Slow Down to Speed Up

It may not seem logical, but you can boost the productivity of a servo system by slowing down its operation. Kevin McCarthy, Chief Technology Officer with Dover Motion, explains the paradox.

In servo system design, the ancient fable of the tortoise and the hare turns out to include some practical real-world advice. As contradictory as it may seem, in many cases the key to higher productivity lies in, of all things, slowing down. The key to this conundrum lies in the fact that human nature tends to regard most systems as roughly linear, and that variables that scale as cubic or fourth powers sneak up on us.

To examine this in detail, we'll have to "do the numbers", and our example will be a simple linear motor positioning stage executing forward and backward moves. To simplify the calculations, we'll assume that the stage uses a 'triangular" move profile, that is, it accelerates for half of the move, and when it hits top speed, it decelerates to a stop for the second half.

In addition, the stage will have a "duty cycle of 100%" meaning that there is no pause between moves. If we ignore friction in the guideways, which is usually a good simplification, the resulting system is quite easy to analyze, and can be characterized by a fairly small set of variables:

Travel	0.1 m
Moving mass	10 kg
Desired move time	0.1 s
Velocity limit	3 m/s
Linear motor K_m	10 N/√(W)
Motor continuous power	100 W

It is helpful to consider where the numerical requirements originated; the travel and moving mass are usually directly tied to the application, although any efforts to reduce them will be richly rewarded. The desired move time can be harder to pin down, but is often linked to some tie-in to units/hour, throughput, and economic justifications. The velocity limit is usually set by one of three factors; the linear encoder; the physical limits of the guideway bearings; or the available amplifier supply voltage.

Figure 1: Linear Motor Stage

The linear motor constant K_m deserves a closer look. This figure of merit, while key to linear motor performance, is often missing from supplier data sheets. This constant (for any given motor) relates the ratio between force output (the reason you bought the linear motor) and waste heat generation (an undesirable by-product, which sets the limit on performance).

While most advertising claims emphasize the linear motor K_f (the force Constant, with units of Newtons/Amp), this is largely a free variable, which can be changed readily by altering the coil wire gauge. The motor constant K_m, on the other hand, is defined only by the copper volume, packing efficiency, and magnetic field strength, and is the true linear motor quality factor.

K_m is also particularly helpful when it comes to calculating system performance. The value varies with the motor size and design, and typically ranges from 5-25 N/√(W) – larger numbers offer higher performance, are physically larger, and cost more because of the increased magnet and coil volume. In this hypothetical example, the linear motor K_m and power rating may have been chosen on the basis of cost, available space, informed motor sizing data, or a "wing and a prayer".

Using the values listed above, let's run a few numbers on the proposed application. First, we'll check the top speed against the stated limit, to see if we have any problems on that front. The top speed in a triangular move can be found using the formula:

$$V = (2 \cdot X) / t$$

Where X is the move size (0.1m), and t is the desired move time. This yields a top speed of (2 x 0.1)/0.1, or 2 m/s. this is comfortably within the system velocity limit of 3 m/s – so far so good.

The next quantity to calculate is the acceleration, which is equal to:

$$A = (4 \cdot X) / t^2$$

Where again, X is the move size and t is the move time. This yields an acceleration of $(4 \times 0.1)/0.1^2$, or 40 m/s^2. Because 9.8 m/s^2 is 1G, the acceleration in this application could also be referred to as just over 4Gs.

The above formulae are not the usual ones that spring to mind when making motion calculations, but they are time saving shortcuts for triangular moves, compared to splitting the move into two parts, calculating for half the move, and so on.

With the acceleration known, the force required comes from Newton's Second Law:

$$F = m \cdot A$$

Multiplying the mass (10 Kg) by the acceleration (40 m/s^2) gives a force of 400 $kg\text{-}m/s^2$ – that is, 400 Newtons. Note that the force is continuous in this application, it applies both during the acceleration phase and (with a direction reversal) during the deceleration phase.

The next step, to calculate the power that the linear motor must dissipate, is equally easy:

$$\text{Power} = (\text{Force} / K_m)^2$$

That is, to calculate the waste heat to be dissipated, simply divide the force by the motor K_m, and square the result. In this application, the power is $(400 / 10)^2$, or 1.6 kW. That's going to take a big (and expensive) motor!

Alternatively, if the existing 10 N/√W motor (with a continuous power rating of 100 W) and 100 ms move time were to be retained, the system would require a pause between each move of 1.5 s, for duty cycle of 1:16, or 6.3%.

In this example, we intentionally chose values that would require a high power linear motor. The method works of course, irrespective of the actual numbers used. Let's examine the way the system scales, and see how we got into this mess. The simple, yet key, insight is to note that, as shown before, the acceleration scales as the inverse square of the desired move time!

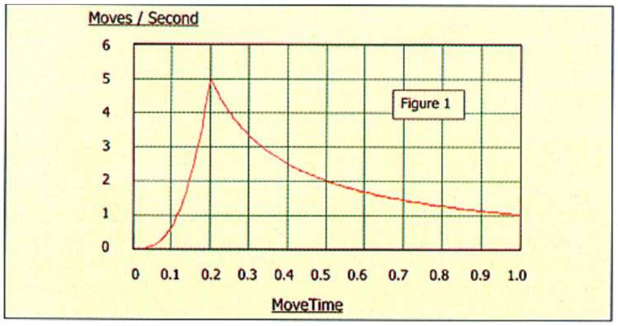
That is a pretty significant observation. In more explicit terms, the relationship between power, move distance, move time, mass, and K_m is as follows:

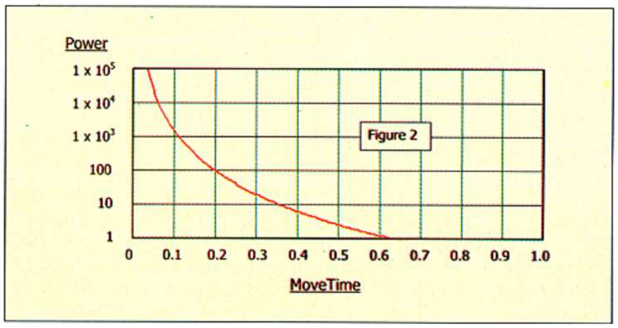
$$\text{Power} = (16 \cdot m^2 \cdot X^2) / K_m^2 \cdot t^4$$

We tend to view the world as a fairly linear place, but fourth power laws like this one tend to whack you before you see them coming. For example, attempting to make a move in half the time (a reasonable request, coming from a project manager or marketing type) will multiply the linear power requirement by a factor of 16! Similarly, relaxing the time by a factor of two will cut the power requirements by a factor of 16. In the above application, that would be just enough to let the original motor work properly. Because the move time turns out to be a very "sensitive parameter", a closer look at how it affects system performance is in order.

To assemble this last piece of the puzzle, let's explore what happens not just to system power, but to throughput (expressed in moves per second), as we vary the move time. Obviously, if we make the move time very long, the number of moves per second will be greatly reduced. For example, if we take a full second to make the above move, we can pretty much count on a throughput of one move per second. As we shorten the move time slowly, the number of "moves per second" (mps) rises inversely with the move time. If we were to graph this function, it would be the standard k/t curve, a hyperbola.

What is not glaringly obvious is that this smooth curve runs into serious trouble as soon as the power equals the power rating of the linear motor. At this point, our smooth curve encounters a “discontinuous inflection point”, and the mps begins an inverse cubic power crash towards zero. This is shown clearly in Fig. 1, where the two very different regions of the curve are readily apparent. In our example, the mps will be a paltry 0.6 at our chosen move time of 100ms. In Fig, 2, the rapidly escalating power requirements associated with short move times are also shown.





The cause of this discontinuity is fairly easy to understand. As soon as the continuous power rating of the linear motor coil is reached, any further reduction of the move time will require longer and longer “cooling off” spells to allow the excess power to dissipate. Despite the intuitive conclusion that this delay is cancelled by the reduction in the move time, the fourth power nature of the relationship stymies our common sense, and leads to vastly lower mps. The decrease in move time does help a little, reducing the fourth power term to a cubic one, but this is little consolation.

The most important observation to draw from this is that for any given system there exists an optimum move time, which may well be longer than one would expect. If you find yourself on the left-hand side of the cusp, then you will actually increase the mps, and hence productivity, by allowing the move time to increase. You truly can “slow down to speed up”!

In our example above, slowing down the move time from 100 to 200ms will increase the mps from 0.6 to 5.0, an 800% improvement! The number of variables that determine the optimum move time are actually quite few – namely, the moving mass, the move distance, and the linear motor Newtons / √W.

We have seen several examples of systems in which the move time was selected based on the goal of a certain number of parts per hour, or an overall cycle time was chosen, and then allocated among the machine operations with little or no attention to proper motor sizing. In some cases, the inevitable cooling-off pauses are inserted automatically by motor coil thermal sensors or an amplifier I^2T function, and the only obvious signs of a problem are that the system is sluggish and “not meeting spec”.

In other cases, technicians add in delays on an empirical basis until the linear motor coils “stop burning out”. For our simple example, it would be fairly easy to size the motor properly to operate at the desired throughput. In real-world systems, however, with considerably more complex motion sequences, it is a rare designer who prepares a detailed power model of the complete system. Additional constraints can include the cost of linear motors, or even the amount of space available to fit the motor.

The concepts in this article also apply to rotary motor systems, although in this case, the moment of inertia of the motor rotor and leadscrew often predominate over the reflected payload mass.

In conclusion, it is clear that the inverse fourth power relationship between power and move time requires us to consider cycle times and motor sizing carefully when designing motion systems. The square law between moving mass and power similarly requires effort to minimize the moving mass.

Casual comments such as “We need to slice 100 ms off the cycle time” need to be evaluated in light of the potentially expensive underlying physics. Finally, the fact that the productivity curve has such a pronounced maximum can help us to design better systems that operate at peak performance.