Survey of Methods of Combining Velocity Profiles with Position control

Petter Karlsson
Mälardalen University
P.O. Box 883
72123 Västerås, Sweden
pkn09001@student.mdh.se

ABSTRACT

In many applications where some kind of motion is performed, for example in robotics, it is of high importance to be able to control how a certain angular motor position is reached. To this end, motion profiles are used. These profiles often define how the velocity varies during the traversal from the starting position to the desired position. It is relatively easy to let a controller act on the set velocity and a velocity feedback to achieve decent velocity following, but not as easy to also make sure that the correct position is kept along the route as well as in the end.

In this paper, four approaches to manage position control alongside velocity profiles are presented and discussed. The first approach is based on continuous velocity control in an ideal environment using a PID controller. If no disturbances are present and the velocity measurement is *very* exact, this approach could work, but it is hardly worth the effort. This approach can be improved by switching to a distance-based control scheme near the end. Another approach is to use position control by incrementally adding to the set position. The last approach discussed is a cascaded P-PI controller where both velocity and position is considered.

Keywords

Motion control, position control

1. INTRODUCTION

In motion control applications, velocity profiles are used in order to achieve a controlled acceleration and deceleration with respect to desired velocity and position. In contrast to pure PID (proportional, integral and derivative) position control, this gives the system a much higher level of determinism. By using velocity profiles, many aspects of the motion can be controlled, such as traversal time, velocity, acceleration/deceleration and jerk (derivative of acceleration). In short, motion profiles are used when it is important where you get but also how you get there.

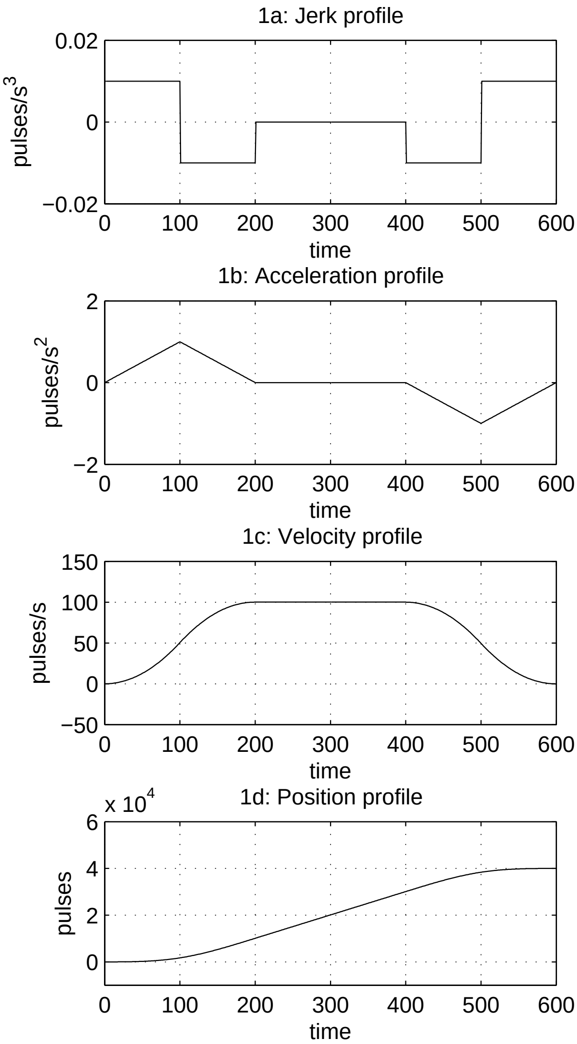

Figure 1: A step shaped jerk profile (1a) and its derivatives, giving continuous acceleration (1b) as well as smooth velocity (1c) and position(1d) profiles.

Depending on what characteristics are needed for the motion, different profiles can be used. The basic profile is the *trapezoidal velocity profile*, where the first segment of the profile is a constant acceleration phase. This phase is followed by a constant velocity phase and the profile ends with a constant deceleration phase ending at the set position. Problems could arise using this approach, because the steps in acceleration will produce impulses in jerk. To remedy this, an S-curve profile can be used. In this profile, the phases of constant acceleration/deceleration are replaced by linearly increasing and decreasing acceleration/deceleration, see Figure 1. This approach will instead produce finite steps of jerk that are much more manageable.

One of the big problems of combining velocity profiles with position control of any system is that of generating a smooth deceleration, stopping at the set position without creep or abrupt stop. If the motion is decelerated too early, forcing a slower motion while reaching for the set position, the throughput of a system can be lowered. An abrupt stop, on the other hand, may cause damage to equipment or have a negative impact on the comfort of, for example, an elevator ride [5]. Both of these problems can give rise to synchronization issues in multi-axis environments such as CNC machinery or robot drives and manipulators.

The purpose of this paper is to survey methods of reaching the set position, using a velocity profile, while minimizing or eliminating the problems of creep and abrupt stop. This paper will not go into details of tuning systems based on the different methods or how to handle disturbance rejection. The methods presented will be evaluated on a proof-of-concept basis using MATLAB/Simulink simulations to determine whether the method is suitable for reaching a set position using a velocity profile.

In section 1.1, the methods involved in producing the results in this paper are presented. It is followed by a short description of the terms and abbreviations used throughout the paper in section 1.2. The different control approaches surveyed in this paper are described in section 2 and the paper ends with the conclusions in section 3.

1.1 Method

The suitability of each approach will be determined by simulation using MATLAB/Simulink. In the simulations I will use the transfer function of a hypothetical motor to test the different approaches. Values for the different gains in the controller, as well as sample intervals where applicable, are chosen arbitrarily to represent a stable system. No disturbances or static friction is modeled, as that would not contribute significantly to the results of this paper. The motor model that is used in the simulations, represented as "Motor" in the controller models, has the following characteristics:

- moment of inertia of the rotor (J) = 0.01 $kg.m^2/s^2$
- damping ratio of the mechanical system (b) = 0.1 Nms
- electromotive force constant (K=K_e=K_t) = 0.01 Nm/A
- electric resistance (R) = 1 Ω
- electric inductance (L) = 0.5 H

giving the approximate Laplace transfer function

$$\frac{\omega}{V} = \frac{K}{(Js+b)(Ls+R)+K^2}$$

where ω is the angular velocity and V is the input voltage [4].

In all the controller models the voltage saturates at $\pm 100V$, meaning that the maximum absolute value of the voltage fed to the motor is $100V$. This is done to ensure that the simulations are somewhat realistic.

For purposes of simplicity, the profile used in all simulations will be a trapezoidal velocity profile.

1.2 Terms

p : Actual (measured) position.
p^* : Set position (desired position).
v : Actual (measured) velocity.
v^* : Set velocity (desired velocity).
S : Distance between p and p^*.
P controller : Proportional controller. The output is proportional to the error.
PI controller : Proportional-integral controller. The output is proportional to the error and the integral of the error.
PID controller : Proportional-integral-derivative controller. The output is proportional to the error as well as the integral and derivative of the error.

2. CONTROL APPROACHES

2.1 Velocity Control

One way of using velocity profiles is to use a PID controller to control the velocity of the motor based on time, see Figure 2. The integral property of the controller will make sure that the set position to be expected at the end of the profile is eventually approached. This is of course based on the (not very plausible) assumption that exact, continuous speed measurements can be performed on the system.

One way to prove this is to note that the angular position p of the motor equals the integral of velocity over time. The integral term of the PID controller, denoted I, will accumulate all of the error between set velocity v^* and measured velocity v over time (see equation 1), thus expressing the remaining distance S to the end point multiplied by a factor k. Now we can conclude that there is a linear relation between the integral term and the remaining distance to set position p^* (see equation 2).

$$I = \int k[v^*(t) - v(t)]dt = k[p^*(t) - p(t)] \qquad (1)$$

$$S = p^*(t) - p(t) = \frac{I}{k} \qquad (2)$$

There are a few things to note here, though, rendering this approach less useful in a real system. In the model of figure 2, friction is not modeled. As can be seen in figure 4,

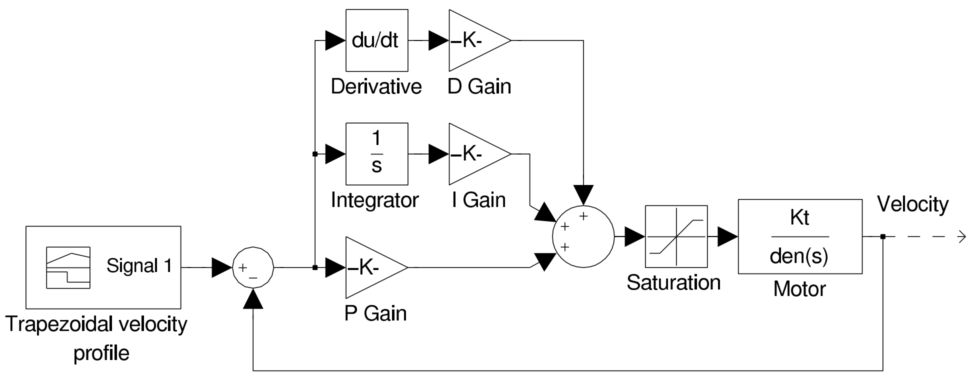

Figure 2: A PID controller based on continuous and exact velocity measurements.

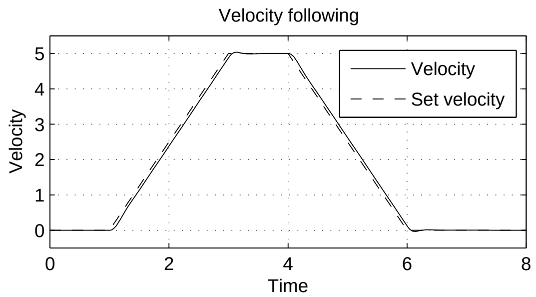

Figure 3: Velocity following of (ideal) PID velocity controller. The graph shows how the measured velocity follows the velocity profile.

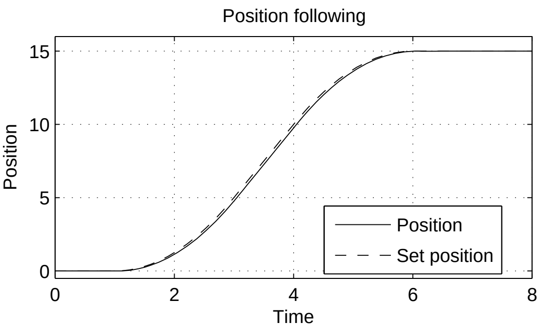

Figure 4: Position following of (ideal) PID velocity controller. The graph shows how the measured position follows the inferred position profile.

the integral term pushes the motor towards the set position after the profile has been traversed. In reality, the friction of the system would probably be too high for the output of the integral term to overcome it, leading to an early stop in this example. Further, most PID controllers are equipped with an anti-windup system, stopping the integral term from growing uncontrollably. Once the anti-windup threshold has been reached, the integral term won't represent the missing distance to the set position any more.

There are, of course, other controllers that can be used with this method but in order to be reasonably sure that the motor will end up close to the set position, an integral term is necessary. The anti-windup threshold of this term would have to be set rather high in order to guarantee that it will not usually be reached and the velocity measurement would have to be continuous and extremely exact. These requirements are very hard to fulfill and this approach is therefore *very* impractical.

2.2 Velocity control with near-end position control

One way of overcoming the requirements of exact speed measurements and high anti-windup threshold of the previous method is to combine it with a distance-based velocity control when the set position is approached [5], as shown in figure 5.

In this approach, the deceleration phase is separated into sub-phases. When a constant deceleration is achieved, the set velocity v will instead be expressed as

$$v^* = \sqrt{2a_{max}(S - S_{off})}$$

, where a_{max} is the maximum acceleration allowed, S is the distance remaining to the set position and S_{off} is an offset distance that is used to guarantee continuity in velocity at the time where velocity pattern is switched. When the target position is approached closely, the pattern is once again switched to

$$v^* = k * S^x$$

where k and x are chosen to guarantee acceptable settling time and continuity of acceleration and velocity. This method will however lead to a longer settling time and requires well-tuned parameters in order to create a continuous velocity

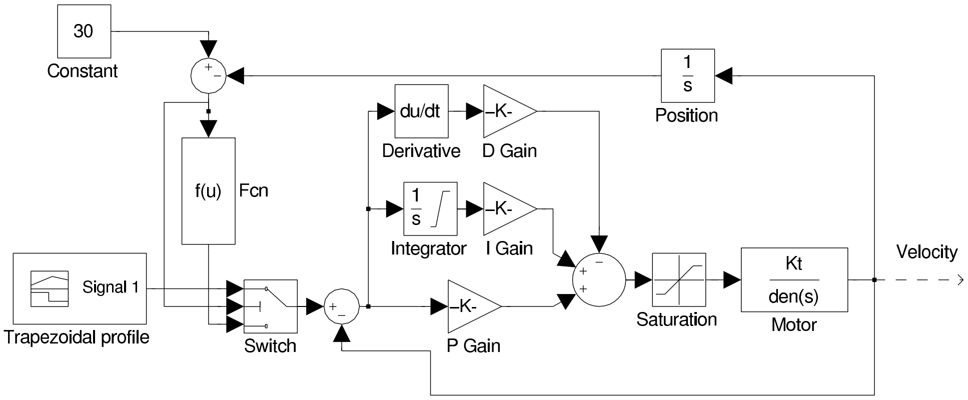

Figure 5: Velocity follower with near-end position-based velocity.

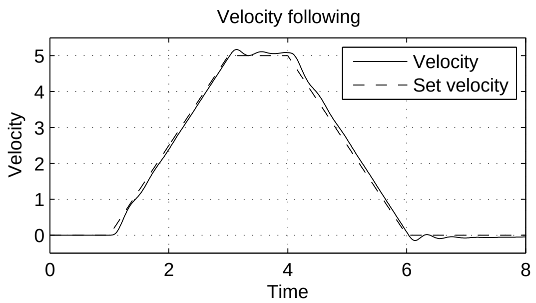

Figure 7: Velocity following of PID controller based on discretized position error. The graph shows how the measured velocity follows the velocity profile.

profile. [5] instead proposes a system based on a cascaded controller, similar to the setup which I will examine more closely in section 2.4.

2.3 Incremental position control

A method that is used in various motion control chips is a kind of incremental position control [1]. Using a velocity profile as a source, the expected position at every sample interval is calculated. The control signal applied is then based on the difference between the expected position at the next time step and the current position. This method will require a discrete system and the velocity is then preferably expressed in the number of sensor pulses per control loop period.

The example setup I have used is based on zero order hold circuits for discretization and uses a PID controller to minimize the position error according to figure 6, giving the position following of figure 8.

One way to avoid integration of the velocity feedback is to use a position sensor instead. This could be an absolute encoder or, which is very common, an incremental quadrature encoder [3].

This method makes the system independent of any velocity measurement, with the possible drawback of decreased precision in velocity which can be hinted in figure 7. In this

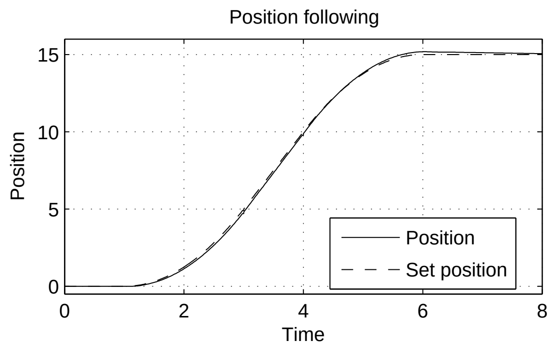

Figure 8: Position following of PID controller based on discretized position error. The graph shows how the measured position follows the inferred position profile.

control system, the integrator works with the error position, so as long as $S > 0$ the integral term will keep growing. This will guarantee that as long as the system is stable, the set position p will eventually be reached. One of the strong advantages of this method is its simplicity; the only input needed is a position feedback.

2.4 Cascaded P-PI control

A setup as depicted in figure 9 will make sure that the velocity profile, as well as the position profile that can be inferred from the velocity profile, is followed. The basic idea is that the inner PI loop controls the velocity and that there is an additional outer loop consisting of the position error multiplied by a gain. The PI loop will make sure that the velocity profile is followed (see figure 10) and the additional P term will compensate for any mismatch of position that occurs (see figure 11).The integral property of the PI loop makes sure that any residual position error will eventually be eliminated.

In a discrete implementation of this controller, the current velocity can quite easily be measured by differentiation of the position feedback. This makes it similar to the incremental PID controller of section 2.3 but with the added functionality of velocity control.

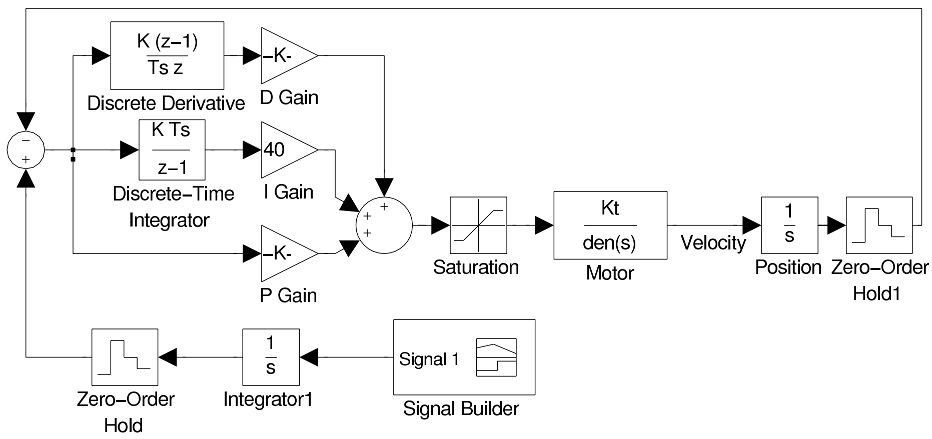

Figure 6: Velocity control based on discretized position error.

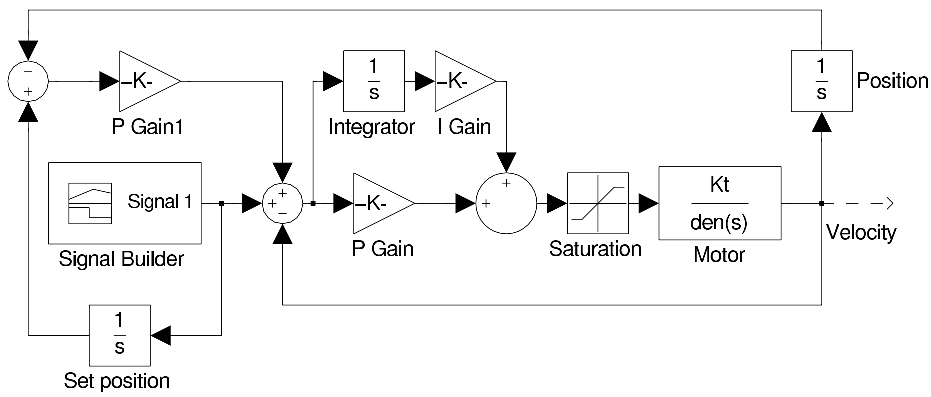

Figure 9: Velocity control using a cascaded P-PI controller.

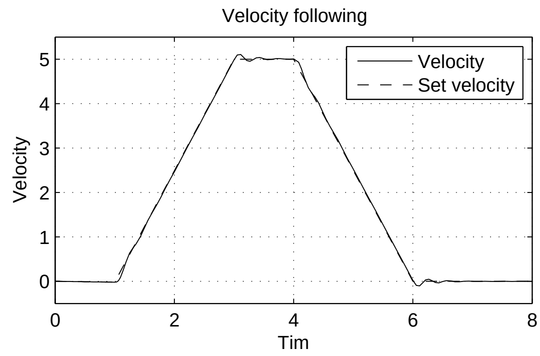

Figure 10: Velocity following of a cascaded P-PI controller. The graph shows how the measured velocity follows the velocity profile.

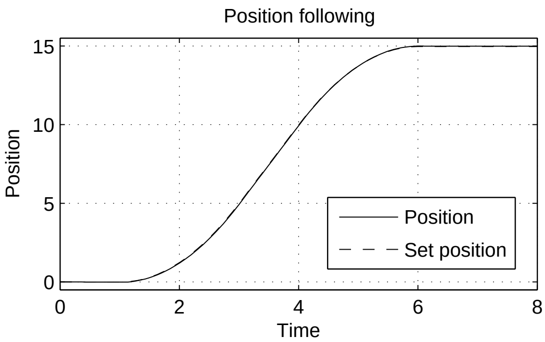

Figure 11: Position following of a cascaded P-PI controller. The graph shows how the measured position follows the inferred position profile.

This approach is very powerful as good following of both velocity and position profiles can be achieved. Concerning disturbance rejection, there are improvements over this method available, but they share the same basic concepts. One of these is PI+, which is described in [2].

3. CONCLUSIONS

We have now seen multiple ways of making sure that a set position is reached, following a velocity profile.

Using only velocity based control will not guarantee that the set position is reached under all conditions. The two factors that make this approach nearly unusable is that a continuous and very exact speed measurement is necessary and that the anti-windup mechanism of the integral term can destroy the linear connection between the terms value and S.

The above situation can be solved by switching to a position-based control method at a certain time, position, velocity or acceleration. This approach has the downsides that the landing time will be increased and that the constants of the equations determining where to switch control method are hard to define.

A totally different approach, that is commonly used in motion control chips, is to control the position and velocity by adding different values to the set position at regular intervals. The value to add is calculated from the input velocity profile. When using this method, no dedicated velocity feedback mechanism is needed and the complexity of the controller is low. On the other hand, no direct control of the velocity is achieved.

To control velocity and position simultaneously, a cascaded P-PI controller can be used. It consists of an inner loop, controlling velocity, and an outer loop adjusting for any position mismatch along the way.

This paper deals only with the problem of making sure that position control is achieved alongside a velocity profile. A continuation of this work could be an investigation of the disturbance rejection properties of these and other methods.

4. REFERENCES

[1] J. U. Cho and J. W. Jeon. A motion-control chip to generate velocity profiles of desired characteristics. *ETRI Journal*, 27(5):563–568, October 2005.

[2] G. Ellis and R. D. Lorenz. Comparison of motion control loops for industrial applications. *IEEE IAS Annual Meeting*, 4:2599–2605, October 1999.

[3] R. O. Jr. Incremental optical encoder system forabsolute position measurement. *US. patent*, 4,079,251, August 1976.

[4] K. Ohishi, K. Ohnishi, K. Miyachi, and K. Tamaki. Microprocessor-based robust control of a dc servo motor. *IEEE Control Systems Magazine*, 6(5):30–36, October 1986.

[5] H.-M. Ryu and S.-K. Sul. Position control for direct landing of elevator using time-based position pattern generation. *IEEE IAS Annual Meeting*, 1:644–649, October 2002.