10.24425/acs.2020.132588

Archives of Control Sciences
Volume 30(LXVI), 2020
No. 1, *pages 123–138*

Motion control algorithm and tuning rules for mechanical devices with low sampling-rate electronics

JORGE JUAN GIL and IÑAKI DÍAZ

Controlling mechanical systems with position and velocity cascade loops is one of the most effective methods to operate this type of systems. However, when using low-rate sampling electronics, the implementation is not trivial and the resulting performance can be poor. This paper proposes effective tuning rules that only require establishing the bandwidth of the inner velocity loop and an estimation of the inertia of the mechanism. Since discrete-time mechatronic systems can also exhibit unstable behavior, several stability conditions are also derived. By using the proposed methodology, a P-PI control algorithm is developed for a desktop haptic device, obtaining good experimental performance with low sampling-rate electronics.

Key words: motion control, cascade control loops, discrete-time systems, stability, mechatronic systems

1. Introduction

Motion control algorithms usually consist of two cascade loops where the inner loop controls the velocity of the mechanism and the outer loop the position. This control scheme is widely used in industry and more specifically in computer numerical control (CNC) systems. The automatic tuning of cascade controllers has been explored in the scientific literature [1–3], and useful guidelines are available in industry [4]. Although each loop could have a complete PID controller, a P-PI scheme, that is, a proportional controller for the position loop and a proportional-integral controller for the velocity loop, is commonly used in robotics [5] and machine tool [6] as the simplest strategy to manage independently each joint.

The Authors are with TECNUN, University of Navarra, and CEIT, San Sebastián, Spain.

Received 10.08.2019.

The main challenge in machine tools is to increase the position bandwidth of the system, which is limited by the first natural frequency of the mechanical elements of the transmission [7]. To overcome this limitation, several solutions have been proposed such as the inclusion of feed-forward paths based on acceleration and velocity [8, 9]. This solution is also suitable to compensate non-linear effects due to friction or backlash [10].

Industrial applications require high position tracking precision and very fast response, and thus, the applied control techniques become more complex. However, some robotic applications do not require such demanding specifications and are driven by cost-effective electronics [11]. These electronic systems usually face limitations due to the relatively slow sampling rate they can achieve, compromising both the performance of the robotic system and its overall stability. This paper analyses thoroughly this limitation and proposes a methodology with effective tuning rules for the P-PI control algorithm that only requires an estimation of the inertia of the mechanism and the desired bandwidth for the internal velocity loop. These recommendations take into account the limitations imposed by cost-effective electronic devices, and stability limits are derived to show achievable safe-boundaries with low sampling rate features. Finally, the proposed methodology is applied to a desktop haptic mechanism showing that good performance can be achieved with this method.

The paper is organized as follows: Section 2 describes the control block diagram of a robotic mechanism controlled by a P-PI cascade control algorithm. Section 3 analyses the gain selection for the proposed control architecture and Section 4 particularizes the strategy for slow sampling rate control. Section 5 derives stability implications for the mechatronic system and Section 6 shows the real implementation of the proposed method and the obtained results. Finally, Section 7 gathers conclusions.

2. System description

Fig. 1 shows the block diagram of a robotic mechanism and the P-PI cascade controller if both position and velocity signals are measured by analog sensors.

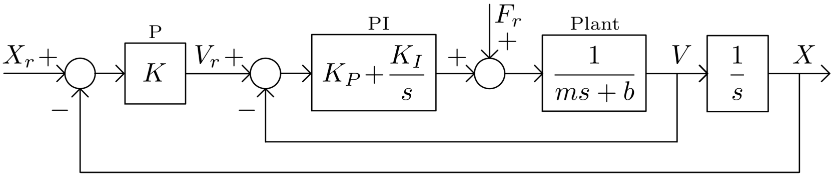

Figure 1: Continuous-time description of the mechanism and the P-PI cascade controller

X is the position of the system in Laplace domain, X_r is the setpoint and F_r represents any external disturbance. The behavior of the mechanism is modeled by a linear transfer function with inertia m and viscous damping b. The output position of the system with respect the setpoint is:

$$\frac{X}{X_r} = \frac{KK_P s + KK_I}{ms^3 + (b + K_P)s^2 + (K_I + KK_P)s + KK_I} \,. \tag{1}$$

The output position of the system with respect the disturbance is:

$$\frac{X}{F_r} = \frac{s}{ms^3 + (b + K_P)s^2 + (K_I + KK_P)s + KK_I} \,. \tag{2}$$

Thus, using a proportional (P) controller for the position loop and a proportional-integral (PI) controller for the velocity loop, the resulting system is of third order.

3. Gain selection based on system model

The inner PI controller forces the system to achieve the velocity reference commanded by the position controller while also rejects the disturbances (e.g., the static friction or external forces). Even in absence of disturbances, it is necessary a PI controller to get zero error in this inner loop. And with this controller, the velocity loop is a second-order system whose output as a function of the reference is:

$$\frac{V}{V_r} = \frac{K_P s + K_I}{ms^2 + (b + K_P)s + K_I} \,. \tag{3}$$

Parameters K_P and K_I could be selected using sensitivity criteria and relative stability [12]. However, with a suitable choice of K_P and K_I, it is possible to obtain any transient characteristics in the inner loop, in terms of natural frequency and damping factor. However, it is necessary to know the inertia m and the viscous damping b of the mechanism. If there are no particular restrictions, we propose to set the poles of transfer function (3) critically damped ($\zeta = 1$) and leave the natural frequency ω_n to the decision of the engineer. Thus, using the following parameters

$$K_P = 2m\omega_n - b, \tag{4}$$

$$K_I = m\omega_n^2, \tag{5}$$

the dynamics of the velocity loop is:

$$\frac{V}{V_r} = \frac{2\omega_n \left(s + \dfrac{\omega_n}{2}\right)}{(s + \omega_n)^2} \,. \tag{6}$$

Besides the two identical real poles, there is a dominant zero, and thus, velocity response always exhibits the same overshoot (around 13% for any ω_n selected). Natural frequency ω_n imposes the bandwidth of the velocity loop, and it is usually set between 60 rad/s and 200 rad/s. Within this range, and for most desktop mechanisms, the effect of viscous damping b on (4) is marginal, and thus, it can be neglected:

$$K_P \approx 2m\omega_n\,. \tag{7}$$

Disturbance rejection (2) depends on both inner and outer controllers, but once the inner PI is chosen depending on ω_n, the response to external forces only depends on the selection of gain K. The root locus of the system is used to choose this gain. The poles of the system are the roots of the following characteristic equation:

$$ms^3 + (b + K_P)s^2 + (K_I + KK_P)s + KK_I = 0. \tag{8}$$

Using parameters (4) and (5), it yields:

$$ms^3 + 2m\omega_n s^2 + (m\omega_n^2 + 2mK\omega_n - Kb)s + Km\omega_n^2 = 0. \tag{9}$$

Neglecting viscous damping b, the characteristic equation does not depend on the inertia:

$$s^3 + 2\omega_n s^2 + (\omega_n^2 + 2K\omega_n)s + K\omega_n^2 = 0, \tag{10}$$

$$1 + K\frac{2\omega_n s + \omega_n^2}{s(s^2 + 2\omega_n s + \omega_n^2)} = 0, \tag{11}$$

$$1 + 2\,\omega_n K\frac{s + \dfrac{\omega_n}{2}}{s(s + \omega_n)^2} = 0. \tag{12}$$

Thus, the positioning of the three poles of characteristic equation (9) changes as a function of gain K and follows the paths of the root locus depicted in Fig. 2.

There is a trade-off between the time constant of the dominant real pole, that moves from $s = 0$ to $s = -0.5\,\omega_n$, and the pair of complex poles, whose damping factor can change from 1 to 0. Thus, trying to shorten the transient response of the disturbance rejection, undamped oscillations could arise. We propose to choose the gain that places the poles approximately in the red crosses depicted in Fig. 2. Applying the magnitude condition in the real pole located in $s = -0.4\,\omega_n$, it yields that:

$$2\,\omega_n K = \frac{0.6\,\omega_n\,0.6\,\omega_n\,0.4\,\omega_n}{0.1\,\omega_n}, \tag{13}$$

$$K = 0.72\,\omega_n\,. \tag{14}$$

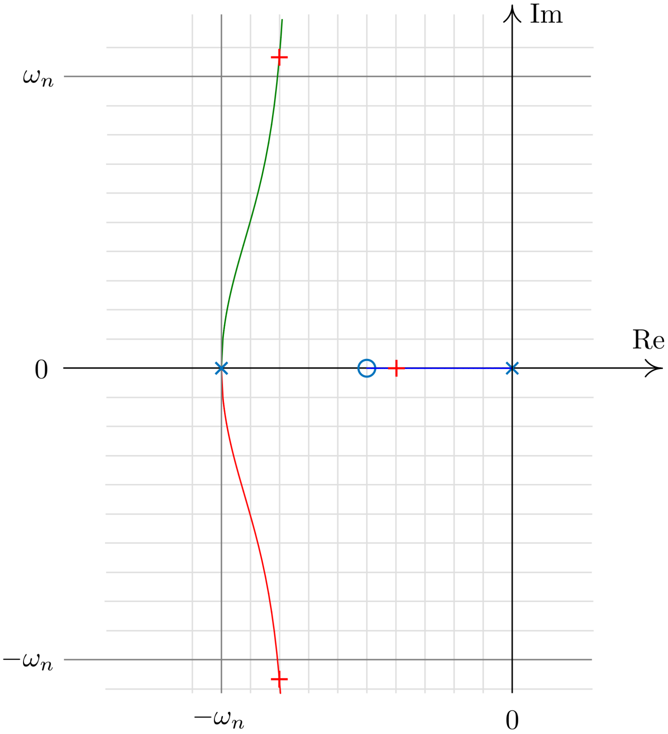

Figure 2: Root locus as a function of gain K

For the proposed gain (14), the damping factor of the pair of complex poles is $\zeta \approx 0.6$. We do not recommend higher gains because the position of the real pole barely changes, while the complex poles become poorly damped. To show the disturbance rejection around the proposed gain, some Matlab simulations are performed with arbitrary values of inertia and damping: $m = 10$ kg and $b = 1$ Ns/m (Fig. 3). The position setpoint is $X_r = 0$, the external disturbance is a step input of magnitude $F_r = 100$ N at $t = 0$ s. PI controller is tuned using (4) and (5) with $\omega_n = 60$ rad/s.

It is worth noting that for the selected bandwidth, a force similar to the weight of the mechanism produces a deviation outside the reference position of approximately 1 mm. Both the magnitude of this deviation and the transient duration can be shortened by increasing the bandwidth.

As a conclusion to this section, the values proposed for the gains of the controllers, (5), (7) and (14), only depend on the velocity bandwidth ω_n and the inertia of the device m. This inertia should be known or at least estimated. The selection of ω_n is not straightforward because the continuous-time model (Fig. 1) does not present any stability limit, and thus, in principle, the higher bandwidth the better performance.

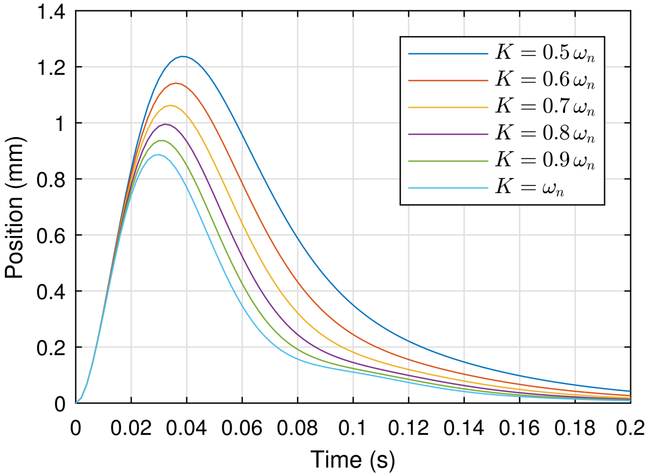

Figure 3: System response to an external disturbance force $F_r = 100$ N

4. Tuning methodology under sampling rate limitations

The use of the proposed gains and, in particular, the selection of ω_n, require a more detailed analysis if the electronics cannot reach a fast sampling rate. In this work, the mechanism used as testbed is the haptic interface PHANToM 1.0 (Fig. 4). The inertia of this device is $m \approx 70$ g. Viscous damping b is supposed to be negligible. The electronics is described in detail in [11], and it includes an Arduino MEGA board. With this electronics the sampling period cannot be smaller than $T = 2$ ms, which is 10 times larger than typical values in industrial PLCs [4], and thus, it is difficult to achieve a high bandwidth for the velocity loop.

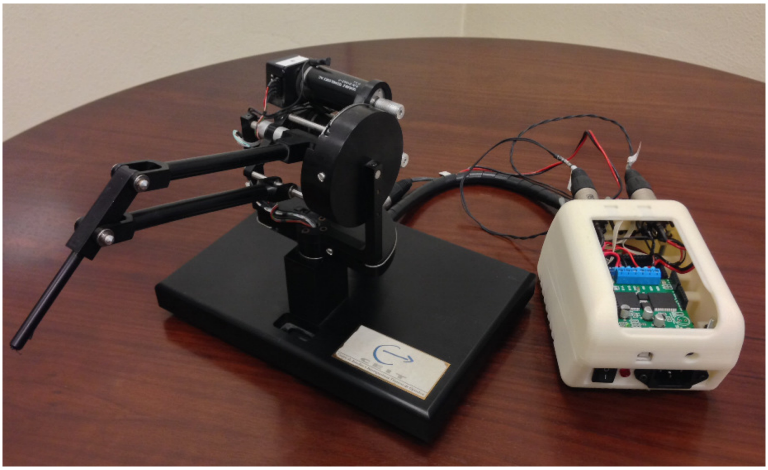

Figure 4: Device and electronics used for the experiments

The discrete-time description of the system (Fig. 5) is used to analyze the limitations imposed by the sampling rate. This model includes three new elements: the sample-and-hold process, the estimation of the velocity using the backwards difference, and a first-order digital filter to reduce the quantization noise.

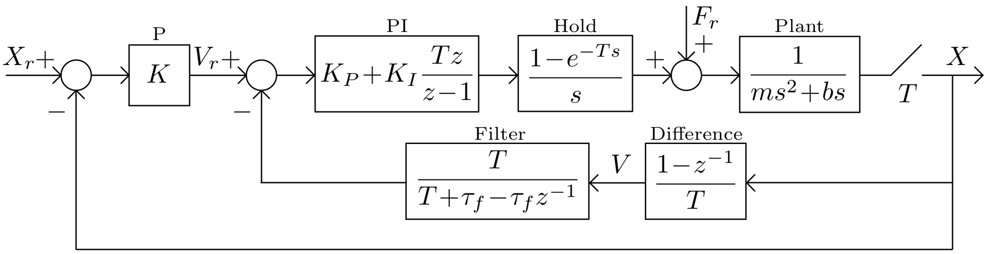

Figure 5: Discrete-time description of the P-PI cascade controller

The filter is the digital version of the following continuous-time transfer function,

$$G_f(s) = \frac{1}{\tau_f s + 1}, \tag{15}$$

where τ_f is the time constant and the cut-off frequency of the filter is $1/\tau_f$ rad/s. The block diagram also includes the digital version of the PI controller.

The discrete-time system behaves similar to the ideal continuous-time description (Fig. 1) if the dynamics of the filter is much faster than the dynamics of the velocity loop, which also must be much faster than the sampling rate:

$$\omega_n \ll \frac{1}{\tau_f} \ll \frac{\pi}{T}. \tag{16}$$

The three frequencies in (16) are given in rad/s. To fulfill this inequation, each frequency should be at least five times greater that the former [13, 14]. Since the sampling period is relatively long, there is small room to select ω_n and τ_f. We have chosen for our system the following values: $\omega_n = 50$ rad/s and $\tau_f = 4$ ms. And the selected values achieve these relations:

$$\omega_n = 50 \text{ rad/s} \quad \xrightarrow{\times 5} \quad \frac{1}{\tau_f} = 250 \text{ rad/s} \quad \xrightarrow{\times 6.3} \quad \frac{\pi}{T} = 1570.8 \text{ rad/s}. \tag{17}$$

5. Stability analysis

An important requirement before the experimental implementation is to guarantee the stability of the system. This analysis is carried out by using the discrete-time description of the system (Fig. 5). The characteristic equation without velocity filtering ($\tau_f = 0$) is a fourth-order polynomial in z (18).

$$\begin{aligned}\left[\left(e^{-\frac{bT}{m}} - 1 + \frac{bT}{m}\right) z + 1 - \left(1 + \frac{bT}{m}\right) e^{-\frac{bT}{m}}\right] \cdot \\ \cdot \left[(K_P + K_I T)z - K_P\right]\left[(KT+1)z - 1\right] + \\ + z(z-1)^2 \left(z - e^{-\frac{bT}{m}}\right) \frac{Tb^2}{m} = 0. \end{aligned} \qquad (18)$$

The stability conditions are derived using the Routh-Hurwitz criterion after applying a bilinear transformation. Substituting the proposed values for K, K_P and K_I using (4), (5) and (14), and the physical estimated values of the device, $m \approx 0.07$ kg and $b \approx 0.6$ Ns/m, we have computed in Matlab the stable values of the two remaining parameters: ω_n and T. However, the exact stability boundary (Fig. 6) barely changes with the physical parameters of the inertia and the viscous damping.

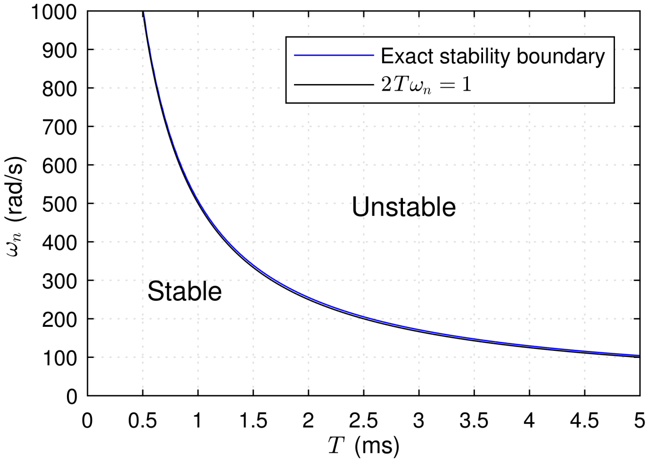

Figure 6: Stability boundary without velocity filtering

Interestingly, the critical bandwidth can be approximated by expression (19). Thus, known the sampling period of the system, this formula indicates the maximum value for ω_n, from the stability point of view, if no filtering is introduced.

$$\omega_n \approx \frac{1}{2T} \,. \qquad (19)$$

This stability limit is further reduced if the estimation of the velocity is filtered. In our case, the inclusion of the first-order digital filter leads to a different characteristic equation (20), but it is still a forth-order polynomial.

$$\begin{aligned}\left[\left(e^{-\frac{bT}{m}} - 1 + \frac{bT}{m}\right) z + 1 - \left(1 + \frac{bT}{m}\right) e^{-\frac{bT}{m}}\right] \cdot \\ \cdot \left[(K_P + K_I T)z - K_P\right] \left[(KT + K\tau_f + 1)z - K\tau_f - 1\right] + \\ + \left[(T + \tau_f)z - \tau_f\right] (z-1)^2 \left(z - e^{-\frac{bT}{m}}\right) \frac{b^2}{m} = 0. \end{aligned} \tag{20}$$

Fig. 7 shows the exact stability boundaries computed in Matlab for the sampling periods $T = [1, 2, 3, 4, 5]$ ms. The longer time constant τ_f, the smaller bandwidth ω_n can be implemented. Again, stability boundaries barely change with the inertia and the viscous damping. A good approximation for all the stability boundaries is:

$$\omega_n \approx \frac{1}{2T + \tau_f}\,. \tag{21}$$

Although stability condition (21) is useful, it only applies to the P-PI controller that uses the parameters proposed in Section 3, and the tuning rule is (16). Plus sign (**+**) in Fig. 7 shows the parameters selected for the experimental setup ($\omega_n = 50$ rad/s and $\tau_f = 4$ ms) and how they fall within the stable region for $T = 2$ ms.

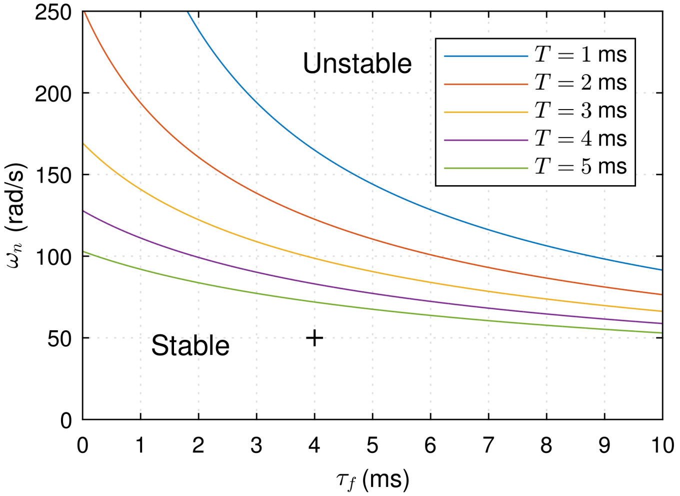

Figure 7: Stability boundaries with first-order digital filtering of the velocity

6. Experimental response to a smooth position setpoint

This section analyzes experimentally the P-PI controller described in Fig. 5 using the proposed bandwidth ω_n = 50 rad/s and the inertia of the device m = 0.07 kg. Substituting these values in (5), (7) and (14), the gains are K = 35 s^{-1}, K_P = 7 Ns/m and K_I = 175 N/m. The time constant of the digital filter is τ_f = 4 ms. For the position setpoint, a S-shaped smooth curve [15] is used instead of a step input. These smooth trajectories limit the velocity and the acceleration of the input reference in order to prevent the excitation of vibrational modes of the mechanism, and they are widely used in industrial applications.

To generate the smooth trajectory, some maximum values for the velocity, $v_{\max}$, the acceleration, $a_{\max}$ and the jerk, $j_{\max}$ should be selected. If all the maximum values are reached, seven consecutive sections arise from the initial to the final position (Fig. 8). The final value for the position, $x_{\max}$, could be

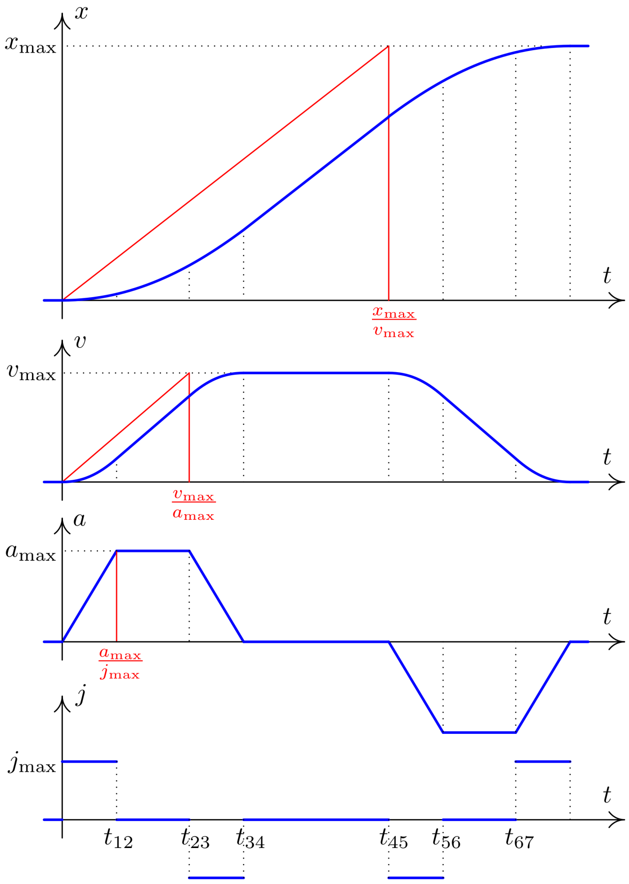

Figure 8: Position, velocity, acceleration and jerk profiles of a smooth trajectory with 7 sections

considered as another restriction for the generation of the smooth trajectory. Three main relations are shown in Fig. 8:

$$t_{12} = \frac{a_{\max}}{j_{\max}}, \tag{22}$$

$$t_{23} = \frac{v_{\max}}{a_{\max}}, \tag{23}$$

$$t_{45} = \frac{x_{\max}}{v_{\max}}. \tag{24}$$

All the transitions between sections can be obtained using these three definitions. From the initial position up to the final position reference $x_{\max}$, it takes the sum of the three main relations: $t_{12} + t_{23} + t_{45}$. In Appendix A, we have included the Matlab code that generates a smooth trajectory after introducing the maximum values of position, velocity, acceleration and jerk. This code generates the smooth trajectory even if the velocity or the acceleration profiles cannot reach their respective maximum values.

The maximum values should be defined depending on the experimental application. In our case, the selected displacement, $x_{\max} = 0.1$ m, is compatible with the workspace of the mechanism. The rest of the parameters, $v_{\max} = 0.5$ m/s, $a_{\max} = 5$ m/s^2, and $j_{\max} = 55.5$ m/s^3, are selected to get a fast response and also to ensure that all the maximum values of the profiles are reached during short intervals of time.

The smooth trajectory (blue line in Fig. 9) is computed in the `setup()` function of the Arduino sketch, using a C version of the code proposed in Appendix A. The P-PI controller is programmed in a dedicated function, and this subroutine is executed at the sampling rate using internal timer interruptions.

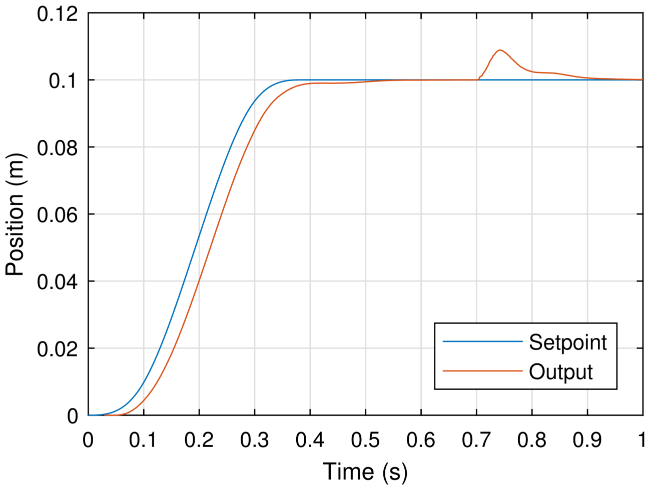

Figure 9: Experimental system response

Fig. 9 shows the experimental response of the device. At $t = 0.7$ s, a disturbance of 3.5 N (five times the equivalent weight of the device) is introduced. The performance of the controlled system is quite satisfactory. The experimental response also shows an observable outcome of gain $K = 35\ \text{s}^{-1}$. This gain is directly related with the apparent delay between the smooth reference and the actual position of the mechanism, which is the inverse of K, that is 0.028 s.

Substituting the selected parameters in characteristic equation (20), the four poles of the system are:

$$\begin{cases} z_1 = 0.9602 \\ z_2 = 0.8714 + 0.0636j \\ z_3 = 0.8714 - 0.0636j \\ z_4 = 0.1538 \end{cases} \xrightarrow{s=\frac{\ln z}{T}} \begin{cases} s_1 = -20.32 \\ s_2 = -67.52 + 36.45j \\ s_3 = -67.52 - 36.45j \\ s_4 = -936.13 \end{cases} . \tag{25}$$

As expected, the selected bandwidth ω_n for the inner velocity loop imposes a dominant real pole located at $s_1 \approx -0.4\,\omega_n$. The other real pole s_4 arises due to the velocity estimation and filtering, and that is why was not included in the root locus of Fig. 2. Nevertheless, if condition (16) is held, the influence of this pole on the disturbance rejection performance is negligible.

The pair of complex poles s_2 and s_3 have a damping factor of 0.88 which differs from the expected 0.6. This is a consequence of time discretization and the presence of the new real pole, that modifies the branches of the root locus associated to these complex poles.

Fig. 10 shows the velocity signal directly estimated by the backwards difference and after applying the digital filter. The device reaches the maximum

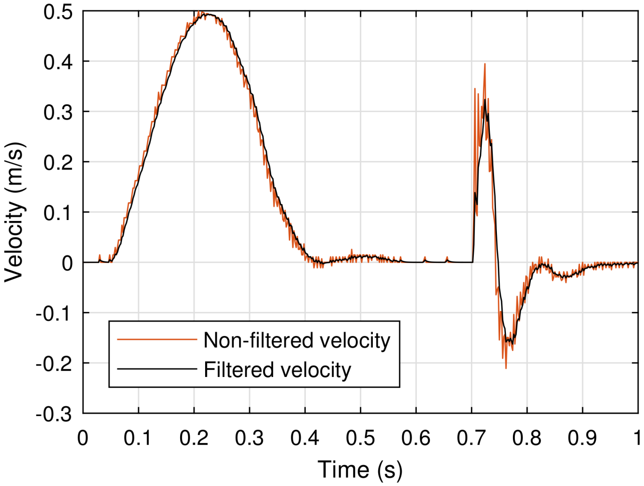

Figure 10: Filtered and non-filtered velocity

velocity value for the smooth trajectory: $v_{\max} = 0.5$ m/s. Filtering the velocity is quite useful for the rejection of external disturbances.

7. Conclusion

This work analyses and describes an effective control algorithm based on two cascade loops to drive mechanical devices. The algorithm uses a P controller for the outer position loop and a PI controller for the inner velocity loop. The gains of the controller are selected based on the continuous-time model of the system. These parameters only depend on the natural frequency ω_n, chosen as the bandwidth of the velocity loop, and the inertia of the device. This control strategy is able to reject external disturbances smoothly and quickly.

When the system is driven by cost-effective electronics, performance and stability can be limited due to the low sampling rate that governs the control loop. This works describes a methodology to effectively tune control gains under this assumption and to evaluate its implication for the stability of the system. The proposed methods are validated on a desktop haptic device showing that good performance can be achieved with cost-effective electronics with proper control strategies and parameter selection.

A. Matlab code to generate smooth trajectories

```
% Script reference.m to generate a smooth setpoint
% Input parameters: final position       (m)
%                   maximum velocity     (m/s)
%                   maximum acceleration (m/s^2)
%                   maximum jerk         (m/s^3)
% Output: time     (s)
%         position (m)

function F = reference(pmax,vmax,amax,jmax)

Dt = 0.002;     % Dt = 0.002 s could be an input
time = 0:Dt:3;  % t_final = 3 s could be an input
leng = length(time);
p = zeros(1,leng);
v = zeros(1,leng);
a = zeros(1,leng);

t12 = amax/jmax;
t23 = vmax/amax;
t45 = pmax/vmax;
t67 = t23+t45;

index_avoidfour = 0.0;
index_noamax = 0.0;
n = 1;
```

```
for k = 2:leng

switch n
  case 1
    if time(1,k)+Dt >= t12
      n = 2;
    end
    a(1,k) = jmax * time(1,k);
    v(1,k) = v(1,k-1) + a(1,k) * Dt;
    % Check if we get vmax before reaching amax
    if 2*v(1,k) >= vmax
      n = 3;
      t12 = time(1,k);
      t23 = time(1,k);
      t67 = t23+t45;
      amax = a(1,k);
    end
    p(1,k) = p(1,k-1) + v(1,k) * Dt;
    % Check if we get pmax before reaching amax
    if 12*p(1,k) >= pmax
      n = 3;
      t12 = time(1,k);
      t23 = time(1,k);
      t45 = time(1,k)+t12;
      t67 = t23+t45;
      amax = a(1,k);
      index_avoidfour = 1.0;
    end
  case 2
    if time(1,k)+Dt >= t23
      n = 3;
    end
    a(1,k) = amax;
    v(1,k) = v(1,k-1) + a(1,k) * Dt;
    p(1,k) = p(1,k-1) + v(1,k) * Dt;
    % We get pmax befofe reaching vmax
    % Jump to 3 with order to omit 4, else jump to 6
    if 2*(p(1,k)+t12*(v(1,k)+amax*t12/3)) >= pmax
      n = 3;
      t23 = time(1,k);
      t45 = time(1,k)+t12;
      t67 = t23+t45;
      index_avoidfour = 1.0;
    end
  case 3
    a(1,k) = amax - jmax * ( time(1,k) - t23 );
    % Check if as in section 5 and then jump to 6
    if index_avoidfour == 1.0
      if a(1,k) <= -amax
        a(1,k) = -amax;
        n = 6;
      end
    % Condition to jump from section 3 to 4
    elseif time(1,k)+Dt >= t23+t12
      n = 4;
    end
    v(1,k) = v(1,k-1) + a(1,k) * Dt;
    p(1,k) = p(1,k-1) + v(1,k) * Dt;
  case 4
    if time(1,k)+Dt >= t45
      n = 5;
    end
    a(1,k) = 0.0;
    v(1,k) = v(1,k-1) + a(1,k) * Dt;
    p(1,k) = p(1,k-1) + v(1,k) * Dt;
  case 5
    if time(1,k)+Dt >= t45+t12
      n = 6;
    end
    a(1,k) = - jmax * ( time(1,k) - t45 );
    v(1,k) = v(1,k-1) + a(1,k) * Dt;
    p(1,k) = p(1,k-1) + v(1,k) * Dt;
  case 6
    if time(1,k)+Dt >= t67
      n = 7;
    end
    a(1,k) = -amax;
    v(1,k) = v(1,k-1) + a(1,k) * Dt;
    p(1,k) = p(1,k-1) + v(1,k) * Dt;
  case 7
    if time(1,k)+Dt >= t67+t12
      n = 8;
    end
    a(1,k) = -amax + jmax * ( time(1,k) - t67 );
    v(1,k) = v(1,k-1) + a(1,k) * Dt;
    p(1,k) = p(1,k-1) + v(1,k) * Dt;
  case 8
    a(1,k) = 0.0;
    v(1,k) = 0.0; % Force zero velocity
    p(1,k) = p(1,k-1) + v(1,k) * Dt;
end

end

pfinal = pmax * p / max(p); % For integration errors
plot(time,pfinal,'k');      % Optional plot
F = [time' pfinal'];        % Output in columns
```

References

[1] S. Song, W. Cai, and Y.-G. Wang: Auto-tuning of cascade control systems, *ISA Transactions*, **42**(1) (2003), 63–72.

[2] R. Vilanova and O. Arrieta: PID tuning for cascade control system design, in *2008 Canadian Conference on Electrical and Computer Engineering*, May 2008, pp. 001 775–001 778.

[3] S. Formentin, A. Cologni, D. Belloli, F. Previdi, and S.M. Savaresi: Fast tuning of cascade control systems, in *18th World Congress The International Federation of Automatic Control*, vol. IFAC Proceedings Volumes, no. 44, Milano, Italy, 2011, pp. 10 243–10 248.

[4] B&R, *TM450 – ACOPOS Control Concept and Adjustment*, B&R Industrial Automation GmbH, 2007.

[5] W. K. Chung, L.-C. Fu, and T. Kröger: *Springer Handbook of Robotics*, 2016, ch. Motion Control, pp. 163–194.

[6] Y. Altintas, A. Verl, C. Brecher, L. Uriarte, and G. Pritschow: Machine tool feed drives, *CIRP Annals – Manufacturing Technology*, **60** (2011), 779–796.

[7] R. Haber, J. Arenas, F. Matia, A. Villalonga, and J. González-Fontanet: Automatic optimal setting of a cascade control system. application to trajectory tracking control with friction and backlash, in *XXXIX Jornadas de Automática*, Badajoz, Spain, 2018, pp. 924–931.

[8] Rockwell, *MOTION-AT005C-EN-P Motion System Tuning*, Rockwell Automation, Inc., May 2016.

[9] Fagor, *CNC 8065 – Programming manual*, Fagor Automation S. Coop., 2014.

[10] M. Nordin and P.-O. Gutman: Controlling mechanical systems with backlash – a survey, *Automatica*, **38** (2002), 1633–1649.

[11] J.J. Gil, I. Díaz, P. Ciáurriz, and X. Justo: Educational haptic controller based on Arduino platform, in *11st Congress of Technologies Applied to Electronics Teaching*, Bilbao, Spain, June 11-13 2014, pp. 205–211.

[12] L. Abdullah, Z. Jamaludin, M. Salleh, B. Abu Bakar, J. Jamaludin, T. Chiew, and N. Rafan: Theoretical analysis of velocity and position loop behaviour of nonlinear cascade feedforward controller for positioning of xy table ballscrew drive system, in *Materials, Industrial, and Manufacturing Engineering Research Advances 1.1*, ser. Advanced Materials Research, vol. 845, Trans Tech Publications, 2 2014, pp. 831–836.

[13] K. Ogata: *Discrete-Time Control Systems*, 2nd ed., Pearson Prentice-Hall, 1995.

[14] *Modern Control Engineering*, Pearson Prentice-Hall, 2002.

[15] R.H. Castain and R.P. Paul: An on-line dynamic trajectory generator, *The International Journal of Robotics Research*, **3**(1) (1984), 68–72.