Path Tracking Controller of Quadruped Robot for Obstacle Avoidance Using Potential Functions Method

Giang Hoang[#1], Hak Kyeong Kim[#1], and Sang Bong Kim[#1]

[#]Department of Mechanical and Automotive Engineering, Pukyong National University,
Busan 608-739, Korea
[1]kimsb@pknu.ac.kr

***Abstract*— *This paper proposes a tracking controller for obstacle avoidance of a quadruped robot using potential functions method. The followings are done for this task. At first, a ceiling-mounted camera system is installed for image processing. The goal point and obstacles are separated and recognized by a color recognition method. Second, a path planning algorithm using potential functions method is proposed to generate the path to avoid obstacles and to plan a path for the quadruped robot to reach from start point to goal point. Third, a quadruped robot is chosen as the mobile platform for this study and the kinematic model for the robot is presented. Fourth, a tracking controller is designed for the quadruped robot to track the trajectory based on the backstepping method using Lyapunov function. Finally, the simulation results are presented to show the effectiveness of the proposed trajectory planning algorithm and the tracking controller.*

[Keywords— *Path tracking; back stepping; obstacles avoidance; potential functions; quadruped robot].*

I. INTRODUCTION

During the last decade, research on walking robots has been focused mainly on gait generation. Many gaits have been studied and implemented in walking machines. Although gait generation has been a very important subject for researching, lack of attention has been paid to path tracking with walking robot. Zhang et al. formulated the wave-crab gait and the circular wave gait in [1]. In [2], a four-legged walking robot called PVII was built by Hirose and Umetani. The PVII can maintain a horizontal body orientation and has contact sensor on each foot to detect contact with the ground. In [3], Ma et al. proposed a gait transition method for a quadruped robot to achieve omni-directional static walking. These papers proposed only walking gaits and stability of the walking robot. In [4], Santos et al. introduced path tracking methods using discontinuous gaits. But they used an open loop control algorithm. Therefore, a closed-loop control with camera feedbacks is needed.

This paper addressed a trajectory tracking problem: control a quadruped robot to track a path generated by a path planning algorithm for obstacles avoidance. The control technology for our system is presented in following sequence. First, the server computer receives image from the camera. The obstacles and goal point are marked. Second, the proposed potential functions method is applied and a trajectory from a start point to a goal point is generated. Third, the kinematic model for the quadruped robot is presented. From the modelling, a backstepping controller is designed for the robot to track the planned trajectory generated by the potential functions method. The simulation results are presented to show the effectiveness of the proposed trajectory planning algorithm and tracking controller of the robot for obstacle avoidance.

II. SYSTEM DESCRIPTION AND PATH PLANNING ALGORITHM USING POTENTIAL FUNCTIONS METHOD

A. System description

In this section, the ceiling-mounted camera system configuration is introduced. The system includes a ceiling-mounted camera, server computer, the quadruped robot and obstacles in random positions in Fig. 1.

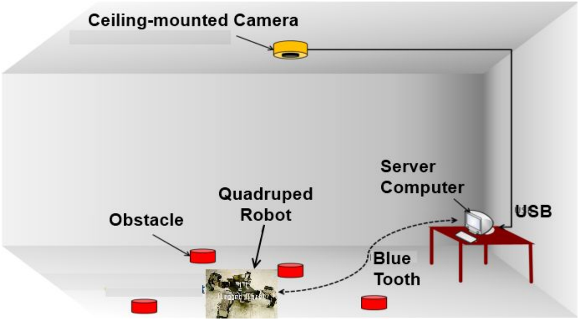

Fig. 1 Configuration of Ceiling-mounted Camera System

The camera is connected to the server computer by USB interface. The server computer receives image from the camera to perform image processing, generates the path and then sends commands to the robot.

B. Potential function

This section describes the use of potential functions in path planning [5]. A potential function $U(\mathbf{p})$ can be viewed as energy. Hence the gradient $\nabla U(\mathbf{p})$ of the potential function can be viewed as force as follows:

$$\nabla U = \left[\partial U / \partial x(\mathbf{p}), \partial U / \partial y(\mathbf{p})\right]^T \tag{1}$$

where $\mathbf{p} = [x\ y]^T$ is a position vector of the robot in the map.

The potential function approach directs the robot as if it is a particle moving in a gradient vector field. Gradients can be intuitively viewed as forces acting on a positively charged particle robot which attached to the negatively charged goal. Obstacles also have a positive charge which forms a repulsive force directing the robot away from obstacles. The combination of repulsive and attractive forces directs the robot from the start point to the goal point while avoiding obstacles. Potential function can be constructed as the sum of attractive and repulsive potentials as follows:

$$U(\mathbf{p}) = U_{att}(\mathbf{p}) + U_{rep}(\mathbf{p}) \qquad (2)$$

where $U_{att}(\mathbf{p})$ is the attractive potential function and $U_{rep}(\mathbf{p})$ is the repulsive potential function.

The attractive potential represents that the robot attract to the goal. The repulsive potential represents that the robot repels from obstacles. The repulsive potential function in terms of individual obstacles can be commonly used the following form.

$$U_{rep_i}(\mathbf{p}) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{d_i(\mathbf{p})} - \frac{1}{Q^*}\right)^2 & , d_i(\mathbf{p}) \le Q^* \\ 0 & , d_i(\mathbf{p}) > Q^* \end{cases} \qquad (3)$$

where $d_i(\mathbf{p})$ is the distance from the robot to obstacle i, Q_i^* is the threshold distance to allow the robot to ignore obstacles and η is the repulsive gain.

Then the repulsive gradient is:

$$\nabla U_{rep_i}(\mathbf{p}) = \begin{cases} \eta\left(\frac{1}{Q^*} - \frac{1}{d_i(\mathbf{p})}\right)\frac{1}{d_i^2(\mathbf{p})}\nabla d_i(\mathbf{p}), & d_i(\mathbf{p}) \le Q^* \\ \\ 0 & , d_i(\mathbf{p}) > Q^* \end{cases} \qquad (4)$$

The total repulsive function and gradient can be calculated as follows:

$$\begin{aligned} U_{rep}(\mathbf{p}) &= \sum_{i=0}^{n} U_{rep_i}(\mathbf{p}) \\ \nabla U_{rep}(\mathbf{p}) &= \sum_{i=0}^{n} \nabla U_{rep_i}(\mathbf{p}) \end{aligned} \qquad (5)$$

The attractive potential can be chosen as follows:

$$U_{att}(\mathbf{p}) = \begin{cases} \frac{1}{2}\zeta d^2(\mathbf{p},\mathbf{p}_{goal}) & , d(\mathbf{p},\mathbf{p}_{goal}) \le d_{goal}^* \\ d_{goal}^*\zeta d(\mathbf{p},\mathbf{p}_{goal}) - \frac{1}{2}\zeta(d_{goal}^*)^2 & , d(\mathbf{p},\mathbf{p}_{goal}) > d_{goal}^* \end{cases} \qquad (6)$$

where $d(\mathbf{p},\mathbf{p}_{goal})$ is the distance from the robot position to the goal position, d_{goal}^* is a threshold distance from the robot position to the goal position where the attractive potential function switch and ζ is the repulsive gain.

And the attractive gradient is:

$$\nabla U_{att}(\mathbf{p}) = \begin{cases} \zeta(\mathbf{p} - \mathbf{p}_{goal}) & , d(\mathbf{p},\mathbf{p}_{goal}) \le d_{goal}^* \\ \frac{d_{goal}^*\zeta(\mathbf{p} - \mathbf{p}_{goal})}{d(\mathbf{p},\mathbf{p}_{goal})} & , d(\mathbf{p},\mathbf{p}_{goal}) > d_{goal}^* \end{cases} \qquad (7)$$

C. Gradient descent

The robot moves from a “high value” state to a “low value” state of the potential function. The vector of the gradient points the direction that the potential function will increase. Therefore, the robot follows a path “downhill” by following the negated gradient of the potential function. Following such a path is called gradient descent. The robot terminates motion when and $\mathbf{p}^*$ is called a critical point. The point $\mathbf{p}^*$ is maximum, minimum or saddle point. We use the second derivative (Hessian matrix) to determine the type of critical point.

The gradient descent algorithm can be defined as:

```
$\mathbf{p}(0) = \mathbf{p}_{start}$
$i = 0$
while ($\|\nabla U(\mathbf{p}_i)\| < \varepsilon$) do
        $\mathbf{p}_{i+1} = \mathbf{p}_i - \alpha \times (\nabla U(\mathbf{p}_i))$
        $i = i + 1$;
end while                                        (8)
```

where the notation $\mathbf{p}_i$ is used to denote the value of $\mathbf{p}$ at the i^{th} iteration and the final path consist of the sequence of iterates $\{\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_i\}$ that is the path for trajectory tracking controller. The value of the scalar α determines the step size of the iteration.

III. Kinematic Modelling of Quadruped Robot

A. Design of quadruped robot

The quadruped robot in this paper consists of one body and four legs. Each leg has three joints. Fig. 2 shows the simplified model of the robot.

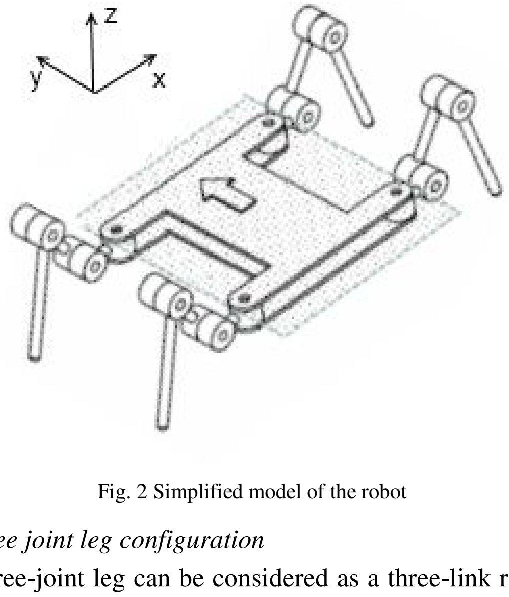

Fig. 2 Simplified model of the robot

B. Three joint leg configuration

A three-joint leg can be considered as a three-link revolute joint manipulator, which is attached to robot body. Therefore, the kinematic modelling is performed by the conventional robotics approaches. The kinematic model is derived according to the Denavit-Hartenberg (DH) convention [6]. In Fig. 3, a representation of a three joint leg is shown with the reference frames and corresponding joint variables.

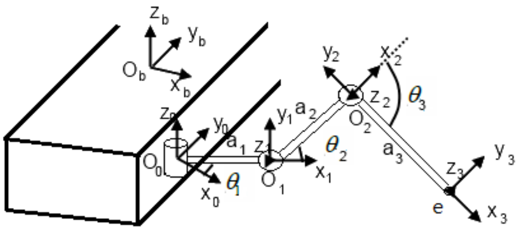

Fig. 3 Three-joint leg

The end-effector position vector of one leg, $\mathbf{p}_e = [x_e\ y_e\ z_e]^T$, can be calculated as follows [6]:

$$\begin{aligned} x_e &= \cos\theta_1(a_1 + a_3\cos(\theta_2+\theta_3) + a_2\cos\theta_2) \\ y_e &= \sin\theta_1(a_1 + a_3\cos(\theta_2+\theta_3) + a_2\cos\theta_2) \qquad (9) \\ z_e &= -a_3\sin(\theta_2+\theta_3) - a_2\sin\theta_2 \end{aligned}$$

where $\mathbf{q} = [\theta_1\ \theta_2\ \theta_3]^T$ is joint angles vector.

The translational velocity vector of the end-effector, $\mathbf{v}_e$, can be expressed as the time derivative of the end-effector position vector.

$$\mathbf{v}_e = \dot{\mathbf{p}}_e = \mathbf{J}_P(\mathbf{q})\dot{\mathbf{q}} = \mathbf{J}_P(\mathbf{q})\left[\dot{\theta}_1\ \dot{\theta}_2\ \dot{\theta}_3\right]^T \qquad (10)$$

where $\mathbf{J}_P = \partial\mathbf{p}_e / \partial\mathbf{q}$ is the analytical Jacobian [6], $\dot{\mathbf{q}} = \left[\dot{\theta}_1\ \dot{\theta}_2\ \dot{\theta}_3\right]^T$ is joint angular velocity vector and $\mathbf{z}_0$, $\mathbf{z}_1$, $\mathbf{z}_2$ are unit vectors as shown in Fig. 3.

IV. Path Tracking Controller Design

A. Path discretization

The path generated by the potential functions method is discretized into appropriate steps. As a simple example, in order for the robot move for a distance of 10.0cm, the motion planner decides to take ten 1.0cm steps. In the execution, the legs start from a specified home position. The desired values are supplied to the robot. The new joint angles are computed by the tracking controller on the robot

B. Tracking controller design

At first, the gait of the robot chosen in this study is the crab gait. The crab gait consists in moving the center of gravity of the body along a trajectory while maintaining the body's orientation constant. The advantages of the crab gait are high stability margin that allows the robot to move in larger direction angle and computation of the joint angles is similar for all legs. A desired tracking controller is designed to track the path obtained from path planning algorithm.

The robot walks in direction with the angle β by moving the end-effector of every leg with the same angle β to a desired point as shown in Fig. 4. The desired point position is calculated as follow:

$$\mathbf{p}_d = \mathbf{p}_{e0} + \mathbf{p}_r \qquad (11)$$

where $\mathbf{p}_d = [x_d\, y_d\ z_d]^T$ is the desired point of end-effector, $\mathbf{p}_r = [r\times\sin\beta, r\times\cos\beta, 0]^T$, $\mathbf{p}_{e0} = [x_{e0}, y_{e0}, z_{e0}]^T$ is the initial position of end-effector and β is then angle of moving and r is the distance of one robot step.

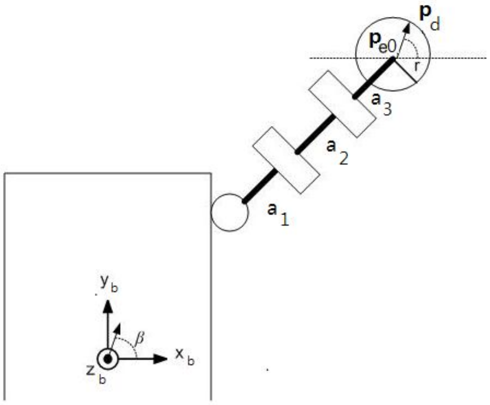

Fig. 4 Crab gait motion for one leg

The operational space error vector between the desired and the actual end-effector is defined as follows:

$$\mathbf{e} = \mathbf{p}_d - \mathbf{p}_e \qquad (12)$$

Then the time derivative of the error vector is:

$$\dot{\mathbf{e}} = \dot{\mathbf{p}}_d - \dot{\mathbf{p}}_e = \dot{\mathbf{p}}_d - \mathbf{J}_P(\mathbf{q})\dot{\mathbf{q}} \qquad (13)$$

In this paper, a controller backstepping method [8] is proposed. The candidate Lyapunov function (clf) is chosen as follows:

$$V(\mathbf{e}) = \frac{1}{2}\mathbf{e}^T\mathbf{K}\mathbf{e} \geq 0 \qquad (14)$$

where K is a symmetric positive definite matrix.

Using Eq. (13), the derivative of Eq. (14) wrt to time is derived as follows:

$$\dot{V} = \mathbf{e}^T\mathbf{K}\dot{\mathbf{e}} = \mathbf{e}^T\mathbf{K}(\dot{\mathbf{p}}_d - \mathbf{J}_P(\mathbf{q})\dot{\mathbf{q}}) \qquad (15)$$

Control input chosen are joint velocity vector, $\dot{\mathbf{q}}$. On the assumption that matrix $\mathbf{J}_P$(q) is square and nonsingular, the control input is chosen as follows:

$$\dot{\mathbf{q}} = \mathbf{J}_P^{-1}(\mathbf{q})(\dot{\mathbf{p}}_d + \mathbf{K}\mathbf{e}) \qquad (16)$$

From Eqs. (15) and (16), we have:

$$\dot{V} = -\mathbf{e}^T\mathbf{K}\mathbf{e} < 0 \qquad (17)$$

The joint velocity vector is designed to make the error converge to zero when $t \to \infty$. In the case of singular $\mathbf{J}_P(\mathbf{q})$. The solution of Eq. (16) can be generalized into

$$\dot{\mathbf{q}} = \mathbf{J}_P^{+}(\mathbf{q})(\dot{\mathbf{p}}_d + \mathbf{K}\mathbf{e}) \qquad (18)$$

where $\mathbf{J}_P^{+}(\mathbf{q}) = (\mathbf{J}_P^{T}(\mathbf{q})\mathbf{J}_P(\mathbf{q}))^{-1}\mathbf{J}_P^{T}(\mathbf{q})$ is a pseudo inverse matrix of $\mathbf{J}_P(\mathbf{q})$.

With the control input of Eq. (18), the operational space error **e** converges to 0 when $t \rightarrow \infty$ [6].

V. Simulation Results

A. The potential function results

In the following simulations, the workspace is within a 10m x 10m and there are some circular obstacles placed in it. The starting position of the robot at point (0, 0) and the goal position at point (8, 8). For the potential functions method, the parameters are chosen as shows in table 1.

TABLE I
POTENTIAL FUNCTIONS METHOD PARAMETERS

Parameters	Description	Unit
η	0.1	
ζ	1	
Q^*	1.5	m
d^*_{goal}	10	m

Fig. 5 shows the trajectory generated by path planning algorithm using potential functions method. The path avoids two obstacles at point (1, 3) and another one at point (5, 7) and reaches goal position at point (8, 8).

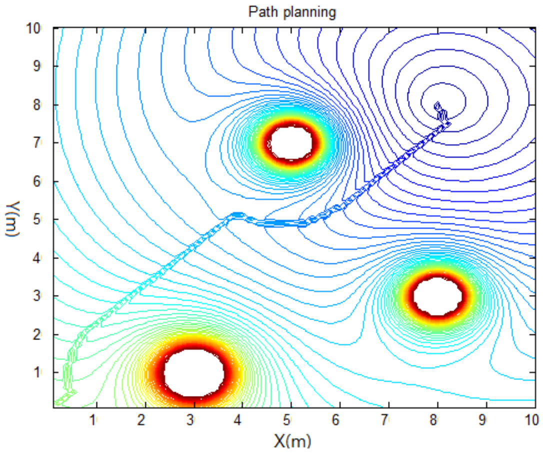

Fig. 5 Path planning algorithm

Fig. 6 shows the potential function in the workspace. There are three maxima at three obstacles and one minimum at the goal point. The robot follows the path to move from a "high value" state to a "low value" state of the potential function.

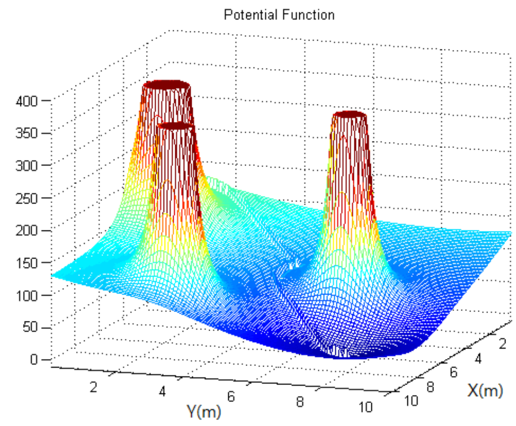

Fig. 6 Potential function

B. The tracking simulation results

1) One path discretization

Fig.7 shows the path generated by the path planning algorithm. The path is discretized into appropriate steps. In this case, a step is the distance of one robot step, r = 2cm. The direction of robot motion of each step, β, is calculated from the path. Then the distance of one step and the direction of each robot step are sent to robot to follow.

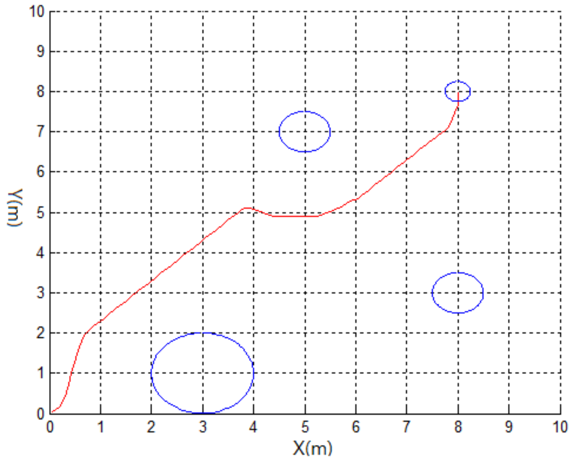

Fig. 7 Path from potential functions method

2) One leg simulation results

For one leg simulation results, the one leg simulation parameters are shown in Table II .

TABLE II
ONE LEG SIMULATION PARAMETERS

Parameters	Description	Unit
β	π/4	*rad*
r	2	*cm*
a_1	2	*cm*
a_2	6	*cm*
a_3	10	*cm*

The initial values are chosen as shows in Table III.

TABLE III
SIMULATION INITIAL VALUES

Parameters	Description	Unit
x_{e0}	3.121	*cm*
y_{e0}	0	*cm*
z_{e0}	-2.829	*cm*
x_d	4.535	*cm*
y_d	1.414	*cm*
z_d	-2.829	*cm*
θ_1	0	*rad*
θ_2	π/4	*rad*
θ_3	-3π/4	*rad*
e_x	1.414	*cm*
e_y	1.414	*cm*
e_z	0	*cm*

Fig. 8 shows a step of one leg from initial point to desired point.

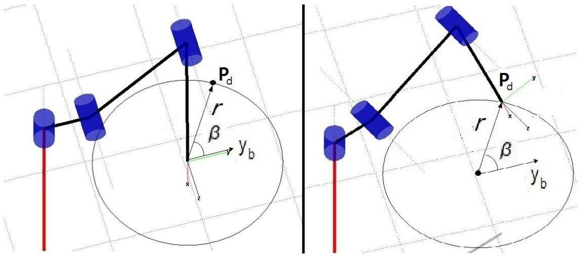

Fig. 8 A step of one leg

Fig. 9 shows the joint angles of the joints of the leg to move one step.

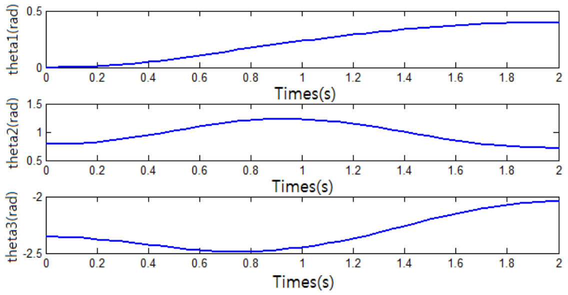

Fig. 9 Joint angles in one step

Fig. 10 shows the operational space error when the robot steps. After 2 seconds, the error converges to zero. The end-effector reaches the desired point Pd.

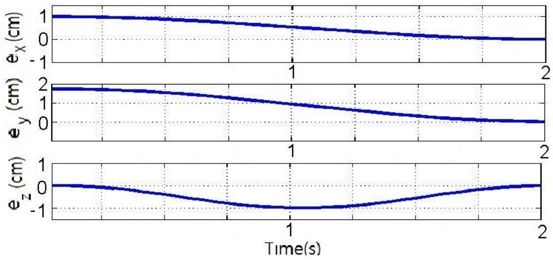

Fig. 10 The operational space error **e**

VI. CONCLUSION

This paper proposed to design a tracking controller for obstacle avoidance of quadruped robot from start point to goal point using a ceiling-mounted camera system method. Ceiling-Mounted Camera system is used for image processing. By the system, the robot and obstacles are recognized by the color recognition method. The kinematic the robot is presented. Potential functions method is proposed as the path planning method to generate the path. Based on the kinematic modelling, the path tracking controller using the backstepping technique is presented to track the path generated by the path planning method. The simulation results of the controller verified the effectiveness of the proposed path planning algorithm using potential functions method and the proposed tracking controller.

ACKNOWLEDGMENTS

This research was supported by a grant from Construction Technology. Innovation Program (CTIP) funded by Ministry of Land, Transportation and Maritime Affairs (MLTM) of Korean government.

REFERENCES

[1] C. Zhang and S. M. Song, “Gaits and Geometry of a Walking Chair for Disabled”, *J. Terramech*, vol. 26, pp. 211-233, Nov. 1989.

[2] S. Hirose and Y. Umetani, “The Basic Consideration on a Feasible Walking Mechanism as a Terrain Vehicle”, *Trans. Soc. Instrument Contr. Engineers*, vol. 15, pp. 298-933, Nov. 1979.

[3] S. Ma, T. Tomiyama and H. Wada, “Omnidirectional Static Walking of a Quadruped Robot”, *IEEE Trans. on Robotics*, vol. 21, pp. 152-161, Apr. 2005.

[4] P. G. D. Santos and M. A. Jimenez, “Path Tracking With Quadruped Walking Machines Using Discontinuous Gaits”, *Computers Elect. Engng*, vol. 21, pp. 383-396, Aug. 1995.

[5] Y. K. Hwang and N. Ahuja, “A Potential Field Approach to Path Planning”, *IEEE Trans. on Robotics and Automation*, vol. 8, pp. 23-32, Feb. 1992.

[6] B. Siciliano, L. Sciavicco, L. Villani and G. Oriolo: *Robotics Modelling, Planning and Control*, 2nd ed., Springer-Verlag, 2009.

[7] R. Fierro and F. L. Lewis, “Control of a Nonholonomic Mobile Robot: Backstepping Kinematics into Dynamics”, *Journal of Robotic Systems*, vol.14, pp. 149-163, Nov. 1997.

[8] Y. Go, X. Yin and A. Bowling, “Navigability of Multi-Legged Robots”, *IEEE/ASME Trans. on Mechatronics*, vol. 11, pp.1-8, Feb. 2006.