SIMULATOR FOR MULTI-SCALE MUSCULOSKELETAL MODELS WITH REFLEX CIRCUITS

Csaba Fazekas, Dávid Csercsik, Gábor Szederkényi, Katalin M. Hangos

Process Control Research Group, Systems and Control Laboratory, Computer and Automation Research Institute,
P.O.Box 63, H-1518, Budapest, Hungary

fazekas@scl.sztaki.hu(Csaba Fazekas)

Abstract

A simulator for multi-scale musculoskeletal models is presented that realizes a multi-scale model of the musculoskeletal system. Each level of the multi-scale model corresponds to a characteristic anatomical parts of the real system. The simulator integrates coherently the different types of partial models with different time and size-scales and can solve the problem of stiffness. The simulator is implemented in MATLAB such that each level is realized as an object. The simulator satisfies the modularity condition, i.e. the partial models can be changed in a user friendly way. The model and the simulator has been extended by models of receptors, their feedbacks and reflex loops. The extended model can be applied to investigate, e.g. the gamma-loop. Some other applications are also presented.

Keywords: Musculoskeletal model, receptor, multi-scale model, feedback, gamma-loop

Presenting Author's Biography

Csaba Fazekas. He is a young researcher in the Process Control Research Group of the Computer and Automation Research Institute of the Hungarian Academy of Sciences. He received his M.Sc. (Eng) in Information Technology from the University of Veszprém in 2002. His research interest is modeling and model calibration of musculoskeletal and energetic systems.

ISBN 978-3-901608-32-2

1 Introduction

Musculoskeletal systems generate mechanical response from electrical excitations to form the motion which also depends on the inner structure of the system and its processes. Therefore, a musculoskeletal model that is applied to deeper diagnosis should contain several partial models and should describe different processes, such as electrical, chemical, mechanical processes. However, different types of models with different time and space constants can be associated with different processes and structures, see e.g. [1]. The necessity of simulation and the complex structure of musculoskeletal system call for the application of multi-scale modeling methods [2].

Multi-scale modeling is able to integrate models from micro-scales to macro-scales in a seamless fashion (see e.g. [3, 4]). This modeling technique offers a systematic way of constructing, analyzing and solving dynamic models of large-scale complex systems [5]. It also provides us with natural mechanism-driven decomposition of the underlying model with any related information and an integration framework to organize the information exchange between the partial models [6]. In addition, developing the abstractions to integrate between scales will lead to a much deeper understanding of the universal or generic features of biological phenomena.

The multi-scale model requires special simulators because it could contain different types of partial models with highly different time- and size-scales. The different time-scales can cause stiffness problems that have to be resolved. The different type of partial models may require different kind of solvers (for ordinary and/or partial differential equations). The simulator can integrate the partial models in a coherent way and and can apply a suitable solver to resolve the problem of stiffness caused by the different partial models.

One of the most popular application area of musculoskeletal dynamic simulators is the motion control. The human control research needs the understanding of various feedback mechanisms present in the human neuromuscular systems. These feedback mechanisms have to be integrated into the musculoskeletal model properly. For describing the effect of these feedback mechanisms, different types of receptors and their connection to the musculoskeletal system have to be modeled. Several publications studied the role of sensory feedback in motor control [7, 8, 9], and deal with the properties of the receptors, which provide the sensory information for this feedback [10, 11].

Our aim was to develop a simulator for multi-scale musculoskeletal model with the following properties.

- It contains anatomically and physiologically meaningful elements,
- it models the muscle structure (different fibers, tendon, aponeurosis),
- it can handle more agonist and antagonist muscles and multi-joint muscles,
- it can handle sub-models with different time and length-scales,
- it can handle feedback mechanisms, for example reflex loops,
- and the applied sub-models can be changed and/or modified in a user-friendly way.

2 Methods

2.1 The ingredients of a multi-scale musculoskeletal model

To develop a multi-scale musculoskeletal system, first the scales have to be identified. After the investigation of musculoskeletal system from anatomical, physiological and modeling point of view the following scales (levels) are defined (see in Fig. 1 and see details in [2]):

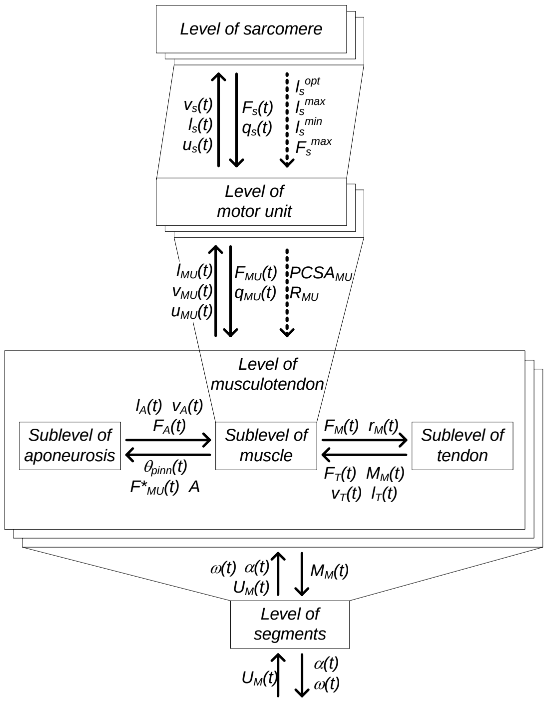

Fig. 1 Levels and their connections in the multi-scale musculoskeletal model

- *Level of sarcomere*: models the active force generators taking into account its mechanical (length and contraction velocity), electrical (activation) and chemical (concentrations) states and models the passive force of filaments. This level contains the model of active force generation, activation dynamics and bio-chemical processes, therefore it consists of dynamic and algebraic equations. Their typical characteristic sizes are several nm or μm and their typical time constants are about $1 - 10$ ms, however, there are some processes with time constant much greater.

ISBN 978-3-901608-32-2

- *Level of motor unit*: generates the force of the motor unit from the forces of sarcomere, computes the activation state of the motor unit from the activation state of the sarcomere and computes the excitation signals of sarcomere from the excitation signal of the motor unit. This level is responsible for modeling passive forces of fibers and motor units. This level is mainly defined to map the internal structure of a muscle. The force integration and the passive force can be described by algebraic equations, while the excitation signal distribution can be described by differential or algebraic equations.
- *Level of musculotendon*: computes the forces and torques of muscles from the forces of motor units, computes the activation of muscles from the activation of motor units, generates the excitation signals of motor units from the excitation signal of muscles and realizes the model of geometric joint transformation. For the integration of forces of motor units belonging to the given muscle, we take into account the effect of tendon, aponeurosis and pinnation. This level contains three sub-levels:
  (a) Sub-level of muscle: integrates the forces of motor units. This integration does not require differential equations. It also computes the activation of muscle and the excitation signals of motor units.
  (b) Sub-level of tendon: computes the effect of tendons. This sub-level can contain differential equations with time constant from 10 ms to several s.
  (c) Sub-level of aponeurosis: computes the effect of aponeurosis. This sub-level can also contain differential equations with time constant from 10 ms to several s.
- *Level of segments*: is responsible for the computation of the movement of segments from the muscle torques and external forces taking into account intersegmental dynamics, body segments and joint kinematics. The computation of the joint angles, their velocities and accelerations take place here. This level contains differential equations with typical time constant s.

We have to note that each level contains nonlinear characteristic and nonlinear sub-models.

Model integration is the process of linking sub-models, that exist at different scales and may have different natures, into a coherent, composite multi-scale model. Connections between the levels can be seen in Fig. 1.

The interface variables have been collected in Tab. 1. The subscript $_. = M, MU, S, T, A$ identifies the object the variable belongs to, where M refers to the musculotendon, MU to the motor unit, S to the sarcomere, T to the tendon and A to the aponeurosis.

The model inputs are the excitation signals of the muscles (e.g. computed from the EMG) and the external effects, while the model outputs are the joint angles and the joint velocities (so called movement pattern).

Tab. 1 Interface variables

Notation	**Unit**	**Definition**
$F_.(t)$	N	Force
$l_.(t)$	m	Length
$v_.(t)$	m/s	Contraction velocity
$q_.(t)$	1	Activation state
$u_.(t)$	1	Excitation signal
$\alpha(t)$	rad	Joint angle (output)
$\omega(t)$	rad/s	Joint velocity (output)
$U_M(t)$	1	Excitation (input)
$M(t)$	Nm	Joint moment
$r_m(t)$	m	Moment arm of muscle force
$\theta_{pinn}(t)$	rad	Pinnate angle

2.2 The structure and operation of the simulator

The models developed under the proposed multi-scale modeling framework have been implemented in MATLAB [12] where the user has a possibility to enter his/her special functions and parameter values to collect and tailor the available sub-models realized as MATLAB functions. Each level in our framework is also realized as an object. It means that a separate $.m$ file contains the model configuration for each level and this main file of the level calls the appropriate files containing the possible partial models of the corresponding mechanisms. The different models of a particular mechanism are stored in a separate file where the different models are labeled with an identifier (a string). It means that if the user would like to implement a new model of a mechanism, he/she has to modify only one file containing the possible models of the mechanism, and he/she has to create an identifier of the new algorithm. If the user would like to use an already implemented model of the mechanism, he/she only has to give the identifier in the required model in the model configuration data file. The full simulator contains about 50 files and ≈ 15000 program lines.

From the mathematical point of view, the model is a set of differential and algebraic equations (a DAE model) where the algebraic equations can - in principle - be substituted into the differential ones (an index 0 model [13]). The resulting DAE is a stiff set of equations, therefore, the *ode15s* solver in MATLAB is used for the numerical solution. This solver is a multistep, variable-order solver based on the appropriate numerical differentiation formulas. Optionally, it may also use the backward differentiation formulas [12].

The simulation is divided into two phases. First an *initialization* is performed with inputting the structural and the source parameters together with the initial conditions for the differential equations. The simulator builds the musculoskeletal system model and loads its parameters. The computed parameters are derived in the initialization phase.

The solution of the DAE models is then performed in the *dynamic simulation* phase. To satisfy modularity requirement (such that the user could modify some kind of partial models), the DAE model of a particular level is stored in a file separately from the DAE model of the other levels. As a consequence, the DAE model of a level is solved separately from the DAE model of the other levels. The order of the solution of them is determined a priori. The solution of the DAE of a level can be generated at any time. However, the levels are not independent from each other, i.e. they have to communicate with each other during their solutions. This communication is only taken place periodically and this period time is adjustable.

The user defined limb structure, the musculotendon parameters and the identifiers of the applied mechanism related algorithms are given in text files. The data structure that stores the segments and their connection is a linked list. It is built in the initialization phase of the simulator based on the input files. Parameters and variables of the levels and sub-levels are stored in structures separately. These structures are created in the initialization phase.

Two detailed test models have been realized and tested in this simulator.

- A shoulder and elbow model, that contains two joints and seven muscles.
- A wrist model, that contains one joint and eight muscles.

The number of differential equations and the number of algebraic equations depend on the number of joints, muscles, motor units and sarcomeres. Our shoulder-elbow test model contains 50 differential equations while the wrist test model contains 48 differential equations and both model contains more than 100 algebraic equations.

The simulator was run on a personal computer with Pentium D CPU 3.20 GHz processor, 2 GB Ram equipped with a Microsoft Windows XP Professional Version 2000 Service Pack 2 operating system. The MATLAB version 6.5.1 (SP1) was used.

Tab. 2 shows the execution times needed to compute the a simulated movements with overall duration 1 *sec.* The values of these execution times are high. It can be the consequence of the dynamic equation organization.

Tab. 2 *Execution times in sec.*

Wrist		Elbow	Shoulder	
Palmarflexion	Dorsalfelxion	Flexion	Anteversion	Retroversion
928	613	437	334	324

Verification of the simulator The simulator has been verified against measured muscle excitation and position data using the above described wrist, elbow and shoulder models. The excitation signal is computed from the measured EMG of the muscles while the movement is measured by passive marker based camera system [14].

An example for the measured inputs and the corresponding measured and simulated outputs of an elbow flexion movement can be seen in Fig. 2 and 3. In the initial position the arm hanged vertically and the upper arm was fixed during the movement.

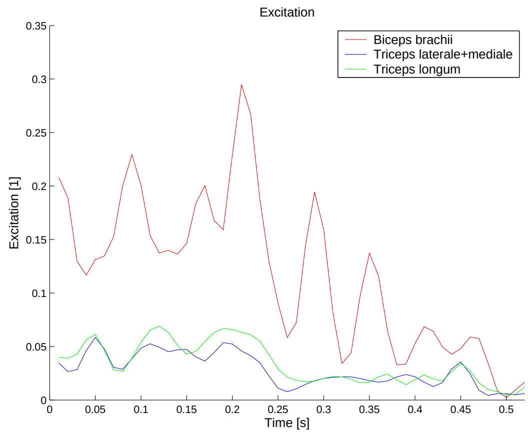

Fig. 2 Measured excitation of muscles during elbow flexion

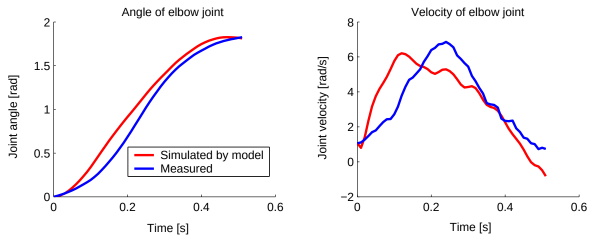

Fig. 3 Measured and simulated movement pattern during elbow flexion

2.3 Extension with receptors and feedbacks

The importance of understanding the way in which sensorimotor feedback influences motor control has increased significantly in the past decades. For investigating these feedbacks the reflex loops should be integrated into the multi-scale musculoskeletal model and its simulator.

The first step of the integration is to determine the location of the receptors (e.g. muscle spindle, Golgi tendon organ) in the model hierarchy. We suppose that the time scale of the receptor is similar to the time constant of its environment, because it ensures that the receptor is sensitive to its environment. Furthermore, receptors apply information that originates only from its environment except for their excitation that originates from the central nervous system (e.g. gamma innervation [1]). This fact enables the receptors to be modeled on the same level as its environment. The receptors have two different kinds of information exchange pathways: the local ones (the monosynaptic reflexes) can be realized as connections between the receptor and objects corresponding the level of the motor unit. The global ones (polysynaptic reflexes and external excitation) can be realized via global interfaces between the receptors and the nervous system.

For example, a muscle spindle is modeled on the level of motor unit (see Fig. 4). The feedbacks of muscle spindle can be realized with the definition of proper interfaces. The information exchange between the muscle spindle and its environment is realized by local connection between them, see *States* and *Excitation from muscle spindle* edges in Fig. 4. The information change between the muscle spindle and the nervous system can be realized using a global interface, see *Output to the nervous system* and *Excitation of the muscle spindle* edges in Fig. 4.

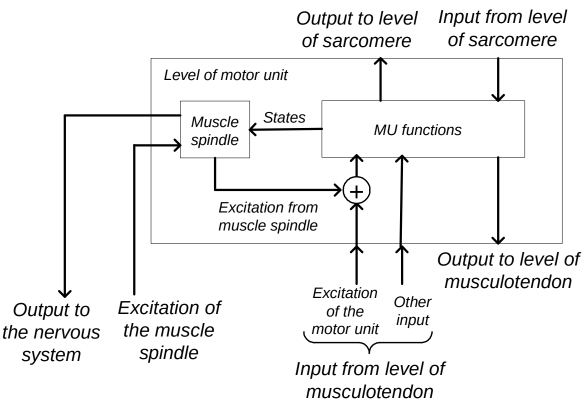

Fig. 4 Muscle spindle model is inserted into the Level of motor unit.

If the model of receptor contains differential equation, its solver can be realized as the other differential equation solver of the model: it can integrate the model at any time, but it provides its data and receives the data of other solvers at every $10\ ms$. The algebraic equations of the receptors and the reflex loops can be evaluated at the same time with the algebraic equations of the corresponding level of the multi-scale model.

3 Applications of the simulators

3.1 Investigation of reflex loops

A possible application of receptor dynamics and feedbacks in the musculoskeletal model is the investigation of reflex loops in human movement.

A simulation study was performed to analyze the feedback effects of a simplified gamma-loop model applying two types of muscle spindle models. The methods and simulation results of this study are detailed in [15].

In figures 5-6 the simulation results of a simple stretch-reflex can be seen in the case of elbow movement. In the beginning of the movement both muscle groups (flexor and extensor) are almost totally inactive, and the limb is in a state around zero degree joint angle (hanging). At about 1.3 sec, a sudden elongation of the flexor muscle was applied for a short period of time. This was im-

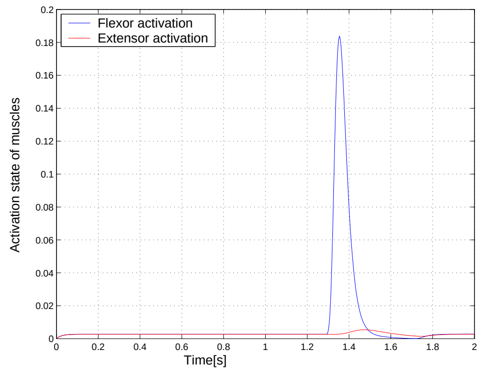

Fig. 5 Muscle activation states

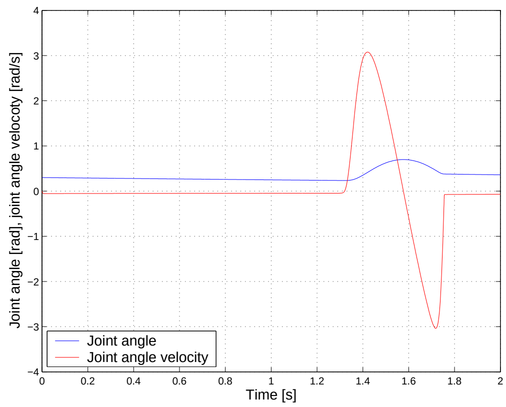

Fig. 6 Limb movement during stretch reflex simulation

mediately sensed by the spindles, and in order to the reflex loop, an increase of activation state appeared in the flexor muscle. This activation provided force and torque to raise the limb. After the disappearance of the stimulus the activation states and the joint angle returned to steady state.

With the tool of taking these simple feedbacks into account, similar to the one described above, many phenomena of interest can be analyzed (alpha-gamma coactivation, polysinaptic feedback etc.) via simulation.

3.2 Other application areas

The proposed simulator has also been applied in the following areas.

- *Analysis of the musculoskeletal systems*: Based on the simulated responses, scale-maps have been constructed, and the detailed model has been reduced based on it (see details in [2]).
- *Parameter estimation*: Movement patterns and EMG signals of muscles have been measured and applied to estimate the maximum isometric force of the muscles via the estimation of a structural parameter (the number of parallel connected sarcomeres) [16]. The simulator has been embedded in

the Nelder-Mead simplex optimization algorithm in this case.

- *Development of controllers*: A simple musculoskeletal limb model (one joint and two muscles) was used for linear and nonlinear controller design purposes [17, 18, 19] in an earlier study.

4 Conclusion and future work

Complex musculoskeletal systems has been modeled with multi-scale modeling technique, that requires a special simulator. In this paper a simulator of the multi-scale musculoskeletal model is presented. It can integrate and solve the different kinds of partial models with different time and size-scales. The different time-scales cause the problem of stiffness, thus a stiff solver has been applied.

The multi-scale model has been extended with receptor models and feedback to simulate reflexes. The anatomical position of receptor determines the location of its partial models in the multi-scale model. Local connection can be defined between the receptors and their environment to realize the information exchange and the monosynaptic reflexes. Only the connection between the receptors and the nervous system require global interfaces.

There are several application areas of the simulator such as analysis of the musculoskeletal dynamics, parameter estimation, analysis of the reflex loops and controller investigations.

Further work is directed towards removing the constraining properties of the simulator: (i) it is only able to simulate planar movement, and (ii) the speed of the solution method does not satisfy our expectations. As a first step, the differential equation solution method will be redesigned to make the simulator faster.

Acknowledgements

This research was supported by the Hungarian Scientific Research Fund and the National Office for Research and Technology through grants no. K67625 and F046223. The third author is a grantee of the Bolyai János Research Scholarship of the Hungarian Academy of Sciences

5 References

[1] R.M. Enoka. *Neuromechanics of Human Movement*. Human Kinetics, Champaign, IL, 2002.

[2] Cs. Fazekas, Gy. Kozmann, and K.M. Hangos. Multi-scale modeling and time-scale analysis of a human limb. *Multiscale Modeling and Simulation, Accepted*, 2007.

[3] E. Villa, L. Mahadevan A. Balaeff, and K. Schulten. Multiscale method for simulating protein-DNA complexes. *Multiscale Modeling and Simulation*, 2:527–553, 2004.

[4] B. Rubinstein, K. Jacobson, and A. Mogilner. Multiscale two-dimensional modeling of a motile simple-shaped cell. *Multiscale Modeling and Simulation*, 3:413–439, 2005.

[5] D.G. Vlachos. A review of multiscale analysis: Examples from systems biology, materials engineering, and other fluid-surface interacting systems. *Advances in Chemical Engineering*, 30:1–61, 2005.

[6] G.D. Ingram, I.T. Cameron, and K.M. Hangos. Classification and analysis of integrating frameworks in multiscale modelling. *Chemical Enginnering Sciences*, 59:2171–2187, 2004.

[7] G.E. Loeb, I.E. Brown, and E.J. Cheng. A hierarchical foundation for models of sensorimotor control. *Experimental Brain Research*, 126:1–18, 1999.

[8] G.E. Loeb and W.S. Levine. Linking musculoskeletal mechanics to sensorimotor neurophysiology. *In: J.M. Winters and S.L-Y. Woo (Edt.), Multiple Muscle Systems: Biomechanics and Movement Organization*, pages 165–181, 1990.

[9] A. Prochazka, D. Gillard, and D.J. Bennet. Positive force feedback control of muscles. *Journal of Neurophysiology*, 77:3226–3236, 1997.

[10] M.G. Maltenfort and R.E. Burke. Spindle model responsive to mixed fusimotor inputs and testable predictions of beta feedback effects. *Journal of Neurophysiology*, 89:797–809, 2003.

[11] Z. Hasan. A model of spindle afferent response to muscle stretch. *Journal of Neurophysiology*, 49:989–1006, 1983.

[12] Mathworks. *Using Matlab version 6.5*. The Mathworks Inc., Natick, 2002.

[13] Cs. Fazekas, Gy. Kozmann, and K.M. Hangos. Hierarchical modelling in biology: a systematic build-up of a limb model. Technical Report SCL-003/2004, Systems and Control Laboratory, Computer and Automation Research Institute of HAS, http://daedalus.scl.sztaki.hu/PCRG, 2004.

[14] Á. Jobbágy, P. Harcos, R. Karoly, and G. Fazekas. Analysis of finger-tapping movement. *Journal of Neuroscience Methods*, 141:29–39, 2005.

[15] D. Csercsik. Construction of simple dynamic models of the gamma-loop mechanism. Technical Report SCL-001/2006, Systems and Control Laboratory, http://daedalus.scl.sztaki.hu/PCRG, 2006.

[16] Cs. Fazekas, Gy. Kozmann, T. Magos, and K.M. Hangos. Parameter estimation of a simple hierarchical arm model. Technical Report SCL-002/2007, Technical Report of the Systems and Control Laboratory, Computer and Automation Research Institute of HAS, http://daedalus.scl.sztaki.hu/PCRG, 2007.

[17] Cs. Fazekas, G. Szederkényi, and K. M. Hangos. Linear and nonlinear control of musculoskeletal systems. *Proceedings of the 25th IASTED International Conference on Modelling, Identification, and Control, MIC 2006, Lanzarote, Spain*, pages 219–224, 2006.

[18] D. Csercsik, G. Szederkényi, and K.M. Hangos. Cascade stabilization and reference tracking of a

simple nonlinear limb model. *Proceedings of the 26th IASTED International Conference on Modelling, Identification and Control, Innsbruck, Austria*, pages 369–374, 2007.

[19] D. Csercsik. *Modern control methods of a simple nonlinear limb model*. Diploma work, Budapest University of Technology and Economics, http://daedalus.scl.sztaki.hu/PCRG, 2007.

ISBN 978-3-901608-32-2