The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Static Balance for Rescue Robot Navigation : Losing Balance on Purpose within Random Step Environment

Evgeni Magid, Takashi Tsubouchi, Eiji Koyanagi and Tomoaki Yoshida

Abstract— **The goal of rescue robotics is to extend the capabilities and to increase the safety of human rescue teams. During a rescue mission a mobile robot is deployed on a rescue site and is operated from a safe place by a human operator. The operator can not see the robot and the environment and a decision on the path selection is very complicated. 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 an account the robot's static and dynamic properties.**

To find a good path we need a special path search algorithm on debris and a proper definition of a search tree, which can ensure smooth exploration. While the main goal of the algorithm is to keep the robot maximally stable at every step of its path, in some cases we need the robot to lose its balance and to change a 3D orientation discontinuously. Losing balance on purpose is an important feature for safe climbing up and going down the debris and it is the central issue of this paper. Exhaustive simulations were used to structure and analyze data and experiments were used to verify our approach to removing unsuitable directions of the search from the search tree.

I. INTRODUCTION

A long standing goal of mobile robotics is to substitute a human crew in environments unreachable or too hazardous to risk human lives. Rescue robotics is the application of robotics to the search and rescue domain, when victims are often buried in unreachable locations. In particular, in advance of a manned rescue operation, the inside of heavily damaged by an earthquake buildings should be investigated by a rescue robot in order to avoid risk of casualties from secondary disaster. The goal of rescue robotics is to extend the capabilities of human rescuers while increasing their safety. During a rescue mission a mobile robot is deployed on a rescue site, while a 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 a robot control subsystem and a remote operation station, connected with a wireless LAN.

In this paper we present our current results in estimation of losing balance on purpose within Random Step Environment(fig.1(b)), which is a simulated debris environment, proposed by National Institute of Standards and Technology (NIST)[4]. To find a good path we implement a special path search algorithm on debris. Because the debris site is dangerous and unstable, the main goal of the algorithm is to keep the robot maximally stable at every step of its path. However, in some cases we need the robot to lose its balance and to change its 3D orientation discontinuously. Without this kind of transition the robot cannot successfully climb up the obstacles and in general is not suitable for debris exploration. Since the real state space of the search is extremely huge, we discretize robot's motion and the state space before the actual search in order to decrease the number of search directions. A search algorithm utilizes a search tree [2]; for our problem dynamically created search tree can not be explicitly presented as a skeleton. To present it as a function $F(Args) = Res$, where arguments $Args$ are the robot's current configuration and the local environment map and output Res is a set of accessible within one step configurations, we need a proper definition of function F which will guide the tree search. Here we present the particular part of function F responsible for losing balance on purpose. Since we must control well all postures of the path, predictable enough balance loss transitions become a part of the path, while all suspicious and unpredictable cases must be forbidden. Out theoretical results were confirmed with exhaustive simulations and experiments, removing all unsuitable directions of the search from the search tree.

Currently rescue robots are operated manually by human operators. The remote operator uses only visual information about the environment, which is usually not sufficient to carry out complex tasks because of the limited visual fields of cameras. 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 and the decision on the traversability of the path becomes more easy. Unfortunately, the off-site operator can not use any of those natural biological sensors and have to judge on the next move on the base of the partial available information, taking subjective and time consuming decisions. Many optional paths from the start to the target position exist and it is very hard for the operator to choose a good and safe path. 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 "pilot system" to propose an operator a good direction or several options to traverse the environment. The operator will receive a proposal on a good path from the "pilot system" on the computer display by means of GUI and apply it in a real scenario driving KENAF robot.

This work was partially supported by NEDO Project for Strategic Development of Advanced Robotics Elemental Technologies, High-Speed Search Robot System in Confined Space

E.Magid and T.Tsubouchi are with ROBOKEN Intelligent Robot Laboratory, Institute of Engineering Mechanics and Systems, University of Tsukuba, Tsukuba, Japan {evgeni,tsubo}@roboken.esys.tsukuba.ac.jp

E.Koyanagi and T.Yoshida are with Future Robotics Technology Center, Chiba Institute of Technology, Japan

978-1-4244-6676-4/10/$25.00 ©2010 IEEE

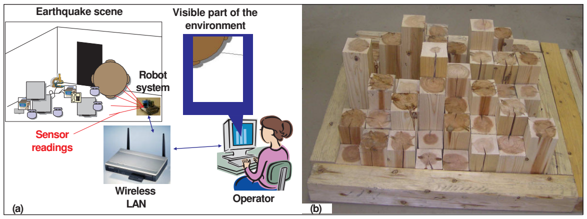

Fig. 1. (a) Standard framework (b) RSE example.

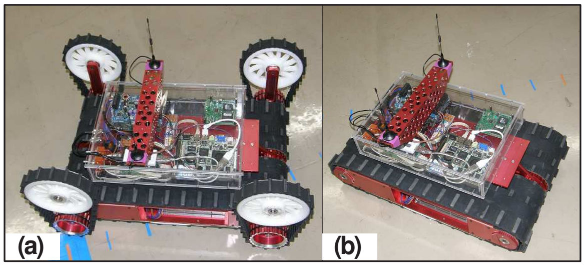

Fig. 2. (a) Full KENAF configuration without sensors (b) Main body without service arms and sensors.

II. The System Framework

The National Institute of Standards and Technology (NIST) created a set of reference test arenas for evaluating the performance of urban search and rescue robots[4]. One of the examples widely used in the RoboCup Rescue competitions and rescue related research is a so-called Random Step Environment(RSE) or Stepfield which simulates cluttered environment with debris[8], [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)); height of each random step may vary from one scene type to another. RSEs are easily reproduced and yet behave in a similar way to real rubble.

In our RSE each cell is a wooden block of 85mm × 85mm size and 0, 90, 180 or 270 mm height, where 0mm hight 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 back pairs of arms, used in experiments and by the simulation ”pilot system”, are given in table I.

TABLE I

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
Hight	270 mm
Weight	17.8 kg

III. Static Stability and Balance Estimation

The debris site is dangerous and unstable and the main goal of the algorithm is to keep the robot maximally stable at every step of its path in the specific configuration without slippering or turning upside-down. A safe and reliable motion of an autonomous vehicle requires continuous satisfaction of static and dynamic constraints[9]. Static stability is a minimal necessary condition for the general vehicle stability. For the static stability satisfaction the robot’s center of mass (CM) must lie above the **support polygon** - a polygon with vertices at the contact points of the robot’s crawlers with RSE[1].

In [6] we presented an algorithm for static balance posture estimation of the robot’s posture in a specified configuration, assuming the centroidal location of robot’s CM. In this section we briefly describe the static balance posture types and assign them color names. From the point of static balance estimation, we distinguish six posture types. **Red state** presents a statically unstable posture, resulting in robot’s turning upside down while trying to climb to an impossible steepness. **Magenta state** appears while the robot has to climb up or to go down the vertical slope of the environment and is legal only for translation motion. **Cyan state** is assigned for a robot’s jumping down situation and is legal only for a rotation motion. Statically stable postures are described with **Green** and **Yellow states**(fig.3(a)); to distinguish those two states we apply Normalized Energy Stability Margin (NESM)[3], which shows how stable the posture is: high quality balance(G) or average quality(Y). Further we denote by R-posture a posture which static balance corresponds to a red state type, M-posture for magenta etc.

Last, **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 do not guarantee a single stable posture since there exist two options and the real one depends on the preceding posture and moving direction. Fig.3(b) demonstrates a side view of an orange state with two possible postures. Orange state is very important, since it affords the robot to lose the balance on purpose, when for example the robot traverses the barrier. Traversing the barrier includes climbing up and going down with loosing balance twice on top of the barrier. We denote by O_1 the first part of the O-posture before the robot looses its balance. O_2 is the second part, which occurs after the robot have lost its initial balance; the robot changes its posture discontinuously at that point and obtains a balance again in a different body orientation. Starting at O_1 posture, we immediately obtain O_2 as a result of inertia and there is no way to obtain O_2 posture without previously obtaining O_1 posture. O-posture is the central issue of this paper.

IV. Describing a Posture

To characterize robot’s posture qualitatively we use the coloring of the states. To decide possible transitions between two successive states, we use 6 variables, whose combinations help us to define legal transitions between the states.

Steepness θ_X - the angle, showing the steepness of the environment at a given robot configuration. Angle between

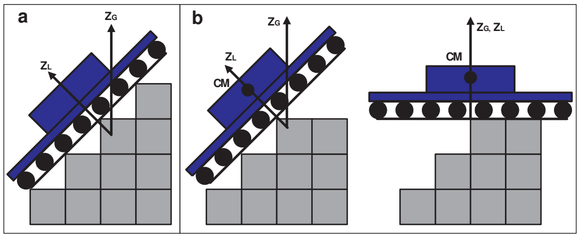

Fig. 3. (a) Green state (b) Orange state: O_1 (left) and O_2 (right).

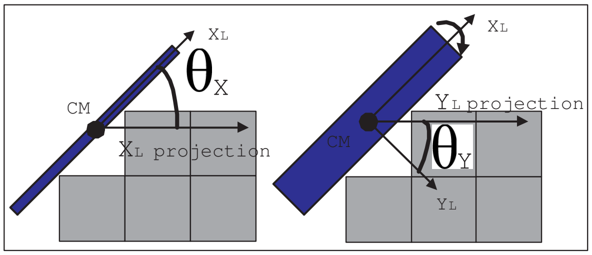

Fig. 4. Steepness θ_X(a) and moment θ_Y(b)

the local axis X_L and its projection on the plane of the global axes X_G, Y_G(fig.4(a)).

Moment θ_Y - the angle, showing the dangerous rotational moment around robot's X_L-axis at a given configuration. Angle between the local axis Y_L and its projection on the plane of the global axes X_G, Y_G(fig.4(b)).

Contact points quality (CPQ) - depends on the angle θ_{CPQ} between the robot's crawlers and the edges of the RSE cells. It affects the robot's ability of climbing the obstacles and going down safely.

Inclination - is the *steepness* angle θ_X sign. We distinguish three groups of posture sets with respect to this parameter: $G_{U_{inc}}$ is a climbing up the steps of the environment posture, $G_{D_{inc}}$ is a going down posture and $G_{Z_{inc}}$ is a neutral inclination posture.

M-sign - is the *moment* angle θ_Y sign. Similar to inclination, group $G_{P_{MS}}$ contains all postures with positive M-sign, $G_{N_{MS}}$ with negative and $G_{Z_{MS}}$ with neutral[1]. Using inclination and M-sign, we define a neutral Z-posture as a posture with robot's body being parallel to the ground level: $G_{Z_{inc}} \bigcap G_{Z_{MS}}$

NESM-stability - shows the probability of the robot's turning upside down due to the CM being too close to one of the edges of the support polygon[3].

Inclination and *M-sign* are the most important variables. They signal about discretization problems, pointing on the missed posture between two successive postures. 4 other variables are emphasized for the experimental work and particulary for creating input, which satisfactorily spans all possible transitioncases. Further we denote each posture as Col(Inc) where Col is the color, Inc is the *inclination*. For example, G(Z) means a green neutral Z-posture and O_1(U) means O-posture O_1 with U_{inc}.

V. Orange posture

KENAF supports two types of motion: translation and rotation. We define **translation step** as a one cell length step forward in the direction of robot local frame's axis X_L. **Rotation step** is a 5 degrees rotation left or right with regard to robot's local frame's axis X_L around axis Z_L. In practice rotational motion within the simulation and in the real world differ a lot. Quality of the surface, number of real contact points, robot's speed, accumulated error in control system, communication delay etc. could result in a high level of imprecision. To ensure that the simulated path could be repeated by a human operator in a real scenario, we forbid any dangerous and unpredictable transitions between two states. Appearance of the O-posture is also a non-trivial and hardly predictable situation, so we forbid it for rotational motion and further deal only with a translation case.

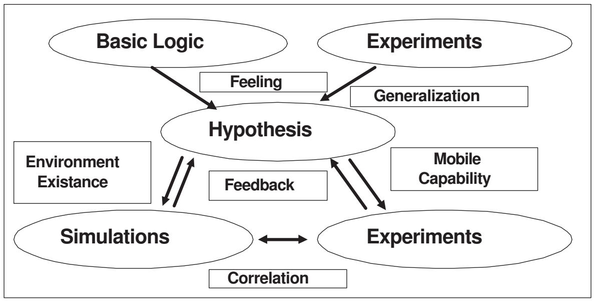

Fig. 5. Theory, simulations and experiments.

A. Orange Posture Contact Type

We distinguish three main types of O-posture with regard to the physical characteristics of the contact of the robot's crawlers with the RSE:

Accidental O-posture (AOP) is obtained while passing through the corner of the RSE-cell[2]. If the robot posture (x,y,θ), preceding AOP, would become (x$\pm\delta_x$,y$\pm\delta_y$,θ) with $\delta_x, \delta_y \in (0, \varepsilon_{shift}]$, this AOP will not be obtained at the next translation step, but a differently colored posture will arise from posture shift in any direction by δ. AOP is theoretically possible, but when the simulated path containing AOP is to be repeated in the real world scenario by the operator, any small deviation will result into drastic path change and even into robot's turning up side down. The detailed discussion on ε_{shift} choice is presented in section VII-B. Only 8 main directions of the posture shift (x$\pm\delta_x$,y$\pm\delta_y$) are chosen based on the properties of RSE: O-posture occurrence is strongly related with edges and vertices of RSE cells.

Inevitable O-posture IOP-1 appears when the robot is passing through an edge of RSE in a *close vicinity* of the RSE corner. Thus, a shift by δ in one direction preserves the current O-posture, while shift in any other direction produces a differently colored posture.

Inevitable O-posture IOP-2 appears when the robot is passing through an edge of RSE far enough from the RSE corner. In this case shift of CM in two opposite directions creates an identical O-posture, while other directions produce G-posture (mainly) or M/R-posture (rarely).

[1] $G_{Z_{inc}} : |\theta_X| \leq \varepsilon$; $G_{Z_{MS}} : |\theta_Y| \leq \varepsilon$; ε=1 degree

[2] From here further we explain the idea of AOP, IOP and O_1 →O_2 transition groups of section VI with simple examples, while both in the simulation and real world experiments the occurrence cases may be much more complicated

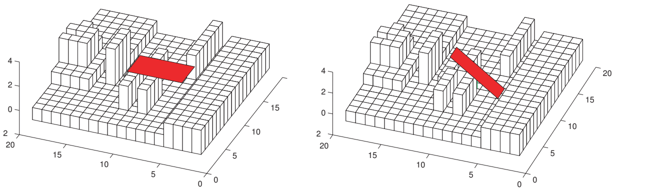

Fig. 6. Translation GG1 from (left) to (right), missing intermediate O-posture.

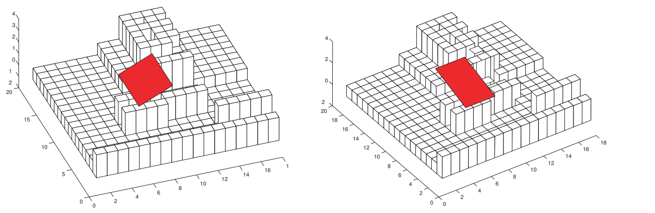

Fig. 7. Translation GG6 from (left) to (right), missing intermediate AOP type O-posture.

B. Discretisation Issue and Forbidden Sequences

Based on a large set of experiments with KENAF robot in several Random Step Environments and exhaustive simulations we summarized main properties of O-posture. Since $O_1 \rightarrow O_2$ transitions should be applied with a special care, we restrict the appearance of O-postures within the search tree neighborhood definition function F. Due to the level of discretization in some cases we miss explicit appearance of O-postures and may only guess on its appearance while carefully comparing two successive postures P_1 and P_2.[7] The results showed that there is a missed intermediate O-posture between P_1 and P_2 for G(Z)→G(D)(fig.6) and G(U)→ G(Z) translations. In three following cases there is a missed AOP between P_1 and P_2, both G or Y-postures :

1) If there is a sign change for θ_X or θ_Y between P_1 and P_2(fig.7).
2) If a change in θ_X and θ_Y both exceed the predefined thresholds $|\Delta\theta_X| > T_{MAX}$ and $|\Delta\theta_Y| > \varepsilon$, where $T_{MAX} \doteq 8$ and $\varepsilon \doteq 1$ degrees .
3) If for θ_X or θ_Y $|\Delta\theta_X|, |\Delta\theta_Y| \in [T_{MIN}, T_{MAX}]$, where $T_{MIN} \doteq 3.5$ degree

For all simulations we used discretization DISC5 - each 85x85 mm cell of RSE turns into 5x5 cells of the internal robot map with the cell size of 17x17 mm. We calculated T_{MIN} by maximizing the difference $\Delta\theta_X$ between two successive G-postures with optimization method, updated it through a set of simulations and finally set to $T_{MIN} \doteq 3.5$ degree; similarly $T_{MAX} \doteq 8$ degrees.

The three later cases together with transition pairs M→O, O→O[3] and O→M are considered to be dangerous sequences and are recolored into R-postures. As an example, consider a pair O→O: just within one step the robot will have to loose the balance twice; this mean climbing and going down through a corner of the RSE-cell with a very small contact square between one of the crawlers and a cell, being close to the AOP case.

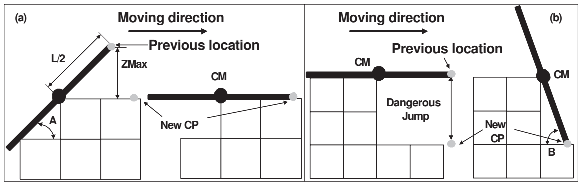

Fig. 8. $O_1 \rightarrow O_2$ transition verification while climbing up(a) and going down(b).

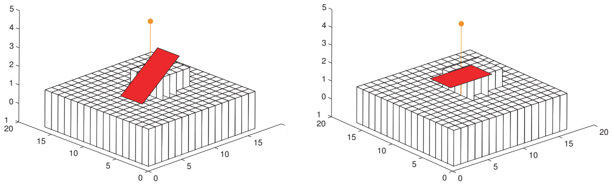

Fig. 9. OO1 IOP-1 - robot's CM is too close to the vertex of RSE cell.

O-posture is more important and has a higher cost in the path planning, so in the case of a missed due to the discretization issue O-posture we recolor the second posture P_2 of the sequence into O-color and then determine its type. Since there is no real data for deciding on the O-posture type in this case, we have to use an approximation.

To ensure a smooth transition from O_1 posture to O_2 we verify that there is no danger for the robot's sensors due to significant changes in θ_x, θ_y(fig.8). All newly obtained contact points of the robot's crawlers with the RSE at O_2 are allowed to change their Z-coordinate for a value within a predefined safety interval with regard to corresponding contact points at O_1. We defined experimentally two intervals of the change : $safe\ interval$ as [0mm,90mm] and $dangerous\ interval$ as [90mm,180mm]. Maximal Z-coordinate change for one of the contact points is the one which defines the interval for the whole posture. While safe interval is preferable and dangerous should be minimized, any higher Z-coordinate change will definitely destroy the sensors.

VI. Orange Transition Groups

Using parameters presented in section IV we define 10 distinct groups of all optional $O_1 \rightarrow O_2$ transitions, resulted by inertia of translational movement.

(OO1): O(Z)→O(D). The robot being in a neutral posture on a flat pattern of the RSE approaches an edge of a barrier or RSE cell, loses its balance at the edge and switches to going down mode. We expect high appearance of this type transitions. Fig.9 demonstrates IOP-1 climbing case, when robot's CM is too close to the vertex of RSE cell. Fig.10 demonstrates a safe IOP-2 climbing case, which is the most usual case of OO1.

(OO2): O(Z)→O(U). The robot being in a neutral posture on a flat pattern cannot lose its balance so that it will switch to climbing up mode.

[3] Two consequent distinct O-postures, not $O_1 \rightarrow O_2$ transition

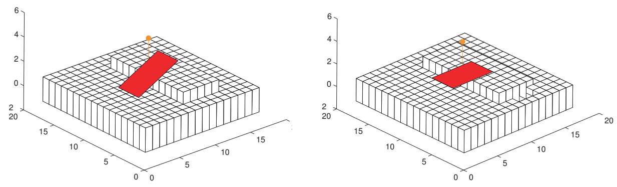

Fig. 10. OO1 IOP-2 - regular safe climbing.

(OO3): O(U)→O(D). The robot cannot immediately switch from climbing up to going down; even in the AOP case, it must pass through a neutral posture, i.e. this change is impossible due to the physical RSE rules.

(OO4): O(U)→O(U). A rare case, when the robot slightly changes the orientation while climbing from a current obstacle to another. Usually the difference in orientation is so small, that it is hardly noticed visually, but only analytically in the transformation matrices. For example, two rotation matrices[4] ${}^{O_1}_{Z}R$ and ${}^{O_2}_{Z}R$ for OO4 case are:

$$
{}^{O_1}_{Z}R = \begin{pmatrix} 0.9993 & 0.0361 & 0 \\ -0.0308 & 0.8526 & 0.5216 \\ 0.0188 & -0.5213 & 0.8532 \end{pmatrix}
$$

$$
{}^{O_2}_{Z}R = \begin{pmatrix} 0.9995 & 0.0263 & 0.0161 \\ -0.0308 & 0.8526 & 0.5216 \\ 0 & -0.5218 & 0.8531 \end{pmatrix}
$$

(OO5): O(U)→O(Z). The robot started to climb up a barrier and upon reaching a top, as CM approaches an edge of the barrier/RSE cell, it switches to a neutral posture on a flat pattern. Similarly to OO1 this is the most regular case.

(OO6): O(D)→O(D). A rare case, when the robot slightly changes its orientation while going down from a current obstacle to another. Similarly to case OO4 the change in orientation usually could hardly be noticed visually.

(OO7): O(D)→O(U). Similar to OO3, this change is impossible due to the RSE rules. In simulation both OO3 and OO7 appear very rarely and only due to accumulated numerical errors.

(OO8): O(D)→O(Z). The robot while in going down mode cannot immediately switch to a neutral posture by losing the balance.

(OO9): Is a group of postures, where θ_y difference between O$_1$ and O$_2$ exceeds 5 degrees while none of O$_1$ and O$_2$ is neutral. In some cases OO9 transition arise due to accumulated numerical errors and after an additional posture check it turns to be not O-posture, but G-posture. In other cases it is AOP, a big point-type change with unpredictable behavior (fig.11). Simulations showed that this case is a rare one and we decided to forbid it completely for IOP1 and try to avoid for IOP2 (fig.12).

(OO10): O-posture appears between two identically orientated G-postures and gives us a hint on a missed M-posture between O$_2$ and next G-posture (fig.13).

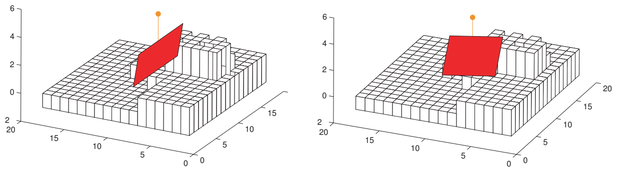

Fig. 11. OO9 AOP case.

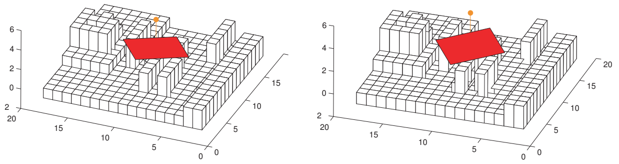

Fig. 12. OO9 IOP-2 case.

We group OO cases into 3 clusters:

1) OO1 and OO5 are two most regular cases of losing the balance while climbing up and going down.
2) Cases OO4, OO6, OO9 and OO10 are rare cases, which we would like to avoid, but yet they could be included as a part of the path if no better option exists.
3) Cases OO2, OO3, OO7 and OO8 theoretically could not exist and may appear in the simulation due to the discretization and accumulated numerical error issues.

Since O-posture is very important, we must take care to distinguish and partially forbid the appearance of AOP, IOP1 and IOP2 cases, as we explain in the next section VII.

VII. Simulation and Experiments

The only real proof of any theoretical hypothesis is an experimental proof. Thousands of different situations can occur in a completely random RSE and it is physically impossible to execute such huge number of experiments. The exhaustive simulations helped to confirm our hypothesis on impossible due to the physical rules of RSE situations. Pairs of postures, impossible in the real world, are also impossible within the simulation. Since the reverse statement is not true, the simulator can not substitute the experiments, but helps to structure the data and remove the impossible types of sequences, saving time and efforts. Exhaustive simulations for environments existence in MATLAB and experiments with KENAF robot in RSE gave a valuable feedback for our

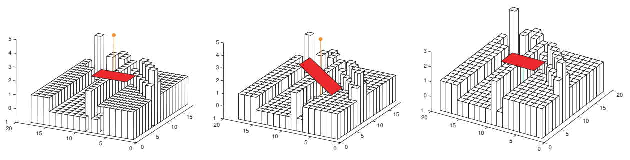

Fig. 13. OO10 IOP-2 case: a missed M-posture between O-posture inertial sequence O$_1$(left) → O$_2$(center) and next G-posture (right).

[4] Rotation from neutral posture Z to the actual posture O$_1$ or O$_2$

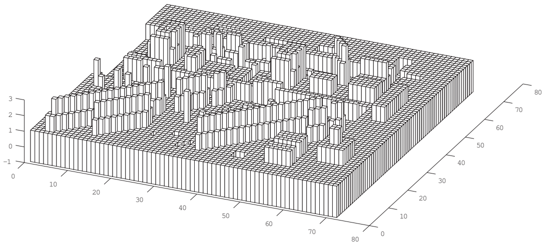

Fig. 14. A complicated environment of size 61x61 cells, covering all main types of the environment obstacles.

theory and finally produced a branch cutting condition for the path search algorithm(fig.5). Successive transition patterns of O_1 $\rightarrow O_2$ will be integrated in the search algorithm as a part of neighbor opening and branch cutting function $F(Args) = Res$.

A. Simulation summary

To simulate all possible combinations of two sequent postures we created a huge environment of 61x61 cells (fig.14) which includes all typical obstacles, usually appearing in the random step environment: horizontal and diagonal barriers, pairs of parallel barriers, valleys, traversable and non-traversable pikes and holes. An exhaustive check of all possible pairs appearance of neighboring postures connected with a translation step were proceeded with voting for each group. As a first robot's CM posture of the pair we took every node of the grid[5]; a second posture of the pair was calculated as a 1-unit length change of CM's location in the direction of the robot's heading direction θ. The simulation included 91 robot orientations $\theta \in \{0, \frac{\pi}{180}, \frac{2\pi}{180}, ..., \frac{89\pi}{180}, \frac{\pi}{2}\}$. In addition to pointing at the impossible (empty) cases the simulation reveals the rare cases and the most often cases.

The total number of posture pairs was more then 6 millions. Among them O-posture appeared in 1.34% of the pairs - 1.3% of them were further proceeded for detection of the O-transition type and only 0.04% were immediately forbidden as was explained in section V-B. As we expected, most of O_1 $\rightarrow O_2$ transitions are of OO1 and OO5 type, they cover 97.6% of all cases; rare cases appear only in 2.1% of the cases, while the rest 0.2% are forbidden cases (table III). To distinguish IOP and AOP cases theoretically within the simulation we used $\delta_x = \delta_y$=0.0017mm; the choice of δ for practical use is explained in the next sectionVII-B.

The results of the simulation are summarized in the following tables, corresponding to the appearance of O-postures. We listed as $main$ transitions IOP1 and IOP2 cases of OO1 and OO5 (table II). $Undesirable$ transitions include IOP1 and IOP2 cases of OO4, OO6, OO10 and IOP2 cases of OO9 since we would prefer to avoid those transitions until we do not have another choice. $Forbidden$ transitions are all AOP cases, IOP1 cases of OO9 and all cases of OO2, OO3, OO7 and OO8. We manually checked a large number of forbidden transitions appearing in the simulator and our expectations we confirmed - all of them appeared due to the accumulated numerical errors. While IOP-1 case of OO9 group is not a numerical error, this kind of transition is very dangerous and is also treated as a forbidden case. Forbidding dangerous and suspicious transition, which still may be theoretically possible in some rare cases, limits our path choice, but increases the security for the practical use.

TABLE II

MAIN, UNDESIRABLE AND FORBIDDEN TRANSITIONS.

Main	Undesirable	Forbidden
93.47	1.54	4.99

TABLE III

ORANGE POSTURE TYPES DISTRIBUTION.

OO1	OO2	OO3	OO4	OO5
44.38	0.05	0.14	0.37	53.22
OO6	OO7	OO8	OO9	OO10
0.5	0.04	0.08	0.85	0.38

TABLE IV

IOP-2 TYPE POSTURE DISTRIBUTION.

OO1	OO2	OO3	OO4	OO5
41.69	0.05	0.14	0.25	50.71
OO6	OO7	OO8	OO9	OO10
0.37	0.04	0.08	0.54	0.37

In total IOP2 is 94.24% of all O_1 $\rightarrow O_2$ transitions and again 92.4% of them are OO1 and OO5 (table IV); IOP1 is 1.07% and AOP is 4.68%. It means, that forbidding dangerous 4.68% cases of all AOP appearances and unsuitable 0.31% of IOP1 and IOP2, we still have enough freedom for loosing balance on purpose - 95%, including 1.54% of undesirable appearances of OO4, OO6, OO9 and OO10.

B. Experimental definition of AOP and IOP

To decide which ε_{shift} and δ to choose for the definition of AOP and IOP we initially conducted a set of simulations. Simulations are very time consuming, so we could not repeat the execution of all 6 million pairs with different choices of δ. As a test group we have chosen 4 orientations: {0,15,45,71} degrees and unfortunately we could not establish a clear dependence of AOP/IOP-1/IOP-2 appearance for different choices of $\delta \in$ {17,8.5,4.25,1.7,0.17,0.017,0.0017}mm. This points again on the important issue of discretization and vulnerability of the computer simulation to the accumulating numerical errors. Any value for ε alone cannot guarantee a proper decision on AOP/IOP-1/IOP-2 type. Shifting the O-posture itself and then taking decision on the type is not effective - we must also check the behavior of previous and next to O-posture poses of the path in the case of their shift by δ in 8 directions. If 3D orientations of shifted in two opposite directions postures would coincident with the non-shifted ones up to some level, we got a IOP-2 type O-posture. If only one direction works - it is IOP-1; otherwise, it is AOP. Our small theoretical shift like 0.0017 mm (0.001 units at DISC5) could not be considered as a repeatable by the human

[5] The nodes too close to the borders of the map were excluded

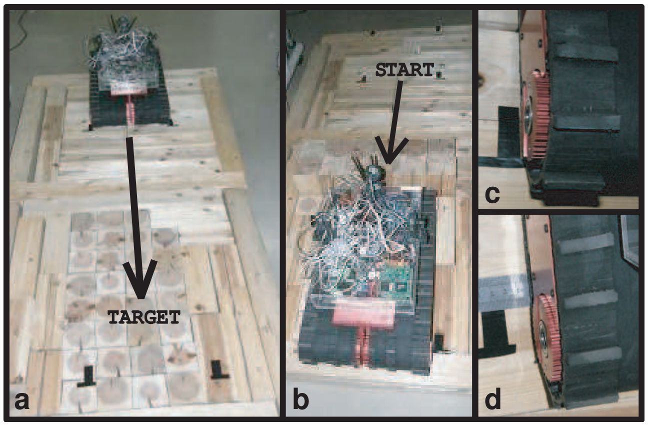

Fig. 15. Side shift detection experiments: (a) flat pattern RSE (b) small horizontal barrier traversal with initial position at start (c) and at target (d)

operator option, so the final choice of ε_{shift} limit and δ for further $pilot\ assistant$ was done experimentally.

For these experiments we created 2 types of RSE - a flat pattern and a horizontal barrier (fig.15(a),(b)) - and marked start S and target T postures. In the first set of the experiments the robot traversed RSE from S to T at a given orientation and we measured the side shift of the robot's corner points relatively to the desired posture T. In the second set the robot went to the target posture T and back and the shift at start posture S was measured after returning.

28 experiments were conducted for orientations θ =45 and 90 degrees for a path length of 117cm (2 robot's length), which is initially set as a short distance path planning range. In the worst case the shift was 77mm together with a significant orientation change, while in all other cases the shift stayed within 20mm range. Fig.15 demonstrates the side shift detection experiments at θ_{CPQ}=90 degrees: flat pattern RSE(a), small horizontal barrier(b), initial position at start posture S of the front right corner point of the crawler(c) and shift of 20mm at target T posture(d). Finally, we concluded that a side shift of 20mm is a good choice for short range path planning.

C. Experimental constraints on $O_1 \rightarrow O_2$ sequences

The goal of this set of experiments was to confirm the proper CPQ angles θ_{CPQ} for the main OO1 and OO5 cases. While the choice of climbing up at OO1 is restricted by the climbing abilities of KENAF robot at M-postures, the choice at OO5 depends purely on $O_1 \rightarrow O_2$ inertial transition. For both climbing up for OO1 and going down for OO5 the best CPQ angle θ_{CPQ} choice is a straight angle of 90 degrees, while there exist angles Θ^1_{b1} and Θ^2_{b1} so that:

- For all CPQ angles $\theta_{CPQ} \in [0,\Theta^1_{b1}]$ the climbing up is impossible
- For all CPQ angles $\theta_{CPQ} \in [0,\Theta^2_{b1}]$ the robot has a high probability of turning upside down while losing the balance on the edge of the barrier when it switches to a going down mode

To detect experimentally Θ^1_{b1} and Θ^2_{b1} we performed 60 experiments with a simple RSE horizontal barrier for climbing up and 58 experiments with a very tricky horizontal barrier for going down. A tricky barrier idea (fig.17(b)) was to make KENAF turning upside down easily at most of the orientations, so that forbidding all of them would minimize the risks while navigating in real environments. Starting from 45 degrees for θ_{CPQ} angle, we explored the interval [0,90] degrees in both directions to decide on the approximate values of Θ^1_{b1} and Θ^2_{b1}, decreasing the search interval each time after success or failure by 2. For each orientation the experiment was repeated 3-4 times.

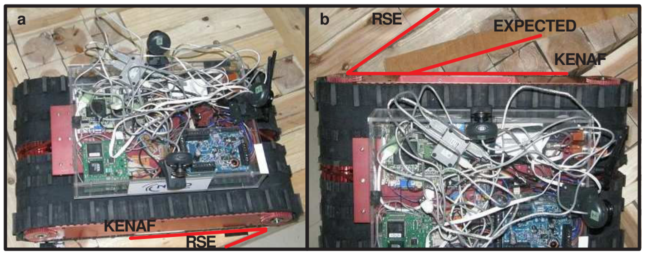

Fig. 16. Significant orientation change while climbing up toward OO1 at θ_{CPQ}=18.7 degrees: (a) initial configuration (b) final configuration

Climbing up experiments at angles, close to a straight angle, showed that for θ_{CPQ} in a range of [85,95] degrees KENAF is slipping on the spot for a while, adjusting its orientation, and starts climbing only after obtaining almost a straight angle value of θ_{CPQ}. So we decided to forbid CPQ angles in ranges [80,90) and (90,95] degrees. A small inessential adjustment within few degrees toward straight angle occurred also when experimenting in the range [80,60] degrees after the first contact of the crawlers with the barrier.

At θ_{CPQ} less then 20 degrees instead of climbing up KENAF slipped on the spot gradually increasing its θ_{CPQ} angle. As θ_{CPQ} reached 35-40 degrees, KENAF started climbing and after OO1 transition on the top of the obstacle it already had a different orientation θ and θ_{CPQ}. This behavior repeated in all cases; while initially for climbing we were using the slow speed of 7cm per second, trying to rush the barrier on a higher speed leaded to the same results. Fig.16 demonstrates a climbing up experiment at θ_{CPQ} = 18.7 degrees[6]; red lines show the RSE cell orientation and the initial KENAF orientation(fig.16a). As the robot climbed the barrier, its orientation significantly changed from the initial one(fig.16b) and the θ_{CPQ} angle change was 16.3 degrees, increasing θ_{CPQ} to 35 degrees.

At angles $\theta_{CPQ} \in$ [20,40] degrees a significant adjustment up to 15 degrees toward 45 degrees angle occurred in about half of the cases for each experiment. At angles $\theta_{CPQ} \in$ [40,45] the adjustment was relatively small. So we set Θ^1_{b1}=40 degrees and thus the good choice of CPQ angle θ_{CPQ} for climbing up and for $O_1 \rightarrow O_2$ transition at OO1 is $\theta_{CPQ} \in [40,80] \bigcup 90$ degrees.

For OO5 at angles θ_{CPQ}=75 degrees and less instead of switching to going down mode KENAF was very close to turning upside down; fig.17 shows the example of the OO5

[6] Angle's high level of precision was obtained after proceeding the photographs and videos of the experiment

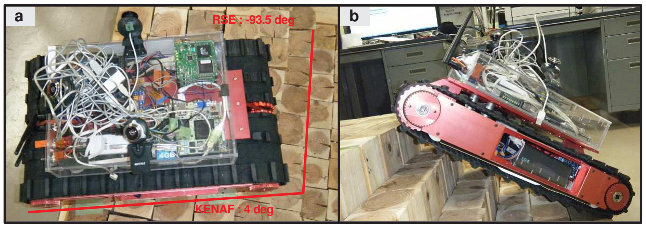

Fig. 17. Going down through OO5 at θ_{CPQ}=82.5 degrees, using a tricky horizontal barrier: (a) precise angle detection (b) final configuration

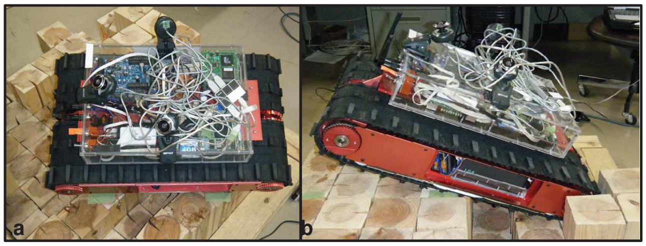

Fig. 18. Identical to simulation experiments: (a)OO9 IOP-2 (b)OO10 IOP-2

experiment. Finally we set Θ^2_{b1}=80 degrees; this leaves only a 20-degrees interval for successful going down CPQ angles θ_{CPQ} ∈[80,100]. However, on the top of the barrier, we usually have a possibility to adjust the orientation and θ_{CPQ} with few rotational steps so that the further process of loosing balance on purpose while going down will be smooth and safe.

D. *Experimental proof of existence for undesirable $O_1 \rightarrow O_2$ sequences*

The goal of this set of experiments was to confirm with a number of examples that undesirable transitions OO4, OO6, OO9 and OO10 are still possible, but dangerous and hard to repeat by a human operator. Since those situations are rare relatively to common OO1 and OO5 cases, as an input we used appearances of the undesirable transitions within the simulator and had each time to rearrange the RSE configuration respectively. For this reason such experiments have a serious lack of spanning a general generic case, but could only confirm the possibility of a particular transition. Fig.18(a) shows an experiment on OO9 IOP-2 case, corresponding to a simulated example of fig.12 and fig.18(b) shows OO10 case corresponding to fig.13. Each experiment for the existence was repeated 10-15 times, since even positioning KENAF more or less precisely at the start posture S did not give exactly the same results every time, emphasizing the complexity of repeating those types of $O_1 \rightarrow O_2$ transitions by the human operator.

VIII. 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.

The search algorithm within the graph requires a proper definition of neighboring states to ensure smooth exploration of the search tree. In this paper we presented our results in estimation of inertial lose of the balance on purpose transition possibilities between two consecutive states. It is an important step toward a proper definition of a search tree neighborhood function $F(Args) = Res$, where arguments $Args$ are the robot's current configuration and the environment and output Res is a set of accessible within one step configurations.

At first glance the experiments revealed much difference between the statistical simulation and experimental results in definition of AOP and IOP states. Actually the goals of the simulation were to model all possible cases of the robot's behavior on RSE, to collect enough statistical data to split up all possible cases into legal and illegal transitions groups, to structure and to analyze our theoretical approach. The experiments were used for the confirmation of the group distribution and updating the simulation with real world results. Since the goal of experiments was to correlate the simulation with real world results, there is no contradiction between their outcomes, but a system update. The feedback from the experiments will be directly implemented into our "pilot system". Dangerous and unpredictable cases of losing the balance on purpose during the path search will be excluded from the search tree, while well predictable and structured cases will become important turning points of a path.

References

[1] T. Bret and S. Lall, A Fast and Adaptive Test of Static Equilibrium for Legged Robots, *IEEE ICRA'06*, 2006, pp 1109-1116.

[2] T. Cormen and C. Leiserson and R. Rivest and C. Stein, Introduction to Algorithms, 2d Ed.,*The MIT Press and McGraw-Hil*, 2001.

[3] S. Hirose and H. Tsukagoshi and K. Yoneda, Normalized energy stability margin: generalized stability criterion for walking vehicles, *Proc. of 1st Int.Conf. On Climbing and Walking Robots*, Brussels, 1998, pp 71-76.

[4] A. Jacoff and E. Messina and J.Evans, Experiences in Deploying Test Arenas for Autonomous Mobile Robots, *Proc. of the 2001 PerMIS Workshop, in association with IEEE CCA and ISIC*, 2001, Mexico City, Mexico.

[5] J. C. Latombe, Robot Motion Planning, *Kluwer Academic Publishersx*, USA, 1991.

[6] E. Magid and K. Ozawa and T. Tsubouchi and E. Koyanagi and T. Yoshida, Rescue Robot Navigation: Static Stability Estimation in Random Step Environment, *Proc. of Int.Conf. on SIMPAR*, 2008, Venice, Italy, pp 305-316.

[7] E. Magid and T. Tsubouchi, Static Balance for Rescue Robot Navigation: Translation Motion Discretization Issue within Random Step Environment,*Proc. of ICINCO*, 2010, Madeira, Portugal, pp 415-422.

[8] R. Sheh and M. Kadous and C. Sammut and B. Hengst, Extracting Terrain Features from Range Images for Autonomous Random Stepfield Traversal, *IEEE Int. Workshop on Safety, Security and Rescue Robotics*, Rome, Sep.2007, pp 1-6.

[9] S. Shoval, Stability of a Multi Tracked Robot Traveling Over Steep Slopes. *IEEE ICRA '04*, Vol.5, 2004, pp 4701-4706.

[10] J. Wang, M. Lewis, and J. Gennari, Interactive Simulation of the NIST USAR Arenas, *Proc. of IEEE SMC*, Washington, DC, Oct.2003, pp 1350-1354.