

LETTER

A Comparative Study among Three Automatic Gait Generation Methods for Quadruped Robots*

Kisung SEO^{†a)}, Member and Soohwan HYUN^{††}, Nonmember

SUMMARY This paper introduces a comparison of three automatic gait generation methods for quadruped robots: GA (Genetic Algorithm), GP (genetic programming) and CPG (Central Pattern Generator). It aims to provide a useful guideline for the selection of gait generation methods. GA-based approaches seek to optimize paw locus in Cartesian space. GP-based techniques generate joint trajectories using regression polynomials. The CPGs are neural circuits that generate oscillatory output from an input coming from the brain. Optimizations for the three proposed methods are executed and analyzed using a Webots simulation of the quadruped robot built by Bioloid. The experimental comparisons and analyses provided herein will be an informative guidance for research of gait generation method.

key words: robot automatic gait generation, quadruped robot, genetic algorithm, paw trajectory, joint trajectory, genetic programming, central pattern generator

1. Introduction

Planning gaits for quadruped robots is a challenging task, because there are many degrees of freedom and, thus many parameters need to be set properly [1], [2]. Providing good locomotion capabilities for robots is very significant for allowing them to carry out useful tasks in a variety of environments [1]–[3]. The automatic generation of gaits is especially important for walking robots because different environments and newly developed robots make it important to generate a variety of gaits in a short period of time [2]. Gaits may be optimized for different properties, including fast velocity and/or high stability, and for specific requirements such as highest or lowest posture, and for various “personality traits.”

Existing automatic generation methods for quadruped gaits include: GA (Genetic Algorithm) based approaches [4]–[6], GP (Genetic Programming) based approach [9], and CPG (Central Pattern Generator) based approaches [11], [12].

However, it is difficult to rate those methods for a given robot, because published studies mainly focus on specific robots and operation environments, specific problems to be solved, and specific algorithms to be improved. In other

words, there is no published research comparing these approaches in the same environments for the same robots.

Therefore, experimental comparisons and analysis for automatic gait generation methods will be a useful guideline for selection of gait generation method.

In this paper, the three representative gait generation methods named above are compared for a quadruped robot. In order to compare the three methods in as fair and unbiased a fashion as possible, the same experimental environments and performance indexes are selected for all. To investigate the characteristics of the three approaches, implementations and experiments on GA, GP and CPG based gait evolution are executed for the Bioloid quadruped robot in the Webots environment. Various performance indexes and important features are summarized including velocity, stability, gait space, and necessity of inverse kinematics. Section 2 describes GA based evolution of gait in the Cartesian space, and Sect. 3 explains GP based evolution of gait in the joint trajectory framework. Section 4 describes CPG based gait generation. Section 5 presents experimental results for evolved gaits, and Sect. 6 concludes the paper.

2. GA Based Gait Generation in Cartesian Space

We use an evolutionary approach based on genetic algorithms to optimizing the locus of the robot’s paw for gait generation in Cartesian space intuitively. To evolve the locus of paw positions for a quadruped robot, in this paper, the shape of the locus is represented by a third-order spline to obtain a flexible shape – not limited rectangular, trapezoid, and arc which are used in widely.

The locus could be obtained by third-order spline interpolation between two points—the end point of the lifting paw and the starting contact point of the lowering paw. The foot locus is shown in an enlarged window in Fig. 1. Let T

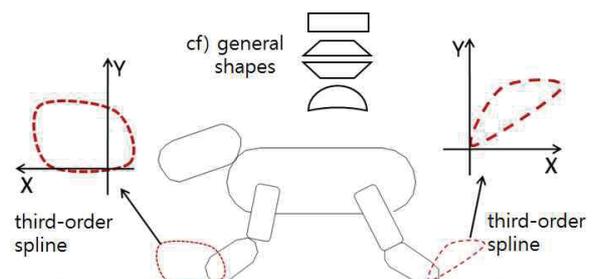


Fig. 1 An example of third-order spline locus.

Manuscript received June 12, 2013.

Manuscript revised September 11, 2013.

[†]The author is with the Dept of Electronics Engineering, Seokyeong University, Seoul, Korea.

^{††}The author is with Hyundai Heavy Industries Research Institute, Yongin, Korea.

*This Research was supported by Seokyeong University in 2010, 2013.

a) E-mail: ksseo@skuniv.ac.kr

DOI: 10.1587/transinf.E97.D.353

be the time to complete one step and D be the length of one step. Assuming that the paws are in contact with the ground at $t = 0$ and $t = T$, we get the following constraints:

$$\begin{cases} X(t) = 0 & t = 0, T \\ Y(t) = 0 & t = 0, T \end{cases} \quad (1)$$

Letting (X_e, Y_e) be the position of end point A when paw is lifting and (X_s, Y_s) be the position of start point B when paw is lowering. During a one-step cycle, assuming that the paw reaches the point A at $t = t_e$ and the point B at $t = t_s$ we get the following constraints:

$$X(t) = \begin{cases} X_e & t = t_e \\ X_s & t = t_s \end{cases} \quad Y(t) = \begin{cases} Y_e & t = t_e \\ Y_s & t = t_s \end{cases} \quad (2)$$

When the paw is in contact with the ground at $t = 0$ and $t = T$, letting $V_{xs}, V_{xe}, V_{ys}, V_{ye}$ be the X and Y velocity, we can get the following constraints:

$$\dot{X}(t) = \begin{cases} V_{xs} & t = 0 \\ V_{xe} & t = T \end{cases} \quad \dot{Y}(t) = \begin{cases} V_{ys} & t = 0 \\ V_{ye} & t = T \end{cases} \quad (3)$$

To satisfy constraint (1) (2) (3), and the continuity conditions of the first derivative and the second derivative, $X(t)$, $Y(t)$ can be characterized by third-order polynomial expressions. Besides the third-order spline loci, a large set of parameters for gait and stance and the number of movement points to complete one full step must be determined as mentioned in [4], [5].

3. GP Based Gait Generation in Joint Space

As a second approach, the concept of the gait generation in joint space is in Fig. 2. The joint trajectories of shoulder and knee for a quadruped robot are represented in 2-D space; the vertical axis is joint angle and the horizontal axis is time. Without the need for conversion of paw position from Cartesian space to a set of joint angles, a gait is determined directly by a series of joint positions (or angles), which corresponds to one cycle of paw locus in Cartesian space.

Specification of gait as a set of joint trajectories is done by evolving a polynomial function of time for each joint as a separate GP tree, but evolving them simultaneously.

The numerical expressions generated by each GP tree resemble those generated when using GP to perform symbolic regression. In Fig. 3, the GP tree on the right side represents some polynomial expression that translates as shown on the left into the joint angle for one of the quadruped robot joints.

Major differences between GA method are as follows. First, a solution for gait is simply represented by each joint's trajectory with fewer parameters. Second, inverse kinematics calculations are not necessary to compute the gait of the quadruped robot. Third, a solution for gait is a form of continuous curve, so no interpolation process is required. Fourth, the shape of paw locus is not known explicitly.

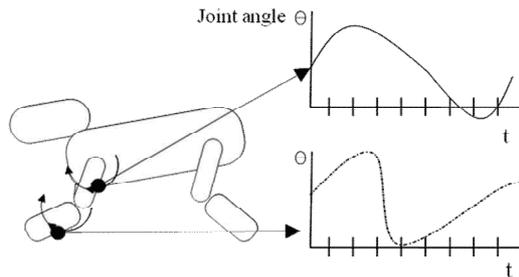


Fig. 2 Gait generation in joint space.

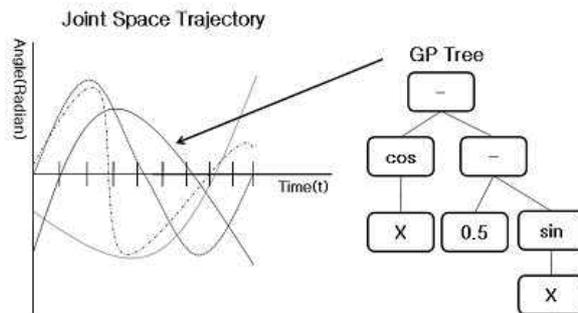


Fig. 3 Representation of trajectory of a joints via a GP tree.

4. Central Pattern Generator Based Gait Generation

CPGs (Central Pattern Generators) are neural circuits capable of producing coordinated patterns of high-dimensional rhythmic output signals while receiving only simple, low-dimensional input signals. To utilize this phenomenon, non-linear oscillators were introduced as mathematical models of the natural CPGs [11], [12]. The oscillator proposed in [11] is based on these differential equations:

$$\dot{\tau}v = -\alpha \frac{x^2 + v^2 - E}{E}v - x, \quad \tau\dot{x} = v \quad (4)$$

where v and x represent the current state of the oscillator, E is a positive constant that represents the energy of the oscillator, determines the rate of convergence towards the limit cycle and τ is the time constant that determines the oscillation's frequency.

By choosing the E and τ parameters, it is possible to control the amplitude and frequency of the oscillations. The result of Eq. (4) is introduced into Eq. (5): where a_{ij} and b_{ij} represent the strength of the coupling of the x and v states of oscillator j into oscillator i .

$$\dot{\tau}v_i = -\alpha \frac{x_i^2 + v_i^2 - E}{E}v - x + \sum_j^N \frac{(a_{ij}x_j + b_{ij}v_j)}{x_i^2 + v_i^2} \quad (5)$$

In this paper, the following topology was used in experiments, as are shown in Fig. 4. Each single node in here corresponds to an oscillator which is mentioned above.

Major features of CPG method are as follows. First, a solution for gait is represented by each joint's trajectory as

Table 1 Evolutionary parameters for three gait generation approaches.

GA Parameters	GP Parameters	PSO[13] parameters for CPG
Number of generations : 100 Population sizes : 150 Selection : Roulette Wheel Crossover : 0.9 Mutation : 0.1	Terminal Set : Random Constants, X Function Set : SIN, COS, +, -, *, / Number of generations : 100 Population sizes : 30*5 multiple populations Initial population : half_and_half Initial depth : 1-6, Max_depth : 15 Selection : Tournament (size=7) Crossover : 0.6, Mutation : 0.1, Reproduction : 0.3	Initial.inertia : 1.0 Inertia.reduction.factor : 0.995 Swarm.size : 50 Particle.velocity.max : 0.2 Particle.confidence.individual : 2.0 Particle.confidence.social : 2.0 Max.trials : 15000

Table 2 Comparison results for three gait generation approaches and built-in gait.

	Maximum Velocity cm/sec	Average Velocity (SD)	Vertical variation, cm(SD)	Average Height(SD), cm	Gait	Number of Parameters	Approach	Features
Built-in	17.82	17.82(N/A)	4.80(N/A)	14.81(N/A)	Cartesian space	Many	Human Design	Intuitive, Conventional I. K. is necessary
GA	21.84	19.25(1.62)	10.15(2.18)	14.63(2.57)	Cartesian space	Many	Optimization	Conventional-like I. K. is necessary
GP	26.53	15.04(4.77)	9.22(1.49)	14.62(2.20)	Joint space	Few	Optimization	Global optimum possibility I. K. is not necessary
CPG	26.02	18.38(3.85)	8.45(1.37)	14.10(1.91)	Joint space	Many	Adaptation Optimization	Oscillatory I. K. is not necessary

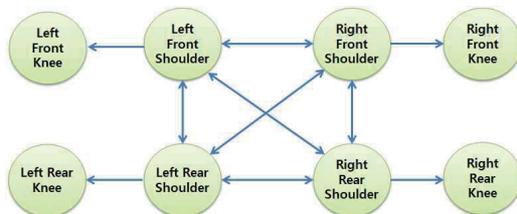


Fig. 4 Topology of CPG model.

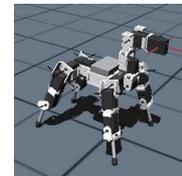


Fig. 5 Simulation model of quadruped robot.

similar as the case of GP method, inverse kinematics calculations are not necessary too. Second, gait solution is a form of continuous curve and the shape of paw locus is not known explicitly as same as GP method. Third, many CPG parameters are used to optimize a gait as similar as GA method.

5. Experiments and Analysis

Here the above three representative gait generation methods are compared for a Bioloid [15] Kit quadruped robot. Figure 5 shows simulation model of quadruped robot. In order to compare the three methods fairly and with as little bias as possible, the same experimental environments, performance indexes and available computational efforts are used.

Evolutionary parameters for three gait generation approaches shown in Table 1.

The fitness function of gait generation is defined to obtain the joint trajectory set that provides the fastest walking with only a small sideways diversion described in Eq. (6), where x is total forward distance reached, z is sideways diversion [2], [5].

$$fitness = ((0.9 \times (-x)) - |(0.4 \times z)|)^2 \tag{6}$$

20 iterations of experiments are executed for each method in Webots [14] robotics simulation software. The tabular results of velocities, the vertical variation and average height of shoulder, space, approach, and features for generated gaits by three methods and the built-in gait of Webots simulator are shown in Table 2. We observe that the GP-based method yielded the best max velocity, and the GA results showed superior average velocity to the other methods. The GP results showed larger deviation even though the max velocity was highest. The results of CPG represented less deviation than those of GP, and the max velocity was very close to the GP case. All three methods are superior to the existing foot trajectory based method.

From comparisons of variation and height of motion data for GA, GP and CPG, the vertical variation of CPG is seen to be lowest, which is closely related to stability of walking. In the other index of shoulder height, the CPG result is lowest, and has the lowest standard deviation, too. The vertical variation means the difference between the highest and lowest y coordinate of the shoulder during walking. The average height represent the mean y coordinate of the shoulder during walking, measured every 32 ms, and its standard deviation. The unit is cm.

Therefore, the movement with the CPG method seems to be more stable than with GA and GP methods in simulation. The GP method is next closest to CPG. Although, the variation of the built-in gait is most stable, it is only due to the manual design of gait to maintaining stability mainly.

6. Conclusions

Three gait generation methods based on GA (Genetic Algorithm), GP (genetic programming) and CPG (Central Pattern Generator) are implemented and compared using the task of a fast locomotion scheme for a quadruped robot including the existing foot trajectory based method. Optimizations using the three proposed methods are executed and analyzed using the Webots simulation for a quadruped robot built by Bioloid.

In simulation, the GP method is superior in terms of max velocity, while the GA and CPG methods show good average speed performance. The CPG method appears to be better than the GA and GP in regard to height of movement and variation of height, which are closely related with balancing and stability.

The three methods (GA, GP, and CPG) all seem to have some unique features for gait generation. The GA method provides intuitive understanding of gait patterns, because they are affected mainly by the locus of the paw. Therefore, we can figure out some characteristics of gait based on analyzing the loci of paw positions. However, this approach depends heavily on the pre-defined shape of the paw locus, so is very different from global optimization.

The GP method with joint trajectory optimization has more possibility to reach global optimization because it is not dependent on a locus shape, but depends only on performance. However, it is difficult to obtain a global optimum because of the enormous size of the search space.

The CPG method gives the possibility of practical optimization in this research, proving to be outstanding for adaptive walking in irregular terrain generally.

These investigations will provide some support for

the conjecture to find new and innovative gaits and will contribute to development of new gait research.

References

- [1] S. Yi, "Reliable gait planning and control for miniaturized quadruped robot pet," *Mechatronics*, volume 20, Issue 4, pp.485–495, June 2010.
- [2] G.S. Hornby, S. Takamura, T. Yamamoto, and M. Fujita, "Autonomous evolution of dynamic gaits with two quadruped robots," *IEEE Trans. Robotics*, vol.21, no.3, pp.402–410, 2005.
- [3] P. Doan, H. Vo, H. Kim, and S. Kim, "A new approach for development of quadruped robot based on biological concepts," *International Journal of Precision Engineering and Manufacturing*, vol.11, no.4, pp.559–568, 2010.
- [4] S. Chernova and M. Veloso, "An evolutionary approach to gait learning for four-legged robots," *Proc. IROS'04*, pp.2562–2567, Sendai, Japan, Sept. 2004.
- [5] H.L. Akın, Ç. Meriçli, T. Meriçli, K. Kaplan, and B. Çelik, *Cerberus'05 Team Report*, Technical Report Boğaziçi University, 2005.
- [6] W. Chen, "Odometry calibration and gait optimisation," Technical Report, The University of New South Wales, School of Computer Science and Engineering, 2005.
- [7] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, 1989.
- [8] J.R. Koza, *Genetic Programming: On the Programming of Computers by Natural Selection*, MIT Press, Cambridge, MA, USA, 1992.
- [9] K. Seo, S. Hyun, and E. Goodman, *Genetic Programming-Based Automatic Gait Generation in Joint Space for a Quadruped Robot*, *Advanced Robotics*, vol.24, no.15, pp.2199–2214, 2010.
- [10] D. Zonker and W.F. Punch, *lil-gp 1.1 User's Manual*, GARAGE, Michigan State University, 1998.
- [11] A.J. Ijspeert, A. Crespi, and J.-M. Cabelguyen, "Simulation and robotics studies of salamander locomotion," *Neuroinformatics*, vol.3, no.3, pp.171–195, 2005.
- [12] A.J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: A review," *Neural Networks*, vol.21, no.4, pp.642–653, 2008.
- [13] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proc. IEEE Int. Conf. on Neural Networks*, Piscataway, NJ, pp.1942–1948, 1995.
- [14] L. Hohl, R. Tellez, O. Michel, and A.J. Ijspeert, "Aibo and Webots: Simulation, wireless remote control and controller transfer," *Robotics and Autonomous Systems*, vol.54, pp.472–485, 2006.
- [15] Robotis Corporation, <http://www.robotis.com>