Learning in a High Dimensional Space: Fast Omnidirectional Quadrupedal Locomotion

Matthias Hebbel, Walter Nistico, and Denis Fisseler

Robotics Research Institute
Information Technology Section
Universität Dortmund
Otto-Hahn-Str.8
44227 Dortmund
forename.surname@uni-dortmund.de

Abstract. This paper presents an efficient way to learn fast omnidirectional quadrupedal walking gaits. We show that the common approaches to control the legs can be further improved by allowing more degrees of freedom in the trajectory generation for the legs. To achieve good omnidirectional movements, we suggest to use different parameters for different walk requests and interpolate between them. The approach has been implemented for the Sony Aibo and used by the GermanTeam in the Four-Legged-League in 2005. A standard learning strategy has been adopted, so that the optimization process of a parameter set can be done within one hour, without human intervention. The resulting walk achieved remarkable speeds, both in pure forward walking and in omnidirectional movements.

1 Introduction

Legged robots are advantageous over wheeled robots when the terrain, in which the robot operates, is jagged or uneven. On the other hand, to control the legs of a robot is a highly complex and challenging task because of the many degrees of freedom in moving a leg and the required properties like stability and achievable speed of the walk.

Our research is based on quadruped walking robots, namely the Sony Aibos. In robot soccer the speed and maneuverability of the robots play an important role. Being faster than the opponent gives a team an invaluable advantage because it will in general be faster at the ball and can control it first.

This paper is structured as follows. Before explaining the commonly used approaches to walk, we will briefly introduce the properties of the Sony Aibo ERS-7 robot. Then, in Sect. 2 we will suggest some improvements for the established and commonly used walking model and will subsequently propose in Sect. 3 a learning strategy which can cope with the problem to find optimal walking parameters in the resulting higher dimensional search space. Section 4 reports on the achieved results of both the extended walking model and the experiences made with the learning approach. Section 5 will finally conclude the paper.

G. Lakemeyer et al. (Eds.): RoboCup 2006, LNAI 4434, pp. 314–321, 2007.
© Springer-Verlag Berlin Heidelberg 2007

1.1 The Experiment Platform

As a robot platform for the presented research, we used the commercially available Sony Aibo ERS-7. The Sony Aibo is a quadruped robot which comes equipped with a CMOS-camera as the most important sensor. Its legs have each 3 degrees of freedom, i.e. a hip abduction, a hip flexion and a knee flexion joint. The Sony Aibo is a truly autonomous robot since all the computation can be done on the on board MIPS IV processor with 576 MHz. For wireless communication the robot is equipped with a WLAN 802.11 compliant ethernet card.

All coordinates mentioned in this text are in the robots' coordinate system and are aligned as follows: the x-axis points to the forward direction, the y-axis points to the left side of the robot and the z-axis points up.

1.2 Related Work

Since the release of the first Sony Aibo model, a lot of research has been made on the walking style of this robot. In 2001 the so called wheel model has been introduced[1], which allows omnidirectional locomotion of the robot by treating the legs as wheels. Any kind of instantaneous movement of a robot can be described by a rotation about a certain point, the so-called *instantaneous center of rotation* (ICR). For walking straight forward without a rotational component, the ICR is located at infinity. The wheel model assumes that the steps of each foot perform a tangential movement on the circle around the ICR. The speed of the step can be calculated in the same way as for a wheel of a differential drive robot.

(a) Forward (b) Diagonal (c) Rotation

Fig. 1. The rotation of the leg locus for different walk requests

Like in a trot gait, the two diagonal opposite legs are always moved at the same time, e.g., two legs are always in the air while the other two legs remain in contact with the ground. Due to the duty factor of 0.5 of this walk, it is only dynamically stable, but it has turned out to be stable enough, even in a robot soccer match with a lot of pushing of other robots.

Further, the wheel model assumes, that the feet of the robot move on a certain locus. The according joint angles needed to control the feet on these loci are calculated by means of inverse kinematics. Figure 1 shows the rotation of these loci about the z-axis for different walk requests. The locus for the feet to move on, is described by several parameters[2,3,4]. In 2004 the team from UT Austin Villa used 17 parameters [5], the German Team 14 parameters [4] and the team from the University of Pennsylvania used 19 parameters[6] to define the walk.

Several approaches have been made to achieve faster walks with different loci, the most commonly used ones are rectangular, half-elliptical and trapezoidal loci. To optimize the parameters which describe the form of the loci, many learning approaches have been used[7,8].

These approaches have in common that only a single parameter set has been used for walking, and such set has only been tuned for fast forward walking. But a parameter set for fast forward walking is not automatically useful for fast backwards or sidewards walking.

2 Enhancements to the Walking Model

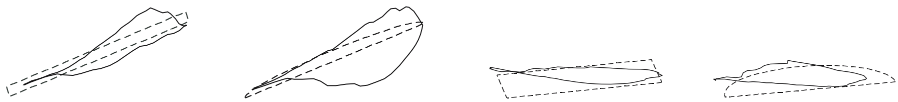
All known walking engines for the Sony Aibo make only use of static inverse kinematics. This means that the calculation of the desired joint angles to reach a specified position with the foot is only based on geometry and does not take into account physical properties like friction, moments of inertia or forces. While in industrial robotics very complex dynamic models are considered to calculate a trajectory[9], in mobile legged robotics this is not feasible due to the lack of computational power and adequate dynamic models.

2.1 Controlled and Real Walking Trajectories

The dashed paths in Fig. 2 show the controlled loci in the xz-plane for forward walking for the fore and hind legs; the loci in Fig. 2(a) and Fig. 2(b) are controlled on a rectangular path while the loci in Fig. 2(c) and 2(d) are controlled on a half-elliptical path. All other parameters like timing, step lengths etc. do not differ between the rectangular and half-elliptical loci of the same foot. The according real trajectories for the controlled trajectories are presented as solid curves in the Figures 2(a) – 2(d). Especially the locus of the front feet differs remarkably between the rectangular (Fig. 2(a)) and the half-elliptical control (Fig. 2(c)). A reason for this might be a slightly different angle of the robots' body while walking or the slipping of the feet on the ground. After having a look on the real trajectory of the feet in the yz-plane, we found out that especially the paths of the fast walks are bent while we would expect them to be a straight line.

2.2 Parameters Defining the Gait

Due to the observations on the real loci and the fact that especially fast walks differed the most from the controlled loci, we reasoned that more flexibility in the control of the feet could give better results in terms of faster walks. We decided to introduce three dimensional polygons instead of the common "flat" two dimensional shapes.



(a) Fore leg rectangular

(b) Fore leg half-elliptical

(c) Hind leg rectangular

(d) Hind leg half-elliptical

Fig. 2. The controlled locus (dashed) and real locus (solid) of the fore and hind legs for different trajectories

As mentioned in Sect. 1.2, the most common walking approaches use comparatively few parameters to describe the walk; this because hand tuning of walks is not feasible and the time to optimize the parameters with learning approaches also profits from a smaller search space due to the long time to evaluate the resulting speed of a parameter set.

However, we decided to use 3 dimensional polygons with n vertices $P_1, \ldots, P_n$. Additionally n timing parameters are needed to specify the amount of time needed for the foot to travel from vertex P_i to P_{i+1}, for $1 \leq i < n$, and from P_n to P_1. Further we restricted the walking gait to a trot gait, i.e. two diagonal opposite legs are at the same time in the air, while the other pair remains on the ground. The amount of overall parameters for a single leg is $3n + n$. Therefore, the number of parameters describing the walk for a 4 legged robot results to $16n$. With the reasonable constraint, that the pair of fore and the pair of hind legs are moving on mirror-symmetrical loci (but countercyclical in time), the number of parameters is reduced to $8n$.

To be able to find an optimum in the large search space, both a determined learning strategy and a fast and reliable measurement of the robots' speed is advantageous.

2.3 Different Parameter Sets for an Omnidirectional Walk

The ability to reach a desired point in any orientation is called omnidirectionality. In robot soccer, this ability gives the robots the advantage to be able to move quickly in any direction without having to rotate in advance, also it makes the control of the robot on the field much easier. The wheel model described in Sect. 1.2 allows omnidirectional movement by rotating the foot trajectories about the z-axis and scaling the legs' speed according to the radius of the circle around the ICR. Unfortunately a parameter set resulting in a fast forward speed (for example) is not necessarily useful for walking sideways or rotating. The fastest walk which was published on the Aibo ERS-210 in [8] was only useful for straight forward walking with only very small rotational components. Due to this fact the GermanTeam has used this parameter set in 2004 only for straight sprints to the ball[4], while in the "normal" game play they switched back to an omnidirectional parameter set.

Since "hard" switching of parameters while walking normally causes stumbling of the robot which can make him fall over or rotate unintentionally, we decided to implement a "soft" interpolation between parameter sets. In this case, we can use optimized parameters for certain walk requests without having the negative effects of stumbling or unwanted directional changes when changing the walk request. A detailed description of the interpolation is given in [10].

3 Parameter Optimization

As described in Sect. 2.2, the gait depends on a set of parameters $\mathbf{x} := (x_1, x_2, \ldots)$ with $x_i \in \mathbb{R}$. The parameters consist of 3 coordinates per vertex for the n vertices of the polygon and the n timing parameters per leg. The major goal for each parameter set is to reach the highest possible robot speed in the desired direction, i.e. to optimize the speed $\mathbf{v}$ of the robot with respect to the parameters $\mathbf{x} := (x_1, x_2, \ldots)$. Since the speed depends on the parameter set, the maximum speed $\mathbf{v}_k$ is a function of the parameter set $\mathbf{x}_k$, i.e. $\mathbf{v}_k := F(\mathbf{x}_k)$.

3.1 The Learning Strategy

For this challenging optimization process, we used the biologically inspired state-of-the-art $(\mu/\rho \overset{+}{,} \lambda)$ evolution strategy with self-adaption[11]. The evolution strategy operates on populations of individuals $\mathbf{a}$ and is based on the paradigm *survival of the fittest*. A parent population $\mathcal{P}_p^{(t)}$ is creating an offspring population $\mathcal{P}_o^{(t)}$ by making use of the operators *replication*, *mutation* and *recombination*. Due to the *mutation* and *recombination* operators the offspring individuals "differ" from their parents, e.g. they have different properties. The *selection* then decides which individuals will form the new parent generation $\mathcal{P}_p^{(t+1)}$, all other individuals will die out. When the *selection* operation is based on the mentioned paradigm *survival of the fittest*, in the course of the evolution process the properties of the parent generation will be optimized with respect to the *fitness* criteria of the selection operator. For more information about the chosen strategy please refer to [10].

3.2 Learning to Walk

As explained in Sect. 2.2, one requirement for the evolution in this high dimensional search space, is to be able to measure the speed of the robot quickly and precisely without constraining the robots' walk, like e.g. in [8]. For this reason, we developed a ceiling camera system which is mounted above the robot soccer field. The camera is attached to a server which is processing the camera images with a frame rate of 25 Hz. After detecting the robot in the image, the server broadcasts the position of the detected robot into the wireless network.

To let the walk evolution be as autonomous as possible, we developed a behavior which lets the robot walk on the field, always in the observation range of the ceiling cam. To determine the fitness F_k of an individual $\mathbf{a}_k$, we let the robot walk with the appropriate parameter set $\mathbf{x}_k$ and measure the speed. To keep the measurement error small, we allow 2 seconds walking, before starting to measure the speed to be sure, not to take acceleration effects into account. After this "warm up" phase, we take the starting position and after another 2 seconds the achieved position of the robot. By dividing the difference between the two taken points by 2, we get the average speed of the last 2 seconds' walk.

4 Results

4.1 Adaption of the Evolution Strategy

The evolution strategy described in Sect. 3.1 offers a lot of parameters to influence its behavior. In our first learning approaches, we used 8 vertices for each polygon and tried different values for the population sizes. In none of the tests, the speed started to converge and the fastest walk found for straight forward walking achieved only 33 cm/s which was still slower than most of the RoboCup teams were walking on the world championship in 2004.

In the next approach, we decided to reduce the search space by allowing only 4 vertices per polygon, i.e. reducing the search space to 32 dimensions. With this configuration, after a few generations a consistent speed improvement and a convergence of the speed was observable. As long as the population size μ and λ was big enough, the size did not have a big effect on the convergence speed; we chose $\mu = 6$ and $\lambda = 24$. This population size was big enough to explore the search space, so that the chance to get stuck in a local optimum was minimized. With these settings the speed converged already after 30 to 40 generations. The curve in Fig. 3 shows the run of the fitness of all offspring individuals during the evolution process to optimize the walking parameters for forward walking. The individuals in the first parent generation were all equal and hand generated. They all resulted in a forward speed of 280 mm/s. The fastest walk has been found after only 29 generations after less than one hour of training.

4.2 Achieved Speeds

The found parameter sets for all walking directions result in a faster speed than the walks presented on the RoboCup 2004. The maximum reached speeds are shown in Fig. 4.

A special parameter set has been found during the evolution, where we wanted to see, if the walking of the Aibo on the "elbows", like all of the RoboCup teams do, is really the most beneficial walk. For this experiment, we created parameter sets for the initial population, which let the Aibo walk with stretched legs, like a real dog. The resulting walk with these very uncommon starting parameters achieved a speed of 510 mm/s, which is certainly by far the fastest forward walk ever found on a Sony Aibo. But besides its fast speed, the walk had some unfortunate properties like unstableness and a lot of vibrations. These vibrations during the walk result in blurred camera images, thus we only used this walk when we wanted to sprint over a longer distance. The unwanted accelerations of the robot body which result in a shaking of the camera while walking of the 510 mm/s walk, our so called boost, are shown in Fig. 5 in comparison with the very few unwanted accelerations of the normal walk.

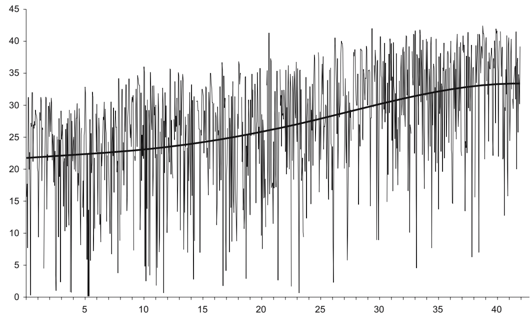
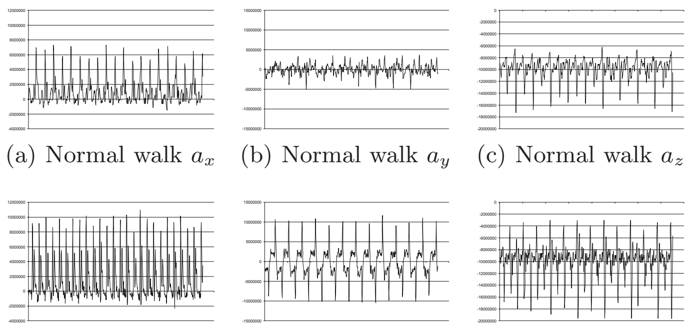


Fig. 3. The fitness of all offspring individuals during an evolution run

forward	backwards	sideways	diagonal	rotation
451 (510)	405	344	421	200

Fig. 4. Achieved speeds in mm/s, respectively $^\circ/s$ for different walk requests



(a) Normal walk a_x (b) Normal walk a_y (c) Normal walk a_z

(d) Boost walk a_x (e) Boost walk a_y (f) Boost walk a_z

Fig. 5. Accelerations during walking straight forward for a normal walk and the "boost" walk

5 Conclusion

In this paper, we presented meaningful enhancements to existing and commonly used walking models for the Sony Aibo. Due to the fact that the enhanced model has more degrees of freedom to define a walk, a learning approach with external measurement of the fitness has been suggested. The learned walks with the described approach were in all directions more than 25% faster than existing walks, e.g. the walk of the German Team from 2004. The here described walk was also one of the reasons why the GermanTeam has won the RoboCup and was one of the fastest teams on the RoboCup championship 2005 in Japan.

Acknowledgment

The authors would like to thank the members of the GermanTeam for the fruitful cooperation and Microsoft MSDNAA for their financial support which made it possible for our team to participate with several team members in the RoboCup world championship in 2005 in Japan.

References

1. Hengst, B., Ibbotson, D., Pham, S.B., Sammut, C.: Omnidirectional locomotion for quadruped robots. In: Birk, A., Coradeschi, S., Tadokoro, S. (eds.) RoboCup 2001. LNCS (LNAI), vol. 2377, pp. 368–373. Springer, Heidelberg (2002)
2. Düffert, U., Hoffmann, J.: Reliable and precise gait modeling for a quadruped robot. In: Bredenfeld, A., Jacoff, A., Noda, I., Takahashi, Y. (eds.) RoboCup 2005. LNCS (LNAI), vol. 4020, Springer, Heidelberg (2006)
3. Kohl, N., Stone, P.: Policy gradient reinforcement learning for fast quadrupedal locomotion. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), IEEE Computer Society Press, Los Alamitos (2004)
4. Röfer, T., Laue, T., Burkhard, H., Hoffmann, J., Jüngel, M., Göhring, D., Lötzsch, M., Spranger, M., Altmeyer, B., Goetzke, V., von Stryk, O., Brunn, R., Dassler, M., Kunz, M., Risler, M., Stelzer, M., Thomas, D., Uhrig, S., Schwiegelshohn, U., Dahm, I., Hebbel, M., Nistico, W., Schumann, C., Wachter, M.: German Team Report 2004. Technical report, HU Berlin, TU Bremen, TU Darmstadt and University of Dortmund (2004)
5. Stone, P., Dresner, K., Fidelman, P., Jong, N.K., Kohl, N., Kuhlmann, G., Sridharan, M., Stronger, D.: The UT Austin Villa 2004 RoboCup four-legged team: Coming of age. Technical Report UT-AI-TR-04-313, The University of Texas at Austin, Department of Computer Sciences, AI Laboratory (2004)
6. Cohen, D., Ooi, Y.H., Vernaza, P., Lee, D.D.: Robocup 2004 Legged Soccer Team. Technical report, University of Pennsylvania (2004)
7. Kohl, N., Stone, P.: Machine learning for fast quadrupedal locomotion. In: The Nineteenth National Conference on Artificial Intelligence, pp. 611–616 (2004)
8. Röfer, T.: Evolutionary gait-optimization using a fitness function based on proprioception. In: Nardi, D., Riedmiller, M., Sammut, C., Santos-Victor, J. (eds.) RoboCup 2004. LNCS (LNAI), vol. 3276, Springer, Heidelberg (2005)
9. Craig, J.J.: Introduction to Robotics, Mechanics and Control. ch. 6, 7 Addison Wesley, Reading (1986)
10. Dahm, I., Fisseler, D., Hebbel, M., Nistico, W.: Learning fast walking patterns with reliable odometry information for four-legged robots. In: ISMCR'05 (2005)
11. Beyer, H.G., Schwefel, H.P.: Evolution strategies – A comprehensive introduction. Natural Computing 1(1), 3–52 (2002)