# Modeling Tension and Relaxation for Computer Animation

Michael Neff* Eugene Fiume

Department of Computer Science
University of Toronto

## Abstract

The use of tension and relaxation in the muscles of real creatures gives rise to nuanced motion that conveys emotion or intent. Artists have long exploited knowledge of this in traditional animation and other areas, but it has been both overlooked and difficult to achieve in physically based animation. The robotically stiff motion that has come to typify physically based approaches belies the fact that dynamics has much to offer in facilitating far more subtle motion in which animators could freely "shape" a motion. We demonstrate that tension and relaxation can be introduced into joint-level, posture based animation. While we show that these modalities can be efficiently incorporated into traditional proportional-derivative control models, we instead formulate a more flexible and better-behaved model based on antagonistic control. This approach is more biomechanically sound, but more importantly it permits the separation of stiffness control from position control, achieving better posture interpolation, better error control, and passive and active dynamics. We introduce effective mechanisms to control the shape of a motion and describe an animation system that efficiently integrates relaxation and tension control in a physically based simulation environment.

**CR Categories:** I.3.7 [Computer Graphics]: Three Dimensional Graphics and Realism—Animation;

**Keywords:** Animation, Physically Based Animation, Human Body Simulation

## 1 Introduction

Imagine that one day you are out for a walk through town. Just ahead, you see a drunken man stumble out of a bar. His movements are loose and uncontrolled as he staggers by you. A little further along you pass an army base. The soldiers are marching, their maneuvers are tight and precise. After turning down a side street, you see a young girl skipping along the sidewalk, swinging her arms in a happy dance. Lost for a moment, you glimpse too late out of the corner of your eye the sight of a soccer ball sailing toward you. You reflexively brace for impact.

These simple but interesting movements create a wide range of impressions. Their differences relate largely to the amount of tension involved in their motions. The drunken man and young girl both moved in a free, relaxed manner. Their movements were loose and less constrained. On the other hand, the marching soldiers, and you as you braced for impact, displayed considerably more tension.

These observations have not been lost on traditional animators. Recall Disney's hand-drawn *Dumbo* in which Dumbo's mother cradles his head with her trunk. Think as well of the sadly sagging unicycle *Red* in the early Pixar computer-aided animation *Red's Dream*. While separated by more than 45 years (1941 and 1987, respectively), the remarkably effective use of tension and relaxation in impressionistic animation sets a high visual standard for technical contributions to physically based character animation.

Turning now to realistic motion synthesis, movement in the human body revolves around the subtle interplay of tension and relaxation. When we are relaxed, our limbs sway freely under the influence of gravity and external forces, loosely guided by our muscles. The force from the movement of one limb will effect the movement of limbs attached to it. When a person is tense, however, their movements will appear tightly controlled and their posture more rigid. The transference of force from a limb's movement will have little effect on neighbouring limbs and there will be less evidence of the effect of gravity in a character's movement. The difference between a loosely moving clown and a precisely moving soldier or robot is largely a difference in tension and relaxation of various joints. Moreover, tension and relaxation will vary across joints and throughout various movements.

To make systematic progress, we must be clearer and more limited in scope. Our aim is to provide animators with the tools needed to model tension in a character's body in order to generate a wide range of expressive animations. Our current focus is on "low-dynamic" behaviours such as gestures and posture adjustments, although the ideas are applicable to the full range of movements. While admitting a rich repertoire of motion, these "low-dynamic" behaviours are not as mechanically constrained as highly dynamic behaviour such as gymnastic maneuvers [7, 22] and falling[3]. Despite this, dynamics has an obvious impact on the expressive nature of movement. This is due to the constant influence on our bodies of gravity, momentum and force transference between joints.

The tension present in a person's body effects the trajectories joints take during a movement. A tightly-controlled, or high tension motion requires exerting additional energy to overcome the effects of gravity and momentum. Looser movements will instead exhibit the influence of gravity, momentum and force transference between joints. These effects give the movements a sense of weight and help create a sense of life in the animation.

Determining kinematic trajectories that reflect tension is a difficult though effective art in traditional character animation. Our system seeks to use the physical dynamics, together with the tension state of the character to determine the final appropriately nuanced joint trajectories. This shifts the work of tediously specifying a joint trajectory to the animation system, easing the animation task. Traditional physics-based approaches to character animation have not captured tension and relaxation well. Proportional derivative control has commonly been used. Here, often spring constants are kept high to minimize joint error and improve position tracking, but this approach will often lead to stiff looking movement unless the trajectories being tracked are of very high quality. Alternately, the gains are kept low and position control is imprecise. In our approach, the dynamics are used to help determine appropriate trajectories, leading to more realistic movement.

Our approach generalizes pose control[21]. Poses, in a manner similar to keyframes, indicate a desired set of joint angles for a character. A pose in our system can involve any subset of the character's degrees-of-freedom and poses can overlap in time. Controllers are built out of collections of poses and can include cycles. At any given time, multiple controllers can be working together to determine a character's movements. Animations are generated

* {neff|elf}@dgp.toronto.edu

from scripts which specify controllers and their start times. Animators can tag poses with transition times and kinematic transition functions that suggest the interpolation path for a DOF, as well as tension parameters. The actual path taken by a DOF will be determined by the interplay in the dynamics simulation of the kinematic transition function and the tension parameters. Position is controlled by regulating the equilibrium point of the forces acting on the character, which allows precise joint positioning. An antagonistic actuator formulation is used which allows joint tension to be freely varied without changing the joint angle.

Our approach has several advantages. Chief among them, it allows very straightforward and flexible control over the amount of tension in a character's body. This is a powerful expressive tool that makes it easier to generate realistic joint trajectories. Unlike traditional PD control, the antagonistic formulation decouples stiffness control from position control. The system also achieves more precise position control as the effect of gravity and external forces are taken into account. Flexible control over the "shape" or "envelope" of a movement is provided both through tension control and by allowing kinematic transition functions to be specified which guide the movement. Transition functions allow for real-time kinematic previews, which are useful for rapidly designing controllers. The formulation is well suited for creating movement that is more richly nuanced and realistic than traditional physics-based approaches to animation.

This paper proceeds by first presenting relevant background material. Equilibrium point control and our antagonistic formulation are presented in some detail, followed by a discussion of how movement "shape" or "envelope" is varied in the system. Finally, a range of results is presented, which include modeling posture, gesture and anticipation effects. Animations are shown both for a full body model and a torso model.

# 2 Background

## 2.1 Computer Animation

Our work relates most closely to previous work done on hand tuned controllers for physics-based animation. Hodgins et al., Wooten, and Faloutsos et al. have presented such controllers for various activities including running [7], gymnastics [7, 22] and reacting to falling [3]. These works use proportional derivative control to calculate the joint torques required for the movements. They also focus on high level control of highly dynamic activities. Our work addresses how to more effectively control the expressive aspects of movement such as timing, tension and shape. This paper reports on our progress on low-dynamic motion, namely fine-grained physically based motion that depends less on impulsive control than continuous control. As far we are aware, the modeling of tension and relaxation has not been discussed in previous work in this field.

The use of PD control is pervasive in the literature. Laszlo et al. use PD control in which the animator interactively varies the set point in order to generate the animation [13]. Zordan and Hodgins use PD control to track motion capture data for the upper body of a character[24]. They scale the gains based on the inertia of the body part attached by the joint and use an optimization method to determine the best gains to track the input data. Once set, the gains are locked for the animation. Our system allows an animator to vary gains throughout a motion and an animation to achieve more nuanced effects.

Kokkevis et al. take a similar approach to animation as ours, tracking desired kinematic trajectories with a control system and using a dynamic simulator to generate the final motion[11]. They divide DOFs into primary and secondary groups and accurately track kinematic trajectories for the primary DOFs, allowing the secondary DOFs to move passively. Our system takes a continuum approach, providing control information for all DOFs, but allowing users to specify how closely the trajectories are to be followed. This can range from closely tracking the trajectory to allowing fully passive motion, and all motions in between.

The work of Lamouret and Gascuel is similar to our system in that they feature rough kinematic trajectories that are loosely followed by a control system[12]. The goals of their work, however, are synchronization of movements, collision handling and object interaction, rather than expressive control of an animation.

Mataric and collaborators have presented a variety of control strategies [15, 14]. For gestures, they use PD control to achieve specified joint angles. End effector positioning is achieved using impedance control where the mechanical impedance of the end effector is modulated by varying the torques at each of the joints. They also present a joint space force field, which is a weighted combination of step and pulse functions multiplied with individual force functions that are similar to PD control for small gain, but drop off for larger errors.

Chi et al. provide a kinematic system that allows the trajectory of arm motions to be varied between end positions[2]. They use Laban Movement Analysis as the basis for an interface that allows a user to select a particular style of movement. Being kinematic, their system in part attempts to explicitly represent some of the effects of tension and relaxation on movement. Our system modifies the tension directly and allows the resulting movement to be generated by the dynamics system. Their movement interpolation interface could be useful in our system to control transitions between poses if it was adapted to make use of the tension parameters.

The work of Perlin uses structured noise to simulate kinematically some of the effects related to tension changes in the body.[18]

## 2.2 Biomechanics

The human body organizes muscles in agonist and antagonist groups around joints. The muscle pairs provide movement in opposing directions around a joint and also act to regulate the stiffness of the joint[8]. The system outlined here follows this organization, using pairs of antagonistic angular springs to control the position and stiffness of each DOF.

The spring-like behaviour of real muscles is important in postural stability. "It ... turns out that for stable equilibrium the neuromusculoskeletal system must possess the attributes of springs, and that for stability, these springs must exceed a certain critical stiffness. The passive, relaxed person is inherently instable at many levels – when someone faints, the head falls if the shoulders are supported, the torso falls if the body is supported .... Certainly the central nervous system is partly responsible for this behaviour, but the primary mode of its influence can be considered to be adjusting muscle 'spring-like' behaviour." [1, p384] Such instability that occurs below certain stiffness levels is also an issue in our system.

Our antagonistic control model corresponds to the *equilibrium point hypotheses* developed by motor control researchers in biomechanics. In this model, originally proposed by Anatol Feldman in 1966[4] for single joint systems, the agonist and antagonist muscles both generate torques around a joint with the equilibrium point of these torques defining a position[20, 9]. This theory would in principle allow the brain to avoid the complicated inverse dynamics problem[6]. Movements are instead achieved by gradually moving the equilibrium position between the movement end points. The limb will follow the equilibrium trajectory[6, 9]. "Experimental observations showed that the *path* between two points exhibited a measurable degree of stability, not just the end-point. Thus, the production of the movement appears to be accomplished by a progressive movement of the neurally-defined equilibrium posture, which has been termed a *virtual trajectory*."[8, p161, emphasis theirs]. In general, the position of the limb will vary from the neurally defined

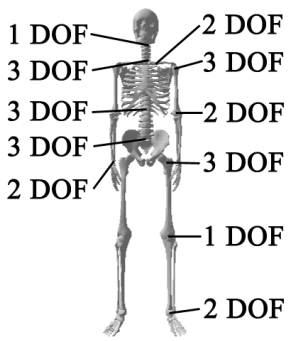


Figure 1: The degrees of freedom for the full skeleton.

virtual trajectory, which is the trajectory of the equilibrium point[9].

The opposing theory in biomechanics, known as the *impulse timing model*, suggests that motor programs instead directly generate short impulses of torque to control joint movement[20]. There is experimental evidence supporting both theories. Our work is compatible with the equilibrium point hypothesis, less because it is a plausible control model for human locomotion than that it provides a convenient computational approach for producing nuanced synthetic motion. However, we explicitly include gravity and external forces in our calculations when deciding how to tune our springs.

It should be noted that while our angular spring model is inspired by the antagonistic grouping of real muscles, it is still a substantial simplification of actual muscle behaviour (e.g., [17, 23]). Real muscles are linear force generators and the torques they generate vary throughout a movement due to variation in the moment arm. Muscle force is also affected by muscle length and velocity. Real muscles have more complex attachments, some spanning multiple joints and some having wide areas of attachment. The three element Hill based model is often used to model musculotendons in the biomechanics community. Our approach attempts to isolate another important feature of real muscle, tension control, while still maintaining computational simplicity.

## 3 Target Prototype

To illustrate our results, we currently employ two skeletal models in our prototype system. The first is a 47 DOF full body skeleton that can freely move in world space. It is shown in Figure 1 and the degrees of freedom at each joint are indicated. The second model is a 28 DOF torso model. It has the same degrees of freedom as the full body model, but the hip is fixed and there are no joints below the hip. Both models include 2 DOF clavicles. Many previous physics-based animation models have not included these joints[22, 3, 15, 24]. In our work, the clavicles have been found to make a significant contribution to the expressive impact of a character's movement.

The animation system is based on the Dynamic Animation and Control Environment (DANCE), a freely available plug-in based software package for building physics-based animation software[16]. The equations of motion are provided by a third party simulation package that uses an O(n) Kane's formulation [10]. This package also converts the external forces acting on the character to joint torques. When the equations of motion are combined with a semi-implicit Rosenbrock integrator [19], the torso model runs at one-tenth real time on a Pentium III 800MHz machine. This provides for a reasonable iterative workflow. Both models can generate motion kinematically in real-time. This is very useful for roughing out an animation.

## 4 Equilibrium Point Control

In physics-based animation, it is necessary to determine the torques that are required at a joint in order to produce the desired movement. Often this is done using a control law such as proportional derivative (PD) control. PD control can be thought of as a spring and damper system and is written as

$$\tau = k_s(\theta_{desired} - \theta) - k_d\dot{\theta}, \tag{1}$$

where $\theta$ is the current angle of the joint, $\theta_{desired}$ is the desired angle of the joint, $\dot{\theta}$ is the velocity of the joint and $k_s$ and $k_d$ are the spring and damper gains respectively.

When a control strategy such as PD control is used, there will in general be an error between the actual position of the joint and the desired position, even at steady state. This is due to the influence of gravity, which is pulling the joint away from the desired angle. The equilibrium point will be the angle at which the proportional term generates enough torque to balance the torque generated by gravity. This same phenomenon occurs in the equilibrium point hypothesis used to characterize human movement and is why the *virtual trajectory* may differ from the actual trajectory, even at steady state[9].

There is a tendency when using PD control to reduce steady state error by increasing the proportional gain. This can lead to very stiff characters. Assuming gravity is the only relevant external force, when PD control is at equilibrium, there will be no damping force, and the net torque will be zero:

$$0 = k_s(\theta_{desired} - \theta) - \tau_G, \tag{2}$$

where $\tau_G$ is the torque due to gravity acting on the limb. Thus the error is given by

$$error = \theta_{desired} - \theta = \frac{\tau_G}{k_s}\ . \tag{3}$$

If the error is to be reduced by increasing the gain, the result will necessarily be overly stiff characters. Consider a 5kg solid cylinder, such as an arm, that is 60 cm long and rotated around a pin joint at one end. The gain to keep it within 3 degrees of horizontal, representing angles as radians, is approximately 280. To keep it within 0.5 degrees, the required gain is 1680.

A relaxed, loosely moving character should have low gains, which means that there will be a large steady state error between the set-point of the PD controller and the actual position of the joint. It is unwieldy for an animator to try to anticipate the amount of error and provide a desired set point that will compensate for this error. For this reason, our system allows the animator to specify directly the actual angle for the joint. The final position of a joint will be the equilibrium point of all the internal and external forces acting on it. The system estimates the external forces and calculates the gains appropriately in order to realize the animator's requested angle with the requested amount of tension. This is what we refer to as *equilibrium point control*. The animator is given control over intuitive parameters, the actual angle and actual tension, and the system determines the gains and set points that will meet these constraints. Explicitly dealing with the external forces allows for accurate position control.

Currently we deal with external forces due to gravity and ground reaction forces, but it is straightforward to add other forces. For instance, the mass of an object the character is carrying could be added, so the character could use the appropriate force to lift it. This allows room for some fun effects as well. If the character was "told" the object was heavier than it is, he would use too much force and jerk the box up quickly. This is the same behaviour humans exhibit when they misestimate the weight of an object they are about to lift. Making the character aware of external forces is equivalent to giving him or her a limited tactile sense.

### 4.1 Antagonistic Formulation

In place of PD control, we employ an antagonistic actuator. Antagonistic control is implemented for each DOF using two ideal angular springs that are set in opposition to each other, and a damper. Unlike PD control, each spring has a fixed set point that remains constant throughout an animation. These set points are placed past the joint limits for the DOF. The control law is written as

$$\tau = k_L(\theta_L - \theta) + k_H(\theta_H - \theta) - k_d\dot{\theta}, \tag{4}$$

where $\tau$ is the torque generated, $\theta$ is the current angle of the DOF and $\dot{\theta}$ is its current velocity. $\theta_L$ and $\theta_H$ are the spring set points which serve as endpoints for the motion, $H$ and $L$ referring to high and low limits, $k_L$ and $k_H$ are the spring gains, and $k_d$ is the gain on the damping term. The tension or stiffness of the joint is taken as the sum of the two spring gains, $k_L + k_H$. The desired angle is not explicitly shown in the actuator, but is controlled by the value of the two gains.

Joint movement is achieved by varying the equilibrium point of a joint, which is the point at which all forces acting on the joint sum to zero. This is done by adjusting the gains on the two springs. At an equilibrium position, both the damping and the net force will be zero. The force on the joint is then

$$0 = k_L(\theta_L - \theta_{eq}) + k_H(\theta_H - \theta_{eq}) + ExternalForce, \tag{5}$$

where $\theta_{eq}$ is the equilibrium point and $ExternalForce$ represents gravity and all other known external forces acting on the system. For any given equilibrium angle, the gains that will achieve it all lie on a line in $k_L$-$k_H$ gain space given by

$$k_H = k_L\frac{\theta_L - \theta_{eq}}{\theta_{eq} - \theta_H} - \frac{ExternalForce}{\theta_H - \theta_{eq}}\ . \tag{6}$$

This line is referred to as the *isoangle* line. Notice that the slope of the line depends only on the limits and the desired angle and that the external force acts to vary the $y$ intercept of the line. Also, because $\theta_{eq}$ is bounded, the slope of the line will always be positive. Since we want the springs to only act in tension, gains are only chosen from the positive quadrant of the $k_L$-$k_H$ space.

We draw particular attention to the linearity of the isoangle relation, and that it depends only on the gains, current joint angle and joint limits. This implies that the tension on any joint, given initial gains, can be adjusted without needing to know the external forces.

The antagonistic formulation reflects the organization of real muscles. This approach would remain suitable even as more complex muscle models are introduced. The formulation permits easy tension adjustment by moving along the linear isoangle curve. It also makes explicit that the rest angle will necessarily be an equilibrium point determined by the spring gains.

### 4.2 PD Formulation of Equilibrium Control

Since the current version of antagonistic control uses linear springs, there is a mathematical equivalence between PD control and antagonistic control. For equivalent values, they will show exactly the same response to error. This is shown in the Appendix. Furthermore, equilibrium point control can be implemented using a PD actuator by rethinking the meaning of the desired angle set point in the PD control law. Instead of using the set point as the desired angle, the set point must be treated as a free parameter that is varied so that at equilibrium, the angle of the joint will be the angle requested by the animator.

We still prefer direct antagonistic control because it is more consistent with animal systems and provides a more natural parameterization. The isoangle line for PD control is a function of stiffness and set point given by:

$$\theta_{eq} = \theta_{desired} - \frac{\tau_G}{k_s}. \tag{7}$$

The isoangle line is a hyperbolic function of the two terms and its shape depends on the external forces. For antagonistic control, the isoangle is a linear function of the two gains.

A person authoring a controller that must deal with an external disturbance is given two intuitive options when using antagonistic control: the spring acting in the direction of the disturbance can be relaxed to reduce torque in the disturbance direction or the spring opposing the disturbance can be tightened to create a torque to resist the disturbance. Tensing or relaxing a joint to deal with a disturbance are the two options humans use, when for instance making balance adjustments(e.g. [5]). Achieving tension and force changes with PD control is less intuitive, and can include moving the set point from one side of the actual equilibrium point to the other.

Significantly, with antagonistic control, it is also possible to resist a disturbance simply by increasing the tension of the joint in order to reduce the error. This behaviour has been observed experimentally in humans[8]. Increasing the stiffness will not generally work for PD control. To make this clear, consider a limb under loose PD control that at steady state is leaning below its set point due to the influence of gravity. If a force hits the limb from below, it will move the limb closer to the set point. Increasing the gain on the PD controller will not restore the arm to its original location, but pull it closer to the set point. Provided the disturbance was not strong enough to knock the arm past the set point, increasing the stiffness of the spring has exactly the wrong effect: it moves the limb further in the direction of the disturbance, away from the desired equilibrium point and closer to the set point. By definition, gain in antagonistic control is varied along the current isoangle line. As the gain is increased in response to a disturbance, the equilibrium point of an antagonistic controller moves closer to the desired equilibrium point, equaling it in the limit. Increasing the gain therefore acts to resist the disturbance. With PD control, increasing gains moves the joint closer to the set point, not the desired equilibrium point. To achieve the same effect using an equilibrium point variation of PD control, the gain and set point would need to be varied in tandem. It is impossible to calculate how much to vary the set point without knowing the size of the disturbance.

### 4.3 Instability at Low Gains

In the presence of gravity, certain low gain configurations can be unstable. This is because the torque due to gravity is proportional to $\sin\theta$, where $\theta$ is the worldspace angle of the limb (0 is up) and this can lead to several equilibrium points in the range $[-\pi..\pi]$. Consider a limb controlled by PD control at equilibrium. (The situation is the same for antagonistic control, but the PD analysis is simpler.) The net torque acting on it will be zero as given by

$$0 = k_s(\theta - \theta_{desired}) - rmg\sin\theta, \tag{8}$$

where $r$ is the distance from the joint to the centre of mass of the limb, $m$ is the mass of the limb and $g$ is the acceleration due to gravity. This equation represents the sum of a linear spring force and a sinusoidal gravity force. For small values of $k_s$ (low gains), the slope of the line will be small compared to the magnitude of the sine function. For a given gain and $\theta_{desired}$, there may be several values for $\theta$ that make the equation zero. In other words, there may be multiple equilibrium points within the limits of the joint. This behaviour is shown for low and high values of gain $k_s$ in Figure 2. Notice, that for low gain, there are three equilibrium points where the torque on the joint will be zero. For high spring stiffness $k_s$, there is only one possible equilibrium point.

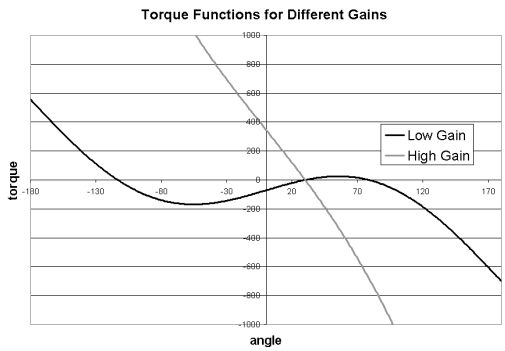


Figure 2: Comparison of the net torque for a limb given high and low gains: $\tau = k_s(\theta - \theta_{desired}) - rmg \sin\theta$. There are three zero torque equilibrium points for the low gain case.

If the limb is at an equilibrium point that is higher in world space than another equilibrium point, the limb will fall to the lower equilibrium point given any slight disturbance. This of course is also a feature of the antagonistic representation, and is also characteristic of people[1]. The isoangles curves for both PD and the antagonistic control cross at low values of gain where there is instability. It is possible to test for unstable low gain configurations and thus avoid them by increasing the gain, but it is important for users of the system to be aware of this phenomenon.

# 5 Shape Control

The "shape" or envelope of a motion refers to how speed is varied over the duration of a motion. This can be plotted as joint position vs. time. Some motions start quickly and end slowly, such as a careful reaching motion. Other motions may start slowly but speed up at the end, like a dabbing motion or a punch. Gentle motions often ease-in toward the end position, while more aggressive striking motions will maintain a high velocity near the end. Motions can be exaggerated by either recoiling backwards before starting the forward path of the motion or overshooting the end position before returning to it.

The shape of a motion is very important for the expressive impact it makes. This has not been explicitly dealt with in previous work in physically based animation. The two common solutions are either to use pure actuated movement or to track spline curves. Actuated movement involves step changes in the set point of a PD controller. The controller will then generate a force to negate the error and move to a new equilibrium point. This movement has a characteristic shape, moving quickly at the beginning when the error is large and slowing down near the end of the movement. It can be appropriate for some types of movements, but looks out of place for gestures like a wave. Spline following simply uses the controller to track a predefined motion curve. This approach is limited by the quality of the input curve.

Our system allows the animator to control a motion's shape in two ways. First, a shape curve can be associated with the transition to a pose that the system will track. Second, tension parameters may be controlled that indicate how closely the curve should be followed. This combination proves to be very flexible and powerful in practice. An animator can precisely control a movement when that is required. Often however, it is sufficient to provide a very simple trajectory and a satisfying shape can be achieved by specifying a tension change that should occur during the motion. It appears easier to generate a desired motion by varying the tension parameters than by trying to anticipate the correct transition curve. Movement is a very natural activity for people, so they can easily develop an intuition about the tension changes they make during a movement. They simply need to make the movement and pay attention to the changes in their body. Trajectory curves are a more abstract concept to work with.

## 5.1 Transition Functions for Shape Control

To control the shape of a motion, animators can associate a cubic Hermite curve with a pose. The curve is parameterized on [0,1] to correspond to progress in time and progress toward the end point. In practice, these transition functions have proven to be sufficiently flexible to allow an animator good control over shape. It allows a wide range of trajectories including recoil and overshoot effects. Animators can also reparameterize the time domain using standard techniques to make, for example, the relative transition time exponential. Transition functions also allow the system to compute kinematic previews, which are very useful in roughing out an animation. These functions are also used when the system needs to estimate a future state of the character.

## 5.2 Updating Gains

Given the control law and the transition functions that are to be tracked, the question remains as to how to update the gains on the springs to achieve the desired motion. The update process must take into account the external forces acting on the character to effect equilibrium control. This is done in two ways in the system: by sampling or by prediction, and they can be used together.

The sampling method takes samples of the forces acting on the figure at a given frequency, converts them to torques and recalculates the two actuator gains so that the equilibrium point at the sample time will be the angle indicated by the transition function. This approach is most appropriate for situations where forces may change in unexpected ways or if the force model used for a given external force does not lend itself to predictive use. In our animations, sampling is used for joints in the lower body that will experience significant variation in ground reaction forces, whereas the predictive approach is used for free upper body movements.

The predictive approach determines a set of gains that match the starting state of the pose, a set of gains that will achieve the end state of the pose and then simply transitions between the two sets of gains to achieve the desired motion. The starting gains can be the gains that are currently active in the system, or they can be recalculated. At the start of a pose, the external forces can be determined by querying all the controllers delivering external forces to the character and converting these forces to torques. For free upper body movements, the most significant force will be gravity. The skeleton state is then estimated for the endpoint. This is done by looking at all the poses that will be active at the end time and consulting their transition functions to determine an estimate of the joint angles for this time. Given the end state, the forces are again determined by consulting the external force controllers and torque estimates are determined. With the external forces and the equilibrium angles, the isoangle lines in gain space can be calculated that will achieve the two end points.

There are arbitrarily many ways to transition between the two lines in $k_L$-$k_H$ space. The path chosen will affect the shape of the resultant motion. By default, a transition path is chosen that keeps the joint stiffness constant. As stiffness is defined as the sum of the two gains, this corresponds to a line with slope $-1$ in $k_L$-$k_H$ space. For sufficiently high gains, if the transition line is traversed at constant speed, the resulting motion will be linear. Refer to Appendix 8.2. A line with slope $-1$ is drawn between the two isoangle lines, starting at the gain specified by the animator. During the motion, gain values are determined by interpolating along this line.

Progress along the line is determined by the user supplied transition function. This produces gains over time that will generate a resulting motion that has the same shape as the transition function. With increasing gain, the provided trajectory will be more closely tracked.

The same approach is used for transitioning ball joints represented by quaternions, except that the progress between the end points must be scaled. If the progress between the start and end (i.e., zero and one) is represented by $p$ and the angle between the two end quaternions is $\theta_{span}$, the update rule is

$$p = \sin p\theta_{span}. \tag{9}$$

Due to the estimates made for the end state, small errors may occur in the end position if the estimates are not accurate. This is consistent with at least some forms of human behaviour such as reaching, where humans often have to make small corrections to position at the end of a motion.

In some situations it is useful not to balance for external forces, and animators can select this option. This is particularly useful if the external forces will have a known effect the animator wishes to take advantage of. For instance, an animator may wish to loosely hold a joint straight, such as a wrist, and allow the effect of gravity to cause it to sway from side to side during a movement.

### 5.3 Tension and Relaxation Control

The power of the system is that very often, desired results can be achieved by specifying a very simple transition function say an ease-in ease-out curve or linear function – and shaping the motion almost completely through tension changes.

The amount of tension in a joint is controlled by the total gains on the two opposing springs. As discussed, for fixed angles the tension of a joint can be varied by shifting gain values along the isoangle line. It is also useful to vary the tension during a movement. For instance, an arm gesture may start by being tightly controlled and end with the arm swaying loosely at the character's side.

An animator can provide tension values for the start and the end position for each DOF in a pose. These are used when the start and end gains are calculated for the motion. If the gains are calculated by resampling the external forces, the gain value at each sample point is determined by the animator's input. For predictive updating, the transition between the end gains can be affected in two ways. The first method draws a line between the two gains and transitions along this line. The second method determines a curved line that will more closely track the provided trajectory.

Lines in gain space having a slope of other than $-1$ will induce a warping on the trajectory. Generally speaking, transitioning from a high gain to a lower gain will cause the motion to start slowly and accelerate towards the end. Moving from a low gain to a high gain will cause the motion to accelerate quickly at the beginning and ease in toward the end. This is shown graphically in Figure 3. Often this warping is enough to provide the desired shape for the motion. A linear transition function can be combined with a tension change and the resulting animation will have the desired appearance. This is especially true for posture changes. The animator is still free to use arbitrary transition functions to warp the motion if additional control is needed.

The second method allows the animator to more closely track the transition function. The transition function can be used to determine desired angles at each moment in time, and from this and the joint limits, the slope of the isoangle line can be calculated for that moment. With point and slope estimates, an isoangle line can be computed at each update time. Tension is then varied by selecting a point on this line that gives the requested stiffness.

With high gains, the resulting motion will track the transition functions closely. As gains are reduced, the momentum of the limbs

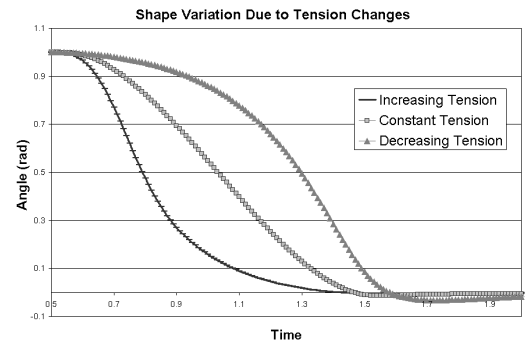


Figure 3: Varying tension during a motion warps the motion's shape. The trajectory function is an ease-in-ease-out curve.

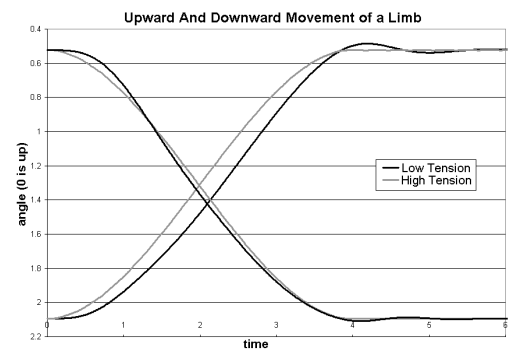


Figure 4: Upward and downward motions for a limb comparing high and low gains.

starts to contribute more strongly to the motion and the trajectory deviates from the transition function. Figure 4 shows examples of both downward and upward movements, each at two different gains. The high gain curve precisely tracks the kinematic transition function. For the downward motion, the lower gain function at first lags behind the high gain function as it takes longer to overcome the inertia of the limb. During the middle section of the movement, the limb leads the kinematic curve under the effect of gravity. The two motions reach the end point at approximately the same time, but the lower gain motion slightly overshoots the desired position. This overshoot is often visually desirable. The results are similar for an upward motion. Once again the high gain motion precisely tracks the kinematic transition function. The lower gain motion again exhibits a lag at the beginning and an overshoot at the end. Since gravity is now acting against the motion, the lower gain transition trails the motion throughout.

Joints that are less stiff will also be more affected by the movement of adjacent limbs. The antagonistic controllers do not compensate for these transference forces, so the effect the forces have depends solely on the tension in the neighbouring joint. Transitory effects will become much more apparent as the character relaxes and this can yield richer movement.

In an authoring system, it is important to give an animator sufficient control over the motion in order to achieve the effects she desires. This control is not useful, however, if it is unintuitive. Controlling gain parameters can be unintuitive. Rather than automating the gain setting and limiting animator control, a calibration method was established that gives the animator some intuition for what gain values mean. The calibration process is performed on the simulated torso skeleton and makes the entire skeleton very stiff except for

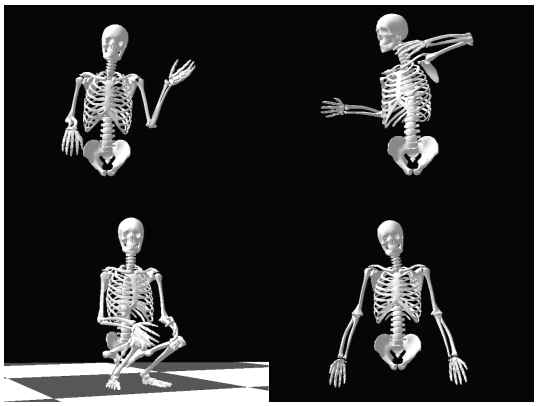

Figure 5: A montage of frames from the animations presented in this paper.

one DOF. The skeleton is then perturbed by pushing it at an end effector appropriate for the DOF with a force of 45N. This corresponds to a firm push. The test is repeated for several different gain values for the DOF and then for the rest of the DOFs in the torso. The animator can then access a set of curves that show the maximum deformation that a push will give at each DOF as a function of gain. These curves have hyperbolic shape and also indicate where instabilities start to occur. This information provides a clear picture of how stiffness values range across different joints of the body, as well as the effect of changing the gain, and allows the parameters to be used in a knowledgeable way.

Damping is the final control parameter in the system. It can be varied to reduce overshoot and transitory effects in a motion. It also creates some lag. If damping is too low and the movements are quite fast, ringing can occur where the motion exceeds the equilibrium point during the transition and then gets pulled back toward it. Increasing the damping can cure this. Generally, damping values that are approximately one tenth of gain values work well.

# 6 Results

The animation system outlined here has been applied to a number of different tasks including posture adjustments, gesture control and anticipation effects. A montage of images from these animations is shown in Figure 5. The results are described here and all animations mentioned in the paper are available online at: http://www.dgp.toronto.edu/~neff All the animations described consist of a small number of poses, the richness coming from the impact of tension variation on the dynamics simulation. Some iteration was involved in generating the animations. In general, less iteration is required when tension changes are the dominant shaping function than when the transition functions are carefully tweaked.

## 6.1 Gestures

Gestures are a prime candidate for expressive variations in tension. An animation is shown that contrasts a relaxed, flamboyant character and a police officer giving the direction “You should go left.”. The police officer’s movements are quick and precise. The flamboyant character’s movements are loose and flowing. His arms swing slightly at his sides, his wrists flick, the movements accelerate and decelerate based on tension changes. All of these effects are due to tension modeling, not the kinematic trajectories.

A wave motion was generated using pure actuated control, kinematically, using tight control and varying the tension through the movement. The actuated control is quite unnatural as the movement accelerates quickly and then eases in toward the end of each section of the motion. The kinematic wave is substantially better, but still looks stiff and not particularly lifelike. Using tight control produces an animation that looks very similar to the kinematic motion. Varying tension produces a significantly more lifelike wave. The wrist is allowed to move passively creating a swaying hand motion, the force transference generates some movement in the collar bone, the trajectories are smoothed and warped due to the influence of gravity and momentum, the arm sways slightly at the character’s side at the end of the movement. These effects rely on being able to vary the tension during a movement. For instance, the arm is relaxed as it is brought to the character’s side, allowing it to sway slightly at the end. If instead the gains were reduced at the start of the motion, the arm would fall in a physically realistic, but unnatural way. This animation uses kinematic trajectories for shape control, but enhances them by adding tension variation.

## 6.2 Postures

Posture adjustments are ideal candidates for tension control. If a tired character is to sag downwards the correct motion shape can be achieved by simply providing a linear transition function and reducing the tension during the motion. Other posture adjustments can be made in a similarly straightforward manner. Importantly, the animator retains the ability to provide arbitrary transition functions to customize a motion.

Examples are presented that contrast a formal greeting nod with a tired character sagging downwards. A shrug is also shown.

## 6.3 Anticipation

Recall the soccer ball that was flying toward you in the story that opened the paper. You changed your tension because you noticed the ball coming. If you had not, you would have remained relaxed. People make tension changes when they anticipate an event. This affects how they react. To demonstrate this a character is shoved with different forces and different amounts of tension in his body. The effect of the shove becomes quite muted as he braces for impact.

Another animation combines tension changes and gestures as the character reacts to being pushed from behind.

## 6.4 Crouch

The previous animations were generated on the torso model. The system also allows full body animation, interaction with ground forces and provides balance control. A crouch animation is included which demonstrates this. This animation uses an automatic balance controller that is based on ideas from [22].

# 7 Conclusion and Evaluation

A system has been presented that allows very flexible character control in a physically based animation framework. It allows the shape of movements to be varied and the tension of a character to be varied within a movement and across movements. Tension control, when combined with an underlying dynamics simulation, is seen to be an effective method to capture many of the subtleties of human motion. This allows richly nuanced motion to be created that requires only a few poses.

The animator interface is still lower level than is desirable and requires a significant amount of input. Future work will investigate higher level interaction techniques for animators that still maintain sufficient control. Another avenue of future research is the creation

of more realistic muscle models that capture additional aspects of human movement.

# 8 Appendix

## 8.1 Equivalence of PD and Antagonistic Control

The equivalence between PD and antagonistic control can be shown by multiplying out the two equations, including external forces, and equating the terms. The net torque with PD control is given as

$$\tau_{net} = k_s\theta_{desired}k_s\theta k_d\dot{\theta} + externalTorque, \tag{10}$$

and the antagonistic control equation becomes

$$\tau_{net} = k_L\theta_L - k_L\theta + k_H\theta_H - k_H\theta - k_d\dot{\theta} + externalTorque. \tag{11}$$

The damping term and external force terms are the same. Equating the $\theta$ terms from both equations gives

$$-k_s\theta = -(k_L + k_H)\theta, \tag{12}$$

$$k_s = k_L + k_H\,, \tag{13}$$

$$k_H = -k_L + k_s\,. \tag{14}$$

Equating the set point terms gives:

$$k_s\theta_{desired} = k_L\theta_L + k_H\theta_H \tag{15}$$

$$k_L = k_s\frac{\theta_{desired} - \theta_H}{\theta_L - \theta_H} \text{ subst. 14}\,. \tag{16}$$

For $\theta_{desired}$ in between $\theta_H$ and $\theta_L$, $k_L$ will be positive and less than $k_s$. This means that for any PD settings, there are a pair of positive values of $k_L$ and $k_H$ such that the antagonistic controller and the PD controller will have the same error response. Note, this is not the case if the linear springs in the antagonistic formulation are replaced with non-linear functions. Also note that the stiffness function used in antagonistic control is $k_L + k_H$, which equals the stiffness gain $k_s$ in PD control.

## 8.2 Linear Transitions

An initial point in $k_L$-$k_H$ space is indicated as $L_1$, $H_1$. The equation passing through this point with slope $-1$ is:

$$H = -L + L_1 + H_1. \tag{17}$$

Considering gravity forces, at equilibrium, the antagonistic control law gives:

$$L(\theta_L - \theta) + k_H(\theta_H - \theta) = -rmg\sin\theta \tag{18}$$

for a joint. For sufficiently high gains, the gain terms will dominate the gravity force and the relation can be approximated as

$$s(\theta_L - \theta) + k_H(\theta_H - \theta) \approx 0\,. \tag{19}$$

Rearranging and substituting 17 gives

$$\theta = \frac{s\theta_L + H\theta_H}{L + H}, \tag{20}$$

$$= \frac{L\theta_L + (-H + L_1 + H_1)\theta_H}{L(-L + L_1 + H_1)}, \tag{21}$$

$$= L\frac{\theta_L - \theta H}{L_1 + H_1} + \theta_H\,. \tag{22}$$

Since $L$ is multiplied by a constant, a linear change in $L$ will produce a linear change in $\theta$. Therefore, for high gains, linear movements along a line in $k_L$-$k_H$ space with slope $-1$ will produce linear motions.


# Acknowledgements

Many thanks go to Petros Faloutsos and Victor Ng-Thow-Hing for writing DANCE, making it available and providing help and useful feedback during this work. Valuable discussions with Joe Laszlo, Dave Mould and Michiel van de Panne greatly benefited the work. Chris Trendall, Joe Laszlo and David Torre helped facilitate the submission. This work was partially supported by an NSERC post graduate scholarship.


# References


[1] Gunnar B. J. Andersson and Jack M. Winters. Role of muscle in postural tasks: Spinal loading and postural stability. In Jack M. Winters and Savio L-Y. Woo, editors, *Multiple Muscle Systems: Biomechanics and Movement Organization*. Springer-Verlag, New York, 1990.

[2] Diane M. Chi, Monica Costa, Liwei Zhao, and Norman I. Badler. The emote model for effort and shape. *Proceedings of SIGGRAPH 2000*, pages 173–182, July 2000. ISBN 1-58113-208-5.

[3] Petros Faloutsos, Michiel van de Panne, and Demetri Terzopoulos. Composable controllers for physics-based character animation. *Proceedings of SIGGRAPH 2001*, pages 251–260, August 2001. ISBN 1-58113-292-1.

[4] A. G. Feldman. Functional tuning of the nervous system with control of movement or maintenance of a steady posture - ii. controllable parameters of the muscles. *Biophysics*, 11(3):565–578, 1966.

[5] R. C. FitzPatrick, J. L. Taylor, and D. I. McCloskey. Ankle stiffness of standing humans in response to imperceptible perturbation: Reflex and task-dependent components. *Journal of Physiology*, 454:533–547, 1992.

[6] Tamar Flash. The organization of human arm trajectory control. In Jack M. Winters and Savio L-Y. Woo, editors, *Multiple Muscle Systems: Biomechanics and Movement Organization*. Springer-Verlag, New York, 1990.

[7] Jessica K. Hodgins, Wayne L. Wooten, David C. Brogan, and James F. O'Brien. Animating human athletics. *Proceedings of SIGGRAPH 95*, pages 71–78, August 1995. ISBN 0-201-84776-0. Held in Los Angeles, California.

[8] Neville Hogan. Mechanical impedance of single- and multi-articulator systems. In Jack M. Winters and Savio L-Y. Woo, editors, *Multiple Muscle Systems: Biomechanics and Movement Organization*. Springer-Verlag, New York, 1990.

[9] Neville Hogan and Jack M. Winters. Principles underlying movement organization: Upper limb. In Jack M. Winters and Savio L-Y. Woo, editors, *Multiple Muscle Systems: Biomechanics and Movement Organization*. Springer-Verlag, New York, 1990.

[10] Michael G. Hollars, Dan E. Rosenthal, and Michael A. Sherman. *SD/FAST User's Manual*. Symbolic Dynamics Inc., 1994.

[11] E. Kokkevis, D. Metaxas, and N. I. Badler. User-controlled physics-based animation for articulated figures. In *Computer Animation '96*, pages 16–26, June 1996. ISBN 0-8186-7588-8.

[12] Alexis Lamouret and Marie-Paule Gascuel. Scripting interactive physically–based motions with relative paths and synchronization. In *Graphics Interface '95*, pages 18–25, May 1995.

[13] Joseph Laszlo, Michiel van de Panne, and Eugene Fiume. Interactive control for physically-based animation. *Proceedings of SIGGRAPH 2000*, pages 201–208, July 2000. ISBN 1-58113-208-5.

[14] Maja J Mataric, Matthew Williamson, John Demiris, and Aswath Mohan. Behavior-based primitives for articulated control. In *Proceedings, From Animals to Animats 5, Fifth International Conference on Simulation of Adaptive Behavior*, pages 165–170. MIT Press, 1998.

[15] Maja J. Mataric, Victor B. Zordan, and Matthew M. Williamson. Making complex articulated agents dance: An analysis of control methods drawn from robotics, animation and biology. *Autonomous Agents and Multi-Agent Systems*, 2(1), july 1999.

[16] Victor Ng-Thow-Hing and Petros Faloutsos. Dynamic animation and control environment, 1999. http://www.dgp.toronto.edu/dgp/software/dance/dance.html.

[17] Benno M. Nigg and Walter Herzog, editors. *Biomechanics of the Musculoskeletal System*. John Wiley and Sons, Ltd., second edition, 1999.

[18] Ken Perlin. Real time responsive animation with personality. *IEEE Transactions on Visualization and Computer Graphics*, 1(1):5–15, March 1995. ISSN 1077-2626.

[19] William H. Press, Saul A. Tukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 2nd edition, 1992.

[20] Richard A. Schmidt and Timothy D. Lee. *Motor Control and Learning: A Behavioral Emphasis*. Human Kinetics, Champaign, IL, third edition, 1999.

[21] Michael van de Panne, Ryan Kim, and Eugene Fiume. Virtual wind-up toys for animation. *Graphics Interface '94*, pages 208–215, May 1994. Held in Banff, Alberta, Canada.

[22] Wayne L. Wooten. *Simulation of Leaping, Tumbling, Landing, and Balancing Humans*. Ph.D. dissertation., Georgia Institute of Technology, 1998.

[23] Felix E. Zajac. Muscle coordination of movement: A perspective. *Journal of Biomechanics*, 26:109–124, 1993. Supll. 1.

[24] Victor B. Zordan and Jessica K. Hodgins. Tracking and modifying upper-body human motion data with dynamic simulation. *Computer Animation and Simulation '99*, September 1999. ISBN 3-211-83392-7. Held in Milano, Italy.