AnyBody - a software system for ergonomic optimization

John Rasmussen[1*], Michael Damsgaard[1], Egidijus Surma[1], Søren T. Christensen[1], Mark de Zee[1], and Vit Vondrak[2]

[1]Institute of Mechanical Engineering, Aalborg University,
Pontoppidanstraede 101,
DK-9220 Aalborg East
Denmark.
[*]Corresponding author: jr@ime.auc.dk

[2] Dept. of Applied Mathematics
VSB Technical University in Ostrava,
17. listopadu 15
708 33 Ostrava-Poruba
The Czech Republic

1. Abstract

This paper describes the efforts of an interdisciplinary research team to design a generally applicable system for ergonomic optimization with respect to the physiological properties of the human body and the body's interaction with the product. The articulation of biological principles in the form of optimization criteria, their implementation into efficient methods for musculoskeletal simulation, and the use of the methods for ergonomic optimization are described. It is concluded that musculoskeletal analysis based on optimization principles holds a large potential in basic research as well as in product development.

2. Keywords: Ergonomics, human interfaces, musculoskeletal analysis.

3. Introduction

Software systems for virtual prototyping and Computer-Aided Engineering (CAE) have revolutionized product development. The available analysis facilities today cover almost any conceivable technical property. From this point-of-view, virtual prototyping is a mature technology. However, many products have an interface to the human body as an important property, and until now there has been no quantitative method available to analyze the ergonomic quality of products, let alone optimize them.

Figure 1. A full-body model in AnyBody comprising several hundred muscles.

In the late nineties, a group at the Institute of Mechanical Engineering at Aalborg University was studying the design of bicycle frames for optimum 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.

In this respect, the bicycle is representative of a large class of products of which the interface to the human body is a primary feature. Furniture, consumer electronics, disability aids, prosthetic and orthotic devices, footwear, sports equipment, cars and motorcycles, and hand tools are just a few of the categories. In view of this fact, it is striking that there has not been a CAE technology available to reliably simulate the interaction between the human body and its environment.

Inspired by the bicycle design problem, a group comprising experts within multibody dynamics, biomechanics, physiology, design optimization, mathematics, and software engineering was established. The group developed three major versions of a software system for modeling the human musculoskeletal system:

1. A hard-coded, procedure-oriented prototype distinctly developed for optimization of bicycles. 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. This version was used for optimization of a handsaw and a tricycle for paraplegics.
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. Figure 1 shows a model of the full human body comprising more than 300 muscles.

The system was named AnyBody to reflect its ability to model "any body" the user desires.

One of the principal problems of analysis of the musculoskeletal system is that it is mechanically redundant. This means that the analysis inherently depends on optimization, and it is crucial that this is implemented efficiently and with regard for the special mathematical properties of the muscle recruitment problem. AnyBody uses a variety of LP and QP techniques for this purpose.

4. Methods

In this work we perceive the human body and other mechanical parts connected to it as a multibody system. The muscles are the actuators of the system and the individual segments are modeled as rigid bodies. A brief review of the attempts to simulate the human body as a mechanical system is best initiated with the observation that the methods traditionally fall into one of two categories, inverse dynamics and forward dynamics, which, as the names indicate, are opposite approaches.

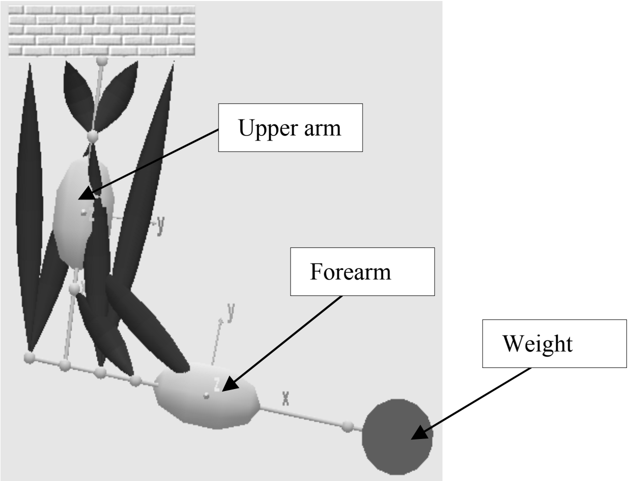

Figure 2. A simplified arm model in a multibody dynamics representation. Notice that five muscles are available to flex the two joints of shoulder and elbow, thus creating a triple statically indeterminate situation known as muscular redundancy.

4.1. Inverse dynamics

In inverse dynamics, the motion and the external loads on the body are assumed known, and the purpose of the computation is to determine the internal forces. When the "internal forces" are mere joint moments and joint reaction forces, this, in most cases, is a straightforward procedure involving the solution of a system of linear equilibrium equations. However, for the purpose of computing individual muscle forces, inverse dynamics is haunted by the so-called redundancy problem: not enough equilibrium equations are available to determine all the muscle forces. Infinitely many different sets of muscle forces, of which the central nervous system (CNS) instantly chooses one, can therefore produce the identified joint moments. This is due to the fact that we have more muscles than strictly necessary to drive most motions, and the situation is in effect statically indeterminate as illustrated in Figure 2. Constructing an algorithm to determine the activation of each muscle therefore entails guessing the motives behind the CNS's function. We are able to repeat movements with considerable precision so many researchers believe that the control of muscle forces must be based on some rational criterion. Indeed, it has been stated [1]: "It is not known why in skilled multi-joint tasks such as cycling, the pattern of muscle activity is rather stereotypical at similar cycling conditions, whereas an infinite number of activity patterns can theoretically be used by the CNS to perform the same task -- to produce the same combination of joint moments".

Assuming that muscles are recruited according to an optimality criterion, we are faced with the task of selecting the right one. Let us briefly state the mathematical form of the inverse dynamics problem:

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

$$\text{Subject to} \quad \mathbf{Cf} = \mathbf{d} \tag{2}$$

$$f_i^{(M)} \geq 0, \qquad i \in \{1,\ldots,n^{(M)}\} \tag{3}$$

where **f** is the vector of $n^{(M)}$ unknown muscle forces, $\mathbf{f}^{(M)}$, and joint reactions, $\mathbf{f}^{(R)}$. **C** is the coefficient matrix, and **d** is the right hand side comprised by external forces and passive elasticity in the tissues of the body. In the AnyBody Modeling System, a min/max criterion is used for the objective function *G*:

$$G(\mathbf{f}^{(M)}) = \max\left(\frac{f_i^{(M)}}{N_i}\right) \tag{4}$$

where N_i is the momentary strength of muscle *i*. The problem can be converted to a linear form via the so-called bound formulation. For full details please refer to [2] and [3].

4.2. Forward dynamics

Several authors, [4,5] used forward dynamics methods for simulation of the human body. Forward dynamics methods are well developed in the field of multibody simulations of mechanical systems, and several sophisticated software packages for this purpose are available commercially. In forward dynamics, internal and external forces are applied to the model leading to acceleration of its segments. By integration over time, the total movement of the model resulting from a given force pattern can be identified, and very complicated phenomena such as electro-chemical latency in the CNS and the muscles, impact, friction and the influence of control systems can be simulated.

When applied to human body simulation, forward dynamics methods can be argued to emulate the function of the CNS directly, in the sense that they use the recruitment of individual muscles as independent variables, thus producing some motion and forces against the environment. The task is then to identify the muscle activations that produce the desired result; much like an infant gradually learns to perform complex manual tasks by training the control of his muscles. Mathematically, this is a problem of optimum control, and such problems are very computationally demanding. They are frequently non-convex and require global optimization strategies such as simulated annealing, genetic algorithms, or random search. This limits the amount of independent variables in the problem and thereby the size of the model and the complexity of the movement.

A fundamental difference between inverse and forward dynamics is that, in inverse dynamics, the model is bound to perform the specified motion, where a forward dynamics model must solve an optimum control problem just to reproduce a known behavior. This means that simulation of human movement by forward dynamics can still be considered as being on the experimental level, and each new case borders a research project. The technology is not mature enough to be used in a clinical setting.

5. The AnyBody Modeling System

The AnyBody Modeling System is designed for constructing complex models of the human body and for determining the environment's influence on the body, and it must consequently exhibit a computational efficiency that can only be obtained by inverse dynamics. This means that the computational procedure is as described in Figure 3.

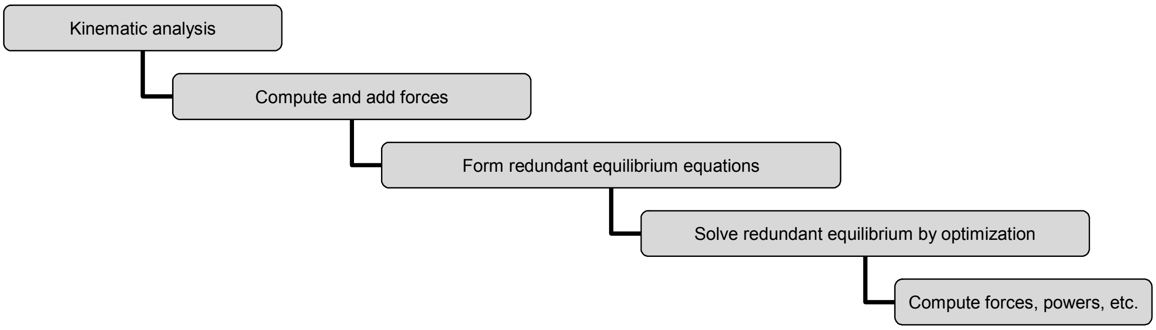

Figure 3. The computational procedure for an analysis of one time step in the AnyBody Modeling System.

As Figure 3 illustrates, kinematics is the initial stage of a musculoskeletal analysis and consequently plays a very important role. While kinematics is a well-developed field and advanced computer systems for mechanism analysis are commercially available, the modeling of the human body requires additional and quite special facilities.

The analysis in the AnyBody Modeling System proceeds through a sequence of time steps defined by the user. A static problem has only one time step, and a dynamic problem has the time span of the analysis divided into steps of equal length. One of the advantages of inverse dynamic analysis is that time steps can be considered independent and without much bearing on the numerical convergence of the analysis. The purpose of the kinematic analysis is to identify each segment's[1] position, velocity and acceleration in each time step. The AnyBody Modeling System uses the Cartesian formulation of the kinematic problem, in which each segment has six independent degrees of freedom, and constraints corresponding to joints are imposed on the full size system of equations.

All segments of the mechanical system are modeled as rigid bodies, neglecting effects such as the wobbly masses of soft tissues. We more or less adopt the formulation of [6]. The position of the i'th segment 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 velocities of the segments is defined as $\mathbf{v}_i=[\dot{\mathbf{r}}_i^T\ \omega_i'^T]^T$, and the accelerations as time-derivatives, i.e., $\dot{\mathbf{v}}$. The vector ω_i' is the angular velocity of the segment, where the apostrophe indicates that it refers to the segment-fixed reference frame.

The kinematic analysis is carried out in terms of all the Cartesian coordinates, i.e., we assemble the coordinate vectors for all n segments of the system, including both human segments and machine parts. This provides the system coordinate vectors $\mathbf{q}=[\mathbf{q}_1^T \,..\, \mathbf{q}_n^T]^T$ and $\mathbf{v}=[\mathbf{v}_1^T \,..\, \mathbf{v}_n^T]^T$. 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 non-linear equations.

$$\Phi(\mathbf{q},t)=\mathbf{0} \tag{5}$$

The number of equations matches the number of unknown coordinates in $\mathbf{q}$ if the problem is kinematically determinate. The position analysis is carried out by solving the equations with a suitable numerical method, for instance Newton-Raphson iteration using the constraint Jacobian, $\Phi_{\mathbf{q}}$. Velocity and acceleration analysis is carried out by solving the linear equations arising from time-derivation of Eq. (5) and transformation into the basis of $\mathbf{v}$.

$$\Phi_{\mathbf{q}*}\mathbf{v}=-\Phi_t \tag{6}$$

$$\Phi_{\mathbf{q}*}\dot{\mathbf{v}}=\gamma(\mathbf{q},\mathbf{v},t) \tag{7}$$

In Eqs. (6) and (7), we use * to indicate that it is the transformed Jacobian, and not $\Phi_{\mathbf{q}}$, we use. For simplicity, $\Phi_{\mathbf{q}*}$ can be derived directly from the constraints without actually using a transformation between $\dot{\mathbf{q}}$ and $\mathbf{v}$. This is carried out by differentiation with respect to time in terms of $\mathbf{v}$. Moreover, the Euler parameter constraints can be left out of Eqs. (6) and (7), as they are redundant in terms of angular velocity. The equation solver applied to the solution of the equations handles other redundant constraints.

Having solved Eqs. (5)-(7), we know the motion completely if the system is properly specified, and we can generate the input to the muscle recruitment problem, Eqs. (1)–(3), and solve to find the muscle and joint forces of the system.

As mentioned, the use of Cartesian coordinates and the Newton-Euler equations for each rigid segment has been adopted here in the view of simplicity of implementation. To enable general application of the method to systems including both the human body and mechanical artifacts,

---

[1] The term "segment" is used for a rigid body in musculoskeletal analysis because the term "body" can be misinterpreted when the context is physiology.

we need to have a general model formulation, which is easily obtained by this approach. A drawback may be low efficiency for large mechanical systems, or the need for efficient sparse system handling, compared to models using fewer (relative) coordinates. Another issue is the large number of parameters to be estimated, hereunder geometric, inertia, and muscle data. Such data can be difficult to estimate and the chosen approach does not provide the best basis for such parameter estimation. However, it must be emphasized that the solution to the muscle recruitment problem does not depend on the actual formulation of the equations of motion. In other words, Eq. (2) can be obtained by any other, more suitable approach depending on the actual application and the data at hand.

The generality of the Cartesian method facilitates the implementation of useful features in the kinematic system. This is a major advantage in musculoskeletal modeling, where the complications of the human body kinematics and interfaces to the experimental techniques used in the field call for special considerations in the software architecture. The following two sections describe two such useful features.

5.1. Kinematic measures

It is tempting to interpret the degrees of freedom of a human body model physiologically, for instance as the flexion of a knee or the twist of a forearm. However, binding the system to express the degrees of freedom in such physiological terms would seriously deplete the system's applicability for studies of humans in free movement and in connection with various types of equipment, and the general statement of the kinematic problem, Eq. (5), in fact allows for more flexible approaches.

In gait analysis, for instance, which is a major field of clinical diagnostics and research, movement is typically recorded by video tracking of optical markers attached to the body. The kinematics of the human body is sufficiently complex to make the conversion of marker positions to physiological joint angles a challenging task, 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.

To enable definition of kinematics in terms of non-anatomical parameters, the AnyBody Modeling System has been equipped with an abstract concept named "kinematic measures". A kinematic measure is just about any dimension that can be measured on the body model. Typical examples could be the distance between two points, the coordinates of a point (such as a video tracking marker) in space, the length of a muscle, or a joint angle. The concept of kinematic measures thereby encapsulates also the anatomical postural dimensions such as joint angles.

As an example of the generality of kinematic measures we can consider the problem of modeling a human body in an unsupported slow squatting movement (Figure 4). 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 feet 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 gravity. 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 position analysis, Eq. (5). The consequence is that a driver on, for instance, the arm position can be neglected from the model, and the arms of the model will automatically attain the position necessary to balance the model in each stage of the movement, exactly as a test person would reach out in front of him during the squat to avoid falling backwards.

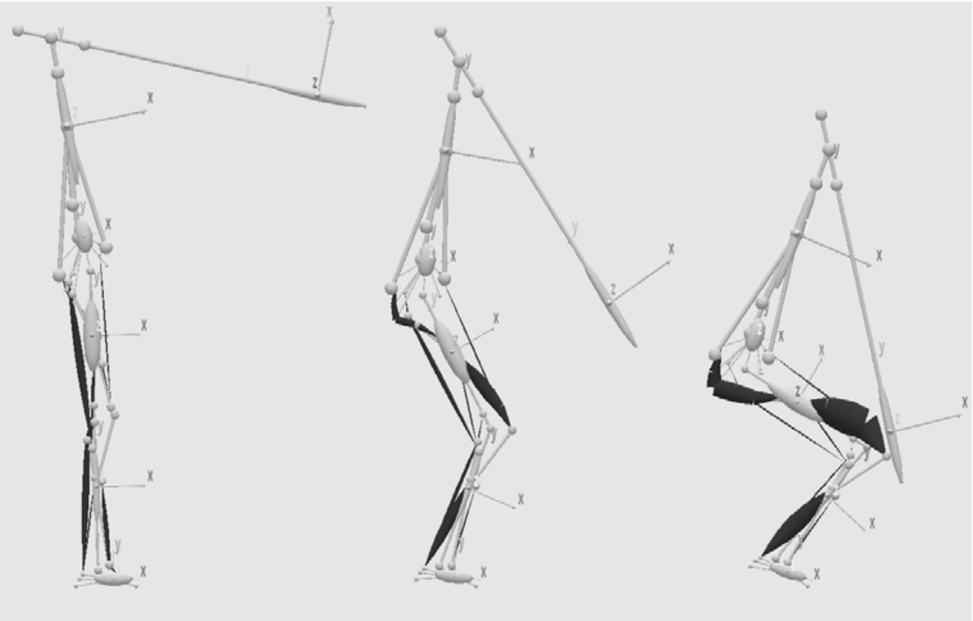

Figure 4. The control of arm positions in a squat model by means of the collective center of mass. The arm moves automatically to maintain the collective center of mass above the ball of the foot. The bulging of the muscles reflects the simulated muscle forces.

5.2. Muscle wrapping

One of the most complicated mechanical aspects of the human body is the fact that muscles wrap over bones and other tissues on their way from origin to insertion. When doing so, they exert forces to multiple parts of the segment surfaces, and as the model moves, the muscles slide over bones. Existing contact surfaces change or disappear, and new ones may arise.

The correct handling of this behavior calls for algorithms of contact mechanics. For a review of state-of-the-art in this field, please refer to [7]. To enable the direct import of geometries from CAD systems into body models, a method has been developed to enable muscle wrapping over triangulated surfaces described as STL files. An STL file is merely a collection of triangles, and almost any CAD system is capable of saving a surface representation on STL format. Thus, the CAD surfaces can be imported directly into the AnyBody model as shown in Figure 5. If the contact between the muscle and the bone is considered frictionless, then the identification of the muscle's path over the bone essentially corresponds to an optimization problem: Minimization of the distance between origin and insertion with the bone surface as a territorial constraint on the path.

We model this problem as a 3-D contact problem of an elastic string and a rigid obstacle representing the bone surface. The goal of this problem is to minimize the potential energy of the string. This way we obtain the shortest path of the string around the obstacle. The most important part in the formulation of this problem is to find correct conditions for non-penetration of the surface. We have chosen linearized contact conditions prescribing non-penetration of the surface by a point on the string in the direction of the outer normal vector to the surface. This model fits very well with triangulated surfaces as described in the STL format as well as with a string discretized by the finite element method. In the discrete case the method of prescribing the contact conditions of non-penetration is based on searching for contact pairs: point of string – closest triangle of surface in normal direction - and collecting these conditions for all points of string discretization. Then the problem could be mathematically written into the form:

$$\min \tfrac{1}{2}\mathbf{u}^T\mathbf{K}\mathbf{u} - \mathbf{u}^T\mathbf{f} \tag{8}$$

$$\text{Subject to } \mathbf{N}\mathbf{u} \le \mathbf{d} \tag{9}$$

where $\mathbf{u}$ is vector of string point positions, $\mathbf{K}$ is stiffness matrix of the string and the $\mathbf{f}$ vector contains external loads and bounding point positions. The inequality constraints $\mathbf{N}\mathbf{u} \le \mathbf{d}$ represent contact conditions of non-penetration, where $\mathbf{N}$ is a matrix collecting normal vectors to the triangles in eventual contact and $\mathbf{d}$ is vector of feasible normal displacements of paired points from the string. This formulation is very suitable for efficient solving in its dual form by fast algorithms as described in more detail in [8].

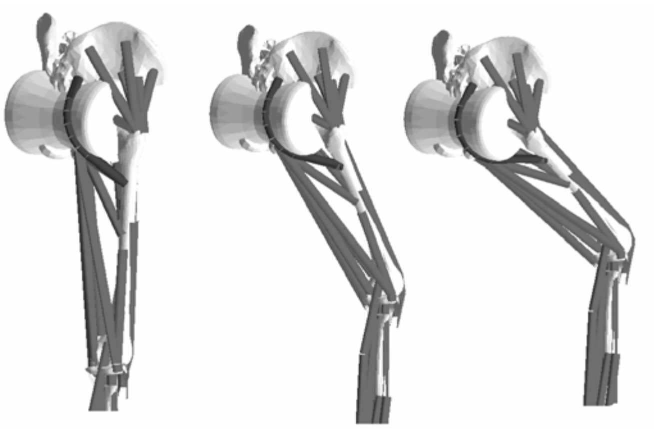

Figure 5. The gluteus maximus muscle wrapping over an artificial STL surface added to the pelvis.

5.3 Sensitivity analysis and design optimization

When basing the analysis on a numerical optimization algorithm, analytical or semi-analytical sensitivity analysis becomes difficult or impossible, and the overall finite difference method appears to be the only viable solution. In this situation, the cost of a sensitivity analysis is similar to the cost of an analysis, so for the problem of ergonomic design optimization the system employs a feasible directions method that only relies on sensitivity analysis only for finding the search direction and solves the line search problem by 0’th order golden section search.

6. Example: Optimization of a hand saw

Ergonomic optimization falls into several categories. Some problems can be understood mostly from the point-of-view of joint moments. In the example of Figure 6 the force between the saw blade and the work piece creates a large moment about the wrist, and a moment about the shoulder joint that forces the shoulder muscles to pull the arm backwards. This creates an inefficient use of muscle forces where the shoulder is working antagonistically to the elbow. In the optimized configuration, this moment arm is much reduced, and the moment about the wrist as well as the shoulder is reduced. For further details please refer to [9]. Other types of problems require detailed modeling of the musculoskeletal system. Examples are optimization of bicycles and optimization of seated postures. In the latter case, the spine plays an important role, and a detailed spine model comprising all vertebrae and individual muscle elements is currently being developed (figure 7).

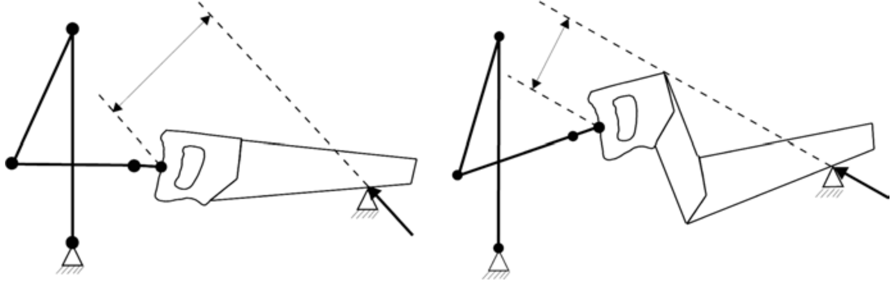

Figure 6. Optimization of a hand saw [9]. Left: Initial configuration. Right: optimized configuration.

6. Conclusions

From the point-of-view of analysis, the efficient, optimization-based methods upon which the AnyBody Modeling System is built allow for handling of models that are significantly larger and more complex than any other technology has accomplished so far. As an example, the full body model of Figure 1 comprises several hundred muscles.

From a design optimization point-of-view, ergonomic optimization has the potential to represent a major step forward in product design as well as in basic research to further our understanding of the function of the human body. This is novel in the sense that there has not previously been a CAE technology available to provide this facility.

We conclude that mathematical programming and the use of optimization principles play major roles in algorithms for analysis as well as for design optimization of models involving the human body and most likely other biological systems as well.

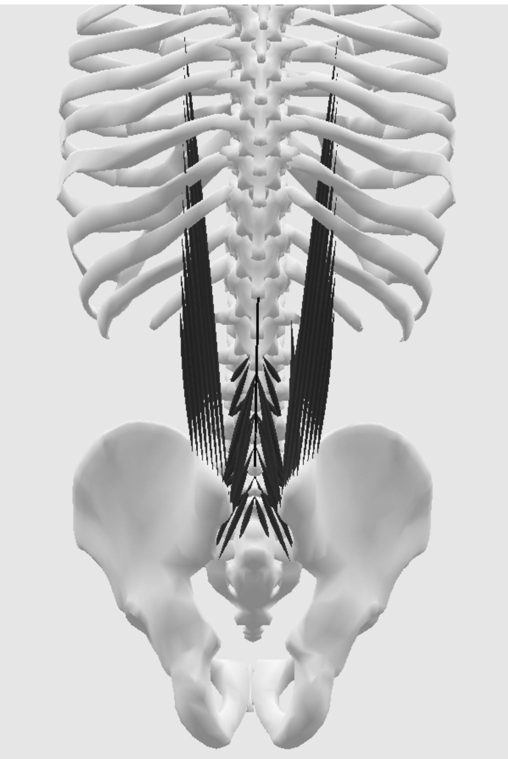

Figure 7. A detailed model of the spine currently under development using the AnyScript Model definition language.

7. Acknowledgements

This work was supported by the Danish Research Agency, the Ford Motor Company, and by the Grant Agency of the Czech Republic, #101/02/0072.

8. References

1. Prilutsky, B.I., Zatsiorsky, V.M. Optimization-Based Models of Muscle Coordination. Exercise and Sport Sciences Reviews, 2002, 30: 32-38.
2. Rasmussen, J., Damsgaard, M., Voigt, M. Muscle recruitment by the min/max criterion – a comparative study. Journal of Biomechanics, 2001, 34(3), 409-415.
3. Damsgaard, M., Rasmussen, J., Christensen, S.T. Inverse dynamics of musculo-skeletal systems using an efficient min/max muscle recruitment model. Proceedings of DETC’01: ASME 2001 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2001, Pittsburgh, Pennsylvania, USA.
4. Anderson F. C., Pandy, M. G. A dynamic optimization solution for vertical jumping in three dimensions. Computer Methods in Biomechanics and Biomedical Engineering, 1999, 2:201-231.
5. Neptune, R. R. Optimization algorithm performance in determining optimal controls in human movement analysis. Journal of Biomechanical Engineering, 1999, 121:249-252.
6. Nikravesh, P. E. Computer-Aided Analysis of Mechanical Systems. Englewood Cliff: Prentice-Hall, Inc., 1988
7. Feng. G., Damsgaard, M., Rasmussen, J., Christensen, S.T. Computational method for muscle-path representation in musculoskeletal models. Biological Cybernetics, 2002, 87(3):199-210.
8. Dostál, Z., Vondrák, V., Rasmussen, J. Efficient algorithms for contact shape optimization problems. Int. series of numerical mathematics, 2001, 138:98-106.
9. Rasmussen, J., Damsgaard, M., Christensen, S.T., Surma, E. Design optimization with respect to ergonomic properties. Structural and Multidisciplinary Optimization, 2002, 24:89-97.