

Analytic Proportional-Derivative Control for Precise and Compliant Motion

Brian F. Allen, Michael Neff and Petros Faloutsos

Abstract—Precise control with proportional-derivative (PD) control generally requires stiffness. The proposed method determines critically damped PD control trajectories that precisely obtain target position and velocity constraints for arbitrary initial conditions. An analytic solution provides the PD control parameters, thereby determining the required impedance. The resulting controller precisely interpolates the target state by solving the full boundary-value problem. Control parameters are time-invariant, and need only be recomputed if the system diverges from the computed trajectory due to unexpected forces or noise. The resulting method provides control with automatically determined compliance, yielding natural response to perturbation.

NOMENCLATURE

$(\theta(t), \omega(t))$	Position and velocity at time t
$m(t)$	Moment of inertia at time t
$(\theta_0, \omega_0, t_0)$	Initial state of the system at time t_0
$(\theta_d, \omega_d, t_d)$	Desired state for the target time t_d
$\tau(t)$	Computed control torque for time t
k	Computed stiffness of PD controller
γ	Computed damping of PD controller

I. INTRODUCTION

Proportional-derivative (PD) control is one of the most commonly used forms of feedback control. This popularity stems from its simplicity, global stability, robustness and broad-applicability[1]. Despite many years of wide-spread use, practical implementations often resort to one of a variety of tuning methods to determine the stiffness and damping constants. One popular example is the Ziegler-Nichols method[2] for tuning proportional-integral-derivative (PID) controllers. A wide variety of automatic tuning methods have been proposed in the robotics and control literature[3].

Our approach considers the use of PD control for precisely interpolating future state. Whereas traditional practice might use a PD controller solely to track an interpolating spline, we propose to eliminate the tracked spline. In its place, our system solves for PD control parameters that will result in a critically damped trajectory that will attain the target state precisely.

With this method, both the target position and the target velocity of the desired state can be satisfied. The calculation of

B. F. Allen and P. Faloutsos are with the Department of Computer Science, University of California, Los Angeles, USA. vector@cs.ucla.edu and pfal@cs.ucla.edu

M. Neff is with the Department of Computer Science, University of California, Davis, USA. neff@cs.ucdavis.edu

This research was funded in part by U.S. Army Medical Research & Materiel Command's Telemedicine and Advanced Technology Research Center and the UCLA Center for Advanced Surgical and Interventional Technology.

this critically damped trajectory is made possible by using an analytic solution for the PD control parameters. Thus, the proposed method finds the unique trajectory curve that solves the full boundary-value problem (BVP) of controlling the system from an initial state $(\theta_0, \omega_0, t_0)$ to a given future state $(\theta_d, \omega_d, t_d)$. By chaining a sequence of BVPs, complex trajectories interpolating a series of desired states are made possible.

There are several advantages to formulating control as the trajectory of a critically damped PD controller. First, since the result of the proposed method is simply a set of constant control parameters, the method's stability and robustness are well understood[3]. Second, PD trajectories may be more similar to muscle-generated trajectories and therefore may appear less "robotic" than trajectories based on, for example, cubic splines[4]. Third, the stiffness of the system is implicit in the trajectory. That is, the system has a stiffness that is uniquely determined by the BVP being solved, and so the system responds to perturbations and disturbances with natural compliance. Thus the system will have many of the properties observed in human muscle control, such as an increase in stiffness with increased torque[5].

II. OVERVIEW

To begin, section III introduces a recent analytic solution [6] of critically damped PD control parameters to robotics. This solution provides a means to determine constant PD control parameters that drive the system through a trajectory starting at the system's initial state $(\theta_0, \omega_0, t_0)$ and precisely interpolating the target position at the target time (θ_d, t_d) .

Following on, section IV provides a method for also honoring a desired target velocity ω_d at the target time t_d , i.e., $\omega(t_d) = \omega_d$. This is shown to be possible, despite the analytic solution of section III already being fully constrained, leaving no (mathematical) degrees-of-freedom to allow control over final velocity. Our approach is to introduce an invertible transform, called the f -adjustment. As the magnitude of the f -adjustment varies, so does the velocity at the target time $\omega(t_d)$ change. Once the f -adjustment that satisfies $\omega(t_d) = \omega_d$ is found, the system is transformed back to the original coordinates. The resulting solution satisfies the full boundary-value problem, i.e., the analytically calculated trajectory satisfies both the initial conditions $(\theta(t_0), \omega(t_0)) = (\theta_0, \omega_0)$ and the target conditions $(\theta(t_d), \omega(t_d)) = (\theta_d, \omega_d)$.

Section V describes the implementation, section VI presents the results of the simulation of the control of a single

actuator, and section VII concludes with a discussion of the limitations and benefits of the method.

III. ANALYTIC SOLUTION TO CONTROL PARAMETERS

PD control computes the control torque based on the difference between the current state θ, ω and a desired state θ_d, ω_d as $\tau = k(\theta_d - \theta) + \gamma(\omega_d - \omega)$. If the moment of inertia m about the axis of the joint is known, then an equivalent second-order differential equation can be written as

$$m \frac{d^2 \theta}{dt^2} + \gamma \omega + k \theta = \gamma \omega_d + k \theta_d. \quad (1)$$

Considering only controllers with critically damped behavior (i.e., $\gamma^2 = 4km$), the homogeneous solution to (1) is

$$\theta(t) = \left(\theta_0 + t \left(\omega_0 + \frac{\gamma \theta_0}{2m} \right) \right) e^{-\frac{\gamma t}{2m}}, \quad (2)$$

setting θ_d to 0 and possibly inverting θ so that $\theta_0 > 0$ without loss of generality.

In robotic systems, the initial conditions (θ_0, ω_0) are generally fixed, as they often represent the current state of the system. As we have restricted the controller to critically damped motion, the system has but a single degree-of-freedom γ . Ideally, we seek the parameter γ which, when applied to control the system, causes the future position to be reached $\theta(t_d) = \theta_d$.

A. Solving for PD Control Parameters

Although an algebraic closed-form expression for γ in (2) does not exist, recent work[6] proposes an analytic solution for the control parameter γ such that constraints on both the initial position and velocity, and the desired position are satisfied exactly. We briefly review this solution.

Expressions of the form $y = x e^x$ can be manipulated using the Ω map, also called the Lambert-W function [7], defined by $x = \Omega(x) e^{\Omega(x)}$. Although Ω has no closed-form expression, it can be represented analytically and is quite fast to compute in practice [7]. $\Omega(\cdot)$ is a multi-valued map, but the principle branch, i.e., the branch passing through the origin, can be expressed as $\Omega_0(x) = \sum_{n=1}^{\infty} \frac{(-n)^{n-1}}{n!} x^n$. For $x \geq -1/e, x \in \mathbb{R}$, $\Omega(x)$ has at least one real solution, and for $x > 0$, $\Omega(x)$ is unique.

By using the Ω map, (2) can be solved for γ in terms of the initial conditions (θ_0, ω_0) , the desired position θ_d and the moment of inertia m ,

$$\gamma = -\frac{2m}{t_d \theta_0} \left(t_d \omega_0 + \theta_0 \left(1 + \Omega \left(-\frac{\theta_d}{\theta_0} e^{-1 - \frac{t_d \omega_0}{\theta_0}} \right) \right) \right). \quad (3)$$

For simplicity, we assume t_0 is zero without loss of generality.

This equation provides the PD control parameter γ (and k by the critically damped assumption) that will yield a trajectory that exactly satisfies the initial conditions and the final position constraint $\theta(t) = \theta_d$. Figure 1 shows several example trajectories generated by varying the initial velocity under the constraint $\theta(10) = 1$. Note that with each different initial velocity, the target position is interpolated exactly.

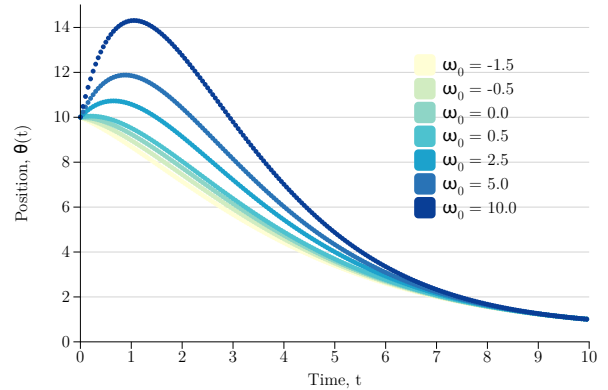


Fig. 1. The analytically computed control parameter γ provides a critically damped trajectory that exactly satisfies the target position. This graph shows a variety of initial velocities resulting in different trajectories, but each honoring the boundary conditions.

B. Trajectory Independence of Moment

For systems where the moment of inertia may not be constant, such as a multi-link robot, we desire that the trajectory of a joint will not be affected as the moment $m(t)$ changes over time. This is made possible by substituting (3) into (2) and defining

$$g \equiv \frac{1 + \Omega \left(-\frac{\theta_d}{\theta_0} e^{-1 - \frac{t_d \omega_0}{\theta_0}} \right)}{t_d} + \frac{\omega_0}{\theta_0}, \quad (4)$$

the expression for the trajectory independent of $m(t)$ is then

$$\theta(t) = e^{tg} (\theta_0 + t(\omega_0 - \theta_0 g)). \quad (5)$$

However, if the system is altered due to collision, perturbation or errors in the model used to calculate the moment $m(t)$, then the trajectory itself must be updated by recomputing a new g value that satisfies the target state given the newly changed current state. Speaking roughly, we may simply factor out the changing $m(t)$ when computing the trajectory, and then multiply it back in when computing the torque, thus leaving the trajectory itself unaffected by changes in the robot's configuration or mass properties.

IV. HONORING A DESIRED FINAL VELOCITY

The analytic solution of the previous section is useful for interpolation, but has limited application to the control of real systems, due to the inability to specify the velocity at the target time. Consider that the velocity of the system at the target time $\omega(t_d)$ is fully constrained by γ from (3). Equation 3 can have at most two solutions due to the $\Omega(\cdot)$ map having either zero, one or two solutions. Thus, the method of the previous section may result in trajectories with unacceptable velocities at the target time.

In this section, we extend the analysis to satisfy an arbitrary given target velocity ω_d at the target time t_d . To accomplish this, an extra mathematical degree-of-freedom is introduced using an invertible transform, called the f -adjustment. Our approach is to solve the f -adjusted version

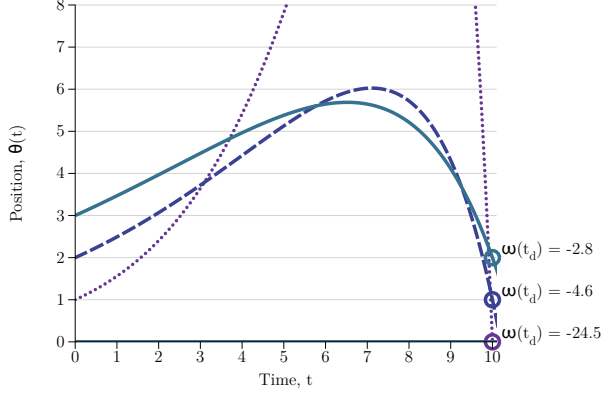


Fig. 2. An extra degree-of-freedom is introduced by translating the system by zero, one and two units. The transformed trajectories reach their desired positions exactly, each decreasing θ by one over the allotted time, but each arrives at the target position with a different velocity.

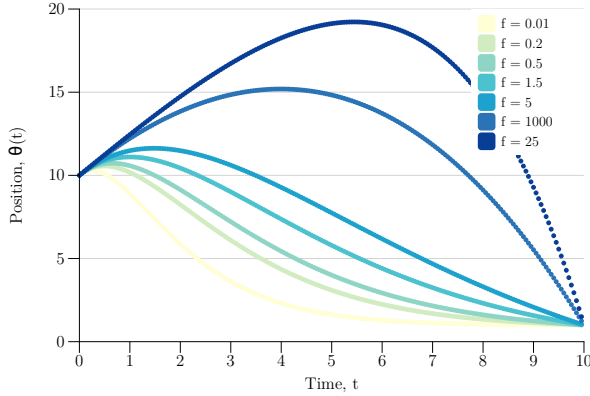


Fig. 3. By translating by f and then solving for γ , then translating back, a range of velocities at t_d are possible. Note however that there is not a monotonic relationship between f and $\omega(t_d)$.

of (3), thereby transforming the problem. Once an appropriate f -adjustment is found, the transform is inverted, leaving constant PD control parameters that satisfy the full boundary-value problem.

The intuition behind the f -adjustment is the observation that changing the magnitude of θ_0 and θ_d , while leaving their relative difference the same, changes the shape of the trajectory. Figure 2 shows three example trajectories, each starting with the same initial velocity and changing position by the same amount over the same time. However, two of the trajectories are offset by a fixed amount before determining γ with (3). Note that each of the velocities at the desired time are different. Thus by choosing an appropriate offset, the desired velocity can be met. Formally, the quantity f offsets the initial and target positions as

$$\hat{\theta}_0 = \theta_0 - \theta_d + f, \quad \hat{\theta}_d = f. \quad (6)$$

These f -adjusted values are denoted by $\hat{\cdot}$, and replace the corresponding terms in (3) to determine a γ with an altered trajectory. This method provides a potential means to determine a trajectory satisfying the full boundary-value conditions (both initial and final position and velocity). To

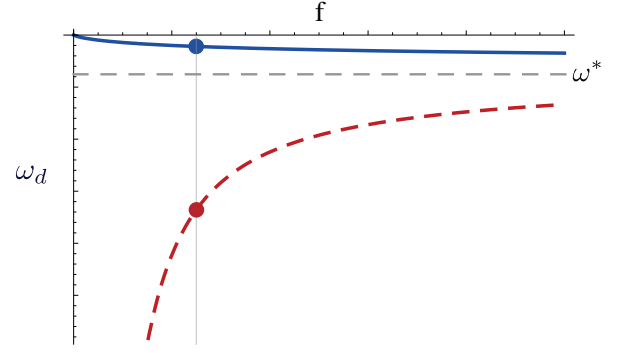


Fig. 4. The relationship between f and $\omega(t_d)$, showing the critical velocity ω^* separating the two branches. The Ω_0 branch is colored blue and the Ω_{-1} is red. Each point on either curve represents a distinct trajectory satisfying the given constraints on initial state and desired position and time.

find such a trajectory, we need only find f such that the desired velocity constraint is satisfied,

$$\frac{\partial \theta(t_d, f)}{\partial t} = \omega_d. \quad (7)$$

By differentiating (2) the velocity is expressed as a function of f and t ,

$$\frac{\partial \theta(f, t)}{\partial t} = -\frac{1}{t_d^2} \left(C + (\hat{\theta}_0 + B)t\xi \right) \exp\left(\frac{Bt}{t_d \hat{\theta}_0}\right) \quad (8)$$

$$\xi \equiv \Omega \left(-\frac{f}{\hat{\theta}_0} \exp\left(-1 - \frac{t_d \omega_0}{\hat{\theta}_0}\right) \right)$$

$$A \equiv 1 + \xi, \quad B \equiv t_d \omega_0 + \hat{\theta}_0 A$$

$$C \equiv t(t_d \omega_0 + \hat{\theta}_0) - t_d^2 \omega_0.$$

Figure 3 demonstrates the effect of the f -offset on the shape of a particular example trajectory. Equation 2 has regions of both no solution and regions of two solutions, due to the use of the Ω map. The two-solution case results in two separate trajectories for a given f , complicating the search for an f satisfying (7). However, the relationship between f and $\omega(t_d)$ is surprisingly simple. Figure 4 illustrates the behavior of the final velocities $\omega(t_d)$ as the f -offset changes. Note that the two branches correspond to the two real branches of the Ω map.

A root-finding search determines f that satisfies the final velocity constraint. The search can be improved using the first derivative of the velocity with respect to f :

$$\frac{\partial^2 \theta(t, f)}{\partial t \partial f} = -\left(\frac{ADf + EF\xi}{Af\hat{\theta}_0^2 t_d^3} \right) t \exp\left(\frac{Bt}{t_d \hat{\theta}_0}\right) \quad (9)$$

$$D \equiv B t_d^2 \omega_0 + A \left(A \hat{\theta}_0 (B t + t_d \hat{\theta}_0) - B^2 t - t_d^2 \omega_0 \hat{\theta}_0 \right)$$

$$E \equiv f t_d \omega_0 - f \hat{\theta}_0 + \hat{\theta}_0^2$$

$$F \equiv t_d (B - t_d \omega_0) + A (B t + t_d \hat{\theta}_0)$$

To determine the appropriate branch for a given desired velocity, we define the critical velocity as

$$\omega^* \equiv \lim_{f \rightarrow +\infty} \frac{\partial \theta(t_d, f)}{\partial t} = \frac{2(\theta_d - \theta_0)}{t_d} - \omega_0. \quad (10)$$

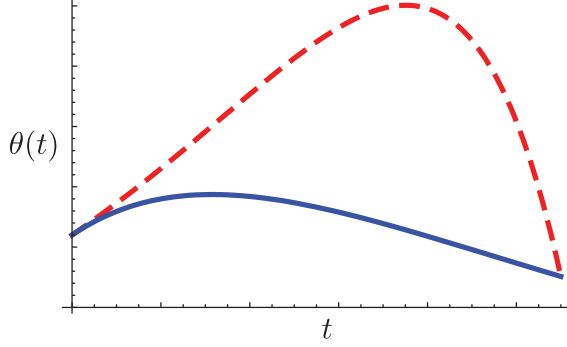


Fig. 5. The two trajectories implied by the two dots in Figure 4. The solid blue curve shows the trajectory corresponding to a small final velocity $\omega(t_d)$, while the dashed red trajectory indicates the curve with the larger magnitude final velocity.

The critical velocity is illustrated in Figure 4 with the two distinct branches of $\partial\theta(t_d, f)/\partial t$ converging at ω^* . These branches of $\partial\theta(t_d, f)/\partial t$ correspond to the two branches of $\Omega(\cdot)$ that provide real solutions. The search for f is constrained to the principle branch Ω_0 if $\omega_d > \omega^*$ and to the Ω_{-1} branch otherwise.

V. IMPLEMENTATION

Implementation of the feedback control loop is summarized in algorithm 1. With each iteration, the system determines whether or not the actual state has diverged from the current trajectory. When divergence is detected, a new trajectory is computed using the current system state for the initial condition, and the original target. Online recomputing of the trajectory allows proper response to collisions and perturbations, while still reaching the target state.

The argmin operation of algorithm 1 is implemented using the Newton-Raphson numerical root finder[8]. Because the derivative of the objective is known analytically (9), and because the objective function is known to be monotonic when constrained to the correct branch as given by (10), the minimization is very fast and generally converges in fewer than ten iterations.

When solving for the f -offset, some trajectories can require very large values of f to reach the target velocity. In these cases, numerical stability can suffer, since the transformation requires the subtraction of generally small values, such as θ_d , by large f . To mitigate this issue, our implementation uses 80-bit extended-precision floating-point values.

VI. RESULTS

To evaluate the method, we consider the control of an simple single-DOF 10 kg arm, illustrated in Figure 6. The system is required to interpolate an arbitrary series of target states, shown as the small circles in Figures 7 and 8, and listed in table I. At the precise time and position of each target state, a 2 kg projectile sphere passes the plane of arm; if the position has been reached correctly, the ball will be

Algorithm 1 ControlLoop()

```

for all time  $t$  do
   $\theta \leftarrow e^{t\gamma} \left( \hat{\theta}_0 + t(\omega_0 - \hat{\theta}_0\gamma) \right)$ 
  if current state diverges from trajectory then
    {Recompute trajectory}
     $f \leftarrow \underset{f}{\operatorname{argmin}} \left( \frac{\partial\theta(t_d, f)}{\partial t} - \omega_d \right)$ 
     $\gamma \leftarrow \gamma(f)$ 
     $k \leftarrow \gamma^2 / (4m)$ 
  end if
  {Compute torque}
   $\tau \leftarrow k(\theta_d + f - \theta) - \gamma\omega$ 
end for

```

Algorithm 2 $\gamma(\cdot)$

```

 $\hat{\theta}_0 \leftarrow \theta_0 - \theta_d + f$ 
 $\Omega \leftarrow \operatorname{Branch}()$ 
 $\xi \leftarrow \Omega \left( -\frac{f}{\hat{\theta}_0} \exp\left(-1 - \frac{t_d\omega_0}{\hat{\theta}_0}\right) \right)$ 
return  $\frac{1}{\hat{\theta}_0} \left( \omega_0 + \frac{\hat{\theta}_0}{t_d} (1 + \xi) \right)$ 

```

deflected. To compare interpolation error quantitatively, the actual state is recorded at each target. Using the proposed method, mean error over the interpolation points is 0.0037 rad of position and 0.0195 rad/s of velocity. This compares well to a PD controller, with parameters hand-tuned to precisely satisfy the first target state, tracking a Hermite spline that interpolates the same states. The Hermite-tracking PD controller has a mean error of 0.0493 rad and 1.0448 rad/s over all of the interpolation points.

To examine the response to perturbation, each controller's error is measured under five arbitrary impulses of 5000 Nm applied during a single time step. The perturbed motions are shown in the bottom graphs of Figures 7 and 8. The Hermite spline tracker is tuned to be sufficiently stiff to completely reject the test perturbations, giving the same position and velocity error at the interpolation points in both cases. However, when subjected to perturbations, the proposed analytic trajectory controller yields mean position error of 0.0029 rad and velocity error of 0.0335 rad/s. In this case, the mean position error is actually lower when perturbations are applied. This effect arises because, as discussed in section V, if the system diverges from the expected trajectory a new trajectory is calculated. The new trajectory will be closer to the interpolation point, and thus less subject to numerical drift and introduction of error due

Algorithm 3 Branch()

```

if  $\omega_d > \frac{2(\theta_d - \theta_0)}{t_d} - \omega_0$  then
  return  $\Omega_0$ 
else
  return  $\Omega_{-1}$ 
end if

```

Time (sec)	Position $\theta(t)$ (rad)	Velocity $\omega(t)$ (rad/s)
0.5	0	0
1.25	1.37	0.35
3.0	-1.18	-0.48
3.4	-1.04	1.48
4.35	1.31	0.15
5.0	0.46	-2.86
5.2	-0.37	-0.63
5.5	0.82	0.44
6.76	-1.37	-0.07
7.26	-0.34	5.39
8.5	-0.7	-1.91

TABLE I
TARGET STATES

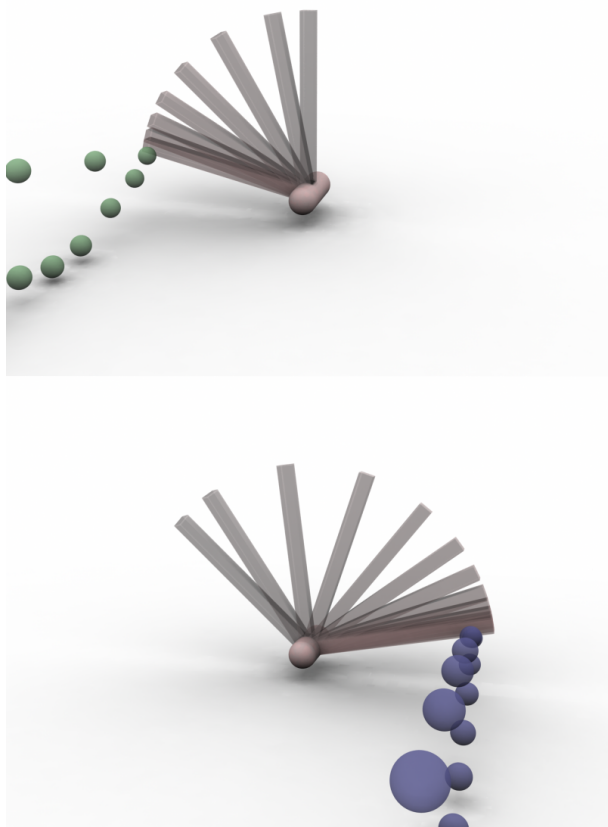


Fig. 6. A single DOF system is used to evaluate the proposed method. In this image, the actuator drives the arm with precise timing to block incoming balls; motion is illustrated using a stroboscopic effect.

to inexact model estimate. Note however, that the velocity error is nearly twice the unperturbed value. For each of these evaluations, the forward dynamics are integrated with a time step of 0.002 seconds.

VII. CONCLUSION

We have described a method to compute constant, linear control parameters for critically damped PD control able to precisely interpolate desired future states. The resulting trajectories are defined by the stiffness of control, leading to a natural method of determining impedance, and thus

behavior that is more similar to damped-spring-like muscle motion. The method also provides an analytic expression for both the trajectory and its derivatives, allowing interactive manipulation of the future trajectories. Finally, we have demonstrated the system is computationally efficient and able to respond robustly to unexpected perturbations.

The proposed method has some inherent limitations. First of all, the system model must be sufficiently accurate in providing the correct moment of inertia. Second, there are target states that simply cannot be reached from given initial conditions using a critically damped PD trajectory. For example, any case that would require two inflection points of the trajectory is impossible, as the critically damped PD trajectory has at most one inflection point. In these cases, the system is still able to reach to target position, but only by sacrificing accuracy in the target velocity. An example of this can be seen in the evaluation problem at 4.35 seconds. When the system is perturbed, the target velocity cannot be satisfied, leading to the larger average error in velocity with perturbations. Note that subsequent trajectories are likely to recover, as the error in velocity at the target becomes the new initial condition to reach the next target position.

Despite these limitations, we believe that the provided analytical solution of a long-standing problem may be of practical use both for the specific, proposed method of control through interpolation, and also more generally, to assist in the tuning of PD controllers.

REFERENCES

- [1] J. Craig, *Introduction to Robotics: Mechanics and Control*. Prentice Hall, 2004.
- [2] J. Zeigler and N. Nichols, "Optimum settings for automatic controllers," *Trans. ASME*, vol. 64, no. 8, pp. 759–768, 1942.
- [3] K. Ogata, *Modern Control Engineering*, 2nd ed. Prentice-Hall, Inc., 1990.
- [4] J. Winters and L. Stark, "Muscle models: what is gained and what is lost by varying model complexity," *Biological Cybernetics*, vol. 55, no. 6, pp. 403–420, 1987.
- [5] H. Gomi and M. Kawato, "Human arm stiffness and equilibrium-point trajectory during multi-joint movement," *Biological cybernetics*, vol. 76, no. 3, pp. 163–171, 1997.
- [6] B. Allen, M. Neff, and P. Faloutsos, "Pose control in dynamic conditions," in *Third International Conference on Motion in Games*, Zeist, 2010.
- [7] R. Corless, G. Gonnet, D. Hare, D. Jeffrey, and D. Knuth, "On the LambertW function," *Advances in Computational mathematics*, vol. 5, no. 1, pp. 329–359, 1996.
- [8] B. Flannery, W. Press, S. Teukolsky, and W. Vetterling, "Numerical recipes in C," *Press Syndicate of the University of Cambridge, New York*, 1992.

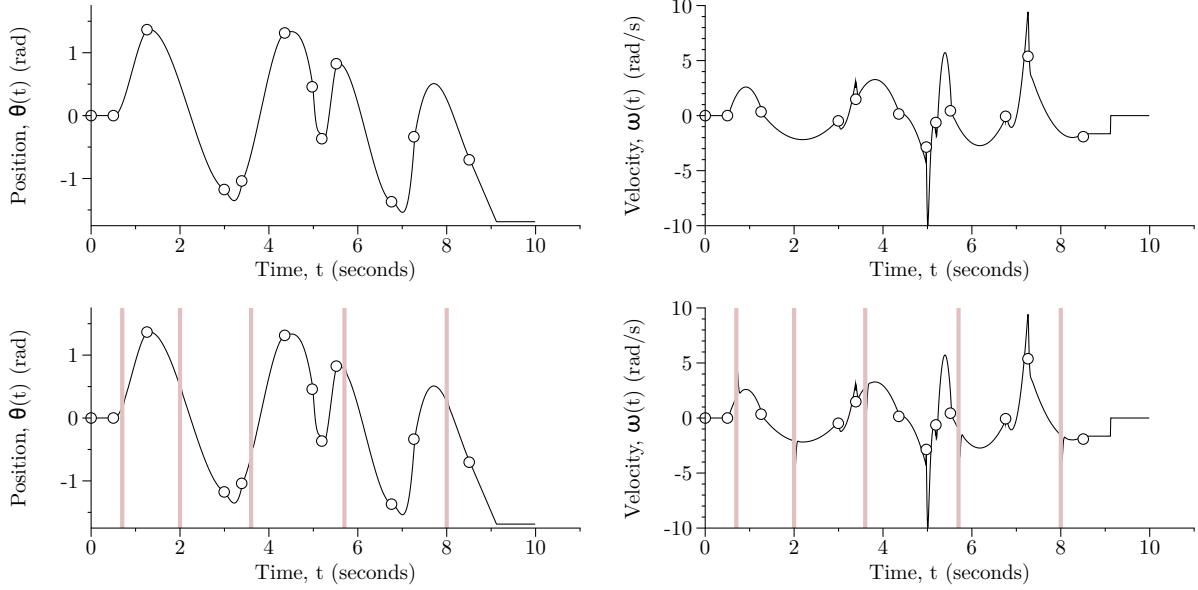


Fig. 7. Tracking an interpolating Hermite spline with a stiff PD controller. Vertical bars indicate unexpected perturbations. The perturbations are quickly rejected by the stiff controller.

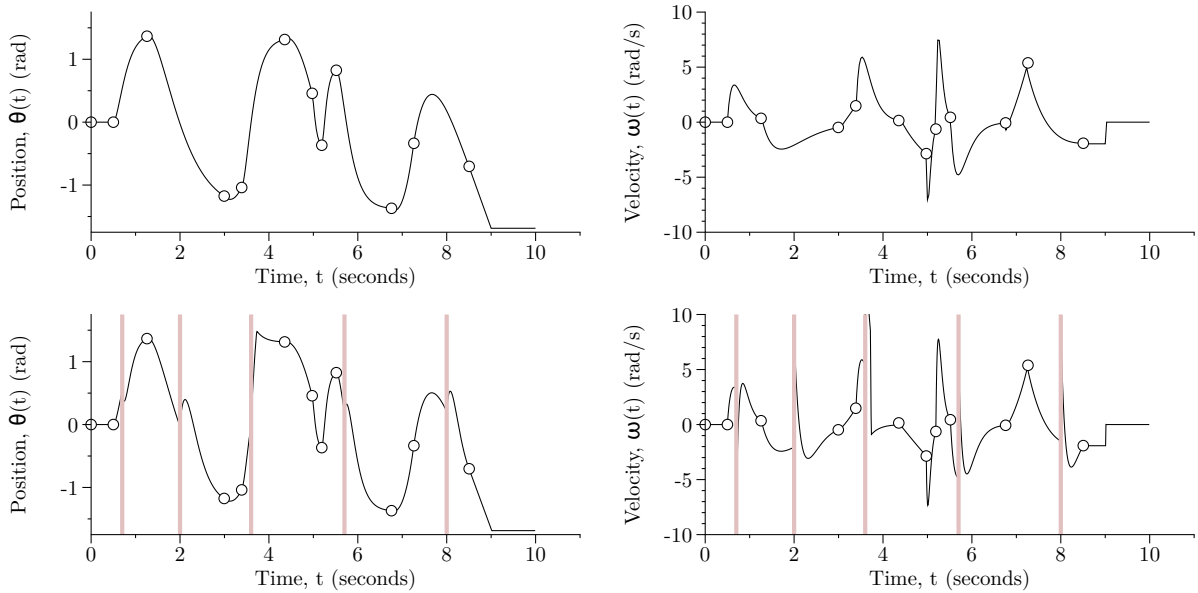


Fig. 8. Analytical PD control trajectory interpolating target states. Vertical bars indicate unexpected perturbations. Note that perturbations are resolved with a compliance that varies with the strength of the motion.