Behavior Selection Based Navigation and Obstacle Avoidance Approach Using Visual and Ultrasonic Sensory Information for Quadruped Robots

Kiyotaka Izumi, Maki K. Habib*, Keigo Watanabe and Ryoichi Sato
Department of Advanced Systems Control Engineering
Graduate School of Science and Engineering, Saga University,
Saga, Japan
{izumi, habib, watanabe}@me.saga-u.ac.jp
*Currently at Department of Mechanical Engineering, School of Sciences and Engineering, the American University in Cairo, Cairo, Egypt. Email: maki@aucegypt.edu

Abstract: A robot functioning in an environment may exhibit various forms of behavior emerge from the interaction with its environment through sense, control and plan activities. Hence, this paper introduces a behaviour selection based navigation and obstacle avoidance algorithm with effective method for adapting robotic behavior according to the environment conditions and the navigated terrain. The developed algorithm enable the robot to select the suitable behavior in real-time to avoid obstacles based on sensory information through visual and ultrasonic sensors utilizing the robot’s ability to step over obstacles, and move between surfaces of different heights. In addition, it allows the robot to react in appropriate manner to the changing conditions either by fine-tuning of behaviors or by selecting different set of behaviors to increase the efficiency of the robot over time. The presented approach has been demonstrated on quadruped robot in several different experimental environments and the paper provides an analysis of its performance.

Keywords: Legged robot, Quadruped robots, Behaviors selection, Robot navigation, Obstacle avoidance, Sensors.

1. Introduction

A large number of robots have been developed to work in critical environment, such as, space, ocean bottom, power plants, as well as, in the fields of clinical medicine, hazard prevention, etc., and researchers continue to design robots with greater capabilities to perform more challenging and comprehensive tasks (Hirose et al., 1986; Ooka et al., 1986; Cruse et al., 1994; Chen et al., 2002a; Habib, 2003a). There are many ways for a robot to move across a solid surface in which wheels, crawlers, and legs were common options for the available robots. The application fields of such robots are naturally restricted, depending on the condition of the ground.

Wheeled mobile robots are mechanically simple, easy to construct, easy to implement a controller, dynamically stable in general, and they are ideal for operation on level and hard surfaces. When the surface is rough and has projections and depressions with dimensions that are greater than the diameter of the wheel or when the surface is soft, resistance to the movement increases drastically and their function as transport machines is almost lost, which leads to poor performance. The crawler type locomotion mechanisms have traverse ability higher than that of the wheel, but its control is hard and the dead-reckoning is difficult to realize, though it is possible to move on different terrains.

In order to have good mobility over uneven and rough terrain a legged robot seems to be a good solution because legged locomotion is mechanically superior to wheeled or tracked locomotion over a variety of soil conditions and certainly superior for crossing obstacles. The path of the legged machine can be (partially) decoupled from the sequence of footholds, allowing a higher degree of mobility. This can be especially useful in narrow surroundings or terrain with discrete footholds (Raibert, 1986; Hirose, 2001). However, creating and controlling a legged machine that is powerful enough, but still light enough is very difficult. Legged robots are usually slower and have a lower load/power ratio with respect to wheeled robot. Autonomous legged robots have distinct control issues that must be addressed. These problems are amplified when the robot is small with an on-board controller that is purposely simple to accommodate weight and expense restrictions. The kinematics and dynamics of legged robots are nonlinear, while robot parameters, such as center of mass position, amount of payload, etc. are not known exactly and might also vary (Nishikawa et al., 1998). In addition, it is difficult to estimate states of the system (Pugh et al., 1990). The system might be unstable without control, and
the goal of keeping balance is difficult to be decomposed into actuator commands. A legged system has a lot of degrees of freedom in which a high motion performance and ground adaptation ability on irregular terrain can be demonstrated. In order to allow a completely decoupled motion over irregular terrain, at least three degrees of freedom per leg are required. Two joints would be enough to place the foot in any desired position, and with the third joint, the robot can climb over much larger obstacles relative to its size and also can climb a slippery hill that a leg with two joints cannot perform. But, this will result in using 12 actuators for a four-legged robot, which yields to increase weight and control complexity compared to six actuators for a traditional industrial manipulator (Waldron et al., 1984). Contact forces, in general, only allow pushing the feet into the surface, not pulling. This directly limits the total downwards acceleration that can be applied to a walking machine. This initiates a challenge to investigate the technical problems involved in the realization of a robot that uses legs to navigate in difficult, partially unstructured and unknown environments.

Autonomous robots operating in an unknown and uncertain environment have to cope with dynamic changes in the environment. Hence, enable a robot to navigate successfully to goals while avoiding both static and dynamic obstacles is a challenging problem for mobile robots in general and for legged robots in specific. There is a large body of work devoted to the navigation of wheeled-mobile robots. Some common approaches are odometry, inertial navigation and landmark navigation. The progress made so far in the design of legged robots has been mostly in the areas of leg coordination, gait control, stability, incorporation of various types of sensors, etc. But what is missing in most of these robots is some perception-based high-level control that would permit a robot to operate with a measure of autonomy. Legged mobile robots do not require avoiding all obstacles by altering the path in the obstacle avoidance task as in the case of wheeled and crawler robots. Because legged robots have the ability to step over and onto obstacles in their path depending on the obstacle configuration and the current state of the robot, they are uniquely suited to overcoming these difficulties. However, many existing navigation planning methods fail to consider these additional capabilities, because they were primarily designed for wheeled mobile robots.

The navigability of an autonomous multi-legged system is a crucial element of its overall capabilities (Go et al., 2006). The design of algorithms to compute robust goal-directed navigation strategies for legged robots is an important area of research. The basic difference between path planning for legged robots and mobile robots is that for mobile robots the path is continuous, while for legged robots it is not. In addition, the ability to follow a particular trajectory also depends on the design of the legged robot, i.e., a path planning method that may work for one type of legged robot may not work efficiently for all other types of legged robots. Navigation planning for legged robots via foot placement planning has enabled several walking type robots to traverse various environments autonomously. The available planning approaches for legged robots can be classified based on how much details are considered during the planning process. One category of planning approaches uses every detail in solving the navigation problem. This involves solving a giant motion planning problem for all degrees of freedom of the robot (Hauser et al., 2005; Kallmann et al., 2004; Kuffner et al., 2002). However, the high computational cost becomes a critical problem for locomotion. Another category of planning systems have used local planning that allows the robot to adjust its gait locally in response to the type of terrain (Bares & Wettergreen, 1999; Hodoshima et al., 2004; Krotkov et al., 1991). In addition, another category of planning approaches ignores all the details of the legs, and treats the robot as a wheeled robot than can be steered within its environment. In general, global navigation strategies for mobile robots aim to search for a collision-free path in a 2D environment. These techniques have been applied to legged robots, resulting in conservative global navigation strategies. However, this always forces the robot to go around obstacles rather than using the ability to traverse obstacles by stepping over or onto them. Other approaches build action models trying to simplify the planning problem. One action model uses straight sequences of footsteps to the edges of obstacles similar to a visibility graph, combined with turning in place and stepping-over actions to cross through obstacle-filled environments (Chenstunt et al, 2004).

Biological systems have a tightly integrated action perception cycle. Hence, for walking robots, to realize their full potential, distal environment sensing must be tightly integrated with the walking cycle. Distal sensing is crucial to allow anticipatory gait adjustment to accommodate varying terrain. Close coupling of the visual and locomotor cycle can lead to rapid, adaptive adjustment of the robot (Lewis, 2002). This problem is even more difficult when the robot is unable to generate accurate global models of the obstacles in its environment. Determining an optimal navigation policy without this information can be difficult or impossible. A legged mobile robot is a free roving collection of functions primarily designed to reach a target location. Equipping robots with more sensors increases the quantity and reliability of information the robot can extract from its environment to support robot’s intelligent behavior (Ferrell, 1994). In order to facilitate flexible obstacle detection and avoidance techniques, it is necessary to acquire the 3-dimensional (3D) information about the surrounding environment. Generally, 3D information is acquired through external sensors, such as binocular cameras, ultrasonic sensors (Ohya et al., 1997), laser range finders, etc. However, a high computational
cost is required to analyze 3D information because the binocular camera needs to process two frames from two cameras (Okada et al., 1999, Okada et al., 2003). In addition, although the ultrasonic sensor can accurately measure the distance to an object, there is a difficult problem in determining the azimuth. Therefore, it remains a challenging task to build a robust real-time obstacle avoidance system for a robot using vision data.

2. Quadruped Robot and Behavior based Solution for Obstacle Avoidance

It is well known that the autonomous walking of a quadruped robot depends on not only the implementation of the locomotion but also the navigation control. Although there are a lot of works published on the implementation of the locomotion in the rough terrain for quadruped robots, few works can be recognized on the adaptive navigation in a cluttered environment. In this paper, a quadruped walking robot TITAN-VIII (Arikawa & Hirose, 1996; Hirose & Arikawa, 2000) has been used as a platform to test and demonstrate the development and implementation of a behavior selection based obstacle avoidance algorithm (See Figure 1.(a)).

TITAN VIII is a walking machine that has four modular legs. The leg mechanism is composed of a planar 2 degrees of freedom link-wire mechanism and a rotating mechanism which rotates the planar. Hence, this leg mechanism has 3DOF. One of the characteristics of this leg is usage of wire and pulley driving system within the leg. The feet of TITAN VIII can be used also as wheels in order to achieve faster motion on flat surfaces. TITAN VIII walks in a walking posture jutting out its legs to each side. This is standard walking posture of TITAN VIII. In such a walking posture, good energy efficiency can be achieved (Arikawa & Hirose, 1996; Hirose & Arikawa, 2000). An ART-based Fuzzy controller for the adaptive navigation of a quadruped robot has been developed (Chen et al., 2002b), and then different type of sensors has been integrated with the robot to support its navigation (Yamaguchi et al., 2002a; Yamaguchi et al., 2002b). Visual and ultrasonic sensors have been integrated with the quadruped robot. The aim of these sensors is to detect and acquire 3D information of obstacles along the path of the robot. The first sensor was the USB camera. The camera is mounted on the front of the robot body. The camera was fixed at the front side of the robot body (See Figure 1.(b)). In addition, three ultrasonic sensors have been used and configured at the tip of each of the front legs (See Figure 1.(c)).

The obstacle is roughly measured by processing the image acquired through the USB camera, and the ultrasonic sensors are used to complement the visual information in relation to obstacle and to perform the selection of the suitable actions at the right time. In order to facilitate this process, a set of behavioral actions is decided, designed and implemented. Currently, the main actions in the list include: default, detour, striding, and climbing-over obstacles actions. Thus, fusing information through the use of different and multiple sensors separately according to the situation and obtaining the information necessary for obstacle avoidance can support the right decision to select the suitable set of actions to avoid obstacles in real-time.

3. Sensory Information and Obstacle Measurement

The size of an obstacle within the environment of the robot is measured by ultrasonic sensors and a USB CCD camera. The maximum measurable distance of the ultrasonic sensor is about 600 [mm]. The image resolution of the camera is set to 320 × 240 [pixel], and the specification of the camera is listed in Table 1. The camera is mounted on the front of the robot body.

| Sizes              | W40.4 × D57 × H79 [mm] |
|--------------------|------------------------|
| Weight             | 118 [g]                |
| Image reception    | 1/4 [in] CMOS sensor   |
| device             |                        |
| Maximum resolution | 640 × 480 [dot]        |
| Frame ratio        | 15 [fps] (VGA)         |
|                    | 30 [fps] (less than 320 [dot]) |
| The number of colors | 16.77 million (24 [bit]) |

Table 1. Specification of USB camera
3.1 3-Dimensional Measurements by Single Camera

The measurement model between a camera and an obstacle in top view is shown in Fig. 2. The parameter definitions relevant to the top view are listed in Table 2.

![Fig. 2. Camera model in top view](image)

| Symbols | Physical meaning |
|---------|------------------|
| L       | Distance between an obstacle and the camera |
| W       | Acquisition range of the camera with distance L |
| w       | Width of an obstacle |
| h       | Height of an obstacle in image coordinate |
| α       | Horizontal projection angle of the camera |
| X       | Maximum image width |
| Y       | Maximum image height |
| x       | The x axis of the image reference frame |
| y       | The y axis of the image reference frame |

Table. 2. Physical parameters of camera model in the top view

The obstacle width \( W \) is calculated by using parameters in the image, such that

\[
W = \frac{W}{X} w
\]  \hspace{1cm} (1)

where

\[
W = 2L \tan \frac{α}{2}
\]  \hspace{1cm} (2)

In an exploratory experiment, the acquisition range \( W \) became 220 [mm] when the distance \( L \) was set to 300 [mm]. Therefore, the projection angle \( α \) was set to 40 [deg].

Next, parameters in a vertical direction are defined as listed in Table 3 and the corresponding side view is shown in Fig. 3.

The obstacle height \( H \) is calculated by using parameters defined for the vertical direction, such that

\[
H = \frac{2H_e}{Y} h
\]  \hspace{1cm} (3)

where \( H_e \) is given by

\[
H_e = L \tan \frac{β}{2}
\]  \hspace{1cm} (4)

and \( L \) is given by

\[
L = \frac{L}{\tan \frac{β}{2}}
\]  \hspace{1cm} (5)

In an exploratory experiment, the acquisition range \( H_e \) became 80 [mm] when the distance \( L \) was set to 300 [mm]. Therefore, the projection angle \( β \) was set to 30 [deg].

If the obstacle shape is assumed to be a rectangular parallelepiped, then the obstacle depth can be obtained by a perspective method. The perspective is the art of making some objects or people in a picture look further away than others. The concept of perspective is shown in Fig. 4, where \( S_1 \) denotes the area of front surface for the object, \( S_0 \) denotes the area of rear surface for the object and \( Z_o \) denotes the obstacle depth. The obstacle depth is given by

\[
Z_o = \left( \frac{S_1}{S_0} - 1 \right) L
\]  \hspace{1cm} (6)
3.2. Image Processing

The raw colored image is first converted into the shade (or gray scale) image and further converted into the monochrome image by image binarization. Then, the 3D size information of obstacle is calculated based on the perceived number of surfaces of the obstacle. The flow of this process is illustrated in Fig. 5.

1. Case of one surface detection

When the acquired image has only the front surface of an obstacle as shown in Fig. 6, the width \( w \) and the height \( h \) of the obstacle front surface are given by

\[
\begin{align*}
w &= x_2 - x_1 \\
h &= y_2 - y_1
\end{align*}
\]

where the image point \( (x_1, y_1) \) is for the apex \( P_1 \) and similarly \( (x_2, y_2) \) is for the apex \( P_2 \).

2. Case of two surface detection

When the acquired image includes the front and top surfaces as shown in Fig. 7, the width \( w \) and the height \( h \) of the obstacle front surface are given by

\[
\begin{align*}
w &= x_3 - x_2 \\
h &= y_3 - y_2
\end{align*}
\]

using the image coordinates for apexes \( P_1, P_3 \) and \( P_2 \). In this case, the vanishing point \( V(y_v, x_v) \) is the point that the straight line passing through points \( P_1 \) and \( P_2 \) intersects the straight line passing through points \( V \) and \( P_1 \). The width of the rear surface is the distance between the points \( P_2 \) and \( P_3 \). Hence, the area \( S_o \) of the rear surface is obtained using the calculated from the height and the width of the rear surface.

Then, the depth \( Z_o \) of the obstacle is calculated according to Eq. (6).

3. Case of three surface detection

When the acquired image includes the front, the top and the side surfaces as shown in Fig. 8, the width \( w \) and the height \( h \) of the obstacle front surface are given by

\[
\begin{align*}
w &= x_4 - x_1 \\
h &= y_4 - y_1
\end{align*}
\]

using the image coordinates for apexes \( P_1, P_4 \) and \( P_2 \). In this case, the height of the rear surface is the distance between the points \( P_2 \) and \( P_4 \), and the width of the rear surface is the distance between the points \( P_2 \) and \( P_3 \).
Fig. 9. Image of three surfaces with the lacking of parts

Let us consider a situation where a part of the obstacle is not reflected in the acquired image, which is shown in Fig. 9.

In this situation, the width and the height of the front surface are defined as

\[ w = x_2 - x_1 \]  \hfill (13)

\[ h = y_1 - y_6 \]  \hfill (14)

In this case, the vanishing point \( V \) is the point that the straight line passing through the points \( P_1 \) and \( P_3 \) intersects the straight line passing through the points \( P_2 \) and \( P_4 \). In this situation, the width of the rear surface for the obstacle is defined as the distance between the point \( P_2 \) and the intersection point at which the horizontal line passing through the \( x \) coordinate of point 2 and parallel to the line passing through the points \( P_1 \) and \( P_3 \) intersects the straight line passing through the points \( P_2 \) and \( P_4 \). The height of the rear surface is the distance between points \( P_2 \) and \( P_4 \).

4. Design of Actions

Primitive actions with different levels of abstraction have been designed and implemented to support formulating the behavior of a robot using a combination of these actions. In general, the description of an action set can have the following form,

\[ \mathcal{A} = \{ A_i | i = 1, 2, \ldots n \} \]  \hfill (15)

where \( A_i \) denotes the symbol of \( i \)th action and \( n \) denotes the number of actions. The action \( A_i \) consists of the series of parameters to move the robot, such as

\[ A_i = \{ c_{i1}, c_{i2}, \ldots c_{im} \} \]  \hfill (16)

where \( c_{ij} \) denotes \( j \)th movement parameter vector and \( m \) denotes the number of movement parameters to perform the action \( A_i \).

In this research, the gait of the quadruped robot is selected to be an intermittent crawl gait (Tsukakoshi et al., 1996). The leg order in one cycle is 4th leg, 2nd leg, 3rd leg and 1st leg. In this paper, the \( j \)th movement parameter vector \( c_j \) composes the traveling distance and the height of swing leg, such as

\[ c_j = [Ap, Az] \]

\[ Ap = [\Delta x, \Delta y, \Delta z] \]

\[ Az = [\Delta z_x, \Delta z_y, \Delta z_z] \]  \hfill (17)

where \( \Delta x \), \( \Delta y \) and \( \Delta z \) are the translational distance for each direction, and \( \Delta s_z \) denotes the upward distance of \( j \)th leg when the \( j \)th leg becomes the swing leg from \( j \)th leg. In addition, \( \Delta s_z \) denotes the downward distance of \( j \)th leg, when the \( j \)th leg becomes the support leg from the swing leg.

The following subsections describe the core actions, which enable the robot to avoid obstacles at different circumstances.

4.1. Default Action: \( A_i \)

The default action \( A_i \) is for a straight translation. Here, we define \( A_i \) as

\[ A_i = \{ c_i \} \]  \hfill (18)

\[ c_i = [200 0 50 50 50 50 50 50 50 50 50] \]  \hfill (19)

where the unit of \( c_i \) element is [mm].

4.2. Striding Action: \( A_i \)

The process sequence of the developed striding action \( A_i \) is listed as follows:

1. The robot approaches an obstacle up to the distance in which the robot can avoid the obstacle,
2. Rear legs of the robot approach the obstacle, and then
3. Front legs of the robot approach the obstacle, then
4. Rear legs of the robot approach the obstacle.

4.3. Climbing-over Action: \( A_i \)

The process sequence of the developed climbing-over action \( A_i \) is listed below:

1. The robot approaches an obstacle up to the distance in which the robot can avoid the obstacle,
2. Rear legs of the robot approach the obstacle,
3. Front legs of the robot approach the obstacle,
4. Rear legs of the robot climb over the obstacle,
5. Front legs of the robot get off the obstacle, and then
6. Rear legs of the robot get off the obstacle.

4.4 Detour Action: \( A_i \)

The detour action \( A_i \) enables the robot to move around the obstacle by generating a crank like path. The process sequence of \( A_i \) action is listed as follows:

1. The robot approaches an obstacle up to the distance in which the robot can avoid it,
2. The robot moves to side as the crab walking up to the distance in which the robot can avoid the obstacle, and then
3. The robot moves forward up to the distance in which the robot passes the obstacle.
5. Action Selection

Autonomous intelligent systems are characterized by the fact that they select one from a set of equivalent action alternatives in a given situation as appropriate (Habib, 2003). Hence, it is important to develop a navigation strategy with efficient action selection mechanism. Currently, the authors have implemented a rule-based logical flow to support the selection of a suitable action according to perceived relation between the robot and the detected obstacle. Brief listing of the rule-based logical flow is shown below,

\[
\begin{aligned}
&\text{if } (L \geq 600 \text{ [mm]}) \} \\
&\quad \text{The robot selects the action } A_1; \\
&\text{else if } (H \geq 320 \text{ [mm]}) \} \\
&\quad \text{The robot selects the action } A_2; \\
&\text{else if } (Z_o \geq 180 \text{ [mm]}) \} \\
&\quad \text{The robot selects the action } A_3; \\
&\text{else} \\
&\quad \text{The robot selects the action } A_4;
\end{aligned}
\]

where \( L \) is the distance between the robot and the obstacle; \( H \) is the obstacle height, and \( Z_o \) is the obstacle depth.

6. Experimental Results

Experiments have been conducted to prove that the designed set of action modules enables the robot to recognize and avoid obstacles in real-time under different situations. The selected gait of the robot during the experiments was an intermittent crawl gait. In addition, a unit cycle has been used to illustrate the total time required to perform each action. A unit cycle represents the time required for moving each of the four legs of the robot once according to the pattern of the selected gait. However, the total number of cycles depends on the environment and the type of the available obstacles. The following subsections highlight the experimental results and achievements.

6.1. Striding Action

This experiment aims to demonstrate a striding action. The observed robot behavior was described by the following set of actions,

\[ A = \{ A_1, A_2, A_3, A_4, A_1, A_1, A_1, A_1 \} \]

The results obtained through this experiment illustrate the ability of the robot to perform the striding action successfully. Figure 10 shows the tips of left side legs of the robot, i.e., the 1st and 3rd legs, didn't have any contact with the obstacle during the avoidance. Figure 11 shows the \( z \) positions for the tip of the 1st leg. Additionally, Action \( A_1 \) is performed with 1 cycle, and the total number of \( A_1 \) action as illustrated in this behavior is 9; Action \( A_2 \) is performed with 6 cycles; and thus, the total number of cycles is \( 1 + 6 + 2 = 15 \) cycles.

6.2. Climbing-over Action

The climbing-over action has been demonstrated in this experiment, and the observed robot behavior was described by the following set of actions,

\[ A = \{ A_1, A_1, A_1, A_1, A_3, A_4, A_1 \} \]

The robot has performed the climbing-over action successfully. The experimental results are illustrated in Figure 12, in which it also highlights the case where the tips of left side legs of the robot didn’t have any contact with the obstacle at anytime during swing phase. Figure 13 shows the \( z \) positions for the tip of the 3rd leg. The results illustrate a contact point between the obstacle and the tip of the robot leg during a support phase while climbing-over. The time performance for executing the set of actions above as illustrated by Figure 13 is shown below,

Action \( A_1 \) is performed with 1 cycle, and the total number of \( A_1 \) action as illustrated in this behavior is 6; Action \( A_2 \) is performed with 10 cycles; and thus, the total number of cycles is \( 6 + 10 + 2 = 18 \) cycles.
6.3 Detour Action

The observed robot behavior during the execution of the detour action was described by the following set of actions,

\[ A = \{ A_1 A_4 A_1 A_4 A_1 A_4 A_1 A_4 \} \]

The experimental result of a detour action is shown in Fig. 14. The results show none of the robot’s legs tips did have any contact with the obstacle during the avoidance. The time performance for executing the set of actions as stated above is,

- Action \( A_1 \) is performed with 1 cycle, and the total number of \( A_1 \) action as illustrated in this behavior is 8;
- Action \( A_4 \) is performed with 15 cycles; and

Thus, the total number of cycles is \( 5 + 15 + 3 = 23 \) cycles.

6.4 Detour and Striding Actions

An experiment was demonstrated to verify the effectiveness of the present approach in case of having multiple obstacles crossing the path of the robot. The successful experimental results with two obstacles are shown in Fig. 15 and Fig. 16. The set of actions that has been selected to formulate the intended behavior is shown below,

\[ A = \{ A_1 A_4 A_1 A_4 A_1 A_4 A_1 A_4 A_2 A_1 A_4 \} \]

During this behavior, the robot approaches the first obstacle with action \( A_1 \). Then, the robot initiates the avoidance of the first obstacle using action \( A_4 \). After clearing the first obstacle, and while the robot approaches the second obstacle using a number of \( A_1 \) actions, the robot selects to avoid it by activating the action \( A_4 \). The time performance for executing the set of actions above as illustrated by Figure 16 is shown below,

- Action \( A_1 \) is performed with 1 cycle, and the total number of \( A_1 \) action as illustrated in this behavior is 9;
Action $A_1$ is performed with 16 cycles; and Action $A_4$ is performed with 7 cycles. Thus, the total number of cycles is $2 + 16 + 5 + 7 + 2 = 32$ cycles.

6.5 Detour and Climbing-over Action
The set of actions that has been selected to formulate this behavior is as follow,

$$\{A_1, A_4, A_1, A_4, A_1, A_4, A_1, A_4, A_1, A_2\}$$

Successful experimental results have been achieved to avoid two obstacles and it is shown in Fig. 17 and Fig. 18 respectively. During the execution of this behavior, first, the robot activates the default action $A_1$ and then it selects the detour action $A_4$ for avoidance. After taking the detour action and avoiding the first obstacle, the robot activates again the default action $A_1$ to proceed with the forward trajectory. During the walking course, the robot detects the second trajectory and according to the situation, it selects the climbing-over action $A_2$ to avoid the second obstacle. Finally, the robot activates again the default action to proceed with its trajectory. The time performance for executing the set of actions above as illustrated by Figure 18 is shown below,

Action $A_1$ is performed with 1 cycle, and the total number of $A_1$ action as illustrated in this behavior is 12; Action $A_4$ is performed with 19 cycles; and Action $A_2$ is performed with 9 cycles. Thus, the total number of cycles is $4 + 19 + 5 + 9 + 3 = 40$ cycles.

7. Conclusions
This paper presented a robust approach to the design of a set of behavioral actions with a strategy that adapt robot actions to the local terrain during legged navigation planning. The developed navigation and obstacle avoidance algorithm has the capability to use a combination of primitive actions to formulate a set of high level behaviors, which supports quadruped robots navigation. The developed algorithm enable the robot to select the suitable behavior in real-time to avoid obstacles based on sensory information through visual and ultrasonic sensors utilizing the robot’s ability to step over obstacles, and move between surfaces of different heights. These adaptive action models have been applied to a quadruped robot with very promising results, allowing these robots to successfully and reliably navigate environments.

7.1 Future Work
Intelligent systems should exhibit emergence property that is not designed into any of its individual sub-components. In order to make these systems adaptable to various situations and type of goals to be pursued in the world, it is necessary to dynamically select behaviors and to change their respective priority to make the system behave appropriately according to the situations it encounters in the real world.

7. Conclusions
This paper presented a robust approach to the design of a set of behavioral actions with a strategy that adapt robot actions to the local terrain during legged navigation planning. The developed navigation and obstacle avoidance algorithm has the capability to use a combination of primitive actions to formulate a set of high level behaviors, which supports quadruped robots navigation. The developed algorithm enable the robot to select the suitable behavior in real-time to avoid obstacles based on sensory information through visual and ultrasonic sensors utilizing the robot’s ability to step over obstacles, and move between surfaces of different heights. These adaptive action models have been applied to a quadruped robot with very promising results, allowing these robots to successfully and reliably navigate environments.

8. References
Arikawa, K. & Hirose, S. (1996). Development of Quadruped Walking Robot TITAN-VIII, *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’96)*, Vol. 1, pp. 208-214, 1996.
Bares J.E., Wettergreen, D. S. (1999). Dante ii: Technical description, results, and lessons learned. *International Journal of Robotics Research*, Vol. 18, No. 7, pp. 621–649, July 1999.
Chen, X., Watanabe, K., Kiguchi, K., Izumi, K. (2002a). Path Tracking Based on Closed-Loop Control for a
Hirose, S., Arikawa, K. (2000): Coupled and Decoupled

388

Cruse, H., Bartling, Ch., Cymbalyuk, G., Dean, J. and

Kanehiro, F., Yoshimi, T., Kajita, S., Morisawa, M.,

Hauser, K., Bretl, T., Latombe, J.-C. (2005). Non-gaited

Ferrell, C. (1994). Robust and Adaptive Locomotion of an

Kallmann, M., Bargmann, R., Matarić, M. (2004). Planning the sequencing of movement primitives. Proc. of the International Conference on Simulation of Adaptive Behavior, 2004.

Kanehiro, F., Yoshimi, T., Kajita, S., Morisawa, M., Fujiwara, K., Harada, Kaneko, K., Hirukawa, H., Tomita, F. (2005). Whole body locomotion planning of humanoid robots based on a 3d grid map. Proc. of the IEEE International Conference on Robotics and Automation, Barcelona, Spain, April 2005.

Krotkov, E., Bares, J., Kanade, T., Mitchell, T., Simmons, R. Whittaker, W. R. L. (1991). Ambler: a six-legged planetary rover. Proc. of the Fifth International Conference on Advanced Robotics, 1991, Robots in Unstructured Environments (ICAR '91), volume 1, pp. 717–722, June 1991.

Kuffner, J., Kagami, S., Nishiwaki, K., Inaba, M. Inoue, H. (2002). Dynamically-stable motion planning for humanoid robots. Autonomous Robots, Vol. 12, No.1, pp.105–118, January 2002.

Lewis, M. A. (2002). Detecting Surface Features during Locomotion using Optic Flow. Proc. of International Conference of Robotics and Automation (ICRA'2002), Washington DC, pp. 305 – 310, 2002.

Nishikawa, N., Murakami, T., Ohnishi, K. (1998). An Approach to Stable Motion Control of Biped Robot with Unknown Load by Torque Estimator. Proc. of International Workshop on Advanced Motion Control, pp. 82-87, 1998.

Ohy, A., Kosaka, A., Kak, A. (1997). Vision-Based Navigation of Mobile Robot with Obstacle Avoidance by Single Camera Vision and Ultrasonic Sensing. Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'97), Vol. 2, pp.704-711, 1997

Okada, K., Kagami, S., Inaba, M., Inoue, H. (1999). Vision-based Action Control of Quadruped Legged Robot JROB-1, Proc. of 9th International Conference on Advanced Robotics (ICAR'99), pp. 451-456, 1999

Okada, K., Inaba, M., Inoue, H. (2003). Integration of Real-time Binocular Stereo Vision and Whole Body Information for Dynamic Walking Navigation of Humanoid Robot, Proc. of International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI'03), pp. 131-136, 2003

Ooka, A., Ogi, K., Takemoto, Y., Okamoto, K. and Yoshida (1986). Intelligent robot system II. Proc. of International Conference on Robotics Research, pp.341-347, 1986.

Pugh, D. R., Ribble, E. A., Vohnout, V. J., Bihari, E. E., Walliser, T. M., Patterson, M. R., Waldron, K. J. (1990). Technical Description of the Adaptive Suspension Vehicle. International Journal of Robotics Research, Vol. 9, No. 2, pp. 24-42, 1990.

Raibert, M. H. (1986). Legged robots that balance. The MIT Press, 1986.

Tsukakoshi, H., Hirose, S., Yoneda, K. (1996). Maneuvering Operations of the Quadruped Walking Robot on the Slope, Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Vol. 2, pp. 863-869, 1996

Waldron, K. J., Vohnout, V. J., Perry, A., McGhee, R. B. (1984). Configuration of the Adaptive Suspension Vehicle. International Journal of Robotics Research, Vol. 3, No. 2, pp. 37-48, 1984.

Yamaguchi, T., Watanabe, K., Izumi, K., Kiguchi, K. (2002a). Acquisition of An Obstacle Avoiding Path in Quadruped Robots, Proc. of Joint 1st International Conference on Soft Computing and Intelligent Systems and 4th International Symposium on Advanced Intelligent Systems(SCIS & ISIS 2002), 2002.

Yamaguchi, T., Watanabe, K., Kiguchi, K., Izumi, K. (2002b). Obstacle Avoidance Strategy for A 4-legged Robot by Getting-over and Striding, Proc. of the 4th Asian Control Conference, pp. 1438-1443, 2002.