Impulse-Based PD Control for Joints and Muscles

Rachel Weinstein*
Stanford University

Eran Guendelman*
Stanford University

Ron Fedkiw*
Stanford University
Industrial Light + Magic

Introduction

We propose a novel approach to proportional derivative (PD) control that exploits the fact that these equations can be solved analytically for a single degree of freedom. The analytic solution tells us what the PD controller would accomplish in isolation without interference from neighboring joints, gravity and external forces, outboard limbs, etc. Our approach to time integration uses an inverse dynamics style formulation that automatically incorporates global feedback so that the per joint predictions are achieved. Stiffness is decoupled from control without the need for estimating external forces as in [Neff and Fiume 2002] so that we obtain the desired target regardless of a joint's stiffness, which merely determines *when* a target angle is hit. Whereas PD is typically applied via torques allowing drift, we follow [Guendelman et al. 2003] working with impulse and velocity as opposed to force and acceleration. This also allows for robust incorporation of collisions and contact. In particular, we use the framework of [Weinstein et al. 2005] making heavy use of post-stabilization to implement our PD control method.

Description

We begin by discussing PD control in a generalized coordinate formulation, where each coordinate represents a degree of freedom. Given a sufficiently smooth target trajectory $\theta_o(t)$, the PD control law specifies the generalized acceleration as $\ddot{\theta} = \ddot{\theta}_o - k_p(\theta - \theta_o) - k_v(\dot{\theta} - \dot{\theta}_o)$ where k_p and k_v are the proportional (position) and derivative (velocity) gains, respectively. Few authors include $\dot{\theta}_o$ and $\ddot{\theta}_o$, instead targeting a zero velocity and acceleration. However, including these allows us to formulate a second order equation for the error, $\ddot{E} + k_v\dot{E} + k_pE = 0$ where $E = \theta - \theta_o$. Given errors in both velocity and position, *critical damping* drives those errors to zero most quickly. This is achieved by setting $k_v = 2\sqrt{k_p}$ reducing the choice of gains to a one dimensional family parameterized by k_p. Increasing k_p causes the trajectory to be tracked more strongly. A key point is that this PD formulation follows the trajectory via a second order system compatible with the physics, in contrast to Ferguson curves or other interpolation schemes.

We exploit the fact that one can integrate analytically to obtain closed form expressions for the error E and thus the exact solution $\theta_E = \theta_o + E$ at the end of a time step. Setting our angular velocity based on the secant to the exact solution curve gives $\hat{\omega} = (\theta_E - \theta^n)/\Delta t$ which guarantees that we achieve the exact solution at the end of an Euler step.

We use the time discertization from [Guendelman et al. 2003], $\omega^{n+1} = \omega^n + \Delta t\alpha^n$ and $\theta^{n+1} = \theta^n + \Delta t\omega^{n+1}$ (where α is angular acceleration), with the position update replaced by $\theta^{n+1} = \theta^n + \Delta t\hat{\omega}$. It can be shown that $\hat{\omega}$ is identically $\dot{\theta}^{n+1/2}$ up to $O(\Delta t^2)$, and thus $\hat{\omega}$ naturally lives at time $t^{n+1/2}$ and our modified position update resembles central differencing. However, while central differencing also includes a step to integrate $\omega^{n+1/2}$ to ω^{n+1}, this approach lacks such a step instead using $\omega^{n+1} = \hat{\omega}$. Consequently, in the limit as $\Delta t \to 0$, only half of the acceleration $-k_pE - k_v\dot{E}$ is accounted for, which is equivalent to using PD control with gains $k'_p = k_p/2$ and $k'_v = k_v/2$.

This problem is trivially corrected by applying our method to a set of PD equations with the gains doubled: $\hat{k}_p = 2k_p$ and $\hat{k}_v = 2k_v$. This becomes an *overdamped* system, but in the limit as $\Delta t \to 0$ the system behaves as though it were critically damped. Note that our scheme remains unconditionally stable *and* drives the error to zero even for large Δt.

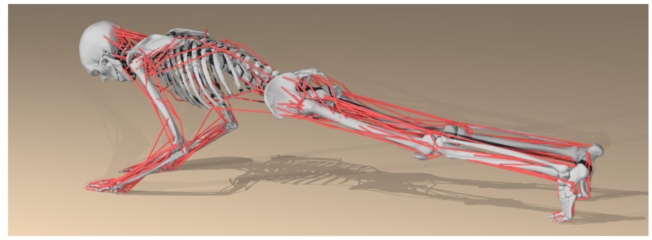

Figure 1: **A skeleton performs push-ups with 228 muscles whose forces were calculated via PD control (118 dof).**

Since PD control is typically integrated into velocity as a force, we incorporate our method into the post-stabilization step that immediately follows the velocity update in the method proposed in [Weinstein et al. 2005]. Extending PD control to multiple joints significantly complicates the situation, because each joint generates forces that interfere with neighboring joints. We use inverse dynamics to alleviate this problem solving for the impulses that take surrounding joints into account by extending post-stabilization from a joint by joint approach to a global approach for the entire articulated rigid body. The global framework requires a matrix relating the impulses at all joints to the change in velocities at all joints. We solve a least squares problem which minimizes both the norm of the solution as well as the residual of the overdetermined part of the equations.

Since global post-stabilization gives global feedback, joints do not fight each other and instead move smoothly towards the target state. Besides obtaining smooth motion, an advantage of dynamic controllers (versus kinematic) is response to unanticipated forces. For example, a character should easily move around their own limbs, but struggle to lift a foreign object.

In addition to smaller examples used to illustrate many of the previously mentioned concepts, we created a skeleton from the Visible Human data set and animated the skeleton flailing in a net and swinging a mace. The skeleton joint movement is defined by an analytic function targeted via our PD control, while the mace and net are freely moving unactuated joints. We also demonstrate an optimization-based method for inverse muscle actuation with an example of our skeleton performing a series of push-ups (Figure 1). The push-up motion was created using analytic functions, and our optimization method determines the muscle actuation required to achieve the desired joint angles throughout the body. Unlike torque actuation, muscle actuation is restricted to lie along the muscle's line of action, and inequality constraints are used to enforce bounds on muscle force. Our method trivially handles muscles spanning multiple joints. Even with the additional calculation of the muscular impulses, this simulation only required three minutes a frame. Examples with lower degrees of freedom ran in interactive time.

References

GUENDELMAN, E., BRIDSON, R., AND FEDKIW, R. 2003. Nonconvex rigid bodies with stacking. *ACM Trans. Graph. (SIGGRAPH Proc.) 22*, 3, 871–878.

NEFF, M., AND FIUME, E. 2002. Modeling tension and relaxation for computer animation. In *Proc. ACM SIGGRAPH Symp. on Comput. Anim.*, 77–80.

WEINSTEIN, R., TERAN, J., AND FEDKIW, R. 2005. Dynamic simulation of articulated rigid bodies with contact and collision. *IEEE Trans. on Vis. and Comput. Graph. (in press)*.

*e-mail:{rachellw,erang,fedkiw}@graphics.stanford.edu