Springer

Journal of Mechanical Science and Technology 23 (2009) 114~123

Journal of Mechanical Science and Technology

www.springerlink.com/content/1738-494x
DOI 10.1007/s12206-008-0920-9

Trajectory optimization with GA and control for quadruped robots †

Key Gew Chae and Jong Hyeon Park*
School of Mechanical Engineering Hanyang University, Seoul, 133-791, Korea

(Manuscript Received June 13, 2008; Revised July 14, 2008; Accepted September 18, 2008)

Abstract

This paper proposes an optimal galloping trajectory, which costs low energy and guarantees the stability of the quadruped robot. In the realization of fast galloping, the trajectory design is important. For a galloping trajectory, we propose an elliptic leg trajectory, which provides simplified locomotion to complex galloping motions of animals. However, the elliptic trajectory, as an imitation of animal galloping motion, does not guarantee stability and minimal energy consumption. We propose optimization based on energy and stability using a genetic algorithm, which provides a robust and globally optimized solution to this multi-body, highly nonlinear dynamic system. To evaluate and verify the effectiveness of the proposed trajectory, a series of computer simulations were carried out.

Keywords: Quadruped robot; Gallop; Genetic algorithm; Elliptic trajectory; Cubic polynomials

1. Introduction

To date, much previous research on legged robots has been conducted. Among those, quadrupeds are able to use the terrain conditions for minimizing energy consumption. There are several merits for using quadruped robots as mobile robots. They can walk or run over an uneven surface, such as in a forest, without leading to undue damage to the environment.

Galloping locomotion can be simplified as throwing a ball with a spring once we assume that the running motion can be viewed as simply iterations of the flight and thrust. If there is no external resistance, a ball with a spring can continually bounce back off the ground. The so-called spring-loaded inverted pendulum (SLIP) was used to generate the motions of a running robot, and the angle between the mass center and the foot; the stiffness of the contact leg and the contact velocity were identified as important variables [1, 2]. The relations among the running speed and the flight phase duration were researched [3]. Many others developed schemes of running based on the SLIP model [4-6].

Raibert also developed the fastest quadruped robot with a running speed of about 2.9 m/s [1]. His machine uses prismatic legs with pneumatic and hydraulic actuation. Marhetka simulated a planar quadruped that sustained its a speed of close to 7 m/s [7]. This system used prismatic legs with electrical actuation and moved at a speed high enough to benefit from galloping. However, those mechanisms are vulnerable to impacts despite the fact that the multi-joint leg can compensate for energetic impacts. Nichol studied a set of models and the design requirements, which facilitate the design of a quadruped machine capable of gallop gaits [8]. Quadruped design begins with the design of the legs. A spring-mass inverted pendulum model is used and has been shown to describe energetic legged locomotion.

Animals show their marvelous abilities in adaptability and energy efficiency as an instinct. Animal-like locomotion is ideal and the ultimate goal we should pursue [9]. A computer simulation of galloping was successfully performed based on elaborated

† This paper was recommended for publication in revised form by Associate Editor Doo Yong Lee
*Corresponding author. Tel.: +82 2 2220 0435, Fax.: +82 2 2298 4634
E-mail address: jongpark@hanyang.ac.kr

trajectories and optimization in [10]. As a simple method of imitating a typical animal gallop, trajectories for galloping were proposed based on the ellipse [11]. However, the timing of foot contact with the ground and lift-off was predetermined and constant, and the phase difference between the front and rear feet remained constant regardless of the galloping state. In time, robot trajectories were modeled as polynomials and the robot state was parameterized with a polynomial function, which was used for optimization by a genetic algorithm [12]. Among the many optimization algorithms, genetic algorithms are the best for finding global solutions in highly nonlinear systems [13, 14].

In the current work, we propose an optimized ellipse-based leg-trajectory of quadruped robots based on a real-coded genetic algorithm (RCGA). Timing of elliptic foot motions is optimized when given a galloping speed. The RCGA is used mainly due to its simplicity, speed, and ease in dealing with complex constraints. In computer simulations of the galloping of a 6-DOF quadruped robot, the commercial software RecurDyn® is used. The simulations show that the proposed trajectory is energy efficient while maintaining galloping stability.

This paper is organized as follows. Section 2 introduces the quadruped robot model in the sagittal plane, which is used in the current work. An ellipse-based trajectory-generation and phase detection are also explained along with the configuration of the robot controller and the impact force control algorithm. Next, in Section 3, the energy and stability optimization with GA is explained. The results of computer simulations of galloping are also covered in that section, followed by conclusions in Section 4.

2. Ellipse-based trajectory generation

2.1 Quadruped robot model

The size of the quadruped robot considered in this paper is similar to that of a typical dog. At each leg, there are two active pitch-joints, one roll-joint, and one passive pitch-joint. The purpose of the passive pitch joint is to function as a spring used in the SLIP during galloping. Fig. 1 shows its 2D model on the sagittal plane, along with the coordinate frames for the front and rear leg. A force sensor is assumed to be installed at the sole of the foot. The length of each link, spring coefficient, damping coefficient, and the total mass are summarized in Table 1. Later, the quadruped robot is simulated with the commercial software RecurDyn®, a computer program for dynamic model simulation.

Table 1.

A. Specifications of the quadruped robot.

	Front		Rear	
	Length	Mass	Length	Mass
Link1	0.13 m	0.47 kg	0.15 m	0.57 kg
Link2	0.18 m	0.48 kg	0.15 m	0.18 kg
Link3	0.1 m	0.18 kg	0.1 m	0.18 kg
Base link	0.6 m	17.7 kg	Total Mass	20 kg

B. Specifications of passive legs

Front spring coefficient	80 Nm	Rear spring coefficient	80 Nm
Front damping coefficient	1 Nms	Rear damping coefficient	1 Nms

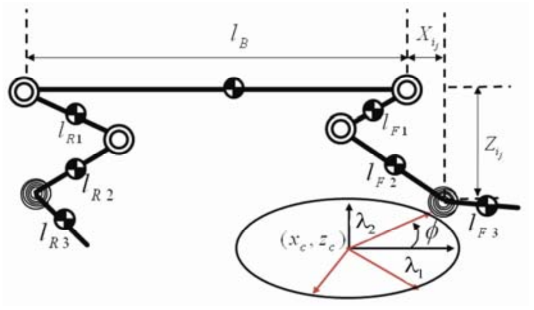


Fig. 1. Frame assignments for the 2D quadruped robot.

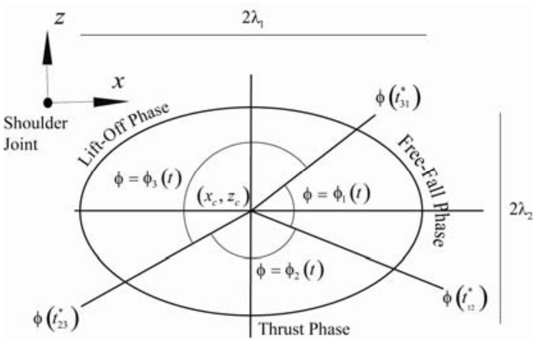


Fig. 2. Elliptic trajectory.

2.2 Elliptic trajectory generation

The trajectory of each leg is based on a trajectory ellipse. More specifically, the shape of the trajectory of each foot relative to its shoulder is an ellipse. An elliptic trajectory of each foot is divided into three different sections depending on the gallop phases of the quadruped robot. The first section is during the time of fall of the robot body (or base) from its peak elevation until the foot touches the ground, or Free-Fall Phase. The second is during the time when the foot is in contact with the ground and the body is pushed forward with respect to the ground, or Thrust Phase. The third is during the time between the moment that the foot and the body are completely off the ground and the moment that the body reaches its peak elevation level, or Lift-Off Phase. Fig. 3 shows these three sections of an elliptic trajectory of a leg. The elliptic trajectory in each section is denoted by $\phi_i(t), i = 1,2,3$. Free-Fall, Thrust, and Lift-Off Phases are associated with elliptic trajectory sections 1, 2, and 3, respectively.

It will be assumed for the time being that the elliptic angles of transition from one section to another are known. Suppose ϕ^*_{12} , ϕ^*_{23} , and ϕ^*_{31} denote the elliptical angle of the leg trajectory at the transitions from section 1 to section 2, from section 2 to section 3, and from section 3 and section 1, respectively. These three angles are to be determined optimally later with a genetic algorithm.

The first section starts when the COG of the robot is its peak elevation level and ends when the corresponding foot comes into contact with the ground. During this motion, the COG velocity is always negative, i.e., $\dot{z}_{COG} < 0$. The ellipse angle for the first section is modeled as a third-order polynomial of time *t*.

$$\phi_1(t) = a_{13}t^3 + a_{12}t^2 + a_{11}t + a_{10} \tag{1a}$$

The elliptic angles for the second and third sections are similarly modeled as a first-order and third-order polynomials, respectively.

$$\phi_2(t) = a_{21}t + a_{20} \tag{1b}$$

$$\phi_3(t) = a_{33}t^3 + a_{32}t^2 + a_{31}t + a_{30} \tag{1c}$$

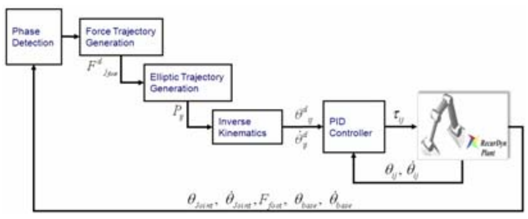

Phase Detection
Force Trajectory Generation
F^d_{foot}
Elliptic Trajectory Generation
P_{ij}
Inverse Kinematics
θ^d_{ij}
$\dot{\theta}^d_{ij}$
PID Controller
τ_{ij}
RecurDyn Plant
$\theta_{ij}, \dot{\theta}_{ij}$
$\theta_{Joint}, \dot{\theta}_{Joint}, F_{foot}, \theta_{base}, \dot{\theta}_{base}$

Fig. 3. Outline of the control system.

There are *ten* unknowns in the polynomial coefficients, which are to be determined. First, the continuity in the angular position and speed at the intersections require:

$$\phi_1(t^*_{12}) = \phi_2(t^*_{12}) \tag{2a}$$

$$\phi_2(t^*_{23}) = \phi_3(t^*_{23}) \tag{2b}$$

$$\phi_3(t^*_{31}) = \phi_1(t^*_{31}) \tag{2c}$$

$$\dot{\phi}_1(t^*_{12}) = \dot{\phi}_2(t^*_{12}) \tag{2d}$$

$$\dot{\phi}_2(t^*_{23}) = \dot{\phi}_3(t^*_{23}) \text{ and} \tag{2e}$$

$$\dot{\phi}_3(t^*_{31}) = \dot{\phi}_1(t^*_{31}) . \tag{2f}$$

where t^*_{ij} denotes the time when the transition between the i-th and the j-th sections occurs.

Note that the transition times introduced here, t^*_{12} , t^*_{23} , and t^*_{31} , also have to be determined. Without loss of generality, the time is set to be zero when the COG of the robot is initially at its peak. And, the moment of touch down can be computed based on the elevation of the COG relative to its position at the moment of touch down. If it is assumed that the whole body of the robot is modeled as a mass, it falls freely under gravity, and with no initial velocity for the distance of H and the duration of the fall is:

$$\Delta t_F = \sqrt{2gH} . \tag{3}$$

Thus,

$$t^*_{12} = \Delta t_F . \tag{4}$$

Also, at the moment of touch down and lift-off, its angular speed is set to be

$$\dot{\phi}_1(t^*_{12}) = \frac{v_0}{\lambda_2} \tag{5}$$

where v_0 is the desired forward speed of the robot and λ_2 is the half of the minor axis of the ellipse. Note that in the second section of leg motion, the speed of the COG of the robot is reversed due to the vertical ground reaction force, which will be explained in Section 2.4, i.e., $\dot{z}_{COG} < 0 \rightarrow \dot{z}_{COG} > 0$.

In addition, suppose that the transition angles are known, and thus

$$\begin{bmatrix} \phi_1(t^*_{12}) & \phi_2(t^*_{23}) & \phi_3(t^*_{31}) \end{bmatrix}^T = \xi . \tag{6}$$

Then, Eqs. (2a-2f), (5) and (6) work as *ten* constraints for solving the unknowns in Eq. (1).

Finally, the cycle time to complete one elliptic trajectory is set to be T:

$$t_{31}^{*} = T \,. \tag{7}$$

If the vertical GRF is symmetric in time, the durations of the flight-off phase (section 3) and the fall-down phase (section 1) will be identical:

$$t_{12}^{*} = T - t_{23}^{*} = \Delta t_F \,. \tag{8}$$

The time span in which the forward thrust is generated, or the duration of the second section of leg motion, becomes

$$t_T = T - 2\Delta t_F \,. \tag{9}$$

Once cycle time T and the maximum vertical elevation of the COG of the robot relative to its vertical level at the moment of the foot contact with the ground, H, are set depending on the gallop pattern, all the timings for the sectional transitions can be determined by Eqs. (7), (8), and (9). This is similar to the case of a SLIP model, where only two parameters of the spring stiffness of the leg and the hopping period determine the vertical motion characteristics.

Now, from the angle information we obtained of each section, the end point of the robot leg is determined by the following elliptic equation.

$$x(t) = \lambda_1 \cdot \cos\phi(t) + x_c(t) \tag{10a}$$

$$z(t) = \lambda_2 \cdot \sin\phi(t) + z_c(t) \tag{10b}$$

where (x_c, z_c) is the location of the center of the trajectory ellipse of the corresponding leg, which will be changed for control of the vertical ground reaction force as explained in Section 2.4.

2.3 Configuration of controller

The controller used for the quadruped robot has five components: phase detection, force control, elliptic trajectory generator, inverse kinematics, a PID controller as shown in Fig. 3. RecurDyn®, a commercial software for simulating dynamics, is used to simulate the dynamics of the quadruped robot. The controller is written in the environment of Matlab. For example, the torque inputs to the motors or the RecurDyn®® inputs are computed in Matlab codes. Status on foot contacts is assumed to be determined by sensors and actually obtained from RecurDyn®® during simulations. The desired trajectories of the legs in Eq. (10) are transformed into the desired joint trajectories of the links of the robot. PID controllers are used to track the desired trajectories. Note also that different values for PID gains are used depending on the trajectory section. This is due to minimized ground impact at the touch down.

2.4 Impact force control

In order to control the balance of the body while maintaining the gallop height at the desired level, it is extremely important to keep the ground reaction force appropriately. The desired vertical ground reaction force should be carefully computed and controlled so that the pitch angle of the body (or base) remains small and the galloping is stable.

The intended force trajectory is derived from the following method. First, the governing equation of momentum is

$$\Delta(\mathrm{mv}) = \int_0^{t_e} F dt \tag{11}$$

where (mv), t_e, and F are the linear momentum, the duration of the contact and the contact force, respectively. Considering the motion of the robot in the vertical direction, the contact force F is the so-called vertical ground reaction force (GRF). The *desired* vertical GRF is proposed to be modeled as a sinusoidal function of time t:

$$F(t) = |F| \sin\left(\frac{\pi}{\hat{t}_e} t\right) \geq 0, \;\; 0 \leq t \leq \hat{t}_e \tag{12}$$

where $\hat{t}_e$ is the *expected* duration of contact. Note that the actual duration of contact is not known

From Eqs. (11) and (12),

$$\Delta(\mathrm{mv}) = \mathrm{M}v_{lo} - \mathrm{M}v_{td} = \int_0^{t_e} F dt = |F| \frac{\hat{t}_e}{\pi}\left[1 - \cos\left(\pi \frac{t_e}{\hat{t}_e}\right)\right] \tag{13}$$

where $v_{lo} = \sqrt{2gh_{COG\,top}}$ which is the COG velocity at liftoff and v_{td} is the COG velocity at touchdown.

Then, under the assumption of $t_e = \hat{t}_e$,

$$|F| = \frac{\pi}{2\hat{t}_e}(\mathrm{M}v_{lo} - \mathrm{M}v_{td}), \tag{14}$$

Assuming further that the maximum height of the center of gravity from the ground is constant at H all the time, that is,

$$v_{lo} = v_{td} = \sqrt{2gH}, \tag{15}$$

$$|F| = \frac{\pi M\sqrt{2gH}}{\hat{t}_e}. \tag{16}$$

In order to keep the pitch angle of the main body (or base), θ_b, steady at zero, the robot has to distribute the vertical GRF to the front and rear legs. This force trajectory is distributed to both legs as shown in the following equations.

$$F_j(t) = |F|\sin\left(\frac{\pi}{\hat{t}_e}t\right)\eta(1-\sin(\theta_b)) \tag{17}$$

where η is 1.04 and 2.73 for front and rear legs, respectively, which are tuned in a simulation.

The desired GRF in Eq. (17) is to be tracked at the foot. The force measured at the feet is compared the desired GRF, and any error would result in the change of the center position of the trajectory ellipse proportional to the error. When the contact force is smaller than the desired force, the corresponding trajectory ellipse is lowered vertically. In the other case, the trajectory ellipse center moves up and reduces the vertical GRF. In this work, the center of the trajectory ellipse changes only in the vertical direction, i.e., z_c, changes depending on the force tracking error. The block diagram of a force tracking control scheme is shown in Fig. 4. Since the information from the force sensor is very fluctuant during the contact with the ground, it is often difficult to adapt the force tracking control. To solve this problem and to increase the robustness of the force control scheme, low pass filters are used for measured GRF and for the output, which is modification to the center position of the trajectory ellipse. The low pass filter is in the form of

$$Y(t+1) = e^{-\omega h}Y(t) + U(t+1)(1-e^{-\omega h})$$

where U and Y are input and output of the filter, and h is the control loop-time.

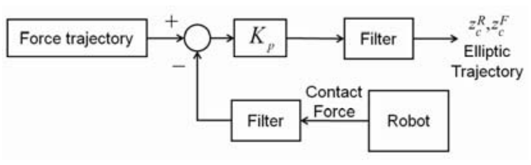


Fig. 4. Force control block diagram.

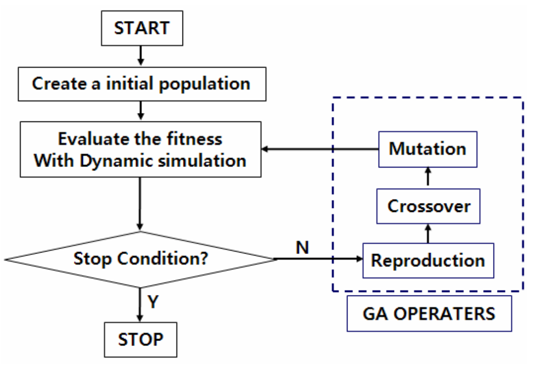
3. Energy and stability optimization with GA

3.1 Genetic algorithm

A genetic algorithm is used to solve many optimization problems because it is robust and can find a globally optimal solution even with a highly nonlinear cost function. Since there are many nonlinear components involved in the galloping of quadruped robots, it is highly desirable to use a genetic algorithm in finding the globally optimal solution.

A GA operates in conjunction with the dynamic model of the quadruped robot, as shown in Fig. 5. At the first step of the process, an initial population is created as a starting point for the search. With a trajectory based on the initial parameters, a numerical dynamic analysis is carried out. From the numerical dynamic simulation, the objective function is evaluated.

Only if the value of the objective function is above a given threshold are the chromosome or the parameters of galloping expected to participate in the reproduction process through crossover and mutations. An efficient individual is selected more frequently and reproduced. On the other hand, an inefficient individual disappears from a population. The crossover operator takes two chromosomes and swaps part of their genetic information to produce new chromosomes. The mutation operator produces new genetic struc-


START
Create a initial population
Evaluate the fitness With Dynamic simulation
Stop Condition?
N
Y
STOP
Mutation
Crossover
Reproduction
GA OPERATERS
Fig. 5. Flow chart of a running cycle.

tures in the population by randomly modifying some of the genes, helping the search algorithm to escape from a local minimum point.

If the quadruped robot falls on a gallop or becomes unstable, the value of the objective function becomes high. Thus, in the next generation, the penalized population will become useless elements from the elite group, securing the stability automatically. The offspring produced by these genetic processes are the next population to be evaluated, and these processes are repeated until a satisfactory individual is found or other stop conditions are satisfied.

3.2 Cost function and design variables for optimization.

The cost function to be minimized by the algorithm is set to be:

$$J = \int_{t_0}^{t} (P^T QP + F^T RF) dt$$

where P and F denote the power consumed at the motors and the squared value of contact force applied at the feet, or GRF, in the vertical direction by the ground, respectively, and Q=R=I. The cost function is to minimize the consumed power during a gallop and at the same time to reduce the size of the vertical GRF. A large difference in the GRFs at the front and rear legs results in a high pitching motion and eventual instability of the quadruped robot. Note that it is not simply the size of the GRF but squares of the GRF. Thus, the term associated with the GRF in the cost function would minimize the possibility of too much imbalanced GRF at the front and rear legs, which would result in a large pitching angle of the body (or base). Thus, the resulting trajectory would be energy efficient and stable.

As indicated in Section 2.2, the elliptic angles at transitions, $\xi \in \Re^3$, are used as the chromosome. Since the elliptic trajectory would result in the feet turning in the clockwise direction,

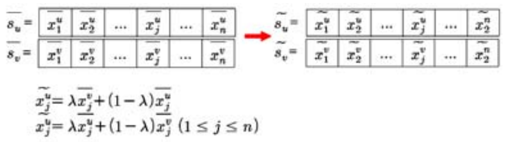
$$0 < \phi_2\left(t^*_{23}\right) < \phi_1\left(t^*_{12}\right) < 2\pi ,$$

which is used as another constraint to be satisfied. If this condition was not met for a certain chromosome, it would not be regarded as a good chromosome and thus would be tossed out of the population.

3.3 Operators for RCGA

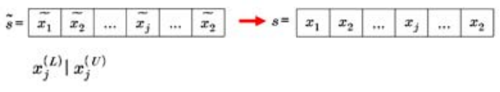
As an RCGA operator, arithmetical crossover and bounded mutation are used. Arithmetical crossover gives a wider variety of population values; however, bounded mutation provides a more global solution.

- Arithmetical crossover


$\bar{s}_u = \bar{x}^u_1 \; \bar{x}^u_2 \; \dots \; \bar{x}^u_j \; \dots \; \bar{x}^u_n$
$\bar{s}_v = \bar{x}^v_1 \; \bar{x}^v_2 \; \dots \; \bar{x}^v_j \; \dots \; \bar{x}^v_n$
$\tilde{s}_u = \tilde{x}^u_1 \; \tilde{x}^u_2 \; \dots \; \tilde{x}^u_j \; \dots \; \tilde{x}^n_2$
$\tilde{s}_v = \tilde{x}^v_1 \; \tilde{x}^v_2 \; \dots \; \tilde{x}^v_j \; \dots \; \tilde{x}^n_2$
$\tilde{x}^u_j = \lambda \bar{x}^v_j + (1-\lambda)\bar{x}^u_j$
$\tilde{x}^u_j = \lambda \bar{x}^u_j + (1-\lambda)\bar{x}^v_j \; (1 \le j \le n)$

This operator is defined as a linear combination of two vectors. If X^u and X^v are to be crossed, this operator uses a random value $\lambda, \lambda \in [0,1]$ as it always guarantees closedness.

- Bounded mutation


$\tilde{s} = \tilde{x}_1 \; \tilde{x}_2 \; \dots \; \tilde{x}_j \; \dots \; \tilde{x}_2$
$s = x_1 \; x_2 \; \dots \; x_j \; \dots \; x_2$
$x^{(L)}_j \,|\, x^{(U)}_j$

This operator also requires a single parent $\tilde{X}_j$ and produces a single offspring X_j. The operator is a variation of the uniform mutation with X_j A being either the ends of the domain, each with equal probability.

3.4 Environment model

In the sagittal plane, a galloping simulation of the quadruped robot was carried out. We defined the four fixed parameters as base initial height of 0.31 m, base initial velocity of 2.5 m/s, center of elliptic trajectory location, and length of elliptic trajectory as shown in Table 2. By using the commercial software, RecurDyn®, the contact force between the foot and the ground is computed with

$$F_n = k\delta^{m_1} + c\frac{\dot{\delta}}{|\dot{\delta}|}|\dot{\delta}|^{m_2}\delta^{m_3}$$

Table 2. Fixed parameters.

	x_c	z_c	Length of Major Axis (m)	Length of Minor Axis (m)
Front	-0.05 m	-0.16 m	0.1	0.03
Rear	-0.02 m	-0.145 m	0.1	0.03

Table 3. GA parameters.

Max. generation	200
Population	60
Chromosome length	4
Crossover ratio	0.9
Mutation ratio	0.02

3.5 Genetic algorithm parameters

In the GA section, for fast convergence and an accurate solution, we adopted the RCGA with gradient-like selection and used simple crossover and boundary mutation methods. The parameters on the GA are summarized in Table 3.

3.6 Simulation results

In this paper, commercial software called RecurDyn®, a dynamic modeling and simulation tool, is used. It offers various kinds of joints, bodies, actuators, sensors and constraint blocks to build up a dynamic model and reflects its physical and dynamical parameters like coordinates, length and mass of links, geometrical constraints, etc. It has the advantage of being convenient in simulations without developing complicated mathematical equations and dynamics.

A simulation of the quadruped robot running on a flat ground is performed. The simulation is executed for 3 seconds. When the simulation of the resulting optimized trajectory is completed, the quadruped robot completes 13 cycles of gallop and moves approximately 7.5 m in distance.

Fig. 6 shows how the value of the cost function changes as the simulation progresses. Approximately, after the 21st generation, it reaches close to its minimum value and becomes almost steady. The stability of the resulting gallop trajectory is verified with the phase plot shown in Fig. 7. In this multi-body and high-nonlinear dynamic quadruped robot, a phase plot of the pitch angle and the pitch angle rate of the body (or the base) is a good choice in verifying the stability of the robot. Regular pattern of the pitch motion of the body (or the base) shown in Fig. 7 also indicates that the galloping is stable. The resulting joint angle and angular velocity are shown in Fig. 8. The trajectory of the COG of the quadruped robot is shown in Fig. 9.

Fig. 10 shows the actual trajectory of the feet, which is not of an elliptical shape at all. This is due to

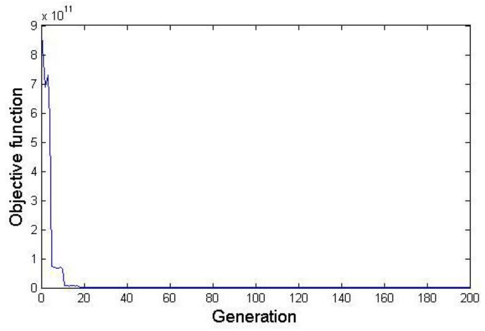


Fig. 6. Objective function.

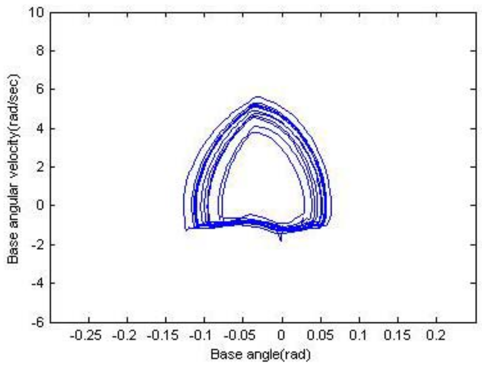
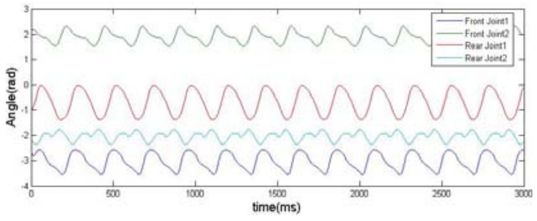
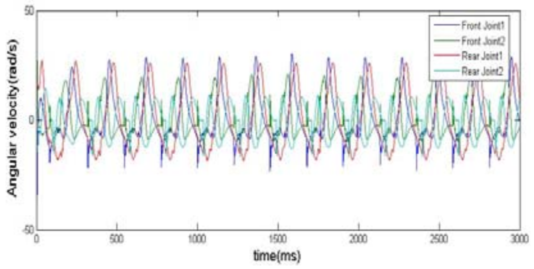


Fig. 7. Phase plot.



(a) Joint angle



(b) Joint angular velocity

Fig. 8. Motion at the joints.

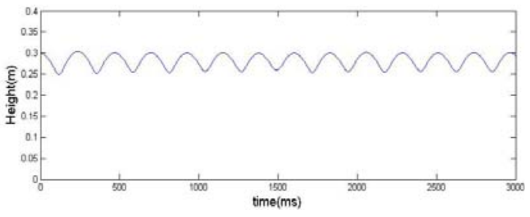
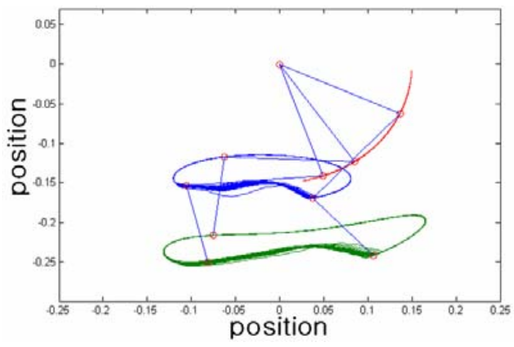
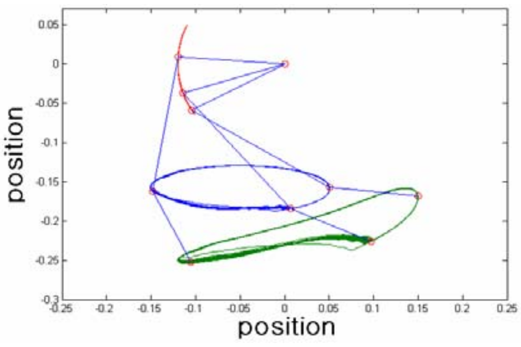


Fig. 9. Trajectory of the body of the robot.

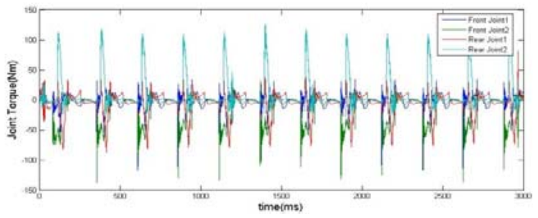


(a) Rear leg

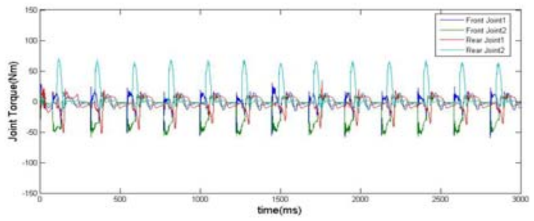


(b) Front leg

Fig. 10. Trajectory of the feet.

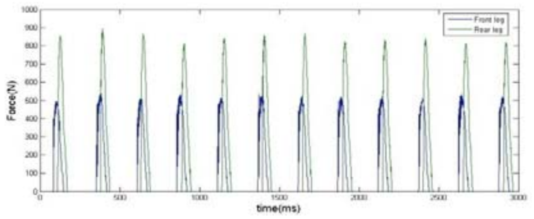


(a) Joint torque with a stable trajectory found by trial-and-error

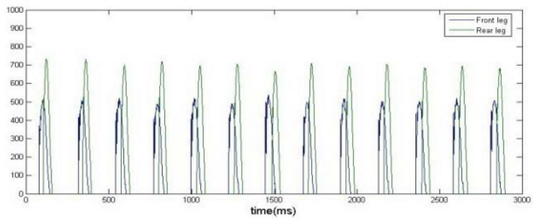


(b) Joint torque with the trajectory found with the genetic algorithm

Fig. 11. Comparison in joint torque.



(a) Contact force with a stable trajectory found by trial-and-error



(b) Contact force with the trajectory found with the genetic algorithm

Fig. 12. Comparison in the vertical contact force or GRF.

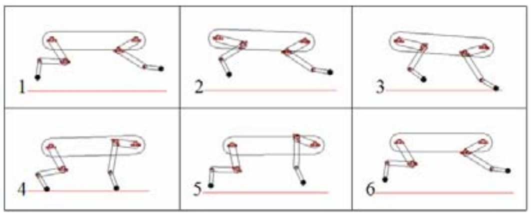


Fig. 13. Consecutive scenes of the quadruped robot galloping on a flat surface.

the fact that changes in the center position of the trajectory ellipse are made by the force feedback. The optimized trajectory resulting from the GA, compared with the previous research results, where trajectory parameters were tuned by trial-and-error, shows significant improvements in joint torque and contact force (or GRF). With the optimized trajectory, the total consumed energy is reduced by 30%. Figs. 11 and 12 compare the joint torque and the ground contact force or GRF, respectively. Fig. 13 shows the consecutive scenes of the quadruped robot galloping with the optimized trajectory.

4. Conclusions

Optimized trajectory is obtained with a genetic algorithm for a galloping quadruped robot, which has four active joints and two passive joints in each leg. The proposed optimized elliptic trajectory makes it possible to generate stable galloping motions based on an elliptic trajectory, which mimics the motions of an animal. A galloping trajectory is divided into three main phases for the purpose of trajectory generation. The trajectory in each phase is represented by polynomials in time, which are then optimized for stability and energy efficiency. Optimal locations of the intersections of the phases are found with a genetic algorithm. The galloping height and body posture are maintained through the impact force control with the optimized leg trajectory. In the near future, the trajectory designed in the work will be applied to a real quadruped robot.

Acknowledgment

This work was supported by a grant from the Korea Science and Engineering Foundation. (R01-2006-000-11041-0).

References

[1] M. N. Raibert, *Legged Robots That Balance*, MIT Press, Cambridge, (1986).

[2] S. H. Hyon and T. Emura, Running Control of a Planar Biped Robot Based On Energy-Preserving Strategy, *Proc. of IEEE Int. Conf. on Robotics and Automation*, (2004) 3791-3796.

[3] J. K. Hodgins and M. N. Raibert, Adjusting Step Length for Rough Terrain Locomotion, *IEEE Tr. on Robotics and Automation*, 7 (3) (1991) 289-298.

[4] H. Zou and J. P. Schmiedeler, Dynamic Modeling of Quadrupedal Running Gaits Using a Simple Template with Asymmetrical Body Mass Distribution, *Proc. of ASME Int. Design Engineering Technical Conferences*, (2004) 717-726.

[5] A. Kawamura, K. Sugio, K. Suzuki and C. Zhu, Simulation Study on One Leg Jumping for Biped Running, *Proc. of IEEE Int. Workshop on Advanced Motion Control*, (2004) 71- 74.

[6] Alexander, R. McN., Three Uses for Spring in Legged Locomotion, *Int. J. of Robotics Research*, 9 (2) (1990) 53-61.

[7] D. W. Marhefka, *Fuzzy Control and Dynamic Simulation of Quadruped Galloping Machine*, Ph.D. Thesis, Ohio State University, (2000).

[8] J. G. Nichol, *Design for Energy Loss and Energy Control in a Galloping Artificial Quadruped*, Ph.D. thesis, Stanford University, (2005).

[9] H. M. Herr and T. A. McMahon, A Galloping Horse Model, *Int. J. of Robotics Research*, 20 (1) (2001) 26-37.

[10] D. P. Krasny and D. E. Orin, Evolution of Dynamic Maneuvers in a 3D Galloping Quadruped Robot, *Proc. of IEEE Int. Conf. on Robotics and Automation*, (2006) 1084-1089.

[11] K. Y. Kim, O. H. Kwon, J. S. Yeon and J. H. Park, Elliptic Trajectory Generation for Galloping Quadruped Robots, *Proc. of IEEE Int. Conf. on Robotics and Biomimetics*, (2006) 103-108.

[12] G. Capi, S. Kaneko, K. Mitobe, L. Barolli and Y. Nasu, Optimal Trajectory Generation for a Prismatic Joint Biped Robot Using Genetic Algorithms, *Robotics and Autonomous Systems*, 38 (2) (2002) 119-128.

[13] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Optimization*, Wiley-Interscience, (1999).

[14] M. Srinivas and L. M. Patnaik, L. M., Genetic Algorithms: A Survey, *Computer*, 27 (6) (1994) 17-26.

Jong Hyeon Park received the B.S. degree in mechanical engineering from Seoul National University, Seoul, Korea, in 1981 and the S.M. and Ph.D. degrees from the Massachusetts Institute of Technology (MIT), Cambridge, in 1983 and 1991, respectively. Since 1992, he has been with the School of Mechanical Engineering at Hanyang University, Seoul, Korea, where he is currently a professor. He was a KOSEF (Korea Science and Engineering Foundation)-JSPS (Japan Society for the Promotion of Science) Visiting Researcher with Waseda University, Tokyo, Japan, in 1999, and a KOSEF-CNR (Consiglio Nazionale delle Ricerche) Visiting Researcher with Scuola Superiore Sant'Anna, Pisa, Italy, in 2000, a Visiting Scholar with MIT, Cambridge, USA, in 2002-2003. He was also associated with Brooks Automation Inc., Chelmsford, MA, in 1991-1992 and 2001-2002. His research interests include biped robots, robot dynamics and control, haptics, and bio-robots. He is a member of the IEEE (Institute of Electrical and Electronics Engineers), KSME (Korea Society of Mechanical Engineers), ICROS (Institute of Control, Robotics and Systems), KROS (Korea Robotics Society), KSAE (Korean Society of Automotive Engineers), KSPE (Korean Society of Precision Engineering) and KSEE (Korean Society for Engineering Education).