Controlling Motors in the Presence of Friction and Backlash

Tim Wescott, Wescott Design Services

Two of the most prevalent issues in controlling moving mechanisms are friction and backlash. These effects are present in nearly every mechanism that one may wish to control, and they have a serious negative impact on the control system designer's ability to accurately control mechanism motion, yet often they can be eliminated only through heroic efforts on the part of mechanism designers. Thus it is often necessary to deal with these effects in the controller. This article explains friction and backlash effects, why they are a source of problems in motion system design, how a control system designer can predict their effects and how to mitigate those effects.

December 12, 2014

Author's Note:

This paper forms part of the basis material for Chapter 8, *Nonlinear Systems*, in the book *Applied Control Theory for Embedded Systems* by Tim Wescott [Wes06]. If you find this paper informative, you may be interested in the rest of the book.

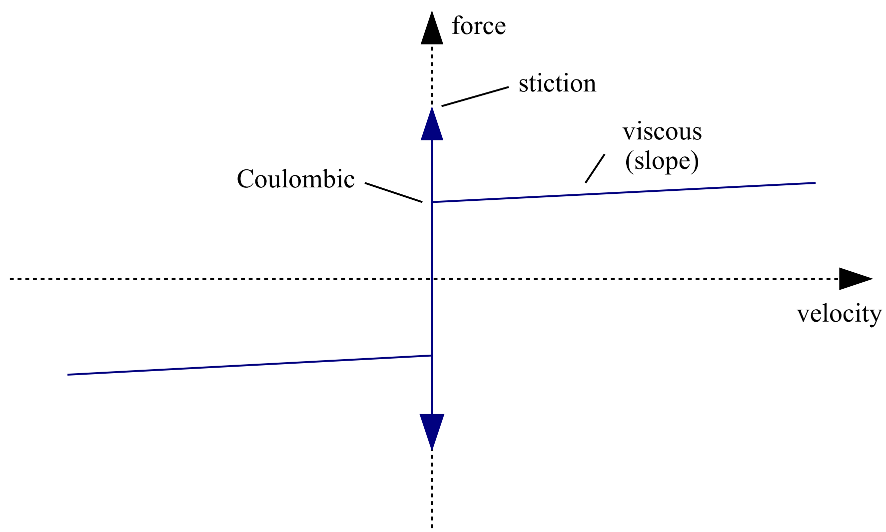

Figure 1: Force vs. Velocity plot for Friction.

Achieving smooth control in mechanisms can be a heartless task. Modeling is difficult, and control strategies that work in "textbook" cases often fail to work in the real world. Two of the factors that often contribute to this difficulty are friction and backlash. These effects are highly nonlinear, difficult to model and analyze even with a fully nonlinear model, yet cannot be ignored.

Fortunately, control systems engineers have developed methods of dealing with these issues. These methods are ad-hoc and often seem old fashioned, but they can often work and work well.

1 Friction and Backlash Behavior

The property that makes both friction and backlash difficult to deal with is the existence of sudden, difficult to quantify discontinuities. This not only makes it difficult to deal with their behavior mathematically, it also means that the performance of common PID and PD controllers will not be close to optimal, and will often not be satisfactory.

Everyone is familiar with friction in everyday life. Friction lets us stand and walk, it lets us hold tools and food, it makes our machines run. Friction also defeats our attempts to build 100% efficient machines, and it can make life difficult for a control systems designer.

Figure 1 on page 2 shows the main characteristics of a mechanical interface with friction. The stiction (or starting friction) is the amount of force required to break the interface loose. The Coulombic (or "dry") friction is that portion of the running friction that is dependent only on the direction of motion but has constant magnitude. Finally most mechanisms with friction also display some viscous drag that is more or less proportional to velocity.

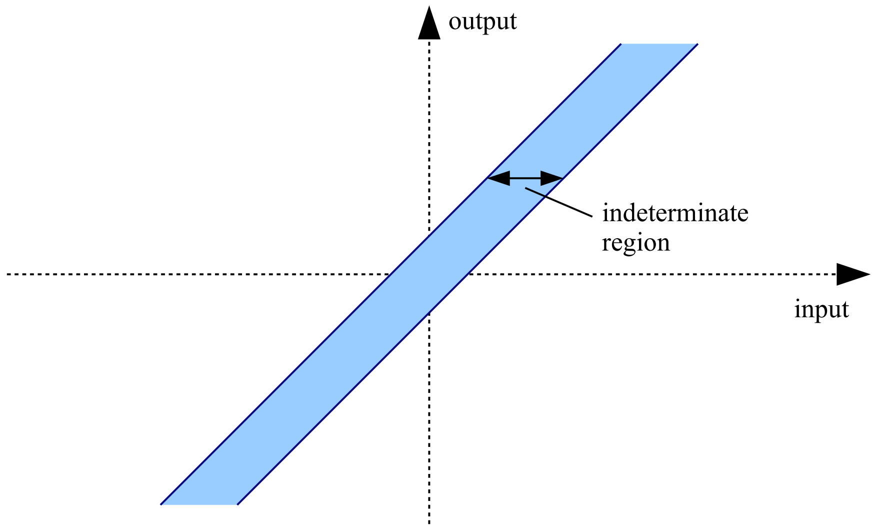

Figure 2: Input vs. Output Displacement for an Element with Backlash.

Looking at Figure 1 there are not one, but two discontinuities around zero velocity. First there's the Coulombic friction, where the force resisting motion is proportional to the sign of the velocity. If that weren't enough, most interfaces with friction also exhibit stiction (or starting friction). Stiction is the effect where, if the interface has remained still for any length of time, the amount of force required to start the relative motion is greater than the amount required to sustain it. The resulting force/velocity relationship not only has discontinuities in it, but it has discontinuities that are, for all effective purposes, large and infinitesimally narrow.

In control systems terms the result of these discontinuities in the force/velocity relationship is to give the relationship an effective gain that is effectively infinite. Worse, the effective gain when the mechanism comes out of stiction is not only large but negative. Neither of these properties are conducive to loop stability.

Backlash is the term that is commonly used to describe any sort of coupling that has slack when it is unloaded. Devices such as gear trains, or mechanical linkages that contain pinned hinges, will exhibit backlash to some extent or another. Such devices require a certain amount of running clearance to work, and this running clearance must be taken up before the output of the device will respond to the input. This all means that for a given input position the position of the output is indeterminate within the limits of the backlash.

The amount of backlash in any given mechanism can often be "bought down" by specifying more precise parts, or by building anti-backlash devices. However, doing so adds system cost, makes the system more sensitive to wear, and often increases problems with friction.

Figure 2 on page 3 shows that backlash presents problems to the would-be plant model that are nearly as severe as those presented by friction. In the case of backlash there is a hidden state – the difference between the input and output positions. This state is only independent of the input position when it is within the indeterminate region — beyond

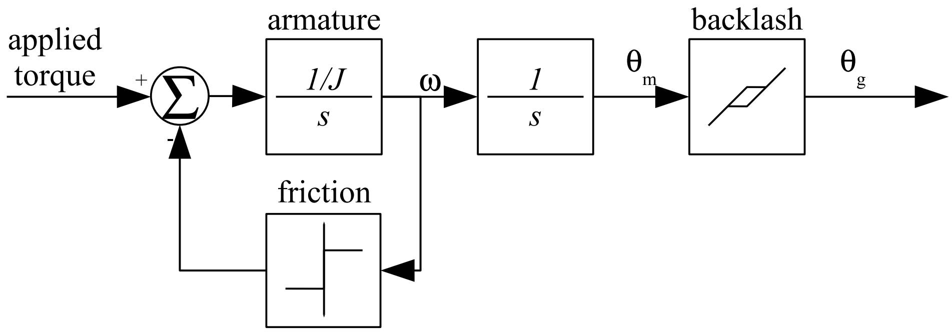

Figure 3: Mechanism Model with Friction and Backlash.

that region the hidden state is at its minimum or maximum value, and the output is pinned to the input. This causes the plant model to have variable structure, with a discontinuous transition between the “free” and “pinned” modes.

Figure 3 on page 4 shows the mechanism model that will be used in this paper. The model is in the context of a motor, however it could easily apply to any mechanism with friction and backlash. The motor armature velocity determines the friction torque available. This friction torque is subtracted from the applied torque. If there is any torque left over this is used to accelerate the motor armature. The armature velocity is integrated into motor shaft position. The motor drives a device (usually a gearbox) that has some backlash; the output of this device is the output of the motor model.

2 Linear Controllers

The preferred way of designing a control system is to either start with a linear plant model, or linearize the model by choosing an operating point and finding the first derivative of the plant output to the “important” plant inputs. Many control systems engineers are so focused on a linearized model that they don’t even realize that they’ve done so — that’s just “the way things are done”. This approach works in cases where the first derivative exists and where it doesn’t vary over too wide a range, and where the plant model doesn’t dramatically change it’s character as a function of it’s state. In the case of friction and backlash, however, the plant’s input-output behavior usually makes this method invalid.

2.1 Continuous Motion

One use case for a controller where friction and backlash *don’t* make the linear approximation invalid is when the system is in continuous motion. If the system to be controlled is going to be running continuously and in such a manner that the backlash is always taken up, then the behaviors shown by Figure 1 on page 2 and Figure 2 on page 3 will be reduced to a fixed friction torque and a fixed positional offset. In this case then the system behavior can be modeled easily by linearization, and simple PID control techniques can be used quite successfully.

Example 1: When Friction Isn't So Bad.

A motor is connected to a small conveyor belt, and is controlled by a PID controller connected to the motor via tachometer feedback as shown in Figure 4 on page 5. The motor and conveyor assembly together have both friction and backlash, and behave as modeled in Figure 3 on page 4. When the assembly is in continuous motion its response to a torque disturbance is a damped pass-band response which can be approximated by

$$\frac{\Omega(s)}{T_d(s)} = \frac{k_d s}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \tag{1}$$

where the damping ratio (ζ) is no less than 0.7, the loop natural frequency (ω_0) is approximately $14^{\text{radians}}/_{\text{sec}}$, and the coupling constant at the motor k_d is no more[1] than $100\text{RPM}/\text{N}\cdot\text{m}/\text{sec}$ ($\text{N}\cdot\text{m}$ = Newton-meter). The breakaway torque is $0.4\text{N}\cdot\text{m}$, the running friction is $0.2\text{N}\cdot\text{m}$, and there are other disturbance torques in the system that can reach $0.4\text{N}\cdot\text{m}$ peak-peak.

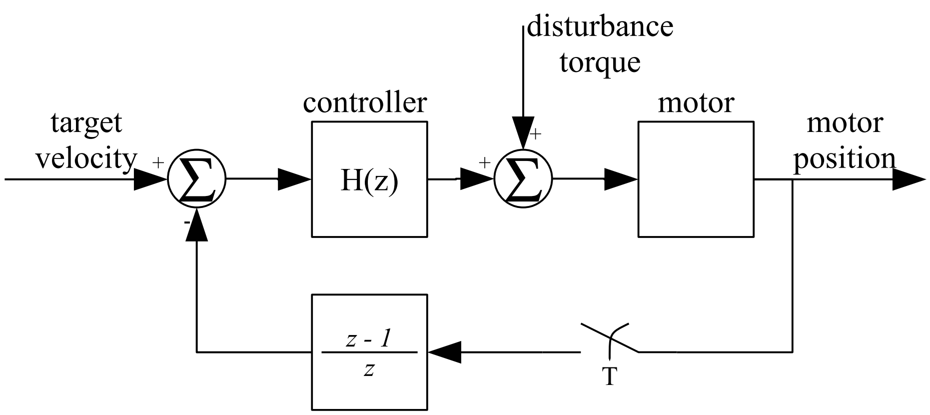

Figure 4: A System with Continuous Rotation.

Make a conservative estimate of the lowest speed of the motor before friction becomes a consideration in the controller design.

Solution:

If the motor never stops then the friction will always be a constant force which can be easily compensated for by the integrator in the controller. Therefore this

[1]Note that this implies a fairly hefty flywheel on the motor.

problem reduces to one of finding the amount of speed variation in the system, and keeping the motor speed above that figure.

As a worst case assume that the system disturbance torque is at the natural frequency where the system response is most sensitive, and furthermore that the system will tend to oscillate at that frequency due to the effect of breakaway torque. The system sensitivity at that point is

$$S\left(\omega_0\right) = \frac{k_d}{2\zeta\omega_0} = \frac{100\mathrm{RPM}/\mathrm{N}\cdot\mathrm{m}/\mathrm{sec}}{(1.4)\left(14^{\mathrm{rad}}/_{\mathrm{sec}}\right)} \simeq 5^{\mathrm{RPM}}/_{\mathrm{N}\cdot\mathrm{m}} \tag{2}$$

The maximum disturbance that will be seen is $0.8\,\mathrm{N}\cdot\mathrm{m}$ peak to peak, so the maximum peak disturbance will be half that, or $0.4\mathrm{N}\cdot\mathrm{m}$. This means that the minimum speed that must be maintained is at least

$$RPM_{min} = \left(T_{dmax}\right)\left(S\left(\omega_0\right)\right) = (0.4\mathrm{N}\cdot\mathrm{m})\left(5^{\mathrm{RPM}}/_{\mathrm{N}\cdot\mathrm{m}}\right) = 2\mathrm{RPM} \tag{3}$$

The above example analyzed the operating region for which one could assume continuous motion. But what happens when you have friction or backlash, and the motion *isn't* continuous? The following sections investigate this.

2.2 PD Control and Discontinuous Motion

When one uses proportional-derivative control in a mechanism with no backlash all of the nonlinear effects of friction will work with the proportional control to help damp the motion of the motor, and the system will tend to have fairly robust stability assuming that the proportional gain isn't pushed to absurdly high levels. The problem with this approach is that the system will never reach zero error, and it isn't sufficient to guarantee stability in a system with backlash.

Figure 5 on page 6 shows the block diagram of the motor-controller that is used for this section and the following one. The motor is driven by a current-output amplifier, so in the absence of friction it acts as a double integrator. The motor output position is sampled

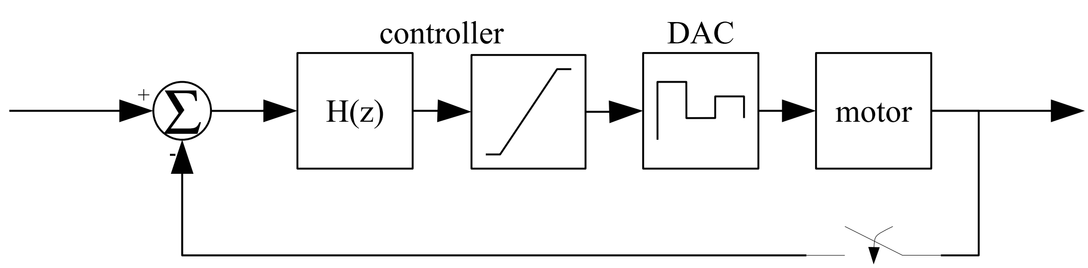

Figure 5: Example System used in the Text.

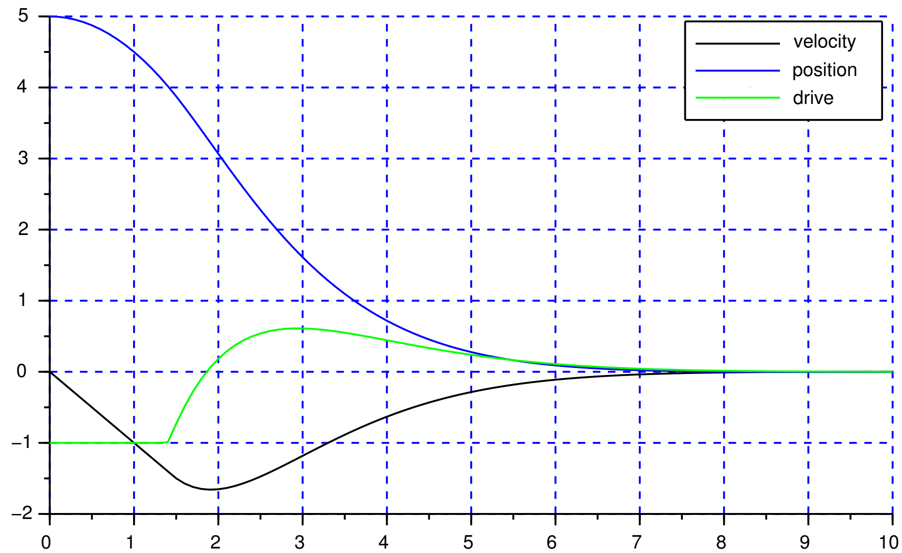

Figure 6: Ideal Motor Response with PD Control.

and compared with the motor command position. This position error is applied to a linear compensator (H), and the resulting the current command is saturated to stay within the limits of the current drive, then the command is applied to a DAC which acts as a zero-order hold.

Figure 6 on page 7 shows a plot of an ideal motor that's being controlled by a PD controller. Figure 7 on page 8 shows a similar plot, only this time the motor has some substantial friction. Compare these plots to get an idea of the difference in behavior. The system with friction never reaches the target position — it gets to about 0.4 units away, then stops. Moreover, the drive to the motor never falls to zero, which both wastes power and heats up the motor. With some determination you can play with the derivative term and the starting point, and you can make the motor with friction settle out to zero — but not reliably. I have seen this lead designers to releasing product that didn't work correctly, because while this may work in the lab, it is not robust to changes in environment or setup. It just doesn't work in general.

The system response shown in Figure 7 can be adequate if the position error is smaller than your needed precision and the power dissipated in the motor will not cause harm. So using a proportional-derivative controller without special accommodation for friction can be a successful strategy — as long as you either control your friction or control your expectations.

Adding backlash into the mix can present difficulties, however. Figure 8 on page 8 shows the response of a motor with backlash and no friction[2]. Initially the motor response is

[2]A model of a motor with backlash and no friction isn't entirely realistic, but it serves to illustrate that a system with backlash can get into "funny" oscillation modes. Indeed, while I cheated somewhat in generating

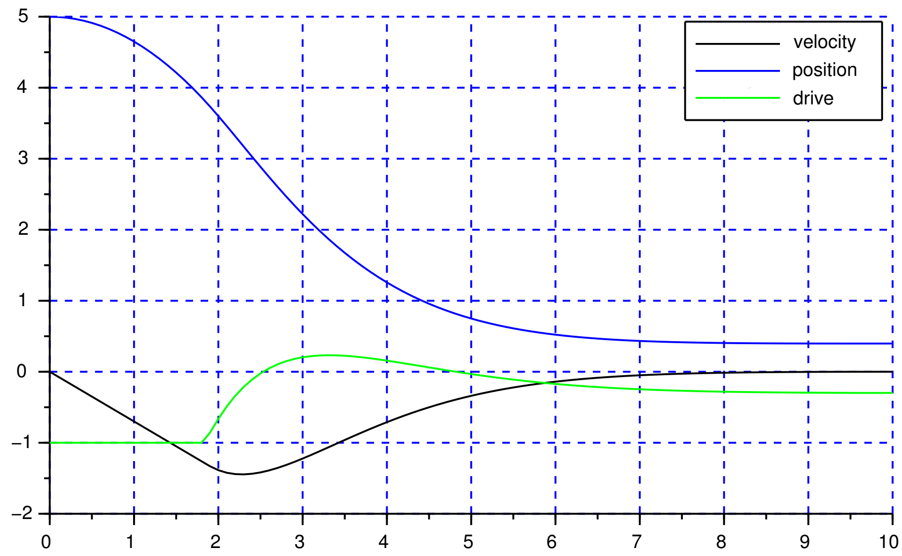

Figure 7: Response of Motor with Friction to PD Control.

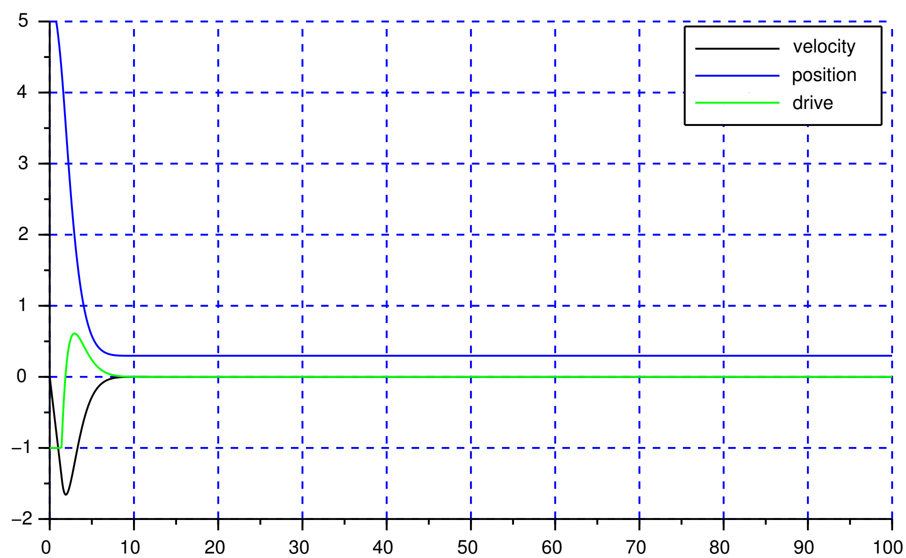

Figure 8: Motor Response with Backlash and PD Control.

more or less correct, but the motor overshoots somewhat, then gains speed in the opposite direction and "thunks" into the backlash, then commences to oscillate with the limit cycle shown.

A PD controller can be used in a system with friction and no appreciable backlash, as long as one is willing to accept the fact that the target point will never be reached, and that as a consequence the drive to the motor may stay on for long periods of time. Applying PD control to a motor with friction and backlash is less safe, as indicated by Figure 8. Such systems can be designed to be stable, but the steady-state error cannot be guaranteed to be zero and stability verification must be done on the nonlinear system model which can be difficult.

Some systems that include a motor with friction can be controlled with a PD controller in a satisfactory manner. The primary requirement is that it not be necessary to come to rest exactly on the target point, that one takes any other significant disturbance torques into account with ones friction torque, and that some residual drive to the motor can be tolerated in the long term. In that case the maximum error that can be expected from such a system is the one where the torque command just equals the breakaway torque. If you assume that the breakaway torque (F_s) and other disturbance torques (F_d) are known, and if you calculate the proportional gain from the angular error to motor torque ($k_p \sim {}^{\mathrm{torque}}/_{\mathrm{angle}}$) then this limiting angle can be found:

$$\theta_s = \frac{F_s + F_d}{k_p} \tag{4}$$

When the system settles, there will generally be a position error and the controller will continue driving the motor. Depending on the system design this may cause excessive power dissipation. If this angular error is within an acceptable range and the power dissipation is acceptable then you don't need to do anything at all; just use the controller as it is. If the angular error or power dissipation is too large then you'll need to use one of the nonlinear compensation methods presented later in this paper.

2.3 Motor Feedback Strategy

In the system shown in Figure 8 on page 8 the limit cycle is due to the uncontrolled acceleration of the motor while the backlash is being taken up. Because the motor does not settle to exactly zero position error, the drive persists and the motor accelerate. During the interval when the output is not moving the motor is building up speed, without the controller having any feedback. Once the backlash is taken up the motor impacts the gears or mechanism that cause the backlash, and the output inevitably overshoots which causes the cycle to repeat. In this case the primary cause of trouble is the fact that the motor accelerates uncontrollably — if the motor speed could be controlled even while the mechanism was within its slack area then the limit cycle could be reduced in amplitude or eliminated altogether.

this example, I have seen instances of systems like this that can seem perfectly innocent in the lab, and only start exhibiting malign behavior in the hands of customers. This is not a situation you want to allow.

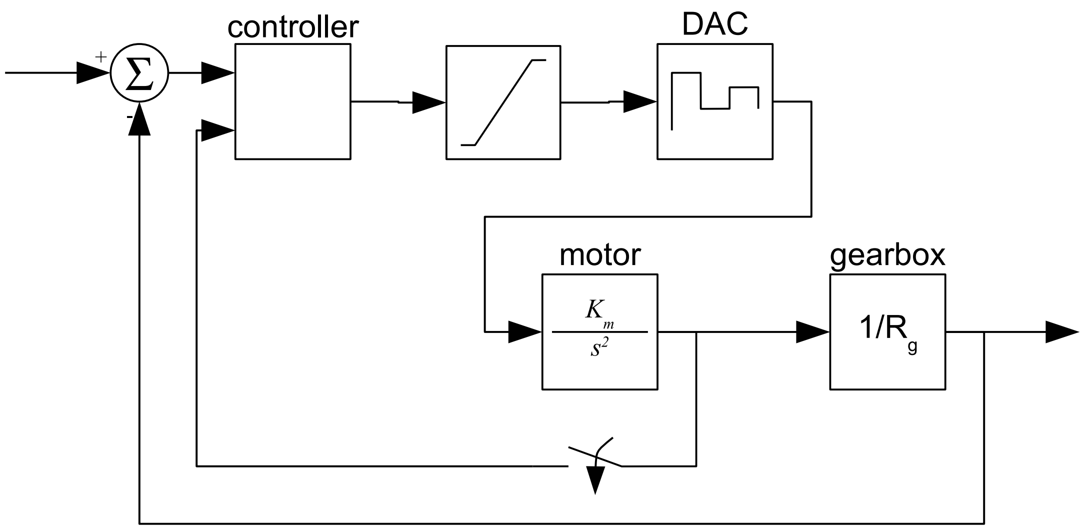

Figure 9: Controller with Motor Position Feedback.

Figure 9 on page 10 is a block diagram of a system with a sensor added to the motor shaft and used to control the motor velocity. The sensor can be a position sensor (which would require a differentiator in the controller) or a tachometer. The controller uses this feedback to determine the speed of the motor shaft itself. The advantage of this feedback is that when the controller can sense the behavior of the motor itself it can prevent the motor from going too fast as it takes up the slack in the gearbox, thereby reducing or eliminating the overshoot when the slack is taken up.

2.4 PID Control

Figure 10 on page 11 shows a plot of an ideal motor that's being controlled by a PID controller. Compare this to Figure 11 on page 12, which shows a similar motor with friction. These plots certainly appear to be remarkably similar, except that the motor with friction is left with some residual drive to the motor. You would like a simple PID controller to be the solution to your problem.

There's a hidden problem, however: the friction in the motor leaves a slight residual position error on the motor that the controller cannot be overcome immediately. After a time, the motor with friction will go into a limit cycle, as shown in Figure 12 on page 12. This can be insidious because it may take quite some time before the limit cycle starts up[3] and because the level of friction generally varies greatly with temperature. If you're not watching for friction as a troublemaker, it can take a lot of head-scratching before you figure out what is causing the problem.

[3] In the simulation that generated Figure 11 on page 12, it took over 6000 seconds before the oscillation showed up — this in a motor that had apparently settled out after 20 seconds!

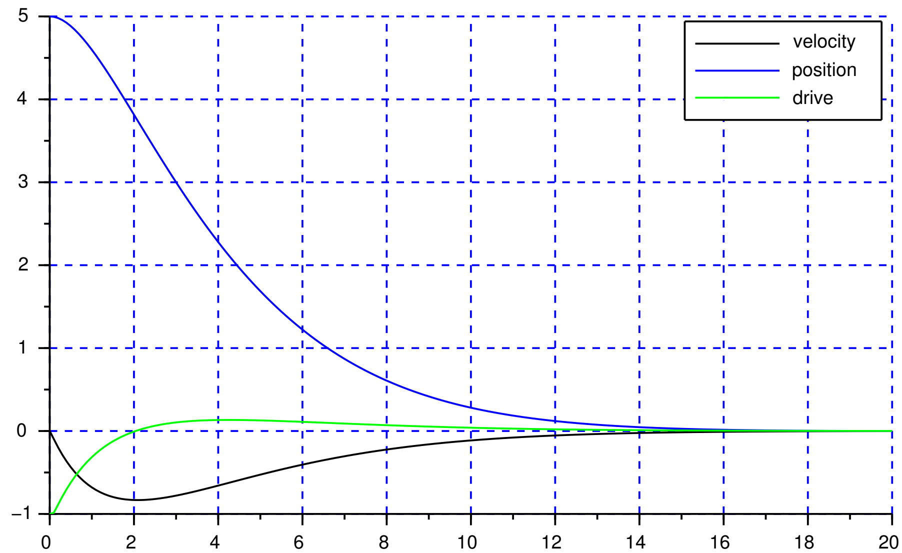

Figure 10: Ideal Motor with PID Control.

Applying a PID controller to a motor with both friction and backlash will likely either result in the behavior seen in Figure 8 on page 8 or Figure 12 on page 12 (or both on a bad day). Because the motor will never settle to exactly the right spot, and because the PID will never quit pushing you're almost guaranteed to see a limit cycle of some sort.

So far I've painted a grim picture — it would seem, at this point, that if you have a mechanism with friction or backlash that there isn't a thing that you can do to make your system work correctly. This is not so — you just can't make it work with "textbook" linear controllers. Friction and backlash are nonlinear effects, and because they are severe ones they demand nonlinear control strategies. Fortunately, these control strategies can be handled without using mathematics that are so advanced that your head explodes.

3 Nonlinear Compensators

When a straight linear controller doesn't provide adequate performance there are two nonlinear compensation schemes that have been developed over the years to deal with friction and backlash. These are pulse-width modulation of the motor drive and the use of deadband in the feedback loop.

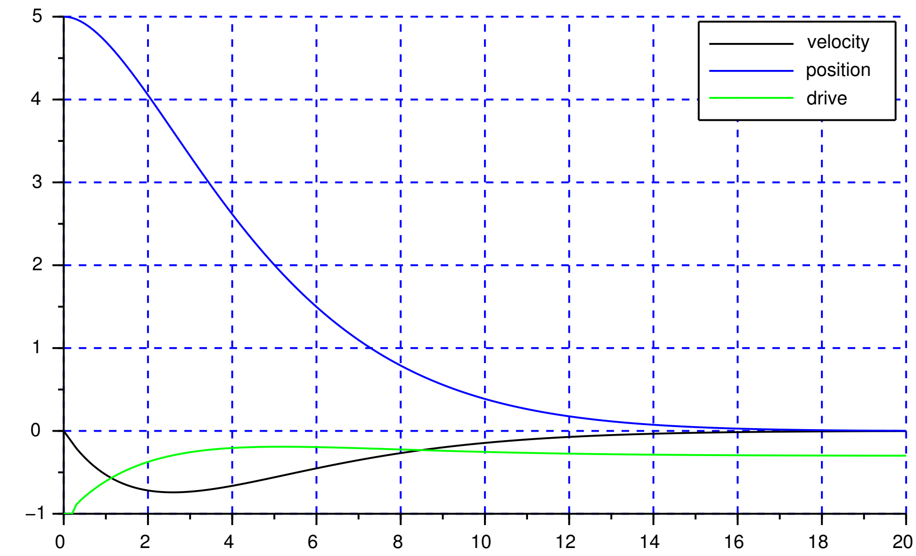

Figure 11: Motor with Friction under PID Control.

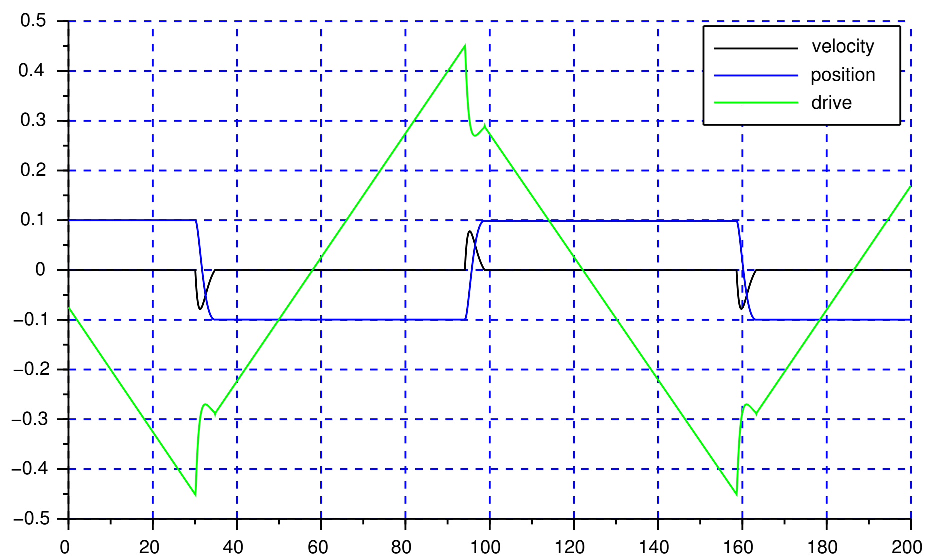

Figure 12: Motor with Friction under PID Control, Oscillating.

3.1 PWM Motor Drive

One fairly easy and effective measure that can be taken to increase servo system performance is to pulse-width modulate (PWM) the drive to the motor[4]. Rather than letting the drive to the motor fall continuously to zero from the maximum drive, the drive is allowed to fall to some point, then it is pulsed on and off with a duty cycle that provides the correct average drive. In cases where the motor can take the treatment the driver circuitry can be simplified by pulsing the full motor voltage on and off.

The advantage of a PWM drive to a motor which is limited by friction is that the motor will always move for small inputs. This means that the motor can be slowed down to a crawl without the jerkiness associated with a continuous drive. In fact, using PWM drive to a motor often means that the controller doesn't have to have an integrator to achieve zero steady-state position error, so you can often drop from a PID loop to a PD loop.

There are disadvantages, however. The motor will always move by a discrete amount, however small, and the difficulty of controlling the system well go up as this amount goes down. The velocity to drive relationship can be quite nonlinear, if not as bad as without PWM. Finally, the current pulses to the motor can be severe; full-voltage PWM drive requires that you use a motor with stout brushes in an assembly that can take the strong torque pulses.

3.1.1 PWM Drive Characteristics

If the motor is stopped, the amount that the motor will move in response to a single pulse depends on the level of friction in the motor, the torque generated at the motor armature in response to a pulse, and the length of the pulse. This can be found by calculating the motor motion in response to a single pulse.

The motor will accelerate in response to a pulse, then it will slide to a stop. The following discussion assumes that the pulse width is at least as long as the motor's electrical time constant and short enough that the motor velocity doesn't get large enough to create any restraining torque other than friction. Given that, and recalling that the motor velocity starts at zero, the motor's velocity and position profile will follow the relationship

$$\omega\left(t_0 + t_{on}\right) = \int_0^{t_{on}} \frac{T_s - T_f}{J_m} dt = T_p \frac{T_s - T_f}{J_m} \tag{5}$$

$$\theta\left(t_0 + t_{on}\right) = \theta_0 + \int \int_0^{t_{on}} \frac{T_s - T_f}{J_m} dt^2 = \theta\left(t_0\right) + \frac{t_{on}^2}{2} \frac{T_s - T_f}{J_m} \tag{6}$$

[4]This PWM is not the PWM that is applied to a motor by a switching amplifier. The PWM drive to the motor for the purposes of compensating for friction is not being done for efficiency as in the case of a switching amplifier, and the effect is lost if it is too fast. PWM motor drive frequencies are generally in the 10Hz to 1000Hz range, where modern switching amplifiers generally operate well above 20kHz. If you are using this technique on a motor driven by a switching amplifier, you need to pulse the drive command to the amplifier at whatever rate is appropriate to deal with friction.

where ω is the motor velocity, θ is the distance traveled by the motor, T_s is the drive torque on the motor, T_f is the friction torque, J_m is the moment of inertia seen by the motor armature and t_p is the pulse duration. After the pulse is removed the motor will slide to a stop in time t_s, which can be found from

$$t_s = \frac{\theta_{on} J_m}{T_f} = t_p \frac{T_s - T_f}{T_f} \tag{7}$$

where ω_{on} is the motor speed at time t_{on} obtained from (5).

The total distance that the motor travels in response to each pulse can then be determined:

$$\theta \left(t_0 + t_{on} + t_s\right) = \theta \left(t_0\right) + \frac{t_{on}^2}{2} \frac{T_s - T_f}{J_m} + \frac{t_s^2}{2} \frac{T_f}{J_m} \tag{8}$$

Substituting in (7), we can eliminate t_s from (8):

$$\theta \left(t_0 + t_{on} + t_s\right) = \theta \left(t_0\right) + \frac{t_{on}^2}{2} \frac{T_s - T_f}{J_m} + \frac{t_{on}^2}{2} \frac{\left(T_s - T_f\right)^2}{T_f J_m}$$

or

$$\theta \left(t_0 + t_{on} + t_s\right) = \theta \left(t_0\right) + \frac{t_{on}^2}{2} \frac{T_s \left(T_s - T_f\right)}{T_f J_m} \tag{9}$$

The stall torque, T_s, in the above equations is the torque generated by the motor when it's driven by the given pulse, and ignoring friction. If you know the characteristics you can derive this from the motor data using the motor torque constant and the drive current:

$$T_s = I_d k_t \tag{10}$$

where I_d is the drive current (determined by your drive circuit) and k_t is the motor's torque constant (which you'll find specified in the motor data sheet). If the motor is being driven by a voltage then the drive current can be found from the motor's armature resistance and drive voltage:

$$I_d = \frac{V_d}{R_a} \tag{11}$$

where V_d is the voltage applied to the motor and R_a is the motor armature resistance.

The above calculations indicate that the motor will always move in response to a pulse, but that the amount that the motor moves will be proportional to the square of the pulse width.

These calculations only hold if the motor is not moving before the onset of the pulse. Once the PWM duty cycle is high enough, the motor will not stop between pulses. The duty cycle at which the motor starts to move continuously is very constant over a fairly large range of pulse on-times; only when the PWM is so fast that the motor never gets a chance to move at all[5], or so slow that the motor's speed limited by other factors at the end of its

[5]Determined by the motor's L/R time constant and by the degree of "spring" in the armature before friction is overcome.

"on" time[6] does this not hold. In between those extremes, the critical duty cycle is equal to the ratio of the motor's friction torque divided by it's frictionless drive torque during a pulse:

$$\rho_c = \frac{t_{on}}{t_s + t_{on}} = \frac{t_{on}}{t_{on}\left(1 + \frac{T_s + T_f}{T_f}\right)} = \frac{T_f}{T_s} \tag{12}$$

Example 2: Setting PWM Duration.

You are designing a motor controller for a motor with substantial friction. The motor has an armature resistance of 1.7Ω, a torque constant of $5.9^{\mathrm{mN \cdot m}}/_{\mathrm{A}}$ (millinewton-meter/amp), a rotor inertia of $3.8\mathrm{gram} \cdot \mathrm{cm}^2$ and a rotor inductance of $110\mu\mathrm{H}$. The motor is attached to a mechanism with a moment of inertia of $5\mathrm{gram} \cdot \mathrm{cm}^2$, a breakaway friction torque that can range from $2\mathrm{mN} \cdot \mathrm{m}$ to $5\mathrm{mN} \cdot \mathrm{m}$, and a running friction torque that can range from $1\mathrm{mN} \cdot \mathrm{m}$ to $2\mathrm{mN} \cdot \mathrm{m}$. You are driving this mechanism from a controlled-voltage source that can deliver from -5V to +5V.

- Find the minimum voltage needed to guarantee that the motor will always move. Set a running voltage that gives you a 20% margin over this value.
- Find the pulse on-time, t_{on}, that is required to limit the motor travel to no more than 5 degrees at a time.
- Find the motor travel for the above pulse time with the maximum expected friction.
- Compare this pulse on time with the motor's electrical time constant.
- Find the duty cycle where the motor no longer stops between pulses.

Solution:

We start by finding the minimum drive voltage to insure that the motor will always move. In order to move, the motor's armature torque must exceed the breakaway torque of $5\mathrm{mN} \cdot \mathrm{m}$. The torque at the motor armature is equal to the armature current times the torque constant, so to generate the breakaway torque this current must exceed:

$$I_{min} = \frac{5\mathrm{mN} \cdot \mathrm{m}}{5.9^{\mathrm{mN \cdot m}}/_{\mathrm{A}}} = 0.85\mathrm{A} \tag{13}$$

The voltage required to develop this current across the motor armature can be found from the motor armature resistance:

$$V_{min} = I_{min} R_a = (0.85\mathrm{A})(1.7\Omega) = 1.45\mathrm{V} \tag{14}$$

[6] In a voltage-drive mode this is determined by the motor's mechanical time constant; in a current-drive mode this depends on the amount of viscous damping in the mechanism.

To get a 20% safety factor the actual drive voltage will be set at $V_d = (1.2)\,V_{min} = 1.74\text{V}$.

The torque developed by a motor that is driven by a voltage depends on armature speed, but when the motor is not moving the developed motor torque at this drive voltage is

$$T_s = \frac{k_m V_d}{R_a} = \frac{\left(5.9^{\text{mN}\cdot\text{m}}/_{\text{A}}\right)(1.74\text{V})}{1.7\Omega} = 6\text{mN}\cdot\text{m}$$

This checks out as being 20% greater than the desired minimum drive torque of $5\text{mN}\cdot\text{m}$.

We can find the displacement vs. time from (6): $\Delta\theta = \frac{t_{on}^2}{2}\frac{T_s\left(T_s - T_f\right)}{T_f\,J_m}$, from which we can deduce the on time

$$t_{on} = \sqrt{^{2T_f\,J_m\Delta\theta}/_{T_s\left(T_s - T_f\right)}} \tag{15}$$

Our worst-case motion occurs when friction torque is at its lowest. So to find the maximum allowable on-time for our pulses we must use the maximum allowable displacement of $\Delta\theta_{max} = 5° = 0.095\text{radian}$ and the minimum friction torque of $T_f = 1\text{mN}\cdot\text{m}$. From this we find the on time:

$$t_{on} = \sqrt{\frac{2\,(1\text{mN}\cdot\text{m})\,(8.8\text{gm}\cdot\text{cm}^2)\,(0.095\text{radian})}{(6\text{mN}\cdot\text{m})\,(6\text{mN}\cdot\text{m} - 1\text{mN}\cdot\text{m})}} = 2.36\text{ms} \tag{16}$$

The displacement, and hence the speed of the motor in PWM mode, changes with friction. At maximum friction, $T_f = 2\text{mN}\cdot\text{m}$, the displacement will be

$$\Delta\theta_{min} = \frac{(2.36\text{ms})^2}{2}\frac{(6\text{mN}\cdot\text{m})\,(6\text{mN}\cdot\text{m} - 2\text{mN}\cdot\text{m})}{(2\text{mN}\cdot\text{m})\,(8.8\text{gm}\cdot\text{cm}^2)} = 0.036\text{rad} = 2.1° \tag{17}$$

Because the motor is being driven by a voltage source, the electrical time constant of the motor is simply the $^L/_R$ constant of the armature. For this motor this number is

$$\tau = \frac{110\mu\text{H}}{1.7\Omega} = 65\mu\text{s} \tag{18}$$

This number is significantly less than our chosen motor on-time, so it will not significantly affect the motor behavior.

The duty cycle at which the motor will run continuously depends on the motor friction, and ranges from

$$\rho_{min} = \frac{T_f}{T_s} = \frac{1\text{mN}\cdot\text{m}}{6\text{mN}\cdot\text{m}} = 0.17$$

to

$$\rho_{max} = \frac{T_f}{T_s} = \frac{2\text{mN}\cdot\text{m}}{6\text{mN}\cdot\text{m}} = 0.33$$

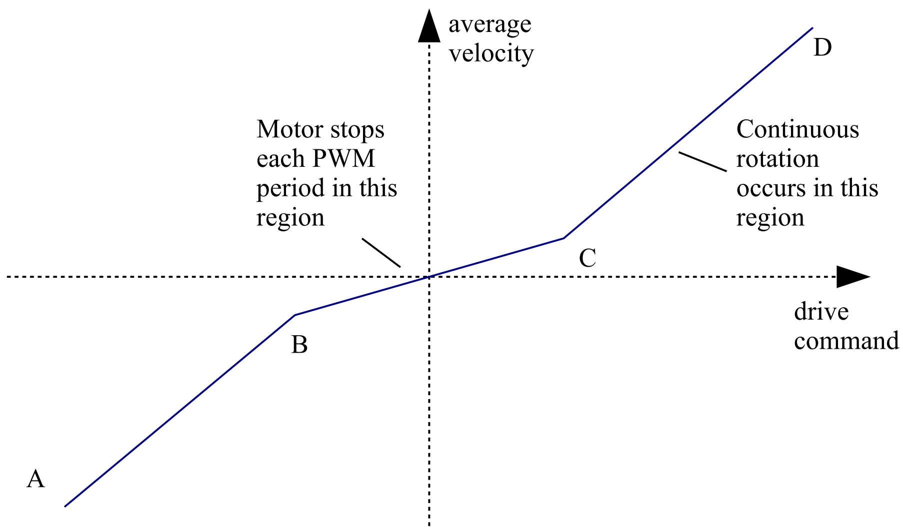

Figure 13: Motor average speed vs. Drive for mixed PWM & Continuous Voltage.

If the on-time of the motor pulse is fixed then the motor speed will depend on the duty cycle up to the point where the motor begins to run continuously. At that point the motor speed will depend on the nature of the motor driver and on any viscous drag in the mechanism. For a current driver the motor speed above the continuous-rotation duty cycle will be limited only by mechanism drag. For a voltage driver the motor speed will also be limited by the motor's back EMF.

As an example, consider a driver that delivers a controlled voltage to the motor, with the voltage transitioning from a constant-voltage drive to a PWM drive to maintain motor motion according to

$$V\left(u,t\right) = \begin{cases} u & \left|u\right| > V_f \\ V_f \cdot \mathrm{pwm}\left(\frac{u}{V_f}, t\right) & 0\mathrm{V} < u \le V_f \\ -V_f \cdot \mathrm{pwm}\left(-\frac{u}{V_f}, t\right) & -V_f \le u \le 0\mathrm{V} \end{cases} \tag{19}$$

where u is the voltage command, V_f is the drive voltage chosen to overcome friction, and $\mathrm{pwm}(r,t)$ generates a PWM pulse train with the specified duty cycle. Given this drive the average motor speed will follow the characteristics shown in Figure 13.

The slope of the velocity/drive curve in the continuous rotation regions (segments AB and CD) are a function of the motor's velocity constant and any viscous drag that may be present. The slope of the velocity/drive curve in the region full-stop region (line BC) depends on the length of the motor on pulse and on the drive current (or voltage). If the pulse on time is held constant and the pulse off time is varied to modify the duty cycle then the cycle time is

$$t_c = \frac{t_{on}}{\rho} \tag{20}$$

and the average velocity is just the position offset from 9 divided by the cycle time:

$$\bar{\omega}(\rho) = \frac{t_{on}^2}{2} \frac{T_s (T_s - T_f)}{T_f J_m} \frac{\rho}{t_{on}} = \rho \frac{t_{on}}{2} \frac{T_s (T_s - T_f)}{T_f J_m} \tag{21}$$

Figure 13 on page 17 and equations (9) and (21) combine to illustrate an important trade off when using PWM drive: Reducing on time will give you a smaller position increment and hence better accuracy, but doing so will reduce the slope of the average velocity vs. drive. So you can buy increased accuracy, but at the cost of making the break in the velocity/drive curve more pronounced. This break will cause difficulties in tuning a controller that get greater as the on time is reduced. This effect can be ameliorated by increasing T_s, but doing so reduces efficiency and can be hard on the motor's brushes and on the mechanical assembly.

Example 3: Gain Variations with PWM Drive.

The motor and PWM drive specified in 2 is to be used in a system with PD position control. The minimum on-time has been set to 2ms. Due to mechanical considerations the control system bandwidth must be held to 20Hz or less. Assume a sampling rate of 200Hz, and a running friction of $2\text{mN}\cdot\text{m}$.

- Find the maximum motor speed
- Find the inflection point in the velocity/drive curve
- Find the slopes high- and low-slope sections of the velocity/drive curve.

Solution:

With a running friction of $2\text{mN}\cdot\text{m}$, the current required from the motor will be

$$I_{run} = \frac{2\text{mN}\cdot\text{m}}{5.9^{\text{mN}\cdot\text{m}}/_{\text{A}}} = 0.35\text{A} \tag{22}$$

With 5 volts applied to the motor terminals the motor armature voltage will be

$$V_a = 5\text{V} - (1.7\Omega)(0.34\text{A}) = 4.4\text{V}$$

Using the torque constant as the speed constant[7], the maximum motor speed will be

$$\omega_{max} = \frac{4.4\text{V}}{5.9^{\text{mV}\cdot\text{sec}}/_{\text{rad}}} = 746^{\text{rad}}/_{\text{sec}} = 7120\text{RPM} \tag{23}$$

[7] In an ideal DC motor the torque and speed constants are the same. In most DC motors they are slightly different, however it is generally safe to assume that they are the same.

From Example 2 the inflection point occurs at a duty cycle of 33% with an on voltage of 1.74V. The average speed will be

$$\bar{\omega}_i = \rho \frac{t_{on}}{2} \frac{T_s \left(T_s - T_f\right)}{T_f \, J_m} \tag{24}$$

or

$$\bar{\omega}_i = 0.33 \frac{2\mathrm{ms}}{2} \frac{\left(6\mathrm{mN \cdot m}\right)\left(6\mathrm{mN \cdot m} - 2\mathrm{mN \cdot m}\right)}{\left(2\mathrm{mN \cdot m}\right)\left(8.8\mathrm{gm \cdot cm^2}\right)} = 4.5\,^{\mathrm{rad}}\!/_{\mathrm{sec}} \tag{25}$$

This duty cycle will be commanded when the voltage command is

$$V_i = (0.33)\,(1.74\mathrm{V}) = 0.574\mathrm{V} \tag{26}$$

So the inflection point is at $(V, \omega) = (0.574\mathrm{V},\ 4.5\,^{\mathrm{rad}}\!/_{\mathrm{sec}})$.

The slope of the low-slope portion of the curve can be found from the inflection point:

$$k_l = \frac{4.5\,^{\mathrm{rad}}\!/_{\mathrm{sec}}}{0.574\mathrm{V}} = 7.84\,^{\mathrm{rad}}\!/_{\mathrm{sec \cdot V}} \tag{27}$$

The two higher-slope curves are symmetrical, so the two slopes are equal. The upper curve ranges from the inflection point at $(0.574\mathrm{V},\ 4.5\,^{\mathrm{rad}}\!/_{\mathrm{sec}})$ to the maximum-speed point of $(5V,\ 746\,^{\mathrm{rad}}\!/_{\mathrm{sec}})$. The slope is

$$k_h = \frac{746\,^{\mathrm{rad}}\!/_{\mathrm{sec}} - 4.5\,^{\mathrm{rad}}\!/_{\mathrm{sec}}}{5\mathrm{V} - 0.574\mathrm{V}} = 168\,^{\mathrm{rad}}\!/_{\mathrm{sec \cdot V}} \tag{28}$$

which closely matches the value predicted from 1/km = 170 rad/volt-sec.

Note that these slopes differ by a factor that exceeds 1:20: in a control system this extreme gain variation would make tuning quite problematical. In a real world application you would either need to work harder to control friction, you would need to use some other control strategy in addition to PWM, or you would need to increase the PWM voltage above $1.74\mathrm{V}$, in an attempt to get a more linear curve.

Figure 14 on page 20 shows what happens when you apply PWM drive to a motor. This shows the same basic motor system as Figure 7 on page 8, but with PWM drive added. Notices that while it doesn't settle as rapidly as the frictionless motor in 6, if there are no disturbance torques it will eventually settle out to zero error – and without using a PID controller. Moreover, you an add integral action to this system and it *will* settle out to a steady value, unlike the oscillation seen in Figure 12 on page 12.

3.1.2 Implementing PWM Drive

There are a number of ways that PWM drive can be implemented to drive a motor. Four methods that are most often convenient are a pure analog implementation, an implementation using digital circuitry (either custom-built or as part of a microcontroller) and implementing PWM drive in software.

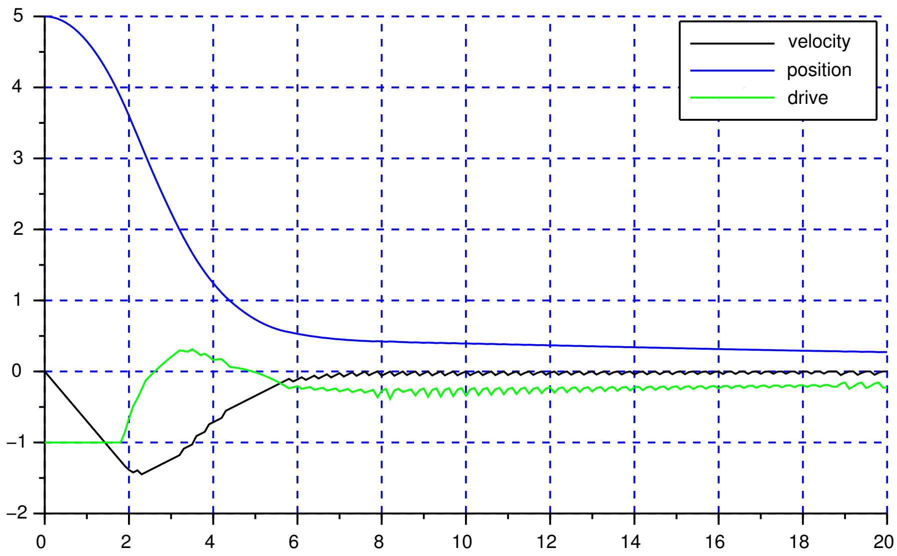

Figure 14: Motor with PWM Drive and PD Control.

Pure Analog: Pure analog PWM generators are most valuable in small control systems that are purely analog, but may also have application where a very small microprocessor is used for control and there aren't enough clock cycles for the processor to generate the PWM.

Figure 15 shows a block diagram of a simple analog PWM generator. The circuit generates a sawtooth wave at some frequency. The sawtooth wave is applied to a comparator along with the command voltage, and the output of the comparator is applied to a power amplifier which drives the motor. Note that above circuit has a disadvantage: the PWM "on" time is not fixed, but rather varies with the drive. Circuits that implement a constant "on" time can be built, but require more circuitry.

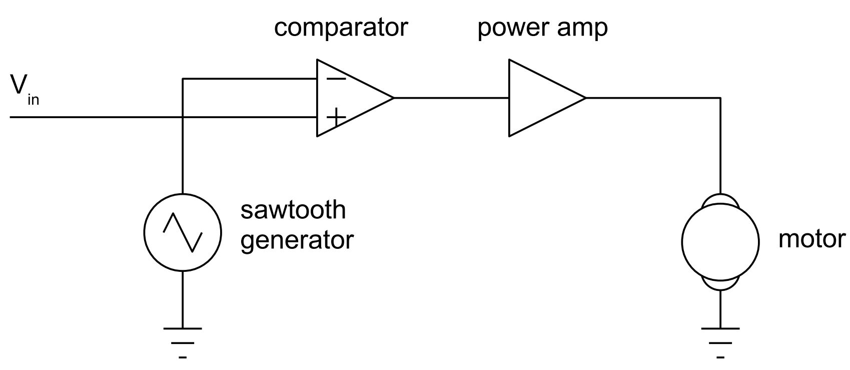

Figure 15: Analog PWM Generator.

Digital Hardware In many motion control systems the control loop is closed in software running on a microcontroller or microprocessor. Feedback from the plant is converted to digital for the processor to read, and plant drive is converted to analog. In systems of this type it often makes sense to generate the PWM signal in the digital realm, then amplify the resulting logic-level pulses to drive the motor.

Most dedicated microcontrollers have dedicated PWM hardware that can be used to generate an appropriate PWM signal to drive a motor. If your system does not, or if it isn't convenient to use the on-board PWM generator, then dedicated digital hardware can be designed instead. In this case it's probably best to study the data sheets for a few microcontrollers, decide which PWM generator features you want, and implement them.

Software Generated In the case where the final motor drive voltage is under software control (either through a DAC and a linear amplifier, or from a high-speed switching amplifier driven directly from the microprocessor) the necessary PWM waveform can usually be generated in software. The software should be structured with an interface routine that determines the necessary voltage and PWM duty cycle, and a driver that actually switches the drive on and off.

When PWM is generated in software it is generally best to switch the PWM signal on and off from an ISR driven by a hardware timer. It also generally works well to have the PWM on-time be determined by the timer cycle time, and the duty-cycle to be controlled by software.

3.2 Deadband

While PWM drive will allow you bring down the resting error in a system with friction, it is not possible to bring this error to zero. Nor will PWM drive give you much advantage at all in a system that has backlash. In a system that has integral action in the controller this small residual error can cause problems with hunting.

Since there is nothing that can be done to prevent the residual error, it is necessary to find a way to cope with it. The method that works best for this is to put some deadband in the controller's error term. Deadband sets the drive to the motor to zero when the input error is within some defined limit, so the motor will come to rest even if it is at a slightly incorrect position. The two most commonly used deadband methods are defined in (29) and (30) and shown in Figure 16.

$$u_{out} = \begin{cases} u_i + u_d & u_i < -u_d \\ 0 & -u_d \leq u_i \leq u_d \\ u_i - u_d & u_i > u_d \end{cases} \tag{29}$$

$$u_{out} = \begin{cases} u_i & u_i < -u_d \\ 0 & -u_d \leq u_i \leq u_d \\ u_i & u_i > u_d \end{cases} \tag{30}$$

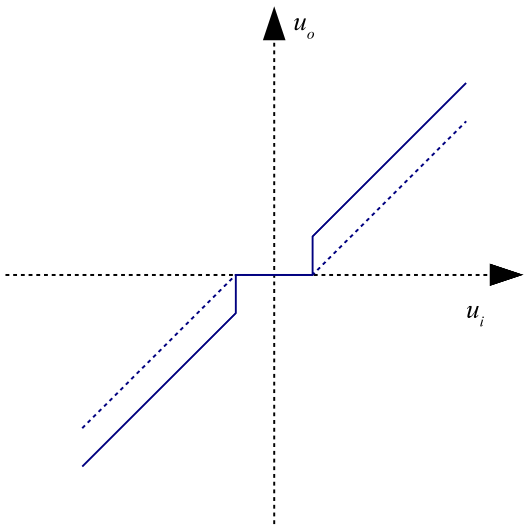

Figure 16: Deadband Input/Output Relationship.

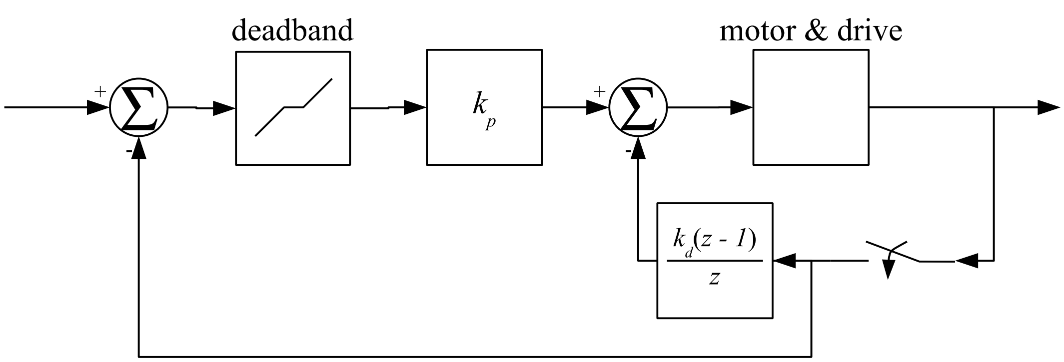

Figure 17: Deadband in a PD Controller.

Figure 17 on page 22 shows the preferred method of designing a controller with deadband for a plant with backlash or friction. The deadband is only applied the proportional action (and integral action, if you include it) — derivative feedback is treated separately. This is so that the mechanism output velocity will always have a damping effect on the motor motion, and so there won't be any "hiccups" in the motor drive if the error term passes through zero. An even better scheme to use is to combine the deadband in Figure 17 on page 22 with the direct motor feedback shown in Figure 9 on page 10.

For all that it solves problems, using deadband in a control loop with a PID controller can present some interesting difficulties, and should be approached with care. Deadband reduces the effective loop gain to zero at small errors. Because a PID driving a motor is only marginally stable as the loop gain decreases this can cause problems with settling, which are compounded by using PWM. These problems can be mitigated by making the integrator "leaky", so that when it gets zero input it's output slowly decays to zero — but if the mechanism is fighting an opposing force, then a leaky integrator means that it will not reach its target exactly. Various combinations of motor shaft feedback and output feedback can be used, but there is no one magic answer—as in all nonlinear control problems, one must experiment to find the best solution for the particular system one is working with.

Also, setting the amount of deadband in a system can be problematic. At a minimum the deadband should be set to the amount that the motor can jump if the error is just outside the deadband region, to prevent small limit cycles. For a mechanism that has significant friction and backlash, however, this deadband will prove to be too small in practice. Normal system design will usually include quite a bit of cut-and-try tuning of deadband parameters to arrive at acceptable system behavior.

4 Mechanical Mitigation

One of the best pieces of advice I ever got when I was a young man studying control theory in school was "well, change the plant, then!" It took me years to appreciate the wisdom of this advise. It can be easy to focus on trying to make the World's Most Clever Controller, and neglect opportunities one may have to fix the root problem. While it is often the case that a mechanism design is cast in stone, it is equally often that making tweaks in the mechanical arrangement that one is trying to control may well be less expensive than trying to design a control rule that is capable of making the mechanism one has behave, and then sourcing hardware that can implement the control rule.

While friction and backlash can be difficult or impossible to eliminate from a mechanism, there are some measures that the mechanical designer can take that may mitigate their effects. These measures must be considered on a whole-system basis, with cooperation between the control system designer and the mechanical designer, and often with input and cooperation between electronics designers and software designers. It is when these difficult cross-discipline problems crop up that the word "system" in "control system design" become more apparent.

Simply reducing backlash and friction is an obvious step. One would like to assume that the mechanical designers will do this as a matter of course. This is usually the case, but a wise control system engineer will gently query as to how much effort was put into the exercise, and whether the mechanical designer understands the trade off. Answers like “this is the same mechanism that we used in last year’s product” (when last year’s product didn’t have the same performance constraints) or “I copied this mechanism from a handbook” should raise red flags. Answers that show that the mechanical designer has done his homework should be respected, although you should review the designer’s calculations if you can.

If you are lucky enough to come into a project at the ground floor, then try to get the attention of the mechanical designers (and the project technical lead, if you can). Offer to do system simulations and calculations early on, to help the mechanical designers establish a budget for friction and backlash that they can use in the design of their mechanisms. This will help your project stay within its time and expense budget, and will help get everyone on board and aware of the issues, instead of being unpleasantly surprised during the integration phase of the project.

Treating the effects of stiction separately from dry friction is often a good idea. Dry friction often (except for its interaction with the integrators in the controller) has a stabilizing effect on a system, while stiction has a destabilizing effect (the limit cycle seen in Figure 12 on page 12, for instance, would not exist if it were not for stiction). Mechanical engineers are sometimes prone to concentrate on dry friction alone when designing a system, yet often it is measures that lower stiction at the expense of higher dry friction that prove to be most beneficial when friction is preventing a mechanism from reaching its desired accuracy.

In cases where motion is continuous, or consists of mostly continuous moves, adding flywheels to motors is a direct method to reduce the effect of friction: as seen in 1 a motor with significant flywheel action can run quite slowly without friction problems.

Remember that backlash effects can be mitigated with motor shaft feedback such as the system shown in Figure 9 on page 10, and note that this feedback can take many forms. Often one doesn’t need the worlds best feedback here, just a smooth and reliable indication of motor speed.

5 Conclusion

While there are no magic control techniques that will eliminate the problems caused by friction and backlash, using the methods outlined in this paper will allow a control system designer to get the most out of a system with these effects.

References

[Wes06] Tim Wescott: "Applied Control Theory for Embedded Systems". Elsevier, 2006

[CHB96] Brian Armstrong and Carlos Canudas de Wit: "Friction Modeling and Compensation", "The Control Handbook", CRC Press, 1996, pp 1369-1382.

About the Author

Tim Wescott has 20 years of experience in industry, designing and implementing algorithms and systems for digital signal processing and closed loop servo control. His extensive practical experience in translating concepts from the highly abstract domain of mathematical systems analysis into working hardware gives him a unique ability to make these concepts clear for the working engineer.

Wescott Design Services

Wescott Design Services provides design and analysis help for companies with problems in control systems, communications systems, and embedded systems design. We specialize in solving difficult problems in embedded control and in detection and estimation problems. With our expertise in understanding the mathematics that underlie modern control and communications theory, and our ability in designing practical circuits and software, we can help you realize your goals quickly, efficiently, and successfully.

http://www.wescottdesign.com

Applied Control Theory for Embedded Systems

This book is written for the practicing engineer who needs to develop working control systems without going back to school for years. It is aimed directly at software engineers who are learning control theory for the first time, however it has found favor engineers of all stripes when they are tasked with implementing control loops. It teaches theory, but it also covers many real-world issues in much greater depth than is taught in the sort of theory course you'd find in many universities.

Applied Control Theory for Embedded Systems is published by Elsevier. Learn more about the book an the author's own web page, http://www.wescottdesign.com/actfes/actfes.html, and from the publisher's page on the book:
http://www.elsevier.com/wps/find/bookdescription.cws_home/707797/description#description