Rescue Robot Navigation: Static Stability Estimation in Random Step Environment

Conference Paper · November 2008

DOI: 10.1007/978-3-540-89076-8_30 · Source: DBLP

5 authors, including:

Evgeni Magid
Kazan (Volga Region) Federal University
89 PUBLICATIONS **505** CITATIONS

SEE PROFILE

Rescue Robot Navigation : Static Stability Estimation in Random Step Environment

Evgeni Magid, Kentaro Ozawa, Takashi Tsubouchi[1]
Eiji Koyanagi, and Tomoaki Yoshida[2]

[1] ROBOKEN - Intelligent Robot Laboratory
University of Tsukuba, Japan
{evgeni,ozw,tsubo}@roboken.esys.tsukuba.ac.jp
[2] Future Robotics Technology Center
Chiba Institute of Technology, Japan

Abstract. Rescue robotics is the application of robotics to the search and rescue domain. The goal of rescue robotics is to extend the capabilities of human rescuers while also increasing their safety. During the rescue mission the mobile robot is deployed on the site, while the human operator is monitoring the robot's activities and giving the orders from a safe place. Thus the operator can not see the robot and the environment and a decision on the robot's path selection becomes very hard. Our goal is to provide a kind of automatic "pilot system" to propose an operator a good direction or several options to traverse the environment, taking into account the robot's static and dynamic properties. In this paper we present an algorithm for estimating the posture of the robot in a specific configuration from the static equilibrium point of view. The results obtained by the simulator agree with our prior expectations and were successfully confirmed by the set of experiments with a real robot.

1 Introduction

A long standing goal of mobile robotics has been to allow robots to work in environments unreachable or too hazardous to risk human lives. Urban search and rescue is one of the most hazardous environments imaginable with victims often buried in unreachable locations. Rescue robotics is the application of robotics to the search and rescue domain. The goal of rescue robotics is to extend the capabilities of human rescuers while also increasing their safety. In particular, the inside of severe earthquake stricken buildings or underground area should be investigated in advance of manned rescue operation in order to avoid risk of suffering from secondary disaster. During the rescue mission the mobile robot is deployed on the rescue site, while the human tele-operator is monitoring the robot's activities and giving the orders from a safe place outside of the site (fig.1(a)). The system consists of two subsystems: robot control system and remote operation station. They are connected with a wireless LAN.

Currently rescue robots are operated manually by human operators. An operator can not see the robot and the environment. He/she can only use the data

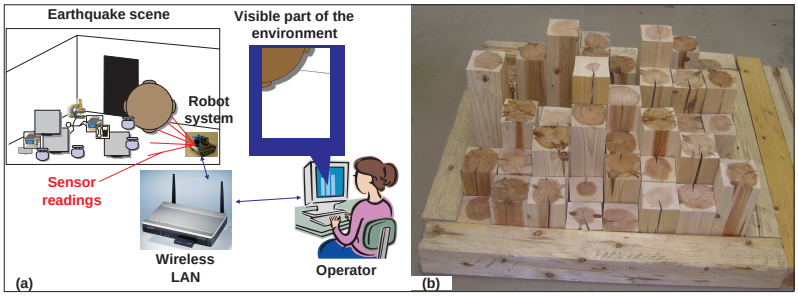

Fig. 1. (a)Standard framework (b) Random Step Environment (RSE) example

obtained by the robot's sensors. With that information only, having no grasp of the environment, it is very difficult for the human to operate the robot. In the case of an on-site operator, which stays inside a crawler-type rescue vehicle, the human can feel the inclination of the vehicle. Using the previous experience, the operator naturally "feels on sight" the steepness of the environment. Then the decision on the traversability and path selection becomes more easy. Unfortunately, the off-site operator can not use any of those natural biological sensors and has to judge on the next move on the base of the partial available information, taking subjective and time consuming decisions. Transferring the function of taking such decisions from a human operator to a computer will decrease the burden on the operator. Our final goal is to provide a kind of automatic "pilot system" to propose an operator a good direction or several options to traverse the environment, taking into account the robot's static and dynamic properties.

To deal with this complicated problem, we must first solve a number of more simple tasks. The first step toward autonomous navigation is the ability to treat the information, provided by the robot's sensors, followed by feature extraction, environment decomposition and further simplification in order to create an internal world model. As soon as the internal world model is available, the robot should take a decision on the path within the internal model and then to apply this path in the real world scenario. Usually there exist more then just a single path, so the path search algorithm needs a good instrument to evaluate the quality of each path.

This paper deals with the quality estimation of the path. The simulator generates a discretized path between the via points proposed by a human operator, predicts the posture of the robot in the discrete points of the path and decides on the quality of each posture with regard to static equilibrium, which is a minimal necessary condition for the equilibrium. The results obtained by the simulator were confirmed with a number of experiments with a real robot in the environments identical to the simulated ones.

2 The System Framework

The National Institute of Standards and Technology (NIST) has created a set of reference test arenas for evaluating the performance of mobile autonomous robots performing urban search and rescue tasks. The arenas are intended to help accelerate the robotic research community's advancement of mobile robot capabilities [5, 6]. One of the examples, simulating cluttered environment with debris is a so-called Random Step Environment(RSE) or Stepfield, which is widely used in the RoboCup Rescue competitions and rescue related research[11]. RSEs are designed to be easily reproduced, and yet behave in a similar way to real rubble and could be extended to other environments[10]. RSE consists of a final number of random steps of some minimal size simulating a heavily damaged environment of the buildings after the earthquake(fig.1(b)).

In our RSE each cell of is a wooden block of size around 100mm $\times$ 100mm and different in height, which may vary from one scene type to another. For our simulations and experiments we used the two sets of heights: {0,100,200,300}mm and {0,100,300}mm, where 0mm height corresponds to the ground level around the RSE-patch.

We assume a simple tractor-like crawler non-reconfigurable robot, corresponding to the main body of "KENAF" robot(fig.2(b)). The main body of "KENAF" consists of two large tracks with a small gap in between; the main specifications of "KENAF" without sensors, front and the back pairs of arms, used in experiments and by the simulation "pilot system", are given in table 1. Further we plan to extend our work to the full-powered "KENAF" with two pairs of service arms(fig.2(a)).As an input for our "pilot system" we use a RSE-map of the environment and a set of via points.

Table 1. Specifications of "KENAF" in basic configuration

Parameter	Measurement
Maximal inclination	
dynamic	60 deg
static	80 deg
Main body length	584 mm
Main body width	336 mm
Track width	150 mm
Height	270 mm
Weight	17.8 kg

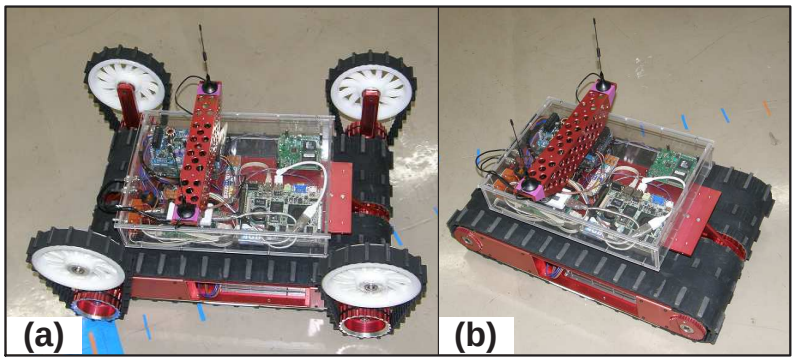

Fig. 2. (a)Full KENAF configuration without sensors (b) Main body without service arms and sensors

3 Stability Analysis

Probably, the most important question which the path search algorithm should be able to answer is if a specific robot configuration is possible or not. This includes not only collisions with the obstacles of the environment case, but also the capability of the robot to keep the current configuration.The robot should be able to stay in the specific configuration without slippering or turning upside-down. In other words, a safe and reliable motion of an autonomous vehicle requires continuous satisfaction of static and dynamic constraints. A vehicle in a stable state can become unstable for several reasons [12]:

1. Large inertial forces arising from rapid acceleration or deceleration, or tight turns;
2. Gravitational and reaction forces due to complex terrain geometries;
3. Surface conditions;
4. Unexpected external disturbances;

Assuming conditions 1, 3 and 4 are satisfied we deal with the second case.

3.1 Static Stability

Static stability is a minimal necessary condition for the general vehicle stability. In most papers dealing with the stability and balance issues the authors deal with a legged robot walking on uneven terrain[1, 4, 7, 8]. Such a robot can avoid falling only by applying contact forces with its feet on the ground that compensate for gravity without causing slip (so-called **static equilibrium**). Assuming robot's motion is slow enough to neglect inertia, the robot must always be able to achieve static equilibrium. While for a flat terrain some simple heuristic tests could be done for checking the stability, irregular and steep terrain requires to check that the robot is in equilibrium at every posture.

Existing approaches for a static stability are linear programming and linear projection. A basic interaction model assumes that the terrain is rigid and that contact occurs at frictional points. Under this assumption, "no slip" means that each contact force is restricted to a second-order friction cone, the shape of which is often approximated by a set of linear inequalities. Likewise, "compensate for gravity" means that the contact forces and center of mass(CM) positions satisfy linear force and moment balance equations. So for specific set of contact points, subject to the above approximation, the set of jointly feasible contact forces and CM positions is a polyhedron. The support polygon is the projection of this polyhedron onto CM-space. Using linear programming, we can search explicitly for a set of contact forces that place a particular CM position in equilibrium without computing the support polygon. Similarly, using linear projection, we can precompute the support polygon and thus determine whether it is possible to place a particular CM position in equilibrium without computing contact forces.

Gravity acts at the robot's center of mass (CM), the position of which may vary as the robot moves. While the change is significant for a legged robot, the CM position is constant for a tracked non-reconfigurable vehicle (tank or tractor) and has slight changes for a reconfigurable[3]. The properties of each contact point determine the range of contact forces possible without slip. So, for specific contact points, static equilibrium jointly constrains both the contact forces and the CM position. On flat terrain with point contacts, we have an intuitive notion of what this constraint means. Simply, the robot's CM must lie above the **support polygon** - a polygon with vertices at the contact points. The vertical prism having this polygon as cross-section is the set of all CM positions at which static equilibrium is possible. If the robot's posture places its CM over the support polygon, then we know contact forces exist that achieve equilibrium without actually having to compute them.

3.2 Modeling the constraint of static equilibrium

The number of contacts with the ground and their position relative to the center of mass is well defined for wheeled vehicles or mobile mechanisms with distinct foot pads. However, in a tracked vehicle there is an uncertainty about the number of contacts and their location along the track. Theoretically, the weight of the vehicle is dispersed evenly over the entire contact area of the track with the ground. Interaction between a track and the ground can vary significantly due to surface geometry, ground texture, track tension etc. There might be only few contacts along each track, affecting the vehicle's response to driving commands and balance.

In our case, RSE results mainly in the number of contact points, rather then the entire track contact. The specific features of RSE constrains all contact points to lie on the edges and at the vertices of the environment cells and on the perimeter of the robot crawlers. If a cell is completely under the crawler, it means or that there is no contact between the crawler and this cell at all, or

[3] CM position relatively to local coordinate frame fixed within the robot's body

3 options of the contact: a single point contact at a vertex of RSE cell, a line contact at an edge of RSE cell or a full-plane contact of the crawler with a cell in the case that the main body of the robot is parallel to the ground[4].

We define the appropriate posture of the robot, based on the literature survey and our mobility experiments. The requirements on the appropriate posture are:

1. The surface is rigid enough to provide the support to the robot;
2. The contact surface between the robot and the terrain provides the friction which is enough to prevent sliding forward, backward or aside;
3. The robot has contacts with the terrain with both crawlers and no contact with its main body, thus escaping the situations of getting stuck;
4. The robot has at least three contact points with the terrain;
5. The surface inclination does not result in slippery or turning upside-down;
6. The location of robot's center of mass prevents the robot from tip over due to the center of mass displacement;

While we assume the satisfaction of requirements 1 and 2, the satisfaction of other requirements is checked explicitly within our algorithm.

3.3 Coarse posture estimation

The output of our algorithm is an actual evaluation of the robot's posture in a specified configuration. Coarse estimation distinguishes three posture types:

Red state: Presents the posture which is impossible or statically unstable. Fig.3(a) demonstrates an impossible posture, where the robot collides with the environment. Fig. 3(b) demonstrates a case when the robot is trying to climb to an impossible steepness, which will result in turning upside down.

Green state: Stands for the statically stable posture, which satisfies all requirements stated in section (3.2). Fig.4(a) demonstrates a green state example.

Orange state: Is something between red and green states. This posture is possible, but not stable. It does not result in robot's turning upside down, but does not guarantee a single stable posture since there exist two options and the real one will depend on the previous posture, moving direction and other parameters. Fig.4(b) demonstrates a side view of an orange state with two possible postures. The orange state is very important, since it affords the robot to lose the balance on purpose, when for example the robot must traverse the barrier(fig.2(a)). Traversing the barrier includes climbing up and going down with loosing balance twice on top of the barrier.

4 The Algorithm

The location of the robot is described with (x,y) coordinates of the robot's CM in the horizontal plane of the global coordinate frame. In our model we assume

[4] If a cell is partially under the crawler, additional options are a single point contact at one of the 4 endpoints of the crawler or a line contact at a part of crawler's perimeter.

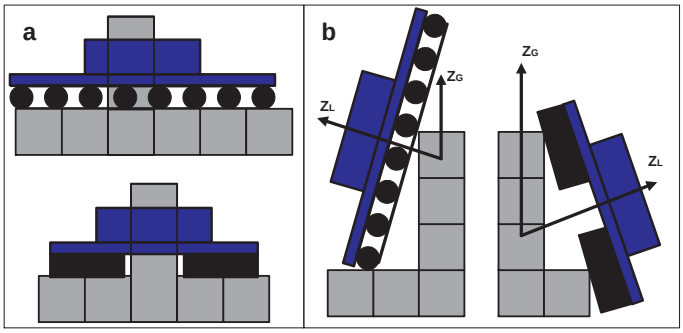

Fig. 3. Red state examples. a) Pikes appear under the main body of the robot side view (up) and front/back view (down) b) The robot is trying to climb to an impossible steepness

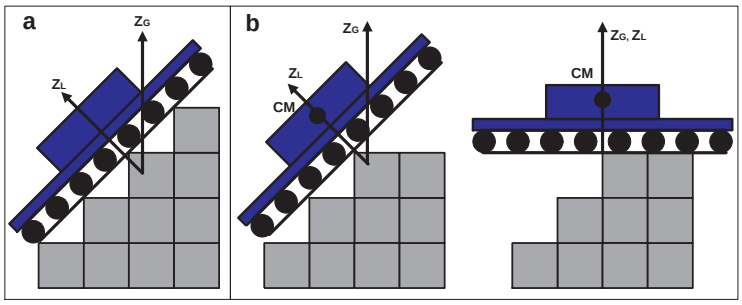

Fig. 4. Acceptable states examples. a) Green state b) Orange state - two possible postures

that CM is located at the physical center of the robot. Orientation θ of the robot is described by the angle between the robot moving direction and X-axis of the global coordinate frame X_G. Four other parameters (z,ω,φ,ψ) describing the posture of the robot, could be derived from the triple (x,y,θ) and the RSE-map with our posture search algorithm. Angles (ω,φ,ψ) are formed by the normal to the support plane with global coordinate system axes X_G, Y_G and Z_G respectively and fixed at the robot's CM.

Given (x,y,θ) at a current configuration, we assume initial posture of the robot as an input for the contact points search algorithm. As an initial guess, the triple (ω_0,φ_0,ψ_0) is passed from the previous posture, assuming initially that there is no change in the orientation of the support plane in the case of successful posture. In the case of starting point or unbalanced previous posture (ω_0,φ_0,ψ_0) = ($\frac{\pi}{2}, 0, \frac{\pi}{2}$). Next we build a list of all candidate points, found under the projection of the robot's posture (x,y,θ,ω_0,φ_0,ψ_0) on the RSE with respect to the requirements, described in section (3.2). Four service points are added for creating a 3D convex hull, based on that list. All points, which can not be contact points are enclosed inside the convex hull. Service facets containing as a

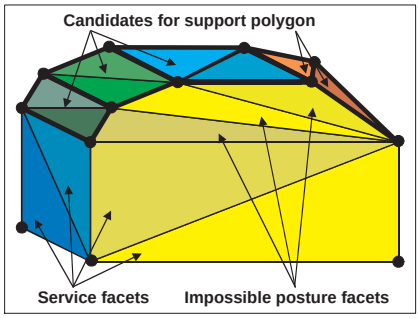

Fig. 5. Service and impossible facets are dropped. Thick black lines show the boundaries of the facets, which may serve as a support polygon.

vertex at least one service point and facets, which has all three vertices within the same crawler, are dropped. In physical world they will refer to impossible postures(fig.5). Each triangle is marked with the plane equation and then the facets with the same mark are recursively joined into a single facet in the case they have two common vertices, forming a set of candidate planes. Among those candidates only one facet contains a projection of CM and thus provides the recommended triple (ω,φ,ψ) for the next iteration. Since the change in (ω,φ,ψ) may be drastic, the obtained triple provides only the direction for the next iteration in (ω,φ,ψ). The iterations stop when the recommended values converge to the previous iteration value.

The obtained facet is checked for its final color. Orange color means that the CM's projection belongs to two distinct adjacent facets, both of which are acceptable postures. A red color facet stands for an impossible posture, while green means a success. An additional color, which was not explained previously is magenta; it appears when we can not obtain convergence during the posture search procedure and means the situation, when between two successive postures the robot's CM position changes its Z-coordinate with a leap while it has to climb a vertical slope of the environment. The magenta posture is also detected between two successive green postures, when CM position change in Z-coordinate exceeds the predefined threshold.

4.1 Qualitative classification of green state

Since the basic algorithm checks only static stability, the results show that almost every posture is green - i.e. statically stable. For this reason we provided a more precise qualitative classification of green posture with regard to static stability.

Throughout the history of walking robots several static and dynamic stability criteria have been defined. In [2] authors presented a comparative analysis of the existing stability criteria. Six case studies were considered and a classification of stability criteria showed that no optimum criterion for static and dynamic stability exists. Yet all of them showed to be valid and as for static stability margins only criterion NESM [3] provided the optimal measurements. We applied

this criterion for estimating the quality of the green state posture, substituting the foot contact points of the walking robot with the distinct contact points of the robot's crawlers.

NESM stability measurement criterion originates from the Energy Stability Margin (ESM) proposed by Messuri in [9], where ESM is defined as the minimum potential energy required to tumble the robot around the edges of the support polygon, that is:

$$S_{ESM} = \min_{i=1}^{N} mgh_i \tag{1}$$

where i denotes the segment of the support polygon considered as rotation axis, N is the number of distinct contact points of the posture, m is the robot's weight and h_i is the variation of CM height during the tumble, coming from

$$h = R_i(1 - \cos\gamma_i)\cos\tau_i \tag{2}$$

where R_i is the distance from the CM to the rotation axis, γ_i is the angle that R_i forms with the vertical axis and τ_i is the inclination angle of the rotation axis relative to the horizontal plane. The ESM gives a qualitative idea of the amount of impact energy supported by the vehicle. Then in [3] the authors normalized the ESM to the robot weight and propose the Normalized Energy Stability Margin, NESM , defined as:

$$S_{NESM} = \frac{S_{ESM}}{mg} = \min_{i=1}^{N} h_i \tag{3}$$

With the help of eq.(3) we define a new state - yellow; green state turns into yellow if the S_{NESM} does not exceed a predefined minimum established in the process of experiment trials with a real robot. As an additional and less important criterion for static stability estimation we employ the square of the support polygon.

5 Simulation and Experiments

The analysis of the proposed algorithm has been performed throughout simulation, constructed in Matlab environment. The goal of the analysis was to estimate the path proposed by the human operator in the given environment with respect to the static stability and to compare with the expected results.

The original map of the RSE is presented with a 2D square grid. While the final output images are scaled to the real RSE map cell of 100mm×100mm size, the calculations are conducted within 10mm×10mm cell discretization. A number of maps were created for the experiments in the real environment and then stored for simulation use. The operator chooses the map and marks start and target points of the robot's CM or a number of via points. The simulator discretises the path, so that we obtain all locations of the CM on the borders of the 10mm×10mm cells of the path and in the middle of each cell; the discretisation of rotation is 1 degree. Then the path, estimated within the simulator as "possible" was repeated in our experiments by the operator in a set of environments identical to the simulated ones.

From our previous experiments and experience of our operator we concluded that in the cases of straight and diagonal barriers the robot should choose the course perpendicular to the barrier. In the case of the pike, if the robot will try to traverse the pike straightly, trying to race through the pike on the high speed, it may tip over or get stuck on the pike; to prevent such situation, the robot should traverse the pike-type of the terrain going a roundabout way.

Next we present a short discussion of the simulation and experimental results and their correspondence for the case of the straight barrier traversing. The height of the bar at each point presents the location of the robot's CM. For magenta case there appear two locations, since at that point the robot have to climb up or down a vertical patch of the RSE cell. For red case there is no real meaning of the height location, since such posture is impossible in general. Thus, for the convenience of the reader, in the red case the simulator shows the previous non-red height and for orange and magenta cases there are additional signs for paying attention. The reader should keep in mind that our final goal is to provide an assistant "pilot system" for an operator of a rescue robot which will search for the good path/paths in the given environment, display it and warn the operator about possible dangerous segments of the path. In other words, the output of the simulator, presented here, is an intermediate result, which will not be completely displayed to the operator.

Fig.6(up) presents the simulation results of the path estimation when the robot traverses the straight barrier with a course at right angle to the barrier. Fig.6(down) presents the results of traversing the straight barrier on the diagonal course. Left figures correspond to 3D view of the barrier and the path, while right figures show the XZ-view. Figures 7(a) and 7(b) show the scenes from the corresponding real robot experiments.

In both cases the robot had to loose the balance twice on the top of the barrier - once when it had to switch from climbing up the barrier to moving parallel to the ground level on the top of the barrier and second time when it started moving down the barrier. In both paths there were detected a number of magenta cases when the robot had to climb up or down a vertical patch of the RSE cell and a number of yellow cases when the static equilibrium was not satisfactory. Even though both paths did not contain any red cases with impossible postures and thus were valid, the quality of the paths was significantly different. While the straight traversal path contained only a small number of yellow points (9 points, 2.45% of the path), the quantity of yellow points gradually increased as the path deviated from the perpendicular to the barrier. Finally, for the diagonal traversal the path contained a significant number of yellow points (91 points, 13.85% of the path). Thus we conclude that the straight traversal path's quality is better then of the diagonal traversal, which agree with our prior expectations.

This quality difference was well-detected during the experiments with the real robot. While to traverse the barrier in the straight manner did not take any serious effort from our operator, the diagonal traversal turned to be tricky. In the yellow points of the path the speed of the vehicle had to be reduced to minimal and required maximal concentration from the operator. In magenta points,

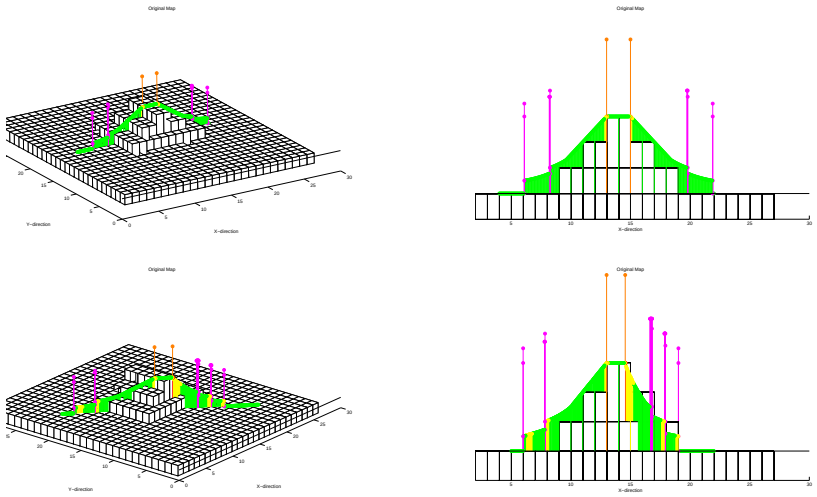

Fig. 6. Straight barrier traversing, map resolution 100×100mm, solution resolution 10×10mm. First row pictures show the straight traversing, second row - diagonal traversing.

when the Z-coordinate was changing from lower to higher level, the climbing ability was not guaranteed and strongly depended on the level change; after some critical level change the vertical climbing became impossible[5]. When the Z-coordinate was changing from higher to lower level, the robot experienced pushes and hammering with a strength depending on the level change. Such hammering may damage the vulnerable sensors and should be avoided as well.

6 Conclusions and future work

The final target of our research is to provide an assistant "pilot system" for an operator of a rescue robot, decreasing the burden on the human operator. As soon as a robot obtains data from the environment and creates an internal world model, a selection on the path within the internal model should be done, followed by applying this path in the real world scenario. Since usually there exist more then just a single path, the path search algorithm needs a good instrument to evaluate the quality of each path.

In this paper we presented the algorithm for estimating the posture of the robot in a specific configuration from the static equilibrium point of view. The results obtained by the simulator agree with our prior expectations and were successfully confirmed by the set of experiments with a real robot. Our future work will concentrate on the path planing algorithm, which will utilize the proposed

[5] Experiments with a high straight barrier set of {0,100,300}mm cell heights showed that the robot would climb it only in a full configuration using service arms (fig.2(a))

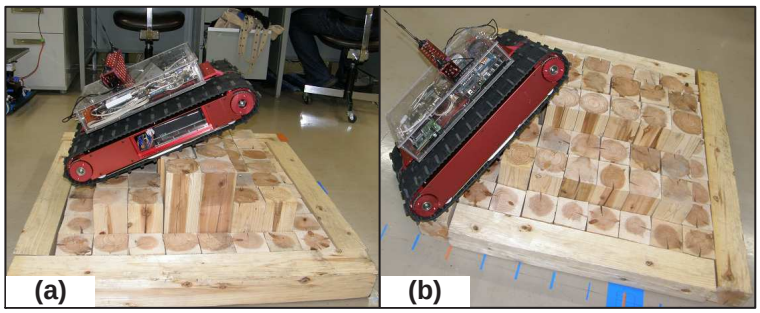

Fig. 7. Scenes from the experiments: (a) straight barrier, straight traversing (b) straight barrier, diagonal traversing

quality estimation of the posture's static equilibrium as a part of configuration evaluation function.

References

1. Bretl, T., Lall, S.: A Fast and Adaptive Test of Static Equilibrium for Legged Robots. ICRA'06, 1109–1116 (2006)
2. Garcia, E., Estremera, J., Gonzalez de Santos, P.: A classification of stability margins for walking robots. Proc. of 5th Int.Conf. on Climbing and Walking Robots and the Support Technologies for Mobile Machines, Paris, France, 799–808 (2002)
3. Hirose, S., Tsukagoshi, H., Yoneda, K.: Normalized energy stability margin: generalized stability criterion for walking vehicles. Proc. of 1st Int.Conf. On Climbing and Walking Robots, Brussels, 71–76, (1998)
4. Hong, D.W., Cipra, R.J.: Optimal contact force distribution for multi-limbed robots. Journal of mechanical design, 2006, Vol.128, 566–573 (2006)
5. Jacoff, A., Messina, E., Evans, J.: A standard test course for urban search and rescue robots. Arch. Proc. of the 2000 PerMIS Workshop, Gaithersburg, MD (2000)
6. Jacoff, A., Messina, E., Evans, J.: Experiences in Deploying Test Arenas for Autonomous Mobile Robots. Proc. of the 2001 PerMIS Workshop, in association with IEEE CCA and ISIC, Mexico City, Mexico (2001)
7. Klein, C.A., Kittivatcharapong, S.: Optimal force distribution for the legs of a walking machine with friction cone constraints. IEEE Trans. on Robotics and Automation, Feb 1990, Vol.6(1), 73–83 (1990)
8. Mason, R., Rimon, E., Burdick, J.: Stable poses of 3-dimensional objects.IEEE ICRA '97, Vol.1, 391–398 (1997)
9. Messuri, D. A.: Optimization of the locomotion of a legged vehicle with respect to maneuverability. Ph. D. Thesis, The Ohio State University (1985)
10. Poppinga, J., Birk, A., Pathak, K.: Hough based Terrain Classification for Realtime Detection of Drivable Ground. Journal of Field Robotics, vol. 25, 67–88 (2008)
11. Sheh, R., Kadous, M.W., Sammut, C., Hengst, B.: Extracting Terrain Features from Range Images for Autonomous Random Stepfield Traversal. IEEE Int. Workshop on Safety, Security and Rescue Robotics, Rome, Sep.2007, 1–6 (2007)
12. Shoval, S.: Stability of a Multi Tracked Robot Traveling Over Steep Slopes. IEEE ICRA '04, Vol.5, 4701–4706 (2004)