DOI: 10.1111/cgf.12863

COMPUTER GRAPHICS *forum*
Volume 36 (2017), number 6 pp. 122–147

Muscle-Based Control for Character Animation

A.L. Cruz Ruiz[1,2], C. Pontonnier[1,2,3], N. Pronost[4] and G. Dumont[1,2]

[1]IRISA/INRIA MimeTIC, Rennes, France
ana-lucia.cruz-ruiz@inria.fr, {charles.pontonnier, georges.dumont}@irisa.fr
[2]ENS Rennes, Bruz, France
[3]Ecoles de Saint-Cyr Coëtquidan, Guer, France
[4]Université Claude Bernard Lyon 1 - LIRIS SAARA, Lyon, France
nicolas.pronost@univ-lyon1.fr

Abstract
Muscle-based control is transforming the field of physics-based character animation through the integration of knowledge from neuroscience, biomechanics and robotics, which enhance motion realism. Since any physics-based animation system can be extended to a muscle-actuated system, the possibilities of growth are tremendous. However, modelling muscles and their control remains a difficult challenge. We present an organized review of over a decade of research in muscle-based control for character animation, its fundamental concepts and future directions for development. The core of this review contains a classification of control methods, tables summarizing their key aspects and popular neuromuscular functions used within these controllers, all with the purpose of providing the reader with an overview of the field.

Keywords: Computer animation, physics-based animation, motion control, motion synthesis, musculoskeletal simulation

ACM CCS: Computing methodologies—Animation

1. Introduction

Character animation is the art of bringing virtual characters to life through the design of solutions, such as motion controllers, which allow the reproduction and/or synthesis of new motions. The way these solutions are designed depends on the requirements of the specific application for which the character will be used, such as the degree of realism. We define realism as the visual degree of similarity between the actions, motions and responses of virtual characters with those of their real counterparts, at both the dynamic and kinematic level. The desired degree of realism also varies depending on the application. For instance, in a game or simulation, a higher degree of realism is sought for the main characters as opposed to background characters.

In this review, we present a recent, but growing trend for the production of realistic character animations: muscle-based control. This solution entails the use of more detailed character models, involving muscles and their controllers. Muscle-based models and controllers already span a variety of areas, such as ergonomic design, rehabilitation therapies, prosthetics, medical diagnosis and even post-surgery predictions. In animation, their usefulness no doubt depends on the requirements described above, along with the time to set up the model, and the desired motion repertoire. Nevertheless, we will show that it is a promising direction for the field of character animation in terms of enhancing motion realism. Thus, the main objective of the review is to provide animators involved in motion control, and experts from other fields (such as robotics and biomechanics) with similar interests, with a wide overview of the field of muscle-based control for animation.

Let us first introduce muscle-based control from a historical point of view. Throughout the years, several solutions have been proposed to mimic how humans and animals control their motions in order to animate virtual characters [Wil87]. These solutions are split into two main approaches: kinematic-based methods and physics-based methods. The former involves animating characters by specifying limb or joint trajectories. The latter involves animating characters by specifying actuator trajectories, such as joint torques or muscle signals.

One of the earliest kinematics-based methods was keyframing. In this technique, the user specified a sequence of positions and their corresponding times, later a computer made an interpolation

© 2016 The Authors
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd. Published by John Wiley & Sons Ltd.

(usually a splining technique) between the specified positions to generate motion. However, this implicated a very low-level control, where the user had to control each degree of freedom of the character. Another technique was the use of control functions, where the motion for each degree of freedom was specified via functions of position versus time, these functions were generally curves composed of a set of control points. This technique had the advantage that the changes could be more easily made on individual degree of freedoms, but the control was still very low level. Later, with the arrival of motion capture systems, animators began using recorded human kinematic data to drive or enhance animations [PB02, Gle98]. One major drawback of this technique was that motion diversity and quality were limited and conditioned by a motion database. Nevertheless, some approaches have extended and diversified the number of motions beyond those of the original databases by applying external forces on the characters and making dynamic corrections on the original motion [MKHK08, PD07].

A different approach to animation is physics-based animation. This technique is based on the development of physics simulators, which aim at replicating real environments by modelling the physical laws and conditions that define it. This has freed animators from worrying about enforcing certain motion characteristics which come implicitly with the presence of physics, and has granted virtual characters with a freedom of motion, which is unrestricted but physically plausible. Once these virtual environments are created, the animator should choose what character will be used for the animation and a strategy for motion synthesis (the strategy for the design of a motion controller).

The character model can exhibit different levels of detail in terms of skeleton, actuators, tissues, etc. (as will be explained in Section 3). A common choice has been simple skeletons actuated by ideal servo-motors, and commanded by servo-based controllers. A very complete review and categorization of these controllers can be found in [GP12]. The animation systems discussed in the latter review featured motions ranging from 2D locomotion [Hod91, vdPF93] and forward flips [HR90], 3D locomotion [RM01, RH02, Sim94], balance [YLvdP07, CBvdP10, CKJ*11, LKL10, AdSP07, MZS09, WZ10] and navigation on uneven terrain [WP10, MdLH10].

Despite these advances, several authors from areas, such as biomechanics, have shown the importance of increasing the level of detail in such models by including muscles as actuators (in the place of servos), and performing a muscle-based control. This has triggered an evolution from servo-based control, to servo-muscle-based control and finally to muscle-based control. Servo-based control assumes that the degrees of freedom (DoFs) of the character are actuated by servo motors, and therefore produces torques. Servo-muscle-based control assumes that a set of DoFs is actuated by servos, while another set is actuated by muscles, and consequently it produces a set of torques and a set muscle signals. Finally, muscle-based control assumes that all the joints of the character are actuated by muscles and produce muscle signals only.

Figure 1 depicts how muscle-based control is integrated in a physics-based framework. In this diagram, the physics simulator updates the state of the character and the environment as a result of the external forces in the environment and internal forces in the character. The external forces in the environment include gravity, force perturbations, and forces due to obstacles, slopes, and changes in the friction coefficient. The internal forces in the character are the muscle, ligament, and joint-contact forces, which are generated due to the muscle signals produced by the muscle-based control method. To design these control methods, knowledge in neuromusculoskeletal simulation is vital since it provides a basis on which to: model the characters, automate the motion generation process and solve the motion redundancy problem. More information on these topics will be featured in Section 3.

The simulator first gathers the internal and external forces, and executes a forward dynamics simulation that helps to compute the resulting state of the character. This state comprises several kinematic and muscle variables, such as the position and velocity of each link (or sub-body) in the character, and the length and velocity of shortening or lengthening of each muscle. Specifically, the simulator first computes the link accelerations. These are then fed to a numerical integrator that updates the link positions and velocities, which are finally used to update the state of the muscles.

As previously mentioned, the main contribution of this review is providing the reader with an overview of the trends in the field of muscle-based control for animation. We begin by explaining our motivation in Section 2. In Section 3, we present an overview of the neuromusculoskeletal simulation tools, used to design control laws, such as musculoskeletal modelling, simulated dynamics and muscle force estimation. The core of the review is Section 4, where we feature a muscle-based control method classification consisting of two categories: controller optimization methods and trajectory optimization methods. This section also contains the main strengths and weaknesses of each method and examples of controllers found in the animation domain. Next, Section 5 features comments on the current and future developments in the field of muscle-based animation. Finally, the review concludes with a set of appendixes including the most relevant aspects presented. Appendix A contains a table summarizing our classification and important control and model features. Appendix B contains a list of neuromuscular cost functions used by the controllers. Appendix C contains a short list of physics simulators and the controllers that use them.

Lastly, we would like to emphasize that the review aims at highlighting the contributions and novelties introduced by state-of-the-art muscle-based controllers and the characters they use. Therefore, it provides brief but comprehensive descriptions of relevant controllers (with a special focus on their muscle-related components). It also provides a general discussion on musculoskeletal modelling, with thorough descriptions only of the models these controllers use. For more details on state-of-the-art muscle modelling, the reader is invited to consult [LGK*10]. We would also like to underline the fact that most of the frameworks perform muscle-based control; however, a few perform servo-muscle-based control, meaning that they involve characters actuated with both servos and muscles. In such frameworks, we focus on the muscle-controlled part.

2. Motivation

Several studies have shown the importance of considering internal forces when describing joint kinematics, especially in joints where

© 2016 The Authors
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

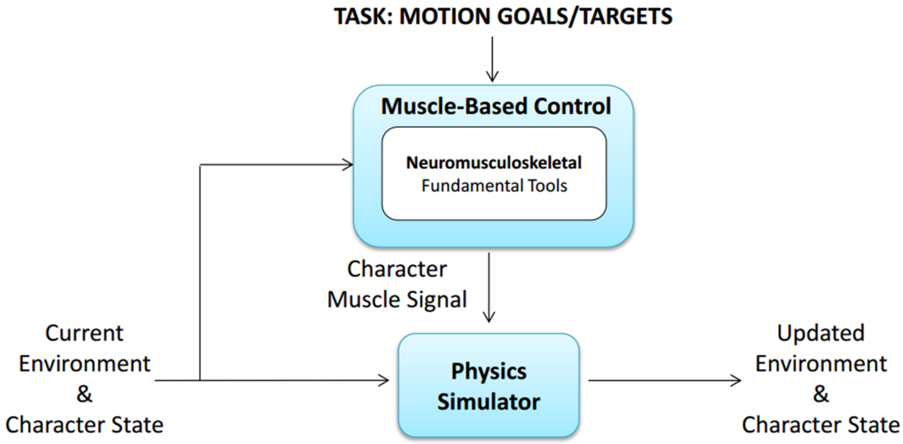

Figure 1: *Animation using a muscle-based control framework.*

complex interactions between muscle actions, soft tissues and cartilage exist (such as spinal disks, shoulders and knees) [ADR11]. Considering muscles, in particular, comprises several advantages: better stability properties and more realistic passive dynamics (Section 2.1), better estimates of energy cost or fatigue (Section 2.2), efficient control via motion mechanics (Section 2.3) and an ease to simulate musculoskeletal defects, pathologies and physical fatigue (Section 2.4). These advantages are a consequence of the non-linear properties present within the muscles, such as the force–length and force–velocity relationships (these properties are detailed in Section 3.1).

Moreover, muscles are at the centre of important motor control theories. For instance, the spring-like behaviour of muscles has been a crucial part of motor control theories such as the variants of the equilibrium point theory [BHMIG92, Fel86]. Their coordinated actions also form the basis to theories such as muscle synergies (both theories will be described in Section 3.4).

2.1. Stability and passive dynamics

The musculoskeletal system is able of achieving passive adjustments and these adjustments are robust across certain changes in the environment and perturbations [vdKdGF*09]. This is due to the fact that the non-linear properties of muscles grant the body with a first defence to counteract mechanical perturbations [HR11]. Their presence gives place to adjustments such as *preflexes*, which are mechanical responses that precede stretch reflexes when a muscle is activated.

The authors of [GvdBHZ98] investigated to what extent these properties contributed to the recovery from perturbations during locomotion by using different models with different actuators: servos, muscle models and models without force–length and force–velocity relationships. They concluded that the character actuated by muscle models (with both properties) had substantially better resistance to both static and dynamic perturbations. The role of these properties has also been investigated in the control of explosive movements, such as vertical jumping [vSB93]. The authors concluded that the force–length–velocity properties of muscles were responsible for a reasonable performance when small perturbations were applied.

2.2. Physiological feasibility and energy estimates

The inclusion of muscles motivates physiologically feasible motions. An example of this can be found in [KSK00], where the physiological infeasibility of interpolating user-input postures was shown, and it was later reduced based on muscle dynamics. Muscles also provide better estimates of energy expenditure. In [WHDK12], visual, kinematic and dynamic comparisons evidenced that walking motions synthesized via energy estimates using muscles were closer to real human data than the estimates based on torques. Recently, comparisons have also been made between torque-actuated and muscle-actuated simulations of human swimming [Si13]. The former yielded plausible results but high control gains and a smaller numerical time step were necessary.

2.3. Control via motion mechanics

Due to the presence of muscles, the mechanical system of the character is granted with the ability to accomplish control functions, not only for counteracting perturbations, but also for tasks such as human walking. For instance, instead of trying to create control models that mimic complex neural circuits (for either torque or muscle-based characters), biomechanisists have discovered that locomotion requires little control if certain principles of legged mechanics are used. In [GH10], these principles were encoded as muscle reflexes, which were used to reproduce human walking, without any higher level controller. These reflexes were inspired in spinal reflexes, which link sensory information directly into muscle activations, bypassing the inputs from the central nervous system (such reflexes will be further described in Section 3.4). Other authors have demonstrated that specific mechanical behaviours observed during walking can be encoded in a single, simple muscle reflex [PGB97, GSB03]. Nevertheless, more evidence of the performance of such legged mechanics principles, in terms of walking on uneven terrain and in various directions, is still needed. This evidence is needed to

© 2016 The Authors
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

show the extent to which reflexes can deal with such tasks without a higher level controller.

2.4. Simulation of musculoskeletal defects, pathologies and physical fatigue

Among other advantages of muscle-based control is the fact that muscles provide a natural solution to the simulation of musculoskeletal defects, pathologies and physical fatigue. By taking into account an anatomical structure (musculoskeletal system), it is easier to simulate phenomena that derive from this structure. For instance, fatigue and recovery muscle models can be used to simulate a motion where a human gradually gets tired [KSK00] by limiting the maximal force that the muscle can produce with respect to the history of muscle force [GML93, GML96]. Cost functions that minimize the force of a specific muscle can be used to synthesize motions with pain avoidance behaviours [LPKL14]. Changing muscle parameters and properties such as maximal strength can be used to weaken muscles, and generate well-known pathologies and defects [WHDK12]. Finally, injuries can also be simulated by displacing muscles [KSK00].

3. Neuromusculoskeletal Simulation Overview

Humans are actuated by muscles controlled by the central nervous system. These muscles produce forces that actuate joints to achieve a given motion. The motion is most of the time realized under external perturbations or forces (such as a voluntary pushes, the force of gravity and ground reaction forces), which the central nervous system should also compensate for.

In the biomechanics field, the neuromusculoskeletal simulation can either consist in finding ways to estimate the muscle forces from a prescribed motion or directly generating motion from computed forces. The link with physics-based animation is straightforward: by applying to a musculoskeletal model (Section 3.1), an optimal set of muscle forces (Section 3.3) with regard to motion and other requirements, it should achieve realistically a specified motion.

The following section features a general review of important tools and concepts to understand in a neuromusculoskeletal simulation: musculoskeletal modelling, simulated dynamics and muscle forces estimation. Most of the references cited here come from the field of biomechanics, as biomechanicians are the historical actors of development of the neuromusculoskeletal simulations. However, several openings and applications to animation are highlighted in order to show the strong link between both fields.

3.1. Musculoskeletal modelling

Simulating physics in a system means specifying the segments, joints, masses, inertias and actuation capacities of these systems. Musculoskeletal modelling consists in describing these different features in a convenient way in order to enable a physics simulation. This modelling is common to different types of virtual characters, such as animals, humans and even imaginary creatures.

3.1.1. *Joints and segments*

The core problem of musculoskeletal modelling is the definition of anatomically realistic segments and joints. In biomechanics, the models are often anatomically based [WSA*02, WvdHV*05] and exhibit a higher level of detail than those used in animation, as they have to be accurate enough to provide clinically relevant biomechanical quantities. In many cases, kinematical closed chains appear in the structure of the model and make both kinematics and dynamics studies more complex [PSVV07, VDH94].

In the animation field, functional DoFs (i.e. the resulting motion of the anatomical ones) are often used as a basis for the kinematical model [H-Anim] and segments directly link the considered articular centres. Nevertheless, some approaches have begun using detailed anatomically based models of certain body sections such as the neck [LT06], hand [SSB*15] and the upper body [Lee08]. Using complete body models still remains an area of growth, with only a handful of models presented for motion synthesis purposes, from which the models presented by [Si13] and [LPKL14] stand out.

3.1.2. *Lengths, masses and inertias*

One of the main issues in musculoskeletal modelling is the scaling of the model. As motion is generally recorded to be analysed (using tracking or motion capture systems), the lengths of the segments are easily computed from marker positions [LAdZR11]. However, when this information is not available, regression laws based on cadaver measurements [Dem55] can be used. Cadavers were also used to scale masses and inertias [dL96, DCV07]. However, advances in medical imaging (e.g. scanner or MRI scanners) opened the door to subject-specific scaling of musculoskeletal models and showed interesting perspectives for clinical applications [BAGD07]. Moreover, other approaches have emerged which rely only on exterior measurements of the body, such as 3D point clouds, to determine subject-specific bone geometry and motion [ZHK15].

In the models used by the control methods in this review, the parameters are manually specified for every new character. However, some exceptions exist, such as [GvdPvdS13] and [HMOA03], which automatically computed the muscle parameters by including them in optimization procedures.

3.1.3. *Servos and muscles*

Historically, as shown in Figure 2 (left), muscles were not represented and the simulation was mainly skeletal. Estimation of joint torques with servo-actuated joints was the main goal, but provided no relevant information about muscle load and fatigue. Progressively, muscles were incorporated in the simulation models as non-direct actuators of the joints.

This has also been the case for animation, where characters have evolved in level of detail, and muscle models are finding their way into motion control frameworks. A more straightforward inclusion of muscles began with the use of mass–spring systems. Contrary to the usual angular-spring dampers and proportional-derivative (PD) controllers, these systems (like muscles) are non-direct actuators of the joints. Meaning that they produce first forces, not torques, which

© 2016 The Authors
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

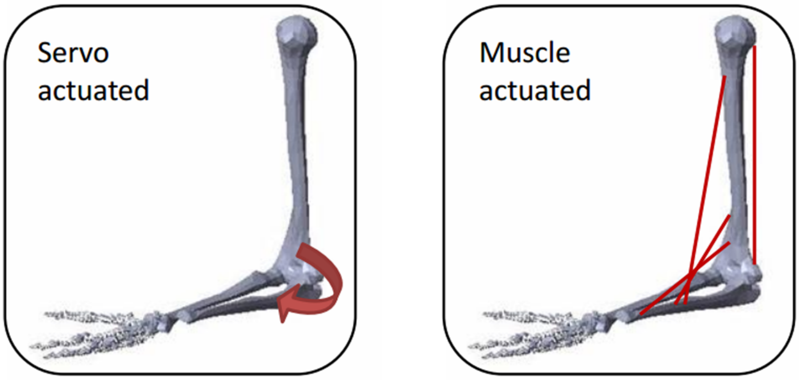

Figure 2: *A biomechanical upper limb model. On the left, the elbow is actuated by a virtual servo. On the right, the elbow is actuated by muscles.*

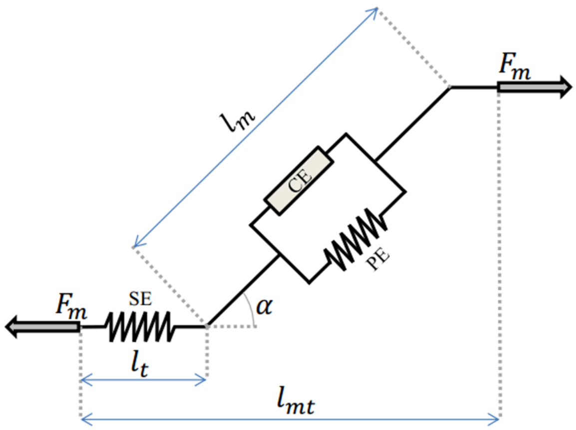

Figure 3: *Commonly used musculotendon model for musculoskeletal simulations. Inspired from [Zaj89] and [EMHvdB07].*

interact with the skeletal system to produce motion. Like muscles, these systems use force action lines, determined by their insertion or attachment site to the skeletal structure. Nevertheless, nowadays, more faithful muscle representations, such as biomechanical muscle models, are actively used.

One popular biomechanical muscle model that has been incorporated into virtual characters for force generation is the Hill muscle model [Hil38]. Although this model was developed decades ago, its current usefulness is evidenced by its various adaptations and implementations within the biomechanics community. These adaptations have come to be known as Hill-type models, such as the Hill–Stroeve model [Str96], and the widely used adaptation made in the late eighties by [Zaj89], for numerical simulations. As shown in Figure 3, the model consists of a contractile element CE (non-linear viscoelastic relationship) in parallel with a passive element PE (non-linear spring). The contractile element represents the active tension, or forces, created by the contractile proteins in the muscle, while the passive element represents the passive tension or the force that results from the elongation of the connective tissue components in the musculotendon unit. The tendon is represented by a serial non-linear spring SE of length l_t, α represents the pennation angle or the orientation of the fibres with regard to the tendon, l_m represents the muscle length and l_{mt} the length of musculotendon unit. The latter is computed by adding the muscle l_m and tendon l_t lengths. This model has widely been used even if the numerous parameters necessary to completely define its behaviour are difficult to obtain *in vivo* [HKVdH*07, IEC10].

The muscle force generation F_m of a musculotendon unit j can be summarized as the sum of the contractile and passive forces:

$$F_{mj} = [f_p(\bar{l}_m) + a_j \cdot f_l(\bar{l}_{mj}) \cdot f_v(\dot{\bar{l}}_{mj})] \cdot F_{0j}, \tag{1}$$

where f_p is the passive force relationship, a_j is the muscle activation, f_l is the force–length relationship, f_v the force–velocity relationship, F_{0j} the maximum isometric force and $\bar{l}_{mj}$ the normalized length of the muscle unit (normalization is usually made using the resting length of the muscle). Several models have been proposed to approximate the f_l and f_v relationships with regard to experimental data [RAPC10]. Example models are presented in Figure 4. The force–length relationship documents how muscle tension varies at different muscle lengths, and it is related to the 'sliding filament theory'. At a microscopic level, muscle fibres are composed of smaller structures called actin and myosin filaments that make bindings to form muscle contractions. Peak muscle force can be generated when most of these bindings or cross-bridges are created. This event corresponds to the resting length of the muscle (usually near the middle of the range of motion) [Knu07]. The force–velocity relationship explains how the force of fully activated muscle varies with velocity. It states that the force the muscle can create decreases with increasing velocity of shortening (concentric actions), while the force the muscle can resist increases with increasing velocity of lengthening (eccentric actions) [Knu07].

The tendon force f_{jt}, output of the musculotendon unit, is simply obtained by taking into account the pennation angle:

$$f_{jt} = F_{mj} \cdot \cos \alpha_j. \tag{2}$$

However, in many studies, the pennation angles are neglected.

© 2016 The Authors
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

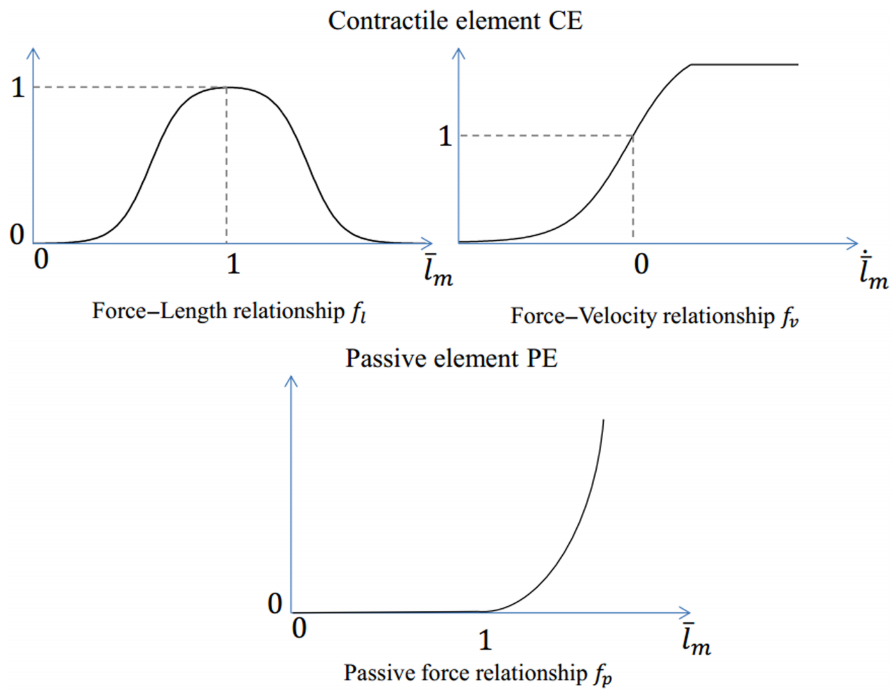

Figure 4: *Force generation capacity of muscles. Inspired from [RAPC10] and [EMHvdB07].*

Complete dynamics of the musculotendon unit also includes the activation dynamics, meaning that there is a non-linear temporal relationship between the neural excitation u_j and the effective activation of the muscle [BLMB04]. In many works [VYN05, VYN06, PD09], this non-linear relationship is approximated by a second-order differential equation, exhibiting different time constants for activation and deactivation:

$$\dot{e}_j = (u_j - e_j)/\tau_{ne}$$
$$\dot{a}_j = \begin{cases} (e_j - a_j)/\tau_{act} \,, & e \geq a \\ (e_j - a_j)/\tau_{deact} \,, & e < a, \end{cases} \tag{3}$$

where u_j is the neural excitation, a_j the muscle activation, e_j an intermediate variable, τ_{ne} the neural excitation constant time (often neglected) and τ_{act} and τ_{deact} the activation and deactivation time constants, respectively. In animation, activation dynamics is sometimes modelled using equal activation and deactivation time constants [GvdPvdS13, WHDK12].

The musculotendon unit modelling remains challenging, since changes in the tendon length affect the final muscle force, and vice versa. A proper evaluation of the muscle length should be done [MD12]. However, such a computation is costly in terms of computation time and is often simplified. For example, the algorithm used in [DRC*06] just iterates once at each simulation time step, assuming that most of the tendon effect will be obtained with only one iteration.

Besides the mechanical model presented previously, other muscle models have emerged which encompass visual characteristics such as muscle deformation or both functional and visual characteristics [LGK*10]. The models can be grouped under three techniques: geometrically based, physically based and data-driven approaches. In geometrically based approaches, muscle deformation is determined by the skeleton arrangement [CHP89, WVG97, TSC96]. In physically based approaches, both the contractile muscle forces and the changing muscle geometry are represented during contraction [NTH01, TZT09, TSB*05]. Finally, data-driven approaches directly model the skin shape that is deformed by the underlying muscle, due to data captured from the surface of subjects [ACP03, PH06, FLP14]. These models offer a next level of fidelity. However, to this date, they are not usually used for the control of virtual characters due to the fact that they would render the control computationally expensive.

3.2. Simulated dynamics

As it has been stated in [EMHvdB07], the musculoskeletal dynamics problem can been presented as follows: let us consider a musculoskeletal system with n DoFs, actuated by m muscles. The DoFs are the joint angles gathered in a vector called q. The state of such a musculoskeletal model, from a dynamics point of view, can be expressed as $(q, \dot{q})$.

The relationship between motion and forces is given by the Newton's second law of motion, which can be expressed in a matrix form as [Pan01]:

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) + R(q)F_m + E = 0, \tag{4}$$

where $M(q)$ is the mass matrix of the system, gathering masses and inertias of all the segments $(n \times n)$, $C(q, \dot{q})$ represents the

© 2016 The Authors
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

Coriolis and centrifugal effects ($n \times 1$), $G(q)$ represents the vector of gravity torques ($n \times 1$) and E represents the external forces. Finally, $R(q)F_m$ represents the action of the muscles on the joints (muscular joint torques, $n \times 1$) , where $R(q)$ is the matrix containing the muscular moment arms ($n \times m$) and F_m the muscle forces ($m \times 1$). In a dynamics simulation, such quantities can be automatically constructed by using algorithms such as the ones developed in [KL96] and [Fea14].

From Equation (4), two different problems can be derived. The first one, called *inverse dynamics*, consists in applying a specified motion and specified external forces to a musculoskeletal model and then computing the forces that generate the considered motion. The equation to solve is a reformulation of Equation (4) and can be expressed as

$$R(q)F_m = -(M(q)\ddot{q} + C(q, \dot{q}) + G(q) + E). \qquad (5)$$

The outputs of Equation (5) are the muscular joint torques that are used to define muscle forces. This equation is most of the time solved due to a top–down strategy also referred to as Newton–Euler algorithm [Win05, Fea14, RHWZ08], which considers each segment separately from distal to proximal. However, more robust methods and methods considering closed loops have also been validated to solve the inverse dynamics problem [Kuo98, vdBS08]. This approach is widely used in motion analysis, as motion capture data are a very common resource and can be used as a kinematical input in such problems.

The second problem is called *forward dynamics* and is the one that interests us the most, as it consists in generating motion from computed forces. Since no direct measurement of muscle forces is available, this approach is generally coupled with an optimization problem to compute a set of forces compatible with a given task. The equation to solve, issued from Equation (4), can be written as follows:

$$\ddot{q} = M^{-1}(q)(-C(q, \dot{q}) - G(q) - R(q)F_m - E). \qquad (6)$$

In animation, the forward dynamics problem is incorporated in a physics simulation involving collision detection (which provides external forces to apply to the system) and a numerical integration (e.g. Runge–Kutta methods) to obtain the current system state $(q, \dot{q})$ from the computed accelerations $\ddot{q}$. In biomechanics, the problem is generally solved with real external force measurements (such as ground reaction forces from force plates), and it is formulated as the optimization problem presented in the next section.

3.3. Muscle forces estimation

Most musculoskeletal models exhibit actuation redundancy ($m > n$) that leads to an infinite number of actuation solutions, as there are less equations (dynamics equations) than unknowns (muscle forces). The models may also exhibit under-actuation, which stems from the fact that a single muscle can actuate several joints simultaneously, such as bi-articular muscles.

These challenges can be solved by defining what is the optimal actuation solution, through the modelling of known motor control laws associated with a motion. Motor control is the process through which humans and animals create motions by using their neuromuscular system. This process involves the computation of higher level commands by the central nervous system to achieve specific motion goals based on sensory information regarding the environment and the current body state. These commands later excite the muscular system, creating skeletal motion [Ros91]. Several models that mimic this process are actively studied and used in the fields of neuroscience, biomechanics and robotics to generate muscle forces.

3.3.1. *Problem formulation*

A popular motor control model is the minimization of a cost function ($f(X)$) encoding task goals and bio-inspired objectives that motivate natural motion. A common formulation of this model consists in a non-linear constrained optimization problem:

$$\text{Find } X \text{ which minimizes } f(X) \qquad (7)$$

subject to,

$$g_j(X) \leq 0, j = 1, 2, \ldots, m$$
$$h_k(X) = 0, k = 1, 2, \ldots, p,$$

where the constraints enforce that:

- the computed forces solve the dynamic equations;
- the muscles are only pulling and they have physiological-based force limits;
- the computed forces may respect any additional set of unilateral (g) or bilateral constraints (h).

The constraints $g_j(X)$ and $h_k(X)$ may be specified as hard constraints (as in the formulation above), or as soft constraints (as additional cost functions). The optimization can be a static or dynamic one (often an optimal control problem [ZDG*96]). A static optimization refers to the process of minimizing or maximizing an objective function at a time instant, while a dynamic optimization refers to the process of minimizing or maximizing an objective function over an interval of time of non-zero duration.

3.3.2. *Forward, inverse and hybrid dynamics-based optimization*

This model can be used both in an inverse dynamics and forward dynamics framework. General schemes of these frameworks are featured in Figures 5 and 6.

In the *inverse-dynamics-based optimization* (Figure 5), motion and external forces are applied to a musculoskeletal model. The generated joint torques are then used in an optimization procedure that computes the muscle forces that satisfy the task, motion objectives and constraints. In biomechanics, the inverse dynamics problem is often solved by minimizing a cost function representing an energetic cost [TBB97, Pan01, EMHvdB07, PD09]. Several cost functions

© 2016 The Authors
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

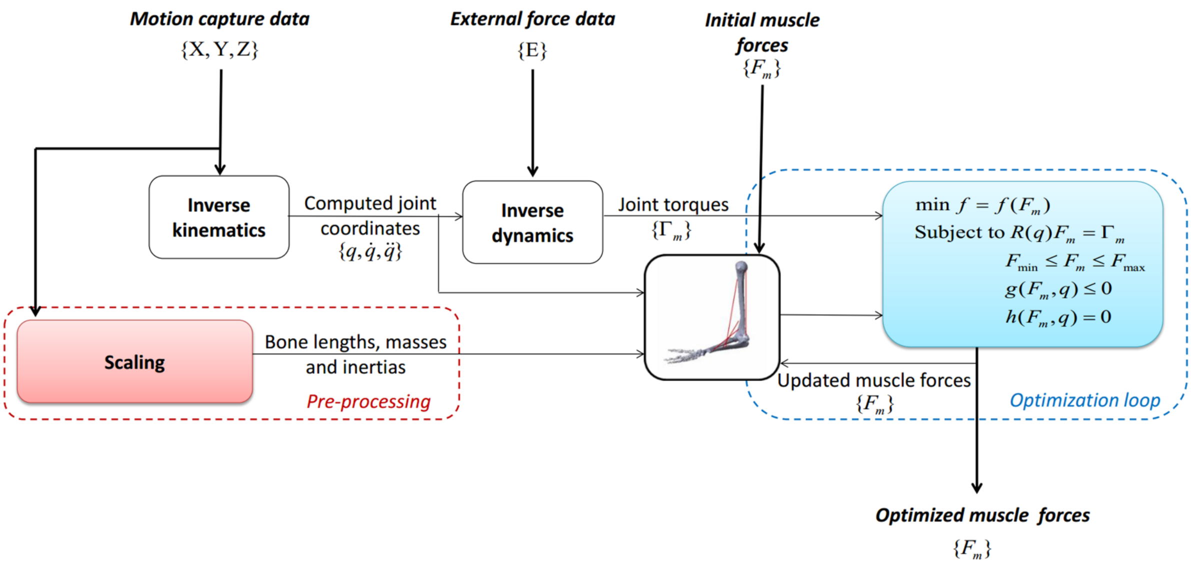

Figure 5: *Inverse-dynamics-based optimization. The optimization problem iterates until the cost function is minimized and the constraints are satisfied. Adapted from [EMHvdB07] and [PDZS*14].*

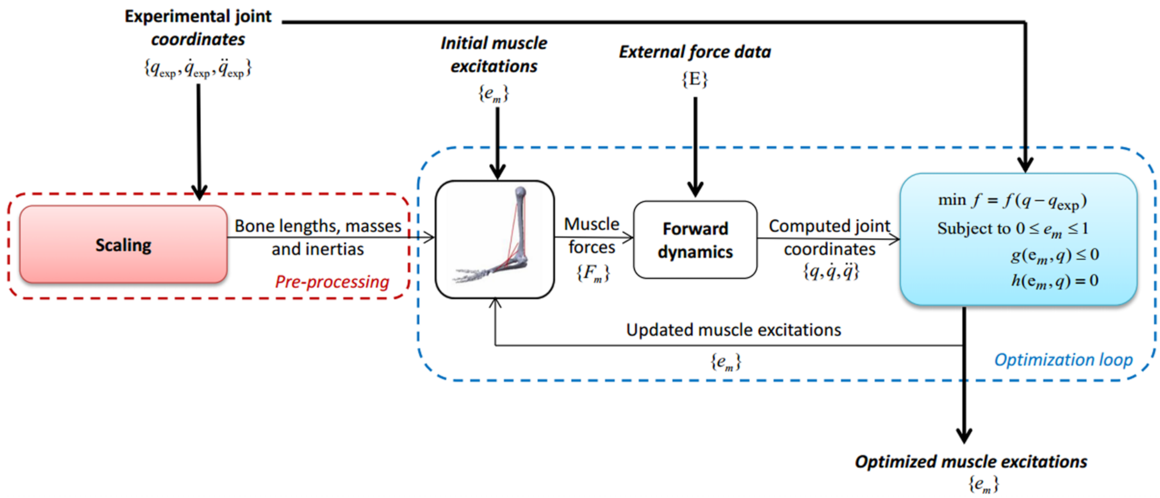

Figure 6: *Forward-dynamics-based optimization. The optimization problem iterates until the cost function is minimized and the constraints are satisfied. Excitations are often computed instead of forces for a more straightforward inclusion of muscle dynamics within the solution (activation and force generation properties are highly influential in high-performance motions). Adapted from [EMHvdB07].*

have been tested. The sum of the squared and cubed muscle forces has been classically used for gait [CB81] and upper arm [AKCM84, Cha97] motions as an image of the metabolic energy consumption, whereas a min/max criterion [RDV01] has widely been used in ergonomics applications as an image of the muscle fatigue. These examples are of importance as they are influencing the cost functions that are currently used in the muscle-based animation field. A widely used commercial software exploiting such a method for musculoskeletal analysis is AnyBody [DRC*06]. One of the purposes of using inverse-dynamics-based optimization for animation is to provide characters with a certain degree of adaptability to perturbations. Computing muscle forces from motion capture data is also interesting because it allows the simulation of new motions (e.g. fatigued or injured motions), by replaying the computed muscle forces while altering muscle parameters [KSK00, WHDK12, LPKL14]. Moreover, recent developments allow the computation of these muscle forces in real time [vdBGEZ*13]. Finally, it is worth noting that an alternative strategy to muscle excitation computation is the *muscle-based variation of Jacobian transpose control* [SADM94, GvdPvdS13], which does not use the complete inverse dynamic model, but a simpler, static equivalent. Essentially, it consists in finding a set of forces and torques in the task space which achieve a target pose, and converting these into individual muscle torques and excitations via the Jacobian transpose.

© 2016 The Authors
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

In the *forward-dynamics-based optimization* problem (Figure 6), only initial muscle excitations and external forces are applied to the model (no pre-recorded motion). The desired motion is instead directly provided to the optimization procedure to evaluate task achievement, and produce the required muscle excitations. Usual cost functions are the distance between computed kinematics (issued from the forward dynamics problem) and experimental kinematics data. A good example of a forward dynamics optimization is the computed muscle control algorithm (CMC) implemented in OpenSim [DAA*07]. Its forward simulation uses a cost function that can be based on the distance to experimental data, or on the distance to pre-defined poses issued from a planner.

Several *hybrid dynamics* methods also exist, trying to use the advantages of both methods to be mechanically and physiologically consistent. These methods basically consist of an inverse-dynamics-based optimization, with the difference that (as in the forwards dynamics problem) initial excitations are provided for some muscles through the use of electromyographic measurements [BLMB04, AM04, ARB10]. The advantage of such approach is that the dimension of the original problem is reduced by removing variables from the optimization problem. Numerous optimal control methods have also been developed to obtain realistic motions from scratch through the actuation of a musculoskeletal model [Pan01, AvdB10]. This is also of great interest for the animation field, as optimal control theory is especially well fitted to synthesize a motion between two body poses.

As previously shown, a central and important component in these frameworks is the choice of the objective function. For this reason, we have grouped the most relevant objective terms used in animation according to the categorization found in [ZW90]. In this categorization, a generalized performance criterion was proposed, which included three components: task-specific (tracking a given trajectory, minimizing jerk), neuromuscular (minimizing muscle stress, neural effort) and bone joint (minimizing contact forces, avoiding certain ranges of motion) objectives. Appendix B contains a detailed description of each of these components and a table listing relevant cost functions. A special focus has been given to the neuromuscular objectives, which represent a novelty in muscle-based control. The reader is invited to consult the reference of each controller for more detailed descriptions on the remaining objective types.

To solve the optimization problem, popular algorithms such as sequential quadratic programming (SQP) [KSK00, ZCCD06, Si13] and simplex methods have been implemented. SQP consists in modelling the non-linear optimization problem as quadratic subproblems and to use the solutions of these subproblems to find better approximations that lead to the optimum [RR09]. If the optimization is unconstrained, other authors have opted for simpler methods such as the simplex method [GT95, DZS08]. This iterative method uses a geometric shape or simplex to explore the solution space and find an optimum [RR09].

Evolutionary algorithms [dG04] have also been recently incorporated into the control of virtual characters [HMOA03, WHDK12, GvdPvdS13]. An example is the covariance matrix adaptation (CMA) [Han06], which uses a multi-variate normal distribution to explore the solution space in search of an optimum.

3.4. Motor control theories

The optimization itself can be used as a controller to generate muscle signals, or it can be used to optimize bio-inspired control laws based on motor control findings and theories (Section 4.1) such as hierarchical systems, central pattern generators (CPGs), equilibrium point theory, muscle reflexes and muscle synergies. The current section briefly presents these theories and some of the application cases found in both biomechanics and animation.

Hierarchical control systems have been designed due to studies [KSJ*00] that outline how the components or neural organs of the motor system work together to generate muscle excitations for voluntary and reflexive actions. This hierarchy has inspired multiple-level controllers in animation, as in [ZCCD06].

CPGs have been proposed for the generation of locomotion, and other kinds of behaviours (such as respiration and swallowing). CPGs are biological neural networks that produce rhythmic patterns without relying on sensory feedback or higher control centres [Hoo01]. This neural rhythmicity is generated from interactions between neurons or between currents within individual neurons. Although these networks do not rely on sensory feedback, higher control centres use this feedback to modulate the CPG outputs. The CPG models created by [Tag98] and [TYS91] for bipedal locomotion have been popular in the robotics domain [Ijs08, AT05, AT06, ENMC05] and are also beginning to be present in the world of animation, specifically for the synthesis of human swimming [Si13] and walking [HMOA03]. Nevertheless, a limitation of these generators is the fact that they only produce a limited set of motions, mainly rhythmic or periodic patterns. Using CPG-based methods alone is not enough to provide a character with a rich motion repertoire. For this reason, in applications requiring periodic and non-periodic motions, it is necessary to use an additional control strategy to handle the non-periodic aspects [Si13].

Control laws based on the *equilibrium point theory* [Fel66] have also been implemented by animators [NF02]. This theory argues that the nervous system controls movement through the specification of final equilibrium positions of the limb. The equilibrium trajectory mirrors properties of the limb and neuromuscular system, such as inertia and viscoelasticity. This trajectory is specified by virtual positions corresponding to variations in muscular activations. The muscle activations move the limb from its real position to the virtual one.

Lower level control laws, such as *muscle reflex models* [GH10], have recently started to be incorporated into character motion synthesis, as we will see in the next section [GvdPvdS13, WHDK12]. The models suggested that reflex inputs (which serve as mediators between the CNS or central nervous system and mechanical environment) dominate in contributions to muscle activations during locomotion. This supports the idea that the function of CPGs may be limited during normal locomotion. These reflexes have been included in animation as positive feedbacks of muscle fibre length and force [GSB03] that altered muscle activation. The effect of such reflexes was a reliance on compliant leg behaviour, joint overextension avoidance and improved gait stability.

Finally, a promising theory (somewhat close to the idea behind CPGs), which is yet to be actively applied within animation

© 2016 The Authors
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

frameworks, is the theory of *muscle synergies*. This theory proposes that by combining a few modules, the CNS may learn new control policies fast and efficiently, for instance, to adapt to perturbations. Evidence of this modular organization has been found due to the low dimensionality of motor signals [dP10]. The existence of such modules has been shown in human arm reaching motions [MBdF10], human postural responses [TOT07], overhead throwing [CRPSD15] and frog kicks [dSB03].

All of these theories are of interest since they have inspired many of the control methods used in animation. In Section 4.1, we will see how CPGs, muscle reflexes and the equilibrium point theory have inspired controllers in animation. In Section 4.2, we will show other control methods which take more inspiration on the work done in the domain of control and robotics. Nevertheless, a handful of these methods have also been inspired by motor control theories, such as the theory of muscle synergies.

4. Muscle-Based Control Methods

Once the character is designed using the musculoskeletal modelling and simulation dynamics concepts presented in Section 3.1 and Section 3.2, a muscle-based controller can be constructed using the force estimation frameworks and techniques featured in Section 3.3, in particular, the forward-dynamics-based optimization framework. As initially shown in Figure 1, the purpose of the muscle-based controller is the computation of adequate muscle signals (muscle excitations or muscle forces) that allow the character to achieve a set of tasks and motion goals. Specifically, we are interested in defining a controller to determine the forces F_m in Equation (6).

The motion goals for these controllers can consist of high-level goals, such as walking speed and direction, or task space and joint trajectories. Therefore, the specification of detailed motion data is not a necessity. In fact, the purpose of designing controllers is to be as independent as possible from motion data. In the first case, one sole procedure computes both kinematics and muscle signals from the high-level goal. In the latter case, either a higher level controller computes the desired kinematics and a low-level muscle controller transforms it into muscle signals, or the animator provides the kinematic trajectories.

We have grouped the controllers into two categories: controller optimization methods (Section 4.1) and trajectory optimization methods (Section 4.2). In the controller optimization methods, the optimization seeks to determine the optimal control parameters that will allow a control law to produce muscle signals that satisfy specific motion goals. Once these parameters are determined, the controllers convert desired kinematic goals into muscle signals. These control laws are based on the motor control findings and theories discussed in Section 3.4. In trajectory optimization methods, the optimization directly generates the muscle signal trajectories that accomplish the desired motion goals. Both methods attempt to optimize motions with sometimes similar cost functions (Table 2), but the controller optimization methods access the equations of motion implicitly through experience. In other words, they see the character and its environment as a 'black box', while the trajectory optimization methods access these equations explicitly. Furthermore, the controller optimization approaches have the characteristic, that (when needed) instead of computing the complete inverse dynamic model, many use a 'simpler' equivalent, such as the *Jacobian transpose* [SADM94] (Section 3.3.2). Finally, it is worth noting that to compute the muscle signals, the controller optimization methods perform the optimization offline, while the trajectory optimization methods perform it online.

A section has been devoted to each control method. Each section consists of a description of the control method, its main strengths and weaknesses and examples from the animation field. Appendix A resumes these characteristics for all controllers discussed in this review, and Appendix B summarizes the details and formulas describing the neuromuscular cost functions used by each controller.

4.1. Controller optimization methods

Controller optimization methods seek to determine a set of control parameters that will yield the desired motion goals throughout an entire period of time. These parameters depend on the specific control law, but in general they can be summarized as: feedback control law gains (such as PD controller gains, force feedback gains and spring gains) and CPG unit weights. An overview of such methods is featured in Figure 7. Animators specify the desired task ($f_{task}(t)$), neuromuscular ($f_{neuromuscular}(t)$) and bone joint objectives ($f_{bone,joint}(t)$), constraints ($g(s_m, q)$, $h(s_m, q)$), external forces (E), initial control (p_m) and character state (s_m), and finally an initial guess of the joint trajectories that fulfill the task. An optimization procedure continually updates the controller parameters and desired joint trajectories until the cost function is minimized. It is worth noting that the final desired joint trajectories might be specified by the animator, or synthesized by the optimization as in Figure 7.

Once the control parameters and joint trajectories have been determined, the control law executes online and directly generates the actuation signals that accomplish natural looking motions, while satisfying task-related objectives. Example of control laws includes: antagonistic control, PD-controllers, muscle reflexes and neural networks. The variety of motions that have been synthesized encompass human locomotion (e.g. walking, running, hopping) [GvdPvdS13, WHDK12] and postural adjustments [NF02].

4.1.1. *PD controllers and muscle reflexes*

Several approaches have made use of the fact that humans minimize muscle effort during locomotion [R*76] within their controller optimization frameworks. Some notable examples are the controllers developed by [WHDK12] and [GvdPvdS13] for synthesizing locomotion in humanoids and imaginary bipedal creatures. These approaches used an optimization procedure to determine the optimal control parameters of PD controllers (PD gains) and muscle reflexes (force and length feedback gains). The optimization had a time horizon of 10 or 20 s, and was based on a muscle effort term called the rate of metabolic energy expenditure [And99], and soft constraints to track kinematic objectives and ensure stable gaits.

© 2016 The Authors
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

Table 1: *Main muscle-based controllers for animation.*

Control type	Control space	Controller Reference	Task	Character	Model	Actuation Type	Control Strategy	Main User Input	Cost Function
Controller optimization methods	**Force space**	[NF02]	B,G,P	Human, Human Torso	Human 47 DoFs 94 Ms	Springs	Antagonistic control	Joint positions	NA
	Motor space	[GvdPvdS13]	B,L	Humanoids, Imaginary creatures	40 DoFs 27 Ms	Hill-type ([Zaj89])	PD controllers Muscle reflexes Constant excitations	Locomotion speed Locomotion direction	Pose tracking Velocity tracking Stability Muscle effort
	Hybrid space	[WHDK12]	B,L	Humanoid	30 DoFs 16 Ms (legs)	Hill-type ([Zaj89]) (Lower-body) Servos (Upper-body)	PD controllers Muscle reflexes Constant excitations	Locomotion speed	Pose tracking Velocity tracking Stability Muscle effort
Trajectory optimization methods	**Force space**	[Mil88]	L	Snakes, Worms	–	Springs	Function primitives	Function primitive parameters	NA
		[TT94]	CI,L	Fishes	91 Ms	Springs	Function primitives	Pre-defined habits	NA
		[TTL12]	B,L	Alphabet Letters	Ex. Letter I 104 DoFs 4 Ms	Springs	No function primitives	Task positions Task velocities	Pose tracking Velocity tracking Momentum tracking Base contact tracking
	Motor space	[GT95]	L,Tr	Snakes, Marine animals	Snake 126 DoFs 40 Ms	Springs	No function primitives	Locomotion speed Distance to goal	Velocity tracking Distance to goal tracking Muscle effort
		[GTH98]	L	Dolphin	12 Ms	Springs	No function primitives	Locomotion speed Distance to goal	Velocity tracking Distance to goal tracking Muscle effort
		[KSK00]	B,K,L	Human	86 Ms (legs)	Hill-type ([DLH*90, Del90])	No function primitives	Joint positions	Stability Muscle fatigue Motion feasibility Joint limit accordance
		[AHS03]	H	Human hand & Forearm	–	Other ([BH93])	Function primitives	Contraction values at keyframes	NA
		[HMOA03]	B,L	Human	19 DoFs 60 Ms	Other ([Hat77])	No function primitives	Locomotion speed Ratio of period of foot-ground contact	*Offline*: Criterion based on locomotion speed and ratio of period of foot-ground contact Energy efficiency and motion smoothness *Online*: Muscle fatigue
		[TSF05]	H	Human hand & forearm	41 Ms 16 DoFs	Hill-type ([Zaj89])	No function primitives	Joint positions Joint velocities Muscle activations	Pose tracking Muscle effort

(*Continued*)

© 2016 The Authors
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

Table 1: *Continued.*

Control type	Control space	Controller Reference	Task	Character	Model	Actuation Type	Control Strategy	Main User Input	Cost Function
		[ZCCD06]	T	Human Torso	5 MGs	Hill-type ([Zaj89])	Function primitives	Function primitive parameters	NA
		[LT06]	CI,N	Human neck & head	72 Ms	Linearized Hill-Type ([NTH01, WC00])	No function primitives	Initial and final head positions	*Offline*: Muscle effort *Online*: Minimal joint displacement
		[SKP08]	A,H	Human hand & forearm	54 Ms	Hill-Type No FV-relationship ([SKP08])	No function primitives	Task velocities	Velocity tracking Muscle effort
		[LST09]	A,T	Human upper body	147 DoFs 814 Ms	Linearized Hill-type ([LT06])	No function primitives	Joint positions Target coactivation	Muscle effort
		[LPKL14]	B,L	Human	25-39 DoFs 62-120 Ms	Hill-type ([Zaj89])	No function primitives	Joint positions Joint velocities Task positions	Pose tracking Velocity tracking Acceleration tracking Stability Muscle effort Energy efficiency Pain avoidance
		[CRPLD15]	A,H	Human arm & hand	3 DoFs 6 Ms	Hill-Type ([Hil38, RAPC10])	Function primitives	Joint positions	Pose tracking
		[SSB*15]	H	Human hand	–	Piece-wise linear muscle model ([SSB*15])	No function primitives	Task positions	Velocity tracking Muscle effort
	Hybrid space	[DZS08]	T	Human Torso	3 MGs	Hill-type ([Zaj89]; [NTH01]; [BLMB04]) Servos (Spine)	Function primitives	Audio track	Pressure tracking
		[MWTK13]	B,J,K,L	Humanoid	36 DoFs 28 Ms (legs)	Hill-type ([MD12, GH10]) (Lowerbody) Servos (Upperbody)	Function primitives	Locomotion speed Jumping height Kicking foot speed	Pose tracking Velocity tracking Virtual force reduction Dynamical consistency Muscle effort Muscle force Physiological consistency Self-collision avoidance Joint limit accordance
Hybrid method *(using: controller and trajectory optimization)*	**Motor space**	[Si13]	L	Human	163 DoFs 823 Ms	Linearized Hill-Type ([LST09])	*Locomotion tasks*: Controller optimization: Neural networks PD controllers *Non-locomotion tasks*: Trajectory optimization: No function primitives	*Locomotion tasks*: Joint positions High level CPG parameters *Non-locomotion tasks*: Trunk orientation Motion capture data	*Locomotion tasks*: Time-varying muscle length tracking *Non-locomotion tasks*: Pose tracking Motion naturalness Self-collision avoidance

© 2016 The Authors
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

Table 2: *Neuromuscular objectives used in animation.*

Type	Controller	Cost function	Formula	Terms
Effort-like	[WHDK12]	Muscle effort	$w_M J_M + w_R J_R + w_L J_L$ $J_M = \dot{A} + \dot{M} + \dot{S} + \dot{W}$	J_M, J_R, J_L = Average rate of metabolic energy expenditure, sum of torques squared, sum of squared soft joint limit torques $\dot{A}, \dot{M}, \dot{S}, \dot{W}$ = Muscle activation, muscle maintenance, muscle shortening heat rates, positive mechanical work rate w = Scalar weights
	[GvdPvdS13]	Muscle effort	$J_M = \dot{A} + \dot{M} + \dot{S} + \dot{W}$	Same terminology as above
	[GT95, GTH98]	Muscle effort	$\frac{1}{2}\left(w_a \left\lvert\frac{du}{dt}\right\rvert^2 + w_b \left\lvert\frac{d^2u}{dt^2}\right\rvert^2\right)$	u = Control signal
	[TSF05]	Muscle effort	$\frac{1}{2}\lVert a\rVert^2$	a = Muscle activation
	[LT06]	Muscle effort	$\lVert W^{-1} f_c\rVert^2$	W = Weights matrix f_c = Muscle forces
	[SKP08, SSB*15]	Muscle effort	$w_a \lVert a\rVert^2 + w_d \lVert a - a^o\rVert^2$	a = Muscle activation a^o = Activation previous timestep w_a, w_d = Regularization and damping scalar weights
	[LST09]	Muscle effort	$\frac{1}{2}\sum(w_i a_i)^2$	w_i = Variable weight (regularizes muscle activation levels)
	[MWTK13]	Muscle effort	$J_M + J_{upper}$ $J_M = \dot{A} + \dot{M} + \dot{S} + \dot{W}$ $J_{upper} = w_m \sum(f^i_{motor})^2 + \ddot{q}^T W_a \ddot{q}$	J_{upper} = Upper-body effort, f^i_{motor} = Active motor torques W_a = Diagonal weight matrix, q =Articular positions
	[LPKL14]	Muscle effort	$\lVert a\rVert^2$	a = Muscle activation
		Energy efficiency	$-\frac{1}{D}\sum_{1}^{Nfall}(J_M)$	J_M = Average rate of metabolic energy expenditure D = Total moving distance before falling $Nfall$ = Number simulation time slots before falling
Fatigue-like	[KSK00, HMOA03]	Muscle fatigue	$\int_{t_0}^{t_f}\sum\left(\frac{f_i}{f^i_{max}}\right)^2 dt$	i = ith-muscle , f_i = Muscle force f^i_{max} = Maximum muscle force, t_0, t_f = Initial and final time
Alternative muscle-based objectives	[HMOA03]	Energy efficiency and motion smoothness	$\frac{1}{S + w_o D}$	S = Specific power D = Rate of change of muscular tensions
	[MWTK13]	Muscle force physiological consistency	$\sum_i \left(F^i_{SE} - F^i_{CE} - F^i_{PE} - m_{CE}\,\ddot{l}^i_{CE}\right)$	$F^i_{SE}, F^i_{CE}, F^i_{PE}$ = Serial-elastic, contractile, passive forces m_{CE} = Pointmass between contractile element and tendon l^i_{CE} = Length contractile element
	[LPKL14]	Pain avoidance	$\sum_{1}^{Nfall}(f^i_c)$	$(f)^i_c$ = ith-muscle force

© 2016 The Authors
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

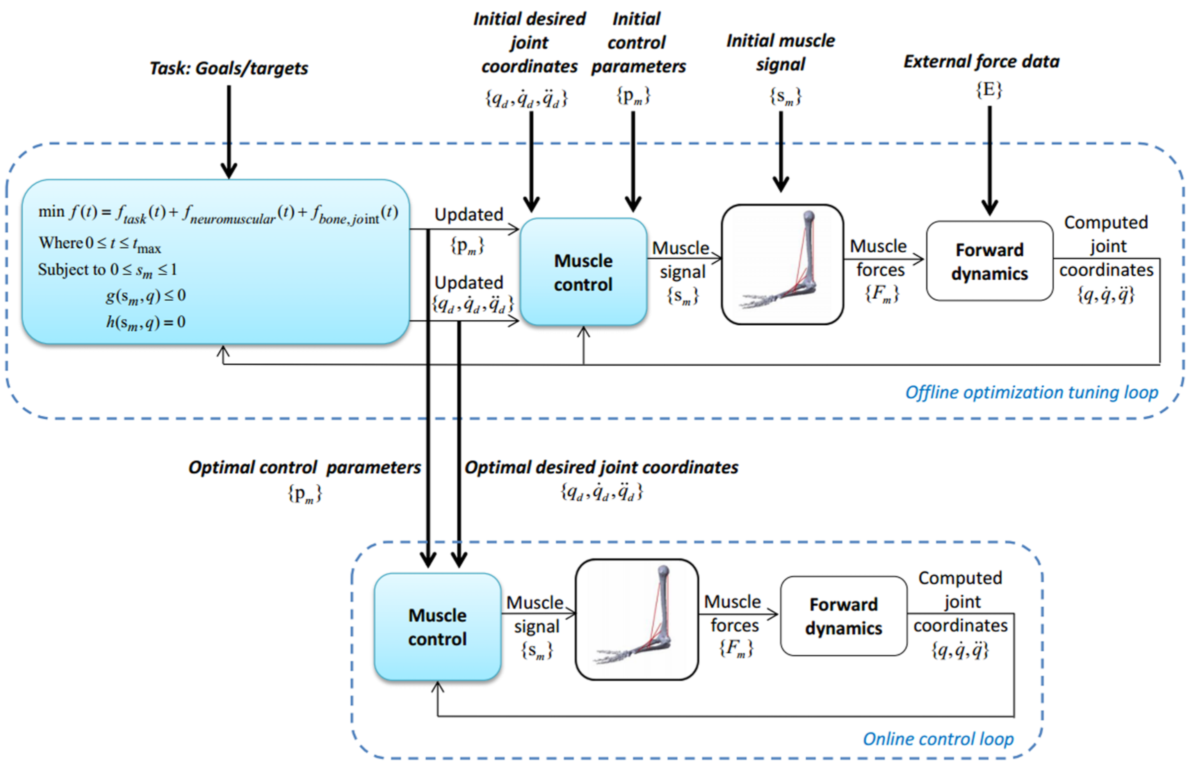

Figure 7: *Controller optimization scheme. The tuning process, which is usually made over a group of time steps [GvdPvdS13], is iterated until the cost function is minimized or the desired motion is obtained (the tuning process may also consist of a locally weighted regression [Si13] or manual parameter and target specification [NF02]). The optimal parameters and joint trajectories are then used in an online closed control loop (instead of joint trajectories [WHDK12] desired muscle lengths [Si13] might also be used).*

The PD control laws and muscle reflexes generated the muscle excitations to make the character fulfill the locomotion tasks. Geijtenbeek *et al.*[GvdPvdS13] used task space PD controllers on different body segments, inspired by the jacobian transpose control of [SADM94], while Wang *et al.*[WHDK12] employed them on each muscle in the character.

These PD controllers worked jointly with the muscle reflexes designed by [GH10]. The reflexes encoded principles of legged mechanics, such as natural joint compliance in stance phase and dorsiflexion during the swing phase. In addition to these control laws, constant excitations were also used to adjust the output of these control laws according to the gait phase, or to produce rhythmic arm and tail movements. Figure 8 features examples of the variety of creatures and motions synthesized by [GvdPvdS13]. Interestingly, in this approach, the muscle routing of the creatures was also optimized such that the task was successfully achieved.

4.1.2. *PD controllers and neural networks*

Recently, PD controllers have also been included for synthesizing human swimming motions. The authors of [SLST14] and [Si13] used a controller optimization method for periodic motions in swimming, and trajectory optimization (consult Section 4.2) for non-periodic motions. The controller optimization method consisted of neural networks (CPGs) for specific body sections and PD controllers. As opposed to the PD controllers in the previous sections, in this approach, the control gains were fixed, and the parameters in the optimization were a set of weights in CPG model.

The character's body was divided into 10 muscle groups (e.g. right leg, left leg muscles) for the CPG modelling. Each CPG was modelled as a set of non-linear differential equations, which contained desirable properties such as trajectory reproduction, modulation and external perturbation compensation. The networks produced desired time-varying muscle lengths for specific swimming modes, and the PD controllers converted them into muscle activations. The learning process of these networks was carried out by an incremental locally weighted regression (ILWR) [SA98]. This process sought the minimization of an error criterion based on the desired muscle lengths obtained from kinematic data of swimming.

The use of muscle groups, and a higher level controller to modulate each CPG, simplified the control task. For instance, turns were induced by decreasing the activation amplitudes of the muscles on one side of the body relative to muscles on the opposite side.

4.1.3. *Antagonistic control*

The authors of [NF02] used this methodology on a controller based on the *equilibrium point hypothesis* proposed in [Fel66] and

© 2016 The Authors
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

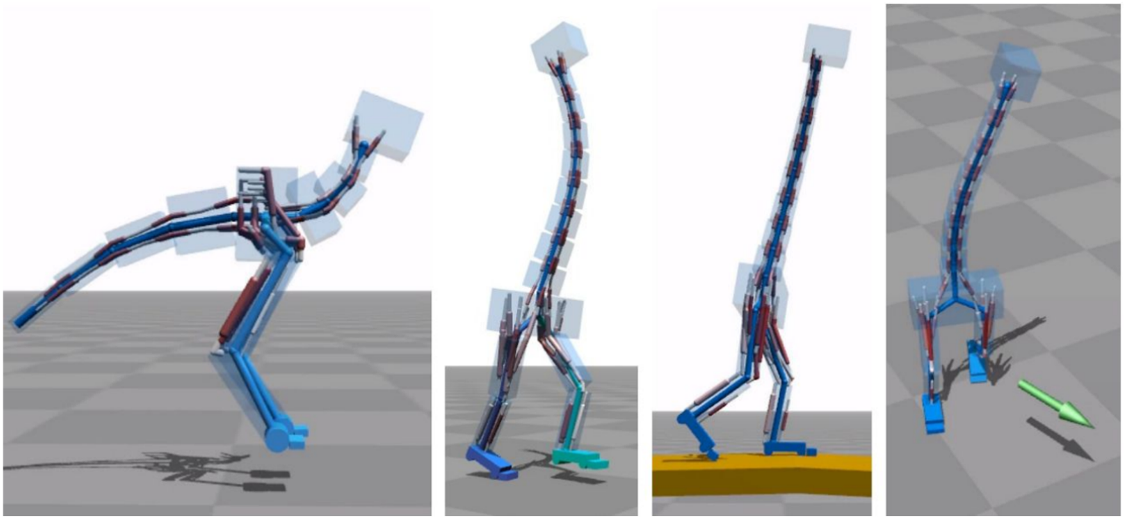

Figure 8: *Locomotion of muscle-based bipeds [GvdPvdS13].*

introduced in Section 3.3. However, an alternative to optimization was used in order to determine the set of parameters that achieved the desired motion. In this approach, each degree of freedom of a human skeleton was actuated by two angular springs representing the antagonistic grouping of muscles around joints. Movement was achieved by varying the equilibrium point of each joint; the equilibrium point was defined as the point where the sum of the forces acting on the joint equalled to zero. The variation of the equilibrium point was made by adjusting the spring parameters according to desired angles specified by the animator and a method that took samples of external forces and recalculated the parameters.

4.1.4. *Summary*

Similarly to controller optimization methods in the joint space, these controllers have generated impressive results in terms of skills (list of possible motions of the character) and robustness to external perturbations. However, because muscles are used instead of servos, new cost functions (such as muscle effort) have been included within the parameter optimization procedure, which further motivate motion realism. These improvements have been evidenced at both the kinematic and dynamic level by [WHDK12].

Controller optimization methods have allowed the implementation of models of biomechanical mechanisms. The use of these laws, such as muscle reflexes [GH10], has generated well-known events during walking, such as joint compliance in stance phase, and dorsiflexion during the swing phase [WHDK12, GvdPvdS13]. This represents an important advantage, since bio-inspired control laws can directly generate desired muscle and joint behaviour at specific stages during the motion.

One of its main drawbacks is computational efficiency. Computation times still need to be improved, since to synthesize 10 s of animation, the controllers yielding the most impressive results require approximately 10 h of tuning [WHDK12].

Nevertheless, we believe that considering and modelling muscle groups and synergies (Section 3.4) could aid in overcoming this setback, by explicitly establishing relationships among muscles and reducing the number of control parameters [ADN*13].

Finally, another interesting aspect of some of these control frameworks was the strategy used for ensuring balance of the characters. For instance, in the case of [GvdPvdS13], trunk stability was maintained by using feedback rules to control orientation in sagittal and coronal planes with respect to the centre of mass velocity and a target heading. Balance was also enforced due to the application of a SIMBICON-style balance correction [YLvdP07] to determine leg orientation in the sagittal and coronal plane. The authors of [WHDK12] also made use of the SIMBICON balance feedback laws to adjust desired hip target angles. Lastly, the approach developed by [NF02] used automatic balance controllers based on ideas from the balancing simulations of [Woo98].

4.2. Trajectory optimization methods

The objective of trajectory optimization methods is to generate control variable trajectories that minimize or maximize a measure of performance while also respecting a set of constraints. Such methods are for now mostly used in robotics. In the domain of character animation, these methods remain mostly a static optimization, solving a given set of equations at each discrete step. Alternatively, they can also be referred to as 'model-predictive control methods' when they are used online and with a finite time horizon.

As detailed in Section 3.3, these optimizations are usually formulated as non-linear constrained optimization problems that use bio-inspired cost functions, such as muscle effort or muscle fatigue. A schematic overview of this approach is featured in Figure 9. Generally, the animator specifies the desired task ($f_{task}(t)$), neuromuscular ($f_{neuromuscular}(t)$) and bone joint objectives ($f_{bone,joint}(t)$), external forces (E) and initial muscle signals (s_m). The optimization

© 2016 The Authors
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

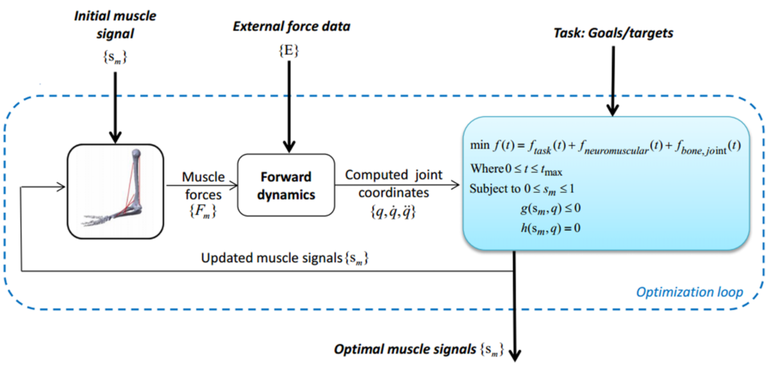

Figure 9: *Trajectory optimization methods. The optimization process directly computes the muscle control signals according to the minimization of a cost function and targets. These signals can be computed at each time step (static optimization) or in a defined simulation period (dynamic optimization or optimal control). Some approaches include additional components, such as neural networks to generate the desired joint stimuli [HMOA03, LT06].*

iterates until the cost function is minimized, producing the optimal muscle signals that fulfill the task.

Among the variety of motions that have been synthesized through this approach are locomotion patterns [Mil88, GT95, GTH98, KSK97, MWTK13, TTL12], and human upper body movements such as breathing [ZCCD06, DZS08], arm flexion [LST09] and hand movements [SKP08, AHS03].

We have distinguished two types of trajectory optimization methods: those that rely on the assumption of a specific function (periodic or polynomial functions) as a control trajectory, and those that do not. The following sections explain each type and provide further insight into how these controllers are used in animation.

4.2.1. *Trajectory optimization based on function primitives*

One of the simplest motion control strategies consists in synthesizing motions through the generation and tuning of periodic signals. These controllers are mainly used for the generation of oscillatory motions, such as those seen in the locomotion of fishes, worms, snakes and in human chest motions. These signals generally drive spring-like muscles, which are gathered into muscle groups to reduce the number of controlled variables.

Early implementations manually tuned periodic functions to generate the desired motions [TT94, Mil88, ZCCD06]. Tu and Terzopoulos [TT94] controlled artificial fishes by converting a desired swim speed into a spring contraction amplitude and frequency. This mechanism was based on the observation that the speed of most fishes can be proportional to the amplitude and frequency of the tail's lateral oscillation.

The authors of [Mil88] used controllers that produced sine waves to generate waves of compression, which replicated the elastic deformation present in the locomotion of snakes and worms. Sine waves have also been used jointly with step functions to generate human motions. Zordan *et al.* [ZCCD06] used these functions as varying parameters to synthesize human breathing. The first parameter was a contraction ratio, which was determined according to modelled and measured human muscle contraction ratios during breathing. The second parameter was a binary timing, which was defined by the desired breathing frequency. Finally, polynomial function primitives, such as spline curves, have also served as a means for human motion synthesis. Albrecht *et al.* [AHS03] synthesized hand motions in real time by specifying muscle contraction values at keyframes and interpolating them via spline functions.

Recent implementations have automated the generation of periodic functions by using optimization procedures. An example is the torso controller of [DZS08] for synthesizing human breathing and laughing. In this approach, an optimization attempted to minimize the tracking error between a desired lung pressure, computed from an audio soundtrack, and the current pressure of the model. This process generated the parameters of a set of sine waves that were used directly as muscle activation signals.

More complex motions have also been synthesized with the use of function primitives. An example is the approach of [MWTK13], where control trajectories were encoded as splines, and trajectory optimization and spacetime constraints [WK88] were used for humanoid motion synthesis. The optimization generated joint coordinates, muscle activation signals and lengths, foot contact points and ground reaction forces. This process was driven by a muscle effort term (metabolic energy expenditure [AP99]), and additional objectives that enforced the equations of motion and encoded other desired muscle behaviours. Although contact points and ground reaction forces, which usually introduce discontinuities into the optimization, were included in the optimization, this approach was still able to achieve impressive computation times. The reason behind this was the use of the *contact invariant optimization (CIO)* framework [MTP12]. The framework smoothed out the discontinuities in the objective function by allowing foot contact points to gradually invoke ground reaction forces at a distance until a real contact was made.

Other approaches have begun incorporating motor control theories, such as muscle synergies (Section 3.4), coupled with static optimization. This is the case of the synergy-based controller for throwing motions developed by [CRPLD15]. Initial control signals

© 2016 The Authors
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

(or synergies) were first extracted from human throwing motions and used to actuate the character. Next, a synergy-based forward dynamics pipeline ensured that the desired throwing motion was reproduced. The pipeline achieved this through two adaptation stages: the first stage consisted in determining the character's unknown muscle parameters; while the second stage consisted in modifying the time-varying part (or shape) of the synergy via a static optimization, such that the desired throw was reproduced. An interesting feature of this controller was that muscle redundancy was reduced, and that the initial synergies were able to reproduce general trends in the throwing motion.

4.2.2. *Trajectory optimization without function primitives*

The majority of trajectory optimization methods discussed in this review do not make assumptions on the trajectory of the control signal, and rely more heavily on biomechanical and motor control concepts, such as the minimization of effort or fatigue. We will first discuss the controllers whose sole purpose is synthesizing rigid body motions; next, we will introduce the controllers that also model the effect of this motion on soft tissues; and we will finish by presenting controllers designed for purely soft bodies (no skeleton).

Important contributions have been made, which focus on synthesizing the rigid body motions of musculoskeletal systems. Komura *et al.* [KSK97] developed an open-loop feedforward controller for the animation of the lower limbs of muscle-based characters. This controller interpolated input postures and computed muscle forces via the inverse dynamics and prediction algorithm introduced by [CB81]. The same authors extended this approach in [KSK00], by converting input physiologically infeasible postures into feasible ones, and simulating fatigued and injured characters. The novelty with respect to their original implementation is that once muscle forces were computed (based on a muscle fatigue minimization), an evaluation took place to determine if these forces respected force limits, and if the muscles had the capacity to produce the desired motion. Infeasible motions were then converted into feasible ones through an optimization based on: the minimization of the total supplementary torques needed when motion is infeasible, stability control [Vuk90], and additional muscle-related objectives which are explained in Appendix B. Another interesting aspect is that these motions could also be easily re-targeted by changing muscle parameters, such as maximum force limits, or even removing muscles.

Feedback controllers have also been developed with an adaptability to different physiological and environmental conditions. For instance, the authors of [LPKL14] synthesized biped gaits, which were adaptable to conditions such as muscle weakness, tightness, joint dislocation, external forces and motion objectives, such as maximization of efficiency and pain reduction. The approach consisted of a muscle optimization and a trajectory optimization. From a given reference motion, the muscle optimization (which minimized muscle effort) found the optimal coordination of muscle activation levels to control the character in a per frame basis. The purpose of the trajectory optimization was to modulate the reference motion and its step locations [KH10] such that: an accurate and robust motion reproduction was achieved, or to adapt the motion to new conditions and requirements. The optimization was based on the minimization of an efficiency term (required) and a muscle force term to simulate a pain avoidance behaviour (optional). It is also worth mentioning that this is one of the first approaches in animation to compare the synthesized muscle signals with human muscle data.

Other approaches couple trajectory optimization methods (as muscle-based controllers) and neuronal networks (as joint controllers). An example is the locomotion controller developed by [HMOA03], which employed a neuronal model and an optimization procedure. The neuronal model was composed of a CPG (Section 3.4) that computed the required joint stimuli based on sensory information such as the position of the centre of gravity and joint displacements. This joint stimuli was later distributed as individual muscle forces through an online static optimization procedure that sought to minimize muscle fatigue [CB81]. Before running this model online, the optimal neural parameters were found through a genetic algorithm and evaluative criteria. This criteria contained a motion smoothness term and an energy efficiency term that considered muscle power and rate of change of muscular tensions.

Similarly, Lee and Terzopoulos [LT06] synthesized neck motions by using neural networks to generate musculoskeletal stimuli and adjusting this stimuli through a static optimization. The networks generated neck poses and stiffness signals according to desired head orientations. While the optimization generated desired neck muscle strain (deformation) and strain rates that ensured that the head converged to the desired orientations via minimal joint displacements. The outputs of the networks were combined into one feedforward signal, which was converted into muscle activations by a PD controller that was constantly monitoring the error in muscle strain and strain rate. An innovation in their design was that the networks were able to control the pose and stiffness of the neck independently, due to their offline training process. For both networks, this offline training process consisted in minimizing a muscle effort term. However, for the network controlling stiffness, the muscle forces were constrained to lie in the null space of the moment arm matrix, and therefore they did not contribute to the joint torque or affect the pose. Once the networks were trained offline, they performed their online tasks, faster than attempting to solve the corresponding optimal control problem online.

In addition to rigid body motion, some control methods also focus on synthesizing the movement of soft bodies (such as tendons and skin) and the interactions of these with rigid bodies. For instance,Tsang *et al.* [TSF05] synthesized hand motions and muscle bulging through a controller which generated muscle activations for a set of desired joint orientations. The authors employed an optimization which minimized joint tracking error and muscle effort. Furthermore, a variety of hand motions were generated by using clinically motivated heuristics for *Repetitive Strain Injuries* diagnosis.

Sueda *et al.* [SKP08] also synthesized hand motions and the movement of tendons and muscles under the skin, due to a novel biomechanical simulator that used target rigid body velocities. The simulator involved an optimization procedure to compute muscle activations and a skin deformation algorithm. The optimization was mainly led by a muscle effort term and task-related objectives which ensured a good tracking of the desired rigid body velocities. These

© 2016 The Authors
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

activations first produced muscle and tendon motion, and afterwards, rigid body or bone motion. It is worth noting that the transformation from activations to rigid body motion was enhanced by complex muscle routings or strands. These strands were modelled as cubic B-splines and were not constrained to pass through a fixed set of via points. Instead, they were allowed to slide along predefined surfaces. Allowing this motion is important, because more realistic changes in muscle length and velocity are achieved, and these changes are known to affect the force generation properties of muscles.

The work of [SSB*15] is closely related to that of [SKP08]. It consisted in a hierarchical control framework for hands and tendinous systems, which performed tasks such as writing, and could simulate clinical deformities of the hand by altering tendon parameters. At the highest level, kinematic controls were computed to track a fingertip reference trajectory, and at the lowest level, an activation controller transformed these controls into muscle activations, using the formulation of [SKP08]. A difference with the latter approach is the fact that the optimization parameters could be determined through a general-purpose learning-based approach requiring no previous system knowledge. Furthermore, the framework has an increased robustness when handling the constraints between tendons and bones, due to the application of the constrained strand framework of [SJLP11]. Finally, another novelty is the ability to deal with highly stiff strands using larger time steps, due to the assumption that strain and stress propagate instantaneously through the strand.

Hand motion has not been the only area of development. The authors of [Lee08] and [LST09] presented one of the most detailed biomechanical models of the human upper body and a controller to track a set of poses while achieving a desired level of muscle co-activation. The activations that satisfied these requirements were computed due to a static optimization procedure that minimized muscle effort, and was constrained by joint torques computed from input poses [Fea14]. The optimization was solved twice (once for each muscle in an antagonistic pair), with the difference that the antagonist activation was constrained by a torque of opposite sign. Finally, once the agonist and antagonist activations were defined, they were modified proportionally to the target co-activation. Secondary skin motion was also synthesized through the simulation technique of [Sif07]. This technique consisted in defining two meshes, a coarse one that simulated the elastic flesh deformation caused by muscle fibres, and a detailed one for rendering and collision handling.

Recently, even more complex and dynamic motions have been synthesized via trajectory optimization methods. As seen in the previous section, the authors of [Si13] synthesized human swimming through two types of controller on a very detailed musculoskeletal system: the first for periodic motions and the second one for non-periodic motions. For the non-periodic motions, such as controlling body orientation, a static optimization was implemented for the purpose of tracking references poses, and imposing motion naturalness through Gaussian process dynamical models (GDPMs) and human motion data.

The locomotion of purely soft-bodied characters has also been an area of study. One of the first examples is the controller created by [GT95] for the locomotion of highly flexible animals. The control was done at multiple levels of abstraction, which granted the characters with the ability to synthesize basic locomotion skills, remember the learned skills and combine them efficiently to perform more complex tasks. The basic locomotion skills were learned through an optimization procedure containing task achievement objectives, such as the shortest distance to a goal, and a muscle effort objective. Next, the learned signals were converted into more compact representations and used within a second optimization procedure with the objective of finding a proper combination of skills to achieve more complex behaviours. Once the compact representations were created, the method could be made to work in real time.

The computational efficiency of this controller was later improved in [GTH98]. The authors devised a simulation method that, unlike most optimal control methods, allowed the computation of gradients and fast gradient-based optimization controller synthesis. The reason behind this was that neural networks were used instead of a physics simulator at each time step. This created a cascade network structure that allowed the use of a 'backpropagation through time', which adjusted the control signals using the chain rule of differentiation within each network.

More recently, the authors of [TTL12] also developed a trajectory-optimization-based controller for soft-bodied characters. This approach sought to control the shape of soft-bodied characters (and therefore the shape of their muscles) to achieve specific locomotion tasks. The procedure used an optimization at each time step to determine muscle lengths, and a contact planner, which predicted how the desired changes in muscle contraction affected contact points, allowing the characters to slide and break contacts with the ground.

4.2.3. *Summary*

Trajectory optimization methods are until now the most common solution for muscle-based control. One of their advantages is that they are more easily adaptable to different character morphologies. The reason for this is their centralized nature. Local controllers, which can be subject-dependent (due to the fact that they are usually assigned to specific muscles or body sections), are not used. Examples of this flexibility are the works of [GT95], who implemented a controller for a variety of virtual fishes, and [TTL12], who used a variety of alphabet letters with diverse muscle arrangements.

One of the main drawbacks of trajectory optimization methods is their implementation. A considerable amount of knowledge is needed in optimization techniques and constrained dynamics. Another drawback is the fact that modelling biomechanical mechanisms (such as muscle reflexes), which would certainly aid in synthesizing motions, is not straightforward. The effect of such mechanisms could be encoded within a cost function, but implementing a model of the mechanisms (as is done by controller optimization approaches) is a more natural and efficient alternative.

The computational requirements and efficiency of these methods depend on many factors, such as the number of objectives and complexity of the character [GP12]. Computational efficiency still needs to be improved since the controllers have not reached real-time performance. However, some controllers have demonstrated impressive results, given the redundancy involved in the control problem. For example, the authors of [GTH98] and [MWTK13] achieved

© 2016 The Authors
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

convergence times comprising a few seconds or minutes, due to the use of fast gradient-based optimization methods. We believe that computation times could be further improved by considering muscle groups and muscle synergies, which would reduce the number of control parameters.

Finally, as with controller optimization frameworks, several of the frameworks in this section were conceived for tasks, such as walking and jumping, which imply the additional and crucial challenge of balancing. Different strategies for balance were used. For instance, the authors of [TTL12] were inspired by the work of [MZS09], and regulated linear, angular momentum, and the area of support of their soft-bodied characters to maintain balance. On the other hand, approaches such as [KSK00] used a stability function based on the ZMP (zero moment point) [Vuk90]. Another interesting balance strategy used by [LPKL14] was based on the work of [KH10] which consists in planning a balance-recovering reference motion instantaneously based on the estimated pendulum state.

5. Conclusions and Future Directions

Muscle-based animation is gradually enhancing character realism by introducing important biological factors through the use of muscle models and muscle-based controllers. However, the main challenge remains how to model these elements, while ensuring a set of desirable characteristics (such as motion realism and an efficient computation time).

As seen in Section 2, the beneficial effects of muscles encompass: better stability properties, more realistic passive dynamics, better estimates of energy cost or fatigue, efficient control via motion mechanics and an ease in simulation of musculoskeletal defects, pathologies and fatigue. All of these benefits have been evidenced by comparative studies and theories from the fields of biomechanics and motor control. Moreover, the presence of viscoelastic and non-linear actuators (Figure 4) supply the body with a limited amount of power. This viscoelasticity also tends to smooth the action of the muscle on the joints, and the non-linearity tends to reduce drastically the efficiency of the muscles when they are contracted in extreme positions. Another benefit of the use of muscles is that their limited force generation abilities tend to frame realistically the capabilities of the body in terms of motion. Finally, muscle routing better represents the action of the muscles on the joint than a servo, and generates a non-linear relationship between forces and torques.

We have presented a classification and description of the main techniques for muscle-based control in the animation field. This classification encompasses two general categories: controller optimization methods and trajectory optimization methods. A synopsis of this classification and key characteristics of each control approach are featured in Table 1. Moreover, in Table 2, we present the muscle-based terms used in each controller's cost function, for detailed explanations on non-muscle-based terms, we invite the reader to consult the list of references.

Each control technique contributes differently to the animation field, at the expense of certain trade-offs. The main advantage of controller optimization methods is that they allow an easier implementation of well-known biomechanical mechanisms and motor control theories through simple control laws that generate the required joint and muscle behaviour at different stages during the motion. Their main drawback is computational efficiency (Section 4.1). As stated in Section 4.2, two of the main advantages of trajectory optimization methods are a higher flexibility to adapt to different character morphologies, and lower computation times with regard to controller optimization methods although most methods have not reached real-time capabilities yet. Among the main drawbacks are a higher difficulty for implementation and for encoding simple biomechanical mechanisms.

For both control strategies, the control of the animator on the style of a motion is still not straightforward. Moreover, the presence of muscles introduces realism to the animation, which could prevent certain desired styles that go against the constraints imposed by these. Nevertheless, there exists a degree of artist control that can be done at different levels, with different impacts on the final motion. On a first level, a high-level motion goal (such as walking at a certain speed) can be specified; at a second level, other desired characteristics pertaining to the âstyleâ of the motion (such as walking in a tired or fatigued fashion) can be described in the objective functions, although this is not intuitive or easy. Therefore, some authors have opted for an indirect, but easier way of controlling style by controlling certain properties of the musculoskeletal system. For instance, the authors of [KSK00], [WHDK12] and [LPKL14] synthesized fatigued, tired, and even injured motions by changing muscle parameters.

Muscle-based control is still a young field with a high potential for realistic motion production. Nevertheless, more exhaustive studies comparing muscle and servo-based characters (such as [GvdBHZ98]) are necessary to evaluate its benefits and setbacks, and to more clearly determine what is the feasible and beneficial level of model abstraction for specific applications. On a higher level, more thorough studies comparing the effects of control complexity versus modelling accuracy on the quality of the final motion would also be interesting.

Regarding the control itself, trajectory optimization methods are, for now, the most common solution employed within the animation community. However, the implementation of controller optimization methods jointly with motor control theories (such as muscle synergies and reflexes, which are popular in robotics [ADN*13, HN05], biomechanics [GH10] and neuroscience [dP10, dSB03]) seems promising.

We believe that both methods could be enhanced by including or considering muscle groups, exploiting their functional relationships, and also by an exploration of linear and non-linear control techniques [SL91, CRPD14]. The field could also benefit from the determination of new cost functions inspired in the fields of biomechanics and motor control, such as the energy cost of head stabilization, and limb stiffening due to ground contact uncertainty. An evaluation or benchmark of the robustness between muscle-based control methods could also be interesting, since there is still no significant study comparing these methods.

Another important area of improvement is usability. Physics-based characters (in general) are scarcely employed in films, games and simulations. This implies solving many challenges among which are: reducing computation times, balance control and

© 2016 The Authors
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

enriching the motion repertoire, but most importantly, incrementing the ease of use of the controllers through achieving a better controllability and reactivity of the character animation.

Muscle-based controllers are meant to impact and enhance the physics-based animation field. Any physics-based animation system that is currently using joint torques (or virtual forces, etc.) has the potential to be extended to a muscle-actuated system, leading to many possibilities. Multi-scale models and simulations will also benefit from the recent advances in musculoskeletal animation where entities from different scales can cohabit and interact within a common environment.

Acknowledgements

We would like to thank the reviewers, for their insightful comments and suggestions, which helped enrich this work. This study was funded by the French National Research Agency (ANR) project ENTRACTE (Grant agreement: ANR 13-CORD-002-01).

Appendix A: Synopsis of Muscle-Based Controllers for Animation

A synopsis of the controller classification and key characteristics of each control approach are featured in Table 1. The controllers were classified according to *control type*, following the classification made in Section 4. An additional 'hybrid method' category was added, since the approach by [Si13] used both a controller optimization and a trajectory optimization.

The control methods are also classified by their *control space*, or the space in which their output signal is generated. This category is an indicator of the degree of detail with which the muscles are modelled. Motor space control methods generate excitations or activations to control characters with detailed muscle representations. Force space control methods generate forces to control characters with less detailed muscle representations (such as spring muscle models). Hybrid space control methods generate both motor signals and servo signals (torques) for characters that are actuated by muscles and servos.

The table also features the main *tasks* for which the controllers were employed. These can be: locomotion (L), kicking (K), jumping (J), balancing (B), gestures (G), posture adjustment (P), torso motion (T), neck motion (N), arm motion (A), hand motion (H), target tracking (Tr) and character interaction (CI).

The *models* commanded by these controllers are described by their DoFs, number of muscles (Ms) and muscle groups (MGs). The muscle *actuators* are classified by the force generation model employed as either spring elements, Hill-type models or other. In parenthesis, we also state the references on which the muscle models were based or extracted.

The most important characteristic is the *control strategy*, which indicates the muscle control method as seen in Sections 4.1 and 4.2. Next, we find the *Main User Input*. The objective of this characteristic is to denote what are the main user input commands to the control frameworks. For instance, the required user input can be compact goal (such as a desired walking speed or direction), it can also be more detailed motion data (recorded or specified by the animator), or sometimes less intuitive variables such as controller parameters directly.

Finally, the *cost function* column contains the objective functions used, if the strategy involved an optimization procedure. In the case that no cost function was used the field reads NA (not applicable).

Appendix B: Cost Functions Used by Muscle-Based Controllers for Animation

Table 2 features a list of neuromuscular objective terms, which correspond to the categorization introduced in Section 3.4, inspired by [ZW90]. We recall that this categorization is based on a generalized performance criterion which includes three components: task-specific (tracking a giving trajectory, minimizing jerk), neuromuscular (minimizing muscle stress, neural effort) and bone joint (minimizing contact forces, avoiding certain ranges of motion) objectives. Task-specific kinematics and bone joint objectives are already popular among servo-based controllers. Therefore, we focus on the neuromuscular objectives, the novelty of muscle-based controllers.

Task-specific kinematics and bone joint objectives

The task objectives mainly consist in reference tracking (pose, velocity, acceleration, moment and distance tracking) and stability control (to ensure balance and avoid sliding). Other objectives include those that attempt to motivate motion naturalness, by encouraged motion fluidity, smoothness and preferences for motions closer to the training data [Si13]. Finally, the bone joint objectives usually enforce joint limits and avoiding self-collision.

Neuromuscular objectives

These objectives were divided into those that describe muscle fatigue and those that describe muscle effort (following the distinction made by [AvdB10]). Thus, cost functions presenting muscle volume weighting and lower exponents were classified as effort-like, while cost functions without muscle volume weighting and higher exponents were classified as fatigue-like. An additional category was added for the functions that did not fit these two categories. The final classification and details on these functions is featured in Table 2.

Fatigue-like functions Muscle fatigue occurs when there is a failure to maintain a required or expected force [Edw08], and it is related to the amount of synergy between the muscles. A high synergy implies that all muscles contribute during the motion in a way that the maximum relative load of any muscle remains as small as possible. In other words, they work well together by helping each other and ensuring that no muscle works more than the rest. The authors of [KSK00] incorporated this function into an optimization to synthesize lower body motion. Another example is found in [HMOA03], where the muscle fatigue term of [CB81] was used to synthesize locomotion.

Effort-like functions Muscle effort is a substitute for muscle energy expenditure. It is related to the volume of activated muscle

© 2016 The Authors
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

tissue [AvdB10]. These functions have been more popular than fatigue-like functions, among animators. In [GT95] and [GTH98], a muscle effort objective was used, which penalized high amplitude and rapid variation in the actuation signals. The authors of [TSF05], [SKP08], [LT06] and [LST09] expressed effort through operations such as weighting, normalizing and adding up muscle activations or forces. More recent approaches [WHDK12, GvdPvdS13, MWTK13] have preferred using the metabolic energy expenditure cost functions as described by [AP99]: the sum of heat released and mechanical work done by the muscles. Others [LPKL14] have used instead a measurement of energy consumption per unit moving distance [AP01].

Alternative muscle-based terms Some control methods have muscle-based terms which do not fall into the two previous categories (effort and fatigue). An example is the approach of [MWTK13] who also ensured consistency between muscle forces and physiology. Another example is the controller of [HMOA03], who used an energy efficiency and motion smoothness term that took into account the rate of change of muscular tensions.

Appendix C: Physics Simulators

The physics simulator is one of the most important tools in muscle-based control frameworks. One of its main objectives is to perform forward dynamics calculations, such that the character's state can be retrieved and updated once the controller is in action. Controller optimization frameworks such as [GvdPvdS13], [WHDK12] and [NF02] used simulators such as ODE (open dynamics engine) [Smi06] and DANCE (dynamic animation and control environment) [SFNTH05]. Trajectory optimization frameworks, such as [ZCCD06] also used ODE, while some [GTH98, KSK97] opted for SD/FAST [HRS91], others [MWTK13] for MuJoCo [TET12] and others [CRPLD15] for MATLAB® SimMechanics. Finally, some approaches have opted for homemade simulators, such as [Si13], who developed their own multi-physics simulation framework for human swimming.

References

[ACP03] Allen B., Curless B., Popović Z.: The space of human body shapes: Reconstruction and parameterization from range scans. *ACM Transactions on Graphics 22*, 3 (July 2003), 587–594.

[ADN*13] Alessandro C., Delis I., Nori F., Panzeri S., Berret B.: Muscle synergies in neuroscience and robotics: From input-space to task-space perspectives. *Frontiers in Computational Neuroscience 7*, 43 (2013), 7–12.

[ADR11] Andersen M. S., Damsgaard M., Rasmussen J.: Force-dependent kinematics: A new analysis method for non-conforming joints. In *Proceedings of the 13th Biennial International Symposium on Computer Simulation in Biomechanics* (2011).

[AdSP07] Abe Y., de Silva M., Popović J.: Multiobjective control with frictional contacts. In *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (San Diego, CA, USA, 2007), Eurographics Association, pp. 249–258.

[AHS03] Albrecht I., Haber J., Seidel H.-P.: Construction and animation of anatomically based human hand models. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (San Diego, CA, USA, 2003), Eurographics Association, pp. 98–109.

[AKCM84] An K. N., Kwak B. M., Chao E. Y., Morrey B. F.: Determination of muscle and joint forces: A new technique to solve the indeterminate problem. *Journal of Biomechanical Engineering 106* (1984), 364–368.

[AM04] Amarantini D., Martin L.: A method to combine numerical optimization and EMG data for the estimation of joint moments under dynamic conditions. *Journal of Biomechanics 37*, 9 (2004), 1393–1404.

[And99] Anderson F. C.: A Dynamic Optimization Solution for a Complete Cycle of Normal Gait. PhD thesis, University of Texas at Austin, 1999.

[AP99] Anderson F. C., Pandy M. G.: A dynamic optimization solution for vertical jumping in three dimensions. *Computer Methods in Biomechanics and Biomedical Engineering 2*, 3 (1999), 201–231.

[AP01] Anderson F. C., Pandy M. G.: Dynamic optimization of human walking. *Journal of Biomechanical Engineering 123*, 5 (2001), 381–390.

[ARB10] Amarantini D., Rao G., Berton E.: A two-step EMG-and-optimization process to estimate muscle force during dynamic movement. *Journal of Biomechanics 43*, 9 (2010), 1827–1830.

[AT05] Aoi S., Tsuchiya K.: Locomotion control of a biped robot using nonlinear oscillators. *Autonomous Robots 19*, 3 (2005), 219–232.

[AT06] Aoi S., Tsuchiya K.: Stability analysis of a simple walking model driven by an oscillator with a phase reset using sensory feedback. *IEEE Transactions on Robotics 22*, 2 (2006), 391–397.

[AvdB10] Ackermann M., van den Bogert A. J.: Optimality principles for model-based prediction of human gait. *Journal of Biomechanics 43*, 6 (2010), 1055–1060.

[BAGD07] Blemker S. S., Asakawa D. S., Gold G. E., Delp S. L.: Image-based musculoskeletal modeling: Applications, advances, and future opportunities. *Journal of Magnetic Resonance Imaging 25*, 2 (2007), 441–451.

[BH93] Brand P. W., Hollister A.: *Clinical Mechanics of the Hand* (3rd edition). Mosby – Year Book, Inc., St. Louis, MO, 1999.

[BHMIG92] Bizzi E., Hogan N., Mussa-Ivaldi F. A., Giszter S.: Does the nervous system use equilibrium-point control to guide

© 2016 The Authors
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

single and multiple joint movements? *Behavioral and Brain Sciences 15*, 4 (1992), 603–613.

[BLMB04] BUCHANAN T. S., LOYD D. G., MANAL K., BESIER T. F.: Neuromusculoskeletal modeling : Estimation of muscle forces and joints moments and movements from measurements of neural command. *Journal of Applied Biomechanics 20* (2004), 367–395.

[CB81] CROWNINSHIELD R. D., BRAND R. A.: A physiologically based criterion of muscle force prediction in locomotion. *Journal of Biomechanics 14*, 11 (1981), 793–801.

[CBvdP10] COROS S., BEAUDOIN P., VAN DE PANNE M.: Generalized biped walking control. *ACM Transactions on Graphics 29*, 4 (July 2010), 130:1–130:9.

[Cha97] CHALLIS J. H.: Producing physiologically realistic individual muscle force estimations by imposing constraints when using optimization techniques. *Medical engineering & Physics 19*, 3 (1997), 253–261.

[CHP89] CHADWICK J. E., HAUMANN D. R., PARENT R. E.: Layered construction for deformable animated characters. *SIGGRAPH Computer Graphics 23*, 3 (July 1989), 243–252.

[CKJ*11] COROS S., KARPATHY A., JONES B., REVERET L., VAN DE PANNE M.: Locomotion skills for simulated quadrupeds. *ACM Transactions on Graphics 30*, 4 (July 2011), 59:1–59:12.

[CRPD14] CRUZ RUIZ A. L., PONTONNIER C., DUMONT G.: A bio-inspired limb controller for avatar animation. *Computer Methods in Biomechanics and Biomedical Engineering 17*, sup1 (2014), 174–175.

[CRPLD15] CRUZ RUIZ A. L., PONTONNIER C., LEVY J., DUMONT G.: Motion control via muscle synergies: Application to throwing. In *MIG '15: Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games* (Paris, France, 2015), ACM, pp. 65–72.

[CRPSD15] CRUZ RUIZ A. L., PONTONNIER C., SOREL A., DUMONT G.: Identifying representative muscle synergies in overhead football throws. *Computer Methods in Biomechanics and Biomedical Engineering 18*, sup1 (2015), 1918–1919.

[DAA*07] DELP S. L., ANDERSON F. C., ARNOLD A. S., LOAN P., HABIB A., JOHN C. T., GUENDELMAN E., THELEN D. G.: Opensim: Open-source software to create and analyze dynamic simulations of movement. *IEEE Transactions on Biomedical Engineering 54*, 11 (2007), 1940–1950.

[DCV07] DUMAS R., CHEZE L., VERRIEST J.-P.: Adjustments to mcconville et al. and young et al. body segment inertial parameters. *Journal of Biomechanics 40*, 3 (2007), 543–553.

[Del90] DELP S. L.: Surgery Simulation: A Computer Graphics System to Analyze and Design Musculoskeletal Reconstructions of the Lower Limb. PhD thesis, Stanford University, 1990.

[Dem55] DEMPSTER W. T.: Space Requirements of the Seated Operator. Wadc technical report, Wright-Patterson Air Force Base, Ohio, 1955.

[dG04] DE GARIS H.: Introduction to evolutionary computing. *Evolutionary Computation 12*, 2 (2004), 269–271.

[dL96] DE LEVA P.: Adjustements to Zatsiorsky-Seluyanov's segment inertia parameters. *Journal of Biomechanics 29* (1996), 1223–1230.

[DLH*90] DELP S. L., LOAN J. P., HOY M. G., ZAJAC F. E., TOPP E. L., ROSEN J. M.: An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures. *IEEE Transactions on Biomedical Engineering 37*, 8 (1990), 757–767.

[dP10] DAVELLA A., PAI D. K.: Modularity for sensorimotor control: Evidence and a new prediction. *Journal of Motor Behavior 42*, 6 (2010), 361–369.

[DRC*06] DAMSGAARD M., RASMUSSEN J., CHRISTENSEN S. T., SURMA E., DE ZEE M.: Analysis of musculoskeletal systems in the anybody modeling system. *Simulation Modelling Practice and Theory 14*, 8 (2006), 1100–1111.

[dSB03] D'AVELLA A., SALTIEL P., BIZZI E.: Combinations of muscle synergies in the construction of a natural motor behavior. *Nature Neuroscience 6*, 3 (2003), 300–308.

[DZS08] DILORENZO P. C., ZORDAN V. B., SANDERS B. L.: Laughing out loud: Control for modeling anatomically inspired laughter using audio. *ACM Transactions on Graphics 27*, 5 (December 2008), 125:1–125:8.

[Edw08] EDWARDS R. H. T.: *Human Muscle Function and Fatigue*. John Wiley & Sons, Ltd., Chichester, UK, 2008, pp. 1–18.

[EMHvdB07] ERDEMIR A., MCLEAN S., HERZOG W., VAN DEN BOGERT A. J.: Model-based estimation of muscle forces exerted during movements. *Clinical Biomechanics 22*, 2 (2007), 131–154.

[ENMC05] ENDO G., NAKANISHI J., MORIMOTO J., CHENG G.: Experimental studies of a neural oscillator for biped locomotion with qrio. In *ICRA 2005: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005* (2005), IEEE, pp. 596–602.

[Fea14] FEATHERSTONE R.: *Rigid Body Dynamics Algorithms*. Springer, US, 2014.

[Fel66] FELDMAN A. G.: Functional tuning of nervous system with control of movement or maintenance of a steady posture. 2. Controllable parameters of muscles. *BIOPHYSICS-USSR 11*, 3 (1966), 565–578

[Fel86] FELDMAN A. G.: Once more on the equilibrium-point hypothesis (λ model) for motor control. *Journal of Motor Behavior 18*, 1 (1986), 17–54.

[FLP14] FAN Y., LITVEN J., PAI D. K.: Active volumetric musculoskeletal systems. *ACM Transactions on Graphics 33*, 4 (July 2014), 152:1–152:9.

[GH10] GEYER H., HERR H.: A muscle-reflex model that encodes principles of legged mechanics produces human walking

© 2016 The Authors
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

dynamics and muscle activities. *IEEE Transactions on Neural Systems and Rehabilitation Engineering 18*, 3 (2010), 263–273.

[Gle98] GLEICHER M.: Retargetting motion to new characters. In *SIGGRAPH '98: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques* (1998), ACM, pp. 33–42.

[GML93] GIAT Y., MIZRAHI J., LEVY M.: A musculotendon model of the fatigue profiles of paralyzed quadriceps muscle under fes. *IEEE Transactions on Biomedical Engineering 40*, 7 (1993), 664–674.

[GML96] GIAT Y., MIZRAHI J., LEVY M.: A model of fatigue and recovery in paraplegicâs quadriceps muscle subjected to intermittent fes. *Journal of Biomechanical Engineering 118*, 3 (1996), 357–366.

[GP12] GEIJTENBEEK T., PRONOST N.: Interactive character animation using simulated physics: A state-of-the-art review. *Computer Graphics Forum 31*, 8 (December 2012), 2492–2515.

[GSB03] GEYER H., SEYFARTH A., BLICKHAN R.: Positive force feedback in bouncing gaits? *Proceedings of the Royal Society of London B: Biological Sciences 270*, 1529 (2003), 2173–2183.

[GT95] GRZESZCZUK R., TERZOPOULOS D.: Automated learning of muscle-actuated locomotion through control abstraction. In *SIGGRAPH '95: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques* (1995), ACM, pp. 63–70.

[GTH98] GRZESZCZUK R., TERZOPOULOS D., HINTON G.: Neuroanimator: Fast neural network emulation and control of physics-based models. In *SIGGRAPH '98: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques* (1998), ACM, pp. 9–20.

[GvdBHZ98] GERRITSEN K., VAN DEN BOGERT A., HULLIGER M., ZERNICKE R.: Intrinsic muscle properties facilitate locomotor control-a computer simulation study. *Motor Control 2*, 3 (1998), 206–220.

[GvdPvdS13] GEIJTENBEEK T., VAN DE PANNE M., VAN DER STAPPEN A. F.: Flexible muscle-based locomotion for bipedal creatures. *ACM Transactions on Graphics 32*, 6 (November 2013), 206:1–206:11.

[H-Anim] URL: www.h-anim.org. Accessed on 25 August 2014.

[Han06] HANSEN N.: The cma evolution strategy: A comparing review. In *Towards a New Evolutionary Computation*. Springer, Berlin, Heidelberg, 2006, pp. 75–102.

[Hat77] HATZE H.: A myocybernetic control model of skeletal muscle. *Biological Cybernetics 25*, 2 (1977), 103–119.

[Hil38] HILL A. V.: The heat of shortening and the dynamic constants of muscle. *Proceedings of the Royal Society of London B: Biological Sciences 126*, 843 (1938), 136–195.

[HKVdH*07] HORSMAN M. K., KOOPMAN H., VAN DER HELM F., PROSÉ L. P., VEEGER H.: Morphological muscle and joint parameters for musculoskeletal modelling of the lower extremity. *Clinical Biomechanics 22*, 2 (2007), 239–247.

[HMOA03] HASE K., MIYASHITA K., OK S., ARAKAWA Y.: Human gait simulation with a neuromusculoskeletal model and evolutionary computation. *The Journal of Visualization and Computer Animation 14*, 2 (2003), 73–92.

[HN05] HUANG Q., NAKAMURA Y.: Sensory reflex control for humanoid walking. *IEEE Transactions on Robotics 21*, 5 (2005), 977–984.

[Hod91] HODGINS J.: Biped gait transitions. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation, 1991* (April 1991), vol. 3, pp. 2092–2097.

[Hoo01] HOOPER S. L.: *Central Pattern Generators*. John Wiley & Sons, Ltd., 2001.

[HR90] HODGINS J. K., RAIBERT M. H.: Biped gymnastics. *The International Journal of Robotics Research 9*, 2 (1990), 115–128.

[HR11] HOUK J. C., RYMER W. Z.: *Neural Control of Muscle Length and Tension*. John Wiley & Sons, Inc., 2011.

[HRS91] HOLLARS M. G., ROSENTHAL D. E., SHERMAN M. A.: *SD/FAST Userâs Manual*. Symbolic Dynamics Inc., Mountain View, CA, USA, 1991.

[IEC10] INFANTOLINO B. W., ELLIS M. J., CHALLIS J. H.: Individual sarcomere lengths in whole muscle fibers and optimal fiber length computation. *The Anatomical Record 293*, 11 (2010), 1913–1919.

[Ijs08] IJSPEERT A. J.: Central pattern generators for locomotion control in animals and robots: A review. *Neural Networks 21*, 4 (2008), 642–653.

[KH10] KWON T., HODGINS J.: Control systems for human running using an inverted pendulum model and a reference motion capture sequence. In *SCA '10: Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Madrid, Spain, 2010), Eurographics Association, pp. 129–138.

[KL96] KANE T. R., LEVINSON D. A.: *Dynamics Online*. OnLine Dynamics Inc., Sunnyvale, USA, 1996.

[Knu07] KNUDSON D.: *Fundamentals of Biomechanics*. Springer Science & Business Media, Springer, US, 2007.

[KSJ*00] KANDEL E. R., SCHWARTZ J. H., JESSELL T. M.: *Principles of Neural Science*, vol. 4. McGraw-Hill, New York, 2000.

[KSK97] KOMURA T., SHINAGAWA Y., KUNII T. L.: A muscle-based feed-forward controller of the human body. *Computer Graphics Forum 16*, 3 (1997), C165–C176.

© 2016 The Authors
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

[KSK00] KOMURA T., SHINAGAWA Y., KUNII T. L.: Creating and retargetting motion by the musculoskeletal human body model. *The Visual Computer 16*, 5 (2000), 254–270.

[Kuo98] KUO A. D.: A least-squares estimation approach to improving the precision of inverse dynamics computations. *Journal of Biomechanical Engineering 120*, 1 (1998), 148–159.

[LAdZR11] LUND M. E., ANDERSEN M. S., DE ZEE M., RASMUSSEN J.: Functional scaling of musculoskeletal models. In *Congress of the International Society of Biomechanics, ISB* (2011).

[Lee08] LEE S. H.: Biomechanical Modeling and Control of the Human Body for Computer Animation. PhD thesis, University of California at Los Angeles, 2008.

[LGK*10] LEE D., GLUECK M., KHAN A., FIUME E., JACKSON K.: A survey of modeling and simulation of skeletal muscle. *ACM Transactions on Graphics 28*, 4 (2010), 1–13.

[LKL10] LEE Y., KIM S., LEE J.: Data-driven biped control. *ACM Transactions on Graphics 29*, 4 (July 2010), 129:1–129:8.

[LPKL14] LEE Y., PARK M. S., KWON T., LEE J.: Locomotion control for many-muscle humanoids. *ACM Transactions on Graphics 33*, 6 (November 2014), 218:1–218:11.

[LST09] LEE S.-H., SIFAKIS E., TERZOPOULOS D.: Comprehensive biomechanical modeling and simulation of the upper body. *ACM Transactions on Graphics 28*, 4 (September 2009), 99:1–99:17.

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

[MBdF10] MUCELI S., BOYE A. T., D'AVELLA A., FARINA D.: Identifying representative synergy matrices for describing muscular activation patterns during multidirectional reaching in the horizontal plane. *Journal of Neurophysiology 103*, 3 (2010), 1532–1542.

[MD12] MILLARD M., DELP S.: A computationally efficient muscle model. In *ASME 2012 Summer Bioengineering Conference* (2012), American Society of Mechanical Engineers, pp. 1055–1056.

[MdLH10] MORDATCH I., DE LASA M., HERTZMANN A.: Robust physics-based locomotion using low-dimensional planning. *ACM Transactions on Graphics 29*, 4 (July 2010), 71:1–71:8.

[Mil88] MILLER G. S. P.: The motion dynamics of snakes and worms. *SIGGRAPH Computer Graphics 22*, 4 (June 1988), 169–173.

[MKHK08] MULTON F., KULPA R., HOYET L., KOMURA T.: From motion capture to real-time character animation. In *Motion in Games*, vol. 5277 of Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 2008, pp. 72–81.

[MTP12] MORDATCH I., TODOROV E., POPOVIĆ Z.: Discovery of complex behaviors through contact-invariant optimization. *ACM Transactions on Graphics 31*, 4 (July 2012), 43:1–43:8.

[MWTK13] MORDATCH I., WANG J. M., TODOROV E., KOLTUN V.: Animating human lower limbs using contact-invariant optimization. *ACM Transactions on Graphics 32*, 6 (November 2013), 203:1–203:8.

[MZS09] MACCHIETTO A., ZORDAN V., SHELTON C. R.: Momentum control for balance. *ACM Transactions on Graphics 28*, 3 (July 2009), 80:1–80:8.

[NF02] NEFF M., FIUME E.: Modeling tension and relaxation for computer animation. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (San Antonio, TX, 2002), ACM, pp. 81–88.

[NTH01] NG-THOW-HING V.: Anatomically-based Models for Physical and Geometric Reconstruction of Humans and Other Animals. PhD thesis, University of Toronto, 2001.

[Pan01] PANDY M. G.: Computer modeling and simulation of human movement. *Annual Review of Biomedical Engineering 3*, 1 (2001), 245–273.

[PB02] PULLEN K., BREGLER C.: Motion capture assisted animation: Texturing and synthesis. *ACM Transactions on Graphics 21*, 3 (July 2002), 501–508.

[PD07] PRONOST N., DUMONT G.: Dynamics-based analysis and synthesis of human locomotion. *The Visual Computer 23*, 7 (July 2007), 513–522.

[PD09] PONTONNIER C., DUMONT G.: Inverse dynamics method using optimization techniques for the estimation of muscles forces involved in the elbow motion. *International Journal on Interactive Design and Manufacturing (IJIDeM) 3*, 4 (2009), 227–236.

[PDZS*14] PONTONNIER C., DE ZEE M., SAMANI A., DUMONT G., MADELEINE P.: Strengths and limitations of a musculoskeletal model for an analysis of simulated meat cutting tasks. *Applied Ergonomics 45*, 3 (2014), 592–600.

[PGB97] PROCHAZKA A., GILLARD D., BENNETT D. J.: Positive force feedback control of muscles. *Journal of Neurophysiology 77*, 6 (1997), 3226–3236.

[PH06] PARK S. I., HODGINS J. K.: Capturing and animating skin deformation in human motion. *ACM Transactions on Graphics 25*, 3 (July 2006), 881–889.

[PSVV07] PENNESTRÌ E., STEFANELLI R., VALENTINI P. P., VITA L.: Virtual musculo-skeletal model for the biomechanical analysis of the upper limb. *Journal of Biomechanics 40*, 6 (2007), 1350–1361.

[R*76] RALSTON H. J.: Energetics of human walking. *Neural Control of Locomotion 18* (1976), pp. 77–98.

[RAPC10] RENGIFO C., AOUSTIN Y., PLESTAN F., CHEVALLEREAU C.: Distribution of forces between synergistics and antagonistics muscles using an optimization criterion depending on muscle contraction behavior. *Journal of Biomechanical Engineering 132*, 4 (2010).

© 2016 The Authors
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

[RDV01] Rasmussen J., Damsgaard M., Voigt M.: Muscle recruitment by the min/max criteriona comparative numerical study. *Journal of Biomechanics 34*, 3 (2001), 409–415.

[RH02] Reil T., Husbands P.: Evolution of central pattern generators for bipedal walking in a real-time physics environment. *IEEE Transactions on Evolutionary Computation 6*, 2 (2002), 159–168.

[RHWZ08] Riemer R., Hsiao-Wecksler E. T., Zhang X.: Uncertainties in inverse dynamics solutions: A comprehensive analysis and an application to gait. *Gait & Posture 27*, 4 (2008), 578–588.

[RM01] Reil T., Massey C.: Biologically inspired control of physically simulated bipeds. *Theory in Biosciences 120*, 3–4 (2001), 327–339.

[Ros91] Rosenbaum D. A.: *Human Motor Control*. Academic Press, San Diego, CA, 1991.

[RR09] Rao S. S., Rao S.: *Engineering Optimization: Theory and Practice*. John Wiley & Sons, 2009.

[SA98] Schaal S., Atkeson C. G.: Constructive incremental learning from only local information. *Neural Computation 10*, 8 (1998), 2047–2084.

[SADM94] Sunada C., Argaez D., Dubowsky S., Mavroidis C.: A coordinated jacobian transpose control for mobile multi-limbed robotic systems. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation, 1994* (May 1994), vol. 3, pp. 1910–1915.

[SFNTH05] Shapiro A., Faloutsos P., Ng-Thow-Hing V.: Dynamic animation and control environment. In *GI '05: Proceedings of Graphics Interface 2005* (Victoria, British Columbia, 2005), Canadian Human-Computer Communications Society, pp. 61–70.

[Si13] Si W.: Realistic Simulation and Control of Human Swimming and Underwater Movement. PhD thesis, University of California, 2013.

[Sif07] Sifakis E.: Algorithmic Aspects of the Simulation and Control of Computer Generated Human Anatomy Models. PhD thesis, Stanford University, 2007.

[Sim94] Sims K.: Evolving virtual creatures. In *SIGGRAPH '94: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques* (1994), ACM, pp. 15–22.

[SJLP11] Sueda S., Jones G. L., Levin D. I. W., Pai D. K.: Large-scale dynamic simulation of highly constrained strands. *ACM Transactions on Graphics 30*, 4 (July 2011), 39:1–39:10.

[SKP08] Sueda S., Kaufman A., Pai D. K.: Musculotendon simulation for hand animation. *ACM Transactions on Graphics 27*, 3 (August 2008), 83:1–83:8.

[SL91] Slotine J., Li W.: *Applied Nonlinear Control*. Prentice-Hall, Englewood Cliffs, NJ, 1991.

[SLST14] Si W., Lee S.-H., Sifakis E., Terzopoulos D.: Realistic biomechanical simulation and control of human swimming. *ACM Transactions on Graphics 34*, 1 (December 2014), 10:1–10:15.

[Smi06] Smith R.: Open dynamics engine v0. 5 user guide.

[SSB*15] Sachdeva P., Sueda S., Bradley S., Fain M., Pai D. K.: Biomechanical simulation and control of hands and tendinous systems. *ACM Transactions on Graphics 34*, 4 (July 2015), 42:1–42:10.

[Str96] Stroeve S.: Learning combined feedback and feedforward control of a musculoskeletal system. *Biological Cybernetics 75*, 1 (1996), 73–83.

[Tag98] Taga G.: A model of the neuro-musculo-skeletal system for anticipatory adjustment of human locomotion during obstacle avoidance. *Biological Cybernetics 78*, 1 (1998), 9–17.

[TBB97] Tsirakos D., Baltzopoulos V., Bartlett R.: Inverse optimization: Functional and physiological considerations related to the force-sharing problem. *Critical Reviews in Biomedical Engineering 25*, 4–5 (1997), 371–407.

[TET12] Todorov E., Erez T., Tassa Y.: Mujoco: A physics engine for model-based control. In *Proceedings of 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (October 2012), pp. 5026–5033.

[TOT07] Torres-Oviedo G., Ting L. H.: Muscle synergies characterizing human postural responses. *Journal of Neurophysiology 98*, 4 (2007), 2144–2156.

[TSB*05] Teran J., Sifakis E., Blemker S. S., Ng-Thow-Hing V., Lau C., Fedkiw R.: Creating and simulating skeletal muscle from the visible human data set. *IEEE Transactions on Visualization and Computer Graphics 11*, 3 (2005), 317–328.

[TSC96] Thalmann D., Shen J., Chauvineau E.: Fast realistic human body deformations for animation and vr applications. In *CGI '96: Proceedings of the 1996 Conference on Computer Graphics International* (1996), IEEE Computer Society, pp. *******166–.

[TSF05] Tsang W., Singh K., Fiume E.: Helping hand: An anatomically accurate inverse dynamics solution for unconstrained hand motion. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Los Angeles, CA, 2005), ACM, pp. 319–328.

[TT94] Tu X., Terzopoulos D.: Artificial fishes: Physics, locomotion, perception, behavior. In *SIGGRAPH '94: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques* (1994), ACM, pp. 43–50.

[TTL12] Tan J., Turk G., Liu C. K.: Soft body locomotion. *ACM Transactions on Graphics 31*, 4 (July 2012), 26:1–26:11.

[TYS91] Taga G., Yamaguchi Y., Shimizu H.: Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. *Biological Cybernetics 65*, 3 (1991), 147–159.

© 2016 The Authors
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.

[TZT09] TANG C., ZHANG G., TSUI C.: A 3d skeletal muscle model coupled with active contraction of muscle fibres and hyperelastic behaviour. *Journal of Biomechanics 42*, 7 (2009), 865–872.

[vdBGEZ*13] VAN DEN BOGERT A. J., GEIJTENBEEK T., EVEN-ZOHAR O., STEENBRINK F., HARDIN E. C.: A real-time system for biomechanical analysis of human movement and muscle function. *Medical & Biological Engineering & Computing 51*, 10 (2013), 1069–1077.

[vdBS08] VAN DEN BOGERT A. J., SU A.: A weighted least squares method for inverse dynamic analysis. *Computer Methods in Biomechanics and Biomedical Engineering 11*, 1 (2008), 3–9.

[VDH94] VAN DER HELM F. C. T.: A finite element musculoskeletal model of the shoulder mechanism. *Journal of Biomechanics 27* (1994), 593–633.

[vdKdGF*09] VAN DER KROGT M. M., DE GRAAF W. W., FARLEY C. T., MORITZ C. T., CASIUS L. R., BOBBERT M. F.: Robust passive dynamics of the musculoskeletal system compensate for unexpected surface changes during human hopping. *Journal of Applied Physiology 107*, 3 (2009), 801–808.

[vdPF93] VAN DE PANNE M., FIUME E.: Sensor-actuator networks. In *SIGGRAPH '93: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques* (Anaheim, CA, 1993), ACM, pp. 335–342.

[vSB93] VAN SOEST A. J., BOBBERT M. F.: The contribution of muscle properties in the control of explosive movements. *Biological Cybernetics 69*, 3 (1993), 195–204.

[Vuk90] VUKOBRATOVIC M.: *Biped Locomotion*. Springer-Verlag, New York, Inc., NY, 1990.

[VYN05] VENTURE G., YAMANE K., NAKAMURA Y.: Identifying musculo-tendon parameters of human body based on the musculo-skeletal dynamics computation and hill-stroeve muscle model. In *Proceedings of the 2005 5th IEEE-RAS International Conference on Humanoid Robots* (2005), 351–356.

[VYN06] VENTURE G., YAMANE K., NAKAMURA Y.: Application of non-linear least square method to estimate the muscle dynamics of the elbow joint. In *IFAC - International Conference on System Identification* (Newcastle, Australia, 2006), pp. 1168–1173.

[WC00] WINTERS J. M., CRAGO P. E. (Eds.): *Biomechanics and Neural Control of Posture and Movement*. Springer New York, New York, NY, 2000.

[WHDK12] WANG J. M., HAMNER S. R., DELP S. L., KOLTUN V.: Optimizing locomotion controllers using biologically-based actuators and objectives. *ACM Transactions on Graphics 31*, 4 (July 2012), 25:1–25:11.

[Wil87] WILHELMS J.: Toward automatic motion control. *IEEE Computer Graphics and Applications 7*, 4 (1987), 11–22.

[Win05] WINTER D. A.: *Biomechanics and Motor Control of Human Movement*. John Wiley & Sons, Inc., NJ, USA, 2005.

[WK88] WITKIN A., KASS M.: Spacetime constraints. *SIGGRAPH Computer Graphics 22*, 4 (June 1988), 159–168.

[Woo98] WOOTEN W. L.: Simulation of Leaping, Tumbling, Landing, and Balancing Humans. PhD thesis, Georgia Institute of Technology, 1998.

[WP10] WU J.-c., POPOVIĆ Z.: Terrain-adaptive bipedal locomotion control. *ACM Transactions on Graphics 29*, 4 (July 2010), 72:1–72:10.

[WSA*02] WU G., SIEGLER S., ALLARD P., KIRTLEY C., LEARDINI A., ROSENBAUM D., WHITTLE M., D'LIMA D. D., CRISTOFOLINI L., WITTE H., SCHMID O, STOKES I; Standardization and Terminology Committee of the International Society of Biomechanics: ISB recommendation on definitions of joint coordinate system of various joints for the reporting of human joint motion part I: Ankle, hip, and spine. *Journal of Biomechanics 35*, 4 (2002), 543–548.

[WvdHV*05] WU G., VAN DER HELM F. C. T., VEEGER H. E. J. D., MAKHSOUS M., ROY P. V., ANGLIN C., NAGELS J., KARDUNA A. R., MCQUADE K., WANG X., WERNER F. W., BUCHHOLZ B.: ISB recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motion–Part II: Shoulder, elbow, wrist and hand. *Journal of Biomechanics 38*, 5 (2005), 981–992.

[WVG97] WILHELMS J., VAN GELDER A.: Anatomically based modeling. In *SIGGRAPH '97: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques* (1997), ACM Press/Addison-Wesley Publishing Co., pp. 173–180.

[WZ10] WU C.-C., ZORDAN V.: Goal-directed stepping with momentum control. In *SCA '10: Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Madrid, Spain, 2010), Eurographics Association, pp. 113–118.

[YLvdP07] YIN K., LOKEN K., VAN DE PANNE M.: Simbicon: Simple biped locomotion control. *ACM Transactions on Graphics 26*, 3 (July 2007).

[Zaj89] ZAJAC F. E.: Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. *Critical Reviews in Biomedical Engineering 17*, 4 (1989), 359–411.

[ZCCD06] ZORDAN V. B., CELLY B., CHIU B., DILORENZO P. C.: Breathe easy: Model and control of human respiration for computer animation. *Graphical Models 68*, 2 (2006), 113–132.

[ZDG*96] ZHOU K., DOYLE J. C., GLOVER K.: *Robust and Optimal Control*, vol. 40. Prentice Hall, NJ, 1996.

[ZHK15] ZHU L., HU X., KAVAN L.: Adaptable anatomical models for realistic bone motion reconstruction. *Computer Graphics Forum 34*, 2 (2015), 459–471.

[ZW90] ZAJAC F., WINTERS J.: Modeling musculoskeletal movement systems: Joint and body segmental dynamics, musculoskeletal actuation, and neuromuscular control. In *Multiple Muscle Systems*. Springer, New York, 1990, pp. 121–148.

© 2016 The Authors
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd.