

Hardware-in-the-Loop Optimization of the Walking Speed of a Humanoid Robot

Thomas Hemker*, Hajime Sakamoto[†], Maximilian Stelzer*, Oskar von Stryk*

*Technische Universität Darmstadt, Department of Computer Science,
Hochschulstraße 10, 64289 Darmstadt, Germany

[†]Hajime Research Institute, Ltd., 5-6-21 Higashi Nakajima
Higashi Yodogawa-ku, Osaka 533-0033, Japan

email: {hemker, stelzer, stryk}@sim.tu-darmstadt.de, sakamoto@hajimerobot.co.jp

Abstract— The development of optimized motions of humanoid robots that guarantee a fast and also stable walking is an important task especially in the context of autonomous soccer playing robots in RoboCup. We present a walking motion optimization approach for the humanoid robot prototype HR18 which is equipped with a low dimensional parameterized walking trajectory generator, joint motor controller and an internal stabilization. The robot is included as hardware-in-the-loop to define a low dimensional black-box optimization problem.

In contrast to previously performed walking optimization approaches we apply a sequential surrogate optimization approach using stochastic approximation of the underlying objective function and sequential quadratic programming to search for a fast and stable walking motion. This is done under the conditions that only a small number of physical walking experiments should have to be carried out during the online optimization process. For the identified walking motion for the considered 55 cm tall humanoid robot we measured a forward walking speed of more than 30 cm/sec. With a modified version of the robot even more than 40 cm/sec could be achieved in permanent operation.

Keywords— Walking humanoid robot, RoboCup, Walking optimization, Black-box optimization, Surrogate optimization

I. INTRODUCTION

The speed of dynamically walking humanoid robots is a critical factor in many applications, especially in autonomous robot soccer games. In this paper we describe the successful approach undertaken for obtaining the fastest walking humanoid robot in the humanoid robot league of RoboCup 2006 (www.robocup2006.org).

Walking humanoid robots are high dimensional nonlinear dynamic multibody systems (MBS) with changing contact situations (impacts) and an underlying control of the joint motors. Many different approaches have already been investigated for improving the walking speed of bipedal and quadrupedal robots. However, all model-based optimization approaches have in common that their outcome critically depends on the quality and accuracy of the robot model. The derivation of highly accurate enough robot models to achieve the best possible walking speed may require too many efforts considering, e.g., the effects of gear backlash, elasticity and temperature dependent joint friction or of different ground properties. With a reasonable effort a MBS dynamics simulation of a humanoid robot can only achieve an error of about 5 to 10 % compared to the real system. Also a lot of additional work has to be spent in the transformation of results from the simulated model

to the real system before they can be used.

Furthermore, robot prototypes with the identical technical design differ significantly in their motion even if the same motion control software is used. This is due to inevitably small differences in the many robot components. For this reason, the final improvement can only be obtained by working on the real robot. An alternative approach to the optimization of walking motions based on detailed MBS dynamics simulations is to start with a reasonable, initial walking motion and then to use online hardware-in-the-loop optimization of the physical robot prototype.

In our approach, a surrogate optimization methods is applied, that is based on recent developments in the field. From our experience, this approach for online optimization of the walking speed is much more efficient in terms of function evaluations, i.e. walking experiments, required than random search methods like genetic or evolutionary algorithms. The latter are usually applied to cope with the robust minimization of noisy objective functions.

The paper is organized as follows. The following section gives a brief overview on the problem to find stable walking motion for walking robots in general and on the approaches to find fast and more robust motions. The third section describes the hardware and software components of the robot HR18 for which the walking optimization is done. The fourth section points out the optimization problem by objective function, variable domain, and gives a short overview on optimization methods that are applicable for the arising kind of problem. The here applied surrogate optimization method is introduced in the fifth section, the sixth section describes the experimental setup and the result obtained from the online optimization. The paper concludes by summarizing the main results.

II. HUMANOID LOCOMOTION

A. Humanoid robots

The legs of all current humanoid robots which are able to reliably perform a variety of different walking motions in experiments (as Honda Asimo, HRP-2, Johnnie or Sony Qrio) consist of rigid kinematic chains with 6 or 7 revolute joints using electrical motors of high performance and with rigid gears for rotary joint actuation. Only few initial approaches exist to insert and exploit elasticities in humanoid robot locomotion, e.g. [1]. Commonly servo motors are used in low- and

medium-cost humanoid robots. However, there exist specialized joint motors, gears and controllers designed for high-cost humanoid robots (e.g. Asimo, Qrio). A large variety of humanoid robots is involved in the Humanoid Robot League of the RoboCup competitions (www.robocup.org).

Walking of humanoid robots is much more difficult than with four-legged robots because during walking there must be phases where only one foot (or even no foot at all) is in contact with ground. Therefore stability control is much more an issue than with four-legged robots. Sensors are needed to detect instabilities in a very early phase. Sophisticated software must evaluate the sensors and modify the motion if needed to guarantee stability.

B. Walking optimization

Walking optimization generally may be done in two different ways: on computational models or on the real robot. Optimization on computational models using optimal control techniques, e.g. [2], [3], [4], for both four-legged and humanoid robots needs careful adaption of the model to the real robot and successive refinements of the model so that the computed trajectories may be implemented to the real robot. The pure optimization however can be done without human assistance. Stability criteria may be used in the optimization; therefore, methods that proved to be useful for four-legged robots may be used for humanoid robots as well.

When optimization is done on the real robot, methods used for four-legged robots may not directly be used for humanoid robot as instabilities may lead to severe damage of the robot. E.g. if the robot repeatedly falls down during automated walking experiments for optimization. Therefore, special approaches must be chosen. For four-legged robots, approaches based on parameterization of the walking motion and optimization of the parameters by evaluation of the walking speed have been used successfully [5], [6], [7]. For biped robots, stability is much more critical than for four-legged robots. [8] suggested to let the humanoid robot walk on a speed ramp during walking optimization as long as it is stable and to measure stability implicitly by the distance the robot has covered without falling down.

III. THE HUMANOID ROBOT HR18

A. Hardware selection and design

The new humanoid robot prototype HR18 (4.5 kg weight, 55 cm height, cf. Fig. 1) was developed by the Hajime Research Institute and consists of 24 actuated rotational joints; 14 of them are the most relevant ones for walking: 6 in each leg in the standard configuration for 6-dof humanoid robot legs (cf. Fig. 2) and two in the waist for forward/backward (pitch) and left/right turning motion of the waist. Motion of the 4-dof arms may be used for stabilization of locomotion. The head joint with the camera has 2-dof.

Metal frames for links and body are only used where really needed, i.e. mainly to connect the motors and



Fig. 1. The humanoid robot prototype HR18

to protect the controller boards from impacts in case of falling down. For all joints Robotis DX-117 servo motors are used which are operated at 14.8 V, have a maximum speed of 0.138 sec/60° and a maximum torque of 33.4 kg-cm. Besides controlling the position of the joints with adjustable control parameters, the servos are able to monitor their operation environment, e.g. temperature of the motor or voltage of the power supply.

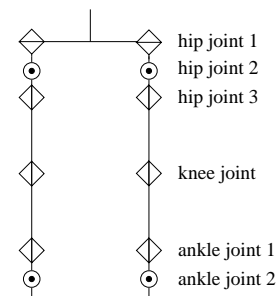


Fig. 2. Kinematic chain of the legs

The servos are connected via a RS485 bus to a controller board equipped with a Renesas SH7145 32bit microcontroller running at 50 MHz and 1 MByte of RAM. This controller board is used for real-time motion generation. Every 10 ms new desired positions are generated and sent to the servos. Furthermore, the controller is used to gather and evaluate data from the inertial sensors of the robot. The controller is connected to the main CPU of the system using a RS232 connection running at 57.6 kbits/s.

Lithium polymer rechargeable battery packages are used for onboard power supply because of their good ratio between mass and capacity, a 4-cell 14.8 volt battery for actuation of the motors, and a 2-cell battery for the controller board. According to the design rules of the RoboCup Humanoid Robot League the foot has a rectangular size of 125 mm times 90 mm, and it is a flat one without any ground contact sensors. Depending on the surface the robot shall walk upon, different materials like rubber or paper may be added to the ground contact area of the feet to avoid or give rise to a desired level of friction properties. The robot is equipped with three 1-axis gyroscopes and one 3-axes accelerometer,

all working with 100 Hz and attached in the hip. They are used to stabilize the walking motion by correction motions of the arms and the legs. For the needs of acting

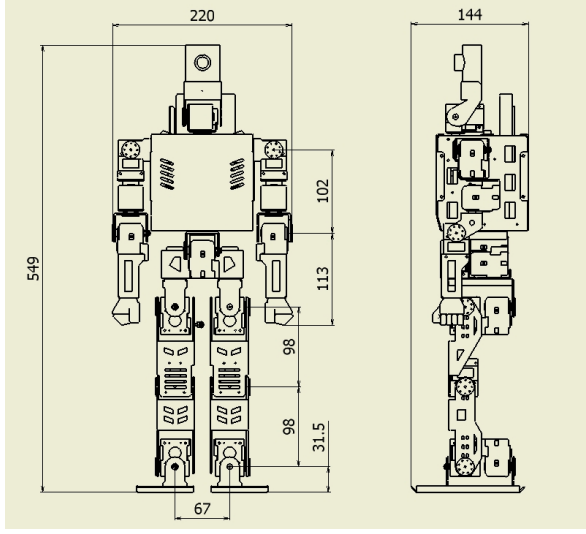


Fig. 3. Technical layout of HR18 (in mm)

in a highly dynamically environment like it is arising for autonomous soccer games in RoboCup, the HR18 has been equipped at TU Darmstadt with two off-the-shelf webcams and a pocket PC to enable autonomous soccer playing.

B. High- and low-level software

Two levels of software are used on the robot. On a pocket PC the high-level software is running. It consists of a framework [9] that coordinates the control architecture with sensor data processing, self localization, world modeling, hierarchical finite state machine for behavior control [10], and finally the motion generation/requests. The motion requests are sent via the described serial port to the controller board.

The low-level software on this board distinguishes between two different kinds of motions: Walking motions and so called special motions. The former ones are parameterized by certain parameters (cf. Section III-C), and the latter ones are teach-in-motions that are stored by joint angle trajectories.

The inertial sensor values are used to reduce vibration of unbalance during walking or other motions with a PD controller. The vibration is caused by a fast, but simplified calculation of the zero moment point (ZMP) or by uneven terrain. The PD control is useful to stabilize especially during the fast walking of the robot. It is calculated by

$$\theta_{new} = \theta + K_p * \omega_{gyro} + K_d * \frac{d}{dt} \omega_{gyro}$$

for the joint motors of the foot pitch, the foot roll, the hip pitch, the hip roll, the waist pitch, the shoulder pitch, and finally the shoulder roll, herein described as θ for the angle calculated by inverse kinematics and

θ_{new} the controlled angle; K_p and K_d are the PD control parameter and ω_{gyro} is the angular velocity of the gyro. The values for K_p and K_d are identified by expert knowledge and experiments.

Direct access to the memory of the robot exists, i.e. all parameters may be changed during run time which allows easy alteration of walking or sensory parameters and debugging.

C. Parameterization of humanoid walking trajectories

Like any motion, walking motion may be described by joint angle trajectories. These however are infinite dimensional and therefore hard to handle. In contrast to special motions like shooting the ball, waving, getting down or standing up, walking is a periodically and comparatively smooth motion. Therefore, the walking

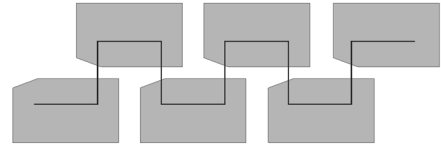


Fig. 4. Footprint of the walking motion

motion may be generated by prescribing trajectories for the hip and the feet and solving the inverse kinematics for the joint angles. The inverse kinematics model can be calculated analytically due to the special kinematic structure of the leg (Fig. 2). The trajectories of hip and feet are parameterized by several geometric parameters, which leads to a high dimensional (finite, however) discretization of the walking motion.

The walking trajectories are calculated in the three dimension by the following formulas. To control the y -position the ZMP is used:

$$y(t) = c_1 * \exp(\sqrt{(G/H_g)} * t) + c_2 * \exp(-\sqrt{(G/H_g)} * t) + X_{ZMP} \quad (1)$$

where G describes the gravity, H_g the height of center of gravity, c_1 and c_2 are constants, and x_{ZMP} is precalculated so that the ZMP moves on a rectangular trajectory between the feet (cf. solid line in Fig. 4).

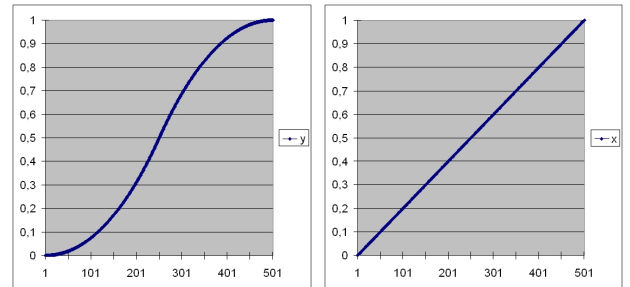


Fig. 5. y -position and x -position trajectory

The x -position is just a simple straight line that is used to control the center of gravity movement of X

axis. When the robot is walking at constant speed, the ZMP of the x -axis is 0 and is neglected.

The curve describing the z -position is moving fast near the ground. It is because to reduce the influence of the ground error and to reduce inertia during swing of the leg. For lifting the foot we use

$$z_{up}(t) = 1 - \frac{\arcsin(1 - t/500)}{\frac{1}{2}\pi}, \quad (2)$$

and

$$z_{dw}(t) = \frac{\arcsin(t/500)}{\frac{1}{2}\pi} \quad (3)$$

for dropping it again. By aggregating these curves for a walking motion we obtain completely parameterized walking motion trajectories.

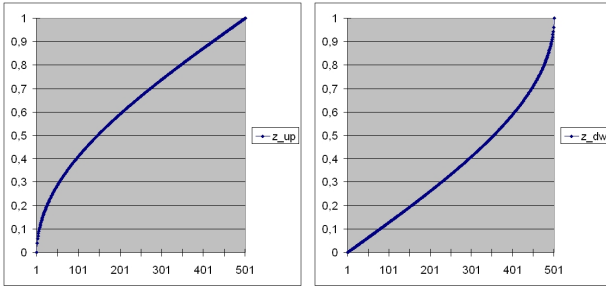


Fig. 6. General z -position trajectory

Finally parameter tuning by hand identified the most important parameters for stable walking. Those are the relation of the distances of the front and of the rear leg to the center of mass, the lateral position, the roll angle and the height above ground of the foot during swing phase, and the pitch of the upper body. These parameters scale and shift the defined walking trajectories in Equations (1) to (3) during the optimization process, where the other parameters are assumed to be fixed with suitable values which are determined by expert knowledge and experiments.

Using finally the symmetry of the walking motions the parameter for the lateral position of the right foot can be set to the negative lateral position of the left foot during the swinging phase, just as well as the roll angle of the right foot has to be the negative roll angle of the left foot. With this procedure we obtain a six dimensional parameterization of the walking trajectories at a fixed step frequency.

IV. THE OPTIMIZATION PROBLEM

A. Modeling of the objective function

As quality or objective function value for a parameter set defining a walking motion we measure the distance the robot covers on the experimental field until it stops or falls. It starts with a small step length 110 mm and the step length is increased every 2 steps by 5 mm, so that the final step length of 240 mm would be reached after 52 steps, always with a constant frequency of ~ 2.85 steps per second. It follows that the

walking speed is only influenced directly by the applied step length.

This definition of the objective function includes constraints for maintaining walking stability of the walking motion implicitly, which otherwise would have to be formulated analytically with a high effort to incorporate them explicitly into the definition of the here considered optimization problem. Only robust walking motions for all step lengths between 110 mm and 240 mm have a chance to reach a high objective function value, because during the measurement 27 different step lengths are applied. This ensure that a found walking motion is a fast but also robust one for different velocities.

B. Feasible domain of optimization variables

The above introduced number of relevant parameters for the walking motion is reduced with this objective function definition by another dimension to finally 5 optimization variables. The variation of the step length during the walking experiment, the x -position of one foot in front of the upper body and the x -position of the other foot behind the upper body during a step are reduced to only one variable. This variable describes the proportion between both, and by the total applied step length, that is increased during the walking experiment, both parameters are described uniquely. During the on-line optimization the five real-valued variables that are influencing the main characteristics of the walking behavior of the robot are varied to maximize the defined objective function only regarding to box constraints. These bounds are set to values for which a wide range for variations of the walking motion is possible. But the bounds also guarantee that no contacts of moving parts of the robot can occur.

C. Classification of the optimization problem

The robot is included as hardware in the loop for the walking experiment to evaluate the objective function f . In this context a non deterministic black-box optimization problem with box constraints arises, where besides a noise function value of f no further information is provided. Especially no gradient information can be provided which would be compulsive for effective gradient based optimization. Even the approximation of gradients by finite differences is not applicable because of the non deterministic experimental results. Additionally we are also not able to proof properties like continuity or differentiability of the objective function because of the black box characteristic of the problem. The resulting mathematical formulation of the optimization problems reads as follows:

$$\max_{p \in \Omega} f(p), \quad f : \mathbb{R}^5 \rightarrow \mathbb{R},$$

with

$$\Omega = \{p \in \mathbb{R}^5 | p^{(l)} \leq p \leq p^{(u)}, p^{(l)}, p^{(u)} \in \mathbb{R}^5\}.$$

This leads to the question which optimization methods can be applied to find a good walking motion with reasonable efforts.

D. Optimization methods

The arising optimization problem impedes the direct use of any kind of gradient based or Newton type methods because only the objective function value disturbed by induced noise is available. But a wide range of different optimization methods is designed for these kind of optimization problems (cf. Fig. 7). These methods can

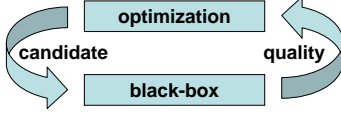


Fig. 7. Direct coupling for black-box optimization

be divided into three different groups of approaches. The first group summarizes the random search methods or stochastic optimization methods with the mostly known Genetic and Evolutionary Algorithms (GA/EA). Simulated Annealing, Tabu Search, Particle Swarm and Ant Colony Methods are also well developed representatives of this group. These methods use stochastic elements or heuristics to generate new promising candidates out of evaluated candidates. The textbook of Siarry [11] gives a wide introduction for the above mentioned methods.

The second group consists of sampling methods. These are on the one hand side direct optimization methods [12] like the Pattern Search [13], [14], or DIRECT [15] which only include the information if improvement is obtained or not. On the other side we have sampling methods that include the objective function values also as quantitative information. These are e.g. the Nelder-Mead simplex approach [16], extension of this idea, or implicit filtering [17], [18] as a projected Quasi-Newton Method. Another way to subdivide the group of sampling method is the characteristic, if the search is grid-based (e.g. pattern search) or not (e.g. Nelder-Mead), or if it is stencil based (e.g. implicit filtering) or not (e.g. DIRECT). But all sampling methods have in common that new candidates are generated during their search for a maximum/minimum by exploring promising areas, or using bigger steps to find promising areas of the search domain.

Both groups of methods are often applied to simulation based optimization problems, which do not differ too much from the here considered walking optimization problem. With the main weight on the number of function evaluation a method needs, it has been shown that sampling methods can outperform random search methods on well modeled problems [19], but are not as easy to apply as the often very user friendly implemented GAs or EAs. In our case each function evaluation needs a physical experiment, which means use and wear of the robot, manpower for observing, and expensive lab time.

The third group of methods are the surrogate optimization methods. Here, the optimization is not performed directly on the objective function but on a ap-

proximation of it as described, e.g., in [20], [15] or [21]. In [22] and [23] such a method based on sequential stochastic approximations in combination with sequential quadratic programming was successful applied to simulation based black-box problems with expensive function evaluations. Because of the efficiency shown for these applications regarding to the needed objective function evaluations, the therein proposed method is adapted to the humanoid robot walking optimization problem and is described in the next section.

V. SEQUENTIAL SURROGATE OPTIMIZATION

A. Design and analysis of computer experiments

The optimization method we apply consists in each iteration of a statistical approximation method that calculates a surrogate function of the original objective function. Many different methods exist and are used in the context of surrogate optimization. As a standard approach designed for deterministic black-box optimization problems, design and analysis of computer experiments (DACE) described by Sacks et al. [20] is widely used, which has its origins in the geostatistical kriging method [24]. But even for non-deterministic functions this method is able to catch the main characteristics if the deviation is in comparison to the mean not too high to mask the main effects of the considered objective function f .

A DACE model \hat{f} is in its general form a two component model,

$$\hat{f}(p) = \sum_{j=1}^k \beta_j h_j(p) + Z(p).$$

The first and more global part approximates the global trend of the unknown function f by a linear combination of a scalar vector $\beta \in \mathbb{R}^k$ and k known fixed functions h_j . Different approaches can be considered for this first part [25]. Universal kriging is predicting the mean of the underlying function as a realization of a stochastic process by a linear combination of k known basis functions on the whole domain, as well as detrended kriging applies a classical linear regression model. In our application we follow the idea of ordinary kriging where a constant mean is assumed on the whole domain with $k = 1$ and one constant h .

The second part Z guarantees that the DACE model \hat{f} fits to f for a number of sets of points $p^{(i)}$, $i = 1, \dots, N$ out of the variable space,

$$f(p^{(i)}) = \hat{f}(p^{(i)}), \quad i = 1, \dots, N.$$

This lack of fit part is assumed to be a realization of a stationary Gaussian random function with a mean of zero, $E[Z(p)] = 0$, and a Covariance

$$\text{Cov}[Z(p^{(i)}), Z(p^{(j)})] = \sigma^2 R(p^{(i)}, p^{(j)}),$$

with $i \neq j$, and $i, j \in \{1, \dots, N\}$. The smoothness and properties like differentiability are controlled by the chosen correlation function R , in our case the product of

one dimensional correlation functions,

$$R(p^{(1)}, p^{(2)}) = \prod_{j=1}^{n_p} R_j(|p_j^{(1)} - p_j^{(2)}|),$$

and the Gaussian correlation function,

$$R_j(|p_j^{(1)} - p_j^{(2)}|) = \exp(-\theta |p_j^{(1)} - p_j^{(2)}|^q),$$

with $q = 2$, as a special case of the generalized exponential correlation function with $0 < q < 2$ and $\theta \in (0, \infty)$ proposed by Sacks et al. [20]. By maximum likelihood estimation the process variance σ^2 , the regression parameter β , and the correlation parameter θ used to define R are estimated most consistent to the present objective function values to which \hat{f} has to fit. As additional information we can evaluate the expected mean square error (MSE) of the actual model. Under the assumption that the approximated function really behaves like a Gaussian random process, the MSE of the DACE model gives an estimation of the approximation quality for a certain p . This is needed later during the sequential update procedure. For a more detailed description on the theory behind these kind of models we refer the reader to [26].

B. Sequential update process

We start the optimization process with an experiment, where the point of the first surrogate function are chosen by prior obtained expert knowledge to provide a stable but possibly slow, initial humanoid robot walking motion. An initial set of experiments is generated around the initial motion by varying each single variable dimension p_i on its own by 10 % of the total range of the feasible domain, like it is known from other optimization methods by sampling [13]. In our case this leads to eleven sampled points for which the objective function value is known. With this knowledge the first surrogate function \hat{f}_1 is built. With this first surrogate function the initial phase of this optimization procedure is finished and the main iteration starts. This analytically given surrogate is maximized regarding to the defined box constraints on p .

The maximizer of the surrogate function \hat{f} is added to the set of basis points for the DACE model in the next iteration step. Such sequential procedures are discussed in greater detail in [27]. However, in contrast to the update strategies discussed therein, we use convergence to a previously determined parameter set as criteria to stop maximizing \hat{f} and to search for a new candidate parameter set. If the distance of a found maximizer to a point already evaluated by experiments falls below a defined limit, not the actual maximizer, but the maximizer of the expected mean square error (MSE) of the surrogate function is searched, evaluated, and added to the set of basis points for approximation. The MSE of a DACE model can be calculated easily if the model itself is determined as described in [20]. This procedure improves the approximation quality of the surrogate function in unexplored regions of the parameter

domain. The objective function value for this new candidate is determined by performing the corresponding walking experiment with the humanoid robot.

After a new point is added, a new surrogate function is generated with a new DACE model and the optimization starts again. The sequential procedure is summarized in a general setting in the following algorithm:

Algorithm 1 Surrogate function update for objective function f and variable p :

- 1: Evaluate $f(p^{(i)})$ for a set of basis points $p^{(i)}$, with $i = 1, \dots, N$.
 - 2: Build surrogate function \hat{f} by running DACE with the set of basis points $p^{(i)}$, for $i = 1, \dots, N$.
 - 3: Search the minimizer \hat{p}^* of \hat{f} .
 - 4: If \hat{p}^* is too close to a basis point, $|\hat{p}^* - p^{(i)}| \leq \epsilon$, $i = 1, \dots, N$, go to 5, else to go 6.
 - 5: Search for the maximizer \hat{p}^{MSE*} of the $MSE(\hat{f})$.
 - 6: Add \hat{p}^* respectively \hat{p}^{MSE*} to the set of basis points, stop or go to step 1.
-

The key feature of surrogate optimization approaches is that the optimization methods avoids from running directly in a loop with the real system that generates the original optimization problem, but calls the surrogate problem on which emerging computational costs to determine new promising points to test with the real system can be neglected if the costs of running a real walking experiment are taken into account.

C. Implementation details

The complete optimization procedure is implemented in Matlab and uses two additional standard toolboxes. The first one is the DACE toolbox [28], that generates in each iteration the DACE model and is called directly by the optimizer for the function values $\hat{f}(p)$ to find the maximizer \hat{p}^* of \hat{f} , respectively the maximizer \hat{p}^{MSE*} of the MSE of \hat{f} . The second component is the optimizer SNOPT [29], a sophisticated implementation of sequential quadratic programming (SQP).

The idee behind SQP methods as described in [30] is to solve a nonlinear programming problem by using a sequence of quadratic subproblems. It is a highly efficient optimization method for smooth analytically defined nonlinear programming including also nonlinear constraints problems, even for large scale problems. In our case only nonlinear programming problems of small scale are arising with simple constant bounds on the variables, which have to be solved during each iteration of the described sequential surrogate optimization approach.

VI. EXPERIMENTAL AND NUMERICAL RESULTS

A. Experimental setup and procedure

All walking experiments with the humanoid robot HR 18 are performed on a standard RoboCup soccer field (cf. Fig. 8). To guarantee constant conditions during the

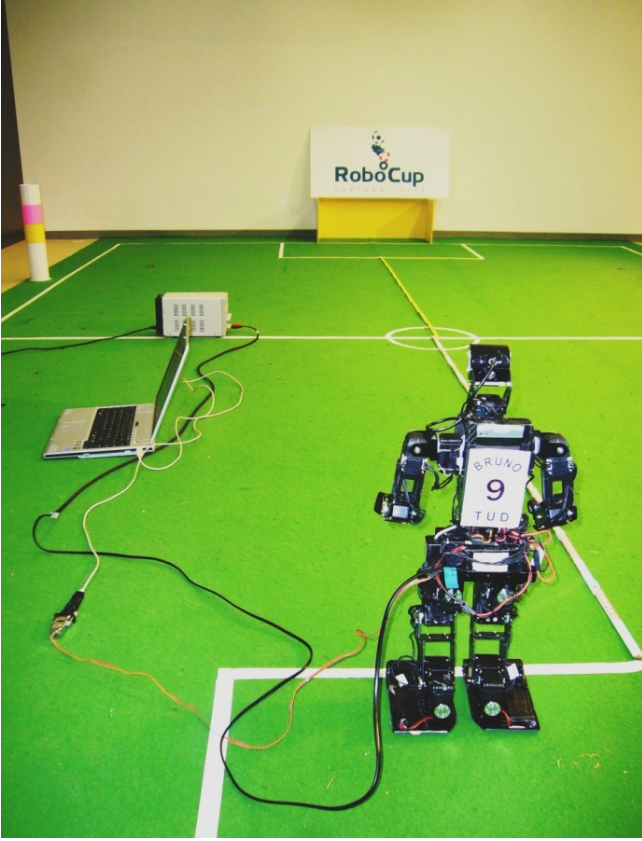


Fig. 8. Setup at the beginning of a walking experiment

experiments the joint motors are supplied by a continuous external power supply where the batteries are only used as weights to provide the operating condition the robot meets in the intended field application. For the walking experiments the Pocket PC is only carried for the same reason as the batteries are. Only the low-level software is running on the robot during the experiments with its joint motor control and the internal stabilization. The controller board is serially connected to an external computer, on which the above described high-level software and the optimization procedure is running. The walking requests generating module therein is extended for the special needs of the modeled objective function and the walking experiments that have to be performed, so that the step length can be increased during runtime. Additionally the joint motor temperatures are observed to guarantee no overheating at any time. The motion generation obtains new walking parameter sets from the optimization procedure as a text file and sends walking requests for walking experiments together with the walking parameters to the robot.

Of course not every new parameter set that is generated by the optimization procedure results in a walking motions where the robot is really walking forward or is even able to walk one step without falling. For such parameter sets p we assign zero as the objective function value. All walking experiments start at the same position heading into the same direction, and the distance that is covered by the robot is measured as the differ-

ence between an orthogonal projection from the final position on the line of an ideal straight walk and the starting point. This way not straight walks are penalized implicitly, and the distances are returned manually back to the interface of the optimization procedure, so that the next parameter set can be computed. Additionally the temperature of the joint motors as a possible critical factor is monitored during all experiments, so that at no time the problem of overheating in a joint motor could accrue. The power cord, the serial connection, and a belt to protect the robot from damage in the case that it falls are carried by a helper, but it is keenly observed that the robot's motions are not influenced by the cables and the belt.

B. Numerical and experimental results

The initial parameter set $p^{(ini)}$ has been obtained earlier by tuning of the parameters by hand and the definition of p as described above is summarized in Table I. As we can see from the results (Fig. 9) during the eleven

TABLE I
DEFINITION PARAMETER p

p_1	backward / forward	each in mm
p_2	step height	in mm
p_3	lateral position	in mm
p_4	roll angle	in degree
p_5	body pitch	in degree

walking experiments of the initial phase of the optimization process, that the results have a nonlinear connection to the parameters because variations according to any component of of $p^{(ini)}$ do not yield to any improvement. After the initial experiments a value of p is found which results in a larger objective function value. But the following iterations of the proposed surrogate optimization approach can be taken as kind of an exploring to catch the characteristics of the black-box function f better. Also the maximizers of the surrogate func-

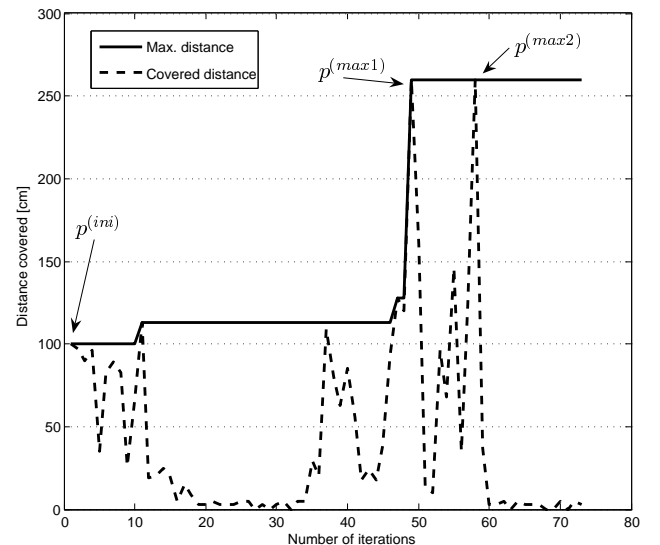


Fig. 9. Walking experiment result during optimization

tions during the first number of iterations do not really have a big chance to be maximizers of the underlying black-box as can be observed from iteration 18 to 32 in Fig. 9. But by finding promising areas, improvements in form of two candidates $p^{(max1)}$ and $p^{(max2)}$ are identified, where the robot covers the same distance after performing all 52 steps of a walking experiment with stopping at end without falling. Even after exploring

TABLE II
BOUNDS, INITIAL EXPERT'S GUESS, AND TWO
CANDIDATES FOR AN OPTIMAL p

	p_1	p_2	p_3	p_4	p_5
$p^{(l)}$	-2.000	0	50.000	-8.000	0
$p^{(ini)}$	-1.000	25.000	75.000	-4.000	15.000
$p^{(max1)}$	-0.958	17.344	81.641	-0.018	20.000
$p^{(max2)}$	-2.000	27.761	50.000	-0.155	17.331
$p^{(u)}$	-0.500	50.000	100.000	0	20.000

promising areas of the variable domain no continuous improvement is guaranteed as observed during the last 10 walking experiments. In a five dimensional space of optimization variables, an appropriate approximation quality on the whole domain can not be obtained only by the number of here performed walking experiments.

As a more technical result affecting the hardware we observed that at no time the external power supply generated more than 3.2 ampere at a voltage of 14.8. During the whole number of walking experiments the joint motor temperature was never getting a major problem with average temperatures of 55 to 60 degree Celsius. The knee joint motor as the most stressed one has been observed during all iteration carefully. The here described optimization run was paused only for one time when the motor temperature exceeds 65 – 70 degree Celsius after about 40 runs. This was done to guarantee that all walking experiments are performed under roughly comparable conditions. The shut down temperature of the joint motors is 85 degree Celsius.

C. Adaption for a fast walking motion

The solutions found during the optimization are successively adjusted by setting up a constant step length and step time dependent on different floor coverings. The result of our approach is a stable humanoid robot walking motion with a speed of 30 cm/sec. This motion provides a robustness that allows to start with the found maximum speed and to perform an abrupt stop without unbalancing the robot, even on different floor coverings.

The robot was right from the start designed to be very lightweight. But during the preparations for RoboCup 2006 additional work was spent on the lightweight aspect, one degree of freedom was reduced in each arm so that 2 DX-117 motors could be removed as well as the left/right turning waist joint. By the same reason the steal frame of the upper body was replaced by a slimmer carbon frame of 0.2 kg less weight (cf. Fig. 11). To lower the center of mass the battery for the joint motors was replaced by two 2-cell batteries of the same

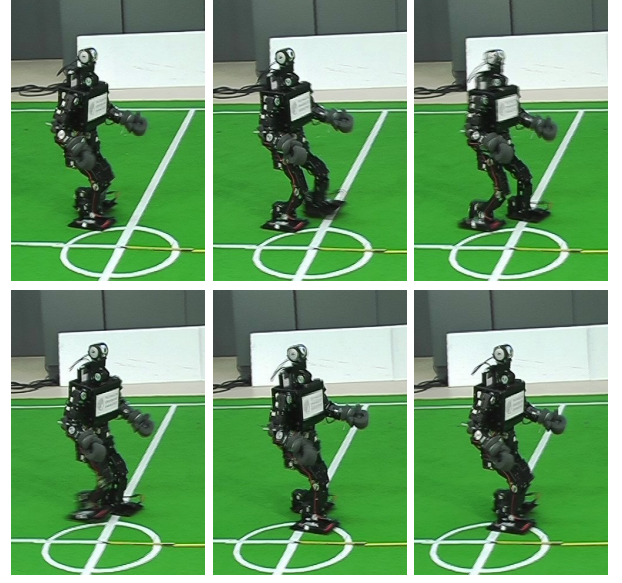


Fig. 10. Optimized walking motion of the humanoid robot prototype HR18

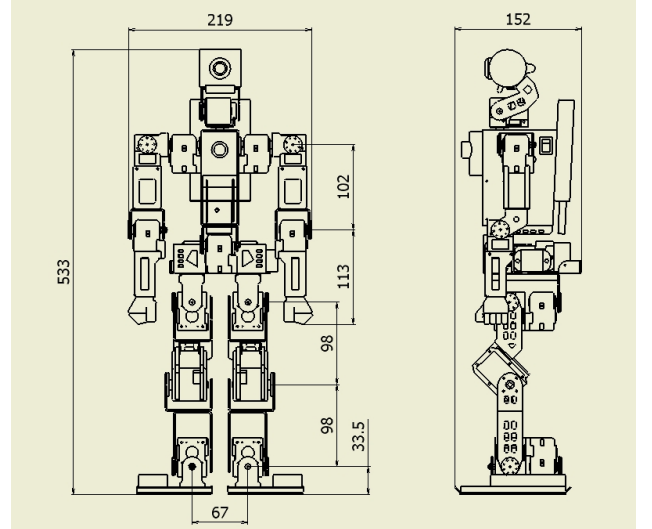


Fig. 11. Technical layout with measurements in mm including the design modifications

kind like it is used for the controller board. These two batteries are placed on the sides of the both feet (cf. Fig. 12).

Another change was done due to the relatively high temperatures in the knee joints, so that during the normal operation time of robot soccer games no heating problems arise. The new motor of type Robotis RX-64 has a higher maximum torque of 53 kg-cm as well as a higher maximum speed of 0.198 sec/60° than the first used DX-117. The modified version of the humanoid robot has a final weight of 3.3 kg and was the basis of the humanoid robot Bruno used successfully by the Darmstadt Dribblers & Hajime Team at RoboCup 2006. For a more detailed report on the modified robot we refer the reader to [31].

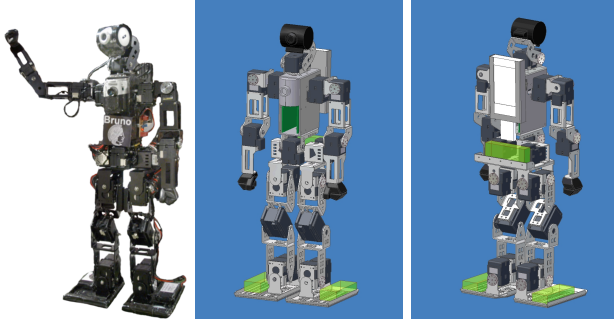


Fig. 12. The humanoid robot Bruno (prototype HR18 after modifications)

With the modified humanoid robot a forward walking speed of more than 40 cm/sec was obtained at a step frequency of 4 steps per second and a 20 mm longer step length. But in difference to the originally found motion on unchanged hardware, this walking motion needs some steps for acceleration with increasing step-length to get into a stable walking motion of 40 cm/sec. But it also performs a stable instant stop in one step. In Table III a comparison of the walking speed of Bruno linearly scaled to human size is given.

TABLE III
WALKING SPEED SCALED TO HUMAN SIZE

	HR18 prototype	approx. human size
size	55 cm	180 cm
walking speed	40 cm/sec 1.44 km/h	~ 130.80 cm/sec ~ 4.70 km/h

During RoboCup 2006 several demo footraces have been performed with other humanoid robots and four-legged Sony Aibo robots where Bruno clearly demonstrated to be the fastest humanoid robot. In a demo race, it also outperformed the much taller finalists of the footrace competition of the TeenSize humanoid robots taller than 65 cm.

VII. CONCLUSION

For the newly developed, 55cm tall humanoid robot prototype HR18 a low dimensional hardware-in-the-loop walking optimization approach has been presented. The optimization problem formulation is based on a suitable parameterization of the walking trajectories, where a small number of parameters are used as optimization variables. During fast walking the underlying joint servo motor control is stabilized using inertial sensing which is part of the low-level software running on the robot's controller board. Under this precondition it was demonstrated that efficient walking speed optimization can be obtained even with a quite small number of functions evaluations. In only about 50 walking experiments a 30 cm/sec fast walking motion has been obtained using a sequential surrogate optimization approach. In further experiments, it was observed that the obtained walking motion is also robust in a sense that the performance does not change much on

different floor coverings. A further improved hardware design of the robot prototype HR18 resulted in the prototype Bruno. With a short starting phase for acceleration from stand still a further increase of the forward walking speed to more than 40 cm/sec was achieved. The described optimization approach can be applied to other types of humanoid robot locomotion like turning or walking sideways as well.

ACKNOWLEDGMENTS

Parts of this research were supported by the German Research Foundation (DFG) within the Research-Training-Group 853 "Modeling, Simulation, and Optimization of Engineering Applications" and the Priority Program 1125 "Kooperierende Teams mobiler Roboter in dynamischer Umgebung".

REFERENCES

- [1] A. Seyfarth, R. Tausch, M. Stelzer, F. Iida, A. Karguth, and O. von Stryk, "Towards bipedal jogging as a natural result of optimizing walking speed for passively compliant three-segmented legs," in *CLAWAR 2006: International Conference on Climbing and Walking Robots, Brussels, Belgium, Sept. 12-14, 2006*.
- [2] J. Denk and G. Schmidt, "Synthesis of a walking primitive database for a humanoid robot using optimal control techniques," in *Proceedings of IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS2001), Tokyo, Japan, November 2001, 2001*, pp. 319-326.
- [3] M. Buss, M. Hardt, J. Kiener, J. Sobotka, M. Stelzer, O. von Stryk, and D. Wollherr, "Towards an autonomous, humanoid, and dynamically walking robot: Modeling, optimal trajectory planning, hardware architecture, and experiments," in *Proc. IEEE/RAS Humanoids 2003. 2003*, Springer-Verlag.
- [4] M. Stelzer, M. Hardt, and O. von Stryk, "Efficient dynamic modeling, numerical optimal control and experimental results for various gaits of a quadruped robot," in *CLAWAR 2003: International Conference on Climbing and Walking Robots, Catania, Italy, Sept. 17-19, 2003*, pp. 601-608.
- [5] N. Kohl and P. Stone, "Policy gradient reinforcement learning for fast quadrupedal locomotion," in *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2004, vol. 3, pp. 2619-2624.
- [6] Th. Roefer, "Evolutionary gait-optimization using a fitness function based on proprioception," in *8th International Workshop on RoboCup 2004 (Robot World Cup Soccer Games and Conferences), Lecture Notes in Artificial Intelligence. Springer, to appear, 2005*.
- [7] M. Hebbel, W. Nistico, and D. Fisseler, "Learning in a high dimensional space: Fast omnidirectional quadrupedal locomotion," in *RoboCup 2006: Robot Soccer World Cup X, Lecture Notes in Artificial Intelligence, Springer, 2006, to appear, 2006*.
- [8] S. Behnke, "personal communication," 2006.
- [9] M. Friedmann, J. Kiener, S. Petters, D. Thomas, and O. von Stryk, "Reusable architecture and tools for teams of lightweight heterogeneous robots," in *Proc. 1st IFAC-Symposium on Multivehicle Systems*, Salvador, Brazil, October 2-3 2006, p. to appear.
- [10] M. Lttsch, M. Risler, and M. Jngel, "Xabsl - a pragmatic approach to behavior engineering," in *Proceedings of IEEE/RSJ International Conference of Intelligent Robots and Systems (IROS)*, Beijing, China, October 9-15 2006, p. to appear.
- [11] J. Dréo, P. Siarry, A. Ptrowski, and E. Taillard, *Metaheuristics for Hard Optimization*, Springer, 2006.
- [12] R.M. Lewis, V. Torczon, and M.W. Trosset, "Direct search methods: then and now," *Journal of Computational and Applied Mathematics*, vol. 124, pp. 191-207, 2000.
- [13] J.D. Griffin and T.G. Kolda, "Asynchronous parallel generating set search for linearly-constrained optimization," Tech. Rep., Sandia National Laboratories, Albuquerque, NM and Livermore, CA, July 2006.

- [14] V.J. Torczon, "On the convergence of pattern search algorithms," *SIAM J. Optim.*, vol. 7, no. 1, 1997.
- [15] D.R. Jones, M. Schonlau, and W.J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, pp. 455-492, 1998.
- [16] J.A. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal*, vol. 7, pp. 308-313, 1965.
- [17] Choi, Eslinger, Gilmore, Patrick, Kelley, and Gablonsky, "IFFCO: Implicit Filtering for Constrained Optimization, Version 2," Tech. Rep. CRSC-TR99-23, Center for Research in Scientific Computation, July 1999.
- [18] C.T. Kelley, *Users Guide for imfil version 0.5*, 2005.
- [19] Fowler, Kelley, Miller, Kees, Darwin, Reese, Farthing, and Reed, "Solution of a well-field design problem with implicit filtering," *Optimization and Engineering*, vol. 5, pp. 207-233, 2004.
- [20] J. Sacks, S. B. Schiller, and W.J. Welch, "Design for computer experiments," *Technometrics*, vol. 31, pp. 41-47, 1989.
- [21] N. M. Alexandrov, "An overview of first-order model management for engineering optimization," *Optimization and Engineering*, vol. 2, pp. 413-430, 2001.
- [22] Th. Hemker, K. Fowler, and O. von Stryk, "Derivative-free optimization methods for handling fixed costs in optimal groundwater remediation design," in *Proc. of the CMWR XVI - Computational Methods in Water Resources*, 19-22 June 2006.
- [23] Th. Hemker, M. Glocker, H. De Gersem, O. von Stryk, and Th. Weiland, "Mixed-integer simulation-based optimization for a superconductive magnet design," in *Proceedings of the Sixth International Conference on Computation in Electromagnetics*, Aachen, Germany, 4-6 April 2006.
- [24] G. Matheron, "Principles of geostatistics," *Econom. Geol.*, vol. 58, pp. 1246 - 1266, 1963.
- [25] J.D Martin and T.W. Simpson, "A study on the use of kriging models to approximate deterministic computer models," in *Proceedings of DETC'03*, 2003.
- [26] J.R. Koehler and A.B. Owen, "Computer experiments," *Handbook of Statistics*, vol. 13, pp. 261-308, 1996.
- [27] Sasena, *Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations*, Ph.D. thesis, University of Michigan, 2002.
- [28] Lophaven, Nielsen, and Sondergaard, "DACE, a matlab kriging toolbox," Tech. Rep. IMM-TR-2002-12, IMM, August 2002.
- [29] P.E. Gill, W. Murray, and M.A. Saunders, "User's guide for SNOPT 7.1: a fortran package for large-scale nonlinear programming," Report NA 05-2, Department of Mathematics, University of California, San Diego., 2006.
- [30] P.E. Gill, W. Murray, and M.A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM REVIEW*, vol. 47, no. 1, pp. 99-131, 2005.
- [31] M. Friedmann, J. Kiener, R. Kratz, S. Petters, M. Stelzer, O. von Stryk, and D. Thomas, "Darmstadt dribblers 2006: Humanoid robot," 2006.