GENERATING HIGH-SPEED THREE-DIMENSIONAL DYNAMIC QUADRUPED WALKING USING AN EVOLUTIONARY SEARCH

Di He, Qining Wang, Chunxia Rong and Guangming Xie
Intelligent Control Laboratory, College of Engineering, Peking University, Beijing 100871, China

Keywords: Quadruped walking, Three-dimensional gaits, Evolutionary search, Legged robots, Locomotion.

Abstract: This paper presents an evolutionary computation approach to generate three-dimensional fast forward gaits for quadruped robots with three motor-driven joints on each limb. We use linear constraints to reduce the high-dimensional space of parameters in order to generate the speed effectively. Real robot experiments show that the evolutionary approach is effective in developing quadruped gaits. Satisfactory results are obtained in about an hour by the autonomous learning process, which starts with a set of hand-tuned parameters.

1 INTRODUCTION

Over the past years, plenty of publications have been presented in the biomechanics literature which explained and compared the dynamics of different high-speed gaits including gallop, canter, bound, and fast trot (e.g. (Alexander and Jayes, 1983), (Alexander et al., 1980)). To study and implement legged locomotion, various robot systems that achieve high-speed and efficient walking gaits have been created (e.g. (Holmes et al., 2006), (Raibert, 1986), (Collins et al., 2005)). Since the walking speed of the robots is one of the most important factors in determining the success of a team in the RoboCup Four-Legged competitions with the standard platform using Sony AIBO ERS-7 robots, there has been significant incentive in the RoboCup community to develop gaits with better performance on the speed (e.g. (Hornby et al., 1999), (Röfer et al., 2004), (Quinlan et al., 2003)).

Theoretically, high-speed gait design can be achieved by dynamics analysis, if it can be simplified into dynamics models. However, quadruped robots' dynamics state are complex with the ground constraints. Especially when dealing with the whole robot, the problem becomes nonlinear and high-dimensional. Therefore, current methods that dealt with gait optimization often resort to Inverse Kinematics Model, which convert the optimizing problem into a gait locus design (e.g. (Quinlan et al., 2003), (Röfer et al., 2004), (Röfer et al., 2005), (Rong et al., 2009)). In previous work, researchers have designed several locus shapes, e.g. rectangular, elliptical, trapezoidal and three-dimensional polygon to describe the walking patterns for AIBO robots. With different machine learning algorithms, the gait locus can be improved. For example, the sharP-KUngfu Team proposed an Adaptive Particle Swarm Optimization (APSO) based approach to generate fast two-dimensional gaits and reaches a speed of $425mm/s$ (Rong et al., 2009). The team from the University Newcastle generated fast gaits which is $420mm/s$ using Genetic Algorithm (GA) and plane loci of arbitrary shape (Quinlan et al., 2003). However, these methods only deal with the generation of two-dimensional quadruped walking. The three-dimensional walking gaits are more close to real walking patterns of the quadruped animals. Only a few studies have been made in the generation of three-dimensional walking gaits. For example, the German Team created a flexible gait implementation that controls the feet on a path described by a three-dimensional polygon and get a speed of $451mm/s$ finally (Röfer et al., 2005). The results indicated that gait optimization with a three-dimensional polygon can obtain stable gaits with higher walking speed. However, as the complexity of the polygon increases, the learning procedure is time consuming and may damage the motor-driven joints of the robots during the optimization process of the high dimensional parameters.

As a matter of fact, the performance of optimization is not the same by using different gait locus shapes or machine learning algorithms. Generally, optimizing with a model of more degrees of freedom

He D., Wang Q., Rong C. and Xie G. (2009).
GENERATING HIGH-SPEED THREE-DIMENSIONAL DYNAMIC QUADRUPED WALKING USING AN EVOLUTIONARY SEARCH.
In *Proceedings of the 6th International Conference on Informatics in Control, Automation and Robotics - Intelligent Control Systems and Optimization*, pages 167-172
DOI: 10.5220/0002184301670172

is likely to generate a better gait, but it takes more time. And an algorithm which has a better global search capability always converges slowly. In this paper, we use three-dimensional polygon as the gait model for the locus shape. To solve the problem of slow convergence of the optimization using three-dimensional polygon, we propose a method based on linear constraints to reduce the degrees of freedom of the model. The whole learning process is running automatically by the robot with onboard processor. In real robot experiments, we achieved an effective gait which speed is higher than the previous known gaits, using AIBO as the test platform.

The remainder of this paper is organized as follows. Section II introduces the parameterization and kinematics for the Sony AIBO ERS-7 robot platform, and the general methods for quadruped gait planning. Section III presents our evolutionary search based on linear constraints. Section IV specifies the evaluation of gait optimization based on the proposed evolution and others. Section V presents the conclusion and discussion.

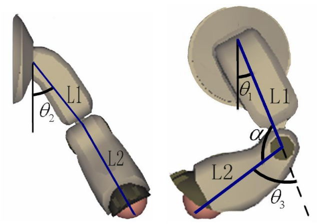
2 DYNAMIC MODEL FOR AIBO ROBOT

2.1 Inverse Kinematics Model

The high-level parameters that we adopt to represent the gait need to be transferred to joint angles of legs before they can be implemented by the robot. An inverse kinematics model can be used to solve this problem. For a linked structure with several straight parts connecting with each other, the position of the end of this structure relative to the starting point can be decided by all angles of linked parts and only one position results from the same angle values. The definition of the kinematics model is the process of calculating the position of the end of a linked structure when given the angles and length of all linked parts.

For the AIBO robots, Given the position of the end of the structure, inverse kinematics calculates out what angles the joints need to be in to reach that end point. In this study, the inverse kinematics is used to calculate necessary joint angles to reach the paw position determined by gait parameters. Fig. 1 shows the inverse kinematics model and the coordinates for AIBO.

The shoulder or hip joint is the origin of the coordinate system. l_1 is the length of the upper limb, while l_2 is the length of the lower limb. Paw position is represented by point (x, y, z). The figures and equations here only give the view and algorithm to get the solution for left fore leg of robot. According to the symmetrical characteristic of legs, all other legs can use the same equations with some signs changing.



(a) (b)

Figure 1: The inverse kinematics model and coordinates for Aibo. (a) is the front view of left fore leg. (b) is the side view of left fore leg.

The following equations shows the inverse kinematics model:

$$\begin{aligned} x &= l_2\cos\theta_1\sin\theta_3 + l_2\sin\theta_1\cos\theta_2\cos\theta_3 + l_1\sin\theta_1\cos\theta_2 \\ y &= l_1\sin\theta_2 + l_2\sin\theta_2\cos\theta_3 \qquad (1) \\ z &= l_2\sin\theta_1\sin\theta_3 - l_2\cos\theta_1\cos\theta_2\cos\theta_3 - l_1\cos\theta_1\cos\theta_2 \end{aligned}$$

The inverse kinematics equation to get θ_1, θ_2, θ_3 by the already known paw position (x, y, z) is as follows:

$$\begin{aligned} \theta_3 &= \cos^{-1}\frac{x^2+y^2+z^2-l_1^2-l_2^2}{2l_1l_2} \\ \theta_2 &= \sin^{-1}\frac{y}{l_2\cos\theta_3+l_1} \\ \theta_1 &= -\tan^{-1}\frac{a}{b} \pm \cos-1\frac{x}{a^2+b^2} \qquad (2) \end{aligned}$$

where $a = l_2\sin\theta_3$, $b = -l_2\cos\theta_2\cos\theta_3 - l_1\cos\theta_2$.

One problem of the inverse kinematics is that it always has more than one solution for the same end point position. However, as to AIBO, only one solution is feasible due to the restriction on the joint structure. As a result, when using inverse kinematics to calculate joint angles, it is necessary to take joint structure limitation into consideration to get the right solution. Otherwise, it will possibly cause some physical damage to the robot platform.

2.2 Control Parameters

Before we run the learning gait procedure, the control parameters representing a gait need to be decided. For the stance parameters, as shown in Section 2.1, point (x, y, z) can be translated into $(\theta_1, \theta_2, \theta_3)$. Therefore, we use coordinates of the paw positions relative to the shoulders to describe the posture, which is called *FootZeroPosition*. As to locus, we choose an arbitrary three-dimensional polygon with n vertices (P_1, P_2, ..., P_n) as the shape. Additionally, n timing parameters which we call *p_length*, are needed to specify the amount of time needed for the foot to travel

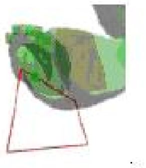


Figure 2: The end of a foot travels as a 3-dimensional polygon.

from vertex P_i to P_{i+1}, where $1 < i < n$, and from P_n to P_1, the total time needs for one cycle is the step length, which we assume here is an integral number, in 0.008 second units. In this way, a total number of parameters for a single leg of $4n+4$. And we suppose the two diagonally feet are traveling at the same time while the other two delay for half a period. Thus, the total number of parameters for the AIBO robot during the proposed gait optimization process is $8n+8$.

2.3 Evolution of the Parameters

We can calculate the walking posture from point to point and moment to moment by the model above. However, because of the constraint of the ground, the end of the foot in fact will not travel exactly as the polygons we design, the relation between gait parameters and speed is impossible to acquire, and there is no sufficiently accurate simulator for AIBO due to the dynamics complexity. As a result, we have to perform the learning procedure on real robots.

In order to automatically acquiring speed for each parameter set, the robot has to be able to localize itself. Since the low resolution of AIBO's camera and limited processing ability, it is faster and more accurate to detect black-white edge than other things. Thus we use a white board with parallel black bars on the field for AIBO to localize (see Fig. 3).

During evaluation procedure, the robot walks to a fixed initial position relative to the board, then loads the parameter set needed to be evaluated, walks for a fixed time, e.g. $6s$, stops and calculates the forward speed. After that, the robot walked back and started another trial. The total time for testing one set of parameter is usually less than $15s$.

Figure 3: The white board with parallel black bars for body adjusting of the AIBO robots.

3 GAIT OPTIMIZATION WITH LINEAR CONSTRAINTS

Since the paw of one foot travels a smooth curve in space, the more complex the polygon that we use to describe the locus shape, the more close the walking gait to that of the real robot. However, in practice, the optimization with a higher dimensional model is more easily to damage the robot in the gait evolution and is time-consuming. Here, we propose a new strategy which reduces the degree of freedom of the model by linear constraints.

Assume that the paw of one foot travels as a three-dimensional polygon with 2n-vertices Q_1, Q_2, … Q_{2n}. The paw travels from Q_i to Q_{i+1} while $i = 1, 2, \ldots, 2n-1$ and Q_{2n} to Q_1. Let t_1, t_2, …, t_{2n} be the run-through-time for the edges relative to the time for the whole polygon, which is under the constraint:

$$\sum_{i=1}^{2n} t_i = 1 \tag{3}$$

Given the polygons for the front and rear legs respectively, it produces a $16n$ dimensional space of the locus for the whole robot. Then the linear constraint is as follows:

$$\begin{cases} Q_{2k+1} = (Q_{2k} + Q_{2k+2})/2, & k = 1, 2, \ldots, n-1 \\ Q_1 = (Q_2 + Q_{2n})/2, & \\ t_{2k-1} = t_{2k}, & k = 1, 2, \ldots, n. \end{cases} \tag{4}$$

Under the linear constraint, the dimension of locus reduces by half. Furthermore, since the vertex Q_{2k+1} is the midpoint of Q_{2k} and Q_{2k+2}, a $2n$ vertex polygon changes into an n vertex polygon in fact. It means the gait optimization can begin with a simple locus shape first, such as a three-dimensional polygon with 4-vertices. When the speed approaches to a certain level, change the locus shape into a polygon with 8-vertices by linear interpolation and continue the procedure.

The optimization process can be divided into two parts. In the first part, the optimization process focuses on global search by using linear constraints. When approaches the optimal solution, the optimization process moves to the second part, where the linear constraints will be removed and the local search

Table 1: Control parameters in gait evolution.

Parameter	Description
foot zero position	relative position of paws and shoulders
points of the polygon	relative position of the locus and Foot Zero Position
p-length	run-through-time for the edge relative to the time for the whole polygon
step length	time for one complete step in 0.008 second units

will be emphasized. We can apply different optimization approaches, e.g. APSO, GA, to different parts of the optimization process.

4 EXPERIMENTAL RESULTS

Using the method described above, we carried out two separate experiments and evaluate the walking results. In the first experiment, since different machine learning algorithms perform differently during optimizations, we evaluate the performance of using GA or APSO. Similar to (Rong et al., 2009), the inertial weight of APSO is determined by the equation:

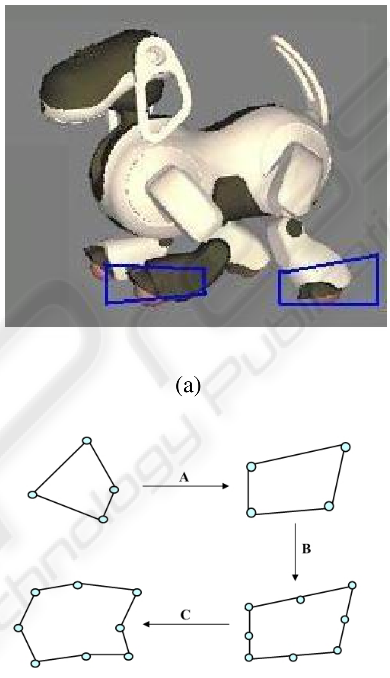
$$\begin{aligned} w &= 1.2 - 0.02 \times k,\ (k \leq 10) \\ w &= 1 - 0.085 \times (k-10),\ (10 < k \leq 20) \\ w &= 0.15 - 0.03 \times (k-20),\ (20 < k \leq 25) \\ w &= 0,\ (k > 25) \end{aligned} \tag{5}$$

where k is the iteration. Three-dimensional polygon with 4-vertices (40-dimensional search space) is chose as the locus shape for the low-dimensional search section. We begin the optimization with a hand-tuned gait which is $250mm/s$ as initial state. After the speed rising to about $400mm/s$, we change the optimized locus into an 8-vertices polygon by linear interpolation and continue the evolution. The process is shown in Fig. 4.

For both of the optimization algorithms, the number of individuals for one population is chosen to be 10. And in the genetic algorithm we use five better individuals to generate the other five by crossover operator and mutation operator.

Fig. 5 shows the best result of each generation in different strategies. We can see that the strategy of using APSO first and optimizing with GA after linear interpolation achieved better result than the other three. Moreover, We can note that there are both advantages and disadvantages comparing these experiments with different algorithms. Fig. 5 indicate that APSO can find the optimal solution area during the early period of the global search.

The optimized speed of the $15th$ generation reaches about $400mm/s$. However, the ability of local search is weak and the convergence is relatively slow. Contrarily, GA grows slowly during the early period of global search, the optimized speed reaches about $400mm/s$ until the optimization process runs to the $25th$ generation. However, if given a relatively optimal solution, GA can find the best result within a short time period. Thus, we select the strategy which integrates APSO and GA to inherit the advantages of them.



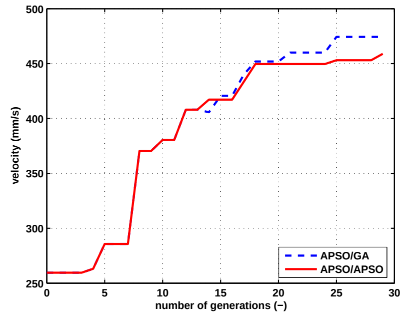
(a)

(b)

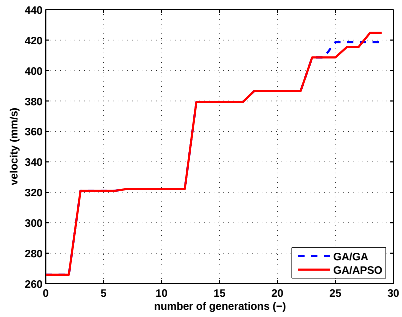
Figure 4: (a) shows the initial locus for the robot, (b) shows the optimization process: in procedure A, we optimize the locus shape of 8-vertices polygon with linear constraint, which is equal to optimize a gait of 4-vertices polygon. In procedure B, we change the locus shape by linear interpolation. In procedure C, we optimize with the high dimensional locus shape of 8-vertices polygon.

In the second experiment, we compare our method with the existing methods (see Fig. 6).

It indicates that during learning, GA performs slowly with speed result $403mm/s$ after 30 generations. APSO grows fast in the first 13 generations, but grows slow then and reaches a speed of $435mm/s$



(a)



(b)

Figure 5: The velocity of the best gait from each generation by using different strategies. e.g. GA/APSO (the red line in (b)) means using GA to optimize the 4-vertices polygon and optimize the 8-vertices polygon with APSO after the linear interpolation.

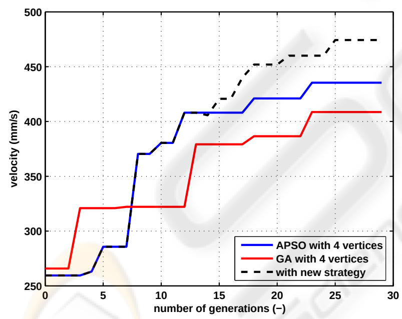
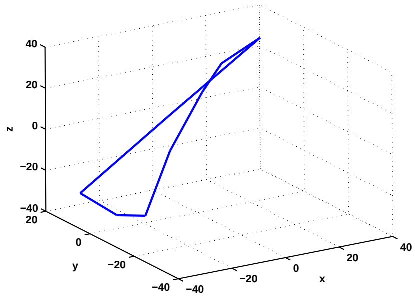


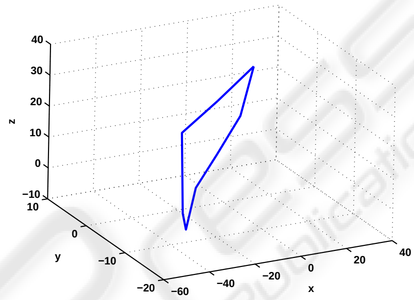
Figure 6: The velocity of the best gait from each generation in real robot experiments.

at last. Our new search strategy achieves better result than the other two, and the speed reaches to $478mm/s$. It suggests that more dimensions of the search space can get a better gait than the less one. And our strategy avoids the slow learning process. Fig. 7 and Fig. 8 show the locus of the paws and the walking performance of the real robot respectively.

The final speed of the optimized quadruped gait is faster the the existing results using AIBO platform.



(a)



(b)

Figure 7: locus.

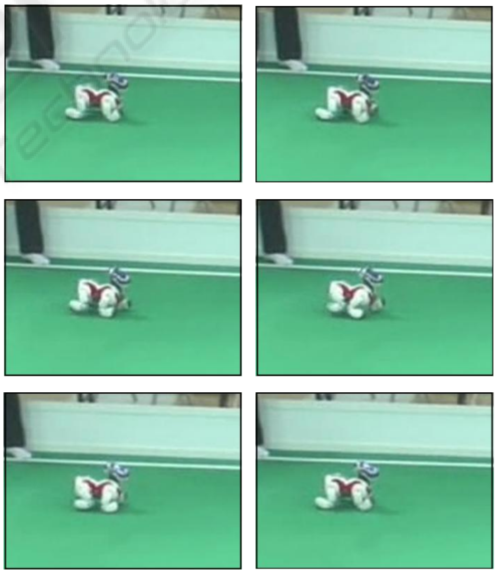


Figure 8: A sequence of photos captured during autonomous walking of the AIBO robot on carpet.

5 CONCLUSIONS

In this paper, we have demonstrated a novel evolutionary computation approach to generate three-dimensional quadruped fast forward gaits using the AIBO robot platform. Our method was easily coded and computationally inexpensive. Moreover, by linear constraints, the evolution converged extremely fast and the training time was largely reduced. It is an essential advantage for physical robot learning, minimizing possible damage to the robot. It reduced the human work as well as generating evolutional results varied a lot in different experiences. Through experiments which took about 60 minutes each, we achieved several high performance sets of gait parameters which differ a lot from each other. The proposed method generated a speed of $478mm/s$ which is faster than the previous known gaits.

One of the useful aspects of the proposed method is that the high-dimensional parameter set is optimized in an effective way. In order to reduce the dimension, we approach a linear constraint for the parameters. Under this constraint, the locus can be optimized fast. Another contribution of our method is the combination of the two algorithms, GA and APSO. The result shows that the strategy has better global searching capability and local searching capability than using each algorithm only.

In the future, we will compare different high-performance gait parameters and analyze the dynamics model of the robot to obtain further understanding of the relation between parameter and its performance. In this study, we find that the gait actually executed by the robot differs significantly from the one that we design. There are several possible reasons. The most important one is the interaction with environment prevents the implement of some strokes of robot legs. Although with learning approach, factors that cause the difference between actual gait and planned gait do not have to be taken into consideration. However, we assume that if the planned gait and actual gait can conform to each other, AIBO will walk more stable with high speed. In order to solve the problem, the analysis of dynamics between the robot and the environment is necessary. In the gait learning procedure of current study, we only evolve fast forward gait and choose forward speed as the fitness. Later on, we will try to learn effective gaits in other directions, for example, gaits for walking backward, sideward and turning. We also consider exploring optimal omni-directional gaits. With gaits working well at all directions, robots will be able to perform more flexibly and reliably.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the contribution of the team members of the sharP-KUngfu Robot Team. Part of the source code and videos can be found on the web page http://www.mech.pku.edu.cn/robot/fourleg/

REFERENCES

R. M. Alexander, A. S. Jayes, A dynamic similarity hypothesis for the gaits of quadrupedal mammals, *J. Zoology*, vol. 201, pp. 135-152, 1983.

R. M. Alexander, A. S. Jayes, R. F. Ker, Estimates of energy cost for quadrupedal running gaits, *J. Zoology*, vol. 190, pp. 155-192, 1980.

S. Collins, A. Ruina, R. Tedrake, M. Wisse, Efficient bipedal robots based on passive dynamic walkers, *Science*, vol. 307, 2005, pp. 1082-1085.

P. Holmes, R. J. Full, D. Koditschek, J. Guckenheimer, The dynamics of legged locomotion: Models, analyses, and challenges, *SIAM Review*, vol. 48, no. 2, 2006, pp. 207-304.

G. S. Hornby, M. Fujita, S. Takamura, T. Yamamoto, O. Hanagata, Autonomous evolution of gaits with the Sony quadruped robot, *Proceedings of the Genetic and Evolutionary Computation Conference*, vol. 2, 1999, pp. 1297-1304.

M. J. Quinlan, S. K. Chalup, R. H. Middleton. Techniques for improving vision and locomotion on the Sony AIBO robot. *Proceedings of the 2003 Australasian Conference on Robotics and Automation*, 2003.

M. J. Quinlan, Nubots team robocup 2005. Technical report, 2005.

M. H. Raibert, *Legged robots that balance*, MIT Press, Cambridge, 1986.

C. Reynolds, Flocks, herds, and schools: A distributed behavioral model, *Comp. Graph.*, vol. 21, no. 4, pp. 25-34, 1987.

C. Rong, Q. Wang, Y. Huang, G. Xie, L. Wang, "Autonomous evolution of high speed quadruped gaits using particle swarm optimization", *Lecture Notes in Artificial Intelligence*, 5399, Springer-Verlag Berlin Heidelberg, 2009.

T. Röfer, H. D. Burkhard, U. Düffert, J. Hoffmann, D. Göhring, M. Jüngel, M. Lötzsch, O. v. Stryk, R. Brunn, M. Kallnik, M. Kunz, S. Petters, M. Risler, M. Stelzer, I. Dahm, M. Wachter, K. Engel, A.Osterhues, C. Schumann, J. Ziegler, GermanTeam RoboCup 2004. Technical report, 2004.

T. Röfer, H. D. Burkhard, U. Düffert, J. Hoffmann, D. Göhring, M. Jüngel, M. Lötzsch, O. v. Stryk, R. Brunn, M. Kallnik, M. Kunz, S. Petters, M. Risler, M. Stelzer, I. Dahm, M. Wachter, K. Engel, A. Osterhues, C. Schumann, J. Ziegler, GermanTeam RoboCup 2005. Technical report, 2005.