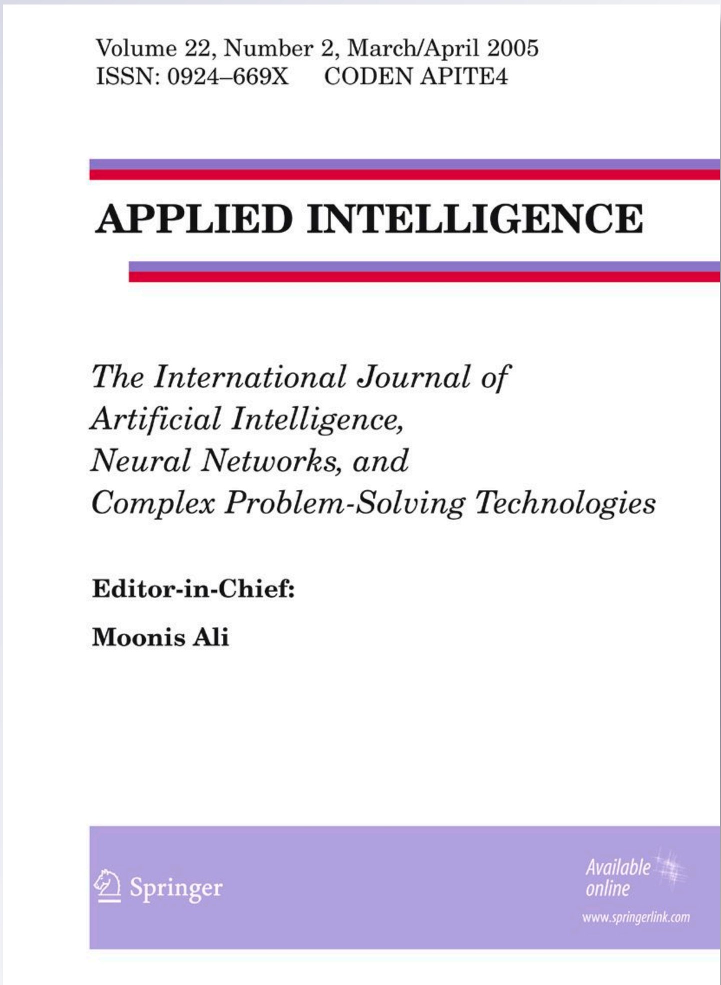
Portable autonomous walk calibration for 4-legged robots

Boyan Bonev, Miguel Cazorla, Francisco Martín & Vicente Matellán

Applied Intelligence
The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies

ISSN 0924-669X
Volume 36
Number 1

Appl Intell (2012) 36:136-147
DOI 10.1007/s10489-010-0249-9



Springer

Appl Intell (2012) 36:136–147
DOI 10.1007/s10489-010-0249-9

Portable autonomous walk calibration for 4-legged robots

Boyan Bonev · Miguel Cazorla · Francisco Martín · Vicente Matellán

Published online: 14 August 2010

Abstract In the present paper we describe an efficient and portable optimization method for calibrating the walk parameters of a quadruped robot, and its contribution for the robot control and localization. The locomotion of a legged robot presents not only the problem of maximizing the speed, but also the problem of obtaining a precise speed response, and achieving an acceptable odometry information. In this study we use a simulated annealing algorithm for calibrating different parametric sets for different speed ranges, with the goal of avoiding discontinuities. The results are applied to the robot AIBO in the RoboCup domain. Moreover, we outline the relevance of calibration to the control, showing the improvement obtained in odometry and, as a consequence, in robot localization.

Keywords Legged locomotion · Walk parameters estimation · Autonomous odometry calibration

B. Bonev · M. Cazorla (✉)
Dpto. de Ciencia de la Computación e Inteligencia Artificial, Universidad de Alicante, P.O. Box 99, 03080, Alicante, Spain
e-mail: miguel@dccia.ua.es

B. Bonev
e-mail: boyan@dccia.ua.es

F. Martín
Dpto. de Ingeniería Temática y Tecnología Electrónica, Universidad Rey Juan Carlos, C/ Tulipan s/n, 28933, Madrid, Spain
e-mail: fmartin@gsyc.es

V. Matellán
Dpto. de Ingenierías Mecánica, Informáticaa y Aeroespacial, Universidad de León, Campus de Vegazana, León, Spain
e-mail: vicente.matellan@unileon.es

1 Introduction

Locomotion and odometry constitute the two main problems in legged robotics. Legged locomotion has many parameters which have to be calibrated. For instance, the speed response differs depending on the surface, as a rough surface requires a different gait than a smooth one. Furthermore, odometry varies depending both on the characteristics of the surface and on the locomotion parameters. Different carpets require different locomotion parameters, therefore the calibration method must be portable. Portability is an advantage in the context of robotics competitions, in which the robotics system has to be calibrated and run in different places and under different conditions. Moreover, in some competitions additional installations such as zenithal cameras, sensors, etc., are not allowed. We have developed a calibration process for both walk parameters and odometry using a simulated annealing learning approach, and the application developed enables such calibration to be fully automated. The input data for the learning algorithm is the measure of the robot's instant speed, and therefore the walk calibration is relatively fast, taking about 40 minutes. The odometry calibration takes 60 minutes.

Our study has been performed in the scope of the RoboCup domain. RoboCup [15] is an international competition, the ultimate goal of which is to develop a team of autonomous robots capable of competing with the human world soccer champion team by the year 2050. The scope of our research is the Four-Legged Robot League with SONY AIBO [18] robots. Currently this league has been replaced by the Standard Platform League with a new robotic platform, the Nao robot, which is not considered in this work. The Four-Legged League field has a size of approximately 6 m × 4 m. The Aibo robot's main exteroceptive sensor is a camera, used to detect objects in the field. Objects are colour coded: there are two uniquely coloured beacons, two goal nets of a different colour, the ball is orange, and the robots wear coloured uniforms. In spite of these facilities, both perception and locomotion can be both very noisy and erroneous. For example, robots can collide with each other, the carpet characteristics are not standard and can vary from one field to another, the referee can manually remove robots and place them in different parts of the field, etc. Perception is also challenging: fast robot movements can lead to failures in perception, as well as occlusions and other external problems.

The rest of the paper is organized as follows. First, the related work in the field is outlined. After that, the experimental procedure is explained in Sect. 3. Then, we describe the learning process (Sect. 4) for both speed maximization (Sect. 4.1) and response optimization (Sect. 4.2). Finally, we describe error measurements and results (Sect. 5), and an application of the technique for enhancing localization in Sect. 5.1, concluding in Sect. 6.

2 Related work

The gait calibration was first approached as a tuning-by-hand procedure. This is a difficult task and the result was not optimal. The need to perform a gait calibration for different surfaces has motivated the use of machine learning approaches. For instance [4, 5, 8, 16] and [12] used genetic algorithms to find the parameter set which best optimized the gait. These methods were designed to improve the speed that the robot could achieve. However we were also interested in obtaining the closest speed to the desired one. Other approaches have used different optimization and machine learning approaches, such as gradient descent methods [3], multi-dimensional minimization based on Powell's method [10] and downhill simplex algorithms [12]. In [17], a method for both fast and camera-stable movement was presented. In the present study we focus on a system with an instant speed measurement, i.e., the current robot speed is measured every few seconds, so that the learning process is faster than other approaches, as explained in the rest of this section. The algorithm used in our approach is based on simulated annealing, as explained later.

Another issue in the approaches for walk calibration is the way in which the speed of the robot is measured. Some approaches have used an external camera situated above the robot's field [3]. However, although this is a very precise means of measuring the instant speed of the robot, such a system is not easy to install in new scenarios. On the other hand, there are some approaches designed to be portable. For example, in [4], robot velocity was measured by making the robot walk for a given period of time and calculating how far it had reached. A localization system was used, based on coloured marks around the field. In several other approaches, such as [12] and [10], the speed of the robot was calculated by measuring the time it took the robot to walk from one of the field's landmarks, to the opposite one. A faster method for measuring a robot's instant speed was proposed in [5], where a black and white pattern was used for localization. In [16] the authors used forward kinematics and the Aibo acceleration sensor. The approach presented in this paper uses white patterns in the field, similar to the approach of [5], in order to enable the robot to measure its speed quickly and autonomously. The patterns are also used to correct the robot's direction during its runs. Note that the camera above the field which we present later in this paper is used as a ground truth for the localization experiments, but the walk calibration system remains fully portable.

Finally, another important issue in gait optimization is the use of different sets of parameters for different speeds. Although such an approach has already been used in [5], the suitability of a parameter set for interpolation was evaluated manually. Our approach considers different parameter sets for different speed ranges. Interpolation is unnecessary because the optimization procedure we propose not only maximizes the speed but also minimizes speed discontinuity between consecutive parameter sets. Thus, when a speed change requires a switch between two parameter sets, the change in the actual speed response is smooth.

An initial version of our walk calibration system was presented in [1]. There, the initial goal was to improve walking stability and speed. In the present study, the goal of the calibration system was manifold. First, we have aimed to optimize the walking style in terms of speed and stability. Second, we have adjusted walk parameters to improve the robot's odometry. In this paper, we present a set of experiments designed to show the contribution of walk parameter calibration to robot performance in terms of localization and control. To this end, we have performed an odometry calibration of the robot with its original walk parameters and we have realised localization experiments. Moreover, we have calibrated the walk engine parameters for the concrete surface used in the experiments. Then we have performed an odometry calibration for this second set of walk parameters and repeated the localization experiments. The results show that a calibrated set of walk parameters improves localization and control of the robot. Furthermore, our method represents an improvement of previous ones as we use fewer samples to learn, speeding up the entire process.

3 Experimental setup

The calibration tool developed was used in the RoboCup competition by the "TeamChaos" team [19]. This team represents a cooperative effort involving the Örebro University of Sweden and the Spanish universities Rey Juan Carlos University (Madrid), the University of Murcia and the University of Alicante. The robot we used for the RoboCup and for the experiments was the Sony AIBO ERS-7 four-legged robot [18]. The walk implementation used was a forward, lateral and rotational locomotion walking style [6, 9]. The parameters we considered are presented in Fig. 2, and their meanings are listed in Table 1. In this paper we have not considered the walk engine itself, as our purpose was to optimize the parameters which the walk engine offers. For a better explanation of the walk parameters and the walk engine, see [3].

Our calibration experiments were based on measuring the robot's speed while walking. In order to measure the speed and guide the robot, we used a camera installed on the robot's head and a series of marks on the ground as shown in Fig. 1. We used a resolution of 208 × 160 pixels in order to segment the white colour [20] and perform a blobs analysis. A white line on the ground determined a straight trajectory for the robot. The measurement was stopped briefly when the robot reached the end of the line before it turned around to follow the line again. If the robot deviated from the line, it was able to find it again, allowing the process to be unsupervised by the user. There were also distance marks along the line, spaced 20 cm apart from each other. The robot detected these marks and calculated its instant speed based on the milliseconds it took it to reach each consecutive mark. Actually, it is not an instant speed, but a time difference between two consecutive marks. In other portable approaches, the speed was measured based on the time it took the robot to walk from one side of the field to the other. A rigorous measurement of instant speed is possible if a non portable system, incorporating a fixed external camera supervising the robot, is employed. We used such a system, but only for experimental validation of the results presented in this paper. Moreover, it is important to remark that the ground truth system was not used for walk calibration due to the portability of our method. We also applied a filter for erroneous measurements in case the robot missed a mark. In order to improve stability of the calculated speeds, we used the mean value of each three instant speed measurements as a learning algorithm input.

In addition, our self-calibration system has been designed to be highly portable to any place or surface, as the only infrastructure needed is the series of white marks, which can easily be incorporated onto any surface. No ground-truth system is required, although in Sect. 5.1 we report its use in order to demonstrate that calibration does actually improve robot odometry and localization. This is a fundamental requirement given that we compete in different countries and on different surfaces.

The control software built by our team uses the three different walking modes of forward walking, lateral walking, and turning (Fig. 1). Forward walking was the crucial mode for the present study, and was the one used in this article to illustrate the learning process. During the execution of the on-line learning algorithm, walk parameters were continuously being changed to obtain different speed responses. In order to measure the error of forward, lateral and rotational walking modes, the parameters were kept constant, as the robot performed several speed measurements for each different speed requested.

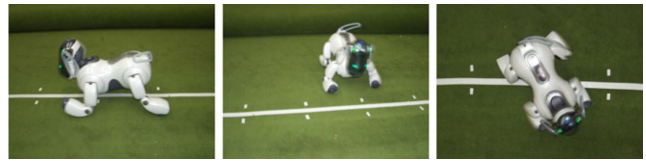


Fig. 1 Robot and distance marks for forward, lateral, and rotational walking

Table 1 Meaning of the walk engine parameters

	Front leg	Back leg
Height of the loci	hf	hb
Height of the leg	hdf	hdb
Sideways offset	fso	bso
Forward offset	ffo	bfo

4 Learning procedure

As mentioned above, our goal was to find parameter sets that both maximized the upper limit of speed and improved the response for the entire range of possible speeds. An appropriate speed response enables greater precision of control, odometry and, consequently, localization, as we will show in Sect. 5.1.

Different techniques can be applied to find a suitable set of walk parameters (see [2]). This can be seen as an optimization problem in a continuous parametric space with 8 highly irregular dimensions. Setting parameters manually takes many hours and the results are not satisfactory. On the other hand, obtaining a general analytical solution is not feasible. Varying each parameter independently and selecting the best value of each parameter is possible but not optimal, as the parameters are not independent due to kinematics constraints (for example, rear knee angles are determined by the `hb`, `hdb`, and `bso` parameters in conjunction (see Fig. 2)). This 1-dimensional search method was used to establish the limits of the parametric space and to determine an initial set of parameters for the subsequent search within the 8-dimensional space of parameters.

The algorithm we chose followed a simulated annealing [11] scheme. It comprised a probabilistic algorithm which

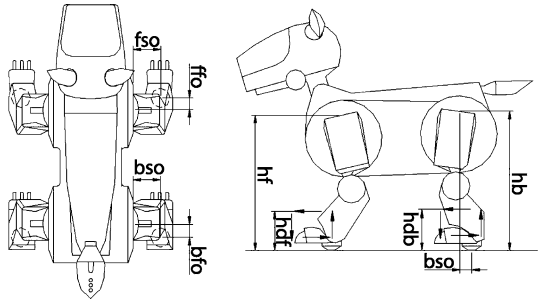


Fig. 2 Scheme of Sony's quadruped robot AIBO with the 8 parameters involved into the calibration: {hf, hb, hdf, hdb, fso, bso, ffo, bfo}

aimed to find a good solution to the global optimization problem. We do not know the function of the parameters space, so we can not provide an analytical solution. We do not even know the local minima of the function, but simulated annealing allowed us to overcome them when the temperature was high (first iterations). As the temperature decreased, the probability of jumping outside the local minima diminished.

There is a wide range of different algorithms which look for good global optimization solutions. Among them there are genetic algorithms [7, 13], tabu search, ant colony [14], and particle swarm optimization approaches. However, due to the cost of evaluating a single parameter set (the robot has to walk more than 20 cm and measure its speed), maintaining a population of solutions results in a higher complexity. In the Robocup domain, one can not use the field more than several hours to adjust our system. Because of that, our optimization algorithm must be as fast and effective as possible. Simulated annealing is a relatively straight-forward solution to our problem and its formulation is suitable for a continuous parameters space. The function to minimize is the negative of the maximum speed achieved (see (3)) plus the response optimization function (see (4)), both explained in the following subsections. As other heuristic optimization techniques, simulated annealing can reach a good solution and lose it afterwards, because of the stochastic nature of the algorithm. That is why we always keep the best evaluated solution.

4.1 Speed maximization

One of our goals was to improve the maximum speed for a given surface. Our simulated annealing algorithm explored the parametric space accepting improvements with high probability, depending on the temperature. An illustration of this procedure is shown in (1),

$$P(C) = \begin{cases} e^{-\frac{f - f_{old}}{\eta T}}, & f > f_{old} \\ 1, & f \leq f_{old} \end{cases} \tag{1}$$

where $P(C)$ is the probability of accepting the change C of parameters, and f_{old} and f are the result of the evaluation function before C and after C. The evaluation function in this problem is the measure of the instant speed of the robot. T is the temperature with an initial value of 2.0 decreasing by 10% with each iteration (each iteration has 3 steps of parameter changes). The factor η is used to scale the evaluation function and its value is the average variation of f in each step of the algorithm. For example, for f_{max} (see (3)), $\eta = 0.01$, while for f_{min} (see (4)), $\eta = 0.0001$. η is set manually. The parameters p_i are updated at intervals depending on the temperature T:

$$\triangle p_i = \mu_i \sqrt{T}\, \text{rand}(-1, 1) \tag{2}$$

Here $i \in [1, 8]$ represents any of the 8 different parameters. μ is used to scale the changes according to the range of each parameter p_i, and it is initialized with a value that allows each p_i to be a 10% maximum of the parameter's range. Therefore, from (2) it can be deduced that $\mu_i = \frac{R(p_i)}{10\sqrt{T_0}}$, where $R(p_i)$ is the range of values for each parameter and T_0 is the initial temperature. The function $\text{rand}(-1, 1)$ returns any random value in the interval $(-1, +1)$.

For the speed maximization problem, the evaluation function depends only on the maximum speed that the robot can achieve. The sign is negative because the algorithm is based on energy minimization.

$$f_{max}(speed) = -speed \tag{3}$$

The initial parameter set can influence the effectiveness of the learning process, so we started from an acceptable one, as explained in the previous subsection. Our algorithm found a good set after one hundred iterations (which took between 10 and 15 minutes), improving the initial speed by 30%. The stop criterion was based on the temperature T. When T was below a certain threshold such that further changes are improbable, the optimization process concluded. An example of speed evolution during annealing is shown in Fig. 3. Due to time restrictions we tuned the simulated annealing temperature to decrease very quickly, for this reason the algorithm could fall into a local minimum in some cases.

A result of the speed maximization is shown in Fig. 4, where requested vs. obtained (real) speed on a particular surface is represented. The maximum speed achieved depends primarily on the surface. With the walk engine used in our experiments we achieve a speed of 0.28 m/s. This speed is close to the upper physical speed limit of this walk engine. There are other walking engines which achieve higher speeds using a different gait design (although these also need parameter optimization). On soft or slippery surfaces, the maximum speed is usually lower than on a hard surface, for an example see Fig. 4.

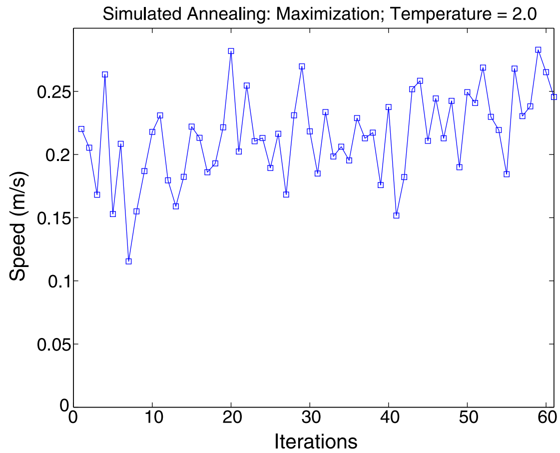


Fig. 3 Evolution of the speeds obtained during execution of the simulated annealing

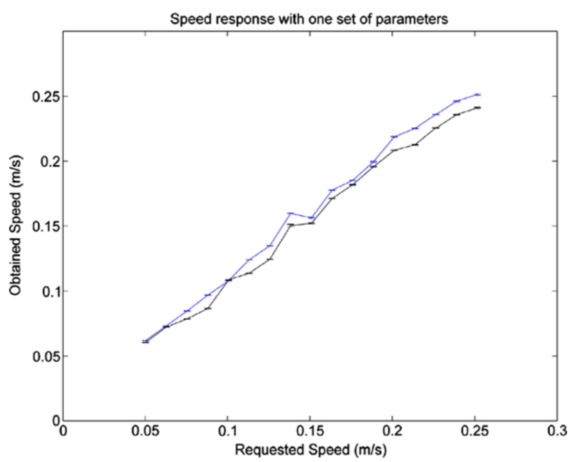


Fig. 4 (Color online) Relation between requested speed and the real speed obtained: *blue line*: on a hard surface; *black line*: on a soft surface

4.2 Response optimization

Regardless of any improvement, the maximum achievable speed is physically limited. The next objective was to improve the relationship between the requested and the obtained speed. This relationship is not linear when the same set of parameters for the entire range of possible speeds is used (Fig. 4), and as a result imprecisions in the control of the robot and incorrect odometric estimations arise. This effect is partially due to the locomotion model and also to the physical characteristics of the surface on which the robot moves. Our goal in this process is to get a linear response in requested versus obtained speeds.

Our proposal is based on using different sets of parameters for different speed intervals. However, one constraint of the problem is the need for continuity between intervals. To perform such a calibration, we modified the learning algorithm in order to obtain the real speeds (v_l', v_u') for the lower and upper limit of the definite speed interval $[v_l, v_u]$ simultaneously.

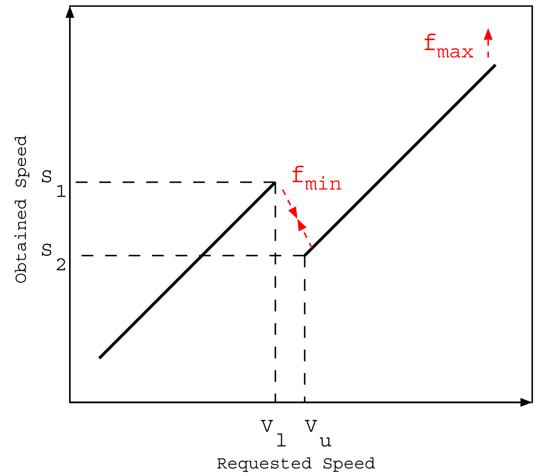


Fig. 5 The roles of the f_{min} and f_{max} objective functions are to minimize discontinuities in the speed response, and to maximize the maximum speed response

When the goal for the simulated annealing is a concrete value of speed, the difference between desired and obtained speed must be minimized. For each pair of consecutive speed intervals we had two goals, so we alternated measurements of the speeds v_l and v_u, using the objective function:

$$f_{min}(s_1, s_2) = |(s_1 - v_l)^2 - (s_2 - v_u)^2| \tag{4}$$

where s_1 and s_2 are the actual speed measurements, and v_l and v_u are the desired speeds sent to the walk engine, as already explained. The aim of this objective function was to achieve a smooth speed change between two consecutive speed intervals. Both objective functions are illustrated in Fig. 5. Thus, for $n > 1$, calibration process parameter sets would be as follows:

1. Find a parameter set by means of speed maximization. Thus the achieved maximum speed is v_{max}. Determine the values $v_{n-1}, v_{n-2} \dots v_1$ so that $v_{max} > v_{n-1} > v_{n-2} > \cdots > v_1 > 0$.
2. Optimize the parameters set S_n for the speed interval $[v_{n-1}, v_{max}]$
3. Similarly, for each i, $n > i \geq 0$ optimize the parameter set S_n for the speed interval $[v_{i-1}, v_i]$.

Thus we attempted to approximate the speed responses of the lower limit of an interval S_i and the upper one of the next interval S_{i-1}.

4.3 Discussion

The number of parameter sets depends on the locomotion model, i.e., different locomotion models need different number of parameter sets. In our model, we have tested different numbers of parameter sets. In Fig. 6 we show the resulting speed responses for 1, 2, and 6 sets, which correspond to the best results obtained. In this figure, the requested speed versus obtained speed is shown. Why a number of a parameter set is better than other? Because we experimentally show that different numbers of parameter sets yield different speed responses. For example, when using one set at 0.15 the obtained speed is 0.17, which is not desirable. We experimentally show that with two parameter sets the obtained speed has less discontinuities and is closer to the requested speed.

Although it can be difficult to avoid discontinuities between the different speed intervals, with two parameter sets it was possible to avoid discontinuity at 15 cm/s (due to the design and implementation of the walk engine [9]), that is, it was possible to minimize the difference between requested and obtained speed. For that reason, we have used 2 parameter sets for the rest of the paper.

5 Experimental validation

Once the walk parameters have been calibrated, our simulated annealing approach obtained a very good approximation of the requested speed. However, odometry calibration was still necessary in order to take into account the differences that might still exist following calibration, as well as any errors and variances. Measuring the error is necessary in order to carry out a precise estimation of accumulated uncertainty while the robot walks. Error depends both on the surface and the parameters set, so these measurements must be taken after each new calibration of the walk.

In order to measure the error we discretized the speed range into 20 values. We took 10 speed measures for each of them, in order to obtain their associated error and variance. Figure 7 shows the speed responses of forward, lateral and rotational walking and their associated error measures. The left column shows the average of the measured speeds. Each measure has an error with respect to the desired speed. The right column represents the mean error from ten experiments. The error of the figures on the right does not refer to the difference between desired and obtained speeds. It is the error among different repetitions of the same experiment. The values of these error measures are used for the odometry calibration and they are important for a good localization, as explained in the following subsections.

The localization experiments presented in the following section were carried out according to Fig. 8, using a calibrated odometry for both unoptimized and optimized walk parameters using the system previously described. The goal of these experiments was to show the contribution of the calibration to the real performance of the robot.

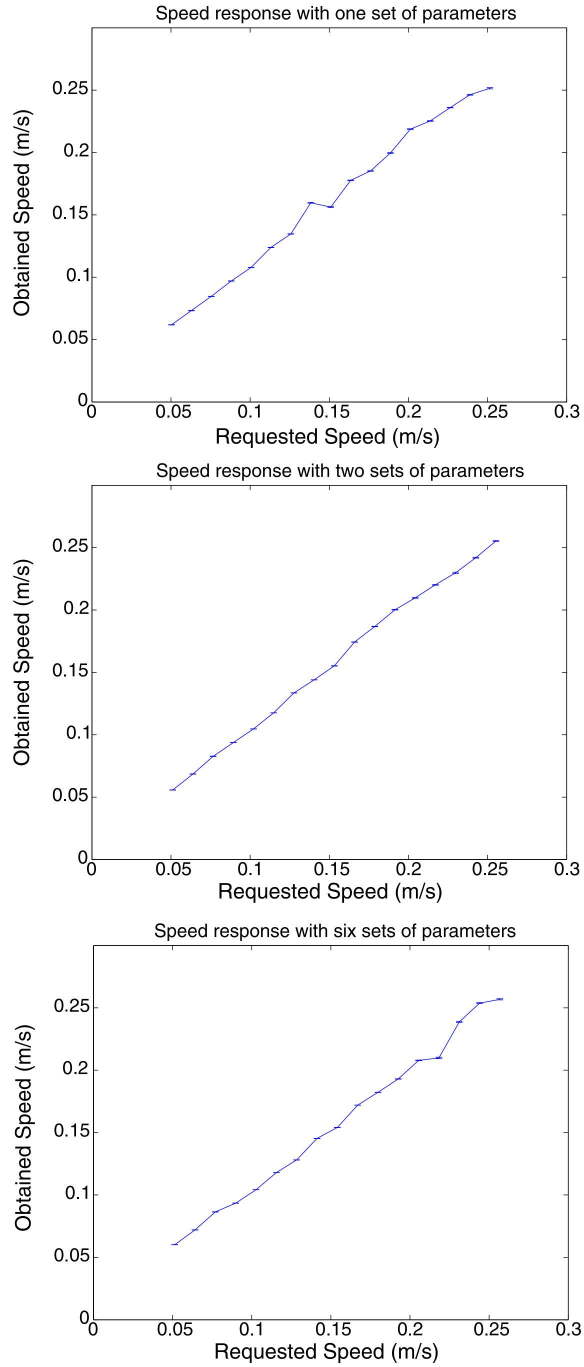


Fig. 6 Relationship between requested and obtained speed for: (**a**) 1 set of parameters; (**b**) 2 sets of parameters; (**c**) 6 sets of parameters

5.1 Localization enhancement

Once the parameters have been obtained, we wanted to test their influence on other player modules. We chose self-localization ability because it significantly influences the global behavior of the player (localization information is used in role assignment, movement decisions, etc.). The majority of probabilistic localization algorithms are based on

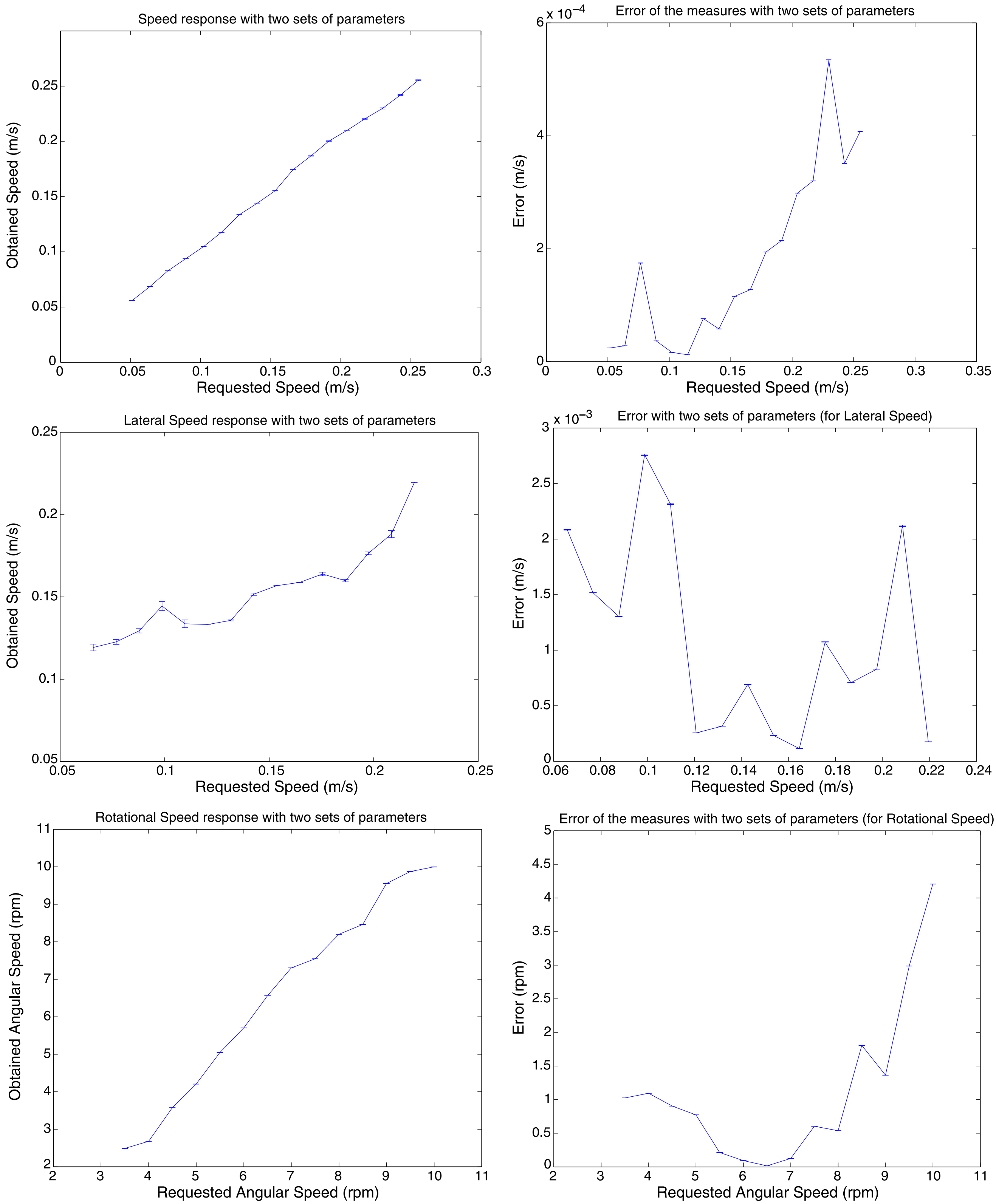


Fig. 7 *On the left*, the speed responses for forward, lateral and rotational walking; *on the right*, their associated error measures (requested speed vs the obtained speed)

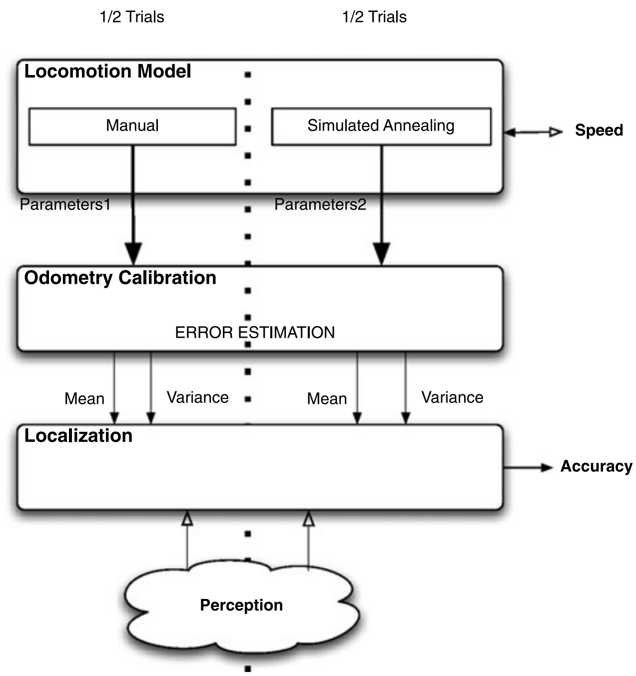


Fig. 8 Flowchart of the experiments

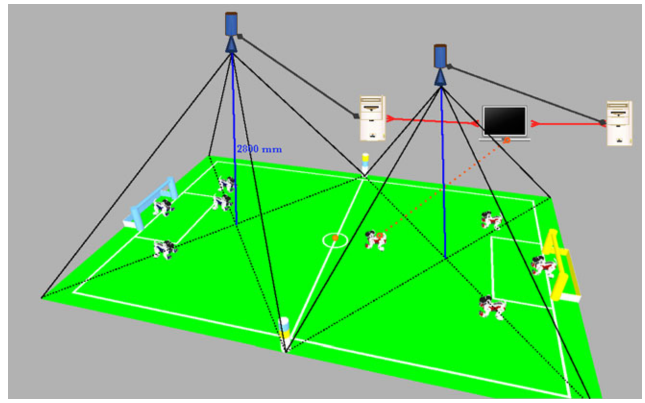


Fig. 9 Ground truth system

Table 2 Precision of the test points (mm)

	$distance_x$	$distance_y$	$distance$
Mean	32	18.2	39.24
Variance	15.9	10.5	13

two phases. In the *prediction phase* we used a movement model to calculate how a new action modified the robot's state. These models were primarily based on the odometry produced by a motion module. In the *correction phase* we used sensory information to adjust the predicted robot's state. Success in the localization process depends to a large extent on the quality of the odometry calculated, and the calculated error of this information.

In this section we will analyze how the localization process was improved by using the proposed method. First, we used the uncalibrated parameters, which had been optimized using the previously described method on a real robot, to describe several trajectories in a Robocup field. We calibrated the walking parameters with the proposed method and then made the robot follow the same trajectories. The results showed how the localization process was improved using the proposed calibration method.

A ground truth system was used to compare the actual position with the one estimated by the robot, as can be seen in Fig. 9. We used two zenithal cameras connected to two computers which functioned as servers. The client (represented as a laptop in the figure) was connected to the servers and to the robot in order to send them a signal to start the tracking process. In this process, the robot started to locally record all the sensor information perceived in a log file, and the servers connected to the cameras started to locally record the actual robot pose in a log file. All the elements reset their clocks and responded to the init signal with an acknowledgement message in order to measure the time difference between log files (always lower than 6 milliseconds). Once the client had sent a signal to stop recording to all the elements, all the log files were mixed. The purpose of this mechanism was to avoid networking latencies.

Ground truth precision had previously been calibrated to ensure the correctness of the calculated robot position. The set of points used in the calibration are shown at the top of Fig. 10. Error in the ground truth system is summarized in Table 2. The maximum error was 64 mm, which corresponds to points at the border of the image. The mean error was 32 mm, which we considered sufficiently accurate to validate the ground truth system (considering that the size of the robot is 310 × 180 mm).

As a result of the movement improvement proposed in this paper, the odometric information was more accurate and the error estimation was also improved. The localization algorithm used was an Extended Kalman Filter (EKF). We used this algorithm because it can be considered a standard tracking algorithm, where both observations and odometry accuracy are key factors in its success. The goal of these experiments was to test how the localization process was improved, only by improving the odometric information. Because of this, in this section we show the localization results using the previous locomotion and the improved one.

We have designed three different experiments where half of the trials was done using sets of walk settings obtained with our previous manual system, and the other half was performed with the simulated annealing-based system settings. In both cases, the localization system was initialized at the robot's real position and updated with the odometry and sensory information. Odometry was calibrated for both experiments using the self-calibration method previously described.

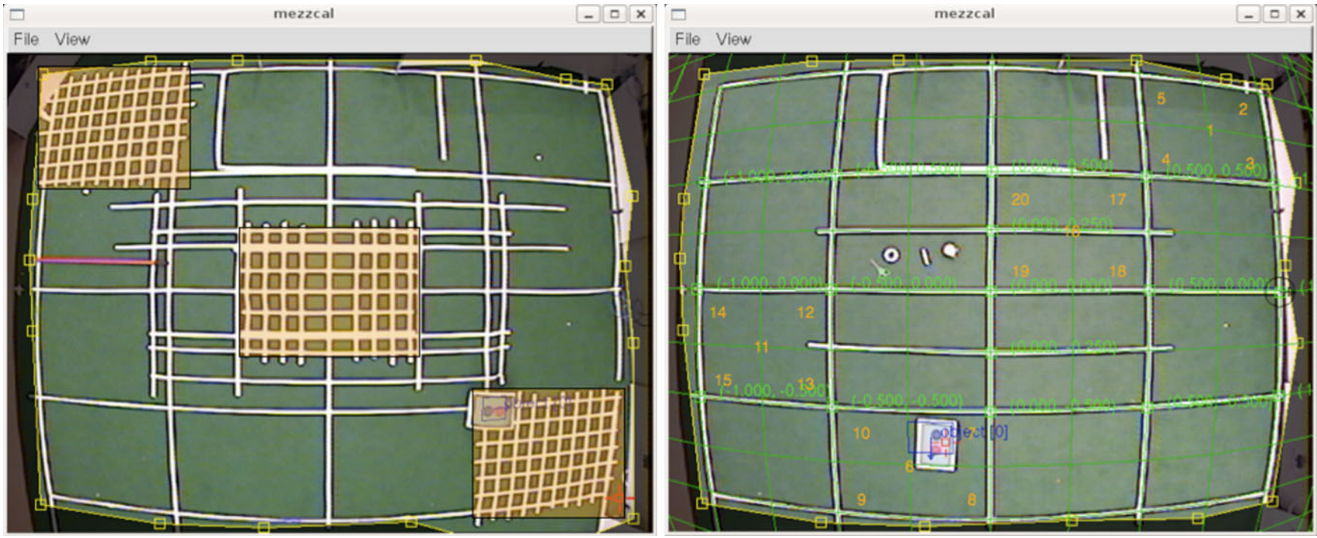


Fig. 10 Calibration and test points

The first experiment consisted in the robot being commanded a rectilinear movement at variable speed. Figure 11-above, shows the trajectory of a single trial carried out during this experiment. The left graph shows the trajectory followed when using the manually obtained settings, and the right graph shows a trial using the learnt walk parameters, both using the calibrated odometry. The trial using manually obtained settings demonstrated that linear odometry is shorter than actual odometry, and the calculated error in the odometry did not let localization system correct the estimation. The trial with calibration obtained using the simulated annealing system was better. In this case, the linear odometry corresponded to the actual displacement, and the deviation in the trajectory could be corrected based on a good estimation of the error. The second experiment comprised two linear movements and a rotational one. The trajectories for this second experiment are shown in Fig. 11-middle. As in the first experiment, the difference between the position calculated by the ground truth and the estimated position was more accurate in the calibrated case. Finally, in the third experiment, the robot moves in a circle making three turns. The results are shown in Fig. 11-bottom and, again, the calibration obtained using the simulated annealing system was crucial for correct estimation of the robot position. In this case the error in the manually calibrated system is very large and the final position and orientation are completely wrong. As the results show, the problem is not due to the localization method, but to the calibration of the odometry.

The numerical results are summarized in Table 3. This table presents an analysis of the error in robot pose estimation and confirms that the calibration obtained using the simulated annealing system was decisive for a correct position estimation.

Table 3 Error in the estimated position (mm)

type	mean	stdev	median	maximun
Original settings	402.52	305.04	366.43	1555.2
Calibrated odo.	208.21	124.87	177.34	586.30

6 Conclusions

The use of walk calibration using machine learning proved to be feasible, as well as necessary, especially when moving the system to different surfaces. The TeamChaos team improved its maximum speed by 30% using a machine learning method based on simulated annealing. Furthermore, the use of multiple parameter sets enabled us to improve the speed response by correcting discontinuities in the speed space. We obtained our best result when using 2 sets of parameters. To the best of our knowledge, this is the first time that this approach has been applied. Moreover, the precise error measurements provided the odometry system with more precise information about motion uncertainty. We have also shown how the system improved robot self-localization, which has greatly increased the efficiency of our player. The process was faster when using our algorithm than applying previously described methods. The system is also fully portable, which is an important issue in many mobile robotics applications, especially in terms of the Robocup competition.

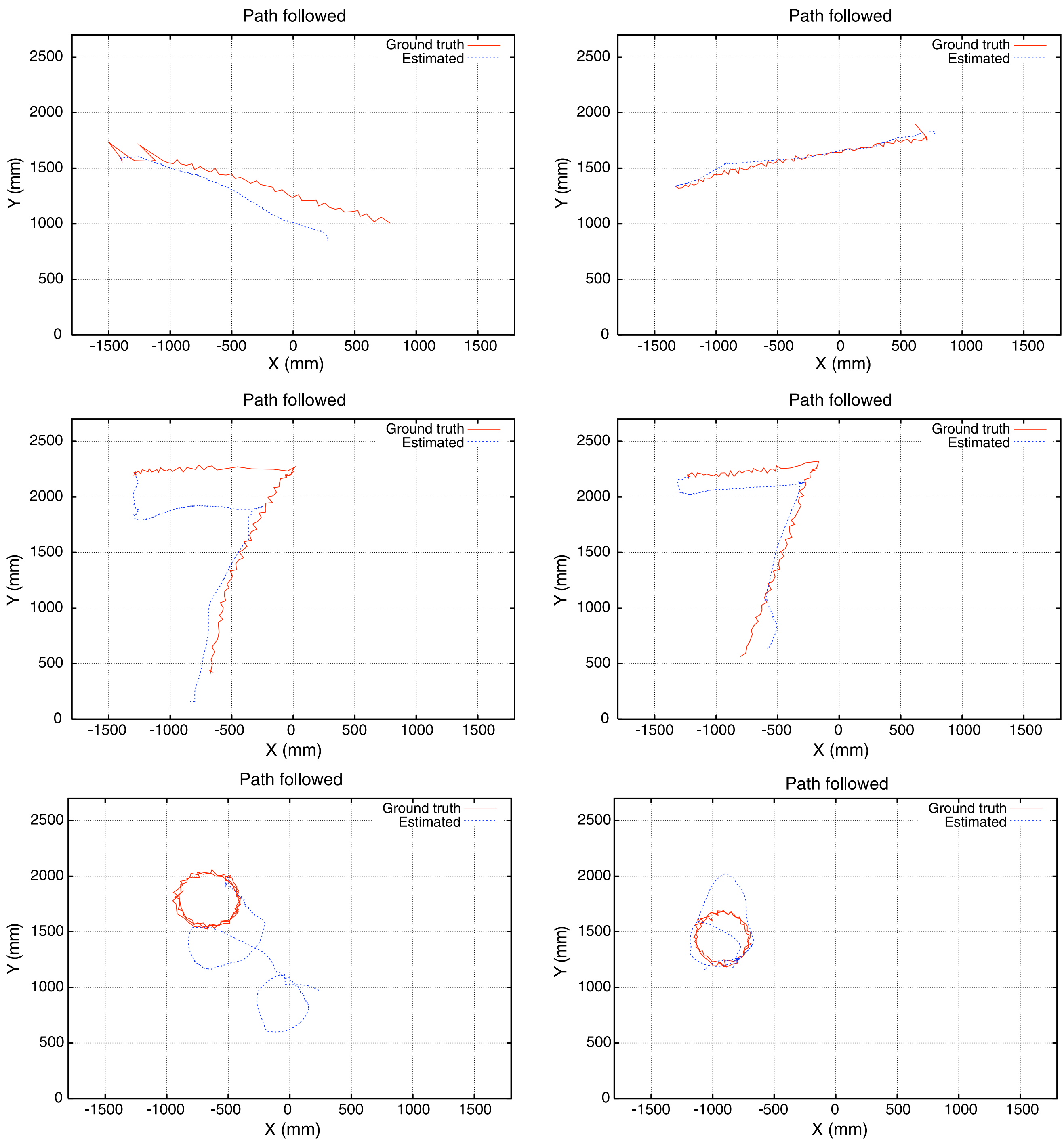


Fig. 11 (Color online) Real trajectory is shown in *red line* (obtained by the ground-truth system). The estimated position is represented by the *blue line*. The *left graphs* give the result for manually calibrated odometry and the *right* ones gives the result for calibration obtained using the simulated annealing system. *Above*: linear trajectory; *middle*: lateral trajectory; *bottom*: circular trajectory

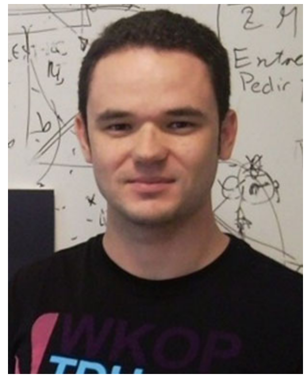
Regarding further research, calibrating the walk parameters for forward speed only constitutes a good solution for the RoboCup domain. Using the infrastructure of our experiments, a study of more complete calibration could be carried out by simultaneously calibrating forward, lateral, and rotational walking. This is not necessary in our domain and it would take much more time; nevertheless, it would be possible thanks to our instant speed measuring procedure.

Finally, a calibration of curved lines could be considered, instead of separating calibration into three different walking types.

Acknowledgements The authors wish to thank the TeamChaos team (http://www.teamchaos.es) for all their support during the experiments. This research has been partially funded by the Spanish Ministerio de Ciencia y Tecnología, DITPA project (DPI2004-07993-C03), by the Generalitat Valenciana project GV06/134, and by the Comunidad de Madrid as part of the RoboCity 2030 project (S-0505/DPI/0176).

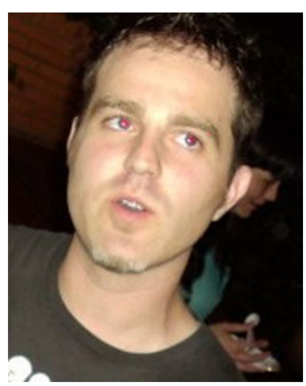
References

1. Bonev B, Cazorla M, Martínez H (2005) Walk calibration in a four-legged robot. In: Proceedings of CLAWAR (climbing and walking robots). London (UK)
2. Chalup SK, Murch CL, Quinlan MJ (2007) Machine learning with AIBO robots in the four-legged league of RoboCup. IEEE Trans Syst Man Cybern 37(3)
3. Chen W (2005) Odometry calibration and gait optimisation. PhD Thesis. The University of New South Wales
4. Chernova S, Veloso M (2004) An evolutionary approach to gait learning for four-legged robots. In: Proceedings of IROS'04, Sendai, Japan
5. Düffert U, Hoffmann J (2006) Reliable and precise gait modeling for a quadruped robot. In: Proc 9th international workshop on RoboCup 2005 (robot world cup soccer games and conferences). LNAI. Springer, Berlin
6. Hafmar H, Lundin J (2002) Quadruped locomotion for football robots. Master Thesis, Chalmers University of Technology, Goteborg, Sweden
7. Hanshar F, Ombuki-Berman B (2007) Dynamic vehicle routing using genetic algorithms. Appl Intell 27(1):8999
8. Hebbel M, Nistico W, Fisseler D (2007) Learning in a high dimensional space: fast omnidirectional quadrupedal locomotion. In: RoboCup 2006: Robot soccer world cup X. LNAI, vol 4434. Springer, Berlin, pp 314–321
9. Hengst B, Ibbotson B, Pham P, Sammut C (2002) Omnidirectional locomotion for quadruped robots. In: Proceedings of RoboCup 2001. Springer, Berlin
10. Kim MS, Uther W (2003) Automatic gait optimisation for quadruped robots. In: Australasian conference on robotics and automation, Brisbane
11. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 4598(13):671–680
12. Kohl N, Stone P (2004) Machine learning for fast quadrupedal locomotion. In: The nineteenth national conference on artificial intelligence
13. Laumanns M, Laumanns N (2005) Evolutionary multiobjective design in automotive development. Appl Intell 23(1):5570
14. Merkle D, Middendorf M (2003) Ant colony optimization with global pheromone evaluation for scheduling a single machine. Appl Intell 18(1):105111
15. RoboCup website http://www.robocup.org
16. Röfer T (2005) Evolutionary gait-optimization using a fitness function based on proprioception. In: Proc RoboCup 2004: Robot world cup VIII. Lecture notes in artificial intelligence, vol 3276. Springer, Berlin, pp 310–322
17. Saggar M, D'Silva T, Kohl N, Stone P (2007) Autonomous learning of stable quadruped locomotion. In: Lecture notes in artificial intelligence. Springer, Berlin, pp 98–109
18. Sony AIBO ERS7 robot website http://www.aibo.com/
19. Team Chaos website http://www.teamchaos.es
20. Wasik Z, Saffiotti A (2002) Robust color segmentation for the RoboCup domain. In: Proc of IEEE int conf on pattern recognition (ICPR)



Boyan Bonev received a BS Degree in Computer Science in 2005 and Ph.D. in Computer Science in 2010, both of them from the University of Alicante. He is currently Research Assistant with the Department of Computer Science and Artificial Intelligence of the University of Alicante. His research interests concern pattern recognition, computer vision and mobile robotics. He has participated in the international RoboCup competition with the Spanish quadruped-league team. The topic of his Thesis is feature selection based on information theory. He is co-author of the book Information Theory in Computer Vision and Pattern Recognition.

Miguel Cazorla received a BS degree in Computer Science from the University of Alicante (Spain) in 1995 and a PhD in Computer Science from the same University in 2000. He is currently Associate Professor with the Department of Computer Science and Artificial Intelligence of the University of Alicante. He has done several postdocs stay: ACFR at University of Sydney with Eduardo Nebot, IPAB at University of Edinburgh with Robert Fisher, CMU with Sebastian Thrun and SKERI with Alan Yuille. He has published several papers on robotics and computer vision His research interest areas are computer vision and mobile robotics (mainly using vision to implement robotics tasks).



Francisco Martín Graduated in Computer Science Engineering from Universidad Rey Juan Carlos in 2002 and received the PhD degree from the same university in 2008. Currently he is teaching Robotics, Operating Systems and Networks at Universidad Rey Juan Carlos. His research interests include self-localization, and control architectures in robotics. He has participed in RoboCup since 2005 in the 4-legged league and Standar League.

Vicente Matellán got his PhD at the Technical University of Madrid (1998), worked as Assistant Professor at Carlos III University (1993-99) and Rey Juan Carlos University (1999-2008). Currently he is Associate Professor in Computer Science at the University of Leùn (Leùn, Spain). His main research interest include multi-robot systems, robotic software architectures, artificial vision, free software, and distributed systems. He has made over 100 contributions in journals, books, and conferences in these areas.