Controlled Balance Losing in Random Step Environment for Path Planning of a Teleoperated Crawler-Type Vehicle

Evgeni Magid and Takashi Tsubouchi
Intelligent Robot Laboratory, University of Tsukuba, Tsukuba 305-8573, Japan
e-mail: evgeni@roboken.esys.tsukuba.ac.jp, tsubo@roboken.esys.tsukuba.ac.jp
Eiji Koyanagi and Tomoaki Yoshida
Future Robotics Technology Center, Chiba Institute of Technology, Chiba 275-0016, Japan
Satoshi Tadokoro
Graduate School of Information Sciences, Tohoku University, Sendai 980-8579, Japan

Received 26 November 2010; accepted 1 June 2011

Rescue robotics is the application of robotics to the search and rescue domain, aimed at extending the capabilities and increasing the safety of the rescuers. Deployed on a site during a rescue mission, a mobile robot is teleoperated by a human operator from a safe place. To suggest to the operator a good direction to traverse the three-dimensional (3D) debris environment, we develop a pilot system, which requires a special path search algorithm on debris and a proper definition of a search tree. Although 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 change a 3D orientation discontinuously through losing its balance. Losing balance on purpose is an essential feature for safe climbing up and going down debris, and it is the central issue of this paper. Exhaustive simulations were used to structure and analyze data. Experiments with a real robot verified our approach to removing unsuitable search directions from the search tree and gave important feedback to the algorithm. © 2011 Wiley Periodicals, Inc.

1. INTRODUCTION

Rescue robotics is the application of robotics to a broad number of domains where there is a requirement to extend the capabilities of human rescuers and increase their safety: urban search and rescue (USAR), planetary and underground exploration, military applications, etc. During a rescue mission a robot teleoperated from a safe place is deployed at the disaster site for initial exploration purposes, substituting for a human scouting team. The teleoperated system consists of a remote operation station [Figure 1(a)] and a mobile robot platform [Figure 1(b)], connected by a wireless LAN.

To find a good trade-off between teleoperation and full autonomy of a rescue robot is a serious challenge (Birk & Kenn, 2002), and due to the complexity of the task in a real-world rescue scenario, teleoperation control mode (Choi et al., 2007) is still preferable to autonomous (Baltes & Anderson, 2003; Birk, Markov, Delchev, & Pathak, 2006) or semiautonomous (Ohno, Morimura, Tadokoro, Koyanagi, & Yoshida, 2007; Suthakorn et al., 2009) rescue robot systems. Teleoperation, manual control by a human operator at a distance that is too great for the operator to see what the robot is doing (Murphy, 2000), requires a constant interaction of the robot with the operator for a path and victims search. The off-site operator cannot use any of the natural biological sensors as if he/she was a driver inside the rescue vehicle. Instead, the operator has to judge the next move on the basis of the limited visual information from the cameras and his/her previous operational experience, taking subjective and time-consuming path choice decisions. While searching a path, the operator has to remember, recognize, and diagnose the scene, considering the camera's positions and orientations; predict the situation of a robot getting stuck; and identify the victims, keeping in mind that any operational error could be fatal for the victim.

Various solutions to decrease the pressure on the operator include improving the operational graphical user interface (GUI), the development of a three-dimensional (3D) mapping sensory system (Zhang, Guo, Nejat, & Huang, 2007), gravity-referenced attitude display (Lewis & Wang, 2007), and improved interfaces with input information filtering (Schilling & Driewer, 2005). Another approach is to assist the operator through automation of particular tasks such as victim detection (Bahadori, Iocchi, Nardi, & Settembre, 2005), extraction of an occupancy grid map from a ranger finder sensor's data (Birk & Carpin, 2006), and even providing a basic autonomous system for both navigation and victim detection (Birk et al., 2006). Our proposed solution is an automatic pilot system, suggesting to the operator a fairly safe path between the current location and the target, which is our main long-term research goal. It will calculate the path with regard to the robot's static and dynamic properties and

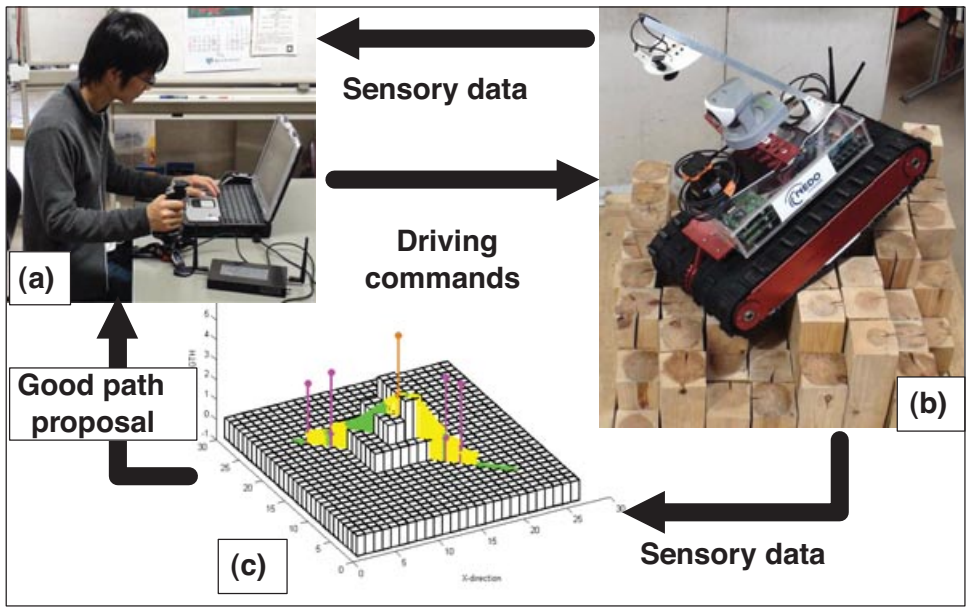

Figure 1. Standard framework includes operator (a) and rescue robot on RSE (b); we propose enhancing the standard scheme with our pilot system (c).

suggest to the operator a good path by means of GUI [Figure 1(c)]. Yet the final decision as to which path to apply in a real scenario driving the robot is taken by the operator.

In this paper we present the estimation of losing balance on purpose within a simulated debris environment, called random step environment (RSE) or random step field [Figure 1(b)] proposed by the National Institute of Standards and Technology (NIST) (Jacoff, Messina, & Evans, 2001). While searching a safe path through dangerous and unstable 3D debris, the main goal of any path search algorithm is to keep the robot maximally stable at every step of its path. However, in some specific cases the robot must be able to lose its balance and to change discontinuously its 3D orientation. Without this kind of transition the robot cannot successfully climb obstacles and in general is not suitable for debris exploration. To decrease the number of search directions and make the search feasible, we discretize the robot's motion and the huge real state space before the actual search (Magid & Tsubouchi, 2010b). A search algorithm utilizes a search tree (Cormen, Leiserson, Rivest, & Stein, 2001); for our problem a dynamically created search tree cannot be explicitly presented as a skeleton (Russell & Norvig, 2003). To present it as a function $F(\text{Args}) = \text{Res}$, we need a proper definition of guiding the tree search function F; here arguments Args are the robot's current configuration and the local environment map and output Res is a set of configurations accessible within one step. The properly defined function F becomes a branching function of Steepest Ascent Hill Climbing, Dijkstra, A^*, or any other path search algorithm, which is out of scope of this paper. This paper presents the particular part of function F responsible for losing balance on purpose. Predictable balance loss transitions become a part of the path, and not enough predictable cases are excluded, gaining a control at each posture of the path. Our theoretical results were confirmed with exhaustive simulations and experiments, removing all unsuitable directions of the search from the search tree. A gap between the exhaustive simulations and the real-world experiments (Magid, Tsubouchi, Koyanagi, & Yoshida, 2010) provided us valuable feedback, which improved the algorithm for final detection and usage permissions of controlled balance losing transitions within the pilot system.

The rest of the paper is organized as follows: The system overview is provided in Section 2. Section 3 deals with the search space discretization issue. We discuss the issue of static equilibrium and present the qualitative estimation of an expected posture and posture's color labeling in Section 4. On the basis of our previous work (Magid et al., 2010), we extended our considerations for controlled balance losing motion. We describe the details of the controlled balance losing in Section 5 and categorize inertial transition groups in Section 6; these sections are the heart of the paper. Exhaustive simulations and experiments are described in Section 7. Section 8 presents new updates for posture-type detection, whose necessity was revealed by Magid et al. (2010). Finally, we conclude in Section 9.

2. THE SYSTEM SETUP

Specific mobility requirements of the USAR domain make crawler-type robots the most suitable choice; wheeled (Sato, Matsuno, & Shiroma, 2008), legged (Campbell & Buehler, 2003), and snake-type (Arai, Tanaka, Hirose, Kuwahara, & Tsukui, 2008) rescue solutions have a more limited usage. In the huge variety of crawler robot modifications in the USAR field (Carlson & Murphy, 2005), one of the state-of-the -art robots in the field is the Kenaf robot (de Hoog, Cameron, & Visser, 2010; Nagatani, Tokunaga, Okada, & Yoshida, 2008; Yoshida et al., 2010) and similarly configured robots (Mihankhah, Kalantari, Aboosaeedan, Taghirad, & Moosavian, 2009; Ohno et al., 2007; Suthakorn et al., 2009) with a heavy main body, consisting of two large tracks with a small gap in between and four supporting subcrawlers. For our simulation pilot system and experiments, we exploit a simple tractor-like crawler, nonreconfigurable robot with the centroidal location of robot's center of mass (CM) corresponding to the main body of Kenaf without subcrawlers [Figure 1(b)]. The specifications of Kenaf fully equipped for a rescue search mission in this basic configuration are presented in Table I.

USAR implies a dangerous, unstable debris site of a building heavily damaged by a disaster, and one of the most popular debris environment models is the so-called RSE or random step field, proposed by NIST (Jacoff et al., 2001). RSE is a set of random steps with equal width and depth but different heights [Figure 1(b)]. Similar to real debris behavior, a big variety of optional assemblies, easy setup, and storage made RSE attractive test arenas for evaluation of USAR robot performance and an obligatory test arena for RoboCup Rescue competitions (Sheh, Kadous, Sammut, & Hengst, 2007; Suthakorn et al., 2009; Wang, Lewis, & Gennari, 2003). Each rescue group uses its own RSE setup, which is more or less similar to an official RoboCup Rescue version. Our RSE is assembled from 85 × 85 mm wooden blocks of 0-, 90-, 180-, 270-, or 360-mm height; 0-mm height corresponds to the ground level around the RSE patch.

Table I. Kenaf specifications.

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	19.6 kg

3. SEARCH SPACE AND SEARCH TREE

Search tree function F(Args) decides on possible next steps of the robot from a given current location and orientation. In a standard two-dimensional (2D) navigation with each cell of the state space defined as "free" or "occupied" (obstacle), a transition between two free cells is automatically legal (Latombe, 1991). In our case of a 3D environment, "possible" states refer to statically stable and "impossible" states to statically unstable postures, as we will explain in Section 4. But even in the case of two adjacent "possible" states, the transition between them is not always possible: after checking two adjacent postures of the path, we must confirm the transition between them as well. To decrease the number of search directions, we discretize the robot's motion and the state space before the search; we have to admit losing some generality to make the actual path planning possible. We studied several levels of search space discretization for XY coordinates of the environment and concluded that discretizing each 85 × 85 mm cell of RSE into 5 × 5 cells of the internal robot map with the cell size of 17 × 17 mm is the best choice for our problem solution (Magid & Tsubouchi 2010b); we refer to it as DISC5 discretization.

Translation and rotation are the two motion patterns of the Kenaf robot. The *translation step* is defined as a one-cell-length step forward in the direction of the robot local frame's axis X_L (Figure 2). The *rotation step* is a 5-deg change in robot orientation θ, rotating the X_L axis clockwise (right) or counterclockwise (left) with respect to Z_L (Figure 2). At each node of the search tree, the branching function validates a three-neighborhood of the node—go straight or turn left/right. All impossible search directions, different for rotation and translation steps, are immediately cut off from the search tree, and the path search proceeds in a most promising direction. When a proper discretization level is chosen and search tree branching function F is well defined, a search algorithm based on F successfully operates in a discretized RSE.

4. STATIC BALANCE AND POSTURE ESTIMATION

In a 3D debris environment with a complicated definition of an obstacle, the path search algorithm should be able to tell whether a specific robot configuration is possible. This includes obstacle collision avoidance and situations when the robot is to keep the current configuration without sliding or turning upside down, i.e., the robot's posture must be stable.

4.1. Static Stability

For general vehicle stability a minimal necessary condition is static stability. In most papers dealing with stability and balance issues, the authors consider a legged robot walking on uneven terrain (Bretl & Lall, 2006; Hong & Cipra, 2006; Klein & Kittivatcharapong, 1990; Mason,

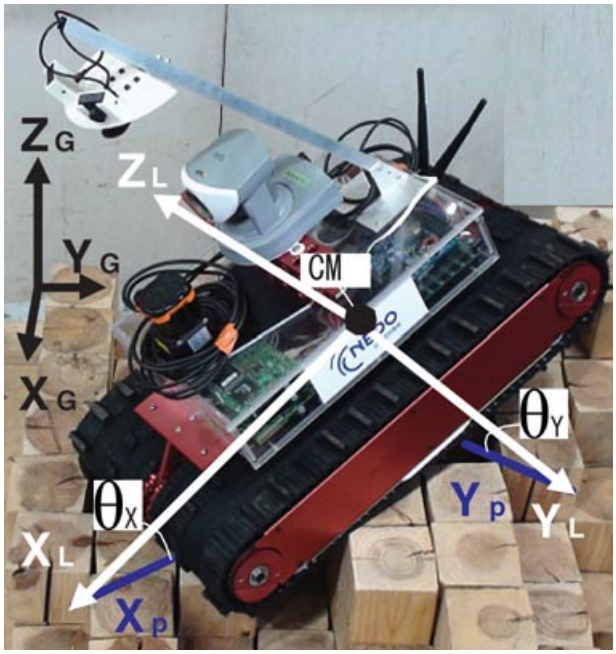

Figure 2. Steepness θ_X and moment θ_Y. X_G, Y_G, Z_G and X_L, Y_L, Z_L are global and local frame axes, respectively. X_p, Y_p are projections of X_L, Y_L on the XY plane formed by X_G, Y_G.

Rimon, & Burdick, 1997); in these cases a robot avoids falling by applying contact forces with its feet on the ground to compensate for gravity without causing a slip (a so-called *static equilibrium*). Assuming a slow enough motion to neglect inertia, the robot always must be able to achieve static equilibrium. Whereas for a flat terrain some simple heuristic tests are enough to check stability, an irregular and steep debris environment requires a check that the robot is in equilibrium at every posture.

To check static stability, we utilize a *support polygon*. Extending a legged robot's definition to our crawler robot, we define a support polygon as a 2D convex hull of the ground projection of all expected contact points of the robot's crawlers with the RSE. If the ground projection of a robot's CM stays inside the support polygon, we know that contact forces exist that achieve equilibrium, and we do not have to compute them explicitly. The specific features of RSE result mainly in the number of contact points, rather than the entire track contact, and constrain all contact points to lie on the edges and at the vertices of the environment cells and on the perimeter of the robot crawlers. Assuming sufficient friction between the crawlers and an obstacle surface, we define an appropriate posture of the robot, based on the literature survey and mobility experiments (Magid, Ozawa, Tsubouchi, Koyanagi, & Yoshida, 2008). If at least one of *conditions A* is not satisfied at posture K, posture K is not appropriate:

1. Both crawlers contact a terrain, and there are no contacts in the gap between them, thus escaping situations of getting stuck.
2. There are at least three distinct noncollinear contact points.
3. A surface inclination does not result in slipping or turning upside down.
4. A proper location of the robot's CM: no tip over due to CM displacement.

4.2. Qualitative Posture Estimation

In Magid et al. (2008) we presented an algorithm for static balance posture estimation of the robot's posture in a specified configuration, assuming the centroidal location of the robot's CM. In this section we briefly describe the static balance posture types and assign them color labels.

From the point of view of static balance estimation, we distinguish six posture types, presented in Table II. We define the *red state* that presents a forbidden posture. At the *magenta state* the robot has to climb up or to slide down a vertical slope of the environment (Figure 3). The *cyan state* is assigned for a robot's jumping up/down. To distinguish statically stable *green* and *yellow states* [Figure 4(a)] we apply normalized energy stability margin (NESM) (Hirose, Tsukagoshi, & Yoneda, 1998), which shows how statically stable the posture is: high-quality balance (G) or average quality (Y).

Last, *orange state* is something between red and green states. This posture is possible, but not stable. It does not result in a robot's turning upside down but does not guarantee a single stable posture because there exist two options, and the real one depends on the preceding posture and moving direction. Figure 4(b) demonstrates a side view

Table II. Color label explanation.

Label	Balance quality	Technical details
Red	Turn upside down or get stuck	Pitch $> \pi/4$ or roll $> \pi/6$ or at least one of conditions A (Section 4.1) is not satisfied
Orange	Lose balance on purpose	Two optional postures exist
Magenta	Climb up or slide down	Oscillations in posture estimation algorithm
Cyan	Jump down	Jump of CM between two stable postures < 50 mm
Yellow	Fair	NESM parameter < 1
Green	Good	NESM parameter ≥ 1

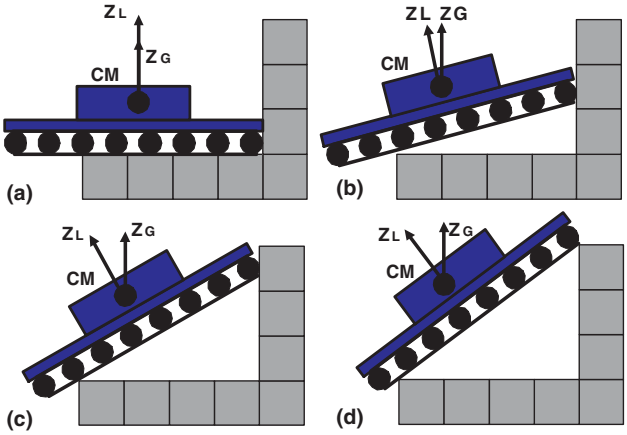

Figure 3. Magenta state: In climbing up mode robot moves from (a) to (d); in sliding down mode from (d) to (a).

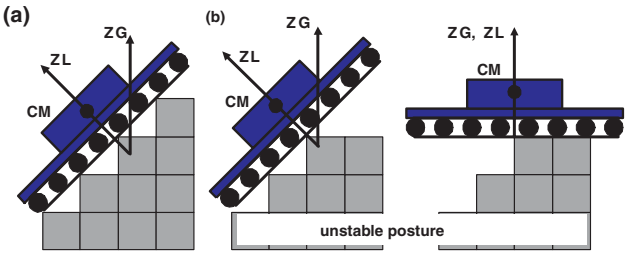

Figure 4. (a) Green state presents a stable posture. (b) Orange state is unstable because the robot's CM location results in two optional postures: O_1 (left) and O_2 (right).

of an orange state with two possible postures. The orange state is very important, because it allows the robot to lose balance on purpose when, for example, the robot traverses the barrier. Traversing the barrier includes climbing up and going down with losing balance twice on top of the barrier. We denote by O_1 the first part of the orange posture before the robot loses its balance. O_2 is the second part, which occurs after the robot has lost its initial balance; the robot changes its posture discontinuously at that point and obtains 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. Orange state is a very gentle state and should be used with special care in the translation motion step (Magid & Tsubouchi, 2010b; Magid et al., 2010) and be completely forbidden for a rotational step (Magid & Tsubouchi, 2010a). Orange posture is the central issue of this paper.

Further, we denote by *R posture* a posture in which static balance corresponds to a red state type, *O posture* for orange, *M posture* for magenta, *Y posture* for yellow, *G posture* for green, and *C posture* for cyan state type. The reader should not be confused by *Z posture*, which is not related to the color label and is described in detail at the end of Section 4.3. For verifying the proposed qualitative posture estimation, we constructed a simulation in a Matlab environment. The path proposed by the human operator in the given environment was analyzed with respect to static stability and compared with the expected results. Then the path evaluated as "good" by the simulator was repeated by the operator in a set of environments identical to the simulated ones, confirming that our qualitative discretization of the posture types is proper (Magid et al., 2008).

4.3. Describing a Posture

To characterize a robot's posture qualitatively, we use the color labels. To decide on legal transitions between two successive states, we use a combination of the five following variables.[1] *Steepness* θ_X is the angle showing the steepness of the RSE at the current robot configuration and rotational moment around the robot's transversal Y_L axis (Figure 2). *Moment* θ_Y is the angle indicating the current rotational moment around robot's lateral X_L axis. The choice of θ_X corresponding to Y_L and θ_Y to X_L emphasizes the tip-over direction in case of a failure. *Contact points quality* (CPQ) depends on the angle θ_{CPQ} between the robot's crawlers and the edges of the RSE cells and affects the robot's ability to climb obstacles, losing balance on purpose and going down safely. *Inclination* (Inc) is the steepness angle θ_X sign—positive while ascending and negative while descending. With respect to this parameter, we specify three groups of posture sets: a climbing up the steps of the RSE posture $\mathrm{Q}_{\mathrm{U_{inc}}}$, a going down posture $\mathrm{Q}_{\mathrm{D_{inc}}}$, and a neutral inclination posture $\mathrm{Q}_{\mathrm{Z_{inc}}}$. Thus, with respect to inclination, the postures are categorized into three types: Z (neutral), U (climbing up), and D (going down). *M sign* (MS) is the moment angle θ_Y sign; similarly group $\mathrm{S}_{\mathrm{P_{MS}}}$ is for all postures with positive M sign, and $\mathrm{S}_{\mathrm{N_{MS}}}$ with negative and $\mathrm{S}_{\mathrm{Z_{MS}}}$ with neutral postures.

Inc and MS signal about discretization problems, pointing to a missed posture between two successive postures; three other variables are emphasized for the experimental work. Combining Inc and MS, we define a neutral *Z posture*, a posture with the robot's body parallel to a horizontal patch of RSE: $Q_{\mathrm{Z_{inc}}} \bigcap S_{\mathrm{Z_{MS}}}$. Further we denote each posture as Col[Inc], where Col is the color and Inc is inclination. For example, we define G[Z] as a green neutral Z posture, G[U] as a G posture with $\mathrm{U_{inc}}$, and G as a G posture, covering all Inc cases if no other technical details are explicitly specified.

5. CONTROLLED BALANCE LOSING STATE

Kenaf supports translation and rotation types of motion, and the controlled balance-losing state naturally appears

[1] In Magid and Tsubouchi (2010a and 2010b) and Magid et al. (2010) we used six variables; however, after generalizing translation and rotation cases, we simplified this to five variables.

in both cases. In practice rotational motion in the simulation and in the real world differ a lot. The quality of the surface, the number of real contact points due to complicated interaction between an obstacle and a rubber crawler, the robot's speed, the accumulated error in the 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 a nontrivial and hardly predictable situation, so we completely forbid it for rotational motion and further deal only with a translation case.

5.1. 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 $(x \pm \delta_x, y \pm \delta_y, \theta)$ with $\delta_x, \delta_y \in (0, \varepsilon_{\text{shift}}]$, 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. Inertial transition through 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 in drastic path change and even in the robot's turning upside down. A detailed discussion on $\varepsilon_{\text{shift}}$ choice is presented in Section 7.2. For Matlab simulations in Section 7.1, only eight main directions of the numerical posture shift $(x \pm \delta_x, y \pm \delta_y)$ are chosen based on the properties of RSE: O-posture occurrence is strongly related to edges and vertices of RSE cells, and particular signs of AOP presence are described in Section 5.2.

Inevitable O-posture IOP-1 appears when the robot is passing through an edge of RSE in *close vicinity* to the RSE corner. Thus, a shift by δ in one direction preserves the current O posture, whereas a 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 by δ creates an identical O posture, whereas other directions produce G posture (mainly) or M/R posture (rarely).

[2]From here further we explain the idea of AOP, IOP, and $O_1 \rightarrow O_2$ transition groups of Section 6 with simple examples, whereas in both the simulation and real-world experiments the cases may be much more complicated.

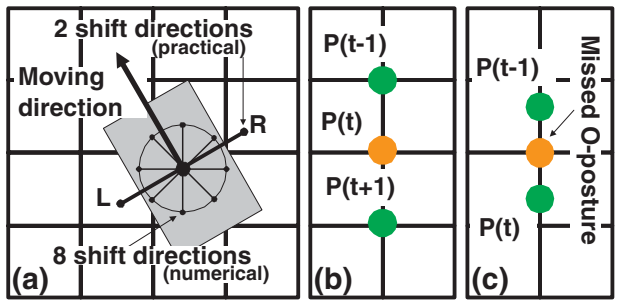

Figure 5. Discretization issue: (a) Shift for AOP/IOP detection, (b) explicit O posture, and (c) missed O posture.

Figure 5(a) presents eight numerical posture shift directions and two practical posture shift directions for the robot's body (the gray rectangle), explained in detail in Section 8. Figures 5(b) and 5(c) present the idea of discretizing a continuous path and evaluating the robot's postures at sample points: Figure 5(b) shows an explicit appearance of O posture between two G postures at $t-1$ and $t+1$ samples; Figure 5(c) displays a missed O posture due to a poor discretization level. AOP can rarely appear explicitly, but in most of the cases it is missed, being a kind of point-type singularity, and both types of IOP have to be continuous in at least one of the practical directions.

5.2. Discretization Issue and Forbidden Sequences

On the basis of a large set of experiments with the Kenaf robot in several RSEs and exhaustive simulations, we summarized the main properties of O posture. Because $O_1 \rightarrow O_2$ transitions should be applied with special care, we restrict the appearance of O postures within the search tree neighborhood definition function F. Owing to the level of discretization, in some cases we miss the explicit appearance of O postures. To overcome the imperfectness of the discretization and detect a missed O posture, we carefully compare any suspicious successive postures P_1 and P_2 (Magid & Tsubouchi, 2010b). The results showed that there is a missed intermediate O posture between first posture P_1 and second posture P_2 of the pair for G[Z] $\rightarrow$ G[D] (Figure 6) and G[U] $\rightarrow$ G[Z] translations. With an infinite level of discretization at a specific point of interest at the edge instead of sequence G[Z] $\rightarrow$ G[D] (Figure 6), we would obtain a chain G[Z]$\rightarrow$ O_1[Z] $\rightarrow$ O_2[D] $\rightarrow$ G[D] with the only difference between G[Z] and O_1[Z]: whereas for a human observer the robot's posture looks completely the same, the closer the robot gets from G[Z] to O_1[Z] the more unstable the posture becomes and finally at O_1[Z] the balance will be lost. Similarly, the posture gradually becomes more stable while moving away from O_2[D] to G[D]. In the three following

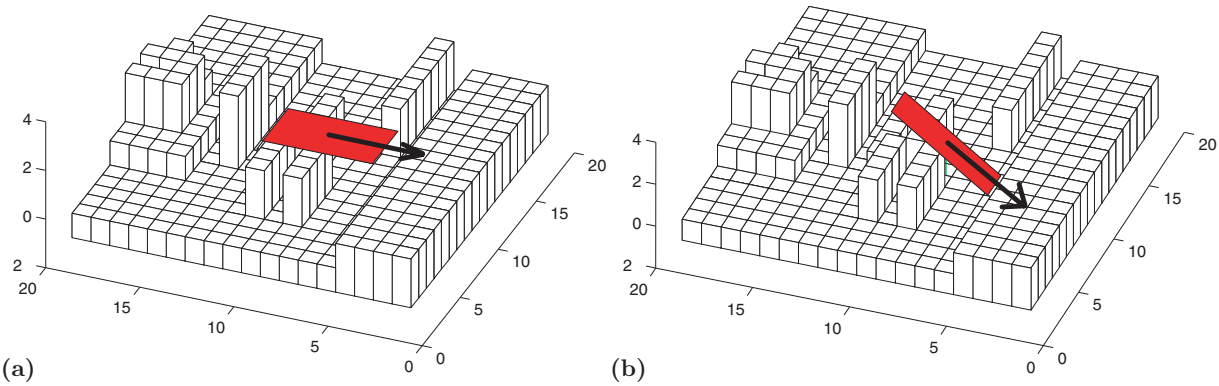

Figure 6. Translation from G[Z] (a) to G[D] (b), missing intermediate O posture; black arrows show the moving direction.

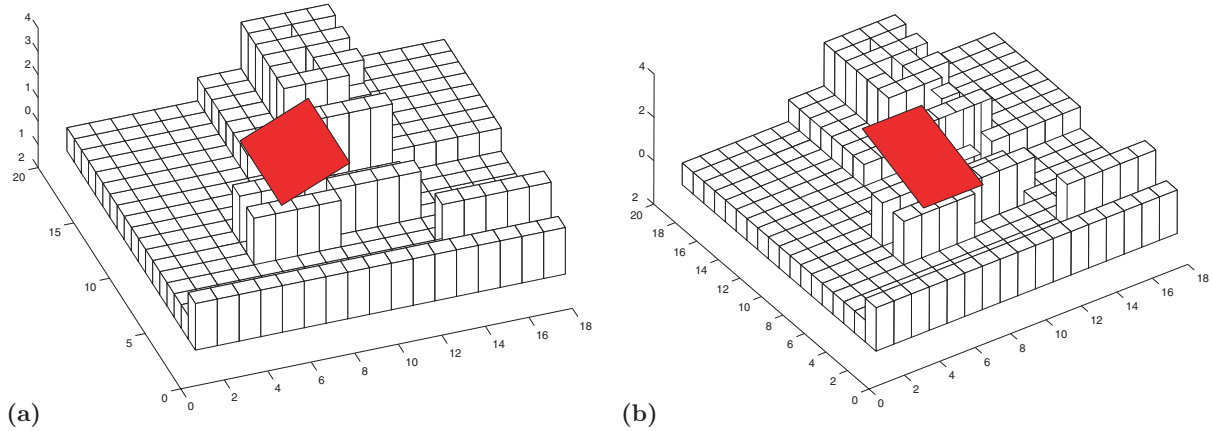

Figure 7. A hypothetically possible transition between two stable postures postures from (a) to (b) has a missing intermediate AOP type O posture, which means that the real transition is impossible.

cases there is a missed AOP between P_1 and P_2, both G or Y postures.

Missed AOP Transitions:

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

Another two special cases in which a rarely appearing missed O posture cannot be detected with the previously suggested methods (Magid & Tsubouchi, 2010b; Magid et al., 2010) are translations G[U] → G[U] and G[D] → G[D]. To solve this problem, we extend the previously declared cases as follows.

Extended Uniform Transitions:

1. Uniform climbing: G[U] → G[U], $\theta_X(P_2) \geq \theta_X(P_1)$, $|\Delta\theta_X| \leq T_{\text{MAX}}$, $|\Delta\theta_Y| \leq \varepsilon$.
2. Missed M posture at climbing: G[U] → G[U], $\theta_X(P_2) \geq \theta_X(P_1)$, $|\Delta\theta_X| > T_{\text{MAX}}$, $|\Delta\theta_Y| \leq \varepsilon$.
3. Missed O posture at climbing: G[U] → G[U], $\theta_X(P_2) \leq \theta_X(P_1)$, $|\Delta\theta_Y| \leq \varepsilon$; further AOP/IOP check is done (Section 8).
4. Uniform descending: G[D] → G[D], $\theta_X(P_2) \leq \theta_X(P_1)$, $|\Delta\theta_X| \leq T_{\text{MAX}}$, $|\Delta\theta_Y| \leq \varepsilon$.

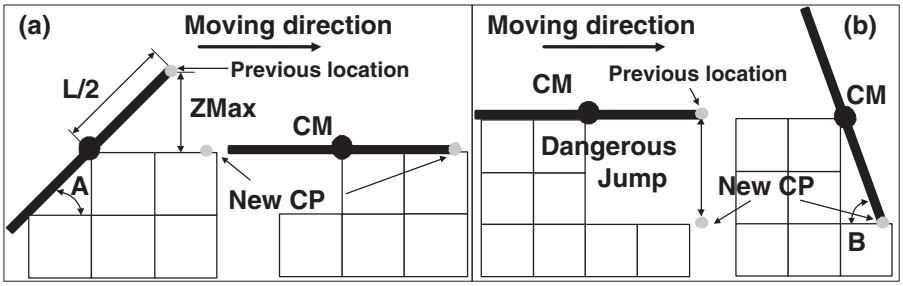

Figure 8. $O_1 \rightarrow O_2$ transition verification while climbing up (a) and going down (b).

5. Missed M posture at descending (i.e., sliding down): G[D] → G[D], $\theta_X(P_2) \leq \theta_X(P_1)$, $|\Delta\theta_X| > T_{\mathrm{MAX}}$, $|\Delta\theta_Y| \leq \varepsilon$.
6. Missed O posture at descending: G[D] → G[D], $\theta_X(P_2) \geq \theta_X(P_1)$, $|\Delta\theta_Y| \leq \varepsilon$; further AOP/IOP check is done (Section 8).
7. Missed AOP: G[U] → G[U] or G[D] → G[D] with $|\Delta\theta_Y| > \varepsilon$.

Even though extended uniform transitions of types 3 and 6 appeared only a few times among a total 6 million cases in the exhaustive simulation of the RSE, we expect a greater appearance in a real-world scenario. For all simulations we used a discretization DISC5. We calculated T_{MIN} by maximizing the difference $\Delta\theta_X$ between two successive G postures with the optimization method, updated it through a set of simulations, and finally set it to $T_{\mathrm{MIN}} \doteq 3.5$ deg; similarly $T_{\mathrm{MAX}} \doteq 8$ deg.

The three cases of missed AOP transitions together with extended uniform transitions of type 7 and 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: within just one step the robot will have to lose 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.

O posture is more important and has a higher cost in path planning, so in the case of O posture miss due to the discretization issue, we recolor the second posture P_2 of the sequence into O color and then determine its type. Because there are no real data for deciding on the O-posture type in this case, we initially had to use an approximation. On the final stage of adjusting the loop theory—simulation—experiments, a more complicated detection mechanism was developed (see Section 8).

To ensure a smooth transition from O_1 posture to O_2, we verify that there is no danger to the robot's sensors due to significant changes in θ_x, θ_y (Figure 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 [0 mm, 90 mm] and *dangerous interval* as [90 mm, 180 mm]. Maximal Z coordinate change for one of the contact points is the one that defines the interval for the whole posture. Although safe interval is preferable and dangerous should be minimized, any higher Z coordinate change will definitely destroy the sensors.

6. ORANGE TRANSITION GROUPS

Using parameters presented in Section 4.3, we define 10 distinct groups of all optional $O_1 \rightarrow O_2$ transitions resulting from 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 of transition. Figure 9 demonstrates the IOP-1 climbing case, when the robot's CM is too close to the vertex of RSE cell. Figure 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.

(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 physically impossible.

(OO4): O[U] → O[U]. A rare case, when the robot slightly changes orientation while climbing from one 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 the OO4 case are

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

[4]Rotation from neutral posture Z to the actual posture O_1 or O_2.

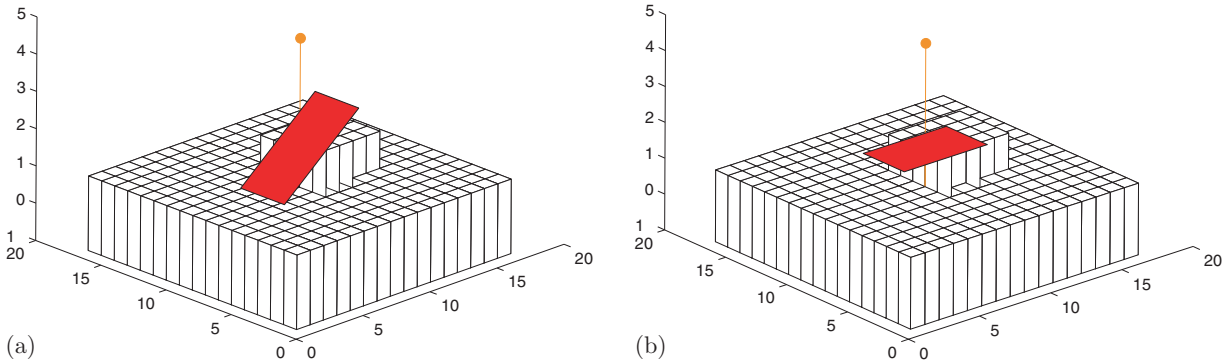

Figure 9. IOP-1: Robot's CM is too close to the vertex of RSE cell. OO1: Going down from (b) to (a). OO5: Climbing up from (a) to (b).

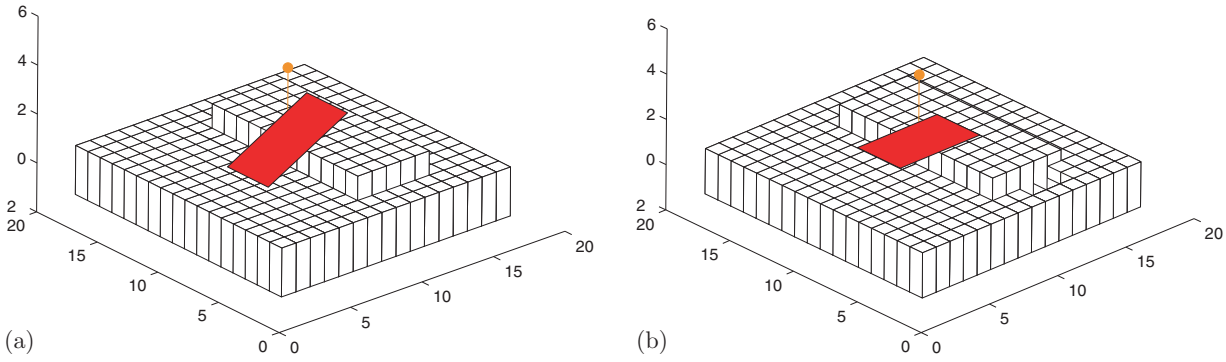

Figure 10. IOP-2. OO1: Regular safe descent from (b) to (a). OO5: Regular safe climb from (a) to (b).

as follows:

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

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

Further, we distinguish two subtypes of this transition:

A. If $\theta_X(P_2) < \theta_X(P_1)$, the robot will keep climbing up, and at the transition point the slope of the environment becomes less steep.
B. If $\theta_X(P_2) \geq \theta_X(P_1)$, the transition is physically impossible.

(OO5): O[U] → O[Z]. The robot started to climb up a barrier and upon reaching the top, as the CM approaches an edge of the barrier/RSE cell, it switches to a neutral posture on a flat pattern. Similar 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. Similar to case OO4, the change in orientation usually could hardly be noticed visually, and two subtypes of the transition are as follows:

A. If $\theta_X(P_2) > \theta_X(P_1)$, the robot will keep descending and at the transition point the slope of the environment becomes more steep.
B. If $\theta_X(P_2) \leq \theta_X(P_1)$, the transition is physically impossible;

(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.

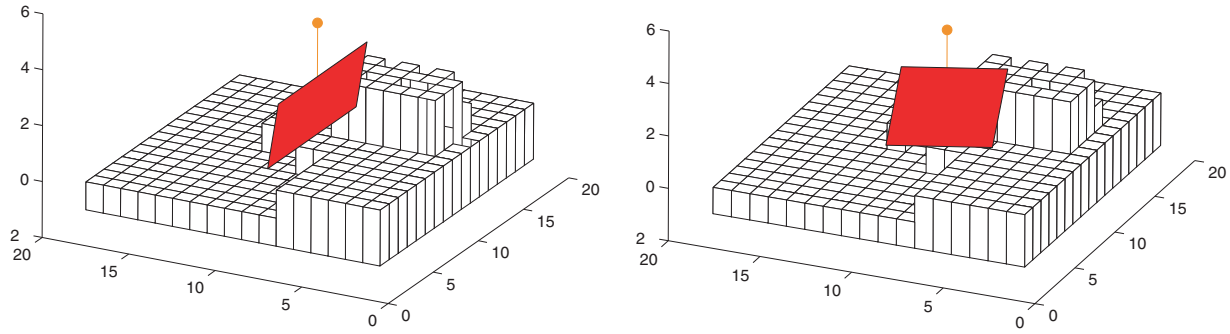

Figure 11. OO9 AOP case.

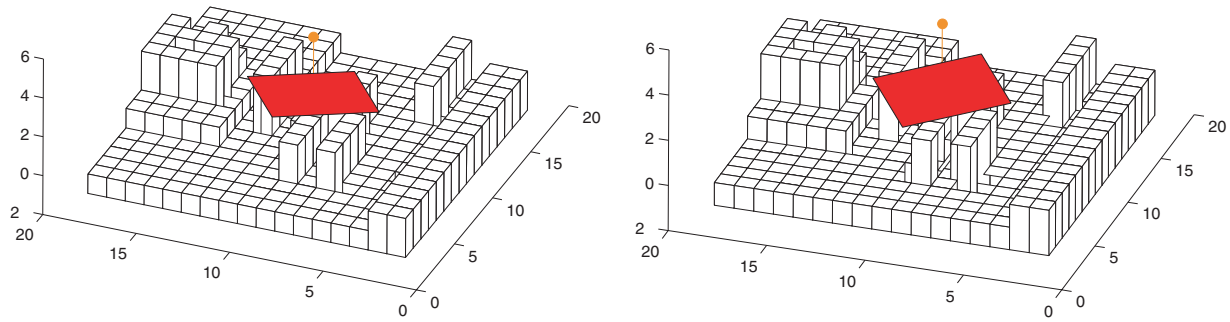

Figure 12. OO9 IOP-2 case.

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

(OO9): A group of postures in which the θ_y difference between O_1 and O_2 exceeds 5 deg and neither O_1 nor O_2 is neutral. In some cases the OO9 transition arises due to accumulated numerical errors, and after an additional posture check it turns out to be not O posture but G posture. In other cases it is AOP, a big point-type change with unpredictable behavior (Figure 11). Simulations showed that this case is a rare one, and we decided to forbid it completely for IOP1 and try to avoid it for IOP2 (Figure 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 the next G posture (Figure 13).

We group OO cases into three clusters:

1. OO1 and OO5 are the two most regular cases of losing balance while climbing up and going down.
2. Cases OO4A, OO6A, OO9, and OO10 are rare cases, which we would like to avoid, yet they could be included as a part of the path if no better option exists.
3. Cases OO2, OO3, OO4B, OO6B, OO7, and OO8 theoretically could not exist in RSE and may appear in the simulation due to discretization and accumulated numerical error issues.

Because O posture is very important, we must take care to distinguish and partially forbid the appearance of AOP, IOP1, and IOP2 cases for regular and undesirable groups (Section 7.1); the detection procedure is described in Section 8.

7. 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

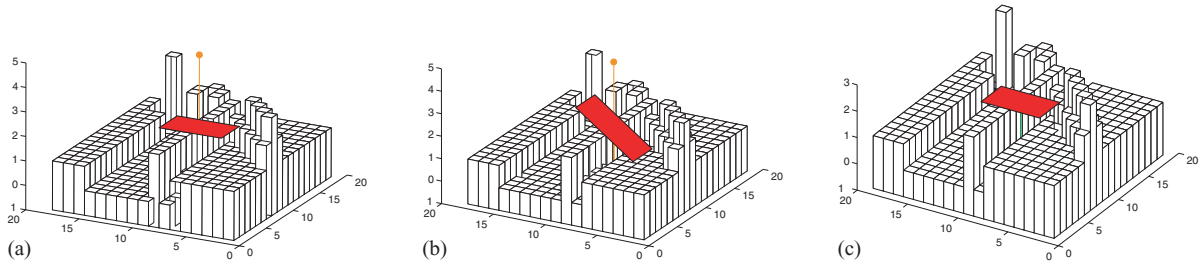

Figure 13. OO10 IOP-2 case: A missed M posture between O-posture inertial sequence O_1(a) $\rightarrow O_2$ (b) and the next G posture (c).

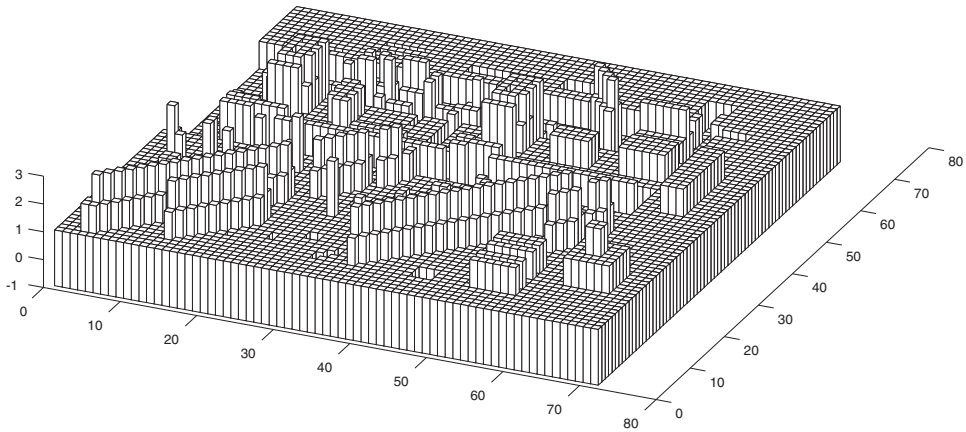

Figure 14. A complicated environment of size 71 $\times$ 71 cells, covering all main types of environmental obstacles.

impossible to execute such a huge number of experiments. The exhaustive simulations helped to confirm our hypothesis on impossible situations due to the physical rules of RSE. Pairs of postures impossible in the real world are also impossible within the simulation. Because the reverse is not true, the simulator cannot substitute for the experiments but helps to structure the data and remove the impossible types of sequences, saving time and effort. Exhaustive simulations for environment existence in MATLAB and experiments with the Kenaf robot in RSE gave valuable feedback for our theory and finally produced a branch cutting condition for the path search algorithm. Successive transition patterns of $O_1 \rightarrow O_2$ will be integrated in the search algorithm as a part of the neighbor opening and branch cutting function $F(\text{Args}) = \text{Res}$. At the end of this section we present a set of experiments in an unstructured 3D debris environment, which confirmed our results for RSE and pointed out some differences between unstructured real-world and RSE approximations.

7.1. Simulation Summary

To simulate all possible combinations of two sequent postures, we created a huge environment of 71 $\times$ 71 cells (Figure 14) that includes all typical obstacles, usually appearing in the RSE: horizontal and diagonal barriers, pairs of parallel barriers, valleys, and traversable and nontraversable pikes and holes. An exhaustive check of all possible pairs of neighboring postures connected with a translation step was done with voting for each group. For first robot posture of the pair, we took every node of the grid as CM location[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 $\theta \in \{0, \pi/180, 2\pi/180, \ldots, 89\pi/180, \pi/2\}$. In addition to pointing at the impossible (empty) cases, the simulation reveals the rare cases and the most common.

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

Table III. Orange transition types distribution.

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

Table IV. Main, undesirable, and forbidden transitions.

Transitions	Appearance percentage
Main	93.47
Undesirable	1.54
Forbidden	4.99

The total number of posture pairs was more than 6 million. Among them O posture appeared in 1.34% of the pairs—1.3% of them were further processed for detection of the O-transition type, and only 0.04% were immediately forbidden, as was explained in Section 5.2. As we expected, most of the $O_1 \rightarrow O_2$ transitions are of OO1 and OO5 type; they cover 97.6% of all cases; rare cases appear in only 2.1% of the cases, whereas the rest (0.2%) are forbidden cases (Table III). To distinguish IOP and AOP cases theoretically within the simulation, we used $\delta_x = \delta_y = 0.0017$ mm; the choice of δ for practical use is explained in Section 7.2.

The results of the simulation are summarized in the next tables, corresponding to the appearance of O postures. We listed as *main* transitions IOP1 and IOP2 cases of OO1 and OO5 (Table IV). *Undesirable* transitions include IOP1 and IOP2 cases of OO4,[6] OO6, OO10, and IOP2 cases of OO9 because we 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 were confirmed—all of them appeared due to the accumulated numerical errors. Whereas the IOP-1 case of the 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 transitions, which still may be theoretically possible in some rare cases, limits our path choice but increases security for the practical use.

[6]On the initial stage of the simulation, OO4 and OO6 were treated as undesirable, and only after the experimental stage were they subdivided into A and B subgroups.

Table V. IOP-2 transition type posture distribution.

Transition type	Appearance percentage
OO1	41.69
OO2	0.05
OO3	0.14
OO4	0.25
OO5	50.71
OO6	0.37
OO7	0.04
OO8	0.08
OO9	0.54
OO10	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 V); IOP1 is 1.07% and AOP is 4.68%. This means that forbidding the dangerous 4.68% of all AOP appearances and the unsuitable 0.31% cases of IOP1 and IOP2, we still have enough freedom for losing balance on purpose—95%, including 1.54% of undesirable appearances of OO4, OO6, OO9, and OO10.

7.2. Experimental Definition of AOP and IOP

To decide which $\varepsilon_{\text{shift}}$ 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 four orientations: {0,15,45,71} deg, 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 to the important issue of discretization and vulnerability of the computer simulation to 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 a decision on the type is not effective—we must also check the behavior of the previous and next to O posture poses of the path. If 3D orientations of previous and next postures shifted in two opposite directions would be coincident with the nonshifted 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) is not an option repeatable by the human operator, so the final choice of $\varepsilon_{\text{shift}}$ limit and δ for further pilot assistance was done experimentally.

For these experiments we created two types of RSE—a flat pattern and a horizontal barrier—and marked start S and target T postures (Figure 15). In the first set of experiments the robot traversed RSE from S to T at a given

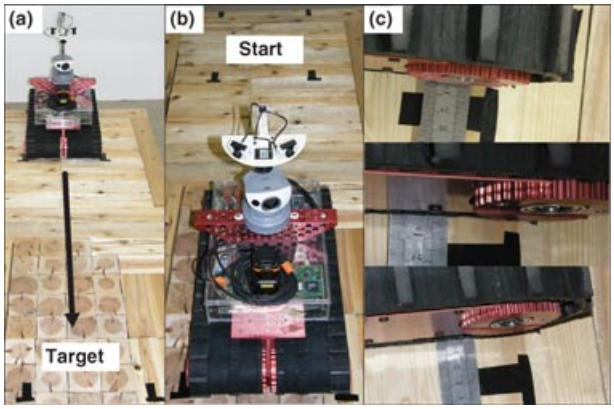

Figure 15. Side shift detection experiments: (a) start, (b) target, and (c) three different shifts at the target position.

Figure 16. Going down through CBL OO1 posture at $\theta_{CPQ} = 87.5$ deg, using a tricky horizontal barrier.

orientation, and we measured the side shift of the robot's corner points relative 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. Twenty-eight experiments were conducted for orientations $\theta = 45$ and 90 deg for a path length of 117 cm (two robot lengths), which is initially set as a short distance path planning range. In the worst case the shift was 77 mm together with a significant orientation change, whereas in all other cases the shift stayed within 20-mm range. Figure 15 demonstrates the side shift detection experiments at $\theta_{CPQ} = 90$ deg at a flat pattern RSE: start (a) and target (b) positions and three different shift results of the front right corner point of the crawler (c) at the target posture. Finally, we concluded that a side shift of 20 mm is a good choice for short-range path planning.

7.3. Experimental Constraints on Inertial Transitions

The goal of this set of experiments was to confirm the proper CPQ angles θ_{CPQ} for the main OO1 and OO5 cases. Whereas the choice of climbing up at OO5 is restricted by the climbing abilities of the Kenaf robot at M postures,[7] the choice at OO1 depends purely on the $O_1 \rightarrow O_2$ inertial transition. For both going down at OO1 and climbing up at OO5, the best CPQ angle θ_{CPQ} choice is a straight angle of 90 deg, whereas there exist angles Θ^1_{b1} and Θ^2_{b1} so that

- for all CPQ angles $\theta_{CPQ} \in [0, \Theta^1_{b1}]$, 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 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 (Figure 16) was to make Kenaf turn 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 deg for the θ_{CPQ} angle, we explored the interval [0, 90] deg 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 three to four times.

Climbing up experiments at angles close to a straight angle showed that for θ_{CPQ} in a range of [80, 100] deg 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, 100] deg. A small inessential adjustment within a few degrees of a straight angle sometimes occurred after the first contact of the crawlers with the barrier when experimenting in the range [80, 60] deg. At θ_{CPQ} less than 20 deg instead of climbing up Kenaf slipped on the spot, gradually increasing its θ_{CPQ} angle. As θ_{CPQ} reached 35–45 deg, Kenaf started climbing and after the OO5 transition on top of the obstacle it already had a different orientation θ and θ_{CPQ}. This behavior was repeated in all cases; whereas initially for climbing we were using the slow speed of 7 cm/s, trying to rush the barrier at a higher speed led to the same results. Figure 17 demonstrates a climbing up experiment at $\theta_{CPQ} = 19$ deg;[8] black lines show the RSE cell orientation and the initial Kenaf orientation [Figure 17(a)]. As the robot climbed the

[7] For the OO5 sequence the controlled balance losing follows immediately the climbing up process, which must succeed to advance the CM toward the O posture at the edge before the inclination of the robot's body becomes critical.

[8] Angle's high level of precision was obtained after processing the photographs and videos of the experiment.

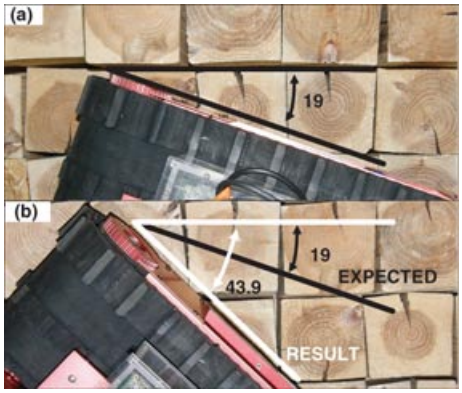

Figure 17. Significant orientation change while climbing up a horizontal barrier at $\theta_{CPQ} = 19$ deg: (a) initial configuration and (b) final configuration at the top (result) vs. expected.

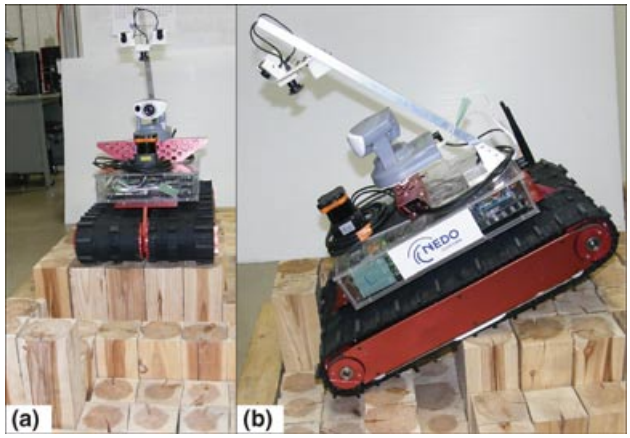

Figure 18. Identical to simulation experiment for OO10 IOP-2 case: (a) front view and (b) side view.

barrier, its orientation significantly changed from the initial one [Figure 17(b)] and the θ_{CPQ} angle change was 24.9 deg, increasing θ_{CPQ} to 43.9 deg; white lines show the RSE cell orientation and the final Kenaf orientation; black line the initial orientation, which was expected on the top of the barrier. At angles $\theta_{CPQ} \in [20, 40]$ deg a significant adjustment up to 15 deg toward a 45-deg angle occurred in about half the cases for each experiment. At angles $\theta_{CPQ} \in [40, 45]$, the adjustment was relatively small. In all cases the maximal self-adjusted θ_{CPQ} after climbing up the barrier was 44–45 deg, which defined the lower boundary of the successful climbing up interval $\Theta^1_{b1} = 40$ deg. Thus, a good choice of CPQ angle θ_{CPQ} for climbing up and for $O_1 \rightarrow O_2$ transition at OO1 is $\theta_{CPQ} \in [40,80] \bigcup 90$ deg.

For OO1 at angles $\theta_{CPQ} = 75$ deg and less, instead of switching to going down mode Kenaf was very close to turning upside down; Figure 16 shows the example of the OO1 experiment. Finally we set $\Theta^2_{b1} = 80$ deg; this leaves only a 20-deg interval for successful going down CPQ angles $\theta_{CPQ} \in [80, 100]$. However, on the top of the barrier, we usually have a possibility to adjust the orientation and θ_{CPQ} with a few rotational steps so that the further process of losing balance on purpose while going down will be smooth and safe.

7.4. Experimental Proof of Existence for Undesirable Inertial Transitions in RSE

The goal of this set of experiments was to confirm with a number of examples that undesirable transitions OO4, OO6, OO9, and OO10 in RSE are still possible but dangerous and hard to repeat by a human operator. Because these situations are rare relative to the 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. Figure 18 shows front (a) and side (b) views of the experiment on the OO10 IOP-2 case, corresponding to a simulated example of Figure 13. Each experiment for the existence was repeated 10–15 times, because 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. In addition transitions of the OO10 type in some cases showed an unstable behavior; for example, during the experiment presented in Figure 18 we repeated the same transition 16 times, and in five trials the robot succeeded in skipping a controlled balance losing O posture and next climbing up M posture and continuing toward the next G posture due to the motion's inertia and some displacement of CM from the ideal assumption.

7.5. Unstructured 3D Debris Field Experiments

The goal of conducting a set of experiments in an unstructured 3D debris field was to confirm the results obtained in RSE and to estimate the level produced by the RSE approximation with regard to a real-world rescue scenario. For this set of experiments, we created a garbage pile of about 2.5 $\times$ 2 m [Figure 19(a)], consisting of wooden blocks (cells of RSE) and concrete blocks (A, F, G), pipes of different diameters (I, J), metal plates (D, H), bricks and stones (B), pieces of furniture (E), clothes (K) and other litter. Prior to the experiments we defined a number of points of interest [marked with red dots in Figure 19(a)], where inertial (or some problematic) transitions were expected, and planned particular paths through those points. We conducted six types of experiments, trying to pass the pile from different directions [white arrows in Figure 19(a)] through the points of interest, and each type consisted of 5–10 trials with slightly different orientations.

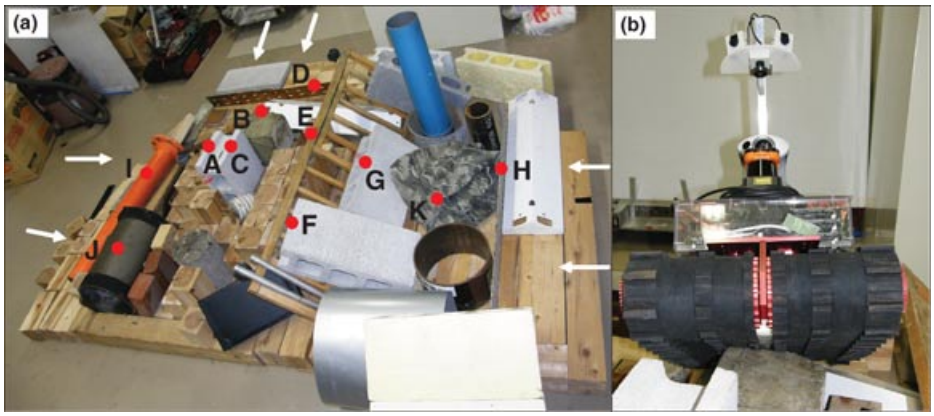

Figure 19. (a) Experimental pile: Red dots with letters mark points of special interest, and white arrows show six traversal directions. (b) OO5 IOP1, appearing at point B.

The main features of RSE that were reflected in our approach to deal with controlled balance losing inertial transitions are presence of sharp edges and corners of RSE blocks, straight angles, parallel to the ground, flat patterns, and a known a priori minimal size of each block. A real-world rescue scene may contain all these features as well but is not so strictly constrained. Sharp edges of real debris will produce exactly the same behavior as in RSE in some cases; for example, Figure 19(b) presents an appearance of OO5 IOP1 inertial transition at point B of Figure 19(a), where any shift right would preserve the posture, whereas shifting left will turn the robot upside down. Appearance of rear in RSE transitions such as OO4A and OO6A will significantly increase after removing the simulation limitations (Figure 20). Absence of the minimal cell size enables two inertial transitions in a row for traversing a thin, 10–20-mm-width barrier, such as the one appearing at point E [Figure 19(a)]. Together with the surfaces nonparallel to the ground, the lack of this limitation produces transitions OO3 (forbidden in RSE) while climbing up a very thin, few-millimeter barrier [like the one demonstrated in Figure 21 occurring at point H; Figure 19(a)] or losing balance on the sharp edge of an inclined obstacle [e.g., at point F or G; Figure 19(a)]. In other cases of the obstacles significantly different from RSE rules, Kenaf did not demonstrate any special behavior: it successfully climbed the rounded surfaces of the pipes of the restricted (Section 7.3) heights [Figure 19(a), I and J] and passed through nonrigid clothes without seizing [Figure 19(a), K].

8. EXPERIMENTAL FEEDBACK

The experimental results in Magid and Tsubouchi (2010b) revealed that a proposed AOP and IOP detection method is insufficient for practical purposes when a number of missed O postures significantly dominates over explicit O-posture appearances. A more complicated detection mechanism is required that is less dependent on search space discretization drawbacks and involves estimation of neighboring postures in the $\varepsilon_{\text{shift}}$ range from the point of interest.

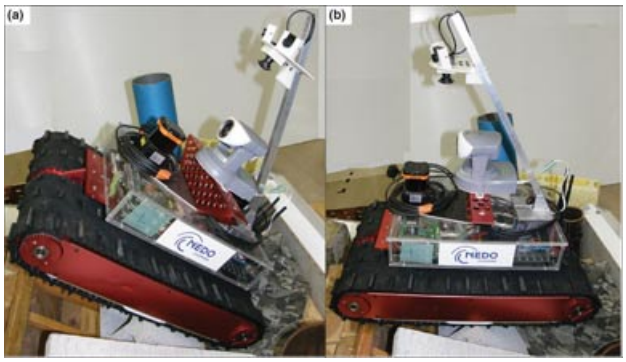

Figure 20. OO4A transition appears at point G: (a) O_1[U] and (b) O_2[U].

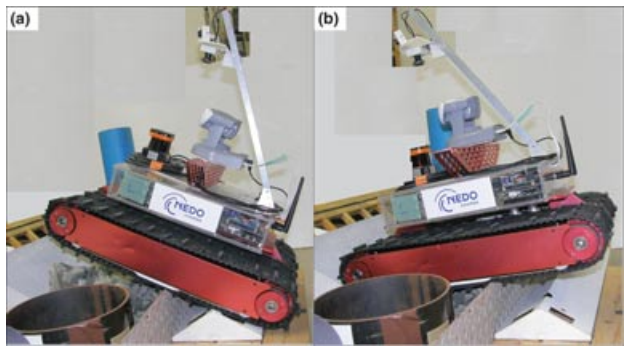

Figure 21. OO3 transition, forbidden in RSE, appears at point H: (a) O_1[U] and (b) O_2[D].

Experiments (Section 7.2) showed that a side shift of 20 mm is a good *practical* choice for short-range path planning, whereas frontal shift in the moving direction strongly depends on the experience of the operator and in general is easier to control.[9] Setting both shift variables to $\varepsilon_{\text{shift}} = 20$ mm, we use the following AOP/IOP detection algorithm:

1. Define P_1 and P_2 as follows:
   a. If O posture appears explicitly [Figure 5(b)] at time t as posture $P(t)$, set $P_1 = P(t-1)$ and $P_2 = P(t+1)$.
   b. If a missed O posture is detected in a time interval $[t-1, t]$ between postures $P(t-1)$ and $P(t)$ [Figure 5(c)], set $P_1 = P(t-1)$ and $P_2 = P(t)$.
2. Shift P_1 by $\varepsilon_{\text{shift}} = 20$ mm in two opposite directions[10] orthogonal to the robot moving direction—left P_1^L and right P_1^R shift [Figure 5(a), *practical* shift].
3. Estimate relationship between P_1, P_1^L, and P_1^R using θ_X, θ_Y angles. For each shifted posture P_1^S, $S \in \{L, R\}$ if $|\theta_X(P_1) - \theta_X(P_1^S)| \le \varepsilon$ and $|\theta_Y(P_1) - \theta_Y(P_1^S)| \le \varepsilon$, we report on a repeating posture; otherwise P_1 and its shift are considered to be different. Then:
   a. If both P_1^L and P_1^R are different from P_1, report AOP presence and exit.
   b. If one shifted posture P_1^S repeats P_1, we conclude on IOP-1 at P_1. Shifted direction is defined as $\overrightarrow{P_1 P_1^S}$. Proceed to step 4.
   c. If both P_1^L and P_1^R repeat P_1, we conclude on IOP-2 at P_1. Shifted direction is defined as $\overrightarrow{P_1^L P_1^R}$. Proceed to step 4.
4. Repeat step 2 for P_2.
5. Repeat step 3 for P_2.
   a. Repeat step 3a for P_2.
   b. Repeat step 3b for P_2. If shift directions at P_1 and at P_2 are the same,[11] conclude on IOP-1 at P_2 and proceed to step 6. Otherwise report AOP and exit.
   c. Repeat step 3c for P_2. Proceed to step 6.
6. If there was no report on AOP, make a decision on the IOP-1 or IOP-2 type referring to Table VI.

This simple detection mechanism successfully closes the gap between initial theoretical simulations trials with a numerical detection of the posture type at explicit appearances only and practical requirements revealed by the experiments.

Table VI. Decision on the O-posture type.

P_1 type	P_2 type	Decision on O posture
IOP-1	IOP-1	IOP-1
IOP-1	IOP-2	IOP-1
IOP-2	IOP-1	IOP-1
IOP-2	IOP-2	IOP-2

Another important feedback is received from the set of experiments conducted within the unstructured 3D debris environment (Section 7.5). In particular, the absence of any physical constraints or assumptions on the environment makes possible the inertial transition forbidden in RSE OO3 or two inertial transitions in a row following one another and significantly increases the number of transitions rare for RSE (OO4A and OO6A). This set of experiments marked future work directions for transferring the "pilot system" from a simplified RSE simulation of a rescue scene to a general elevation map approximation of a real-world rescue scenario.

9. 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 path selection within the internal model should be done, followed by applying this path in the real-world scenario. Because usually there exists more than 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 on estimation of inertial loss of balance on purpose transition possibilities between two consecutive states. It is an important step toward a proper definition of a search tree neighborhood function $F(\text{Args}) = \text{Res}$, where arguments Args are the robot's current configuration and the environment and output Res is a set of configurations accessible within one step.

The experiments revealed much difference between the statistical simulation and experimental results in the definition of AOP and IOP states. 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, and 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

[9]Permitting the robot to follow a path chosen by the operator autonomously in our future work will decrease a maximal frontal shift to few millimeters at a single straight segment of the path.

[10]Whereas in the theoretical algorithm we used eight shift directions at the explicit O posture itself with a very small shift value of $\varepsilon_{\text{shift}} = 0.0017$ mm to detect the *numerical* change, we verified that two significant shifts with $\varepsilon_{\text{shift}} = 20$ mm are enough in *practical* application.

[11]If P_1 has $\overrightarrow{P_1^L P_1^R}$ shift direction, any shift direction at P_2 is suitable.

results. The gap between RSE experiments and simulation refined the pilot system and produced a system update for the detection algorithm of different types of controlled balance losing transitions. Moreover, while closing the gap we redefined several permissions on acceptable and forbidden controlled balance losing transitions for the pilot system. Dangerous and unpredictable cases of losing balance on purpose during the path search will be excluded from the search tree, whereas well-predictable and structured cases will become important turning points of a path. The set of experiments conducted in the unstructured 3D debris environment marked future work directions for transferring the pilot system from RSE to an elevation map approximation of a general rescue scene. Even though our solution deals only with the static stability of a vehicle and suffers from a number of drawbacks and limitations such as strong assumptions on rigid and stable environment, absence of slippery and external disturbances, centroidal location of the robot's center of mass, and loss of some generality of the proposed path due to environment discretization, we believe that our unique approach to the path planning in RSE from the stability point of view and in particular the important property of losing balance on purpose makes a significant contribution to the rescue robotics domain.

ACKNOWLEDGMENTS

This work was partially supported by NEDO Project for Strategic Development of Advanced Robotics Elemental Technologies, High-Speed Search Robot System in Confined Space. The generous financial support of Atsumi International Foundation is gratefully acknowledged.

REFERENCES

Arai, M., Tanaka, Y., Hirose, S., Kuwahara, H., & Tsukui, S. (2008). Development of Souryu-IV and Souryu-V: Serially connected crawler vehicles for in-rubble searching operations. Journal of Field Robotics, 25(1), 31–65.

Bahadori, S., Iocchi, L., Nardi, D., & Settembre, G. P. (2005, September). Stereo vision based human body detection from a localized mobile robot. In IEEE Conference on Advanced Video and Signal Based Surveillance, Como, Italy.

Baltes, J., & Anderson, J. (2003, August). The Keystone Fire Brigade 2003. In AAAI Mobile Robot Competition Workshop, Acapulco, Mexico.

Birk, A., & Carpin, S. (2006). Rescue robotics—A crucial milestone on the road to autonomous systems. Advanced Robotics Journal, 20(5), 595–605.

Birk, A., & Kenn, H. (2002). A control architecture for a rescue robot ensuring safe semi-autonomous operation. In RoboCup-02: Robot Soccer World Cup VI. Berlin: LNAI Springer.

Birk, A., Markov, S., Delchev, I., & Pathak, K. (2006, August). Autonomous rescue operations on the IUB Rugbot. In IEEE SSRR, Gaithersburg, MD.

Bretl, T., & Lall, S. (2006, May). A fast and adaptive test of static equilibrium for legged robots. In IEEE ICRA, Orlando, FL.

Campbell, D., & Buehler, M. (2003, September). Stair descent in the simple Hexapod RHex. In IEEE ICRA, Taipei, Taiwan.

Carlson, J., & Murphy, R. R. (2005). How UGVs physically fail in the field. IEEE Transactions on Robotics, 21(3), 423–437.

Choi, Y., Jeong, K., Kang, J., Seo, Y., Lee, S., Jung, S., & Kim, S. (2007, October). A remotely operated mobile robot with modular track mechanisms. In ICCAS, Seoul, Korea.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2001). Introduction to algorithms (2nd ed.). Cambridge, MA: MIT Press and McGraw-Hill.

de Hoog, J., Cameron, S., & Visser, A. (2010, July). Dynamic team hierarchies in communication-limited multi-robot exploration. In IEEE SSRR, Bremen, Germany.

Hirose, S., Tsukagoshi, H., & Yoneda, K. (1998, November). Normalized energy stability margin: Generalized stability criterion for walking vehicles. In Proceedings of 1st CLAWAR, Brussels, Belgium.

Hong, D. W., & Cipra, R. J. (2006). Optimal contact force distribution for multi-limbed robots. Journal of Mechanical Design, 128(3), 566–573.

Jacoff, A., Messina, E., & Evans, J. (2001, September). Experiences in deploying test arenas for autonomous mobile robots. In PerMIS Workshop, Mexico City, Mexico.

Klein, C. A., & Kittivatcharapong, S. (1990). Optimal force distribution for the legs of a walking machine with friction cone constraints. IEEE Transactions on Robotics and Automation, 6(1), 73–85.

Latombe, J.-C. (1991). Robot motion planning. Boston: Kluwer Academic Publishers.

Lewis, M., & Wang, J. (2007, October). Gravity-referenced attitude display for mobile robots: Making sense of what we see. In IEEE SMC, Part A: Systems and Humans, Montreal, Canada.

Magid, E., Ozawa, K., Tsubouchi, T., Koyanagi, E., & Yoshida, T. (2008, November). Rescue robot navigation: Static stability estimation in random step environment. In Proceedings of SIMPAR, Venice, Italy.

Magid, E., & Tsubouchi, T. (2010a, November). Static balance for rescue robot navigation: Discretizing rotational motion within random step environment. In Proceedings of SIMPAR, Darmstadt, Germany.

Magid, E., & Tsubouchi, T. (2010b, June). Static balance for rescue robot navigation: Translation motion discretization issue within random step environment. In Proceedings of ICINCO, Madeira, Portugal.

Magid, E., Tsubouchi, T., Koyanagi, E., & Yoshida, T. (2010, October). Static balance for rescue robot navigation: Losing balance on purpose within random step environment. In IEEE IROS, Taipei, Taiwan.

Mason, R., Rimon, E., & Burdick, J. (1997, April). Stable poses of 3-dimensional objects. In IEEE ICRA, Albuquerque, NM.

Mihankhah, E., Kalantari, A., Aboosaeedan, E., Taghirad, H. D., & Moosavian, S. (2009, February). Autonomous

staircase detection and stair climbing for a tracked mobile robot using fuzzy controller. In IEEE ROBIO, Bangkok, Thailand.

Murphy, R. R. (2000). An introduction to AI robotics. Cambridge, MA: MIT Press.

Nagatani, K., Tokunaga, N., Okada, Y., & Yoshida, K. (2008, October). Continuous acquisition of three-dimensional environment information for tracked vehicles on uneven terrain. In IEEE SSRR, Sendai, Japan.

Ohno, K., Morimura, S., Tadokoro, S., Koyanagi, E., & Yoshida, T. (2007, October–November). Semi-autonomous control system of rescue crawler robot having flippers for getting over unknown-steps. In IEEE IROS, San Diego, CA.

Russell, S., & Norvig, P. (2003). Artificial intelligence: A modern approach (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Sato, N., Matsuno, F., & Shiroma, N. (2008). Development of a high mobility wheeled rescue robot with a 1-DOF arm. International Journal of Advanced Mechatronic Systems, 1(1), 10–23.

Schilling, K., & Driewer, F. (2005, July). Remote control of mobile robots for emergencies. In 16th IFAC World Congress, Prague, Czech Republic.

Sheh, R., Kadous, M. W., Sammut, C., & Hengst, B. (2007, September). Extracting terrain features from range images for autonomous random stepfield traversal. In IEEE SSRR, Rome, Italy.

Suthakorn, J., Shah, S., Jantarajit, S., Onprasert, W., Saensupo, W., Saeung, S., Nakdhamabhorn, S., Sa-Ing, V., & Reaungamornrat, S. (2009, February). On the design and development of a rough terrain robot for rescue missions. In IEEE ROBIO, Bangkok, Thailand.

Wang, J., Lewis, M., & Gennari, J. (2003, July). Interactive simulation of the NIST USAR arenas. In IEEE SMC, Washington, DC.

Yoshida, T., Nagatani, K., Koyanagi, E., Hada, Y., Ohno, K., Maeyama, S., Akiyama, H., Yoshida, K., & Tadokoro, S. (2010). Field experiment on multiple mobile robots conducted in an underground mall. Field and Service Robotics, Springer Tracts in Advanced Robotics (Vol. 62, pp. 365–375). Berlin: Springer.

Zhang, Z., Guo, H., Nejat, G., & Huang, P. (2007, April). Finding disaster victims: A sensory system for robot-assisted 3D mapping of urban search and rescue environments. In IEEE ICRA, Rome, Italy.