

Analysis of musculoskeletal systems in the AnyBody Modeling System

Michael Damsgaard ^{a,*}, John Rasmussen ^a, Søren Tørholm Christensen ^a,
Egidijus Surma ^a, Mark de Zee ^{a,b}

^a Institute of Mechanical Engineering, Aalborg University, 9220 Aalborg East, Denmark

^b Department of Health Science and Technology, Center for Sensory-Motor Interaction, Aalborg University, 9220 Aalborg East, Denmark

Available online 24 October 2006

Abstract

This paper reviews the simulation software the *AnyBody Modeling System*, which was originally developed by the authors. AnyBody is capable of analyzing the musculoskeletal system of humans or other creatures as rigid-body systems. The paper introduces the main features of the system; in particular, the inverse dynamic analysis that resolves the fundamental indeterminacy of the muscle configuration. In addition to the musculoskeletal system, a model can comprise external objects, loads, and motion specifications, thereby providing a complete set of the boundary conditions for a given task. The paper also describes the basic ideas of structured model development in AnyBody.

© 2006 Elsevier B.V. All rights reserved.

Keywords: Musculoskeletal models; Muscles; Multibody dynamics; Inverse dynamics; Simulation software

1. Introduction

The musculoskeletal systems of humans and animals are mechanically very complex and computational models must be highly simplified in order to be reasonably efficient. Typically, the musculoskeletal system is assumed to be a rigid-body system allowing for standard methods of multibody dynamics to be applied.

In addition, however, the model must have reasonable representations of the muscle geometry and the recruitment pattern of the muscles, which are both complicated issues. The muscles consist of soft tissue and they wrap about each other and the bones, ligaments, and other anatomical elements in a complicated fashion. Reasonable modeling of these geometries is essential for the mechanical model. Moreover, the muscles are activated by the Central Nervous System (CNS) by mechanisms that are not understood well enough for detailed modeling. Therefore, the modeling of these mechanisms is based on assumptions, typically some kind of optimality condition. The fundamental problem is that there are more muscles than necessary to drive the degrees of freedom of the system, which implies that there are infinitely many muscle recruitment patterns that are acceptable from a dynamical point of view. This problem is often referred to as the redundancy problem of the muscle recruitment.

* Corresponding author.

E-mail address: md@ime.aau.dk (M. Damsgaard).

Nomenclature

\mathbf{C}^0	coefficient matrix of equilibrium equations
\mathbf{d}^0	right-hand side of equilibrium equations
\mathbf{f}^0	vector of forces. f_i is the i th element
\mathbf{g}^0	vector of forces corresponding to \mathbf{v} 's basis
G	objective functions
\mathbf{J}_i	inertia tensor of the i th body
$l_i^{(oi)}$	origin-insertion length of the i th muscle
m_i	the mass of the i th body
n^0	integer number of, e.g. bodies in the system
N_i	normalizing factor, typically muscle strength
p	polynomial degree
\mathbf{p}_i	vector of Euler parameters of the i th body
\mathbf{q}	vector of position coordinates
\mathbf{q}^*	virtual positions corresponding to \mathbf{v}
\mathbf{r}_i	translation vector of the i th body
\mathbf{v}	vector of velocity coordinates
γ	right-hand side of acceleration constraints
Φ	vector of kinematic constraints violations
Φ_q	Jacobian matrix with respect \mathbf{q}
ω_i	angular velocity vector of the i th body
Quantities marked with 0 may have superscripts (M), (R) referring to ‘muscles’ and ‘reactions’, respectively.	

Musculoskeletal models can be divided into two groups: forward and inverse dynamic models. Forward dynamics computes the motion based on a predicted muscular activation. While this is attractive in the view of the detailed modeling of various physical phenomena, it is a very computationally demanding optimal control problem and requires a costly optimization to make the model perform a specific task. Inverse dynamics computes the muscle activation based on a specified task, i.e., known motion. This puts many restrictions on the model, but it is computationally much more efficient. This efficiency can be exploited to build more complex models comprising more muscles, i.e., a finer level of details of the mechanical model of the body.

The literature on biomechanics contains many such models. Refs. [1,2,17] show applications of forward dynamics and a review can be found in [19]. Many of the other references made later in this paper show applications of inverse dynamic models, which we shall discuss in more details in the following section.

The AnyBody Modeling System, which was initiated at Aalborg University by the authors, is a general modeling system for making such musculoskeletal models. It was designed to meet four goals:

- (1) It should be a modeling system, i.e., a tool that allows users to construct models from scratch or use or modify the existing models to suit different purposes.
- (2) The system should facilitate model exchange and cooperation on model development, and it should allow models to be scrutinized.
- (3) If possible, it should have sufficient numerical efficiency to allow ergonomic design optimization on inexpensive computers.
- (4) The system should be capable of handling body models with a realistic level of complexity such as that shown in Fig. 1.

Currently, AnyBody allows only for inverse dynamic analysis of the models and therefore this paper will focus on this approach. In the rest of this paper, we shall attempt to give an overview to the functionality of AnyBody, hereunder the theory behind the basic analysis capabilities and aspects of the software design.



Fig. 1. Full body model comprising several hundreds of muscles made with AnyBody.

2. Muscle recruitment

The solution of the muscle recruitment problem in the inverse dynamics approach is generally formulated as an optimization problem on the form

$$\text{Minimize } G(\mathbf{f}^{(M)}) \quad (1)$$

$$\text{Subject to } \mathbf{C}\mathbf{f} = \mathbf{d}, \quad (2)$$

$$0 \leq f_i^{(M)} \leq N_i, \quad i \in \{1, \dots, n^{(M)}\}, \quad (3)$$

where G is the objective function, i.e., the assumed criterion of the recruitment strategy of the CNS, stated in terms of the muscle forces, $\mathbf{f}^{(M)}$. G is minimized with respect to all unknown forces in the problem, $\mathbf{f} = [\mathbf{f}^{(M)T} \mathbf{f}^{(R)T}]^T$, i.e., muscle forces, $\mathbf{f}^{(M)}$, and joint reactions, $\mathbf{f}^{(R)}$. Eq. (2) is the dynamic equilibrium equations, which enter as constraints into the optimization. \mathbf{C} is the coefficient-matrix for the unknown forces and the right-hand side, \mathbf{d} , contains all known applied loads and inertia forces. The non-negativity constraints on the muscle forces, (3), state that muscles can only pull, not push, and the upper bounds limit their capability, i.e., N_i is the strength of the muscle.

The most popular form of the objective function, G , is *the polynomial criteria* (4) and a somewhat less known form, introduced by Siemienski [25], is *the soft saturation criteria* (5).

$$G(\mathbf{f}^{(M)}) = \sum_{i=1}^{n^{(M)}} \left(\frac{f_i^{(M)}}{N_i} \right)^p, \quad (4)$$

$$G(\mathbf{f}^{(M)}) = - \sum_{i=1}^{n^{(M)}} \sqrt[p]{\left(1 - \frac{f_i^{(M)}}{N_i} \right)^p}. \quad (5)$$

Both of these forms of G are stated with a variable power, p , and a normalizing function for each muscle, N_i . The normalized muscle force is often referred to as the *muscle activity*.

The most physiologically reasonable choice of N_i is some measure of the strength of the muscle. This can either be approximated by a constant or computed by some model taking the muscle's operating conditions into account. The modified Hill model introduced by Zajac [30] is a popular choice using the length and the length-rate of the muscle as input. These will be available from the kinematical analysis of the system.

More details about these criteria can be found in the literature, e.g. [8,9,14,20,24] apply the polynomial criteria with various powers, p , and Sieminski [25] introduces the soft saturation criterion with $p = 2$. Thorough comparison and discussion of different criteria are found in [6,7,10–12,23].

A third possibility is the so-called min/max formulation that takes the form

$$G(\mathbf{f}^{(M)}) = \max \left(\frac{f_i^{(M)}}{N_i} \right), \quad (6)$$

i.e., minimization of the maximal muscle activity. A number of properties make this criterion attractive compared to (4) and (5).

Firstly, it can be transformed into a linear problem, which makes it numerically efficient and possible to solve with a finite algorithm, see [10,12]. (4) and (5) also have this feature for $p = 1$; however, it is generally agreed that $p = 1$ leads to a physiologically unreasonable result, namely that the stronger muscles do all the work and the real muscles are known to share the loads whenever possible.

For higher powers, p , (4) and (5) become less and less numerically attractive. Moreover, the polynomial criteria must be equipped with the upper bound constraints in (3) on the muscle activity, whereas both the soft saturation and the min/max criteria have these constraints implicitly fulfilled for sub-maximal loads. The soft saturation criteria, however, cause numerical problems when activities are close to the upper limit, whereas the min/max criterion simply utilizes the muscles optimally so that the activities do not exceed the limit before becoming absolutely unavoidable. This way of handling of the upper bound makes (6) numerically attractive too.

Finally, one should notice that Rasmussen et al. [23] showed that (4) and (5) converge towards each other and towards (6) for increasing power, p . In addition, we notice that the activity's dependency on the magnitude of external load converges towards a linear function for large p and it is indeed a linear relationship for the min/max criterion, see [23].

This convergence is an important result in the view that Challis and Kerwin in [7] compared the polynomial criteria with experiments and found the best agreement for very large powers ($p = 100$). Numerically, this is equivalent to a solution obtained using (6).

Apart from being numerically attractive, the criterion in (6) also appears to be physiologically attractive. Assuming that muscle fatigue and activity are proportional, the criterion postpones fatigue as much as possible; in other words it is a minimum fatigue criterion. Based on this line of argumentation, the criterion (6) was chosen as the foundation of the inverse dynamics analysis in the AnyBody software. It should, however, be emphasized that none of the presented criteria have been generally proven superior over the others from a physiological point of view.

Some believe that (6) leads to too much muscle synergism [12], and indeed it does exploit the muscles with very poor working conditions, i.e., small moment arms, to a degree that may be questionable. Also a minor numerical difficulty of (6) should be mentioned. This min/max criterion inherently contains some indeterminacy for certain groups of sub-maximally activated muscles. This may be obvious since the objective only involves the maximally activated muscles, and it must be dealt with in order to obtain a unique solution out of (6) for complex models. This is solved by an additional criterion by [4], whereas a dedicated iterative solution, which is implemented in AnyBody, is suggested by [10 and 12].

3. Mechanical model

The mathematical model of the mechanical system must produce the equations of motion in the form of (2). We adopt a general multibody system dynamics approach using a set of Cartesian coordinates for each body. This is a choice of generality and ease of implementation over efficiency of the kinematical analysis. This is a

reasonable choice because the kinematics is a minor part of the analysis, where we typically deal with many more muscles than segments considering, for instance, a full body model.

All segments of the biomechanical system are modeled as rigid bodies,¹ neglecting effects such as the wobbly masses of soft tissues. We, more or less, adopt the formulation in the textbook by Nikravesh [18]. The position of the i th body is described by the coordinates $\mathbf{q}_i = [\mathbf{r}_i^T \mathbf{p}_i^T]^T$, where \mathbf{r}_i is the global position vector of the center of mass and \mathbf{p}_i is a vector of four Euler parameters. The velocity of the bodies is defined as $\mathbf{v}_i = [\dot{\mathbf{r}}_i^T \omega_i'^T]^T$, where the vector ω_i' is the angular velocity of the body measured in the body-fixed reference frame.

The kinematical analysis is carried out in terms of all the Cartesian coordinates by solving a set of imposed kinematical constraints of the form

$$\Phi(\mathbf{q}, t) = \mathbf{0}, \quad (7)$$

where $\mathbf{q} = [\mathbf{q}_1^T \cdots \mathbf{q}_n^T]^T$ is the assembled coordinate vector for all n segments. The explicit time, t , indicates that some of the constraints are kinematical drivers in addition to normal holonomic constraints arising from the joints. In the case of inverse dynamic analysis the imposed constraints must specify the motion completely, implying that we have a full set of equations in (7). Note that, we must also include the unity constraints on the Euler parameters. Eq. (7) is generally a non-linear system of equations and it is solved by a modified Newton–Raphson scheme. Subsequently, we solve the linear velocity and acceleration constraints, (8), but in terms of \mathbf{v} and $\dot{\mathbf{v}}$ instead of time-derivatives of \mathbf{q} :

$$\Phi_{\mathbf{q}^*} \mathbf{v} = -\Phi_t \quad \text{and} \quad \Phi_{\mathbf{q}^*} \dot{\mathbf{v}} = \gamma(\mathbf{q}, \mathbf{v}, t), \quad (8)$$

where $\Phi_{\mathbf{q}^*}$ is the Jacobian constraint with respect to \mathbf{q}^* . \mathbf{q}^* contains a virtual set of positions that correspond to \mathbf{v} . They are not meaningful as finite values due to the rotational entries in \mathbf{v} , but as infinitesimal values in differentiations they make sense.

Now, we know the motion completely in \mathbf{q} , \mathbf{v} , and $\dot{\mathbf{v}}$. We can now turn towards the target of setting up the dynamics equilibrium (2). For each segment, we have the Newton Euler equations, which in this case may take the form

$$\begin{bmatrix} m_i \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}'_i \end{bmatrix} \dot{\mathbf{v}}_i + \begin{bmatrix} \mathbf{0} \\ \tilde{\omega}'_i \mathbf{J}'_i \omega'_i \end{bmatrix} = \mathbf{g}_i, \quad (9)$$

where m_i and J'_i are the mass and the inertia tensor referring to the centroidal body-frame, respectively. The right-hand side, \mathbf{g}_i , is the forces, having six entries, firstly three forces and then three moments in body-fixed coordinates. It consists of muscle forces, $\mathbf{g}_i^{(M)}$, reaction forces, $\mathbf{g}_i^{(R)}$, and known applied loads, $\mathbf{g}_i^{(\text{app})}$, that may depend explicitly on \mathbf{q} , \mathbf{v} , and t . $\mathbf{g}_i^{(M)}$ and $\mathbf{g}_i^{(R)}$ enter (2) on the left-hand side, whereas the remaining entries in (9) enter \mathbf{d}_i ; thus the full right-hand side of (2) is assembled as $\mathbf{d} = [\mathbf{d}_1^T \cdots \mathbf{d}_n^T]^T$, where

$$\mathbf{d}_i = \mathbf{g}_i^{(\text{app})} - \begin{bmatrix} m_i \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}'_i \end{bmatrix} \dot{\mathbf{v}}_i - \begin{bmatrix} \mathbf{0} \\ \tilde{\omega}'_i \mathbf{J}'_i \omega'_i \end{bmatrix}. \quad (10)$$

Similar to the unknown forces, $\mathbf{f} = [\mathbf{f}^{(M)T} \mathbf{f}^{(R)T}]^T$, their coefficient matrix, \mathbf{C} , can be partitioned according to muscle and reaction forces, i.e., $\mathbf{C} = [\mathbf{C}^{(M)} \mathbf{C}^{(R)}]$ that define $\mathbf{g}^{(M)} = \mathbf{C}^{(M)} \mathbf{f}^{(M)}$ and $\mathbf{g}^{(R)} = \mathbf{C}^{(R)} \mathbf{f}^{(R)}$.

$\mathbf{C}^{(R)}$ is in principle the transposed of the Jacobian constraint, $\Phi_{\mathbf{q}^*}$. This follows from standard forms of the constrained equations of motion for a multibody system where $\mathbf{f}^{(R)}$ is given by Lagrange multipliers associated with the constraints, see e.g. [18]. A constraint imposed on the motion by a mechanical device corresponds to a reaction force component in that device, i.e., each row in $\Phi_{\mathbf{q}^*}$ corresponds to a column in $\mathbf{C}^{(R)}$. However, the kinematical constraints in (7) and (8) also contain pure motion specification of the system's degrees of freedom. We shall therefore exclude such motion constraints from $\mathbf{C}^{(R)}$ so that $\mathbf{C}^{(R)}$ becomes the subset of the columns from the transposed Jacobian, $\Phi_{\mathbf{q}^*}$, that are associated with the real mechanical devices. The neglected columns correspond to the system's degrees of freedom and these will be kinetically supported by muscles.

¹ In biomechanics we speak of segments instead of (rigid) bodies, because “body” can be confused with the human body.

The muscle coefficient matrix, $\mathbf{C}^{(M)}$, requires a geometric model of the muscles. We model the muscles geometrically as elastic strings spanning between two or more points and in cases wrapping over rigid obstacles. It is beyond the scope of this paper to describe the wrapping model, but in the simple case without wrapping, we can express the muscle's *origin-insertion length* as $l^{(oi)} = |\mathbf{r}_i^{(p)} - \mathbf{r}_j^{(p)}|$, where $\mathbf{r}_i^{(p)}$ and $\mathbf{r}_j^{(p)}$ are the positions of the spanned points that depend on \mathbf{q} . Any other model of the muscle path must similarly provide us with this length as a function $l^{(oi)}(\mathbf{q})$ and its time-derivative for calculation of the strength, N_i . By the principle of virtual work, we can show that the coefficients in $\mathbf{C}^{(M)}$ are the derivatives of $l^{(oi)}$ with respect to the system coordinates in \mathbf{q}^* . These derivatives shall be denoted by $l_{i,q^*}^{(oi)}$. Firstly, we can express the virtual work produced by the muscles as the sum of muscle forces times their virtual length change:

$$\delta W = \sum_{i=1}^{n^{(M)}} \delta l_{i,q^*}^{(oi)} f_i^{(M)} = \delta \mathbf{q}^{*T} \sum_{i=1}^{n^{(M)}} l_{i,q^*}^{(oi)} f_i^{(M)} = \delta \mathbf{q}^{*T} \begin{bmatrix} l_{1,q^*}^{(oi)} & \dots & l_{n^{(M)},q^*}^{(oi)} \end{bmatrix} \mathbf{f}^{(M)}. \quad (11)$$

Secondly, we can express the same virtual work as the scalar product of the generalized force vector for all muscles, $\mathbf{g}^{(M)}$, and the virtual change of the system coordinates \mathbf{q}^* :

$$\delta W = \delta \mathbf{q}^{*T} \mathbf{g}^{(M)} = \delta \mathbf{q}^{*T} \mathbf{C}^{(M)} \mathbf{f}^{(M)}. \quad (12)$$

Comparison of (11) and (12) shows that $l_{i,q^*}^{(oi)}$ is indeed the i th column of $\mathbf{C}^{(M)}$.

We have now established all entries of the muscle recruitment problem in (1)–(3). In order to speed up the actual solution, we can optionally reduce the number of equilibrium equations in (2). Since there are no other conditions on the reaction forces, $\mathbf{f}^{(R)}$, than equilibrium equations, we can apply a standard factorization to eliminate elements of $\mathbf{f}^{(R)}$ using (2). We shall, however, not go into further details about this.

4. The AnyBody software

The AnyBody software consists of two applications, a Windows graphical user interface (GUI) and a console application, that both have the same modeling facilities but differ in the ways they can be used. The console application can be called from other programs whereas the GUI application contains much more facilities for viewing the model and its results, thereby providing a better foundation for manual model development and analysis.

Modeling in AnyBody is done by a text-based input. For this purpose a special modeling language named AnyScript has been developed. A text-based user input has been chosen for two reasons: (1) From the software developer's point-of-view, it is easy to develop and maintain and (2) we believe that it is the only way to meet the goals of the AnyBody system mentioned in the Introduction. In particular, Goals 1 and 2 require a very versatile and flexible input.

4.1. The AnyScript modeling language

AnyScript is a declarative, object-oriented language for development of multibody dynamics models, particularly models of the musculoskeletal system. An AnyScript model is roughly divided into two main sections:

- (1) The model section containing the definition of the mechanical system, the body and the surrounding objects, i.e., the boundary conditions.
- (2) The study section containing lists of analyzes and other operations that can be performed on the model. These can then be executed from the software.

The declarative nature of the language means that the language has a number of predefined classes that the user can create objects from. The predefined classes comprise (1) basic data types, such as numbers and strings, (2) mechanical objects types such as segments, joints of various types, drivers, forces, and muscles, and (3) operational and model management classes.

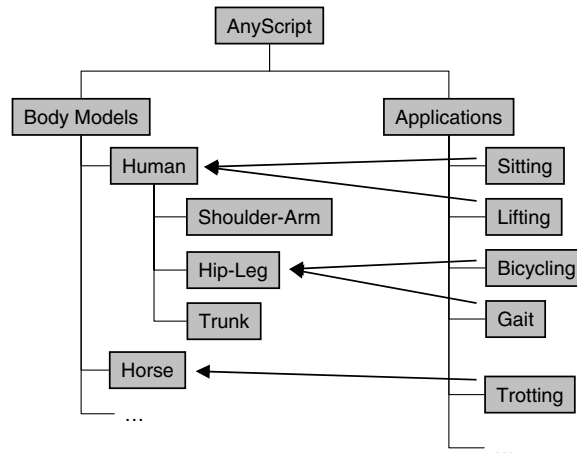


Fig. 2. Recommended structure to facilitate model exchange. The tree structures show the grouping of the models whereas the arrows examples of how applications use block from the Body Model repository. The labels in the boxes are merely examples and each Body Model box will typically comprise several versions with different features and complexity. Fig. 3 shows a practical example.

The user cannot use operational code like ‘do’ loops and ‘if-then-else’ clauses and neither can new classes with additional functionality be defined.² Nevertheless, classes that function as containers of others do exist and they play an important role in structuring large models hierarchically.

The study section of the model allows for specification of various operations to be performed on the model such as kinematical analysis, kinetic analysis, muscle calibration, and systematic parameter variations. Studies can refer to the entire model or to subsections of the model. From a software design point-of-view, the definition of “studies” and “operations” as classes in the modeling language has enabled a clear relationship between the modeling data structure and the user interfaces of the software. Whenever a new “operation” or “study” is implemented in the basic data structure, it is immediately available in the modeling language as well as in the user interfaces of the software.

Also the graphical appearance of the model in the viewer of the GUI is programmed by the user in AnyScript by means of special objects with visualization capabilities. This puts the user in full control of the view via the model input.

4.2. Structuring models for multiple purposes

Structure is essential when creating large models such as the full body model of Fig. 1. AnyScript contains functionality for structuring models in a tree structure much similar to a file system. Folder-objects can be utilized to contain specific parts of the model and information can be accessed from many places through references. Typically, the data of large models are organized in several files and these can simply be combined by C-style including statements.

These features allow the user to structure the data according to the nature of the model and the practical use of the model. In particular, a model can easily be built by several modelers working on different files, and models and model parts can be exchanged.

It appears that Goals 1 and 2 from the Introduction have been reached. However, the system provides much freedom for the user for structuring the models in different ways, and this may prevent the interfacing of differently structured model parts with each other. If, for instance, one user has developed an arm model, and another user a hand model, it is likely that they will want to combine these, but different model structures might make this impossible. To facilitate model merging, the AnyBody Research Group has developed a model structure, which splits the model into two distinct parts (Fig. 2):

² Future versions of AnyBody will get extended possibilities for user input of this kind.

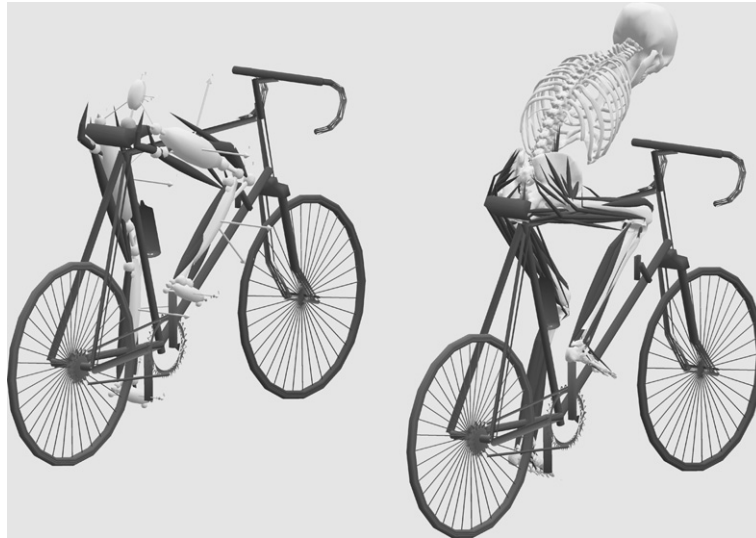


Fig. 3. A typical example of a structured model: the two bicycles are identical applications within the same model and are combined with a simple, 2-D lower extremity and a more complicated, 3-D lower extremity, respectively.

- (1) *The body model.* This part contains segments, joints, muscles, and other anatomical data, but no boundary conditions. Additionally, it may contain setups for calibrating parameters of the body model. Typically, muscle parameters need adjustment for given body anthropometrics.
- (2) *The application model.* Application-specific data about movements, loads, and external objects, such as tools, bicycles, or the like, are placed in this part.

The idea is that users can exchange body models and connect them with different types of applications as illustrated in the diagram of Fig. 2.

As illustrated in Fig. 2, the body model itself is broken into parts. The idea of this is that different scientists can build models of different parts of the body and put them together as a virtual Frankenstein's monster. This is important in view of the large effort that must go into building and validating good models. The question then remains in view of the large effort that must go into building and validating good models. The question then remains of how the elements that cross the interface between two body models or between a body model and the application can be handled. How can the developer of a hand be sure that the developer of the arm has provided muscle attachment points on the arm for the muscles spanning the wrist? To solve this problem, the AnyScript language has been equipped with a facility to semantically allow addition of the necessary elements outside the basic objects declaration. This means that the hand model to some extent can contain the necessary additions to the arm model to make the parts compatible. Fig. 3 is a picture of a bicycle model exploiting the structured body model repository.

5. Example

In this section, we shall show an application of AnyBody for gait modeling. This example is not intended to be a scientific validation of the methods, but merely a demonstration to accompany the description in the previous sections.

Gait is a very relevant application, since it is an obvious choice for validation of the methods. Moreover, gait is an important clinical application. Gait analysis plays an important role in the diagnostics of various diseases and for rehabilitation. Designers of leg prostheses and implants naturally see gait as an important case, perhaps the single most important from a functional point of view.

For the demonstration purpose here, we shall use a publicly available data set by Vaughan et al. [29]. In this data set, positions of 15 markers attached to the lower extremity as well as a full set of ground reaction forces on the two feet are available. These have been measured in a gait laboratory using video camera based motion

capture techniques with reflective markers and two force platforms in the floor. The force platforms capture all six degrees of freedom of the ground reaction force.

With these data as input, it is possible to analyze a limited model comprising the lower extremity only. This can easily be assembled in AnyBody from the building blocks in the repository [26]. The model is scaled in

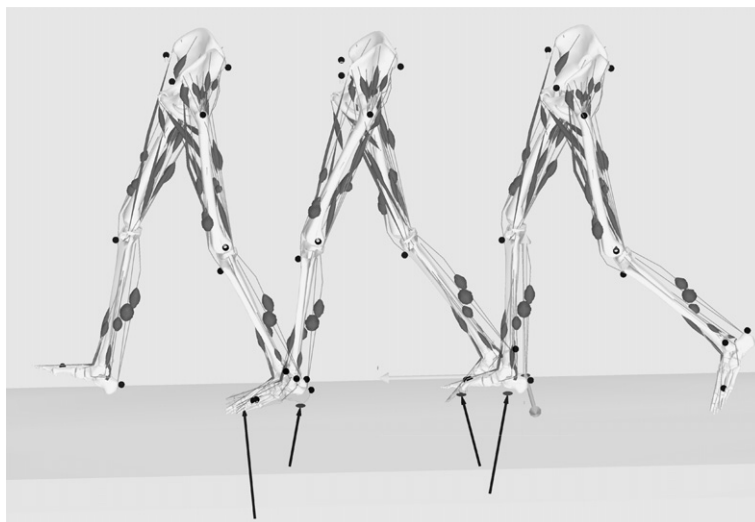


Fig. 4. Three frames of the gait model driven by data from Vaughan et al. [29]. The dark spheres are the markers.

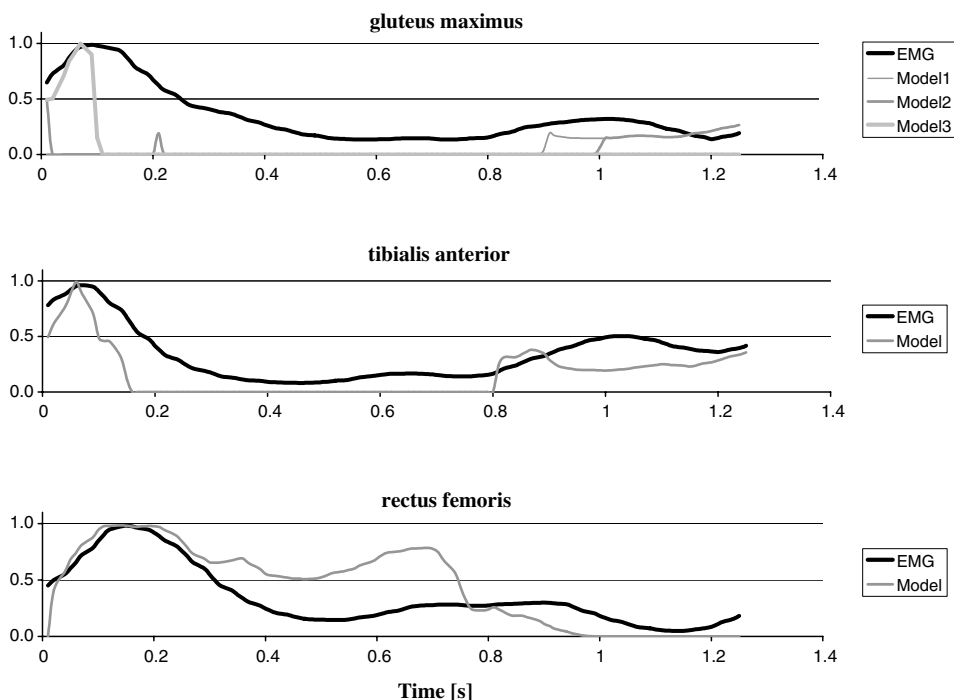


Fig. 5. Comparison of measured and simulated muscle activity. The measured EMG values are from Vaughan et al. [29] and the “Model” values are simulated activities, i.e., muscle force divided by its strength. EMG is short for electromyography, i.e., the electrical signals that can be measured from active muscles. All values are normalized to 1.0 over the period of time, so they are comparable. Time equal to zero is right leg heel strike.

order to fit the dimensions of the person given from the marker positions relative to each other. We assume that all markers are attached to rigid segments, thereby neglecting soft tissue related motion.

The model of the lower extremity has 18 degrees of freedom. Therefore, following all three times 15 marker coordinates are practically impossible; the system is in principle kinematically over-determined. Instead, we manually construct 18 kinematical constraints; most of them constrain a single marker coordinate of the model to be equal to the measurement, but a few of them actually use a combination of marker coordinates. By a careful choice of marker constraints, the motion is reconstructed.

The kinetic boundary conditions of the legs consist of the measured ground reaction force, which is applied to the feet, and a six degrees of freedom reaction force at the pelvis; the latter accounts for the upper body being truncated from the model. The model is depicted in Fig. 4.

In Fig. 5, the computed muscle activities together with the EMG signals measured by Vaughan et al., are depicted for three selected muscle groups. These curves illustrate how good agreement we can expect. Periods of activation can to some degree be resembled by the simulation. The magnitude of the activations seems worse, but it should be noticed that there is in general no obvious and well-established relationship between EMG signal levels and real muscle force, so this fact is not only due to inadequacies in the model. Measuring real muscle forces is indeed very difficult.

Notice that in one case there is more than one model output, because the particular muscle is divided into multiple branches. Also in the measurements, the EMG signal is more related to the activity in the parts of a large muscle closest to the electrodes, but more detailed information about electrode location has not been made available in the data used here.

6. Conclusions

We have in this paper described the functionality and the principal considerations behind the AnyBody Modeling System; in particular, we have sought to explain how the four system design goals from the Introduction have been met. Meeting all four goals, we believe, makes AnyBody a unique software system for the presented type of analysis. To our knowledge, no other software offers similar muscle recruitment analysis capabilities together with general model building facilities.

A number of commercially available software packages do, however, deal with modeling of the body. Many of these are merely geometric models, occasionally referred to as digital manikins, which for instance function together with CAD systems aimed at investigating the geometric compatibility between the human and a product. Examples are Jack and eM-Human by UGS, Human Builder by Safework, and Ramsis by Human Solutions.

Musculoskeletal modeling capabilities more like in the AnyBody Modeling System are only found in a limited number of available systems. The most known packages are probably SIMM/FIT by Musculographics, Inc. [16], BRG.LifeModeler by Biomechanics Research Group, Inc. [5], and Armo by G-sport, Inc. [13]. Armo does actually offer similar muscle recruitment analysis but with a limited model building interface. SIMM/FIT and BRG.LifeModeler are both general modeling systems, which are fundamentally based on forward dynamics engines in contrast to AnyBody. Both of them do also offer some facilities for inverse dynamic analysis, but they are in their current versions significantly different from the muscle recruitment analysis in AnyBody.

We shall not attempt to speculate on which of the methods in these and other packages could be generally better. All available methods for this type of musculoskeletal analysis are still in a state where their accuracy and efficiency depend highly on the actual application as well as the topology and parameters of the mechanical model being used.

The efficiency of the muscle recruitment solver in AnyBody has made it possible to handle models of full body complexity. The current full body model, Fig. 1, contains more than 400 muscles and a single time step can be solved in a few seconds on a standard personal computer. This allows for systematic parameter studies and optimization, which gives hope for computer-assisted design of man-driven machines, tools, and exercises like that demonstrated in [21,22].

The scientific search for the “real” and general muscle recruitment criterion is ongoing and it may never be established. Neither of the presented criteria are based upon detailed knowledge about the real control system (the Central Nervous System), but are rather based upon an overall assumption about its optimal function.

This may sound weak and less accurate than we are used to in typical engineering applications. The optimality approach does, however, fulfill basic conditions such as the dynamic equilibrium and the better of the criteria do provide physiologically reasonable results. Therefore, this should be considered as a reasonable approach while regarding the accuracy of the results with scrutiny.

Indeed biomechanical systems are generally difficult to handle accurately, also in the experimental setups that are used to validate computational methods. Muscle activation and in particular muscle forces cannot be measured accurately and the nature of the system makes it impossible to measure all muscles. Naturally, this is a problem in the search for validation of computational models, but it also gives the models a special importance, since they in many cases are the only way to estimate certain valuable information such as the internal forces in the body. Many of the references for this paper contain comparisons of inverse dynamics methods to experimental results similar to the above example; for instance [15,28] contain results produced using AnyBody.

The optimality assumption and the use of inverse dynamics in general imply some restrictions on the use of the formalism. We neglect certain properties of the control system such as muscle activation dynamics and we assume optimality. We must therefore restrict the methodology to relatively slow and skilled motion, i.e., tasks that the human is familiar with. The gait example is a case where these assumptions can be expected to hold. For unskilled tasks, one cannot expect the same degree of optimality. Whether the methodology can be extended to cover certain types of unskilled motion and more dynamical effects is indeed an interesting question. The first experiments with combinations of forward and inverse dynamic methods are being done these years, see for instance [27], and maybe such approaches can eliminate some of the above-mentioned restrictions, retaining some efficiency from the inverse dynamics approach.

Modeling humans and other creatures is a very demanding task and no single scientist or even institution could expect to accomplish this by themselves. Experiences from many years of research must be combined, and therefore it has been essential in the design of AnyBody to facilitate collaboration between scientists about the task of developing good body models. To meet this demand for model exchange, the AnyBody Research Group has made and maintains a public repository of models [26].

Acknowledgements

The development of the first versions of AnyBody was supported by Aalborg University, grants from the Danish Research Council, and projects partially financed by Ford Werke and Ford Forschungszentrum Aachen, Germany. AnyBody Technology A/S [3], now develops and distributes the system. AnyBody is a trademark of AnyBody Technology A/S.

References

- [1] F.C. Anderson, M.G. Pandy, A dynamic optimization solution for vertical jumping in three dimensions, *Computer Methods in Biomechanics and Biomedical Engineering* 2 (1999) 201–231.
- [2] F.C. Anderson, M.G. Pandy, Dynamic optimization of walking, *Journal of Biomechanical Engineering* 123 (5) (2001) 381–390.
- [3] AnyBody Technology A/S. Available from: www.anybodytech.com.
- [4] J.C. Bean, D.B. Chaffin, A.B. Schultz, Biomechanical model calculation of muscle contraction forces: a double linear programming method, *Journal of Biomechanics* 21 (1988) 59–66.
- [5] Biomechanics Research Group, Inc. Available from: www.lifemodeler.com.
- [6] B.M. van Bolhuis, C.C.A.M. Gielen, A comparison of models explaining muscle activation patterns for isometric contractions, *Biological Cybernetics* 81 (1999) 249–261.
- [7] J.H. Challis, D.G. Kerwin, An analytical examination of muscle force estimations using optimization techniques, *Proceedings of the Institution of Mechanical Engineers* 207 (1993) 139–148.
- [8] R.D. Crowninshield, Use of optimization techniques to predict muscle forces, *Journal of Biomechanical Engineering* 100 (1978) 88–92.
- [9] R.D. Crowninshield, R.A. A Brand, Physiologically based criterion of muscle force prediction in locomotion, *Journal of Biomechanics* 14 (1981) 793–801.
- [10] M. Damsgaard, J. Rasmussen, S.T. Christensen, Inverse dynamics of musculo-skeletal systems using an efficient min/max muscle recruitment model, in: *Proceedings of IDETC: 18-th Biennial Conference on Mechanical Vibration and Noise*, Pittsburgh, 2001.
- [11] J. Dul, G.E. Johnson, R. Shiavi, M.A. Townsend, Muscular synergism—I. On criteria for load sharing between synergistic muscles, *Journal of Biomechanics* 17 (1984) 663–673.

- [12] E. Forster, Predicting Muscle Forces in the Human Lower Limb during Locomotion. Ph.D. Thesis, Ulm University, Germany, VDI Research Report, 17, 241, VDI Verlag, Düsseldorf, 2003, ISBN 3-18-324117-x.
- [13] G-sport, Inc. Available from: www.gsport.co.jp.
- [14] W. Herzog, Individual muscle force estimations using non-linear optimal design, *Journal of Neuroscience Methods* 21 (1987) 167–179.
- [15] K. Meijer, J.A. van Geenhuizen, H.T.J. Liebrand, L.I. Meijer, R.J. Pas, M. de Zee, H.H.C.M. Savelberg, The functional consequences of muscle adaptation: a simulation in AnyBody, in: *Proceedings of the International congress on computational bioengineering*, Zaragoza, Spain, 2003.
- [16] MusculoGraphics, Inc. Available from: www.musculographics.com.
- [17] R.R. Neptune, Optimization algorithm performance in determining optimal controls in human movement analysis, *Journal of Biomechanical Engineering* 121 (1999) 249–252.
- [18] P.E. Nikravesh, *Computer-Aided Analysis of Mechanical Systems*, Prentice-Hall Inc., Englewood Cliff, NJ, 1988.
- [19] M.G. Pandy, Computer modeling and simulation of human movement, *Annual Review of Biomedical Engineering* 3 (2001) 245–273.
- [20] A. Pedotti, V.V. Krishnan, L. Stark, Optimization of muscle force sequencing in locomotion, *Mathematical Biosciences* 38 (1978) 57–76.
- [21] J. Rasmussen, M. Damsgaard, Optimization of biomechanical multibody systems, in: W. Gutkowski, Z. Mróz (Eds.), *Proceedings of The Second World Congress of Structural and Multidisciplinary Optimization*, vol. 2, Institute of Fundamental Technological Research, Warsaw, Poland, 1997, ISBN 83-905454-7-0.
- [22] J. Rasmussen, M. Damsgaard, S.T. Christensen, E. Surma, Design optimization with respect to ergonomic properties, *Structural and Multidisciplinary Optimization* 24 (2002) 89–97.
- [23] J. Rasmussen, M. Damsgaard, M. Voigt, Muscle recruitment by the min/max criterion – a comparative study, *Journal of Biomechanics* 34 (3) (2001) 409–415.
- [24] H. Rohrlé, R. Scholten, C. Sigolotto, W. Sollbach, H. Kellner, Joint forces in the human pelvis-leg skeleton during walking, *Journal of Biomechanics* 17 (1984) 409–424.
- [25] A. Siemienski, Soft saturation, an idea for load sharing between muscles, Application to the study of human locomotion, in: A. Cappozzo, M. Marchetti, V. Tosi, (Eds.), *Proceedings of the Symposium “Biocomotion: a century of research using moving pictures”*, Promograph, Rome, Italy, 1992, pp. 293–303.
- [26] The AnyBody Research Project and the AnyBody Repository. Available from: www.anybody.aau.dk.
- [27] D.G. Thelen, F.C. Anderson, S.L. Delp, Generating dynamic simulations of movement using computed muscle control, *Journal of Biomechanics* 36 (2003) 321–328.
- [28] M.S. Thompson, M. Voigt, M. de Zee, Evaluation of lower extremity musculo-skeletal model using sit-to-stand movement, in: *19th Congress of the International Society of Biomechanics*, Dunedin, 6–11 July, 2003.
- [29] C.L. Vaughan, B.L. Davis, J.C. O’Connor, *Dynamics of gait*, 1st ed., Human Kinetics Publishers, 1992. The Man data set provided at the website of the International Society of biomechanics, <http://isbweb.org/o/content/view/66/73/>.
- [30] F.E. Zajac, Muscle and tendon: properties, models, scaling and application to biomechanics and motor control, *Critical Reviews in Biomedical Engineering* 14 (1989) 359–411.