2012 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 7-12, 2012. Vilamoura, Algarve, Portugal

Optimization-based generation and experimental validation of optimal walking trajectories for biped robots

Alexander Werner, Roberto Lampariello and Christian Ott

***Abstract*— In this paper the generation of walking gaits for biped robots is addressed as a nonlinear optimization problem. The latter presents an efficient formulation, which only requires parameterizing the joint states and does not require to integrate the equations of motion. The results of the optimization are applied to a real robot, with the aid of a suitable stabilizing controller. The final gain in optimized cost is assessed, for the real system. The experimental results confirm the effectiveness of the method.**

I. Introduction

The task of generating walking gaits for bipedal robots was addressed extensively in the literature in recent years. The problem may be formulated as a control problem [1]–[4] or alternatively as a nonlinear optimization problem [5]–[8]. We address this latter formulation here in detail. The aim of this paper is to assess the overall gain in the cost function, when an optimal solution is commanded on a real system, which introduces modeling uncertainties and closed-loop controller corrections. We find that such analyses are hardly found in other optimization-related works.

Different approaches may be found in the formulation of the related optimization problem, from collocation [8] to multiple-shooting [5]. We present here an alternative approach, already applied in [9] to a fixed base robot, based on the sole parameterization of the robot joint states, which does not require integration of the equations of motion.

We also solve the global optimization problem, which is hardly addressed in the literature. As is well known, local minima are abundant in highly constrained nonlinear optimization problems [10], as is shown here to be the case for the walking task. We chose here to minimize the mechanical energy, to arrive at a statistically consolidated global optimum, for any given desired walking gait. Initial guesses are provided by means of an RRT-algorithm.

To finally determine how useful the optimization is, when applied on a real system, we perform closed-loop trajectory tracking experiments with the DLR Biped (see Fig. 1). The trajectory parameterization is adapted to the controller structure. At this stage, we can assess the loss in the cost function, which arises from modeling errors in the model used in the off-line optimization procedure (e.g. joint compliance is not included), and also from the inevitable implementation of a closed-loop controller.

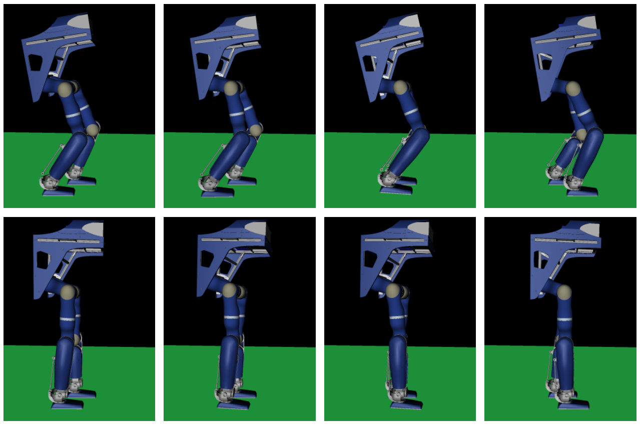

Fig. 1. Simulation environment for the DLR Biped robot: two samples of an optimized trajectory with different step length. Top row: result with constraint knee joint angle. Bottom row without constraint.

In [7] Djoudi *et al.* describe a way to find torque-optimal robotic gaits for a planar system. Based on forth-order polynomial functions five joint angles are defined. The dynamic model describing the single support phase is augmented by impact equations emulating an instantaneous support change. Due to the simple model structure the system has no actuation at the feet, resulting in an underactuated system.

Denk and Schmidt proposed a solution for three-dimensional robotic gaits with a rigid body model in [8] [11]. The concept includes the modeling of single support, double support, heel- and toe-motions. It is solved via a direct collocation approach discretizing both torque input and joint trajectories. Feasibility is expressed via constraints on the zero-moment point (ZMP). The concept is designed to run on the humanoid robot Johnnie. The parameterization is done via cubic splines for the joint angle trajectories and piecewise linear functions for the joint torques.

Buschmann *et al.* describe in [6] a method to generate full-body walking motions, parameterized with reduced degrees of freedom. The motion is described in the coordinates of the center of mass (COM) and joint angles determining the configuration of the active upper body parts. The trajectory of the swing leg is not part of the optimization problem. Cubic splines are used to parameterize this coordinate set. The dynamics are calculated with a free-floating three-dimensional rigid body model. The optimization process makes use of algebraic

*All authors are with the Institute of Robotics and Mechatronics (DLR). E-mail: firstname.lastname@dlr.de

978-1-4673-1736-8/12/$31.00 ©2012 IEEE

derivatives which yield a six-times lower execution time compared to numerical derivatives.

In [5] Schultz *et al.* address the problem of optimizing the running gait of a multibody model of the human body. An optimal control problem is formulated that minimizes a weighted sum of joint torques and of torque variations. It is solved using an efficient direct multiple-shooting algorithm, which however involves integration of the equations of motion and provides generally discontinuous torque profiles. Results are provided in simulation.

In [12] pre-calculated humanoid paths, provided by kinematic planning methods, are optimized with respect to joint torques, to achieve smooth and stable human-like motions. Function gradients are derived analytically for greater numerical efficiency. However, the optimization problem only addresses the torque time profiles, leaving the planned trajectory unaltered.

The main contribution of this paper is to provide a detailed analysis of the issues involved in nonlinear optimization for generating walking gaits and an experimental evaluation of the final optimal gait. The paper is then structured as follows: the rest of Section I presents a literature survey, while Section II presents the problem statement and the modeling of the robot. Section III describes the formulation of the optimization problem and Section IV the method of solution. Section V analyzes the results and includes a discussion, while Section VI presents the conclusions. The adopted notation is such that all vectors are expressed in the inertial frame of reference unless otherwise noted.

II. Problem statement

The problem of walking is split into periods of single support phases, defined as the trajectory between the lift-off and touch-down of the swing foot. A double support phase is not included. This way, it is possible to reduce the problem to the smallest possible unit which is to be optimized.

The configuration of the robot is described in joint angles $\boldsymbol{q} \in \mathbb{R}^N$ with N as the number of joints and $t = [0, t_f]$ as time. The coordinate systems $\mathcal{L}$ and $\mathcal{R}$ are attached at the sole of each foot. Let $\boldsymbol{H}_{LR} \in SE(3)$ define the relative frame between right and left foot. A minimal representation of position and orientation is given by $\boldsymbol{x}_{LR} \in \mathbb{R}^3$ and $\boldsymbol{\phi}_{LR} \in \mathbb{R}^3$.

To simplify the description, we assume the left foot as the stance foot and the right foot as the swing foot. Reaction forces and torques created at the contact point are denoted by $\boldsymbol{W}_L$ for the wrenches acting at the $\mathcal{L}$ coordinate system. For periodic walking and a symmetric robot, it is easily possible to calculate a trajectory pattern where stance and swing leg have changed their roles.

In order to describe the step pattern on the ground, the stride length l and width w are introduced. The step starts at $t = 0$ and ends at $t = t_f$. The robot is oriented in the x-direction and the footsteps are located on the ground, as shown in Fig. 2. Without loss of generality the feet are set to be aligned at both the start and end of the step.

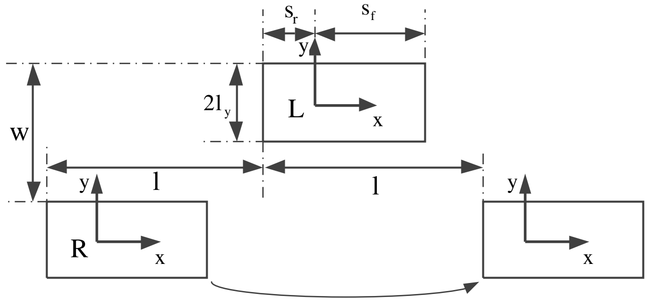

Fig. 2. Step geometry with measures w and l, coordinate systems $\mathcal{L}$ and $\mathcal{R}$. Foot geometry with measures s_r,s_f and l_y.

The kinematics of the robot during the single support phase is a pure serial chain of links. As the left foot resides at a fixed point during the whole trajectory it is convenient to treat the robot as a fixed base system. This way the joint angles $\boldsymbol{q}$ are a valid set of minimal coordinates. In order to also derive the expression for the constraint force at the foot, we first refer to a free floating robot dynamic model, as follows:

$$\begin{bmatrix} \boldsymbol{M}_b & \boldsymbol{M}_{bm} \\ \boldsymbol{M}_{bm}^T & \boldsymbol{M}_m \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{v}}_b \\ \dot{\boldsymbol{\omega}}_b \\ \ddot{\boldsymbol{q}} \end{bmatrix} + \underbrace{\begin{bmatrix} \boldsymbol{C}_b \\ \boldsymbol{C}_m \end{bmatrix} + \begin{bmatrix} \boldsymbol{g}_b \\ \boldsymbol{g}_m \end{bmatrix}}_{\boldsymbol{h}(\boldsymbol{q},\dot{\boldsymbol{q}})} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{\tau} \end{bmatrix} + \begin{bmatrix} \boldsymbol{W}_L \\ \boldsymbol{0} \end{bmatrix}, \tag{1}$$

where $\boldsymbol{v}_b$ and $\boldsymbol{\omega}_b$ refer to the velocity of the free floating base, the first term on the right hand side refers to the actuation forces $\boldsymbol{\tau}$ ($\boldsymbol{0}$ for the base) and the second to the constraint forces. $\boldsymbol{W}_L$ expresses the constraint force on the base for a fixed base system. To derive its expression we apply the algebraic constraint written as:

$$\begin{bmatrix} \boldsymbol{v}_b \\ \boldsymbol{\omega}_b \end{bmatrix} = \boldsymbol{0}, \tag{2}$$

which by differentiation and substitution in (1) provides the following expressions:

$$\begin{aligned} \boldsymbol{M}_m(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}_m(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}_m(\boldsymbol{q}) &= \boldsymbol{\tau} \\ \boldsymbol{M}_{mb}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}_b(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}_b(\boldsymbol{q}) &= \boldsymbol{W}_L \end{aligned} \tag{3}$$

III. Optimization Problem

The cost function and constraints described in the following sections define the optimization problem:

$$\begin{gathered} \min_{\boldsymbol{q}(t)} \ \Gamma(\boldsymbol{q}(t)) \\ \boldsymbol{g}(\boldsymbol{q}(t)) = 0 \\ \boldsymbol{h}(\boldsymbol{q}(t)) \leq 0 \end{gathered} \tag{4}$$

where $\boldsymbol{g}(\boldsymbol{q}(t))$ represents the equality and $\boldsymbol{h}(\boldsymbol{q}(t))$ the inequality constraints.

A. Cost Function

The chosen cost function is based on the mechanical energy the robot consumes. It is designed to prevent putting energy into the mechanical system and removing it later, as the drawing of energy does not result in a lower cost. This choice is closely related to the fact that the harmonic drive gears used are hardly backdrivable. The expression for the cost function is defined as follows:

$$\Gamma(\boldsymbol{q}(t)) = \sum_{i=1}^{N} \int_0^{t_f} \begin{cases} \dot{q}_i(t) \cdot \tau_i(t) & \\ 0 & \end{cases} \text{dt} \quad \begin{matrix} \text{if } \dot{q}_i(t)\tau_i(t) \geq 0 \\ \text{if } \dot{q}_i(t)\tau_i(t) < 0 \end{matrix} \tag{5}$$

Differently to the classical squared version $\int (\dot{q}_i\tau_i)^2$, the chosen function is the correct expression of the consumed energy.

B. Equality Constraints

Step geometry is defined by an equality constraint that sets the position of the swing foot at $t = 0$ using l, which is equivalent to the stride length, and w to the lateral distance between the feet:

$$\begin{aligned} \boldsymbol{x}_R(0) &= \begin{bmatrix} -l \\ -w \\ 0 \end{bmatrix} & \dot{\boldsymbol{x}}_R(0) &= \boldsymbol{0} \\ \boldsymbol{\phi}_R(0) &= \boldsymbol{0} & \dot{\boldsymbol{\phi}}_R(0) &= \boldsymbol{0} \end{aligned} \tag{6}$$

The same set of constraints is required for the final position of the swing foot:

$$\begin{aligned} \boldsymbol{x}_R(t_f) &= \begin{bmatrix} l \\ -w \\ 0 \end{bmatrix} & \dot{\boldsymbol{x}}_R(t_f) &= \boldsymbol{0} \\ \boldsymbol{\phi}_R(t_f) &= \boldsymbol{0} & \dot{\boldsymbol{\phi}}_R(t_f) &= \boldsymbol{0} \end{aligned} \tag{7}$$

These constraints restrain the space for optimization, in that there is no impulse transfer possible at the touch-down of the swing-foot.

Continuity of the trajectories is ensured by the following equality constraints. The values $q_{i,start}$ and $q_{i,end}$ are given by the boundary conditions from Section IV-B The following equality constraint also addresses the second derivative to prevent discontinuous torques in the joints:

$$\begin{aligned} q_i(0) &= q_{i,start} & q_i(t_f) &= q_{i,end} \\ \dot{q}_i(0) &= \dot{q}_{i,start} & \dot{q}_i(t_f) &= \dot{q}_{i,end} \\ \ddot{q}_i(0) &= \ddot{q}_{i,start} & \ddot{q}_i(t_f) &= \ddot{q}_{i,end} \end{aligned} \tag{8}$$

Care must be taken that (6),(7) and (8) are compatible.

C. Inequality Constraints

Joint limits for position, speed and torque constraints are implemented with the following box constraints:

$$\begin{aligned} q_{min,i} \leq q_i(t) &\leq q_{max,i} \\ \dot{q}_{min,i} \leq \dot{q}_i(t) &\leq \dot{q}_{max,i} \\ \tau_{min,i} \leq \tau_i(t) &\leq \tau_{max,i} \end{aligned} \tag{9}$$

Self collisions are avoided by restraining the distance between the knees, d_k, and that between the ankle joints, d_a, so that a collision is not possible in the proximity of the optimal trajectories. This is implemented in the following equations:

$$\begin{aligned} 0 &\leq d_a - d_{a,min} \\ 0 &\leq d_k - d_{k,min} \end{aligned} \tag{10}$$

Collisions with the environment are expressed with the help of the distance between the foot edges and the floor. Given the pose of the foot link this can be calculated easily.

The optimizer is likely to find a solution where the foot slides along and very close to the floor, since this requires the least amount of energy. To prevent this situation, the real floor shape is modified such that the floor collision constraint imposes the foot to have a variable distance to the ground. The following equation transforms a vector $\boldsymbol{x}_{edge}$, defining a point on the foot outline described in the local coordinate system, into the inertial coordinate system, where it is denoted $\boldsymbol{x}_e(t)$:

$$\boldsymbol{x}_e(t) = \boldsymbol{H}_{LR}(t) \begin{bmatrix} \boldsymbol{x}_{edge} \\ 1 \end{bmatrix} \tag{11}$$

The inequality constraint then follows as:

$$0 \leq x_{e,z}(t) - f(x_{e,x}(t), x_{e,y}(t)) \tag{12}$$

The function $f(x, y)$ defines the obstacle height at the coordinates (x, y). This function is used to describe the clearance for the whole floor surface with the exception of the start and end footprint.

Contact stability is assured by constraining the reaction wrench $\boldsymbol{W}_L$. This results from the equations modeling friction, balance and unilaterality at the stance foot. The friction element can be ensured by using a friction cone as a model for the maximum forces permissible [8]:

$$\sqrt{F_{L,x}^2 + F_{L,y}^2} + \left| \frac{\tau_{L,z}}{\kappa} \right| \leq \mu F_{L,z} \tag{13}$$

where μ is the friction coefficient between the materials in contact, κ the effective radius and $\boldsymbol{F}_L$ and $\boldsymbol{\tau}_L$ correspond to the force and torque components of the contact wrench, respectively. The unilaterality constraint $0 \leq F_{L,z}$ is included in the friction cone constraint.

The zero-moment point [13] is used to calculate if the reaction force applied to the contact would induce rolling around any of the edges of the foot. With the definition of the outline in Fig. 2 the following inequalities must be fulfilled:

$$\begin{aligned} s_r \leq -\frac{\tau_{L,y}}{F_{L,z}} &\leq s_f \\ -l_y \leq \frac{\tau_{L,x}}{F_{L,z}} &\leq l_y \end{aligned} \tag{14}$$

where s_r,s_f and l_y are defined in Fig. 2.

IV. Solution Method

In our direct optimization method, the infinitely dimensional functions $\boldsymbol{q}(t)$ are reduced by an end-to-end parameterization $\tilde{\boldsymbol{q}}(t, \boldsymbol{b})$, that depends on a finite dimensional vector $\boldsymbol{b} \in \mathbb{R}^{N_b}$ (see Section IV-A). Differently from the classical single shooting formulation, we do not require integration of the equations of motion, since these are solved exactly for any choice of $\boldsymbol{b}$. As a result of this, the input torque $\boldsymbol{\tau}$ is a continuous function and we eliminate the problem, generally found in shooting methods, in obtaining sufficiently accurate derivatives for the optimization routine. This method is inspired by a solution to optimal robotic ball catching [9].

In order to incorporate the path inequality constraints $\boldsymbol{h}(\boldsymbol{b})$ into the non-linear programming problem, a set of via points

$$\boldsymbol{v} = \{t_0 < t_1 < ... < t_N = t_f\} \tag{15}$$

is defined. The infinite dimensional path inequality constraints (5) - (14) are reformulated into a vector of inequality constraints:

$$\begin{bmatrix} \boldsymbol{h}\left(\tilde{\boldsymbol{q}}\left(t_0\right)\right) \\ \boldsymbol{h}\left(\tilde{\boldsymbol{q}}\left(t_1\right)\right) \\ \vdots \\ \boldsymbol{h}\left(\tilde{\boldsymbol{q}}\left(t_N\right)\right) \end{bmatrix} \leq 0 \tag{16}$$

This method enforces the path inequality constraints at points in $\boldsymbol{v}$ only. The constraint values are processed by a sequential quadratic programming based optimization algorithm with numerical derivatives [14].

A. Parameterization

The necessary discretization of the configuration space is done with quartic basic splines [15] transforming the optimal control problem to a non-linear programming problem. From the parameter set $\boldsymbol{b} \in \mathbb{R}^{N_b}$, N_j elements are taken to form the joint parameterization vector $\boldsymbol{B}$ which is an input for the following function:

$$q_i(t) = \frac{1}{6}\begin{bmatrix}1 & u & u^2 & u^3\end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & -1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} B_j \\ B_{j+1} \\ B_{j+2} \\ B_{j+3} \end{bmatrix}, \tag{17}$$

used to parameterize one coordinate in $\boldsymbol{q}(t)$. $B_j \dots B_{j+3}$ is a vector of parameterization values extracted from the whole set $B_1 \dots B_N$ as follows:

$$\begin{aligned} j &= \text{floor}(\tfrac{t}{t_f} \cdot (N_j - 3)) \\ u &= \tfrac{t}{t_f} \cdot (N_j - 3) - j \end{aligned} \tag{18}$$

The basic spline provide a local support and through the inverse dynamics, guarantees a continuous function $\boldsymbol{\tau}(t)$.

B. Boundary Conditions

In order to reduce the number of necessary equality constraints used to ensure continuous joint angles, velocities and accelerations at the boundaries, boundary conditions are used to calculate as many parameters as possible without limiting the search space to optimize the cost-function. This results in combing (8) and (17).

In the case of a step from a pre-defined start to a pre-defined end configuration (joint angles), the equality constraints (6),(7) and (8) are not required. For each joint, three parameters of $\boldsymbol{B}$ per boundary are determined by these pre-defined conditions.

For generation of a periodic gait instead, it is the boundary conditions that determine part of $\boldsymbol{b}$, such that the following equation holds:

$$\left.\begin{aligned} q_{l,i}(0) &= m(i) q_{r,i}(t_f) \\ \dot{q}_{l,i}(0) &= m(i) \dot{q}_{r,i}(t_f) \\ \ddot{q}_{l,i}(0) &= m(i) \ddot{q}_{r,i}(t_f) \\ \\ q_{r,i}(0) &= m(i) q_{l,i}(t_f) \\ \dot{q}_{r,i}(0) &= m(i) \dot{q}_{l,i}(t_f) \\ \ddot{q}_{r,i}(0) &= m(i) \ddot{q}_{l,i}(t_f) \end{aligned}\right\} \text{for } i = 1..\frac{N}{2} \tag{19}$$

$$m(i) = \begin{cases} 1 & \text{joint axis direction lateral} \\ -1 & \text{all other joints} \end{cases}$$

where $q_{r,i}$ and $q_{l,i}$ designate the joints of the legs in the order they appear in the kinematic chain seen from the hip.

C. Global search

The experience shows that local minima are ubiquitous in this description of the problem [10]. It is also worth mentioning that these global minima are not only related to the cost function but also to convergence into feasible areas. It is therefore necessary to carry out a global search to arrive at a global optimum, as shown by the results in Section VI-A.

It is also essential, for convergence efficiency, to provide the optimizer with initial guesses which are in the proximity of a valid trajectory, or which fulfill as many of the constraints as possible. Our first approach is to generate start and end configurations which comply with the equality constraint (6). These are generated by means of the inverse kinematics, as follows. Given the relative position of the feet, six degrees of freedom remain undetermined. These are computed by feeding a uniformly-distributed random position and orientation of the hip (in a valid interval) to the inverse kinematics. From these conditions, part of the initial guess for parameter vector $\boldsymbol{b}$ follows after algebraic manipulation. The remaining parameters in $\boldsymbol{b}$, relative to the optimization of the trajectory, were defined by a straight line connecting the initial and final configurations.

Due to the foot constraint (12), the above approach was found to have a very low convergence rate. Therefore, a valid initial guess for the complete trajectory is first computed with an RRT-based kinematic path planning method [16] (as done in [12]). The resulting sequence of configurations is then fitted to the parameterization functions. This way the self- and floor collision constraints are also satisfied by the initial guess.

V. Stable Trajectory Execution

Due to disturbances and modeling uncertainties it is not possible to execute the optimal walking trajectories by only feed forward torque control or high gain position control. An inner control loop is needed to ensure that the actual ZMP does not reach the borders of the support polygon. In this work we use the ZMP based balance controller from [3] which modifies the COM trajectory to maintain a balanced gait. Fig. 3 shows an overview of the control loop. Given the optimized basic spline parameters, we use (17) to compute the joint angle trajectory $\boldsymbol{q}$, and its derivatives $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$. From (3) we obtain the corresponding joint torques $\boldsymbol{\tau}$, which are used directly as a feed-forward component in the low level position controller from [17].

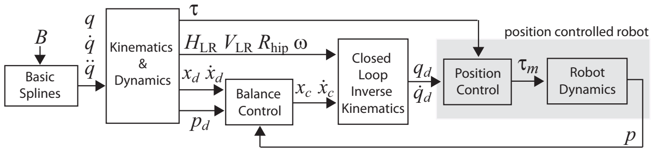

Fig. 3. Closed loop system for online generation of the optimized joint level trajectories via an underlying balancing controller.

In order to use the joint level trajectories in combination with the underlying COM based balance controller, we compute the corresponding trajectories of the COM $\boldsymbol{x}_d \in \mathbb{R}^3$, the swing leg pose $\boldsymbol{H}_{RL}$ and its body twist $\boldsymbol{V}_R \in se(3)$, and the hip orientation $\boldsymbol{R}_{\text{hip}} \in SO(3)$ and its angular velocity $\boldsymbol{\omega} \in so(3)$ via forward kinematics. Moreover, the desired ZMP trajectory $\boldsymbol{p}_d$ is computed from the contact wrench $\boldsymbol{W}_L$ obtained from 3.

The balancing controller from [3] considers the errors between the actual ZMP $\boldsymbol{p}$ and COM $\boldsymbol{x}$ from the planned trajectories $\boldsymbol{p}_d$ and $\boldsymbol{x}_d$ in order to compute a modified COM trajectory

$$\dot{\boldsymbol{x}}_c = \dot{\boldsymbol{x}}_d - k_p(\boldsymbol{p}_d - \boldsymbol{p}) + k_c(\boldsymbol{x}_d - \boldsymbol{x}) \tag{20}$$

with positive scalar gains k_p and k_c [3].

From $\boldsymbol{x}_c$ and the desired trajectories for the swing foot and the hip orientation, the desired joint angle trajectory $\boldsymbol{q}_d$ is computed by a closed loop inverse kinematics algorithm [18]. The resulting joint angle trajectory is finally commanded to an underlying position controller, which uses the planned joint torques $\boldsymbol{\tau}$ as a feed-forward component.

It should be noted that the described solution of stabilizing an optimized joint level trajectory with an inverse kinematics based balancing algorithm has the conceptual disadvantage that it is limited to trajectories which a priori avoid the singularities of the leg kinematics. As a consequence, conservative constraints of the joint angles had been used in the optimization in choosing artificially restrictive knee joint angles limits, see Fig. 4. The extension to a joint level balancing algorithm that is able to handle singular configurations, however, would require a completely different control approach and is out of the scope of this paper.

The trajectories generated by the algorithm described in Sections II-IV consider the planning of consecutive single support phases with instantaneous support change (i.e. without double support phase). In practice a double support phase still exists due to modeling uncertainties and non-ideal position tracking. This double support phase is handled by the underlying balance controller (20). However, due to the lack of a planned double support phase, the desired ZMP $\boldsymbol{p}_d$ is discontinuous at the instant of the support change. While this is not a fundamental problem for the balancing algorithm, it still leads to a discontinuity in the commanded COM velocity, which may be undesired in practice. In the experiments reported in Section VI-C, we avoided this discontinuity by utilizing a fast interpolation rather than an ideal jump of the desired ZMP at the support change.

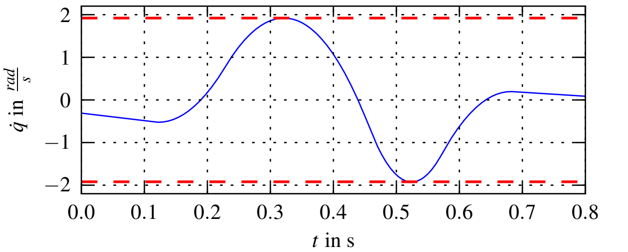

Fig. 4. Profile of the knee joint speed for the swingfoot and a 340mm stride length and a 0.8s step time. The joint speed limits are shown as red dashed lines.

VI. Results

The proposed optimization approach is evaluated for the DLR-Biped [19], a bipedal robot with $N = 12$ joints. The robot is equipped with rather compact feet 95mm wide and 190mm long. In each foot a force/torque sensor is measuring the ground reaction force for computation of the ZMP. The joints have integrated joint torque sensors and harmonic drive gears, which make the drive units hardly backdrivable if used without torque feedback. The system was designed for evaluating compliant impedance control laws for bipedal balancing and walking. As a consequence of the hardware design, the robot has relatively low maximum joint velocities of about 2 rad/s (except for one joint allowing for 3.14 rad/s), which poses a challenge for the trajectory optimization.

Figures 4 and 5 show some aspects of a typical result of an optimized trajectory, for which the inequality constraints are clearly satisfied. A typically resulting trajectory is illustrated in Fig. 1. To compare the results of the optimization with other approaches it is useful to analyze the COM and ZMP trajectories for the optimal trajectory defined in joint space. These are shown in Fig. 6 where it is evident that the optimizer satisfies the ZMP constraint while making full use of its limits (e.g. Point C).

A. Results for global search

For the optimization of the periodic gait we set $N_p = 144$, resulting in $12 - 3$ parameters per joint (3 for the boundary conditions (19)). For the global search 500 random trails were performed as described in Section IV-C. The number was chosen according to the dependency of

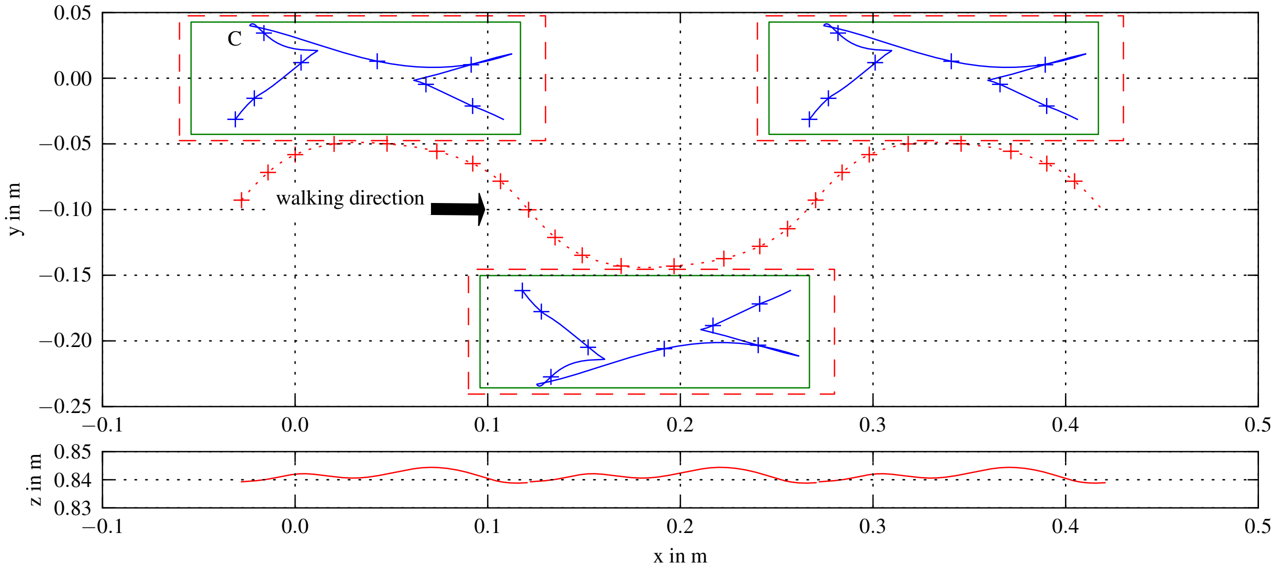

Fig. 6. Top: COM and ZMP trajectory, shown as red dotted and blue solid line respectively, for a sequence composed of three optimized steps trajectories. The ZMP trajectory is discontinuous between sucessive steps. This is caused by the approximation of the double support phase. The red dashed rectangle denotes the support polygon during the single support phase, the inner green rectangles denotes the support polygon as defined in the optimization. (Cross markers denote 100ms intervals) Bottom: Height variantion of the COM.

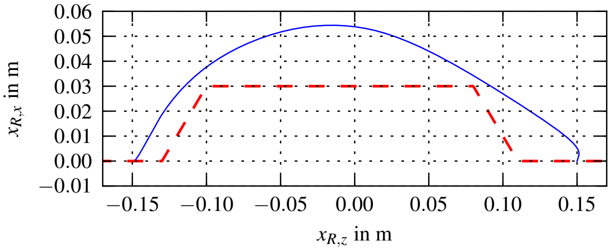

Fig. 5. Foot trajectory for an energy optimal step for 0.3m stride length and 0.8s step time, with $x_{R,z}$ the height of the foot above the ground and $x_{R,x}$ the coordinate along the walking direction. The border of the restricted area is shown by the red dashed line.

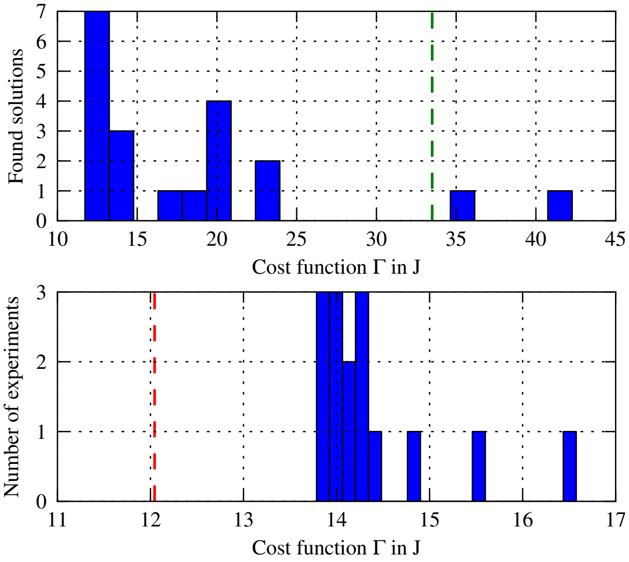

Fig. 7. Histograms of the cost function for 0.3m stride length and 0.8 s step time. Top: results of the global search. Comparison result generated with standard control method - shown in dashed green. Bottom: experimental results for commanded trajectory; the cost of the optimal planned trajectory (which was commanded) is shown in dashed red (approx. 12J). This illustrates the loss in the cost function.

the cost function in relation to the number of parameters. This relation is shown in Fig. 8.

Typical run times for a successful optimization is in the order of 1800s, related to an average of 800 iterations on an Intel Xeon W3530 2.80GHz processor.

Fig. 7 shows a typical result for the converged solutions (top diagram), giving an indication of the convergence rate. From it, it is evident that local minima exist. Notice the cost range between 12 J (approx. 15W) and 42 J (approx. 56W). In the same figure the value for a motion generated with a control method [4] is shown.

B. Simulation Results

To verify the trajectories found by the optimizer, the simulation tool OpenHRP [20] was used. It features a free-floating rigid body and contact simulation. The control structure used is the same as the one for the experimental system consisting of a combination of a joint position controller and the ZMP based stabilizer.

Without the stabilizer it is only possible to walk some steps before numerical disturbances and the non-ideal joint controller make the robot fall. Despite this, we tried to take full advantage of the optimizer capabilities and generated a trajectory where the knee joints have no artificial limits. As a result the optimizer could stretch the knee to the maximum, which would result in a kinematic singularity. The solution is shown in Fig. 1 (bottom row). This results in a lower cost function value.

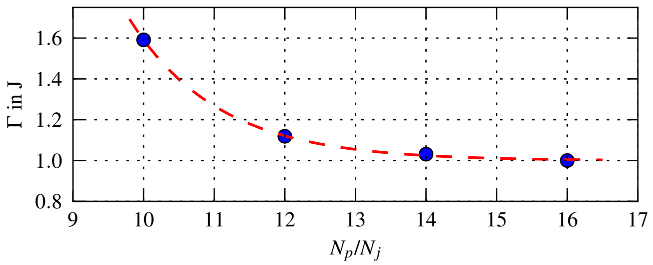

Fig. 8. Minimal cost function value found for given number of parameters per joint N_p/N_j, normalized to the lowest found cost value. Blue markers denote results from distinct global search executions. The red dashed line is an exponential fit against the raw data. Above 12 parameters only limited gain is achieved.

C. Experimental Results

The experimental results can be used to show how the on-line control modifies the trajectories and thus modifies the cost of the optimized trajectory. As stated above, the controller modifies the joint level trajectories to guarantee the stability of the robot in case of model uncertainties and disturbances.

To calculate the cost value of the measured trajectory, the joint angles where recorded. Multiple recordings of the trajectory for the most optimal solution shown in Fig. 7 (top) where examined with respect to their cost function value. The results are shown in Fig. 7(bottom), from which it, it is evident that the loss in optimality is small. The end result is an 55% gain in the cost function, with respect to the reference feed-back control solution.

VII. Conclusions and Future Work

A method for generating energy optimal walking gaits for a biped robot is presented as a nonlinear optimization problem. Results are given for the implementation of the resulting optimal trajectories on a real robot.

The method fully exploits the robot dynamics. The formulation of the optimization problem is based on the parameterization of the states and not of the inputs. This eliminates the need to integrate the equations of motion, resulting in an efficient method of solution. The method also provides smooth trajectories in torque, which is important for systems with passive compliance at the joints.

A detailed account of the optimization problem at hand is presented, with respect to local minima, convergence rate and cost minimization for the real system. A comparison is given to a solution obtained with optimal control.

It was found that, for the example given, the overall cost improvement was of factor of 3 with relation to the non-optimized solution. This implies that the control effort does not have a noticeable impact on the cost function. A future statistical analysis will consolidate this fact.

Future work will address generalizing global optimal solutions generated off-line. The dynamic model will be enhanced to include a double-support phase, joint compliance and impacts.

References

[1] T. Sugihara, Y. Nakamura, and H. Inoue, “Realtime humanoid motion generation through zmp manipulation based on inverted pendulum control,” in *IEEE Int. Conf. on Robotics and Automation*, 2002, pp. 1404–1409. 1

[2] H. Diedam, D. Dimitrov, P.-B. Wieber, K. Mombaur, and M. Diehl, “Online walking gait generation with adaptive foot positioning through linear model predictive control,” in *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2008, pp. 1121–1126. 1

[3] Y. Choi, D. Kim, Y. Oh, and B.-J. You, “Posture/walking control for humanoid robot based on kinematic resolution of com jacobian with embedded motion,” *IEEE Transactions on Robotics*, vol. 23, no. 6, pp. 1285–1293, 2007. 1, 5

[4] J. Englsberger, C. Ott, M. A. Roa, A. Albu-Schäffer, and G. Hirzinger, “Bipedal walking control based on capture point dynamics,” in *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2011. 1, 6

[5] G. Schultz and K. Mombaur, “Modeling and optimal control of human-like running,” *IEEE/ASME Transactions on Mechatronics*, vol. 15, pp. 783–792, 2010. 1, 2

[6] T. Buschmann, S. Lohmeier, H. Ulbrich, and F. Pfeiffer, “Optimization based gait pattern generation for a biped robot,” in *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, dec. 2005, pp. 98 –103. 1

[7] D. Djoudi, C. Chevallereau, and Y. Aoustin, “Optimal reference motions for walking of a biped robot,” in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005. 1

[8] J. Denk and G. Schmidt, “Synthesis of walking primitive databases for biped robots in 3d-environments,” in *International Conference on Robotics and Automation, ICRA2003, Proceedings of the IEEE*, 2003. 1, 3

[9] R. Lampariello, D. Nguyen-Tuong, C. Castellini, G. Hirzinger, and J. Peters, “Trajectory planning for optimal robot catching in real-time,” in *ICRA*, 2011. 1, 4

[10] R. Fletcher, *Practical methods of optimization.* Wiley, 1987. 1, 4

[11] J. Denk and G. Schmidt, “Synthesis of a walking primitive database for a humanoid robot using optimal control techniques,” in *International Conference on Humanoid Robots*, 2001. 1

[12] W. Suleiman, E. Yoshida, J.-P. Laumond, and A. Monin, “Optimizing humanoid motions using recursive dynamics and lie groups,” in *Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. 3rd International Conference on*, 2008. 2, 4

[13] M. Vukobratovic and D.Jurizcic, “Contribution to the synthesis of biped gait,” in *Proceedings IFAC Symp. Technical and Biological Problem on Control*, 1968. 3

[14] H.-D. Joos., “Mops - multi-objective parameter synthesis, user’s guide v1.21,” DLR, Tech. Rep. IB-515-02-01, 2002. 4

[15] D. Rogers and J. Adams, *Mathematical elements for computer graphics.* McGraw-Hill Higher Education, 1989. 4

[16] S. LaValle, “Rapidly-exploring random trees: A new tool for path planning,” *In*, no. 98-11, 1998. 4

[17] A. Albu-Schäffer, Ch. Ott, and G. Hirzinger, “A unified passivity-based control framework for position, torque and impedance control of flexible joint robots,” *The International Journal of Robotics Research*, vol. 26, no. 1, pp. 23–39, January 2007. 5

[18] L. Sciavicco and B. Siciliano, *Modeling and Control of Robot Manipulators.* McGraw-Hill Companies, 1996. 5

[19] C. Ott, C. Baumgärtner, J. Mayr, M. Fuchs, R. Burger, D. Lee, O. Eiberger, A. Albu-Schäffer, M. Grebenstein, and G. Hirzinger, “Development of a biped robot with torque controlled joints,” in *Humanoids*, 2010. 5

[20] F. Kanehiro, H. Hirukawa, and S. Kajita, “Openhrp: Open architecture humanoid robotics platform,” *The International Journal of Robotics Research*, vol. 23, pp. 155–165, 2004. 6