VisSim Tutorial Series

Simulation of Motion Control Systems

William Erickson, *Indramat-Rexroth.*

VisSim Tutorial Series

Simulation of Motion Control Systems

Excerpted with permission from *Modeling and Visual Simulation in Industry*, A. Mulpur and P. Darnell, International Thomson Computer Press, Boston, MA, 1997.

Other books in the VisSim Tutorial Series include:

- *Biomedical Systems: Modeling and Simulation of Lung Mechanics and Ventilator Controls Design*. Mike Borrello, Metran America, Inc.
- *Fundamentals of Mathematical Modeling and Simulation*. Peter Darnell and Arun Mulpur, Visual Solutions, Inc.
- *Heating, Ventilation and Air Conditioning (HVAC) Controls: Variable Air Volume (VAV) Systems*. Nebil Ben-Aissa, Johnson Controls, Inc.
- *Introduction to 6-DOF Simulation of Air Vehicles*. Robert Josselson, ITT Aerospace Systems Group.
- *Simulation of Communication Systems*. Eugene Estinto, Eritek, Inc.

Table of Contents

Introduction

A typical motion control system incorporates a number of embedded feedback loops. The output from one loop becomes the input, or command, to the next loop. Each of these loops has a specific purpose.

The motion control system illustrated here contains three embedded feedback loops, one each for current (or torque), velocity, and position.

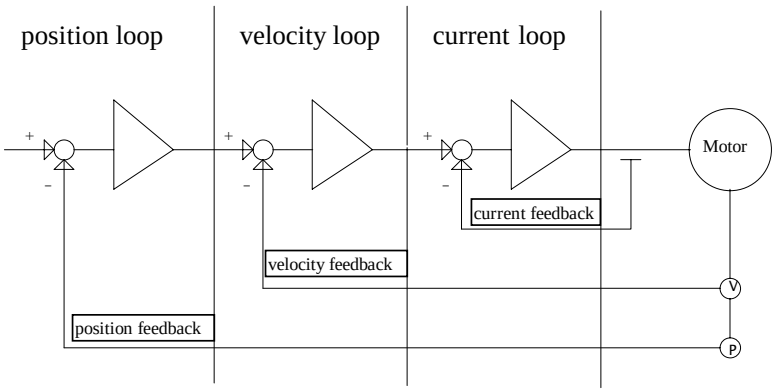

Figure 1. Three loop control system

The position loop receives a position command and compares it to the actual position (X), the difference (error) is processed and issued as a velocity command to the velocity loop. The velocity loop compares this command with the actual velocity (V), the difference is the velocity error which, in turn, is processed and issued as a current command to the current loop. The current loop takes this command, compares it with the actual current, processes it, and outputs the current to the motor.

The function of each embedded feedback loop is different. The position loop provides positional accuracy and static stiffness while the velocity loop provides dynamic stiffness. The current loop ensures that current quickly and accurately reaches the commanded value.

By definition $stiffness = \frac{\Delta F}{\Delta X}$, as shown below.

Figure 2 shows this relationship pictorially. A weight attached to a spring sits at rest. When a force is applied the weight changes position, Δx, causing the spring to compress. The stiffness is the ratio of applied force divided by the change in position, Δx.

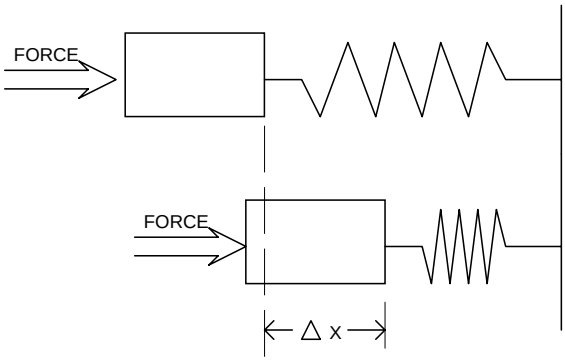

Figure 2. Static stiffness

In English units, stiffness is expressed in lbs/inch for linear motion and in-lb/radian for rotary motion. In this example, the stiffness is called the *static stiffness* because it is provided by the position loop. If the applied force is time variant, stiffness is called *dynamic stiffness* and is controlled by the velocity loop.

Depending upon the intended application, the topology of the motion control system may change. Some systems will include additional process loops, while others will eliminate them. The downside of closing multiple imbedded loops is the need to ensure that the loops don't fight each other. The innermost loop must be the fastest (the current loop in Figure 1), with each successive loop slower than the one before.

Dynamic Performance Measurement

A design engineer measures the dynamic performance of a motion control system by determining the system's bandwidth. Bandwidth is the measure of the system's ability to follow a command. The higher the bandwidth the closer the match.

The traditional approach for measuring bandwidth is to issue a sinusoidal command to the system and compare it to the system's response. At low frequencies, the two will be nearly identical. As the frequency of the sinusoid increases, a phase delay is seen between command and response.

Figure 3 shows a modified motion control system. The command is changed from a step with a constant amplitude to a sinusoid whose amplitude is constantly changing. The period of the sinusoid is 100 sec (0.01 Hz) and the system's response is virtually identical to the command.

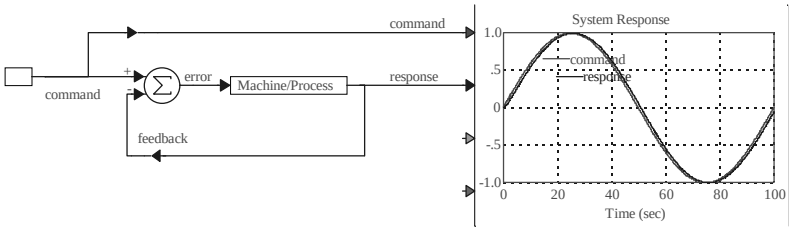

Figure 3. System response to a low frequency sinusoidal command

Increasing the frequency of the command causes a dramatic change to occur in the system's response. As shown in Figure 4, the command is increased in frequency to 1 radian/sec (0.16 Hz). Not only is the system's response delayed from the command, but the amplitude of the waveform is also much lower.

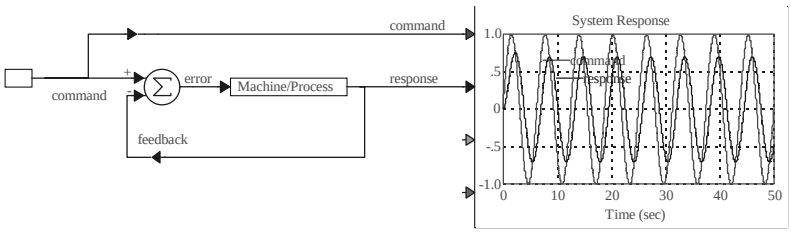

Figure 4. System response to a higher frequency sinusoidal command

Typically, system bandwidth is defined as the frequency that yields a 70.7% (-3db) response of the commanded value. The commanded frequency in the simulation above produces such a response; therefore, the bandwidth of the system is 1 radian/sec (0.16 Hz).

The previous plots have been time domain plots that look at amplitude over time. An alternative approach is to look at either magnitude or phase versus frequency plots, called Bode plots. It is much easier to determine bandwidth from a Bode plot than from a time plot; however, it is also much more difficult to create a frequency domain plot than a time domain plot – at least it was until simulation programs such as VisSim came about.

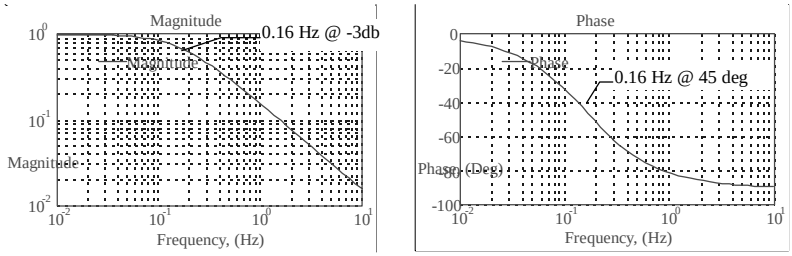

Figure 5. Bode plots of magnitude and phase for the system in Figure 4

The bandwidth can be taken from either a magnitude or phase plot. In a magnitude plot, bandwidth is the frequency at which the amplitude falls to 0.7 of the input. In a phase plot, it is the frequency at which a 45° phase shift occurs. Both plots in Figure 5 show a bandwidth of approximately 0.16 Hz, the same value seen in the time plot of Figure 4.

Classical Control System Analysis

The foundation of the preceding discussion has been the closed-loop feedback system. Since these systems are dynamic in nature the mathematical analysis of these systems generally involves the use of differential equations. The mathematics involved in the analysis can be greatly simplified through the use of the Laplace transform, which changes the frame of reference from one of time to one of frequency.

For the purpose of this discussion, it isn't necessary to be comfortable working with differential equations or even Laplace transforms. All that will be required is a basic understanding of algebra. The objective of this analysis is to describe how the output of a system changes as a function of a change in input to the system. The mathematical representation of this is called the transfer function. The blocks used to derive the closed-loop transfer function are directly applied to model the system with VisSim.

Deriving a Simple Transfer Function

A block diagram of a simple closed-loop feedback system is shown in Figure 6. The objective is to obtain the transfer function (the ratio $C_{(s)}/R_{(s)}$) for the simple system.

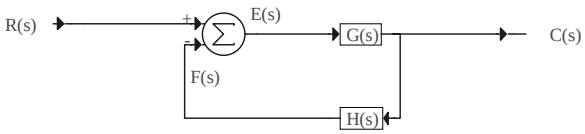

Figure 6. Simple closed-loop feedback system block diagram

In Figure 6, the following applies:

R(s) = input command

C(s) = output response

G(s) = forward gain

H(s) = feedback gain

F(s) = feedback

E(s) = error

(s) = Laplace operator

The error, E(s), is equal to the input command, R(s), minus the feedback value, F(s).

$$E(s) = R(s) - F(s)$$

The feedback value, F(s), is equal to the output, C(s), multiplied by the feedback gain, H(s).

$$F(s) = C(s) * H(s)$$

Substituting the equation for the feedback, F(s), into the equation for error, E(s), yields the following relationship:

$$E(s) = R(s) - C(s) * H(s)$$

The output, C(s), is equal to the error, E(s), multiplied by the forward gain G(s).

$$C(s) = E(s) * G(s)$$

Substituting the equation for E(s) into the above equation for C(s) yields

$$C(s) = G(s) * \left[R(s) - C(s) * H(s)\right]$$

Multiplying this through to eliminate the outermost parentheses yields

$$C(s) = R(s) * G(s) - C(s) * G(s) * H(s)$$

Collecting all of the C(s) terms on one side of the equation yields

$$R(s) * G(s) = C(s) + C(s) * G(s) * H(s)$$

which simplifies to

$$R(s) * G(s) = C(s) * \left[1 + G(s) * H(s)\right]$$

The objective is to isolate the input command term, R(s), and the output term, C(s), on one side of the equation. This is accomplished in two steps: first divide both sides of the equation by R(s) and simplify.

$$\frac{R(s) * G(s)}{R(s)} = \frac{C(s) * \left[1 + G(s) * H(s)\right]}{R(s)}$$

This simplifies to

$$G(s) = \frac{C(s) * \left[1 + G(s) * H(s)\right]}{R(s)}$$

The second and final step is to divide both sides of the equation by [1+G(s)*H(s)] and again simplify.

$$\frac{G(s)}{\left[1 + G(s) * H(s)\right]} = \frac{C(s) * \left[1 + G(s) * H(s)\right]}{R(s) * \left[1 + G(s) * H(s)\right]}$$

Simplifying this gives the desired relationship, or transfer function, between output and input terms.

$$\frac{G(s)}{1 + G(s) * H(s)} = \frac{C(s)}{R(s)}$$

This simple transfer function is the fundamental building block used in developing the transfer functions for more complex models. The transfer function for this model is the forward gain, G(s), divided by the product of forward gain, G(s), multiplied by feedback gain, H(s).

Deriving a Second Order Transfer Function

Figure 7 illustrates a velocity controlled servo system. It is similar in appearance to the closed-loop feedback system described in the previous section. It contains a single feedback loop with only one element (Kf). The forward loop is a bit more complicated with a total of four elements. The terms Kt and J are motor-defined terms and are usually obtained from the motor's data sheet. The terms Kp and Ki are drive velocity loop parameters. In this age of digital drives, these are usually programmable. The element Kf is used for scaling the output velocity to the input velocity and can be left as a value of 1 if the velocity units used throughout are rads/sec.

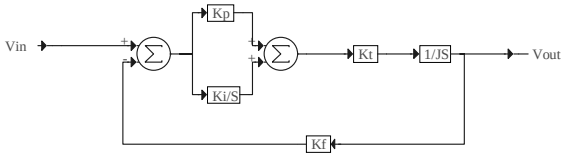

Figure 7. Velocity controlled servo system

In Figure 7, the following applies:

Kp = velocity loop proportional gain (amp-sec/radian)

Ki = velocity loop integral gain (amp-sec^2/radian)

Kt = motor torque constant (in-lb/amp)

J = system inertia (in-lb-sec^2)

S = Laplace operator

Vin = velocity command

Vout = actual velocity

The first step in determining this system's transfer function is to identify the forward gain, G(s), and the feedback gain, H(s) by multiplying all the terms in the forward path

$$G_{(s)} = Kp + \frac{Ki}{S} * \frac{Kt}{JS} = \frac{Kt(Kp + Ki/S)}{JS} \qquad H_{(s)} = Kf$$

Through substitution, the equation for the transfer function Vout/Vin = G(s)/[1+G(s)H(s)] becomes

$$\frac{Vout}{Vin} = \frac{\frac{Kt(Kp + Ki/S)}{JS}}{1 + \frac{KfKt(Kp + Ki/S)}{JS}}$$

Multiplying out the terms in parentheses yields

$$\frac{Vout}{Vin} = \frac{\frac{KtKp}{JS} + \frac{KiKt}{JS^2}}{1 + \frac{KtKfKp}{JS} + \frac{KtKfKi}{JS^2}}$$

To simplify the equation, multiply both the numerator and denominator by JS^2 to yield

$$\frac{Vout}{Vin} = \frac{\frac{KtKp}{JS} + \frac{KiKt}{JS^2}}{1 + \frac{KtKfKp}{JS} + \frac{KtKfKi}{JS^2}} * \frac{JS^2}{JS^2} = \frac{KtKpS + KiKt}{JS^2 + KtKfKpS + KtKfKi}$$

Next, normalize the equation by multiplying both the numerator and denominator by 1/J (the intent is to leave the S^2 term in the denominator with a coefficient of 1).

$$\frac{Vout}{Vin} = \frac{KtKpS + KiKt}{JS^2 + KtKfKpS + KtKfKi} * \frac{1/J}{1/J} = \frac{\frac{KtKp}{J}S + \frac{KiKt}{J}}{S^2 + \frac{KtKfKp}{J}S + \frac{KtKfKi}{J}}$$

For the purpose of this exercise, set Kf = 1, and Vin and Vout in units of rads/sec to yield

$$\frac{Vout}{Vin} = \frac{\frac{KtKp}{J}S + \frac{KiKt}{J}}{S^2 + \frac{KtKp}{J}S + \frac{KtKi}{J}}$$

The denominator in this equation is called the characteristic equation and can be used to predict the natural frequency (Wn) and damping factor (Z) of the second order system.

The generic form of the characteristic equation is

$$S^2 + 2ZWnS + Wn^2$$

Solving for Wn and Z yields

$$Wn = \sqrt{\frac{KtKi}{J}} \qquad Z = \frac{Kp}{2} * \sqrt{\frac{Kt}{KiJ}}$$

In the above two equations, only four parameters dictate the dynamic performance of the velocity loop. A question that arises frequently in the application of motion control products is what to do with the drive tuning parameters to accommodate changes in load inertia. Since the motor torque constant, Kt, is fixed and not subject to change, the values for Kp and Ki must be modified to accommodate inertia changes. For a specified dynamic performance, both Kp and Ki must change in proportion to inertia. For example, if the total inertia is doubled, both Kp and Ki must be increased by a factor of 2 to maintain consistent dynamic performance.

Example Bandwidth Calculation

Using the following motor / drive system, determine the system's velocity loop bandwidth and damping:

torque constant, Kt = 5.93 in-lb/amp

motor inertia, J = 0.2036 in-lb-sec^2

proportional gain, Kp = 3 amp-sec/rad

integral gain, Ki = 375 amp-sec^2/rad

Solving for Wn and Z yields

$$Wn = \sqrt{\frac{Kt * Ki}{J}} = \sqrt{\frac{5.93 * 375}{0.2036}} = 104 rad / \sec = 16.5 hz$$

$$Z = \frac{Kp}{2}\sqrt{\frac{Kt}{Ki * J}} = \frac{3}{2}\sqrt{\frac{5.93}{375 * 0.2036}} = 0.42$$

Second Order VisSim Model Development

The second order system is simulated using VisSim by utilizing the same model just used to derive the second order transfer function. A step velocity command of 1 rad/sec is applied at time 0. Figure 8 shows VisSim's predicted response in the time domain.

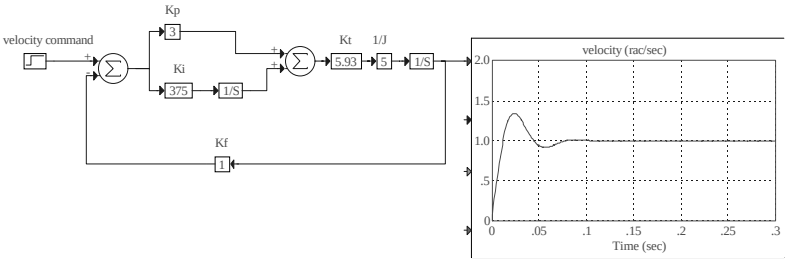

Figure 8. Second order VisSim model

Figure 9 shows VisSim's response in the frequency domain, which yields direct information on the bandwidth.

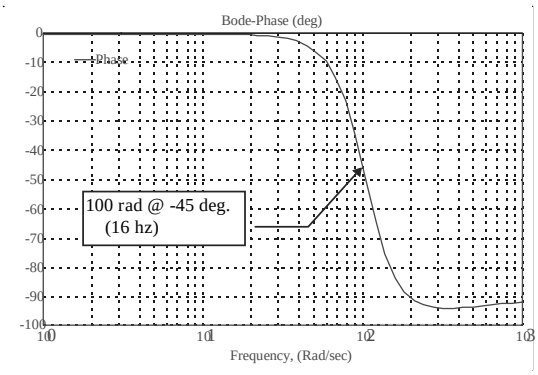

Figure 9. Phase response of second order model

Simulation Accuracy

By comparing the simulation's step response with the "real" system, as shown in Figure 10, an almost perfect match is revealed, indicating that the second order model is quite adequate.

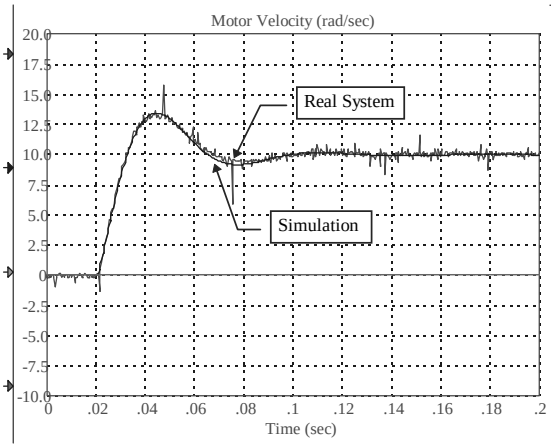

Figure 10. Comparison of simulated and actual step responses

Bear in mind earlier comments about embedded loops – *the inner loop must run faster (have higher response) than the outer loop, and the closer the loops run, the greater the interaction* – the second order system just modeled assumes a perfect current loop (infinite response), which outputs the commanded current the instant the command is given.

In the real world, however, this is never the case. The motor windings contain inductance and resistance that limit the rate at which current can rise. As higher and higher performance is required from the velocity loop, the influence of the real-world current loop combined with the motor's resistance and inductance becomes important and can no longer be ignored.

To illustrate this point, the second order model can be expanded to include the effects caused by the motor's winding resistance and inductance. A simplified model of the motor's electrical circuit is shown in Figure 11. The intent is to introduce a first order delay equal to the electrical time constant of the motor. In this example, the electrical time constant of the motor is 5 msec (that is, in one time constant, the actual motor current reaches 63% of the commanded value). It will take five time constants for the current to achieve the full commanded value.

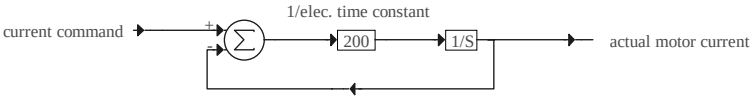

Figure 11. Simplified motor model

This is confirmed in Figure 12. A step current command of 1 amp is given. In 5 msec, the actual current rises to approximately 0.6 amps and in 30 msec (5 time constants), it achieves the commanded value of 1 amp.

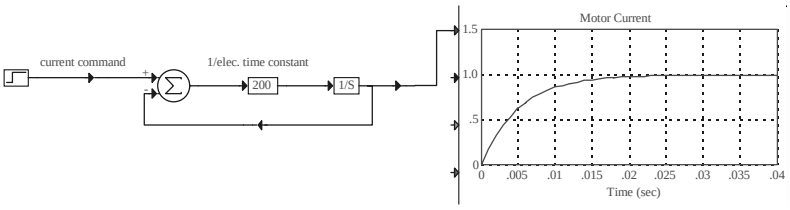

Figure 12. Current step response of the simplified motor model

The impact of this crude motor model can now be tested on the second order model. Shown in Figure 13 is the response to a step command in velocity of 1 rad/sec. Comparing this with the earlier step response makes it obvious that the effects of this motor can't be ignored.

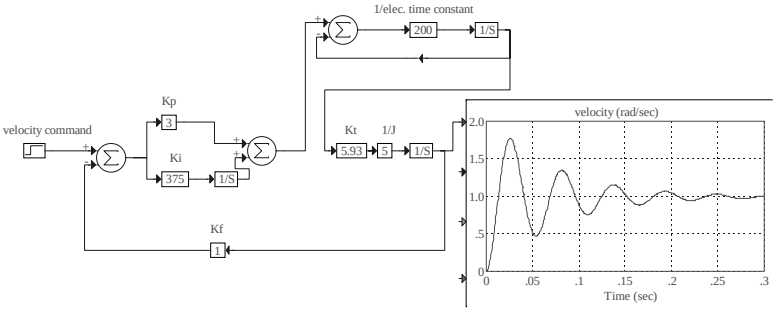

Figure 13. Second order VisSim model including the motor's delay

This same problem arises when dealing with a closed position loop. At low position loop gains, it is possible to ignore the influence of the velocity loop; however, as gains are increased, it becomes necessary to include the velocity loop within the model.

Simulation Guidelines

Another potential problem concerns the simulation process. The simulation step size (or sample time) must be faster than the fastest loop in the model to prevent erroneous results. In some instances, the model "blows up" when run, generating simulation errors. In other situations, the simulation appear normal but the results are skewed.

The models presented thus far have used a sample time of 1 msec. Simulations presented later will require faster sampling. A good habit is to decrease sample time by a factor of 10, once the model is functioning appropriately, to determine if performance changes. If performance does change, faster sampling is required. The tendency may be there to leave sample times very short to obtain the greatest accuracy, but it can become quite frustrating to wait minutes for each run of a simulation to complete.

A basic rule for developing models is KIS, which stands for "Keep It Simple." The more complex the model the more information will be required about the real system and the more difficult the model will be to use — especially if it is intended to be used by others. Use only the complexity necessary to achieve reliable results.

Developing a Second Order Velocity Loop Model

This section builds on the previously developed second order velocity loop model by improving the motor model, adding a current loop, and enclosing the model within a position loop.

Servo Motor Design

The motor model illustrated earlier in Figure 8 incorporates the mechanical aspects (torque and inertia) of the motor. The new model addresses the electrical aspects of the motor, including its resistance, inductance, and counter electromotive force (emf).

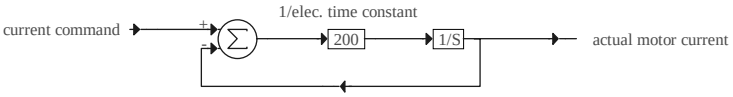

Figure 14. Simplified motor electrical model

In Figure 14, the motor model is represented as a simple first order delay function with the delay due to the motor's electrical time constant, L/R. This model is overly simplified. In actuality, the input to the motor is not current, but voltage. A voltage is applied to the motor's terminals causing a current to flow in the motor's windings. The initial rate of current rise is dictated by the motor's inductance, while the final current flow is dictated by the motor's resistance. The corrected model is shown in Figure 15.

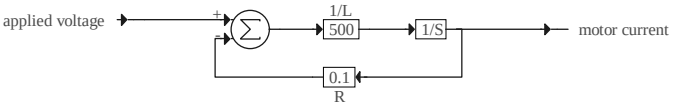

Figure 15. Corrected motor electrical model

As models are developed it is always good to test individual components before incorporating them into the main design. It is easier to identify and correct a problem when the model is simple than when it has been incorporated into a complete system, which may include hundreds of individual blocks. For the above model, it is easy to prove if it is correct. The final (steady-state) current is V/R, where V is applied voltage and R is motor resistance. The electrical time constant for the motor, τ_c , is L/R, where L is motor inductance and R is motor resistance.

$$I_{ss} = \frac{V}{R} = \frac{100}{0.1} = 1000 amps \qquad \tau_c = \frac{L}{R} = \frac{.002}{0.1} = 0.02 \text{ sec}$$

Figure 16 shows that the simulation results are exactly as predicted above: the final current is 1000 amps while the time constant is 0.02 sec (630 amps at time = 0.02 sec). Thus, this component of the model is correct and can be incorporated into the main model, as shown in Figure 17.

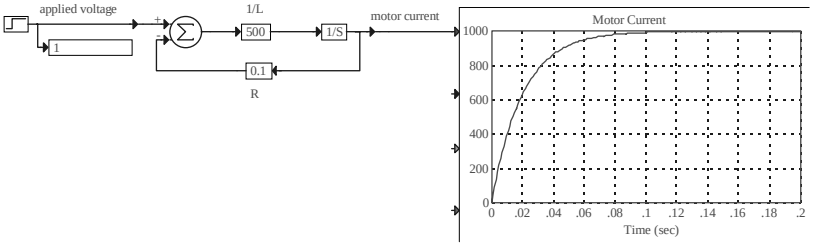

Figure 16. Motor current response to a step in applied voltage

The next element to add to the model is the motor's back electromotive force constant (bemf) or voltage constant (Kv). As motor speed increases, it generates a voltage that is opposite in polarity to the applied voltage, which has the effect of reducing the voltage available to force current into the motor. The net effect is that for a given applied voltage there is a finite speed that the motor can achieve. Assuming a lossless motor, the final speed is equal to the applied voltage divided by the voltage constant. However, this final speed will be reduced somewhat due to frictional losses in the motor.

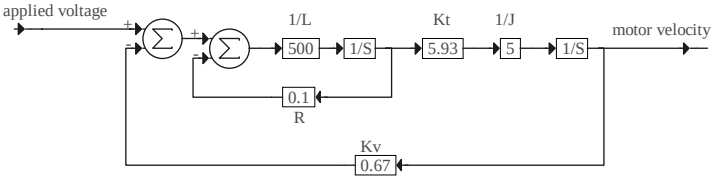

Figure 17. Complete motor model

In Figure 18, the motor's inductance and resistance are added to the earlier model, which incorporated only the motor's torque constant and inertia. In addition, the motor's voltage constant, Kv, is added. To test this model, the same 100 volt step input used in Figure 16 is applied. The final velocity (ω) equals

$$\omega = \frac{V}{Kv} = \frac{100}{.67} = 149.25\, rad / \sec$$

As shown in Figure 18, the final velocity is approximately 150 rad/sec. There are now two integrators in the motor model (one tied to the inductance and the other to the inertia) creating a second order model. Therefore, the time constant is not as easy to predict as with the previous motor model. The fact that this is a second order model also explains the overshoot seen in the velocity response.

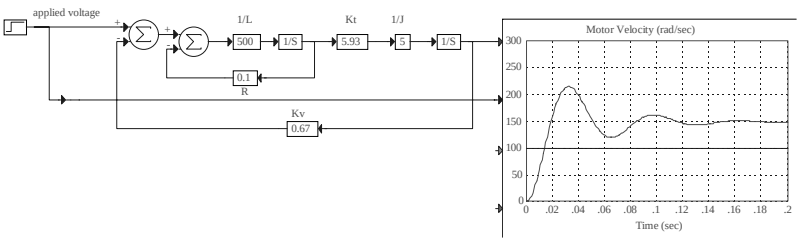

Figure 18. Motor velocity resulting from a step in applied voltage

The real motor reacting to an applied voltage does not exhibit the amount of overshoot indicated above, nor does it achieve the predicted velocity. This is due to viscous damping that can be added to the model and is nothing more than a speed dependent loss. While adding viscous damping to the model is easy, it is frequently difficult to determine how much to apply from the manufacturer's motor data sheet. The model in Figure 19 has an unrealistically high level of viscous damping added (10 in-lb/rad/sec) to show its effect and the motor's response to a 100 volt command plotted. The overshoot and settling time are reduced, as well as the final velocity, which is only 120 rad/sec instead of 150 rad/sec, as in the simulation in Figure 19.

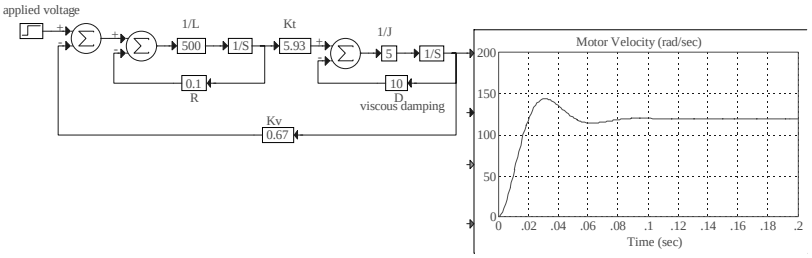

Figure 19. Complete motor model's response to a step in applied voltage

This motor model can now be incorporated into larger models of an entire motion control system. It requires a voltage command for input and provides motor velocity as the output. A total of six parameters, as listed in Table 1, are required for it to function.

Table 1. Motor Model Parameters

Parameter	**Units**	**Description**
input	volts	voltage applied to motor
L	henry	motor inductance
R	ohm	motor resistance
Kt	in-lb	motor torque constant
J	in-lb-s^2	motor inertia
Kv	v-s/rad	motor back emf constant
D	in-lb-s/rad	viscous damping
output	rad/s	velocity

Current Loop

The just completed motor model requires a voltage input, while the previously developed velocity loop model outputs a current to the motor. The addition of a current loop will make these two models compatible. The motor model shown in Figure 20 is outfitted with a current loop by adding two more elements: a current loop gain (KpI) and a summing junction. The gain of the current loop has the units volts/amp. A current command is summed with the current feedback, the error is amplified by the current loop gain, KpI, and output as a voltage command to the motor.

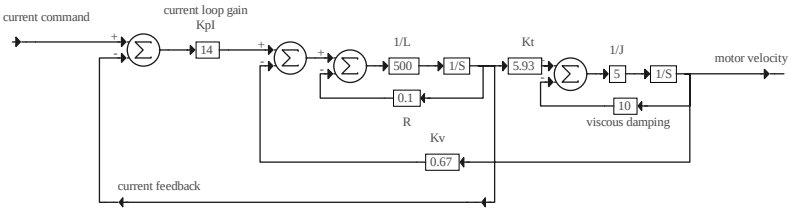

Figure 20. Motor model with current loop added

If the model is functioning correctly, a step current command at the input causes the velocity to ramp up while the current feedback levels off quickly to the commanded value. This is shown in Figure 21.

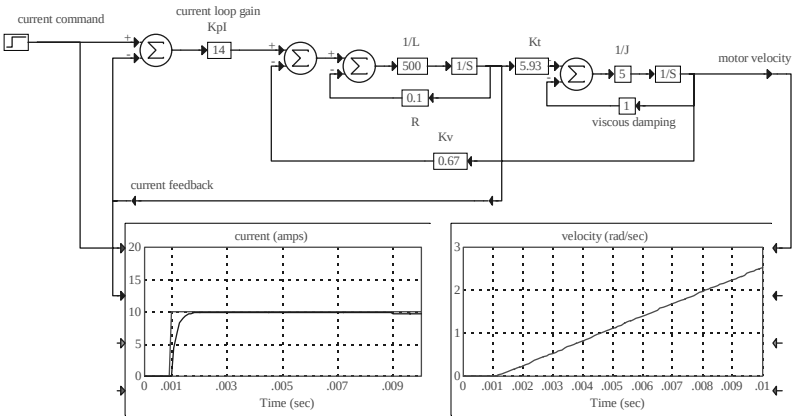

Figure 21. Motor's velocity response to a step current command

A 10 amp current command is issued 0.001 sec into the simulation. Within 1 msec the current feedback reaches 10 amps and, as anticipated, the velocity begins ramping. The slope of this ramp is

$$accel\ rate = \frac{torque}{inertia} = \frac{10 * 5.93}{.2} = \frac{296.5\ rad}{s^2}$$

$$velocity = accel\ rate * time = 296.5 * .009 = \frac{2.6\ rad}{S}$$

Ten msec into the simulation (9 msec after the current command is issued), the velocity is 2.5 rad/s, exactly as predicted.

Velocity Loop

The next step in the model development closes the velocity loop around the motor and current loop model. This is accomplished by adding the following elements to the model:

- Two summing junctions
- One velocity loop proportional gain (Kp)
- One velocity loop integral gain (Ki)
- One integrator (1/S)

A velocity command is summed with the velocity feedback signal and the difference (or error) is applied to both the proportional gain loop (Kp) and the integral gain loop (Ki). The result of each is summed together and becomes the command to the current loop.

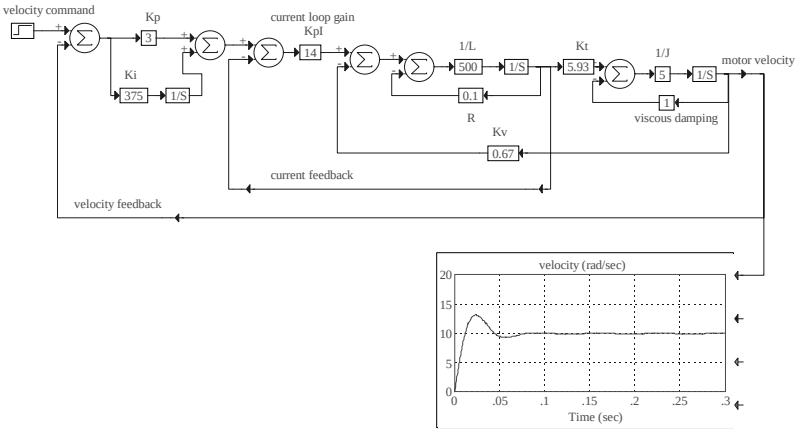

Figure 22. Complete third order velocity loop model

The simulation shown in Figure 22 shows the system's response to a 10 rad/sec step velocity command. The response is virtually identical to the one shown earlier in Figure 8 for a simple second order system which demonstrates that under the conditions of this simulation, the simple model would be more than sufficient.

Table 2 lists the parameters required for the model shown in Figure 21. As models are developed, especially when used by others, it is wise to limit the complexity of the model and only include those elements that are likely to impact the model's intended use. The model in Figure 22 requires nine parameters; the simplified model in Figure 8 requires only four parameter (Kp, Ki, Kt, and J). Both have equally reliable results.

As with any model it is always possible to further refine and define the system. Not included in the model shown in Figure 22 are such elements as voltage and current limits, an integrating current loop, and power bridge (that is, pulse width modulation effects). In the vast majority of motion control simulations motor/drive model complexity beyond what has been demonstrated is unnecessary. In practice, the simple second order model is sufficient for most situations.

Table 2. Velocity Loop Model Parameters

Parameter	Units	Description
input	rad/s	velocity
Kp	amp-sec	vel. loop proportional gain
Ki	amps	vel. loop integral gain
KpI	volts/amp	current loop prop. gain
L	henry	motor inductance
R	ohm	motor resistance
Kt	in-lb	motor torque constant
J	in-lb-s^2	motor inertia
Kv	v-s/rad	motor back emf constant
D	in-lb-s/rad	viscous damping
output	rad/s	velocity

Position Loop

The final step in developing the system model is to incorporate a position loop into the diagram. This involves closing one additional loop around the already-closed velocity and current loops. Only two additional elements are added to the velocity loop model to accomplish this: another summing junction and position loop gain (PLG). The units of PLG are 1/sec.

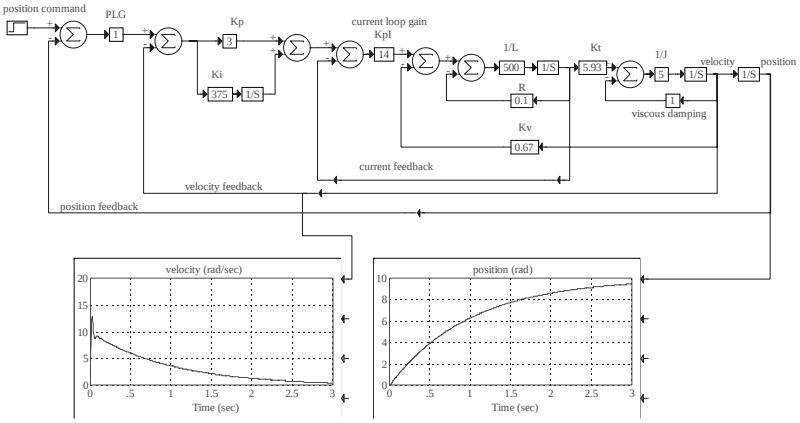

Figure 23. Complete position loop model

Position loop gain (PLG) is most easily described as the inverse time constant of the position loop.

$$PLG = \frac{1}{t_c}$$

In Figure 23, the time constant is 1 sec. Thus, given a step command in position, the system's response is such that in 1 sec, 63% of the final position is achieved and in 3 sec, 95% of the final position is achieved.

Another way to look at position loop gain is in terms of its output-to-input relationship (transfer function). The output is the command to the velocity loop and therefore has the units rad/sec, while the input is position in radians. In industry, PLG is most frequently expressed as

$$PLG = \frac{units / \min \; of \;\; velocity \; command}{0.001 \, unit \; of \;\; error}$$

Traditionally, a PLG of 1 is the "magic" gain (everybody seems to start with this gain — and frequently ends with this gain). For example, if the units in use are English, the PLG is expressed as "1 inch per minute per mil," meaning one inch per minute of velocity command for each thousandth of an inch of position error. Examining this further yields

$$PLG\,1 = \frac{1\,in / \min}{0.001\,in} = \frac{1\,in}{\min} * \frac{1}{0.001\,in} = \frac{1}{0.001 \min} * \frac{1 \min}{60 \sec} = \frac{1}{0.06 \sec}$$

Recall that earlier PLG was also expressed as

$$PLG = \frac{1}{t_c}$$

Therefore, for a PLG of 1 IPM/mil, the time constant of the position loop is 60 msec and the PLG is

$$PLG = \frac{1}{.06} = 16.67$$

Programming PLG = 16.67 into the system model causes the system to respond to a step command in such a manner that at 0.06 seconds, 63% of the command value will be achieved and after 0.18 seconds (3 time constants), 95% of the commanded value is achieved.

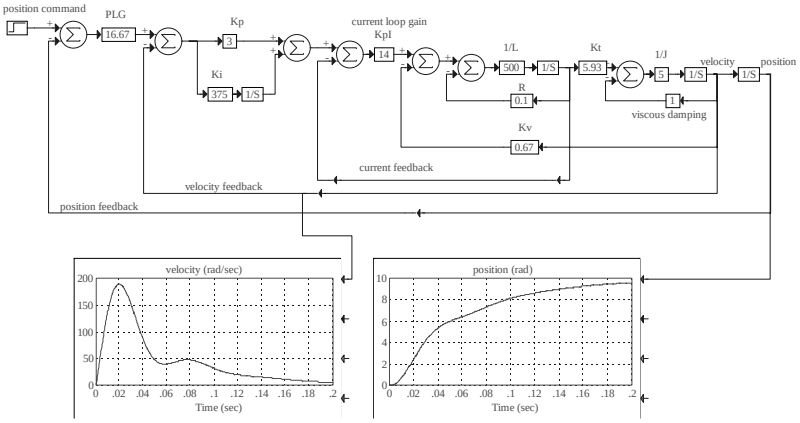

Figure 24. System response to a step position command

As predicted, with a command of 10 radians the model in Figure 24 shows 6.3 radians at t = 0.06 sec and 9.5 radians at t = 0.18 sec. Notice also that the position curve is not a smooth exponential curve but rather has a perturbation at about 0.04 sec. This effect is due to the velocity loop response. Each embedded loop must operate faster than the previous loop. In fact, to rule out the affects of an inner loop, it must operate an order of magnitude (10x) faster. In practice, a loop separation of 5:1 is usually acceptable.

The next simulation shown in Figure 25, increases both Kp and Ki by a factor of 10, which increases the bandwidth of the velocity loop. The position response is now a perfect exponential curve.

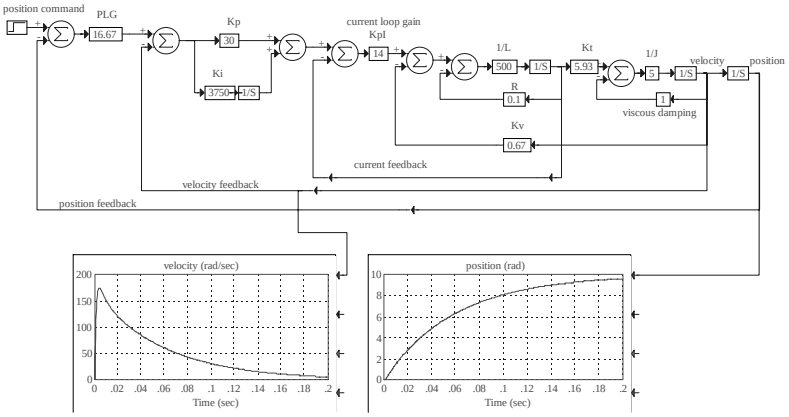

Figure 25. System response with proper separation of position and velocity loops

The position loop just described is a following error system. This means that if the system is operating at some steady-state velocity, there is a constant following error. For example, if the system is operating with a gain of 1 IPM/mil at a speed of 500 in/min, the actual position lags behind the commanded position by 0.5 inch. This is seen by re-arranging the PLG formula as shown

$$PLG = \frac{in / \min}{.001\, in\ error}$$

Rearranging this yields

$$error = \frac{in / \min * .001}{PLG} = \frac{500 * .001}{1} = .5\ in$$

A speed dependent error is acceptable for point-to-point positioning applications, such as pick-and-place robots, but cannot be tolerated for applications, such as printing and packaging. For these applications, the solution is known as velocity feed forward. With velocity feed forward, a position command is issued to the position loop, and a velocity command is also issued to the velocity loop, bypassing the position loop.

To look at the affects of adding velocity feed forward to the model, a different command function is required. Instead of a simple step in position, a velocity-ramp-to-speed is used. To model a velocity-ramp-to-speed, two ramps of equal rate (100 rad/sec^2) are incorporated in the diagram, as shown in Figure 26. One ramp is delayed by 1 sec and of opposite direction. This provides a ramp to a velocity of 100 rad/sec. The resulting velocity is then integrated to obtain position.

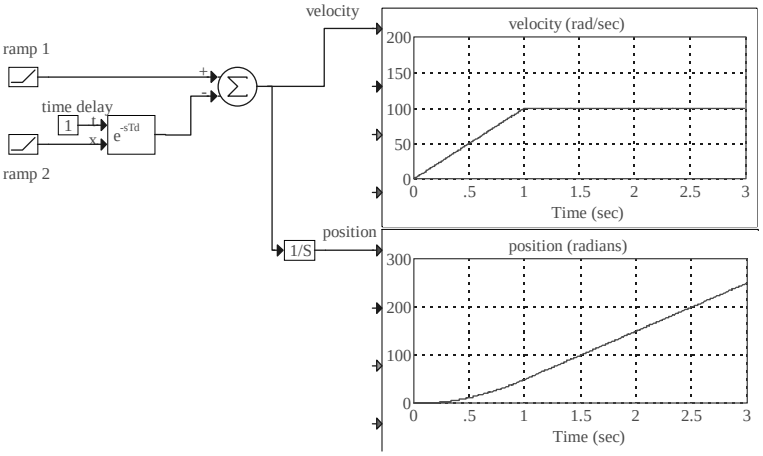

Figure 26. Velocity-ramp-to-speed command generator

The position output from the ramp generator is applied to the position input of the system model while the velocity output from the ramp generator is summed with the output of the position loop (velocity command). The first simulation, shown in Figure 27, is run with the old model and the just-developed ramp generator for command. An additional plot is added to show following error. The following error is predicted as

$$error = \frac{rad / \sec * 0.001}{PLG} = \frac{100 * 0.001}{16.67} = 6\ rad$$

This is exactly what the position error plot indicates in Figure 27.

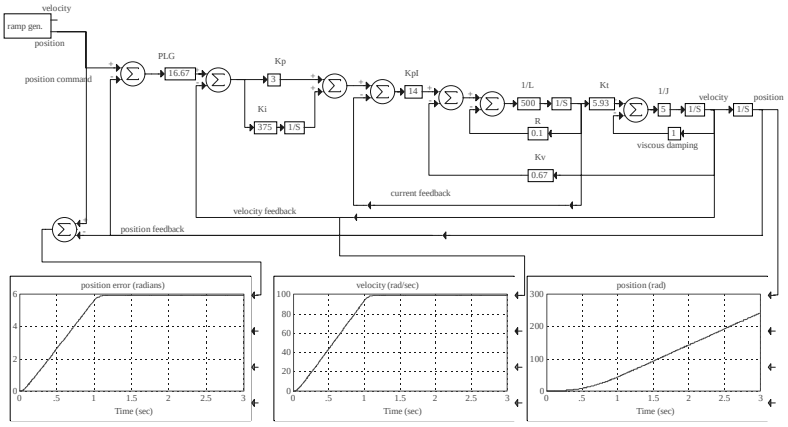

Figure 27. Position error without velocity feed forward

The simulation is run again, this time with velocity feed forward added, as shown in Figure 28. The velocity output from the ramp generator is applied directly to the velocity loop.

Notice the dramatic decrease in following error. What was 6 rads of error without feed forward is reduced to zero at steady-state. Even during acceleration, the error remains under 0.01 radian.

The model just presented is likely the most useful model for the simulation of motion control systems. The ramp generator can be replaced with any block that synthesizes the desired command profile. Velocity feed forward can be included or eliminated simply by connecting the velocity output of the generator block to the summing junction. If a less complex model of the motor is desired, the output of the velocity loop summer can be applied directly to the Kt block, effectively removing the current loop and the motor's electrical characteristics. The parameters, which must be provided to the model, are described in Table 3.

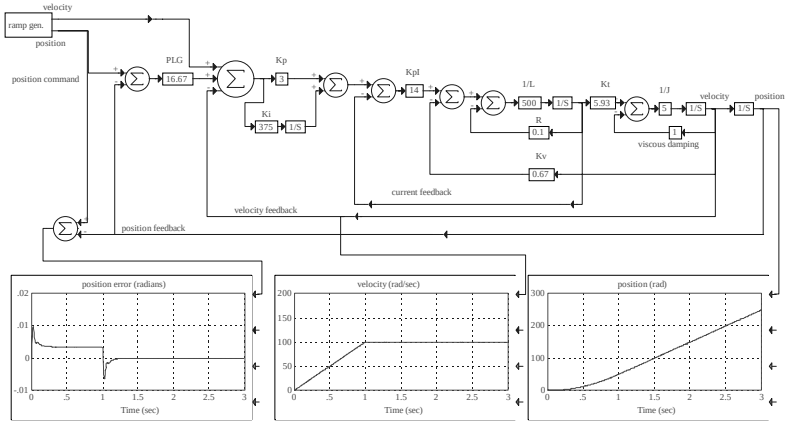

Figure 28. Position error with velocity feed forward

Table 3. Parameter Values

Parameter	**Units**	**Description**
input	rad	position
PLG	1/sec	position loop gain
Kp	amp-sec	velocity loop proportional gain
Ki	amps	velocity loop integral gain
KpI	volts/amp	current loop prop. gain
L	henry	motor inductance
R	ohm	motor resistance
Kt	in-lb	motor torque constant
J	in-lb-s^2	motor inertia
Kv	v-s/rad	motor back emf constant
D	in-lb-s/rad	viscous damping
output	rad	position