Gait Optimization on a Humanoid Robot using Particle Swarm Optimization

Cord Niehaus #1, Thomas Röfer +2, Tim Laue #3

#Fachbereich 3 - Mathematik und Informatik, Universität Bremen,
Postfach 330 440, 28334 Bremen, Germany

1cniehaus@informatik.uni-bremen.de, 3timlaue@informatik.uni-bremen.de

+DFKI Lab Bremen, Enrique-Schmidt-Str. 5,
28359 Bremen, Germany

2Thomas.Roefer@dfki.de

***Abstract*— This paper describes the application of Particle Swarm Optimization (PSO) for gait optimization on a humanoid robot. The biped gait is modeled by a number of parameterizable trajectories. To achieve omni-directional walking, different sets of gait parameters are optimized for specific walk directions and interpolated later. By using a fitness test based on an acceleration walk, the optimized sets of parameters are suitable for a wide range of walk speeds. We tested the applicability of the approach by performing gait optimization for several walk directions on a modified Kondo KHR-1 robot.**

***Keywords*—gait modeling, gait optimization, particle swarm optimization (PSO), acceleration walk, omni-directional gait**

I. Introduction

In RoboCup competitions, one major task is the robot locomotion on a flat field and its optimization in terms of quickness, flexibility, and stability. Especially in the leagues with legged robots, such as the four-legged league and the humanoid league, good omni-directional walking skills are essential for winning games. The optimization of the robot's gait, i.e. the parameters of the gait, is a good area to apply optimization methods that are able to deliver quick results. On the one hand, it is prudent to avoid time-consuming optimization runs that wear out the robot hardware, and on the other hand it is also beneficial to be able to adapt the robot gait to different surface conditions at the competition site in a reasonable amount of time. Therefore, we examine Particle Swarm Optimization (PSO) as suitable alternative to other established optimization methods that have been used in RoboCup leagues.

The paper is organized as follows: In Section II, we shortly point out previous gait optimization approaches in the field of legged RoboCup leagues, for both four-legged and biped robots. Afterwards, we introduce the robot platform and the gait modeling with its parameters. This is followed by the description of the PSO algorithm in section V. We then outline the experimental setup that includes a description of the optimization process. In section VII we present the experimental results of the gait optimization performed. Finally, section VIII concludes this work and gives a short outlook on ongoing and future work on this topic.

II. State of the Art

During recent years, the RoboCup Four-Legged League has become a popular testbed for gait optimization approaches. The reason is probably the reliable standard platform AIBO by Sony that allows researchers to easily compare and apply approaches and their results. Variations of evolutionary algorithms were applied in [1] and [2] to achieve fast and stable omni-directional gaits on the AIBO robot. In a later approach of [3], machine learning methods were used to optimize the fast walking with a focus on head stability to maintain the robot's visual capabilities.

In most cases, the knowledge achieved in the Four-Legged League can be transferred to the Humanoid League [4]. There, some approaches already show exciting results in walk speed and stability. The team Darmstadt Dribblers applied "Sequential Surrogate Optimization" to speed-up their 55 cm tall robot to 40 cm/s [5]. The KHR-1 robot that is similar to the one used in our approach, but without carrying a PDA and batteries, could be accelerated to walk speeds up to 22 cm/s, by using evolutionary algorithms as shown by [6]. This work also provides a survey of suitable optimization techniques. In parallel to our work, Faber and Behnke [7] optimized the speed of the forward gait of one of their soccer robots to 34cm/s by using Policy Gradient Reinforcement Learning.

III. The Robot Platform

The commercially available robot kit KHR-1 from Kondo was used as a basis for the robots that competed as the joint team "BreDoBrothers" of the Universität Bremen and the University of Dortmund during RoboCup 2006. The original version has a total of 17 degrees of freedom. There are five in each leg, three in each arm and one in the head (cf. Fig. 2). The hinge joints are realized by servo-motors of the type Kondo KRS-784ICS. The humanoid structure enables the robot to walk and stand up. The major deficit of the leg structure is the absence of a hip joint that would allow the robot to rotate and walk curves as a human. The rotation can therefore only be achieved by sliding the soles of the feet on the floor in different directions and using the friction to change the orientation of the robot.

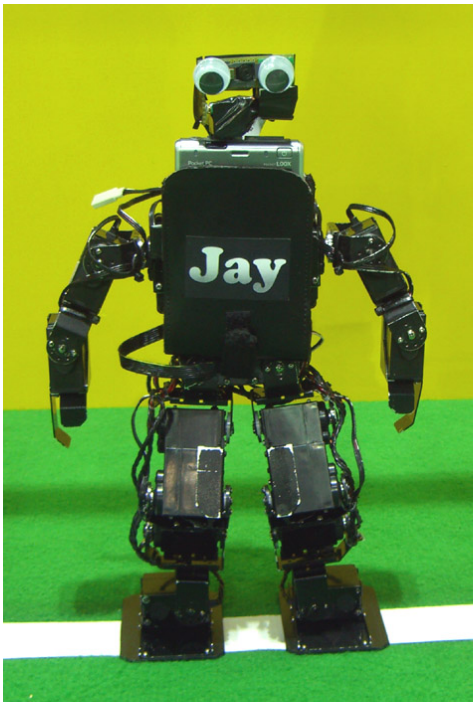


Fig. 1. The modified Kondo KHR-1 with the added PDA on the upper body. The camera head with two additional hinge joints was removed during optimization.

Several modifications of the basic robot kit were necessary to use the robot in RoboCup competitions (cf. Fig. 1). The original controller board was replaced by an more reliable and faster custom-made board that includes an Atmel ATMega128 controller, three accelerometers and one gyroscope. The head of the robot was extended by a pan-tilt unit that consists of two small servo motors and carries a camera. A Pocket LOOX PDA by Fujitsu Siemens was added to the robot for all on-board computations. In addition, the servo motors were upgraded with more reliable and stronger metal gears. Also the original batteries were exchanged by lighter Lithium-Polymer batteries and the base areas of the feet were enlarged to 90×65 mm^2 each in order to get more static stability. With these modifications, the robot weights about 1500 g and is 38 cm tall when standing; the leg length is 21 cm.

IV. Gait Modeling

The control software of the robot is based on the widespread software framework of the GermanTeam [8], ported to run on Microsoft Windows Mobile on the PDA. On the Kondo, the framework sends new joint angles to the servos every 12 ms. These joint angles are computed by the *walking engine* from a vector that describes the desired motion speeds in forward, sideward, as well as rotational direction and a set of parameters that describe the gait in general. The trajectories of the feet are first calculated in Cartesian space, and transformed into sequences of joint angles afterwards using inverse kinematics. The parameters of a gait shape these trajectories as well as how much the robot's weight is shifted during walking, and

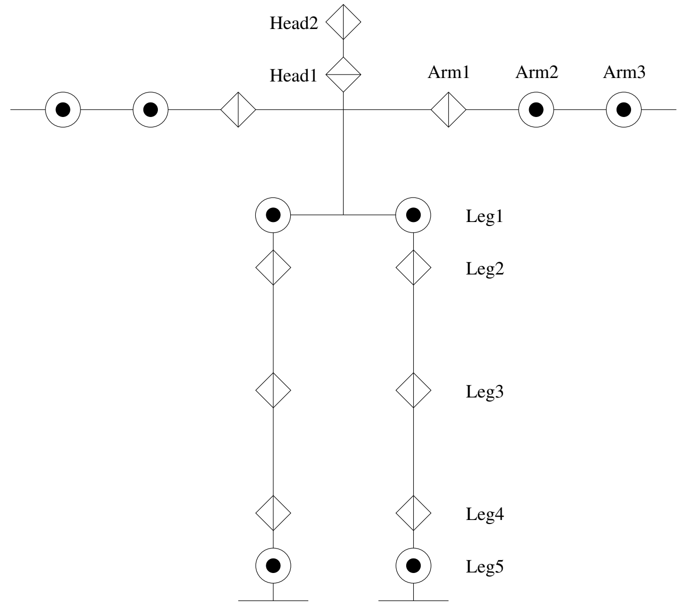


Fig. 2. The kinematic chain of the humanoid robot Kondo KHR-1 with additional head joints

whether or how much the arms swing.

A complete trajectory covers the duration of a walk motion cycle and can be constructed by a sequence of two-dimensional points. The values between the points are described by parts of standard functions, e.g., sine or sigmoid.

The foot positions relative to the center of the hip can be calculated by a combination of four trajectories. Two of these trajectories shape the foot movement and the other two the movement of the robot's upper body. The trajectory *stepX* (cf. Fig. 3) describes the foot movement in walk direction and the trajectory *stepHeight* (cf. Fig. 3) controls the lifting of the foot off the ground. The robot bends from the waist according to the trajectory *bodyTilt* (cf. Fig. 4). Furthermore, the trajectory *bodyShift* (cf. Fig. 4) shifts the upper body sideways to allow lifting the foot without load off the ground. In addition to these trajectories, the motion of the arms is also described by two additional parameters.

There is a set of parameters which control the size and the shape of the trajectories. The general parameter *stepDuration* defines the duration of a walk motion cycle that contains two steps. The three parameters *stepOriginX*, *stepOriginY* and *stepOriginZ* set the origin of the steps in Cartesian space relative to the center of the hip. The ratio between the step length in front and behind the *stepOriginX* is set in the parameter *stepRearFrontRatio*. The parameter *doubleSupport* influences the phase during a walk motion cycle, where both feet have ground contact within the trajectory *stepX*. The trajectory *stepHeight* can be adjusted by a total of six parameters. A walk motion cycle is basically divided into four phases for each leg (ground, lift, air, down). While the foot is lifted off the ground, the trajectory is described in dimensions by three parameters. The parameter *stepHeightAir* sets how high the robot lifts a foot off the ground. The parameter

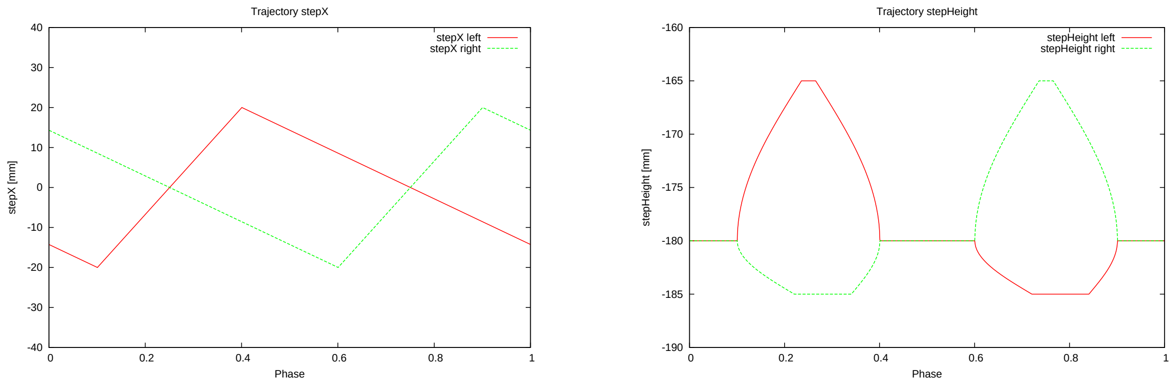


Fig. 3. The trajectories *stepX* and *stepHeight* shape the movement of the legs.

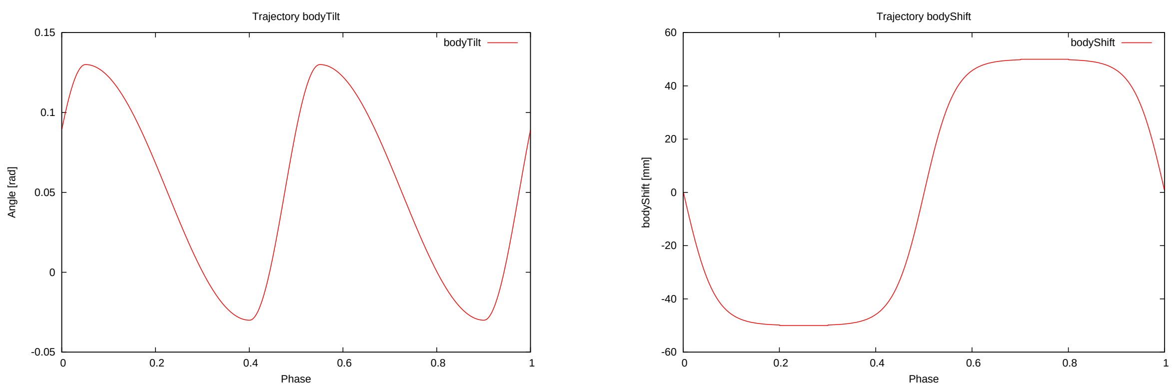


Fig. 4. The trajectories *bodyTilt* and *bodyShift* regulate the movement of the robot upper body.

stepHeightAirPick decides at which point during the lift phase the maximum height is reached, and the parameter *stepHeightAirLength* specifies for how long it remains in the maximum. The parameters *stepHeightGround*, *stepHeightGroundPick* and *stepHeightGroundLength* have the corresponding meaning for the phase when the opposite leg is lifted.

For the trajectory *bodyShift*, the parameter *bodyShiftOrigin* sets its origin, the parameter *bodyShiftFootDiff* defines the magnitude of the oscillation in dependency to the *stepOriginY*. The parameter *bodyShiftPause* defines the duration of a pause in the maximum positions. In addition, the parameter *bodyShiftPhaseShift* specifies a temporal shift relative to the trajectory *stepX*. The parameters define the trajectory *bodyTilt* as angle values with *bodyTiltOrigin* as the origin, *bodyTiltScale* as the magnitude of the oscillation in body direction and *bodyTiltBackFrontRatio* as the ratio between leaning backward and forward. Equal to the trajectory *bodyShift*, the parameter *bodyTiltPhaseShift* marks the temporal shift relative to the trajectory *stepX*. The movement of the arm is synchronized with the opposite leg's trajectory. The scale of the arm movement is defined by the *armTiltScale* and the parameter *armRollOrigin*, which defines the origin of the hand in sideway direction.

V. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is an optimization approach with stochastic characteristics that imitates the behavior of biological populations, e.g. a flock of birds searching for food. In order to optimize the gait parameter set, PSO has been chosen with a focus lying on quickly finding suitable results.

PSO was introduced by Kennedy and Eberhart in [9], [10], and it is inspired by studies of fish and bird flocks by [11]. The basic idea is to use the biological swarm behavior of animal populations to resolve numerical optimization problems in multi-dimensional search spaces.

Therefore, PSO is an algorithm that makes use of the advantages of social systems and their dynamic to solve optimization problems. Each member of the swarm, which is referred to as *particle*, has a position and velocity inside the borders of the search space and displays a possible solution to the problem. In each iteration, the particles test their position and collect experience about the search space. After each iteration, each particle communicates its best known position to a number of other particles. The new position of each particle in the next iteration is based on its own knowledge, the information from the other particles, and its last position and velocity. Through the mass of collected information and its usage by the whole swarm, a directional and parallel search for the optimum is achieved. The stochastic weighting of the different available information provides the stochastic character of the optimization process.

The PSO algorithm proposed as the *Standard PSO Version 2006* requires the adjustment of a number of parameters that – in combination with the given optimization problem – change the behavior of the algorithm and its performance strongly. The following table gives a short overview about the parameters used and their meaning:

- *sizeOfSwarm*: Determines the number of particles in the swarm.
- *formOfNeighbourhood*: The form of neighborhood within the swarm describes in which way the particles communicate with each other and how the information flows within the swarm. Using the form *full*, each particle is informed by all other particles. Using the form *k-random*, each particle is informed by only k other particles.
- *first cognitive confidence coefficient*: The first coefficient c_1 describes the confidence of the particle in its own current velocity.
- *second cognitive confidence coefficient*: The second coefficient c_{max} describes the confidence in the information about the best known positions both of itself p_d and other particles g_d. For the update, these values are multiplied by random values between 0 and 1 ($r(0,1)$).
- *maximum velocity*: The maximum velocity sets the maximum distance that a particle is able to move inside the dimension borders within a single iteration.

In each iteration, first the velocity v_d and afterwards the position x_d of the particle in each dimension is updated according to the equations 1 and 2.

$$v_d = c_1 v_d + c_{max} r(0,1)(p_d - x_d) + c_{max} r(0,1)(g_d - x_d) \quad (1)$$

$$x_d = x_d + v_d \quad (2)$$

VI. Optimization Process and Setup

The optimization was performed in two stages. First a simulator was used to find suitable parameters for the PSO itself, and afterwards these settings were used to optimize the gait of the real robot.

A. Optimization in the Simulator

Before the optimization on the real robot, different sets of parameters for the PSO were evaluated using a simulated model of the KHR-1 robot (cf. Fig. 5). The simulations were conducted in SimRobot [12], a physical robot simulator that is capable to simulate user-defined robots in three-dimensional space since it includes a physical model based on rigid body dynamics. The goal was to find out how the results differ when the size of the swarm and the form of the neighborhood within the swarm varied. Four combinations from two values for the size of the swarm (12, 20) and the neighborhood (*full* and *k-random*) were chosen. The maximum number of iterations was set to 25 in order to test whether a good walk parameter set can be found within this range. Similar to the testing performed later on the robot, an accelerated walk with a time limit was chosen, and the distance covered in walk direction during the test defined the fitness value. All other PSO parameters were set to the recommended values proposed in [13].

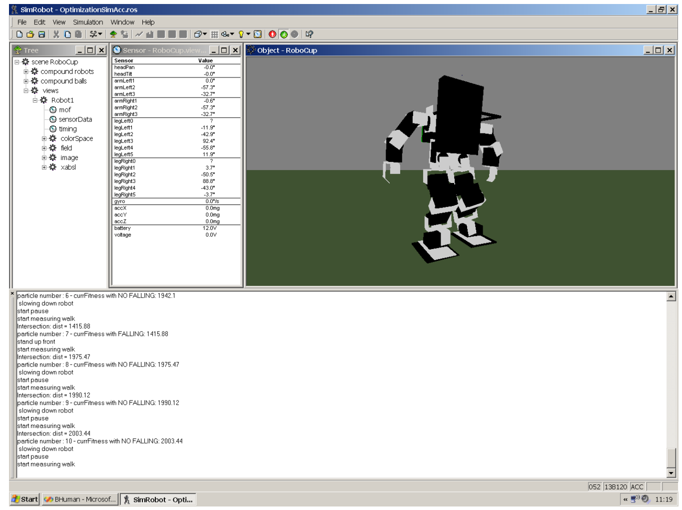


Fig. 5. Optimization in SimRobot

The main result of testing in the simulator was that the most hardware-saving combination of 12 members and the neighborhood form *full* are capable to bring up fast and good results. Therefore, this combination was chosen to be used for the parameter optimization on the real robot KHR-1.

B. Optimization on the Real Robot

The optimization scenario (cf. Fig. 7) is set up on a *RoboCup Humanoid Kid-Size League* field. The green carpet is of comparable quality to the surface that is used at official competitions. In order to get rid of the task of changing batteries, the power supply of the robot is covered via cable by a power supply unit. Therefore the robot operates on a constant voltage level during the test runs. However the robot still carries the battery that it would require for a fully autonomous operation. The distance measurement is performed by manual metering from a laid-out tape. Furthermore, a PC – connected via WLAN with the PDA on the robot – is used to enter the measured fitness values, and thus it completes the setup.

During a complete test run, we optimized five of the main eight walk directions (cf. Fig. 6). The main walk directions are: straight ahead and backward walking, walking sideways to the left and the right, and also walking in all four diagonal directions. To obtain parameters for all eight main walk directions, we mapped the optimized parameter set for diagonal walking to the walk directions on the other body sides.

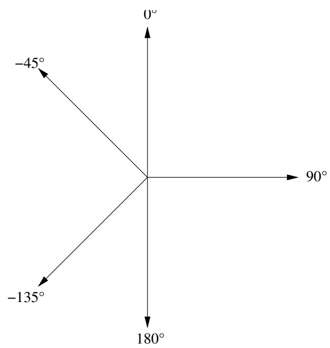


Fig. 6. The walk directions optimized that are later mirrored along the body direction.

For each walk direction, a test run lasts over 25 iterations of the PSO. A single test of a parameter set is performed by letting the robot run an almost constantly accelerated walk within a time limit of 20 seconds. The step size is increased after the duration of two walk motion cycles, which equals four steps. The increasing margin was chosen in a way that the robot starts with zero speed and increases its step size to end up at the theoretical maximum step size within the time limit. In case of straight ahead walking, the increasing margin was 20 mm.

The fitness value is defined by the distance covered in the desired walk direction during the time limit of 20 seconds. The distance covered by the robot is measured along the tape and entered manually by using the PC that sends the data to the robot via WLAN. A particle test ends early if one of the following abort situations occurs:

- The robot falls down. If the distance covered until the fall is higher than 10 cm, the distance is taken as the fitness value, otherwise a fitness value of 0 is rewarded.
- The walk parameter set tested results in no movement into the desired walk direction. This is rewarded with a fitness value of 0.
- The robot's walk direction drifts away more than 45 degree from the desired walk direction. This is rewarded with a fitness value of 0.

The search space is limited by minimum and maximum values for each dimension (cf. table I). On the basis of knowledge about the robot platform, walk trajectories, and their synchronizations, the limits were chosen in a reasonable way. Furthermore, a number of parameters were set to constant values to favor symmetric trajectories, e.g. for the trajectory *stepHeight*. At the initial iteration, the particle's positions were uniformly spread within the search space.

VII. RESULTS

The result section is structured as follows: First we present the results from simulator test runs to confirm our choice concerning the PSO parameters. After this, we show the fitness charts for forward walking and list the best parameters found for forward and sideways walking, followed by the results of the average and variance tests for the best parameter sets found. A summary of other test results collected concludes this section.

TABLE I
MINIMUM AND MAXIMUM DIMENSION BORDERS

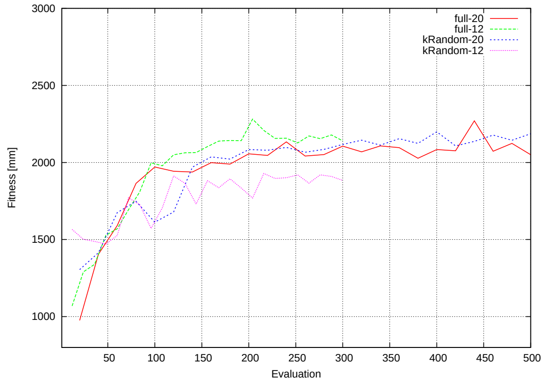
Parameter	**min value**	**max value**
stepDuration	1000 ms	2000 ms
stepOriginX	-10 mm	10 mm
stepOriginY	30 mm	50 mm
stepOriginZ	-190 mm	-170 mm
stepRearFrontRatio	0,5	2,0
doubleSupport	0,0	0,2
stepHeightAir	10 mm	30 mm
stepHeightAirPick	0,5	0,5
stepHeightAirLength	0,0	0,0
stepHeightGround	0 mm	0 mm
stepHeightGroundPick	0,5	0,5
stepHeightGroundLength	0,0	0,0
bodyShiftOrigin	0,0 mm	0,0 mm
bodyShiftFootDiff	-15 mm	15 mm
bodyShiftPause	0,0	0,0
bodyShiftPhaseShift	-0,1	0,1
bodyTiltOrigin	0,0	0,05
bodyTiltScale	-0,005	0,0
bodyTiltPhaseShift	0,0	0,2
bodyTiltBackFrontRatio	0,1	0,3
armTiltScale	0,0	0,5
armRollOrigin	-1,0	-1,0

The cursively written parameters are constant. All values are chosen by expert knowledge and experiments.

The results of the PSO parameter testing in the simulator showed the best performance by combining 12 particles with the neighborhood form *full* (cf. Fig. 8). This parameter combination *full-12* converges after about 200 evaluations and performs at least as good as the other combinations do, although they need more evaluations. Therefore, it was chosen to optimize the gait parameters on the real robot.

Among the results of the five walk directions optimized, the fitness chart for the walk direction 0 shows a usual optimization progress (cf. Fig. 8). The chart rises up to the

Fig. 7. The optimization scenario for the real robot on a RoboCup Humanoid Kid-Size League field



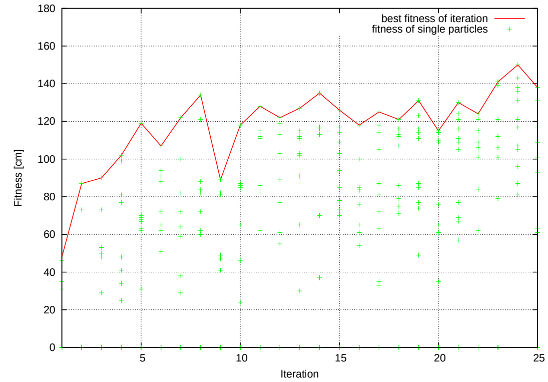


Fig. 8. The figure on the left shows the results of the PSO parameter testing in the simulator, while on the right, the fitness chart for gait optimization in walk direction 0 on the real robot is displayed.

maximum fitness of 150 cm at iteration 24. It also shows a steadily rising average level of all particles. With respect to the almost constantly accelerated motion, an approximated maximum speed of 17 cm/sec for straight forward walking was achieved.

The table II presents the best parameter sets found for the forward and sideward walking. The values of the walk parameters for both directions lie within the preset borders of the search space.

After the optimization was finished, we ran the best parameter sets found another 30 times for each walk direction to evaluate their average fitness value and its standard deviation. The average values of the best parameter sets found showed that the results that were reached during the optimization, could be repeated, respectively outperformed. The average values lie about 10% under the best values, and they also show a relatively high standard deviation.

In addition, we examined the best 10% of all test runs for each walk direction to find out that their parameter values form a narrow cluster within the search space. This information can be used to limit the borders of the search space further for future optimizations.

In closing tests the optimized gait parameter sets were integrated into the walking engine and linear interpolation was used to achieve omni-directional walking. The robot was controlled via a joypad, and it showed good results in terms of interpolation between different walk directions and speeds.

The gait optimization for a single walk direction with 25 iterations and 12 particle tests per iteration lasts between 3 and 3.5 hours. This represents a relative short duration in comparison to similar optimization approaches on a Kondo KHR-1, such as, e.g., [6].

TABLE II

BEST WALK PARAMETER SETS

Parameter	0	90
stepDuration	1399,52 ms	1154,03 ms
stepOriginX	-3,46 mm	-3,98 mm
stepOriginY	37,90 mm	49,88 mm
stepOriginZ	-182,16 mm	-190 mm
stepRearFrontRatio	1,703	1,680
doubleSupport	0,062	0,0
stepHeightAir	21,42 mm	13,71 mm
bodyShiftFootDiff	8,06 mm	-11,52 mm
bodyShiftPhaseShift	-0,030	-0,069
bodyTiltOrigin	0,048	0,042
bodyTiltScale	-0,0025	-0,0023
bodyTiltPhaseShift	0,027	0,091
bodyTiltBackForthRatio	0,166	0,100
armTiltScale	0,351	0,433
Fitness	150 cm	96 cm

Results after a total of 25 iterations for the walk directions of 0° and 90°.

VIII. Conclusion and Ongoing Work

In this paper we presented a fast and hardware-saving optimization approach based on Particle Swarm Optimization to optimize the gait of a biped humanoid robot. The parameters of a gait were optimized for a large range of walk speeds and different walk directions. The best parameter set for the main walk direction was interpolated to achieve omni-directional walking with speeds of up to 17 cm/sec. The main advantage of the PSO approach is the possibility to obtain results comparably fast.

An ongoing work is the transfer of the optimization approach to the RoboCup Simulation League, since a physical model of a humanoid robot was introduced in this year's 3D Simulation Competition. The gait modeling and optimization is integrated into the framework of the team *Virtual Werder*.

TABLE III

DISTRIBUTION OF BEST FITNESS VALUES

walk direction	0	90	180	-45	-135
avg value	140,1	89,1	84,1	89,47	70,87
std deviation	12,027	14,072	5,938	9,677	12,998
min value	100	54	68	64	36
max value	152	109	91	105	83

Average fitness values and standard deviations of the best parameter sets found for the different walk directions.

Using the PSO approach for behavior optimization, e.g., approaching the ball with integrated obstacles avoidance, is also a research topic.

ACKNOWLEDGEMENTS

The authors would like to thank all members of the BreDoBrothers team as well as the B-Human team for their supporting work. This work has been funded by the Deutsche Forschungsgemeinschaft in context of the Schwerpunktprogramm 1125 (Kooperiende Teams mobiler Roboter in dynamischen Umgebungen) as well as in context of the Sonderforschungsbereich TR8 (Spatial Cognition).

REFERENCES

[1] T. Röfer, "Evolutionary gait-optimization using a fitness function based on proprioception," in *RoboCup 2004: Robot World Cup VIII*, ser. Lecture Notes in Artificial Intelligence, no. 3276. Springer, 2005, pp. 310–322.

[2] 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*, ser. Lecture Notes in Artificial Intelligence. London, UK: Springer-Verlag, 2006.

[3] M. Saggar, T. D'Silva, N. Kohl, and P. Stone, "Autonomous learning of stable quadruped locomotion," in *RoboCup-2006: Robot Soccer World Cup X*, G. Lakemeyer, E. Sklar, D. Sorenti, and T. Takahashi, Eds. Berlin: Springer Verlag, 2007, to appear.

[4] T. Laue and T. Röfer, "Getting Upright: Migrating Concepts and Software from Four-Legged to Humanoid Soccer Robots," in *Proceedings of the Workshop on Humanoid Soccer Robots in conjunction with the 2006 IEEE International Conference on Humanoid Robots*, E. Pagello, C. Zhou, and E. Menegatti, Eds., 2006.

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

[6] M. Hebbel, R. Kosse, and W. Nistico, "Modeling and learning walking gaits of biped robots," in *Proceedings of the Workshop on Humanoid Soccer Robots of the 2006 IEEE-RAS International Conference on Humanoid Robots*, Genoa, Italy, 2006, pp. 40–48.

[7] F. Faber and S. Behnke, "Stochastic Optimization of Bipedal Walking using Gyro Feedback and Phase Resetting," in *Proceedings of the IEEE-RAS 7th International Conference on Humanoid Robots (Humanoids2007)*, Pittsburgh, PA, USA, 2007.

[8] T. Röfer, T. Laue, M. Weber, H.-D. Burkhard, M. Jüngel, D. Göhring, J. Hoffmann, B. Altmeyer, T. Krause, M. Spranger, O. v. Stryk, R. Brunn, M. Dassler, M. Kunz, T. Oberlies, M. Risler, and etc., "GermanTeam RoboCup 2005," 2005, 247 pages. [Online]. Available: http://www.germanteam.org/GT2005.pdf.

[9] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," vol. 4, 1995, pp. 1942–1948 vol.4.

[10] R. C. Eberhart and J. Kennedy, "A new optimizer using particles swarm theory," in *Sixth International Symposium on Micro Machine and Human Science*, 1995, pp. 39–43.

[11] C. W. Reynolds, "Flocks, herds, and schools: A distributed behavioral model," *Computer Graphics*, vol. 21, no. 4, pp. 25–34, 1987. [Online]. Available: http://www.red3d.com/cwr/papers/1987/boids.html

[12] T. Laue, K. Spiess, and T. Röfer, "Simrobot - a general physical robot simulator and its application in RoboCup," in *RoboCup 2005: Robot Soccer World Cup IX*, ser. Lecture Notes in Artificial Intelligence, A. Bredenfeld, A. Jacoff, I. Noda, and Y. Takahashi, Eds., no. 4020. Springer, 2006, pp. 173–183.

[13] M. Clerc, "Standard PSO version 2006," http://www.particleswarm.info/Standard_PSO_2006.c, 2006.