Analysis of two evolutionary gait generation techniques for different coordinate approaches

Soohwan Hyun and Kisung Seo

Dept. of Electronic Engineering, Seokyeong University, Jungneung-Dong 16–1, Sungbuk-Gu, Seoul 136–704, KOREA

Abstract: Planning gaits for legged robots is an important and challenging task that requires optimizing parameters in a highly irregular and multidimensional space. Two evolutionary gait generation techniques using GA (Genetic Algorithm), GP (genetic programming) based on Cartesian and joint space are compared to develop fast locomotion for a quadruped robot. Optimizations for two proposed methods are executed and analyzed using a Webots simulation and real experiment of the quadruped robot. The performance and motion features of GA-, GP-based methods are compared and analyzed.

Keywords: robot automatic gait generation, Cartesian space locus, joint space trajectory, Genetic Algorithm, genetic programming, quadruped robot

Classification: Science and engineering for electronics

References

1 Introduction
It is difficult to plan gaits for quadruped robots, because there are many degrees of freedom and, therefore, parameters, to be set properly [1]. Forward velocity of gait is widely used as a performance index in previous works [1, 2, 3, 4], which rewards forward motion and penalizes sideways diversion.

Existing automatic generation methods for quadruped gaits include: GA (Genetic Algorithm [5]) based approaches [1, 2, 3], GP (Genetic Programming [6]) based approach [4]. Most of current approaches use GA to optimize a pre-selected set of parameters [1, 2, 3].

Another approach was proposed [4] by authors to use genetic programming to optimize joint trajectories instead of the locus of paw positions in Cartesian space. The joint-space-oriented method has to optimize only 4–6 joint trajectories, rather than the more numerous parameters of Cartesian space.

However, it is hard to choose a specific gait generation method for a given robot, because there are many different factors to that influence the decision, including objectives, robot features and characteristics of problem spaces. Therefore, it is useful to provide comparisons and analysis for above two typical gait generations.

In this paper, both representative gait generation methods named above are compared for the quadruped robot represented by the Bioloid Kit. In order to compare the two methods in as fair and unbiased a fashion as possible, the same experimental environments and performance indexes are selected for all.

2 GA-based gait generation in Cartesian space
The problem of optimizing the gait speed becomes a parameter optimization problem in multi-dimensional space. As a first method, we use an evolutionary approach based on genetic algorithms to optimizing the locus of the robot’s paw for gait generation. A conventional SGA (Simple Genetic Algorithm) [5] is chosen.

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 [3]. The locus could be obtained by third-order spline interpolation by two points which are the end point of the lifting paw and the start point of the lowering paw. $X_s$, $Y_s$ are the position of start point, $X_e$, $Y_e$ are the position of end point, and $V_{xs}$, $V_{ye}$ are velocities of X and Y respectively as shown in Fig. 1 (top). For the front and rear leg, it compose a 22-dimensional search space.

3 GP-based gait generation in joint space
The other approach using GP (genetic programming) was proposed to optimize joint trajectories instead of the locus of paw positions in Cartesian space for simulation of Sony Aibo robot [4].

The concept of the gait generation in joint space is shown in Figure 1 (botom). The joint trajectories of shoulder and knee for a quadruped
Fig. 1. (top) An example of third-order spline locus and stance parameters for quadruped robot, (bottom) Gait generation in joint space and representation of trajectory of a joint via a GP tree.

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 Figure 1 (bottom), 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.

4 Experiments and analysis

4.1 Simulation environments

The Webots [8] mobile robotics simulation software developed by Cyberbotics provides the user with a rapid prototyping environment for modeling, programming and simulating mobile robots. Webots relies on ODE (Open Dynamics Engine) to perform accurate dynamic physics simulation.

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 equation (1), where $x$ is total forward distance reached, $z$ is sideways diversion.

$$fitness = ((0.9 \times x) - |(0.4 \times z)|)^2 \quad (1)$$
4.2 Experimental results

Two gait generation methods—GA (Genetic Algorithm), GP (Genetic Programming)—are executed using Webots simulation. The number of evaluations to be used by the two methods is set to be equal, to make a fair comparison (100–500 populations and 100 generations were tested).

15–20 experiments with various steps in a cycle are executed for each method. These experiments were run on a single Core 2 Duo 2.13 GHz PC with 2 GB RAM. Success rate is calculated by the number of walking properly of total trials. Success rate of GA approach is 80% which is bigger than of GP approach (66.67%).

Ten runs of each experiment are chosen for use in summarizing velocity statistics, eliminating those that do not work properly during replay of the simulation even though they fared well in the evolution process.

The results of average and max velocities for generated gaits by GA and GP, displayed is provided in Fig. 2 (a).

Joint graphs for each method are displayed in Fig. 2 (b) and (c). The average velocity is an average of 10 iterations for the max velocity in each experiment. The maximum velocity represents the best values among the 10 experiments. In the GA-based experiment, the best performance is 21.84 cm/s for max velocity (velocity) and 17.2 cm/s for average velocity of 10 experiments. The max velocity of best performance of the GP-based method is 26.53 cm/s and the average velocity is 13.28 cm/s. We observe that the GP-based method yielded the best max velocity, and the GA results showed superior average

![Comparison between GA and GP](image)

![Best GA Spline Data](image)

![Best GP Data](image)

**Fig. 2.** (a) Comparison Results between GA and GP (b) Best GA Gait Data (c) Best GP Gait Data
velocity to the other methods. The GP results showed larger deviation even though the max velocity was highest.

This is due to the property of gait representation and generation method rather than a qualitative property for GP and GA. The GA based gait generation method provides intuitive understanding of gait shapes and, because they are affected mainly by the locus of the paw. They naturally resemble the conventional and stable gaits. In other words, GA method can generate more similar ones to typical gait. This effect makes less chance of failure for gait relatively. However, this approach depends heavily on the pre-defined shape of the paw locus, so is more 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 joint trajectories for performance. However, it is difficult to obtain a global optimum because of the enormous size of the search space. Therefore the average speed performance of GP method is lower than that of GA.

Cumulative results of joint trajectories for best individuals in experiments are represented in Fig. 3. The numbers in parenthesis are the number of trials. As we can see, cumulative joint trajectories of GA in Fig. 3 (a) and (c) represent regular and uneven distributions. This seems that search space of GA approach is limited within specific ranges. However, the cumulative joint trajectories of GP in Fig. 3 (b) and (d) show irregular and considerably wide distributions since it seems that search space of GP method is more widely spread than that of GA’s.

![Fig. 3.](image)

(a) Cumulative results of joint trajectories (a) GA Gait (10), (b) GP Gait (10), (c) GA Gait (30), (d) GP Gait (30)
4.3 Bioloid real robot results

Experiments for the real robot are executed through transferring simulation data of the best individual for each method into the controller of the Bioloid quadruped robot. The results of the GP method appear more similar to the results in the simulation images than they did with the GA method. Most corresponding frames in the real and simulation runs appear almost identical.

The reason for the GP method that showed very similar movement between simulation and real experiment seems as follows. The GP method uses direct joint trajectories which are not necessary to transform into other space. Therefore it can generate more similar to the results in the simulation images. However, because GA method adopts a generation of paw locus in Cartesian space, it requires a transformation to joint space values for actuators by inverse kinematics calculation. During transformation from a point in Cartesian space to joint space, there may be existed some more discrepancies.

5 Conclusions

Two gait generation methods using GA (Genetic Algorithm), GP (genetic programming) with different coordinates are implemented for comparison to develop a fast locomotion scheme for a quadruped robot. Optimizations using the two proposed methods are executed and using the Webots simulation for a quadruped robot built by Bioloid. The resulted phenomena are analyzed carefully for the aspects of evolutionary approaches and different coordination space.

In simulation, the GP method based on joint space shows superior results in max velocity, while the GA method based on Cartesian space shows good performance in average speed. In real experiments, the GA and GP approaches showed relatively similar movements to their respective simulation data. Especially, the GP method showed very similar movement between simulation and real experiment.

Acknowledgements

This Research was supported by Seokyeong University in 2010.