Towards an Autonomous, Humanoid, and Dynamically Walking Robot: Modeling, Optimal Trajectory Planning, Hardware Architecture, and Experiments

Martin Buss[3], Michael Hardt[2], Jutta Kiener[2], Marion Sobotka[1],
Maximilian Stelzer[2], Oskar von Stryk[2], and Dirk Wollherr[1]

[1] Control Systems Group, Technical University Berlin
Einsteinufer 17/EN 11, D-10587 Berlin, Germany
Dirk.Wollherr@TU-Berlin.de   sobotka@rs.tu-berlin.de
http://www.rs.tu-berlin.de
[2] Simulation and Systems Optimization Group, Technische Universität Darmstadt
Alexanderstr. 10, D-64283 Darmstadt, Germany
{hardt, kiener, stelzer, stryk}@sim.tu-darmstadt.de
http://www.sim.informatik.tu-darmstadt.de
[3] Institute of Automatic Control Engineering, Technische Universität München
D-80290 München, Germany
Martin.Buss@ei.tum.de
http://www.lsr.ei.tum.de
(research performed while author was with TU Berlin)

Abstract. The development process to achieve walking motion with a recently constructed humanoid robot is discussed. The desired motion is based on the solution of an optimal control problem whose constraints depend upon the high-dimensional nonlinear multibody system dynamics of the 17 DoF humanoid and physical contact constraints with the environment. On-line control strategies are developed to track the precalculated trajectories. Experimental walking results with the humanoid robot are presented.

1   Introduction

Many efforts to achieve autonomous biped locomotion may be divided into two main approaches: gait trajectories are computed online according to the actual intention and perception data of the robot [15, 17, 24, 26], or a large set of trajectories is computed offline [1, 2, 11, 25] and the robot selects one of these precalculated trajectories depending upon its situation [18].

Suitable gait trajectories for a biped robot are subject to numerous physical constraints, and the gait is additionally required to fulfill some "aesthetic" criteria [12], e.g. smoothness, energy efficiency and effectiveness. Recursive optimization techniques provide a means for updating the gait trajectories in the absence of a closed-form solution. Such computations are still difficult to execute

on-line with current computing capacity, though on-line finite horizon methods are in development. This work, however, is based on the offline generation of gait trajectories.

A precise modeling of the robot kinematics and dynamics is particularly crucial for the development of dynamically stable trajectories. An accurate modeling of legged locomotion systems must rely on high-dimensional nonlinear multibody system (MBS) dynamics subject to constraints. Additional complex tasks include the generation, optimization and control of stable motions for such systems. High-level modeling and simulation tools can be vital in the development of autonomous biped locomotion and can also assist in the selection and integration of hardware and software.

The intrinsic disadvantage of relying upon precalculated trajectories is the difficulty of accomplishing alternative motions for which trajectories are not available. Thus, another problem arises on how to modify trajectories such that they can be applied to slightly different situations than they originally were designed for, whereby the modification task must further take into account stability of locomotion. For this purpose, a method called Jacobi Compensation has been developed which can make trajectory modifications producing changes in the motion of certain parts of the body in a selected Cartesian direction.

Many efforts by other research groups related to the online modification of precalculated trajectories have focused on ensuring gait stability by controlling the Zero-Moment-Point (ZMP) [13] or adapting existing trajectories for walking in the plane to walking on slopes [27]. Preliminary investigations towards an alternative trajectory modification scheme are conducted in the experiments. The adaptation method described here is, in principle, applicable to a wider range of problems including gait stabilization, adaptation to slopes and compensation of trajectory imperfections, though this has yet to be demonstrated.

For the organization of this article: Section 2 presents the kinematic parameters, inverse leg kinematics, and the dynamic parameters of the developed humanoid prototype. An efficient dynamic modeling paradigm proposed in Section 3 is the basis for a complete dynamic model used to obtain stride primitives by numerical optimal control as discussed in Section 4. A method for online modification of gait trajectories is introduced in Section 5. Experimental results with a biped humanoid walking machine are presented in Section 6.

2 Kinematic Model

Fig. 1 displays a schematic of the current humanoid prototype (see Appendix A and Figs. 7 and 9 for a picture of the real robot and for details of the hardware (HW) and software (SW) system architecture. The humanoid construction consists of 17 actuated joints:

- two legs each with 6 actuated joints
  - hip with 3 DoF (*Degrees of Freedom*) rotating about x-, z- and y-axes
  - knee with 1 DoF rotating about y-axis
  - ankle with 2 DoF rotating about y- and x-axes

- waist with 1 actuated joint rotating about z-axis
  - two shoulders each with 2 actuated joints rotating about y- and x-axes

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

  - 6 DoF describing a fictitious 3D rotation and translation joint between the reference free-floating body (torso) and an inertial reference frame and
  - 17 DoF for the existing internal joints.
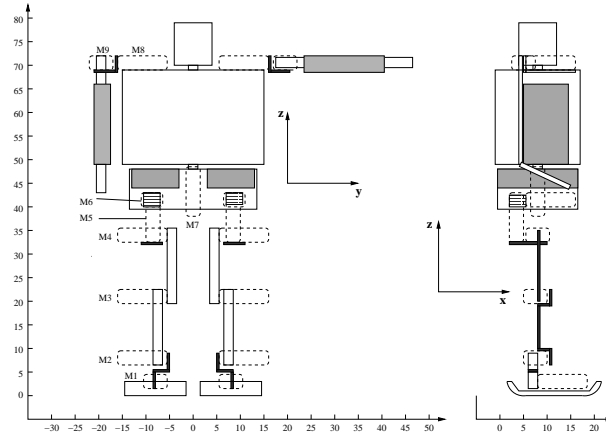
Fig. 1. Humanoid kinematic structure

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.

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

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

For the experiments reported in this paper, 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 $\boldsymbol{u} \in \mathbb{R}^{12}$. Future experiments will integrate these joints to exploit the additional DoF in order to further optimize the performance specifications (Sect. 4).

The humanoid robot is 80 cm tall and has a weight of about 12 kg without batteries and mainboard. Its kinematic structure complies with the regulations

for the Humanoid League of RoboCup [20], an annual worldwide autonomous robot soccer competition. 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 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.

2.1 Inverse Kinematics

For the evaluation of the reduced dynamics presented in Sect. 3, 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 [22]. 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.

2.2 Dynamic Parameters

For dynamical calculations (Sect. 3), 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 be as exact as possible. The real humanoid is divided into geometrical primitives (cylinder, ellipsoid, or box) which are then individually measured and weighed. One leg, for example, is divided into more than 20 units. Assuming a uniform mass density in object, the link inertias may be approximated with the help of the parallel and perpendicular axis theorem.

3 Efficient Dynamic Modeling

Biped constructions generally consist of a minimum of five links with two to six degrees of freedom per leg. Dynamical simplifications allow one to analyze certain predominant behaviors of the dynamic system, but many other important features are lost. A more complete dynamical system description contains more

significant dynamical effects yet a control solution for these models based on an analytical approach is usually not possible and results must be sought for numerically. The modeling and optimization approaches presented here are thus strongly dependent upon numerical methods.

Various approaches exist for modeling the multibody system (MBS) dynamics of a tree-structured legged robot subject to unilateral contact constraints, all with quite different characteristics regarding efficiency and accuracy in simulation and optimization. The selected MBS modeling and computational approach relies upon the Articulated Body Algorithm (ABA) due to its superior modularity and computational efficiency for high dimensional systems [3, 21]. These methods belong to the class of numeric, recursive, $\mathcal{O}(\mathcal{N})$ complexity methods, where N is the number of links in the system, further demonstrate their superior modularity and flexibility when parts of the kinematical structure or the kinetical data have to be changed and refined as occurs frequently during the design and operation cycle of a humanoid robot. It is also desirable to use the same dynamic modeling framework during the entire development and operation period of a legged robot, e.g., for the selection of actuators using dynamic optimization (Sect. A.1), and for the optimization of reference trajectories for dynamic walking (Sect. 4.2), for the calibration of model parameters by optimization, for the model-based estimation of dynamic state variables, and for the future development of nonlinear dynamic model-based controllers realizing dynamically stable legged locomotion.

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

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

(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 $\boldsymbol{u}(t) \in \mathbb{R}^m$ are the control input functions which are mapped with the constant matrix $B \in \mathbb{R}^{N \times m}$ to the actively controlled joints. The ground contact constraints $\boldsymbol{g_c} \in \mathbb{R}^{n_c}$ represent holonomic constraints on the system from which the constraint Jacobian may be obtained $J_c = \frac{\partial \boldsymbol{g_c}}{\partial \boldsymbol{q}} \in \mathbb{R}^{n_c \times N}$, while $\boldsymbol{f}_c \in \mathbb{R}^{n_c}$ is the ground constraint force. $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, and $\ddot{\boldsymbol{q}}$ are the generalized position, velocity and acceleration vectors respectively.

A property prevalent in legged machines is that their constrained contact legs often have unique inverse kinematic solutions for their joint angles and angle velocities. This lends itself to the use of reduced dynamics algorithms for simulation and optimization. The projection of the dynamics (1) onto a reduced set of independent states converts the differential-algebraic (DAE) contact system (1) into a system of first order ordinary differential equations (ODEs) of minimal size. Define the independent $\boldsymbol{q_I}$ and dependent $\boldsymbol{q_D}$ states as:

$$\boldsymbol{q_I} = \text{global orientation, position; swing leg(s) states}$$
$$\boldsymbol{q_D} = \text{contact leg(s) states}$$

from which $\boldsymbol{q_I} = Z\boldsymbol{q}$, where Z is a constant mapping. The reduced dynamics

$$\ddot{\boldsymbol{q}}_I = Z\mathcal{M}(\boldsymbol{q})^{-1}\Big(B\boldsymbol{u} - \mathcal{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) - \mathcal{G}(\boldsymbol{q}) + J_c^T\boldsymbol{f}_c\Big) . \tag{2}$$

is computed using a recursive numerical multibody algorithm [10]. The second time derivative of the contact constraints are then satisfied with the simulation of this ODE.

3.1 Constraints

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 [10]:

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

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

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:

$$
\begin{bmatrix} q_1^e \\ q_2^e \\ q_3^e \\ q_4^e \\ q_5^e \\ q_6^e \end{bmatrix} = \begin{bmatrix} -1 & & & & & \\ & 1 & & & 0 & \\ & & -1 & & & \\ & & & 1 & & \\ & 0 & & & -1 & \\ & & & & & 1 \end{bmatrix} \begin{bmatrix} q_1^0 \\ q_2^0 \\ q_3^0 \\ q_4^0 \\ q_5^0 \\ q_6^0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ step \\ 0 \\ 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} q_7^e \\ q_8^e \\ q_9^e \\ q_{10}^e \\ q_{11}^e \\ q_{12}^e \end{bmatrix} = \begin{bmatrix} -1 & & & & & \\ & 1 & & & 0 & \\ & & -1 & & & \\ & & & 1 & & \\ & 0 & & & -1 & \\ & & & & & 1 \end{bmatrix} \begin{bmatrix} q_7^0 \\ q_8^0 \\ q_9^0 \\ q_{10}^0 \\ q_{11}^0 \\ q_{12}^0 \end{bmatrix}
$$

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,

$$\boldsymbol{f}_{c,i} = \mathbf{0}_{6\times1} \qquad \text{where leg } i \text{ is breaking its contact with the ground.}$$

3. **Constraints during whole half-stride:**
   The following constraints must hold for all times during SLS resp. DLS.
   – *stability:*
   ZMP lies in convex hull of contact points, i.e. distance from ZMP to convex hull HULL of contact points is negative (for both SLS and DLS):

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

   – *swing foot orientation:*
   the swing foot rotational position must not divert too far from its starting and ending configuration, i.e. norm of vector **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 (differential-algebraic set of equations) 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 con-
straints are not violated [5, 19]. 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 rota-
tional contact forces $\boldsymbol{T}_j = [T_{j,x}\ T_{j,y}\ T_{j,z}]^T$ must satisfy (otherwise a slipping
contact state is entered)

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

with friction coefficients μ_t, μ_d.
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 $\boldsymbol{T}_j = [T_{j,x}\ T_{j,y}\ T_{j,z}]^T$,

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

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 2).
– *discontinuity of velocity states:*
the jump in the generalized system velocities due to an inelastic collision
with the ground are calculated using a collision dynamics algorithm [9]

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

– *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 deter-
mines the foot position **fp** and orientation **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 $\mathbf{I}_3$:

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

4 Gait Generation by Numerical Optimal Control

4.1 Measures for stable locomotion

The difficult task of maintaining stability of fast legged locomotion has been a main obstacle in the construction of such systems. The notion of *static stability*, often used to enforce *postural stability*, does not suffice for fast motion. Static stability requires the ground projected center of mass (GCoM) to lie within the *support polygon*, the convex hull about the leg's contact points. This highly conservative measure generally results in very slow legged motions. The notion of *dynamic stability* is required for faster legged motion. A dynamically stable gait is one without static stability that is sustainable indefinitely [7]; however, an adequate measure suitable for gait generation and control design is not currently available.

The Center-of-Pressure (CoP), equivalent to the Zero-Moment-Point (ZMP), is a point on the ground where the net vertical ground reaction force acts. This point has often been used in previous research efforts [14] to provide a dynamic measure for postural stability by computing its distance to the support polygon boundary. Instability occurs when the CoP reaches the boundary, then a change in the system's contact condition generally occurs, and the system will begin to rotate about that edge. The CoP's deficiency is that it always remains within the contact polygon, even during periods of instability and does not provide information as to neither the degree nor the direction of postural instability.

The Foot-Rotation-Indicator (FRI) [7], the point on the ground where the net vertical contact force would have to act to keep the foot stationary, is more informative than the CoP. It coincides with the CoP when under the foot's contact surface. When not under the surface, its location gives information about the degree and direction of postural instability. When multiple feet are in contact, individual foot instabilities are indicated by the FRI, though not by the system-wide net CoP should they occur within the support polygon. In the moment a foot's FRI exits its ground contact surface, regardless of whether it continues in the support polygon, the foot changes its contact condition and rotates about that edge thus changing the system's dynamic behavior.

The FRI is calculated as follows. It is assumed that contact forces $\boldsymbol{f_c}$ cannot be measured with sensors and must be deduced from a dynamic equilibrium equation,

$$\boldsymbol{f_c} = -(J_c \mathcal{M}^{-1} J_c^T)^{-1} Q \dot{V}_c \,,$$

where $Q\dot{V}_c$ represents the accelerations of the unconstrained system along the constrained motion DoF at some reference point $\boldsymbol{p_r} \in \mathbb{R}^3$ on the foot's contact surface. In the case of flat foot contact, 6 motion DoF are constrained (3 linear and 3 rotational) so that $\boldsymbol{f_c} = [N_x \, N_y \, N_z \, F_x \, F_y \, F_z]^T$. The FRI point $\boldsymbol{p_f} \in \mathbb{R}^3$ is the point where an equivalent force $\boldsymbol{f_c}$ may be applied on the foot and for which $\{N_x = 0, N_y = 0\}$. This can be calculated from the spatial transformation of a force acting on a rigid body. Let $\boldsymbol{p_r} = [p_{r,x} \, p_{r,y} \, 0]^T$.

$$\boldsymbol{p_f} = [p_{r,x} - N_y/F_z \quad p_{r,y} + N_x/F_z \quad 0]^T \,. \tag{3}$$

This point is calculated for each foot and constrained to lie inside the foot contact surface during an optimization. Its distance to a central position may also be minimized in the performance for optimal dynamic postural stability.

4.2 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. We list here three performance indices currently used in our humanoid gait generation investigations.

Postural Stability Performance: Distance in the ground plane between foot i's FRI point $_i\boldsymbol{p_f}$ and a central reference point under the foot $_i\boldsymbol{p_r}$

$$\boldsymbol{J}_{s1}[\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{u}] = \int_0^{t_f} \sum_i \left(\frac{_iN_x^2 + {_i}N_y^2}{_iF_z^2} \right) \, \mathrm{d}t \tag{4}$$

where $\frac{_iN_x^2 + {_i}N_y^2}{_iF_z^2} = (_ip_{f,x} - {_i}p_{r,x})^2 + (_ip_{f,y} - {_i}p_{r,y})^2$.

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 [16]. The integral of this value over a gait period is

$$\boldsymbol{J}_{e1}[\boldsymbol{u}] = \frac{1}{s} \int_0^{t_f} \sum_{i=1}^N R_i \left(\frac{u_i}{G_i K_i} \right)^2 \, \mathrm{d}t \tag{5}$$

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 ϵ as used in [8] 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

$$\boldsymbol{J}_{e2}[\dot{\boldsymbol{q}}, \boldsymbol{u}] = \int_0^{t_f} \frac{\sum_{i=1}^N |u_i \dot{q}_i|}{mgv} \, , \tag{6}$$

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 (4) 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.

Numerical optimization tools have advanced sufficiently [23] such that the many modeling and stability constraints can be incorporated into the problem formulation together with a relatively complete dynamical model so as to obtain truly realistic energy-efficient, stable and fast motions. The optimization approach is based on a discretization of the control problem in time using direct collocation and its subsequent formulation as a nonlinear programming problem (NLP) then solved with a sparse sequential quadratic programming algorithm [6].

The optimization of the stability or energy performance indices subject to the first order form of the system dynamics $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t)$, $\boldsymbol{x} = (\boldsymbol{q_I}, \dot{\boldsymbol{q_I}})$, $\boldsymbol{q_I}$ and $\boldsymbol{u}$ from Eq. (2), and constraints leads to optimal control problems. Their solution delivers optimal open loop trajectories $\boldsymbol{x}^*(t)$, $\boldsymbol{u}^*(t)$, $0 \leq t \leq t_f$. The method DIRCOL [23] uses the method of sparse direct collocation and approximates the states $\boldsymbol{x}$ with spline functions and the controls $\boldsymbol{u}$ with linear functions on a discrete and successively refined time grid. The method is equipped to handle the complexities of the walking problem: unknown liftoff times, different ground contact combinations for the legs, discontinuous states at collision times of the legs with the ground, switching dynamics, and actuation limits.

4.3 Computation of humanoid reference trajectories
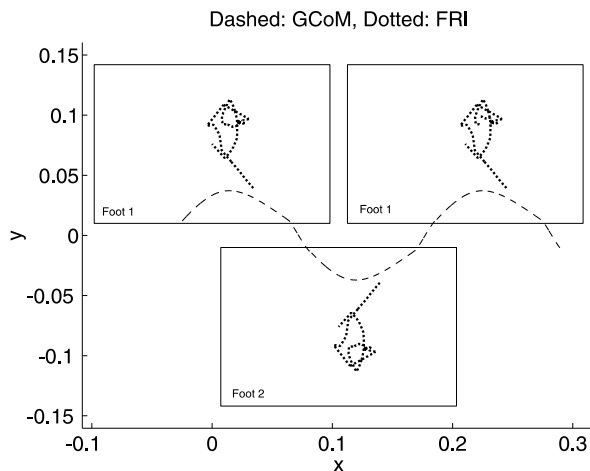
Fig. 2. GCoM and individual foot FRI trajectories during two steps of an optimized statically stable walk.

Inverse kinematic algorithms that were developed for the humanoid prototype in order to compute the reduced dynamics (2) also facilitated the generation

of heuristic joint angle and angle velocity reference trajectories satisfying the physical modeling constraints (Sect. 3.1). 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 (5) subject to the statically stable and dynamic postural stability nonlinear constraints (Sect. 4.1). 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 (Sect. 3.1). An optimization using 44 time grid points required 1584 NLP variables (Sect. 4.2) 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 Fig. 2. Note that the system remains statically stable and that the FRI points remain centered about the middle of their respective foot contact surfaces.

5   Online Compensation

When precalculated optimal control trajectories are applied in practice there usually occur some deviations of stability criteria or constraints due to modeling errors, link flexibilities, gear loss, backlash, joint control errors, and external disturbance forces acting on the robot from the environment. These may result in a degraded walking performance. Commonly, a sensor-based control strategy has to be applied to cope with such deviations.

In this section a method termed *Jacobi Compensation* is proposed [28], which modifies precalculated trajectories in selected task coordinate directions in order to reduce stability criteria deviation and thereby improve walking performance. Task coordinates can be selected Cartesian directions of e.g. the hip coordinate or other task-dependent criteria such as the projected CoM, etc.

The goal of the method is to move a specific set of coordinates $\boldsymbol{p}_c \in \mathbb{R}^{m_c}$ of points on the humanoid, e.g. the center of the hips or an ankle, in the direction $\Delta\boldsymbol{p}_c$ in Cartesian space to reduce deviations. The joint angles $\boldsymbol{q}_t$, e.g. obtained from a precalculated trajectory, are modified by $\Delta\boldsymbol{q} = \boldsymbol{h}(\Delta\boldsymbol{p}_c)$, where $\boldsymbol{h}(\cdot)$ transforms the Cartesian motion $\Delta\boldsymbol{p}_c$ into a joint space motion $\Delta\boldsymbol{q}$. As shown in Fig. 3, this correction $\Delta\boldsymbol{q}$ is linearly superimposed with the joint configuration $\boldsymbol{q}_t$ resulting in a new posture $\boldsymbol{q}_d = \boldsymbol{q}_t + \Delta\boldsymbol{q}$ of the robot.
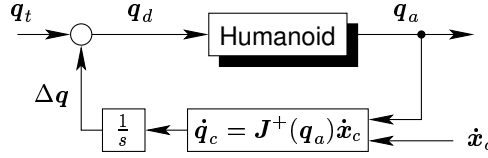
Fig. 3. Jacobi Compensation: The precalculated trajectory is modified so that the motion of one part of the body is increased in direction $\boldsymbol{p}_c$.

The relationship between Cartesian motion and joint space motion is described by the Jacobian

$$\boldsymbol{J}(\boldsymbol{q_a}) = \left[\frac{\partial \boldsymbol{p}_c}{\partial q_{a1}} \cdots \frac{\partial \boldsymbol{p}_c}{\partial q_{aN}} \right] \in \mathbb{R}^{m_c \times N} \ ,$$

a function of the actual joint angles $\boldsymbol{q}_a \in \mathbb{R}^N$ which maps the velocity $\dot{\boldsymbol{q}}_c$ in joint space to the velocity $\dot{\boldsymbol{p}}_c \in \mathbb{R}^m_c$ in Cartesian space according to

$$\dot{\boldsymbol{p}}_c = \boldsymbol{J}(\boldsymbol{q_a}) \, \dot{\boldsymbol{q}}_c. \tag{7}$$

To invert this relationship the pseudoinverse $\boldsymbol{J}^{\#} := \boldsymbol{J}^T (\boldsymbol{J}\boldsymbol{J}^T)^{-1}$ minimizing the Euclidian norm $\|\dot{\boldsymbol{q}}_c\|_2$ is used to obtain the joint motion; here, $m_c < N$ is assumed for existence of a solution, i.e. the motion modification is along less task coordinates $\boldsymbol{p}_c$ than degrees-of-freedom N of the system. The velocity for correction of deviations is the obtained as

$$\dot{\boldsymbol{q}}_c = \boldsymbol{J}^{\#}(\boldsymbol{q_a}) \, \dot{\boldsymbol{p}}_c \ , \tag{8}$$

which is integrated to obtain the position modification $\Delta\boldsymbol{q}$ in joint space. Superimposing it with the precalculated trajectory $\boldsymbol{q}_t$ allows one to adapt the trajectory to the actual requirements, cf. Fig. 3.

Depending on the control problem there exist a variety of possibilities to compute the correction velocity $\dot{\boldsymbol{p}}_c$. For the experiments described in Sect. 6, the velocity has been chosen proportional to the control error $\dot{\boldsymbol{p}}_c = K \, \Delta\boldsymbol{p}_c$ of the task coordinates, where K is a positive definite (und usually diagonal) matrix.

Applications of this method are plentiful: In the experiments in Sect. 6 the method has been used to alter the posture of the robot and thus modify precalculated trajectories to improve walking stability and performance. Other applications include the possibility to adapt precalculated gait trajectories to fit for walking on slopes. If the robot walks up a slope, the center of mass can be slightly shifted to the front by Jacobi Compensation. Apart from that modification the trajectories can remain unchanged, hence less trajectories in a step data base need to be precalculated in advance.

6 Experimental Results

In the following, results of three experiments are described: In the first experiment, the trajectories generated according to Sect. 4.3 are scaled in duration

and applied to the humanoid without further modification as reference trajectories to local joint PD position controllers. The resulting walking performance is sometimes not stable. To improve the precalculated trajectories the Jacobi compensation method presented in Sect. 5 is used to modify certain cartesian task coordinate points of the robot in the second experiment. The last experiment shows walking performance with completely unmodified trajectories.

6.1 Trajectory Following Using Slow Trajectories

The precalculated trajectories obtained by the algorithms discussed in Sect. 4.3 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. Fig. 4 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.
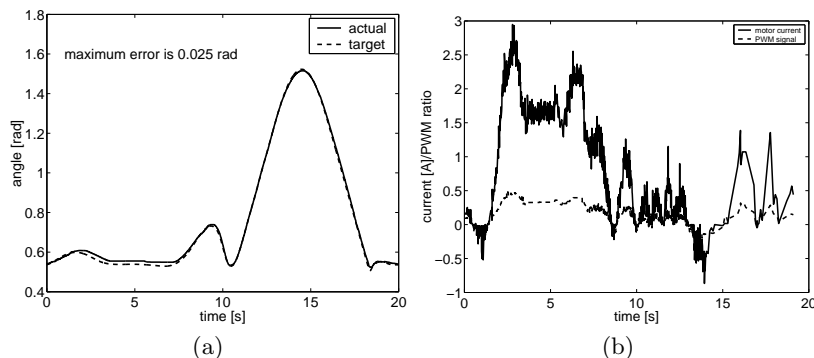
Fig. 4. First experiments with trajectory following control (knee joint of left leg).

From Fig. 4(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, see Appendix A for more implementation details. The corresponding motor current (solid) and PWM ratio (dashed) are shown in Fig. 4(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. In an attempt to alleviate the errors in the torso inertial reference trajectory, a teach-in Jacobi Compensation (cf. Sect. 5) strategy has been applied as follows.

6.2 Jacobi Compensation on Slow Trajectories

In experiments with precalculated trajectories it turned out that the CoM is not shifted sufficiently far over the supporting leg.

To improve stability of the gait Jacobi compensation has been applied. The desired Δp_c has been trained in a teach-in cycle. The precalculated trajectory is stopped every 2 s and the operator modifies Δp_c or directly at the joint level a Δq to achieve a statically stable trajectory point by keyboard commands. These trained modifications of the precalculated trajectory are then linearly interpolated and superimposed with q_t during normal operation. The improvement of walking behavior, in particular stability, is significant as a result from this manual teach-in compensation method.

Fig. 5(a) again shows the desired and the measured trajectory of the left knee. As the robot now has to support its complete weight by the knee, the control error is higher than before; the maximum error is 0.042 rad. From Fig. 5(b) one can see, that the joint has reached its maximum load capabilites as the motor current saturates and the PWM ratio is close to 100%. This is not surprising, as the robot has been designed for fast locomotion where the required motor torque is smaller than the torque necessary for statically balancing on one leg.

(a)   (b)
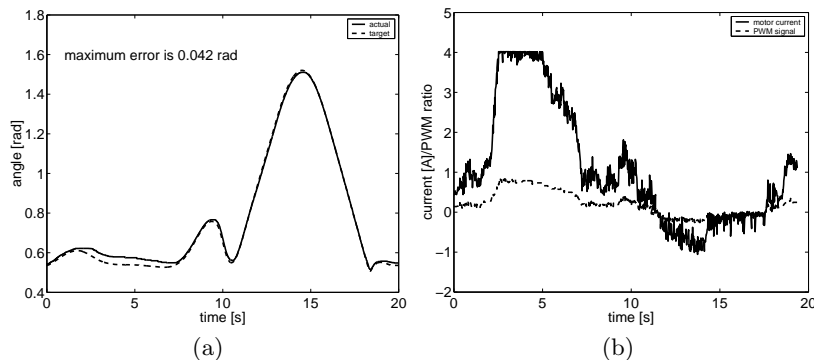
Fig. 5. Trajectory following with manually modified trajectories (knee joint of left leg).

The effect of the compensation is shown in Fig. 6(a) where the target trajectory and the modified trajectories of the left ankle joint are presented. The

compensation mainly affects the support phase, where the robot has to be balanced on the left leg and hence small errors in the joint angle degrade static stability. Fig. 7 shows snapshots of some stages in a walking sequence.
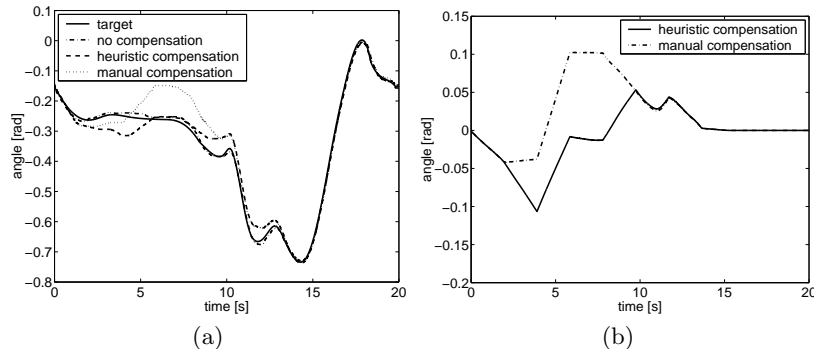
Fig. 6. Experimental result with teach-in compensation: (a) knee joint of left leg; (b) compensation trajectory Δq_3 for knee joint.

The success of this experiment demonstrating the applicability of Jacobi Compensation is regarded a first step towards sensor based stability control. In the following, walking performance with an unmodified trajectory with a duration of 6 s computed as discussed in Sect. 4.3 is described.

6.3   Trajectory Following Using Fast Optimal Control Trajectory

The result of conducting the same experiment as in Sect. 6.1 with a faster, unscaled trajectory is shown in Fig. 8. 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 Fig. 8a. 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 shown in Fig. 5. This shows that the robot dynamics have successfully been exploited in the gait generation process.

7   Conclusions

This paper presents the control development procedure to accomplish stable walking in a humanoid robot. A key fundamental element is the approximation of an accurate dynamic model for the robot. Gait trajectory generation is achieved by numerically solving an optimal control problem subject to numerous physical and non-physical constraints. The cost function for the optimization is a weighted combination of several objectives penalizing high energy consumption and postural instabilities. The optimal gait trajectories are consequently tested
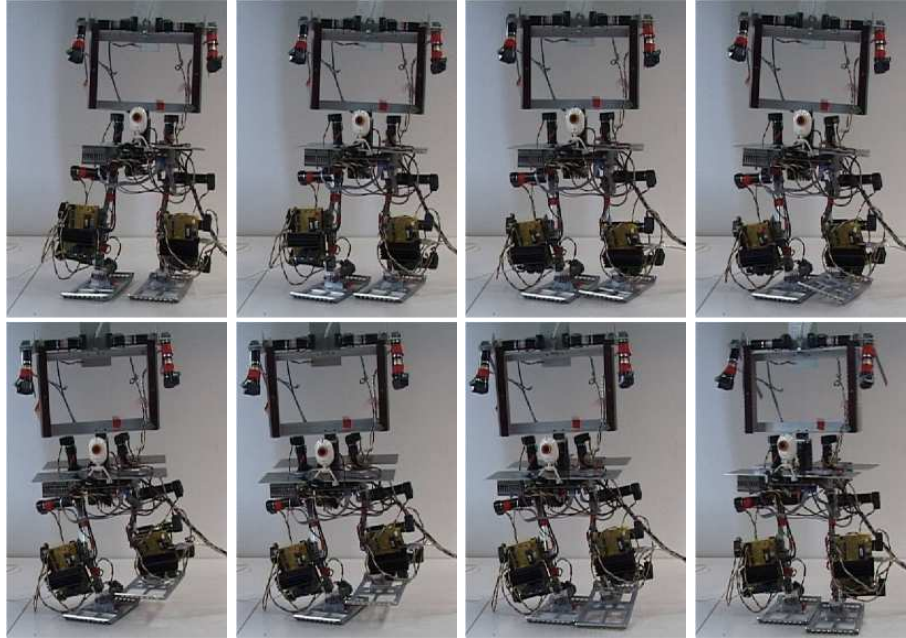
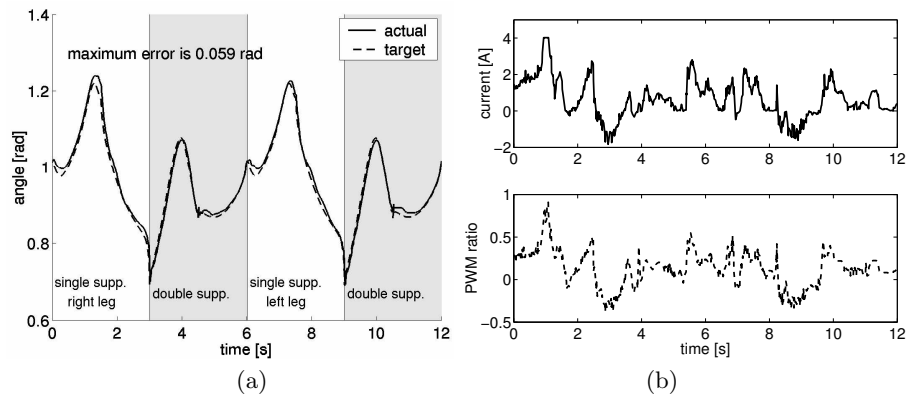**Fig. 7.** Snapshots of a step sequence.

(a)

(b)

Fig. 8. Double stride, duration per stride 6 s.

on the developed humanoid robot prototype hardware in experiments. A gait trajectory modification strategy is used on-line to robustify the calculated reference trajectories and is a first step towards sensor based stability control. The experiments point out the importance of exact modeling as the system stability while following generated trajectories increases significantly with the quality of the model. Although very accurate trajectories cannot replace online stability control, it can significantly reduce the amount of corrective control effort allowing the system to operate closer to calculated optimal performance levels. Future development will aim towards more accurate modeling and faster gait trajectories. Furthermore, sensor-based control using accelerometers, gyroscopes, and an inclination sensor as well as foot contact force sensors will be used to improve the walking performance with respect to stability and increased speed of the robot.

A   Hard- and Software System Architecture

This appendix discusses the hardware realization of the 17 DoF humanoid robot prototype hardware, the control, computing and software environment used to control the robot.

A.1   Hardware Design and Software Environment

For simplicity of design, one important aim was to assemble the robot from as many identical modules as possible. Therefore all joints are variations of an elementary joint: The shaft of the motor-gear-unit is fixed to an L-shaped base plate. Attached to the axis of the motor is a lever arm whose far end is connected to the base plate of the next joint. Though this lightweight construction spares additional bearings the motor axis is still sufficiently stable to support the exerted load. For the links between the motors ordinary steel with a rectangular profile of $3 \times 15 \, \text{mm}^2$ is used. This slightly flexible construction was chosen to incorporate an additional mechanical shock absorption mechanism damping the impact of the feet hitting the ground. Joints requiring more than one DoF, like the ankle or the hip, are realized by two or more sequential motors with orthogonally oriented axes of rotation, see Fig. 9.

The mechanical robot construction arising from linking these elementary modules is shown in Fig. 9. The robot carries 3 batteries as power supply, two

of them visible on the picture at the height of the hips and below the navel joint. The third battery is located symmetrically behind the hips together with an ATX power supply for the main PC mounted on the upper body. The chosen Sony BP-L90A batteries provide a capacity of 90 Wh each, hence allowing for approximately 45 min autonomous walking.

The motors are chosen in consideration of the best use of power supply and weight of the necessary batteries and the remaining robot to gain the optimal torque for the proposed tasks. By optimizing under these constraints motors with 20 W and 42 V have been chosen [11].
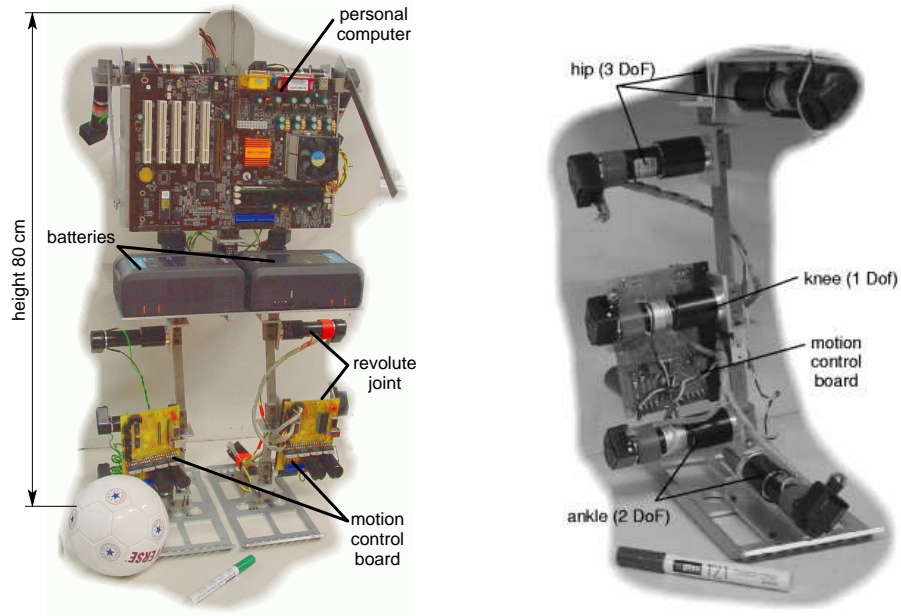
Fig. 9. Mechanical realization of the robot (left) and left leg rear view (right).

The motors are accessed using the microcontroller board shown in Fig. 9 which was developed at the Control Systems Group in Berlin [4]. The core of the board is a Motorola MC68HC908BD48 8 bit microcontroller including: 3 USB endpoints, a 6 channel A/D converter and a 16 channel pulse-width modulator (PWM). These PWM signals are amplified by a National LMD18200 mosfet H-bridge, hence a motor load of up to 3 A at 55 V is admissible. The actual position of a motor is determined by evaluating the signals of pulse encoders attached to each motor using US Digital LS7266 quadrature decoders. To each board weighing 170 g, 4 motors can be connected. Hence this board represents a lightweight motion control solution.

With these components position PD control loops are implemented on the microcontroller. The A/D converters on the microcontroller are wired to sense

the motor current which also allows to drive the motors with current control. These motion control boards are linked with the main PC on the robot via an USB connection. Through this link, new control inputs are delivered to the board retrieving the measured values at the same transfer stage.

For a main computer carried along by the robot a fullsize ATX mainboard is used. Being similar in weight compared to most full sized single board computers with equivalent computational power, a fullsize computer can be tolerated. The PC is equipped with an Athlon 1300 MHz CPU providing enough computational power for motion control and additional tasks such as object recognition using a camera system.

To obtain a graphical interface to the robot and the motion control boards, a MATLAB S-function has been implemented allowing to drive the robot from within the SIMULINK simulation environment. Experiments showed, that this rapid prototyping environment handles well the soft realtime constraints for the outer control loop without the need for a hard realtime environment. This may be attributable to the efficient task scheduling capabilities of the Linux kernel.

References

1. J.-M. Bourgeot, N. Cislo, and B. Espiau, "Path-planning and tracking in a 3D comlex environment for an anthropomorphic biped robot," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*, (Lausanne, Switzerland) (2002) 2509–2514
2. J. Denk and G. Schmidt, "Synthesis of a walking primitive database for a humanoid robot using optimal control techniques," in *Proceedings of the IEEE/RAS International Conference on Humanoid Robots*, (Tokyo, Japan) (2001) 319–326
3. R. Featherstone, D. Orin: "Robot Dynamics: Equations and Algorithms." *IEEE Intern. Conf. on Robotics and Automation* (2000) 826–34
4. K. Gänger, "Implementieren einer Regelungselektronik fuer Humanoide Fussball-Roboter (*in German*)," tech. rep., Control Systems Group Berlin (2002)
5. M. Gienger, K. Löffler, F. Pfeiffer: "Towards the Design of a Biped Jogging Robot." *IEEE Int. Conf. on Robotics and Automation* (2001) 4140–45
6. Gill, P.E.; Murray, W.; Saunders, M.A.: "User's Guide for SNOPT 5.3: a Fortran Package for Large-Scale Nonlinear Programming." Mathematics Department, Univ. of California, San Diego (1999)
7. A. Goswami: "Postural Stability of Biped Robots and the Foot-Rotation Indicator (FRI) Point." *Intern. J. of Robotics Research* **18**(6) (1999) 523–533
8. P. Gregorio, M. Ahmadi, M. Buehler: "Design, Control, and Energetics of an Electrically Actuated Legged Robot." *IEEE Transactions on Systems*, Man and Cybernetics, Part B **27**(4) (1997) 626–634
9. 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, Univ. of California, San Diego, U.S.A. (1999)
10. M. Hardt, O. von Stryk: "Dynamic modeling in the simulation, optimization and control of legged robots." To appear in: *Zeitschrift für Angewandte Mathematik und Mechanik* (2003)
11. M. Hardt, D. Wollherr, M. Buss, and O. von Stryk, "Design of an autonomous fast-walking humanoid robot," in *Proc. of the 5th Intern. Conf. on Climbing and Walking Robots*, (Paris, France) (2002) 2491–2496

12. Q. Huang, S. Kajita, N. Koyachi, K. Kaneko, K. Yokoi, T. Kotoku, H. Arai, K. Komoriya, and K. Tanie, "Walking pattern and actuator specification for a biped robot," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*, (Kyongju, Korea) (1999) 1462–1468

13. Q. Huang, K. Kaneko, K. Yokoi, S. Kajita, T. Kotoku, N. Koyachi, H. Arai, N. Imamura, K. Komoriya, and K. Tanie, "Balance control of a biped robot combining off-line pattern with real-time modification," in *Proceedings of the IEEE International Conference on Robotics and Automation*, (San Francisco, CA) (2000) 3346–3352

14. Q. Huang, K. Yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi, K. Tanie: "Planning Walking Patterns for a Biped Robot." *IEEE Trans. on Robotics and Automation* **17** (2001) 280–89

15. S. Kajita, T. Yamaura, and A. Kobayashi, "Dynamic walking control of a biped robot along a potential energy conserving orbit," *IEEE Transactions on Robotics and Automation*, vol. 8 (1992) 431–438

16. H. Kimura, I. Shimoyama, H. Miura: "Dynamics in the Dynamic Walk of a Quadruped Robot." *Advanced Robotics* **4**(3) (1990) 283–301

17. M. Inaba, F. Kanehiro, S. Kagami, and H. Inoue, "Two-armed bipedal robot that can walk, roll over and stand up," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS* (1995) 297–302

18. O. Lorch, A. Albert, J. Denk, M. Gerecke, R. Cupec, J. F. Seara, W. Gerth, and G. Schmidt, "Experiments in vision-guided biped walking," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*, (Lausanne, Switzerland) (2002) 2484–2490

19. F. Pfeiffer, C. Glocker: "Multibody Dynamics with Unilateral Contacts.", Wiley Series Nonlinear Science, New York (1996)

20. RoboCup Federation: "RoboCup Humanoid League 2002 Rule.",
`http://www.robocup.org/regulations/humanoid/rule_humanoid.htm`

21. A. Jain, G. Rodriguez: "Computational Robot Dynamics Using Spatial Operators." *IEEE Intern. Conf. on Robotics and Automation* (2000) 843–49

22. M.W. Spong, M. Vidyasagar: Robot Dynamics and Control. J. Wiley & Sons (1989)

23. O. von Stryk "User's Guide for DIRCOL Version 2.1," Simulation and Systems Optimization Group, Technische Universität Darmstadt,
`http://www.sim.informatik.tu-darmstadt.de/sw/dircol`

24. K. Tani, K. Ikeda, T. Yano, S. Kajita, and O. Matsumoto, "The concept of model free robotics for robots to act in uncertain environments," in *Proc. of the IEEE/Tsukuba Intern. Workshop on Advanced Robotics*, (Tsukuba, Japan), (1993) 85–90

25. D. Wollherr, M. Hardt, M. Buss, and O. von Stryk, "Actuator selection and hardware realization of a small and fast-moving, autonomous humanoid robot," in *Proc. of the IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems IROS*, (Lausanne, Switzerland) (2002) 391–398

26. F. Yamasaki, K. Endo, M. Asada, and H. Kitano, "A control method for humanoid biped walking with limited torque," in *RoboCup 2001* (A. Birk, S. Coradeschi, and S. Tadokoro, eds.), Springer (Berlin, Heidelberg) (2002) 60–70

27. Y. F. Zheng, J. Shen, and F. R. Jr. Sias, "A motion control scheme for a biped robot to climb sloping surfaces," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2 (1988) 814–816

28. T. Sugihara, and Y. Nakamura, "Whole-body cooperative balancing of humanoid robot using COG Jacobian," in *Proc. of the IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems IROS*, (Lausanne, Switzerland) (2002) 2575–2580