Efficient Dynamic Modeling, Numerical Optimal Control and Experimental Results for Various Gaits of a Quadruped Robot

M STELZER, M HARDT and **O von STRYK**
Simulation and Systems Optimization, Technische Universität Darmstadt, 64283 Darmstadt, Germany

This is a preprint of the paper that appeared in:
CLAWAR: International Conference on Climbing and Walking Robots, Catania, Italy, September 17-19 (Professional Engineering Publishing Ltd., 2003) 601-608.

ABSTRACT

Numerical simulation and optimization of gaits for quadruped robots based on nonlinear multi-body dynamics models of legged locomotion have made progress recently. A fully three-dimensional dynamical model of Sony's four-legged robot is used to state an optimal control problem for a symmetric, dynamically stable gait. The optimal control problem is solved by a sparse direct collocation method. Numerical problems related to the high-index differential algebraic equations of motion are avoided by substituting the differential algebraic equations by an equivalent set of reduced dynamics ordinary differential equations. Numerical and experimental results validate the model and the methods used for gait generation.

1 INTRODUCTION

Generating dynamically stable symmetric gaits for legged robots is still a challenge. For a given gait pattern, landing time and point of each leg are prescribed, i.e. they depend on parameters. The trajectory of each joint between lift-off and landing is not uniquely determined. To overcome this problem of redundancy, the problem of finding a dynamically stable symmetric gait is formulated as an optimal control problem, involving the robot's dynamics and several additional constraints. Efficient methods are needed for both the dynamics and the resulting optimal control problem. The resulting trajectories are implemented on the robot using the given trajectory tracking control on joint level. Therefore, the optimal control problem formulation for the computation of reference trajectories for a dynamic gait must account for possible inaccuracies in the dynamic model and parameters as well as for external disturbances. All aspects of generating a reference trajectory by dynamic optimization, that can be implemented on the real robot, are presented here for Sony's four-legged robot AIBO (www.aibo.com).

The organization of the paper is as follows. Section 2 describes the robot dynamics model and the algorithm for evaluating the dynamics. The formulation of the optimal control problem and the numerical method used for solving it are presented in Section 3. Numerical and experimental results are given in Section 4.

2 EFFICIENT DYNAMIC MODELING

2.1 SONY'S FOUR-LEGGED ROBOT

In this paper we will concentrate on finding dynamically stable gaits for Sony's AIBO (Figure 1). Nevertheless, the methods presented here may be used for other legged robots, too. The kinematical structure of the robot consists of several actuated joints as described in the next subsection. Originally designed as a toy and entertainment robot, there also exist worldwide competitions for teams of autonomous soccer playing robots (www.RoboCup.org). For this application fast and stable gaits are mandatory. The robot contains an onboard CPU of 196 MHz, which in autonomous soccer competitions mainly must be used for image processing using the integrated CCD-camera and image understanding (localisation 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 [2], 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.

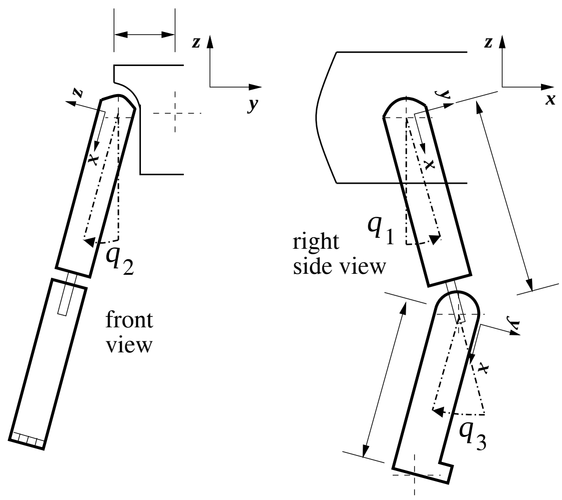

Figure 1: The four-legged Sony robot (left) and the kinematical structure of one leg (right).

2.2 Dynamic Model

Our 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 kinetical data for each link (length, mass, center of mass, moments of inertia) have been provided by Sony to the authors.

2.3 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 of each of the joints $u_{i,\max}$ 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 described in Section 3 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. As already mentioned 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, by 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.

2.4 Articulated Body Algorithm

Various approaches exist for dealing with multibody systems' equations. We chose articulated body algorithm (ABA) due to its several advantages over other methods like symbolic methods, composite rigid body algorithms or natural coordinates. ABA is a numerical, recursive, order N algorithm (where N is the number of links in the multibody system). It shows a high modularity with respect to exchange of submodels within the dynamic structure, useful when considering different actuators or limb complexity, and varying contact situations of different feet having contact with ground. It is useful to be able to work with the same model, even if parts of the model are substituted by improved models. Also a model that may be used in all different situation when considering the robot is desired: ABA may be used in off-line trajectory calculation, estimation of parameters in the model and model-based on-line controllers as well.

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

$$\begin{aligned} \ddot{\mathbf{q}} &= \mathcal{M}(\mathbf{q})^{-1}\Big(B\mathbf{u} - \mathcal{C}(\mathbf{q},\dot{\mathbf{q}}) - \mathcal{G}(\mathbf{q}) + J_c(\mathbf{q})^T\mathbf{f}_c\Big) \\ 0 &= \mathbf{g}_c(\mathbf{q}) \end{aligned} \tag{2.1}$$

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

The articulated body algorithm exploits the tree structure of the multibody system by calculating dynamics in several sweeps from base to tip and tip to base. Three sweeps are needed for evaluating forward dynamics, two additional sweeps treat contact situations. The transfer operators from one link to the subsequent resp. preceding link have equivalents in Kalman Filter theory, making the algorithm interesting from a mathematical point of view. Details may be found in [5]. We use the implementation SOAFOR (Spatial Operator Algebra Fortran routines [5]) of the articulated body algorithm.

2.5 Reduced Dynamics Algorithm

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, treating explicitly only the independent states $\mathbf{q}_I$, which are global orientation and position and states related to legs in contact with ground, and using inverse kinematics to determine the dependent states $\mathbf{q}_D$ of the other legs:

$$\begin{aligned} \mathbf{q}_I &:= \text{global orientation, position; swing leg(s) states} \\ \mathbf{q}_D &:= \text{contact leg(s) states} \end{aligned}$$

$\mathbf{q}_I$ may be computed from all states $\mathbf{q}$ using a constant mapping Z, i.e. $\mathbf{q}_I = Z\mathbf{q}$. The solution of the reduced dynamics

$$\ddot{\mathbf{q}}_I = Z\mathcal{M}\left(\tilde{\mathbf{q}}\right)^{-1}\left(B\mathbf{u} - C\left(\tilde{\mathbf{q}},\dot{\tilde{\mathbf{q}}}\right) - \mathcal{G}\left(\tilde{\mathbf{q}}\right) + J_c^T\mathbf{f}_c\right),$$

where $\tilde{\mathbf{q}}$ consists of the independent states and of the dependent states determined from inverse kinematics, then may be proven to be the solution of the initial system of differential algebraic equations (2.1) [5]. Inverse kinematics for each leg of the robot here has a unique solution if agreements concerning the bending of joints are made. This makes it possible to deal with reduced dynamics algorithms. As a result, 24 instead of 36 states can describe the model and a set of ordinary differential equations only instead of a system of differential algebraic equations may be considered.

3 OPTIMAL CONTROL

3.1 Optimal Control Problem

The problem of finding a symmetric gait for the robot is stated as an optimal control problem. Different gait patterns such as walk, trot, rack, canter and rotary or transverse gallop differ in the duty factor of each foot, i.e. the fraction of a total stride cycle during which the foot is in contact with ground, and relative phases of feet, i.e. the order and time displacement of feet reaching ground [1]. However there is no essential difference in the modeling and optimization approach presented in this paper when applied to different gait patterns. We consider the optimal control problem for a trot, which is characterized by a duty factor of 0.5 and diagonal legs lifting off and landing synchronously. The trot gait has a special property of symmetry (which, for example, gallop has not) that enables us to reduce the problem formulation to a half stride, consisting of one phase. For more complicated gait patterns, more phases may be needed: During one phase contact situation may not change for sake of having to deal with a well defined set of considered states (states related to legs with feet in contact with ground are not considered when using the reduced dynamics approach). Since additional conditions on change of phases are similar to those on the boundary, it is easy to transfer the following considerations to other gait patterns. The optimal control problem is stated as follows:

$$\begin{aligned}
&\min \mathcal{J}[\mathbf{q},\dot{\mathbf{q}},\mathbf{u},t_f] \ \text{ subject to} && \text{minimize the merit function } \mathcal{J} \text{ subject to} \\
&\mathcal{M}\ddot{\mathbf{q}} = B\mathbf{u} - C\left(\mathbf{q},\dot{\mathbf{q}}\right) - \mathcal{G}\left(\mathbf{q}\right) + J_c^T\mathbf{f}_c, && \text{system of MBS ODE} \\
&\mathbf{g}_c\left(\mathbf{q}\right) = 0 && \text{contact algebraic conditions} \\
&\mathbf{b}\left(\mathbf{q}(t_0),\mathbf{q}(t_f),t_0,t_f\right) = 0 && \text{boundary conditions} \\
&\mathbf{n}(\mathbf{q},\mathbf{u}) \geq 0 && \text{nonlinear inequality constraints,} \\
&\mathbf{q}_{\min} \leq \mathbf{q} \leq \mathbf{q}_{\max}, \ \ \mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max} && \text{box constraints on state and control variables.}
\end{aligned}$$

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.

Useful merit functions are, for example, time t_f, energy $\int \sum_{i=1}^{m} u_i^2$, or combinations of both [7]. 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 proper sine curve. This property increases stability by avoiding contact with the ground resulting from deflexions of bodies and joints, and 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, while of course implicitly it is ensured by periodicity of the generated gait and may be checked by one of the criteria given in [7]. More details on each of the constraints may be found in [3], where the constraints are stated for a humanoid robot.

3.2 Solving the Optimal Control Problem

For solving the optimal control problem, the method DIRCOL [8] is used. The states and controls are approximated by piecewise cubic resp. piecewise linear polynomials on a discrete and successively refinable time grid. The optimal control problem is thereby transcribed into a nonlinear program with the coefficients of the polynomials as variables, which may be solved by a sequential quadratic and, due to the special structure of the variables, sparse programming method [4]. For more details we refer to [6, 8].

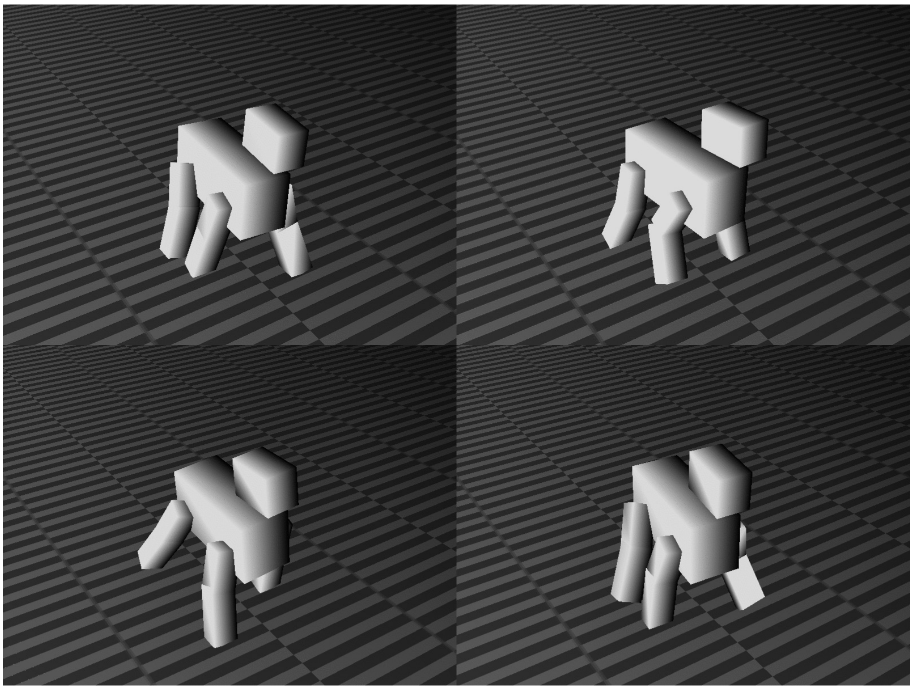

Figure 2: Four scenes from an animation of a computed trot gait.

4 RESULTS

4.1 Numerical Results

Here, we present numerical results for generating an optimal trajectory for a trot gait. 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 visualised not only by plotting the trajectories (as in Figure 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 2 four single images of a computed trot are shown, each for a different contact situation.

4.2 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 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

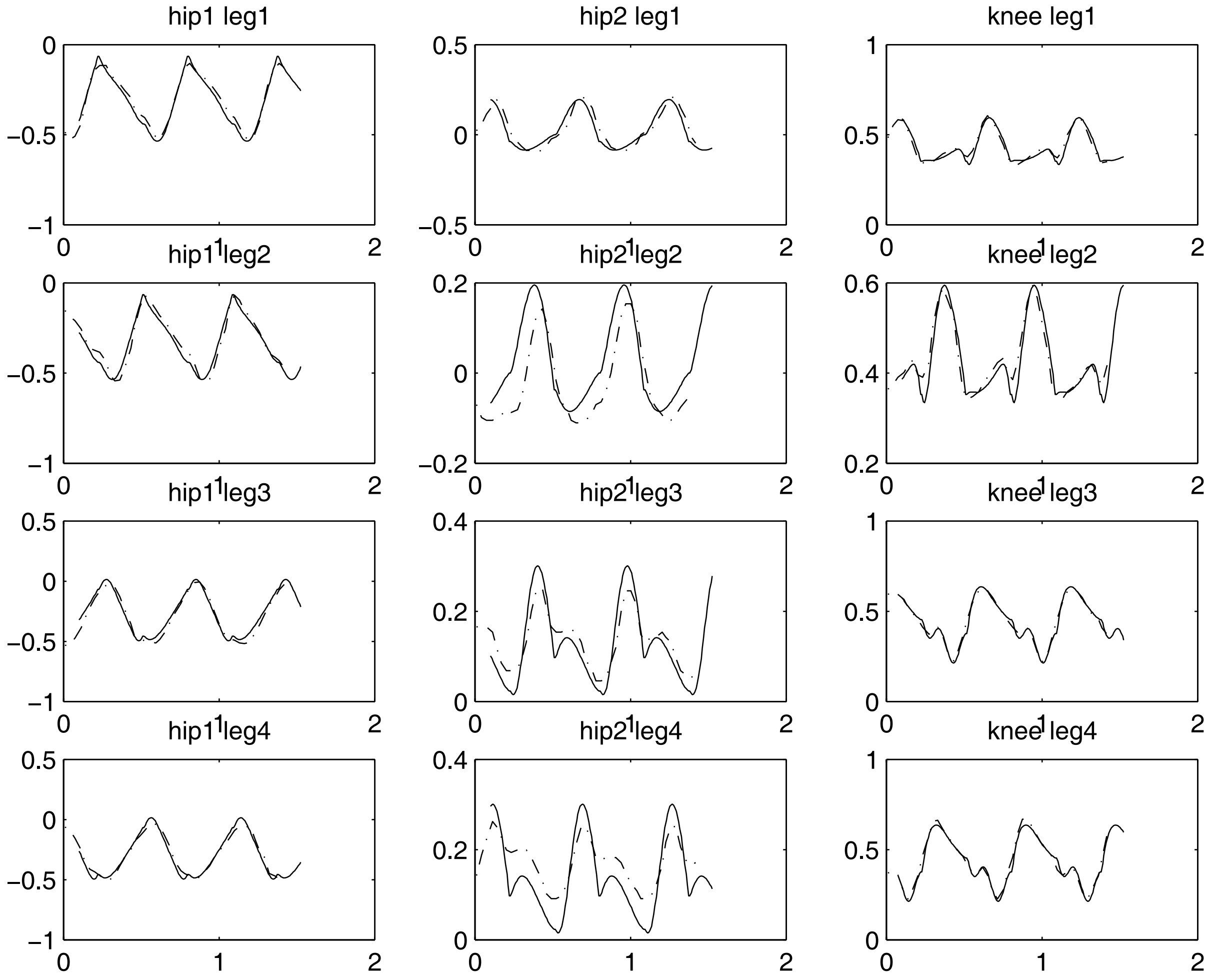

Figure 3: The experimentally measured joint angle trajectories (dotted lines; joint angle [rad] versus time [s]) for the first hip joints and the knee joints match the computed reference trajectories (solid lines) quite well after considering improved estimates for maximum torque and velocity constraints. 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.

very well match the calculated trajectories for each of the twelve actuated joints (cf. Figure 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 yielding 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 movements. 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.

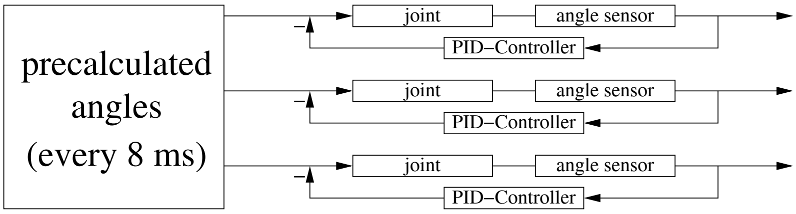

Figure 4: Decentralized PID-control scheme for the three joints of one leg.

5 CONCLUSION AND OUTLOOK

Based on first experimental results, the underlying optimal control problem has been refined by additional conditions to avoid slipping, and the dynamic robot model has been refined by a more detailed motor model. Having detailed and accurate models also of the actuators, even faster gaits than trot may be implemented, e.g. gallop. Using the described methods for simulation and optimization of refined dynamic robot models a whole suite of different quadruped gaits may be realized quickly.

Future work will include the refinement of motor characteristics using experiments and inverse dynamics algorithms, the investigation of other gaits than trot, and walking on underarms instead of the feet for the front legs which reduces the possibility of falling down in case of a lateral shock through collision with another robot.

REFERENCES

[1] R. Alexander: “The Gaits of Bipedal and Quadrupedal Animals,” *The International Journal of Robotics Research*, Vol. 3, No. 2, pp. 49–59, 1984.

[2] H.-D. Burkhard et al.: ”GermanTeam 2002”. Technical Report (2002), preprint available from `http://www.sim.informatik.tu-darmstadt.de/publ/`

[3] M. Buss, M. Hardt, J. Kiener, M. Sobotka, M. Stelzer, O. von Stryk, D. Wollherr: ”Towards an Autonomous, Humanoid, and Dynamically Walking Robot: Modeling, Optimal Trajectory Planning, Hardware Architecture, and Experiments”, Third IEEE International Conference on Humanoids Robots, TU München and Universität Karlsruhe, Germany, October 1-3, 2003, submitted.

[4] P.E. Gill, W. Murray, and M.A. Saunders: “User’s Guide for SNOPT 5.3: a Fortran Package for Large-Scale Nonlinear Programming”, Report, Math Dept., UC San Diego, 1997.

[5] M. Hardt: *Multibody Dynamical Algorithms, Numerical Optimal Control, with Detailed Studies in the Control of Jet Engine Compressors and Biped Walking*. Ph.D. Thesis, Electrical Engineering, University of California San Diego, U.S.A. (1999)

[6] M. Hardt, O. von Stryk: “Towards optimal hybrid control solutions for gait patterns of a quadruped.” In: M. Armada, P. Gonzalez de Santos (eds.): Proc. CLAWAR 2000 - 3rd Int. Conference on Climbing and Walking Robots, Madrid, October 2-4, 2000 (Bury St. Edmunds and London, UK: Professional Engineering Publishing, 2000) pp. 385-392.

[7] M. Hardt, O. von Stryk: “Dynamic modeling in the simulation, optimization, and control of legged robots.” *Z. Angew. Math. Mech., Vol. 83* (2003) to appear.

[8] O. von Stryk: ”User’s Guide for DIRCOL Version 2.1”, Simulation and Systems Optimization Group, Technische Universität Darmstadt (1999) `http://www.sim.informatik.tu-darmstadt.de/sw/dircol`