Preprint of the paper submitted to ECCOMAS Multibody Dynamics 2005, Advances in Multibody Dynamics, Madrid, June 21-24, 2005.

MULTIBODY DYNAMICS 2005, ECCOMAS Thematic Conference
J.M. Goicolea, J. Cuadrado, J.C. García Orden (eds.)
Madrid, Spain, 21–24 June 2005

FROM ROBOTS TO HUMANS: TOWARDS EFFICIENT FORWARD DYNAMICS SIMULATION AND OPTIMIZATION EXPLOITING STRUCTURE AND SENSITIVITY INFORMATION

Maximilian Stelzer, Oskar von Stryk

Simulation and Systems Optimization Group, Technische Universität Darmstadt
Hochschulstrasse 10, D-64289 Darmstadt, Germany
e-mail: [stelzer|stryk]@sim.tu-darmstadt.de,
web page: http://www.sim.informatik.tu-darmstadt.de

Keywords: biomechanics, dynamics algorithms, forward dynamics simulation, walking robots, optimization, optimal control techniques

Abstract. *The locomotion of quadruped and humanoid robots shares some characteristics with the one of animals and humans: a kinematic tree structure and a free-floating base, switches in the model due to different contact situations and a high number of degrees of freedom consisting of many links and many actuated joints. One main difference is in actuation: while robots commonly are driven by (at most) one motor for each joint, animals and humans usually use much more than one muscle for each joint. Although a single muscle may also be connected to several joints, the tree-like structure of the multibody system may be preserved. This enables the use of efficient, recursive dynamics modeling methods exploiting tree structure which have already been successfully developed for legged robots .*

Optimization techniques for exploiting redundancy in dynamic motions are presented. Biomechanical systems inhibit two levels of redundancy. First, the motion (e.g. point-to-point) of a whole limb or body can be realized by many different joint trajectories. This is common for robots with legs and arms as well as for animals and humans. Second, the joint torques (and as a result the joint motions) can be generated by different forces of the individual muscles involved in the motion of the joint. Several optimization hypotheses exist how the participating muscles are actuated during joint motion. The backward dynamics simulation and optimization starts from measured joint trajectories to compute an approximation of the required control for the observed motion. The forward dynamics simulation and optimization results in a high dimensional, nonlinear optimal control problem. In principle, it enables the forecast of a free motion for a validated model but is much more computationally expensive. Compared to current approaches reach a computational speed-up by two orders of magnitude using tailored, efficient dynamics modeling, recursively obtained, explicit sensitivity information of the system dynamics and efficient direct transcription methods for solving the optimal control problem by direct collocation (already successfully used for optimization of gaits of legged robots).

Numerical results are presented for walking robots and for the kicking motion of a leg with two joints and five muscle-tendon-groups.

1 INTRODUCTION

A strong need for more insight to biomechanics comes from motion analysis or prediction and prosthesis design, where muscle excitations are of interest but often may cannot be obtained easily, e.g. only by surgery. In this cases simulation may provide a sensible method to determin muscle excitation.

Efficient numerical methods for finding optimal gaits for walking robots have successfully been used and validated in experiments. This paper shall review these methods and show the extensions made to handle biomechanic systems.

The outline of the paper is as follows: Section 2 reviews an efficient dynamics algorithm for walking robots and explains the extension to biomechanics systems. Optimization techniques used are presented in Section 3. Numerical and experimental results are presented in Section 4. Section 5 concludes the paper.

2 DYNAMICS ALGORITHMS

Efficient dynamics algorithms are needed, especially when optimizing high dimensional systems. Walking robots show some characteristics (Section 2.1) which may be exploited in the dynamics algorithms (Section 2.2). Sensitivity information which is useful for optimization may be obtained from this algorithm at low cost (Section 2.3). General extensions needed for biomechanics systems compared to walking robots are explained in Section 2.4, while Section 2.5 describes the special modifications in actuation.

2.1 Characteristics of walking robots

Walking robots generally are high dimensional dynamic systems with a large number of actuated joints and rigid links. Each joint is actuated by at most one motor. Motors are characterized by maximum (short-time or permanent) torque resp. current, angle constraints, gear ratio and axis inertias. Due to different contact situations the kinematic structure of the robot changes periodically. When cutting the contacts, the systems show tree structure, i.e. there are no kinematic loops except with ground contacts.

2.2 Dynamics algorithms for tree structured systems

The basic equations of motion are those for a rigid, multibody system (MBS) experiencing contact forces

$$\begin{aligned} \ddot{\boldsymbol{q}} &= \mathcal{M}(\boldsymbol{q})^{-1}\Big(B\boldsymbol{u} - \mathcal{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) - \mathcal{G}(\boldsymbol{q}) + J_c(\boldsymbol{q})^T\boldsymbol{f}_c\Big) \\ 0 &= \boldsymbol{g}_c(\boldsymbol{q}) \end{aligned}$$

where N equals the number of links in the system, m equals the number of actively controlled joints, $\mathcal{M} \in \mathbb{R}^{N\times N}$ is the square, positive-definite mass-inertia matrix, $\mathcal{C} \in \mathbb{R}^N$ contains the Coriolis and centrifugal forces, $\mathcal{G} \in \mathbb{R}^N$ the gravitational forces, and $\boldsymbol{u}(t) \in \mathbb{R}^m$ are the control input functions which are mapped with the constant matrix $B \in \mathbb{R}^{N\times m}$ to the actively controlled joints. The ground contact constraints $\boldsymbol{g}_c \in \mathbb{R}^{n_c}$ represent holonomic constraints on the system from which the constraint Jacobian may be obtained $J_c = \frac{\partial \boldsymbol{g}_c}{\partial \boldsymbol{q}} \in \mathbb{R}^{n_c\times N}$, while $\boldsymbol{f}_c \in \mathbb{R}^{n_c}$ is the ground constraint force. $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, and $\ddot{\boldsymbol{q}} \in \mathbb{R}$ are the generalized position, velocity and acceleration vectors respectively.

These equations may be established with several algorithms. We use articulated body algorithm (ABA) due to its numerous advantages over other methods. ABA is a recursive numerical algorithm of order N (with N the number of links in the MBS). It is tailored to tree structured, fully three dimensional systems and shows a high flexibility in exchange of parts of the model (kinematic and kinetic data, actuation, contact situations). ABA may be formulated analytically in operator formulation, which due to the special stacked structure of the operators involved numerically may be realized by recursive calculations in three sweeps from base to tip and vice versa [7, 23]. Additional sweeps may be added to handle contact forces and sensitivity information.

The main idea of the algorithm lies in the fact that the mass matrix may be inverted explicitly using a factorization of the mass matrix:

$$\begin{aligned}\mathcal{M} &= (I - K\Theta H)^T D (I + K\Theta H),\\ \mathcal{M}^{-1} &= (I - K\Psi H) D^{-1} (I + K\Psi H)^T,\end{aligned}$$

where the occurring operators have physical interpretations [17]. A review of all the occurring operators, the recursive algorithm and an approach for an object oriented implementation of the algorithm tailored to its structure may be found in [13].

2.3 Sensitivities

Information about sensitivities are essential not only in numerical optimization but also non-linear analysis, parameter identification and calibration. Exact sensitivities are superior to approximations (e.g. by finite differences) but often not available at reasonable cost. Jain [16] showed that in the operator formulation sensitivity information may be gained at low cost from ABA. The resulting iterative algorithms provide sensitivity information. Manipulator Jacobian may be calculated as well as sensitivities of inverse dynamics $\partial \boldsymbol{u}$ and forward dynamics $\partial \ddot{\boldsymbol{q}}$ w.r.t. position, velocity and control variables for tree-structured rigid MBS:

$$\begin{aligned}\partial \boldsymbol{u} &= \nabla_q \boldsymbol{u} \partial \boldsymbol{q} + \nabla_{\dot{q}} \boldsymbol{u} \partial \dot{\boldsymbol{q}} + \nabla_{\ddot{q}} \boldsymbol{u} \partial \ddot{\boldsymbol{q}},\\ \partial \ddot{\boldsymbol{q}} &= \nabla_u \ddot{\boldsymbol{q}} \partial \boldsymbol{u} + \nabla_q \ddot{\boldsymbol{q}} \partial \boldsymbol{q} + \nabla_{\dot{q}} \ddot{\boldsymbol{q}} \partial \dot{\boldsymbol{q}}.\end{aligned}$$

The occurring partial derivatives may be stated in stacked operator notation. The resulting recursive algorithm is an extension of the forward dynamics recursive algorithm with modified inboard sweep and two additional sweeps.

2.4 Extensions to biomechanical systems

Biomechanical systems differ from walking robots in several points. One main difference lies in actuation: While robots are driven by motors, at most one motor for each joint and each motor attached only to one joint, in biomechanics, joints are actuated by muscles, which primarily exert linear forces. Muscles may span over several joints and commonly one joint is connected to several muscles. Nevertheless tree structure may be conserved when muscles are assumed to have no mass or having its mass rigidly attached to the bones. Knowing the force insertion points (which depend on the joint angle, cf. muscle paths Section 2.5.4), torques may be calculated and inserted directly into the problem.

Control variables for walking robots are torques or motor currents, for biomechanic system controls are the muscle activations (cf. Section 2.5.3). Once the linear force for each muscle is determined, each joint's torque is calculated taking into account the muscle path (Sections 2.5.5, 2.5.4).

In contrast to the robots rigid links, biomechanic structure show high flexibility. Especially the wobbling masses should be taken into account (but are not considered yet in this paper).

Contact situation of human feet with the ground are much more complex than rigid robot's feet's contact with the ground. For the example of kicking investigated in Section 4.2 however, this is not necessary because there is no contact that has to be modeled at all.

2.5 Muscle modeling

Each muscle shows some characteristic behavior due to its structure. We review the resulting relations ([24]) and short explanations for them; the structure itself shall not be reviewed here. The relations give factors to be multiplied with the maximum isometric force.

2.5.1 Force-velocity relation

The active force a muscle may exert depends on its velocity. It is equal to the muscle maximum isometric force at zero velocity and equal to zero at the maximum contraction velocity. The active force is higher than the maximum isometric force if the muscle has excentric velocity. The overall relation not only depends on the maximum velocity but also on parameters c_3, c_4 that indicate how fast the force converges to zero with contractive velocity resp. how fast the force converges to the maximum force with excentric velocity. For fast muscles $c_3 \in [0.25, 1]$, while for slow muscles, $c_3 \in [0.1, 0.25]$. The overall force-velocity relation is given by:

$$f_{FV}\left(v^M\right) = \begin{cases} \frac{1-\frac{v^M}{v^M_{max}}}{1+\frac{v^M}{v^M_{max}c_3}} & , v^M \leq 0 \\ \\ \frac{1-1.33\frac{v^M}{v^M_{max}c_4}}{1-\frac{v^M}{v^M_{max}c_4}} & , v^M > 0. \end{cases}$$

Figure 1 shows two examples of the force-velocity relation for a fast and a slow muscle.

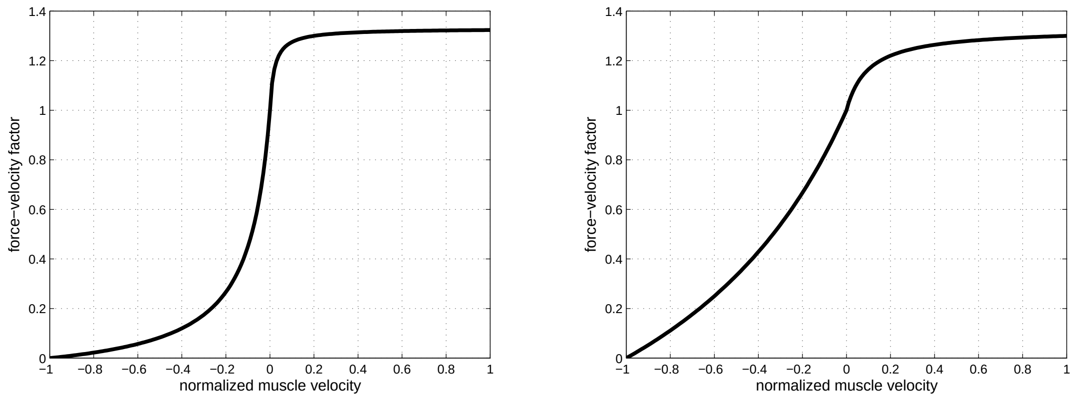

Figure 1: Force-velocity relation for a slow ($c_3 = 0.1, c_4 = 0.02$; left) and a fast ($c_3 = 1.0, c_4 = 0.1$; right) muscle.

2.5.2 Tension-length relation

Muscle forces result from biochemical structures that grip into each other an thereby cause the movement respective force. It is obvious, that the more overlapping structures exist, the

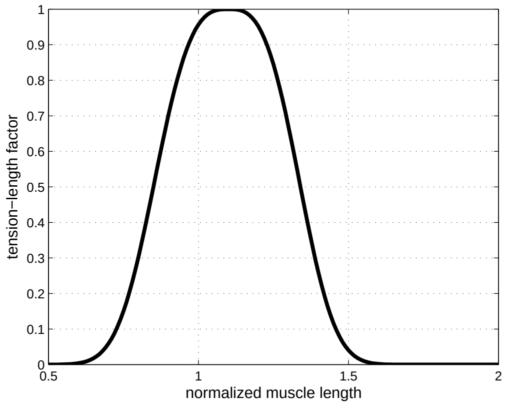

Figure 2: Tension-length relation with $c_1 = 0.017$ and $c_2 = 0.015$.

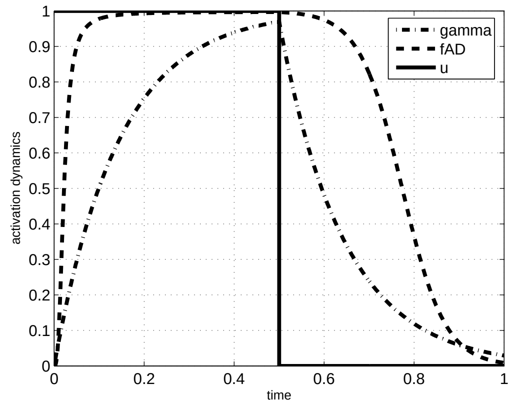

Figure 3: Activation dynamics.

higher are the forces that may be established. If the muscle is expanded, less overlapping area and thus less force exists. If the muscle on the other hand is shortened, the structures obstruct each other and also less force may be exerted. This is modeled with the following equations, where c_1 and c_2 are parameters for the effect of decrease of forces when expanding resp. shortening the muscle:

$$f_{TL}\left(l^M\right) = \begin{cases} e^{-\frac{1}{c_1}\left(1-\frac{l^M}{1.1 l_0^M}\right)^3} & , l^M \leq 1.1 l_0^M \\ e^{-\frac{1}{c_2}\left(\frac{l^M}{1.1 l_0^M}-1\right)^3} & , l^M > 1.1 l_0^M \end{cases}$$

Figure 2 gives an example of the relation.

2.5.3 Activation dynamics

Muscles may not exert force instantaneously. Muscle excitation u leads to increased calcium ion concentration γ in the muscle which finally results in force exertion. This is modeled by:

$$\dot{\gamma} = b_2(b_3 u - \gamma)$$

How the calcium ion concentration relates to the force exerted is given by the following equation:

$$f_{AD}\left(\gamma(u)\right) = \frac{(b_1\gamma(u))^3}{1 + (b_1\gamma(u))^3}$$

The overall muscle activation dynamics is shown in Figure 3.

2.5.4 Muscle path

The muscle lengths and velocities needed for the relations above may be expressed by joint angles and joint angular velocity:

$$\begin{aligned} l^M &= l(q_1, q_2, ...), \\ v^M &= v(q_1, q_2, ..., \dot{q}_1, \dot{q}_2, ...) \end{aligned}$$

To calculate the torques that result from the linear muscle forces, the muscle path, i.e. the force insertion points and force exertion direction (or the resulting lever directly), have to be modeled. Anyway the resulting lever depends on the joint angles only (the first index i indicates the number of the muscle or muscle group, the second index j the number of the joint, the muscle has effects on; not all combinations of i, j are needed):

$$d_{i,j} = d_{i,j}(q_1, q_2, ...).$$

2.5.5 Total muscle force

With the factors given in the previous section, the total muscle force may be stated as:

$$F(\gamma, l^M, v^M) = F^{iso}_{max} f_{AD}(\gamma) f_{TL}(l^M) f_{FV}(v^M).$$

2.5.6 Resulting active torques

The torque in joint j that results from the muscle forces is (with appropriate index sets I_j that indicate with muscles have effect to joint j):

$$\tau_{j,a} = \sum_{i \in I_j} d_{i,j} F_i(\gamma_i, l^M_i, v^M_i).$$

2.5.7 Passive torques

In addition to the active torques, passive torques that depend on $\boldsymbol{l^M}, \boldsymbol{v^M}, \boldsymbol{\gamma}$ (bold letters indicate the vector of all occurring lengths, velocities, calcium ion concentrations), and the joint angles and that model passive effects of tendons, ligament and the connective tissue (especially at the boundaries of the feasible joint angle intervals) have to be considered [12, 32]:

$$\tau_{j,p} = \tau_{j,p}(\boldsymbol{l^M}, \boldsymbol{v^M}, \boldsymbol{\gamma}, \boldsymbol{q}).$$

The total torque applied to joint j is $\tau_j = \tau_{j,a} + \tau_{j,p}$ Note that for robotic systems $\boldsymbol{u}$ is the torque and is equal to the control in the optimal control problem if no detailed motor model is used. For biomechanic systems $\boldsymbol{u}$ is the control (i.e. the muscle activations) and $\boldsymbol{\tau} = (\tau_1, \tau_2, ...)$ are the torque for the dynamics calculations.

3 OPTIMIZATION TECHNIQUES

3.1 Forward vs. inverse dynamics solution

Simulation of a time dependent behavior of a human movement modeled with the techniques stated in Section 2 not only means numerical integration of a high dimensional ODE system but also the solution of a static or dynamic optimization problem for the redundant muscle groups involved. If you consider a sequence of static postures of a movement this results in a sequence of static optimization problems. Their solution however only for slow movements give approximations of acceptable quality to the solution of the dynamic optimization problem over the whole time horizon of the movement (i.e. to the optimal control problem) [2, 9].

Inverse dynamics simulation and optimization

Inverse dynamics simulation for a given, e.g. measured movement calculates the muscle activations of the muscles involved under the assumption of certain criteria for solving the redundancy problem. By this approach practically only given movements may be analyzed; new movements may not be calculated and goal oriented movements (e.g. reaching certain joint angles) may not at all or may only very limitedly be optimized, e.g. [5].

Approaches to extend inverse dynamics simulation to the optimization of human movements rely on very specialized assumptions (like min/max criteria) to the objective function for solving the redundancy problem of the muscles and use a low dimensional parameterization of the free parameter space to efficiently solve the resulting optimization problem numerically [21, 22].

For slow movements dynamic properties of wobbling masses have no effect to the quality of the solution and only for slow movements special min/max-criteria for solving the redundancy problem of the human musculoskeletal system on muscle-tendon-level may be justified. The overall forces and torques at one joint then are distributed to the muscles according to different parameters of the muscles. But if faster movements shall be investigated other optimality criteria have to be used.

From the biomechanic point of view not only faster movements but also other optimality criteria are of interest. By now there are no methods to solve these problems with inverse dynamics simulation satisfactorily. First approaches to the efficient treatment of loops of parallel muscles, may be found in [18]. Inverse dynamics however here also is not solved for any general optimality criterion. In a two-level algorithm first the joint torques and then the muscle forces are calculated.

Forward dynamics simulation and optimization

With forward dynamics simulation, in contrast, analysis of given movements as well as the calculation and optimization of free movements is possible. Starting with the muscle activations (that are to be determined) forward dynamics simulation calculates the resulting movement. By forward dynamics simulation it is possible to analyze movements of parts of the human body or the whole body. This leads to a high dimensional nonlinear optimal control problem.

One advantage of analyzing human movements with forward dynamics simulation is that differences of measured and calculated movements may be integrated into the optimality criterion which allows compensation of measurement errors (e.g. [27]), while with inverse dynamics simulation small measurement errors may result in large errors of the muscle forces.

3.2 Common approaches to forward dynamics optimization

Optimization using forwards dynamics simulation up to now numerically most often is treated by methods that are not optimally tailored to the problem's structure. Most methods transform the optimal control problem into a finite-dimensional, constrained, nonlinear optimization problem (NLP) by parameterization of the controls (direct shooting [31]). The resulting NLP is solved using sequential quadratic programming methods.

For numerical calculation of the gradients of the objective function and constraints w.r.t. the optimized parameters the sensitivity matrix of the solution of the system of differential equations w.r.t. the optimized parameters has to be computed. For human movements this is usually done by external numerical differentiation with differences approximation which is a numerical quite expensive approach [19, 20, 28] because the differential equations of the system have to be integrated numerically at least as often as grid points in the discretization of the controls exist. This leads to overall very high computing times for movements with a higher number of muscle groups.

For example the computing times for human jumping with a leg model with 9 muscle groups and three joints [27, 6] have been reported to be in the region of days on a workstation ([25]). For a three-dimensional model of the whole body with 54 muscle groups computing times on workstations in the region of months have also been reported ([1]).

In [3] computing times are compared when using MIMD parallel computers and vector parallel computers. The method from [20] is used for a 14 dof model with 46 muscle-tendon groups. Computing times were up to three month on a normal computer (SGI Iris 4D25), 77 h on a vector parallel computer and 88 h on a MIMD parallel computer.

The problem investigated in Section 4.2 with 2 joints and 5 muscle groups required computing time in the region of hours on 1996's workstations [26].

3.3 Direct collocation

The direct collocation method DIRCOL ([30]) is used to solve the resulting optimal control problems. States and controls are approximated by piece cubic resp. linear polynomials on a time discretization grid which can be refined successively. The differential equations and nonlinear implicit and boundary conditions are discretized this way to a nonlinear problem with the piecewise coefficients as variables. The resulting NLP is solved using efficient sequential quadratic programming method SNOPT ([8]), which exploits sparsity that is a result of the special structure of the discretization.

3.4 Use of sensitivity information

The sequential quadratic programming method SNOPT does not need user defined derivative information, but may also compute derivatives by difference approximation. However, exact derivatives are useful. First experiments comparing results with SNOPT's derivatives and exact sensitivities gained from the dynamics algorithm showed slight improvements ([14, 15]).

4 NUMERICAL AND EXPERIMENTAL RESULTS

In this section we present numerical and experimental results for walking robots and a biomechanic systems. All optimization calculations have been performed with the method of Section 3.3. DIRCOL. Articulated body algorithm was used for dynamics calculations, except for the biomechanic example, where analytic equations have been implemented directly.

4.1 Robotics: Walking robots

4.1.1 Four legged robot

A full three dimensional models of Sony's four legged robot AIBO ERS-210(A) (see Figure 4) has been optimized ([29]). The robot's four legs consist of three joints each; the main body has been modeled to be rigid. Therefore, the optimal control problem consists of 6 states for the virtual 6-dof-joint for the robot's torso's position and orientation, 12 joint angles and 18 derivatives of those position and angle coordinates due to the transformation of the differential

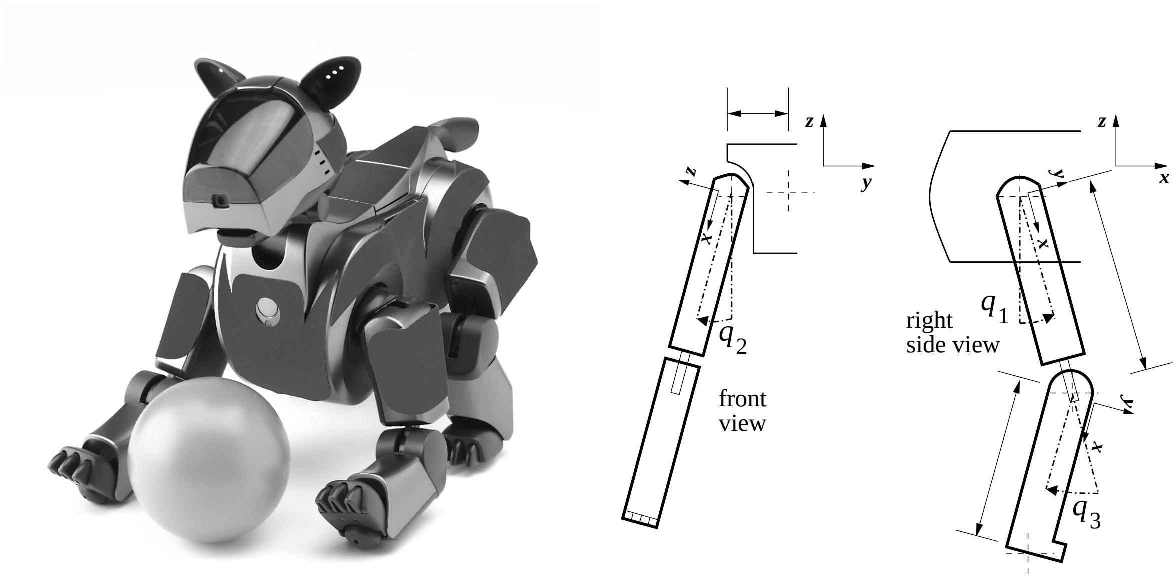

Figure 4: The four-legged Sony robot (left) and the kinematical structure of one leg (right).

equation to a system of first order. The optimal control problem is stated in the following form:

$\min \mathcal{J}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{u}, t_f)$ subject to	minimize the merit function $\mathcal{J}$ subject to
$\mathcal{M}\ddot{\boldsymbol{q}} = B\boldsymbol{u} - \mathcal{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \mathcal{G}(\boldsymbol{q}) + J_c^T \boldsymbol{f}_c,$	system of MBS ODE,
$\boldsymbol{g}_c(q) = 0$	contact algebraic conditions,
$\boldsymbol{b}(\boldsymbol{q}(t_0), \boldsymbol{u}(t_0), \boldsymbol{q}(t_f), \boldsymbol{u}(t_f), t_0, t_f) = 0$	boundary conditions,
$\boldsymbol{n}(q, u) \geq 0$	nonlinear implicit conditions,
$\boldsymbol{q}_{min} \leq \boldsymbol{q} \leq \boldsymbol{q}_{max}, \boldsymbol{u}_{min} \leq \boldsymbol{u} \leq \boldsymbol{u}_{max}$	box constraints on states and controls.

Note that the optimal control problem in this notation contains the differential algebraic equation of multibody system differential equations and contact algebraic equations. However, when solving the optimal control problem, this system of differential algebraic equations is replaced by the reduced dynamics equations (ordinary differential equations of reduced size that, involving the inverse kinematics of the robot's legs, have the same solution like the differential algebraic equations), see [10].

Possible merit functions are for example time, energy or combinations of both. Boundary conditions contain conditions for

- symmetry resp. anti-symmetry of states at boundaries (as only a half stride is optimized),
- foot placement, i.e. conditions that force the feet to be placed on desired positions (which may depend on parameters and therefore are also subject to the optimization),
- contact forces at the end of the stance phase, that allow the foot to lift off.

Nonlinear implicit conditions are:

- Hips of legs in contact with the ground must stay within maximum radius of the leg, so that the inverse kinematics solution required for reduced dynamics has a well-defined solution.

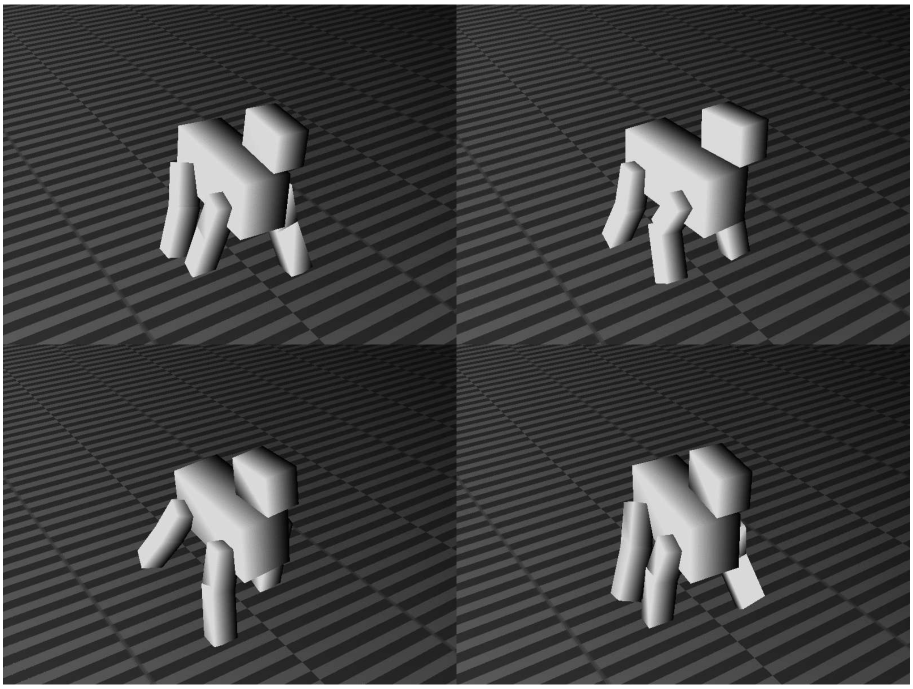

Figure 5: Four scenes from an animation of a computed trot gait.

- The swing feet must move above a certain height relative to the ground, for example a sine curve. This increases stability by avoiding contact with the ground resulting from deflexions of bodies and joints, and which would lead to stumbling of the robot.
- Slipping is avoided by limiting horizontal contact forces relative to vertical contact forces.
- Vertical contact forces must be positive, i.e. the robot may only push to ground but may not pull from ground.
- The next implicit conditions to be introduced to the problem are detailed motor characteristics. By now box constraints for minimal and maximal values of angular velocities and torques give a rough model of the motor.

Note that stability is not enforced explicitly, because it is not that essential for four legged robot and may be checked by one of the criteria given in [11]. More details on each of the constraints may be found in [4], where the constraints are stated for a humanoid robot.

Kinematic and dynamic data has been provided by SONY. Motor data has not been provided but had to be tuned iteratively by implementing the optimized gaits to the real robot and measuring differences in desired and realized joint angle trajectories.

Finally, parameters have been found that result in optimal gaits which may be implemented to the real system and show the behavior expected from simulations. Figure 6 show the joint angle trajectories calculated and measured with the 2nd hip motor not yet tuned; Figure 5 gives some postures of AIBO from an optimized trajectory.

4.1.2 Biped robot

Fig. 7 displays a schematic of a humanoid prototype built in cooperation with a group from TU Berlin (now TU Munich) [4]. See Figure 7 (left) for a picture of the real robot. The

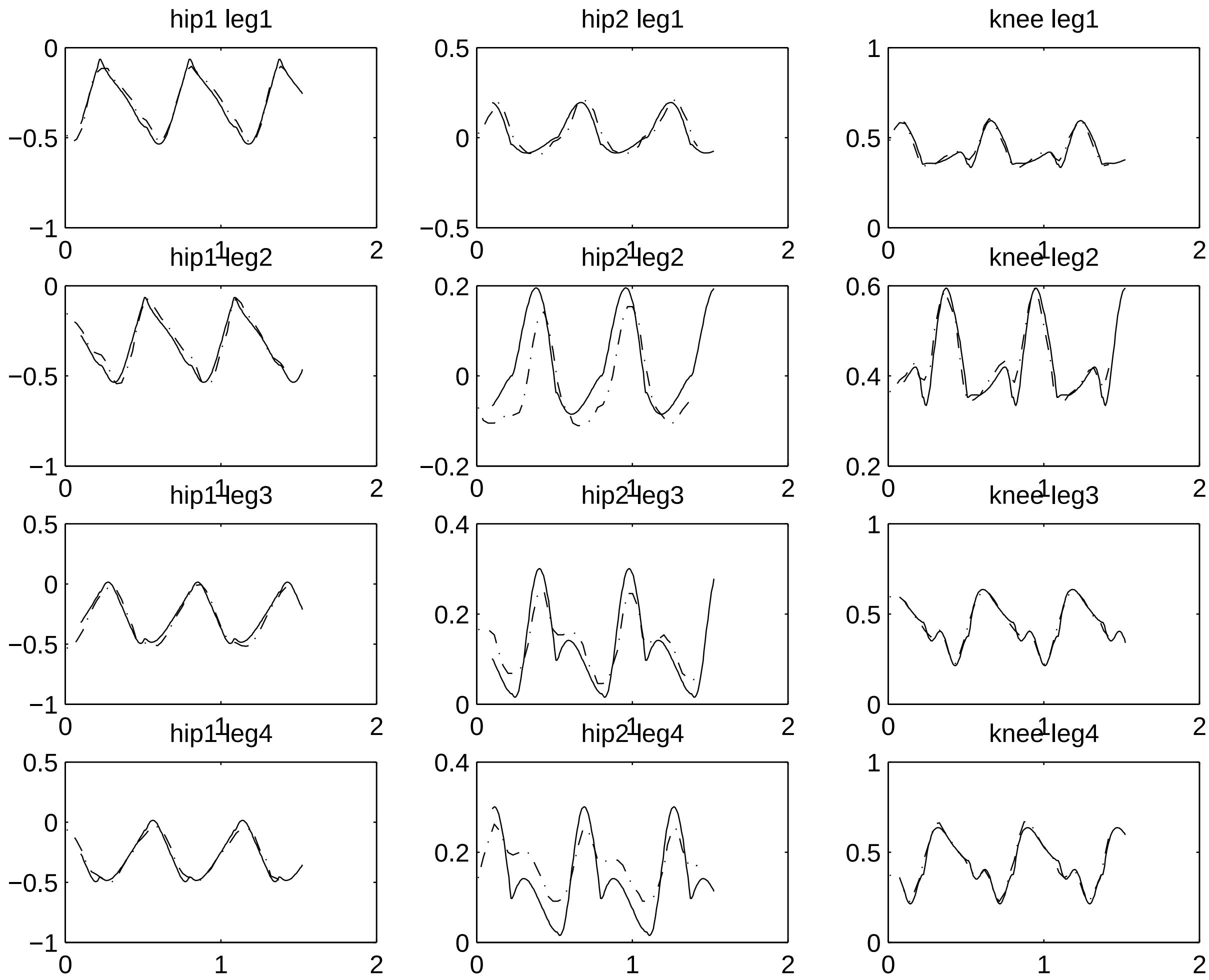

Figure 6: Numerical and experimental results for Aibo ERS-210(A) robot: The experimentally measured joint angle trajectories (dotted lines; joint angle [rad] versus time [s]) for the first hip joints and the knee joints match the computed reference trajectories (solid lines) quite well after considering improved estimates for maximum torque and velocity constraints. For the second hip joints, the constraints have not yet been adapted resulting in the depicted difference. The joint trajectories are shown for about two and half strides of the trot gait.

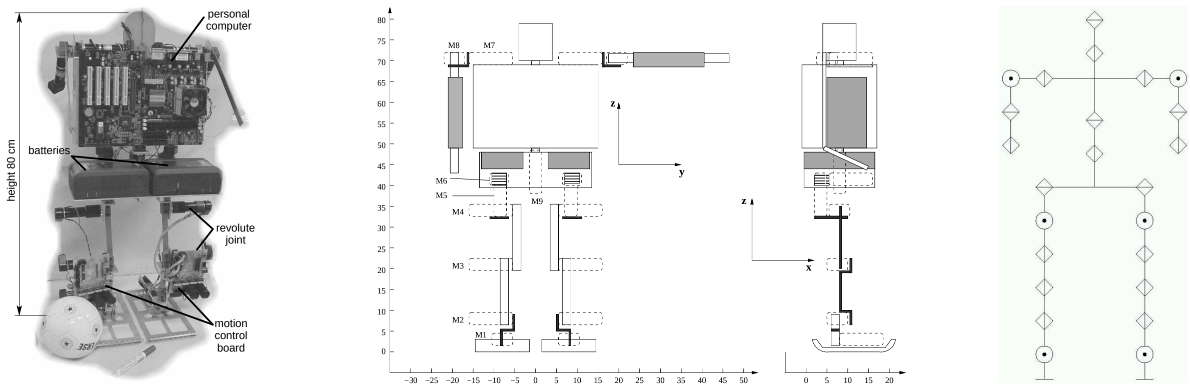

Figure 7: Humanoid Kinematic Structure.

humanoid construction consists of 17 actuated joints:

- two legs each with 6 actuated joints, namely
  - a hip with 3 DoF (*Degrees of Freedom*) rotating about the x-, z- and y-axes,
  - a knee with 1 DoF rotating about the y-axis,
  - an ankle with 2 DoF rotating about the y- and x-axes,
- a waist with 1 actuated joint rotating about the z-axis,
- two shoulders each with 2 actuated joints rotating about the y- and x-axes.

The head is currently fixed to the body, though it is planned to equip the head with 2 actuated joints (pan-tilt) and a CCD-camera.
The humanoid dynamic model consists of:

- 6 DoF describing a fictitious 3D rotation and translation joint between the reference free-floating body (torso) and an inertial reference frame and
- 17 DoF for the existing internal joints.

A total of 23 position and 23 velocity states $(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t))$ resulting in 46 ordinary differential equations describe the system configuration.

$$
\begin{aligned}
\boldsymbol{q} &= \begin{bmatrix} q_{1-3} \\ q_{4-6} \\ q_{7-12} \\ q_{13-46} \end{bmatrix} = \begin{bmatrix} \text{Euler angles for system orientation} \\ \text{System linear translation vector} \\ \text{System angular and linear velocity vector} \\ \text{Legs, waist and shoulder angles and angle velocities} \end{bmatrix} \\
\boldsymbol{u} &= \begin{bmatrix} u_{1-12} \\ u_{13-17} \end{bmatrix} = \begin{bmatrix} \text{Applied torques to legs} \\ \text{Applied torque to waist and shoulders joint} \end{bmatrix}
\end{aligned}
$$

Trajectories for a full three dimensional model of the robot (with the joints of the upper body fixed) have been optimized. The optimal control problem is simular to the one stated for the four-legged robot in Section 4.1.1. Experimental results have shown good matching to the optimized walking trajectories. However, stability even with optimized trajectories is difficult to handle in experiments ([4]). Figure 8 shows a walking sequence of the robot.

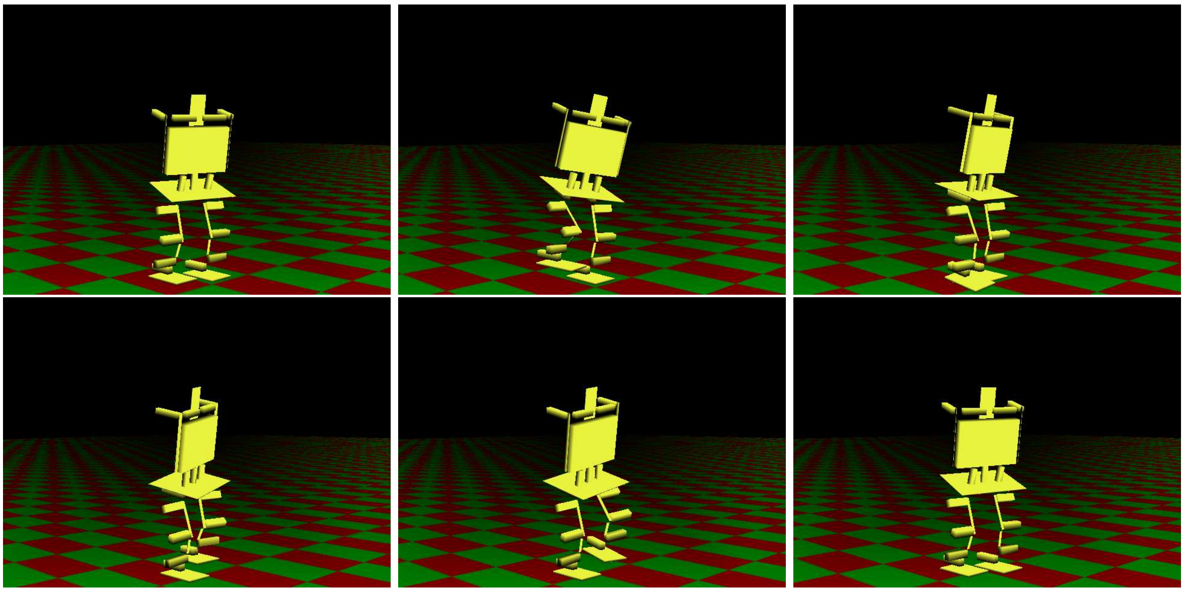

Figure 8: Snapshots of a step sequence.

4.2 Biomechanics: Kicking movement

A time optimal kicking movement has been investigated. Kinematic and kinetic data of the musculoskeletal system as well as muscle model parameters and measured reference data have been taken from Spägele ([24, 27]). The model (cf. Figure 9) consists of two joints, two rigid links and five muscle groups.

The problem is formulated as an optimal control problem with 9 states and 5 controls as follows:

$$\boldsymbol{x} = \begin{bmatrix} q_1 \\ q_2 \\ \dot{q}_1 \\ \dot{q}_2 \\ \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_4 \\ \gamma_5 \end{bmatrix} = \begin{bmatrix} \text{hip angle} \\ \text{knee angle} \\ \text{hip velocity} \\ \text{knee velocity} \\ \text{ca}^{2+} \text{ concentration muscle 1} \\ \text{ca}^{2+} \text{ concentration muscle 2} \\ \text{ca}^{2+} \text{ concentration muscle 3} \\ \text{ca}^{2+} \text{ concentration muscle 4} \\ \text{ca}^{2+} \text{ concentration muscle 5} \end{bmatrix}, \boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} = \begin{bmatrix} \text{activation of muscle 1} \\ \text{activation of muscle 2} \\ \text{activation of muscle 3} \\ \text{activation of muscle 4} \\ \text{activation of muscle 5} \end{bmatrix}$$

The kicking movement was optimized to be time optimal, i.e. the merit function is

$$\mathcal{J} = t_f .$$

Compared to the measured movements (and the results of [24, 27], which match the measured data very well), our results show a shorter time and higher maximum angles. The reason therefore is, that in [24] the maximum muscle forces were modified to match the optimized time of the measurement. Obviously our optimal movement is another local minimum. Nevertheless, the controls (Figure 12) show the same characteristics.

Computing time and size of the resulting NLP are shown in table 10. The direct shooting approach used in [24, 27] for 11 grid points required in the region of hours to compute the solution ([26]). Comparing the computing time with our approach (Figure 10) and considering how computational speed has progressed since 1996, we still obtain a speed up of two orders of magnitude.

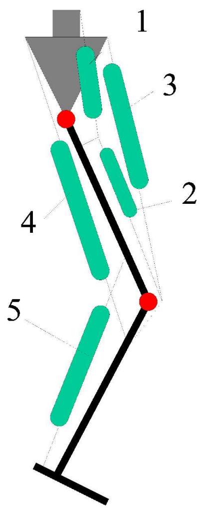

Figure 9: Kinematic structure of the leg with 5 muscle groups.

grid points	10	60
nonlinear constraints	81	829
nonlinear variables	129	531
computing time	1.2 s	6.3 s

Figure 10: Size of the resulting NLP and computation time on a 1700 MHZ+ Athlon XP for two different numbers of grid points in the discretization.

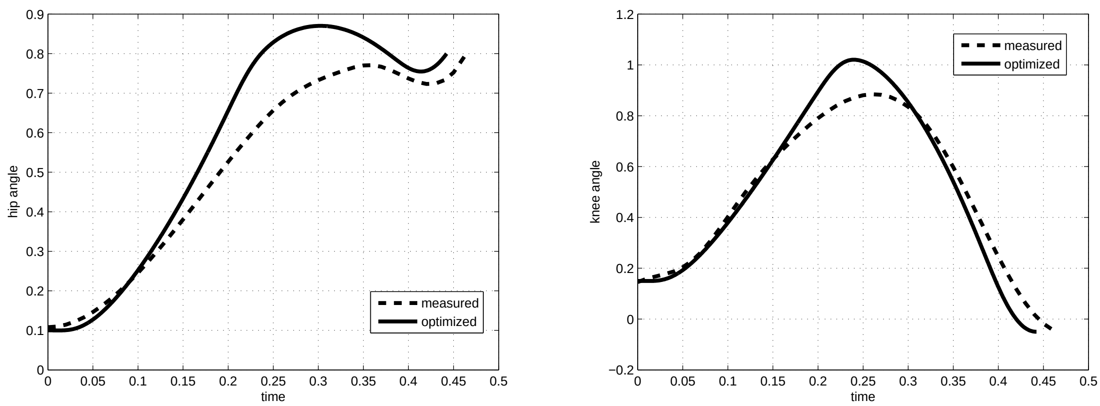

Figure 11: Hip (left) and knee (right).

5 CONCLUSIONS AND OUTLOOK

We reviewed efficient numerical multibody systems dynamics algorithms and optimization techniques that allow solving the forward dynamics optimization in biomechanics two orders of magnitude faster than present methods.

Future work includes refinements of the model:

- wobbling masses and
- a contact situation of the foot, which shall be modeled by a detailed foot model.

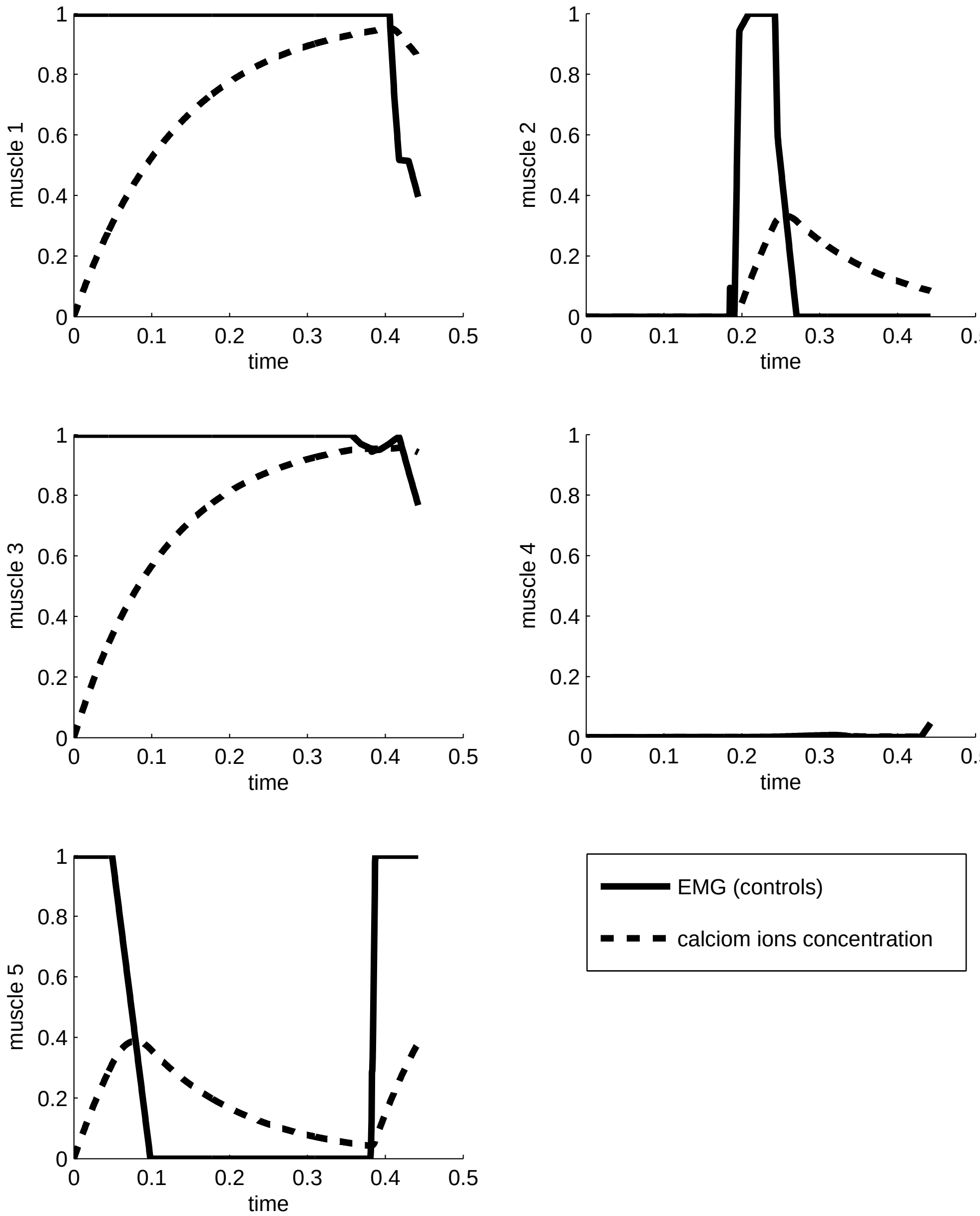

Figure 12: Results from optimization: Controls (corresponding to EMG) and calcium ions concentrations.

Further movements of larger parts of the human body or of the complete human body shall be investigated. Therefore measurements of joint angle trajectories, ground reaction forces and the anthropometric data of the proband are needed.

REFERENCES

[1] F. C. Anderson and M. G. Pandy. A dynamic optimization solution for vertical jumping in three dimensions. *Comp. Meth. Biomech. Biomed. Eng.*, 2:201–231, 1999.

[2] F. C. Anderson and M. G. Pandy. Static and dynamic optimization solutions for gait are practically equivalent. *Journal of Biomechanics*, 34:153–161, 2001.

[3] F. C. Anderson, J. Ziegler, M. G. Pandy, and R. T. Whalen. Application of high-performance computing to numerical simulation of human movement. *Journal of Biomechanic Engineering*, 117:155–157, 1995.

[4] M. Buss, M. Hardt, J. Kiener, J. Sobotka, M. Stelzer, O. von Stryk, and D. Wollherr. Towards an autonomous, humanoid, and dynamically walking robot: Modeling, optimal trajectory planning, hardware architecture, and experiments. In *Proc. IEEE/RAS Humanoids 2003*, page erscheint. Springer-Verlag, 2003.

[5] M. Damsgaard, S.T. Christensen, and J. Rasmussen. An efficient numerical algorithm for solving the muscle recruitment problem in inverse dynamics simulations. In *International Society of Biomechanics, XVIIIth Congress, July 8-13, 2001, Zurich, Switzerland*, 2001.

[6] P. Eberhard, T. Spägele, and A. Gollhofer. Investigations for the dynamical analysis of human motions. *Multibody System Dynamics*, 3:1–20, 1999.

[7] R. Featherstone. The calculation of robot dynamics using articulated-body inertias. *The Int. J. Robotics Res.*, 2(1):13–30, 1983.

[8] P.E. Gill, W. Murray, and M.A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Journal on Optimization*, 12:979–1006, 2002.

[9] R. Happee. Inverse dynamic optimization including muscular dynamics, a new simulation method applied to goal directed movements. *Journal of Biomechanics*, 27:953–960, 1994.

[10] M. Hardt. *Multibody Dynamical Algorithms, Numerical Optimal Control, with Detailed Studies in the Control of Jet Engine Compressors and Biped Walking*. PhD thesis, Electrical Engineering, University of California San Diego, U.S.A., 1999.

[11] M. Hardt and O. von Stryk. Dynamic modeling in the simulation, optimization, and control of legged robots. *Z. Angew. Math. Mech.*, 83(10):648–662, 2003.

[12] A. L. Hof and J.W. van den Berg. Emg to force processing ii: estimation of parameters of the hill muscle model for the human triceps surae by means of a calferometer. *Journal of Biomechanics*, 14:759–770, 1981.

[13] R. Höpler. *A unifying object-oriented methodology to consolidate multibody dynamics computations in robot control*. PhD thesis, Technische Universität Darmstadt, 2004.

[14] R. Höpler, M. Stelzer, and O. von Stryk. Integrated, object-oriented dynamics modeling for design, trajectory optimization, and control of legged robots. In *Proc. Euromech 452 – Advances in Simulation Techniques for Applied Dynamics*, Halle, Germany, 2004.

[15] R. Höpler, M. Stelzer, and O. von Stryk. Object-oriented dynamics modeling for legged robot trajectory optimization and control. In *Proc. IEEE Conf. on Mechatronics and Robotics (MechRob)*, page (to appear), Aachen, September 13-15 2004.

[16] A. Jain and G. Rodriguez. Linearization of manipulator dynamics using spacial operators. *IEEE Trans. Syst., Man, Cybern.*, 23(1):239–248, 1993.

[17] A. Jain and G. Rodriguez. Diagonalized lagrangian robot dynamics. *IEEE Trans. Robotics and Automat.*, 11:571–584, 1995.

[18] Y. Nakamura, K. Yamane, I. Suzuki, and Y. Fujita. Dynamic computation of musculoskeletal human model based on efficient algorithm for closed kinematic chains. *Proceedings of the 2nd International Symposium on Adaptive Motion of Animals and Machines, Kyoto, March 4-8, 2003*, 2003.

[19] R. R. Neptune and M. L. Hull. Evaluation of performance criteria for simulation of submaximal steady-state cycling using a forward dynamic model. *Journal of Biomechanical Engineering*, 120:334–340, 1998.

[20] M. G. Pandy, F. C. Anderson, and D. G. Hull. A parameter optimization approach for the optimal control of large-scale musculoskeletal systems. *Journal of Biomechanical Engineering*, 114:450–460, 1992.

[21] J. Rasmussen, M. Damsgaard, and S. T. Christensen. Optimization of human motion: to invert inverse dynamics. *International Society of Biomechanics, XVIIIth Congress, July 8-13, 2001, Zurich, Switzerland*, 2001.

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

[23] G. Rodriguez, K. Kreutz-Delgado, and A. Jain. A spatial operator algebra for manipulator modeling and control. *International Journal of Robotics Research*, 10(4):371–381, 1991.

[24] T. Spägele. *Modellierung, Simulation und Optimierung menschlicher Bewegungen*. PhD thesis, Universität Stuttgart, 1998.

[25] T. Spägele. personal communication. 2002.

[26] T. Spägele. personal communication. 2005.

[27] T. Spägele, A. Kistner, and A. Gollhofer. Modelling, simulation and optimisation of a human vertical jump. *Journal of Biomechanics*, 32(5):521–530, 1999.

[28] T. Spägele, A. Kistner, and A. Gollhofer. A multi-phase optimal control technique for the simulation of a human vertical jump. *Journal of Biomechanics*, 32(1):89–91, 1999.

[29] M. Stelzer, M. Hardt, and O. von Stryk. Efficient dynamic modeling, numerical optimal control and experimental results for various gaits of a quadruped robot. In *CLAWAR 2003: International Conference on Climbing and Walking Robots, Catania, Italy, Sept. 17-19*, pages 601–608, Aachen, 2003.

[30] O. von Stryk. User's guide for DIRCOL version 2.1: A direct collocation method for the numerical solution of optimal control problems. Technical report, Fachgebiet Simulation und Systemoptimierung, Technische Universität Darmstadt, 2001. http://www.sim.informatik.tu-darmstadt.de/sw/dircol.

[31] O. von Stryk and R. Bulirsch. Direct and indirect methods for trajectory optimization. *Annals of Operations Research*, 37:357–373, 1992.

[32] Y. S. Yoon and J. M. Mansour. The passive elastic moment at the hip. *Journal of Biomechanics*, 15:905–910, 1982.