Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2012)
P. Kry and J. Lee (Editors)

# Misconceptions of PD Control in Animation

B. F. Allen[1] and P. Faloutsos[2]

[1]Nanyang Technological University, Singapore
[2]York University, Toronto, Canada

**Abstract**

*In this paper, we address certain misconceptions that have been perpetuated in the animation practice and research for quite some time related to the proportional-derivative (PD) control of physics-based systems. Because in animation we often think in terms of targeting keyframes, we tend to forget that PD control, in its simple form, has a very specific asymptotic behavior that approaches zero (or an offset) with zero velocity as time approaches infinity. We pay particular attention to the issue of introducing a "desired" or "end" velocity term in the equation of a proportional-derivative controller, and discuss how this term should be interpreted and how it relates to feedforward control rather than an end derivative problem.*

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

## 1. Introduction

A linear harmonic oscillator (a.k.a *linear spring*) is one of the most fundamental physical models in science. It can be found virtually in every elementary book on dynamics, (e.g. [Bar98]). Within the control domain it is commonly used as a regulator and often referred to as a proportional-derivative (PD) controller. Animation has borrowed the concept of proportional-derivative control for many applications. For example, it is widely used for pose to pose control (e.g [ANF10]), motion tracking (e.g. [ZH02]), and stabilization of constraint systems [Bar01]. It is also used as part of more sophisticated control systems such as [LT06, WGF08], and has been extended and treated in a variety of interesting ways (e.g [TLT11, ANF11]). The concept of PD control is so fundamental, simple and well studied, that we often forget its actual mathematical behavior especially as it applies to the control of discreetly simulated physical systems, such as human characters. Many papers in animation, especially those that deal with the problem of motion tracking, use the PD control equation in a fashion that makes it easy to misinterpret the function of the terms that appear in the equation.

To be clear, in this paper, we do not aim to correct or criticize any specific paper, rather we aim to remind the community of certain fundamental aspects of PD control and address certain misconceptions about its behavior. We also do not claim that we are the only ones who have identified these issues. There are papers that at certain places briefly address these misconceptions, such as [TLT11, WGF08]. However, these papers have a focus other than teaching the basic issues of PD control, and many of their readers seem to overlook the related statements.

We have verified that these misconceptions actually exist by talking to students and researchers in the field. To our surprise, even established researchers with years of experience in PD control, seem to have them. Thus, we feel that it is important to publicize these issues for the benefit of those who are not aware of them. This year's short papers track seems to us as a suitable place for a paper with a focused educational contribution.

## 2. The simplest example: a spring mass particle

In this section we will review the fundamental mathematical behavior of a PD controller with the simplest possible example, a single 1D particle whose position is controlled by a damped linear spring force. However, everything we discuss below applies directly to a 1D rotation spring as well.

*Spring Mass Particle.* Assume a one-dimensional particle with mass, $m$, attached to a massless linear spring with zero rest length, constant stiffness $k > 0$, damping $b > 0$, and anchored at $x = 0$. Given initial conditions on the position and velocity of the particle, and assuming that there are no other forces present, the motion of the particle is defined by a sec-



DOI: 10.2312/SCA/SCA12/231-234

ond order ordinary differential equation:

$$m\ddot{x}(t) + b\dot{x}(t) + kx(t) = 0, \quad (1)$$
$$x(0) = x_0, \dot{x}(0) = v_0,$$

where the dot indicates derivation with respect to time $t$. It is well known that depending on the relative values of $k, b$ the solution of this ODE can be one of three cases: over-damped, critically damped, and under-damped, see Figure 1. From the three solutions, the last two are the most interesting for animation purposes.

The solution for the critically damped case, which arises when $b = 2\sqrt{km}$, is a dropping exponential scaled by a second order polynomial:

$$x(t) = e^{-gt}\left(x_0 + (v_0 + x_0 g)t\right), \quad (2)$$

where $g = \sqrt{k/m}$. The effect of the polynomial is evident at the transient phase while at steady state the solution is dominated by the exponential term.

The solution for the underdamped case, which arises when $\omega_0^2 > g^2$, is

$$x(t) = Ae^{-gt}\cos\left(t\sqrt{\omega_0^2 - g^2} + \phi\right), \quad (3)$$

where $g = b/(2m)$, $\omega_0 = \sqrt{k/m}$, and $A, \phi$ are determined by the initial conditions. At steady state the effect of the exponential term dominates the solution.

In a discreet simulation setting such as those that arise in animation, these solutions would be computed with numerical integration. As long as the system is numerically stable, both of these solutions have a transient initial behavior that depends on the initial conditions, and asymptotically approach zero as time goes to infinity (the steady state) with zero velocity.

### 2.1. Non zero target value

To make the particle reach a desired location, $x_d$, the rest length of the spring is set to $x_d$ and equation (1) becomes:

$$m\ddot{x}(t) + b\dot{x}(t) + k(x(t) - x_d) = 0, \quad (4)$$
$$x(0) = x_0, \dot{x}(0) = v_0.$$

It is easy to see that if $x_d$ does not depend on time, equations (1) and (4) are equivalent under a change of variable approach. Indeed, for $y(t) = x(t) - x_d$, we have $\dot{y}(t) = \dot{x}(t), \ddot{y}(t) = \ddot{x}(t)$ and equation (4) becomes:

$$m\ddot{y}(t) + b\dot{y}(t) + ky(t) = 0, \quad (5)$$
$$y(0) = x_0 - x_d, \dot{y}(0) = v_0,$$

which is equivalent to equation (1).

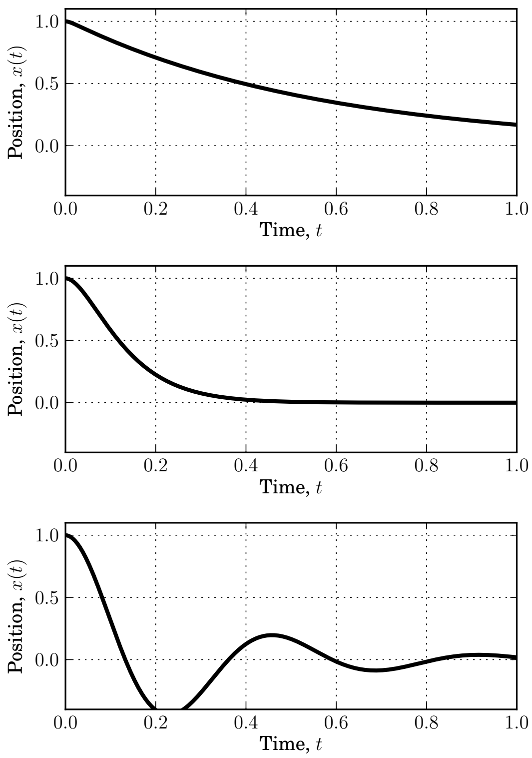


**Figure 1:** *The three classes of proportional-derivative behavior: overdamped (top), critically damped (middle), and underdamped (bottom).*

## 3. Misconception I

*We can add an end condition on the ODE to set the final position of $x(t)$ to $x(t_f) = x_d$.*

Equation (1) and its equivalent equation (4) are initial value problems with an asymptotic behavior. They go asymptotically to $x = 0$ and $y = 0$ respectively. The second form does not introduce a different final value, it simply offsets the solution by a constant amount, $x_d$ as shown in Section 2.1. Although this offset has the desired effect, it is not mathematically equivalent to solving a final value problem. Assuming non-trivial initial conditions, mathematically the two formulations never reach 0 or $x_d$ respectively in finite time.

## 4. Misconception II

*We can control the time that the particle will reach its target value.*

The time that it takes for the particle to go from its initial conditions to the target position (or arbitrarily close to the target), depends on both the PD parameters and the initial

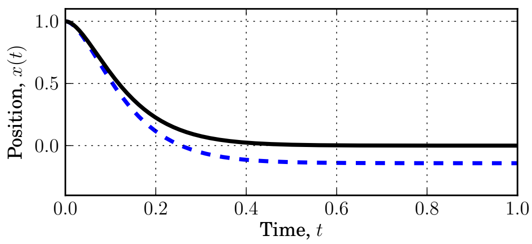


**Figure 2:** *The effect of a desired velocity term is an offset in steady-state position. The solid black trajectory corresponds to the target frame (pos = 0, vel = 0) whereas the dashed blue to (0,-1). The blue trajectory reaches the target 0 faster but has a steady state error.*

conditions. In most applications of interest, the initial conditions are the current keyframe, and thus, generally, change as the particle moves from frame to frame. Therefore, unless we solve a form of an optimal control problem every time we want the particle to go from an initial state to a target position, we do not have an accurate way of controlling the time that the particle will reach the target position. To address this, most often we resort to stiff springs, which ensure that the the particle will reach its target close enough at some point in time between the two keyframes. For the critically damped case, there are certain approximations and extensions over the simple PD control formulation that can control the timing for key framing reasons, for example [ACSF07].

## 5. Misconception III

*If we want our particle to reach location $x_d$ with velocity $v_d$, then we simply introduce the desired velocity to equation (4) as follows*:

$$m\ddot{x}(t) + b(\dot{x}(t) - v_d) + k(x(t) - x_d) = 0, \quad (6)$$
$$x(0) = x_0, \dot{x}(0) = v_0.$$

Whether we call $v_d$ a desired velocity or anything else it is only an offset on $x$ not $\dot{x}$. Therefore it is really a steady state error. This is easy to see if we rearrange equation (5) as follows:

$$m\ddot{x}(t) + b(\dot{x}(t) - v_d) + k(x(t) - x_d) = 0 \rightarrow$$
$$m\ddot{x}(t) + b\dot{x}(t) - bv_d + k(x(t) - x_d) = 0 \rightarrow$$
$$m\ddot{x}(t) + b\dot{x}(t) + k(x(t) - x_d - \frac{b}{m}v_d) = 0 \rightarrow$$
$$m\ddot{x}(t) + b\dot{x}(t) + k(x(t) - x'_d) = 0, \;\; x'_d = x_d + \frac{b}{m}v_d. \quad (7)$$

Therefore the new steady state is $x_\infty = x'_d = x_d + \frac{b}{m}v_d$.

Figure 2 demonstrates this issue with two experiments. In both cases the same critically damped spring is moving the particle from initial conditions $x_0 = 1, v_0 = 0$ to the target value of $x_d = 0$ using equation (6). The black trajectory corresponds to $v_d = 0$, while the blue trajectory to $v_d = -1$. The effect of the non-zero $v_d$ is evident. Although the blue trajectory reaches a steady state faster, it exhibits a steady state deviation from 0 of $x_\infty = -b/m$ as computed by equation (7).

## 6. Motion tracking

Misconception III has been created by many papers that have used PD control in animation and in particular those that address motion tracking. We are referring to this issue as a misconception rather than an error, because using a desired velocity term in the PD equation for motion tracking is generally beneficial. Unfortunately, very few papers state or explain this issue. Even when they do in a line or so (e.g. [WGF08, TLT11]), as it is not the main point of these papers many of their readers tend to overlook it. It therefore seems that many of us have adopted the impression that the desired velocity term has the same effect on the final velocity that the desired position has on the final position.

It is worth looking at the classic trajectory tracking problem. A typical motion tracking approach, first interpolates a set of given keyframes using splines, and then samples and tracks the spline using critically damped PD control. For good results, the spline is sampled at least at the display frequency and the PD controllers are stiff and critically damped. In other words we can assume that at each frame, we solve numerically a short PD control problem with initial conditions the current keyframe, and final position the end keyframe.

However, the PD control formulation does not target final velocities. As we showed earlier (equation (7)) the desired velocity is an error in steady state. If we look at the simulated trajectory between keyframes we will see that it is not as smooth as it looks at the display rate. For example, for a critically damped spring the in-between trajectories are short versions of the classic critically damped curve that we see in the basic literature, figure 3. As long as the spring is stiff enough for a given sampling rate, the trajectory will appear smooth, while it is actually a collection of piece-wise exponentially dropping curves.

So what is the benefit of the velocity term then? During motion tracking we are not interested in the asymptotic behavior of the curve. The fact that we have a non-zero final velocity means that we no longer want to stay at the final value over time. The velocity term acts as a feedforward term. In simple terms, it often points to the direction of the new location, and therefore adds to the PD control force that is moving the particle towards the new location. The practical effect of this term is that it reduces the stiffness of the spring that it is required to reach the new keyframe, or, for the same spring stiffness, it reduces the lag in the tracking [TLT11]. Looking at figure 2 we can again verify that with the velocity term the particle reaches is target location faster earlier.

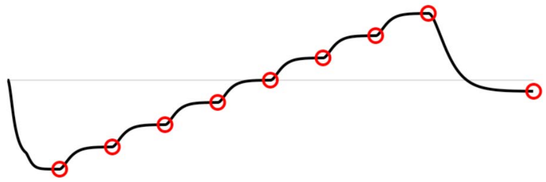

**Figure 3:** *Zoomed-in trajectory of a critically damped tracking PD controller following a smooth curve with an exaggerated integration step to trajectory sample ratio.*

## 7. PD control as a boundary value problem

Recent work [ANF11, ANF10] solves the PD control problem as a boundary value problem with full boundaries (position, velocity), and at the same time controls exactly the time that it takes to move between the two boundaries. This work solves the final value problem analytically: it computes the necessary stiffness for a critically damped spring that moves the particle from its initial state to the its final position precisely, and at the exact time desired. To achieve this, it solves equation (2) for the exponent $g$, using the Lambert-W function. Intuitively, the approach fits a critically damped trajectory between the initial state and the final position of the particle. Given a desired end velocity the paper introduces an efficient search method that adjusts the trajectory so that it passes at the final location with the desired velocity.

These works offer one of the most in-depth and detailed looks into the behavior of PD control, and at the same time serve as an example of how one can take an asymptotic problem and manipulated into a boundary value problem.

## 8. Articulated body systems

This paper is primarily concerned with the behavior of the 1D mass-spring system. For completeness, we briefly consider on the general problem of PD control for the articulated body case.

Controlling a full, simulated character often involves tens of interrelated DOFs. To reduce this complex control problem to the much simpler problem of a series of independent joints requires accounting for the dynamics of the system. This idea is referred to by a variety of names, including inner-loop control, inverse dynamics control [YCP03], feedback linearization and, broadly, computed torque control. In addition, approximations of inverse dynamics control have been suggested within the animation literature, including compensating for total distal mass [ZH02], and using a single-pass to compensate for proximal joint torque [ACSF07]. Machine learning has also been used to build approximations of the inverse dynamics from a biological perspective [LT06] and specifically aiming to mimic the role of the cerebellum [GK93].

## 9. Conclusion

We have discussed and addressed a few misconceptions that seem to have perpetrated in the animation community regarding the behavior of PD control. Because PD control is so well studied and so commonly used, we often forget what really happens between keyframes and how PD control actually works. This paper points out known but often forgotten issues for the benefit of the animation community.

## References


[ACSF07] Allen B. F., Chu D., Shapiro A., Faloutsos P.: On the beat!: Timing and tension for dynamic characters. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2007), pp. 239–247. 3, 4

[ANF10] Allen B. F., Neff M., Faloutsos P.: Pose control in dynamic conditions. In *Proceedings of the Third International Conference on Motion in Games (MIG)* (2010), vol. 6459, pp. 48–58. 1, 4

[ANF11] Allen B. F., Neff M., Faloutsos P.: Analytic proportional-derivative control for precise and compliant motion. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (2011). 1, 4

[Bar98] Baruh H.: *Analytical Dynamics*. McGraw-Hill, 1998. 1

[Bar01] Baraff D.: Physically based modeling: Rigid body simulation. *SIGGRAPH Course Notes, ACM SIGGRAPH 2*, 1 (2001), 2–1. 1

[GK93] Gomi H., Kawato M.: Neural network control for a closed-loop system using feedback-error-learning. *Neural Networks 6*, 7 (1993), 933–946. 4

[LT06] Lee S.-H., Terzopoulos D.: Heads up! Biomechanical modeling and neuromuscular control of the neck. *ACM Transactions on Graphics 25*, 3 (2006), 1188–1198. 1, 4

[TLT11] Tan J., Liu K., Turk G.: Stable proportional-derivative controllers. *Computer Graphics and Applications, IEEE*, 99 (2011), 1–1. 1, 3

[WGF08] Weinstein R., Guendelman E., Fedkiw R.: Impulse-Based Control of Joints and Muscles. *IEEE Transactions on Visualization and Computer Graphics 14*, 1 (2008), 37–46. 1, 3

[YCP03] Yin K., Cline M. B., Pai D. K.: Motion perturbation based on simple neuromotor control models. In *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications* (2003), p. 445. 4

[ZH02] Zordan V. B., Hodgins J. K.: Motion capture-driven simulations that hit and react. In *In Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2002), pp. 89 – 96. 1, 4