DESIGNING A GENERAL SOFTWARE SYSTEM FOR MUSCULOSKELETAL ANALYSIS

John Rasmussen, Michael Damsgaard, Egidijus Surma, Søren T. Christensen, and Mark de Zee

The AnyBody Group, Institute of Mechanical Engineering, Aalborg University,
Pontoppidanstraede 101, DK-9220 Aalborg East, Denmark
e-mail: jr@ime.auc.dk , web page: http://anybody.auc.dk

Keywords: Multibody Dynamics, Musculoskeletal modeling, Inverse dynamics, Muscle systems, Numerical analysis

Abstract. This paper describes the considerations that went before the development of the AnyBody Modeling System, a system for computer modeling and analysis of the musculoskeletal system. The paper explains the considerations and choices behind the system's design and basic methodology, and it presents a number of examples to illustrate the consequences. It is concluded that much of the necessary functionality has been provided at the cost of a technical interface that might be difficult for non-experts to master.

---

1. Introduction

Software systems for virtual prototyping and Computer-Aided Engineering have revolutionized product development in the past two decades. The available analysis facilities today cover almost any conceivable technical property including strength, heat, flow, magnetism, and dynamics, and very few if any modern technical products are produced without prior computer simulation of some form.

On the other hand, tools for analysis of the mechanics of the human body are still in an initial stage of development. Advanced applications have been demonstrated (for instance F.C.Anderson and M.G.Pandy, 2001, H.v.d.Kooij et al, 2003, D.G.Thelen et al, 2003), but the technology has not yet reached a level where it is practically useful to a broad population of users. This paper describes the efforts of an interdisciplinary research team to design a generally applicable system for modeling and analysis of musculoskeletal systems.

In the late nineties, a group at the Institute of Mechanical Engineering at Aalborg University was studying the mechanics of bicycle frames with the goal of optimizing performance. It transpired over a period of time that the problem is ill posed unless the model includes the rider. In other words, the bicycle and the rider form one machine, and one part cannot be optimized in the absence of the other.

The Institute of Mechanical Engineering comprises experts within numerical mathematics, multibody dynamics, optimization, and software engineering and a group of scientists representing these fields started to review state-of-the-art within musculoskeletal simulation and found that it with its combination of competencies might be able to contribute to the field. The group developed three major versions of a software system for musculoskeletal modeling:

1. A hard-coded, procedure-oriented prototype

distinctly developed for simulation of pedaling. This prototype demonstrated the feasibility of the basic numerical methods.

2. An object-oriented prototype capable of handling different models by means of object definitions, albeit still hard-coded into the software.

3. A version capable of handling object-oriented models in a specially developed model description language. In this version, users can develop their own models, and models can be combined and easily exchanged between users.

The system was named AnyBody to reflect its ability to model "any body" the user desires, be it a human, a horse, or a dinosaur.

An interest group comprising more than 200 members, many of whom are biomechanical researchers, was founded. The purpose of the group was to monitor the development of the technology and give advice about its implementation and possible use. About 50 particularly interested members of the interest group signed up as alpha testers. These users tested the functionality of the system, influenced its final development, and provided a collection of models on which new users could build.

Since the interests of the Institute of Mechanical Engineering comprise the process of design, development of software for Computer-Aided Engineering to some extent falls within the its scientific paradigm. However, at a certain stage, a software tool becomes so mature that its further development is not a research task, and the AnyBody Modeling System was consequently transferred to a company, AnyBody Technology, that will undertake its dissemination. It has been decided that while the system is copyrighted and will become a commercial product, its models will remain in the public domain.

The AnyBody Modeling system is currently in a beta testing phase in which it is being distributed from the web page of AnyBody Technology, www.anybodytech.com.

This paper describes the considerations that went before the development of the system and explains some of the choices behind the system's design and basic methodology.

2. Methods

This section explains the goals of the system development and how they led to the choice of the methods upon which the system is built.

From the beginning of the development, the aim was to create a system with the following characteristics:

1. It should be a modeling system, i.e. a tool that allows users to construct models from scratch or use or modify existing models to suit different purposes.
2. The system should facilitate model exchange and 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 the example of Figure 1.

Figure 1. A full body model comprising more than 300 individual muscles

2.1 Structural considerations

The scientific field of design optimization has for two decades recognized the fundamental difference between system changes on the parameter level and on the topological level. Where the former is relatively uncomplicated, the latter has been perceived as a major challenge. This distinction also applies to software systems for mechanical simulation. Designing a system that allows the user to build the topology of the model is at least one order of magnitude more complicated than a system in which the user merely controls the parameters of a predefined model.

Systems for Computer-Aided Design (CAD) and word processors are good examples of the complexities involved in handling model topology. Enormous resources have been invested in their development, and the technology is still far from perfect from a user's point of view. The philosophy behind the system development has been to provide the user with ever more sophisticated menu-driven interfaces to assist in the manipulation of the model. This has had the undesirable side effect of separating the user from the actual data to such an extent that documents and models when reaching a certain level of complexity start to behave irrationally. Any author of a document with a high content of mathematical formulas in a modern word processor is painfully aware of this behavior.

As a reaction to the problems of handling complex data through graphical interfaces, a few systems have appeared with a converse philosophy, and they have obtained a considerable and enthusiastic community of supporters. Examples of such systems are the LaTeX type setting system and the Matlab system for numerical mathematics and data processing.

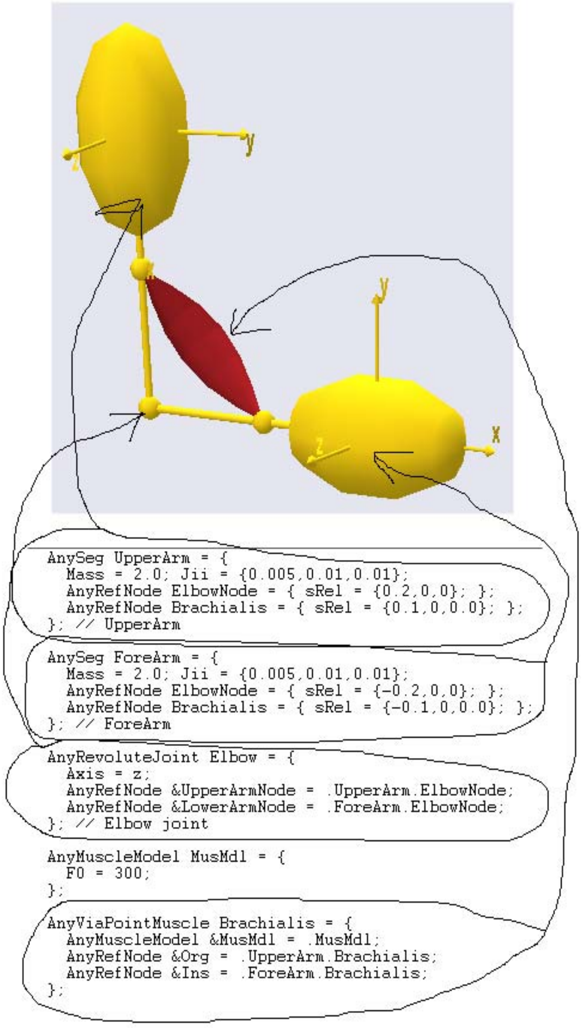
A brief review of the task of building a realistic body model such as the example of Figure 1 reveals that the amount and complexity of data is likely to make the development of a menu-driven user interface difficult. Furthermore, the requirement for model exchange and investigation of models would require that the data of the model is directly readable. It was consequently decided to develop a body modeling language that would express models in clear text. The result was the AnyScript language.

2.2 The AnyScript 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:

- The body model containing the definition of the mechanical system.
- The study section containing lists of analyses and other operations to perform on the model.

The language has a number of predefined classes that the user can create objects from. The declarative nature of the language means that an element of the model is defined simply by declaring an instance of a class. It is not necessary to write operational code like 'do' loops and 'if-then-else' clauses. For instance, a model with two joined segments articulated by a muscle would be defined as in figure 2.



```
AnySeg UpperArm = {
  Mass = 2.0; Jii = {0.005,0.01,0.01};
  AnyRefNode ElbowNode = { sRel = {0.2,0,0}; };
  AnyRefNode Brachialis = { sRel = {0.1,0,0.0}; };
}; // UpperArm

AnySeg ForeArm = {
  Mass = 2.0; Jii = {0.005,0.01,0.01};
  AnyRefNode ElbowNode = { sRel = {-0.2,0,0}; };
  AnyRefNode Brachialis = { sRel = {-0.1,0,0.0}; };
}; // ForeArm

AnyRevoluteJoint Elbow = {
  Axis = z;
  AnyRefNode &UpperArmNode = .UpperArm.ElbowNode;
  AnyRefNode &LowerArmNode = .ForeArm.ElbowNode;
}; // Elbow joint

AnyMuscleModel MusMdl = {
  F0 = 300;
};

AnyViaPointMuscle Brachialis = {
  AnyMuscleModel &MusMdl = .MusMdl;
  AnyRefNode &Org = .UpperArm.Brachialis;
  AnyRefNode &Ins = .ForeArm.Brachialis;
};
```

Figure 2. AnyScript definition of two segments and a joint articulated by a muscle.

It is an important feature of the system that once the elements of a model are declared, the underlying mathematics is also defined, and the user does not explicitly work out the equations of motion and code them into the system. In the example of figure 2, the revolute elbow joint will automatically lead to elimination of all other mutual degrees of freedom between the two segments than a rotation about the z axis.

The study section of the model allows for specification of various operations to perform on the model such as kinematic analysis, kinetic analysis, muscle calibration, and systematic parameter variations. Studies can refer to the entire model or to subsections of the model.

2.3 Structuring models for multiple purposes

The structure of an AnyScript model is fairly high level and therefore legible and rather compact. This can be attributed to the fact that much of the model's functionality is defined implicitly by the mutual references between its elements. A joint, for instance, needs only refer to the two points it is joining.

The compactness and legibility allows for easy exchange of models between users, so it would appear that one of the goals of developing the system has been reached. However, the system provides much freedom for the user to structure 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 merge them, and different model structures might disable this option. To facilitate model merging, the AnyBody Research Group has developed a model structure and a few facilities have been added to the system.

In addition to using the ISB recommendation for coordinate systems, the model structure recommends to split the model into two distinct parts (figure 3):

(i) The body model. This part contains segments, joints, muscles, and other anatomical data, but it does not contain any movement specifications, load specifications, or specifications of external equipment such as bicycles, tools, or workplaces.

(ii) The application model. Application-specific data such as loads and movements 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 figure 4.

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 objects. This means that the hand definition can contain the necessary additions to the arm model to make the two parts compatible.

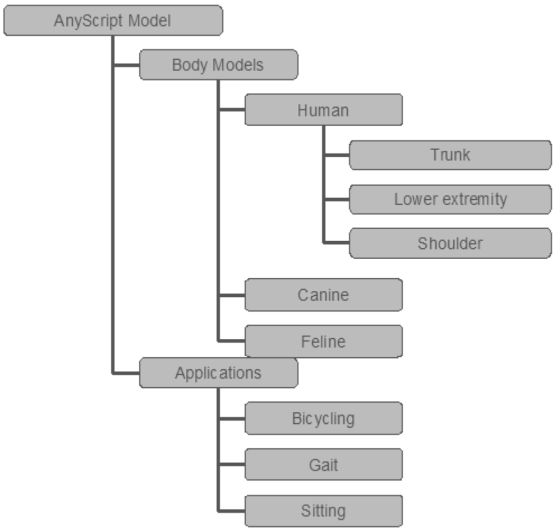


Figure 3. Recommended structure to facilitate model exchange. The labels in the boxes are only examples.

2.4 Parametric models

It is imperative for the applicability of the system that models can be parametric and that relationships between elements can be built into the model. The developer of a given application is likely to want to investigate the consequences of different design choices such as the saddle height or pedal arm length of a bicycle, or the backrest inclination of a chair. Similarly, body models should be scaleable to represent different body anthropometrics.

The use of a programming language for model definition fortunately makes the use of mathematical expressions a natural part of the data structure. This means that a mathematical expression can replace a number anywhere in the model and the length of a femur segment, for instance, can be represented as a fixed fraction of an overall body height parameter. However, while the basic facility to enable body

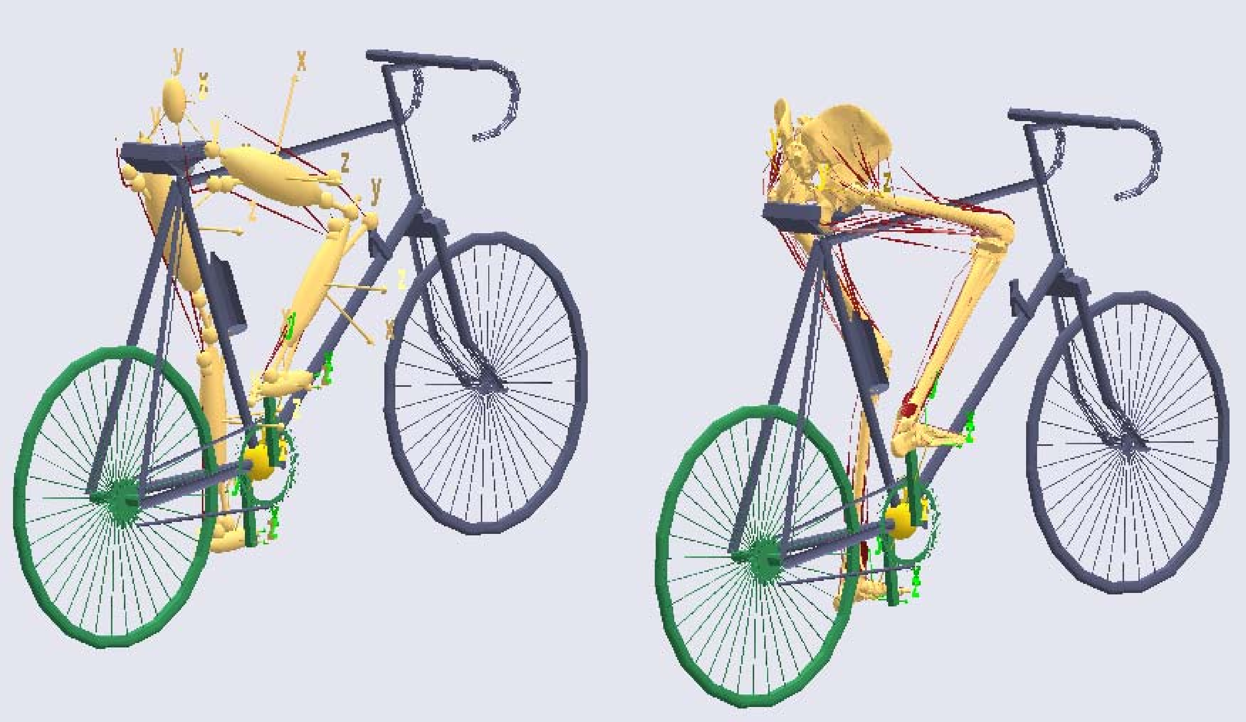


Figure 4. A typical example of a structured model: The two bicycles are identical applications within the same model and are combined with a simple, 2-D lower extremity and a more complicated 3-D lower extremity respectively.

scaling is in place, the correct “formula” to do so remains an open scientific question.

2.5 Inverse dynamics

Investigations of possible methods for simulation of dynamics made it clear that computation times would be prohibitive for models with a realistic number of muscles unless the approach was based on inverse dynamics. For identification of individual muscle forces as opposed to joint moments, this requires use of an efficient and physiologically credible approach for distributing forces between redundant muscles. It also became clear that the quantitative differences between plausible criteria were not very large (J.Rasmussen et al, 2000). The AnyBody Modeling System currently uses a min/max criterion, but the implementation of other criteria is planned.

2.6 Kinematic versatility

It is tempting to interpret the degrees of freedom of a human body model physiologically, i.e., as joint angles. However, binding the system to express the degrees of freedom in such physiological terms would seriously deplete the system’s applicability.

In gait analysis, for instance, movement is typically recorded by video tracking of optical markers and it would be desirable to be able to use the marker positions directly to drive the model. Another example is when the posture or movement of the human body is defined by its interaction with an artifact such as a bicycle, a hand tool, a chair, or a workplace. In the case of the bicycle, for instance, the movement of the feet is defined by the pedal cycle rather than the anatomical joint angles, and the natural postural definition would be in terms of the crank angle rather than joint angles.

To enable definition of kinematics in terms of non-anatomical parameters, the AnyBody Modeling System has been equipped with an abstract concept called “kinematic measure”. A kinematic measure is any dimension that can be measured on the body model.

The technology behind this facility is a general formulation of the kinematical problem in terms of Cartesian coordinates (P.E.Nikravesh, 1988). The coordinate vectors for all n segments of the system, including both human segments and application parts are assembled. This provides the system coordinate vectors $\mathbf{q}=[\mathbf{q}_1^T \;..\; \mathbf{q}_n^T]^T$ and velocities $\mathbf{v}=[\mathbf{v}_1^T \;..\; \mathbf{v}_n^T]^T$. Furthermore, we assemble all kinematic constraint equations associated with joints, drivers, and the constraints on the Euler parameter stating that all $\mathbf{p}_i$ are unit vectors. This provides $7n$ independent nonlinear equations.

$$\Phi(\mathbf{q},t)=\mathbf{0} \qquad (1)$$

The position analysis is carried out by solving the equations with a suitable numerical method.

This formulation derives its generality from equation (1) into which any nonlinear equation specifying the position of the mechanism, i.e. kinematic measures, can be inserted.

As an example of the generality of kinematic measures we can consider the problem of modeling an unsupported slow squatting movement (Figure 5). When performing such a task, care must be taken to maintain the position of the collective center of mass vertically above the contact line of the two foot on the ground, lest the model would fall over due to lack of forward/backward support. Since a squat involves individual movements of arms, trunk, thighs, shanks, and feet, it would be a very challenging task to specify a set of anatomical joint movements that would constrain the collective center of mass. In the concept of kinematic measures, the collective center of mass is simply a point in the model, and it is consequently possible to drive it by inserting the specifications of its position into the system of equations (1). The consequence is that a driver on, for instance, the arm position can be neglected from the model, and the arm of the model will automatically attain the position necessary to balance the model in each stage of the movement, exactly as a test person might reach out in front of him during the squat to avoid falling backwards.

3. Conclusions

The AnyBody Modeling System meets most of the requirements to a general musculoskeletal analysis system; users can build models of realistic complexity, models can be exchanged between users and are open to investigation because they are written in clear text, and the kinematics is sufficiently general to handle machine interfaces and point trajectory movement specifications.

These facilities are largely obtained by using a model description language, and the system therefore has a rather technical character as illustrated by figure 2. This means that the system is mainly directed towards experts in the field of human movement.

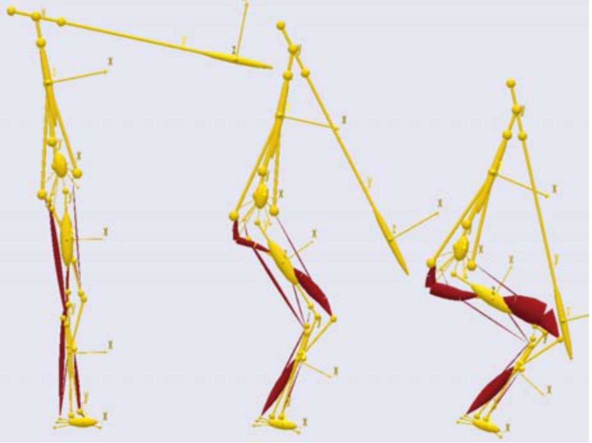


Figure 5. Three different squat positions balanced by arm extensions. The bulging of each muscle is proportional to the muscle force.

4. Acknowledgements

This work was supported by the Danish Research Agency and the Ford Motor Company.

References

F.C.Anderson, M.G.Pandy, *Dynamic Optimization of Human Walking*, Journal of Biomechanical Engineering, 123, 381-390, 2001.

H.v.d.Kooij, R.Jacobs, B.Koopman, F.v.d.Helm, *An alternative approach to synthesizing bipedal walking*, Biol.Cybern. 88, 46–59, 2003.

P.E.Nikravesh, *Computer-Aided Analysis of Mechanical Systems*. Prentice-Hall, Inc., 1988.

J.Rasmussen, M.Damsgaard, M.Voigt. *Muscle recruitment by the min/max criterion – a comparative study*, Journal of Biomechanics, 2001, 34(3), 409-415.

D.G.Thelen, F.C.Anderson, S.L. Delp, *Generating dynamic simulations of movement using computed muscle control*, Journal of Biomechanics 36, 321–328, 2003.