Heads Up!
Biomechanical Modeling and Neuromuscular Control of the Neck

Sung-Hee Lee[*] Demetri Terzopoulos[†]
University of California, Los Angeles

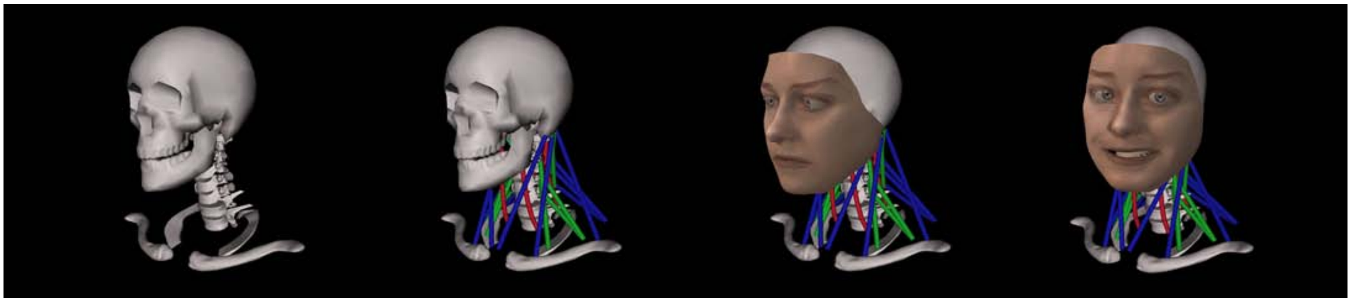


Figure 1: Our biomechanical system comprises a skeleton, muscles, neural control system, and expressive face.

Abstract

Unlike the human face, the neck has been largely overlooked in the computer graphics literature, this despite its complex anatomical structure and the important role that it plays in supporting the head in balance while generating the controlled head movements that are essential to so many aspects of human behavior. This paper makes two major contributions. First, we introduce a biomechanical model of the human head-neck system. Emulating the relevant anatomy, our model is characterized by appropriate kinematic redundancy (7 cervical vertebrae coupled by 3-DOF joints) and muscle actuator redundancy (72 neck muscles arranged in 3 muscle layers). This anatomically consistent biomechanical model confronts us with a challenging motor control problem, even for the relatively simple task of balancing the mass of the head in gravity atop the cervical spine. Hence, our second contribution is a novel neuromuscular control model for human head animation that emulates the relevant biological motor control mechanisms. Incorporating low-level reflex and high-level voluntary sub-controllers, our hierarchical controller provides input motor signals to the numerous muscle actuators. In addition to head pose and movement, it controls the tone of mutually opposed neck muscles to regulate the stiffness of the head-neck multibody system. Employing machine learning techniques, the neural networks within our neuromuscular controller are trained offline to efficiently generate the online pose and tone control signals necessary to synthesize a variety of autonomous movements for the behavioral animation of the human head and face.

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

Keywords: neck animation, biomechanical modeling, hierarchical neuromuscular control, neural network learning, facial animation

1 Introduction

Biomechanics-based animation research continues to expand its horizons. In the important area of human modeling, substantial effort has been devoted to the physical simulation and control of complete anthropomorphic figures (see, e.g., [Faloutsos et al. 2001; Hodgins et al. 1995]). In an effort to improve realism, researchers have also been developing increasingly sophisticated biomechanical models of individual body parts, such as hands [Tsang et al. 2005; Albrecht et al. 2003], torsos [Zordan et al. 2004], and especially faces [Sifakis et al. 2005; Kähler et al. 2001; Lee et al. 1995]. Pacing this progress, multiple efforts have been directed at the modeling of individual muscles [Irving et al. 2004; Ng-Thow-Hing 2001; Chen and Zeltzer 1992], the preferred class of actuators for use in biomechanical modeling.

Given the voluminous literature on human body and facial modeling, it is surprising that the neck has been largely overlooked in computer graphics. This may be due in part to the complexity of cervical anatomy and biomechanics. Yet the realistic modeling of the neck is a significant problem in human animation, because the neck determines the global movement of the head and face relative to the body. Indeed, the neck plays a crucial role in supporting the mass of the head, balanced in gravity, atop the cervical spine while generating the controlled head movements that are essential to so many aspects of human behavior.

In this paper, we introduce the first biomechanical model of the human head-neck musculoskeletal system for computer animation. In particular, we model the head and each vertebra in the cervical spine as a dynamic rigid body with appropriate mass distribution and three rotational degrees of freedom (DOF), coupling the bones with joints to emulate the biological assembly of interest. The resulting articulated multibody system is actuated by contractile muscles. Each actuator is also modeled biomechanically as a simplified Hill-type muscle model, which is frequently used in biomechanics research. The complexity of the musculoskeletal model, especially its kinematic and muscular redundancy, which imitates that of its biological counterpart, confronts us with a challenging control problem. We believe that the best way to tackle this problem is via an approach inspired by biological motor control mechanisms, all the more so because our long-term goal is to create lifelike characters that are able to synthesize a broad range of human motions. Hence,

[*] www.cs.ucla.edu/∼sunghee
[†] www.cs.ucla.edu/∼dt

our second major contribution in this paper is a novel neuromuscular control model for human (head) animation that emulates the relevant biological motor control mechanisms.

A distinctive feature of the mammalian motor control architecture is that it is hierarchical [Kandel et al. 2000]—multiple neural organs, such as the cerebral cortex, basal ganglia, cerebellum, and spinal cord, participate in generating the signals finally transmitted by motor neurons innervating muscles. This suggests that simple, flat control strategies may be incapable of synthesizing a large repertoire of human motions. Hence, we take a hierarchical approach, proposing a bi-level motor control architecture whose lower level corresponds to *reflex* (or *feedback*) *control* in the human body, and whose upper level corresponds to *voluntary* (or *feedforward*) *control*. Our hierarchical head-neck controller provides the inputs to the numerous muscle actuators necessary to maintain the stability of the cervical spine and autonomously generate a variety of head movements for the behavioral animation of the human head and face.

A key technical contribution of this paper is the development of a voluntary controller that is able to control independently the *pose* and *tone* of the head-neck musculoskeletal system. By "tone", we mean the stiffness or tension of the musculoskeletal system, which humans can control by coactivating agonist and antagonist muscles. Our voluntary controller comprises a pose signal generator and a tone signal generator, the sum of whose outputs yields the voluntary, feedforward control signal. Meanwhile, the lower-level, reflex controller continually monitors the strain and strain rate of each muscle, generating an involuntary, feedback control signal such that the muscle can maintain its desired length in the presence of external force disturbances.

Our hierarchical control model has additional features of interest. The computational mechanisms underlying the implementation of the voluntary controller are artificial neural networks sustained by machine learning techniques. Neural networks are trained to generate the appropriate pose and tone control signals necessary for the musculoskeletal system model to synthesize a variety of autonomous humanlike movements for the behavioral animation of the head and face. The training data are precomputed by solving repeated optimal control problems. Aside from their structural resemblance to biological neural networks, our artificial neural networks are efficient feedforward controllers—once trained offline, they can do their online jobs orders of magnitude faster than attempting to solve the corresponding optimal control problems online.

Fig. 1 illustrates our implementation of the above ideas, and more, as a self-animating virtual human neck, head, and face. In a simulated physical environment with gravity, our autonomous system naturally selects, alters, and maintains head pose and gaze direction, and it can adjust its tone in response to external disturbances.

The remainder of this paper is organized as follows: Section 2 reviews relevant research in the graphics and biomechanics literature. Section 3 provides a functional overview of our face-head-neck animation system. Section 4 details our biomechanical musculoskeletal model. Section 5 develops our hierarchical, neuromuscular control framework, including the reflex and voluntary controllers, and the associated control learning algorithms. Section 6 reports selected results. Section 7 discusses our modeling approach vis-a-vis alternative schemes. Section 8 presents conclusions and proposes avenues for future work in our highly fertile domain.

2 Related Work

To our knowledge, there are no prior reports in the computer graphics literature on the biomechanical modeling and control of the neck. The closest related effort has been by Monheit and Badler [1991] who proposed a purely kinematic spine and torso model, where the total bending angle is distributed to each joint according to weighting parameters. The neck has been studied to some extent, however, in the biomechanics and neurophysiology literature. Keshner and Peterson [1995] investigated the multiple neurological mechanisms underlying human head stabilization. Vasavada et al. [1998] constructed a 3D human neck muscle model and measured the moment-generating capacity of each muscle. They visualized human neck motion in their work, but once again the movement is generated kinematically, with no dynamics.

Chen and Zeltzer [1992] introduced the biomechanical modeling of muscles for computer animation, modeling muscle tissue with large finite elements and simulating muscle deformation by applying a Hill-type force in the muscle. Parametric muscle models have been proposed that deform geometrically, and they have been used to simulate skin shape change due to the bulging of underlying muscles using kinematic [Scheepers et al. 1997; Wilhelms and Gelder 1997] and dynamic [Kähler et al. 2001] skin. Recently, more sophisticated muscle deformation methods have been proposed, such as B-spline solids [Ng-Thow-Hing 2001], invertible finite elements [Irving et al. 2004], and muscle strands [Pai et al. 2005]. We do not simulate solid muscles in this paper. Our muscle model is strictly a force generating uniaxial actuator, but it is more complex than those used by Lee et al. [1995] in their biomechanical face model or by Tu and Terzopoulos [1994] in their biomechanical fish model.

Albrecht et al. [2003] proposed an anatomy-based hand animation system where they modeled two types of muscles—geometric muscle for simulating muscle deformation and pseudo-muscle for actuating bones—but their controller is manually-tuned. Tsang et al. [2005] proposed a heuristic technique for solving the necessary muscle activation to acquire target poses for a muscle-actuated human hand model.

Komura et al. [2000; 1997] computed optimal feedforward muscle activation levels given several key poses of human lower extremities for solving inverse kinematics or "physiological retargeting" of the motion. These references and [Tsang et al. 2005] are relevant to our work in that they perform inverse dynamics to compute necessary muscle activation level for Hill-type muscle models. However, their controllers are not as comprehensive as ours, inasmuch as they disregard muscle coactivation and must solve expensive space-time optimization problems online, making them impractical for interactive, autonomous animation. Also [Tsang et al. 2005] and [Komura et al. 1997] disregard feedback control. It should be noted that inverse dynamics does not guarantee stability; in fact, inverse dynamics control without feedback control can easily become unstable even under the slightest disturbance.

Not surprisingly, neuromuscular control approaches are common in the biomechanics literature. With the advent of artificial neural networks, researchers have adopted the technique to the study of human motor learning. For example, Kawato et al. [1987] constructed a hierarchical neural network that learns inverse dynamics of a simple arm model. This forward simulation/learning model is biomimetic but computationally expensive. Kim and Hemami [1998] performed a similar study with a simplistic human head and torso model. In graphics, Yin et al. [2003] briefly mentioned the importance of neuromuscular control for animation, but they performed inverse dynamics analysis of mocap data, and used this as a feedforward control input. The control scheme itself is essentially computed torque control, a common technique in robotics. Grzeszczuk et al. [1998] applied artificial neural networks and the backpropagation learning algorithm to training feedforward controllers for dynamic objects, among them a locomotion controller for a biomechanical dolphin model.

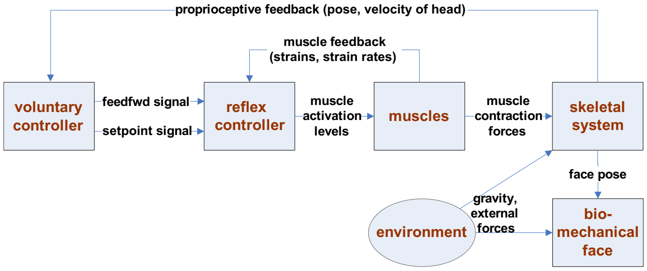


Figure 2: Face-Head-Neck System Architecture.

A unique feature of muscle is that its stiffness increases with increasing neural signal. Consequently, by coactivating agonist and antagonist muscles, humans and other animals can increase stiffness while maintaining pose. They effectively use such tone control to mitigate instability under external loads or to increase the accuracy of the limbs in motor tasks. It is also well known that coactivation occurs when humans learn new motions. Hogan [1984] studied tone (a.k.a. impedance) modulation by coactivating agonist and antagonist muscles. In computer animation, Neff and Fiume [2002] proposed a joint-actuated control technique in which they attached two opposing PD feedback controllers to every joint of an articulated anthropomorphic figure, controlling the tension and relaxation of the resulting body motion by modulating the two proportional feedback gains. Their work falls short of our richly muscle-actuated model in that it does not include feedforward control and its joint controllers cannot accurately model the characteristics and functions of real muscles, especially when these muscles span multiple joints as many neck muscles do.

3 Neck-Head-Face System Overview

Fig. 2 shows the overall architecture of our head-neck system model, which comprises the skeleton, muscles, and hierarchical controller. The voluntary sub-controller generates feedforward and setpoint control signals: The feedforward signal is generated to attain the desired pose and tone. The setpoint signal specifies the desired strain and strain rate of each muscle, as well as the magnitude of the feedback gain. Comparing the strain and strain rate against their desired values, the reflex controller generates a feedback signal and adds it to the feedforward signal, thus determining the activation level of each muscle. Given an input activation signal, each muscle generates a contraction force depending on its length and velocity. Finally, the skeleton produces articulated motion in response to the internal muscle forces and external environmental forces, such as gravity and applied forces. Physics-based animation is achieved by numerically integrating the equations of motion of the biomechanical model through time. Including control computations, our simulation runs about 10 times slower than real time on a PC with a 3.2 GHz Mobile Intel Pentium 4 CPU and 1 GB of RAM.

Although this paper does not dwell on facial animation, we have augmented the realism of our biomechanical head-neck model for the demonstrations that we present in Section 6 by coupling a biomechanical face model (the lower right box in Fig. 2) to the front of the skull as shown in Fig. 1. This expressive, behaviorally-capable face model [Terzopoulos and Lee 2004] is an improved version of the second-generation biomechanical model reported in [Lee et al. 1995]. Conceptually, the face model decomposes hierarchically into several levels of abstraction related to the (FACS) control of facial expression, the anatomy of facial muscle structures, the histology and biomechanics of facial tissues, as well as facial geometry and appearance. Like our biomechanical model of the neck, the face model is muscle-driven. Its 44 facial muscles are arranged in an anatomically consistent manner within the bottom layer of a synthetic facial soft tissue. The tissue is modeled as a lattice of uniaxial viscoelastic units assembled into multilayered prismatic elements with epidermal, dermal, sub-cutaneous fatty tissue, fascia, and muscle layers. The elements enforce volume preservation constraints and model contact response against the bone substrate. Expressive facial tissue deformations are animated by numerically simulating the physical response of the element assembly to the stresses induced by appropriately coordinated facial muscle contractions. The face simulation runs at real-time, interactive rates on the aforementioned PC.

Bone	Mass	k_s: x,z-axes	k_s: y-axis
Skull	3.5	50	25
C1–C7	0.21	50–70	25–35

Table 1: Physical parameters of the skeleton. The masses are in kilograms. The k_s quantities are in $\mathrm{N}\cdot\mathrm{m}/\mathrm{rad}$. The k_d are set to 10% of the corresponding k_s. The y axis is in the vertical direction.

4 Musculoskeletal Model

Our musculoskeletal model comprises a model of the skeleton and a model of the muscles of the neck, which we will describe in turn.

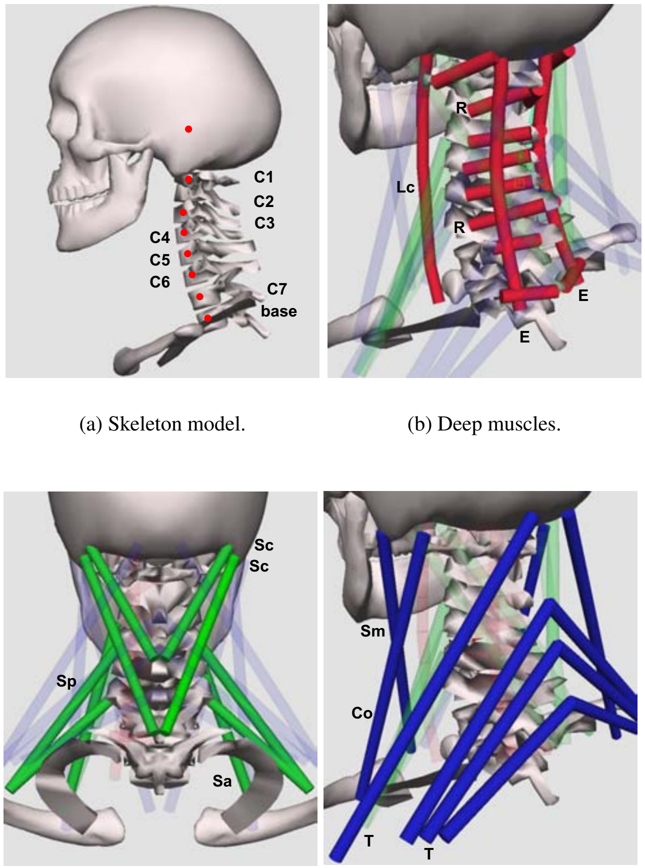
4.1 Skeleton Model

The relevant skeletal structure is modeled as an articulated multibody system. It includes a base link, seven cervical bones, C1–C7, and a skull, as shown in Fig. 3(a). In the human spine, disks are sandwiched between adjacent vertebrae, allowing 6-DOF motion. By carefully locating pivot points as in [Kapandji 1974], we simplified each joint to a 3-DOF rotational joint. To each joint angle, we attach a rotational damped spring in order to model the stiffness of the ligaments and disks, as follows: $\tau_s = -k_s(q - q_0) - k_d\dot{q}$, where q is the joint angle, q_0 is the joint angle in the natural, rest configuration, k_s is the spring stiffness, and k_d is the damping coefficient. The linear damping increases the stability of the system. Table 1 specifies the physical parameters of the skeleton.

The equations of motions of the skeletal system are

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{K}_s\mathbf{q} + \mathbf{K}_d\dot{\mathbf{q}} - \mathbf{P}(\mathbf{q})\mathbf{f}_P = \mathbf{P}(\mathbf{q})\mathbf{f}_C + \mathbf{J}(\mathbf{q})^T\mathbf{f}_e, \tag{1}$$

where $\mathbf{q}$, $\dot{\mathbf{q}}$, and $\ddot{\mathbf{q}}$ are 24-dimensional vectors containing all the joint angles (generalized coordinates), the angular velocities, and the angular accelerations, respectively. Since our muscle model is massless and purely force-based, the mass of the head is incorporated into the skull and the mass of the neck is distributed among the cervical vertebrae. $\mathbf{M}(\mathbf{q})$ denotes the inertia matrix of the skeleton. The vector $\mathbf{c}(\mathbf{q},\dot{\mathbf{q}})$ represents the Coriolis forces, centrifugal forces, and gravity. The diagonal stiffness $\mathbf{K}_s$ and damping $\mathbf{K}_d$ matrices are due to the aforementioned rotational springs. Since the equations of motion (1) are expressed in joint space, $\mathbf{J}(\mathbf{q})$ is the Jacobian matrix that transforms the external force $\mathbf{f}_e$ into joint torques. The muscle forces are divided into passive, elastic forces $\mathbf{f}_P$ produced by the muscles' material properties as they are stretched, and active, contractile forces $\mathbf{f}_C$ generated by the muscles in response to the neural control signal. The moment arm matrix $\mathbf{P}(\mathbf{q})$ maps muscle forces to joint torques, and it is computed using the principle of virtual work



(a) Skeleton model.

(b) Deep muscles.

(c) Intermediate muscles.

(d) Superficial muscles.

Figure 3: Musculoskeletal model. (a) The red dots represent the pivots of the eight joints of the cervical column. The pivots of vertebra C2 to C7 are in their supporting bones. Geometric mesh data were acquired from www.3dcafe.com. The deep muscle layer (b), intermediate muscle layer (c), and superficial muscle layer (d) of the neck are shown. Table 2 details the muscles and attachments.

[Delp and Loan 1995], as detailed in Appendix A. We compute $\ddot{\mathbf{q}}$ in (1) using Featherstone's dynamics algorithm and numerically integrate through time to obtain $\dot{\mathbf{q}}$ and $\mathbf{q}$ using the explicit Euler method.

4.2 Muscular Structure

There are more than 20 types of muscles in the neck, and there are many muscles of each type. Individual muscles often have multiple origins and insertions. Since it would be difficult and computationally very costly to model all the muscles accurately, we were motivated to reduce the number of muscles modeled. In an effort to minimize the total number of actuators in the synthetic musculoskeletal system, we first attempted to model only the major superficial muscles of the neck. We discovered, however, that even though these muscles outnumbered the total number of degrees of freedom of the system, the system was uncontrollable, apparently because most of the major muscles span multiple bones. The solution was to dauntlessly emulate the considerable muscular redundancy of the target biological system.

Layer	Muscle	#m	Origin / Insertion	w
Deep	Longus colli (Lc)	16	adjacent vertebrae (anterior vertebral bodies)	1.0
	Erector (E)	16	adjacent vertebrae (behind transverse pro)	1.0
	Rotator (R)	16	adjacent vertebrae (transverse pro / spinous pro)	1.0
Intermediate	Scalenus anterior (Sa)	4	base (lateral) / C5 C3 (transverse pro)	2.0
	Scalenus posterior (Sp)	4	base (lateral) / C6 C4 (transverse pro)	2.0
	Splenius capitis (Sc)	4	C7 C5 (spinous pro) / skull (superior nuchal line)	2.0
Superficial	Sternomastoid (Sm)	2	base (sternum) / skull (mastoid pro)	3.0
	Cleidooccipital (Co)	2	base (clavicle) / skull (superior nuchal line)	3.0
	Trapezius (T)	8	base (posterior) / C6 C4 C2 (behind spinous pro) skull (external occipital prot)	3.0

Table 2: The subset of neck muscles that are modeled and their origins/insertions. Legend: number of muscles (#m); strength weight factor (w); process (pro); protuberance (prot).

Consulting references on anatomy [Warfel 1985; Kapandji 1974], we incorporated 72 individual muscles into the musculoskeletal model, as shown in Fig. 3(b)–(d). The neck muscles are arranged in three layers—deep, intermediate, and superficial. In the deep layer (Fig. 3(b)), there are a total of 48 muscles, which improve controllability. Six muscles are attached across each cervical joint, such that they cover the 3 DOFs of the joint. This increases, if not guarantees, controllability and affords greater freedom to model the major muscles of the intermediate and superficial layers, each of which include 12 muscles arranged as shown in Fig. 3(c) and (d).

Notwithstanding the rather large number of modeled muscles, note that we have disregarded many of the muscles of the neck, such as the muscles attached to the hyoid bone, in an effort to simplify our model. Table 2 details the muscular structure of our biomechanical system.

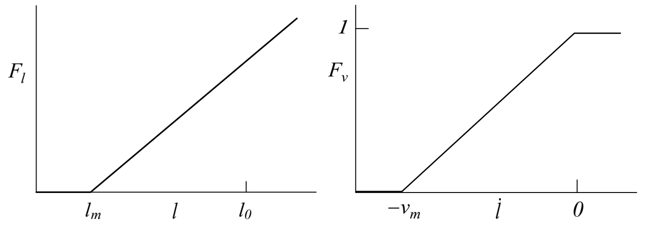
4.3 Hill-Type Muscle Model

To model each muscle actuator, we employ a popular muscle model in biomechanics research, which is known as a Hill-type model. Good introductions to this model can be found elsewhere [Ng-Thow-Hing 2001; Winters and Crago 2000]. If we assume that the length of the tendon remains constant as the muscle is stretched, the muscle force comes from two sources: A parallel element (PE), which passively produces a restoring force f_P due to the material elasticity of the muscle, and a contractile element (CE), which actively generates a contractile force f_C in response to excitation from the motor neurons. The total muscle force is: $f_m = f_P + f_C$.

The PE is modeled as a uniaxial exponential spring:

$$f_P = \max(0, k_s(\exp(k_c e) - 1) + k_d \dot{e}),$$

where k_s and k_c are *elastic coefficients*, k_d is the *damping coefficient*, $e = (l - l_0)/l_0$ is the *strain* of the muscle, with l and l_0 its *length* and *slack length*, respectively, and $\dot{e} = \dot{l}/l_0$ is the *strain rate* of the muscle. Since f_P is determined by the state of the musculoskeletal system rather than by its neural activation, it is not treated as a control input in (1).



(a) Force-length relation (b) Force-velocity relation

Figure 4: Linearized Hill-type model.

The contractile force from the CE is typically expressed as

$$f_C = aF_l(l)F_v(\dot{l}), \tag{2}$$

where $0 \leq a \leq 1$ is the *activation level* of the muscle (i.e., the input signal from the motor neuron innervating the muscle). F_l denotes the *force-length relation* (i.e., the muscle force as a function of its length) and F_v denotes the *force-velocity relation* (i.e., the muscle force as a function of its shortening velocity).

We use a simple, linearized Hill-type model with F_l and F_v as shown in Fig. 4. In particular, $F_l(l) = \max(0, k_{\max}(l - l_m))$, where $k_{\max}$ is the maximum stiffness of a fully activated muscle and l_m is the minimum length at which the muscle can produce force, and $F_v(\dot{l}) = \max(0, 1 + \min(\dot{l}, 0)/v_m)$, where v_m is the maximum contraction velocity under no load. Per [Ng-Thow-Hing 2001], we set $l_m = 0.5l_0$ and $v_m = 8l_0$ sec^{-1}. The coefficient k_c is set to 7 for all the muscles. The coefficients k_s, k_d, and $k_{\max}$ for each muscle are scaled by its *strength weight factor* w, which is set roughly proportional to the cross sectional area of the muscle. Table 2 specifies the strength weight factors and attachment sites of the muscles.

Note that the original Hill model includes a negative stiffness range as the muscle is stretched. This range is seldom reached in everyday movement (see Ch. 7 of [Winters and Crago 2000] and references therein). It is known that negative stiffness can de-stabilize musculoskeletal systems such as ours. We have avoided this by modifying the model. Even though our $F_l(l)$ increases monotonically (the same F_l was used in [Hogan 1984]), the difference relative to the original Hill model is modest, because the stretch of the neck muscles is limited by the constrained motions of the bones.

5 Hierarchical Control System

Like the human muscle control architecture, that of our biomechanical neck model is hierarchical. In our system (Fig. 2), the higher-level voluntary controller (Fig. 5) delivers a kinematic signal (*setpoint signal*) as well as a dynamic signal (*feedforward signal*) to the lower-level reflex controller. The reflex controller then determines the required motor neural signal for each muscle while monitoring the state of the muscle, specifically its strain and strain rate. Since the output signal from the voluntary controller normally changes more slowly than that of the reflex controller, we can run the two controllers at different speeds. The hierarchical structure offers a practical advantage in view of the fact that the computational cost of the voluntary controller is significantly higher than that of the reflex controller. In our system, the voluntary controller updates every 40 milliseconds whereas the reflex controller updates once per integration time step; i.e., approximately every millisecond.

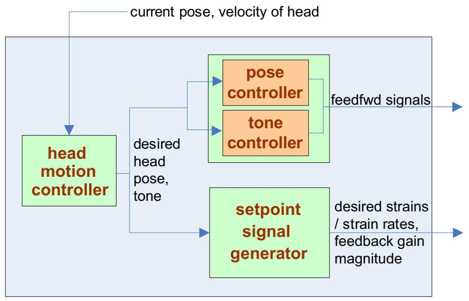

current pose, velocity of head
pose controller
tone controller
feedfwd signals
head motion controller
desired head pose, tone
setpoint signal generator
desired strains / strain rates, feedback gain magnitude

Figure 5: The sub-controllers in the voluntary controller.

5.1 Reflex Control

The reflex controller generates a neural activation level a for each muscle by summing the feedforward signal a_f generated by the voluntary controller with an internally-generated feedback signal a_b that is computed by comparing the strain and strain rate of each muscle with their desired values. In terms of its biological basis, our reflex controller emulates the stretch reflex in human motor control, which is believed to be modulated by the gamma motor neural signal and is activated when the muscle is elongated beyond the desired length [Kandel et al. 2000]. The length and velocity of the muscle are measured by its proprioceptive sensory organs, among them the spindles inside the muscle.

Our reflex control model is as follows:

$$\begin{aligned} a_b &= s\left(k_p(e - e_d) + k_d\,\mathrm{sat}_m(\dot{e} - \dot{e}_d)\right), \\ a &= \min(1, \max(0, a_f + a_b)), \end{aligned} \tag{3}$$

where k_p and k_d are proportional and derivative gains, s is the feedback gain scaling factor, and e and $\dot{e}$ are the muscle's strain and strain rate, respectively (given in Section 4.3). Note that s along with the desired strain e_d and desired strain rate $\dot{e}_d$ are determined by the setpoint signal generated by the voluntary controller. In our experience, a large derivative feedback force overwhelms the proportional feedback force and tends to make the system unstable, so we employ the function

$$\mathrm{sat}_m(x) = \begin{cases} x & \text{if } |x| < m, \\ m\,\mathrm{sgn}(x) & \text{otherwise}, \end{cases}$$

which saturates its input at the value m (we set m to 2.0). With this saturated derivative feedback, we found that we can use a reasonable derivative gain $k_d = 0.05$ relative to the proportional gain $k_p = 8$ without having to decrease the integration time step.

5.2 Voluntary Control and Learning

A distinctive feature of human motor control is that one can increase the stiffness or tone of the body by coactivating opposing (agonist and antagonist) muscles. Humans are known to use coactivation to increase their stability when subjected to external disturbances or to improve accuracy when performing certain difficult motor control tasks. From the mechanical perspective, higher tone can be advantageous, because it increases the stiffness of the musculoskeletal system, thus improving robustness against perturbation. However, the issue of tone control has been more or less neglected in animation research [Neff and Fiume 2002]. Biomechanics researchers have suggested that humans can independently control the coactivation and movement [Yamazaki et al. 1994]. To emulate this feature

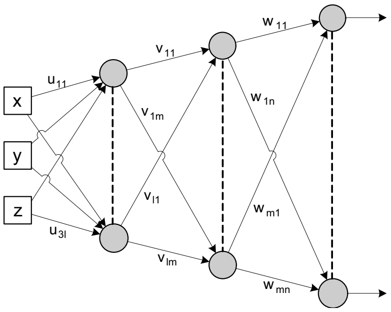


Figure 6: A 3-layer neural network.

of human motor control, we have designed our voluntary controller to be capable of controlling the pose and the tone of the neck independently.

In our system, the pose signal $\mathbf{a}_p$ and tone signal $\mathbf{a}_t$ are independently generated by two neural networks, and the feedforward signal is obtained by summing the two signals:

$$\mathbf{a}_f = \mathbf{a}_p + \mathbf{a}_t.$$

This separation is possible because the tone signal is computed to be orthogonal to the pose signal, in the sense that the tone signal does not affect the pose of the system.

Another distinctive feature is that through trial and error, humans and other animals are able to learn how to control their muscles in order to move effectively and efficiently. This can be regarded as an optimization process that solves for the necessary neural input to the muscles required to achieve a desired motion [Grzeszczuk and Terzopoulos 1995]. Throughout this incremental learning process, the brain generates increasingly more appropriate motor signals to accomplish the desired motion and it becomes decreasingly dependent on feedback control. From the robotics perspective, this feedforward signal enables the animal to use lower feedback gains, which enhances the naturalness of the motion, among other factors. Similarly, the voluntary controller in our system generates its feedforward signal through machine learning. In particular, we solve offline for optimal neural inputs that achieve sampled target poses and tones, and use them to train neural network controllers to efficiently output optimal solutions online [Grzeszczuk et al. 1998].

5.2.1 Neural Networks

Since the computational structure of artificial neural networks is based on insights into biological nervous systems, we employ them in our pose and tone controllers. Moreover, the well-known function approximating ability of neural networks is attractive and compatible with our training strategy. Our offline learning process generates sample input-output training pairs by solving repeated optimization problems, as we will explain in the subsequent two sections, and then it trains neural networks on numerous such precomputed pairs, thus obtaining a suitable function approximator. It takes less than 10 hours to train each neural network on our 3.2 GHz CPU PC. Once trained, the neural network can approximate suitable outputs for particular inputs orders of magnitude faster than one can hope to do by solving the associated optimization problem. This makes the trained neural network suitable for online use, especially for interactive animation.

Fig. 6 shows the fully connected, feedforward neural network that we employed for our pose and tone controllers. The inputs to the neural network are the normalized three components of the quaternion coordinate $\mathbf{h}$ (orientation) of the head. Each neuron is modeled as a sigmoid function, $y = \tanh(b + \sum_{i=1}^{k} w_i x_i)$, where b is a bias term and the w_i are the weights of the inputs x_i from the k neurons in the previous layer. The output of the neural network is the normalized pose signal a_p (or tone signal a_t). The dimension of the network output vector is 72, the total number of muscles. We use a 3-layer network with two hidden layers of sizes 20 and 40 neurons. The trainable parameters of the network are the weights and bias terms associated with the neurons, and they are computed using the backpropagation learning algorithm, as in [Grzeszczuk et al. 1998]. Although free and commercial neural network packages are available, we used our own simple implementation.

5.2.2 Pose Controller

To train the pose controller neural network, we randomly sample the head pose space. For the i-th sample pose $\mathbf{h}_d^i$, the desired pose signal $\mathbf{a}_p^i$ is the solution of the constrained optimization problem

$$\mathbf{a}_p^i = \underset{\mathbf{a}}{\operatorname{argmin}} \|\mathbf{f}_C^w\|^2 \tag{4}$$

$$\text{subject to} \quad \mathbf{c}(\mathbf{q}_d^i, \mathbf{0}) + \mathbf{K}_s \mathbf{q}_d^i - \mathbf{P}(\mathbf{q}_d^i)\mathbf{f}_p = \mathbf{P}(\mathbf{q}_d^i)\mathbf{f}_C, \quad \mathbf{a} \in [0,1]^m.$$

Eq. (4) minimizes weighted muscle contraction forces $\mathbf{f}_C^w = \mathbf{W}^{-1}\mathbf{f}_C$, where $\mathbf{W} = \operatorname{diag}(w_1, \ldots, w_m)$ for the m muscles. The strength weight factors w_i (see Table 2) encourage muscle forces in proportion to muscle strengths. The primary constraint in the minimization stems from (1) with $\ddot{\mathbf{q}} = \dot{\mathbf{q}} = \mathbf{0}$ (to maintain $\mathbf{h}_d^i$ statically), $\mathbf{f}_e = \mathbf{0}$ (no external forces other than gravity), and with the joint angles $\mathbf{q} = \mathbf{q}_d^i$ provided by the setpoint signal generator to yield the desired $\mathbf{h}_d^i$ (i.e., $\mathbf{h}_d^i = \mathbf{g}(\mathbf{q}_d^i)$, where $\mathbf{g}(\cdot)$ is the forward kinematics function). To solve (4), we use DONLP2 [Spellucci], which is based on the sequential equality constrained quadratic programming method. On the order of $20,000 \approx N$ training pairs $\{\mathbf{h}_d^i, \mathbf{a}_p^i\}_{i=1}^N$ are generated offline to train $\mathbf{n}_p$ using backpropagation.

Given a desired head pose $\mathbf{h}_d$, the trained pose controller network efficiently computes a feedforward signal online to maintain $\mathbf{h}_d$ with minimal muscle contraction forces $\mathbf{f}_C$:

$$\mathbf{a}_p = \mathbf{n}_p(\mathbf{h}_d).$$

Given the form of the objective function, $\mathbf{n}_p$ cannot coactivate opposing muscles to increase musculoskeletal stiffness.

5.2.3 Tone Controller

Due to muscle redundancy, there are usually many combinations of muscle coactivations that can increase tone. It remains an open research problem as to how humans choose opposing muscle coactivations. Instead of formulating some explicit stiffness criterion that the musculoskeletal system should maximize, our intuitive assumption is that to achieve maximum stiffness one maximizes the muscle contraction forces while not actuating the musculoskeletal system. Similarly to $\mathbf{n}_p$ above, the tone neural network $\mathbf{n}_t$ is trained offline with on the order of $20,000 \approx N$ training pairs $\{\mathbf{h}_d, \mathbf{a}_t\}_{i=1}^N$, where the maximum tone signal $\mathbf{a}_t^i$ is obtained by solving the constrained optimization problem

$$\mathbf{a}_t^i = \underset{\mathbf{a}}{\operatorname{argmax}} \|\mathbf{f}_C^w\|^2 \quad \text{subject to} \quad \mathbf{P}(\mathbf{q}_d^i)\mathbf{f}_C = \mathbf{0}, \quad \mathbf{a} \in [0,1]^m.$$

Given a desired head orientation $\mathbf{h}_d$ and *tone parameter* c, the tone signal is computed online using the trained network $\mathbf{n}_t$ as

$$\mathbf{a}_t = c\,\mathbf{n}_t(\mathbf{h}_d). \tag{5}$$

Since we should have $a_f = a_p + a_t \leq 1$, then $0 \leq c \leq 1 - \max(a_p)$.

It may at first seem surprising that arbitrary tone can be achieved by simply scaling the output of $\mathbf{n}_t$. However, this is to be expected because the resulting muscle force $\mathbf{f}_C$ is constrained to lie in the null space of $\mathbf{P}(\mathbf{q})$, thus it does not contribute to the generalized force τ. Furthermore, this is possible because the muscle force and the neural signal are linear in the Hill-type model (2); hence, scaling the neural signal retains the muscle force in the null space of $\mathbf{P}(\mathbf{q})$. Note that, aside from c, the tone signal $\mathbf{a}_t$ depends only on the configuration of the system $\mathbf{q}_d$. It is not affected by the external force field (gravity) or by the global orientation of the system, whereas the pose control signal does have such dependencies.

5.2.4 Setpoint Signal Generator

Given a desired head pose $\mathbf{h}_d$, the setpoint signal generator computes the desired strain $\mathbf{e}_d$ and strain rate $\dot{\mathbf{e}}_d$ of each muscle. The former is given by

$$\mathbf{e}_d = \mathbf{n}_g(\mathbf{h}_d).$$

Unlike the pose and tone controllers, we do not implement the function $\mathbf{n}_g$ as a neural network. Rather, it entails the solution of the constrained optimization problem

$$\mathbf{q}_d = \underset{\mathbf{q}}{\operatorname{argmin}} \|\mathbf{q}^v\|^2 \quad \text{subject to} \quad \mathbf{h}_d = \mathbf{g}(\mathbf{q}), \tag{6}$$

where $\mathbf{q}^v = \mathbf{V}^{-1}\mathbf{q}$ with $\mathbf{V} = \operatorname{diag}(v_1, \ldots, v_n)$ and n the number of joints. Here, v_i is the weighting factor of joint q_i, which we set to the range of the joint in accordance with [Hay and Reid 1988], and $\mathbf{g}(\cdot)$ is the forward kinematics function. Having computed $\mathbf{q}_d$ (i.e., the smallest joint angles that achieve $\mathbf{h}_d$), we then obtain $\mathbf{e}_d$ from $\mathbf{g}(\mathbf{q}_d)$.

Finally, we compute the desired strain rate as

$$\dot{\mathbf{e}}_d = \frac{\mathbf{n}_g(\mathbf{h}_d(t+\Delta t)) - \mathbf{n}_g(\mathbf{h}_d(t))}{\Delta t},$$

where $\mathbf{h}_d(t)$ and $\mathbf{h}_d(t+\Delta t)$ are the desired orientation of the head at time t and at the subsequent time step $t+\Delta t$, respectively.

Although simple, the objective in (6) yields natural looking results. We did not implement the setpoint signal generator as a neural network for several practical reasons. First, due to its simplicity, (6) can be solved faster online than by using a neural network. We solve (6) using the gradient descent method, which typically achieves the solution within 3 iterations. Second, this direct computation yields an accurate result, whereas a neural network would incur some error. The error issue is potentially crucial here, as the setpoint signal serves as a reference signal for feedback control in the reflex controller.

5.2.5 Head Motion Controller

At the topmost level of our control hierarchy is a voluntary controller that produces movements which take the head from a current orientation to a desired new orientation. It does its job by providing a series of commands to the neck feedforward and setpoint signal generators to modify the pose/tone of the biomechanical system. We will discuss two approaches next.

Interpolation: Given quaternion representations of initial $\mathbf{h}_i(t_i)$ and desired final $\mathbf{h}_f(t_f)$ orientations of the head, a natural trajectory $\mathbf{h}_d(t)$ from $t_i \leq t \leq t_f$ may be computed as the spherical linear interpolation $\mathbf{h}_d(t) = \operatorname{slerp}(r(t), \mathbf{h}_i, \mathbf{h}_f)$. The interpolation parameter $r(t)$ is determined so that the time derivative of r is bell shaped; i.e., $\dot{r}(t_n) = 1 - \cos(2\pi t_n)$, where $t_n = (t - t_i)/(t_f - t_i)$. The head motion controller also modulates the tone c in (5) and feedback gain scaling factor s in (4) by comparing the actual and desired orientations of the head. If the total accumulated error over a time window exceeds a threshold, the controller increases the tone and feedback gain gradually until the error falls below threshold. By decreasing the error threshold, the neck maintains the pose better and is stiffer. Conversely, by increasing the error threshold, the neck produces more relaxed motion and allows greater perturbation during the movement.

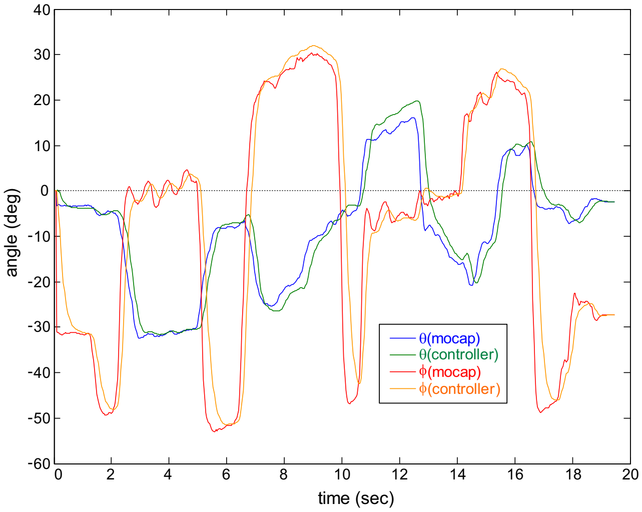
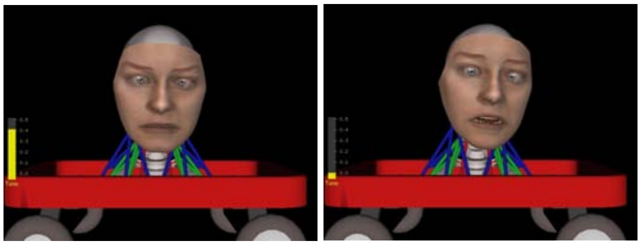


Figure 7: An example of the sensorimotor controller following head motion capture data. The time plots compare the longitudinal θ and latitudinal ϕ angles of the synthetic head (controller) and real head (mocap).

Sensorimotor Control: Although the interpolation generator produces reasonable head-neck motion for the purposes of character animation, an approach that is more consistent with biological control mechanisms is sensorimotor control. At every command-generating instant of the voluntary controller, a desired head orientation and velocity command are generated on the fly based on sensory feedback. For example, given initial $\mathbf{h}_i(t_i)$ and desired final $\mathbf{h}_f(t_f)$ orientations of the head, the sensorimotor controller initiates a head movement towards $\mathbf{h}_f(t_f)$. The inertia of the head yields a natural angular acceleration. During movement, the instantaneous head angle error $\|\mathbf{h}_f(t_f) - \mathbf{h}(t)\|$ is sensed at a fast rate and corrective "steering" is applied to continually reduce the error. When the error decreases to below some threshold, the sensorimotor controller begins to slow the head so that it comes to rest in pose $\mathbf{h}_f(t_f)$.

Fig. 7 shows an example of our sensorimotor controller tracking head motion capture data from the CMU database (mocap.cs.cmu.edu—subject #79, motion #83 (shaving)). The sequence of head orientations from the motion capture data are set as target head orientations to the head controller. The head controller computes the desired head angular velocity as $\dot{x}_d(t) = (x_d(t+d) - x(t))/d$, and the desired orientation as $x_d(t+\Delta t) = x(t) + \dot{x}_d(t)\Delta t$, where $x(t)$ is the angular representation of the head orientation at time t, and d is the time in which the system is allowed to reach the target $x_d(t+d)$. In this example, we set $d = 10\Delta t$ with Δt = 0.033 sec. Fig. 7 reveals that our dynamic head-neck system follows the motion capture data while smoothing noise in the data.



(a) Both tone c and feedback gains s are modulated.

(b) No tone control; only feedback gains s are modulated.

Figure 8: Different head motions result depending on the tone control. (a) and (b) are snapshots taken at the same time with identical perturbations of the red wagon.

6 Experiments and Results

We have conducted several experiments with our biomechanical, neuromuscular face-head-neck animation system.

6.1 Basic Simulations

Even with the rotational springs (which represent ligaments and disks) attached to each cervical joint, the skeletal system appropriately collapses in gravity, exhibiting the expected passive dynamics. Without active control, the complete musculoskeletal system appropriately collapses as well, albeit in a more damped manner. However, simulating the passive dynamics of the musculoskeletal system was crucial for adjusting the parameters of the 72 muscles. Since each muscle's stiffness and damping parameters are not known precisely and, even if they were, since we cannot model all of the muscles in the neck (thus our actuators must also assume the roles of neighboring unmodeled muscles), we cannot naively use empirical data reported in the biomechanics literature. Hence, we tuned the muscle parameters in our model by visually assessing the plausibility of the resulting passive dynamics.

With the feedforward and feedback control networks trained, we ascertained the importance of feedforward control by turning it off and attempting to animate the head using only feedback control. With the feedback gain set at its nominal value, feedback control alone fails to maintain the upright stance of the cervical spine with the head in balance. However, feedback control is important for maintaining the stability of the musculoskeletal system.

6.2 Tone Control Experiments

In a different experimental scenario, we apply perturbations to the base link of the head-neck system that are analogous to riding on a vehicle over a bumpy road (Fig. 8). As the head motion controller senses excessive error between the desired and the actual orientation of the head, it gradually increases the feedback gain s (to its maximum value of 3.0) and tone c (to its maximum value of 0.4) until the error drops below an acceptable threshold or until the maxima are reached. Not surprisingly, the head wobbles less when both the tone and feedback gain are increased, compared to increasing the feedback gain alone. However, we also observed that increasing the tone alone is insufficient to suppress the wobble. This implies that reflexive stiffness also plays an important role in the overall stiffness of the musculoskeletal system. Appendix B discusses reflexive stiffness and intrinsic stiffness.

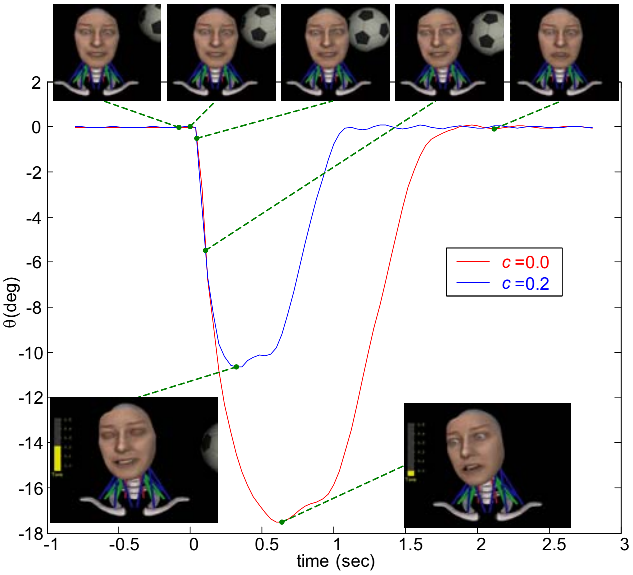


Figure 9: Head orientation longitudinal angle θ over time during an impact simulation. When controlled with zero tone signal, $c = 0$, (red), the head is perturbed more by the impact with the ball than when controlled with tone $c = 0.2$ (blue). All snapshots except for the lower left one are sampled from the zero tone (red) case.

In a second set of perturbation experiments, we apply with a ball external impacts to the head under various tone conditions (Fig. 9). After impact, the head motion controller issues head stopping commands to the lower-level neuromuscular controllers; i.e., set the desired pose to the current pose and the desired velocity to zero. When the head approaches stationariness, the controller issues a command for the head to return to its original upright pose. Since the stiffness of the musculoskeletal system is greater when it increases its tone by coactivating opposing muscles, it is less perturbed by the same impact. This illustrates the fact that even passive human motion differs markedly depending on the internal state of muscle activation.

6.3 Gaze Behavior

Human vision is foveated. The foveal region of the retina, which spans roughly 5 degrees of visual arc, is specialized for high-acuity, color vision. To see an object clearly, gaze-shifting eye movements are usually needed to direct the eye to the visual target. Since the resulting eye motion disrupts vision, these movements are executed as quickly as possible and are called *saccadic* eye movements. As a visual target moves closer, the two eyes must also converge onto the target; these are called *vergence* eye movements. The oculomotor system, which positions the eyes relative to the head, and its interaction with head movement has been the subject of intense research (see, e.g., [Carpenter 1988]).

Given the significantly greater mass of the head relative to the eye, head dynamics are much more sluggish than eye dynamics. For example, in a voluntary head-eye movement to direct the gaze at an off-axis visual target in the horizontal plane, the eye movement is an initial high-speed saccade in the direction of the head movement, presumably to facilitate rapid search and visual target localization,

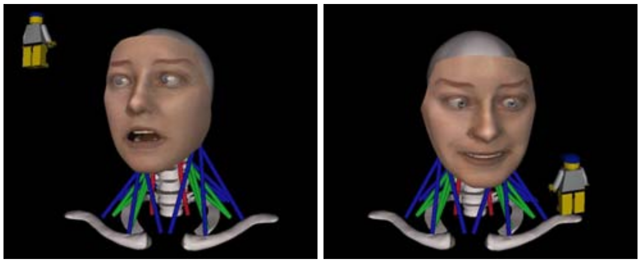


Figure 10: Head-Eye gaze behavior. Snapshots of the model gazing at a target in different directions.

followed by a slower return to orbital center, compensating for the more sluggish head movement that follows.

As Fig. 10 shows, our biomechanical model can synthesize coordinated head-eye movements that emulate at least the primary head-eye movement phenomena reported in the literature. When we present a moving visual target (the doll) to the model, the eyes are directed to make a saccadic ocular rotation (with maximum angular velocity of 200 degrees/sec) to point in the direction of the visual target relative to the head. Simultaneously, the head motion sub-controller of the neck neuromuscular controller issues a high-level command to rotate the head in the direction of the gaze. As the head executes the desired rotation via the low-level physical simulation, the eyes make a continuous compensatory movement such that they remain directed at the visual target. Fig. 10 shows the head gazing at the target in two different directions. Employing a rule-based behavior routine, the biomechanical face automatically synthesizes baby-like facial expressions as the eyes and head track the target. It appears awed when the doll is situated above the head, pleased when the doll is around eye level and held still, and angry when the doll is shaken.

6.4 Autonomous Multi-Head Interaction

Fig. 11 illustrates three autonomous face-head-neck systems interacting in a multi-way behavioral facial animation scenario, which was inspired by a more primitive demonstration in [Terzopoulos and Lee 2004] not involving neck models. In our version, each of the faces is supported by our head-neck musculoskeletal system, which automatically synthesizes all of the head motions necessary to sustain a highly dynamic multi-way interaction. As in the above demonstration, the synthesized head movements must cooperate with eye movements in order to direct the gaze at visual targets in a natural manner. The middle head in the figure acts as a "leader" synthesizing random expressions and alternating its attention between the other two heads, which act as "followers". Once a follower has the leader's attention, the follower will observe the leader's expression and engage in expression mimicking behavior. However, excessive mimicking will lead to behavior fatigue—the follower will lose interest in the leader and attend to its fellow follower. A complete explanation of the behavioral modeling is beyond the scope of this paper; reference [Terzopoulos and Lee 2004] provides additional details.

7 Discussion

Biomechanical musculoskeletal simulation governed by neuromuscular and behavioral control layers seems to be the scientifically principled approach to building self-animating, lifelike characters. In particular, our head-neck model aspires to be significantly more biomimetic than simpler joint-torque-driven articulated models inspired by robotics [Neff and Fiume 2002; Faloutsos et al. 2001; Hodgins et al. 1995]. At least for the time being, we believe that it addresses the modeling challenge at the right level of detail. It is also compatible with the biomechanical face model that we have employed in our work and supports simple behavioral animation in response to interesting external stimuli, such as other face-head-neck systems. Our work has made progress toward a complete and fully integrated cervical-craniofacial simulation in anticipation of an inevitable biomechanical and functional emulation of the entire human body for the purposes of computer animation.

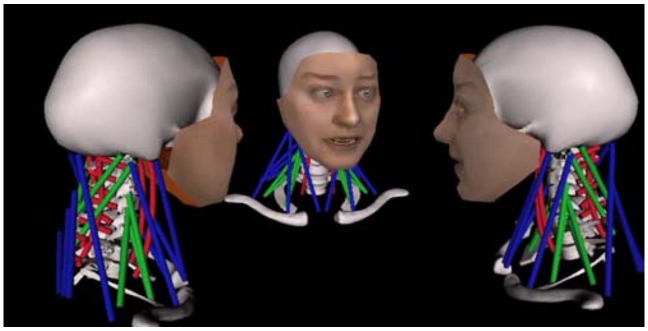


Figure 11: Autonomous behavioral-based interaction between three face-head-neck systems.

The salient details of human neck movement cannot easily be mimicked using conventional joint-actuated skeletal models. In particular, the moment-generating capacity of each joint varies—it is determined by the geometry and capacities of the associated muscles. The muscle itself cannot simply be replaced with a PD-servo—it has nontrivial passive dynamic and force-generating properties, as approximated by the Hill model. Our controllers compute the activation level of each muscle, and this could provide a natural approach to simulating local skin deformation due to underlying muscle contraction and bulging [Kähler et al. 2001; Scheepers et al. 1997; Wilhelms and Gelder 1997]. For example, even in a constant skeletal pose, the sternocleidomastoid and trapezius muscles bulge as the head reacts to applied forces, producing externally salient shape changes of the neck. Simpler approaches than ours will no doubt become increasingly complex as they are augmented in an attempt to capture some of these nuances of human neck movement.

In our work, we used static optimization during the offline training of the pose controller in order to compute optimal muscle activations that generate desired static poses. By contrast, some motion control schemes employ more costly dynamic optimizations to solve for optimal actuator input temporal functions that generates desired output motions. Since most head motions lack vigorous dynamics, our static optimization yields satisfactory results. Moreover, the difference between static optimization and dynamic optimization may not be significant; in the context of normal human gait, Anderson and Pandy [2001] argue that static optimization and dynamic optimization solutions are virtually equivalent.

8 Conclusion and Future Work

We have introduced a biomechanical model of the human head-neck system. Emulating the relevant anatomy, our model is characterized by kinematic redundancy (7 cervical vertebrae coupled by 3-DOF joints), as well as muscle actuator redundancy (72 neck muscles arranged in 3 muscle layers). To control the biomechanical model for the purposes of human head animation, we developed a hierarchical neuromuscular control model that mimics the relevant biological motor control mechanisms. Incorporating a low-level reflex sub-controller, an intermediate-level voluntary sub-controller, and a high-level head motion controller, our novel head-neck control system not only provides inputs to the numerous muscle actuators, but also affords control over muscle tone, which determines the stiffness of the craniocervical multibody system independently of head pose and movement. We showed that it is possible to train the neural networks in our neuromuscular controller offline so that they can efficiently generate the online pose and tone control signals that are required to produce a variety of head movement behaviors for the autonomous animation of the human head and face.

In view of the complexity of the neck, our biomechanical model is inevitably incomplete. For some applications, it would be necessary to model not only additional neck muscles, but also ligaments and the disks (cartilage filled with a gelatinous substance) that deform to cushion the vertebrae of the spinal column. A more complete model would enable us to simulate cervical injuries such as whiplash.

Since bulging muscles play an important role in the externally salient deformation of flesh, in future work we plan to include dynamic neck muscles of anatomically consistent 3D shape and volume, which bulge appropriately as they contract. We also plan to wrap the neck in a dynamically simulated skin compatible with the one on the synthetic face.

We need to tighten the coupling between the biomechanical neck and face models. Currently, the dynamics of the neck do not adequately propagate to the face or vice versa. A tighter coupling will yield more interesting dynamic animations of the face, including facial soft tissue deformations when the head is moved vigorously.

Our pose controller assumes that the global orientation of the musculoskeletal system (i.e., the orientation of the base link) is upright. In other words, it would not output the proper feedforward signal if the system is oriented horizontally. In future work, we plan to incorporate the global orientation of the system as an additional input to the pose controller. This will require suitably augmented neural networks and re-training on augmented data incorporating global orientation.

Applying the methodology introduced in this paper, it should be possible to model and animate the necks of lower animals, such as gorillas, dogs, horses, and even giraffes. Finally, as the demonstrations in Figs. 10 and 11 suggest, a further developed version of our biomechanical model with refined neuromuscular controllers and expanded behavioral repertoire shows promise as an essential component of future autonomous, intelligent virtual humans.

A Moment Arm Matrix Computation

The moment arm matrix $\mathbf{P}(\mathbf{q})$ is defined as $\tau = \mathbf{P}(\mathbf{q})\mathbf{f}$, where $\tau = [\tau_1, \ldots, \tau_n]^T$ is the vector of joint torques (generalized forces) and n is the number of joints, and $\mathbf{f}$ is the vector of muscle contraction forces. Let $\mathbf{l}_j$ be the vector from the origin to the insertion of muscle j. Let $\delta l_j = \langle \dot{\mathbf{l}}_j, \mathbf{l}_j / \|\mathbf{l}_j\| \rangle$ and $\delta \mathbf{l} = [\delta l_1, \cdots, \delta l_m]^T$, where m is the number of muscles. The principle of virtual work $\langle \mathbf{f}, \delta \mathbf{l} \rangle = \langle \tau, \delta \mathbf{q} \rangle$ yields the relation $\mathbf{P}(\mathbf{q})^T \delta \mathbf{q} = \delta \mathbf{l}$. If we set $\delta \mathbf{q}$ to be the i-th basis vector $\mathbf{e}_i$ in the joint space, then the resulting $\delta \mathbf{l}$ is the same as the i-th row of $\mathbf{P}$. Thus, we can compute $\mathbf{P}(\mathbf{q})$ as follows:

```
Require: $\mathbf{q}$
Update the transformation matrix of each bone
for $i = 1$ to $n$ do
    Set $\dot{\mathbf{q}} = \mathbf{e}_i$
    Compute generalized velocity of each transformation matrix
    Compute $\delta \mathbf{l}$ as defined above
    Set the $i$-th row of $\mathbf{P}$ to $\delta \mathbf{l}$
```

B CE Contribution to Stiffness

From $\mathbf{P}(\mathbf{q}) = (\partial \mathbf{l} / \partial \mathbf{q})^T$ and $\frac{1}{2} \delta \mathbf{q}^T \mathbf{K}_J \delta \mathbf{q} = \frac{1}{2} \delta \mathbf{l}^T \mathbf{K}_M \delta \mathbf{l}$, where $\mathbf{K}_M = \mathrm{diag}(k_1, \ldots, k_m)$ and k_i is the stiffness of muscle i, we obtain the joint space representation of muscle stiffness $\mathbf{K}_J = \mathbf{P}(\mathbf{q}) \mathbf{K}_M \mathbf{P}(\mathbf{q})^T$. Since k_i is always positive, $\mathbf{K}_J$ is a positive definite matrix, thus increasing the overall stability of the system. Consider the stiffness of a muscle due to its contractile element k_C in our muscle model. From (2),

$$k_C = \frac{\partial f_C}{\partial l} \propto k_{\max} a + \frac{\partial a}{\partial l} F_l.$$

Here, $k_{\max} a$ is the intrinsic stiffness of a muscle, which is effective regardless of the frequency of a perturbation. The reflexive stiffness due to the reflex control is $(\partial a / \partial l) F_l \propto k_p F_l$. Note that, unlike the intrinsic stiffness, the reflexive stiffness is effective only for slower perturbations, since there is a time lag for a reflexive response due to the low speed of neural information delivery. Coactivating muscles increases intrinsic stiffness; hence it is more effective for suppressing quicker perturbations than reflex control.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. IIS-0326388. SHL was supported in part by the IT Scholarship Program supervised by IITA and the Ministry of Information and Communication, Republic of Korea. The biomechanical face model was implemented by Yuencheng Lee; we thank him for his contribution to this work. Our appreciation also goes to Wei Shao and Jinwook Kim for valuable discussions. We acknowledge the helpful comments from the anonymous referees, which improved our presentation. The majority of the research reported herein was done while the authors were at the Media Research Lab of the Courant Institute of Mathematical Sciences at New York University.

References

ALBRECHT, I., HABER, J., AND SEIDEL, H.-P. 2003. Construction and animation of anatomically based human hand models. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA'03)*, 98–109.

ANDERSON, F., AND PANDY, M. 2001. Static and dynamic optimization solutions for gait are practically equivalent. *Journal of Biomechanics 34*, 153–161.

CARPENTER, R. 1988. *Movements of the Eyes*, 2nd ed. Pion, London.

CHEN, D. T., AND ZELTZER, D. 1992. Pump it up: Computer animation of a biomechanically based model of muscle using the finite element method. In *Computer Graphics (Proceedings of ACM SIGGRAPH 92)*, vol. 26, 89–98.

DELP, S., AND LOAN, J. 1995. A software system to develop and analyze models of musculoskeletal structures. *Computers in Biology and Medicine 25*, 21–34.

FALOUTSOS, P., VAN DE PANNE, M., AND TERZOPOULOS, D. 2001. Composable controllers for physics-based character animation. In *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, 251–260.

GRZESZCZUK, R., AND TERZOPOULOS, D. 1995. Automated learning of muscle-actuated locomotion through control abstraction. In *Proceedings of ACM SIGGRAPH 95*, Computer Graphics Proceedings, Annual Conference Series, 63–70.

GRZESZCZUK, R., TERZOPOULOS, D., AND HINTON, G. 1998. Neuroanimator: Fast neural network emulation and control of physics-based models. In *Proc. of ACM SIGGRAPH 98*, Computer Graphics Proceedings, Annual Conference Series, 9–20.

HAY, J., AND REID, J. 1988. *Anatomy, Mechanics, and Human Motion*, 2nd ed. Prentice-Hall, Englewood Cliffs, NJ.

HODGINS, J. K., WOOTEN, W. L., BROGAN, D. C., AND O'BRIEN, J. F. 1995. Animating human athletics. In *Proceedings of ACM SIGGRAPH 95*, Computer Graphics Proceedings, Annual Conference Series, 71–78.

HOGAN, N. 1984. Adaptive control of mechanical impedance by coactivation of antagonist muscles. *IEEE Transactions on Automatic Control AC-29* (Aug.), 681–690.

IRVING, G., TERAN, J., AND FEDKIW, R. 2004. Invertible finite elements for robust simulation of large deformation. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA'04)*, 131–140.

KÄHLER, K., HABER, J., AND SEIDEL, H.-P. 2001. Geometry-based muscle modeling for facial animation. In *Graphics Interface 2001*, 37–46.

KANDEL, E., SCHWARTZ, J., AND JESSELL, T. 2000. *Principles of Neural Science*, 4th ed. McGraw Hill, New York.

KAPANDJI, I. 1974. *The Physiology of the Joints. Vol. 3: The Trunk and the Vertebral Column*. Churchill Livingstone, Edinburgh.

KAWATO, M., FURUKAWA, K., AND SUZUKI, R. 1987. A hierarchical neural network model for control and learning of voluntary movement. *Biological Cybernetics 57*, 169–185.

KESHNER, E., AND PETERSON, B. 1995. Mechanisms controlling human head stabilization. I. Head-neck dynamics during random rotations in the horizontal plane. *Journal of Neurophysiology 73*, 2293–2301.

KIM, J., AND HEMAMI, H. 1998. Coordinated three-dimensional motion of the head and torso by dynamic neural networks. *IEEE Trans. on Systems, Man and Cybernetics. B 5*, 653–666.

KOMURA, T., SHINAGAWA, Y., AND KUNII, T. L. 1997. A muscle-based feed-forward controller of the human body. *Computer Graphics Forum 16*, 3 (Aug.), 165–176.

KOMURA, T., SHINAGAWA, Y., AND KUNII, T. L. 2000. Creating and retargeting motion by the musculoskeletal human body model. *The Visual Computer 16*, 5, 254–270.

LEE, Y., TERZOPOULOS, D., AND WATERS, K. 1995. Realistic modeling for facial animation. In *Proceedings of ACM SIGGRAPH 95*, Computer Graphics Proceedings, Annual Conference Series, 55–62.

MONHEIT, G., AND BADLER, N. I. 1991. A kinematic model of the human spine and torso. *IEEE Computer Graphics & Applications 11*, 2 (Mar.), 29–38.

NEFF, M., AND FIUME, E. 2002. Modeling tension and relaxation for computer animation. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA'02)*, 81–88.

NG-THOW-HING, V. 2001. *Anatomically-Based Models for Physical and Geometrical Reconstruction of Humans and Other Animals*. PhD thesis, University of Toronto, Department of Computer Science.

PAI, D. K., SUEDA, S., AND WEI, Q. 2005. Fast physically based musculoskeletal simulation. In *Proceedings of Sketches & Applications of ACM SIGGRAPH 2005*.

SCHEEPERS, F., PARENT, R. E., CARLSON, W. E., AND MAY, S. F. 1997. Anatomy-based modeling of the human musculature. In *Proceedings of ACM SIGGRAPH 97*, Computer Graphics Proceedings, Annual Conference Series, 163–172.

SIFAKIS, E., NEVEROV, I., AND FEDKIW, R. 2005. Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Transactions on Graphics 24*, 3 (Aug.), 417–425. Proceedings of ACM SIGGRAPH 2005.

SPELLUCCI, P. Donlp2. www.netlib.org/ampl/solvers/donlp2/.

TERZOPOULOS, D., AND LEE, Y. 2004. Behavioral animation of faces. In *Facial Modeling and Animation*, J. Haber and D. Terzopoulos, Eds., vol. 60 of *ACM SIGGRAPH 2004 Course Notes*. ACM SIGGRAPH, Aug., 119–128.

TSANG, W., SINGH, K., AND FIUME, E. 2005. Helping hand: An anatomically accurate inverse dynamics solution for unconstrained hand motion. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA'05)*, 319–328.

TU, X., AND TERZOPOULOS, D. 1994. Artificial fishes: Physics, locomotion, perception, behavior. In *Proceedings of ACM SIGGRAPH 94*, Computer Graphics Proceedings, Annual Conference Series, 43–50.

VASAVADA, A., LI, S., AND DELP, S. 1998. Influence of muscle morphometry and moment arms on the moment-generating capacity of human neck muscles. *Spine 23*, 412–422.

WARFEL, J. 1985. *The Head, Neck, and Trunk*, 5 ed. Lea & Febiger, Philadelphia.

WILHELMS, J., AND GELDER, A. V. 1997. Anatomically based modeling. In *Proceedings of ACM SIGGRAPH 97*, Computer Graphics Proceedings, Annual Conference Series, 173–180.

WINTERS, J., AND CRAGO, P., Eds. 2000. *Biomechanics and Neural Control of Posture and Movement*. Springer-Verlag, New York.

YAMAZAKI, Y., OHKUWA, T., ITOH, H., AND SUZUKI, M. 1994. Reciprocal activation and coactivation in antagonistic muscles during rapid goal-directed movements. *Brain Research Bulletin 34*, 587–593.

YIN, K., CLINE, M., AND PAI, D. K. 2003. Motion perturbation based on simple neuromotor control models. In *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications(PG'03)*, IEEE Computer Society.

ZORDAN, V. B., CELLY, B., CHIU, B., AND DILORENZO, P. C. 2004. Breathe easy: Model and control of simulated respiration for animation. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA'04)*, 29–37.