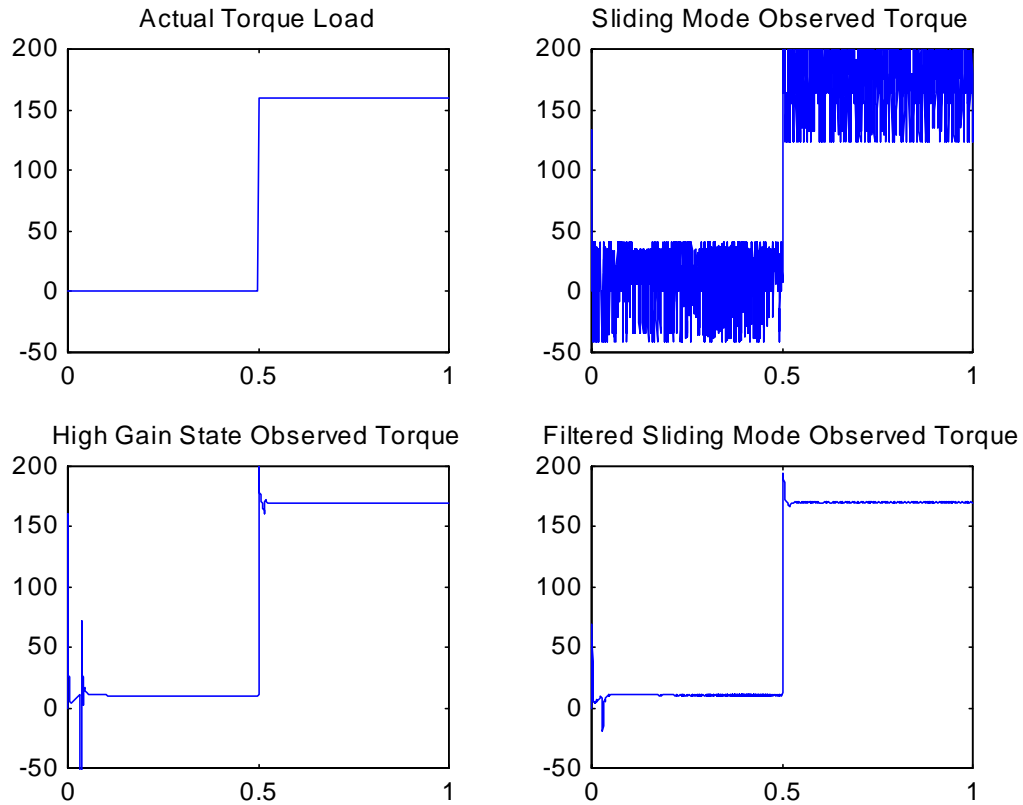


Figure 4.4. Comparison of High Gain and Sliding Mode Observers. 4.4a (upper left). A step change in torque load. 4.4b (upper right). The unfiltered output of the sliding mode torque observer. 4.4c (lower left). The output of the high gain torque observer. 4.4d. The filtered output of the sliding mode torque observer.

Use of Torque Observations in a Feedforward Compensator

A simple feedforward design is all that is necessary to demonstrate the use of torque load information in improved control. The scheme shown in Figure 4.5 uses a PI velocity controller and cancels the effects of a torque disturbance by using a torque observer. The observed torque is divided by K_t to find the current needed to cancel the disturbance. This current is then fed forward and added to the current request from the error compensator.

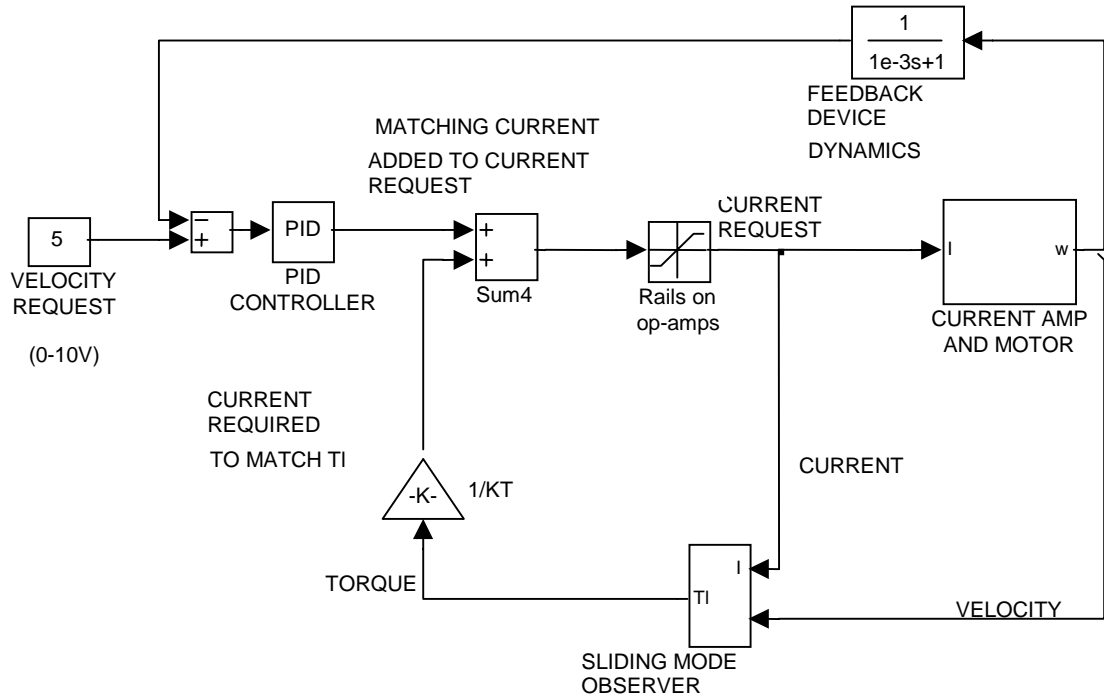


Figure 4.5. Block diagram of a system with a sliding mode observer and feedforward current compensation.

Figures 4.6, 4.7, and 4.8 compare a PI controller, a sliding mode controller, and a PI controller with a sliding mode observer and the feedforward torque cancellation scheme of Figure 4.5. The feedforward scheme achieves the disturbance rejection capabilities of the sliding mode controller while maintaining the smooth velocity and current profiles of the PI controller.

An important assumption made in the design of the torque observer was that the inertia of the system was known. Recall that for stability it was necessary to assume the minimum possible inertia of the system. Figure 4.5 shows the system operating at the assumed inertia. Figures 4.7 and 4.8 show the system operating at two and ten times the assumed inertia, respectively. The sliding mode observer and torque feedforward technique is still provide superior results at ten times the assumed inertia.

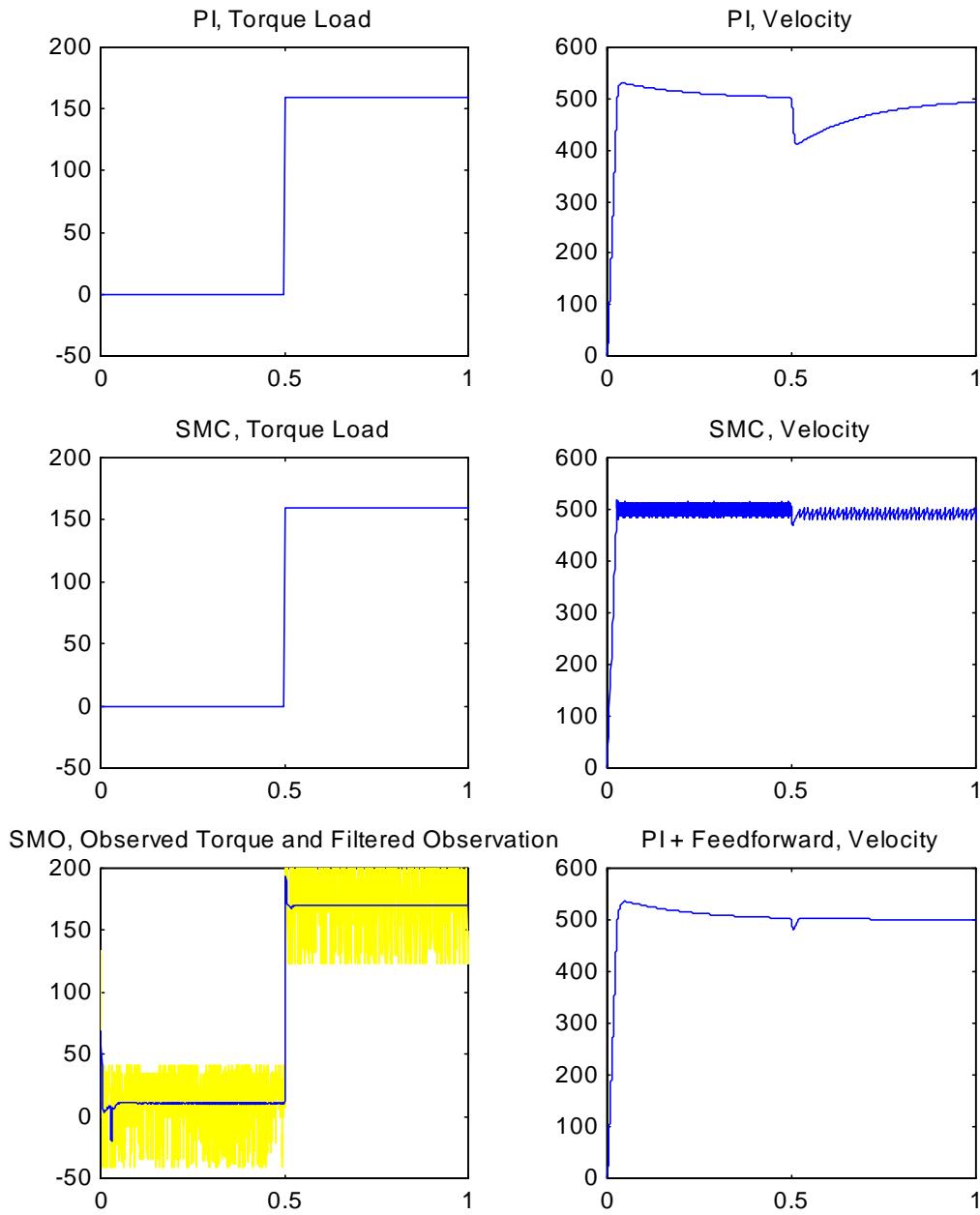


Figure 4.6. Comparison of three control strategies with $J_{actual} = 1 * J_{assumed}$ ($J=1$ p.u.)

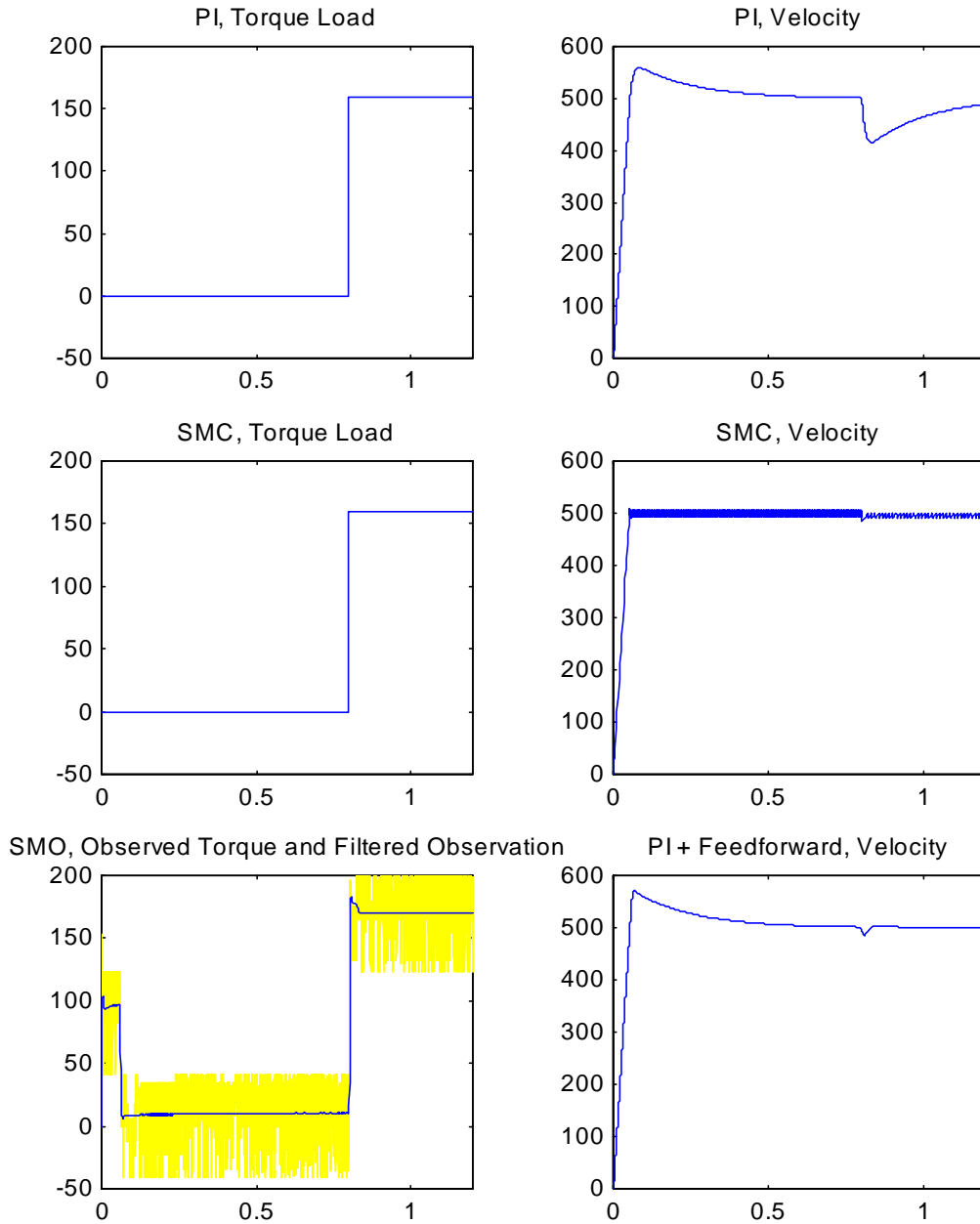


Figure 4.7. Comparison of three control strategies with $J_{actual} = 2 * J_{assumed}$ ($J=2$ p.u.)

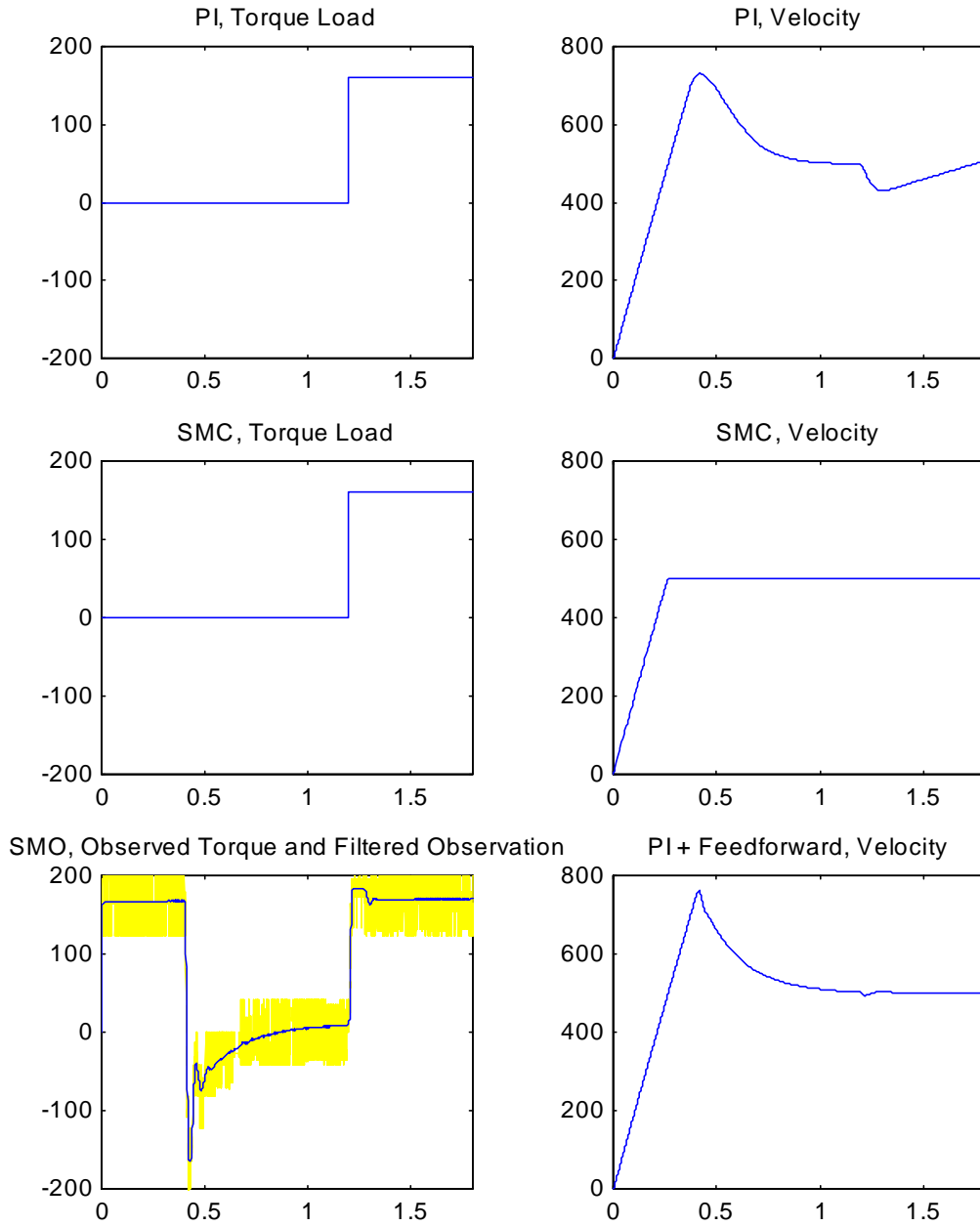


Figure 4.8. Comparison of three control strategies with $J_{actual} = 10 * J_{assumed}$ ($J=10$ p.u.)

Conclusion

This chapter covers motor modeling in the state space domain, transformations from the stationary three-phase reference frame to the rotating reference frame, and simplified state equation models of DC motors. The sliding mode controller is shown to have superior disturbance rejection over linear controllers but is impractical due to chattering. The sliding mode observer with disturbance canceling feedforward is demonstrated as a method of maintaining superior performance and removing the chattering problem. The method both improves disturbance rejection with a step disturbance and improves disturbance rejection and transient times with a sinusoidal disturbance.

Many other control schemes are possible, and every other advanced control scheme known to mankind has probably also been applied to motors. All of these schemes assume some knowledge about the inertia or the torque load. When all assumptions of knowledge about the system and its load are removed, all these advanced control schemes reduce to P or PI like schemes. With a plant with only one input and one output, even sliding mode control reduces to bang-bang control, and bang-bang control is really P control with a very high gain.

Another example of an advanced linear control scheme that turns out not to help much is H-infinity control. H-infinity control is the solution to the Problem of Differential Games, also known as the minimax problem. The problem goes like this: Assume that all parameter errors and system disturbances will combine in the worse possible way, the way that causes the most error. Find the set of feedback gains that will minimize the maximum error. These gains are the H-infinity gains. Because compensator designers are trying to minimize one thing and maximize another using differential equations, they are playing a differential game. In the case where there is only one output and one feedback measurement the H-infinity control reduces to a P controller where a particular gain is the H-infinity gain. A variation of H-infinity is to add an integrator to assure zero steady-state error. With only one output and one feedback measurement, the H-infinity control with integrator reduces to a PI controller.

All linear control techniques suffer from another problem. In PID control the input to a plant is a weighted sum of output values and the output error goes into a compensator. In LQR and LQG control the input to the plant is a weighted sum of states that have been measured or observed from the output of the plant. In the LQR control it is assumed that some Gaussian noise has been added to the output and a Kalman filter is used to calculate the states from the output.

In the PID controller proportional, integral, and derivative gains are adjusted until the plant behaves as desired. In LQR and LQG control the cost of using control, R , and the cost of error, Q , are adjusted until the plant behaves as desired. Both schemes are hard to tune analytically because of the inherent non-linearity of the plant and the quality of the feedback signal. In practice both schemes reduce to experimentally adjusting gains or costs until the system's behavior is as close to the desired behavior as possible.

The quality of the feedback signal turns out to be the limiting factor in almost all control schemes. Increasing gains reach the point where the compensator is no longer providing more negative feedback but is amplifying the noise in the system. It is common to see time wasted trying to find a set of gains that compensates for a poor quality feedback signal. Academia and industry need to place more stress on cleaning up and filtering feedback signals before attempting to optimize a system by adjusting gains. Unfortunately high quality and high signal-to-noise ratio feedback devices are expensive and extra engineering time spent turning knobs is relatively inexpensive, so this situation is not likely to change soon. It does represent an unexploited market opportunity.

One novel alternative from the usual low-pass and notch-pass filter designs of undergraduate academia that has potential in the motor control industry is the IIR predictive filter research led by S. J. Ovaska for smooth elevator control. In [26] Ovaska et. al. give a good overview of polynomial predictive filters. These filters provide smooth and delayless feedback when the motor is operating with a smooth profile but have transient errors when systems have discontinuous acceleration. In [27] Välliviita gives a method that provides the smooth predicted derivative of a signal that is useful in allowing high derivative gains. Derivative gains benefit most from this filtering process because derivation is a noise amplifying process. In [28] Välliviita and Ovaska solve some of the

problems Väliiviita had in [27] with the varying DC gain of the filter. With future papers this method may become more applicable to general purpose motor control.

There are some control schemes that can improve motor control performance with the same quality feedback signal. One is the use of S-curves and the plotting of velocity profiles discussed in the last chapter to overcome the problem of the double integrator. Another is the two degree-of-freedom (DOF) PID controller in which two PID controllers are used to separately control the characteristics of the transient and the steady state response. This controller is difficult to tune and is mostly ignored because acceleration and velocity feedforward gains are already available to control the transient during a setpoint change. The advantage of 2-DOF PID is its ability to control transients caused by process disturbances, the problem addressed here. Hiroi [29] [30] [31] has received three US Patents for 2-DOF controllers and methods of implementing them easily that are in use by his company but still too complicated for more general use. These controllers have great potential in industrial control when their use becomes much simpler. The remaining methods that can achieve better control with the same information are the soft computing techniques of fuzzy logic and neural networks.

Chapter 5. Soft Computing

A Novel System and the Proposed Controller

A specific example created by Lewis et. al. in [32] will be used to show how Fuzzy Logic, a soft computing technique, can improve on the system performance achievable by either a PID or SMC controller alone. The system is a variation on the classic inverted pendulum problem. It is an inverted pendulum pinned onto a rotating disk as shown in Figure 5.1. The pendulum is free to rotate within the plane normal to the disk at the point of the pin. This plane is itself rotating with the disk.

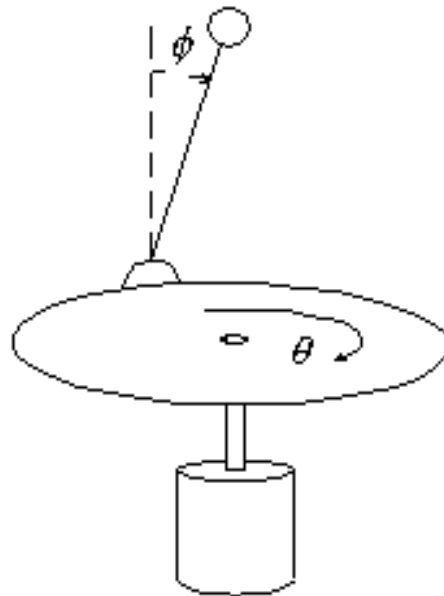


Figure 5.1. An inverted pendulum of a disk.

In [32] and [33] Lewis derived the following state equations for the system using LaGrangian dynamics:

$$\begin{aligned}
 \dot{x}_1 &= x_3 \\
 \dot{x}_2 &= x_4 \\
 \dot{x}_3 &= \frac{\tau + rlx_4^2 \sin x_2 - mrg \sin x_2 \cos x_2}{(I + mr^2 - r^2 \cos^2 x_2)} \\
 \dot{x}_4 &= \frac{(I + mr^2)mg \sin x_2 - (r\tau \cos x_2 + r^2lx_4^2 \sin x_2 \cos x_2)}{l - r^2l \cos^2 x_2}
 \end{aligned} \tag{5.1}$$

Where θ and ϕ are the angles of the disk and the pendulum, respectively, as shown in Figure 5.1 and the state variables are:

$$\begin{aligned}
 x_1 &= \theta \\
 x_2 &= \phi \\
 x_3 &= \dot{\theta} \\
 x_4 &= \dot{\phi}
 \end{aligned} \tag{5.2}$$

The parameters are:

τ = torque applied to the disk, the controlling input

r = radius of the disk

l = length to the center of mass of the pendulum

m = mass of the pendulum

g = acceleration due to gravity

A simulation of the system was created using (5.1) and both PID and SMC controllers were constructed to control the angle of the pendulum to upright. During the tuning of both controllers the authors became experts on the behavior of the system and made observations about the system such as the following:

- The pendulum angle and pendulum speed are the most important states when controlling the pendulum angle.
- When the pendulum must be righted from large angles and speeds the SMC performs best
- Once the angles and speeds are small, the PID controller performs best.

The latter observations are because the SMC works by exerting the full available torque on the disk to rotate it in one direction or the other. This works well for large errors, but when the error become small the SMC chatters the pendulum around the upright position. Contrarily, when the errors are small PID control behaves smoothly and bring the pendulum to a stop in the upright position. When the errors are large the PID control also provides a large response but creates integrator windup that can cause unnecessary overshoot and instability.

The solution proposed here is the hybrid control system of Figure 5.2. Here both a PID and SMC controller calculate a controlling torque based on the error of the pendulum angle. A Fuzzy Logic controller acts as a soft switch that decides on a weighted average of the two torques to use as the actual controlling torque based on the angle and velocity of the pendulum. As noted by the tuning experts, the disk position must be controlled much more slowly than the pendulum angle. Therefore the pendulum angle is not controlled to zero, but controlled to bring the disk position to zero with a much slower, lower gain disk position loop wrapped around it. This outer loop is a PID controller and is of little interest to object of this example, a fuzzy logic approach to getting the best qualities of two different controllers. The outer loop is included for a practical reasons: a spinning disk with an erect pendulum is dizzying to watch, hard to graph, and requires the pendulum angle measurement device to be connected wirelessly or with a slip ring.

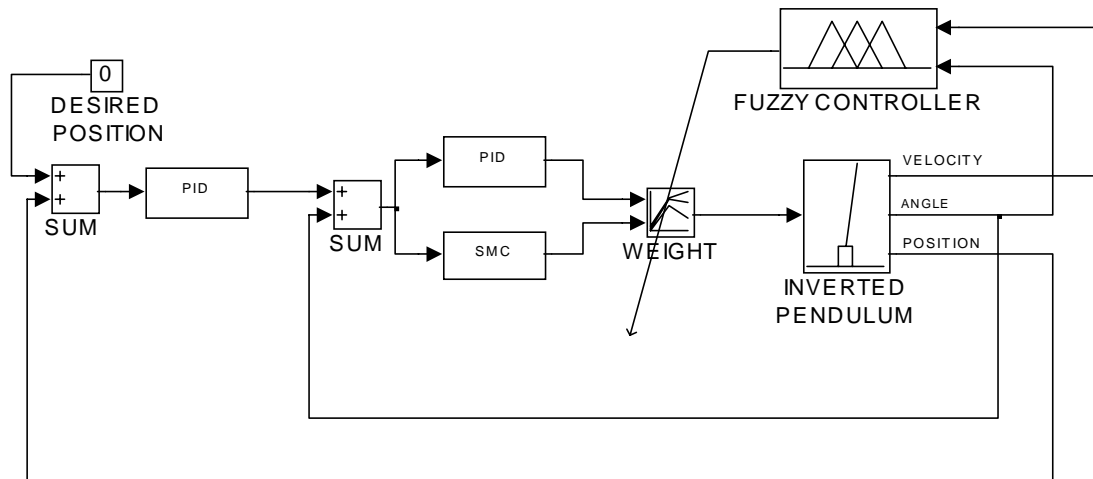


Figure 5.2. Inverted Pendulum on a disk and its control system.

The Fuzzy Controller

Experts tuning the system can come up with a set of linguistic rules describing when it is best to use which controller based on the pendulum angle and velocity. These rules can be put in an IF-THEN form using the linguistic variables Small, Medium, and Large to describe pendulum angle and velocity and the appropriate weight of the SMC controllers output. Some of these rules are:

- IF the Angle is Small AND the Velocity is Small THEN the SMC weight should be Small.
- IF the Angle is Medium AND the Velocity is Medium THEN the SMC weight should be Small.
- IF the Angle is Large AND the Velocity is Large THEN the SMC weight should be Large.

With two measured states and three linguistic variables describing each there are nine such possible rules. One of the advantage of Fuzzy Logic controllers is that it is not necessary to have every possible linguistic rule. This is especially advantageous as the number of rules increases. This and other variation and complexities of a Fuzzy Logic controller are given a thorough discussion by Jang et. al. [34]. The system here has only

nine possible rules so it is reasonable to use each rule. These rules are presented in a more compact form in Table 5.1.

Table 5.1 Weight Given to PID Controllers Torque Command.

THEN the SMC's weight is...		AND the Pendulum Velocity is...		
		Small	Medium	Large
IF the Pendulum Angle is...	Small	Small	Medium	Large
	Medium	Medium	Medium	Large
	Large	Large	Large	Large

The basis of Fuzzy Logic is that the concepts of Small, Medium, and Large can represent fuzzy sets instead of crisp sets. There is no single value above which the pendulum angle is Large. Instead, the angle has varying degrees of largeness that increase with the angle until it can be described as wholly large. In this example the angle and velocity are limited to a discrete Universe of Discord. The absolute values of pendulum angle and velocity are limited to 40° and 30 RPM respectively and discretized to one hundred distinct values. Each value is assigned an amount of smallness, mediumness, or largeness between zero and one.

The output is also described in terms of a membership functions on an output universe of discord of zero to 100 percent use of the SMC's output. The input and output membership functions are illustrated in Figure 5.3. For creating the output universe of discord 100 discrete values are used but the resulting output is not rounded to the nearest whole value. There are 10,000 possible combinations of two inputs with 100 outputs, so the entire input/output space of the system can be stored as a look-up table with 10,000 outputs. Choosing 16 bit integers for 65,536 possible outputs provided adequate resolution in simulation and results in a 20,000 byte look-up table, making it practical for implementation on a DSP. The surface mapped by this table is illustrated in Figure 5.4.

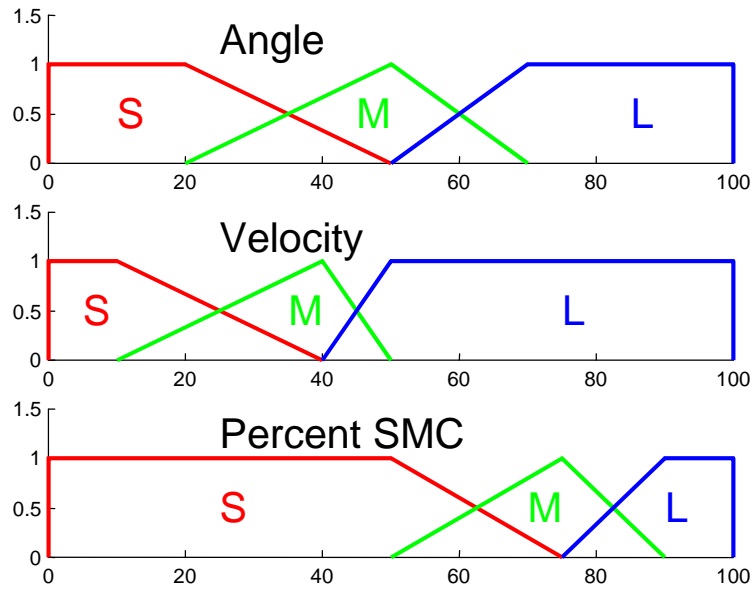


Figure 5.3. Input (Angle and Velocity) and Output (Percent SMC) Membership Functions

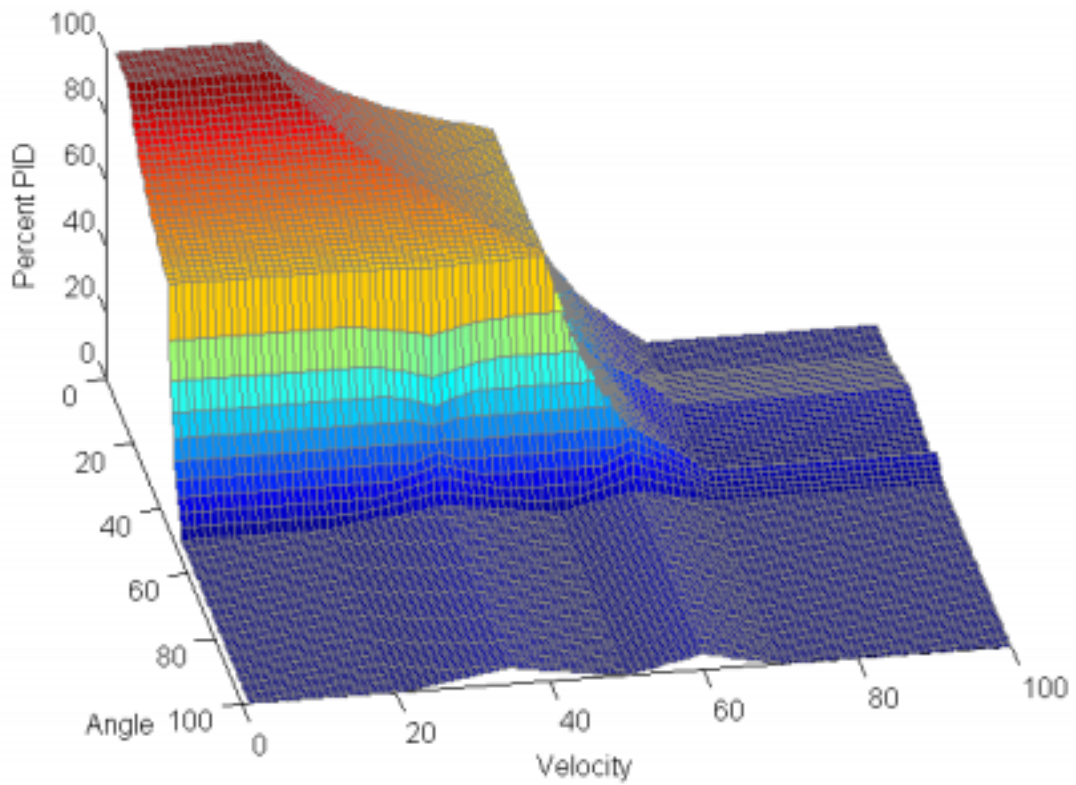


Figure 5.4. This surface maps the input/output behavior of the controller.

A fuzzy inference system is used to generate the input/output mapping. Jang et. al. [34] suggest several such systems. A simple system can be used here because of a restriction placed on the membership functions: At any velocity or angle in Figure 5.3 the sum of a given value's membership in each linguistic variable is unity.

The measured angle's membership in the Small, Medium, and Large set is calculated from the angle's Universe of Discord in Figure 5.3. The measured velocity's membership in each set is also calculated. Then each linguistic rule is evaluated. The THEN part of each linguistic rule is taken to be as true as the minimum value of each part of the ANDed conditions. This evaluation of THEN statements is shown in Table 5.2. The membership values are given for an angle and velocity both at 33 in their discrete Universe of Discord, as shown in Figure 5.3.

Table 5.2 Weight Given to PID Controllers Torque Command.

THEN the SMC's weight is...		AND the Pendulum Velocity is...		
		Small (25%)	Medium (75%)	Large (0%)
IF the Pendulum Angle is...	Small (50%)	Small (25%)	Medium (50%)	Large (0%)
	Medium (50%)	Medium (25%)	Medium (50%)	Large (0%)
	Large (0%)	Large (0%)	Large (0%)	Large (0%)

The output weight's Universe of Discord is then redrawn with each membership function's value limited to the maximum value of that membership function allowed by the linguistic rules. In this case each membership function is limited to the magnitude:

Small 25%

Medium 50%

Large 0%

In the output universe the resulting shape is the yellow area shown in Figure 5.5. The x centroid of this shape, \bar{x} , is used as the output value of the system. The range of possible outputs is scaled so that centroid of the purely small shape results in 100% PID control and the centroid of the purely large shape results in 100% SMC control. For the example in Figure 5.5 the x centroid corresponds to about half PID control, which is consistent with that point on the mapping in Figure 5.4. Many “centers” of the shape other than the centroid, such as the mean of the maximum value, may be used.

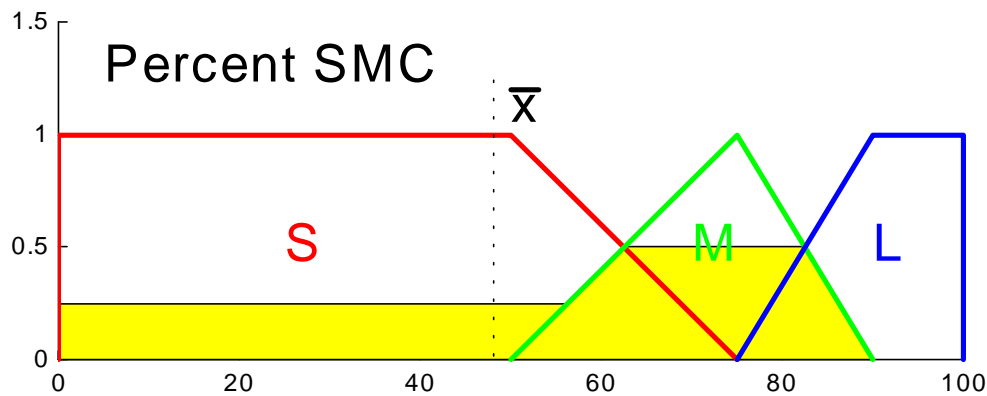


Figure 5.5. The final shape used to calculate the output and its centroid.

Results and Conclusion

Simulation results for the system are shown in Figures 5.6, 5.7, and 5.8. Figure 5.6 shows that for a small initial displacement of 10° the hybrid controller behaves similar to the PID controller and the SMC controller has problems with oscillations around the setpoint.

Figure 5.7 shows a moderate disturbance of 25° . Here the PID still slowly converges and the SMC converges quickly but oscillates. The hybrid controller shows the best response by any of the usual measures, it both converges quicker and has less overshoot than either of the other methods.

Figure 5.8 shows a large disturbance of 45° . Here the PID actually goes unstable, falls down to 180° , and keeps spinning the disk. The hybrid controller is still stable, converges quickly, and does not oscillate like the SMC.

This chapter shows how the performance of a PID system can be improved by adding an SMC and using Fuzzy Logic to create a soft switch between them. The model in (5.1) is only used to simulate the system, not to design the controller. The resulting hybrid system can be tuned automatically with a neuro-fuzzy tuner or manually by an expert as was done here without the need to do a complicated mathematical analysis of the system. The ability to tune a system and improve performance without requiring a detailed system model and expensive or difficult to gather parameters makes Fuzzy Logic and other soft computing methods appealing to industry.

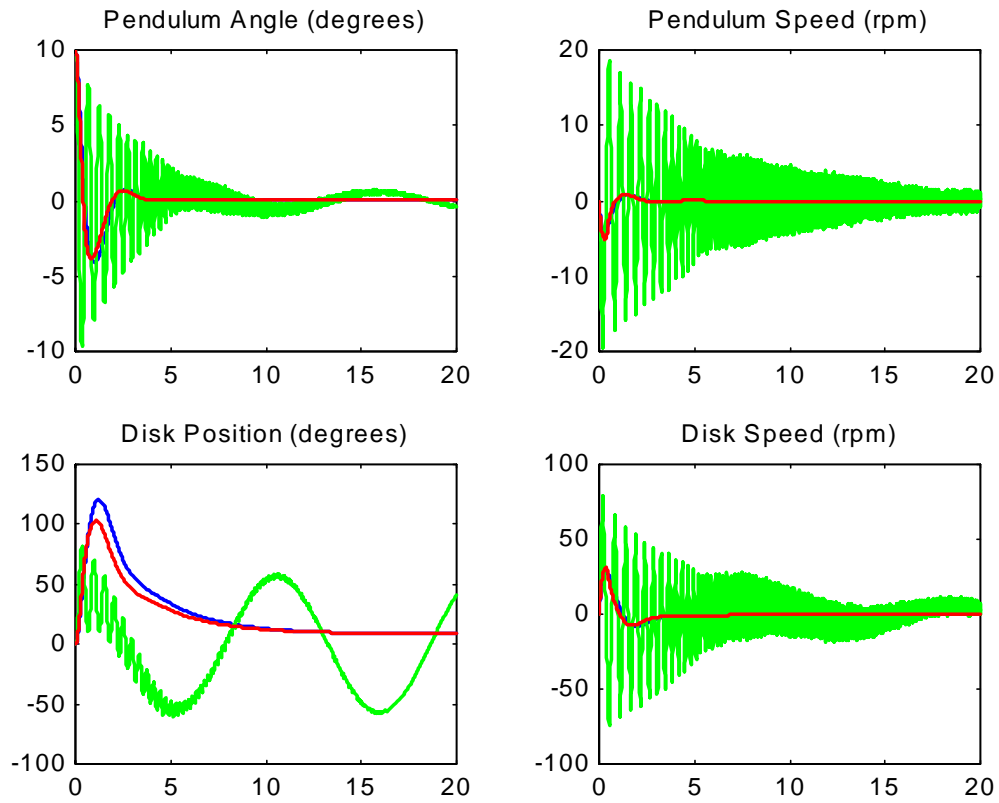
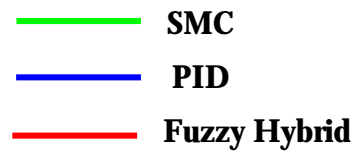


Figure 5.6. The pendulum and disk angle and speed in response to a 10° disturbance.



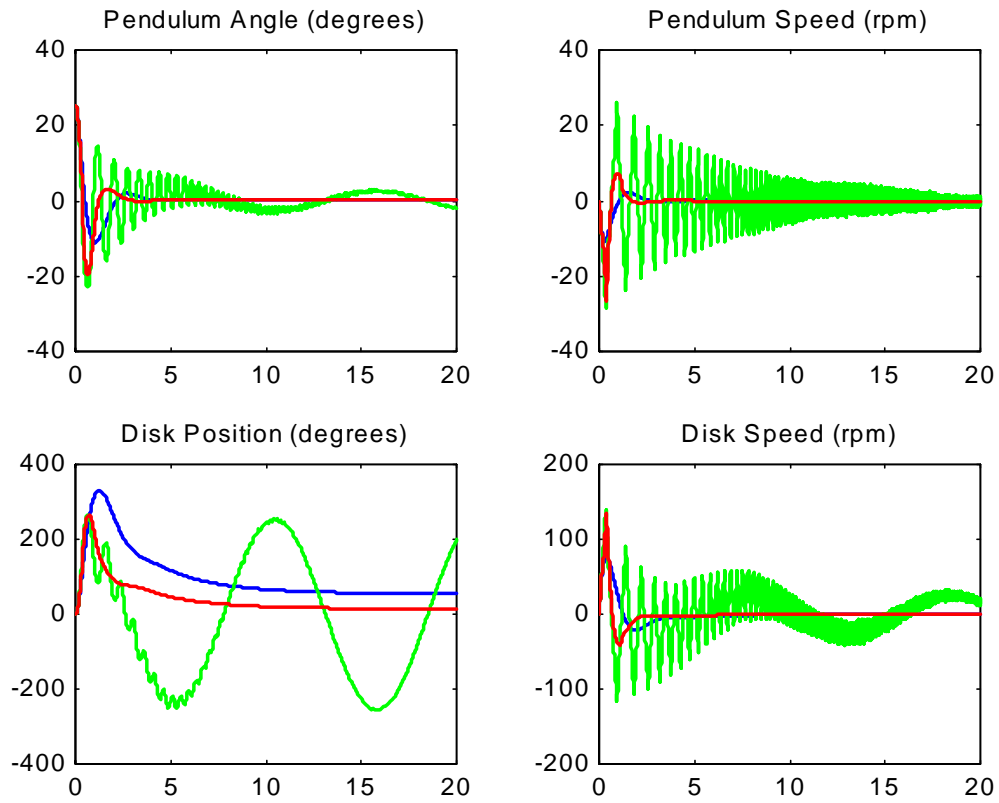
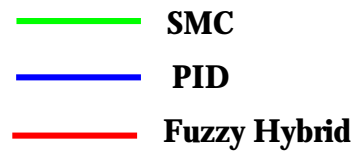


Figure 5.7. The pendulum and disk angle and speed in response to a 25° disturbance.



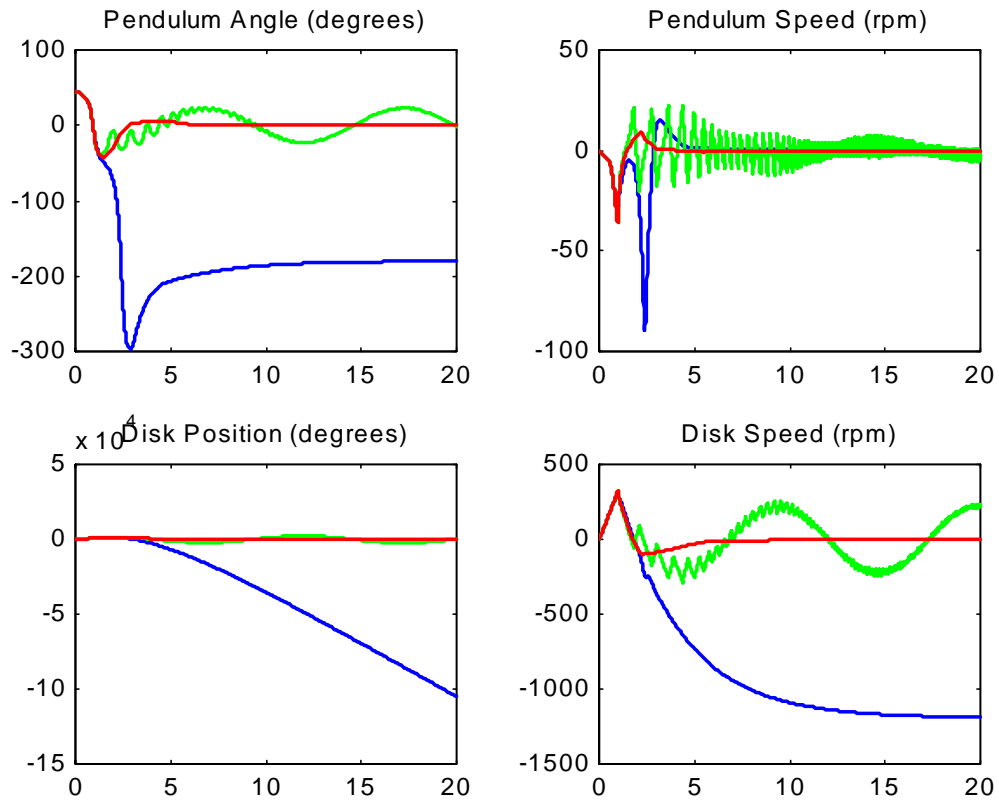


Figure 5.8. The pendulum and disk angle and speed in response to a 45° disturbance.



Chapter 6. A Practical Implementation

Purchasing Considerations

Paralleling almost all other branches of engineering and commerce, motor selection and purchasing can now be done quickly and efficiently over the Internet. In addition to numerous industry-specific web sites there is even a Yahoo! category [35] for electric motors. The motor and motion control industries are huge fragmented market with many players large and small.

The first part of the purchase decision should be a make v buy decision about the entire packaged system. Not all companies can provide a complete packaged and delivered system from the buyer's program interface specifications clean through to installed moving parts. It is assumed here that the target audience of this thesis has some interest in tuning a system and therefore may be interested in at least purchasing parts piecemeal and assembling the components.

It is not practical to do an exhaustive search of the entire industry before purchasing a small quantity of motors and controllers. It is, however, important to select or at least specify each component before purchasing the entire system piecemeal as some features are unexpectedly expensive and many components have varied combinations of features.

For small quantities of motors a local, well known, or well recommended supplier may be the best choice. Long lead times are the norm and are the biggest reason for calling numerous suppliers. When specifications are flexible a surplus or lab equipment catalog may be the best choice for small quantities in a hurry. Unlike the computer industry that is haunted by huge volumes and low margins, local motor and controller manufacturers can often be courted into giving donations or deep discounts to local academic institutions. For researchers on a true shoestring budget it is still more economical to shop surplus or solicit donations than to attempt to manufacturer, refurbish, or rewind a motor.

The next important criterion is the choice of feedback device. The oldest common feedback devices are still the most economical, tachometers and potentiometers for speed and position respectively. Dollar for dollar a tachometer may now cost more than an inexpensive encoder, and the conventional wisdom is that those costs will reverse when the price of encoder interpretation hardware is included. This is an industry where the analog v digital hardware debate is alive and well. In general encoders and resolvers give much better quality feedback than tachometers, but much like motors the bottom line is that price per performance is somewhat consistent across technologies.

A harder trade off than what feedback device to use exists in how to achieve better performance: better feedback v smarter control. In low volumes better hardware that allows for much higher PID gains is cheaper than engineering time and talent to develop a better control scheme. The engineer's role is to create an accurate assessment that makes the issue almost a cost accounting decision.

Though velocity and position compensators and current amplifiers are discussed together they are often sold separately. Most current amplifiers use PWM switching to provide a desired current and also contain a simple PI velocity control loop. A second black box or card must be purchased to do position or precise velocity control. Usually the current amplifier will contain a single axis, a Numerical Machining (NC) term for the ability to turn one motor. Separate control cards are usually multi-axis and can provide an exotic list of features. There are some all-in-one units that provide a current amplifier and velocity and position compensators.

Position and velocity control cards have features lists that show their manufacturers' struggle for bragging rights. Many offer fiber-optic connectors. Several manufacturers offer sophisticated S-curve options. A few manufacturers tout their 64-bit position control systems which, one points out, is enough precision to divide the distance between the Earth and the sun to under 9 nanometers. Such features are more likely designed to prevent users who configure rapidly rotating machinery to work in absolute position mode from seeing an error in their lifetime than for doing wafer stepping on the cosmic scale.

However, the most important features of a control card are that they support the number of axes in use and that they use the same I/O hardware and software as higher and lower parts of the system. Cards are available with every bus, serial, and parallel communications standard in even obscure use. When given the choice, it is worth the investment in a more noise-immune standard that utilizes fiber optics or differential signals such as RS-485. PWM switching and motor brush arc can cause unbearable and difficult to solve noise problems. Beware of the Servo Control Pulse (SCP) standard used in most RC cars, servos, transmitters, and receivers: it is a duty-cycle standard but it is not the same as, or compatible with, PWM. Cards may also be available with “freebie” features that alleviate the need for other system components, such as extra A/D, D/A, or digital I/O pins.

Motion Control Chips

Two chips have the lion’s share of popularity for embedded motion control and custom built applications. They are the National Semiconductor LM628 [36] and the Agilent (formerly HP) HCTL-1100 [37]. The author has also included the Analog Devices ADMC331 [38] as a potential alternative. The price of each chip is shown in Table 6.1.

Table 6.1 Motion Control Chips and Prices

Chip	Price (distributors as of April 2, 2000)
National LM628	\$20.00
Agilent HCTL-1100	\$35.45
Analog ADMC331	\$14.95

Both the National LM628 [36] and the Agilent HCTL-1100 [37] provide position and velocity PID loops. Both assume an encoder is used for feedback. Both can also operate a brushless motor by sequencing the phases on an off correctly. This results in square waves instead of sinusoids on the motor leads which lowers efficiency and performance but does work. They will both do trapezoidal velocity profiling and signal filtering. Both can be configured to provide a PWM or analog output current request that must be connected to a current amplifier. For many motion control applications either chip will perform well, and the data sheets of each should be checked for the minor feature differences that may make one a better choice for a given task.

For those who can tolerate configuring a chip using a DSP assembly language or can afford the proprietary C compiler, the Analog ADMC331 [38] is an excellent recent entry in motion control chips. This chip is actually a 26 MIPS ADSP-2100 with lots of on-board motor controller features. This chip has the features needed for high-performance DC Brushless motors that the others lack: a three-phase PWM generator to drive three sinusoidal motor lead signals and internal blocks designed to do the Park-Clarke and inverse Park-Clarke transform. It also has a timer, three additional PWM channels, 7 analog inputs, and 24 digital I/O pins. Combined with the serial ports, memory, and other usual features of a DSP turned microprocessor and monolithic controller this chip can go beyond motor control and solve all the embedded computing needs of simple robotic vehicles.

A good DC Brush motor H-bridge current amplifier on a chip is the National LMD18200 [39]. It can handle 3A continuous, 6A peak, and 55 volts. Its use in Virginia Tech's Mechatronics class [40] creates available local experience. Beyond 3A or 55V an H-bridge should be purchased or constructed from discrete components. The best document on how to do this comes from Blanchard [41], a consultant whose explanation is only available on-line. The user best heed the warning about gates switching off before switching states. This mistake will result in both switches in one leg of an H-bridge turning on at once and possibly "latching up" and staying on, creating a short across the DC supply. On very large H-bridge's utilizing IGBT's this has resulted in fatal explosions.

Other Considerations

Another practical consideration in the design of vehicle power systems is the need for separate power busses for motor drive and control. Isolated systems of some type are required to keep motor noise out of control electronics. Separate batteries are recommended to keep voltage drops from motor stall current surges from resetting equipment and keep regenerative energy from a motor from damaging control electronics. Figure 6.2 shows two scope captures where electric vehicle motors with a combined 180A stall surge pulled two lead-acid automotive batteries in series from 24V down to 15V for times as short as 110ms. 180A was well within the Cold-Cranking Amps rating of the batteries and the voltage drop did not appear at all on the averaging digital voltmeters originally used to troubleshoot the problem. This short voltage dip was enough to intermittently reboot every computer on the vehicle.

The problem is caused by real batteries and motor leads that have resistance, inductance, and capacitance. The charge in a battery is maintained by a chemical reaction that can only change rates so fast. From an electrical engineering point of view they behave with inductance and capacitance even though internal resistance is usually used as the most important internal battery characteristic.

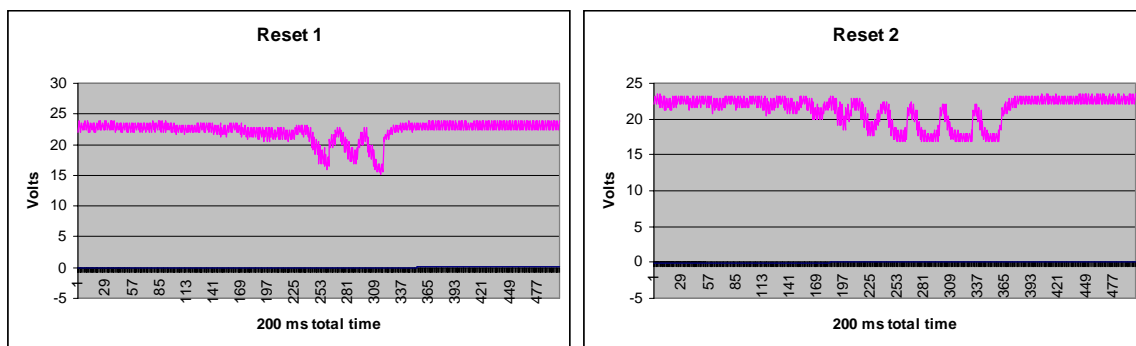


Figure 6.1. Voltage captures during two quick motor stall current surges.

No mention of battery chemistry is complete without a personal weigh-in on the question of whether lead-acid batteries left on a cold concrete floor go dead quicker than other batteries. The claim of most battery manufacturers is that this used to be true but no

longer is in today's world of sealed maintenance-free batteries where hydrogen gas cannot escape -- the usual explanation. A literature search found a lack exhaustive studies on the part of battery manufacturers to back up these claims of unaffected lifetime. The question is whether batteries stay warm like the air or cold like the floor. The author does not store his batteries on cold concrete floors.

Finally, the biggest practical consideration in motor system construction is proper time management. Table 6.1 shows how motor selection and control ranked 9th and 5th respectively on the author's list of time consuming activities in the construction of autonomous electric vehicles. It is a wise idea to schedule any such research project with twice the time and cash required by the best available estimate. Every projects should be designed with at least a safety factor of two.

Table 6.2. Top 10 Time Consuming Tasks in the Design of Autonomous Electric Vehicles

Time Consuming Activities in the Construction of Autonomous Electric Vehicles (most time first)
1. Making computer vision systems robust enough for outdoor use
2. Team management
3. General debug and troubleshooting
4. Documenting work and explaining work to peers
5. Control system design and implementation
6. Navigation Algorithms design, implementation, and testing
7. Power distribution and power electronics design and implementation
8. Physical construction
9. Component selection and system level design
10. Sponsor solicitation and team member recruitment

Chapter 7. A Conclusion with an Example

Conclusion

Part I of this thesis is concerned with motor selection and control design with a concentration on electric vehicles. Two problems of interest to the controls engineer are addressed. The problem of the double integrator and its resulting position overshoot is addressed and the industry's solution of S-curve profiling is explained. The problem of transient and disturbance rejection performance is addressed by high gain observers when a detailed system model is available and by fuzzy hybrid control when a detailed system model is not available. Each of these explanations has a practical example built in.

A larger theme of this thesis is the need to weight motor and control design requirements against the concerns of the entire electric vehicle project as a whole. Comments about this have been spread throughout Part I and no single example is given. Part I concludes with a case study of ZAPWORLD.COM [42]. There are many exciting and innovative examples of electric vehicles, but ZAPWORLD was chosen because their commercial success in "taking it out of the lab" proves that they see the big picture in electric vehicle design.

ZAPWORLD.COM

ZAP is a pioneer in the design of electric motor assisted bicycles. These are hybrid vehicles with two power plants: an electric motor system with battery pack and a human. Their product line has expanded to include scooters, go-carts, motorcycles and some novel vehicles of their own. Their target market is bicycle police and other people looking to move or commute local distances for business or pleasure.

The ZAP Powerbike® and Electricruizer® are the standard power-assist models available with three varieties of the ZAP Power System. All three varieties use a DC Brush motor. The ZAP Electricruizer is shown in Figure 7.1 along with the Lectra Motorbike. The Motorbike is designed to be a worldclass electric vehicle using a brushless motor technology that achieves higher performance but requires a more complex controller.



Figure 7.1. The ZAP Electricruizer (left) and Lectra Motorbike (right).

These products are a good example of when to choose brush or brushless technology. The Electricruizer uses brush technology and has the other features and support that keep people paying \$699.99 for the basic model. There are competitors that offer DC Brushless based electric bikes for just under \$1000 for the standard model, but the performance specifications are comparable so the customer is buying technology, not benefits. As motor size increases to the size of the Lectra Motorbike brushless technology is probably essential to the Lectra's performance. The motorbike has a top speed of 45 MPH and a range of 35 miles.

The control system on each vehicle probably has the most complexity at the power electronics level. Like their full size Hybrid Electric Vehicle counterparts [1] these

vehicles utilize regenerative braking. When the vehicle is stopping the kinetic energy of vehicle motion is returned to the chemical potential energy of a more fully charged battery. Demanding consumers and the need to reduce customer support costs have probably resulted in a well developed, debugged, and documented system. Manufacturing and system construction have been standardized to the point where an entire bicycle can be purchased assembled or the power system can be purchased separately as a kit.

ZAP products are a good example of what it takes to turn a working research lab or academic prototype into a commercial product. They have avoided a premature entry into the full size electric passenger vehicle market that is only now becoming mature enough to flourish. They have left the most complex control tasks usually assigned to electric vehicles, steering and navigation, to the system that still does it best: the human operator.

Part II. Automated Navigation

Chapter 8. Introduction to Navigation Systems

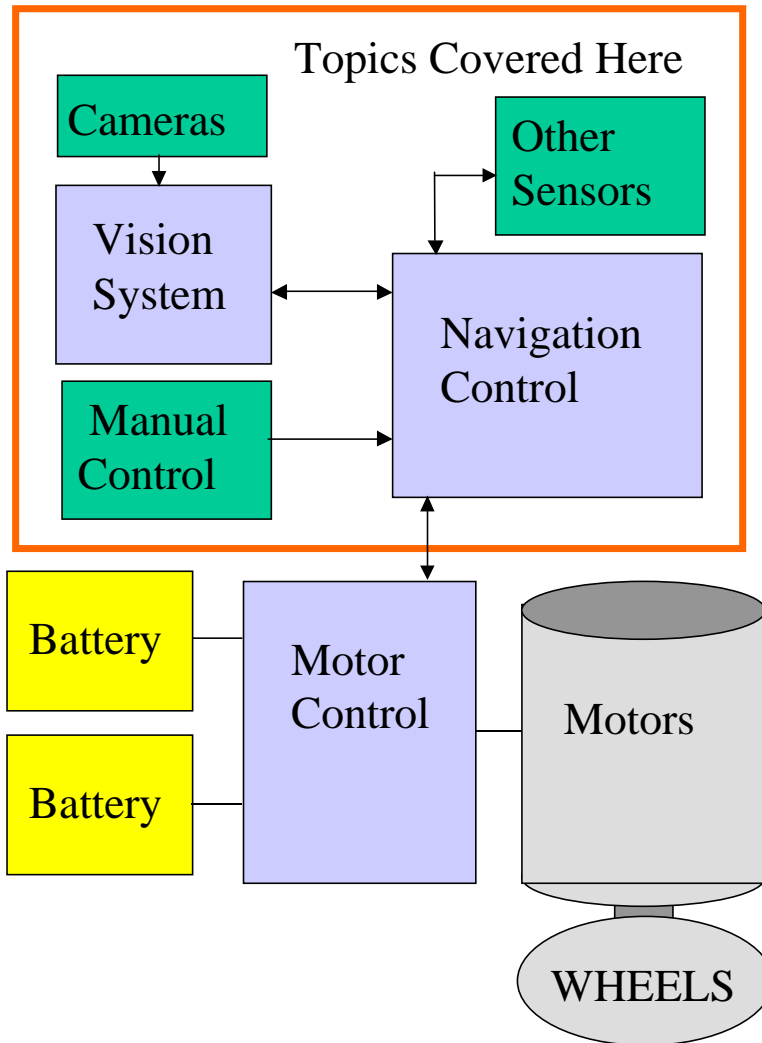


Figure 8.1. A typical autonomous vehicle system showing the parts discussed here.

When many people think about electric vehicles they think about autonomous electric vehicles. Autonomous vehicle, electric vehicle, and hybrid electric vehicle technologies are likely to converge as these technologies mature. The motor control

aspect of designing electric vehicles was the subject of Part I. The object of Part II is taking the human out of the loop and getting vehicles to drive themselves.

All robots need some sort of sensors to measure the world around them. Other than cameras the staples of robotics sensors are ultrasonics, tactile sensors, and infrared sensors. Ultrasonic sensors are used as range finders, emitting a ping of high frequency sound and measuring the time-of-flight until the echo. Tactile sensors are bumpers with switches that only provide information about when a robot has hit something. Infrared sensors can be used as rangefinders or as motion detectors. All these sensors are not only economical but relatively easy to process -- they provide only a single point of information and only so much can be done with that single point. Successful designs using these sensors quickly move to designs using arrays of sensors or multiple measurements over time.

Cameras are a huge step up from these other sensors because the two-dimensional array of information they provide is a flood of data compared to the single drops of information provided by other devices. Quality cameras, interfaces, video capture hardware, and computers are required to quickly and effectively process entire images. In [43] Mentz provides a good example of image processing hardware selection. In a general sense an image can be a 2D array from sonar, radar, camera, or other data and the techniques that apply to each are captivatingly similar.

Three recent or emergent technological advances are underutilized in the development of new computer vision hardware. The first two address the problem of dynamic range when using cameras outdoors. Outdoor images often have single points where direct sunlight is reflected into the camera. When this happens the Charged Coupler Devices used to detect light in conventional CCD cameras will completely fill with charge and overflow, or bloom, into adjacent cells. CCD cameras also have very poor rejection of infrared light. This causes hot objects, usually the dark objects that are not reflecting sunlight, to emit enough infrared radiation to alter a pixel's correct intensity. The result is that nothing in the image looks good and outdoor images are almost always grainy and washed out compared to images captured in a lab.

The technological advance that is doing the most to solve the problem of poor dynamic range is the CMOS camera. These cameras use a photodiode or photogate and active transistor amplifier at every pixel site to create a camera chip out of conventional fabrication processes and virtually eliminate the problems associated with CCD's. A few major semiconductor manufacturers are making CMOS cameras, but the industry leader is Photobit [44]. They have a CMOS camera right for every need, robotics or otherwise, and are poised to become the ubiquitous standard in camera chips.

The next new technology is a new thermoelectric material by Chung et. al. [45] that exploits the thermoelectric cooling effect. When current flows through a PN junction both electrons and holes move away from the interface and take heat with them. The new material exploits this phenomenon better to produce solid state refrigerators on a card that can be driven to 50°C below ambient. This is useful because dark currents, falsely detected light caused by the electron drift in all materials over absolute zero, plague CCD chips. CCD chips are also heated by their own high power consumption. A thermoelectric heat sink could keep these chips cool enough to significantly improve performance. Thermoelectric cooling will also benefit any infrared camera technology because this band is so sensitive to heat.

Finally, recent advances in embedded computing performance offer the opportunity for significant increases in real-time image filtering and color space transformation. The TI C64x DSP [46] offers 8800 MIPS of performance, or over 954 instructions per pixel for a 30 frame per second VGA signal. This has the potential to move many major image processing tasks into real-time hardware and make them invisible to higher level software. A discussion of image processing techniques will generate an appreciation for the need for brawny vision processing computer power.

Chapter 9. Image Processing Techniques

Image processing techniques are as varied as their possible application but the general steps of filter, segment, and measure are common. The steps required in a typical electric vehicle application will be reviewed here as an example. The vehicle may be on a paved or unpaved roadway but is attempting to navigate along a path that usually contains a solid or striped line on one or both sides. There are arbitrary obstacles along the path. The vehicle is equipped with a single color forward-looking camera that must detect the line segments on either side of the path and any obstacles.

The first step is to acquire an image from the camera and measure the dynamic range and contrast of each color channel. This information is used to adjust the gain and exposure time of the next frame. The camera parameters that need to be automatically controllable, especially focus and zoom, greatly affect the cost of the entire system.

The next step is to stretch the image across the full dynamic range of the colorspace to create as much distinction as possible for future operations. In the process an attempt may also be made to correct for inconsistent shading, speculars, noise, and blooming. After this the image will be converted from the camera's native colorspace, usually Red-Green-Blue, to the Hue-Saturation-Intensity colorspace. The hue describes a color pixel's angle from red around the color wheel, and is mostly invariant with changes in lighting conditions. The hue is the only component of the new color space that will be kept for this example. Saturation describes how vivid a color is and is difficult to use in image processing; it is the first piece of information lost in most image compression schemes and the first piece of information completely corrupted by sunlight and infrared. The best use of saturation may be in textured surfaces where the saturation changes with the viewing angle and shape-from-saturation transforms may be possible. Intensity is the black-and-white channel of the image and is all that is available on monochrome cameras. It may be wise to repeat an image processing algorithm with just the hue and just the intensity and compare the results.

After transforming the image and keeping the hue channel, a series of linear convolutions are usually applied to clean up noise in the image. The main complication

arises when there is a need to apply non-linear filters to an image. For example, the Adaptive Wiener 2 filter [47] is a non-linear blurring filter that blurs areas with edges less than areas with low variance in an attempt to remove noise without blurring the edges of objects. Color space transformation are generally non-linear and the order of all the non-linear operations can have a tremendous effect on the resulting image.

A threshold or a series of morphological operators may be applied to further remove spurious features from the image. The image is then segmented into objects of interest through either connected component labeling or a clustering algorithm. A popular variation of these classic segmenting methods is the region-of-interest or ROI. When using ROI's a finite number of windows from the original image are kept and the entire image processing sequence is only performed on these windows. After the process is complete the center of the object in each ROI is found and the coordinates of the ROI are adjust in an attempt to get the object closer to the center of the ROI on the next image. If there is no object in the ROI sophisticated searching algorithms may be employed to move the ROI in search of an object. This technique was originally used because image processing hardware did not have the power to perform the desired operations on the entire image fast enough. It is still in use because it turns out to be an excellent method of tracking objects from one frame to the next.

The final task is to extract some useful information about the components that have been separated from the image. This system assumes all components are either long skinny line segments or blob-like obstacles. Sophisticated pattern matching techniques including Bayesian classifiers and neural networks may be used to compare a segment to a library of known objects. The optics of the camera and geometry of its location on the vehicle will be used to carry out a ground plane transform, a transform that determines the coordinates of a pixel in an image by assuming that the object lies on a level ground plane. The vision system then passes along information useful to the navigation system: a list of line segments and obstacles and their coordinates in the ground plane.

Chapter 10. A Novel Navigation Technique

Every autonomous vehicle navigation strategy will undergo many revisions and incremental improvements before it works reliably. The result of evolving a navigation strategy for the example of the previous chapter with line segment and obstacle data is presented here. All obstacles will be represented by the potential field shown in Figure 10.1. This scheme has been named Mexican Hat Navigation because of Figure 10.1's shape.

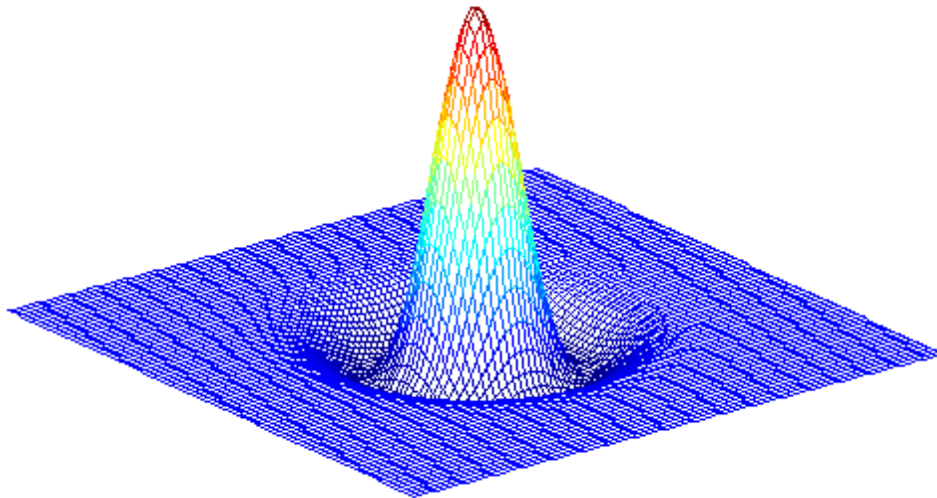


Figure 10.1. The Mexican Hat. A potential field that will be used to represent an obstacle.

This shape is known as the 2D Laplacian of Gaussian and is a statistical distribution that has been commonly used in edge detection ever since vision pioneer David Marr [48] suggested that it is the edge detection convolution carried out by the human retina. The Laplacian of Gaussian is a well established function that will be used in a novel way. In the human eye a bright dotted line activates the retina with the activity map of Figure 10.1. The troughs on each side of the dotted line combine to form two valleys of dark outlining a mountain range of light peaks which are then perceived as a single line. This function's penchant for well behaved superpositioning makes it an ideal basis for an entire navigation strategy. The trough around the peak has been placed at a

distance that corresponds to a safe distance for a vehicle to pass an obstacle based on the width of the vehicle. When multiple obstacles are present their fields overlap to create troughs in places through which the vehicle can safely navigate.

The world chosen for this example contains only two objects, obstacles and line segments. All line segments will be represented by the potential field shown in Figure 10.2. This Figure is known as the Shark Fin. It has a Laplacian of Gaussian distribution perpendicular to the line segment and a Gaussian distribution parallel to the line segment.

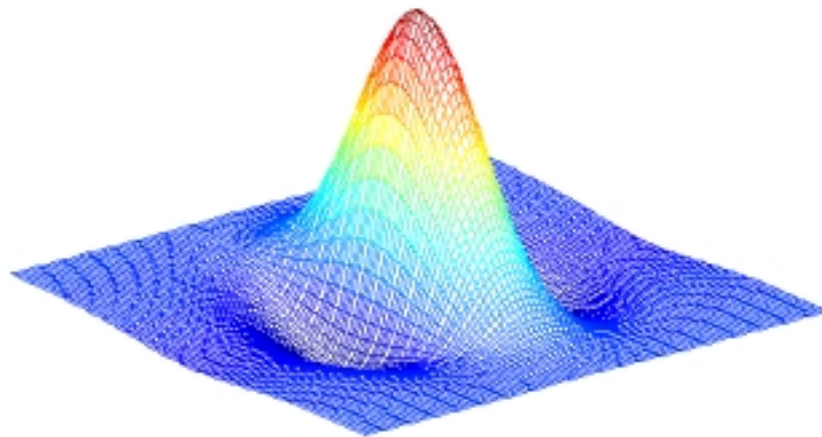


Figure 10.2. The Shark Fin. A potential field that will be used to represent a line segment.

All the line segments and obstacles detected by the vision system and any obstacles detected by other systems are mapped together onto the empty grid in Figure 10.3. At each obstacle location a Mexican Hat mask is added to the grid. For each line segment a Shark Fin must be translated and rotated before it is added to the grid. The result of all these superpositioned masks is the potential field of Figure 10.4.

The vehicle, which starts at the bottom center of the map, navigates by driving forward down the potential alley of least resistance. This path is shown on the field in Figure 10.5 and again on the original map in Figure 10.6.

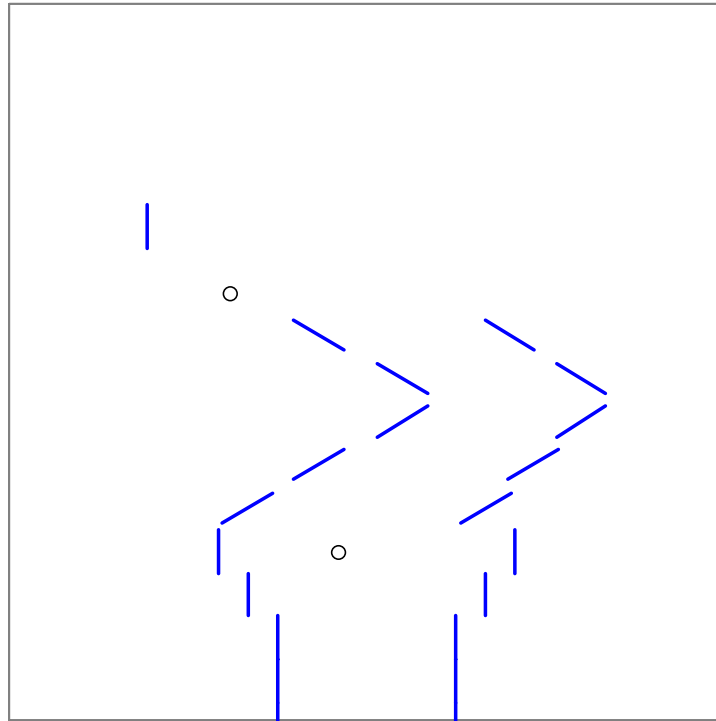


Figure 10.3. A map of obstacles and line segments.

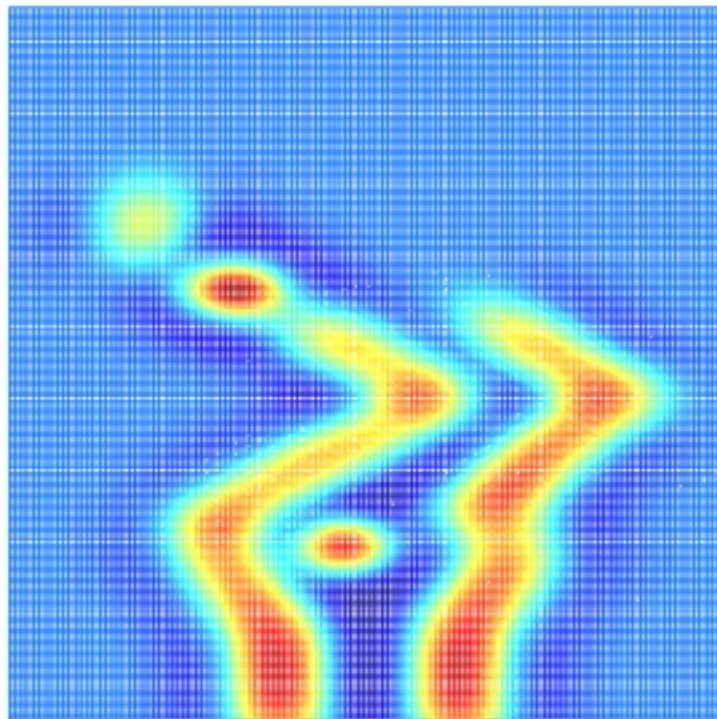


Figure 10.4. The potential field created by Mexican Hat Navigation.

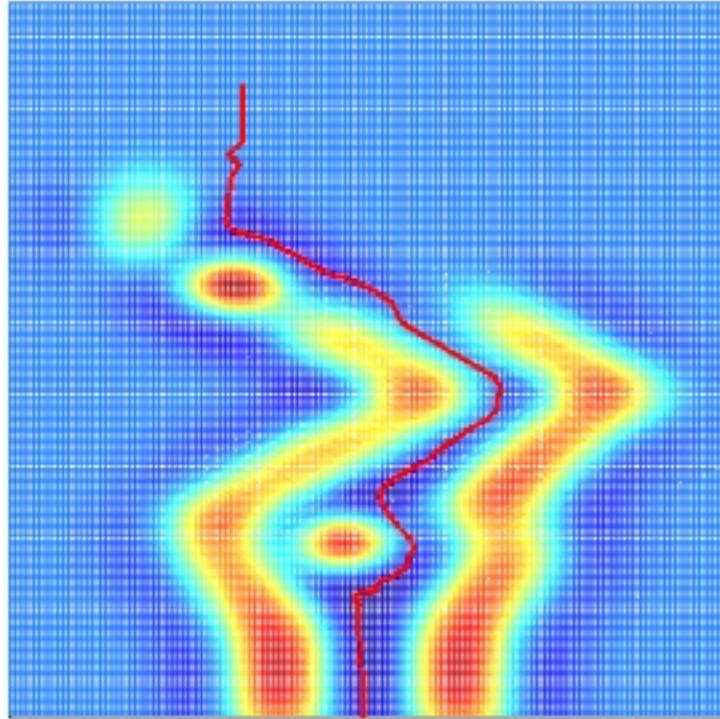


Figure 10.5. The path of least resistance through the potential field.

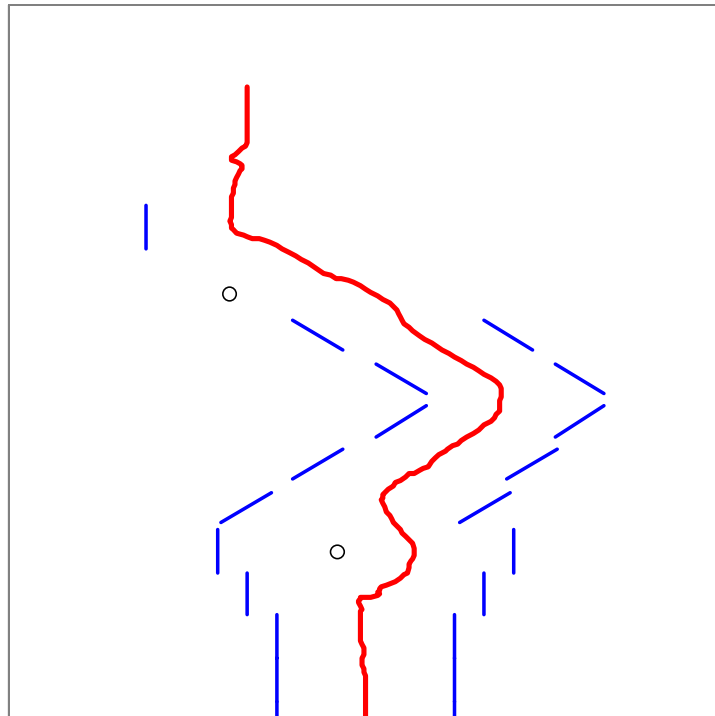


Figure 10.6. The resulting path through the course.

There are several simple variations to Mexican Hat Navigation. The first ones involves smoothing the path. This can be done by calculating the same path and then going back and applying an averaging filter to each point. Significant smoothing can be added by eliminating excursions where the path goes sideways for a single step and then returns to its original path. A more sophisticated method of smoothing the path is to assume a sphere has to roll down the path of least resistance instead of a point and then adjusting the radius and momentum of the sphere to gain the desired smoothness.

Another variation is to change or increase the number of objects allowed in the universe beyond line segments and obstacles. New potential fields must then be contrived to describe the new objects. Contrived fields with more general geometric shapes are known as artificial potential fields and are also used for obstacle avoidance and approach by Khosla and Volpe [49]. A wide variety of alternate navigation strategies is available in the literature.

Kim and Khosla in [50] and Akishita et. al. in [51] use an artificial potential function to get around an obstacle to a goal sink. In [52] Megherbi and Wolovich use potential fields and complex conformal mapping to do obstacle avoidance in 3D. Nam et. al. [53] use artificial potential fields to do collision planning and avoidance when multiple objects are moving. In [54] Joarder and Raviv exploit another characteristic of the human retina to perform obstacle collision. They mimic the looming reflex where objects close to the eye that change texture suddenly cause one to flinch in the other direction.

DeMuth and Springsteen [55] use a neural network with a map of the world as the input and an Autonomous Underwater Vehicle's (AUV's) rudder and speed command as the output. The network is trained manually to give closer obstacles more urgent weights. The resulting system looks like a potential field map because of the geometrically pleasing choice of weights.

In [56] Borenstein and Koren use potential fields along with histogram information and a probability map to perform obstacle detection. In [57] Trahanias and Komninos use uncertainty fields to perform obstacle avoidance. Both of these techniques were popular before embedded computers and vision systems had replaced ultrasonics

and microcontrollers. These methods still have merit but have been replaced by other methods that are now computationally practical.

A large number of navigation and path planning papers deal with determining a contour or path to a goal after all possible paths in a network have been discovered. Sundar and Shiller [58] do this using the Hamilton-Jacobi-Bellman equation, Yahja et. al. [59] use quadtree decompositions, and Chohra et. al. [60] use a Neuro-Fuzzy expert system. Fraichard and Mermond [61] show a path planning method that reduces collisions by accounting for some of the kinematics of car-like robots.

Each of these other navigation methods require a goal state to be known so that an optimal path can be found to that goal state. The Mexican Hat algorithm is unique in its ability to continue to traverse a course without an inherent particular goal. In a universe with only one obstacle and no line segments, the potential field would look like the Mexican Hat of Figure 10.1. Other potential field models would deflect an autonomous vehicle off into nowhere, but here the vehicle will circle in the rim of the hat until a line segment or obstacle became visible and caused the vehicle to head off in another direction. The ability of the Mexican Hat Technique to continue to operate without a goal makes it novel and appealing to the artificial intelligence community; this behavior is more human.

Obstacle avoidance is an important but incomplete part of what navigation systems do. The most common shortfall of vision and navigation systems is that they calculate a whole new trajectory every cycle instead of integrating and storing data over time. Adding mapping and memory can multiplicatively increase the computational and storage requirements of an embedded system. Navigation soon enters the larger issue of mapping and map tracking, and Mayhew [62] covers these topics well.

Chapter 11. Conclusion

Two technologies are converging to create the cars of tomorrow. The first is the industrial and commercial robotics, motor control, battery and power electronics technology required to bring full size electric and hybrid electric passenger cars out of the lab, through the showroom floor and onto America's driveways and highways. Part I discussed the state of the motor control industry and addressed some of the difficult or common problems. Interesting examples show that there is already a market for new electric vehicles and products.

The other maturing technology is smart car technology. Part II surveys the vision hardware and navigation software necessary for an autonomous robotic vehicle. A novel obstacle avoidance strategy has been added to the body of obstacle avoidance work available. As America's highways grow more crowded and room for expansion grows more scarce, autonomous vehicles will viewed as less of a convenience and curiosity and more of a safety device that saves lives. America will have a different definition of the word *automobile*.

The content of this work has been aimed at the autonomous electric vehicle designer or potential future robotics designer. Like all processes, design and learning require good feedback. Motor control performance is ultimately limited by the signal to noise ratio of the feedback device, computer vision systems are ultimately limited by the quality of the image the camera can acquire, and human being are incrementally bettered with each new learning experience.

Bibliography

References for Part I

- [1] W. Luttrell, B. King, S. Postle, R. Fahrenkrog, M. Ogburn, and D. J. Nelson, "Integration of Fuel Cell Technology into the Virginia Tech 1999 Hybrid Electric FutureCar," Animul H2 -- HEVT's Fuel Cell Hybrid, http://fbox.vt.edu:10021/org/hybridcar/documents/fcc99/fcc99final_report.pdf (12 March 2000).
- [2] Oriental Motor USA Corporation, *Oriental Motor General Catalog 1997*. Oriental Motor USA Corp. Torrance, CA, 1997.
- [3] D. La Ree, *EE 3354 Power Lab SPRING 1999*, A-1 Copies, Blacksburg, VA, 1994.
- [4] J. Pyrhönen, J. Haataja, and K. Luostarinen, "Specifications of Requirements for High Efficiency Induction Motors 'Hi-Motors,'" *I motivan moottorikilpailu*. Lappeenranta University of Technology and Heikki Härkönen. <http://info.lut.fi/ente/sahko/Hi-motors/hiapp1.htm> (12 March 2000).
- [5] Canon USA, "Encoders: Super High Resolution Encoder X-1M Introduction," <http://www.usa.canon.com/indtech/encoders/x1mnonrot.html> (12 March 2000).
- [6] P. Krause, O. Wasynczuk, and S. Sudhoff, *Analysis of Electric Machinery*, IEEE Press, NY, 1995.
- [7] K. Ramu, *Electronic Control of Machines. ECpE 4324*, A-1 Copies, Blacksburg, VA 1998. From the forthcoming book *Electronic Control of Machines*, Prentice-Hall, USA.
- [8] J. Mentz, "A Motor Torque-Speed curve plotting MATLAB GUI," http://www.ee.vt.edu/jmentz/TS_GUI.ZIP (14 March 2000).
- [9] Kollmorgen Inc. *MOTIONEERING*. <http://kmtg.kollmorgen.com/Products/Software/motioneering.html> (13 March 2000)
- [10] Galil Motion Control, *Galil Motion Component Selector. [v4.05]*, <http://209.220.32.26/cgi-bin/checkreg.pl?mcs>, from <http://www.galilmc.com/support/download.html> select MCS (13 March 2000).

- [11] Analog Devices, Inc., *Product Index: Sensors & Signal Con, Accelerometers*, http://products.analog.com/products_html/list_gen_121_2_1.html (13 March 2000).
- [12] SICK Inc., *PLS Proximity Laser Scanner*, <http://www.sickoptic.com/plsscan.htm> (13 March 2000).
- [13] Kollmorgen Motion Technologies Group, *BDS-5 USER'S MANUAL M93102 - ISSUE 3*, Industrial Drives, Radford, VA 1995.
- [14] Delta Tau Data Systems, Inc., *PMAC Executive For Windows PEWIN Help*, <ftp://www:1234@ftp.deltatau.com/PMACManual/helpman.exe> (14 March 2000).
- [15] N. S. Nise, *Control Systems Engineering 2nd Ed.*, Addison-Wesley, NY, 1995.
- [16] J. D. Cutnell and K. W. Johnson, *Physics 3rd Ed.*, John Wiley & Sons, New York, pp. 41, 1995.
- [17] J. S. Bay, *Fundamental of Linear State Space Systems*, WCB McGraw-Hill, New York, 1999.
- [18] D. Dozar and N. Hemati, "A Torque Improving Stationary Frame Controller for Permanent Magnet Synchronous Machines," *Conf. Record IAS Annual Meeting. v 1 1994*, IEEE, Piscataway, New Jersey, pp. 416-423, 1994.
- [19] W. Leonard, *Control of Electric Drives*, Verlag, Berlin, 1985.
- [20] S. Chung et. al. "A Robust Speed Control of Brushless Direct Drive Motor Using Integral Variable Structure Control with Sliding Mode Observer," *Conf. Record IAS Annual Meeting. v 1 1994*, IEEE, Piscataway, NJ, pp. 393-400, 1994.
- [21] H. Khalil, *Nonlinear Systems, 2nd Ed.*, Prentice Hall, New Jersey, 1996.
- [22] J. Lee et al. "Design of Continuous Sliding Mode Controller for BLDD Motor with Prescribed Tracking Performance," *Conf. Rec IEEE PESC '92*, pp. 770-775, 1992.
- [23] C. Ünsal and P. Kachroo, "Sliding Mode Measurement Feedback Control for Antilock Braking Systems," *IEEE Trans. On Control Systems Technology v. 7 no. 2*, pp. 271-281, 1999.

- [24] D. Young et. al., "A Control Engineer's Guide to Sliding Mode Control," *IEEE Trans. On Control Systems Technology* v. 7 no. 3, pp. 328-342, 1999.
- [25] J. Slotine and W. Li, *Applied Nonlinear Control*, Prentice Hall, New Jersey, 1991.
- [26] S. Ovaska, O. Vainio, and T. I. Laakso, "Design of predictive IIR filters via feedforward extension of FIR forward predictors," *IEEE Trans. Instr. and Meas.* vol. 46, Oct 1997, pp. 1196-1201, 1997.
- [27] S. Väliiviita and O. Vainio, "Delayless differentiation algorithm and its efficient application for motion control applications," *Proc. IEEE Instrum. Meas. Tech. Conf.*, St. Paul, Minnesota, May 1998, pp. 881-886, 1998.
- [28] S. J. Ovaska and S. Väliiviita, "Delayless recursive differentiator with efficient noise attenuation for motion control applications," *Proc. of the 1998 24th Annual Conf. of the IEEE Industrial Electronics Society, IECON, Part 3 (of 4), v 3 1998*, pp. 1481-1486, 1998.
- [29] Kazuo Hiroi, *US4755924: Process controller having an adjustment system with two degrees of freedom*, Kabushiki Kaisha Toshiba, Kawasaki, Japan, July 5, 1988.
- [30] Kazuo Hiroi, *US5105138: Two degree of freedom controller*, Kabushiki Kaisha Toshiba, Kawasaki, Japan, April 14, 1992.
- [31] Kazuo Hiroi, *US5195028: Two degree of freedom controller*, Kabushiki Kaisha Toshiba, Kawasaki, Japan, March 16, 1993.
- [32] J. Lewis, J. Mentz, and H. Meshref, "Fuzzy Hybrid Control of an Inverted Pendulum on a Horizontal Disk," Final Report for ECpE 5724: Neural and Fuzzy Systems, Fall Semester 1999 with Dr. Hugh F. VanLandingham, November 30, 1999.
- [33] J. Lewis and J. Mentz, "Hybrid Control of a Rotary Inverted Pendulum," *Virginia Tech Signals and Systems Seminar*, February 25, 2000.
- [34] J. Jang, C. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing. A Computational Approach to Learning and Machine Intelligence*, Prentice Hall, New Jersey, 1997.
- [35] Yahoo! Inc. Electric Motors, Yahoo category "Home > Business and Economy > Companies > Electronics > Business to Business > Electric Motors >" (April 1, 2000).

- [36] National Semiconductor, "LM628 Precision Motion Controller," <http://www.national.com/pf/LM/LM628.html> (April 2, 2000).
- [37] Agilent, "General Purpose Motion Control ICs HCTL-1100," <http://www.semiconductor.agilent.com/motion/hctl1100.html> (April 2, 2000).
- [38] Analog Devices, "Single Chip DSP Motor Controller ADMC331," http://www.analog.com/pdf/ADMC331_a.pdf (April 2, 2000).
- [39] National Semiconductor, "LMD18200 3A, 55V H-Bridge," <http://www.national.com/pf/LM/LMD18200.html> (April 2, 2000).
- [40] VT Mechatronics, "VT Mech: VT84 Parts List," <http://mechatronics.me.vt.edu/VT84Construction/partslist.html> (November 1, 1999).
- [41] E. Blanchard, "The Using MOSFETs Page," <http://www.cadvision.com/blanchas/hexfet/> (Nov 28, 1999).
- [42] ZAPWORLD.COM, "ZAP Electric Bikes, ZAPPY Scooters and other Zero Air Pollution transportation!," <http://zapworld.com/>, Site content used with permission, (April 2, 2000).

References for Part II

- [43] J. Mentz, "An Automated Washer Identification System," EE5554 – Theory and Design of Machine Vision Systems with Richard Conners, Ph.D. http://www.ee.vt.edu/jmentz/vision_hw_1_washer_sorter.pdf (October 2, 1998).
- [44] Photobit, <http://www.photobit.com/> (April 2, 2000).
- [45] D. Chung, T. Hogan, P. Brazis, M. Rocci-Lane, C. Kannewurf, M. Bastea, C. Uher, and M. G. Kanatzidis, "CsBi₄Te₆: A High-Performance Thermoelectric Material for Low-Temperature Applications," *Science* 2000 February 11, 287, pp. 1024-1027, 2000.
- [46] Texas Instruments, "TMS320C6000™ Highest Performance DSP Platform," <http://www.ti.com/sc/docs/products/dsp/newcores/c64x.htm> (April 3, 2000).

- [47] The MathWorks, Inc., "Adaptive Filtering," Image Processing Toolbox User's Guide Version 2.1, pp. 7-23, 1998.
- [48] David Marr, Vision, W. H. Freedman and Company, New York, 1982.
- [49] P. Khosla and R. Volpe, "Superquadric Artificial Potentials for Obstacle Avoidance and Approach," 1988 IEEE Int. Conf. on Robotics and Automation, Philadelphia, PA, pp. 1778-1784, 1988.
- [50] J. Kim and P. Khosla, "Real-Time Obstacle Avoidance Using Harmonic Potential Functions" Proc. of the 1991 IEEE Int. Conf. on Robotics and Automation, Sacramento, CA, pp 790-796, 1991.
- [51] S. Akishita, T. Hisanobu, and S. Kawamura, "Fast Path Planning Available for Moving Obstacle Avoidance by Use of Laplace Potential," Proc. of the 1993 IEEE Int. Conf. on Intelligent Robots and Systems, Yokohama, Japan, pp. 637-678, 1993.
- [52] D. Megherbi and W. A. Wolovich, "Real-Time Velocity Feedback Obstacle Avoidance Via Complex Variables and Conformal Mapping," Proc. of the 1992 IEEE Intl. Conf. of Robotics and Automation, Nice, France, pp 206-213, 1992.
- [53] Y. S. Nam, B. H. Lee, and N. K. Ko, "An Analytic Approach to Moving Obstacle Avoidance Using an Artificial Potential Field," Proc. of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems, Part 2 (of 3), Pittsburgh, PA, pp. 482-487, 1995.
- [54] K. Joarder and D. Raviv, "A New Method to Calculate Looming for Autonomous Obstacle Avoidance," Proc. of the 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Seattle, WA, pp. 777-780, 1994.
- [55] G. DeMuth and S. Springsteen, "Obstacle Avoidance Using Neural Networks," Proc. of the Symposium on Autonomous Underwater Vehicle Technology - AUV '90, Washington, DC, pp. 213-215, 1990.
- [56] J. Borenstein and Y. Koren, "Histogram In-Motion Mapping for Mobile Robot Obstacle Avoidance," IEEE Trans. on Robotics and Automation, Vol. 7, No. 4, August 1991.

- [57] P. E. Trahanias and Y. Komninos, "Robot Motion Planning: Multi-Sensory Uncertainty Fields Enhanced with Obstacle Avoidance," Proc. IEEE IROS 1996, pp. 1141-1148, 1996.
- [58] S. Sundar and Z. Shiller, "Optimal Obstacle Avoidance based on the Hamilton-Jacobi-Bellman Equation," Proc. of the 1994 IEEE International Conference on Robotics and Automation, San Diego, CA, pp. 2424-2429, 1994.
- [59] A. Yahja, A. Stentz, S. Singh, and B. L. Brumitt, "Framed-Quadtree Path Planning for Mobile Robots Operating in Sparse Environments," Proc. of the IEEE Intl. Conf. on Robotics and Automation, Leuven, Belgium, pp, 650-655, May 1998.
- [60] A. Chohra, A. Farah, and M. Belloucif, "Neuro-Fuzzy Expert System E_S_CO_V for the Obstacle Avoidance of Intelligent Autonomous Vehicles (IAV)," Proc. of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems, Part 3 (of 3), pp.1706-1713, 1997.
- [61] Th. Fraichard and R. Mermond, "Path Planning with Uncertainty for Car-Like Robots," Proc. of the 1998 IEEE Intl. Conf. on Robotics and Automation, Leuven, Belgium, pp. 27-32, May 1998.
- [62] D. Mayhew, Multi-rate Sensor Fusion for GPS Navigation Using Kalman Filtering, VT EDT Collection, <http://scholar.lib.vt.edu/theses/available/etd-062899-064821/> April 18, 1999.

Vita

James Mentz

James Mentz entered Virginia Polytechnic Institute and State University as an undergraduate in the Engineering program in fall of 1993. James graduated with his Bachelor of Science in Electrical Engineering in May of 1998. James then remained at Virginia Tech and completed his Masters of Science Degree in Electrical Engineering in May of 2000. James may choose to continue his career at Blacksburg and Virginia Tech.