Forward Dynamics Simulation and Optimization of Walking Robots and Humans

Am Fachbereich Informatik der
Technischen Universität Darmstadt
eingereichte

Dissertation

zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)

von

Dipl.-Tech. Math. Ludwig Maximilian Stelzer
(geboren in München)

Referenten der Arbeit:	Prof. Dr. Oskar von Stryk
	Prof. Dr.-Ing. Peter Eberhard, Universität Stuttgart
Tag der Einreichung:	10. 04. 2007
Tag der mündlichen Prüfung:	24. 05. 2007

Darmstadt 2007

Meinen Eltern!

Contents

Chapter 1

Introduction

1.1 Motivation: Simulation-based Motion Optimization for Legged Robots and Humans

Computer-based modeling, simulation, and optimization enables the study of complex engineering applications and natural phenomena that would be too expensive or dangerous, too complex or even impossible to study by direct experimentation. Besides an important role in engineering design, simulation-based optimization can also be applied to achieve optimal operation of existing systems, e.g., robots. Understanding human walking and running and developing humanoid robots with close-to-human performance and efficiency in locomotion is one of the great challenges in robotics and human biodynamics research.

The mathematical modeling of the motion dynamics of robots and humans leads to large systems of differential equations. Algorithms for efficient and flexible computation of the motion dynamics for simulation and optimization must be employed. The parameters of the motion dynamics models have crucial influence on the quality of the results of the simulation, and therefore their determination and validation are important. Numerical integration of the systems of differential equations yields the time-dependent state trajectories of the motion system.

However, not only simulation techniques are involved when investigating motions of legs or arms. A motion from an initial to a final position of a foot or hand can usually be realized in infinitely many ways by a robot or a human. Redundancy in the actuated motion system offers the opportunity of choosing the ”best” out of a large number of alternative actuation strategies for a leg or arm that all lead to the same final goal position. The redundancy and degrees of freedom in the actuation strategies is much higher in human biodynamics where several muscles are involved in moving a single joint than in robot dynamics where usually only one rotary electrical

motor actuates one rotational joint. Optimization models and methods are needed to determine optimal motion trajectories subject to the motion dynamics models. Whereas engineers may select suitable objectives for a robot motion like minimum time or minimum energy, it is not clear which objective is used mainly unconsciously for a human motion.

Besides the differential equations modeling the system dynamics, several additional conditions (like boundary constraints, nonlinear inequality constraints) must be taken into account to properly state optimization problems for motion dynamics. The task in the resulting optimal control problems is to determine the time-dependent control trajectories (e.g., motor torques or muscle activations) that lead to state trajectories (e.g., joint angles and joint angle velocities) of the system, which represent a physically valid motion by fulfilling the motion dynamics model and satisfying all constraints, and which are optimal in the sense of a given objective function (like time needed or energy consumed for execution of a motion). Optimal control problems for analysis, prediction, and optimization involving computational models are helpful whenever a detailed model of the system is available, when the system investigated is not yet present physically or when the system can not be easily controlled to perform any arbitrary desired task.

Formulating and solving optimal control problems of detailed biomechanical models involves high computational cost. For numerical biodynamical investigations of motions of larger parts of the human body, computational times of weeks or even months have been observed. One common way to reduce the computational burden to a practical amount is to assume special properties of the muscle control strategies and to consider the inverse problem to determine the muscle forces which are required to obtain specific joint torques that lead to a motion measured in experiments. This inverse dynamics approach however has the drawback that it can not be used for general objective functions for muscle actuation, that it is cumbersome to include the activation dynamics, i.e. the dynamics of the chemical reactions that make the muscle exert its force, in this approach and that it can not directly be used to predict human motion. Also, small measurement errors may result in large errors in the calculated forces when analyzing human motion.

While for mechanical multibody dynamics systems like robots or motor vehicles, forward dynamics simulation approaches are well established nowadays, the extension to the motion dynamics of biomechanical systems is still a wide open field. Forward dynamics simulation and optimization approaches in human biodynamics in principle allow a general objective function for muscle actuation, consideration of the activation dynamics, to predict motions based on validated models, and to compensate for measurement errors. However, today's approaches severely suffer from too high computational cost which rule them out for practical use. Numerical methods for robots already allow prediction of motion. For biodynamical applications of human motion, computational times currently are far too high for use on standard computers, which

however would give rise to completely new application scenarios of human body dynamics simulation in medicine and biology.

In this thesis the efficient modeling of locomotion dynamics of robots with four and two legs and of humans, the proper modeling of optimal control problems for optimal motion trajectories as well as efficient optimization techniques for solving numerically the resulting forward dynamics simulation and optimization problems will be investigated.

1.2 Application Scenarios of Legged Robot and Human Motion

The following applications shall give examples of where optimal control problems in the context of this thesis arise. Informal descriptions will be given. More detailed and formal problem statements are the content of Chapter 5.

Optimization of Gaits of Legged Robots

Legged robots can walk in many different ways. Depending on the number of feet, already the sequence of feet touching the ground bears several options. Once this sequence has been selected, usually an infinite number of endeffector (i.e. foot) trajectories leads to the same stepping sequence. The problem of optimizing gaits for legged robots is then to find joint angle controls that lead to a ”best” walking motion of, e.g., highest speed available or lowest energy consumption while ensuring postural stability of the robot and leading to a physically valid motion.

Figure 1.1: Sony AIBO ERS-210; taken from [107].

Simulation based optimization of gait patterns are useful for the design and operation of robots like Sony’s AIBO, cf. Figure 1.1. However, the computational results strongly depend on the quality of the simulation model of the robot’s locomotion dynamics and careful adaption of the computed gaits to the real robot may be necessary. This can be avoided by hardware-in-the-loop optimization, where the real robot is used to evaluate the objective function. However, for the real robot no derivative information, which is needed for most optimization approaches, is available directly. Furthermore, hardware-in-the-

loop optimization of course is not possible if the hardware is physically not present yet. Therefore in pre-build optimization, model based optimal control is the method to be chosen.

Analysis and Prediction of Human Motion

In biomechanical systems, redundancies occur in two different ways: First, one overall motion of legs and/or arms from an initial to a final position generally may be performed by an infinite number of joint angle trajectories; second, as human joints are actuated by redundant muscle groups, a specific kinematic joint angle trajectory may be realized by an infinite number of different activations of the muscles involved. The central problem statement addressed in this thesis is as follows: Based on a simulation model of human motion dynamics find the activations of each muscle involved so that the resulting calcium ion concentrations caused by the activation of each muscle lead to forces which cause joint torques that finally result in a motion of all joints which

1. is equal or as ”close” as possible to the kinematic and/or kinetic data of a human body motion (given, e.g., as joint angle trajectories) measured in experiments (analysis problem), or

2. best fulfills some motion goal like maximum jump height or width or fastest possible walking or running (prediction problem).

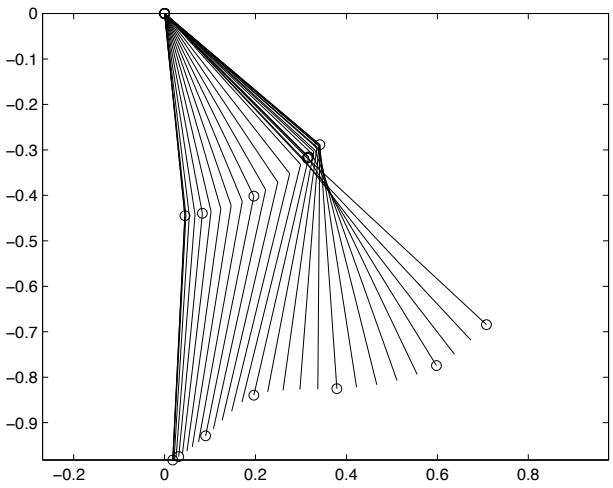


Figure 1.2: Side view of schematic motion samples of a human kicking motion.

While in the first case only the redundancy of the muscles actuating each single joint must be considered, the second case incorporates also the additional level of redundancy with respect to the overall motion. ”Close” in the first case may be measured by an objective function, e.g. the integral over the time of motion of the difference of measured (in terms of joint angles, see Figure 1.2 for a series of leg poses for a kicking motion) and calculated joint angle trajectories. The goal achievement in the second case can be measured as well by a suitable objective function like time or energy required.

Identification of Objective Functions for Human Motion

The objective function involved in solving the redundancies of the muscles plays an important role both in analysis and prediction of human motion. Although there are several heuristic objective functions to be found in literature, the objective function that applies for one specific test person at a specific training level and even for a specific motion task is supposed to differ. Therefore identification of the objective function underlying certain human motions, which can not be measured directly, from motion data that can be measured directly, is an important issue. Identification methods need repeated evaluation of the system, i.e. in this case repeated solution of the underlying optimal control techniques, which requires efficient solution of the optimal control problems to be applicable in reasonable computational time.

1.3 Current State of Research

Accurate and efficient numerical investigation of the forward dynamics simulation and optimization problem in case of the dynamic behavior of large parts of or even the complete human body, consisting of coupled submodels for skeleton, wobbling masses, muscles and tendons and the control mechanisms of the redundant muscle groups involved in a motion, is yet not satisfyingly solved. Kinetic modeling of the muscle-skeleton-apparatus leads to very large systems of differential equations. Usually a large number of controls results from the many redundant muscle groups involved. Moreover, several different hypotheses on suitable objectives and constraints exist for determining the controls of each single muscle involved by simulation and optimization.

Therefore, *forward dynamics simulation* of a human motion leads to high dimensional, nonlinear optimal control problems. Current approaches even for problems with reduced models of the whole human body require computation times of days or weeks on workstations, cf. [7]. Forward dynamics simulation based on a validated dynamics model and model parameters has the important potential of *predicting* certain motions. While forward dynamics simulation is state of the art in vehicle and robot dynamics, e.g. [42, 44, 90, 91], it is still at an early stage in the area of human motion.

On the other hand, *inverse dynamics simulation* investigates given kinematic position and velocity trajectories of a human motion (e.g. by measurement). Together with appropriate modeling approaches it allows a comparatively fast numerical calculation of the controls of each muscle group if very restrictive assumptions on the underlying model like special objective functions for control of the muscles involved are made. Inverse dynamics simulation for a measured human motion gives an *interpretation* of the acting forces and torques on the level of the single muscles involved.

Dynamics modeling

Modeling of the dynamics of human motion involves mainly the following three components of motion generation: (i) skeleton and wobbling masses as a mechanical multibody system (MBS), (ii) muscles and tendons as the (redundant) actuators of the system with inherent dynamic behavior, (iii) control concepts for the activation of the muscle groups involved in generating the motion.

Multibody dynamics modeling of skeleton and wobbling masses

Several methods and programs for modeling and simulation of the dynamics of general multibody systems of various structures exist which are in principle also applicable to the dynamic modeling of the human motion apparatus, for modeling walking or grasping motion of parts of or the whole human body, e.g. ADAMS, DADS or SIMPACK [90, 91].

The assumptions underlying these general-purpose methods do usually not allow to exploit special structure in MBS. E.g. a standard MBS dynamics formulation with constraints is the descriptor form resulting in a possibly large system of differential algebraic equations (DAEs) of index 3 [90]. By exploitation of special properties of the MBS, e.g. a smaller system of ordinary differential equations (ODEs) with a minimum number of state variables may be obtained which can numerically be solved more robustly and efficiently. Furthermore, only few general purpose tools for MBS modeling and simulation are prepared for the numerical solution of an optimal control problem of the redundant muscle groups involved in a motion.

On the other hand, for four-legged and bipedal walking robots efficient methods for modeling of the robot dynamics have been established in recent years. Dynamic motion behavior of walking robots is characterized by a high number of degrees of freedom and many actuated joints and a tree structured MBS with switching contact situations. Recursive methods like [22, 59] are especially well suited for MBS with a large number of degrees of freedom. For tree structured MBS with constraints and inverse kinematics models (like four-legged or bipedal walking robots) modeling of the constraints by DAEs may be transformed using a reduced dynamics approach to a numerically more efficient and robust solvable system of fewer ODEs [42]. It is therefore worth to investigate the extension from modeling the dynamics of humanoid robots [52, 53, 54, 55] to human body dynamics.

Dynamics modeling of muscles and tendons

For modeling of the dynamic motion and force behavior of muscles as contracting actuators with serial and parallel elasticities and active contractile elements a number of well investigated

models have been developed. They describe the muscle forces in relation to muscle length, muscle velocity and muscle activation as the many models based on the fundamental approaches of Hill and Huxley, cf. [80, 82]. Almost all models from literature assume that the muscle forces act at a point. For non-punctual areas of force application the muscles are divided into several muscles with single points of actuation. Several approaches exist for modeling the muscle paths as the straight line method (modeling the muscle path to connect the points of application in a straight line), the centroid line method (modeling the muscle path to connect the centers of mass of the muscle cross sectional areas) or the obstacle set method (modeling the muscle path to move freely sliding along the bones). A survey of these approaches may be found, for example, in [26, 82].

Control of redundant muscle groups

Investigation of the real control mechanisms of muscles, that apply to reflexes or controlled motion by the central nervous system, is still a wide open subject of research in neurophysiology. Up to now, only few validated approaches for mathematical models exist. In biomechanics, however, it is a widely accepted hypothesis, that the control of the redundant muscles involved in a motion usually follows some optimality criteria. For different types of motion and different test persons different optimality criteria $\mathcal{J}$ have been suggested, e.g. uniform distribution of the weighted forces $\boldsymbol{F} = (F_1, ..., F_{n_m})^T$ needed for a certain joint motion to the muscles involved in some k-norm, where $k = 1, 2, 3, 4$ or ∞, see e.g. [80, 88]. The constant weights are positive characteristics $\boldsymbol{N} = (N_1, ..., N_{n_m})^T$ of the muscle capabilities like cross sectional areas or maximum muscle strength. For $k = \infty$ minimization is performed with respect to the maximum load of the muscles:

$$\mathcal{J} = \sum_{i=1}^{n_m} \left(\frac{F_i}{N_i} \right)^k = \| \boldsymbol{F}./\boldsymbol{N} \|_k^k \,,\ k \in \{1, 2, 3, 4\} \qquad \text{resp. } \mathcal{J} = \left\| \frac{F_i}{N_i} \right\|_\infty = \max_i \left(\boldsymbol{F}./\boldsymbol{N} \right). \tag{1.1}$$

Here, $\boldsymbol{F}./\boldsymbol{N}$ denotes the element-wise quotient of the vectors $\boldsymbol{F}$ and $\boldsymbol{N}$ like in MATLAB notation. Another approach is to minimize the energy consumed by all muscles, consisting of resting heat, activation heat, maintenance heat, shortening heat, and the mechanical work performed [120].

Simulation of dynamic motion

Investigations on human motions has a long tradition; an extensive survey may be found e.g. in [58]. First investigations on dynamic human motion using optimization techniques have been

done in the early 1970ies [16]. However, muscles have not been included to the models; using a criterion of minimum power, the hip and knee torques that lead to an optimal trajectory of the respective joints, have been calculated. In [43] in the mid 1970ies a complete model of the human leg with two joints and five muscle groups was used to investigate a kicking motion.

Simulation of time-dependent behavior of a human motion that is modeled according to the previously mentioned aspects not only means the numerical integration of an ODE or DAE system of large size, but also the solution of a static or dynamic optimization problem for the controls of the redundant muscle groups involved. If a sequence of static frames (snapshots) of a motion is considered, this results in a sequence of static optimization problems. Their solution however is only for very slow motions a reasonable approximation to the solution of the dynamic optimization (i.e. optimal control) problem over the continuous time span of the whole motion, see, e.g. [6, 40].

Inverse dynamics approach

Inverse dynamics simulation for a given, usually measured, motion obtains the activations for the muscle groups involved under the assumption of certain criteria for solving the redundancy problem. Thus, practically only given motions can be analyzed; predictions of motions that are goal-oriented as optimal reaching of a certain position, jumping as high or far as possible, running as fast or energy-efficient as possible etc. can not or can only very limitedly be obtained, e.g. [19].

Approaches to extend inverse dynamics simulation to the optimization of human motion are based on very special assumptions (like min/max criteria) to the optimality criterion for solving the redundancy problems of the muscles and use a low dimensional parameterization of the free parameter space for being able to numerically solve the resulting optimization problem efficiently, see. e.g. [87, 88]. For slow motions dynamic properties of the wobbling masses do not effect the quality of the solution, and only for slow motions special min/max criteria for solving the redundancy problems of the human motion apparatus on the level of muscles and tendons are justified. Distribution of the total forces that act at one joint and of the torques to the muscles then is done according to different parameters of the muscles. But if faster motions shall be investigated, other optimality criteria must be used.

From a biomechanical point of view it is desired not only to investigate fast motion but also to use and evaluate different optimality criteria. Up to now there are no methods to solve these problems with inverse dynamics simulation satisfyingly. First approaches to the efficient treatment of loops that occur due to parallel muscles, may be found in [76]; inverse dynamics also is not solved for general optimality criteria there. In an approach of two stages first the joint torques and then the

muscle forces are calculated.

Activation dynamics can not be directly included into inverse dynamics models. First approaches to combine inverse dynamics analysis with muscle activations are given in [36].

The inverse dynamics approach is also used for the Anybody project [2], which allows setting up high dimensional models and is used for several applications like automotive, rehabilitation, aerospace, and sports.

Forward dynamics approach

With forward dynamics simulation, on the other hand, both analysis of given motion and calculation and optimization of free motion are possible in principle. Starting from the muscle activations (which are to be determined) forward dynamics simulation calculates the resulting motion. Analysis of motion of parts of or even the whole human body is possible with it and leads to a high dimensional nonlinear optimal control problem. Advantageous with the analysis of human motion by forward dynamics simulation and optimization is the fact, that differences of measured and calculated motion may be included in the optimality criterion by an additional term consisting e.g. of the integral of the square of the deviation. Thus, measurement errors may be compensated for, while with inverse dynamics simulation small measurement errors in the given kinematic trajectories may lead to large errors in the calculated muscle forces.

Numerical optimization using forward dynamics simulation currently most often is treated by application of methods of transforming the optimal control problem by parameterization of the controls (direct shooting) [115] to a finite-dimensional, constrained, nonlinear optimization problem, which is solved by methods of sequential quadratic programming (SQP) type. These approaches are usually not tailored to the problem structure. For the numerical calculation of gradients of the objective function and constraints with respect to the optimization parameters of the control parameterization the sensitivity matrix of the solution of the (ODE or DAE) state differential equations with respect to the optimization parameters has to be calculated [63]. For human motion dynamics this is often done by external numerical differentiation (END) with difference approximation [78, 84, 98]. END is not only computationally very expensive because the differential equations have to be integrated at least as often as the number of grid points in a piecewise polynomial discretization of the controls and, thus, leads to extremely high computation times for motions with a large number of muscle groups. But also additional errors caused by uncoordinated variable step size integration may cancel many if not all valid digits of the gradient approximation. Therefore, so-called internal numerical differentiation (IND) methods are preferable [63] which efficiently and reasonably accurate compute the sensitivity matrix with an extended numerical integration method and using the ODE Jacobian as additional input.

For example the calculation times for vertical jumping motions of a planar leg model with 9 muscle groups [21, 99] on a workstation have been reported to be within days [95]. For a spatial model of the whole human body with 54 muscle groups even computation times on workstations in the region of months have been reported [7]. In [9] computation times using a normal computer are compared with those using MIMD (multiple instruction multiple data) parallel and vector parallel computers. The method from [84] is applied to a 14 DOF model with 46 muscle tendon groups. Computation times range from one to three months on a normal computer (SGI Iris 4D25), 77 h on a vector parallel computer and 88 h on a MIMD parallel computer. Approaches to reduce computation time of forward dynamics approaches using pre-defined profiles of muscle activations are given in [109].

Identification of objectives for muscle control

Identification of the objective functions that are supposed to be minimized for the involvement of the redundant muscles is a key topic since only with justified objective functions reasonable results can be computed. In [94], load sharing patterns are obtained directly from the muscle forces by an inverse approach. In [110], indirect approaches are described. General identification methods of objective functions on the solution of the optimal control problem are based on sensitivity analysis [64, 68, 69, 71, 72] and its extension to constraints on controls and states [70, 73, 74]. Methods exist both for sensitivity analysis of the discretized optimal control problem [12] and the optimal control problem itself [14].

Application scenarios investigated in the literature

Due to the high computational effort for treating the whole human body, currently only parts of the human body and its interaction with the world are considered, e.g. [7, 31, 43, 77, 79, 122]. In [10, 29, 61, 78] cycling motion is investigated, in [29] to find an optimal cycling machine. In [61] to solve the optimal control problem, the differential equations are not, like commonly done, treated by direct shooting but with a direct collocation approach. A model of a single leg is used for handling a vertical jumping motion in [99]. In [8] a walking motion is optimized. Here, a three dimensional model with 10 segments, 23 DOF (including a 6 DOF free floating base) and 54 muscle-tendon-units is used. In [39] approaches for foot and muscle modeling for generating stable walking motion have been investigated. Skeletal dynamics, muscle paths, muscle tendon actuators and the relationship between muscle activation and muscle contraction have been examined in [82]. An extended approach to muscle path modeling may be found in [25]. From data of the ”Visible Human Male” project [1] and in vivo measurements a dynamical model has been

established [26] whose kinematic structure was published in [27]. The necessity of taking into account the special properties of wobbling masses was stated in [38, 67]. Approaches to coupling of wobbling masses to the rigid body model of the skeleton may be found in [37, 67, 92]. Properties of 26 muscle groups of shoulder, elbow, and hand joint are presented in [24]. In [83] a three dimensional model of the knee may be found. Geometric data was gained from dead bodies. The contact areas of thigh and tibia are modeled to be deformable, those between thigh and patella to be rigid. 12 elastic elements describe ligaments and capsules; in total 13 muscle-tendon-units are modeled. An optimization is performed not for a complete motion, but for single points of time. Investigation of control concepts, which are supposed to be applied in bipedal walking in nature, have been made in [48].

1.4 Aims and Outline

This thesis deals with model-based optimal control of walking motions of legged robots and of biomechanical systems using forward dynamics simulation and optimization. The aim is to develop a methodology for the forward dynamics problem in biomechanics that will achieve a speed-up of computational time of two orders of magnitude compared to common methods. This will allow for solving the problem repeatedly in methods to identify the objective functions that occur in human motions.

The outline of the thesis is as follows. Chapter 2 is about dynamics modeling of legged robots and the extensions for modeling of humans. Starting from general multibody systems and motor and gear models, it is investigated how the biomechanical actuation with muscles can be taken into account.

Dynamics algorithms to deal with the system of differential equations are explained in Chapter 3. The Articulated Body Algorithm and its extension to contact and collision situations are chosen because it is tailored to tree-structured systems and well-suited for an efficient object-oriented implementation.

The different kinds of actuators have influence also to Chapter 4, which deals with simulation based optimal control techniques for walking robots and humans. While for walking robots the only kind of redundancy lies in the fact that certain motions goals, like the foot tip position, orientation and velocity at final time, may be reached by usually an infinite number of joint angle trajectories, biomechanical systems bear another level of redundancy: Due to the fact that one joint is usually driven by more than one muscle (often even more than two antagonistic muscles) and that there are muscles, that have effect directly to more than one joint, one specific joint angle trajectory may be realized by an infinite number of muscle activations and muscle

forces. Optimal control problems are used to overcome both levels of redundancy. First, general purpose optimal control techniques based on direct collocation are used, while in the second part of Chapter 4 adaption of the techniques to the special structure of the underlying models are investigated. The last section of Chapter 4 gives insight into the identification methods.

Numerical and experimental results are given in Chapter 5. For both a quadruped and a biped robot walking motions have been optimized. A human kicking motion has been investigated both in terms of analysis and prediction of the motion and a jumping motion was analyzed. The last part of Chapter 5 gives first results for identification of objective functions for a human kicking motion.

The last chapter gives a summary and outlook to conclude this thesis.

Chapter 2

Modeling the Motion Dynamics of Walking Robots and Humans

Biped or quadruped walking robots and the biomechanical motion apparatus in humans or four-legged mammals share some kinematic and kinematical characteristics. However, the differences increase with the level of details that are included into the models of biomechanical systems. The basic properties of both kinds of systems and especially the extensions of models of walking robots to the modeling of biomechanical systems are discussed in this chapter.

2.1 Walking Robots

2.1.1 Multibody Systems

Walking robots and their large scale motions may be treated as tree structured systems of rigid bodies, which are actuated by at most one, usually rotational, motor per joint. In contrast to finite element methods, multibody dynamics simulation does not take into account elastic deformations in the (almost rigid) links. However, this most often is not considered necessary because elasticity is an undesired side effect in standard robotic mechanisms composed of rigid joints and links and is avoided by construction and design as much as possible. Nevertheless, springs may be used to model elasticity in the joints, which for typical robots is about one order of magnitude larger than the undesired elasticities in the rigid links. This is useful not only for biomechanical systems but also for flexible robots, cf. Section 5.1.3.

Walking robots show a special kinematic structure. Usually they can be modeled as tree structured, i.e., the legs and, if needed, arms all start from a common trunk, cf. Figure 2.1. The trunk

is modeled as a joint (which is a virtual joint which physically is not existent) with six degrees of freedom (DOF) to allow the robot for any position and orientation in the space. Each link is followed by one or more (usually driven) joints, which again are followed by one or more links. Thus, no kinematic loops are introduced. This structure gives rise to special dynamics algorithms that can exploit the structure, see Chapter 3. Attention must be paid to the contacts with the ground which at first sight introduce kinematic loops. Those however can be avoided by separate treatment (Section 3.1.2).

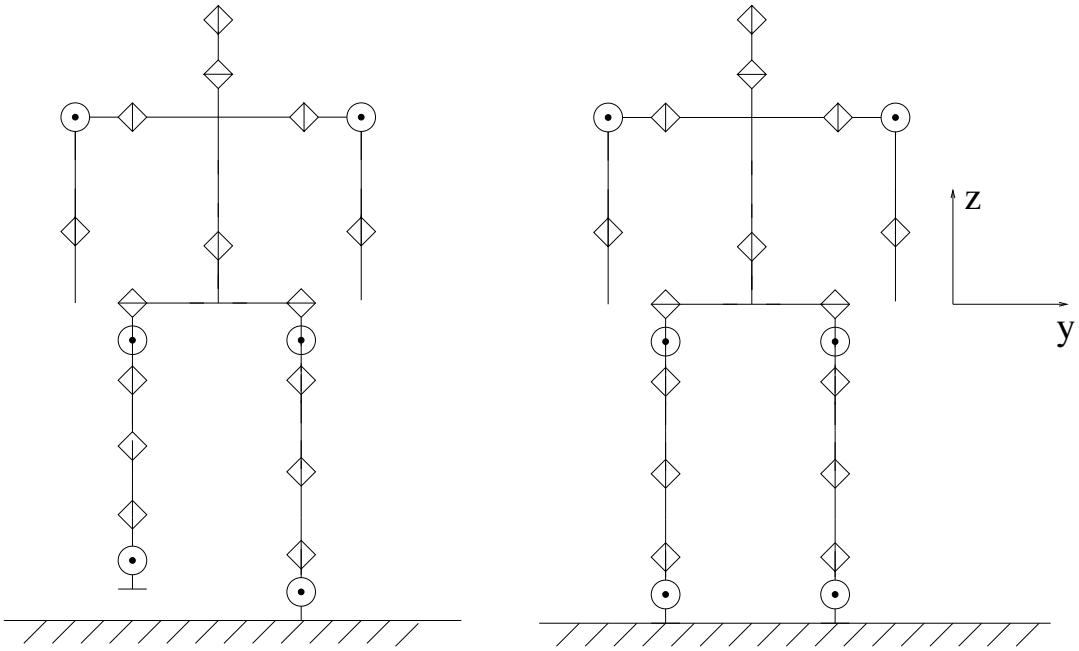


Figure 2.1: Tree structure of a humanoid robot consisting of rigid links and 21 rotational joints: the head, the legs and the arms are attached to a common upper body. Contact with the ground must be treated separately, so that different contact situations (left: single limb support phase, right: double limb support phase) can be treated.

Nevertheless the kinematic structure of the system is changing periodically due to the different contact situations with the ground. For a biped robot phases with two feet, with only one foot or, if the robot performs a real running motion, even no foot in contact with the ground are found. Details on different gait patterns for four- and two-legged robots are stated in Subsection 2.1.3.

Walking motions of robots usually are fast, dynamic motions. Thus, simulation models not only must include the kinematic properties of the system like link lengths or joint orientation but also the kinetic (dynamic) properties like masses, centers of mass, inertia tensors and contact with the ground.

One difficult objective for obtaining models detailed enough for use in motion optimization is the acquisition of their kinetic data. While kinematic data may more or less easily be measured by

careful investigation of the robot with a ruler, for obtaining masses, centers of mass and inertia tensors the robot has to be disassembled for direct measurements, or experiments have to be deducted to identify the parameters numerically from the complete robot, e.g. with the methods described in [45]. While the former is risky with expensive industrially produced robots, the latter requires extensive and eroding experiments and often cannot be done due to missing sensors on the robot. The required data for the results in Section 5.1 were delivered by the manufacturer (for Sony AIBO; parameters for the simple ground reaction model were estimated iteratively by experiments) respectively gained from construction schemes (for the biped robot developed in our group).

Actuation of humanoid robots is comparatively simple: One joint commonly is driven by (at most) one motor and there usually are no motors that drive more than one joint. An example of a robot that follows a more complex actuation structure is shown in Section 5.1.3.

General multibody systems are usually high dimensional nonlinear systems. Both for integration and optimization it is necessary to compute the forward dynamics, i.e. to calculate the joint angle accelerations from the joint angles, joint angle velocities, and the joint torques.

The general MBS differential equation for a rigid, multibody system experiencing contact forces, is given by

$$\mathcal{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} = B\boldsymbol{\tau} - \mathcal{C}\left(\boldsymbol{q}, \dot{\boldsymbol{q}}\right) - \mathcal{G}\left(\boldsymbol{q}\right) + J_c(\boldsymbol{q})^T \boldsymbol{f_c}, \tag{2.1}$$

$$0 = \mathbf{c}(\mathbf{q}). \tag{2.2}$$

Here, with the number N of joints in the system, the number m 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 $\boldsymbol{\tau}(t) \in \mathbb{R}^m$ 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 $\mathbf{c} \in \mathbb{R}^{n_c}$ represent holonomic constraints on the system from which the constraint Jacobian may be obtained, $J_c = \frac{\partial \mathbf{c}}{\partial \mathbf{q}} \in \mathbb{R}^{n_c \times N}$, while $\mathbf{f}_c \in \mathbb{R}^{n_c}$ is the ground constraint force. $\mathbf{q}$, $\dot{\mathbf{q}}$, and $\ddot{\mathbf{q}}$ are the generalized position, velocity and acceleration vectors respectively.

Computation of $\ddot{\boldsymbol{q}}$ requires to solve the linear system, for which in the most general form the computational cost is of order $\mathcal{O}(N^3)$. The mass matrix however is not dense and thus, sophisticated methods can be applied, see Section 3.1. It should be noted that for robotic systems $\boldsymbol{u}$ usually describes the torques in the actuated joints which are equal to the control in the optimal control problem if no detailed motor model is used, see Section 2.1.2. For biomechanic systems $\boldsymbol{u}$ denotes the controls (i.e. the muscle activations, Section 2.2.3.4) and $\boldsymbol{\tau} = (\tau_1, \tau_2, \ldots)$ are the torques for the dynamics calculations, see Section 2.2.3.

2.1.2 Motor and Gear Models

Detailed models of the actuators that drive the robots can substantially improve the computation results. The most basic way to model a motor and gear is to simply model it as a torque-exerting unit. The maximum torque, maximum velocity and maximum acceleration should me limited to avoid motions that cannot be fulfilled by the real robot.

More sophisticated modeling not only includes velocity-torque characteristics and gear efficiency but also may take into account the inertia of the interior parts of the motor and gear [42]. Torque-velocity characteristics are given by a rotational speed of the motor r_0 without load at a specific nominal voltage u_n and a characteristic slope d_m of the torque-rpm relationship. Thus, the maximum rotational velocity of the motor r_m at a certain load τ_m with a certain supplied voltage u_s is given by

$$r_m(\tau_m) = \frac{u_s}{u_n} r_0 - d_m \tau_m.$$

The relationships of the motor-gear combination (with rotational velocity r_r, torque τ_g) hold analogously when taking into account the gear ratio g and the gear efficiency e. Note that the unloaded speed changes linearly with motor gear while the slope shows quadratic relationship:

$$r_r(\tau_r) = \frac{u_s e}{u_n g} r_0 - d_m \frac{e}{r^2} \tau_r.$$

In addition to those relationships, the kinetic parameters of the motor and gear parts may be considered. Note that even comparatively light parts before the gear move with speed multiplied by the gear ratio compared to parts after the gear so that they can contribute considerably to the overall dynamic effects.

This considerations are useful not only for constraints on the optimization problems but also for design of robots where, knowing some motion trajectories and the corresponding torque-rpm-trajectories, appropriate motor-gear combinations may be chosen. This can be done by choosing the motor, gear and power supply such that in the torque-rpm diagram the needed trajectory always is below the limits given by the relationships above. For an example see Figure 2.2.

2.1.3 Gait Patterns

Walking motions of both humanoid and quadruped robots are mimicking walking motions of humans or four-legged mammals like dogs or horses, which usually are periodic motions. Therefore, for walking motions consisting of several periodic steps, only one step has to be investigated. Different gait patterns may be characterized by $2n-1$ parameters for n-legged locomotion [5]:

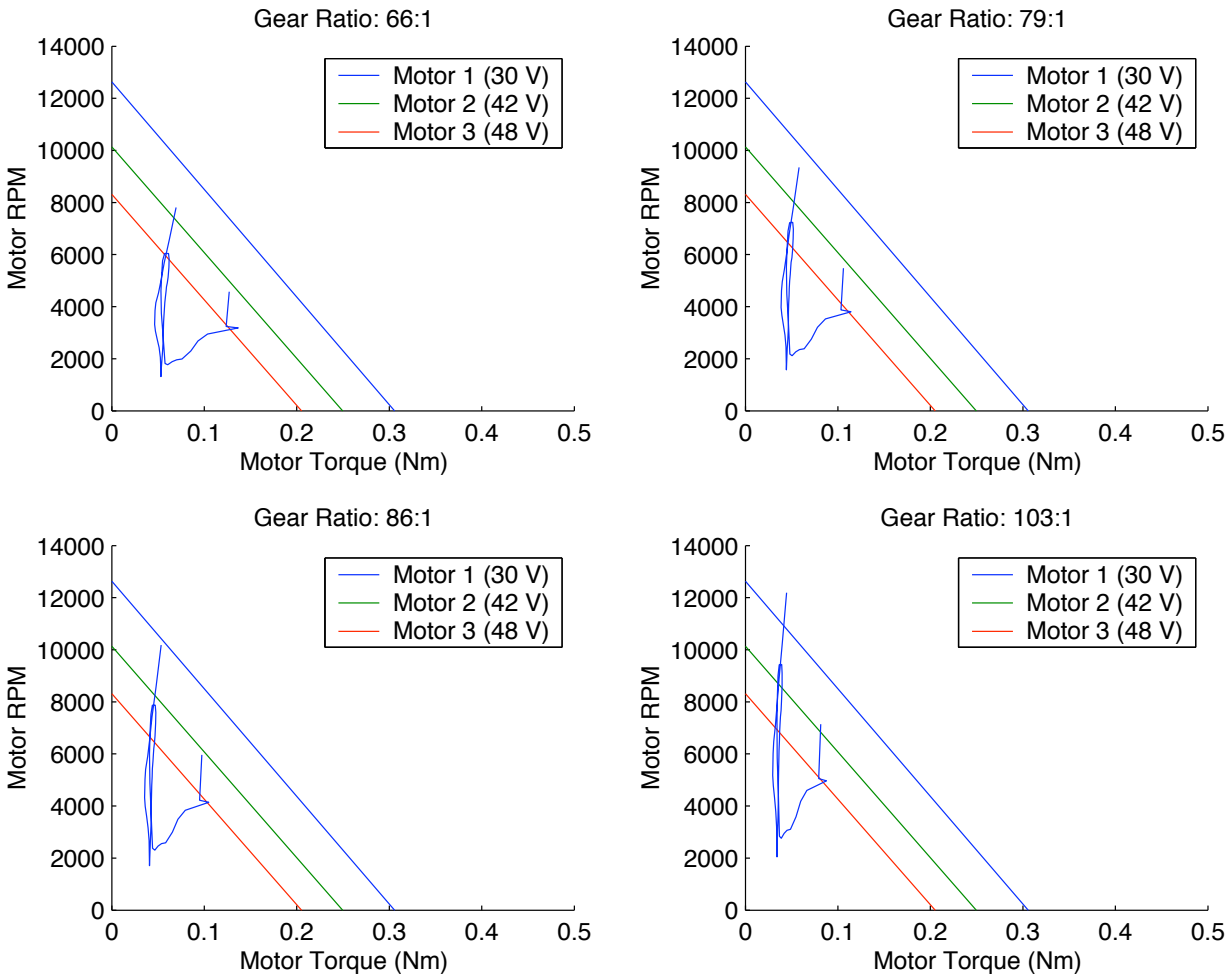


Figure 2.2: Torque-rpm diagrams of several motor gear combinations and a motion trajectory. Motor-gear combinations are capable for the desired motion if the torque-rpm trajectory of the desired motion lies completely below the characteristic line of the motor-gear combination; taken from [42].

- the *duty factor*, i.e., the fraction of the total stride cycle during which each foot has contact with the ground, and
- the *relative phases of each of the legs*, i.e., the fraction of a stride cycle that the respective foot lags behind the motion of the reference foot.

Parameters for different gait patterns of four-legged and biped locomotion may be found in Tables 2.1 and 2.2. Some walking patterns lead to symmetric walking motions, i.e., the second half-stride is equal to the first one with only the legs and some rotational orientations switched. For four-legged gaits, this is the case with walk, trot and rack, for biped gaits with walk, jog and run, and can be exploited if joint angle trajectories are to be optimized; then only one half-stride with

Table 2.1: Parameters for gait patterns of quadruped locomotion

gait pattern	**walk**		**trot**		**rack**		**canter**		**traverse gallop**		**rotary gallop**	
duty factor	> 0.5		$0.3 - 0.5$		$0.3 - 0.5$		$0.3 - 0.5$		< 0.4		< 0.4	
	L	R	L	R	L	R	L	R	L	R	L	R
rel. phase fore	0	0.5	0	0.5	0	0.5	0	0.8	0	0.8	0	0.8
rel. phase hind	0.75	0.25	0.5	0	0	0.5	0.8	0.5	0.6	0.5	0.5	0.6

Table 2.2: Parameters for gait patterns of biped locomotion.

gait pattern	**walk**		**hop**		**jog**		**run**	
duty factor	> 0.5		< 0.5		< 0.5		$<< 0.5$	
	L	R	L	R	L	R	L	R
rel. phase	0	0.5	0	< 0.5	0	0.5	0	0.5

appropriate boundary conditions in the optimization problem formulation to ensure symmetry must be optimized.

Optimization of general walking motions, i.e. finding joint angle trajectories with no certain gait pattern prescribed, leads to a mixed-integer optimal control problem, because every phase of the walking motion has its own kinematic structure due to the different contact situations and thus the underlying mechanical model properties change. This is the reason why usually the desired gait pattern is chosen in advance, see Sections 5.1.1 and 5.1.2. If parameters of a central pattern generator (CPG) are to be optimized like in Section 5.1.3, the gait pattern is a result of the parameters and does not need to be and usually even can not be determined in advance.

In nature smooth transition from one gait pattern to another may be observed, while for conventional legged robots this is a very complex task which is not treated in this work.

2.1.4 Postural Stability

Stability is a critical issue especially for biped locomotion. While four-legged robots may walk at reasonable speed with always at least two feet in contact with the ground, e.g. with a trot gait, biped robots even for the slowest possible walking (except pure gliding on the ground) must lift off one foot and thus only one foot remains in contact with the ground. Therefore a lot of humanoid robots have large, flat feet which ensure a large support area during single limb support phase.

However, the issue of stability can not be solved only by constructional approaches. Also control approaches for offline trajectory computation or online control techniques must be involved.

For both kinds of approaches, postural stability must be judged and measured in some way. For this task several stability criteria and indexes have been introduced during the last decades. The

two important indexes of projected center of mass for static motion and zero moment point for dynamic motion will be discussed in this subsection.

2.1.4.1 Projected center of mass

For the definition of a statically stable walking motion the convex hull of the ground contact points of the feet of the robot is considered. While the projection of the center of mass of the robot along the direction of the gravitational force onto the ground is inside the convex hull of the support area, the robot will not fall. This condition is called *static stability*. If it is fulfilled during a walking motion the motion is called *statically stable*.

2.1.4.2 Zero Moment Point (ZMP)

Ensuring the *dynamic stability* of walking robots, especially those of biped robots, is one of the major challenges in the field of robotics.

For the following consideration all entities are, where reasonable and not mentioned otherwise, given w.r.t. an inertial coordinate system, where the robot walks forward into x-direction, the z-axis points upwards. Ground contact is made in (x, y)-plane.

Stability of a biped robot is influenced (among others) by the force distribution of the ground reaction forces w.r.t. the convex hull of the support area. Two kinds of forces apply: normal and tangential forces.

Normal forces. The normal forces $\boldsymbol{f}_{n_i}$ act normal to the support area, i.e.

$$\boldsymbol{f}_{n_i} = \begin{pmatrix} 0 \\ 0 \\ f_{n_i,z} \end{pmatrix}.$$

The *Zero Moment Point* is the point P where the resulting normal force $\boldsymbol{F}_n = \sum \boldsymbol{f}_{n_i}$ applies:

$$\boldsymbol{r}_{ZMP} := \overrightarrow{OP} := \frac{\sum \boldsymbol{r}_{n_i} f_{n_i}, z}{\sum f_{n_i}, z}.$$

Here, $\boldsymbol{r}_{n_i}$ is the point of application of the force $\boldsymbol{f}_{n_i}$. Because all $f_{n_i,z} > 0$, P lies inside or on the boarder of the convex hull of the points where ground reaction forces apply. In this point P no momentum is exerted due to the normal forces.

Tangential forces. The tangential forces $\boldsymbol{f}_{t_i}$ apply in the contact plane, i.e.

$$\boldsymbol{f}_{t_i} = \begin{pmatrix} f_{t_i,x} \\ f_{t_i,y} \\ 0 \end{pmatrix}.$$

In the point P the applied tangential force can be described by the resulting force $\boldsymbol{F}_t = \sum \boldsymbol{f}_{t_i}$ and the resulting momentum $\boldsymbol{N}_t = \sum \boldsymbol{d}_i \times \boldsymbol{f}_{t_i}$, where $\boldsymbol{d}_i$ is the point of application of the force $\boldsymbol{f}_{t_i}$, w.r.t. the ZMP.
The tangential forces exert a momentum around P, which, however, will not lead to a falling of the robot:

$$\boldsymbol{N}_t = \sum \boldsymbol{d}_i \times \boldsymbol{f}_{t_i} = \begin{pmatrix} * \\ * \\ 0 \end{pmatrix} \times \begin{pmatrix} * \\ * \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ * \end{pmatrix}.$$

ZMP as stability indicator. If the robot falls, the ZMP lies on the boundary of the convex hull of the support area. Goal of a ZMP based stability control thus must be to keep the ZMP strictly inside the convex hull of the support area. The name "Zero Moment Point" is not justified literally because at the ZMP still a moment occurs; however, the components that might lead to a falling of the robot are equal to zero. In literature several slightly different definition of the zero moment point are found [116], which all basically mean the same. Besides ZMP there are several other dynamic stability indicators that may be used for stability control.

Computation of the ZMP. The ZMP can be evaluated by measurement of the contact forces $f_{n_i,z}$ with force sensors at the feet of the robot.

Another option is to directly compute the ZMP from the joint angles, velocities and accelerations of the robot. This is especially useful if no contact force sensors but joint angle, velocity and acceleration sensors are available on the robot.

The moment $\boldsymbol{N_E}$, that is exerted by the reaction forces and reaction moments w.r.t. any arbitrary point $\boldsymbol{r}_E$ on the robot, is calculated by the Euler equation:

$$\boldsymbol{N_E} = \dot{\boldsymbol{L}}_{\boldsymbol{c}} + m_R(\boldsymbol{r_c} - \boldsymbol{r}_E) \times (\ddot{\boldsymbol{r}}_{\boldsymbol{c}} - \boldsymbol{g})$$

Here

- $\boldsymbol{L_c} =^c \boldsymbol{I}\omega$ denotes the angular momentum around the center of mass,
- $\boldsymbol{r_c} = (r_{c,x}, r_{c,y}, r_{c,z})^T$ the position vector of the center of mass of the robot,

- $\boldsymbol{g}$ the vector of gravitation, and
- m_R the total mass of the robot.

Now $N_{E,x} = N_{E,y} = 0$ is implied, which leads to

$$\begin{pmatrix} 0 \\ 0 \\ N_{E,z} \end{pmatrix} = \begin{pmatrix} \dot{L}_{c,x} \\ \dot{L}_{c,y} \\ \dot{L}_{c,z} \end{pmatrix} + m_R \begin{pmatrix} r_{c,x} - r_{E,x} \\ r_{c,y} - r_{E,y} \\ r_{c,z} - r_{E,z} \end{pmatrix} \times \begin{pmatrix} \ddot{r}_{c,x} \\ \ddot{r}_{c,y} \\ \ddot{r}_{c,z} + g \end{pmatrix}$$

and after multiplication of the cross product

$$\begin{pmatrix} 0 \\ 0 \\ N_{E,z} \end{pmatrix} = \begin{pmatrix} \dot{L}_{c,x} \\ \dot{L}_{c,y} \\ \dot{L}_{c,z} \end{pmatrix} + m_R \begin{pmatrix} (r_{c,y} - r_{E,y})(\ddot{r}_{c,z} + g) - \ddot{r}_{c,y}(r_{c,z} - r_{E,z}) \\ (r_{c,z} - r_{E,z})\ddot{r}_{c,x} - (\ddot{r}_{c,z} + g)(r_{c,x} - r_{E,x}) \\ * \end{pmatrix} .$$

Solving the first component for $r_{E,y}$ and of the second component for $r_{E,x}$ finally gives:

$$\begin{aligned} r_{E,x} &= \frac{-\dot{L}_{c,y} + m_R\, r_{c,x}(\ddot{r}_{c,z} + g) - m_R(r_{c,z} - r_{E,z})\ddot{r}_{c,x}}{m_R(\ddot{r}_{c,z} + g)} \\ r_{E,y} &= \frac{\dot{L}_{c,x} + m_R\, r_{c,y}(\ddot{r}_{c,z} + g) - m_R(r_{c,z} - r_{E,z})\ddot{r}_{c,y}}{m_R(\ddot{r}_{c,z} + g)}. \end{aligned}$$

2.2 Humans

2.2.1 Bones

The human body consists of 206 bones in total. For locomotion, the most important ones are those 64 of the lower extremities. In the content of this work, the physiological functions of the muscles (formation of blood cells and storage of calcium) can be neglected. The mechanical properties are the important ones. Basically, bones have three mechanical functions [80], where the first and second one are of importance to motion studies:

- support of the body against external forces like gravity,
- lever system to transfer forces like muscle forces, and
- protection of internal organs like the brain.

The bones are modeled to be rigid, i.e. they do not show any deformation under external forces. Because forward dynamics analysis of large-scale motions cannot be done using FEM techniques, the bones are modeled in the sense of MBS. This implies, that geometry of the bones cannot be directly considered, see Figure 2.3. A simple hinge joint is modeled to be at both ends of the bones.

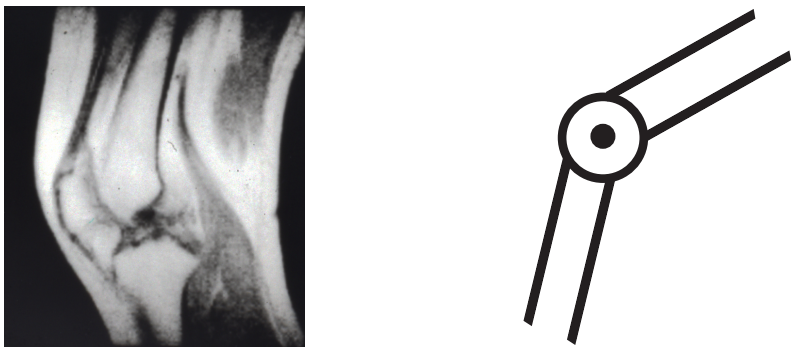


Figure 2.3: MRT image of the human knee joint and its representation in the MBS model. Notice the problem of finding the right axis of rotation and the muscle paths.

One problem that arises from the simplified modeling is that the axis of rotation may not be fixed for bones with certain shapes. While for hinge joints like the hip a fixed axis of rotation may be a reasonable approximation, this is not obvious for more complicated joints like the knee or the shoulder.

The models used for this work have fixed axes but there certainly is space for closer investigations of this issue. Moving axis e.g. could be modeled by extending the model by additional axis (i.e. by standard components) or establishing a new component that takes into account the additional offset, orientation change and forces and torques that occur due to the moving axis.

2.2.2 Wobbling Masses

The human body does not consist of rigid material only. All material that can not be actively controlled and that is not rigid is classified as ”wobbling mass”. What kind of tissue is regarded as wobbling mass depends on the problem that is actually investigated. For a jumping motion for example the complete waist may be regarded as wobbling mass although there are muscles in it; the muscles then would be modeled to belong to the wobbling masses as they are not needed for force exertion in that specific scenario.

There are approaches to model wobbling masses by rigid masses that are attached to the body by spring-damper elements [37, 67, 92]. This introduces additional degrees of freedom to the system. The number of controls remains unchanged because the wobbling masses are not driven.

One interesting issue about wobbling masses is that they may change their characteristic behavior, i.e., the properties of the springs and dampers. One example for this is the different stiffness of the waist when muscles are strained resp. are not strained.

For the examples given in Section 5, wobbling masses are not taken into account yet.

2.2.3 Muscles, Tendon and Ligament

Muscles are the force-exerting elements of the human body. Muscles are classified into three kinds: skeletal, cardiac and smooth. Only skeletal muscles can be controlled consciously and only they are considered for this work. Muscles actively can only exert forces into the direction of shortening. This is the reason why muscles act antagonistically on each joint: There are usually at least two muscles for one joint, one for each direction. Furthermore, there are muscles that span more than one joint, i.e. muscles that have effect to more than one joint. The human body comprises more than 600 skeletal muscles.

In the following subsections, a short review of the most important properties shall be given. The outline follows basically the one from [97].

2.2.3.1 Sliding Filament Theory

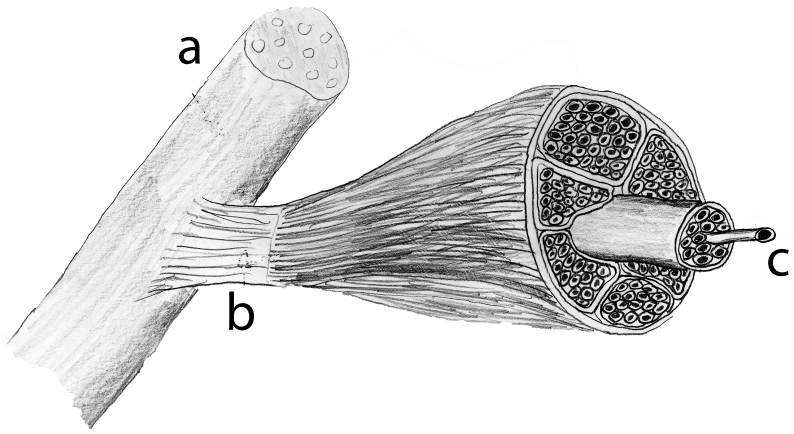


Figure 2.4: Muscles are connected by tendon (b) to the bone (a) and consist of several fibers (c).

Each muscle consists of several parallel muscle fibers, see Figure 2.4. The fibers consist of a row of force exerting units, the sarcomeres (Figure 2.5). Each sarcomere can shorten to a minimum length. Thus, the longer the fibers are the higher is the absolute maximum shortening of the

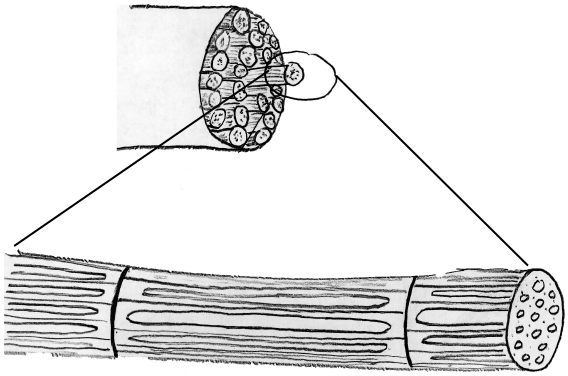


Figure 2.5: Every muscle fiber consists of several sarcomeres.

muscle and the maximum contraction velocity. The more fibers the muscle has in parallel, the higher is the maximum force the muscle may exert.

The most basic level of force generation lies in the sarcomeres. One widely used but not yet fully proved hypothesis is the sliding filament theory. The sarcomeres are supposed to consist of myosin filaments and actin filaments. The myosin filaments can slide along the actin filaments. This motion can be due to external forces, e.g. forces from antagonist muscle, or by active motion of several small heads that tilt and thus make the complete filament slide, see Figure 2.7.

The molecular properties can be investigated using models of the Huxley type [56, 57]. These models are very detailed and thus far to complex for use in efficient forward dynamics simulation and optimization. The models of choice are Hill type models [49] based on phenomenological approximation of the properties of the muscle which however have strong motivation from the sliding filament theory.

2.2.3.2 Force-length-relation

The passive force a muscle exerts when it is lengthened above its rest length by external influences increases exponentially. The total force of an activated muscle at different length, the maximum isometric force, has been measured in [34]. If the passive forces are subtracted from the maximum isometric forces the active force is received, see Figure 2.8.

The sliding filament theory with the moving myosin heads supplies strong explanation for the shape of this relationship. Force is generated by heads in the myosin filament gripping into the actin filament. The higher the overlapping area of actin and myosin filaments is during the lengthening of a muscle, the more heads may get grip and thus the higher is the active force, see

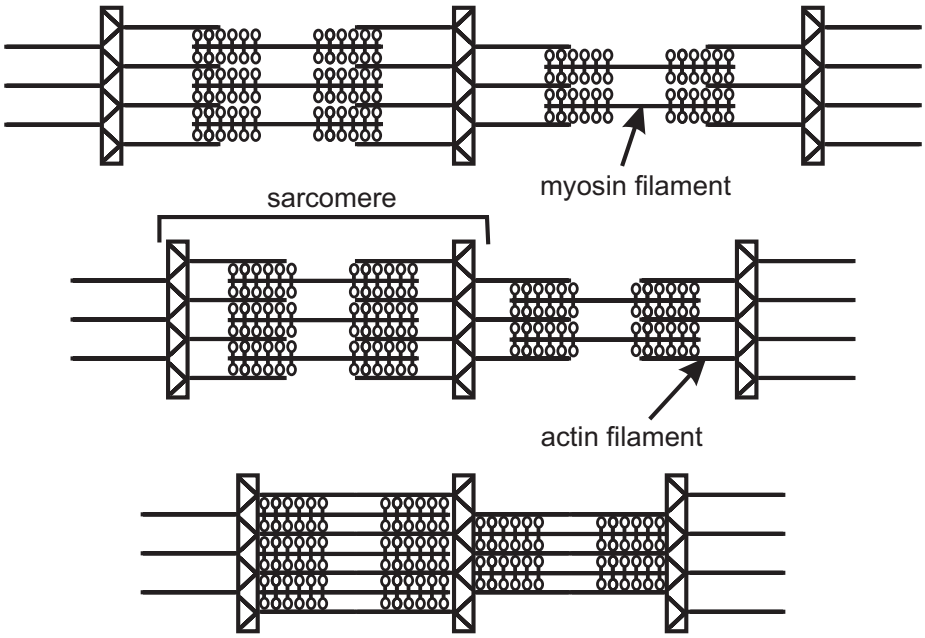


Figure 2.6: Sarcomere in different contraction states (relaxed (top), contracted (middle) and maximally contracted (bottom)).

Figure 2.9. If, on the other hand, the muscle is highly shortened, the myosin heads obstruct each other and thus the resulting active force decreases.

This phenomenological relationship between muscle length and muscle force is modeled by the following equations, where l^M is the length of the muscle, l_0^M its rest length, c_1 and c_2 are parameters for the effect of decrease of forces when expanding resp. shortening the muscle [28, 86], and f_{TL} is the force-length factor (also called tension-length factor) of muscle force. Figure 2.10 gives an example of this numerical relation.

$$f_{TL}\left(l^M\right) = \begin{cases} e^{-\frac{1}{c_1}\left(1-\frac{l^M}{1.1l_0^M}\right)^3} & , l^M \leq 1.1l_0^M \\ \\ e^{-\frac{1}{c_2}\left(\frac{l^M}{1.1l_0^M}-1\right)^3} & , l^M > 1.1l_0^M \end{cases} \tag{2.3}$$

2.2.3.3 Force-velocity-relation

The maximum force a muscle may exert at a certain speed depends on its velocity v^M [23, 62, 111, 118]. It is equal to the muscle maximum isometric force at zero velocity and equal to zero at the maximum contraction velocity. While shortening, the number of myosin heads that actually

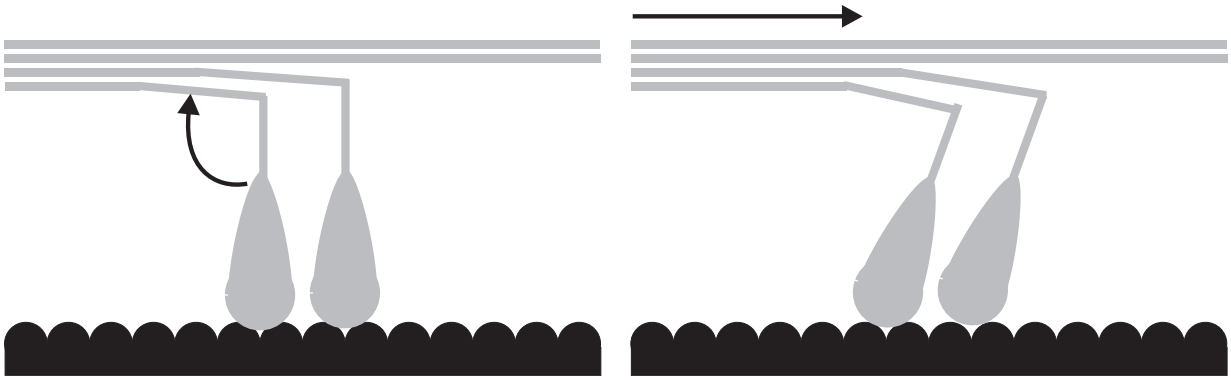


Figure 2.7: The myosin heads may tilt and thus lead to the force exertion.

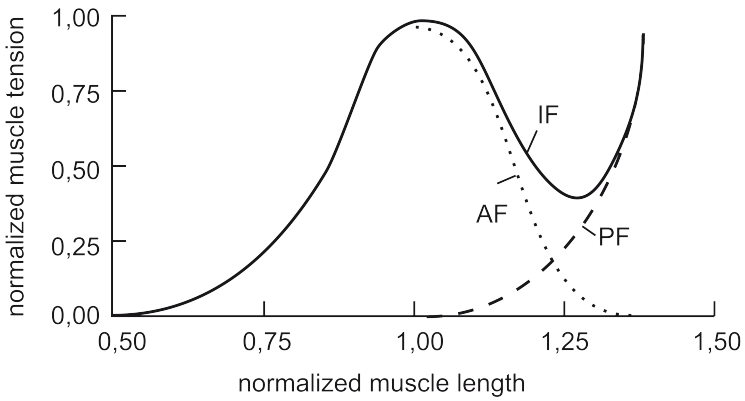


Figure 2.8: The total active force (AF) is received by subtracting the passive force (PF) from the maximum isometric forces (IF).

grip into the actin filament determines the force. When shortening faster, heads have to move back to get grip again, the number of heads that have simultaneously grip is lower and thus the resulting force is lower. When lengthened above the rest length, the force that can be generated is even higher than the isometric force due to effect of the tissue around the muscle.

The overall relation not only depends on the maximum velocity v_{max}^M 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, $c_3 \in [0.1, 0.25]$. c_4 is given by $c_4 = \frac{-0.33}{2}\frac{c_3}{1+c_3}$ [119]. The overall force-

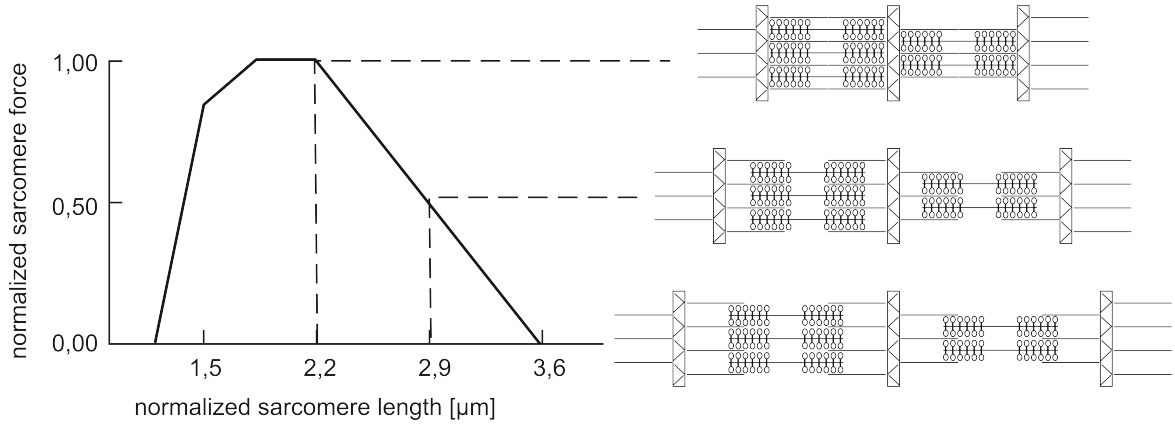


Figure 2.9: The amount of overlapping area of actin and myosin filaments defines the muscle force. If the muscle is shortened below a certain extend, the myosin heads obstruct each other.

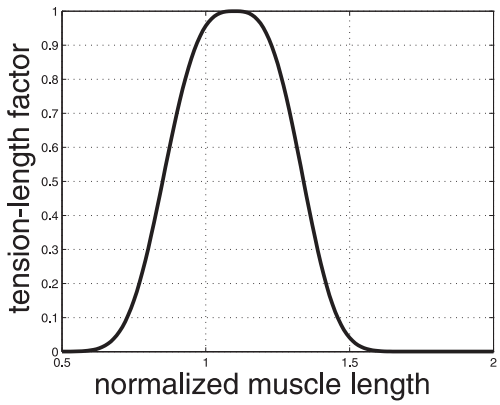


Figure 2.10: Numerical tension-length relation for example parameter values ($c_1 = 0.017$, $c_2 = 0.015$)

velocity relation is given by:

$$f_{FV}\left(v^M\right) = \begin{cases} \frac{1-\frac{v^M}{v^M_{max}}}{1+\frac{v^M}{v^M_{max}c_3}} & , v^M \leq 0 \\ \\ \frac{1-1.33\frac{v^M}{v^M_{max}c_4}}{1-\frac{v^M}{v^M_{max}c_4}} & , v^M > 0. \end{cases} \tag{2.4}$$

Figure 2.11 shows two examples of the force-velocity relation for a fast and a slow muscle.

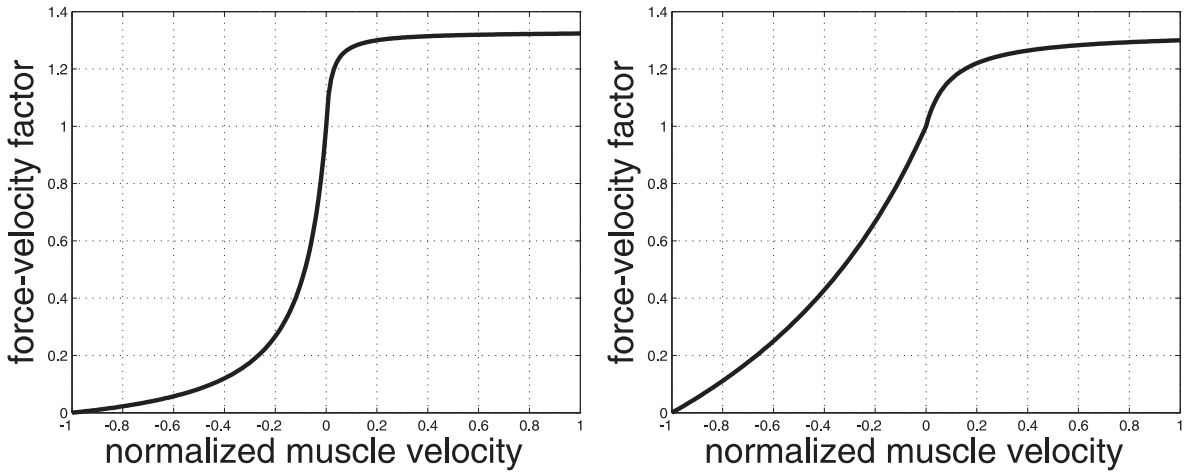


Figure 2.11: Force-velocity relation for a slow muscle ($c_3 = 0.1, c_4 = 0.02$; left) and for a fast muscle ($c_3 = 1.0, c_4 = 0.1$; right).

2.2.3.4 Activation Dynamics

Chemical reaction that lead to force generation take place in the muscle. The myosin heads only move in the presence of Ca^{2+} ions. Those ions are set free in the muscle according to the muscle activation but this process cannot be performed instantaneously. The calcium ion concentration γ lacks behind the muscle activation u both when activation is increased and decreased, which is known as activation dynamics. Activation dynamics is taken into account by the following ordinary differential equation, where b_2, b_3 are muscle specific parameters:

$$\dot{\gamma} = b_2(b_3 u - \gamma). \tag{2.5}$$

How the calcium ion concentration relates to the force exerted is given by the following algebraic equation, where b_1 is a muscle specific parameter and f_{AD} is the activation dynamics force factor:

$$f_{AD}\left(\gamma(u)\right) = \frac{(b_1\gamma(u))^3}{1 + (b_1\gamma(u))^3}. \tag{2.6}$$

Figure 2.12 shows an example for the relationships above.

2.2.3.5 Parallel Elastic and Damping Elements

Besides the actively generated muscle force, passive properties of the muscle must be considered. The common Hill type model uses an elastic element in series with the contractile element. Parallel to the actively force generating muscle elements, a parallel elastic element (PEE) and a

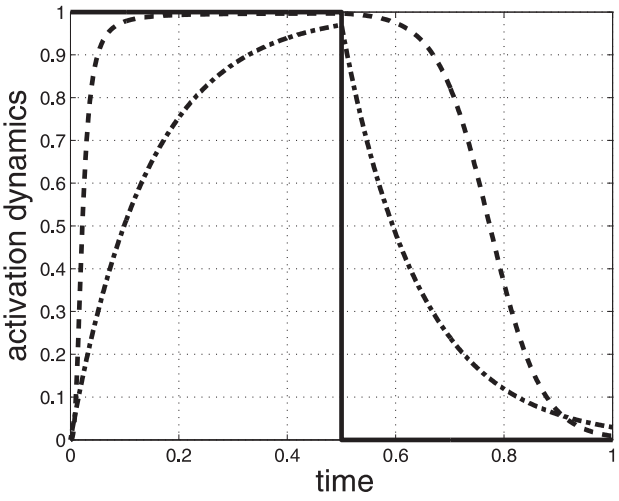


Figure 2.12: Muscle activation dynamics (solid line is u, dotted line is γ, dashed line is f_{AD})

parallel damping element (DE) are considered and modeled by the following equations, where the k_i are muscle specific parameters:

$$F^{PEE}(l^M) = k_1(e^{k_2(l^M-k_3)} - 1) + k_4(e^{k_5(l^M-k_6)} - 1), \tag{2.7}$$

$$F^{DE}(v^M) = k_0 v^M. \tag{2.8}$$

2.2.3.6 Total Muscle Force

With the previous equations the overall force of the muscle tendon unit F^{MTE} is given by

$$F^{MTE}(\gamma, l^M, v^M) = F^{\text{iso}}_{\max} f_{AD}(\gamma) f_{TL}(l^M) f_{FV}(v^M) \tag{2.9}$$

and the overall muscle force F^M is

$$F^M(\gamma, l^M, v^M) = F^{MTE}(\gamma, l^M, v^M) + F^{PEE}(l^M) + F^{DE}(v^M). \tag{2.10}$$

See Figure 2.13 for a schematic overview of the data flow.

2.2.3.7 Muscle Paths

Muscles are not tightly attached to the bones. Thus the direction of force exertion relatively to the bones they are attached to changes with the joint angle, and the muscle lengths and velocities

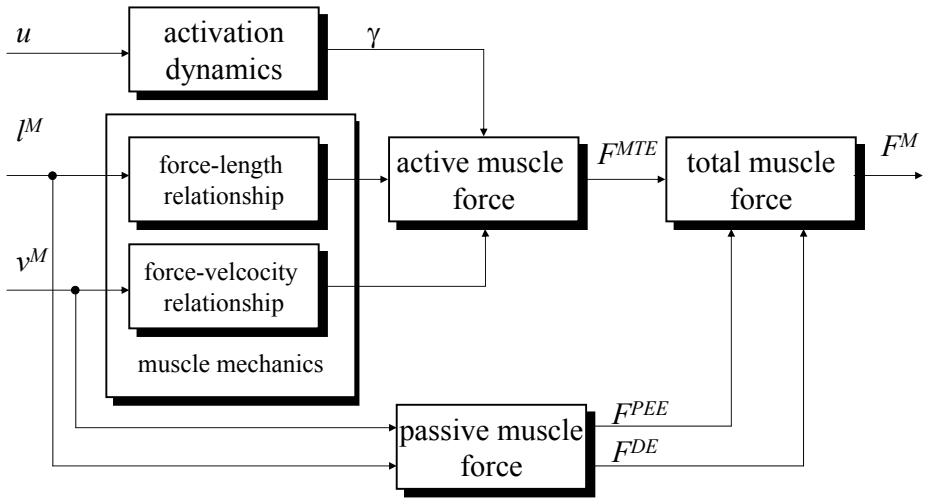


Figure 2.13: Data flow for calculation of the overall force of a muscle tendon unit.

needed for the relations above may be expressed by joint angles and joint angular velocity:

$$l^M = l(q_1, q_2, ...), \tag{2.11}$$

$$v^M = v(q_1, q_2, ..., \dot{q}_1, \dot{q}_2, ...). \tag{2.12}$$

To calculate the torques that result from the linear muscle forces, the muscle paths, i.e., the points and directions of application (or the resulting lever arm directly), have to be modeled. The resulting lever arm 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):

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

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

$$\tau_{j,a} = \sum_{i \in I_j} d_{i,j} F_i^M(\gamma_i, l_i^M, v_i^M), \quad j = 1, \ldots, n_q. \tag{2.13}$$

2.2.3.8 Fatigue

Muscle fatigue occurs when a muscle performs (similar) actions over a long period of time. There exist muscle models that take into account fatigue. The model of Liu [66] for example models fatigue and leads to the percentage of the muscle that can be activated as outcome. Thus,

this model can easily be included into the considerations of this work. It has not been done yet, however, because data acquisition needed for validation was not possible in the context of this work.

2.2.3.9 Passive Torques

In addition to the active torques, passive torques that depend on $\boldsymbol{l}^{\boldsymbol{M}}, \boldsymbol{v}^{\boldsymbol{M}}, \boldsymbol{\gamma}$ (bold letters indicate the vector of all occurring lengths, velocities, calcium ion concentrations), and the joint angles have to be considered [50, 121]. These model passive effects of tendons, ligament and the connective tissue (especially at the boundaries of the feasible joint angle intervals)

$$\tau_{j,p} = \tau_{j,p}(\boldsymbol{l}^{\boldsymbol{M}}, \boldsymbol{v}^{\boldsymbol{M}}, \boldsymbol{\gamma}, \boldsymbol{q}). \tag{2.14}$$

Chapter 3

Multibody Dynamics Algorithms and Object Orientated Computation

To obtain physically valid motion from simulation and optimization, the motion dynamics of the respective robot or human has to be considered. The motion dynamics mathematically can be described by differential equations involving the joint angles, joint velocities and joint accelerations. The numerical solution or optimization of systems described by those systems needs the computation of the joint angle accelerations depending on the joint angles and joint velocities, i.e., the forward dynamics. One common idea is to compute the dynamics recursively in sweeps from the base to the tips of the tree structured systems and vice versa, e.g. the Composite Rigid Body Algorithm CRBA [117], the Recursive Newton-Euler Algorithm RNE [17] or the Articulated Body Algorithm ABA [89]. These algorithms generally lower the computational effort. Which method is best suited for the actual application mainly depends on the size of the system, i.e., the number of (driven) joints [108].

The Articulated Body Algorithm is of linear order $\mathcal{O}(N)$ but has a high coefficient in the effort-DOF relation. It is efficient for systems of 7 or more DOF [108]. Both the number of joints of walking robots and of human biomechanical systems is comparatively high, so that the ABA here is best suited.

3.1 Articulated Body Algorithm ABA

3.1.1 Forward Dynamics Algorithm

The main idea of ABA is to factorize the mass matrix in a way that allows easy calculation of its inverse. The matrices needed for this have special stacked structure which after the analytical investigations allows for fast computation of the forward dynamics using a recursive algorithm in several sweeps.

In more detail, the factorization of the mass matrix as follows is achieved [41]

$$\mathcal{M} = [I + K\Phi H]^T D[I + K\Phi H]. \tag{3.1}$$

Here, D is a block-diagonal matrix and $[I + K\Phi H]$ is left block-triangular. The physical interpretation of the entities in this and the following equation may be found in [51, 60]. The inverse of the mass matrix is then given by [41]

$$\mathcal{M}^{-1} = [I - K\Psi H]D^{-1}[I - K\Psi H]^T \ . \tag{3.2}$$

This leads to the symbolic forward dynamics of tree structured systems without contact. In [41, 89] the special stacked structure of the occurring matrices is exploited to gain a recursive formulation of the forward dynamics:

1. sweep 1 (outboard)
   (a) compute velocities, gravitational acceleration and bias acceleration
2. sweep 2 (inboard)
   (a) compute articulated inertia, spatial forces and intermediate terms
3. sweep 3 (outboard)
   (a) compute intermediate terms and joint acceleration.

3.1.2 Contact Algorithm

Forward dynamics in the case of one or more contacts differs from the case of free motion. This section will shortly summarize the basic idea and the results for extending the ABA to contact situations. A more detailed derivation of the equations may be found in [41].

The basic idea is to calculate and virtually apply additional forces that result in additional joint angle accelerations which finally lead to a preservation of the algebraic contact constraint 2.2. Therefore, the equations of motion of the system without contacts

$$\mathcal{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}_f = B\boldsymbol{\tau} - \mathcal{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \mathcal{G}(\boldsymbol{q}) \tag{3.3}$$

and with contacts

$$\mathcal{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} = B\boldsymbol{\tau} + J_c(\boldsymbol{q})^T \boldsymbol{f_c} - \mathcal{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \mathcal{G}(\boldsymbol{q}) \tag{3.4}$$

are investigated. Note that the joint angles and joint angle velocities do not change due to the contact case, only joint angle accelerations are assumed to be influenced by the presence of contact. Not only the algebraic contact constraint 2.2 have to be fulfilled, but also its first two time derivatives:

$$J_c\dot{\boldsymbol{q}} = 0, \tag{3.5}$$

$$J_c\ddot{\boldsymbol{q}} + \dot{J}_c\dot{\boldsymbol{q}} = 0. \tag{3.6}$$

Combining equations 3.4, 3.3 and 3.6 leads to

$$-J_c(\boldsymbol{q})\mathcal{M}(\boldsymbol{q})^{-1}J_c(\boldsymbol{q})^T\boldsymbol{f_c} = J_c(\boldsymbol{q})\ddot{\boldsymbol{q}}_{\boldsymbol{f}} + \dot{J}_c(\boldsymbol{q})\dot{\boldsymbol{q}}$$

This equation may be solved for $\boldsymbol{f_c}$. Using the operator notation of ABA, there even exists a way to compute $\boldsymbol{f_c}$ without explicitly calculating $\dot{J}_c(\boldsymbol{q})$, see [41] for more details. The contact algorithm then is:

1. calculate unconstrained joint angle accelerations using forward dynamics for system without contacts,
2. calculate unconstrained spatial accelerations at points of contact using inverse dynamics without contacts,
3. calculate the contact forces,
4. calculate the joint angle acceleration update,
5. add the joint angle acceleration update to the unconstrained joint angle accelerations to get the final joint angle accelerations.

This algorithm also may be handled by recursive computations in sweeps from base to tip and vice versa.

3.1.3 Collision Algorithm

Before a system gets into contact with the ground, usually collisions occur. They lead to jumps in the joint angle velocities and a loss of energy of the system.

The calculation of the velocities after a collision follows similar ideas like for the contact algorithm from Section 3.1.2. The main idea here is to calculate and apply an instantaneous impulsive force that occurs due to the collision. Details on the derivation of the algorithm may be found in [41].

This leads to the following collision algorithm:

1. calculate the velocity-jump that occurs at collision,
2. calculate the change of impulse that occurs due to the collision,
3. calculate the resulting change in joint angle velocities,
4. add those changes to the joint angle velocities before collision to get joint velocities after collision.

Again, this algorithm may be expressed by recursive sweeps and thus be integrated into the ABA framework.

3.2 Reduced Dynamics Algorithm

For optimization of multibody systems experiencing contact the differential algebraic system of Equations 2.1 and 2.2 must be considered. The numerical difficulties associated with the system of differential algebraic equations of high index, resulting from the general modeling approach of multibody dynamics and algebraic equations for contact, can be avoided. This is done by a reduced dynamics method from [41], treating as variables of the optimization problem explicitly only the independent states q_I, which are global orientation and position and states related to branches in contact with the ground, and using inverse kinematics to determine the dependent joint angles q_D and the relation $J_{c,I}\dot{q}_I + J_{c,D}\dot{q}_D$ (where $J_{c,\{I,D\}}$ are the constraint Jacobians for the independent resp. dependent joints) to determine the dependent joint angle velocities of the other branches of the tree structured multibody system:

$$\begin{aligned} q_I &:= \text{global orientation, position; free branch(es) states} \\ q_D &:= \text{contact branch(es) states} \end{aligned}$$

q_I may be computed from all states q using a constant mapping Z, i.e. $q_I = Zq$. The solution $\tilde{q}$ of the collision dynamics

$$\ddot{\tilde{q}} = \mathcal{M}\left(\tilde{q}\right)^{-1}\left(Bu - \mathcal{C}\left(\tilde{q}, \dot{\tilde{q}}\right) - \mathcal{G}\left(\tilde{q}\right) + J_c^T f_c\right),$$

then may be proven to give the solution of the initial system of differential algebraic equations [41]. $\ddot{q}_I$ may then be calculated from $\tilde{q}$ by $\ddot{q}_I = Z\ddot{\tilde{q}}$ and be treated as the only states representing the system. The dependent states q_D are obtained by inverse kinematics, which for each branch of the MBS has to be determined. This is not possible analytically for all configurations, however, simple criteria exist to ensure analytical solvability [17].

Besides avoiding DAE and allowing to treat ODE only instead, the size of the model is reduced by the degrees of freedom of the contact branches.

3.3 Object Oriented Approach

In [51] a unifying object-oriented methodology to consolidate multibody dynamics computations in robot control has been presented. The very general and multi-purpose approach was reduced to some of the basic ideas and was extended for application in biomechanical systems as well. An implementation of this modified algorithms has been established for use in the context of this work. The main focus was to get high computational efficiency, easy set up and modification of (sub-)models and easy extension for additional components especially for biomechanical applications like special muscle models, wobbling masses or joint models.

Each part of the robot or the biomechanical system is modeled by a special class that is derived from one base class `DynamicForward`. This gives several component classes like `Base`, `Joint`, `Displacement` (a link with mass), etc. For building tree structured systems, besides simple alignment of components, branchings `Branching` and terminations `Terminator` are necessary. A termination may also be an external force `ExtForce` or a contact `Contact`.

`DynamicsForward` gives the framework for the different sweeps. The different component classes fill the sweeps with the respective computations and contain the component parameters like inertia and mass for a rigid body or the axis of rotation for a joint. Furthermore they have two data ports called *protocol* that take the incoming and give the outgoing variables for the recursive computation. One protocol is shared by the preceding and succeeding components for the shared data, i.e., the outgoing variables of the predecessor are the incoming ones for the successor.

The user only has to define the components of the model, set the parameters and define the topology of the system. The latter is done by adding the components of the model in the order

in which they appear in the system starting from the base. Each branch must be determined by `Terminator`, `ExtForce` or `Contact`. After that, the next parallel branch that started at the same branching can be added.

From this topology of the components an internal pointer structure `Sequence` is established which handles the internal data flow.

For example to model a 5-axes robot the following code could be used:

```
BaseDynamic base;
JointRevoluteDynamic joint1;
LinkDynamic disp1;
JointRevoluteDynamic joint2;
BranchingDynamic branch;
LinkDynamic disp2l;
JointRevoluteDynamic joint3l;
LinkDynamic disp3l;
JointRevoluteDynamic joint4l;
LinkDynamic disp4l;
BranchTermDynamic term_l;
LinkDynamic disp2r;
JointRevoluteDynamic joint3r;
LinkDynamic disp3r;
BranchTermDynamic term_r;

// ... set the parameters for the components

ModelImplDyn model;

model.add(&base);
model.add(&joint1);
model.add(&disp1);
model.add(&joint2);
model.add(&branch);
model.add(&disp2l);
model.add(&joint3l);
model.add(&disp3l);
model.add(&joint4l);
model.add(&disp4l);
```

```
model.add(&term_l);
model.add(&disp2r);
model.add(&joint3r);
model.add(&disp3r);
model.add(&term_r);


SequenceDyn s;
s.setSequence(model);
s.doFwdDynamics();
```

This would lead to the model structure and the internal data shown in Figure 3.1

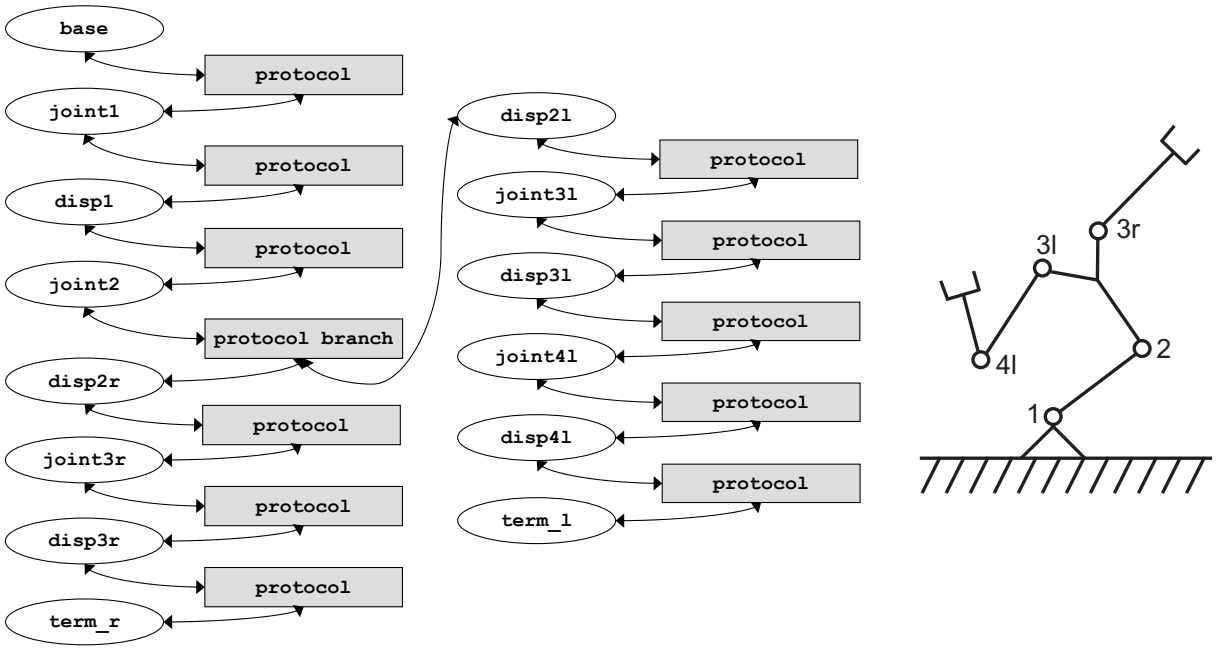


Figure 3.1: Example for structure of object oriented dynamics modeling of a 5-axes robot.

Note that multiple-branchings may be described in different ways. However, the resulting internal data representation and the calculated sweeps are identical even for the different branch descriptions.

Chapter 4

Optimal Control Methods for Problems of Motion Dynamics

4.1 General Optimal Control Problem

To make efficient use of the different redundancies in the robotic or biomechanical systems, model-based optimization approaches can be used to find the best among all possible actuation solutions for different motion tasks. A measure of optimality is described by a certain objective function $\mathcal{J}$ which in a general form may be stated as

$$\mathcal{J} = \varphi(\boldsymbol{x}(t_f), \boldsymbol{p}, t_f) + \int_0^{t_f} L(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{p}, t)dt,$$

i.e. it may consist of a scalar (Mayer term) and of an integral term (Lagrange term) involving the state vector $\boldsymbol{x}$, the control vector $\boldsymbol{u}$, free but constant over time parameters $\boldsymbol{p}$ and the final time t_f.

Differential equations of both first (activation dynamics in biomechanical systems) and second order (multibody system differential equations for robots and biomechanical systems) may occur. The differential equations of second order may be transformed into a system of differential equations of first order but double size by introducing new variables for the first derivatives $\dot{x}$ of the state for every second order state x:

$$\ddot{x} = f(x, \dot{x}, u, t) \text{ is transformed into } \boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} x \\ \dot{x} \end{pmatrix}, \dot{\boldsymbol{y}} = \begin{pmatrix} y_2 \\ f(y_2, y_1, u, t) \end{pmatrix}.$$

This transformation is necessary to apply standard methods to the optimal control problem. An

adapted discretization method for systems of mixed differential equations of first and second order is described in Section 4.3.

Thus, the optimal control problems in the context of this work may be stated in the following form of a general constrained optimal control problem for state differential equations of first order:

$\min_{\boldsymbol{u}} \mathcal{J}$	Minimize the objective function $\mathcal{J}$ consisting of Mayer (scalar) and Lagrange (integral) term
subject to	
$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{p}, t)$	system of ordinary differential equations
$\boldsymbol{r}(\boldsymbol{x}(0), \boldsymbol{x}(t_f), \boldsymbol{p}, t_f) = \boldsymbol{0}$	boundary constraints
$\boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{p}, t) \geq \boldsymbol{0}, 0 \leq t \leq t_f$	nonlinear state and control constraints.

Two basically different classes of methods exist for solving this optimal control problem. Indirect methods formulate optimality conditions to the problem and solve these conditions [115]. This leads to multipoint boundary value problems. These methods give high precision solutions but need a good initial estimate for the solution and high effort is needed to formulate the necessary conditions.

Direct methods on the other hand discretize the optimal control problem and solve the discretized problem. Direct shooting methods [115] discretize the controls only and solve the differential equations of the dynamic system several times to gain sensitivity information that is needed for solving the optimization problem. This leads to high computation times. Direct collocation methods [11, 115] discretize both the states and the controls and then optimize the resulting nonlinear problem. This leads to an implicit and simultaneous solution of the differential equations. Direct methods can be applied directly without formulating any additional conditions to the optimal control problem and do not need initial estimates as good as needed for indirect methods. The results are less precise than from indirect methods but can be used as good initial estimates for indirect methods.

4.2 Direct Collocation

4.2.1 Discretization

There are many different approaches for solving optimal control problems. Here, we consider the computation of optimal trajectories $\boldsymbol{x}^*,\ \boldsymbol{u}^*$ subject to a large and highly nonlinear dynamical system. For this class of problems, so-called direct (transcription) methods have been developed

in recent years showing remarkable performance [11, 115]. Instead of using one of the direct shooting approaches mentioned in Section 1.3 which require feasibility with respect to the ODE constraints in each iteration of the optimization method, a simultaneous approach for solving the ODE integration and optimization problems inherent in the optimal control problem is selected. In direct collocation the implicit integration for a sequence of steps from initial to final time is included as a set of explicit nonlinear equality constraints in the optimization problem. Without the restriction to feasibility to the ODE constraints in each iteration as in direct shooting, only the final solution of direct collocation iteration must satisfy them. Without the restriction of feasible iterates and with much easier computable gradients the solution may be obtained much faster.

The direct collocation method DIRCOL [112, 114] is based on the discretization of both the states and the controls, i.e. $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$ are approximated by $\tilde{\boldsymbol{x}}(t)$ and $\tilde{\boldsymbol{u}}(t)$ on a grid $0 = t_1 < t_1 < \ldots < t_{n_t} = t_f$ (cf. Figure 4.1):

$$\tilde{\boldsymbol{x}}(t) = \sum_l \alpha_l \hat{\boldsymbol{x}}_l(t),\ \tilde{\boldsymbol{x}} \in S^4_\Delta \text{ (cubic)}, \tilde{\boldsymbol{u}}(t) = \sum_j \beta_j \hat{\boldsymbol{u}}_j(t),\ \tilde{\boldsymbol{u}} \in S^2_\Delta \text{ (linear)},$$

where $\hat{\boldsymbol{x}}_l, \hat{\boldsymbol{u}}_j$ are basis functions (e.g. monomials or Hermite basis functions). α_l, β_j are the coefficients of the piecewise polynomial approximation of the states resp. controls and are one option of the variables in the resulting nonlinear constrained optimization problem (NLP). Thus, the resulting large-scale NLP becomes:

$$\boldsymbol{y} = (\alpha_1, \alpha_2, \ldots, \beta_1, \beta_2, \ldots, t_{n_t})^T,\ \min_{\boldsymbol{y}} \phi(\boldsymbol{y}) \text{ s.t. } \boldsymbol{a}(\boldsymbol{y}) = 0,\ \boldsymbol{b}(\boldsymbol{y}) \geq 0,$$

where the equality and inequality constraints $\boldsymbol{a}$ and $\boldsymbol{b}$ are the following:

$\dot{\tilde{\boldsymbol{x}}}(t_\diamond) - f(\tilde{\boldsymbol{x}}(t_\diamond), \tilde{\boldsymbol{u}}(t_\diamond), t_\diamond) = 0, t_\diamond = t_k, t_{k+1/2}, t_{k+1}$	collocation constraints at the grid points and the midpoints of all intervals $[t_i, t_{i+1}]$
$r(\dot{\tilde{\boldsymbol{x}}}(t_1), \dot{\tilde{\boldsymbol{x}}}(t_{n_t}, \boldsymbol{p}, t_{n_t})) = 0$	boundary values
$g_i(\tilde{\boldsymbol{x}}(t_k), \tilde{\boldsymbol{u}}(t_k), \boldsymbol{p}, t_k) \geq 0, i = 1, \ldots, n_g, k = 1, \ldots, n_t$	inequality constraints.

By solving the NLP, the differential equations of motion are solved simultaneously with the optimization problem. This leads to a considerable improvement of efficiency compared to standard methods if all structure and sparsity in the NLP is utilized using a sparse sequential quadratic programming method SNOPT of [32]. The time grid is refined based on local error estimates resulting in a sequence of NLPs with increasing dimensions which are solved successively. DIRCOL brings its own computation of finite differences instead of using the implementation of SNOPT. Adjoint variables, which may be useful for sensitivity analysis of the computed trajectories are computed as well.

The controls are discretized by piecewise linear polynomials, i.e., the following functions $\hat{u}_i(t)$ give a basis for the controls:

$$\hat{u}_i(t) = \begin{cases} \frac{t-t_{i-1}}{t_i-t_{i-1}} & \text{for} \quad t \in [t_{i-1}, t_i), \\ \frac{t_{i+1}-t}{t_{i+1}-t_i} & \text{for} \quad t \in [t_i, t_{i+1}], \\ 0 & \text{otherwise.} \end{cases}$$

The states are discretized by piecewise cubic polynomials. Monomials with local support form a basis for the states:

$$\hat{x}_{l,i} = \begin{cases} \left(\frac{t-t_{i-1}}{t_i-t_{i-1}}\right)^l & \text{for} \quad t \in [t_i, t_{i+1}] \\ 0 & \text{otherwise} \end{cases}$$

The approximation of the controls $\tilde{u}(t)$ and of the states $\tilde{x}(t)$ is thus given by

$$\tilde{\boldsymbol{u}}(t) = \sum_{i=0}^{n_t} \boldsymbol{\alpha}_i \hat{u}_i(t) \tag{4.1}$$

$$\tilde{\boldsymbol{x}}(t) = \sum_{i=0}^{n_t} \sum_{l=0}^{3} \boldsymbol{\beta}_{l,i} \hat{x}_{l,i}(t) \tag{4.2}$$

The vectors of coefficients of the control vector and the state vector $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_{l,i}$ represent the discretized optimal control problem.

The coefficients $\boldsymbol{\alpha}_i$ are given by the values $\boldsymbol{u}_i$ of the controls at the grid points. Continuity of the approximation is given by the definition of the basis functions.

Taking a closer look at those constraints on the grid yields to a special choice of the coefficients $\boldsymbol{\beta}_{l,i}$. For sake of simplicity, $h := t_{i+1} - t_i$ constant is assumed. Then with the values $\boldsymbol{u}_i$ resp. $\boldsymbol{x}_i$ of the controls resp. states at the grid points the following coefficients lead to the fulfillment of the collocation constraints at the grid points and thus only the collocation constraints at the mid points $t_{i+1/2}$ have to be imposed to the nonlinear problem:

$$\begin{aligned} \boldsymbol{\beta}_{0,i} &= \boldsymbol{x}_i, \\ \boldsymbol{\beta}_{1,i} &= h\boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{u}_i, ih), \\ \boldsymbol{\beta}_{2,i} &= -3\boldsymbol{x}_i - 2h\boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{u}_i, ih) + 3\boldsymbol{x}_{i+1} - h\boldsymbol{f}(\boldsymbol{x}_{i+1}, \boldsymbol{u}_{i+1}, (i+1)h), \\ \boldsymbol{\beta}_{3,i} &= 2\boldsymbol{x}_i + h\boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{u}_i, ih) - 2\boldsymbol{x}_{i+1} + h\boldsymbol{f}(\boldsymbol{x}_{i+1}, \boldsymbol{u}_{i+1}, (i+1)h). \end{aligned} \tag{4.3}$$

To define the coefficients, only the vectors $\boldsymbol{x}_i$ has to be stored. Note that by this choice also the number of variables for the nonlinear optimization problem is reduced and continuity of $\tilde{\boldsymbol{x}}$ is

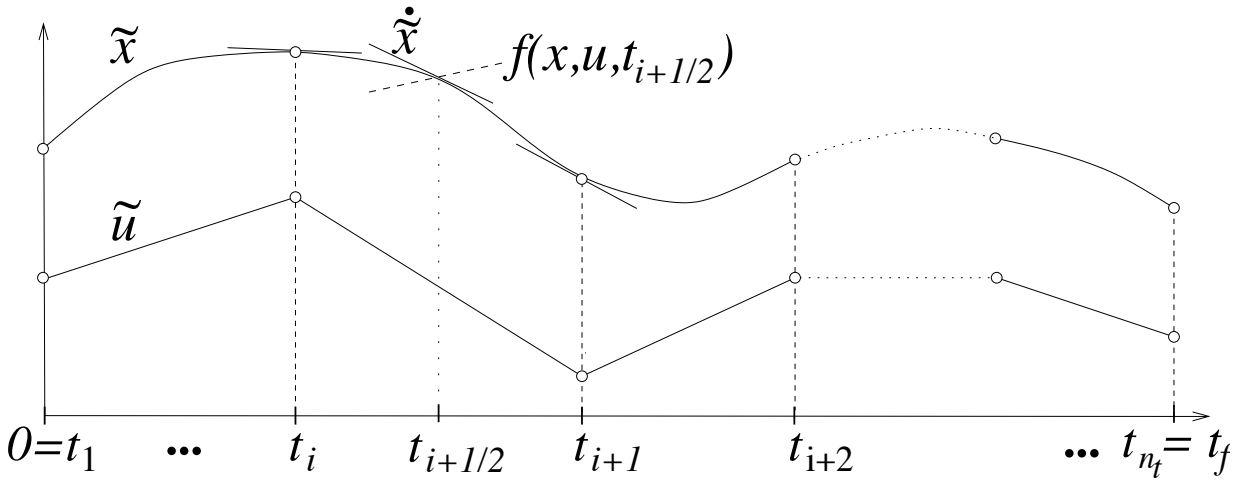


Figure 4.1: Discretization of the controls by piecewise linear polynomials and of the states by piecewise cubic polynomials; collocation constraints at the grid and at the mid points.

ensured simultaneously.

In summary, the optimization variables for the discretized problem that leads to a nonlinear, constrained optimization problem are the values of the controls at the grid points and the values of the states at the grid points.

A special case of nonlinear constraints are box constraints, where the controls or states are bounded by a fixed lower or upper bound. In the discretized problem, those constraints may directly be imposed to the optimization variables. In the case of explicit boundary constraints for states or controls, the respective state or control at the initial or final grid point does not have to be constrained but is simply removed from the set of optimization variables and the boundary value is used directly instead, which reduces the problem size.

4.2.2 Sequential Quadratic Programming

The discretization leads to a nonlinearly constrained optimization problem with optimization variables $\boldsymbol{y}$ that represent the coefficients of the piecewise polynomials of the controls and states. By the special choice of the discretization of the previous subsection this can be done by simply the values of the controls and states at the grid points:

$$\boldsymbol{y} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n_t}, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{n_t}).$$

For explicit boundary conditions imposed to controls or states, the respective variable may be set to a fixed value and excluded from $\boldsymbol{y}$.

The optimization problem then reads as

$$\min_{\boldsymbol{y}} \tilde{\mathcal{J}}(\boldsymbol{y}),\ \text{s.t.}\boldsymbol{a}(\boldsymbol{y}) = 0, \boldsymbol{b}(\boldsymbol{y}) \geq 0,$$

where $\tilde{\mathcal{J}}$ is the discretized objective function, $\boldsymbol{a}$ comprises the collocation constraints and the discretized equality and implicit boundary constraints, and $\boldsymbol{b}$ consists of the discretized inequality constraints.

This optimization problem can be solved using a sequential quadratic programming method (SQP), in the case of DIRCOL, SNOPT [32] is used. Details shall not be explained here, because in the context of this thesis, no modification to the SQP method had to be made.

4.3 Adaption to Systems of Second Order

4.3.1 Preliminaries

In [15, 81, 113], direct collocation approaches with special treatment of differential equations of motion of second order are given. The approach from [81] was used for the $n_{x,2}$ differential equations of second order and states of second order in the problems described above to avoid the transformation into differential equations of first order and the resulting increase of the problem size. This section shortly summarizes the approach and basically follows the structure from [81].

Divided Differences. Let $f \in C[a,b]$ a continuous function on the interval $[a,b]$. Then the recursive definition on the grid Ω

$$f[t_i] := f(t_i),\ 0 \leq i \leq n \tag{4.4}$$

$$f[t_i, \ldots, t_{i+j}] := \frac{f[t_i+1, \ldots, t_{i+j}] - f[t_i, \ldots, t_{i+j-1}]}{t_{i+j} - t_i},\ 0 \leq j \leq n, \tag{4.5}$$

defines the so called divided differences $f[.]$ of the function f.

B-Splines. Let $k \in \mathbb{N},\ t \in [a,b]$, the grid Ω and the functions $g_k(t;s)$ be given. Then the functions

$$M_{i,k} := g_k[t_i, \ldots, t_{i+k}; t], \tag{4.6}$$

$$M_{i,j} := 0,\ \text{for } j = 0 \text{ or } -j \in \mathbb{N} \tag{4.7}$$

are called *B-Splines of order* k and defined to be continuous from the right side at the grid points.

$$N_{i,k} := (t_{i+k} - t_i) M_{i,k}$$

are called *normalized B-Splines of order* k. They can be evaluated with the following formula [20], which is especially well suited for higher order B-Splines, because evaluation of higher order powers is avoided:

$$M_{i,k} = \frac{t - t_i}{t_{k+i} - t_i} M_{i,k-1} + \frac{t_{i+k} - t}{t_{k+i} - t_i} M_{i+1,k-1}.$$

For $k = 1$ the following holds:

$$M_{i,1} = \frac{(t_{i+1} - t)_+^0) - (t_i - t)_+^0}{t_{i+1} - t_i} \tag{4.8}$$

$$= \begin{cases} (t_{i+1} - t_i)^{-1} > 0 & \text{if } t \in (t_i, t_{i+1}) \\ 0 & \text{otherwise} \end{cases}, \tag{4.9}$$

i.e., $\mathrm{supp}(M_{i,k}) = (t_i, t_{i+k})$. The $M_{i,k}$ resp. $N_{i,k}$ form a basis of the vector space $S_m(t_0, \ldots, t_n)$ of splines of maximum degree m over the grid points $t_0, \ldots, t_n$, i.e. for a $s \in S_m(t_0, \ldots, t_n)$ the following holds:

$$s(t) = \sum_i \alpha_{i+m-1} N_{i,k}(t).$$

In the case of equidistant grid points, the boundary intervals may be treated by including $m - 1$ virtual grid points at each boundary. Thus, each $s \in S_m(t_0, \ldots, t_n)$ has the following form:

$$s(t) = \sum_{i=-m+1}^{n-1} \alpha_{i+m-1} N_{i,k}(t).$$

4.3.2 Discretization of the Problem

For evaluating the approach of discretizing the states of second order directly, cubic splines ($m = 4$) are chosen to avoid problems with oscillations. An equidistant grid is used to keep implementation comparatively easy due to the simple way to handle boundary intervals. The basis functions are the same for each interval. With $\delta := t_{i+1} - t_i$, $\mu_j(t) := (t - t_j)/\delta$, and

$I_j := (t_j, t_{j+1}]$, the functions $N_{i,4}$ may be computed:

$$N_{i,4}(t) = \begin{cases} b_0(t;t_i) := \mu_i(t)^3/6, & t \in I_i, \\ b_1(t;t_{i+1}) := (1 + 3\mu_{i+1}(t) + 3\mu_{i+1}(t)^2 - 3\mu_{i+1}(t)^3)/6 & t \in I_{i+1}, \\ b_2(t;t_{i+2}) := (4 - 6\mu_{i+2}(t)^2 + 3\mu_{i+2}(t)^3)/6 & t \in I_{i+2}, \\ b_3(t;t_{i+3}) := (1 - 3\mu_{i+3}(t) + 3\mu_{i+3}(t)^2 - \mu_{i+3}(t)^3)/6 & t \in I_{i+3}, \end{cases} \tag{4.10}$$

Furthermore, $b_j(t;t_i) := 0$ for $t \notin I_i,\ j = 0, 1, 2, 3$. $s(t)$ now may be stated as

$$s(t) = \sum_{i=0}^{n+2} \alpha_i \sum_{j=0}^{3} b_j(t; t_{i+j}).$$

On the additional virtual intervals, there is no contribution of the basis functions. Thus, resorting of the sum leads to:

$$s(t) = \sum_{i=0}^{n-1} \sum_{j=0}^{3} \alpha_{i+j} b_j(t; t_i).$$

$\mathrm{supp}\,(b_j(t;t_i)) = I_i$, and thus the first sum may be eliminated:

$$s(t) = \sum_{j=0}^{3} \alpha_{i+j} b_j(t; t_i), \text{ if } t \in I_i.$$

At the grid points, this gives

$$s(t_i) = \frac{1}{6}\alpha_i + \frac{2}{3}\alpha_{i+1} + \frac{1}{6}\alpha_{i+2}. \tag{4.11}$$

The derivatives of the b_j are given by

$$\begin{array}{lll} \dot{b}_0(t;t_i) & := \mu_i(t)^2/(2\delta), & t \in I_i, \\ \dot{b}_1(t;t_{i+1}) & := (1 + 2\mu_{i+1}(t) - 3\mu_{i+1}(t)^2)/(2\delta), & t \in I_{i+1}, \\ \dot{b}_2(t;t_{i+2}) & := (-4\mu_{i+2}(t) + 3\mu_{i+2}(t)^2)/(2\delta), & t \in I_{i+2}, \\ \dot{b}_3(t;t_{i+3}) & := (-1 + 2\mu_{i+3}(t) - \mu_{i+3}(t)^2)/(2\delta), & t \in I_{i+3}, \end{array}$$

and

$$\begin{array}{lll} \ddot{b}_0(t;t_i) & := \mu_i(t)/\delta^2, & t \in I_i, \\ \ddot{b}_1(t;t_{i+1}) & := (1 - 3\mu_{i+1}(t))/\delta^2, & t \in I_{i+1}, \\ \ddot{b}_2(t;t_{i+2}) & := (-2 + 3\mu_{i+2}(t))/\delta^2, & t \in I_{i+2}, \\ \ddot{b}_3(t;t_{i+3}) & := (1 - \mu_{i+3(t)})/\delta^2, & t \in I_{i+3}. \end{array}$$

Thus,

$$\begin{aligned}\dot{s}(t) &= \sum_{j=0}^{3} \alpha_{i+j}\dot{b}_j(t;t_i), \quad \text{if } t \in I_i,\\ \ddot{s}(t) &= \sum_{j=0}^{3} \alpha_{i+j}\ddot{b}_j(t;t_i), \quad \text{if } t \in I_i,\end{aligned}$$

and on the grid points

$$\begin{aligned}\dot{s}(t) &= \tfrac{1}{2\delta}\alpha_i + \tfrac{-1}{2\delta}\alpha_{i+2},\\ \ddot{s}(t) &= \tfrac{1}{\delta^2}\alpha_i + \tfrac{-2}{\delta^2}\alpha_{i+1} + \tfrac{1}{\delta^2}\alpha_{i+2}.\end{aligned} \tag{4.12}$$

4.3.3 Application to the Optimal Control Problem

The states related to the differential equations of second order shall now be approximated by spline functions $S_4(t_0,\ldots,t_n)$. For this purpose, the approximating functions are defined as follows:

$$\begin{aligned}\tilde{x}_k(t) &:= \sum_{i=0}^{n-1}\sum_{j=0}^{3} \alpha_{i+j,k} b_j(t;t_i),\\ \dot{\tilde{x}}_k(t) &:= \sum_{i=0}^{n-1}\sum_{j=0}^{3} \alpha_{i+j,k}\dot{b}_j(t;t_i),\\ \ddot{\tilde{x}}_k(t) &:= \sum_{i=0}^{n-1}\sum_{j=0}^{3} \alpha_{i+j}\ddot{b}_j(t;t_i).\end{aligned}$$

$k = 1,\ldots,n_{x,2}$. Thus, discretization of the $n_{x,2}$ states of second order of the optimal control problem leads to the vector of $n_{x,2}(n+3)$ parameters for the nonlinear optimization problem. x_k is represented by

$$\boldsymbol{\alpha}_k = (\alpha_{k,0}, \alpha_{k,1}, \ldots, \alpha_{k,n+2})^T$$

In the case of differential equations of mixed first and second order, both parameters $\boldsymbol{\alpha}$ for the states of second order and parameters $\boldsymbol{x}_l$ for the states of first order and $\boldsymbol{u}_m$ for the controls must be taken into account. In the case where the controls may be computed from the states, the $\boldsymbol{u}_m$ may be eliminated. Because this is not the case for forward dynamics biomechanics formulation, this shall not be investigated more closely.

For boundary conditions, nonlinear inequality or nonlinear equality constraints and the collocation constraints, Equations 4.11 and 4.12 may be used to efficiently evaluate the approximation. Note that box constraints, that may be directly implied to the coefficients $\boldsymbol{x}_l$ in the case of first order states now have to be considered as nonlinear inequality constraints.

4.4 Identification of Objective Functions for Muscle Control

4.4.1 Problem Statement

To calculate free goal oriented motion (like kicking as fast as possible or jumping as high as possible) suitable objective functions are needed. They are determined by the goal to reach.

The objective function of muscle involvement on the other hand is chosen (unconsciously) by the human executing the motion. If a measured motion shall be analyzed, i.e. if for a given human motion the muscle activations that lead to that specific motion shall be determined, a objective function for muscle involvement must be chosen. Several heuristic objective functions may be found in literature, see Section 1.3. It is however still an open question which of those objective functions really applies for certain human motions. State of the art for justification of those objective functions is to compare computed data with measurements. This gives an idea that the objective functions are somehow reasonable but do not clarify the real objective functions.

We here present an approach to numerically identify the objective function applied to a human motion. We start with n_h hypotheses $\mathcal{J}_i$ of possible objective functions. The overall objective function $\mathcal{J}$ shall be statable as a weighted sum of the hypothesis:

$$\mathcal{J} = \sum_{1}^{n_h} \omega_i \mathcal{J}_i,$$

where ω_i are the weights. Without loss of generality, $0 \leq \omega_i \leq 1 \forall i$. The problem of identifying the objective functions thus is transformed into the problem of finding the weights ω_i.

The goal is to identify the parameters just from the motion data, i.e., the joint angles and joint angle velocities, in a way that the measured motion agrees best possible with the motion that may be calculated under the assumption of the respective objective function.

This leads to the following minimization problem:

$$\min_{\boldsymbol{\omega}} \mathcal{I} := \min_{\boldsymbol{\omega}} \sum_{i=1}^{n_j} \int_0^{t_f} \left(x_i(\boldsymbol{\omega}; t) - \varphi_i(t)\right)^2 dt, \text{ s.t. } \mathbf{0} \leq \boldsymbol{\omega} \leq \mathbf{1}, \text{ where} \tag{4.13}$$

$\boldsymbol{\omega}$	is the vector of weight parameters ω_i
$x_i(\boldsymbol{\omega}; \cdot)$	is the calculated motion data of the i-th joint with parameters $\boldsymbol{\omega}$ in the objective
φ_i	is the measured motion data for the i-th joint
n_j	is the number of joints taken into account for the investigation.

4.4.2 Sensitivity Analysis of the Discretized Optimal Control Problem

To obtain an efficient identification method, derivative information is needed. The approach of computing the sensitivities from the optimization problem that results from discretizing the optimal control problem is described in [12]. In [14], it was investigated as one of two possible ways to identify the basic behavior of car drivers. It is based on calculating the sensitivities for the optimization problem gained from discretizing the optimal control problem. Starting from the general optimal control problem from Section 4.2.2, i.e.

$$min_{y \in \mathbb{R}^{n_y}} \tilde{\mathcal{J}}(\mathbf{y}, \omega)$$

subject to

$$\begin{aligned} \mathbf{a}(\mathbf{y}, \omega) &= \mathbf{0}, \\ \mathbf{b}(\mathbf{y}, \omega) &\geq \mathbf{0}, \end{aligned}$$

with the vector of variables $\mathbf{y}$ containing the coefficients of the piecewise polynomials of the discretization, $\mathbf{a}$ representing the discretized collocation, boundary and nonlinear equality constraints and $\mathbf{b}$ representing the discretized nonlinear inequality and box constraints.

For a solution $\mathbf{y}^\star$ of the optimization problem, the active constraints can be collected in a vector function $\bar{\boldsymbol{a}}$:

$$\bar{\mathbf{a}}(\mathbf{y}^\star, \omega) = \mathbf{0}.$$

Thus, the Lagrange function has the following form:

$$L(\mathbf{y}^\star, \mu^\star, (\omega)) = \tilde{\mathcal{J}}(\mathbf{y}^\star, \omega) - \mu^{\star T} \bar{\boldsymbol{a}}(\mathbf{y}^\star, \omega),$$

where $\mu^\star$ are the corresponding Lagrangian multipliers.

If $\tilde{\mathcal{J}}, \mathbf{b}, \bar{\mathbf{a}}$ are differentiable twice, some optimality condition of second order and the gradients of the active constraints are linearly independent (which shall not be investigated here, see [12] for details), then the solution $(\mathbf{y}^\star, \omega^\star)$ of the optimization problem is locally differentiable w.r.t. ω and for $i = 1, \ldots, n_\omega$ the sensitivities can be computed from

$$\begin{pmatrix} \frac{\partial \mathbf{y}}{\partial \omega}(\omega) \\ \frac{\partial \mu}{\partial \omega}(\omega) \end{pmatrix} = - \begin{pmatrix} \nabla_{\mathbf{yy}} L(\mathbf{y}^\star, \mu^\star, (\omega)) & \nabla_{\mathbf{y}} \bar{a}(\mathbf{y}^\star, \omega)^T \\ \nabla_{\mathbf{y}} \bar{a}(\mathbf{y}^\star, \omega) & 0 \end{pmatrix}^{-1} \begin{pmatrix} \nabla_{\mathbf{y}\omega} L(\mathbf{y}^\star, \mu^\star, \omega) \\ \nabla_{\omega} \bar{A}(\mathbf{y}^\star, \omega) \end{pmatrix}.$$

The solution vector $\frac{\partial \mathbf{y}}{\partial \omega}$ can be used to compute the sensitivities the same way the states are

computed. Recognizing the structure of the solution vector

$$\frac{\partial \mathbf{y}}{\partial \omega} := (\mathbf{d_i x_0}, \ldots, \mathbf{d_i x_{n_t}}, \mathbf{d_i u_0}, \ldots, \mathbf{d_i u_{n_t}}),\ i = 1, \ldots, n_\omega,$$

the sensitivities $\partial_i \mathbf{x} := \partial \mathbf{x}/\partial \omega_i,\ \partial_i \mathbf{u} := \partial \mathbf{u}/\partial p_i,\ i = 1, \ldots, n_\omega$ are given by

$$\partial_i \tilde{\mathbf{x}}(t) = \begin{cases} \sum_{s=0}^{3} \mathbf{d_i c_j^s} \left(\frac{t-t_j}{h_j}\right)^s, & t_j \le t < t_{j+1}, j = 0, \ldots, n_t - 2, \\ \sum_{s=0}^{3} \mathbf{d_i c_{N-1}^s} \left(\frac{t-t_{N-1}}{h_{N-1}}\right)^s, & t_{N-1} \le t \le t_N, \end{cases}$$

and

$$\partial_i \tilde{\mathbf{u}}(t) = \begin{cases} \mathbf{d_i u_j} + \frac{t-t_j}{h_j}(\mathbf{d_i u_{j+1}} - \mathbf{d_i u_j}), & t_j \le t < t_{j+1}, j = 0, \ldots, n_t - 2, \\ \mathbf{d_i u_{N-1}} + \frac{t-t_{N-1}}{h_{N-1}}(\mathbf{d_i u_N} - \mathbf{d_i u_{N-1}}), & t_{N-1} \le t \le t_N \end{cases}$$

where for the definition of the $\mathbf{d_i c_j^s}$ the variables $\mathbf{d_i x_j}$ are used instead of $\mathbf{x_j}$ in Equations 4.3.

4.4.3 Identification Methods

The problem statement in Section 4.4.1 leads to an optimization problem. Two different approaches for solving this problem are given in the following two subsections.

4.4.3.1 Derivative-free Approach

One approach to optimization when derivatives are not available or difficult or costly to compute is black box optimization. Black box optimizers solve the optimization problem with evaluations of the objective function only. However, they have the drawback that usually more evaluations of the objective function are needed than with approaches making benefit of derivative information.

So in our case, where derivatives are available in principle, black box optimizers should be used only to get a first idea of how to handle the problem. For the real application, more sophisticated methods that use derivatives should be used.

Details on the black box method that has been used for first results of Section 5.3 may be found in [46, 47].

4.4.3.2 Approach using Derivatives

Several approaches to gain benefit from evaluating not only the objective function $\mathcal{I}$ but also the derivatives with respect to the parameters $d\mathcal{I}/d\boldsymbol{\omega}$ exist. The derivatives involve the sensitivities of the (discretized) optimal control:

$$\begin{aligned}
\frac{\partial \mathcal{I}(\boldsymbol{\omega})}{\partial \omega_i} &= \frac{\partial}{\partial \omega_i} \sum_{i=1}^{n_j} \int_0^{t_f} \left(x_i(\boldsymbol{\omega}; t) - \varphi_i(t)\right)^2 dt \\
&= \sum_{i=1}^{n_j} \int_0^{t_f} \frac{\partial}{\partial \omega_i} \left(\left(x_i(\boldsymbol{\omega}; t) - \varphi_i(t)\right)^2\right) dt \\
&= 2 \sum_{i=1}^{n_j} \int_0^{t_f} \left(x_i(\boldsymbol{\omega}; t) - \varphi_i(t)\right) \frac{\partial x_i(\boldsymbol{\omega}; t)}{\partial \omega_i} dt \\
&= 2 \int_0^{t_f} \left(\boldsymbol{x}(\boldsymbol{\omega}; t) - \boldsymbol{\varphi}(t)\right)^T \partial_i \mathbf{x}\, dt
\end{aligned}$$

The special structure of the identification optimization problem, i.e., the sum (or sum of integrals) of squares of differences that are to be minimized, is exploited by special methods. One widely used approach is the Levenberg-Marquardt approach. Details on this method may be found in [75].

Chapter 5

Problems, Models and Results

5.1 Walking Robots

Walking motions have been optimized both for a quadruped and a biped robot. The optimal control problems for the different robots share some characteristics. Whenever possible, only a half stride of the walking motion should be optimized. This introduces additional boundary constraints but ensures that the resulting walking motion is exactly symmetric and reduces the problem size by one half. The details on how the boundaries have to be chosen are given in the respective sections.

Stiff contact between the feet and the ground are assumed as an initial setup of the model. The collision algorithm from Section 3.1.3 has been used to handle the landing of the foot or feet, the contact algorithm (Section 3.1.2) in combination with the reduced dynamics algorithm (Section 3.2) was used during the stance phases. The parameters of the contact model were iteratively adjusted.

Finding an optimal and stable walking trajectory was an iterative process even while using a computational model. The optimized trajectories were implemented to the real robot and the model (e.g. parameters of the motors, ground contact model) and optimal control problem (e.g. constraints on motor parameters) were modified according to the observed experimental results. The projected center of mass (see Section 2.1.4) was used to ensure stability of the optimized walking motions. No control besides joint angle PID control from the servo motors was used to gain stability in the experiments which allowed judgment of the quality of the stability of the optimized motions.

The motions were optimized for time or energy. The resulting joint angle trajectories were used for control of the real robot. The motor currents were not used as control because even more

detailed motor models would have been necessary for this.

The following three subsections summarize work that has been presented at the International Conference on Climbing and Walking Robots CLAWAR 2003 ([107], Section 5.1.1, joint work with M. Hardt and O. von Stryk), the IEEE International Conference on Humanoid Robots ([13], Section 5.1.2, joint work with M. Buss, M. Hardt, J. Kiener, M. Sobotka, O. von Stryk, and D. Wollherr), and CLAWAR 2006 ([93], Section 5.1.3, joint work with A. Seyfarth, R. Tausch, F. Iida, A. Karguth and O. von Stryk) and been included into the respective conference proceedings. Modeling and optimization in those publications was the main topic of the author of this thesis.

In the following chapters, the x-axes is chosen to point into the walking direction of the robot, the z-axes points upwards and the y-axes is chosen to gain a right-handed coordinate system.

5.1.1 Gait Optimization for a Quadruped Robot

5.1.1.1 Model of Sony's Four-legged Robot AIBO ERS-210(A)

Originally designed as a toy and entertainment robot, Sony's AIBO ERS-210(A) (see Figure 5.1) was also involved into the worldwide RoboCup competitions for teams of autonomous soccer playing robots [3] between 2001 and 2004. For this application fast and stable gaits are mandatory. The robot contains an onboard CPU, which in autonomous soccer competitions mainly must be used for image processing using the integrated CCD-camera and image understanding (localization of the robot itself, of team mates, opponents, goals and the ball). For RoboCup, a software architecture based on Sony's real-time operating system AperiOS and Sony's OPEN-R-library has been developed [18], which gives a comfortable way of implementing off-line computed joint trajectories in the robot and receiving the resulting sensor data for joint angles during execution.

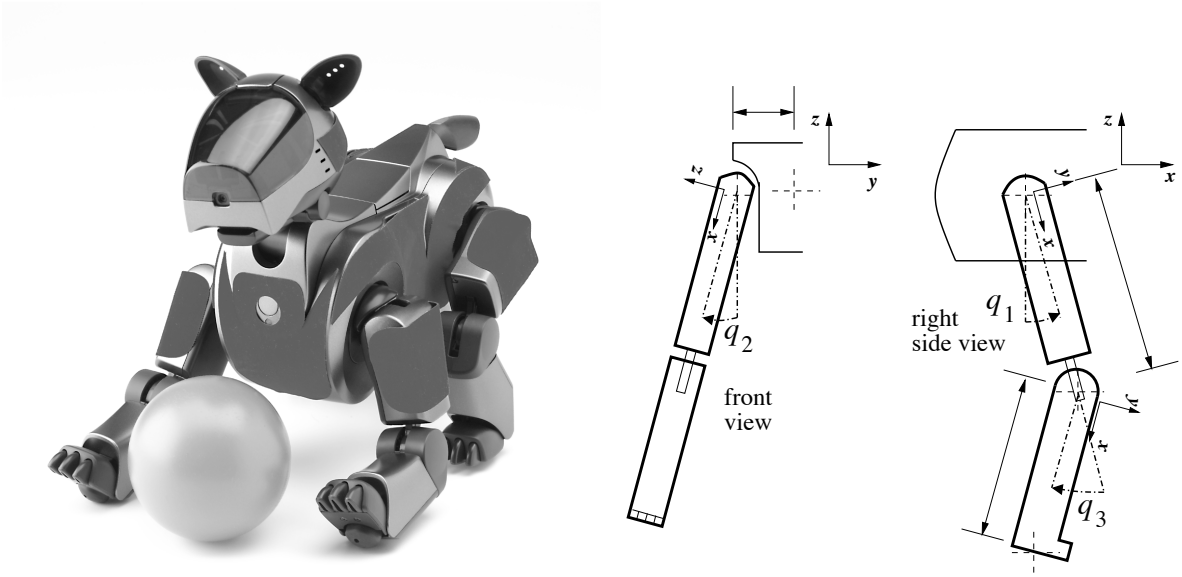


Figure 5.1: The four-legged Sony robot (left) and the kinematical structure of one leg (right); taken from [107].

In the context of this work, the model of Sony's four-legged robot consists of a 9-link tree-structured multibody system (MBS) with a central torso attached to a relatively heavy head at a fixed position and four two-link legs. Each leg contains a 2 DOF universal joint in the hip and a 1 DOF rotational joint in the knee. A minimum set of coordinates consists of 18 position and 18 velocity states $(\mathbf{q}(t), \dot{\mathbf{q}}(t))$ which include a three-parameter Euler angle vector for the orientation, a three-dimensional global position vector, and their time derivatives for the torso, and additionally three angles and their velocities for each leg. The 12 control variables $\mathbf{u}(t)$ correspond to the applied torques in the legs. The required kinematical and kinetic data for each link (length, mass, center of mass, moments of inertia) have been provided by Sony.

5.1.1.2 Motor Characteristics Model

Motor restrictions are of essential importance when calculating optimal gait trajectories. As no further details are available for this robot, motor characteristics have to be estimated. As a first step, maximum angular velocities $q_{i,\max}$ and maximum torques $u_{i,\max}$ of each of the joints have been estimated by an iterative comparison of calculated trajectories and sensor data for each joint: Estimates of the maximum angular velocities and maximum torques used as constraints in the optimal control problem for computing reference trajectories for a dynamic gait have been reduced successively until the observed error between calculated joint angle trajectories and measured joint angle trajectories becomes small. In the experiments set-point trajectory tracking control in each joint was used as provided by the manufacturer of the robot.

This procedure ensures that the calculated trajectories can be implemented on the real robot. However, the restrictions obtained in this manner are likely to be too restrictive. A more detailed model of the joint actuators will most likely lead to better gait trajectories in simulation and experiment.

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 can be replaced by the reduced dynamics equations like described in Section 3.2. The special treatment of states related to differential equations of second order, see Section 4.3, by which the problem size could have been further decreased, had not yet been established when the investigations were taken.

Useful objective functions are, for example, time t_f, energy $\int \sum_{i=1}^{m} u_i^2$, or combinations of both [42]. Boundary conditions contain conditions for

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

Nonlinear inequality constraints are:

- Hips of legs in contact with the ground must stay within a maximum radius of the leg, so that the inverse kinematics solution required for reduced dynamics has a well-defined solution.
- The swing feet must move above a certain curve above ground, for example a sine curve. This property increases stability by avoiding contact with the ground resulting from deflexions of bodies and joints, which could lead to stumbling of the robot.
- Slipping is avoided by limiting the horizontal contact forces relative to the vertical contact forces.
- Vertical contact forces must be positive, i.e. the robot may only push to ground but may not pull from ground.
- Further constraints to be considered in the problem formulation are detailed motor characteristics. By now the box constraints for minimal and maximal values of angular velocities and torques only give a rough estimate of the real actuator data.

Note that stability is not enforced explicitly but may be checked by one of the criteria given in Section 2.1.4 and [42]. More details on each of the constraints may be found in Section 5.1.2, where the constraints are stated for a humanoid robot.

5.1.1.3 Results

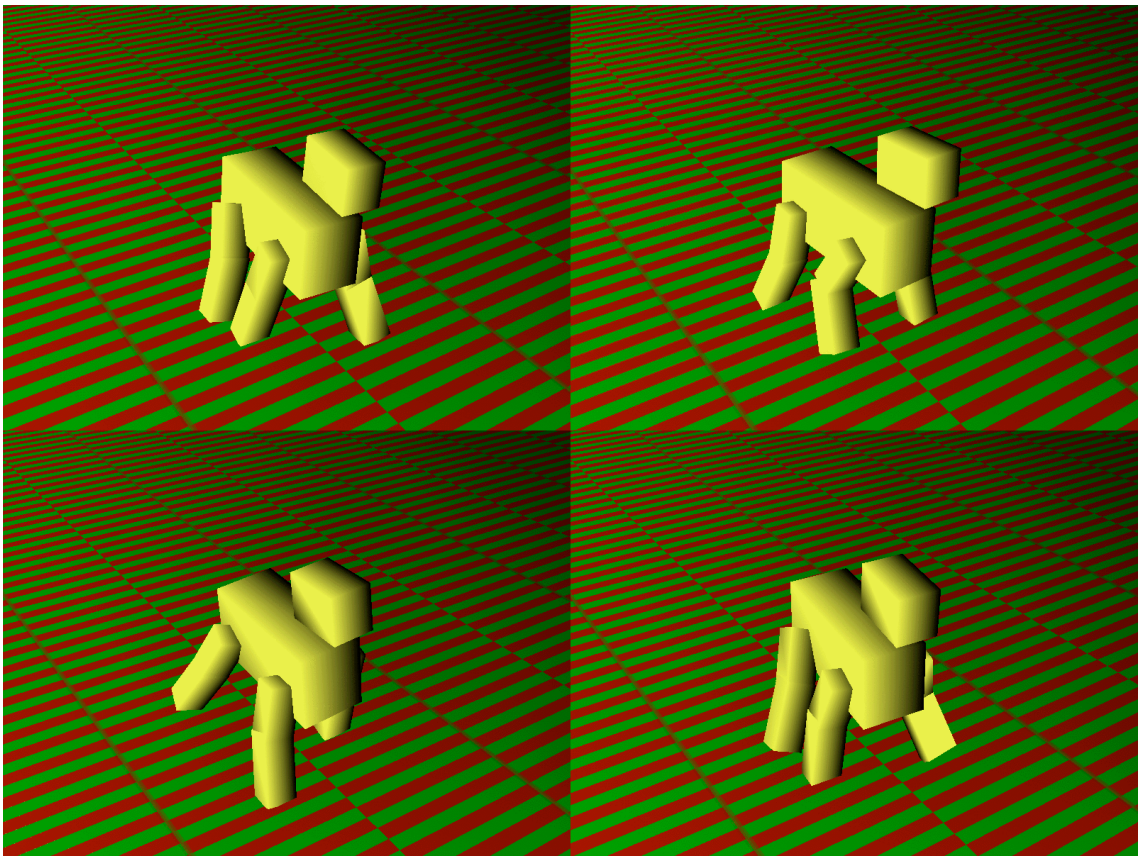


Figure 5.2: Four scenes from an animation of a computed trot gait; taken from [107].

Numerical Results. In this section, the numerical results for generating an optimal trajectory for a trot gait are given. Note that although the states of the optimal control problem include velocities and orientation of the main body, those states are not used for implementation. For the calculations, of course, these states are necessary. The interesting part of the solution is the set of twelve trajectories, one for each of the robot's leg joints. The solution may be visualized not only by plotting the trajectories (as in Figure 5.3, where the calculated trajectories are plotted in comparison to the sensor data received when implementing the trajectory), but also by animating the robot for each of the calculated states. In Figure 5.2 four single images of a computed trot are shown, each for a different contact situation.

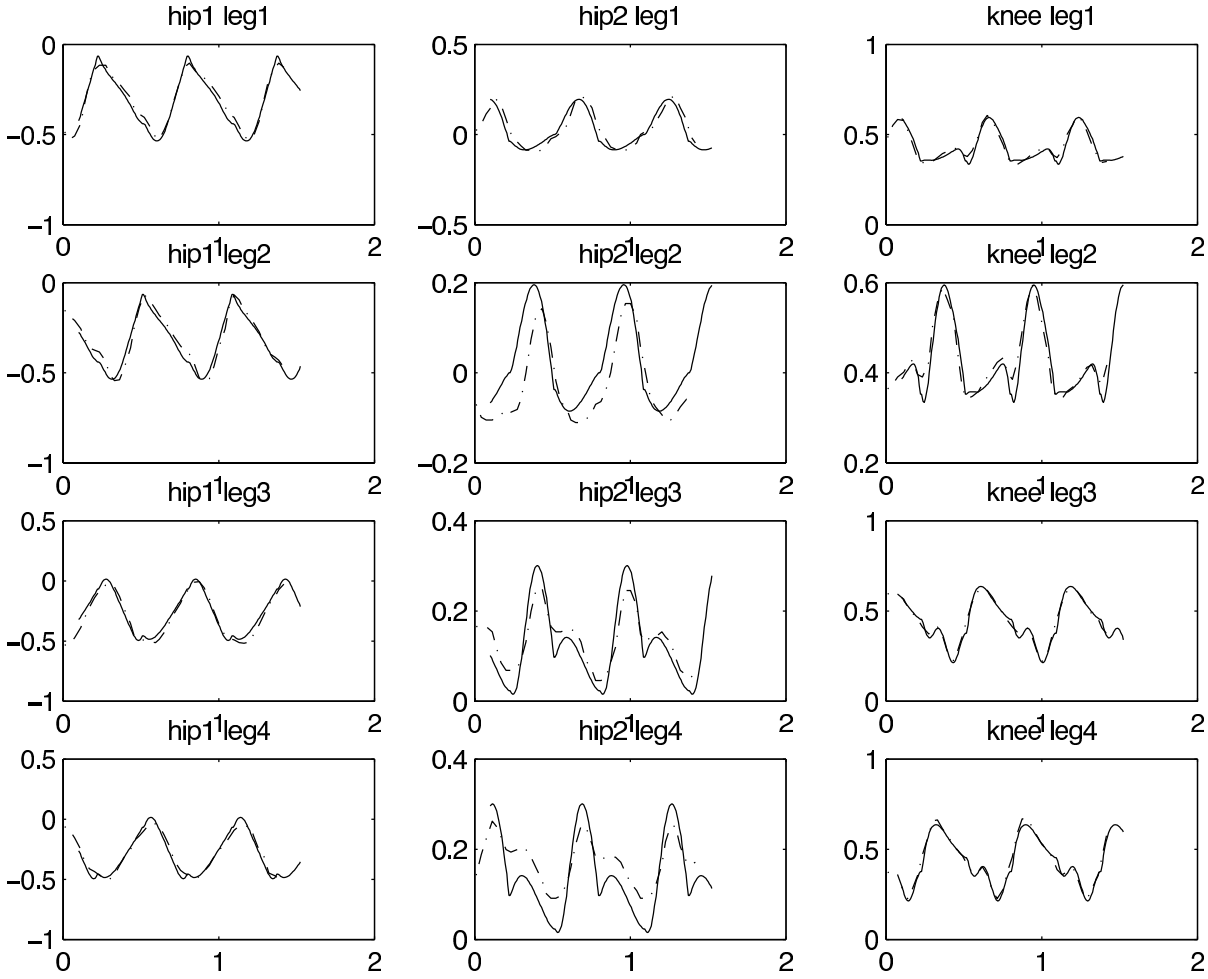


Figure 5.3: The experimentally measured joint angle trajectories (dotted lines; joint angle [rad] versus time [s]) for the first hip joints and the knee joints match the computed reference trajectories (solid lines) quite well after considering improved estimates for maximum torque and velocity constraints. 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; taken from [107].

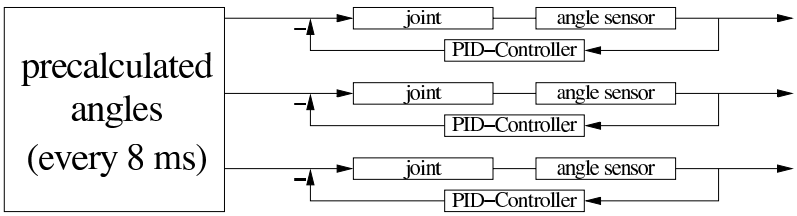


Figure 5.4: Decentralized PID-control scheme for the three joints of one leg; taken from [107].

Experimental Results. To implement the computed joint position trajectories on the robot, a decentralized trajectory tracking control scheme is used which utilizes the available servo motor control (e.g., PD or PID) of each actuated joint of the robot. Involving the data from joint angle sensors, the difference q_e between actual joint angle q_a and reference joint angle q_r of each joint can be measured. An additional error compensation angle q_d is calculated, e.g., in case of a PID controller, by $q_d(t) = k_p q_e(t) + k_i \int q_e(t)dt + k_d \frac{\partial q_e}{\partial t}(t)$ involving the three PID-controller constants k_p, k_i, k_d. This angle is added to the actual angle to get a new wanted angle $q_r = q_a - q_d$, as shown in Figure 5.4 for all joints of one leg.

The first experiments for a trot were not quite successful because the robot's feet slipped on ground and the measured angle trajectories did not match the calculated ones, despite a well working PID-controller scheme. The second problem was solved by adjusting maximum velocities and torques in the optimal control problem as described in Section 3. Sensor data now very well match the calculated trajectories for each of the twelve actuated joints (cf. Figure 5.3). Slipping was avoided by adding an additional constraint on the horizontal contact forces, cf. Section 3. No further control scheme for ensuring stability of the system is used or available. Therefore, the calculated reference trajectories and the system itself have to be robust against errors in the model concerning deflexions of bodies and joints. For the trajectories this is considered by involving the constraints on the swing height. A major practical problem for this approach results from the construction of the robot's feet and the not actuated small tip body, which makes it difficult to guarantee well-defined contact situations. This problem may be circumvented by a slight hardware modification: if "shoes" are attached, unchanged joint angle trajectories lead to good experimental results. Proper shoes also have a good effect on slipping, which may be reduced by choosing appropriate materials for the shoes, thus allowing faster motions. Without proper shoes and using the conservative velocity and torque restrictions in dynamic optimization, a maximum speed of 18 cm/s is achieved in simulation and experiment for the trot gait.

5.1.2 Gait Optimization for a Biped Robot

5.1.2.1 Kinematic Model of Humanoid Robot Prototype

A humanoid robot prototype has been constructed in co-operation with the control system group at TU Berlin [13]. For this robot, optimal walking trajectories have been calculated. Figure 5.5 displays a schematic sketch of the humanoid prototype. The humanoid robot consists of 17 actuated joints:

- two legs each with 6 actuated joints,
  - hip with 3 DOF rotating around x-, z- and y-axes,
  - knee with 1 DOF rotating around y-axis,
  - ankle with 2 DOF rotating around y- and x-axes,
- waist with 1 actuated joint rotating around z-axis, and
- two shoulders each with 2 actuated joints rotating around y- and x-axes.

The head was fixed to the body, though it was 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 virtual 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 first order differential equations describe the system configuration.

$$\boldsymbol{q} = \begin{bmatrix} q_{1-3} \\ q_{4-6} \\ q_{7-12} \\ q_{13-46} \end{bmatrix} = \begin{bmatrix} \text{Euler angles for system orientation} \\ \text{System linear translation vector} \\ \text{System angular and linear velocity vector} \\ \text{Legs, waist and shoulder angles and angle velocities} \end{bmatrix}$$

$$\boldsymbol{u} = \begin{bmatrix} u_{1-12} \\ u_{13-17} \end{bmatrix} = \begin{bmatrix} \text{Legs applied torques} \\ \text{Waist and shoulders joint applied torque} \end{bmatrix}$$

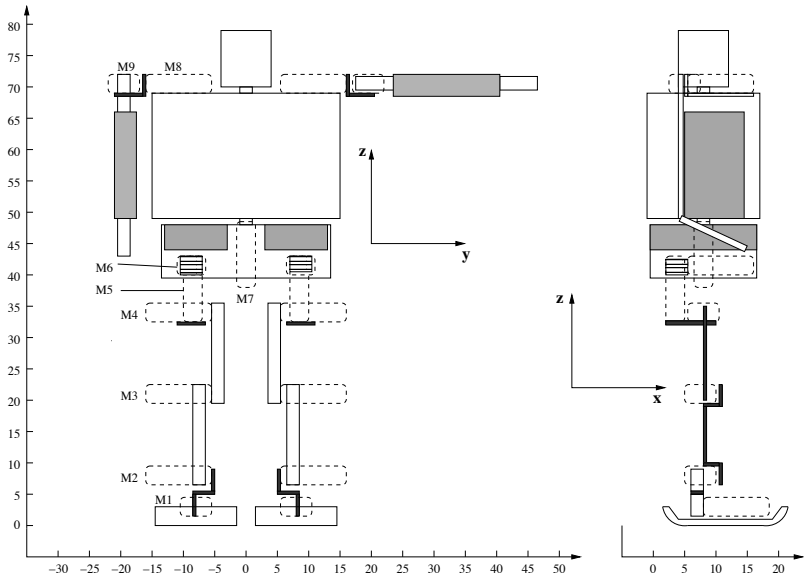


Figure 5.5: Humanoid kinematic structure; taken from [13].

For the trajectory optimization, the waist and both shoulder joints are held fixed, so the above state and control vectors simplify to $\boldsymbol{q} \in \mathbb{R}^{36}$ and $\boldsymbol{u} \in \mathbb{R}^{12}$ which can further be reduced by involving the reduced dynamics approach from Section 3.2. For the single limb support phase, only the states of the free floating base and the swing leg have to be considered, i.e. 24 states are used in the optimal control problem. For the double limb support phase, no states related to the robot's leg have to be considered which results in a reduction of the number of states to 12. The problem size further could be reduced by the special discretization for differential equations of second order (Section 4.3), which however had not yet been implemented when those investigations were taken.

The humanoid robot is 80 cm tall and has a weight of about 12 kg without batteries and mainboard. Its kinematic structure complied with the regulations for the RoboCup Humanoid League in 2003 [3]. Other kinematic design decisions take into account dynamical aspects. For example, the hip flexion/extension joint performing most of the work in the hip was placed as the last of the three hip joints. Thus, the needless work of swinging the other two hip joints is saved. On the other hand, the flexion ankle joint is placed higher than the abduction joint so that at collision of the heel with the ground the impulsive force will disperse better throughout the body rather than influence primarily only the ankle joints.

5.1.2.2 Inverse Kinematics

For the evaluation of the reduced dynamics (Section 3.2), the solution of the inverse kinematics problem is required for a leg which may also be considered as a 6-link manipulator. It is well-known that an inverse kinematics solution does not always exist for a 6-link manipulator, yet for this humanoid kinematic structure a unique solution may be found. The problem is solved by first modeling the three successive planar rotations (last hip joint, knee joint, first ankle joint) as a single rotation. Then the inverse kinematics problem reduces to the identification of four angles given the hip and ground contact positions. Taking compositions of homogenous transformations with symbolic programming tools produces equations from which the four joint angles may be solved. The remaining joint angles are determined from the inverse kinematics solution of a planar 3-link manipulator [100]. The correct solution out of the finite number of possible solutions is determined from considering the direction of knee rotation and relative lateral displacement of the foot with respect to the hip. The calculation of the 6 joint angle velocities given the hip linear and angular velocities is a linear problem for which standard methods may be used.

5.1.2.3 Dynamic Parameters

For dynamical calculations the humanoid model must include dynamic parameters which are estimated based on its kinematic structure and mass measurements. When striving for precise optimization results, it is important that these estimated values are as exact as possible. The real humanoid is divided into geometrical primitives (cylinder, ellipsoid, or box) which were individually measured and weighed. One leg, for example, is divided into more than 20 units. Assuming a uniform mass density in an object, the link inertias may be approximated with the help of the parallel and perpendicular axis theorem.

5.1.2.4 Constraints for Bipedal Walking Model

An important aspect of formulating a gait optimization problem is establishing the many constraints on the problem. For a biped, the gait cycle consists of several phases describing different contact situations and being separated by events. The order of contact events is straightforward and depends primarily upon the speed of locomotion. A summary of the *physical* modeling constraints for a *half-stride* consisting of a single limb support phase (SLS) and a double limb support phase (DLS) of a periodic gait cycle in 3-dimensions is [13]:

1. **Magnitude constraints on states and controls:**

$$L_q \leq \mathbf{q} \leq U_q, \;\; L_u \leq \mathbf{u} \leq U_u,$$

where $L_{(.)}$ and $U_{(.)}$ are constant vectors with length equal to the length of their argument containing upper and lower bounds of their argument.

2. **Boundary conditions at end of half-stride:**

- *symmetry resp. anti-symmetry of states* $\mathbf{q}$ *and contact forces* $\mathbf{f}_c$*:*
  Rotational states, controls and contact forces are symmetric about inertial y-axis and anti-symmetric about x- and z-axes, while linear states and contact forces are symmetric about inertial x- and z-axes and anti-symmetric about y-axis. $q^e_{(.)}$ and $q^0_{(.)}$ denote the value of state $q_{(.)}$ at final resp. initial time:

$$\begin{bmatrix} q^e_1 \\ q^e_2 \\ q^e_3 \\ q^e_4 \\ q^e_5 \\ q^e_6 \end{bmatrix} = \begin{bmatrix} -1 & & & & & \\ & 1 & & & 0 & \\ & & -1 & & & \\ & & & 1 & & \\ & 0 & & & -1 & \\ & & & & & 1 \end{bmatrix} \begin{bmatrix} q^0_1 \\ q^0_2 \\ q^0_3 \\ q^0_4 \\ q^0_5 \\ q^0_6 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ step \\ 0 \\ 0 \end{bmatrix} \text{ and } \begin{bmatrix} q^e_7 \\ q^e_8 \\ q^e_9 \\ q^e_{10} \\ q^e_{11} \\ q^e_{12} \end{bmatrix} = \begin{bmatrix} -1 & & & & & \\ & 1 & & & 0 & \\ & & -1 & & & \\ & & & 1 & & \\ & 0 & & & -1 & \\ & & & & & 1 \end{bmatrix} \begin{bmatrix} q^0_7 \\ q^0_8 \\ q^0_9 \\ q^0_{10} \\ q^0_{11} \\ q^0_{12} \end{bmatrix}$$

  Periodic constraints are also placed between the contact forces experienced by the stance foot at the beginning of the SLS and from the foot about to become the sole stance foot at the end of the DLS. These constraints satisfy the same symmetric and anti-symmetric relationship as above. The periodicity constraints for the leg's 6-DOF are implicitly enforced through the 6-dimensional foot contact constraints, contact location and their periodicity constraints.

- *lift-off force:*
  The leg to lift off at the end of the half-stride is able to lift off from the ground iff the vertical component of the contact force is zero at the foot's center of pressure. This point is in general unknown so that one must additionally restrict the rotational contact forces about the ground planar axes to be zero at some reference point underneath the foot at the end of the DLS. Friction constraints simultaneously require that the remaining components of the 6-dimensional contact force vector be zero,

$$\boldsymbol{f}_{c,i} = \mathbf{0}_{6\times 1} \qquad \text{where leg } i \text{ is breaking its contact with the ground.}$$

3. **Constraints during whole half-stride:**
   The following constraints must hold for all times during SLS resp. DLS.

- *stability:*
  ZMP lies in the convex hull of contact points, i.e. the distance from ZMP to the convex hull HULL of the contact points is negative (for both SLS and DLS):

$$\mathrm{dist}(\text{ZMP}, \text{HULL}) \leq 0.$$

- *swing foot orientation:*
  The swing foot rotational position must not divert too far from its starting and ending configuration, i.e. norm of vector $\mathbf{FO}$ containing the Euler angles of foot orientation received from a forward kinematics algorithm FK must be smaller than a user given value U_{orient} (for SLS and the swing foot only):

$$\mathbf{FO} = \text{FK}(\mathbf{q}, \mathbf{u}) \quad \text{and} \quad \|\mathbf{FO}\| \leq U_{orient}$$

- *leg reach:*
  Working with a reduced dynamics model has the advantage of dealing with an ODE dynamical model rather than with a DAE model. The disadvantage is that extra constraints must be added to ensure that there exists a solution for the dependent states in the model. In this case, the hip(s) connected to the leg(s) in contact with ground must remain within a maximal distance from the contact point(s) i.e.

$$\mathbf{P}_{hip} = \text{FK}(\mathbf{q}) \quad \text{and} \quad dist(\mathbf{P}_{cntct}, \mathbf{P}_{hip}) \leq l_{hip-leg}$$

  where $\mathbf{P}_{cntct}$ are the coordinates of the contact point(s) and FK is a forward kinematics algorithm that computes the position of the hip $\mathbf{P}_{hip}$ from the states. This condition must hold for all legs in contact with the ground, that is for one leg in SLS and for two legs in DLS, so that the inverse kinematics solution for the leg has a well-defined solution.

- *swing height:*
  The swing foot must move above a pre-defined tolerance zone above the ground due to robustness concerns, i.e. the z position coordinate of the foot tip q_{tip} (calculated by forward kinematics FK) must be greater than e.g. a sine curve of pre-specified amplitude:

$$q_{tip,z} = FK(\mathbf{q}) \quad \text{and} \quad q_{tip,z} \geq A_z \sin\frac{\pi t}{t_1} \quad \text{with} \quad t_1 = \text{duration of SLS}.$$

- *avoidance of slipping:*
  Ground contact forces lie within the friction cone and unilateral contact constraints are not violated [30, 85]. In the case of stationary flat foot contact, the ground linear contact forces for foot j, $\boldsymbol{F}_j = [F_{j,x}\ F_{j,y}\ F_{j,z}]^T$ and rotational contact forces $\boldsymbol{T}_j = [T_{j,x}\ T_{j,y}\ T_{j,z}]^T$ must satisfy

$$\sqrt{F_x^2 + F_y^2} \leq \mu_t F_z \quad \text{and} \quad |T_z| \leq \mu_d F_z$$

  with friction coefficients $\mu_t,\ \mu_d$. Otherwise a slipping contact state is entered.
  To prevent a foot lying flat on the ground from entering a rotational contact state, the center of pressure must be constrained to lie underneath the foot surface which may also be expressed

in terms of the rotational contact force vector $\boldsymbol{T}_j$,

$$|T_x| \leq 0.5 F_z l_y \quad \text{and} \quad |T_y| \leq 0.5 F_z l_x \,,$$

where l_x and l_y are the length and width of the foot respectively.

- *positive contact forces:*
  All legs in contact with ground may only push to ground but may not pull from ground, so the z-component of contact force of each leg in contact with ground must be positive:

$$F_z \geq 0.$$

4. **Conditions at change of phase from SLS to DLS:**
   Here t_- resp. t_+ denotes time just before and just after a phase change.

- *continuity of position, angle, and objective function states:*

$$q_i(t_+) = q_i(t_-),$$

  for all components q_i of $\mathbf{q}$ related to positions or angles (see Section 5.1.2.1).

- *discontinuity of velocity states:*
  The jump in the generalized system velocities due to an inelastic collision with the ground are calculated using the collision dynamics algorithm from Section 3.1.3:

$$\text{CDA} : \dot{\boldsymbol{q}}(t_+) = \text{CDA}\left(\boldsymbol{q}(t_-), \dot{\boldsymbol{q}}(t_-)\right)$$

- *foot placement and foot orientation:*
  As both feet are in contact with ground in the double limb support phase and are stationary, foot placement and orientation must be considered at the beginning and end of the SLS. A forward kinematics algorithm FK determines the foot position $\mathbf{fp}$ and orientation $\mathbf{fo}$. Foot position must agree with the unknown fixed parameters $\mathbf{fp}_d$ in the optimization formulation while foot orientation must be in-line with the inertial reference system in which case its relative rotation matrix is the identity matrix $\mathbf{I}_3$:

$$(\mathbf{fp}, \mathbf{fo}) = \text{FK}\left(\mathbf{q}(t_-)\right), \quad \mathbf{fp} = \mathbf{fp}_d \quad \text{and} \quad \mathbf{fo} = \mathbf{I}_3$$

5.1.2.5 Optimization of Stability and Performance Indices

Algebraic control strategies for legged systems cannot yet be constructed to handle the high dimension and many modeling constraints present in the locomotion problem. Heuristic control

methods, on the other hand, tend to have poor performance with respect to power efficiency and stability. The remaining proven approach is the use of sophisticated numerical optimization schemes which can incorporate the numerous modeling constraints to generate optimal trajectories. The resulting trajectories may later be tracked or used to approximate a feedback controller in the portion of state space of interest. The following three performance indices currently are used in the humanoid gait generation investigations.

Postural Stability Performance: Distance in the ground plane between foot i's stability criterion (cf. Section 2.1.4) ${}_i\boldsymbol{p_f}$ and a central reference point under the foot ${}_i\boldsymbol{p_r}$

$$\boldsymbol{J}_{s1}[\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{u}] = \int_0^{t_f} \sum_i \left(({}_ip_{f,x} - {}_ip_{r,x})^2 + ({}_ip_{f,y} - {}_ip_{r,y})^2 \right) \, \mathrm{d}t. \tag{5.1}$$

Energy Performance: In legged systems where a high torque is generated by a large current in the motor, the primary form of energy loss is called the Joule thermal loss [65]. The integral of this value over a gait period is

$$\boldsymbol{J}_{e1}[\boldsymbol{u}] = \frac{1}{s} \int_0^{t_f} \sum_{i=1}^{N} R_i \left(\frac{u_i}{G_i K_i} \right)^2 \, \mathrm{d}t \tag{5.2}$$

where R_i, G_i, K_i, and u_i are the armature resistance, gear ratio, torque factor, and applied torque for link i respectively, while s is the step length or total distance traveled over one stride.

Efficiency Performance: The specific resistance as used in [35] measures the output power in relation to the mass moved and the velocity attained and is a dimensionless quantity. It represents a normalized form of the required kinetic energy

$$\boldsymbol{J}_{e2}[\dot{\boldsymbol{q}}, \boldsymbol{u}] = \int_0^{t_f} \frac{\sum_{i=1}^{N} |u_i \dot{q}_i|}{mgv} \, , \tag{5.3}$$

where mg is the weight of the system, $\dot{q}_i$ is the joint i angle velocity and v is the average forward velocity.

The availability of a fully validated dynamic model combined with optimization tools permits one to make conclusive investigations into which stability or efficiency measures are most effective, though no one measure is sufficient for gait generation. The stability performance 5.1 cannot be used alone to verify or design a dynamically stable control strategy and must be combined with additional dynamic system measures. Efficiency is secondary in importance to stability in legged systems, but it can also have a strong influence in the successful design of an autonomous biped. A challenge for systems with limited power supply is to combine energy conserving motion with the robust stability properties discussed previously.

5.1.2.6 Computation of Humanoid Reference Trajectories

Inverse kinematics algorithms that were developed for the humanoid prototype in order to compute the reduced dynamics (see Section 3.2) also facilitated the generation of heuristic joint angle and angle velocity reference trajectories satisfying the physical modeling constraints (Section 5.1.2.4). The reference trajectories served as start trajectories for the complex 3-dimensional humanoid gait optimization.

Several stages of gait optimizations were performed with varying complexity until all physical and stability constraints were included in the 3-D optimizations. An energy performance index was chosen (Equation 5.2) subject to the statically stable and dynamic postural stability nonlinear constraints (Section 2.1.4). First investigations using a dynamic model considering only the 12 joint DOF in the legs were made using statically stable gaits, walking on flat feet, with one swing phase composing 80–85% of the gait period and a double contact phase composing the remainder of the gait period. This conservative gait was chosen to facilitate first experiments with the humanoid robot prototype. Thus a 25-dimensional ODE (including the objective) has been optimized subject to numerous explicit and implicit nonlinear boundary constraints and nonlinear inequality constraints (Section 5.1.2.4). An optimization using 44 time grid points required 1584 NLP variables (Section 5.1.2.5) with 1079 nonlinear equality constraints and 220 nonlinear inequality constraints. The necessary run-time after two automatic grid refinements using a reasonable starting solution was 1418 seconds on a Pentium III, 1150 MHz. The GCoM and individual foot FRI trajectories from the optimal gait are displayed in Figure 5.6. Note that the system remains statically stable and that the FRI points remain centered about the middle of their respective foot contact surfaces.

5.1.2.7 Experimental Results

In the following, results of two experiments are described: In the first experiment, the trajectories generated according to Section 5.1.2.6 are scaled in duration and applied to the humanoid without further modification as reference trajectories to local joint PD position controllers. The second experiment shows walking performance with completely unmodified trajectories.

Trajectory Following Using Slow Trajectories

The precalculated trajectories obtained by the algorithms discussed in Section 5.1.2.6 are scaled in time by a factor of 20 for debugging purposes. Those prolonged trajectories are applied to the humanoid as reference trajectories to joint level PD position controllers. Figure 5.7 shows the

measured data of the left knee. Since the knee joint supports a significant part of the robot total weight, the load in the other joints is similar or less than the knee load.

From Figure 5.7(a) one can see, that the error of the commanded joint trajectory (dashed) and the measured position (solid) is quite small and does not exceed 0.025 rad for a complete stride. This validates the performance of the PD joint position control with a sampling rate of 250 Hz. The corresponding motor current (solid) and PWM ratio (dashed) are shown in Figure 5.7(b). This plot similarly displays that the knee joint of the robot operates well below its limits with currents of 3 A (below the maximum H-bridge amplifier current of 4 A) and the PWM ratio always less than 50%. Another insight from this result is that the commanded PWM ratio is roughly proportional to the current in the motors, which indicates that in principle torque command control is realizable with the given hardware architecture. The quality of the (unfiltered) current measurement is promising for future application of to be developed external disturbance force estimation algorithms.

Despite small errors in trajectory following in joint space, the robot gait was slightly tottering. One cause may be attributed to unmodeled backlash in the gears and other effects such as link flexibilities. Furthermore by stretching the time scale of the stride for debugging reasons, dynamic effects incorporated in the gait planning were reduced.

Trajectory Following Using Fast Optimal Control Trajectory

The result of conducting the same experiment as in Section 5.1.2.7 with a faster, unscaled trajectory is shown in Figure 5.9. In this experiment, two strides are performed, where each stride consists of a single-support phase and a double support phase, each taking 3 s, see Figure 5.9a. Although the absolute error in the knee joint is 0.06 rad and hence slightly higher than in the case of the slow trajectory, the average load on the joints is less when compared to the slow trajectory. The robot was able to do the double stride without additional control, i.e. with joint angle control only, without falling after adjusting the distance of the foot, see Figure 5.8. This shows that the robot dynamics have successfully been exploited in the gait generation process.

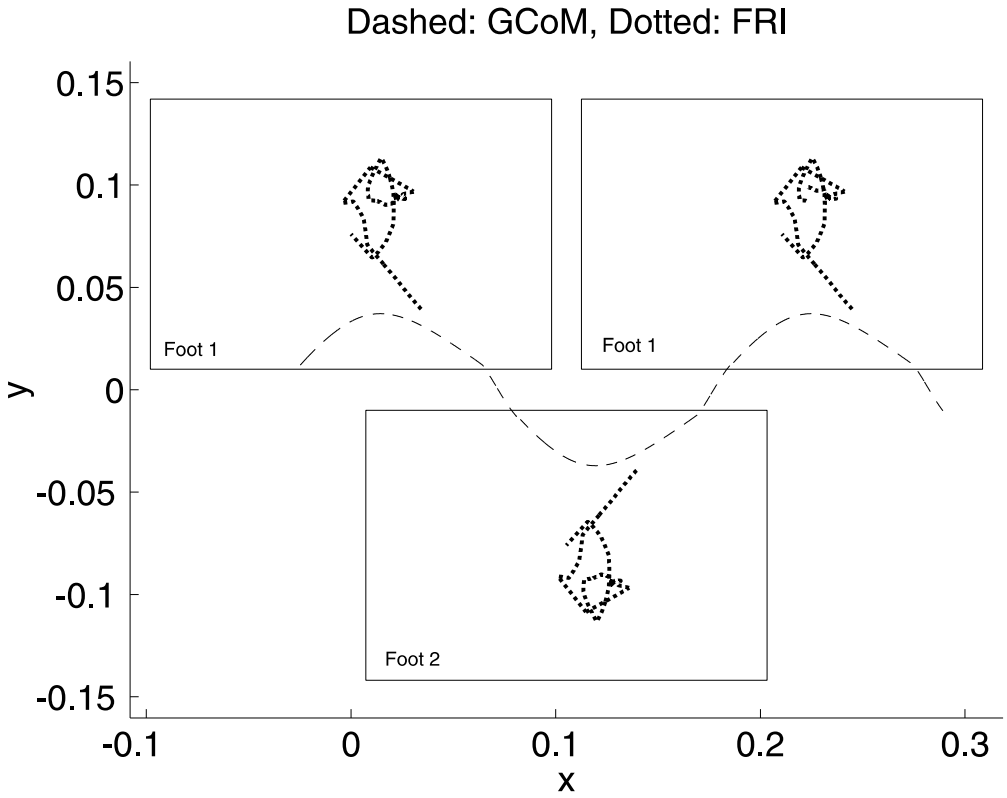
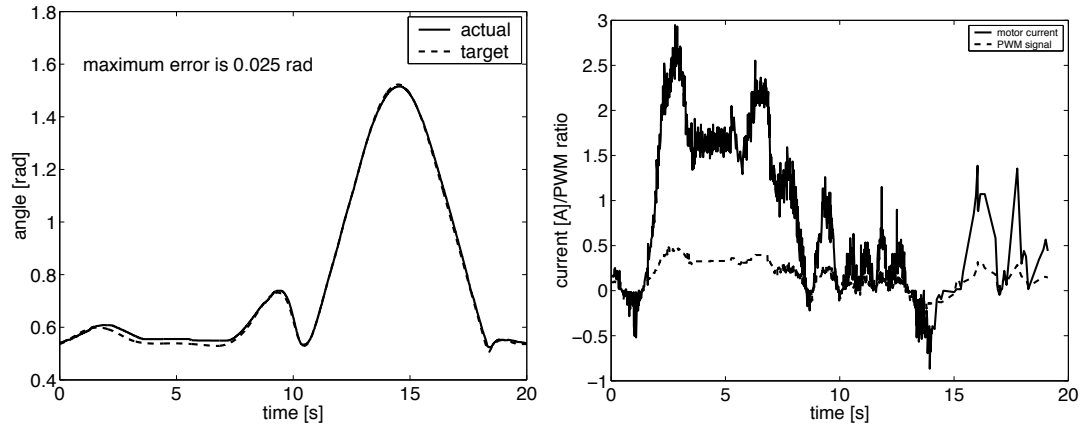


Figure 5.6: GCoM and individual foot FRI trajectories during two steps of an optimized statically stable walk; taken from [13].



(a) (b)

Figure 5.7: Joint angle (left) and motor signals (right) of first experiments with trajectory following control (knee joint of left leg); taken from [13].

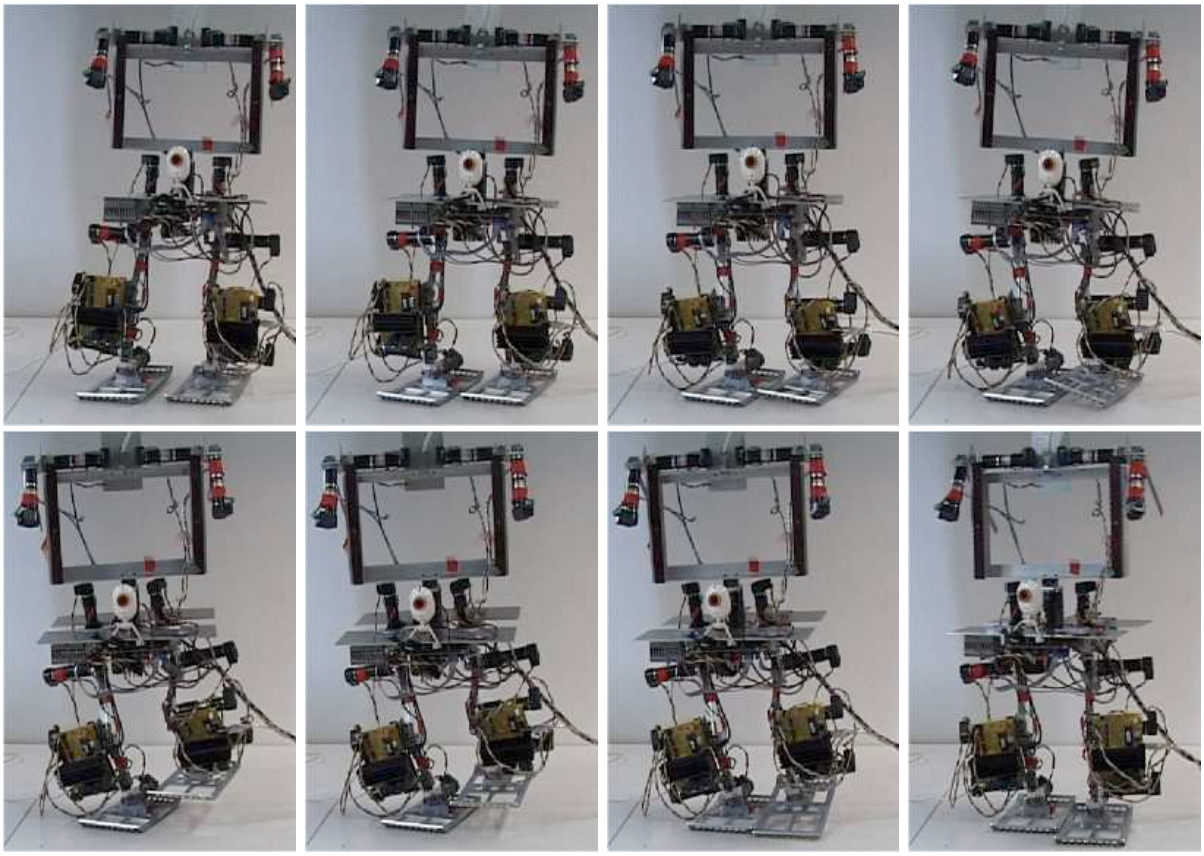
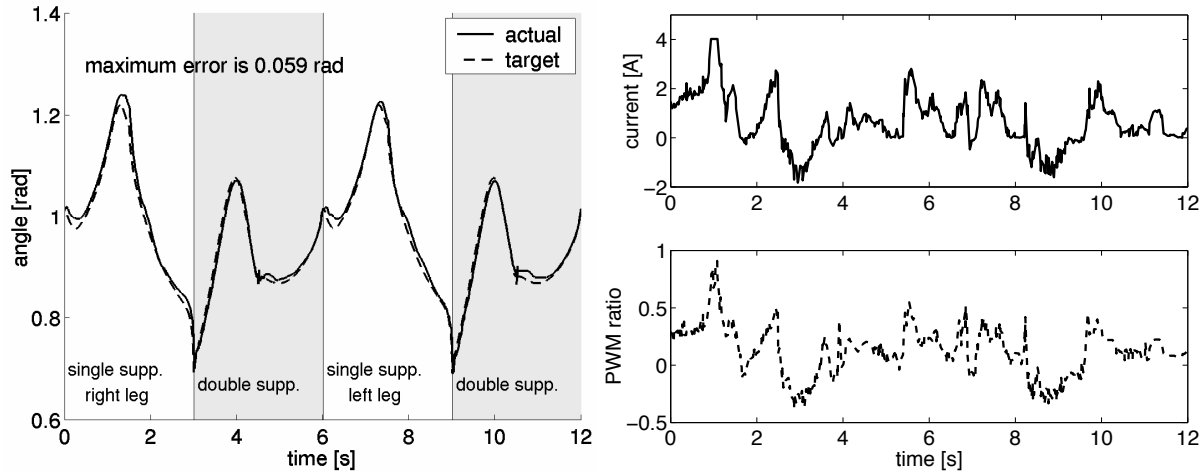


Figure 5.8: Step sequence with manual compensation; taken from [13].



(a) (b)

Figure 5.9: Joint angle, current and pwm signal for a double stride of 6 s; taken from [13].

5.1.3 Walking Optimization for a Biologically Inspired Biped Robot

Optimal control techniques are well suited for optimization problems where the controls can be virtually any arbitrary function. For problems with controls of simple shape in contrary discretization of the controls would increase the problem to an unnecessary big size. In [93], the optimization of the biologically inspired elastic biped JenaWalker II (see Figure 5.10) is investigated. Only the hips are driven. The knee and the ankle joints are not driven directly but effected by the motion of the hip via muscle like springs.

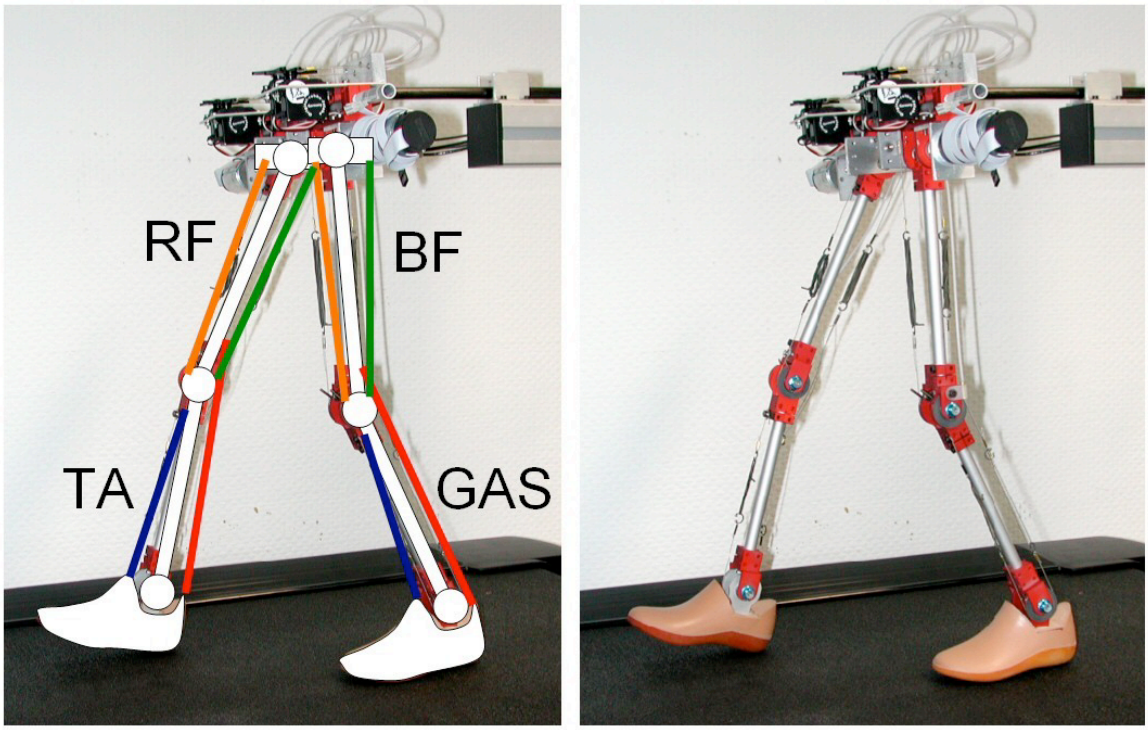


Figure 5.10: The JenaWalker II. RF, BF, TA and GAS denote the springs inspired by the human muscles Rectus Femoris, Biceps Femoris, Tibalis Anterior und Gastrocnemius; taken from [93].

The biped is driven by a central pattern generator: parameters for the frequency, the amplitude and the offset angle generate sinusoidal hip angle trajectories for both hips. Discretization would lead to a much higher number of parameters to be optimized. Besides the parameters for the control of the hip, also the parameters of the muscles (like rest length, spring coefficients) may be optimized.

Different optimization techniques like Implicit Filtering [33] and NoMad [4] have been used to optimize a Matlab Simulink SimMechanics model of the robot scaled to human size (leg length of $1m$, mass $80kg$). Starting from a set of hand tuned parameters (walking speed about

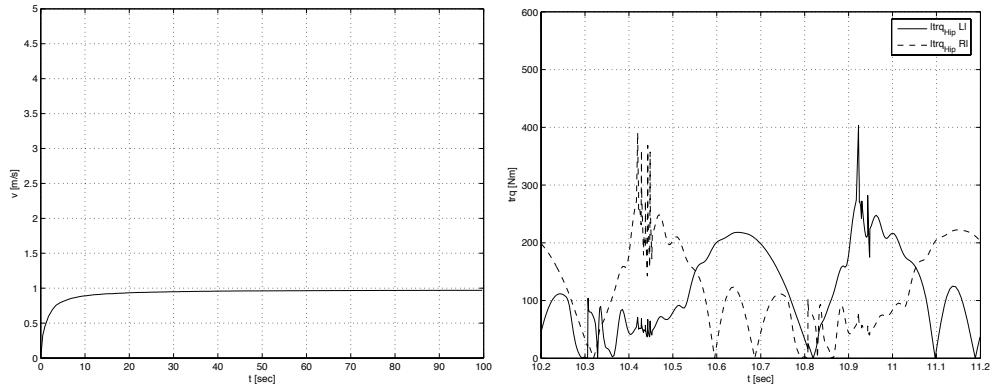


Figure 5.11: Walking speed (left) and hip torques (right) with initial parameter set; taken from [93].

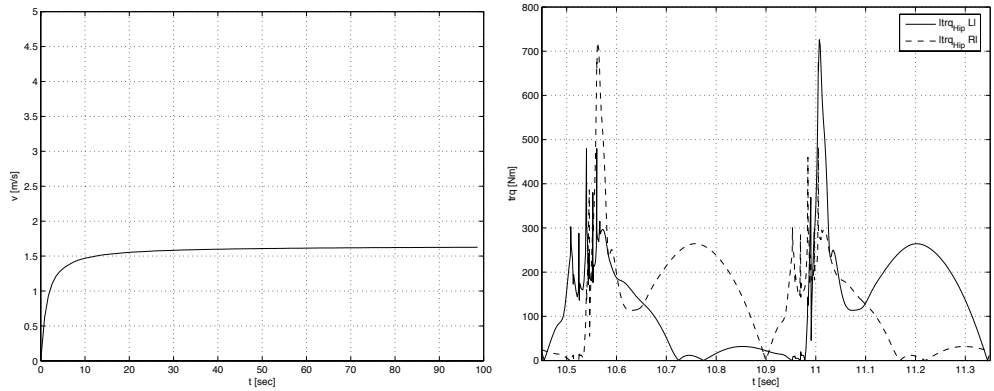


Figure 5.12: Walking speed (left) and hip torques (right) with parameter set optimized for speed using Implicit Filtering; taken from [93].

$1m/s$, Figure 5.11), several optimization studies have been conducted. First the walking speed was optimized without any constraints; this lead to an increase of velocity to $1.6m/s$ but also increased the hip torques to more than $700Nm$ (Figure 5.12). The hip torques then have been limited and the walking speed has been optimized in the second optimization study. Involving NoMad even increased the constrained walking speed to about $3.6m/s$ at reduced hip torques of about $410Nm$ (Figure 5.13). Obviously, Implicit Filtering got stuck in a local minimum. In the third study finally, the speed was constrained to be equal or higher than $2m/s$ and the hip torques have been minimized. This lead to hip torques of less than $300Nm$ at a final speed of $2.5m/s$, see Figure 5.14. Speed is evaluated 10 seconds after the robot starts from standing, but still increases after this time.

Both in the numerical optimizations and in experiments with the real robots, running with flight phases has been observed. This also explains the velocities that become higher than the theoret-

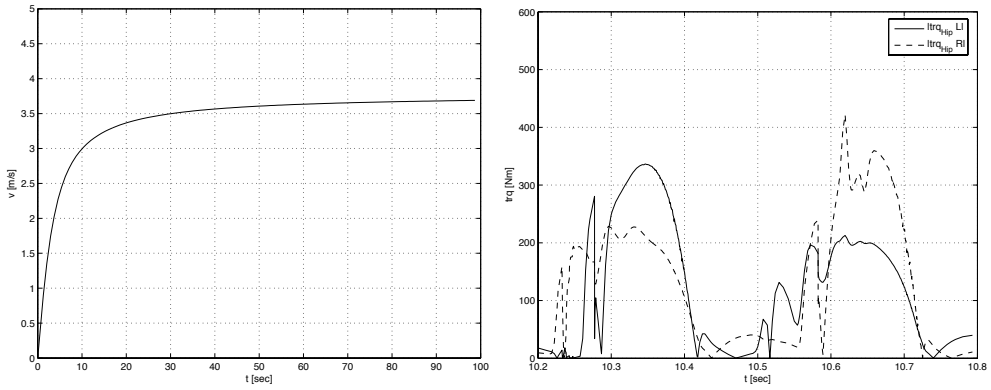


Figure 5.13: Walking speed (left) and hip torques (right) with parameter set optimized for speed and bounded torques using Nomad; taken from [93].

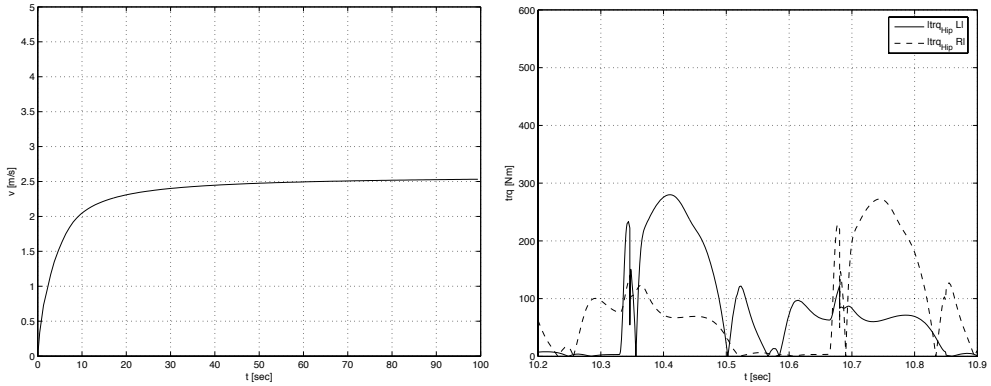


Figure 5.14: Walking speed (left) and hip torques (right) with parameter set optimized for low hip torques and bounded velocity using Nomad; taken from [93].

ical maximum walking speed of angular velocity of the hip joints multiplied by the leg length.

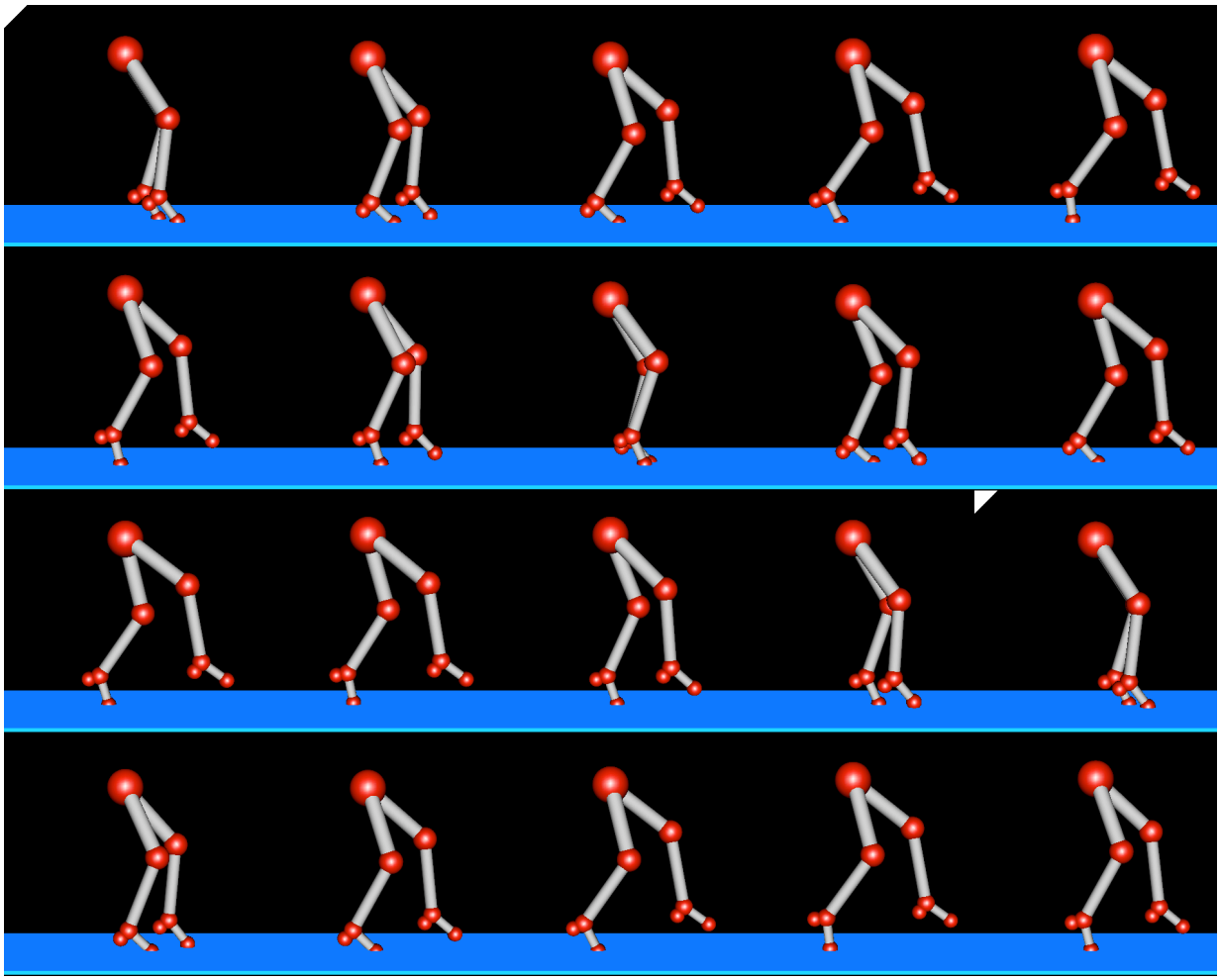


Figure 5.15: Walking motion with parameter set optimized for low hip torques and bounded velocity using Nomad; taken from [93].

5.2 Human Biodynamics

In the following subsections, numerical results for biomechanical applications are presented. The kicking motion has been previously published [101, 102, 103, 104, 105, 106]. Examples have been chosen for which complete data, i.e. captured motion data, electromyography (EMG) data (i.e., the potentials produced by the muscles that finally lead to the exertion of muscle forces), anthropometric and physiological data of the test person, was available in literature, to allow validation.

5.2.1 Kicking Motion

5.2.1.1 Prediction of Motion

A time optimal kicking motion has been investigated [101, 102, 103, 104]. Kinematic and kinetic data of the musculoskeletal system as well as muscle model parameters and measured reference data have been taken from Spägele [97, 99]. The model (cf. Figure 5.16) consists of two joints, two rigid links and five muscle groups. The problem is formulated as an optimal control problem

1. Ilio Psoas group
2. Vastus group
3. Rectus Femoris
4. Hamstring group
5. Gastrocnemius group

q_1 hip angle

q_2 knee angle

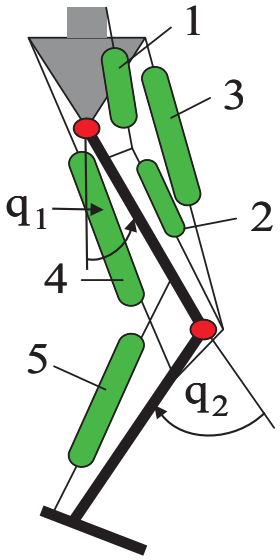


Figure 5.16: Kinematic structure of the leg with five muscle groups.

with 9 states (hip angle q_1, knee angle q_2, the corresponding joint velocities and 5 calcium ion

concentrations) and 5 controls (activations of the muscles) as follows:

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

The kicking motion was optimized to be time optimal, i.e. the objective function is $\mathcal{J} = t_f$. The muscle lengths (cf. Equation 2.11, subscript here denotes the number of the muscle) are calculated according to [97]:

$$\begin{aligned} l_1^M &= 0.287 - 0.0497q_1, \\ l_2^M &= 0.300 + 0.0330q_2, \\ l_3^M &= 0.517 + 0.045\cos(1.128q_1 + 0.748) + 0.033q_2, \\ l_4^M &= 0.483 - 0.062\cos(1.047q_1 + 0.838) + 0.07\cos(1.076q_2 + 0.28), \\ l_5^M &= 0.088 + 0.019\cos(1.16q_2 + 0.464)\,. \end{aligned} \tag{5.4}$$

The velocities are the time derivatives of the lengths, $v_i^M = \dot{l}_i^M, i = 1, \dots, 5$. The resulting lever arms (Equation 2.2.3.7) are also taken from [97]:

$$\begin{aligned} d_{1,q_1} &= 0.024 + 0.0188q_1, \\ d_{2,q_2} &= 0.036 + 0.03e^{-4.33(0.17-q_2)^2}, \\ d_{3,q_1} &= 0.052\cos(q_1 - 0.63) - 0.002, \\ d_{4,q_1} &= 0.037\cos(1.309q_1 - 0.916) + 0.026, \\ d_{4,q_2} &= 0.058(q_2 + 0.685)^2 e^{-1.187q_2}, \\ d_{5,q_2} &= 0.055\,. \end{aligned} \tag{5.5}$$

The passive moments (Equation 2.14) are stated in [97] to be:

$$\begin{aligned} \tau_{1,p} &= 0.8e^{-3.41q_1} + 0.084e^{-15q_1} - 0.753e^{2.55q_1} - (7.9e^{-2.72q_1} + 0.09e^{1.8q_1})\dot{q}_1 \\ \tau_{2,p} &= 1.25 \cdot 10^{-7}e^{8.5q_2} - 6.3e^{-2.9q_2} - 20.1e^{-16.1q_2} + 2.1 - (0.3e^{1.02q_2} + 1.85e^{-3.43q_2})\dot{q}_2\,. \end{aligned} \tag{5.6}$$

The multibody system parameters (mass, inertia w.r.t. point of rotation, center of mass, length)

for the thigh and shank are

$$\begin{array}{llll} m_1 = 8.692, & I_1 = 0.480, & z_1 = 0.189, & l_1 = 0.447, \\ m_2 = 15.492, & I_2 = 4.700, & z_2 = 0.501, & l_2 = 0.538. \end{array} \tag{5.7}$$

The activation dynamics are taken into account with equations 2.5 and 2.6, the parallel elastic element and damping element with equations 2.7 and 2.8, the force-length- and force-velocity-relationship with equations 2.4 and 2.3. The total force of the muscle-tendon element and the overall muscle are calculated with equations 2.9 and 2.10. The parameters for the respective equations may be found in Table 5.1. The boundary conditions were set to

$$\begin{array}{rcl} q_1(t_0) & = & 0.1, \\ q_2(t_0) & = & 0.15, \\ \dot{q}_1(t_0) & = & 0 \\ \dot{q}_2(t_0) & = & 0, \\ \gamma_1(t_0) & = & ... = \gamma_t(t_0) = 0 \\ q_1(t_f) & = & 0.8, \\ q_2(t_f) & = & -0.05, \\ \dot{q}_2(t_f) & = & 0. \end{array} \tag{5.8}$$

Box constraints are imposed on the states and controls

$$\begin{array}{rcl} 0 \leq & q_1 & \leq 1.5, \\ -0.05 \leq & q_2 & \leq 1.5, \\ 0 \leq & u_i, \gamma_i & \leq 1, i = 1, \ldots, 5. \end{array} \tag{5.9}$$

An overview of the data flow in the kicking model may be found in Figure 5.18.

Compared to the measured motions (and the results of [97, 99], which match the measured data very well), our results show a shorter time and higher maximum angles (cf. Figure 5.19). The reason for this is, that in [97] the maximum muscle forces were modified to match the optimized time of the measurement. Obviously our optimal motion is another local minimum. Nevertheless, the controls (Figure 5.17) show the same characteristics. Computing time and size of the resulting NLP are shown in Figure 5.2. The direct shooting approach used in [97, 99] for 11 grid points required hours to compute the solution [96]. Comparing the computing time with our approach (Figure 5.2) and considering how computational speed has progressed since 1996, we still obtain a speed up of two orders of magnitude.

Table 5.1: Parameters for the muscles of the kicking model (taken from [97]).

parameter	muscle 1	muscle 2	muscle 3	muscle 4	muscle 5
b_1	7.23	7.23	7.23	7.23	7.23
b_2	7.46	7.61	8.37	7.46	7.61
b_3	1.0	1.0	1.0	1.0	1.0
$k_0[N]$	0.0	0.0	257.1	378.0	0.0
$k_1[N]$	0.0	0.0	5.393	64.7	0.0
$k_2[m^{-1}]$	0.0	0.0	90.4	23.95	0.0
$k_3[m]$	0.0	0.0	0.58	0.48	0.0
$k_4[N]$	0.0	0.0	0.0	0.0068	0.0
$k_5[m^{-1}]$	0.0	0.0	0.0	239.8	0.0
$k_6[m]$	0.0	0.0	0.0	0.53	0.0
c_1	0.017	0.017	0.017	0.017	0.017
c_2	0.015	0.015	0.015	0.015	0.015
$l_0^M[m]$	0.258	0.309	0.500	0.486	0.085
c_3	0.50	0.33	0.33	0.50	0.50
c_4	0.09	0.02	0.08	0.10	0.03
$v_{max}^M[m/s]$	-1.6	-0.5	-2.0	-1.8	-0.5
$F_{\max}^{\text{iso}}[N]$	4800.0	5300.0	1200.0	1500.0	700.0

Table 5.2: Size of the resulting NLP and computation time on a 1700+ Athlon XP for two different numbers of grid points in the discretization.

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

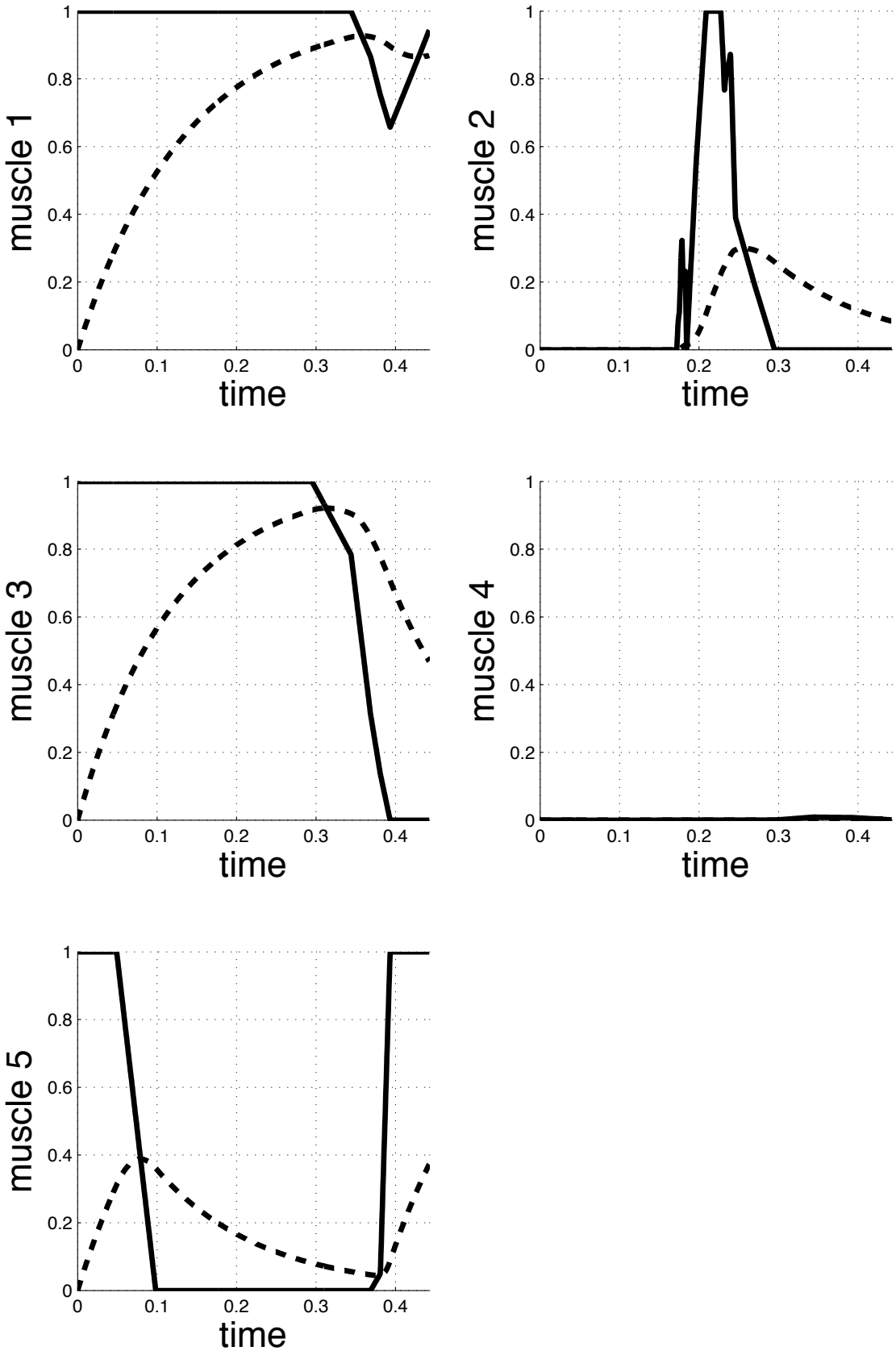


Figure 5.17: Results from optimization: Controls (corresponding to EMG, solid line) and calcium ions concentrations (dashed line).

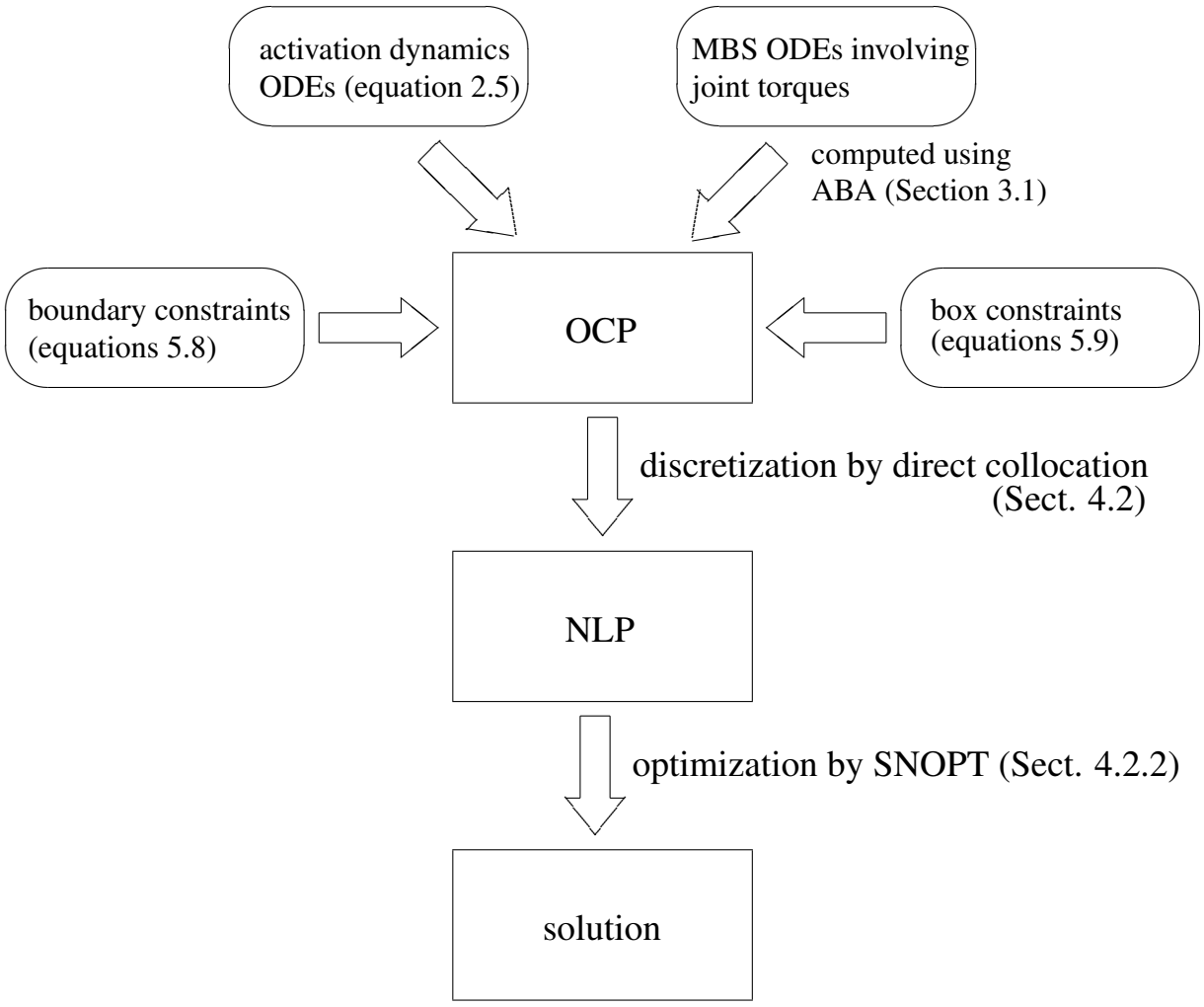


Figure 5.18: Schematic summary of the method.

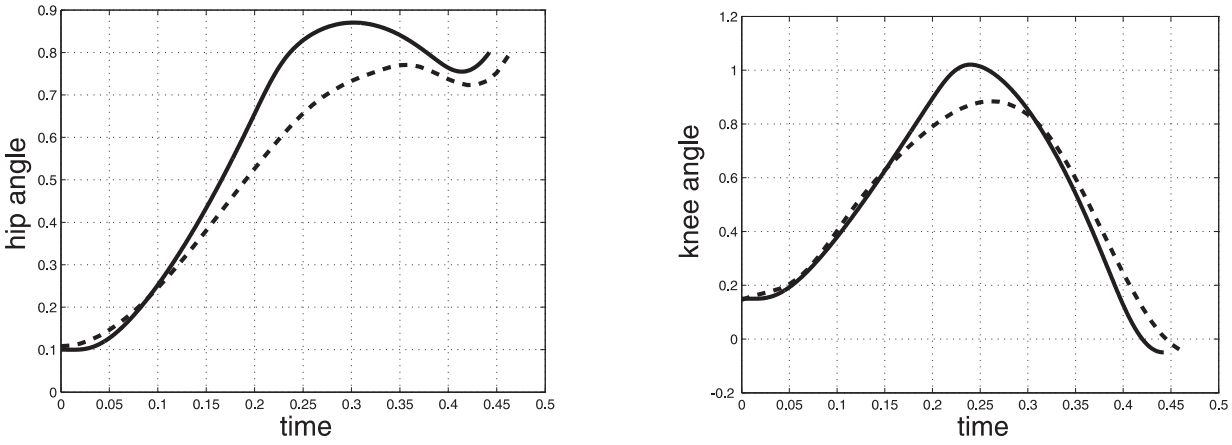


Figure 5.19: Measured (dashed line) and optimized (solid line) joint angle trajectory of hip (left) and knee (right).

5.2.1.2 Analysis of Measured Motion

If not a free, goal oriented motion, shall be predicted and optimized but a measured motion is to be analyzed, the forward dynamics approach may be used in a similar way. The only difference lies in the objective function. Let φ_{hip} and φ_{knee} be the measured hip and knee angle trajectories. Then, to calculate the muscle activations that lead to the measured motion and take into account human motion control which is supposed to minimize the activation effort, the objective function is chosen to be

$$J = \int_0^{t_f} (q_1 - \varphi_{hip})^2 + (q_2 - \varphi_{knee})^2 + c\boldsymbol{u}^T\boldsymbol{u}dt,$$

where c is a weight factor for taking into account human motion control. Note that by minimizing the differences of measured and calculated joint angle trajectories, the optimization result must not exactly match the measured motion like it is the case when applying the inverse dynamics approach. Thus, measurement errors may be implicitly compensated for which avoids that small measurement errors lead to large errors in the computation results like with inverse dynamics simulation and optimization.

The calculated and measured joint angle trajectories now of course better match (cf. Figure 5.21). The results for the activations are given in Figure 5.20. Computation times are about only 20% of that of the free goal oriented motion from the previous section. The reason for this is, that as a starting solution for the joint states, the measured motion may be taken as it is known and involved in the objective function.

Compared to the prediction of the kicking motion, the controls do not reach their boundary values which seems sensible because in this case not the time optimal solution is calculated.

5.2.1.3 Second Order Discretization Approach

The second order approach from Section 4.3 has been tested for the kicking motion. The results are equivalent to those from the standard approach of transforming the differential equations to first order. The number of optimization variables for a time grid of 60 time points was reduced from 829 to 734 compared to the previous subsection. Computation time for this simple example was reduced by about one order of magnitude compared to the standard direct collocation approach. This, however, is a preliminary result because it has been tested in a prototype MATLAB implementation only.

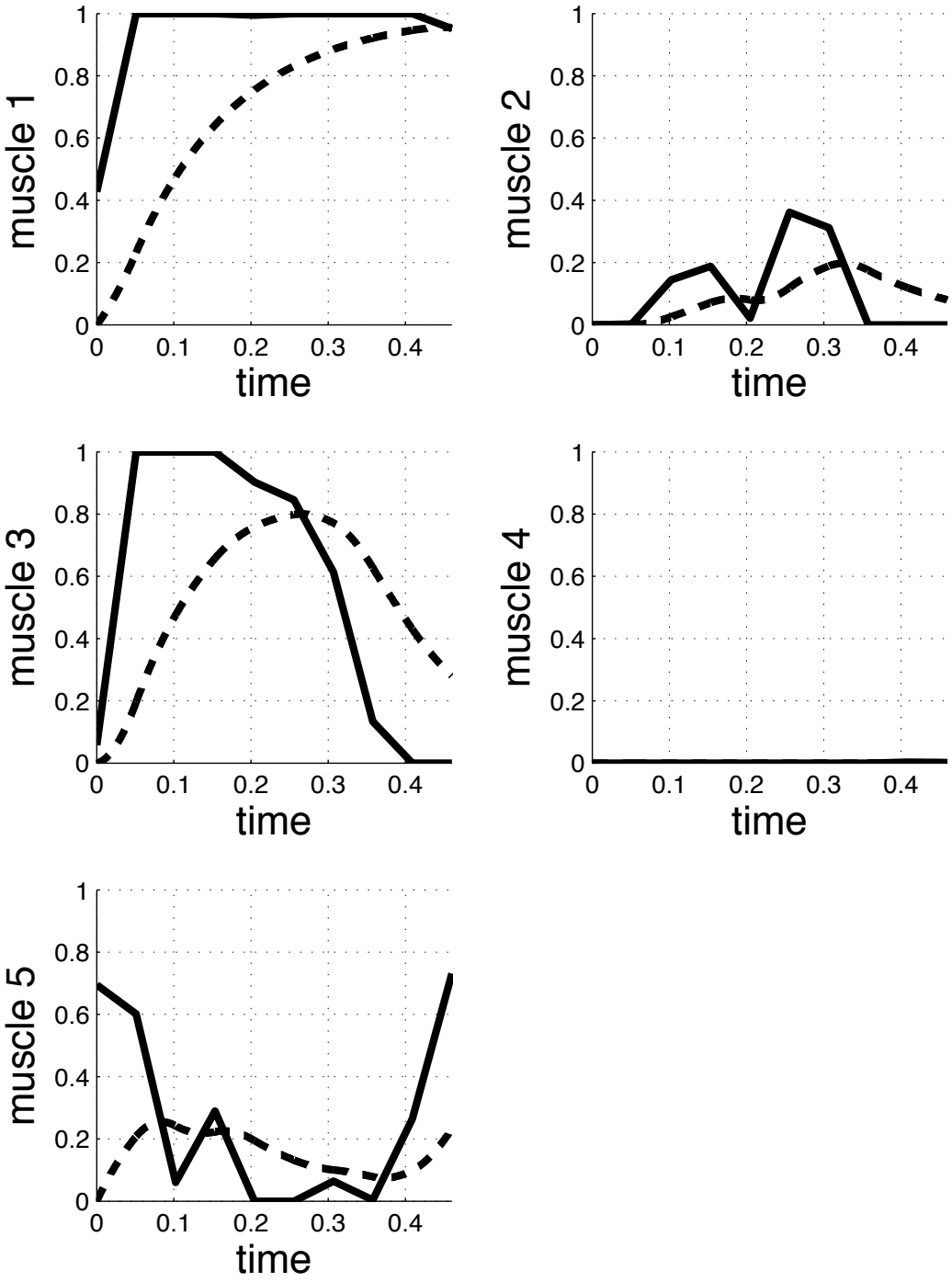


Figure 5.20: Results from optimization for analysis of measured motion: Controls (corresponding to EMG, solid line) and calcium ions concentrations (dashed line).

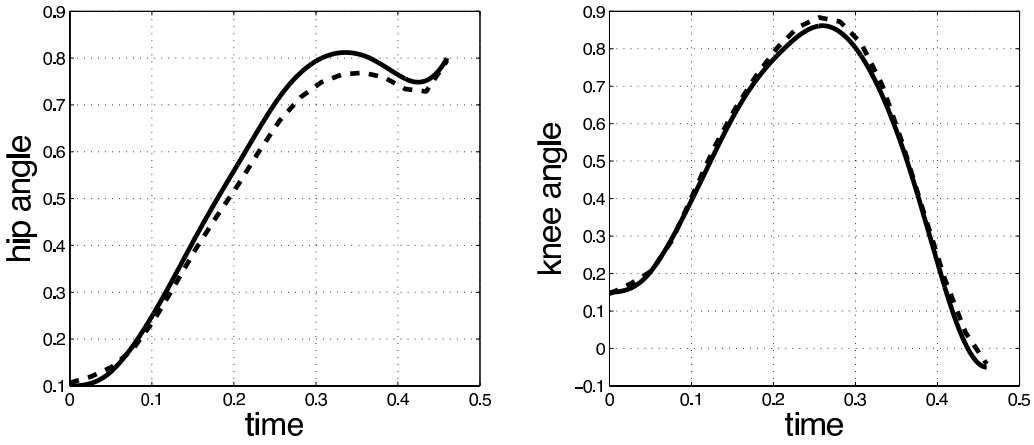


Figure 5.21: Measured (dashed line) and optimized (i.e. reconstructed from analysis; solid line) joint angle trajectory of hip (left) and knee (right).

5.2.2 Jumping Motion

1. Ilio Psoas group
2. Rectus Femoris
3. Vastus group
4. Glutaeus group
5. Hamstring group
6. Biceps Femoris
7. Gastrocnemius group
8. Soleus
9. Tibalis Anterior

q_1 hip angle
q_2 knee angle
q_3 ankle angle

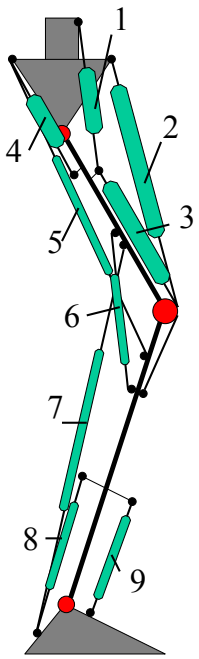


Figure 5.22: Kinematic structure of the leg with nine muscle groups.

A more advanced example that has been investigated is analysis of a human jumping motion. The motion data has been collected by [97], where the motion has been analyzed with a control parameterization approach. Computation there took several days [95]. All motion data and parameters of the model for this section are taken from that work.

The human leg is modeled to be plain and consist of three single-DOF joints and nine muscle groups, see Figure 5.22. The kinematic parameters of the leg are displayed in Figure 5.23.

Besides the kinematical data, the external ground reaction forces (including its point of attack), the motion of the hip (its acceleration) and the muscle forces are needed to compute the forward dynamics of the leg model. The equations of motion may be analytically computed like in [97] or by the recursive dynamics algorithm ABA (Section 3.1).

The motion is made up of three different phases: pre-flight phase, flight phase, and landing phase. The external ground reaction force and its point of attack are displayed in Figure 5.24. During the computations of the muscle activations, it was observed that the high and fast oscillating ground reaction forces in forward direction of landing lead to numerical problems with finer grids. While for rough time grids, they are filtered implicitly, for finer grids, they have been externally filtered (in the most simple case been set to a reasonable maximum force of $100N$).

$$
\begin{aligned}
l_1 &= 0.470m \\
l_2 &= 0.472m \\
l_3 &= 0.075m \\
l_4 &= 0.292m \\
z_1 &= 0.204m \\
z_2 &= 0.202m \\
z_3 &= 0.042m \\
z_4 &= 0.065m \\
m_1 &= 9.000kg \\
m_2 &= 4.185kg \\
m_3 &= 1.305kg \\
I_1 &= 0.207kgm^2 \\
I_1 &= 0.085kgm^2 \\
I_1 &= 0.007kgm^2
\end{aligned}
$$

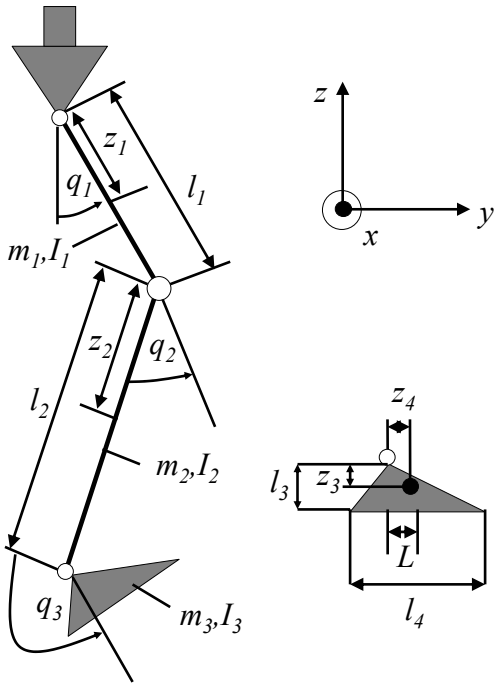


Figure 5.23: Kinematic parameters of the human leg model.

The acceleration of the hip have been gained from interpolation the measured hip position and computing the derivatives of the interpolation. This leads to smooth derivatives, cf. Figure 5.25.

All parameters of the biomechanical part of the model are given in Table 5.3. Activation dynamics according to equation 2.5 is

$$\dot{\gamma}_i = b_{2,i}(u_i - \gamma_i),\ 1 \leq i \leq 9 \tag{5.10}$$

and the activation factor $f_{AD,i}$ is computed according to Equation 2.6

$$f_{AD,i}\left(\gamma_i(u_i)\right) = \frac{(b_{1,i}\gamma(u_i))^3}{1 + (b_{1,i}\gamma(u_i))^3},\ 1 \leq i \leq 9. \tag{5.11}$$

The total muscle force is calculated according to Formulae 2.7, 2.8, 2.9, and 2.10 (note the new numbering of the parameters; some parameters are set fixed values). The parallel elastic element is given by

$$F_i^{PEE}(l_i^M) = k_{2,i}(e^{k_{1,i}(l_i^M - l_0^M)} - 1),\ 1 \leq i \leq 9,$$

the damping element by

$$F_i^{DE}(v_i^M) = k_{3,i}v_i^M,\ 1 \leq i \leq 9.$$

Table 5.3: Parameters for the muscles of the jumping model (taken from [97]).

	$F^{iso}_{max,i}[N]$	$b_{1,i}$	$b_{2,i}[s^{-1}]$	$k_{1,i}[m^{-1}]$	$k_{2,i}[N]$	$k_{3,i}[Nsm^{-1}]$
muscle 1	2800.0	7.24	50.0	15.0	5.85	90.0
muscle 2	900.0	7.24	50.0	15.0	5.40	100.0
muscle 3	5400.0	7.24	50.0	15.0	6.80	100.0
muscle 4	3800.0	7.24	40.0	15.0	9.10	90.0
muscle 5	2400.0	7.24	40.0	15.0	4.10	90.0
muscle 6	1500.0	7.24	30.0	15.0	1.60	70.0
muscle 7	2400.0	7.24	50.0	15.0	8.25	90.0
muscle 8	4200.0	7.24	30.0	15.0	6.50	70.0
muscle 9	1400.0	7.24	30.0	15.0	1.30	70.0

	c_1	c_2	$l^M_{0,i}[m]$	c_3	$c_4[m/s]$	$v^M_{max,i}[m/s]$
muscle 1	0.017	0.0015	0.288	0.27	0.039	-1.1
muscle 2	0.017	0.0015	0.533	0.43	0.045	-0.9
muscle 3	0.017	0.0015	0.331	0.28	0.040	-1.1
muscle 4	0.017	0.0015	0.247	0.24	0.038	-1.2
muscle 5	0.017	0.0015	0.472	0.23	0.025	-0.8
muscle 6	0.017	0.0015	0.280	0.20	0.022	-0.8
muscle 7	0.017	0.0015	0.508	0.27	0.036	-1.0
muscle 8	0.017	0.0015	0.348	0.18	0.023	-0.9
muscle 9	0.017	0.0015	0.282	0.18	0.023	-0.9

	$r_{i,j,0}[m]$	$r_{i,j,1}[mrad^{-1}]$	$r_{i,j,2}[mrad^{-2}]$	$r_{i,j,3}[mrad^{-3}]$	$r_{i,j,4}[mrad^{-4}]$
$i=1, j=1$	0.0420	0.01	0.0	0.0	0.0
$i=2, j=1$	$4.003 \cdot 10^{-2}$	$3.059 \cdot 10^{-2}$	$-2.164 \cdot 10^{-2}$	$-2.807 \cdot 10^{-3}$	0.0
$i=2, j=2$	$6.358 \cdot 10^{-2}$	$3.951 \cdot 10^{-2}$	-0.1793	0.1465	$-3.338 \cdot 10^{-2}$
$i=3, j=2$	$6.358 \cdot 10^{-2}$	$3.951 \cdot 10^{-2}$	-0.1793	0.1465	$-3.338 \cdot 10^{-2}$
$i=4, j=1$	0.0620	0.0	0.0	0.0	0.0
$i=5, j=1$	$4.853 \cdot 10^{-2}$	$3.843 \cdot 10^{-2}$	$-1.932 \cdot 10^{-2}$	$-1.132 \cdot 10^{-2}$	$3.813 \cdot 10^{-3}$
$i=5, j=2$	$2.725 \cdot 10^{-2}$	$4.714 \cdot 10^{-2}$	$-1.990 \cdot 10^{-2}$	$-9.532 \cdot 10^{-3}$	$5.267 \cdot 10^{-3}$
$i=6, j=2$	0.0521	0.0143	0.0	0.0	0.0
$i=7, j=2$	$3.186 \cdot 10^{-2}$	$-4.897 \cdot 10^{-3}$	$-6.119 \cdot 10^{-2}$	$8.377 \cdot 10^{-2}$	$-3.216 \cdot 10^{-2}$
$i=7, j=3$	0.3706	-1.0169	1.1080	-0.4994	$7.886 \cdot 10^{-2}$
$i=8, j=3$	0.3706	-1.0169	1.1080	-0.4994	$7.886 \cdot 10^{-2}$
$i=9, j=3$	0.0327	0.0077	0.0	0.0	0.0

	$a_{0,i}$	$a_{1,i}[\text{deg}^{-1}]$	$a_{2,i}[\text{deg}^{-1}]$	$a_{3,i}[\text{deg}^{-2}]$	$a_{4,i}[\text{deg}^{-1}]$	$a_{5,i}[m]$
muscle 1	0.642	$-1.94 \cdot 10^{-3}$	0.0	0.0	0.0	0.470
muscle 2	1.107	$-1.50 \cdot 10^{-3}$	$1.99 \cdot 10^{-3}$	0.0	0.0	0.470
muscle 3	0.671	0.0	$1.29 \cdot 10^{-3}$	0.0	0.0	0.470
muscle 4	0.494	$-2.06 \cdot 10^{-3}$	0.0	0.0	0.0	0.470
muscle 5	1.021	$2.05 \cdot 10^{-3}$	$-1.87 \cdot 10^{-3}$	0.0	0.0	0.470
muscle 6	0.600	0.0	$1.03 \cdot 10^{-4}$	$-1.21 \cdot 10^{-5}$	0.0	0.470
muscle 7	0.897	0.0	$-5.60 \cdot 10^{-4}$	0.0	$2.14 \cdot 10^{-3}$	0.472
muscle 8	0.563	0.0	0.0	0.0	$1.93 \cdot 10^{-3}$	0.472
muscle 9	0.715	0.0	0.0	0.0	$-1.30 \cdot 10^{-3}$	0.472

	$j=1$	$j=2$	$j=3$
$p_{1,j}[Nm]$	2.8	3.1	3.1
$p_{2,j}[\text{rad}^{-1}]$	5.9	5.9	5.9
$p_{3,j}[Nm]$	8.7	10.5	9.0
$p_{4,j}[\text{rad}^{-1}]$	1.3	11.8	5.0
$\Theta_{1,j}[\text{rad}]$	1.92	0.03	1.92
$\Theta_{2,j}[\text{rad}]$	0.25	-1.92	1.35

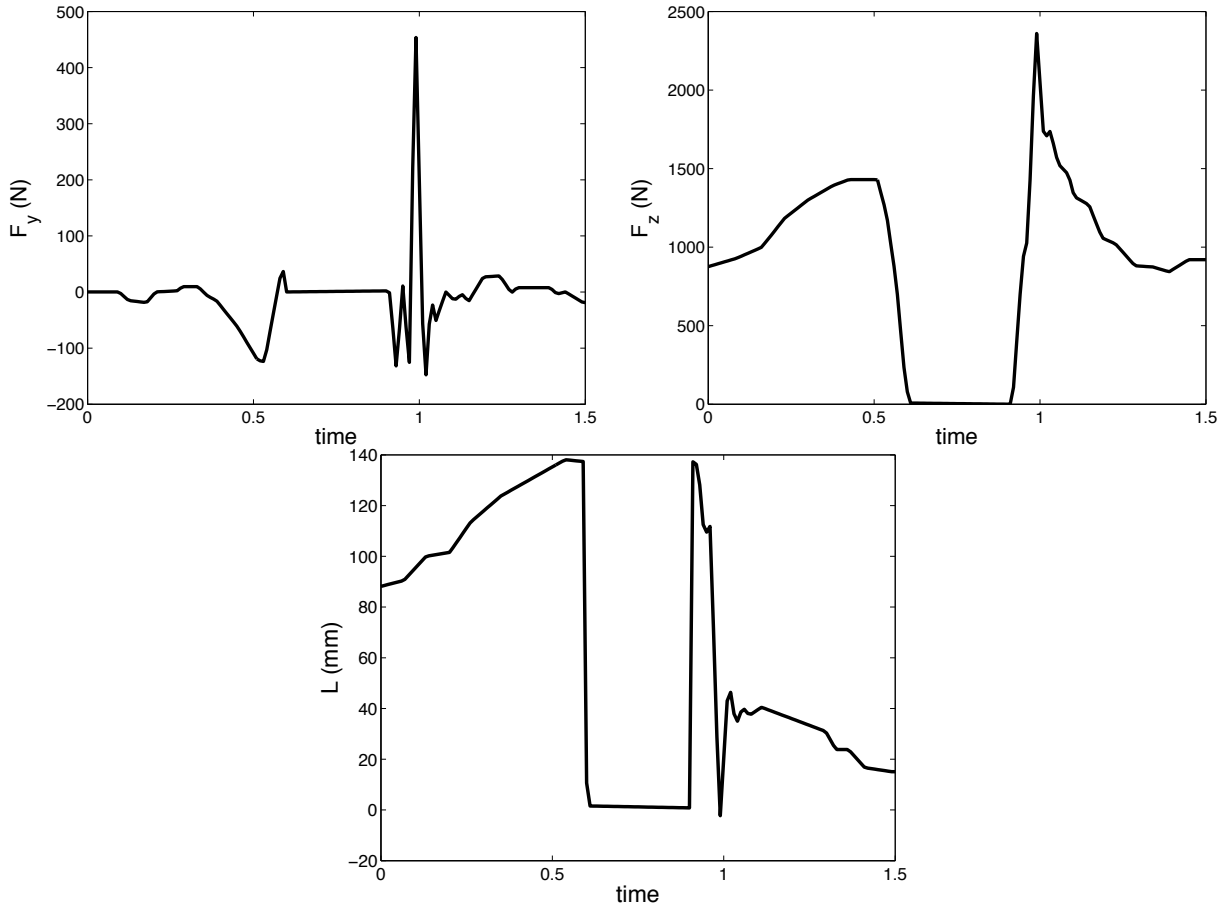


Figure 5.24: Measured ground reaction force (F_y (forward direction) and F_z (upwards direction)) and its point of attack L, cf. Figure 5.23.

The overall force of the muscle tendon element is then

$$F_i^{MTE}(\gamma, l^M, v^M) = F_{\max,i}^{\text{iso}} f_{AD,i}(\gamma_i) f_{TL,i}(l_i^M) f_{FV,i}(v_i^M),\ 1 \le i \le 9,$$

and the total muscle force

$$F_i^M(\gamma_i, l_i^M, v_i^M) = F_i^{MTE}(\gamma_i, l_i^M, v_i^M) + F_i^{PEE}(l_i^M) + F_i^{DE}(v_i^M),\ 1 \le i \le 9.$$

The muscle lengths and velocities are given by

$$l_i^M \quad = \quad a_{5,i}(a_{0,i} + a_{1,i}q_1 + a_{2,i}q_2 + a_{3,i}q_2^2 + a_{4,i}q_3),\ 1 \le i \le 9,$$

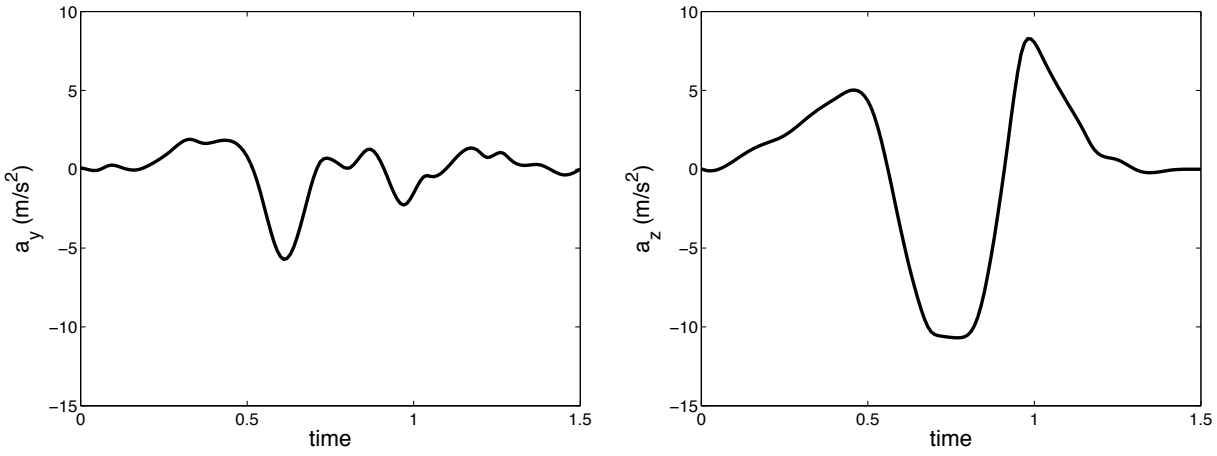


Figure 5.25: Measured hip acceleration (a_y (forward direction) and a_z (upwards direction)).

and

$$v_i^M = a_{5,i}(a_{1,i}\dot{q}_1 + a_{2,i}\dot{q}_2 + 2a_{3,i}q_2\dot{q}_2 + a_{4,i}\dot{q}_3),\ 1 \leq i \leq 9.$$

Two kinds of torques apply to the joints: the active joint torques τ_a resulting from the muscle forces and the passive joint torques τ_p caused by the respective tissue. The lever arms $d_{i,j}$, where i denotes the number of the respective muscle group and j the number of the joint, are given by (note that not all $d_{i,j}$ are needed):

$$d_{i,j} = r_{i,j,0} + r_{i,j,1}q_j + r_{i,j,2}q_j^2 + r_{i,j,3}q_j^3 + r_{i,j,4}q_j^4,\ 1 \leq i \leq 9,\ 1 \leq j \leq 3.$$

Thus, the active torques are given by

$$\begin{aligned}
\tau_{p,1} &= F_1^M d_{1,1} + F_2^M d_{2,1} - F_4^M d_{4,1} - F_5^M d_{5,1},\\
\tau_{p,2} &= F_2^M d_{2,2} + F_3^M d_{3,2} - F_5^M d_{5,2} - F_6^M d_{6,2} - F_7^M d_{7,2},\\
\tau_{p,3} &= F_9^M d_{9,3} - F_7^M d_{7,3} - F_8^M d_{8,3}.
\end{aligned}$$

The passive torques model the maximum joint angle range by a torque that increases once the joint is close to the maximum values. They are given by

$$\begin{aligned}
\tau_{p,1} &= p_{1,1}e^{-p_{2,1}(q_1-\theta_{2,1})} + p_{3,1}e^{-p_{4,1}(\theta_{1,1}-q_1)},\\
\tau_{p,2} &= p_{1,2}e^{\ p_{2,2}(q_2+\theta_{2,2})} + p_{3,2}e^{-p_{4,2}(\theta_{1,2}+q_2)},\\
\tau_{p,3} &= p_{1,3}e^{-p_{2,3}(q_3-\theta_{2,3})} + p_{3,3}e^{-p_{4,3}(\theta_{1,3}-q_3)}.
\end{aligned}$$

The resulting optimal control problem of finding muscle activations $u_i,\ 1 \leq i \leq 9$, that lead to calcium ion concentrations $\gamma_i,\ 1 \leq i \leq 9$ in the muscle and finally to a motion (i.e. computed hip, knee and ankle joint angle trajectories q_1, q_2, q_3) that resembles the measured motion (i.e. measured hip, knee and ankle joint angle trajectories $\varphi_1, \varphi_2, \varphi_3$) may be represented by the following objective function $\mathcal{J}$:

$$\mathcal{J}(u) = \sum_{i=1}^{3} \int_0^{1.5} (\varphi_i(t) - q_i(t))^2 dt + \sum_{i=1}^{9} \omega_i \int_0^{1.5} (u_i(t))^2 dt$$

The first sum of $\mathcal{J}$ makes the computed motion be close to the measured one and the second sum takes into account the muscular effort for achieving this motion and is closely related to the muscle forces. For the computations presented here, $\omega_i = 10^{-3},\ 1 \leq i \leq 9$.

Boundary conditions are imposed to the joint angles and velocities and to the calcium ion concentrations that are necessary to keep the leg standing during the starting phase:

$$\begin{aligned}
q_1(t_0) &= 0.624 \\
q_2(t_0) &= -0.594 \\
q_3(t_0) &= -0.072 \\
\dot{q}_1(t_0) &= 0 \\
\dot{q}_2(t_0) &= 0 \\
\dot{q}_2(t_0) &= 0 \\
\gamma_2(t_0) &= 0.2 \\
\gamma_3(t_0) &= 0.2
\end{aligned}$$

Box constraints are imposed to the controls and states of calcium ion concentration:

$$0 \leq u_i,\ \gamma_i \leq 1,\ 1 \leq i \leq 9.$$

In [97], the optimal control problem was divided into three phases: pre-flight phase, flight phase, landing phase. This was done because of two reasons: first, the ground reaction forces are zero during the flight phase and thus the computation of it consumes computation power where not necessary, and second, the shorter flight phase should be discretized with a finer grid. For the results presented here, only one phase was modeled because the computation time for taking into account the ground reaction force (although zero) is believed to be computationally less expensive than adding additional discretized state and control variables and imposing boundary constraints to the problem. Furthermore, DIRCOL allows for successive grid refinement and

Table 5.4: Size of the resulting NLP for two different numbers of grid points in the discretization of the jumping motion.

grid points	15	40
nonlinear constraints	367	624
nonlinear variables	224	992

thus if more grid points are needed during the flight phase, they will be inserted automatically. Nevertheless, future investigations shall more precisely show, whether splitting up the problem into several phases will help to increase the convergence region of the method.

Computations have started with a grid of 15 time points and have successively been refined to a grid of 40. Details on the number of constraints and variables may be found in Table 5.4. Computation times depend on when the grids are refined. Furthermore, the solution may not be obtained by a single run of the optimization method but needs careful insertion of critical parts of the model (especially the passive joint torques, muscle lengths and velocities and lever arms of the muscles) one after one to allow the method to converge to a feasible and optimal solution. Future extensions of the methods should use automated homotopy methods for optimal control to allow computation with less supervision by the user. The total computation time lies in the region of few hours on a 1700+ Athlon XP for a time grid of 40 points. Compared to the method used in [97], where computation times in the region of days have been reported for a time grid of 16 points [95], and taking into account the progress of computers since then, this gives a speedup in the region of what it has been observed for the kicking motion (where one single run of the method leads to a final result).

The measured and computed joint angle trajectories are plotted in Figure 5.26, where good quantitative agreement is achieved. The measured EMG data and the computed muscle activations are given in Figures 5.27 and 5.27. Of course, the computed muscle activations can not resemble the unfiltered EMG data. Nevertheless, a reasonable agreement may be observed between the regions of activity and the computed results from [97]. The extension of the leg during the pre-lift-off-phase is a result of the activations of muscle groups 2 and 3 (Rectus Femoris and Vastus). For lift-off, especially muscle groups 7 and 8 (Soleus and Tibalis Anterior) are excited. During the flight phase (from $0.6s$ to $0.91s$), the muscle activations are generally low due to the absent of external forces. For landing especially muscle groups 2 and 3 (Rectur Femoris and Vastus) are activated.

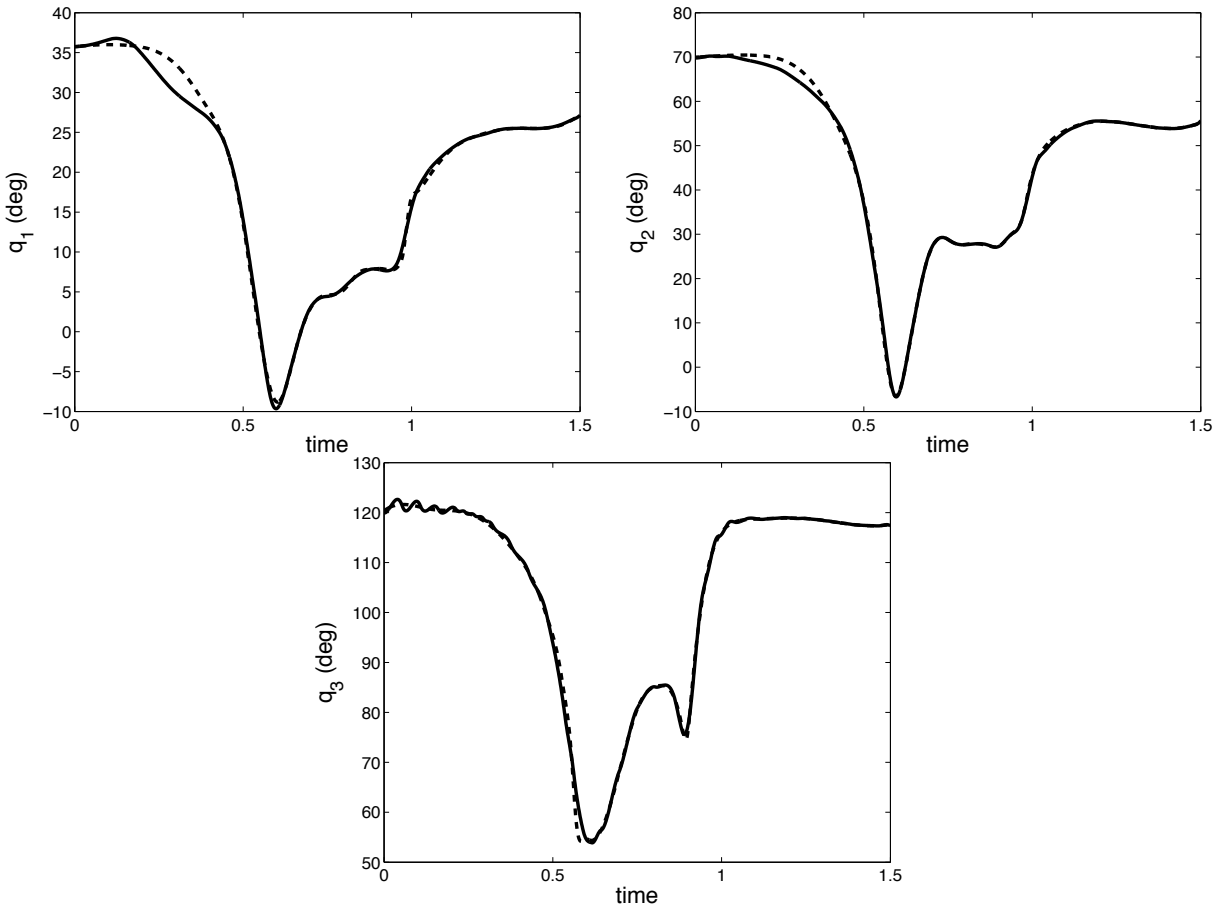


Figure 5.26: Measured (dashed) and computed (solid line) joint angle trajectories for the jumping motion.

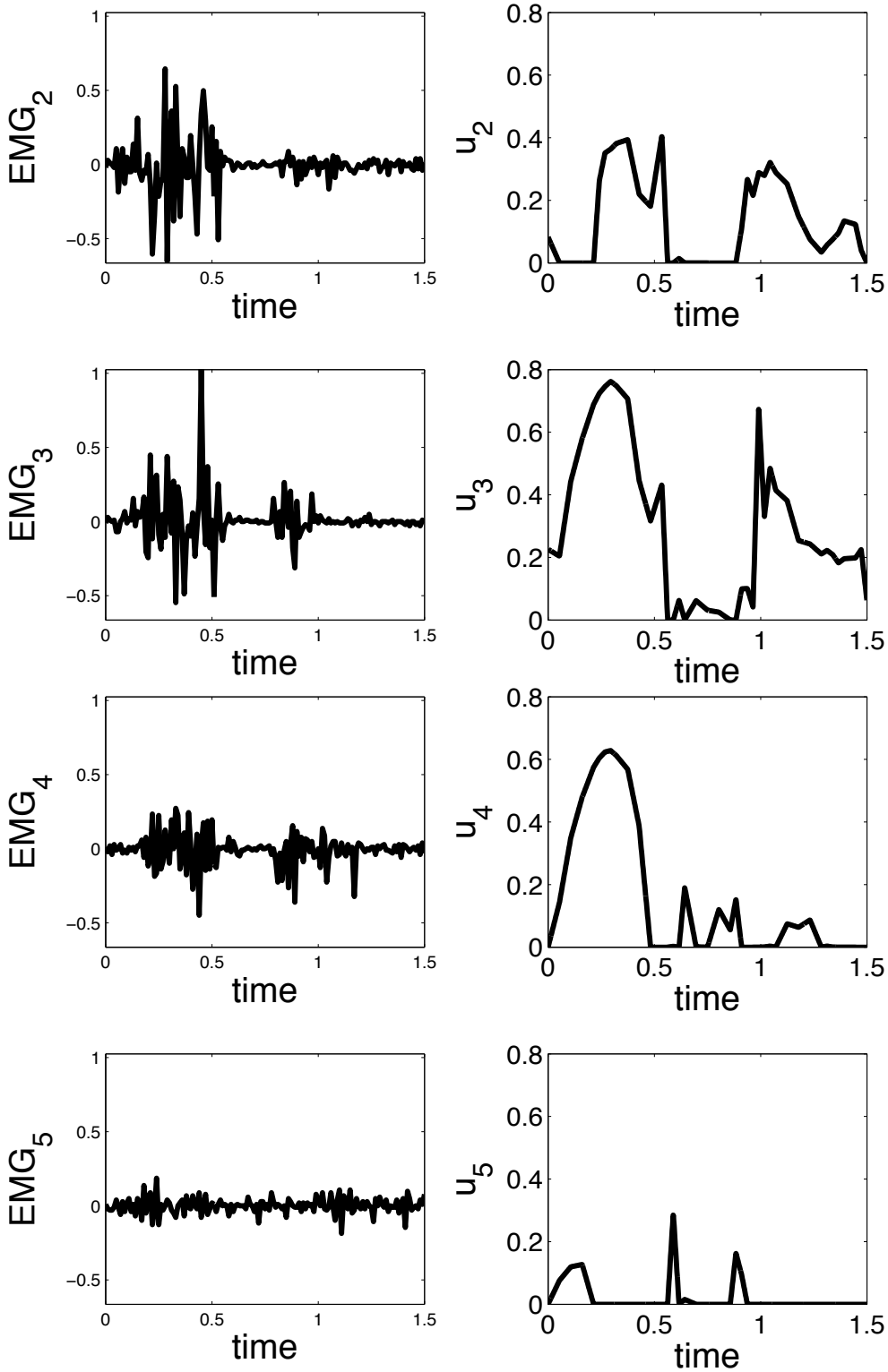


Figure 5.27: EMG data (left) and computed muscle activation (right) for the muscle groups 2-5. EMG data for the Ilio Psoas group is not available using surface EMG and has thus not been measured. Note the different scale of EMG and computed data.

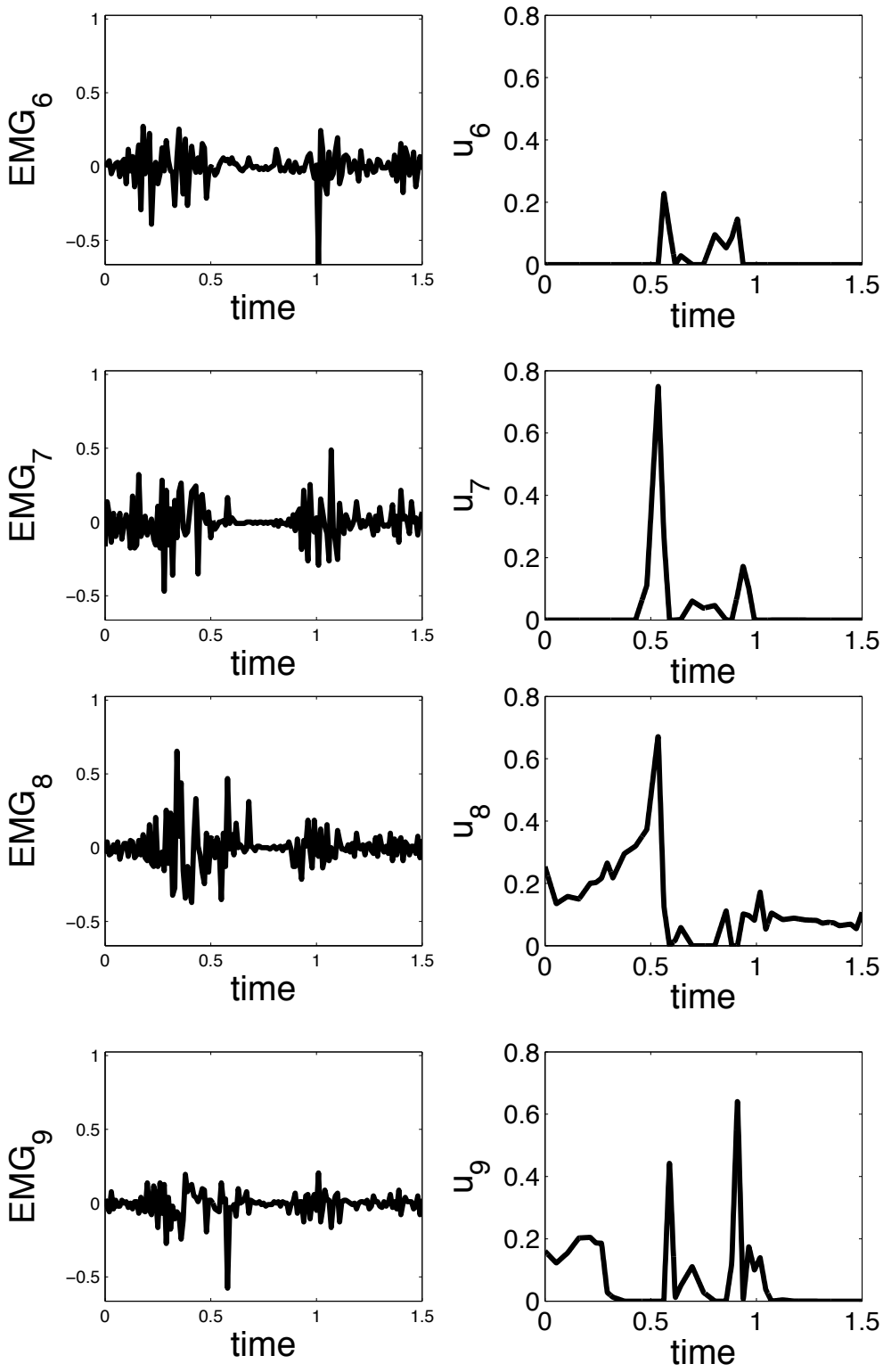


Figure 5.28: EMG data (left) and computed muscle activation (right) for the muscle groups 6-9. Note the different scale of EMG and computed data.

5.3 Identification of Objective Functions for Muscle Control in Human Motion

Predicting human motion (like the kicking motion of Section 5.2.1.1) besides the motion target to reach requires an objective function concerning how the redundant muscles are involved into the motion. Once this objective function is determined for a certain test person and a certain kind of motion, motions really may be predicted.

Moreover, objective functions that include different involvement of each single muscle (group) may give information about training status of certain muscle groups or might identify problems with that specific muscles even before they become obvious to the test person. Change of the objective functions may judge training success for specific muscles.

The problem of identifying this objective function can be formulated as optimization problem, see Section 4.4. In this section some first examples of identifying the the objective functions for a kicking motion (Section 5.2.1) using the methods described in Section 4.4.3 are given.

5.3.1 Re-Identification from Computed Data

First numerical results have been obtained from computing optimal motion using certain objective functions. To have an objective function that purely is based on the behavior of the muscles, this functions were chosen to be the sum of squared controls, i.e., the muscle activations which are closely related to the muscle forces:

$$\mathcal{J}_i(u) = \sum_{j=1}^{5} \int_0^{t_f} \omega_{i,j} u_i(t)^2 dt.$$

The following options for $\boldsymbol{\omega_i}$ were investigated, where $*$ denotes, that the value was kept fixed to 1, i.e. has not been subject to identification:

$$\begin{aligned}
\boldsymbol{\omega_1} &= (0.2, *, *, *, *)^T, \\
\boldsymbol{\omega_2} &= (0.2, 1, *, *, *)^T, \\
\boldsymbol{\omega_3} &= (0.2, 1, 1, *, *)^T, \\
\boldsymbol{\omega_4} &= (0.2, 1, 1, *, 1)^T.
\end{aligned}$$

Muscle group number four is not activated during the complete motion and thus does not influence the motion. Obviously its contribution factor thus can not be re-identified.

Table 5.5: Results of re-identification from computed motion data.

	$i = 1$	$i = 2$	$i = 3$
$\boldsymbol{\omega_i}$	$(0.2, 1, *, *, *)^T$	$(0.2, 1, 1, *, *)^T$	$(0.2, 1, 1, *, 1)^T$
$\boldsymbol{\omega_i^*}$	$(0.200293, 1, *, *, *)^T$	$(0.18954, 1, 1, *, *)^T$	$(0.21306, 1, 1, *, 1)^T$
$\mathcal{I}(\boldsymbol{\omega_i^*})$	$3.8 \cdot 10^{-10}$	$4.3 \cdot 10^{-8}$	$6.3 \cdot 10^{-8}$

The resulting computed joint angle trajectories $q_1(t), q_2(t)$ are displayed in Figure 5.31 for different objective functions.

The identification problem according to 4.13 has been solved. For first results, the black-box approach has been used. Table 5.5 shows the computation results. The weights identified from the computed motion match the weights used for the computations very well, i.e., the objective function was very well re-identified from the computed data.

For additional tests, the computed motion data were modified by adding Gaussian noise of mean value 0 and standard deviation $5\,\mathrm{deg}$. Because of the high noise that was added, several approximately equally good minima are found by the identification method. The best among them re-identified the values of $(0.2, 1, 1, *, 1)^T$ to $(0.26183, 0.90982, 1.00000, *, 0.99996)^T$ and delivered a value of the identification merit function $\mathcal{I}$ of $1.0 \cdot 10^{-5}$. Please note that for evaluation of the identification objective function $\mathcal{I}$ the values of the computed and measured data at the grid points only have been used. Figure 5.29 shows the noisy computed motion data and one solution of the identification method.

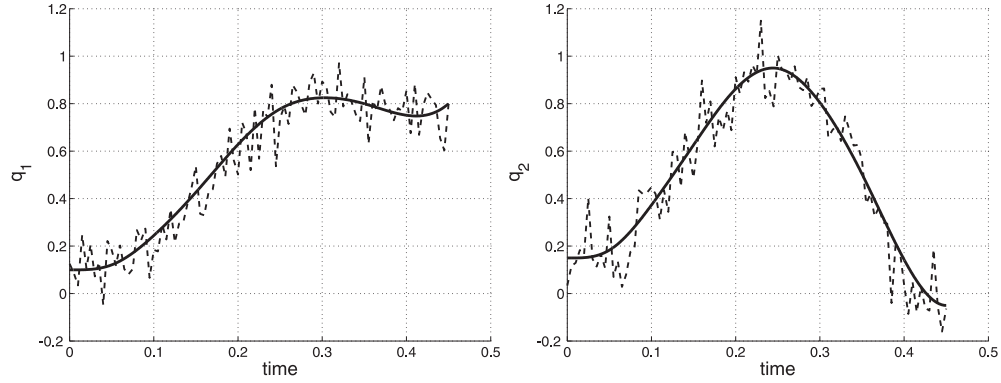


Figure 5.29: Noisy computed hip and knee joint angle trajectories (dashed) and identified trajectories (solid).

Further investigations showed that the approach based on sensitivity analysis of the discretized problem lead to singular matrices. Thus, for further investigations two options exist: Overcome the singularities or handle the problem with another method, e.g. the one from [14].

5.3.2 Test with Identification from Experimental Data

Now the objective function is identified from measured motion data which has been used in Sections 5.2.1.1 resp. 5.2.1.2 for comparison with the predicted motion resp analysis. The identified objective function is

$$\mathcal{J}(u) = \int_0^{t_f} (0.000000u_1(t)^2 + 0.97920u_2(t)^2 + 0.49935u_3(t)^2 + u_4(t)^2 + 1.000000u_5(t)^2)dt.$$

The identification objective function $\mathcal{I}$ was $1.6 \cdot 10^{-3}$. The degree of agreement is comparable to that of the analysis of measured motion, cf. Section 5.2.1.2 and Figure 5.30. Of course, these identified values can not be validated directly. One possible indirect approach is to identify the objective function for certain motions of a specific test person and then compare other motions of the same person with results from numerical prediction of the motion using the identified objective function.

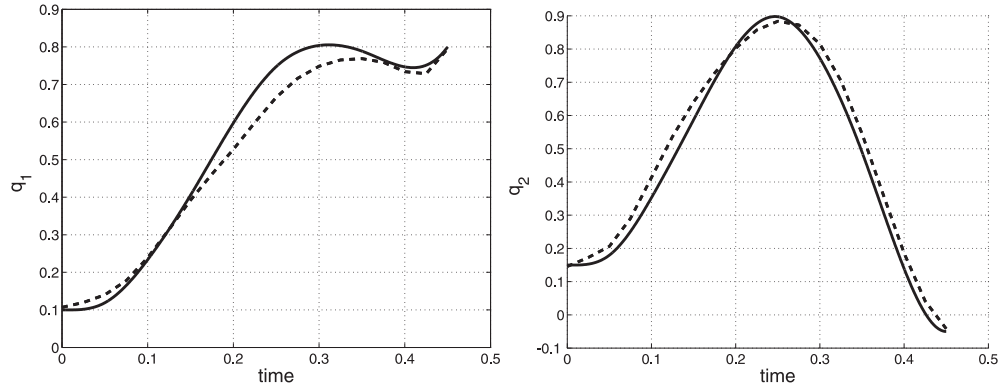


Figure 5.30: Measured hip and knee joint angle trajectories (dashed) and identified trajectories (solid).

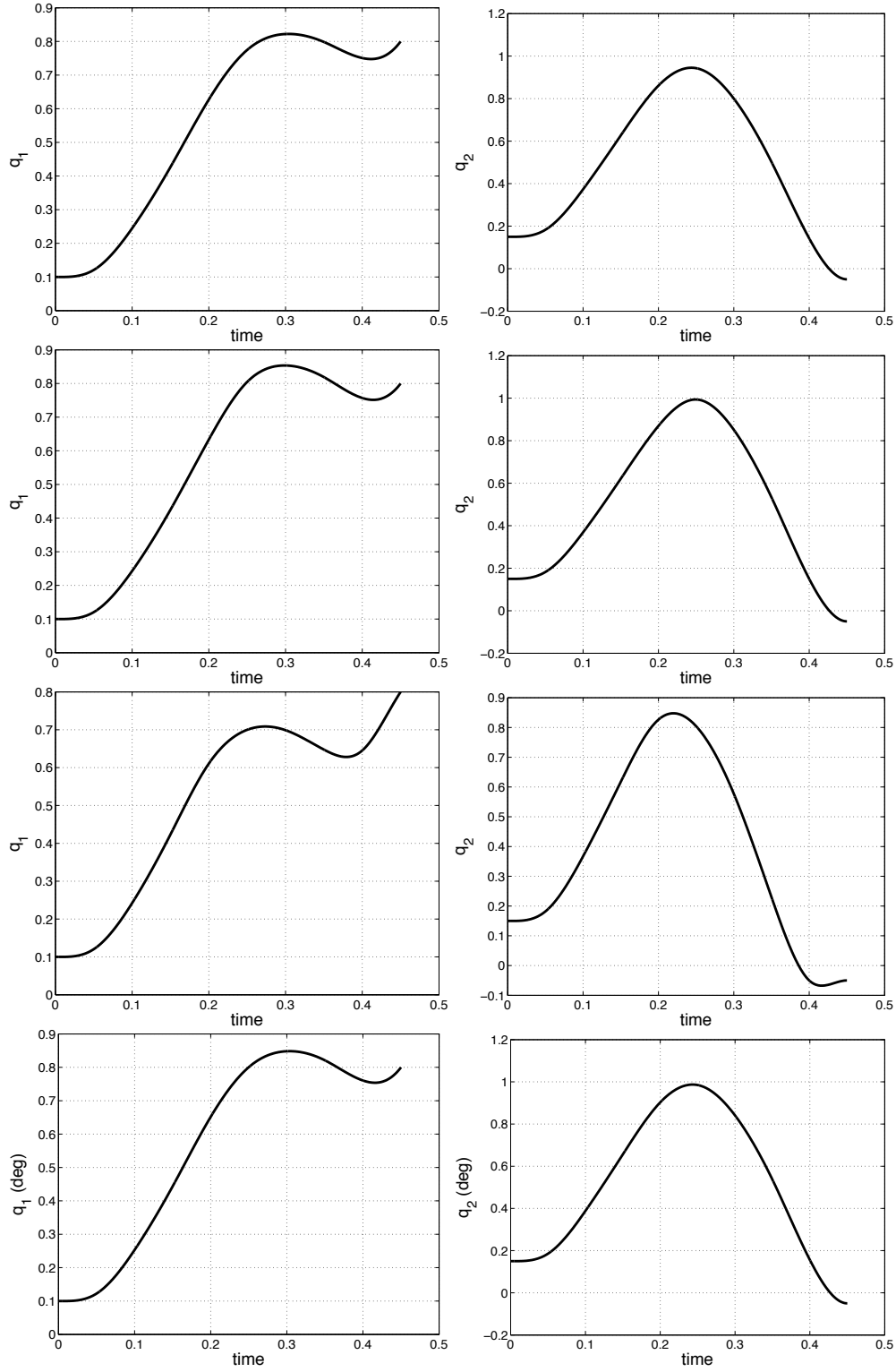


Figure 5.31: Computed hip and knee joint angle trajectories for different objective functions (from top to bottom: $\omega = (0.2, 1, 1, 1, 1)^T$, $(1, 0.2, 1, 1, 1)^T$, $(1, 1, 1, 0.2, 1)^T$, $(1, 1, 1, 1, 0.2)^T$, hip left, knee right).

Chapter 6

Summary and Conclusion

In this thesis, forward dynamics simulation and optimization of legged robot and of human motion has been investigated.

Models of motion dynamics lead to systems of (ordinary) differential equations. An recursive articulated body algorithm has been utilized for an efficient $O(n)$ modeling of the multibody system dynamics of tree-structured legged robots. An extension for biomechanical problems has been developed by replacing the motor-gear by muscle-tendon units. Both active properties of the muscle like force-velocity and force-length relation and passive properties of the tissue and its influence to the joint torques are modeled, as well as muscle paths and the resulting lever arms. The chemical process of activation dynamics is taken into account by additional differential equations. The extended dynamics algorithm has been implemented in an object-oriented manner. It is superior to other algorithms in terms of computational time for systems with a high number of degrees of freedom and modularity, i.e., the exchange of components.

Simulation-based optimization problems for motions of legged robots or humans like walking result in optimal control problems, where the control variable trajectories are to be determined in a manner that the motion from an initial to a final position is optimal with respect to a certain objective like time or energy. Besides the differential equations describing the dynamic behavior of the multibody system, proper constraints like boundary conditions and nonlinear inequality constraints have to be considered in the optimization problem formulation to obtain reasonable results. For human biodynamical systems not only the motion dynamics model is usually larger than for comparable robotic systems, also the degrees of freedom of the optimization problem is much higher due to the many, redundant muscle groups involved in a leg or arm motion. While the inverse dynamics simulation and optimization of human motion dynamics is well established and efficient for special objectives for muscle control, the forward dynamics simulation and optimization approach is yet cumbersome and suffers from extraordinary high computational cost.

However, it was chosen here because of its generality concerning different objective functions and activation dynamics, its implicit filtering properties for the analysis of measured motion, and its principle capability of predicting goal oriented motion if validated models are used. For future applications of validation of hypotheses in biomechanics, first steps towards an optimization technique for identification of unknown objectives for muscle control as a linear combination of elementary types of objectives have also been taken.

For efficient numerical solution of the optimal control problems for forward dynamics simulation and optimization, a direct collocation approach is used which avoids repeated integration of the differential equations of motion by discretizing both the control and the state variables. A tailored approach directly treats the systems of second order multibody system dynamics equations, avoids the inefficient transformation to first order systems required by general numerical optimal control methods, and thus further decreases computational time.

Several problems of simulation-based motion optimization for legged robots and humans have been investigated through proper modeling and successful application of the developed optimization approach. First, the walking motion of a four-legged Sony AIBO ERS-210 robot was optimized using a forward dynamics simulation model based on data partially provided by the manufacturer and partially estimated from repeated experiments. Finally, a fast and stable, upright walking motion was achieved in good agreement of simulated and experimental robot motion. Second, stable walking motions have also been obtained for a slow gait for a humanoid robot using the described methods. Implementation of the computed walking trajectories lead to a stable walking experiment of the 80 cm high prototype using only joint angle PID-control and without using additional sensors for monitoring and maintaining postural stability.

As first examples for the application to human biodynamics, kicking and jumping motions of a human leg have been investigated in terms of analysis for a measured motion and of prediction assuming a reasonable objective function. The muscle activations in the first case and the resulting motion in the second case very well resembled data found in literature and measured data. First steps towards identification of the objective functions used in muscle control have been taken by re-identifying the objective functions from (artificially disturbed) simulated measurement data, and identifying the objective function from a measured kicking motion.

The computational costs for the biomechanics scenarios investigated were reduced by two orders of magnitude compared to commonly used methods due to the applied efficient methods for dynamics modeling and optimal control. As a consequence, it may now become feasible to use forward dynamics simulation and optimization of human motion dynamics in the near future on desktop PCs and in an interactive manner. Thus, many new applications in biodynamics research may be opened which may have been thought intractable before, like validation of biomechanical hypotheses or calibration of biomechanical model parameters.

Future work might include consideration of fatigue for repeated motion. Furthermore, introduction of wobbling masses and a detailed foot model will be needed for accurate investigation of fast motions. More complex examples can be investigated; for this, detailed data of not only the motion but also the test person must be collected. Automated homotopy methods will be useful to enlarge the convergence region of the optimization approach. The developed identification method for muscle control objectives can be improved by methods based on solving an additional optimization problem to obtain the sensitivity information. Identification methods in combination with dynamics simulation and optimization can be used in several other applications like offline or online recognition and classification of human motion. The dynamics algorithms can be extended to the computation of derivative information.

As the forward dynamics simulation and optimization methods are getting fast and if they will also be robust enough to solve problems of medium or large size on common desktop computers, several new applications in medicine and biology will arise.

Chapter 7

Zusammenfassung

In dieser Arbeit wurde die Vorwärtsdynamiksimulation und -optimierung von Bewegungen bei Laufrobotern und Menschen untersucht.

Modelle für die Dynamik von Bewegungen führen auf Systeme von (gewöhnlichen) Differentialgleichungen. Der rekursive „Articulated Body" Algorithmus, dessen Aufwand linear mit der Anzahl der Gelenke anwächst, wird für die effiziente Modellierung der Mehrkörpersystemdynamik von baumstrukturierten Robotern eingesetzt. Die Modellierung wurde vom Einsatz für Roboter erweitert auf den Einsatz bei biomechanischen Problemen; dazu wurden die Motor-Getriebe-Einheiten durch Muskel-Sehnen-Einheiten ersetzt. Sowohl die aktiven Eigenschaften des Muskels, wie die Kraft-Geschwindigkeit- und die Kraft-Längen-Beziehungen, als auch die passiven Eigenschaften des Gewebes und dessen Einfluß auf die Gelenkdrehmomente werden modelliert. Die Muskelpfade und die resultierenden Hebelarme werden berücksichtigt. Der chemische Prozess, der im Muskel zur Kraftentfaltung führt, wird als Aktivierungsdynamik durch zusätzliche gewöhnliche Differentialgleichungen berücksichtigt. Der erweiterte Dynamikalgorithmus, der verglichen mit anderen Algorithmen insbesondere für hochdimensionale Systeme mit mehr als sechs Freiheitsgraden sehr effizient und dabei modular im Hinblick auf die Austauschbarkeit von Modellkomponenten ist, wurde objektorientiert in C++ umgesetzt.

Simulationsbasierte Optimierungsprobleme für Bewegungen von Laufrobotern oder Menschen, wie beispielsweise Gehen, führen auf Optimalsteuerungsprobleme, bei denen die Trajektorien der Steuervariablen so zu bestimmen sind, dass die Bewegung von einer Anfangs- zu einer Endposition optimal bezüglich einer gegebenen Zielfunktion, z.B. Zeit oder Energie, ist. Neben den Differentialgleichungen, die das dynamische Verhalten des Mehrkörpersystems beschreiben, müssen geeignete Nebenbedingungen wie Randbedingungen und nichtlineare Ungleichungsbeschränkungen im Optimalsteuerungsproblem berücksichtigt werden, um sinnvolle Ergebnisse zu erhalten. Für menschliche biodynamische Systeme ist nicht nur das Bewegungsdynamik-

modell normalerweise größer als für vergleichbare Robotersysteme, sondern auch die Anzahl der Freiheitsgrade im Optimierungsproblem ist durch die vielen, redundant an Arm- oder Beinbewegungen beteiligten Muskelgruppen viel größer. Während die Inversdynamiksimulation und -optimierung menschlicher Bewegungsdynamik für spezielle Gütekriterien zur Muskelsteuerung bewährt und effizient ist, ist die Vorwärtsdynamiksimulation und -optimierung durch den außerordentlich hohen numerischen Aufwand immer noch schwerfällig. Trotzdem wurde der Vorwärtsansatz wegen seiner Allgemeinheit in Bezug auf verschiedene Zielfunktionale und die Aktivierungsdynamik, seine implizit-filternden Eigenschaften für die Analyse von gemessenen Bewegungen und seiner prinzipiellen Fähigkeit, Bewegungen unter Verwendung von validierten Modellen vorherzusagen, gewählt. Für zukünftige Anwendungen, wie die Validierung von Hypothesen der Biomechanik, wurden erste Schritte hin zu Optimierungsverfahren zur Identifikation von unbekannten Zielfunktionalen für die Muskelsteuerung als Linearkombination von Elementarzielfunktionalen unternommen.

Zum effizienten numerischen Lösen des Optimalsteuerungsproblems der Vorwärtsdynamiksimulation und -optimierung wird ein direkter Kollokationsansatz verwendet, der die wiederholte Integration der Bewegungsdifferentialgleichungen durch Diskretisierung sowohl der Steuerungen als auch der Zustände vermeidet. Ein angepasster Ansatz behandelt die Mehrkörpersystemdynamikgleichungssysteme zweiter Ordnung direkt, vermeidet so die ineffiziente Transformation auf Systeme erster Ordnung, die von allgemeinen numerischen Optimalsteuerungsverfahren benötigt werden, und reduziert dadurch den Rechenaufwand weiter.

Mehrere Szenarien zur simulationsbasierten Optimierung bei Laufrobotern und Menschen wurden durch geeignete Modellierung und Anwendung des entwickelten Optimierungsansatzes erfolgreich untersucht. Zuerst wurde die Laufbewegung des vierbeinigen Roboters Sony AIBO ERS-210 unter Verwendung eines Vorwärtsdynamikmodells, das auf teilweise vom Hersteller gelieferten und teilweise auf aus wiederholten Experimenten geschätzten Daten basiert, optimiert. Damit konnte eine schnelle und stabile, aufrechte Laufbewegung mit guter Übereinstimmung zwischen simulierter und experimenteller Roboterbewegung berechnet werden. Mit den beschriebenen Methoden wurden stabile Laufbewegungen auch für ein langsames Gehen eines humanoiden Roboters erzielt. Die Implementierung der berechneten Laufbewegung führte zu einer stabilen Laufbewegung im Experiment mit dem 80 cm hohen Prototypen unter Verwendung von nur einer Gelenkwinkel PID-Regelung und ohne zusätzliche Sensoren zur Überwachung oder Regelung der Stabilität der Laufbewegung.

Als erste Beispiele für die Anwendung auf menschliche Biodynamik wurden eine Kick- und eine Sprungbewegung eines menschlichen Beines untersucht, sowohl im Hinblick auf die Analyse einer gemessenen Bewegung als auch für die Vorhersage unter der Annahme eines sinnvollen Zielfunktionals. Die Muskelaktivierungen für den ersten Fall und die resultierende Bewe-

gung für den zweiten Fall stimmen sehr gut mit Daten aus der Literatur und gemessenen Daten überein. Erste Schritte hin zur Identifizierung des Zielfunktionals, das in der Muskelsteuerung Anwendung findet, wurden durch die Re-Identifizierung des Zielfunktionals von (künstlich verrauschten) simulierten Bewegungsdaten und durch die Identifikation des Zielfunktionals für eine gemessene Kickbewegung gemacht.

Der Rechenaufwand für die untersuchten biomechanischen Szenarien konnte im Vergleich zu herkömmlichen Verfahren durch die angewandten effizienten Methoden zur Dynamikmodellierung und Optimalsteuerung um zwei Größenordnungen reduziert werden. Folglich scheint es machbar, in naher Zukunft die Vorwärtsdynamiksimulation und Optimierung der menschlichen Bewegungsdynamik auf Arbeitsplatzrechnern und in interaktiver Art und Weise einzusetzen. So können neue Anwendungen in der Biodynamikforschung erschlossen werden, die zuvor unlösbar erschienen, wie die Validierung biomechanischer Hypothesen oder die Kalibrierung von Parametern biomechanischer Modelle.

Zukünftige Arbeiten könnten die Betrachtung von Ermüdungserscheinungen für wiederholte Bewegungen enthalten. Die Einführung von Schwabbelmassen und eines detaillierten Fußmodells wird nötig sein, um sehr schnelle Bewegungen genau untersuchen zu können. Komplexere Beispiele können untersucht werden, wenn dafür detaillierte Daten nicht nur der Bewegung sondern auch der Testperson aufgenommen werden. Automatisierte Homotopieverfahren werden hilfreich sein, um den Konvergenzradius des Optimierungsansatzes zu vergrößern. Das entwickelte Identifizierungsverfahren für die Gütekriterien der Muskelsteuerung kann durch Verfahren, die auf der Lösung eines zusätzlichen Optimierungsproblems zur Berechnung der Ableitungsinformationen basieren, verbessert werden. Identifizierungsverfahren in Kombination mit Dynamiksimulation und -optimierung können in zahlreichen weiteren Anwendungen wie offline- oder online-Erkennung und -Klassifizierung von menschlichen Bewegungen eingesetzt werden. Die Dynamikalgorithmen können auf die Berechnung von Ableitungsinformationen erweitert werden.

Mit den Entwicklungen in dieser Arbeit konnte gezeigt werden, dass die Vorwärtsdynamiksimulations- und -optimierungsverfahren noch beschleunigt werden können; wenn sie nun auch bei Problemen mittlerer oder hoher Größe robust genug sind, um sie auf Arbeitsplatzrechnern einzusetzen, werden sich zahlreiche neue Anwendungen in Medizin und Biologie auftun.

Bibliography

[1] URL: http://www.nlm.nih.gov/research/visible, [accessed 11 April 2007].

[2] URL: http://www.anybodytech.com/, [accessed 11 April 2007].

[3] URL: http://www.robocup.org, [accessed 11 April 2007].

[4] M. A. Abramson. *Pattern Search Filter Algorithms for Mixed Variable General Constrained Optimization Problems*. PhD thesis, Rice University, 2002.

[5] R. M. Alexander. The gaits of bipedal and quadrupedal animals. *The International Journal of Robotics Research*, 3(2):49–59, 1984.

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

[7] 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.

[8] F. C. Anderson and M. G. Pandy. Dynamic optimization of human walking. *J. of Biomechanic Engineering*, 123:381–390, 2001.

[9] 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.

[10] T. Angeli, M. Gföhler, T. Eberharter, P. Lugner, L. Rinder, and H. Kern. Optimization of the pedal path for cycling powered by lower extremity muscles activated by functional electrical stimulation. In J. Middleton, M. Jones, N. Shrive, and G. Pande, editors, *Computer Methods in Biomechanics and Biomedical Engineering-3*. Gordon and Breach, 2001.

[11] J. Betts. *Practical Methods for Optimal Control Using Nonlinear Programming*. SIAM, 2001.

[12] C. Büskens. *Optimierungsmethoden und Sensitivitätsanalyse für optimale Steuerprozesse mit Steuer- und Zustandsbeschränkungen*. PhD thesis, Fachbereich Mathematik und Informatik, Universität Münster, Germany, 1998.

[13] M. Buss, M. Hardt, J. Kiener, M. 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 International Conference on Humanoid Robots*. Springer-Verlag, Sep.30 - Oct 3 2003.

[14] T. Butz. *Optimaltheoretische Modellierung und Identifizierung von Fahrereigenschaften (Optimal Control Models and Parameter Estimation of Vehicle Driver Behaviors)*. Fortschritt-Berichte VDI, Technische Universit"at Darmstadt, Darmstadt, Germany, December 3 2004.

[15] Y.-C. Chen. Solving robot trajectory planning problems with uniform cubic b-splines. *Optimal Control Applications and Methods*, 12:247–262, 1991.

[16] C. K. Chow and D. H. Jacobson. Studies of human locomotion via optimal programming. *Mathematical Biosciences*, 10:239–306, 1971.

[17] J. J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison Wesley Publishing Company, 2nd edition, 1989.

[18] I. Dahm, U. Düffert, J. Hoffmann, M. Jüngel, M. Kallnik, M. Lötzsch, M. Risler, T. Röfer, M. Stelzer, and J. Ziegler. German Team 2003. Team Description Paper. Technical report, HU Berlin, U Bremen, TU Darmstadt, U Dortmund, 2003. 4 Seiten.

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

[20] C. de Boor. On calculating with B-splines. *Journal of Approximation Theory*, 6:50–62, 1972.

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

[22] R. Featherstone. *Robot Dynamics Algorithms*. Kluwer Academic Publishers, 1987.

[23] Y. C. Fung. *Biomechanics*. Springer-Verlag, New York, 1981.

[24] B. A. Garner and M. G. Pandy. Estimation of musculotendon properties in the human upper limb. *Annals of Biomedical Engineering*, 31:207–220, 2003.

[25] B. A. Garner and M. G. Pandy. The obstacle-set method for representing muscle paths in musculoskeletal models. *Computer Methods in Biomechanics and Biomedical Engineering*, 3:1–30, 2000.

[26] B. A. Garner and M. G. Pandy. Musculoskeletal model of the upper limb based on the visible human male dataset. *Computer Methods in Biomechanics and Biomedical Engineering*, 4:93–126, 2001.

[27] B. A. Garner and M. G. Pandy. A kinematic model of the uppe limb based on the visiblie human project (VHP) image dataset. *Computer Methods in Biomechnics and Biomedical Engineering*, 2:107–124, 1999.

[28] H. S. Gasser and A. V. Hill. The dynamics of muscular contraction. *Proceedings of the Royal Society of London*, 96 B:398–437, 1924.

[29] M. Gföhler, T. Angeli, and P. Lugner. Optimal control of cycling by means of functional electrical stimulation - a dynamic simulation study. In *VIII International Symposium on Computer Simulation in Biomechanics July 4th-6th 2001, Milan, Italy*, 2001.

[30] M. Gienger, K. Löffler, and F. Pfeiffer. Towards the design of a biped jogging robot. In *IEEE Int. Conf. on Robotics and Automation*, pages 4140–4145, 2001.

[31] L. Gilchrist and D. Winter. A two-part, viscoelastic foot model for use in gait simulations. *J. Biomechanics*, 29(6):795–798, 1996.

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

[33] P. Gilmore and C. Kelley. An implicit filtering algorithm for optimization of functions with many local minima. *SIAM Journal on Optimization*, 5:269–285, 1995.

[34] A. Gordon, A. Huxley, and F. Julian. The variation in isometric tension with sarcomere length in vertebrate muscle fibres. *Journal of Physiology*, 184:170–192, 1966.

[35] P. Gregorio, M. Ahmadi, and M. Buehler. Design, control, and energetics of an electrically actuated legged robot. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 27 (4):626–634, 1997.

[36] F. D. Groote, G. Pipellers, I. Jonkers, and B. Demeulenaere. A physiology based inverse dynamic analysis of musculotendon forces during human movement. *Journal of Biomechanics*, 39, suppl. 1, 2006.

[37] K. Gruber. *Entwicklung eines Modells zur Berechnung der Kräfte im Knie- und Hüftgelenk bei sportlichen Bewegungsabläufen mit hohen Beschleunigungen*. PhD thesis, Eberhard-Karls-Universität, Tübingen, 1987.

[38] K. Gruber, H. Ruder, J. Denoth, and K. Schneider. A comparative study of impact dynamics. wobbling mass model versus rigid body models. *Journal of Biomechanics*, 31: 439–444, 1998.

[39] M. Günther. *Computersimulation zur Synthetisierung des muskulär erzeugten menschlichen Gehens unter Verwendung eines biomechanischen Mehrkörpermodells*. PhD thesis, Eberhard-Karls-Universität zu Tübingen, 1997.

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

[41] 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.

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

[43] H. Hatze. The complete optimization of a human motion. *Mathematical Biosciences*, 28: 99–135, 1976.

[44] E. Haug. *Computer Aided Kinematics and Dynamics of Mechanical Systems: Basic Methods*. Allyn and Bacon, Boston, 1989.

[45] A. Heim and O. von Stryk. Documentation of parest — a multiple shooting code for optimization problems in differential-algebraic equations. Technical report, Report TUM-M9616, Mathematisches Institut, Technische Universität München, 1996. http://www.sim.informatik.tu-darmstadt.de/sw/parest.

[46] T. Hemker, K. Fowler, and O. von Stryk. Derivative-free optimization methods for handling fixed costs in optimal groundwater remediation design. In *Proc. of the CMWR XVI - Computational Methods in Water Resources*, 19-22 June 2006.

[47] T. Hemker, H. Sakamoto, M. Stelzer, and O. von Stryk. Hardware-in-the-loop optimization of the walking speed of a humanoid robot. In *CLAWAR 2006: 9th International Conference on Climbing and Walking Robots*, pages 614–623, Brussels, Belgium, September 11-14 2006.

[48] A. Henze. *Dreidimensionale biomechanische Modellierung und die Entwicklung eines Reglers zur Simulation zweibeinigen Gehens*. PhD thesis, Eberhard-Karls-Universität zu Tübingen, 2002.

[49] A. V. Hill. The heat of shortening and the dynamic constants of muscle. *Proceedings of the Royal Society of London*, 126 B:136–195, 1938.

[50] A. L. Hof and J. 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.

[51] R. Höpler. *A Unifying Object-oriented Methodology to Consolidate Multibody Dynamics Computations in Robot Control*. PhD thesis, Technische Universität Darmstadt, 2004.

[52] R. Höpler, M. Stelzer, and O. von Stryk. Integrated, object-oriented dynamics modeling for design, trajectory optimization and control of legged robots. In *EUROMECH Colloquium 452 on Advances in Simulation Techniques for Applied Dynamics*, Halle (Saale), Germany, March 1-4 2004.

[53] R. Höpler, M. Stelzer, and O. von Stryk. Object-oriented dynamics modeling for legged robot trajectory optimization and control. In *Proc. IEEE Intl. Conf. on Mechatronics and Robotics (MechRob)*, pages 972–977, Aachen, Sept. 13-15 2004. Sascha Eysoldt Verlag.

[54] R. Höpler, M. Stelzer, and O. von Stryk. Object-oriented dynamics modeling for simulation, optimization and control of walking robots. In *Proc. 18th Symposium on Simulation Technique, ASIM, Erlangen, September 12-15*, pages 588–593, 2005.

[55] R. Höpler, M. Stelzer, and O. von Stryk. Object-oriented dynamics modeling of walking robots for model-based trajectory optimization and control. In I. Troch and F. Breitenecker, editors, *Proc. 5th MATHMOD Vienna, February 8-10*, number 30 in ARGESIM Reports, 2006.

[56] A. F. Huxley. Muscle structure and theories of contraction. *Progress in Biophysics and Biophysical Chemistry*, 7:257–318, 1957.

[57] A. F. Huxley. Muscular contraction. *Journal of Physiology*, 243:1–43, 1974.

[58] V. G. Ivancevic and T. T. Ivancevic. *Natural Biodynamics*. World Scientific Publishing, 2006.

[59] A. Jain. Unified formulation of dynamics for serial rigid multibody systems. *J. Guidance, Contr., Dynamics*, 14(3):531–542, 1991.

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

[61] M. L. Kaplan and J. H. Heegaard. Predictive algorithms for neuromuscular control of human locomotion. *Journal of Biomechanics*, 34(8):1077–1083, 2001.

[62] B. Katz. The relation between force and speed in muscular contraction. *Journal of Physiology*, 96:45–64, 1939.

[63] M. Kiehl. Sensitivity analysis of ODEs and DAEs — theory and implementation guide. *Optimization Methods and Software*, 10(6):803–821, 1999.

[64] J.-H. R. Kim and H. Maurer. Sensitivity analysis of optimal control problems with bang-bang controls. In *Proceedings of the 42nd IEEE Conference on Decision and Control, Maui*, pages 3281–3286, 2003.

[65] H. Kimura, I. Shimoyama, and H. Miura. Dynamics in the dynamic walk of a quadruped robot. *Advanced Robotics*, 4(3):283–301, 1990.

[66] J. Z. Liu, R. W. Brown, and G. H. Yue. A dynamical model of muscle activation, fatigue, and recovery. *Biophysical Journal*, 82(5):2344–2359, 2002.

[67] W. Liu and B. M.Nigg. A mechanical model to determine the influence of masses and mass distribution on the impact force during running. *Journal of Biomechanics*, 33:219–224, 2000.

[68] K. Malanowski. Sensitivity analysis of optimization problems in Hilbert spaces with applications to optimal control. *Applied mathematics and optimization*, 21:1–20, 1990.

[69] K. Malanowski. Stability and sensitivity of solutions to nonlinear optimal control problems. *Applied mathematics and optimization*, 32:111–141, 1995.

[70] K. Malanowski and H. Maurer. Sensitivity analysis for parametric optimal control problems with control-state constraints. *Computational Optimization and Applications*, 5:253–283, 1996.

[71] H. Maurer and N. P. Osmolovskii. Second order optimality conditions for bang-bang control problems. *Control and Cybernetics*, 33:555–584, 2003.

[72] H. Maurer and H. J. Pesch. Solution differentiability for parametric nonlinear control problems. *SIAM Journal on Control and Optimization*, 32:1542–1554, 1994.

[73] H. Maurer and H. J. Pesch. Solution differentiability for parametric nonlinear control problems with control-state constraints. *Journal of Optimization Theory and Applications*, 86(2):285–309, 1995.

[74] H. Maurer and S. Pickenhain. Second order sufficient conditions for control problems with mixed control-state constraints. *Journal of Optimization Theory and Application*, 86 (3):649–667, 1996.

[75] J. J. More. Levenberg-Marquardt algorithm: implementation and theory. *Conference on numerical analysis; 28 Jun 1977; Dundee, UK*, 1977.

[76] 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.

[77] R. R. Neptune and M. L. Hull. A theoretical analysis of preferred pedaling rate selection in endurance cycling. *Journal of Biomechanics*, 32:409–415, 1999.

[78] 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.

[79] R. R. Neptune, I. C. Wright, and A. J. Van den Bogert. A method for numerical simulation of single limb ground contact events: application to heel-toe running. *Computer Methods in Biomechanics and Biomedical Engineering*, 3:321–344, 2000.

[80] B. M. Nigg and W. Herzog. *Biomechanics of the Musculo-skeletal System*. Wiley, 1999.

[81] M. Olbrich. Ein strukturausnützendes Verfahren zur Bahnoptimierung bei Robotern. Master's thesis, Technische Universität Münchent, Germany, 1995.

[82] M. G. Pandy. Computer modeling and simulation of human movement. *Annu. Rev. Biomed. Eng.*, 3:245–273, 2001.

[83] M. G. Pandy and K. Sasaki. A three-dimensional musculoskeletal model of the human knee joint. part 2: Analysis of ligament function. *Computer Methods in Biomechnis and Biomedial Engineering*, 1:265–283, 1998.

[84] 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.

[85] F. Pfeiffer and C. Glocker. *Multibody Dynamics with Unilateral Contacts*. Wiley Series Nonlinear Science. New York, 1996.

[86] R. W. Ramsey and S. F. Street. The isometric length-tension diagram of isolated skeletal muscle fibres of the frog. *Journal of Cellular and Comparative Physiology*, 15:11–34, 1940.

[87] 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.

[88] 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.

[89] 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.

[90] W. Schiehlen, editor. *Multibody Systems Handbook*. Springer-Verlag, 1990.

[91] W. Schiehlen, editor. *Advanced Multibody System Dynamics: Simulation and Software Tools*. Kluwer, 1993.

[92] A. Seyfarth, A. Friedrichs, V. Wank, and R. Blickhan. Dynamics of the long jump. *Journal of Biomechanics*, 32:1259–1267, 1999.

[93] A. Seyfarth, R. Tausch, M. Stelzer, F. Iida, A. Karguth, and O. von Stryk. Towards bipedal running as a natural result of optimizing walking speed for passively compliant three-segmented legs. In *CLAWAR 2006: 9th International Conference on Climbing and Walking Robots*, pages 396–401, Brussels, Belgium, September 12-14 2006.

[94] A. Siemienski. Direct solution of the inverse optimization problem of load sharing between muscles. *Journal of Biomechanics*, 39, suppl. 1, 2006.

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

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

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

[98] 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.

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

[100] M. Spong and M. Vidyasagar. *Robot Dynamics and Control*. J. Wiley & Sons, 1989.

[101] M. Stelzer and O. von Stryk. Efficient forward dynamics simulation and optimization of human body dynamics. *ZAMM, Zeitschrift für Angewandte Mathematik und Mechanik, Journal of Applied Mathematics and Mechanics*, 86(10):828–840, 2006.

[102] M. Stelzer and O. von Stryk. Efficient forward dynamics simulation and optimization of locomotion: from legged robots to biomechanical systems. In *Proc. 3rd Intl. Symposium on Adaptive Motion in Animals and Machines (AMAM)*, 2005.

[103] M. Stelzer and O. von Stryk. From robots to humans: Towards efficient forward dynamics simulation and optimization exploiting structure and sensitivity information. In *Proceedings of ECCOMAS Multibody Dynamics 2005, Advances in Computational Multibody Dynamics, Madrid, June 21-24*, 2005.

[104] M. Stelzer and O. von Stryk. A novel approach to efficient forward dynamics simulation and optimization of human motion and its application. *Journal of Biomechanics*, 39, suppl. 1, 2006.

[105] M. Stelzer and O. von Stryk. Human kicking motion using efficient forward dynamics simulation and optimization. In *Proceedings of BioMed 2006, Kuala Lumpur International Conference on Biomedical Engineering*, 2006.

[106] M. Stelzer and O. von Stryk. Applications of efficient forward dynamics simulation in biomechanics. In *Proceedings of the XVth International Conference on Mechanics in Medicine and Biology, Singapore*, 2006.

[107] 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, 2003.

[108] W. Stelzle, A. Kecskeméthy, and M. Hiller. A comparative study of recursive methods. *Archive of applied mechanics*, 66(1–2):9–19, 1995.

[109] D. Strobach, A. Kecskeméthy, G. Steinwender, and B. Zwick. A simplified approach for rough identification of muscle activation profiles via optimization and smooth profile patches. In *Proc. of ECCOMAS Multibody Dynamics 2005, Advances in Computational Multibody Dynamics, Madrid, June 21-24*, 2005.

[110] D. Tsirakos, V. Baltzopoulos, and R. Bartlett. Inverse optimization: functional and physiological considerations related to the force-sharing problem. *Critical Reviews in Biomedical Engineering*, 25, 1997.

[111] A. J. van Soest. *Jumping from structure to control*. Phd thesis, Vrije Universiteit Amsterdam, 1992.

[112] 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.

[113] O. von Stryk. Optimal control of multibody systems in minimal coordinates. *Z. Angew. Math. Mech.*, 78(Suppl. 3):1117–1120, 1998.

[114] O. von Stryk. *Numerical Hybrid Optimal Control and Related Topics*. Habilitationsschrift, Technische Universität München, 2003.

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

[116] M. Vukobratović and B. Borovac. Zero-moment point thirty five years of its life. *International Journal of Humanoid Robotics*, 1(1):157–173, 2004.

[117] M. W. Walker and D. E. Orin. Efficient dynamic computer simulation of robotic mechanisms. *ASME J. of Dynamic Systems Meas. and Control*, 104:205–211, 1982.

[118] D. R. Wilkie. The mechanical properties of muscle. *British Medical Bulletin*, 12:177–182, 1956.

[119] J. M. Winters and L. Stark. Muscle models: what is gained and what is lost by varying model complexity. *Biological Cybernetics*, 55:403–420, 1987.

[120] R. C. Woledge, N. A. Curtin, and E. L. Homsher. *Energetic Aspects of Muscle Contraction*. Academic Press, 1985.

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

[122] D. Zlatnik. *Intelligently Controlled Above Knee Prosthesis*. PhD thesis, ETH Zürich, 1998.