Stable Path Planning for Reconfigurable Robots Over Uneven Terrains

Mohammad Norouzi

School of Electrical, Mechanical and Mechatronic Systems

Faculty of Engineering and IT

The University of Technology, Sydney

A thesis submitted for the degree of

Doctor of Philosophy (PhD)

Supervisor	:	A.Prof Jaime Valls Miro
Co-Supervisor	:	Prof Gamini Dissanayake

May 2014

Declaration

I, **Mohammad Norouzi**, certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Student: _____

Date: _____

Abstract

Autonomous mobile robots are required to find safe and feasible routes in the environment when operating over challenging terrains. The most influential tip-over stability measures are based on two criteria; the robot's centre of mass (CM) and the support polygon defined by the convex area spanned between the ground contact points. The force angle (FA) stability margin is employed in this work given its widespread use and simple geometric interpretation.

A method to compute the contact points between a tracked robot and rugged terrain and predict robot's stability axes on 3D meshed maps reconstructed from 3D point clouds using the open dynamics engine (ODE) is presented. The validity and the need for stability computations based on the proposed contact points prediction algorithm is established through experiments over two common indoor obstacles i.e. ramps and stairs.

An analytical strategy to generate stable paths for reconfigurable robots whilst also meeting additional navigational objectives is hereby proposed. The suggested solution looks at minimizing the length of the traversed path and the energy expenditure in changing postures, and also accounts for additional constraints in terms of sensor visibility and traction.

A statistical analysis of stability prediction to account for the uncertainties associated with the actual robot's dynamic model, its localisation in the ground, and the terrain models is introduced. Probability density function (PDF) of contact points, CM and the FA stability measure are numerically estimated, with simulation results performed on the ODE simulator based on uncertain parameters. Two techniques are presented: a conventional standard Monte Carlo (SMC) scheme, and a structured unscented transform (UT) which results in significant improvement in computational efficiency.

A novel probabilistic stability criterion derived from the cumulative distribution of the FA margin is introduced that allows a safety constraint to be dynamically updated by available sensor data as it becomes available. The advantages of planning with probabilistic stability is demonstrated using a grid based A* algorithm as well as a sampling based RRT planner. The validity of the proposed approach is evaluated with a multi-tracked robot fitted with a manipulator arm and a range camera using two challenging 3D terrains data sets: one obtained whilst operating the robot in a mock-up urban search and rescue arena, and a second one from a publicly available on-line data from a quasi-outdoor rover testing facility.

This thesis is dedicated to my wife; *Minoo* for her love, patience, understanding, support and encouragement to finish this dissertation.

To *Aileen*, who
I adore her smile,
I cherish her hugs,
I admire her heart,
but most of all ...
I love that she is **my daughter.**

Acknowledgements

I would like to express my gratitude to my supervisors, Dr. Jaime Valls Miró and Professor Gamini Dissanayake for their support, patience, and encouragement throughout my graduate studies.

I would also like to express an enormous amount of gratitude to all the people from Centre for Autonomous Systems (CAS) and Faculty of Engineering and IT at UTS who have offered me help and support during my study.

Finally, and most importantly, I would like to thank my wife Minoo for her understanding and love during the past few years. Her support and encouragement was in the end what made this dissertation possible. I thank my parents, Zahra and Shokrollah, for their dedication and the many years of support during my undergraduate studies that provided the foundation for this work.

Contents

List of Figures

List of Tables

List of Abbreviations & Symbols

α	Heuristic factor for A* algorithm
βnom_i	The nominal β for the given ith tip-over axis $\mathbf{a_i}$
β	The normalised value of the force angle stability measure
β_μ	Mean of β
β_σ	Variance of β
β_{min}	The minimum stability margin
γ	Weighting factor for stability/reconfiguration or distance
$\mathbf{a_i}$	The ith tip-over axis
$\mathbf{f_i}$	The component of effective net force $\mathbf{f_r}$ which acts about the tip-over axis $\mathbf{a_i}$
$\mathbf{f_r}$	The effective net force
$\mathbf{l_i}$	The tip-over axis normal
$\phi_a nom$	Nominal arm angle
ϕ_a	Robot's arm angle
$\phi_f nom$	Nominal flippers angle
ϕ_f	Robot's flippers angle
$\mathcal{T}$	RRT tree
θ_i	The angle between $\mathbf{f_i}$ and the tip-over axis normal $\mathbf{l_i}$

L_a	Length of arm
L_f	Average length of flippers's CM
m_b	Total mass of robot's main body
m_f	Flippers's mass
m_h	Sensor head's mass
m_{tot}	Total mass of robot
n	Number of out-most contact points
nrj	Number of the robot's joints
nwp	Number of way-points in the trajectory
SC_{min}	Minimum safety confidence
U_c	The reconfiguration cost
Ud	Accumulated grid distance cost
Ue_i	The energy consumption cost
Up_i	The desired position cost
Us	Normalised stability cost
g	**g**ravitational acceleration
P	candidate points
TP	touched points
ASP	**A**ltered **S**upporting **P**olygon
CAS	**C**entre for **A**utonomous **S**ystems
CDF	**C**umulative **D**istribution **F**unction
CM	**C**entre of **M**ass

DEM	**D**igital **E**levation **M**ap
DS	**D**iagonal **S**tep-field
ERC	**E**nergy **R**econfiguration **C**
ESM	**E**nergy **S**tability **M**argins
FA	**F**orce **A**ngle stability measure
HS	**H**ill **S**tep-field
IMU	**I**nertial **M**easurement **U**nits
INS	**I**nertial **N**avigation **S**ystem
ISP	**I**deal **S**upport **P**olygon
ITM	**I**rregular **T**riangulation **M**eshes
LQG-MP	**L**inear-**Q**uadratic **G**aussian **M**otion **P**lanning
LRF	**L**aser **R**ange **F**inders
MHS	**M**oment **H**eight **S**tability measure
ODE	**O**pen **D**ynamics **E**ngine
OSHV	**O**ptimum **S**table **H**igh **V**isibility
PDF	**P**robability **D**ensity **F**unction
PRM	**P**robabilistic **R**oad**M**ap
ROS	**R**obot **O**perating **S**ystem
RRT	**R**apidly exploring **R**andom **T**ree
SC	**S**afety **C**onfidence
SMC	**S**tandard **M**onte **C**arlo
SRM	**S**tochastic **R**esponse **S**urface **M**ethod

SRR	**S**ample **R**eturn **R**over
SSM	**S**tatic **S**tability **M**argin
USAR	**U**rban **S**earch **A**nd **R**escue
UT	**U**nscented **T**ransform
UTIAS	**U**niversity of **T**oronto **I**nstitute for **A**erospace **S**tudies
UTS	**U**niversity of **T**echnology, **S**ydney
VSP	**V**ariable **S**upport **P**olygon
ZMP	**Z**ero **M**oment **P**oint

1

Introduction

1.1 Statement of Problem and Motivations

Mobile vehicles are frequently employed in unstructured environments where high levels of mobility are required. These systems have been performing a significant role in field environments, missions such as factory [12] and mining [13] automation, planetary surface exploration [10], agricultural vehicle [14], mobile manipulators [15] and search and rescue [2]. In such missions, avoiding tip-over is a major concern because it often results in collateral damage to the robot and the general surrounding environment [16]. Moreover, in most of these applications, since the safety of the robot and the people around are of prime importance, the robot should be equipped with a navigation system which is able to combine a guidance strategy to drive the robot's safely towards the goal. In general, this issue is not restricted to the autonomy aspects of mobile robot navigation, but also concerns tele-operated robots or those controlled by on-board human operators carrying heavy payloads. The safety and performance of a vehicle with these characteristics can be improved by automatically detecting and preventing tip-over instabilities. Some works have addressed the autonomous self-righting after a tip-over [17], however, such unstable events are dangerous for robot and by-standers i.e. *prevention is better than cure*.

Stability prediction is an important task for a successful safe path planning on challenging terrain. When the geometry of the terrain and the inertial properties of the vehicle are available, one of a number of well studied stability measures can be used to determine whether a vehicle can remain at a given point in the environment without tipping over. On the other hand, knowledge of the mechanical properties describing the complex vehicle-terrain interactions and/or a

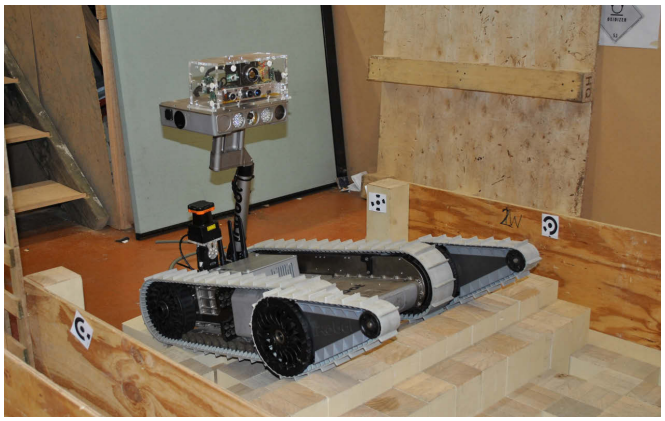


Figure 1.1: The iRobot Packbot explorer robot with a 1 DoF arm, pan-tilt sensor unit and two small front sub-tracks (flippers).

suitable traction control strategy are essential in guaranteeing that the vehicle is able to pass through a point in a given direction.

The robot's tip-over stability is a function of the robot's configuration, environment geometry as well the position of the robot over the terrain. The majority of the tip-over stability criteria are predicated on the position of the CM and the support polygon of the robot, defined by the contact points of the robot on the terrain. For an articulated rescue robot such as the iRobot Packbot shown in Figure 1.1, moving the on-board arm attached to the robot significantly impacts the position of the CM. Moreover, swinging the flippers (the two small front sub-tracks) can generally lead to changes in the support polygon. Furthermore, in specific scenarios such as search and rescue, other functions such as observing the environment become objectives for the robot, thereby making it desirable to position the sensor head as high as possible. This clearly leads to a higher CM and a potential reduction in stability.

Another major difficulty in robot navigation over challenging terrain is how to deal with the correlated noise in the data arising from the robot's sensors which results in uncertainty in both localization and mapping systems. While there are propositions that take into account the uncertainties associated with some parameters, most interpretations of the vehicle's stability still rely on deterministic analysis that assumes precise knowledge of the terrain physical model, robot's pose and vehicle true configurations. Yet generally only sparse (often occluded) uncertain terrain model estimates can be drawn from sensors such as point-cloud cameras, vision or 3D lasers while operating in realistic unstructured environments. Moreover, localisation in ruggedised 3D environments is significantly more challenging and uncertain than

in well-known structured areas. Also, a robot's configuration may not be accurately known given controller residuals due to noisy feedback data from the robot's joints and mechanical play. There is little research that explicitly addresses the challenges of robot stability analysis in uneven terrain while considering a wider range of the parametric uncertainties involved. The navigation system of a robot requires identifying which parts of the terrain are traversable. Moreover this system should also be able to predict the stable configuration during the path.

In this work then, the problem of stable path planning for reconfigurable mobile vehicles over rough terrains is addressed. The strategy is based on automatic reconfigurability by predicting robot-surface interactions, including articulated tracked mobile robots operating in non-homogeneous uneven terrains. The proposed algorithm tries to reposition the CM to prevent robot tip-over, while at the same time negotiates the additional constraint of keeping the arm head as high as feasibly possible with optimum reconfiguration cost. A statistical framework for stability prediction using the FA criteria is presented that takes into account a realistic set of the uncertainties that can be expected to be present when planning in unstructured domains. Through an iterative dynamics simulation process, it will be shown how a probabilistic representation of the contact points prediction and the stability margin can be derived. The technique is leveraged in the well known A* path planner [18] to generate routes that can direct the robot to move safely through the terrain, particularly accounting for out-of-bound areas where the terrain "roughness" is out of the range that can be accommodated by the robot change of posture ability (e.g., highly sloped terrains).

It is assumed that the location of the robot and 3D model of the environment is available through external means such as a 3D SLAM algorithm [19, 20, 21]. Furthermore, the significantly more complex issue of maintaining traction along the path is not considered by assuming that either the vehicle can be driven along the prescribed path by a human operator or that a suitable traction controller is available.

The validity of the proposed approach is evaluated with a multi-tracked robot fitted with a manipulator arm and a range camera using two challenging 3D terrains data sets: one obtained whilst operating the robot in a mock-up Urban Search and Rescue (USAR) arena, and a second one from an on-line data from the quasi-outdoor rover testing facility at the University of Toronto Institute for Aerospace Studies (UTIAS) [22].

1.2 Contributions

In summary the main contributions arising from the work on which this thesis is based are:

- An algorithm to predict robot-terrain interactions based on calculating the projection of the robot's underside on the points defining the terrain underneath in a dynamic physics simulator engine. The algorithm aims to derive and interpret intermittent robot-terrain contact points required to calculate the FA stability margin to improve the safety of robot mobility over ruggedised terrains. The methodology is particularly applicable, although not restricted to, reconfigurable tracked platforms that can actively assume safer poses to reduce potential instabilities, such as those leading to vehicle tip-over when operating in uneven terrains.

- An optimal high-visibility reconfiguration algorithm for the robot on uneven terrain. The proposed mechanism incorporate stability estimates when planning for safer high visibility navigational routes based on predicted robot-terrain interactions. In general, tracks and flippers should also remain in contact with the ground as much as possible to enhance locomotion [2]. In search and rescue applications for instance, the rationale is that the highest arm configuration will serve as a realistic proxy for the widest coverage paths in searching for victims, but this objective conflicts with the more stable postures which usually desire lower arm heights. To address these conflicting objectives, an analytical function to minimize the reconfiguration energy consumption cost whilst maximizing visibility, is introduced which is able to address the limitations in joints angular movement and velocity. The proposed cost function is employed in the well known A* graph search algorithm to enhance the capacity of the robot to explore the environment.

- A computationally efficient statistical approach for stability analysis of mobile robots under uncertainty concomitant with a mission in a challenging environment. The algorithm is capable of accounting for the uncertainties associated with the actual robot's dynamic model, its localisation in the environment, and the terrain models, particularly in uneven terrain. The proposed method has been validated using two techniques: a conventional

standard Monte Carlo scheme, and a structured sampling approach which shows significant improvement in computational efficiency.

- A statistical strategy for motion planning with stability uncertainty over rough terrains. The algorithm is able to exploit information gained from a statistical stability analysis to plan safe and effective routes under the presence of uncertainty in the robot's kinematics, terrain model and localisation on the terrain. Simulation results in an indoor rescue arena and an outdoor rover testing facility demonstrate how planning based on statistical information can result in more effective, and at the same time, safer routes when compared to a more deterministic stability planning approach.

1.3 Thesis Overview

This thesis is structured as follows:

Chapter 2 The detail of stability analysis methodology is presented in this chapter. In order to determine the most suitable criterion for the robot model exploited in this work, the most common stability margin algorithms are investigated. The mechanical detail of the robot model that was employed to validate the work hereby is introduced. The formulation of FA measure (as the best choice for this robot model) is explained in detail. The algorithm used to predict robot-terrain interactions to derive contact points required to calculate the FA stability margin is reported. An overview to the hardware and software design and implementation is provided. An experiment in a reconfigurable USAR arena is carried out to assess the validity of the approximate solution in a practical setting. The need for stability computations based on the proposed contact points prediction scheme when planning for safer navigational routes have also been established through experiments over two common indoor obstacles i.e. ramps and stairs.

Chapter 3 This chapter proposes a motion planning strategy for reconfigurable mobile robots operating in challenging environments. In some employment of mobile robots in an uneven terrain a high position of the arm (higher CM positions) is however preferred, i.e to afford wider sensor views in a search and rescue robot for instance. Another natural aim of the robot is to adopt its configuration to move CM to lower heights as it is possible to increase the safety margin. To trade-off between these conflicting objectives an

analytical function to generate stable paths for reconfigurable mobile robots is hereby proposed. The suggested solution minimises the length of the steady traversed path which also meets additional navigational constraints: energy expenditure in changing postures, sensor visibility (i.e arm configurations closer to those orthogonal to the horizontal global plane which can afford a wider sensor view) and traction (i.e. flipper angles that provide the largest track-terrain interaction area). This is particularly applicable to operations such as search and rescue where observing the environment to locate victims is the prevailing goal, although the proposed stable path planning technique can be generalized to incorporate other potentially conflicting objectives (e.g. maximizing ground clearance for a legged robot). The effectiveness of the technique in practice is shown in planning over two challenging adjacent diagonal and hill step-field arrangements. This experiment verifies how the proposed technique can reconfigure the robot to pass from uneven terrain which would not be traversable for such a robot with a constant posture, while at the same time also maintains the robot's safety from potential instabilities.

Chapter 4 This chapter proposes a statistical analysis of stability prediction to account for some of the uncertainties associated with many parameters in the planning pipe-line: the actual robot's dynamic model, its localisation in the ground, and the terrain models, particularly in uneven terrain. Probability density function (PDF) of contact points, CM and the FA stability measure are numerically estimated, with simulation results performed on ODE based on uncertain parameters. Two techniques are presented: a conventional Monte Carlo scheme, and a structured sampling approach which results in significant improvement in computational efficiency.

Chapter 5 This chapter proposes a probabilistic approach to account for robot's stability uncertainty when planing motions over uneven terrains. A novel probabilistic stability criterion derived from the cumulative distribution of a tip-over margin is introduced that allows a safety constraint to be dynamically updated by available sensor data as it becomes available. The proposed safety constraint authorizes the planner to generate more conservative motion plans for areas with higher levels of uncertainty, whilst avoiding unnecessary caution in well-known areas. The proposed systematic approach is particularly applicable to reconfigurable robots that can assume safer postures when required, although it is equally valid for fixed-configuration platforms to choose safer paths to follow. The advantages of planning with the proposed probabilistic stability margin are

demonstrated with data collected from an indoor rescue arena, as well as an outdoor rover testing facility.

Chapter 6 In this chapter, conclusions from the work described in this thesis are given, together with a summary of the findings of this research.

1.4 Publications

1. **M. Norouzi**, J. V. Miro, G. Dissanayake and T Vidal-Calleja. "Path Planning with Stability Uncertainty for Articulated Mobile Vehicles in Challenging Environments" In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Chicago, Illinois, USA, September 2014, pp. 1748-1753.

2. **M. Norouzi**, J. V. Miro, and G. Dissanayake. "Probabilistic Stable Motion Planning with Stability Uncertainty for Articulated Vehicles On Challenging Terrains" In *Autonomous Robots Journal*, under review.

3. **M. Norouzi**, J. V. Miro, and G. Dissanayake. "Optimal High-Visibility Stable Paths for Reconfigurable Robots On Uneven Terrain" In *Advanced Robotics Journal*, pp. 1-17, 2014, under review.

4. **M. Norouzi**, J. V. Miro, and G. Dissanayake. "Planning Stable and Efficient Paths for Articulated Mobile Robots On Challenging Terrains" In *Proc. Australasian Conference on Robotics and Automation (ACRA)*, Sydney, Australia, December 2013, pages 1-10.

5. **M. Norouzi**, J. V. Miro, and G. Dissanayake. "A Statistical Approach for Uncertain Stability Analysis of Mobile Robots" In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 2013, pp. 191-196.

6. **M. Norouzi**, J. V. Miro, and G. Dissanayake. "Planning High-Visibility Stable Paths for Reconfigurable Robots On Uneven Terrain" In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vilamoura, Algarve Portugal, October 2012, pp. 2844-2849.

7. **M. Norouzi**, F. D. Bruijn, and J. V. Miro. "Planning Stable Paths for Urban Search and Rescue Robots" In *Proc. of the RoboCup International Symposium*, Istanbul, Turkey, July 2011, vol. 7416, pp. 12.

also published as a book chapter:

8. **M. Norouzi**, F. D. Bruijn, and J. V. Miro. "Planning Stable Paths for Urban Search and Rescue Robots", Robocup 2011: Robot Soccer World Cup XV, Lecture Notes in Computer Science (LNCS), Röfer, T., Michael Mayer, N., Savage, J, Saranli, U. (Eds.), 2012, Springer, ISSN 0302-9743, pp. 90-101.

2

Stability Analysis

2.1 Introduction

Planning safe and reliable motions of an autonomous vehicle over rough terrain requires continuous satisfaction of the stability safety margin. For successful vehicle operation in inhospitable environments like disaster areas, it is necessary to consider an effective stability margin to evaluate the tip-over boundary of the robot whilst travelling over rough terrains. The robot's tip-over stability is a function of the robot's configuration, environment geometry as well the position of the robot over the terrain. The majority of the tip-over stability margins are predicated based on the net force and momentum acting on the system's CM and its position over the support polygon, defined by the out-most contact points between the robot and the terrain.

A common approach to control the robot's stability in the literature is to predict the support polygon solely from only inclination data obtained from an on-board IMU by considering constant contact points in the robot's frame. For example contact points are usually supposed to be under wheels or sprockets for wheeled [23] and tracked [16] robots respectively. In the case of a wheeled robot equipped with a suspension system capable of keeping wheels in constant contact with the terrain, this may be a fairly plausible presupposition. But this is a strong assumption for the case of tracked robots operating in highly unstructured terrains, such as those featuring rubble, oversized obstacles, stairs etc. In such uneven environments, contact points can lay anywhere along the track of the robot. A technique to predict the interacting forces between the robot-terrain to derive the contact points which results in a variable support polygon is presented in this chapter. The contact points prediction scheme is based on simulating robot-terrain behaviour in a dynamics physics engine.

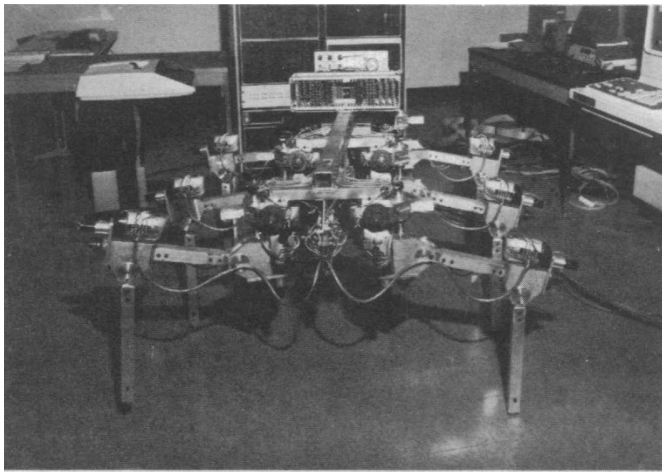


Figure 2.1: The OSU hexapod vehicle [1].

The detail of stability analysis methodology is presented in this chapter. In order to determine the most suitable margin for the robot model exploited in this work, the most common stability margin algorithms are investigated in the next section. The robot model that was employed to validate the work is introduced in Section 2.3. The formulation of FA measure (as the best choice for this robot model) is explained in Section 2.4. The methodology used to describe the shape of the 3D polyhedral terrain model for the simulator is detailed in Section 2.5. The algorithm to predict robot-terrain interactions for basic robot model is reported in Section 2.6. An experiment in a reconfigurable USAR arena is carried out in Section 2.8 to assess the validity of the approximate solution in a practical setting. Section 2.6.2 will have a look into the the contact points prediction algorithm for the complete robot model including flippers. The efficacy of the proposed contact points prediction algorithm for predicting the variable support polygon is verified through comparing the results of deriving Packbot robot with different configurations over a ramp and stairs in Section 2.8.2.

2.2 Review of Stability Margins

There have been a number of propositions to address the issue of stability in mobile robots. Stability indices have understandably played a decisive role in the history of walking robots like the example shown in Figure 2.1, and a number of measures have been proposed in the literature e.g. the static stability margin (SSM) [24] or the energy stability margins (ESM) [25]. The static stability of a walking vehicle was examined by SSM for the first time in 1968 [24].

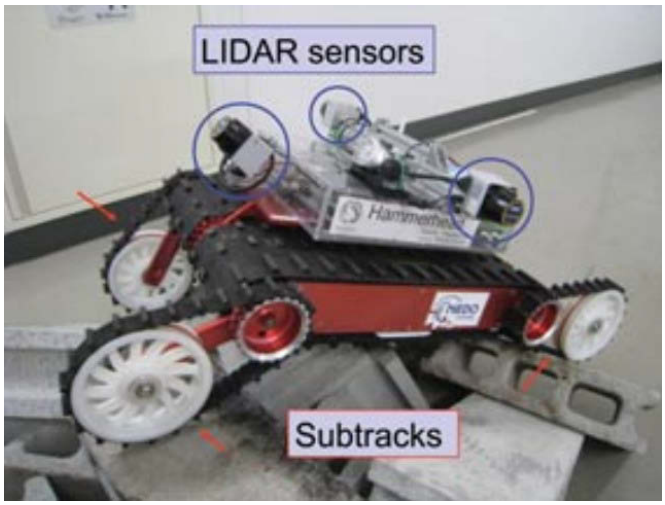


Figure 2.2: The Kenaf robot with three laser range finders [2].

It was claimed that the vehicle was stable if the horizontal projection of the CM lay within the support polygon. The corresponding SSM was defined as the smallest distance between the projected CM and the edge of the polygon. The SSM was later adapted to uneven terrain and slightly modified to reduce the complexity of calculation in [1]. The main disadvantage of these purely projective-based approaches was the insensitivity to the height of the CM. The ESM [25] solved this problem by determining the potential energy that was needed to tumble the vehicle and this represents a reliable static stability margin. This measure was normalised in [26] to obtain a more general and meaningful measure of stability.

More general approaches for the stability control of reconfigurable mobile robots have also taken into account other constraints, e.g. traction optimization or shared controls. The minimization of a performance index that considered the stability measure for each potential tip-over axis and the nominal values of the joints have been suggested to provide the most favourable configuration of the robot for increased traction in [28]. A combination of the stability measure with an artificial function to obtain the demanded actuator values was used in [29]. Both works were however inadequate for certain robot configurations, or in general for robots with low CM.

Some research has focused on the analysis of the robot's CM to find suitable controls to cope with specific scenarios like overcoming obstacles and small ditches in [30] or climbing stairs in [31]. A multi tracked robot on a steep slope was examined in [32] to determine boundaries for the CM and a strategy was devised to traverse a given slope. A real-time roll-over

Figure 2.3: Tracked mobile manipulator Alacrane (right) and its trailer (left) during an emergency response exercise [27].

protection strategy based on whole-body touch sensors that are embedded in the tracks of the robot and an energy stability margin that serves as an indicator for unsafe robot configurations has also been developed [33]. Stabilising actions to protect the robot from roll-over based upon empirical flipper movements were also proposed.

A shared autonomy controller, specifically for the case of tracked vehicles to be able to safely traverse rough terrain with active flippers has also been studied in [2]. The Kenaf's robot system comprised of a manual controller for the main tracks and an autonomous controller for the flippers is based on continuous nearby terrain scanning with laser range finders (LRF) as shown in Figure 2.2. The stability of the robot was evaluated using the normalised energy stability margin [34]. This stability criteria is based on the vertical distance between the initial position of the CM, and its highest position during tumbling. The proposed approach is not applicable for robots with only front or rear sub-tracks such as Packbot because their sub-tracks would not take strong control of the attitude of the robot's body.

An alternative real-time tip-over stability criteria for a reconfigurable tracked mobile platform on slopes was derived in [9] on the basis of load transfers by judging the supporting force generated at the concerned tracked-terrain contact points. The proposed avoidance algorithm considered the contact points to be fixed under the sprockets in order to describe the interactions between tracks and terrain.

More recent work [27] analysed the effect on static tip-over stability margins of towing a

Figure 2.4: A photograph of the modified Packbot Fido from iRobot used in tip-over validation tests [16].

single-axle trailer by a tracked mobile robot on slopes as shown in Figure 2.3. It introduced the concept of altered supporting-polygon (ASP) for both the leading tractor (robot) and the trailer. ASPs are based on a force-torque static equilibrium analysis that reshapes the corresponding ground contact supporting polygons. The tractor ASP is reshaped given the changes in the relative orientation between the units, and the position of both tractor and trailer CMs. In the case of a single vehicle subjected uniquely to gravity, its ASP coincides with its ground contact SP, and in the work the actual supporting-polygon is considered to be a fixed rectangular polygon, with four contact points under the main sprockets, hence a constant location in the robot's frame. Experimental results presented a scenario where the robotic trailer was guided manually on a light (5°) slope.

In the work recently published in [16] the three most common stability margin algorithms, zero-moment point (ZMP), [35], FA stability margin [36] and moment-height stability measure (MHS) [37] have been verified on the iRobot Packbot tracked robot, the same platform used in this work. The robot is shown in Figure 2.4. This study looked at operating the robot with a constant configuration over a small set of ramps and obstacles to evaluate how the tip-over margins were able to assess the stability of the robot. Both the FA and MHS proved to be more effective measurements of potential tip-over instabilities than ZMP.

The original FA measure considered the angles between the vector through the position of the CM and each tip-over axis, and the vector of the resulting force through the CM. This proved sufficient for robots with relatively high CMs that do not change significantly, but is not representative of the actual stability in many other cases, and was in general inadequate for

certain robot configurations, such as those with low CM, or those more strongly subjected to external forces and moments. A revised version of the original FA was proposed in [15] by Papadopoulos and Rey to allow for dynamical changes in the robot configuration. This constituted a more suitable stability measure for mobile robots/manipulators as it exhibited a more simplistic geometric interpretation and thus could be more easily computed. As explained, the margin was introduced in two different versions, and these are explained in the Section 2.4 to better understand the influence of the CM's height for platforms that can significantly reposition their CM to improve stability in uneven terrains.

A fuzzy controller for autonomous negotiation of stairs was presented in [38]. More recently a combination of a stability measure with an artificial function was proposed in [29] to obtain the demanded actuator values also in pursuit of traction enhancements. In both works, the original FA was used. Another active control strategy where the FA margin was used for a reconfigurable wheel-legged mobile robot was based on ground clearance, orientation, gradient stability margin, and wheel traction efficiency on irregular terrain [23]. Contact points were assumed located at the leg terminations resting on planar contact terrain. The performance of the active control strategy was shown to depend heavily on the quality of the feedback data, and wheel-terrain interactions were highlighted for revision.

FA was proven to be one of the most effective stability measures in the literature, and is generally more widely used in the robotics community as it exhibits a more simplistic geometric interpretation as well. Therefore the FA stability margin was found also to be a suitable choice for the robot model exploited in this work. It is briefly described in Section 2.4 for reference.

2.3 Robot Model

A large variety of tracked mobile vehicles have been designed to operate in irregular environments, from collapsed buildings to off-road natural terrains, deployed in a wide range of applications such as mining, search and rescue, forestry, earth moving or planetary exploration. In this kind of terrain, tracked locomotion has been confirmed to provide better mobility and controllability than wheeled or legged robots, yet is has also demonstrated that more complex track-terrain interactions are at play[9].

The multi-tracked iRobot Packbot robot [39] depicted in Figure 2.5 was employed to validate the work hereby proposed. The robot has been suitably modified for deployment in rescue

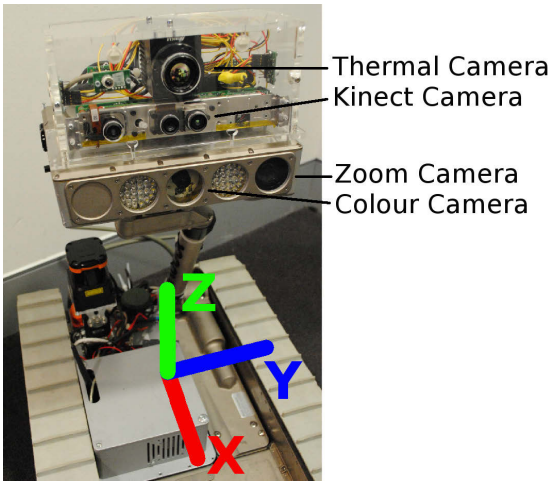


Figure 2.5: Packbot robot's coordinate system and the additional sensor head.

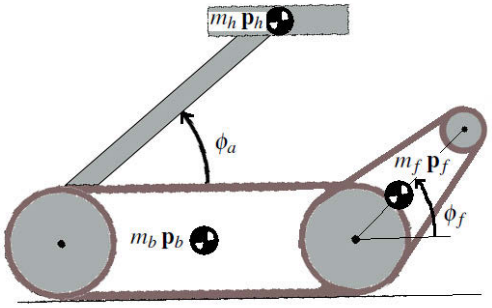


Figure 2.6: The side view of the lumped-mass model.

missions with additional sensors for victim detection, localization and mapping added to the off-the-shelf unit. It consists of a skid-steer vehicle base, two flippers and a manipulator arm attached via a 1 Degree of Freedom (DoF) shoulder joint. It carries a 2-DoF pan-and-tilt unit equipped with several cameras and lights. The robot is battery powered and features battery compartments on the left hand side. More details about the hardware and software system are provided in Section 2.7.

The robot's CM with respect to the robot's frame is defined by

$$\mathbf{CM} = \frac{\sum_j \mathbf{p}_{mass_j}\, m_j}{m_{tot}} = [CM_x\, CM_y\, CM_z]^T \tag{2.1}$$

where m_j is the jth lumped mass at location $\mathbf{p}_{mass_j}$ in the robot's frame and m_{tot} is the total

mass of robot. Since the influence of head panning and tilting on the robot's CM is very small in comparison to the effects that arise from the position of the arm, the head is assumed to be a point mass at the end of the arm. Here m_{tot} is assumed to consist of the three parts

$$m_{tot} = m_f + m_h + m_b \tag{2.2}$$

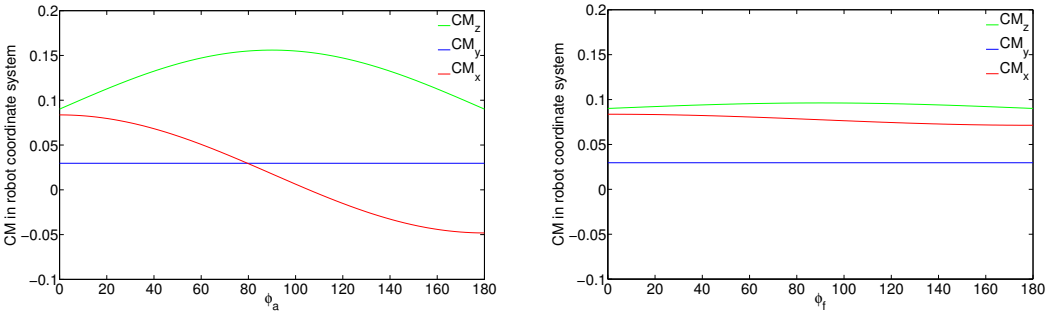
where m_f is mass of the flippers, m_h is the sensor head mass and m_b is the total mass of all other parts of the robot when their position is constant in the robot's coordinate frame. Figure 2.6 depicts a side view of the lumped-mass model. The Equation 2.1 can be reformulated for this robot model as

$$\mathbf{CM} = \frac{m_f\,\mathbf{p}_f + m_h\,\mathbf{p}_h + m_b\,\mathbf{p}_b}{m_{tot}} \tag{2.3}$$

where $\mathbf{p}_f$, $\mathbf{p}_h$ and $\mathbf{p}_b$ are the overall point location of m_f, m_h and m_f in the robot's frame respectively. As mentioned earlier $\mathbf{p}_b$ is constant but $\mathbf{p}_h$ and $\mathbf{p}_b$ will change based on the robot's arm angle ϕ_a and the angle of the robot's flippers ϕ_f. The relation between $\mathbf{p}_h$ and ϕ_a can be formulated as

$$\begin{aligned} p_{hx} &= p_{ax} + L_a\cos(\phi_a) \\ p_{hy} &= 0 \\ p_{hz} &= p_{az} + L_a\sin(\phi_a) \end{aligned} \tag{2.4}$$

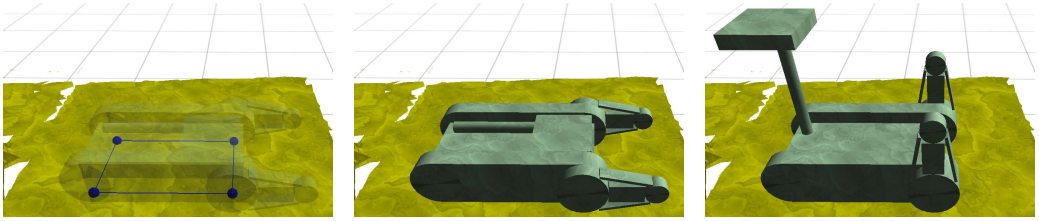
where L_a is the length of arm and $\mathbf{p}_a$ is the location of the joint between the arm and the robot body. As depicted in Figure 2.5, X_R is the robot's *roll* axis, Y_R the *pitch* axis and Z_R the direction normal to the platform, or yaw's axis. To avoid singularities in 3D spatial rotations, a parametrization with quaternion representation has been employed over Euler angles with an *wxyz* convention to transmit rotations between frames. Since the head moves in an $X_R - Z_R$ plane, the changes in ϕ_a have no effect on p_{hy}. The origin of the robot's frame is assumed to be located at the centre of the rectangular support polygon formed when the robot's body is parallel to the horizontal plane. The mass of the Packbot is $19.97kg$ and the robot's head mass is $2.557kg$, with the additional sensor head payload depicted in Figure 2.5 of an MS-Kinect camera, an infra-red camera, cabling and enclosure adding up to $1.314kg$. It is clear that for these type of robots, changing the angle of the arm will change the location of the CM.



(a) Effect of ϕ_a on the robot's CM (when $\phi_f = 0$).

(b) Effect of ϕ_f on the robot's CM (when $\phi_a = 0$).

Figure 2.7: Effects of the arm and flippers configurations (degrees) on the robot's CM.



(a) Shape of the ideal support polygon.

(b) Robot configuration with $(\phi_a = 0, \phi_f = 0)$.

(c) Robot configuration with $(\phi_a = 90, \phi_f = 90)$.

Figure 2.8: The standard robot configurations (units in degrees) and shape of the ideal support polygon when the contact points are assumed to be fixed under the main sprockets.

Figure 2.7 illustrates the effects of the arm and flippers configurations on the robot's CM. A comparison of Figures 2.7a and 2.7b shows that while CM_y is independent from the posture of the robot, the arm configuration has a larger effect on the CM_x and CM_z than the flippers' angle.

In the same way, the relation between $\mathbf{p}_f$ and ϕ_f can be formulated as

$$\begin{aligned} p_{fx} &= p_{fjlx} + L_f \cos(\phi_f) \\ p_{fy} &= 0 \\ p_{fz} &= p_{fjlz} + L_f \sin(\phi_f) \end{aligned} \tag{2.5}$$

where L_f is the length of the flippers's CM and $\mathbf{p}_{fjl}$ is the location of the left joint lump mass (between flipper and robot body). Note that because of symmetry in robot coordinate system, we could equally use the right joint for this calculation, i.e. $\mathbf{p}_{fjr}$.

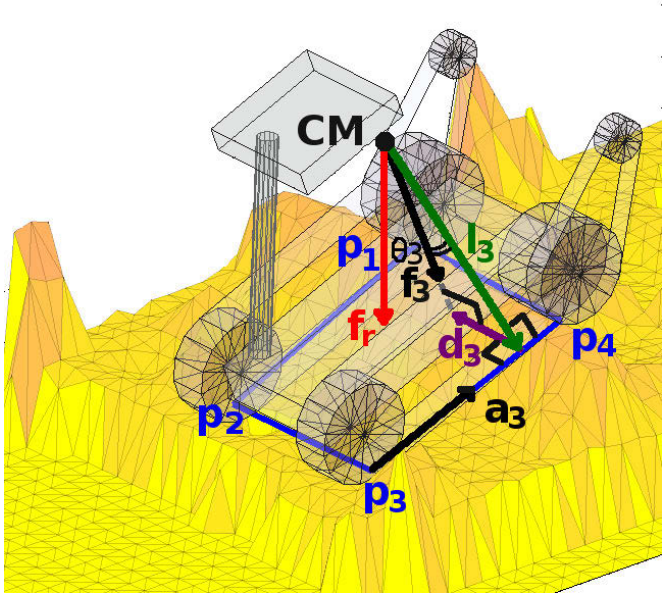


Figure 2.9: The 3D FA stability measure for $n = 4$ and $i = 3$ (for third axis of a support polygon with four contact points). (CM's position has been shifted up and vectors scaled for easier visualization). The FA measure can be intuitively described as the effect of the net force and moment over CM projected on the SP e.g. $\beta_3 = \theta_3 \ \|\mathbf{d_3}\| \ \|\mathbf{f_3}\|$.

When operating in uneven terrains these changes, in turn, have an effect on the stability measure β, as will be explained in the next section. When pursuing high vantage configurations for the sensor head so as to increase visibility of the environment, these stability changes need to be accounted for when suggesting paths the robot should follow.

Figure 2.8 shows the two standard robot configurations and shape of the ideal support polygon when the contact points are assumed to be fixed under the main sprockets, where the configuration of the highest CM position $(\phi_a = 90, \phi_f = 90)$ is depicted in Figure 2.8c. Although this arrangement provides the best field of view for the sensor head, it would increase the risk of a tip-over instability. The posture illustrated in Figure 2.8b with $(\phi_a = 0, \phi_f = 0)$ results in the best stable CM position i.e. CM would be at the lowest height and closest to the front of the robot base. For this exercise flippers have been considered at $(\phi_f = 0)$ instead of the angle which keeps them tangential to the flat terrain i.e. $(\phi_f = -12)$.

2.4 Force Angle Stability Margin

The FA stability margin [15] was principally proposed for mobile machines with manipulators operating in construction, mining, and forestry. Computation of the FA only requires knowledge of the system's CM position and the ground contact positions relative to the system CM to calculate the tip-over mode axes. This simple criterion can then be computed based principally on the minimum angle between the effective net force and the tip-over axis normal. The normalized FA measure will be between zero (borders of instability) to one (most stable configuration). Negative values of the FA measure for an axis indicate that occurring tip-over instability about that axis is in progress. As shown in Figure 2.9, the criterion β_i for the ith tip-over axis $\mathbf{a_i}$ can be principally described by

$$\beta_i = \theta_i \ \|\mathbf{d_i}\| \ \|\mathbf{f_i}\|, \quad i = \{1, \ldots, n\} \tag{2.6}$$

where n is the number of out-most contact points. $\mathbf{f_i}$ is the component of effective net force $\mathbf{f_r}$ which acts about the tip-over axis $\mathbf{a_i}$. θ_i is the angle between $\mathbf{f_i}$ and the tip-over axis normal $\mathbf{l_i}$. $\mathbf{d_i}$ is the minimum length vector from $\mathbf{a_i}$ to $\mathbf{f_i}$. For example in this work $\mathbf{a_1}$, $\mathbf{a_2}$, $\mathbf{a_3}$ and $\mathbf{a_4}$ are left, rear, right and front axis respectively as illustrated in Figure 2.9. The angles are in reference to the support pattern, which is the convex polygon derived from the ground contact points of the robot, and are sensitive to changes in CM's height. The overall robot's FA measure β, is given by

$$\beta = \min(\beta_i), \quad i = \{1, \ldots, n\} \tag{2.7}$$

In general, mobile vehicles operate at low speed when travelling over rough terrain and quasi-static robot dynamics can be safely assumed [7]. Thus, the net force $\mathbf{f_r}$ acting on the system's CM will come from the gravitational loading term

$$\mathbf{f_r} = \sum \mathbf{f}_{grav} = m_{tot}\, \mathbf{g} \tag{2.8}$$

where m_{tot} is the total system mass from 2.2 and $\mathbf{g}$ is the gravitational acceleration.

The location of the ground contact points of the vehicle relative to the system's CM is needed to compute β_i. Given all the vehicle's possible contact points of the tracks with the terrain, only the outermost points $\mathbf{p_i}$, which form a convex support polygon when projected onto a horizontal plane, need to be considered.

$$\mathbf{p_i} = [p_x\, p_y\, p_z]_i^T, \quad i = \{1, \ldots, n\} \tag{2.9}$$

These points $\mathbf{p_i}$ will thereon be referred to as the ground contact points, while the axes that join the ground contact points are the candidate tip-over axes, $\mathbf{a_i}$

$$\mathbf{a_i} = \begin{cases} \mathbf{p_{i+1}} - \mathbf{p_i} & i = \{1, ..., n-1\} \\ \mathbf{p_1} - \mathbf{p_i} & i = n \end{cases} \tag{2.10}$$

Assuming $\hat{\mathbf{x}}$ indicates the normalized vector for an arbitrary vector $\mathbf{x}$, i.e.

$$\hat{\mathbf{x}} = \frac{\mathbf{x}}{\|\mathbf{x}\|} \tag{2.11}$$

The tip-over axis normals $\mathbf{l_i}$ through the system's CM are calculated by subtracting from $(\mathbf{p_{i+1}} - \mathbf{CM}_w)$ the portion that lies along $\hat{\mathbf{a}}_\mathbf{i}$, i.e.,

$$\mathbf{l_i} = (\mathbf{I} - \hat{\mathbf{a}}_\mathbf{i}\, \hat{\mathbf{a}}_\mathbf{i}^T)(\mathbf{p_{i+1}} - \mathbf{CM}_w) \tag{2.12}$$

where $\mathbf{I}$ is 3×3 identity matrix and $\mathbf{CM}_w$ is the location (in world coordinate) of the system's CM. The criterion is shown to depend on the component of $\mathbf{f_r}$ which acts about the tip-over axis $\mathbf{a_i}$, so that $\mathbf{f_i}$ is defined as

$$\mathbf{f_i} = (\mathbf{I} - \hat{\mathbf{a}}_\mathbf{i}\, \hat{\mathbf{a}}_\mathbf{i}^T)\mathbf{f_r} + \frac{\hat{\mathbf{l}}_\mathbf{i} \times ((\hat{\mathbf{a}}_\mathbf{i}\, \hat{\mathbf{a}}_\mathbf{i}^\mathbf{T})\mathbf{n_r})}{\|\mathbf{l_i}\|} \tag{2.13}$$

where $\mathbf{n_r}$ is the net moment acting about the system CM. For the case of a quasi-static vehicle and supposing smooth posture changes, the $\mathbf{n_r}$ can be neglected to speed-up the calculations. The minimum length vector $\mathbf{d_i}$ from the tip-over axis $\mathbf{a_i}$ to $\mathbf{f_i}$, (shown in violet for $\mathbf{d_3}$ in Figure 2.9), is calculated as

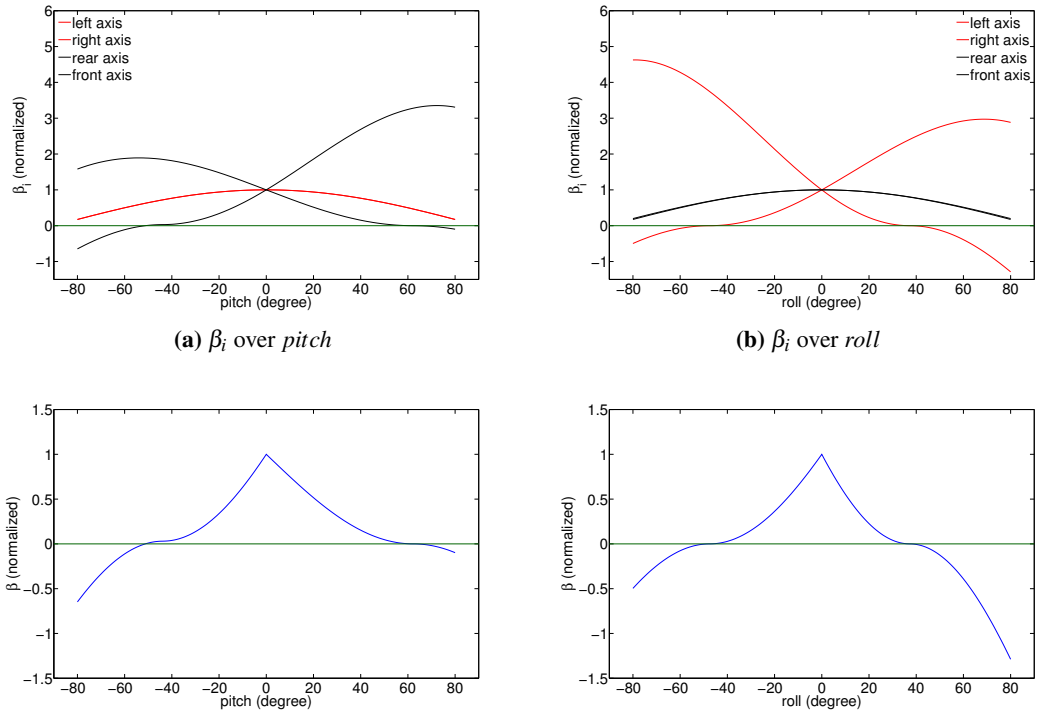
$$\mathbf{d_i} = -\mathbf{l_i} + (\mathbf{l_i} \cdot \hat{\mathbf{f}}_\mathbf{i})\hat{\mathbf{f}}_\mathbf{i} \tag{2.14}$$

so that the angle θ_i can be obtained by

$$\theta_i = \sigma_i \arccos(\hat{\mathbf{f}}_\mathbf{i} \cdot \hat{\mathbf{l}}_\mathbf{i}) \tag{2.15}$$

with

$$\sigma_\mathbf{i} = \begin{cases} +1 & (\hat{\mathbf{f}}_\mathbf{i} \times \hat{\mathbf{l}}_\mathbf{i}) \cdot \hat{\mathbf{a}}_\mathbf{i} > 0 \\ -1 & otherwise \end{cases} \tag{2.16}$$



(a) β_i over *pitch*

(b) β_i over *roll*

(c) Overal β over *pitch*

(d) Overal β over *roll*

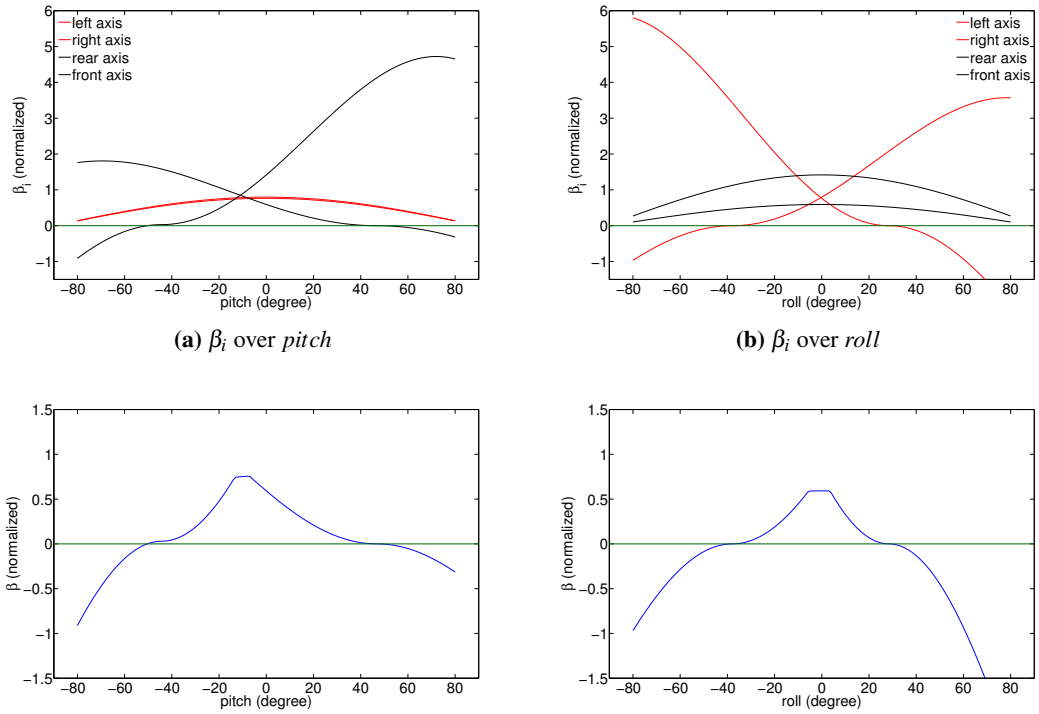
Figure 2.10: Effect of robot's inclination on the FA measure $(\phi_a = 0, \phi_f = 0)$. The horizontal lines indicate the reference points at which $\beta = 0$, or the line of instability.

For more details on these derivations, the reader is referred to [15]. In order to facilitate the interpretation of the stability measurements and be able to define a reasonable stability margin, a normalized value of tip-over instability about each axis is defined as

$$\hat{\beta}_i = \frac{\beta_i}{\beta nom_i} \tag{2.17}$$

where βnom_i is the nominal β for the given ith tip-over axis when the robot is assumed to be in its most stable configuration, i.e. on a level surface with the arm folded back and flippers in front of the robot $(\phi_a = 0, \phi_f = 0)$ as shown in Figure 2.8b. As in the rest of the work only the normalised value of β is used everywhere, for simplicity reasons the symbol of β (without hat) will be employed to point to the normalised value of the FA stability margin.

To better understand the effect of *pitch* and *roll* on the robot's tip-over axes, Figure 2.10 illustrates the evolution of β_i about the four axes when the robot is assumed sitting on flat terrain with the configuration shown in Figure 2.8b and the flat terrain inclination (*pitch* and



(a) β_i over *pitch*

(b) β_i over *roll*

(c) Overal β over *pitch*

(d) Overal β over *roll*

Figure 2.11: Effect of robot's inclination on the FA measure $(\phi_a = 90, \phi_f = 90)$. The horizontal dark green dash-dot lines are indicating the reference points where $\beta = 0$ or the line of instability.

roll) is changed. Due to the flippers being considered at $(\phi_f = 0)$ instead of the angle which keeps them tangential to the flat terrain i.e. $(\phi_f = -12)$, the contact points are assumed to be fixed under the main sprockets which would form an ideal support polygon, as depicted in Figure 2.8a. The FA stability margin about each four tip-over axes are normalized with their corresponding value for this configuration on flat terrain. The progress of the β_i with *pitch* and *roll* are depicted in Figure 2.10a and 2.10b respectively. The changes of the overall β as the minimum FA among these four axes with *pitch* and *roll* are illustrated in Figure 2.10c and 2.10d sequentially. It can be seen from Figure 2.10a how, for example, when the vehicle is tilted backwards (positive *pitch* angles), β_4 about the front tip-over axis gets increasingly larger, indicating that the stability about this axis is not compromised. On the other hand, tip-over instability about the rear axis starts to occur at around the 56° mark in *pitch*. Please note how in Figures 2.10 and 2.11 β are not symmetrical in *roll* since the head arm does not pivot around the centre of the robot base, but to one side.

Figure 2.11 summarises the outcome of these physical simulations. As depicted in Figure 2.11a and 2.11b, the FA measure experiences bigger changes with *pitch* and *roll* in comparison to the results shown in Figure 2.10a and 2.10b. The changes of the overall β as the minimum FA among these four axes with *pitch* and *roll* are illustrated in Figure 2.11c and 2.11d sequentially. As the FA stability margin about each four tip-over axes are normalized with their corresponding value for the configuration $(\phi_a = 0, \phi_f = 0)$ shown in Figure 2.8b on flat terrain, the overall FA values of Figure 2.11c and 2.11d would not even reach to one when *pitch* and *roll* are zero.

Since this model of Packbot has just two flippers at the front and it is not equipped with rear flippers, it is more stable coming down from a ramp or stairs than climbing up. According to Figure 2.10a and 2.11a and supposing the robot is seated in an incline with $pitch = 35^{\circ}$, the FA measure about the rear axis descends from 0.227 to 0.042 for the arrangements of $(\phi_a = 0, \phi_f = 0)$ and $(\phi_a = 90, \phi_f = 90)$ respectively. Remember that these values are calculated based on the assumption of an ideal support polygon, therefore the safety margins would be worse if the stability axes were closer to CM i.e. when the robot is over stairs instead of a ramp. The minimum stability margin is depend on the robot model, the environment and the type of the application. For the modified Packbot model used in this research, a minimum FA baseline margin of 0.05 is considered for comparison purposes in the path planning algorithms provided in the following chapters. This value has been empirically estimated through testing with the robot in the USAR arena.

2.5 Terrain Modelling

For vehicles with the capacity to alter their posture, predicting the contact areas and support polygons is a challenging task as it requires derivation of the contact forces of the robot on the terrain. This is highly influenced by the position of the CM, which changes often given the need to accommodate varying, often conflicting objectives. The search and rescue robot shown in Fig1.1, for instance, will favour arrangements that allow for a greater field of view for the sensors mounted on the arm head while scanning the environment for victims, while at the same time will naturally aim to adopt postures with lower CM heights to prevent robot tip-over instabilities. Therefore an effective algorithm able to prioritize between the need for maximum situational awareness (e.g arm up) and maximum stability (e.g arm down) appears advantageous.

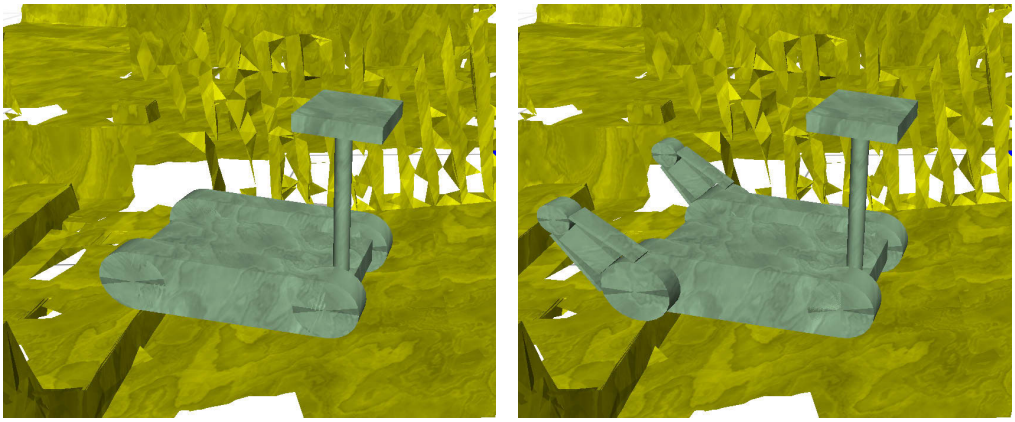
The algorithm to predict robot-terrain interactions requires as input a 3D model of the terrain where the robot is to navigate. While any of a variety of technologies commonly used for 3D perception could have been used, namely stereoscopic vision [40], or tilting LRFs [41], RGB-D cameras based on the PrimeSense sensor (MS-Kinect, Xtion) are naturally designed to provide point clouds from a small and lighter package, which make them more adept sensors for mobile indoor platforms such as search and rescue robots. In this work, the 3D point cloud obtained from a MS-Kinect camera [42, 43] and LRF were used to generate continuous 3D triangulated surfaces [19, 44, 45].

2.6 Robot-Terrain Interaction Analysis

To anticipate the stability measure, some works like [9] considered an ideal support polygon (ISP) for a tracked vehicle, i.e. the contact points are assumed to be fixed under the sprockets of the robot. Some experiments in Section 2.8.2 will illustrate how this is a strong assumption for the case of highly unstructured terrains, where contact points can lay anywhere along the robot's track and in general form a variable support polygon (VSP). In this work no ISP is assumed, and in this section the details of the process to derive the VSP of a robot on a terrain are presented.

The scheme is predicated on calculating the projection of the robot's geometric underside on the points defining the terrain underneath so as to derive the contact points. Determining the exact contact forces necessary to satisfy non-penetration constraints is NP-hard [46]. Given the complexities, the well known ODE [47] physics library for simulating rigid body dynamics has been used to approximate the interacting forces between the Packbot robot and the support surfaces, and derive the contact points. While straightforward geometry-based propositions can possibly be derived for simpler convex robot's surfaces, this is not necessarily the case for more complicated shapes. As depicted in Figure 2.12, the robot model is described for the simulator using ODE's standard geometric primitives (i.e. boxes, cylinders, ...) connected together via ODE's fixed joints and three variable joints for the arm and the flippers.

To better describe and validate the proposed algorithm, the flippers are first excluded from the robot model in the next section. Following this there will be a full stability analysis with flippers arrangement.



(a) The basic robot model is composed of nine components.

(b) The robot with flippers' model is composed of 19 components.

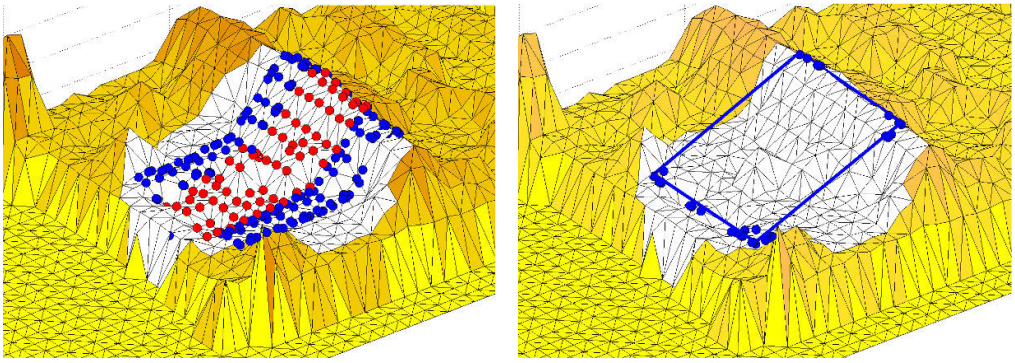
Figure 2.12: Description of Packbot for the simulator using ODE's standard geometric primitives.

2.6.1 Contact Points Prediction for Basic Robot Model

The basic robot model is composed of nine components which comprise the main body, four big main sprockets, two side tracks, one cylinder to represent the arm and a box to model the sensor head as shown in Figure 2.12a. As illustrated in Figure 2.12b, each flipper can be modelled using five primitives for the big and small sprockets, upper and lower tracks and another primitives to configure the middle part which links the big and small sprockets together. In summary, the number of geometric primitives for the basic (Figure 2.12a) and the complete robot model including flippers (Figure 2.12b) will be 9 and 19 respectively.

In the case of the basic robot model, the mathematical description of the robot's bottom surface considers the two main tracks and a middle step, rendering the polygon concave as depicted in Figure 2.13. Under the assumption of quasi-static dynamics, the influence of gravitational force 2.8 for a given robot's pose and configuration can be calculated in an iterative numerical process. To that end, the concave polygon describing the vehicle is first assumed to be sitting on a hypothetical plane with no $pitch$ or $roll$ at any given position and orientation in world coordinates.

The arm angle ϕ_a, flippers angle ϕ_f, coordinates of the centre point (x,y,z) and robot's orientation $(yaw, pitch, roll)$ - generally referred to as heading, elevation and banking - in the global reference frame fully describe the model. The 2D robot position (rx, ry, yaw) in the global reference frame and the angle of the arm and flippers joints (ϕ_a, ϕ_f) constitute the input



(a) Projected *candidate points* (**P**) on the terrain black+blue (tracks in blue).

(b) The *touched points* (**TP**) and the final support polygon, in this case resulting in 4 stability axes.

Figure 2.13: Contact point prediction.

variables. Then z of the hypothetical plane will be set to maximum z of the point which is in the terrain underneath. Supposing the initial values of $pitch = 0$, $roll = 0$, ODE simulates the behaviour of the robot if it is dropped from this point and determines its primary position.

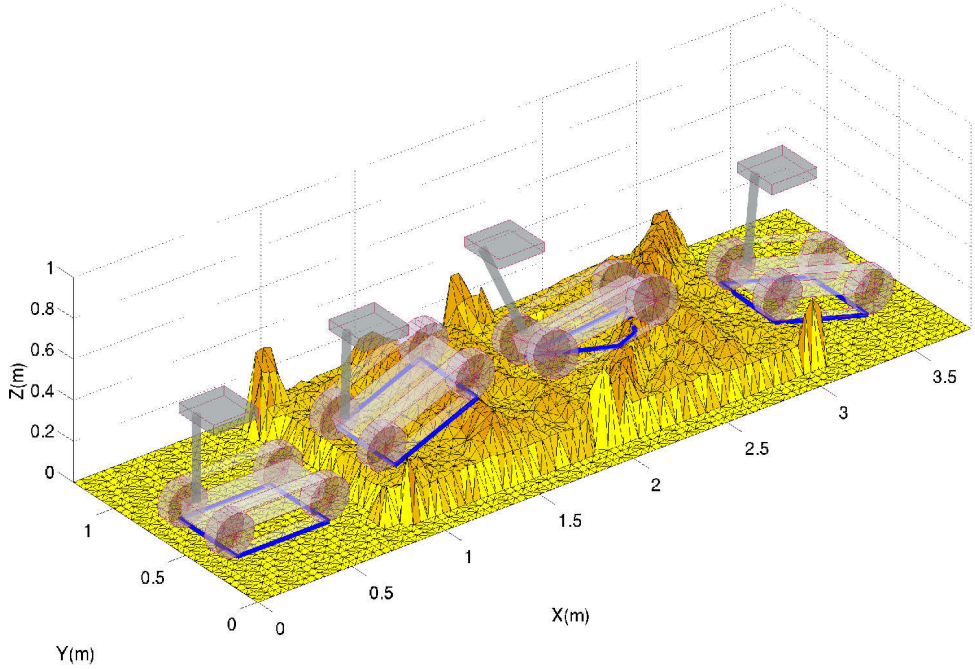
The projection of the main tracks and the middle body on the gridded terrain results in a set of *candidate points* (**P**) which are shown in Blue and Red in Figure 2.13a respectively. The point set located within a threshold distance set by the uncertainty in the grid modelling and rubbery track properties will be considered as *touched points* (**TP**) as depicted in Figure 2.13b. The outermost points form **TP** will become the ground contact points needed to represent the support polygon of the vehicle when sitting on the terrain at that location. The tip-over axes $\mathbf{a_i}$ are defined as the segments that join the ground contact points.

The example shown in Figure 2.13 corresponds to the pose shown in Figure 2.9 (also the second pose in the sequence depicted in Figure 2.14a). A maximum of four possible contact points are assigned to form the vertices of the support polygon, while at least three contact points are needed for the pose to be regarded stable. Therefore the robot will be regarded stable at a given location if the resulting support polygon fulfils the following criteria:

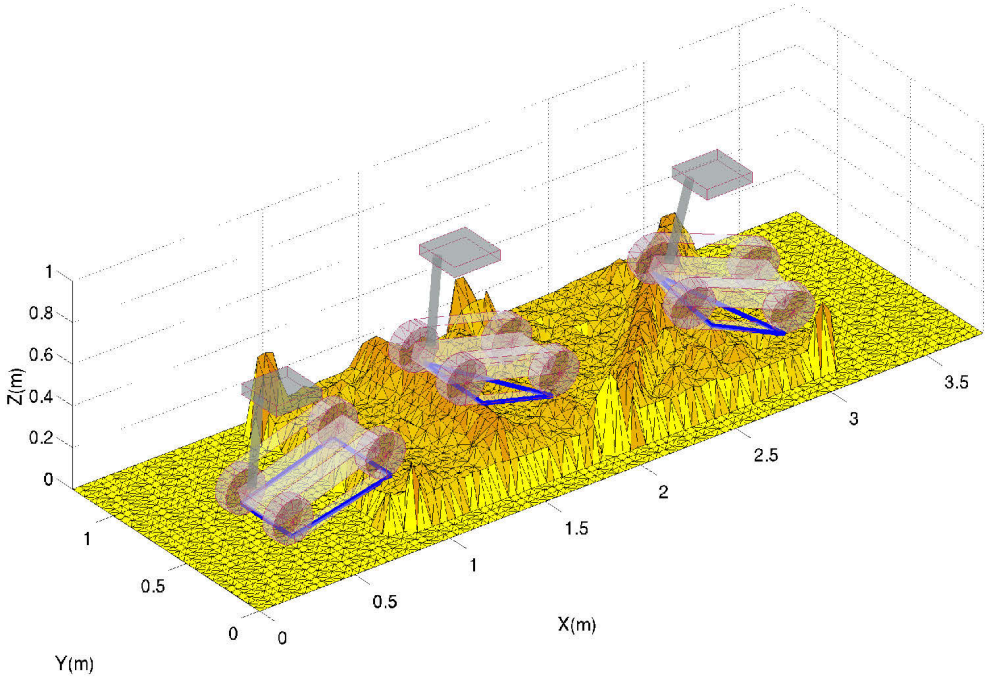
(a) $n \geq 3$

(b) $\beta > \beta_{min}$

where n and β are defined in 2.6 and 2.7 respectively and β_{min} is the minimum stability margin. The scheme is able to model scenarios where robot slippage may be present at a

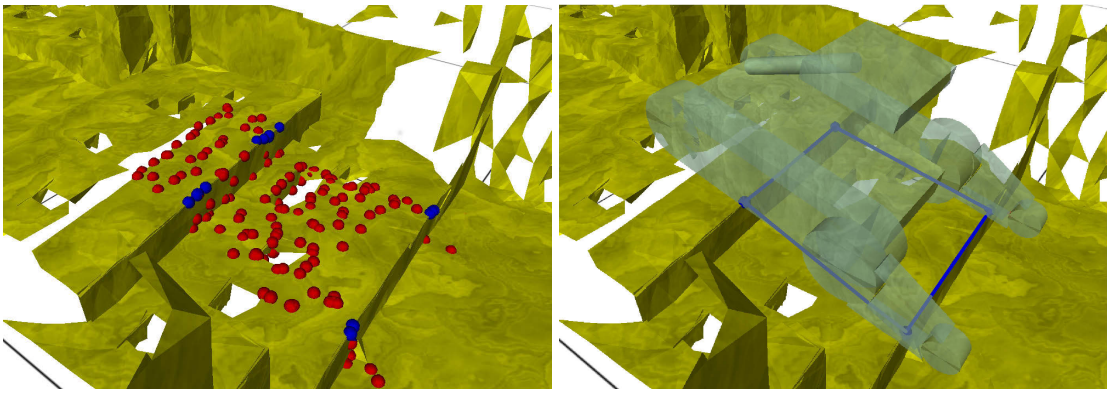


(a) Four robot poses



(b) Three robot poses

Figure 2.14: The shapes of support polygons over two step-fields.



(a) *candidate points* (**P**) in red+blue and *touched points* (**TP**) in Blue.

(b) The final support polygon, in this case resulting in 4 stability axes.

Figure 2.15: Projection of the robot's bottom surfaces on the grid terrain to predict contact points.

given point in the terrain. Large variations of the final (x, y, yaw) from the initial values fed by the planner in any direction will reject the point as slip-prone, hence unstable for planning. This is of course only a crude approximation of slippage. Further knowledge about terrain properties are needed to make more assertive predictions about slippage. When these variations are small, the final resting location of the robot will become the new candidate way-point for planning a path through. To better illustrate the effect of the stability criteria and high vantage configurations in the robot pose, an experiment is presented where the robot is left to assume the most stable configuration while keeping the arm as close as possible to 90 degrees in the global horizontal plane. This was repeated at a number of locations along the step-fields. NIST step fields [48] are used as a recognized artificial analogue for real rubble. They consist of approximately $1.5m^2$ terrains formed by blocks of $64cm^2$ and up to $40cm$ in height. These can be combined to generate standard patterns, a generally accepted practice for replicable tests in 3D navigation. The detailed results are shown in Figure 2.14, where the resulting stability axes in the terrain are also depicted in blue.

2.6.2 Contact Points Prediction with Flippers

The contact points prediction algorithm for the robot configuration with flippers is fairly similar to the procedure described in Section 2.6 for the basic robot model. As illustrated in Figure 2.15, the only difference here is that after determination of the robot's primary position by ODE, in addition to projection of main tracks and the middle body, the tracks of the flippers

are also considered to calculate **P** which are shown in red+blue in Figure 2.15a. Therefore **P** will be including the gridded points under five rectangular areas i.e. two main tracks, one middle body and two smaller tracks of flippers.

It should be mentioned that for a given robot configuration ODE will take care about the rigid body interaction between terrain, robot and flippers. The ODE is able to simulate the robot's final position with a given flippers configuration. The rest of the procedure will remain the same. The shape of the support polygon for FA measure does not necessarily need to be a 2D polygon spanned in a planar surface. The contact points can take place anywhere in the 3D space. Therefore the contact points can be solely from the main track (flippers are not in contact with the terrain) or can be a combination of the points from main tracks and flippers. Again four out-most points from **TP** will form the support polygon as depicted in Figure 2.15b. The conditions for a stable posture are identical to the basic model. The Figure 2.16 shows the result of the contact point prediction algorithm and the shape of the final support polygon with flippers in eight different positions.

2.7 Design and Implementation

2.7.1 Hardware Overview

As described in Section 2.3, the basic mechanical structure of the mobile robot employed to verify the results of this research consists of an iRobot Packbot, shown in Figure 2.17. The robot accepts basic commands for controlling the main tracks, steering, pan/tilt of the sensor head, lights and flippers. The on board processing unit of the robot also provides the data from the internal sensors such as absolute shaft encoder of the flippers, the state of the batteries and the main tracks's encoders data which are used to estimate the robot's position via dead reckoning. The external control unit can communicate with the robot through wired or wireless interfaces.

The robot is configured for the search and rescue mission, although the utilization can be easily adapted to other services like surveillance/reconnaissance explosive, ordnance disposal, vehicle and personnel inspections. The main task of the search and rescue robot is to search for victims within a collapsed building or unstructured environment while localizing itself and mapping the environment concurrently. Moreover an efficient rescue robot design should provide a user friendly human robot interaction interface to supply the rescue team with as much information about the status of the robot, the surrounding area and the potential victims around.

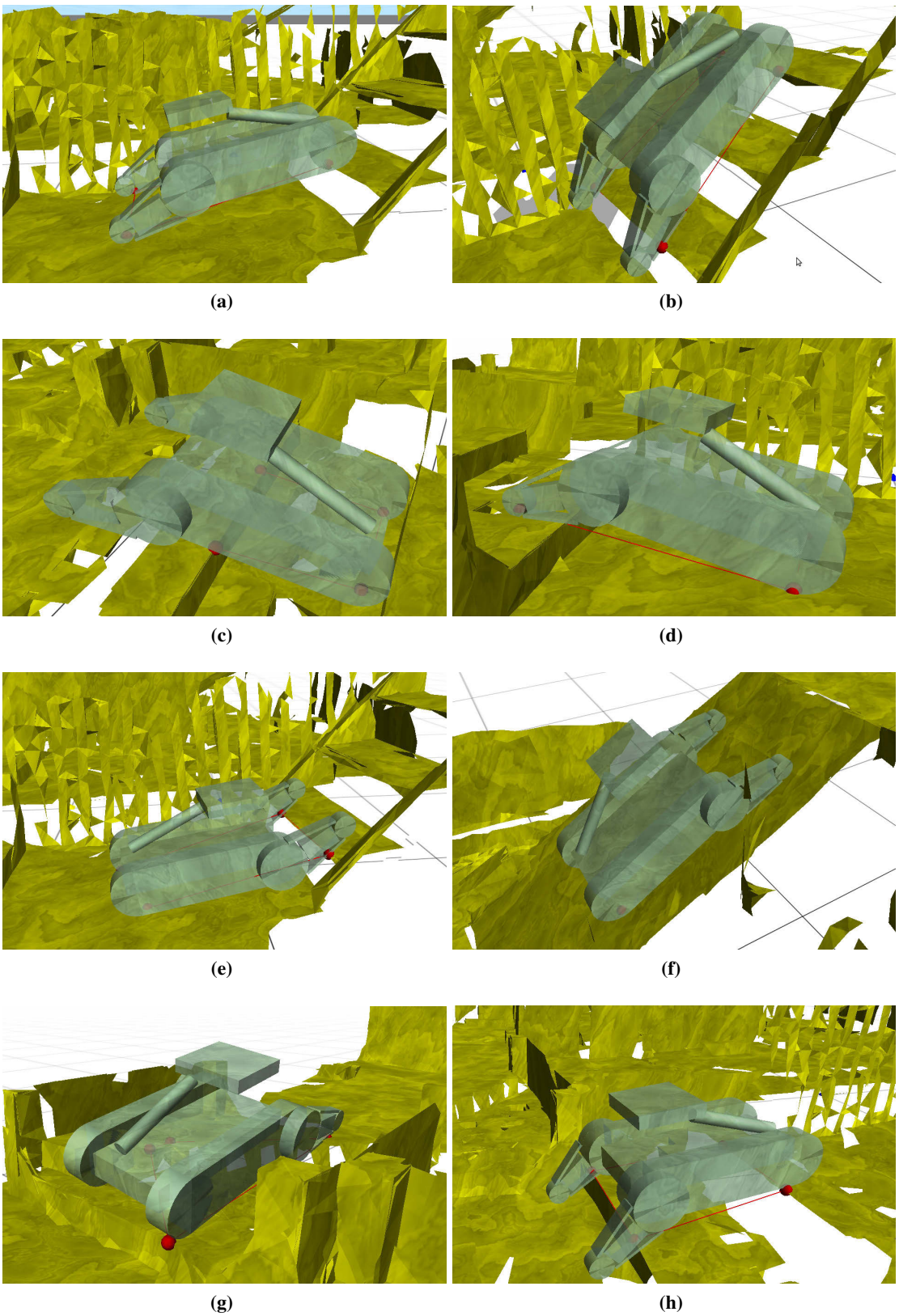


Figure 2.16: The shape of the support polygon with flippers in different positions.

Figure 2.17: The basic model of Packbot robot without additional sensors and processing units [39].

No mechanical modifications were made to the Packbot other than the addition of two boxes for the supplementary accessories. In addition to the standard mobile platforms, the robot is equipped with a variety of advanced sensors for mapping and victim identification, as well as highly effective fit-PC2 (http://www.fit-pc.com) to enhance the robot's processing and communication power. The fit-PC2 is a small, light, fan-less computer based on Atom CPU. The on board fit-PC2 of the Packbot is equipped with an Intel Atom 1.6 GHz CPU, 2GB RAM, WiFi, Dual Gbit Ethernet and a couple of USB2 ports to connect the external sensors and modules. As depicted in Figure 2.5, a typical configuration of the Packbot as a search and rescue robot will include the following sensors:

- **Hokuyo UTM-30LX laser scanning range finder:**
  The okuyo UTM-30LX (http://www.hokuyo-aut.jp) is a lightweight (370g) LRF providing a range of data through a 270° scan in 0.25° increments at 40Hz, up to a distance of 30m. Its small size allows it to be mounted on an auto leveller servo system. This sensor is used for position tracking of the robot and for the SLAM algorithm. Because of the auto leveller we can only calculate two dimensional information from this sensor.

- **Xsens MTi inertial measurement unit:**
  The robot use Xsens MTi (http://www.xsens.com) heading/attitude sensors. The MTi contains gyroscopes, accelerometers and magnetometers in 3D. These sensors provide 3D orientation to a high degree of accuracy, assisting in automatic map generation and situational awareness. These sensors are also used to auto level the laser sensors.

- **Thermoteknix Miricle IR thermal camera:**
  This camera from Thermoteknix (http://www.thermoteknix.com) has a high resolution 384x288 sensor in the 7-14 micron range. This camera forms the core of the autonomous victim identification system on the Packbot.

- **TPA81 thermopile array:**
  The TPA81 (http://www.robot-electronics.co.uk) is a thermopile array detecting infrared in the 2um-22um range. This is the wavelength of radiant heat. These pyro-electric sensors, however can only detect a change in heat levels, hence they are movement detectors. Although useful in robotics, their applications are limited as they are unable to detect and measure the temperature of a static heat source.

- **Kinect Sensor:**
  In addition to an RGB camera, the Kinect (http://www.xbox.com) features a depth sensor which provides full 3D motion capture. The Kinect sensor is widely used in the robotics community for the live 3D rendering, basic 3D visual SLAM, human gesture prediction as well as 3D obstacle detection. The sensor is low weight and the resolution of the RGB and IR depth-finding camera is 640x480 pixels @ 30Hz which makes it suitable for 3D terrain model reconstruction.

2.7.2 Software Overview

To facilitate a modular implementation, data logging, and ease of software integration, the algorithms proposed in this work are implemented using the framework of the open source Robot Operating System (ROS) [3] under Linux Ubuntu. One of the desirable features of ROS (http://ros.org) is that the system can connect a number of processes even from different hosts at runtime in a peer-to-peer topology as depicted in Figure 2.18. For example, in the search and rescue set-up, the driver station laptop is bridged via wireless LAN to the on-board fit-PC2 while the fit-PC2 itself is connected to the robot and another potential fit-PC2i via Ethernet as well. Although ROS supports very different languages including Python, Octave and LISP, the majority of the algorithms implemented in this project are coded in C++.

The key components of the framework are:

- **Libraries, Drivers, and recording data:**
  Nodes, messages, topics, and services form the fundamental concepts of the ROS implementation. Nodes are self contained modules that run independently and communicate

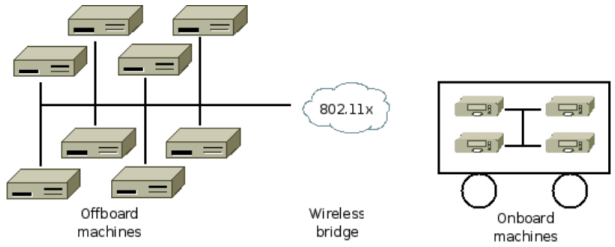


Figure 2.18: A typical ROS network configuration [3].

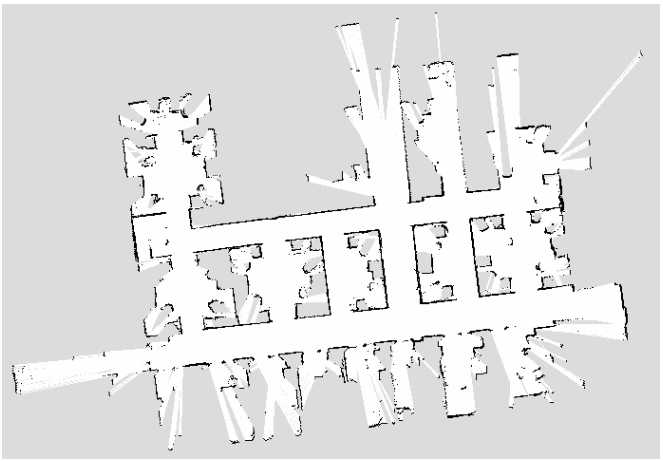


Figure 2.19: The floor map of our lab at UTS using the slam-gmapping node. The main corridor measures just over 20m.

with each other over so called topics using a one-to-many subscriber model and the TCP/IP protocol. Software libraries and ROS nodes are organized into packages, stacks, and ultimately Apps. Moreover the libraries and tools for software development, ROS provides visualizers, hardware abstraction and standard drivers for common sensors and modules in robotics applications. The driver for the other special equipment for this project such as the drivers for the Packbot and the explorer head are developed with our team under ROS standards. The rosbag package is used for recording from and playing back to ROS topics. It is possible to use the rosbag package using either command-line tools or a launch file including calls to other packages.

- **Localization and mapping:**

  The OpenSlam's Gmapping package [49] (http://wiki.ros.org/slam_gmapping) is employed for simultaneous localization and mapping. The odometry data provided by the

Packbot's ROS driver and the auto-levelled laser range-finder scans constitute the input informations for the slam-gmapping node. The slam-gmapping node will attempt to create a 2D occupancy grid map from laser and pose data collected by the mobile robot. A 2D occupancy grid map represents the environment as an array of cells holding a probability value for the presence of an obstacle at that location in the environment. An example of a map from our centre built by the slam-gmapping node is depicted in Figure 2.19.

- **Navigation and control:**
  The ROS Navigation stack (http://wiki.ros.org/navigation) is used to send the velocity commands to move and control the robot base as a differential driving tracked vehicle. While some planners based on Dijkstra's algorithm (e.g. navfn package [50]) are available in the Navigation stack, we have developed a new package called *stable planner* which supports other planners such as A* and RRT, and can take into account the other desired expenditures like energy and reconfiguration cost under stability and limitation in mechanical movement limitations constraints.

- **Physical simulation:**
  The physical simulations are performed using a novel package called "fa_ode". The fa_ode is employing the standard ROS opende package (a ROS wrapper for ODE) for collision detection. The proposed node provides "fa_ode_srv" service, which is defined by a pair of Request/Reply messages: The client node should provide the robot's configuration and pose in the request message and awaits for the node to perform the simulation and respond contact points as well as the FA measures as the reply message.

A block diagram of the overall software components, including the contact points prediction and FA stability analyser, is illustrated in Figure 2.20. The process starts with a request from a client node (usually the planner) for the stability analysis of the robot given a certain configuration of interest in a given 3D terrain model, assumed to be available as an input to the system. In a practical deployment scenario the system would need to generate 3D maps on the fly, however 3D mapping is out of scope of this research and off-the-shelf point-cloud based mapping modules have been employed for that purpose in this work. A pre-built mechanical robot model description is used by the ODE physics simulator engine for its stability calculations, where 2D positions of the robot in the terrain are estimated by a 2D SLAM node with the

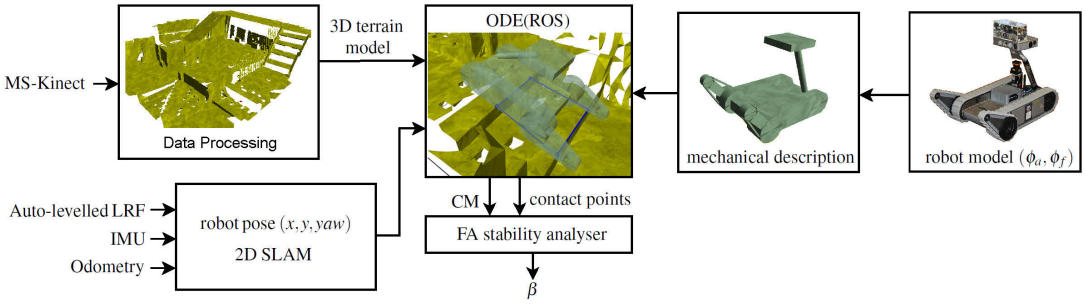


Figure 2.20: The block diagram of the contact points prediction and FA stability analyser.

aid of the auto-levelled laser scanner data, IMU and robot odometry. The ODE node analyses the expected behaviour of the robot and generates a list of contact points as well as the location of robot's CM in the global reference frame. This information constitutes the input variables used by the FA stability analyser node to calculate the overall β.

2.8 Experimental Results

The performance and importance of the proposed contact point prediction technique has been evaluated via two trials. Firstly, an experimental validation in a reconfigurable mock-up $6m \times 8m$ USAR arena in the next section will demonstrate the approach and the close correlation between measured and simulated results. Then, an experiment in the following section will be described to provide evidence for the need for a stability analysis based on VSP versus the more simplistic ISP over two common indoor obstacles i.e. ramps and stairs.

2.8.1 Inclination Prediction over Step-fields

The proposed prediction scheme represents a numerical simulation method, and therefore a trade-off between accuracy and numerical stability. An experiment in a reconfigurable $6m \times 8m$ USAR arena was carried out to assess the validity of the approximate solution in a practical setting. Figure 2.21 shows a snapshot of the robot in the arena approximately half-way through first step-field.

Getting accurate feedback from the actual contact points under the tracks require instrumenting either the whole arena, a highly impractical exercise, or the platform with specialized equipment like pressure sensor arrays, which cannot be easily adapted to the track locomotion arrangement of the robot. Instead, the robot was made to assume a fixed configuration of the

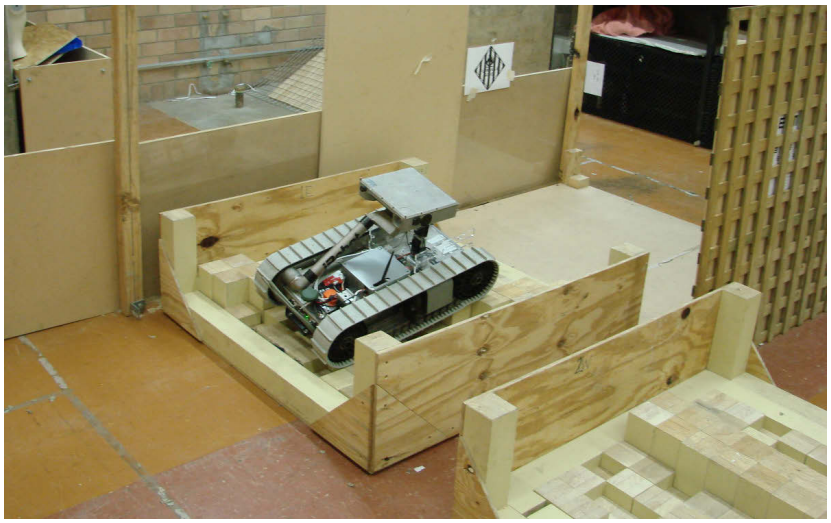
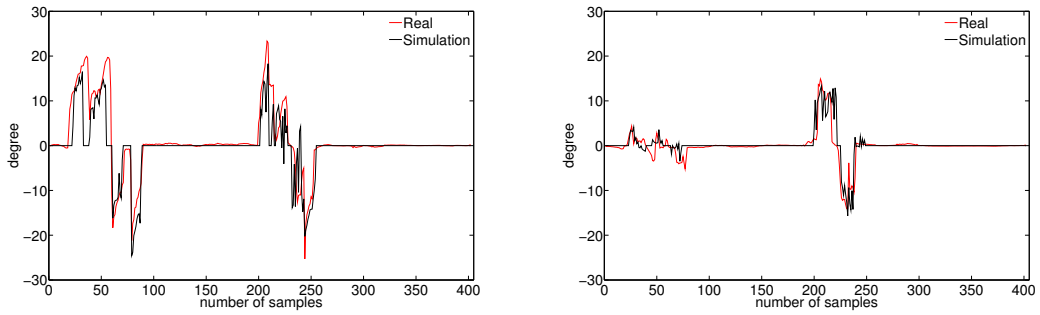


Figure 2.21: The rescue robot on a fully autonomous mission.

arm and programmed to slowly track a given path so that fast dynamics and slippage could be neglected, focusing the stability analysis on the gravitational and reaction forces considered for this work.

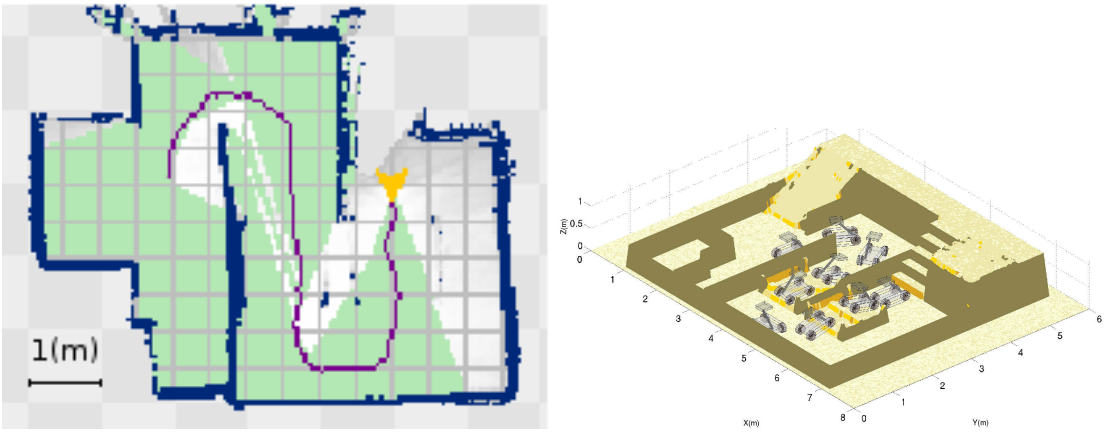
The robot was made to traverse a particularly challenging path including two step fields. Two step fields were arranged in diagonal and hill configurations. A localizer running of 2D range data from the auto-levelled LRF was used to derive an estimate of the robot pose (x, y, yaw) with a previously built map of the arena.



(a) *pitch*. **(b)** *roll*.

Figure 2.23: *pitch* and *roll* prediction results.

Data was then recorded at 402 locations along the path traversed by the robot, depicted



(a) 2D view of the actual path traversed by the robot on a previously built map of the USAR arena.

(b) 3D visualization of the traversed path.

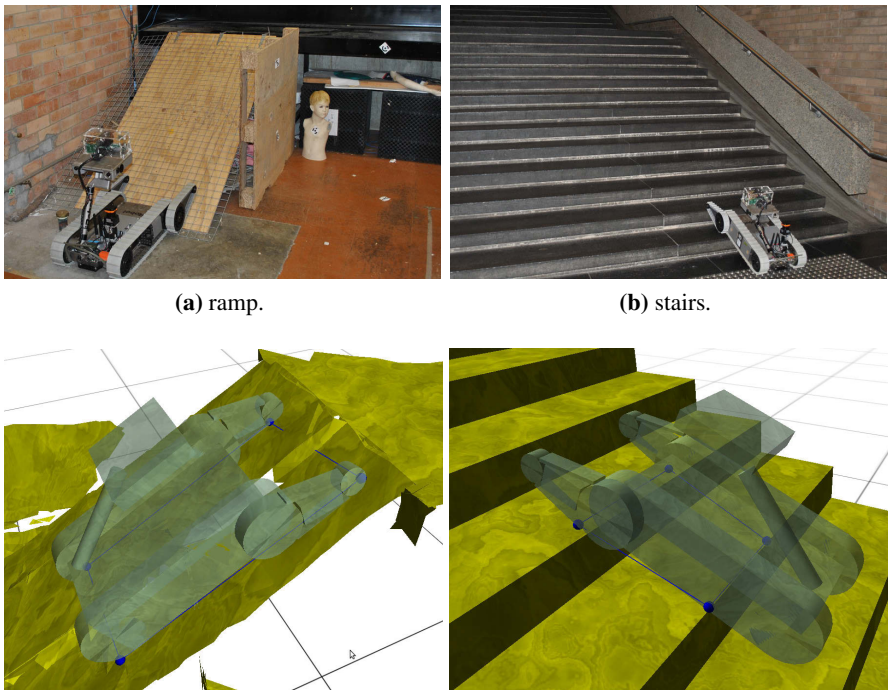
Figure 2.22: Path traversed by the robot during the experiments.

in Figure 2.22a. The measured 2D poses were then used off-line for the verification of the proposed contact point prediction algorithm. As the platform has no suspension and the terrain is rigid, *pitch* and *roll* measurements from an on-board MTi-Xsens inertial measurement units (IMU) can be assumed to be a veracious reflection of the vehicle's attitude when sitting on the terrain.

A comparison between the measured and predicted vehicle *pitch* and *roll* derived from the contact point calculation at these locations was carried out to indirectly assess the proposed algorithm for calculating contact surfaces and stability measures. The results presented in the Figure 2.23 clearly indicate a close correlation between the real values and those inferred from the derivation of the contact point surfaces, with *pitch* and *roll*'s RMS prediction errors over two step-fields of 5.63 and 2.57 degrees respectively.

2.8.2 Significance of ISP and VSP on stability prediction

The relevance of the proposed algorithm to compute the robot's variable support polygon (VSP) over the more simplistic ideal support polygon (ISP) defined by the robot sprockets is verified in this section with experimental results in two representative environment settings. The robot was commanded to move slowly at a constant speed over two different impediments, a ramp and a flight of stairs as shown in Figure 2.24a and 2.24b respectively, both with similar inclination.



(a) ramp. **(b)** stairs.

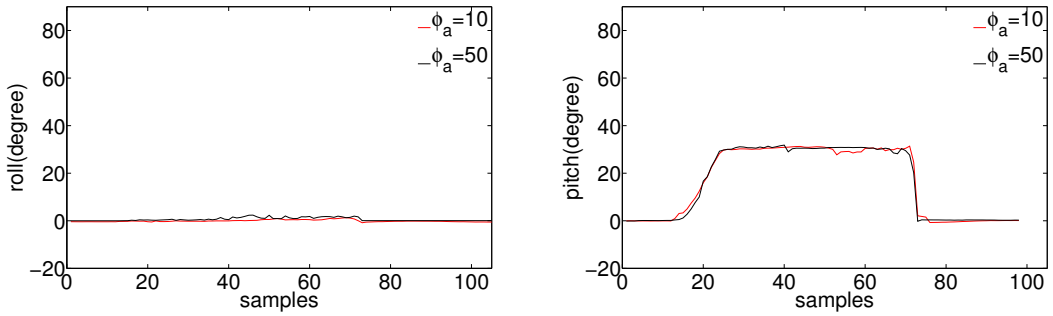
(c) SP over the ramp. **(d)** SP over the stairs.

Figure 2.24: Ramp and stairs and the shape of the support polygons (SP).

The arm configuration was fixed ($\phi_a = 10^\circ$ and $\phi_a = 50^\circ$) during each test. To allow for maximum traction, flippers were repositioned to remain tangential to the terrain ($\phi_f nom$).

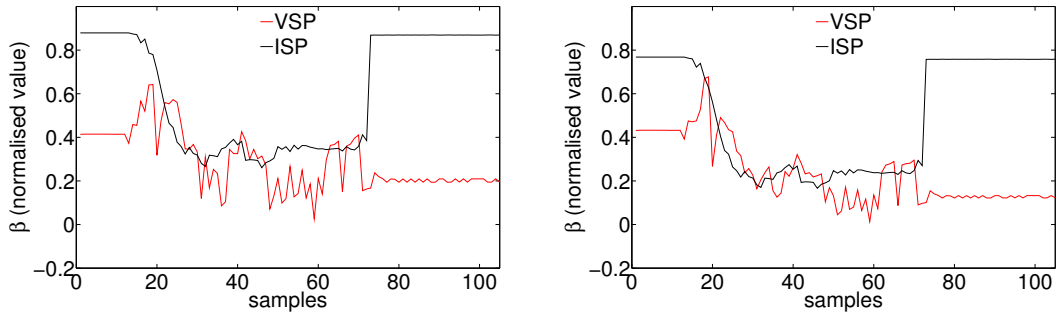
Pitch increased from zero (robot on flat terrain) to approximately 30° and 25° over the ramp and stairs respectively, but there was no significant *roll* involvement as illustrated in Figure 2.25 and 2.27 consequently. IMU data, localization and the robot's posture was recorded during the tests. FA stability margins were then calculated off-line using real IMU data and robot posture from the experiment while assuming an ISP, and also using the robot localization information, robot posture and 3D environment data from the experiment, and the VSP attained from the 3D ODE simulations of the robot in the environment, as described in Section 2.6.

The results of calculating FA margins for an ideal and a variable SP and a fixed arm configuration of $\phi_a = 10^\circ$ are shown in Figure 2.26a and 2.28a. It can be seen how computing stability solely based on inclination data and an assumption of ISP, as in [16], will result in relatively similar safe tip-over margins for both topologies (shown in black in the figures). However, as illustrated in Figure 2.24c and 2.24d, the shape of the SP at these two places is quite different despite similar average *pitch* angles. *Pitch* oscillates in the transition between two steps i.e. at



(a) *roll*. **(b)** *pitch*.

Figure 2.25: The inclination measures in two trials over the ramp.

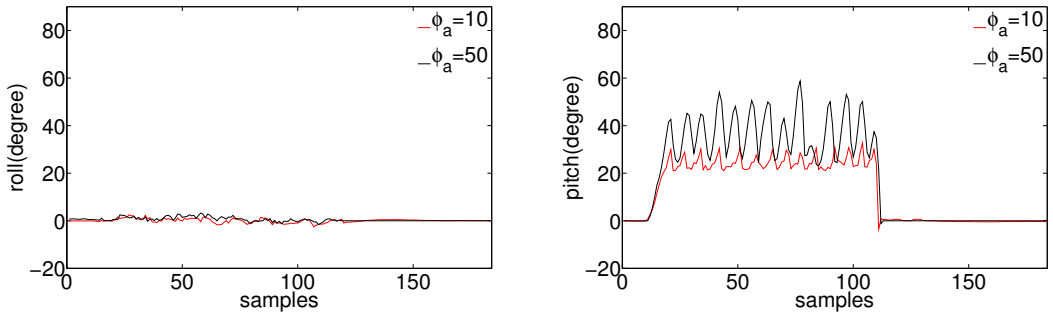


(a) β ($\phi_a = 10$). **(b)** β ($\phi_a = 50$).

Figure 2.26: The FA stability measures based on VSP and ISP in two trials over the ramp.

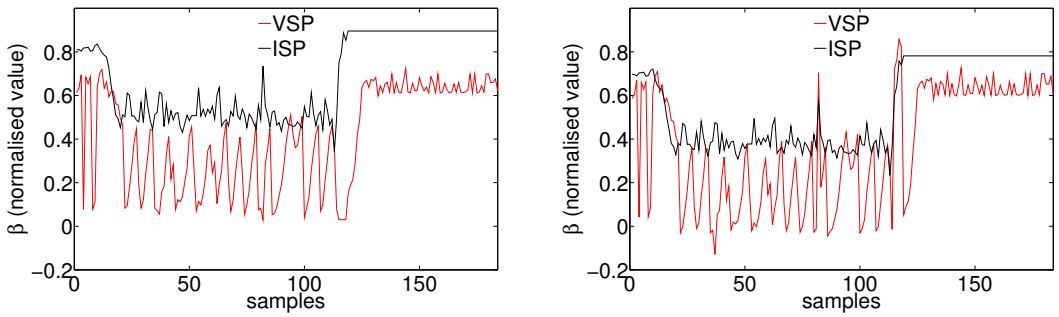
the instances when the robot was loosing contact with the rear step until the time new contact was made with the next step. The SP is therefore smaller on the stairs when compared to that on the ramp. As such, when considering stability margins, the effect of a VSP on a ramp is confidently positive, but it borders on instability on the stairs.

This is further observed in the second test where the robot was driven over both topologies with a different arm configuration, ($\phi_a = 50^\circ$), resulting in a higher CM position. The results indicate how, despite lower stability margins, the robot was able to come up the ramp safely, yet it proved to be unstable over the stairs and was about to tip-over around the transitions between steps - and it had to be manually handled to return it back to the terrain to prevent a fatal crash. Figure 2.26b and 2.28b depict FA margins for the ramp and stairs respectively. VSP reflects how FA measures for the ramp are always positive while at times reach negative values over the stairs. On the other hand, stability measures based on ISP always remain positive for both



(a) *roll*. **(b)** *pitch*.

Figure 2.27: The inclination measures in two trials over the stairs.



(a) β ($\phi_a = 10$). **(b)** β ($\phi_a = 50$).

Figure 2.28: The FA stability measures based on VSP and ISP in two trials over the stairs.

topologies and are not able to accurately predict instability over the stairs. *Pitch* angle evolution over the stair test can be seen in Figure 2.27b, clearly showing sudden crests where tip-over was starting to occur. This example clearly shows the need for a reliable stability predictor for planning purpose based on the available 3D terrain models, which could be formed using on-board low-cost point-cloud cameras like the MS-Kinect camera.

2.9 Summary

A methodology to predict the stability of a vehicle on rough terrains is presented in this chapter. The methodology is based on a mathematical description of the robot's mechanical structure in the ODE dynamic physics simulator. This program is able to import 3D model of terrain and simulate the behaviour of a rigid body over the modelled environment. The simulator aims to derive and interpret intermittent robot-terrain contact points required to calculate the FA stability margin to improve the safety of robot mobility over ruggedised terrains. The methodology is particularly applicable, although not restricted to, reconfigurable tracked platforms that can actively assume safer poses to reduce potential instabilities, such as those leading to vehicle tip-over when operating in uneven terrains. Close correlations between simulation experiments of a multi-tracked robot fitted with a sensor arm and flippers have been observed when contrasted with real data, illustrating the validity of the proposed scheme as a mechanism to increase the robustness of practical open-loop planning of stable paths.

The need for stability computations based on VSP when planning for safer navigational routes have also been established through experiments over two common indoor obstacles i.e. ramps and stairs. The benefit of defining this stability margin, using based on a variable support polygon for safer navigation on 3D meshed map models reconstructed from real point clouds, is shown in the next chapter.

3

Stable Path Planning

3.1 Introduction

The early stage result of this work is published in [51], where a study on the influence of a variable CM and the shape of the support polygon to plan safe traversable paths for the case of reconfigurable robots is presented. The Packbot tracked robot in a basic configuration (without flippers) was fitted with a manipulator arm and a range camera to verify the result of the work. The fundamental motivation of that work considered a suitable arm angle that could afford the widest sensor view and keep stability within safe margins. The main drawback of the work was the partially heuristic nature of the proposed algorithm in choosing the desired postures of the search graph.

This chapter proposes a motion planning strategy for reconfigurable mobile robots operating in uneven terrain. An analytical strategy to generate stable paths for reconfigurable mobile robots such as those equipped with manipulator arms and/or flippers, operating in an uneven environment whilst also meeting additional navigational objectives is hereby proposed. The suggested solution looks at minimizing the length of the traversed path and the energy expenditure in changing postures, and also accounts for additional constraints in terms of sensor visibility and traction. This is particularly applicable to operations such as search and rescue where observing the environment for locating victims is the major objective, although this technique can be generalized to incorporate other potentially conflicting objectives (e.g. maximizing ground clearance for a legged robot).

The proposed planning strategy looks at exploiting the (possibly incomplete) environment information available to the robot and/or operator as it explores novel terrain. The validity of

Figure 3.1: The rescue robot on the mock-up urban search and rescue test arena.

the proposed planning approach is evaluated with a multi-tracked robot fitted with flippers and a range camera at the end of a manipulator arm whilst navigating over two challenging 3D terrains data sets: one in a mock-up Urban Search and Rescue arena, and a second one from a publicly available quasi-outdoor rover testing UTIAS facility.

Autonomous mobile robots are required to find safe and feasible routes in the environment. This need is particularly relevant when operating over challenging terrains, where failure to avoid potential tip-over scenarios in dangerous regions can have catastrophic consequences. In fact these issues are not restricted to the autonomy aspects of mobile robot navigation, but also concern tele-operated robots or those controlled by on-board human operators carrying heavy payloads.

Despite recent advances [7, 28, 40, 41, 51, 52, 53, 54] autonomously navigating uneven terrain remains a significant challenge. Two major factors influence the ability of a vehicle to traverse a challenging terrain: stability and traction. Stability is one of the critical elements that bears a significant influence on the ability of a vehicle to traverse a given path. Assuming knowledge of the geometry of the terrain and the robot's inertial characteristics, a number of well studied stability measures can be employed to determine whether a vehicle will rest at a given location in the environment without tipping over. On the other hand, either the knowledge of the mechanical properties describing the vehicle-terrain interaction and/or a suitable traction control strategy is essential in guaranteeing that the vehicle is able to pass through a point in a given direction.

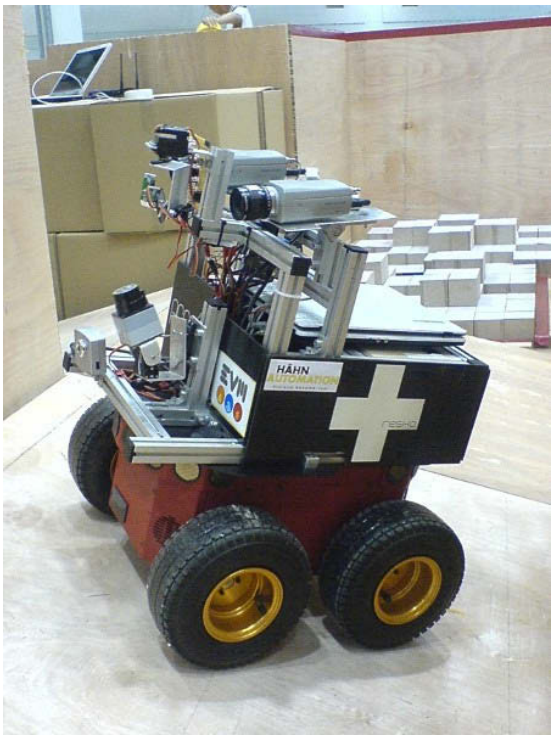


Figure 3.2: The Robbie rescue robot [4].

The robot's mass distribution and the shape of the support polygon have the most critical role in quasi-static stability evaluation for articulated vehicles. For a reconfigurable robot, such as the one seen on the USAR arena in Figure 3.1, the stability is a function of the robot configuration as well the terrain geometry. Lifting the arm attached to the robot or using the flippers can change the contact geometry as well as the location of the CM of the robot. At the same time, observing/mapping the surrounding environment is usually one of the key objectives of robots in many applications, including search and rescue, and for that it is often desirable to position the sensor head as high as possible. This clearly leads to a higher CM and a potential reduction in stability. In this chapter, a strategy to generate stable paths for a reconfigurable vehicle in the presence of conflicting objectives of visibility, traction and stability is presented. It is assumed that the geometry of the terrain is known (although not necessarily complete). We argue that this is a reasonable assumption in a practical scenario where the local geometry of the terrain has been captured by a sensor on-board the robot and the goal location for the robot is defined such that it lies within the part of the terrain that is visible. Clearly a re-planning exercise is required when the robot gradually explores the environment and acquires

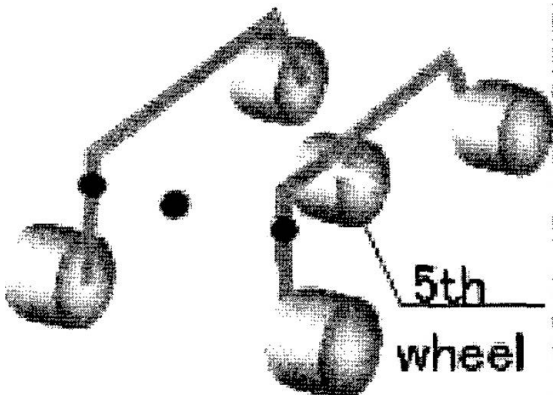


Figure 3.3: The Micro5 suspension system [5].

more information about the geometry of the terrain resulting in a new goal location being set.

It is assumed that the location of the robot and 3D model of the environment is available through external means such as a 3D SLAM algorithm [19]. In general, tracks and flippers should also remain in contact with the ground as much as possible to enhance locomotion [2]. This strategy will result in both better traction and smooth movement but due to sophisticated physical interactions sometimes it is not enough to come over the rough obstacles. Therefore, the significantly more complex issue of maintaining traction along the path is also not considered by assuming that either the vehicle is driven along the prescribed path by a human operator or that a suitable traction controller is available.

3.2 Related Work

Mobile robots have been used to solve a varied spectrum of problems to replace, augment or support human activities in several indoor and outdoor applications, such as factory and mining automation, exploration of hazardous environments, military systems and search and rescue operations. The various stability criteria proposed in the literature to analyse the qualitative performance of robot stability have mostly been adopted for tip-over monitoring and control, or off-line trajectory optimization. In contrast, alternative methods to pre-plan safer paths are fundamentally on continuous and smooth concepts such as using potential fields and irregular triangular meshes to model unstructured terrains and removing triangles from the navigational regions whose slope has proven to be too steep [41]. This is a conservative approach for reconfigurable robots as it will regard certain areas of the terrain non-traversable, whereas adopting alternative configurations may indeed render some discarded area safe to travel. Other

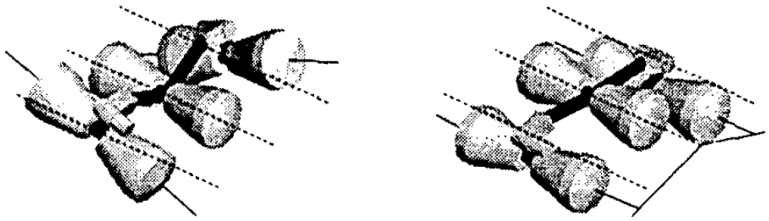


Figure 3.4: The three passive articulated of the chassis of robot Lama during an autonomous rough terrain traverse [55].

planners have considered reconfigurability within the context of numerical methods such as the Fast Marching geodesic approach. The solution to the boundary value partial differential equations guarantees smoothness and stability, yet needs to assume vast prior knowledge from the surroundings [56], which is not realistic.

The design of Robbie, a wheel-driven platform autonomous rescue robot has been presented in [4]. The robot generates 2D maps on the fly using an auto-levelled LRF, while low and overhead obstacles are detected using a 3D LRF as depicted in Figure 3.2. These 3D local scans are used to classify the terrain around the robot. The grid cells are classified into *unknown*, *occupied* or *free* based on the statistical properties of the 3D local scan points within the cell's square area without considering the robot's direction in that location. The autonomous search for victims is performed using exploration transform [57] on the basis of the acquired occupancy grid map. In this approach, the frontier-based exploration is combined with path transform and the cost function is calculated based on the cost of a path that goes to a close frontier instead of the cost of a path to a certain target. Whereby a novel discomfort cost function forces the robot to stay away from obstacles, there is no extra cost as long as the robot stays beyond a certain range to the closest landmark. The path is not checked for possible tip-over instabilities and some potential path-ways like stairs are marked as outright obstacles.

The tip-over stability was also used as a margin for path planning problems over rough terrains using a Digital Elevation Map (DEM) in [5] and [55]. In [5] a path planning method for a micro-rover wheeled robot illustrated in Fig3.3 on a planetary surface is discussed. The terrain traversability analysis is determined by prediction of roll, pitch and the height of the plane from the ground by supposing normal distribution for all parameters. In [55] the tip-over stability is used as a margin for safe elementary motion planning for a six wheeled rover

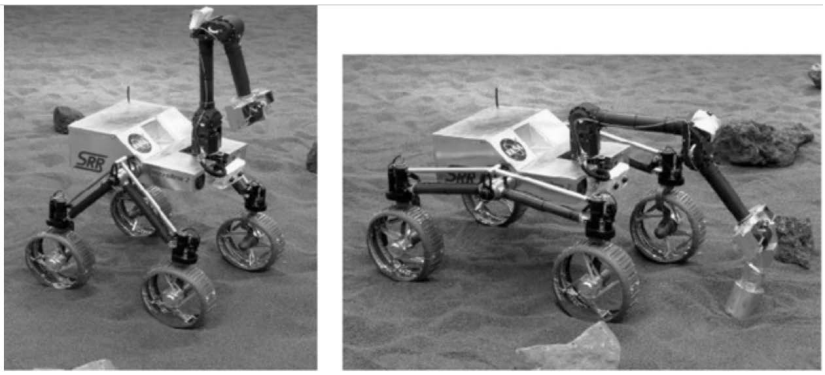


Figure 3.5: Jet Propulsion Laboratory Sample Return Rover (SRR) [6, 7].

Figure 3.6: EHR in the Amazon rain forest, stopped with one wheel placed on a tree trunk. The orientation control is enabled, and the robot body is parallel to the horizontal plane [23].

depicted in Fig3.4 with an articulated chassis. The proposed algorithm explicitly takes into account the geometric constraints on the chassis based on a digital elevation map built as the rover moves. The pitch and roll angles are considered as stability constraints on each of the three axles. In both works the CM of the system is not considered and the stability criteria used can't deal with extremely rough terrains like stairs and step-fields.

The FA stability measure has the advantage of clear geometrical description and low computational effort and is very appropriate to evaluate the tip-over stability. In [6] a performance index based on FA is optimized by Iagnenmma *el al* subject to vehicle constraints for stability-based kinematic control to change the configuration of wheel-legged Jet Propulsion Laboratory Sample Return Rover (SRR) robot as seen in Fig3.5. They assume that the vehicle navigates at low speed, and the robot dynamics are disregarded. Later Iagnenmma *el al* in [7] performed

Figure 3.7: The reconfigurable tracked wheelchair robot [8].

a technique to control tip-over stability of a wheeled rover with an actively reconfigurable suspension system. The FA margin is employed supposing a quasi-static model for the robot. The terrain is assumed to be rigid, and the wheels are assumed to make point contact with the terrain. Supposing rigid-body kinematic equations, wheel-ground contact angles are predicted by utilizing an extended Kalman filter to fuse on-board sensor signals.

More recently an active control strategy using the FA margin was developed for reconfigurable mobile robots operating over irregular terrains [23]. The algorithm proposed a multi objective optimization approach for ground clearance, orientation, gradient stability margin and wheel traction efficiency. The validity of this technique was verified through numerical simulations and experimental tests using a reconfigurable wheel-legged mobile robot, the Environmental Hybrid Robot (EHR), shown in Figure 3.6. Contact points were assumed located at the leg terminations resting on planar contact terrain. The robot frame origin coincides with its CM so the configuration of the robot has no affect on the CM position in the robot coordinate system. The performance of the active control strategy was shown to depend heavily on the quality of the feedback data, and more precise wheel-terrain interactions were highlighted for revision.

More general approaches for the stability control of reconfigurable mobile robots have also taken into account other constraints, e.g. traction optimisation [28], [29]. Both these works

Figure 3.8: Reconfigurable tracked mobile modular manipulator [9].

employed the original FA. In [28], the performance index considered the stability measure for each potential tip-over axis and the nominal values of the joints. The minimization of this performance index provided the most favourable configuration of the robot. A combination of the stability measure with an artificial potential field to obtain the demanded actuator values was used in [29].

The tip-over stability control of a tracked wheelchair robot Fig3.7 equipped with a variable-geometry-tracked mechanism during stair-climbing is proposed in [8]. The tip-over stability analysis and simulation are performed with original FA stability measure [36], and the variation of the tip-over stability margin of the robot under different conditions of passenger's attitude and action during stair-climbing is performed. To simplify the analysis, it is assumed that the robot will not yaw during stair-climbing, so the resultant stability axes have a constant angle in the robot coordinate system. The ground contact points are predicted based on the attitude of the robot and the distance travelled on the stairs. The FA stability measure has good performance and is very appropriate in evaluating the tip-over stability of stair-climbing of this new-style wheelchair robot during which the varieties of robot configuration, passenger attitude and ground contact points are all very large [8].

The recent work on the stability control of a reconfigurable tracked robot is presented in [9]. A real-time tip-over stability criteria for a reconfigurable tracked mobile platform Fig3.8 on slopes was derived on the basis of load transfers by judging the supporting force generated at the concerned tracked-terrain contact points. The proposed avoidance algorithm considered

the contact points to be fixed under the sprockets in order to describe the interactions between tracks and terrain. This is a strong assumption for the case of highly unstructured terrains such as those featuring rubble, over-sized obstacles, stairs etc. For a car-like vehicle having wheels greater than three the prediction of contact points constitutes an indeterminate problem [58] because it cannot be ensured that all wheels touch the ground at all instants. Consequently considering contact points under sprockets also causes an indeterminate problem.

3.3 Stable Path Planning

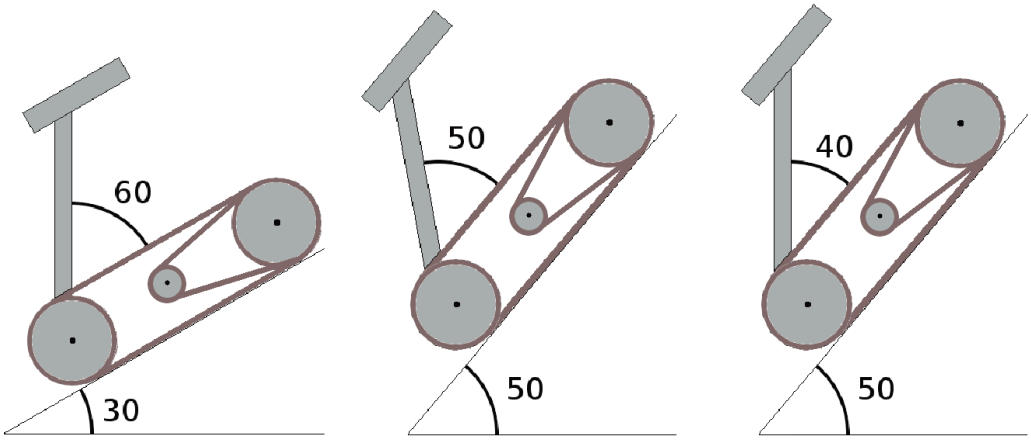
It has been argued in previous sections how the posture of the robot along a path is a critical factor that should be accounted for when planning safe paths. A planner therefore needs to prove stability at a given location and a suitable algorithm predicated on the contacts forces between robot and terrain has been proposed in Section 2.6. Furthermore it has also been argued how other objectives may also be accounted for to come up with practical routes to follow. To leverage how the proposed stability constraint and the reconfiguration cost can be accounted for planning, a method is proposed in this section which integrates these constraints within the well known A* planner algorithm, briefly described in Section 3.3.2 for reference. It should be noted, however, that any cost-driven planner would have been equally suitable.

3.3.1 Reconfigurability Objective Function

There are in general a large number of conflicting objectives that can play a significant role when planning paths in the context of realistic scenarios. For reconfigurable platforms in particular, favouring some nominal poses over others is very much dependent on the purpose of the application. For instance in search and rescue applications, a high position of the arm is generally preferred to afford wider sensor views, but this is in conflict with more stable posture which prefers a lower arm position to keep CM as low as possible. As another example, during controlling legged or wheel-legged planetary rovers, a ground clearance safety margin should always be applied during the operation, otherwise the resulting configurations may be inconsistent and put the robot is jeopardy [23].

In this work, oriented towards search and rescue operations, the nominal configuration for the robot at a given point is assumed as:

1. The arm angle that affords the highest sensor height ($\phi_a nom$).



(a) $pitch = 30, \phi_a = 60$. **(b)** $pitch = 50, \phi_a = 50$. **(c)** $pitch = 50, \phi_a = 40$.

Figure 3.9: Robot's *pitch* and nominal arm angles ϕ_a at two consecutive way points, 3.9a and 3.9c, and the final arm configuration obtained by the reconfigurability function 3.9b.

2. The flippers angle that provides the largest track-terrain interaction area ($\phi_f nom$).

To gain better traction, it is recommended that the tracks of the robot make the maximum possible contact with the terrain, however in some cases a smaller interaction area is better e.g if the robot is turning. An optimal solution for flipper-terrain control interaction is undoubtedly an important challenge in itself that has been left out of the scope of this work. Hence, in the rest of this work where the key objective remains stability planning it is assumed that either the vehicle can be driven along the prescribed path by a human operator or that a suitable traction controller is available to do so.

Moreover, increments in the joint configuration between way-points should also be considered to make sure the robot can follow the planned motions. Energy consumption is also a particularly relevant criteria when platforms are battery powered, so that changes in posture are kept to a minimum. The stability of the robot remains, however, the critical constraint so that if robot is ever found to be unstable, the optimality of any other parameters should be scarified to always satisfy the stability margin. The reconfiguration cost function of the robot U_c is proposed as

$$U_c = \sum_{i=1}^{nrj} Ui \tag{3.1}$$

where $nrj = 2$ is the number of the robot's joints (i.e. for the Packbot model in this work $i \in \{1,2\}$ are chosen for flippers and arm joints respectively) and U_i is the reconfiguration cost function for ith joint which is proposed as

$$U_i = Up_i + Ue_i \tag{3.2}$$

where Up_i is the cost associated to the desired position and Ue_i is an energy term associated to the reconfiguration of the joint. To better understand the meaning of these terms, the example depicted in Figure 3.9 will be employed. For simplification only the arm angle is presented and the flippers are excluded from this case and they are supposed to be fixed at $\phi_f = 180^\circ$. Suppose that the robot is sitting in a 30° ramp and current arm angle is $\phi_a = 60^\circ$ (Figure 3.9a). The path planner has suggested the next way-point to move to is sitting in a 50° incline where the nominal joint position will be $\phi_a nom = 40^\circ$ (Figure 3.9c). Supposing the movement limitation of the arm joint is $\phi_a min = 0^\circ$, $\phi_a max = 180^\circ$ and the maximum permitted arm movement between way-points has been set to $\Delta\phi_a max = 40^\circ$. It will be shown that by considering other costs, the optimal movement between these two way-points will be 50° as shown in Figure 3.9b. It should be noted that the derivation of the proposed optimal solution is dependent on the joint's parameters, current angle and nominal position at a given time and location in the map, and therefore this needs to be recalculated at each point again. As depicted in Figure 3.10a, Up_i is composed of Upc_i a quadratic function on the joint cost, and a boundary function Upl_i which depicts angle limitations.

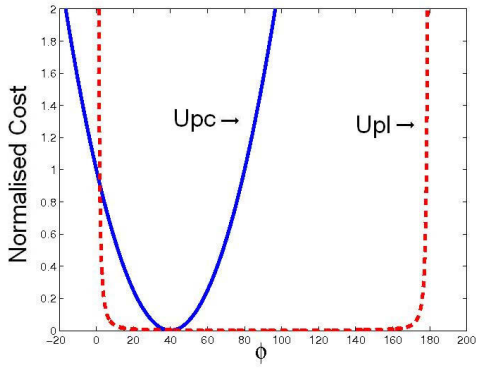
$$Up_i = Upc_i + Upl_i \tag{3.3}$$

The quadratic function Upc_i is introduced to penalize joint positions that deviate from the nominal joint position with quadratically raising costs

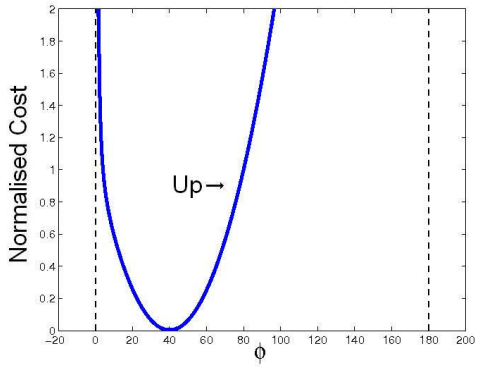
$$Upc_i = Kpc_i\,(\phi_i - \phi_i nom)^2 \tag{3.4}$$

where ϕ_i and $\phi_i nom$ denote the angle of the arm joint, and its nominal position respectively. Kpc_i is a normalizing factor dependent on $\phi_i nom$ by

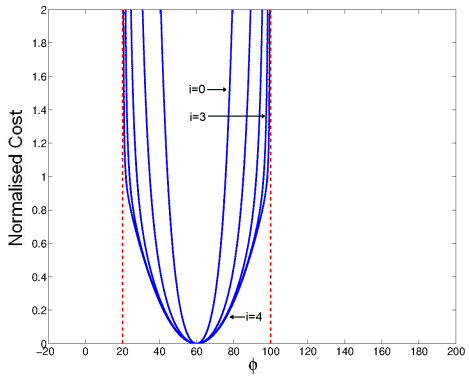
$$Kpc_i = \frac{1}{(\phi_i closer - \phi_i nom)^2} \tag{3.5}$$



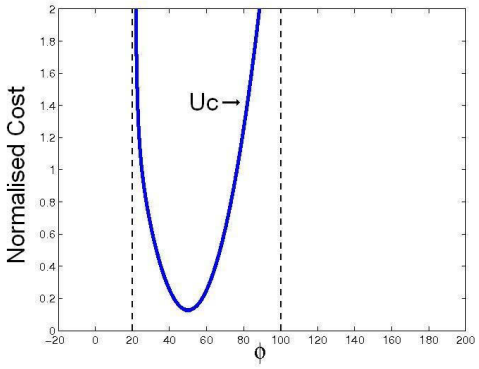
(a) Joint position and movement limitation costs.



(b) Desired position cost functions.



(c) The shape of the energy cost Ue for different values of $Kel_i = 10^{-n}$, $n = \{0,\ldots,4\}$.



(d) The reconfiguration cost function.

Figure 3.10: Optimal arm joint configuration and its constraints. In this example it is supposed that the current arm angle is $\phi_a = 60°$ and the nominal joint position is $\phi_a nom = 40°$.

where

$$\phi_i closer = \begin{cases} \phi_i min & |\phi_i min - \phi_i nom| < |\phi_i max - \phi_i nom| \\ \phi_i max & otherwise \end{cases} \tag{3.6}$$

i.e. $\phi_i closer$ indicates the joint limit closer to $\phi_i nom$ that makes Upc_i equal one, as depicted in Figure 3.10a.

The boundary function Upl_i in Equation 3.3 is defined to account for joint constraints as

$$Upl_i = \begin{cases} Kpl_i \left(\left(\frac{1}{\phi_i - \phi_i min} \right)^2 + \left(\frac{1}{\phi_i max - \phi_i} \right)^2 \right) & \phi_i min < \phi_i < \phi_i max \\ \infty & elsewhere \end{cases} \tag{3.7}$$

where Kpl_i is a constant gain defined so that Upl_i will have a significant effect on Up_i near the extremes, whereas Upc_i will stay dominant in the rest of the domain, i.e. the minimum value for Up_i is near $\phi_i nom$. The domain of Upl_i function is limited to $\phi_i min < \phi_i < \phi_i max$, which will in turn limit the domain of the resulting function Up_i to this area as well. The resulting Up_i cost function is pictorially depicted in Figure 3.10b. The nominal position for the arm is the around the middle of the motion range at 90°. Therefore this function would smoothly push the posture away from the extreme conditions at the boundaries. Moreover this function is differentiable at each point in its domain, in comparison with discontinuous hard limits. It is therefore possible to analytically introduce joint limits in the overall cost function.

Ue_i, depicted in Figure 3.10c, is defined following a similar reasoning: it consists of both a quadratic and a boundary term to restrict the general joint motion in postures between waypoints, as well as to address the limitations of the maximum angular speed of each of the joints respectively

$$Ue_i = \begin{cases} Kec_i\,(\Delta\phi)^2 + Kel_i\left(\frac{1}{\Delta\phi_i max - |\Delta\phi_i|} - \frac{1}{\Delta\phi_i max}\right)^2 & |\Delta\phi_i| < \Delta\phi_i max \\ \infty & elsewhere \end{cases} \tag{3.8}$$

In the same way, as Kpl_i, Kel_i is a small constant gain, Figure 3.10c shows the shape of Ue_i (defined in Equation 3.8) for different values of $Kel_i = 10^{-n}$, $n = \{0,\ldots,4\}$. As shown for $n \geq 3$, the shape of Ue_i is not improving anymore, therefore in this work $n = 3$ (i.e. $Kpl_i = Kel_i = 0.001$) is chosen. $|\Delta\phi_i|$ is the absolute difference between two consecutive positions of the joint and $\Delta\phi_i max$ is the maximum possible gradient for the joint angle. Kec_i is a constant gain defined as

$$Kec_i = \left(\frac{1}{\Delta\phi_i max}\right)^2 \tag{3.9}$$

The suggested weighting factors defined by Equation 3.5 and 3.9 can be suitably modified to place more emphasis on nominal positioning or energy expenditure in the motion of the joint. The final joint reconfiguration cost function U_a for the example discussed is depicted in Figure 3.10d, where it can be analytically seen that the joint angle with the minimum cost at this location is 50°.

3.3.2 The A* Planner Algorithm

The A* planner is a heuristic algorithm widely applied in graph searching applications to find minimum cost paths. This algorithm ensures the optimality of the resulting path according to

Algorithm 1 The A* planner algorithm

```
1:  closed ← ∅
2:  open ← cell(start)
3:  while (open ≠ ∅) do
4:      cell(i) = min(open)
5:      closed ← closed + cell(i)
6:      open ← open − cell(i)
7:      for all cell(j) ∈ {8 successors of cell(i)} do
8:          if (cell(j) ∉ closed & cell(j) ≠ obstacle) then
9:              if (cell(j) ∈ open) then
10:                 refresh_node(i, j)
11:             else
12:                 add_open_node(i, j)
13:             end if
14:         end if
15:     end for
16:     if (cell(goal) = min(open)) then
17:         return path
18:     end if
19: end while
20: return path = ∅
```

any given cost function $g(n)$. The cost of each possible node is evaluated and ranked by a priority function $f(n)$

$$f(n) = g(n) + \alpha\, h(n), \quad \alpha \in \{0, 1\} \tag{3.10}$$

where $g(n)$ represents the accumulated cost from the node n back to the start point, and $h(n)$ is an estimate of the remaining cost to get from node n to the target node. The A* algorithm is widely employed in the robotics community to find minimum *distance* paths from *start* to *goal* nodes in a given static or evolving dynamic occupancy grid map. α is simply a weighting factor to make the search heuristic if an admissible function $h(n)$ exists. When $\alpha = 0$, A* is equivalent to Dijkstra's shortest path search algorithm [59]. [1]

As summarized in Algorithm 1, A* algorithm classifies the graph nodes into two sets re-

[1] When $\alpha > 1$, it is possible to make a trade off between optimality and speed (the so-called satisfying solutions) [18], but here only the optimum choice is selected. The admissibility of the A* algorithm is discussed in sub-section 3.5.2.3 giving an example where a planner employs two different objective functions of distance and stability/reconfiguration costs.

Algorithm 2 The A* function to refresh the cost of a node that is already in the *open* list.

```
1: function refresh_node(i, j)
2:     new_g = cell(i).g + dis(i, j)
3:     if (new_g < cell(j).g)) then
4:         cell(j).g = new_g
5:         cell(j).f = cell(j).g + α dis(j, goal)
6:         cell(j).parent = cell(i)
7:     end if
8: end function
```

Algorithm 3 The A* function to add a new node to the *open* list.

```
1: function add_open_node(i, j)
2:     cell(j).g = cell(i).g + dis(i, j)
3:     cell(j).f = cell(j).g + α dis(j, goal)
4:     cell(j).parent = cell(i)
5:     open ← open + cell(j)
6: end function
```

ferred to as *closed* and *open*. The nodes that have already been explored will be labelled as *closed* to avoid being considered again during the remaining path search. The cells in the *open* list are the nodes that will be considered next during the expansion process. At the beginning *closed* is an empty set and the cell of robot pose will be inserted to the *open* list. During planning, the minimum cost node in the *open* set will be considered as the *current* node. In each iteration, the *current* node will be moved to *closed* and all eight cells adjacent to this node will be processed with the functions $refresh\_node()$ and $add\_open\_node()$ as described in Algorithm 2 and Algorithm 3 respectively. If the adjacent node was neither in the *open* nor in the *closed* set and it is not an obstacle, $add\_open\_node()$ function will add it to the *open* list considering *current* node as its parent. If the adjacent node was already in the *open* list - meaning it already has gotten a parent - $refresh\_node()$ will calculate the new $g(n)$ cost to traverse through this node. Should the new cost be less than the current cost associated with traversing through this adjacent node, the *current* node will become its new parent, and A* costs for the node will be refreshed accordingly. It is an iterative process where the cell in the *open* list with minimum cost will be explored next and their parents will be updated in search of more optimal cost paths. The search ends when the goal node has the smallest f-value in the *open* list, or *open* is empty and no path to the target has been found. Once the goal cell is selected as minimum cost node, the final path will be the chain of parents from the goal back

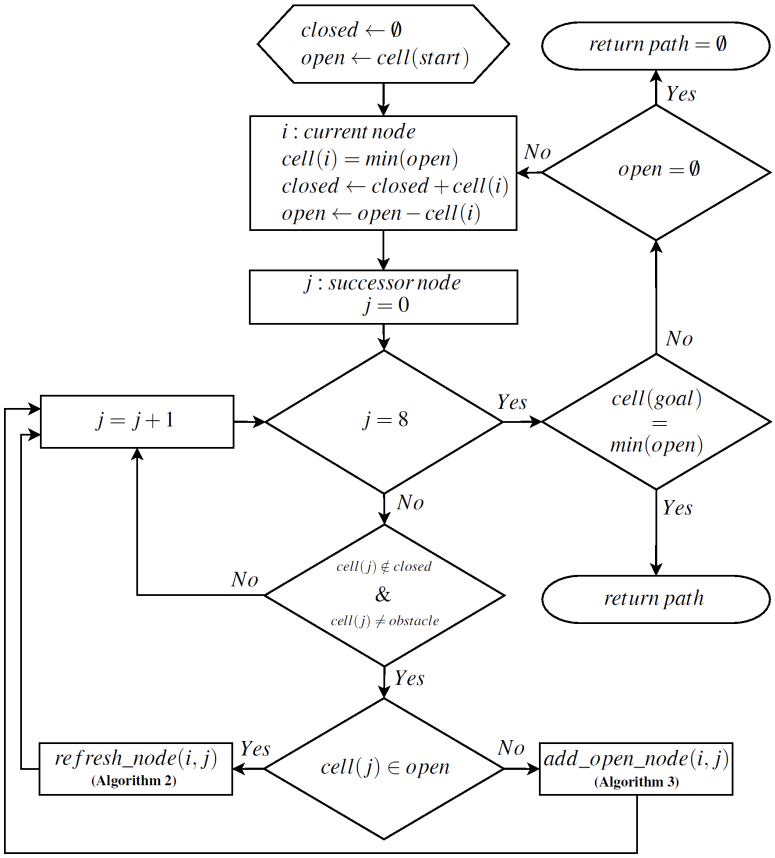


Figure 3.11: The block diagram of the original A* algorithm.

to the start node. The block diagram of the overall procedure is illustrated in Figure 3.11.

3.3.3 Optimal Kinematic Reconfiguration Algorithm

As described in Section 3.3.2, in A*, the accumulated term $g(n)$ in Equation 3.10 can be based on different costs such as Euclidean distance, time, energy, etc. In this section, a modification is proposed to also account for the additional parameters of interest, including energy expenditure, visibility, traction and an additional constraint to guarantee the stability of the path.

The reconfiguration cost of the proposed optimum stable high-visibility (OSHV) function is optimized between two successor nodes as described in the previous section, resulting in a 2D grid-based search space with two peculiarities: a) to ensure stability, a successor node will be added to *open* list or refreshed (gets new parent) if a safe transition is admissible by the reconfiguration ability of the robot, and b) the cost function is modified to take into account the stability and reconfiguration cost, and will be biased towards poses with better visibility and

Algorithm 4 Robot reconfiguration function

```
1:  function reconfig_robot()
2:      φ_a ← π/2
3:      φ_f ← π/2
4:      ode_simulate(φ_a, φ_f)
5:      φ_a nom ← π/2 − pitch
6:      φ_f nom ← φ_f tangential
7:      minimize Uc
8:      ode_simulate(φ_a opt, φ_f opt)
9:      if (β > β_min) then
10:         return Uc(φ_a opt, φ_f opt) // Uc_min
11:     else  // find suboptimal stable arrangement
12:         search_direction ← sign(pitch)
13:         while (φ_a ≥ φ_a min & φ_a ≤ φ_a max) do
14:             φ_a ← φ_a − search_direction ∗ search_step
15:             φ_f ← φ_f tangential
16:             ode_simulate(φ_a, φ_f)
17:             if (β > β_min) then
18:                 return Uc(φ_a, φ_f)
19:             end if
20:         end while
21:         if (search_direction has not changed) then
22:             search_direction ← −search_direction
23:             go to 13
24:         end if
25:         return ∞ // no stable arrangement exists
26:     end if
27: end function
```

traction. The reconfiguration cost described in Section 3.3.1 can be then reflected in the $g(n)$ term as

$$g(n) = \gamma Ud(n) + (1-\gamma)\,(Uc(n) + Us(n)), \quad 0 \leq \gamma \leq 1 \tag{3.11}$$

where Ud is the accumulated grid distance cost between successors (1 or $\sqrt{2}$), Uc is the normalized robot reconfiguration cost according to Equation 3.1 and Us is the normalised stability cost as

$$Us = 1 - \beta \tag{3.12}$$

The stability/reconfiguration cost of the graph is known up to where the search has expanded. Therefore the heuristic cost will be purely based on grid distance cost. γ is a simple weighting factor to be able to place more emphasis on stability/reconfiguration or distance as desired. (i.e. no need to process stability and reconfiguration calculations when navigating flat terrains for instance). The introduction of γ makes it possible to plan purely based on distance, while keeping the stability as a constraint and optimizing the reconfiguration of the robot between two successor way-points.

Despite the proposed analytical function Uc for the reconfiguration cost, the optimization cannot be expressed as a straightforward analytical problem. This is due to the stability constraint, and the fact that changes in the robot configuration may also change the robot's final pose in the terrain. In order to address these considerations the robot reconfiguration algorithm needs to be formulated as an iterative procedure, as summarized by Algorithm 4.

The algorithm first considers how the robot would sit on the terrain with a 90° arm and flippers configuration. The nominal configuration for the arm to provide the highest possible view point at that point in the terrain is considered the orthogonal angle in the global frame and the nominal posture for the flippers will be the angle which keeps them tangential to the terrain. The optimal configuration extracted by analytical minimization of Equation 3.1 is considered as the initial candidate values. Contact points prediction and robot stability analysis will be performed for this configuration as described in Section 2.6 and 2.4 respectively. If the robot with the analytical optimum configuration is proven unstable, the arrangement will then deviate from the true optimum to a near-optimum, but stable one.

If $pitch$ angle is positive (robot nose up), the *search_direction* will be positive to search the postures with smaller arm angles first. This process will be repeated iteratively to find the minimum configuration that meets the threshold of stability criteria with the lowest cost. Otherwise, Uc will be regarded as exceedingly large at that location in the map, effectively rendering the grid untraversable for planning purposes.

In order to merge the proposed stability constraint and the reconfiguration cost, the normal A* algorithm of Algorithm 1 will be replaced with the modified stable planner as summarized in Algorithm 5. The stable planner is different from the normal A* algorithm in the ninth and tenth lines of Algorithm 5. In the ninth line before processing the adjacent node, the algorithm first analyses the stability and reconfiguration cost of the robot and calculates the stability margin β using the $reconfig\_robot()$ function. The stability constraint would be checked at the

Algorithm 5 The stable A* planner algorithm

```
1:  closed ← ∅
2:  open ← cell(start)
3:  while (open ≠ ∅) do
4:      cell(i) = min(open)
5:      closed ← closed + cell(i)
6:      open ← open − cell(i)
7:      for all cell(j) ∈ {8 successors of cell(i)} do
8:          if (cell(j) ∉ closed & cell(j) ≠ obstacle) then
9:              reconfig_robot()
10:             if (β > β_min) then
11:                 if (cell(j) ∈ open) then
12:                     refresh_node(i, j)
13:                 else
14:                     add_open_node(i, j)
15:                 end if
16:             end if
17:         end if
18:     end for
19:     if (cell(goal) = min(open)) then
20:         return path
21:     end if
22: end while
23: return path = ∅
```

tenth line and if the given node has been found to be unstable, it will not be added or refreshed from the current direction. However, the node will still be considered for planning as it may be stable when approached from other directions. Under these conditions, the stability of the resulting path - that with the minimum reconfiguration cost - can be guaranteed. It should also be noted that the stable algorithm is using the modified g as defined in Equation 3.11. Figure 3.12 summarises the procedure in a block diagram form. Note also that this algorithm is using the normalised stability cost suggested in Equation 3.12, and also checks for unstable points. A non-normalised stability cost, tending towards infinity at unstable boundaries, makes the overall cost distribution uneven thus has the undesirable effect of being highly biased towards the stability cost parameter. Moreover, by checking stability against an established minimum stability metric, if it is desirable to plan with an additional safety margin for caution, it is possible to simply increase the stability margin of β_{min}. If stability cost was instead tending towards infinity at unstable boundaries, without a normalization, then the planner would always come up

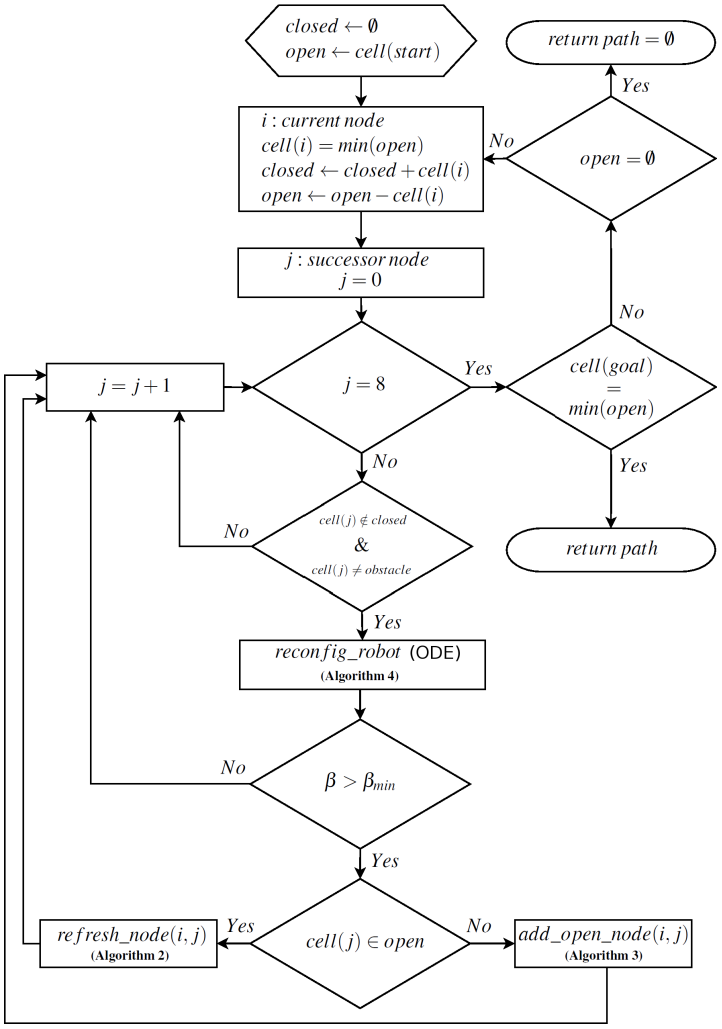
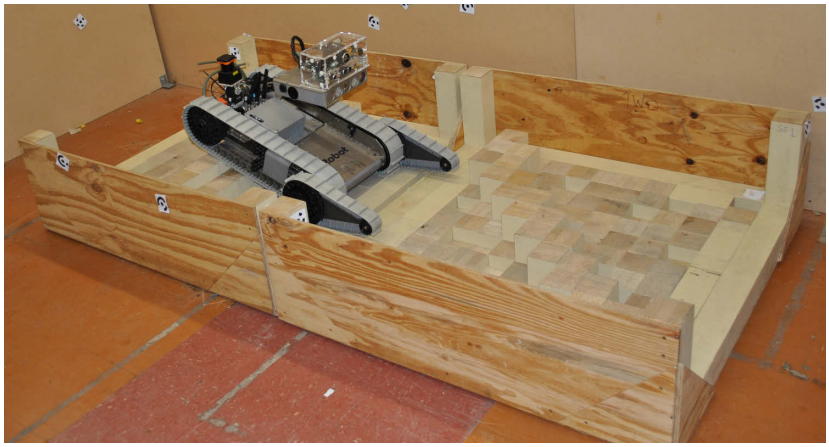


Figure 3.12: The block diagram of the stable A* algorithm.

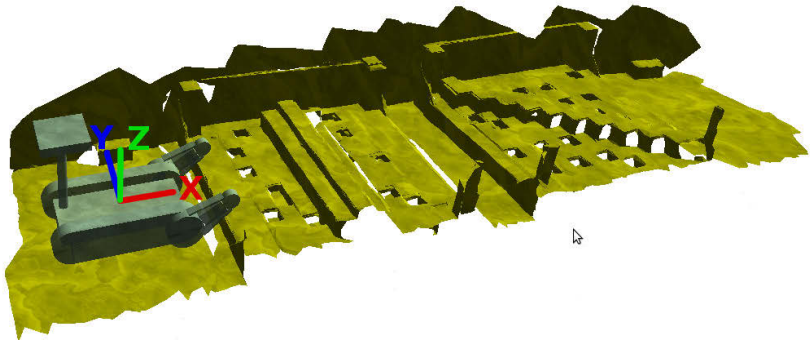
with a path even if a stable trajectory does not exist, an unwelcome situation in most scenarios, but unacceptable in many critical ones.

3.4 Experimental Results

The aim of this experiment was to evaluate the performance of the proposed algorithm in planning over a known step-field arrangement consisting of adjacent diagonal and hill configurations, as depicted in Figure 3.13. As depicted in Figure 3.13a, the two step-fields are placed in a straight line alongside each other. Consequently the resulting trajectory is expected to be



(a) The hill and diagonal step-fields



(b) The 3D meshed model

Figure 3.13: The hill and diagonal step-fields and their 3D meshed models.

a straight route. Therefore this experiment, in particular, is going to address the validity of the posture planning aspect of the search algorithm in a practical scenario.

In the beginning the global coordinate system and robot local frame coincide with each other. As depicted in Figure 3.14, X is the robot's roll axis, Y the pitch axis and Z the direction normal to the platform, or yaw's axis. In addition to the robot's position, the path way-points also include the robot's posture at the desired locations. Setting the start point in the beginning of the first step-field and the goal node at the end of second one, a relatively straight trajectory with 32 way-points was generated with off-line running the proposed planning algorithm in the ODE simulator. Each way-point includes 2D position of the robot on the grid as well as the flipper and arm angles in that location.

The robot was then placed at the same starting position in the real arena, and the result-

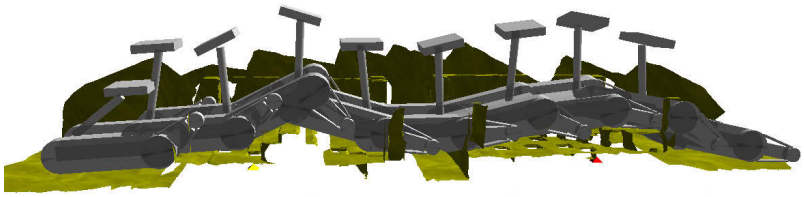


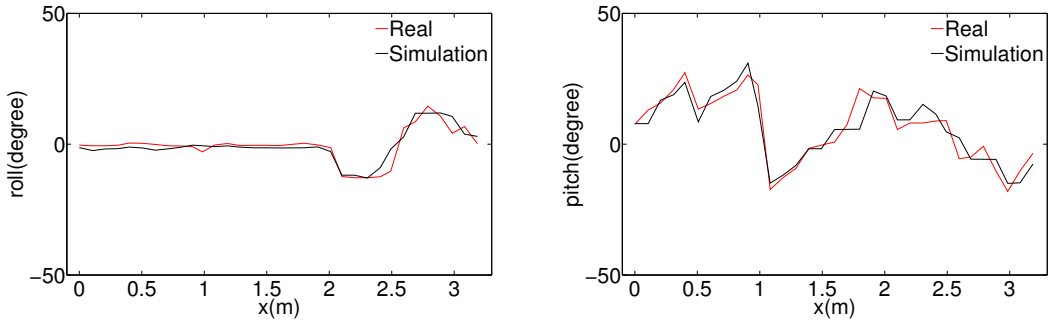
Figure 3.14: The side view of the robot configurations along the trajectory (direction: left to right).

ing way-points were loaded to a controller to follow the suggested trajectory and reconfigure the robot to the corresponding planned posture at the way-points. A localizer running of 2D range data from an auto-levelled laser scanner was used to derive an estimate of the robot pose (x, y, yaw) with a previously built 3D mesh of the arena depicted in Figure 3.13. Based on the position of the robot over the map, the controller aims to reconfigure the robot to the corresponding planned posture in the way-points. Data was recorded at 376 locations along the path traversed by the robot. A side view of a subset of the path with the robot arrangements suggested by the planner is depicted in Figure 3.14, where the robot configurations have been omitted in some places to increase clarity.

The recorded angles of the arm and flipper joints are shown in Figure 3.16a and 3.16b respectively. The robot inclination information measured by IMU is depicted in Figure 3.15a and 3.15b. A comparison between the measured and predicted vehicle $pitch$ and $roll$ derived from the ODE simulation at these locations was carried out to indirectly assess the proposed algorithm for calculating contact surfaces and stability measures. The results presented in the figures Figure 3.15b and 3.15a clearly indicate a close correlation between the real values and those inferred from the derivation of the ODE.

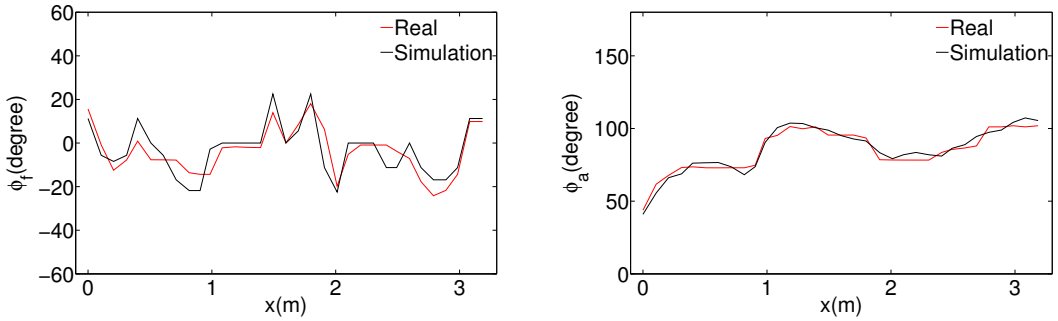
The comparison between the measured and the values set at the way-points for flipper and arm angles shows a reasonable controller performance, as depicted by Figures 3.16a and 3.16b respectively. These results report the outcome of one such trial of driving the robot over the step-fields. Further repeated tests have been performed which indicate that so long as the robot moves gently over the unstructured environment, hence keeping localization errors bounded, results are very similar.

This example shows the utility of the employed flippers behaviour - keeping the flippers tangential to the terrain - to overcome two common standard obstacles for the robot mobility test. The algorithm seems a practical choice to deal with the other two predominant rough impediments in indoor unstructured environments i.e. stairs and ramps as well.



(a) *roll* **(b)** *pitch*

Figure 3.15: Comparison between the measured and predicted inclination data.



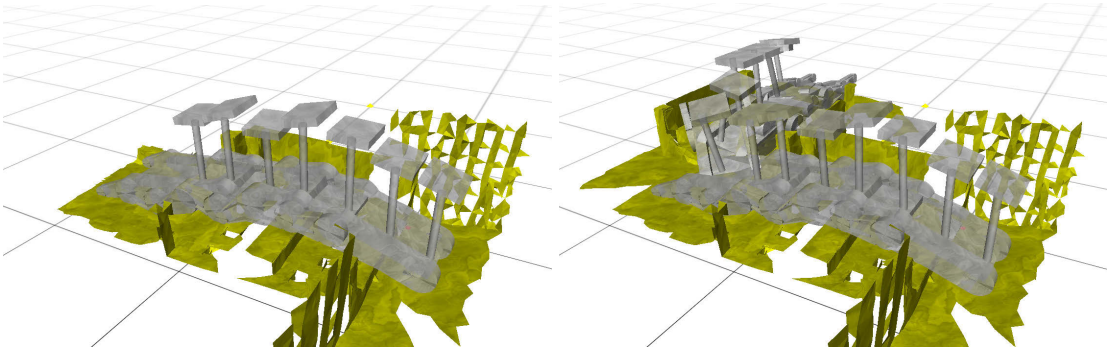
(a) flippers angle ϕ_f **(b)** arm angle ϕ_a

Figure 3.16: Comparison between the measured and set robot's configuration data.

3.5 Simulation Results

3.5.1 Path Planning in an Exploratory Setting

Assuming full prior knowledge of the terrain is not a realistic setting for fundamentally exploratory operations such as search and rescue where coverage increases as navigation progresses. Under these circumstances, the robot or the operator needs to make decisions as to where to move next as new information is collected. Results are now presented where the proposed strategy is exploited within this context in an experimental set-up where only partial information from the surrounding terrain is assumed in the planning process. As coverage increases, the algorithm is able to make more informative decisions, e.g. to suggest alternative routes to revisit places, but at any given time it is shown how local stable paths can be



(a) Path planning in terrain just scanned. **(b)** The path as further terrain knowledge is acquired.

Figure 3.17: Path planning with partial terrain knowledge (direction: right to left).

planned to guarantee safe motion towards goals set within the part of the terrain that has been mapped. This can be seen in the figures Figure 3.17a and 3.17b, where a path to a given goal is planned that takes advantage of the proposed stable reconfiguration strategy. The map shown in Figure 3.17a represents the partial information available to the robot in the first instance. As the robot traverses the proposed path, more information is gathered. The map shown in Figure 3.17b is the result of tracking the initial path (and acquiring more detailed information in the process for future planning), and planning again when the first section has been fully traversed towards a new goal set within the range of the newly acquired terrain data. The outcomes of the proposed algorithm are presented in such a scenario in Figure 3.18, where the robot configurations have been omitted in some places to increase clarity. The proposed optimal configuration planner results in smoother arm trajectories as depicted in Figure 3.19a and 3.19b, where comparison between the nominal trajectories planned by the nominal high-visibility stable paths planner [51] and the optimal postures obtained by the proposed planner are shown. Table 3.1 summarizes the energy reconfiguration costs (ERC) of the nominal and proposed optimal posture trajectories, showing improvement of 25.92% and 32.88% for arm and flippers respectively. Where ERC is defined as

$$ERC = \sum_{i=1}^{nwp} Ue_i \tag{3.13}$$

where nwp is the number of way-points in the trajectory and Ue_i is the energy cost according to Equation 3.8.

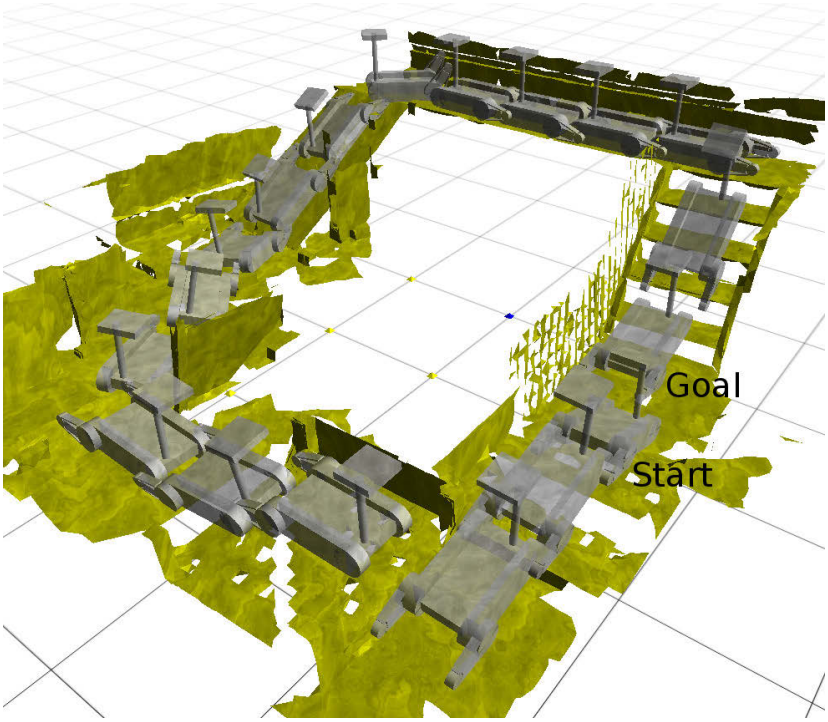


Figure 3.18: The outcomes of the path planning in an exploratory setting. The robot configurations have been omitted in some places to increase clarity. Robot moving clockwise.

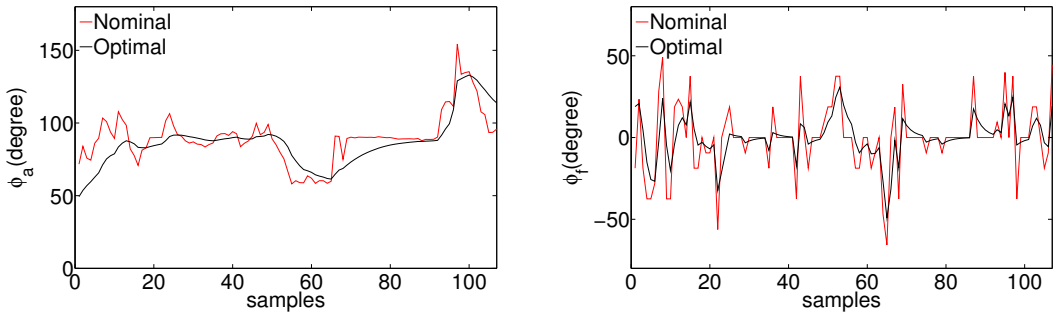
	Nominal	Optimal	Improvement
arm	909.73	673.93	25.92%
flippers	1323.41	888.27	32.88%

Table 3.1: Overall energy costs (*ERC*) for the arm and flippers trajectories shown in Figure 3.19.

3.5.2 Path Planning with Prior Terrain Knowledge

3.5.2.1 Path Planning in an indoor USAR arena

The effect of the proposed planner is further illustrated on the USAR arena depicted in Figure 3.20 in a comparison with the standard A*. For this experiment complete knowledge of the terrain was assumed. For these experiments, the layout has been changed to allow for longer paths. Eight possible directions of motion for each grid were considered. In order to make a fair comparison between the two planners a pre-processing algorithm was first applied to the terrain model to label out obvious untraversable areas, e.g. walls and markedly steep slopes. Results show the path derived from the original A* in blue. A stability analysis of this path



(a) arm. **(b)** flippers.

Figure 3.19: Arm and flippers trajectories planned by a high-visibility stable path planner [51] only, and with the additional posture optimisation proposed in this chapter.

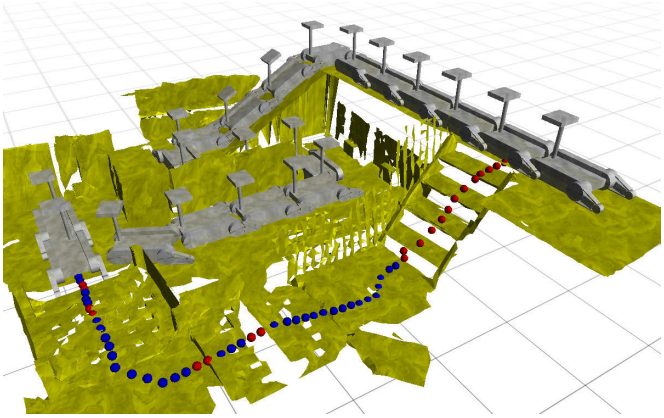
found sections (shown in red) where the robot was not stable for the fixed vertical arm and flipper pose assumed during a path with no reconfigurability. The values of the stability measures (βs) are plotted in Figure 3.21a, where negative values correspond to the red robot poses along the trajectory.

The most stable with lowest reconfiguration cost path (α and γ equal to zero in Equation 3.10 and 3.11 respectively) found by the modified A* is also shown with an outline of the robot pose at regular intervals. It illustrates how an alternative path was preferred by going up the ramp and avoiding the steps and the step fields, hence opting for more stable and lower reconfiguration cost poses over those leading to shorter paths.

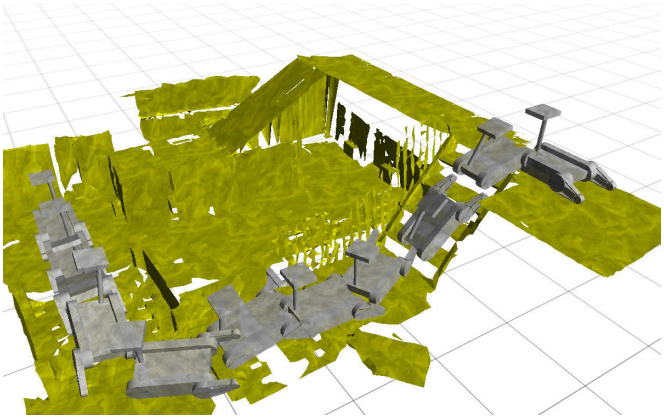
3.5.2.2 Path Planning in quasi-outdoor UTIAS arena

Apart from indoor applications such as search and rescue missions, mobile robots have been frequently employed in the outdoor and planetary surface exploration tasks as well. In such missions, mobile robots are required to predict their stability to avoid hazards such as tip-over or collision with obstacles.

In this example the proposed algorithm is tested with 3D terrain data collected from the quasi-outdoor UTIAS rover testing arena [60]. The UTIAS testing facility consists of a large dome structure, which covers a workspace area 40m in diameter. A 3D model of the environment has been obtained using an infrastructure-based ground-truth localization system suitable for deployment in large work-site environments. The set-up is able to provide full 6D relative localization for three-dimensional LRFs with centimetre-level accuracy in translation, and half



(a) Most stable and ordinary path.

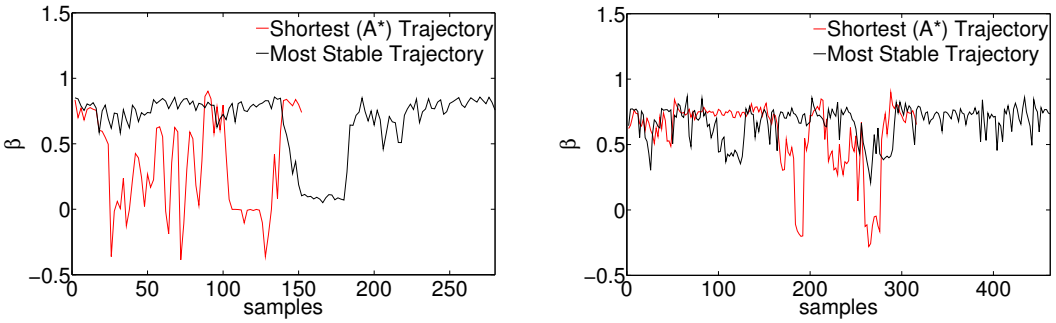


(b) Stable path.

Figure 3.20: Results of stability criterion in A* algorithm in the USAR arena. Standard A* path is shown in blue, with unstable regions depicted in red. Robot poses derived from the most stable with lowest reconfiguration cost A* path are outlined in light grey.

degree accuracy in orientation. Figure 3.22 depicts a panoramic image of the UTIAS indoor rover testing facility, with the rover used to gather the data in the foreground, and three of the four reflective signs used by the ground-truth localization system mounted on the dome structure in the background. These datasets are available online, and for more information, the reader is referred to [60].

Given the large dimension of the UTIAS arena in relation to the robot's size, path planning results are illustrated from a top-down view in Figure 3.23. Grey-scale colour coding indicates the height of the terrain from 0 to 2.76m. A pre-processing algorithm based on terrain gradients was first applied to the model to label out obviously untraversable steep slopes, shown in dark brown. This effectively filtered out regions with a more than a 56° gradient which corresponds



(a) USAR arena. **(b)** UTIAS arena.

Figure 3.21: Stability margins (βs) along the trajectories.

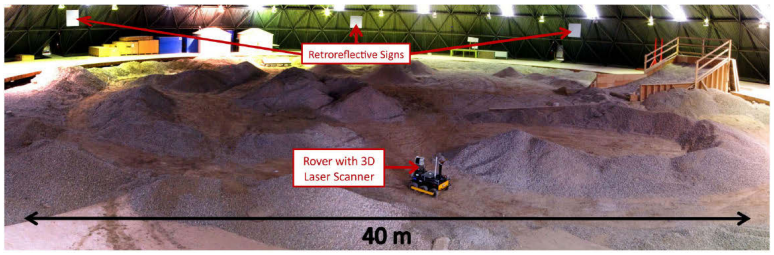


Figure 3.22: A panoramic image of the UTIAS indoor rover testing facility, with the rover used to gather the data in the foreground [22].

to the critically unstable point in *pitch* for a robot sitting flat with an arm configuration at 90°, i.e. the highest visible point in flat terrain. Results show the path derived from standard shortest A* in blue, where unstable points with a fixed configuration of 90° arm and flipper are shown in red. The proposed shortest stable and most stable paths are shown in black and yellow respectively.

It can be seen how the standard and proposed A* paths more or less coincided up to the middle of the second hill (in lighter grey), where the robot's stability appears compromised for a fixed configuration vehicle. The values of the stability measures (βs) are plotted in Figure 3.21b, where negative unstable values correspond to the red spots along the trajectory. The 3D model of the UTIAS arena and the robot configurations along the shortest stable trajectory is illustrated in Figure 3.24, where the robot configurations have been omitted in some places to increase clarity.

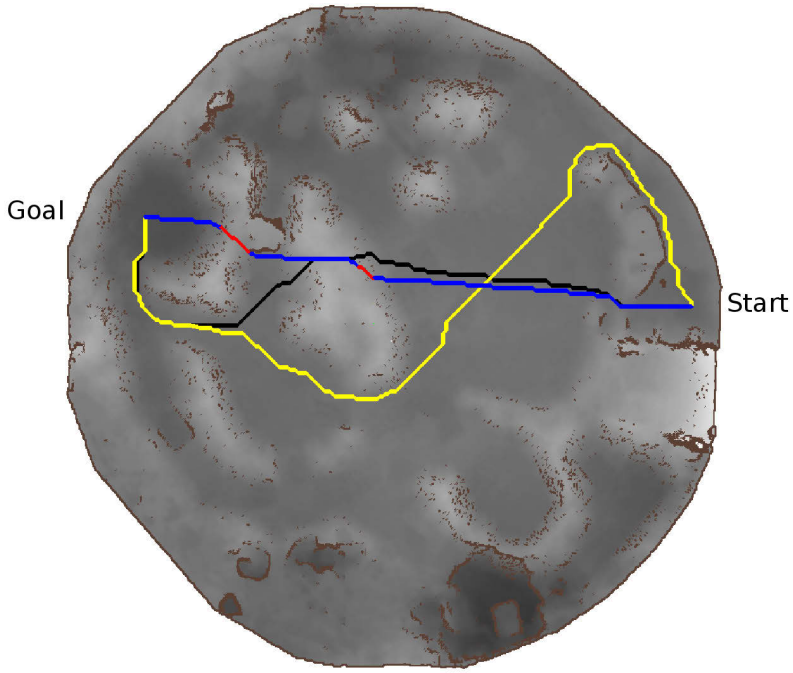
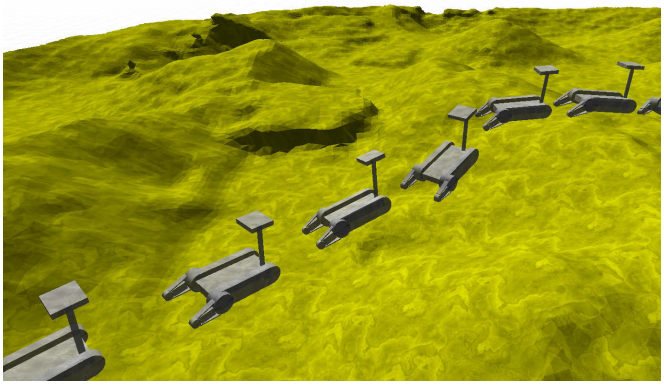


Figure 3.23: Results of stability criterion in A* algorithm in the UTIAS arena. Grey-scale colour coding indicates the height of the terrain from 0 to 2.76m. Obvious untraversable steep slopes are shown in dark brown. Results show the path derived from the original A* in blue with unstable points in red. The shortest stable path (α and γ equal to one in Equation 3.10 and 3.11 respectively) is shown in black. The most stable with lowest reconfiguration cost path (α and γ equal to zero in Equation 3.10 and 3.11 respectively) found by the modified A* is also shown in yellow.

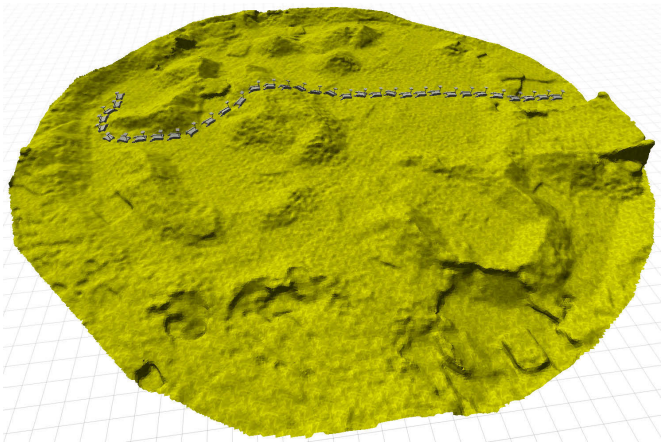
3.5.2.3 The Admissibility of A*

According to the Equation 3.10 and supposing existence of a path form start point to the goal node, the A* heuristic algorithm in graph searching would result in the minimal cost path ($g = g_{min}$, $f = f_{min}$) if it is admissible. The A* is admissible if the heuristic function $h(n)$ is a lower bound on the true minimal cost from node n to a goal node [61]. An admissible A* is able to find the shortest path by investigating the minimum number of the nodes [18]. Employing an heuristic function $h(n)$ may mislead the planner to a path which is not the minimum cost trajectory ($g \neq g_{min}$, $f = f_{min}$). In fact in this case the priority function f would be minimum, but the objective function g is not guaranteed to be the minimum anymore.

When searching for the shortest path, a straight line is an admissible option for h as it is always the minimum distance to the goal. But when the stability and reconfiguration parameters are contributing to g as suggested by Equation 3.11, it is hard to estimate the minimum cost to the goal and keep the A* planner admissible. Hence, the admissibility of the stable A* planner



(a) Zoomed view of the robot configurations around the middle of the trajectory.

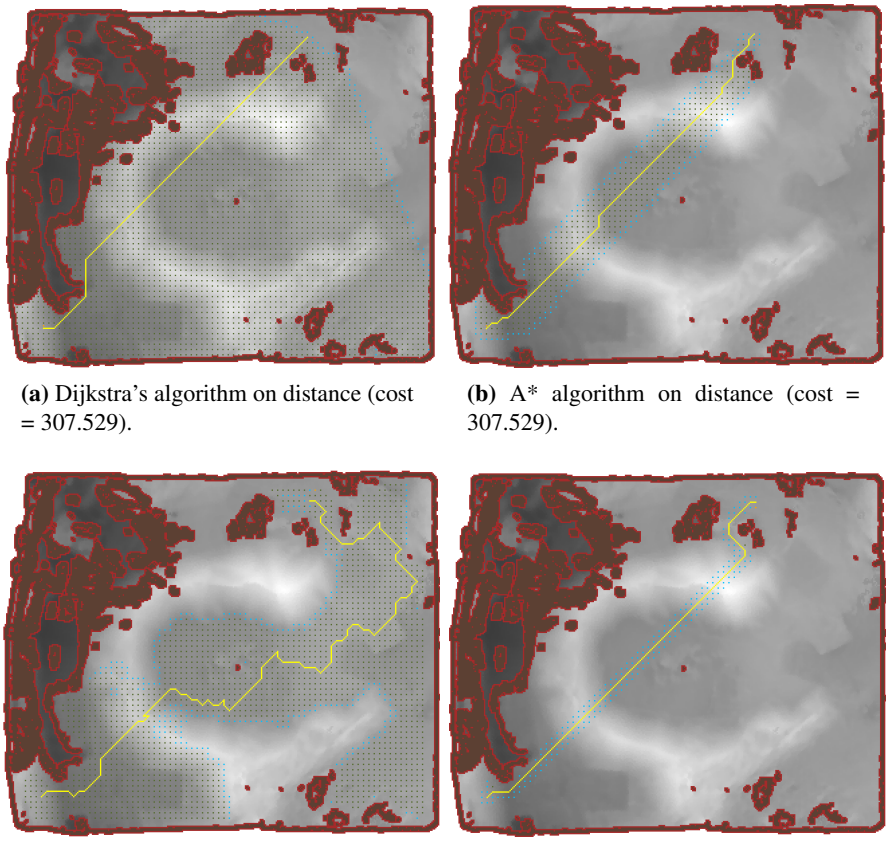


(b) Whole 3D model of the UTIAS arena.

Figure 3.24: The 3D model of the UTIAS arena and the robot configurations along the shortest stable trajectory. The robot configurations have been omitted in some places to increase clarity.

can be guaranteed if g remains solely based on distance .i.e when the $\gamma = 1$ in Equation 3.11. Although using an Euclidean distance heuristic function results in reasonable outcomes for planning stable paths, finding an admissible heuristic function for stability and reconfiguration cost has been left for future work of this research. It is important to note though that setting a constraint on stability still ensures the safety of the resulting trajectory and this does not impose a limitation to the admissibility of the stable heuristic A*.

To investigate this discussion more, the example shown in Figure 3.25 is comparing the resulting trajectories and the amount of investigated graph of the Dijkstra and heuristic A* algorithms for two objective functions of distance and stability/reconfiguration costs. The outcome of the Dijkstra and A* algorithm on distance are shown in Figure 3.25a and 3.25b respectively



(a) Dijkstra's algorithm on distance (cost = 307.529).

(b) A* algorithm on distance (cost = 307.529).

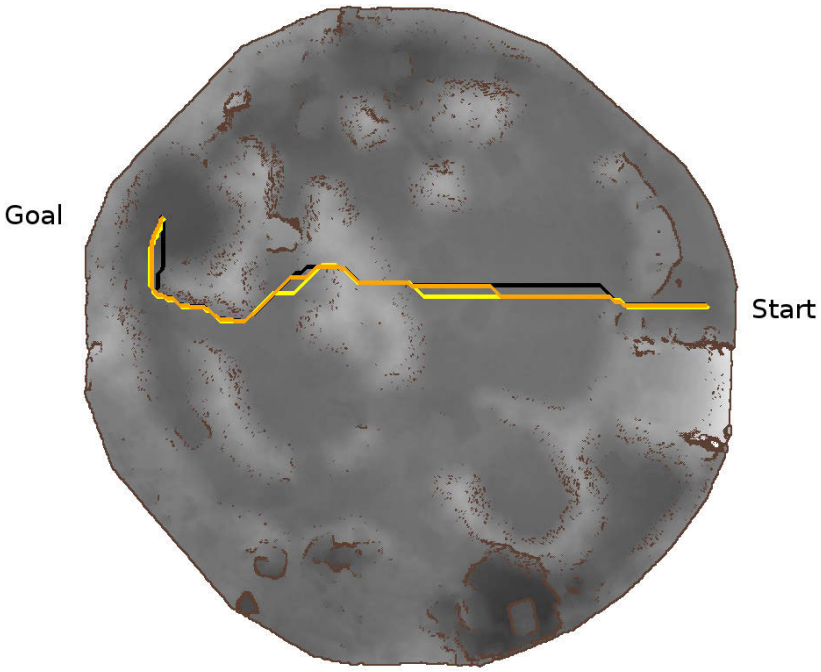
(c) Dijkstra's algorithm on stability (cost = 142.109).

(d) A* algorithm on stability (cost = 193.614).

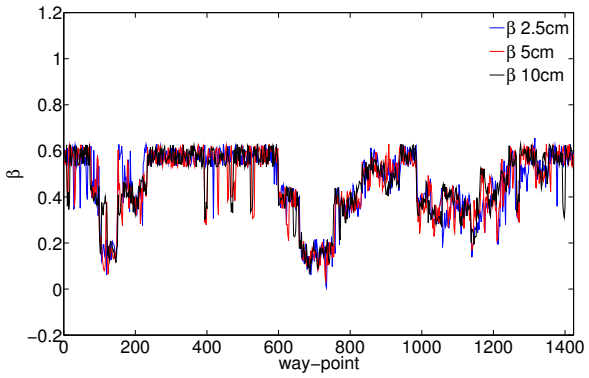
Figure 3.25: The Dijkstra's algorithm vs heuristic A* algorithm.

while Figure 3.25c and 3.25d are illustrating the resulting trajectories of the Dijkstra and A* algorithm on stability/reconfiguration consequent cost.

Both examples of Figure 3.25a and 3.25c which are employing the Dijkstra algorithm have investigated bigger area of graph in comparison to the corresponding cases shown in Figure 3.25b and 3.25d that are using the A* algorithm. When the objective function is distance, regardless of the bigger search area, both Dijkstra and heuristic algorithms result in two paths with the same cost of 307.529. It is proven that if the minimum path was unique, both techniques result in the same trajectory[18], but in this example there is more than one solution to true minimum path and therefore these two algorithms have found different routes, but again with the same cost of 307.529.



(a) The three trajectories from top view.



(b) The stability margin β over three trajectories. The values of β are repeated in the trajectories with lower resolutions.

Figure 3.26: The effect of the grid resolution on the resulting trajectories in the UTIAS arena. The resolution of the way-points in yellow, orange and black are 2.5cm, 5cm and 10cm respectively.

According to the outcomes shown in Figure 3.25c and 3.25d when the objective in g is just stability/reconfiguration cost and the heuristic part in h is purely distance, neither the resulting trajectories nor the overall costs are the same which proves that the A* algorithm is not admissible anymore.

3.5.2.4 The effect of the grid resolution on the resulting trajectory

The resolution of the grid has a big effect on A* planner speed and efficiency. If the resolution of the grid is scaled by factor n, the number of grid cells will increase by the square of the scale factor i.e. n^2. Therefore a small grid resolution may make the search space inefficiently large and in return a big grid size may filter out a potential path or cause a big unstable jump between two successor way-points.

The effect of the grid resolution on the resulting trajectories is investigated via a simulation in the UTIAS arena in Figure 3.26 where Figure 3.26a is illustrating the trajectories from top view in yellow, orange and black for 2.5cm, 5cm and 10cm resolutions respectively and Figure 3.26b is depicting the stability margin β over these three trajectories accordingly. This example shows that planning with 10cm resolution in an outdoor arena is still fairly close to a grid with a four times bigger resolution of 2.5cm. Planning in an outdoor terrain gives more freedom to a robot than an indoor arena, therefore in this work we have chosen 5cm and 10cm as the grid resolutions for indoor and outdoor terrains respectively.

3.6 Summary

A strategy for motion planning in challenging environments for reconfigurable robots such as those equipped with manipulator arms and/or flippers has been presented in this chapter. The proposed mechanism incorporates stability estimates when planning for safer high visibility navigational routes based on predicted robot-terrain interactions. The algorithm is able to minimise the length of the traversed path and optimise over conflicting robot reconfiguration cost between consecutive graph nodes in terms of visibility, traction and energy consumption in the presence of stability constraint and limitations in the robot's joint movement. To enhance the locomotion and maximise the track-terrain contact area, the nominal configuration for flippers is assumed to be the angle which keeps them tangential to the terrain. The performance of this influential flippers behaviour is presented by the experimental results over common indoor obstacles in a mock-up USAR arena. Experimental and simulation results with a reconfigurable tracked robot model have been presented to validate the proposed approach over more simplistic approaches that do not account for reconfiguration in the path estimates. This chapter has not been concerned with the uncertainty in the map, pose and robot posture in the proposed planning process. A statistical analysis of stability prediction to account for these uncertainties is presented in the next chapter.

4

Uncertainty Analysis

4.1 Introduction

Stability prediction is an important concern for mobile robots operating in rough environments. Having the capacity to predict areas of instability means pro-actively being able to plan safer traversable paths. The most influential tip-over stability measures are based on two criteria: the robot's CM and the supporting-polygon defined by the convex area spanned between the ground contact points. However, there is significant uncertainty associated with many parameters in the planning pipe-line: the actual robot dynamic model, its localisation in the ground, and the terrain models, particularly in uneven terrain. This chapter proposes a statistical analysis of stability prediction to account for some of the uncertainties. This is accomplished using the FA stability measure for the Packbot robot model. Probability density function (PDF) of contact points, CM and the FA stability measure are numerically estimated, with simulation results performed on ODE based on uncertain parameters. Two techniques are presented: a conventional Monte Carlo scheme, and a structured unscented transform (UT) [62] approach which results in significant improvement in computational efficiency. Experimental results on maps obtained from a range camera fitted on the sensor head while the robot traverses over a ramp and a series of steps are presented confirming the validity of the proposed probabilistic stability prediction method.

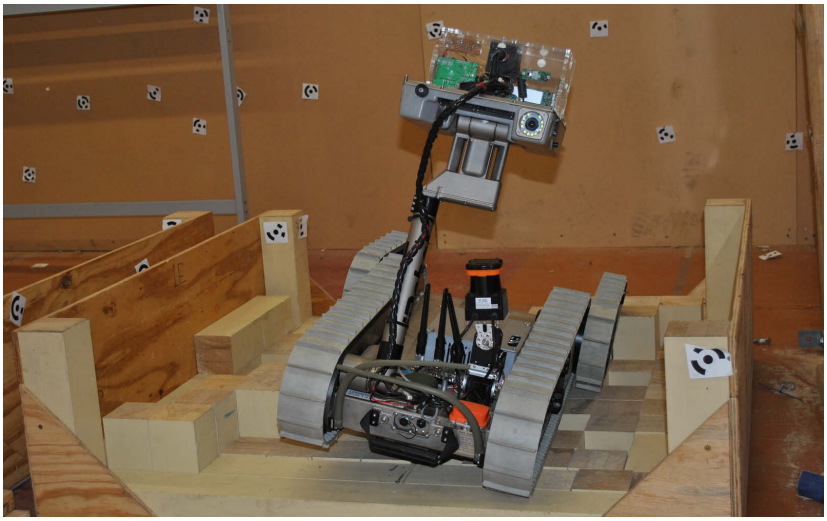


Figure 4.1: The iRobot Packbot robot, with an additional sensor head.

4.2 Related Work

Mobile vehicles are frequently employed in unstructured environments where high levels of mobility are required. These systems have been performing a significant role in field environment missions such as search and rescue [2], agricultural vehicles [14], mobile manipulators [15] and planetary surface exploration [10]. In such missions, avoiding tip-over is a major concern because it often results in collateral damage to the robot and the general surrounding environment [16].

Stability prediction is an important task required to successfully plan safe paths on challenging terrain. A robot's tip-over stability is a function of the robot configuration, environment geometry as well the position of the robot over the terrain. For an articulated rescue robot such as the iRobot Packbot shown in Figure 4.1, moving the on-board arm attached to the robot significantly impacts the position of the CM. Moreover, swinging the flippers (the two small front sub-tracks) can generally lead to changes in the supporting-polygon of the robot, defined by the contact points of the robot in the terrain.

As mentioned earlier, several tip-over stability measures have been proposed in the literature to evaluate the stability of a robot and predict unstable conditions. A study of the influence of a variable CM and the shape of the supporting-polygon to plan safe traversable paths for the case of reconfigurable robots was also proposed in previous chapters, where known local terrains and robot poses were assumed. Given uncertainties in the various parameters involved,

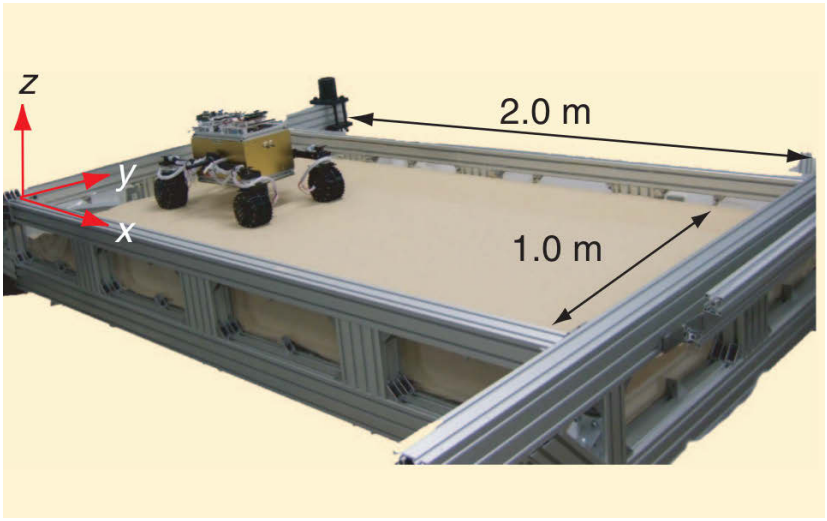


Figure 4.2: Articulated vehicle and the tilt-able test field for mobility prediction [10].

a number of works have been developed in the literature. A motion planning algorithm for a wheeled mobile robot model that included some arbitrary fixed uncertainties in the terrain measurements and rover localisation was presented in [63]. A statistical method for planetary surface exploration focused their efforts on incorporating uncertainty in two deformable soil parameters: cohesion and internal friction angle [10]. The mobility prediction for the wheeled robot was performed via calculating wheel-terrain interaction forces and vehicle dynamic motions. A stochastic response surface method (SRSM) [64] was used as a functional approximation technique to obtain an equivalent system model with reduced complexity. The validity of the proposed technique is confirmed through the comparison between predicted and experimental motion paths of the rover in a tiltable test field as shown in Figure 4.2. The computational efficiency of SRSM was confirmed through comparison to a standard Monte Carlo (SMC) method [65], and the Latin hypercube sampling Monte Carlo method [66].

More recently, two polynomial chaos approaches for mobile robot dynamics prediction given uncertainty in an obstacle's height, vehicle mass and surface type was presented in [11]. A ground vehicle simulator based on the Open Dynamics Engine (ODE) [47] was utilised to compare the results from the experiments conducted with a wheeled mobile robot (the Pioneer P3-AT). The accuracy of these polynomial chaos approach methods was validated through a baseline reference with the results obtained from the SMC approach. The robot was made to traverse over small step obstacles as illustrated in Figure 4.3. The robot's inclination was computed based on the strong assumption that the wheels always remained in contact with the

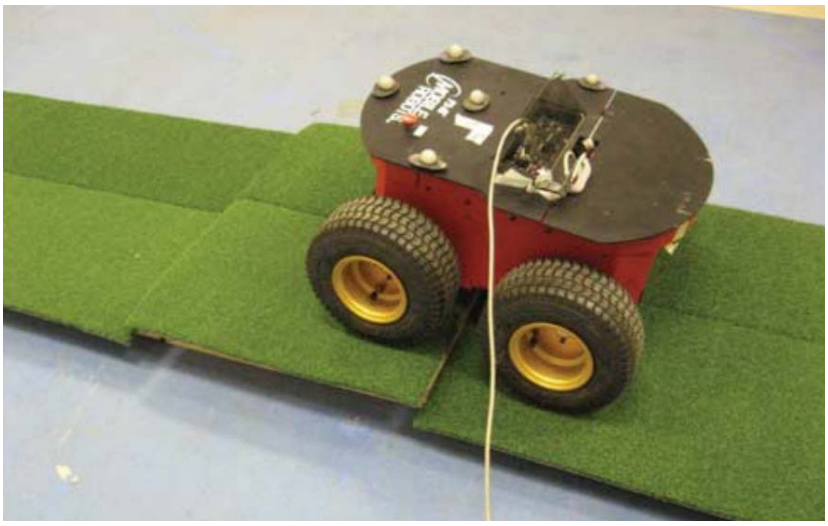


Figure 4.3: Experiment using a Pioneer P3-AT mobile robot over small step obstacles [11].

terrain, not generally the case in more unstructured environments, e.g. over rubble, or when traversing larger inclined surfaces.

The majority of the tip-over stability criteria mentioned above is predicated on the position of the CM and the SP. While there are propositions that take into account the uncertainties associated with some parameters, most rely on deterministic analysis that assume precise knowledge of the terrain physical model, robot pose and vehicle true configurations. However, generally only sparse (often occluded) uncertain terrain model estimates can be drawn from sensors such as point-cloud cameras, or vision or 3D lasers whilst operating in realistic unstructured environments. Moreover, localisation in ruggedised 3D environments is significantly more challenging and uncertain than in well-known structured areas. Also, robot configuration may not be accurately known given controller residuals due to noisy feedback data from the robot joints and mechanical play. There is little research that explicitly addresses the challenges of robot stability analysis in uneven terrain while considering a wider range of the parametric uncertainties involved.

In this chapter, a statistical framework for stability prediction using the FA criteria is presented that takes into account a realistic set of the uncertainties that can be expected to be present when planning in unstructured domains. Through an iterative dynamics simulation process, it will be shown how a probabilistic representation of the contact points prediction and the stability margin can be derived. Simulation and experimental results of Packbot multi-tracked robot and local (up to $5\,m.$) terrain data obtained from a ranging camera while traversing over a

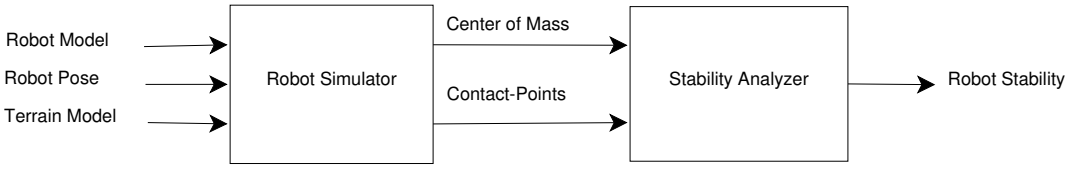


Figure 4.4: High-level pipe-line of the standard robot stability analyser. input: Robot Model (ϕ_a, ϕ_f), Robot Pose (rx, ry, yaw) and 3D Mesh Terrain Model.

ramp and a hill step-field are presented to validate the effectiveness of the proposed statistical approach.

The remainder of this Chapter will first present an overview of the need for an alternative approach when stability calculations cannot be regarded as deterministic in Section 4.3. Section 4.4 will introduce the proposed sampling technique to account for uncertainties, within the context of the generic Monte Carlo random sampling. The robot model, and how it was used in the simulations results as well as in the experimental results over a number of generic terrains will then be described, finishing with concluding remarks.

4.3 The Need for a Probabilistic Approach

4.3.1 Uncertainty in the input data

A block diagram of the traditional robot stability analysis pipeline can be seen in Figure 4.4. If the input parameters can be assumed to be known deterministically, the stability criteria can be computed as described in Section 2.4. In this work we argue there is a need to further introduce a mechanism to account for the uncertainty inherent in the the key parameters that play a role in this analysis, e.g.:

1. Robot Configuration. Variations in ϕ_a and ϕ_f, mainly due to mechanical wear-and-tear and encoder resolution, are assumed to be represented by a normal random distribution with a variance of 3°, as indicated in the technical documentation of the robot.

2. Robot Pose. The noise associated with the robot's 2D pose is dependent on the accuracy of the localisation node. In this work, the study of the stability analysis is restricted to the local terrain in front of the robot, up to $5\,m$ in length. Hence, a conservative variance estimate of 1 cm. per meter in x and y - linearly increasing with a distance of up to $5\,cm$, and 2° in orientation are assumed. This work has not been concerned with the mapping

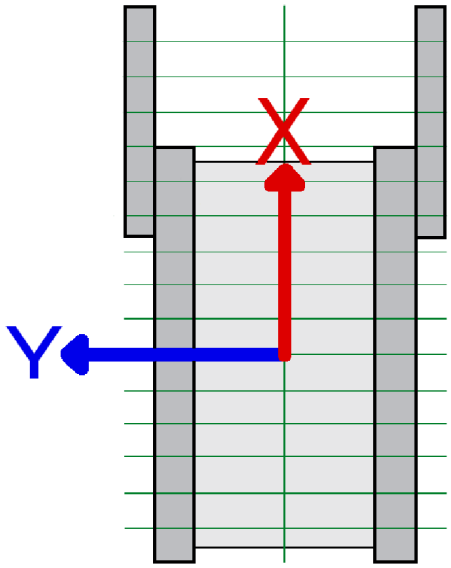


Figure 4.5: The robot frame and (16×2) terrain sections considered in this work in the stability analysis.

or localisation aspects needed to guarantee a complete navigation solution for safe exploration of rough terrains. These are undoubtedly important challenges in themselves, and the effect that map and pose uncertainty will play in the stability prediction is being studied within the framework of this work.

3. Terrain Model. The terrain, shown in Figure 4.10 has been modelled by irregular triangulation [1] of three-dimensional (3D) point-cloud data obtained with an MS-Kinect [42] range camera. The model was built off-line by 3D ICP applied to the depth-image logs of the robot when manually navigating over the terrain at low speeds. The uncertainty values employed in this work are derived from [43], where it has been shown how the MS-Kinect sensor model error in the observed depth measurements increases quadratically with increasing distance, reaching $4\,cm$ at the maximum range of $5\,m$.

   The z components of the vertices that describe the local section of the terrain where the robot may be located will be shifted up/down at the same time to analyse the effect of noise in the terrain model. For tracked vehicles such as the iRobot Packbot, contact

[1] It should be noted that the proposed algorithm would work equally with regular triangulation meshes, height maps or other mesh simplification methods.

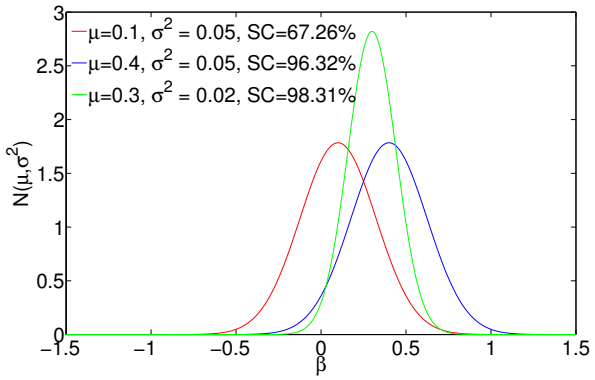


Figure 4.6: Three possible distributions for β and the corresponding values for SC. Although the mean value of the green distribution is smaller than the blue one, it shows a larger SC value which makes it the most certain configuration.

points are almost invariably located under the tracks. Hence, the proposition is for the robot to be divided into sections and the effect of variations in the terrain underneath be considered. Experiments where conducted with various sections until finer selections made little difference in the output of the contact point predictions, settling for two sections in latitude, and 16 sections in the longitude direction as depicted by Figure 4.5.

4.3.2 Probabilistic Stability Margin

As the input parameters are assumed to be represented by a non-deterministic probabilistic distribution, the stability margin would also be a probabilistic variable. In this work β is thus assumed to be a normally distributed random variable with a mean and standard deviation that can be determined based on the statistics of the input parameters. The PDF of a normal distribution $N(\mu, \sigma^2)$ can be formulated as:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{4.1}$$

Assuming a standard normal distribution $N(0,1)$ for β, the cumulative distribution function (CDF) is formulated as:

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \mathrm{e}^{-\frac{t^2}{2}}\, \mathrm{d}t \tag{4.2}$$

This function describes the probability that β will be found at a value less than or equal to x, where $\Phi(-\infty) = 0\%$, $\Phi(0) = 50\%$ and $\Phi(\infty) = 100\%$. For a generic normal distribution

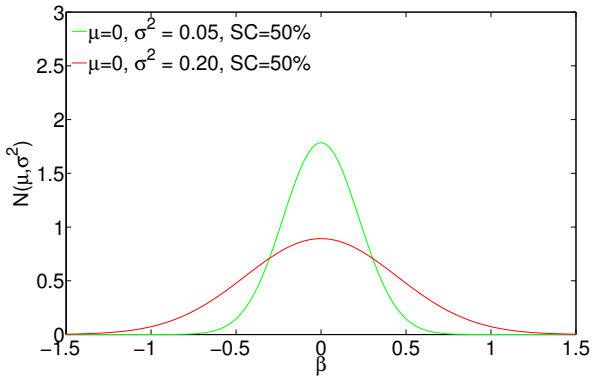


Figure 4.7: Two possible distributions for β and the corresponding values for SC with zero mean and different σs. In this special case of $\mu = 0$, the SC will be multiplied by $(1-\sigma^2)$ to prefer the distribution with smaller σ i.e. the green posture.

$N(\beta_\mu, \beta_\sigma^2)$ for β, the cumulative distribution function can be transformed by

$$F(x, \beta_\mu, \beta_\sigma) = \Phi(\frac{x-\beta_\mu}{\beta_\sigma}) \tag{4.3}$$

Therefore $F(0, \beta_\mu, \beta_\sigma)$ will indicate the probability that β will assume negative values (i.e. a tip-over is in progress). We can now define a "Safety Confidence" margin (SC) to encapsulate our confidence in the stability prediction as

$$SC(\beta) = (1 - F(0, \beta_\mu, \beta_\sigma)) \times 100 \tag{4.4}$$

To intuitively understand the meaning of SC the example in Figure 4.6 is provided. The graph illustrates possible distributions for β, and the corresponding values for SC, based on three different robot postures at a given location on a terrain. Although the mean value of the green distribution is smaller than the blue one, a larger SC value indicates more certainty in this configuration. A conservative fixed large β will unnecessarily push the robot away from many potentially feasible trajectories. On the other hand, critically small safety margins may put the robot in jeopardy, particularly when traversing highly challenging terrains (e.g. stairs or rubble). By employing the proposed SC margin instead, the system can benefit from a dynamic safety boundary that represents reliability in the output predictions.

For the special case when the mean value is exactly zero, the SC calculation would be independent of σ^2 ($SC = 50\%$ always, as illustrated by Figure 4.7). In this case, although both

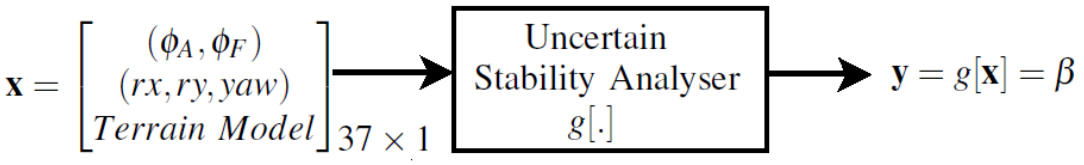


Figure 4.8: High-level pipe-line of the uncertain robot stability analyser. input: Robot Model (ϕ_a,ϕ_f), Robot Pose (rx, ry, yaw) and 3D Mesh Terrain Model.

distributions result in the same value for *SC*, for stability purposes a distribution with smaller σ^2 should be preferred (green curve in this example), indicating that the true β is generally expected to be closer to zero and away from negative tip-over instability. Therefore, for the special case when $\mu = 0$, *SC* will be multiplied by $(1-\sigma^2)$ to lean towards configurations with smaller σs.

4.4 Uncertainty Analysis Method

To state the general problem, we have an n-dimensional vector random variable $\mathbf{x}$ with known mean $\hat{\mathbf{x}}$ and covariance $\mathbf{P}_{xx}$ and would like to predict the mean $\hat{\mathbf{y}}$ and covariance $\mathbf{P}_{yy}$ of a m-dimensional vector random variable $\mathbf{y}$, where $\mathbf{y}$ is related to $\mathbf{x}$ by the non-linear transformation

$$\mathbf{y} = g[\mathbf{x}] \tag{4.5}$$

This is represented by Figure 4.8 in the context of the rescue robot problem presented in this chapter, and described in more detail later in Section 4.5. For the system hereby considered, the arm and flipper angles (ϕ_a, ϕ_f) that determine the posture of the robot, the 3D model of a given terrain and the robot's position on it, constitute the input parameters, i.e. $\mathbf{x}_{37\times1} = (\phi_a, \phi_f, rx, ry, yaw, 32 \times terrain\ sections)$. The output vector includes a list with (up to) four CP, the CM and the FA stability measure, i.e.
$\mathbf{y}_{16\times1} = (4 \times (CP\_x, CP\_y, CP\_z), (CM\_x, CM\_y, CM\_z), \beta)$.

In filtering there are two such transformations: $\mathbf{x}$ could be $\hat{\mathbf{x}}(k|k)$ and $\mathbf{y}$ is $\hat{\mathbf{x}}(k+1|k)$ for predicting state, and $\mathbf{x}$ is $\hat{\mathbf{x}}(k|k)$ and $\mathbf{y}$ is $\hat{\mathbf{z}}(k+1|k)$ for predicting observation. If $g[.]$ is a continuous and differentiable function, it is usually feasible to approximate it by a first-order Taylor series [67] expansion about the point $\hat{\mathbf{x}}$

$$\mathbf{y} = g[\mathbf{x}] \approx g[\hat{\mathbf{x}}] + \mathbf{J}(\mathbf{x} - \hat{\mathbf{x}}) + \dots \tag{4.6}$$

where the ijth element J_{ij} of the Jacobian matrix $\mathbf{J}$ of g is given by

$$J_{ij} = \left.\frac{\partial g_i(\mathbf{x})}{\partial x_j}\right|_{\hat{\mathbf{x}}} = \left.\frac{\partial y_i}{\partial x_j}\right|_{\hat{\mathbf{x}}} \tag{4.7}$$

Now the first order estimation of covariance matrix $\mathbf{P}_{yy}$ can be calculated using general error propagation law [68].

$$\mathbf{P}_{yy} = E((\mathbf{y} - \hat{\mathbf{y}})(\mathbf{y} - \hat{\mathbf{y}})^T) \approx E(\mathbf{J}(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{J}(\mathbf{x} - \hat{\mathbf{x}}))^T) \approx \mathbf{J}\,\mathbf{P}_{xx}\,\mathbf{J}^T \tag{4.8}$$

However, when the transformation is highly non-linear or a deterministic analytical function does not exist, linear approximation methods are not valid and other statistical approaches should be employed. For instance, a well known algorithm for estimating probability density function of a general system's output response from a large set of repeated random inputs is the SMC method. The SMC approximation improves as a greater number of random inputs are used and they are more compatible with the assumed probability distribution. There are different approaches to implement SMC methods, but a typical algorithm is as follows:

a) determine the domain of random inputs and guess a probability distribution over it.

b) select a random set from input vectors according to respective probability distributions.

c) run the simulation or transformation for each vector in the input set and form random set of output.

d) calculate mean and covariance of output set. The results of SMC will be more accurate as the size of the input set increases, and the distribution of the pool is closer to the assumed pattern.

This tendency towards a bigger set of inputs generally makes SMC computationally expensive, so other structured sampling techniques such as Latin hypercube sampling [66] or importance sampling [69] can be used to improve computational efficiency.

4.4.1 Nonlinear transformation of means and covariances

To speed up the transformation of means and covariance, [62] introduces a unscented transform (UT) filter to a smaller set of input data. This technique suggests that, rather than approximate the Taylor series to an arbitrary order, we can approximate the first three moments of the prior distribution accurately using a set of samples. Generally this method superficially resembles the SMC method, yet the samples are not drawn at random. Rather, the samples are deterministically chosen so that they capture specific information about the distribution. This filter is significantly easier to implement because it does not involve any linearisation steps, thereby eliminating the derivation and evaluation of Jacobian matrices.

The n-dimensional random variable $\mathbf{x}$ with mean $\hat{\mathbf{x}}$ and covariance $\mathbf{P}_{xx}$ is approximated by $2n+1$ weighted samples or sigma points selected by the algorithm

$$\begin{aligned} X_0 &= \hat{\mathbf{x}} \\ W_0 &= k/(n+k) \\ X_i &= \hat{\mathbf{x}} + \left(\sqrt{(n+k)\,\mathbf{P}_{xx}}\right)_i \\ W_i &= 1/(2(n+k)) \\ X_{i+n} &= \hat{\mathbf{x}} - \left(\sqrt{(n+k)\,\mathbf{P}_{xx}}\right)_i \\ W_{i+n} &= 1/(2(n+k)) \\ i &= \{1,\ldots,n\} \end{aligned} \tag{4.9}$$

where $\left(\sqrt{(n+k)\,\mathbf{P}_{xx}}\right)_i$ is the ith row or column of the matrix square root of $(n+k)\,\mathbf{P}_{xx}$ and W_i is the weight that is associated with the ith point. $k \in \mathbb{R}$, can be any number (positive or negative) providing that $(n+k) \neq 0$. For Gaussian distribution, a useful heuristic is to select $n+k=3$ to minimise the difference between the moments of the standard Gaussian and the sigma points up to the fourth order [70].

Given the set of samples generated by 4.9, the transformation of the means and covariances procedure is as follows:

1. Each sigma point is instantiated through the process model to yield a set of transformed samples

Algorithm 6 The nonlinear structured unscented transform (UT)

function $ut\_transform(\phi_a, \phi_f, rx, ry, yaw, 32 \times terrain\,sections, k)$
  $\hat{\mathbf{x}} \leftarrow mean(\phi_a, \phi_f, rx, ry, yaw, 32 \times terrain\,sections)$ // 37x1 i.e. n=37
  $\mathbf{P}_{xx} \leftarrow 0$ // 37x37
  $\mathbf{P}_{xx}ii \leftarrow sigma(\phi_a, \phi_f, rx, ry, yaw, 32 \times terrain\,sections)$ // main diagonal
  $X_0 \leftarrow \hat{\mathbf{x}}$
  $W_0 \leftarrow k/(n+k)$
  for $i = 1 \rightarrow n$ **do**
    $X_i \leftarrow \hat{\mathbf{x}} + \left(\sqrt{(n+k)\,\mathbf{P}_{xx}}\right)_i$
    $W_i \leftarrow 1/(2(n+k))$
    $X_{i+n} \leftarrow \hat{\mathbf{x}} - \left(\sqrt{(n+k)\,\mathbf{P}_{xx}}\right)_i$
    $W_{i+n} \leftarrow 1/(2(n+k))$
  end for
  for $i = 0 \rightarrow 2n$ **do**
    $(\mathbf{CP}, \mathbf{CM}) \leftarrow ode\_simulate(X_i)$
    $\beta \leftarrow FA(\mathbf{CP}, \mathbf{CM})$
    $Y_i \leftarrow (\mathbf{CP}, \mathbf{CM}, \beta)$ // 16x1 i.e. 4x(CP_x,CP_y,CP_z)+(CM_x,CM_y,CM_z)+β
  end for
  $\hat{\mathbf{y}} \leftarrow \sum_{i=0}^{2n} W_i Y_i$ // 16x1
  $\mathbf{P}_{yy} \leftarrow \sum_{i=0}^{2n} W_i \{Y_i - \hat{\mathbf{y}}\} \times \{Y_i - \hat{\mathbf{y}}\}^T$ // 16x16
  return $(\hat{\mathbf{y}}, \mathbf{P}_{yy})$
end function

$$Y_i = g[X_i], \quad i = \{0, \ldots, 2n\} \tag{4.10}$$

2. The transferred mean is computed as

$$\hat{\mathbf{y}} = \sum_{i=0}^{2n} W_i Y_i \tag{4.11}$$

3. The transferred covariance is computed as

$$\mathbf{P}_{yy} = \sum_{i=0}^{2n} W_i \{Y_i - \hat{\mathbf{y}}\} \times \{Y_i - \hat{\mathbf{y}}\}^T \tag{4.12}$$

The mean and covariance are calculated using standard vector and matrix operations, which means that the algorithm is suitable for any choice of process model, and implementation is convenient because it is not necessary to evaluate the Jacobian matrixes, as in e.g. an Extended Kalman Filter [71]. The Algorithm 6 summarises this procedure.

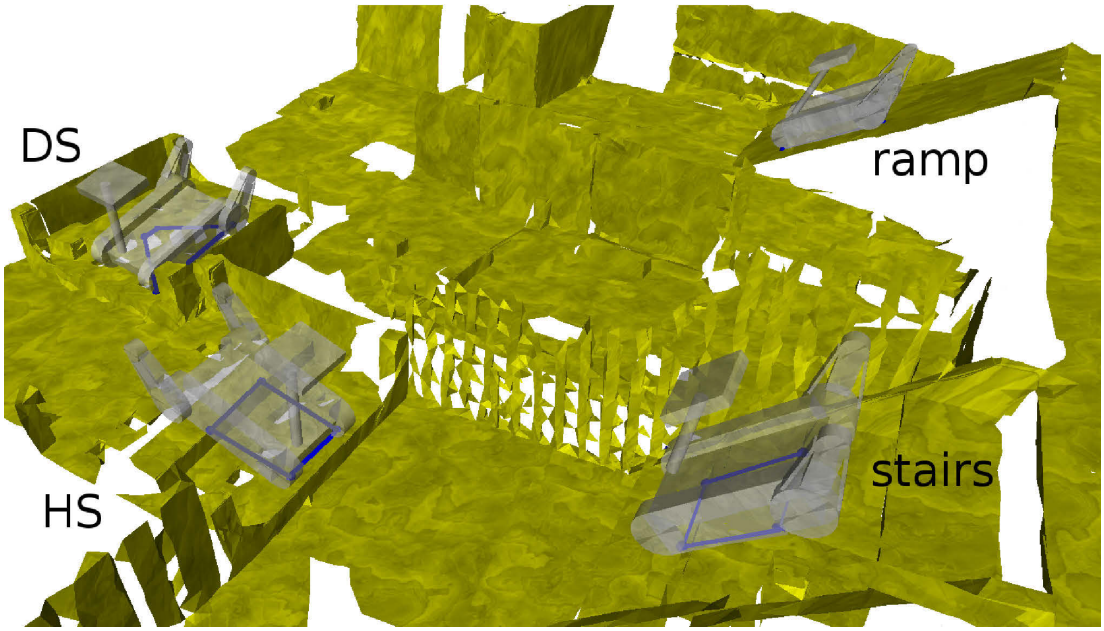
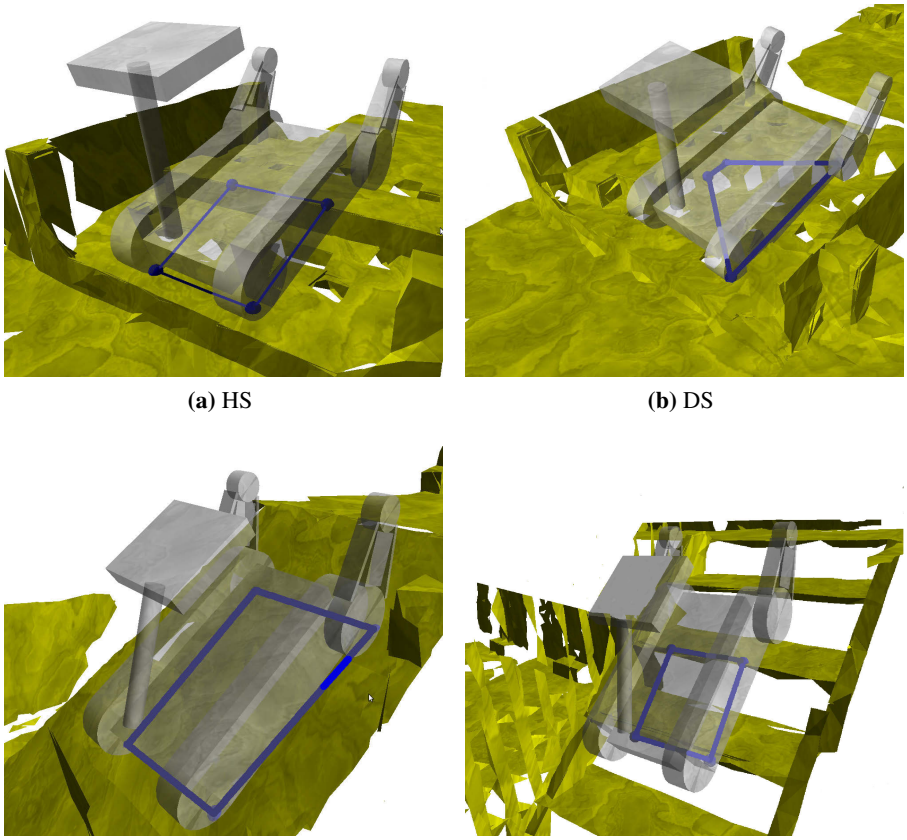


Figure 4.9: Whole 3D USAR test arena and robot model at four locations (HS, DS, ramp and stairs).

It is important to note that while only the FA distributions $(\beta_\mu, \beta_\sigma)$ are exploited for path planning purposes in this work, the output vector $\mathbf{y}$ also provides probabilistic information about the robot's CP and CM. It is envisaged that it may well be possible to take advantage of these useful statistics in other stability margins, or for other purposes (e.g. computer graphics rendering applications).

4.5 Simulation Results

The simulation aims to estimate the distributions of contact points and stability measure β under the assumption of a Gaussian model distribution of uncertainty in the robot configuration, environment geometry as well as the position of the robot over the terrain. The distribution of the contact points is not really necessary for the uncertainty analysis, yet it is provided here to quantify the validity of the UT calculations. The result of SMC is compared with the more efficient structured sampling approach UT explained in Section 4.4.1. The dimension of $\mathbf{x}$ in Equation 4.9 is considered as $n = 3 + 2 + 2 \times 16 = 37$, i.e. 3 degrees of freedom for the robot's 2D position (rx, ry, yaw), 2 for the robot's configuration (ϕ_a, ϕ_f) and 16 for the number



(a) HS **(b)** DS

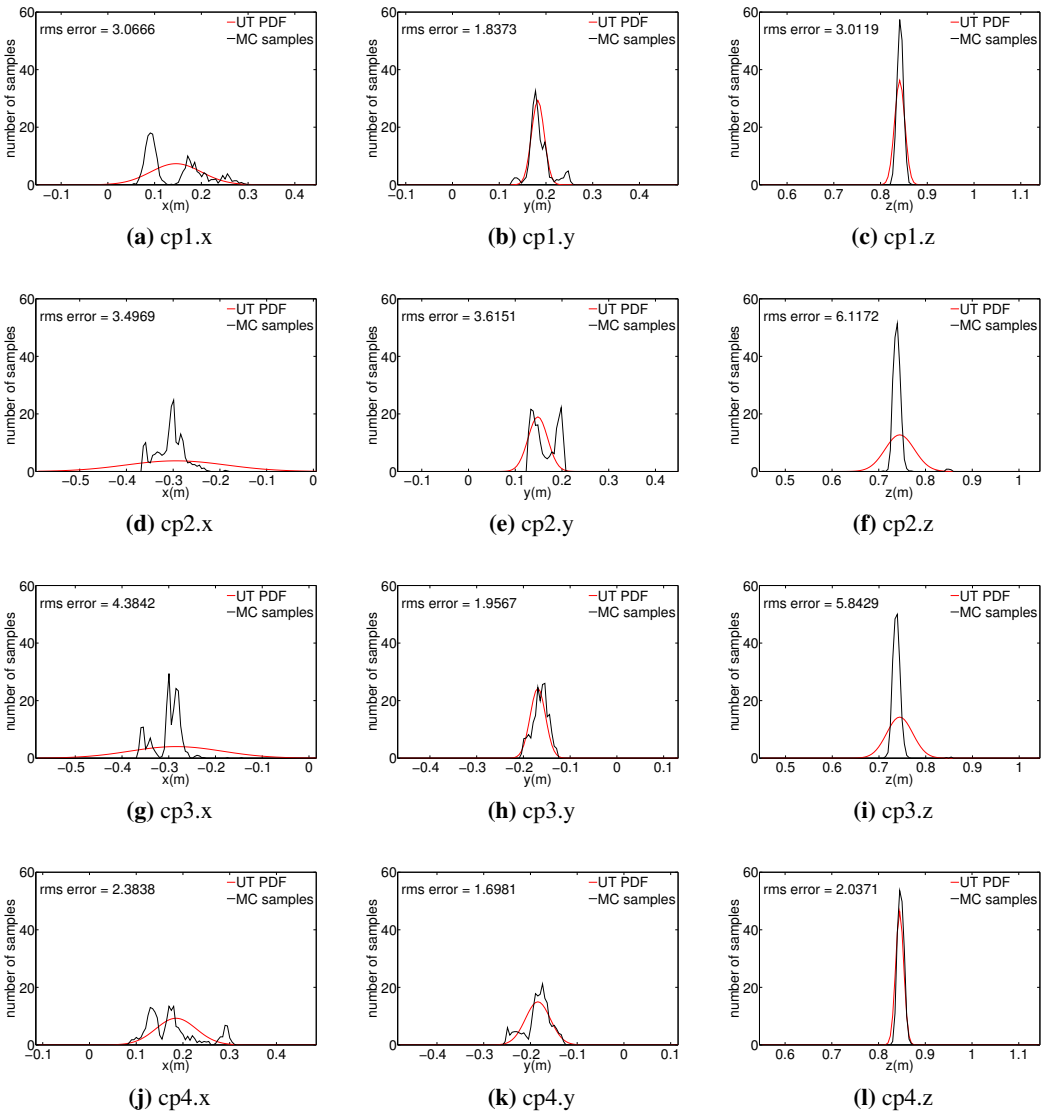
(c) ramp **(d)** stairs

Figure 4.10: Detail of robot model configurations and support polygons at the four different topologies in the map (HS, DS, ramp and stairs).

of sections in longitude direction as shown in Figure 4.5.

The robot was placed in four different terrain topologies including a ramp, stairs and two reconfigurable wooden cubic obstacles arranged in diagonal step-field (DS) and hill step-field (HS) configurations. The 3D model of the USAR test arena and the four robot positions for this experiment are shown in Figure 4.9 and 4.10 (detail). In each position, the uncertainty analysis of contact points and β are first carried out with the well known SMC technique with 1000 iterations, while UT requires only 75 iterations. The distribution of contact points over HS, DS, ramp and stairs are illustrated in Figure 4.11, 4.12, 4.13 and 4.14 respectively.

The distribution of βs in these four different positions are shown in Figure 4.15. The "ideal" PDF using the resulting UT mean and σ values are plotted in black. The original SMC



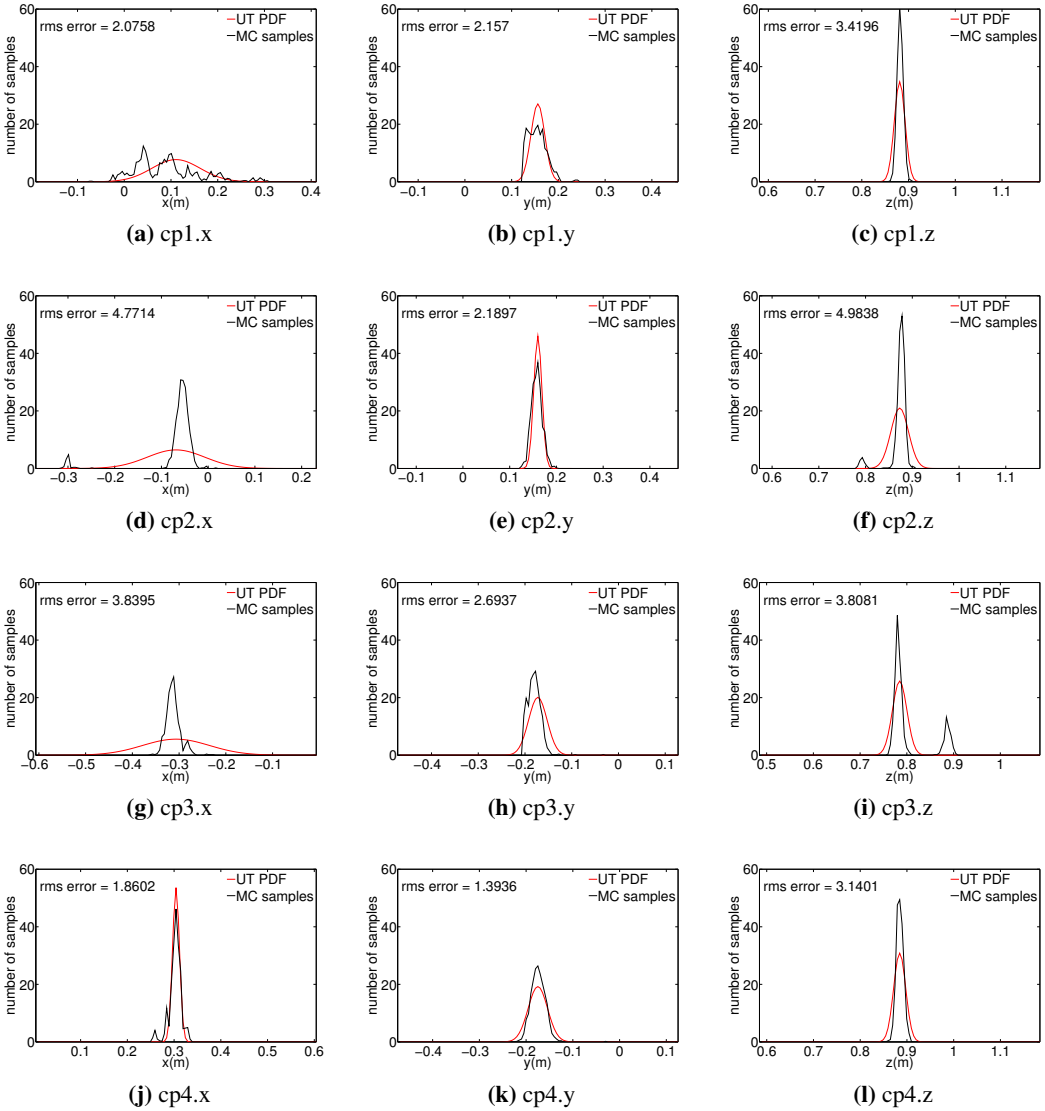
(a) cp1.x **(b)** cp1.y **(c)** cp1.z

(d) cp2.x **(e)** cp2.y **(f)** cp2.z

(g) cp3.x **(h)** cp3.y **(i)** cp3.z

(j) cp4.x **(k)** cp4.y **(l)** cp4.z

Figure 4.11: The distribution of contact points over HS (x and y in robot frame, z in global frame), $(k = 1)$.



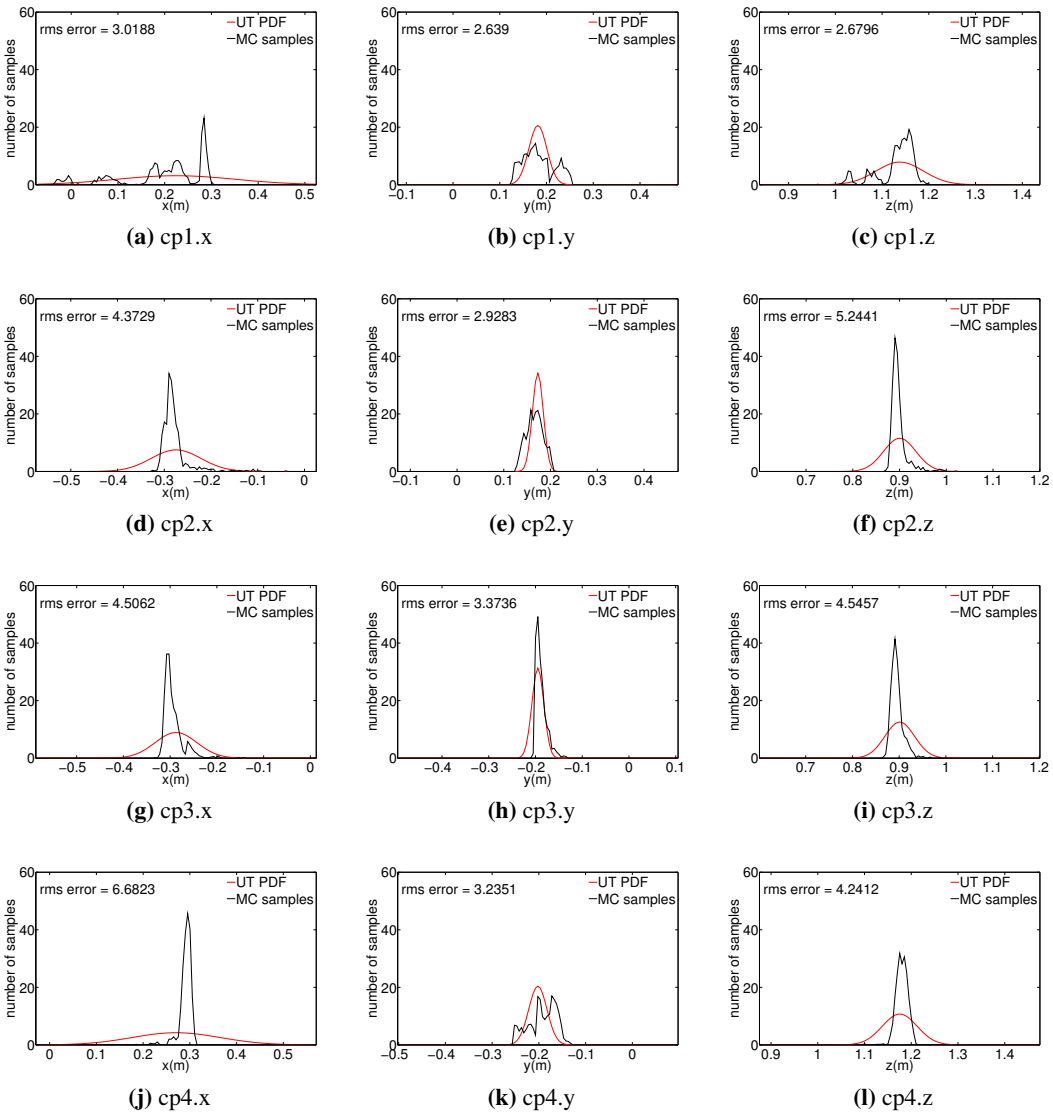
(a) cp1.x **(b)** cp1.y **(c)** cp1.z

(d) cp2.x **(e)** cp2.y **(f)** cp2.z

(g) cp3.x **(h)** cp3.y **(i)** cp3.z

(j) cp4.x **(k)** cp4.y **(l)** cp4.z

Figure 4.12: The distribution of contact points over DS (x and y in robot frame, z in global frame), $(k = 1)$.



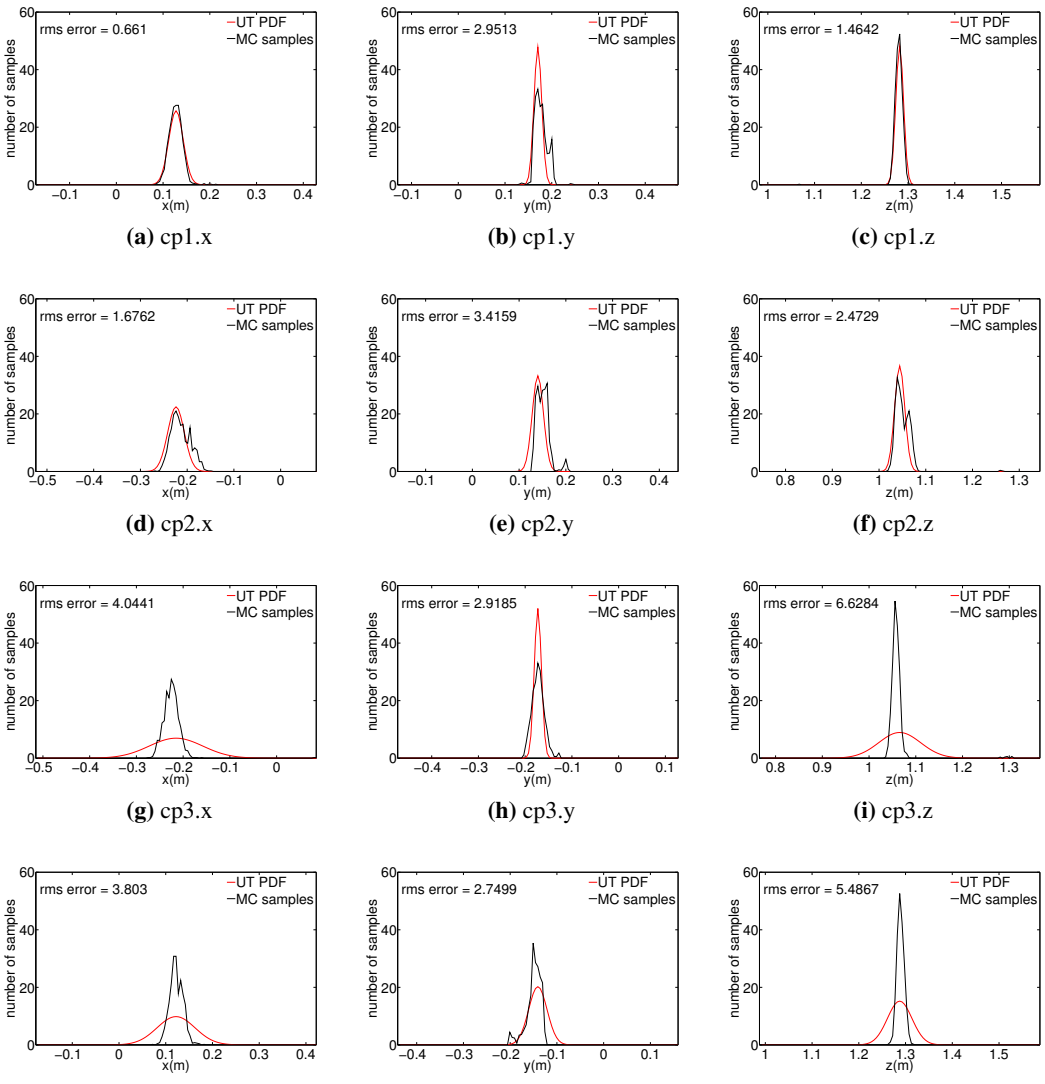
(a) cp1.x **(b)** cp1.y **(c)** cp1.z

(d) cp2.x **(e)** cp2.y **(f)** cp2.z

(g) cp3.x **(h)** cp3.y **(i)** cp3.z

(j) cp4.x **(k)** cp4.y **(l)** cp4.z

Figure 4.13: The distribution of contact points over ramp (x and y in robot frame, z in global frame), $(k = 1)$.



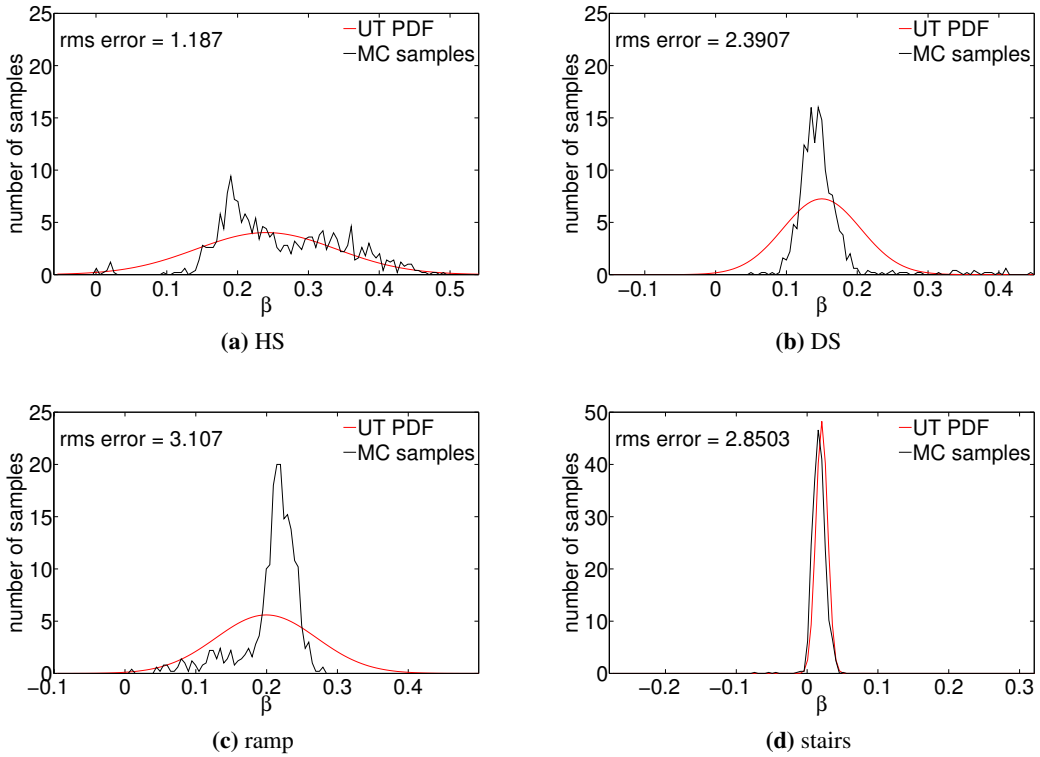
(a) cp1.x **(b)** cp1.y **(c)** cp1.z

(d) cp2.x **(e)** cp2.y **(f)** cp2.z

(g) cp3.x **(h)** cp3.y **(i)** cp3.z

(j) cp4.x **(k)** cp4.y **(l)** cp4.z

Figure 4.14: The distribution of contact points over stairs (x and y in robot frame, z in global frame), $(k = 1)$.



(a) HS **(b)** DS **(c)** ramp **(d)** stairs

Figure 4.15: The distribution of βs in different positions $(k = 1)$

samples are plotted in dashed blue around this PDF for comparison purposes.

The mean and covariance were first extracted using the UT method, with $k = 3 - n = -34$ as suggested in [70]. When k is negative, it is possible that the predicted covariance will not be positive semi-definite. In such cases, the resulting PDFs don't closely follow the SMC samples, and trial-and-error is recommended to evaluate the covariance about the projected mean [70]. As suggested, simulations were repeated to find a more suitable value for k, with a positive value of $k = 1$ thereby significantly improving the rms error over a range of values. The average rms error of each contact point and β in the four positions is summarised in Table 4.1 for two values of $k = 1$ and $k = -34$.

	cp1	cp2	cp3	cp4	CM	β	mean
HS	2.639	4.410	4.061	2.040	1.888	1.127	2.694
DS	2.342	3.826	3.148	2.056	1.763	2.331	2.577
ramp	2.705	4.006	3.790	4.559	1.455	3.004	3.253
stairs	1.582	2.464	4.475	3.828	0.710	2.600	2.610

(a) $k = 1$, average = 2.7080%.

	cp1	cp2	cp3	cp4	CM	β	mean
HS	3.884	5.310	3.918	3.543	5.504	2.743	4.150
DS	4.442	5.332	5.004	1.905	2.697	2.918	3.716
ramp	3.296	3.084	4.901	5.284	3.518	3.013	3.849
stairs	2.191	2.628	1.419	3.110	5.911	3.830	3.181

(b) $k = -34$, average = 3.7244%.

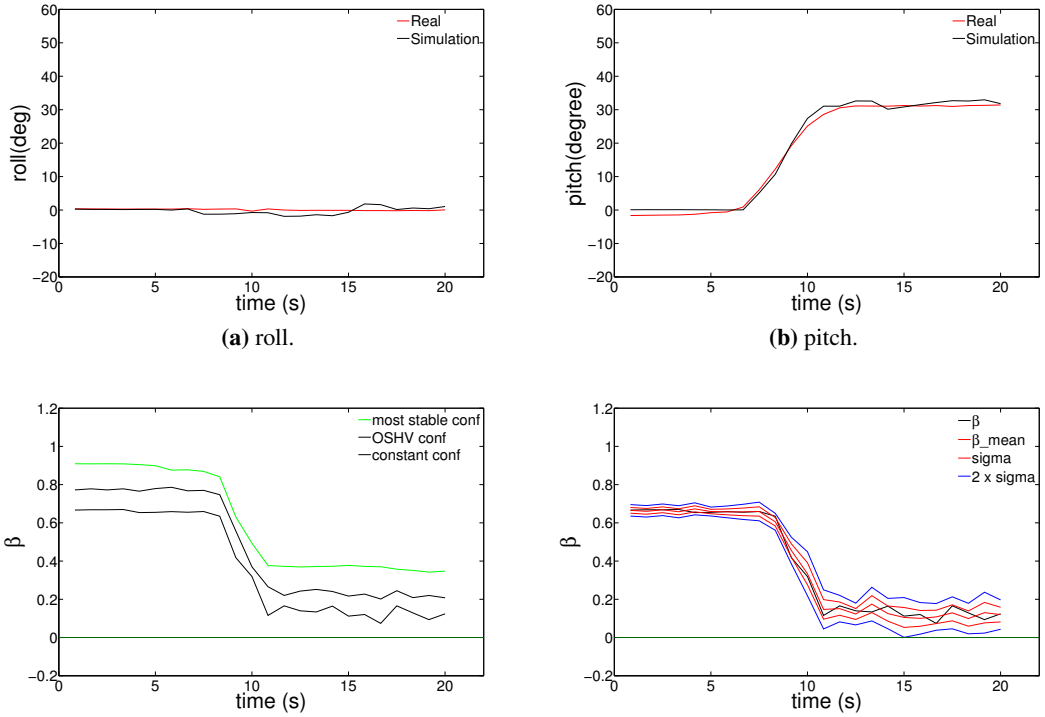
Table 4.1: The rms errors(%) between UT and Monte Carlo samples.

4.6 Experimental Results

To validate the results the robot was made to traverse over the actual ramp and HS following a straight trajectory and constant reduced speed. A localiser running of odometry and 2D range data from an auto-levelled laser scanner was used to derive an estimate of the robot pose (rx, ry, yaw) with a previously built 3D mesh of the arena, depicted in Figure 4.9. As the platform has got no suspension and the terrain is rigid, *pitch* and *roll* measurements from an on-board IMU can be assumed to be a veracious reflection of the vehicle's attitude when sitting on the terrain. The robot's pose (ϕ_a, ϕ_f) was recorded from the actual on-board encoders during the experiments. The data from these tests was then analysed off-line to calculate the statistical properties of contact points and stability measures.

The inclination of the ramp illustrated in Figure 4.10 is 30 degrees. The result of the ramp experiment is illustrated in Figure 4.16. As shown in Figure 4.16a and 4.16b, real inclination data is very close to that inferred by the simulator. The stability measure from a single simulation and mean value driven using UT in each point is depicted in black and red in Figure 4.16d respectively. Also the standard deviation σ (68%) and $2 \times \sigma$ (95%) around the mean are depicted in dashed red and blue. The measured β and its mean value up to σ is always positive, which shows a convenient stability.

The patterns of β acquired by three different configuration planning strategies along the same straight trajectory is illustrated in Figure 4.16c. The solid black line is equal to the β in Figure 4.16d and it is achieved while deriving the robot with a fairly constant configuration



(a) roll.

(b) pitch.

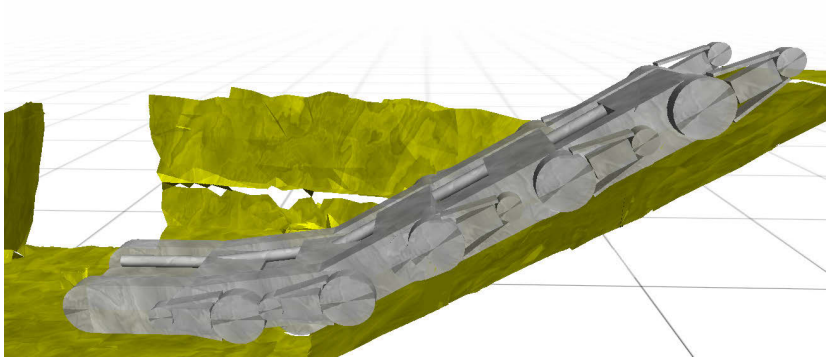
(c) Comparison of βs for different configurations.

(d) β.

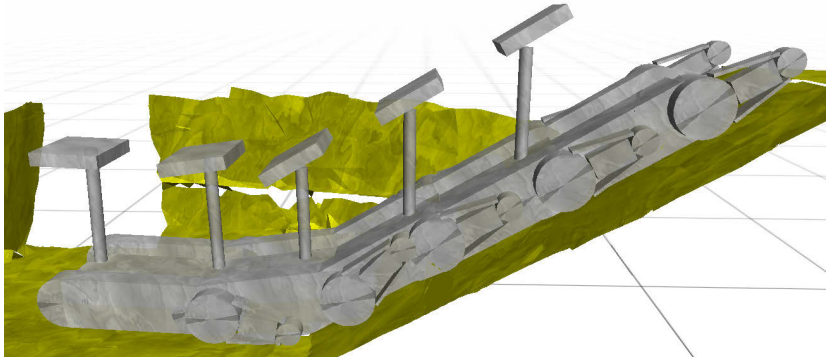
Figure 4.16: Experimental results over ramp.

$(\phi_a = 90^\circ, \phi_f = 45^\circ)$ and simulating the robot with recorded configuration and position over the 3D model of the terrain. For comparison purposes, the stability measures of the OSHV planner (described in Section 3.3.3) with $\beta_{min} = 0.2$ and the most stable (providing the highest SC) configurations, are depicted in dashed black and green respectively. In this ramp case, the β of the OSHV posture lies between the constant and the most stable stability margin. For safer posture trajectory the safety stability margin, β_{min} should be increased which will shift up the dashed black plot. The minimum value of β in the most stable plot is around 0.4, hence if the minimum β in the planning was set to a value larger than this, the ramp trajectory would be regarded as unstable.

A side view of the path with the robot arrangements suggested by both planners are depicted in Figure 4.17 - omitted in some places to increase clarity. Comparing the results in the beginning of the ramp in Figure 4.17a and 4.17b shows that planning purely based on the stability margin has resulted in sudden big changes for the flippers, while the OSHV planner



(a) Most stable configurations.



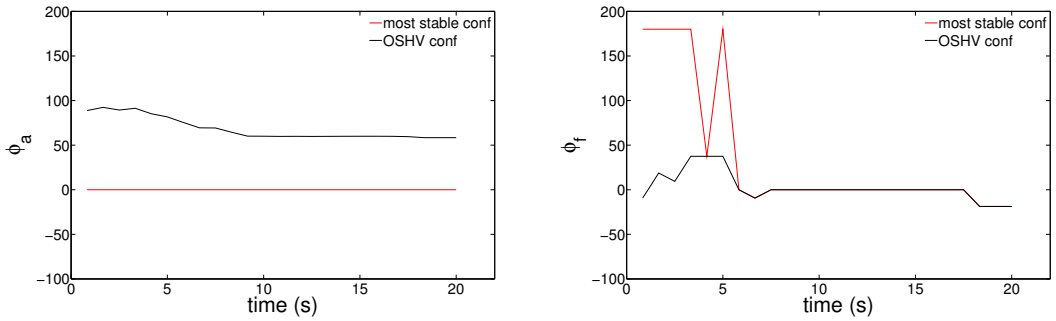
(b) OSHV configurations.

Figure 4.17: The side view of the robot configurations along the ramp (direction: left to right).

produced a soft and continuous kinematic trajectory. The corresponding posture changes for the arm and flippers joints are illustrated in Figure 4.18.

HS is an example to simulate common block obstacles, like rubble or unlevelled floors. The HS set-up illustrated in Figure 4.10 is composed of three successive $10\,cm$ steps: two traversed "up", and one "down". The results of the experiment over the HS is illustrated in Figure 4.19 in the same way as was earlier depicted for the ramp. As can be seen in Figure 4.19a and 4.19b, the real inclination data is also closely captured by the simulator except at around $8\,s$ and $17\,s$, when the robot tipped-over and had to be manually handled and returned to the HS to prevent a fatal crash. Although the calculated mean value for β can be seen to be just positive over the path at those instances, σ uncertainty analysis shows the robot tipping-over at those instances (when the crossing over the steps takes place).

Comparing these two examples shows that, despite the smaller inclination in the HS configuration, the robot is still more stable over the ramp than HS. Assuming that a fixed supporting-



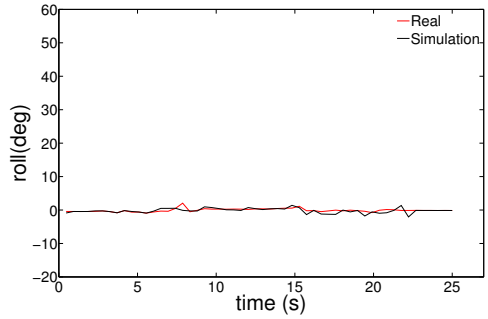
(a) arm. **(b)** flippers.

Figure 4.18: The corresponding posture changes for the arm and flippers in the results of the Figure 4.17 over the ramp.

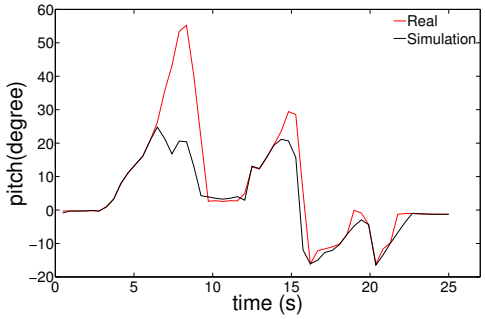
polygon and calculations of stability based on IMU data (like the approach in [16]) will lead to apparent stability, yet that is not the case. The traditional deterministic stability analysis method with variable supporting-polygon can be regarded as fairly reliable over simple topologies like ramps, but can't predict instability over more challenging obstacles like HS where the uncertainty in the input parameters can have a significant influence on the output stability margin.

Another interesting observation can be seen in Figure 4.15d, where on the stairs β mean is only marginally stable, yet in the real experiments the robot was more often than not tipping over, vindicating once again the need to plan for the margins of the stability that this work advocates for.

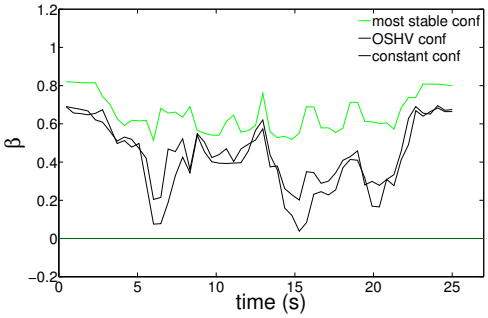
In the same way, the patterns of β acquired by three different configuration planning strategies along the same straight trajectory are illustrated in Figure 4.19c. The solid black line is equal to the β in Figure 4.19d and it is achieved while deriving the robot with a constant configuration $(\phi_a = 90^\circ, \phi_f = 45^\circ)$ and simulating the robot with recorded configuration and position over the 3D model of the terrain. For comparison purposes, the stability measures of the OSHV planner with $\beta_{min} = 0.2$ and the most stable configurations are depicted in dashed black and green respectively. It can be observed how for the OSHV posture β is always smaller than the most stable stability margin. It can moreover be seen how in some places it is also smaller than the constant configuration's stability margin, as in that case there is no accounting for the additional visibility constraints in the robot pose. Thus in contrast to ramp traversing, at some places the constant configuration ends up marginally more stable than the calculated



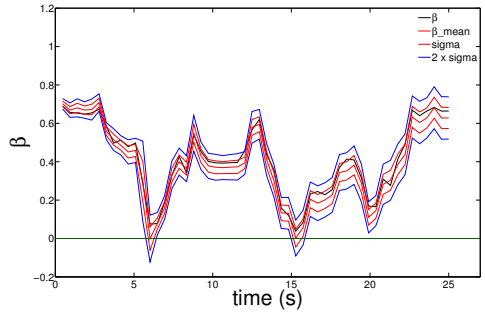
(a) roll.



(b) pitch.



(c) Comparison of βs for different configurations.

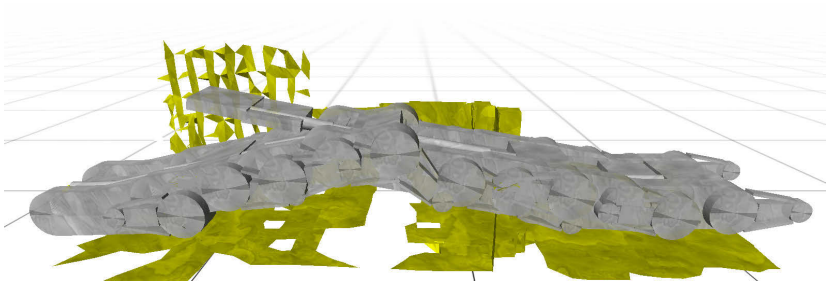


(d) β.

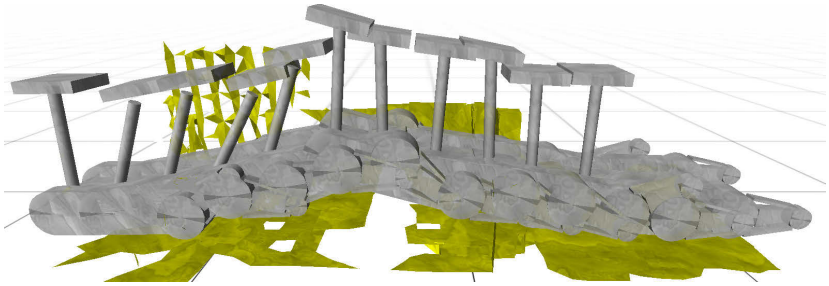
Figure 4.19: Experimental results over hill step-field

OSHV posture. Of course, for trajectories where increased safety posture is desired, β_{min} can be increased, effectively shifting the dashed black plot up so that it is always above the constant posture.

A side view of the path with the robot arrangements suggested by both planners is depicted in Figure 4.20 - omitted in some places to increase clarity. Comparing the results in the beginning, middle and the end of the HS in Figure 4.20a and 4.20b shows that planning purely based on the stability margin has resulted in sudden big changes for the flippers and arm while the OSHV planner produced a soft and continuous kinematic trajectory. The corresponding posture changes for the arm and flippers joints are illustrated in Figure 4.21.

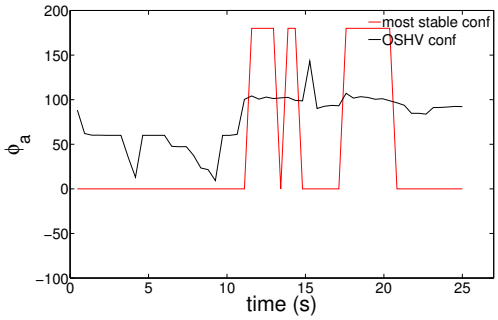


(a) Most stable configurations.

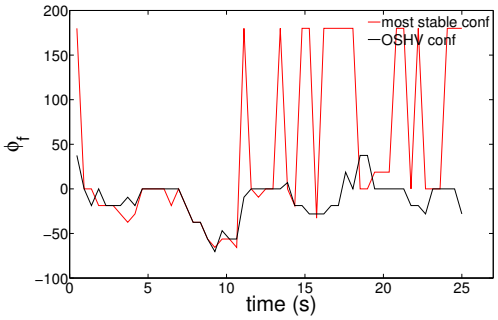


(b) OSHV configurations.

Figure 4.20: The side view of the robot configurations along the HS (direction: left to right).



(a) arm.



(b) flippers.

Figure 4.21: The corresponding posture changes for the arm and flippers in the results of the Figure 4.20 over the HS.

4.7 Summary

Mobile robots are frequently employed in ruggedised environments where stability prediction is critical to the success of the mission. The most influential tip-over stability measures are based on two criteria, the CM and the SP. This chapter argues that, in uneven terrain, significant uncertainties are associated with the robot model, localisation and terrain parameters and these need to be taken into account. A statistical method for stability prediction using the well known FA stability measure has been presented which can take into account these uncertainties. Probability distribution of contact points and CM and FA stability measure have been numerically estimated, with simulation results from the uncertain parameters performed on the ODE. The proposed method has been validated using two techniques: a conventional standard Monte Carlo scheme, and a structured sampling approach, which shows significant improvement in computational efficiency. Experimental results of a multi-tracked robot fitted with flippers, a manipulator arm and a range camera travelling over a ramp and a hill step-field have been presented that confirm the utility of the proposed statistical stability prediction method. A novel probabilistic stability criterion derived from the cumulative distribution of a tip-over margin is introduced that allows a safety constraint to be dynamically updated by available sensor data as it becomes available. The proposed safety constraint authorizes the planner to generates more conservative motion plans for areas with higher levels of uncertainty, while avoids unnecessary caution in well-known areas. A probabilistic approach to the problem of global path planning under stability uncertainty within the context of the work described here is provided in the next chapter, where the emphasis has been mainly placed on paths that guarantee stability in a practical dynamic stability setting.

5

Path Planning with Stability Uncertainty

5.1 Introduction

This chapter looks at the challenging problem of global path planning over ruggedised terrains by formally accounting for stability uncertainty in the process. Based on the findings of the statistical stability analysis technique described in Chapter 4, this chapter proposes to further combine it with the deterministic stable path planning strategy described in Chapter 3, proven to be particularly suitable for search and rescue missions, with the goal of improving robot navigation safety in scenarios where the model of the system and the sensory data available to the robot may be imperfect.

The stability cost/constraint can be considered in two different ways: as a deterministic cost with a fixed stability boundary based on an ideal system model, and as a probabilistic cost with a dynamic safety margin based on the mean and covariance of the measure. This is done by exploiting the proposed probabilistic safety confidence (SC) introduced in Equation 4.4, which is dynamically updated by available sensor data when the mobile robot moves around.

As will be shown, the main drawback of employing deterministic constant stability margins to path planning is that whilst producing safer paths with larger, more conservative stability margins, they may also easily end up being overly restrictive, and filtering out many probable pathways. On the other hand, planning on the boundary of tip-over could easily jeopardise stability if uncertainties are present. The proposed probabilistic approach allows to search paths with a minimum "safety confidence" instead, so that model uncertainties can be taken

into consideration when finding paths, instead of resorting to restrictive fixed minimum safety margins. Moreover, while in Chapter 3 the mechanisms where provided to exploit stability both as a constraint and also as an added cost to the A* search optimization process, in the overall path planning strategy proposed here we take the stand that simply using it as a constraint is appropriate to guarantee paths that are "confidently" stable. In essence we are advocating for the fact that so long as we are confident the final path found will be stable, it is less relevant whether another one might be slightly more stable, as that's ultimately less relevant to the final outcomes in a realistic setting, and we suggest not spend computational resources in doing that.

The advantages of planning with probabilistic stability will be demonstrated using a grid based A* algorithm as well as a sampling based RRT planner with a model of the Packbot robot shown in Figure 1.1, through comprehensive simulations in the USAR and UTIAS arenas.

5.2 Related Work

The demand for autonomous robots in industry and field application is increasing with the technological advances in modern sensors, actuators, hardware and software facilities which make employing of robotics technology economical and feasible. In field application, mobile robots are required to operate fully or semi-autonomously in harsh, unstructured environments such as agriculture [72], mining [73, 74, 75], planetary exploration [76] and search and rescue [77] missions for example. One of the most difficult problems of navigation over unstructured and unforgiving environments is how to address the uncertainties emanating from imperfect actuators and poor environmental sensor information.

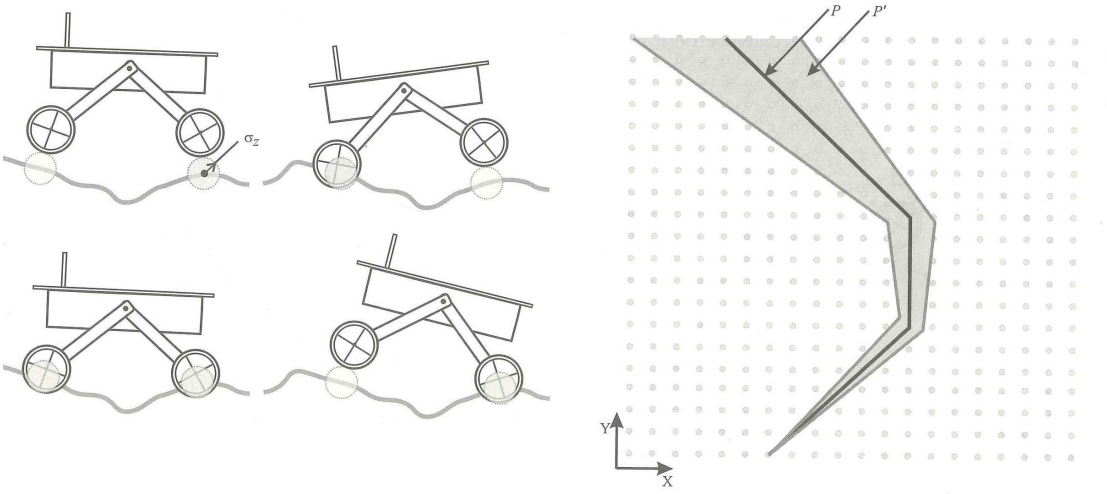
Although uncertainty is often overlooked in classical motion planning techniques [78, 79], more recent search optimization techniques have investigated different approaches to take into account imperfect robot motion or sensing models [80]. One of the well studied approaches developed in the literature to explicitly deal with uncertainties in the input data and system model parameters is the partially observable Markov decision process (POMDP) [81, 82, 83]. For example a POMDP model for finding belief-feedback policies for a team of robots cooperating to extinguish a spreading fire is presented in [84]. The proposed planning algorithm is able to employ user-supplied domain knowledge for the synthesis of information feedback policies. In general the state-space in these domains grows very large and the well documented "curse of dimensionality" is a factor that still hampers progress of these techniques in realistic

settings, making this a very active area of research in the robotics and AI community. Linear-quadratic Gaussian motion planning (LQG-MP) strategies have also been developed which are able to take into account the motion and sensing uncertainty [85]. Assuming a Gaussian model of uncertainty and having a linear-quadratic controller, the LQG-MP method aims to characterise a-priori probability distributions of the state of the robot in advance. The performance of LQG-MP is studied using simulation experiments where the rapidly exploring random tree (RRT) [86] is employed to generate the candidate paths.

Motion planning in dynamic uncertain environments is another challenge for mobile robots operating in close proximity with many other moving agents; e.g. a service robot acting as a waiter in a restaurant, or mobile robots in exhibitions and trade fairs. In this case, the future evolution and uncertainties of the states of the moving agents and obstacles needs to be addressed as well. A strategy to account for future information gathering in the planning in dynamic, uncertain environments is presented in [87]. The uncertainty in location of the robot and obstacles is considered using a partially closed-loop receding horizon control algorithm that is able to integrate the prediction, estimation, and planning and approximately solve the stochastic dynamic programming problem.

The path following with uncertainty has also been studied by the control community. A Kalman-based active observer controller for the path following of wheeled mobile robots subject to non-holonomic constraints is presented in [88]. The effect of external disturbances, general model errors, and uncertainties present in the system are reduced by adding an extra state (the "active state") to the controller design. The effectiveness of the proposed path-following controller is evaluated via simulation results for a wheelchair robot following a straight line and a circular path. More recently, a path following controller design approach for articulated manipulators based on transverse feedback linearisation is presented in [89]. The Lyapunov redesign [90] method is employed to make the proposed controller robust against modelling uncertainty. Experimental results of a four DoF manipulator with a combination of revolute and linear actuated links are provided where the end-effector was set to move in a circular path.

In general, uncertainty in a system can be represented as a stochastic process in two ways: non-deterministic (a boundary is assumed for uncertainties), and probabilistic (the uncertainties are described using probability distributions) [87]. While we have advocated for the benefits that a a probabilistic formulation brings to the work presented in this thesis. other authors have looked at the problem of non-deterministic incorporation of uncertainty at the planning stage, e.g. by considering variations in the 2.5D terrain elevation data and localisation errors, e.g. as



(a) A set of potential worst-case robot configurations to address the terrain measurement uncertainty.

(b) The potential path P and the superset P' to address the localisation uncertainty.

Figure 5.1: A non-deterministic method for incorporating the uncertainty in robot path following a 2.5D terrain elevation data into a path planning algorithm for an articulated wheeled mobile robot model [63].

described in [63] for an articulated wheeled mobile robot. The original FA measure [36] was employed to evaluate the stability of the rover in the elevation map. Consequently as explained in Section 2.4, the position of CM and the ground contact points would be the essential inputs to calculate the safety margin. The contact points are assumed to be under the wheels and are calculated based on the robot's kinematic and its position over the elevation map. A conservative path planning approach is adopted that considers terrain measurement uncertainty, where a set of potential worst-case robot configurations at boundary locations in the terrain are examined to make sure that the vehicle would remain stable for a given arbitrary fixed variance in the elevation map, as illustrated in Figure 5.1. If any posture in this set is proven unstable, the corresponding location in the map will be regarded as untraversable. To address the localisation uncertainty for a given path, all points along the path within a distance proportional to the assumed robot localisation uncertainty are examined given all possible configurations. A point in the terrain would be considered as a feasible point for path finding purposes only if all configurations in the overall search have been proven to be stable. The output of this brute-force approach is a simple failure or success, with no concern for the probability of a tip-over instability.

While in Chapter 3 the benefits of high-visibility stable paths for rescue operations have

Algorithm 7 The statistical stable A* planner algorithm

```
1:  closed ← ∅
2:  open ← cell(start)
3:  while (open ≠ ∅) do
4:      cell(i) = min(open)
5:      closed ← closed + cell(i)
6:      open ← open − cell(i)
7:      for all cell(j) ∈ {8 successors of cell(i)} do
8:          if (cell(j) ∉ closed & cell(j) ≠ obstacle) then
9:              ut_transform()
10:             if (SC > SC_min) then
11:                 if (cell(j) ∈ open) then
12:                     refresh_node(i, j)
13:                 else
14:                     add_open_node(i, j)
15:                 end if
16:             end if
17:         end if
18:     end for
19:     if (cell(goal) = min(open)) then
20:         return path
21:     end if
22: end while
23: return path = ∅
```

been heralded, the fundamental motivation behind the proposal in this paper is the ability to identify safe paths that can also accommodate uncertainty in the stability of the proposed manoeuvres in a probabilistic manner, with the ultimate goal of making it viable for a real implementation whilst considering imperfect sensors and noisy data. To this end, this study has proposed the probabilistic stability measure *SC* in Equation 4.4 based on the cumulative distribution of the FA measure which indicates the probability that β will be found to be positive. The *SC* criterion can be exploited as the objective cost function for planning purposes as well as a stability constraint without contributing to the accumulated cost. As mentioned at the introduction of the Chapter, in this work *SC* is employed as a constraint to guarantee that the path will be stable at any points along the path subject to a minimum *SC_min* percentage of confidence, and we will not introduce this in the search optimization process, as previously done in 3.11 when we explained the A* search methodology with a deterministic β. In this Chapter we therefore concentrate efforts in the search for a path that will be confidently stable

independently of whether others could be marginally more so, focusing the optimization search process in confidently stable paths exhibiting highest possible field of view configurations.

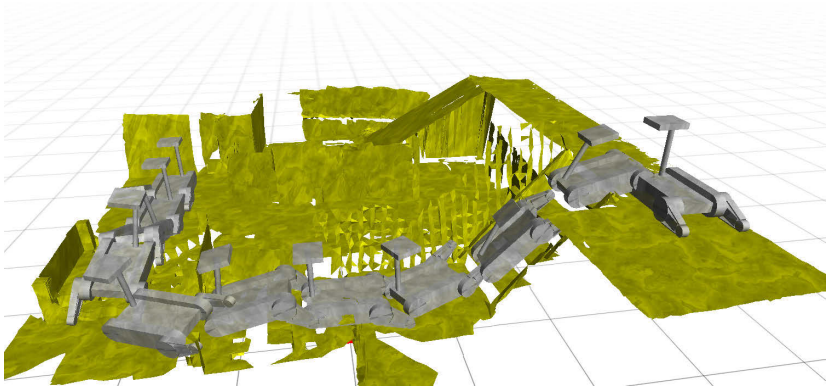
The following sub-section illustrates the implementations with grid based A* algorithm. The integration of the proposed strategy in a sampling-based RRT planner will also be presented in Section 5.4 for completeness. As before, the effectiveness of the approach has been evaluated using two challenging USAR and UTIAS terrain data sets, and then compared to the deterministic stable path planning described in Chapter 3. The USAR test arena is chosen to investigate the performance of the technique in an indoor setting with distinctive features such as stairs, rubble etc., whereas the UTIAS arena is an example of a larger outdoor scenario. In both instances, the robot is expected to come up with configurations aimed at keeping the arm as high as possible to achieve the best possible field of view whilst satisfying the constraints imposed by the corresponding algorithms (β_{min} or SC_{min}).

5.3 Implementation with A* planner

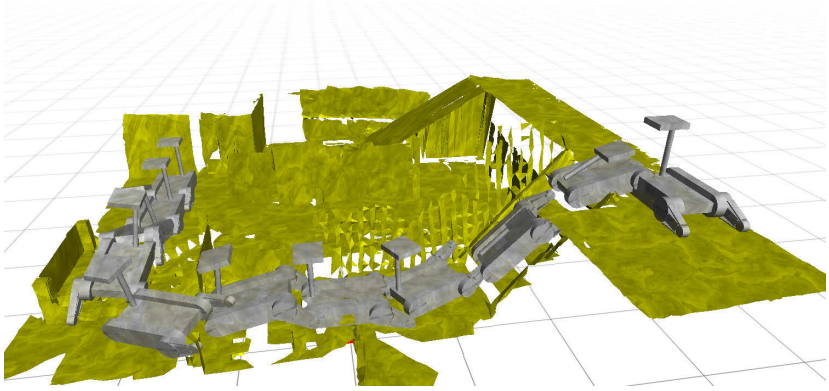
For comparison purposes, lets first briefly review the deterministic planning approach which was introduced in Chapter 3 as summarised in Algorithm 5. The key contribution on this algorithm was the introduction of a stability constraint to a cost-based planner (A* graph searching described in Chapter 3), although the strategy is equally applicable to any other planner driven by an objective function, e.g. RRT, PRM etc. Essentially, the stable A* algorithm first examines the stability of the robot when opening a new search node at a new location with a given configuration. The node is considered stable if β is larger than some nominal β_{min} that is satisfied. The present proposal, abstracted by Algorithm 7, takes into account SC as described by Equation 4.4 through the $ut\_transform()$ Algorithm 6, effectively transforming the fixed stability constraint ($\beta > \beta_{min}$) into a minimum confidence threshold ($SC > SC_{min}$) representative of the certainty in the stability prediction.

5.3.1 Results of A* Planner in the USAR Arena

In order to make a fair comparison between the two planners, a pre-processing step was first applied to the terrain model to label out obvious untraversable areas, e.g. walls and markedly steep slopes. Two sets of experiments of planning based on varying allowable boundaries for β_{min} and SC_{min} are studied in order to highlight the advantages of the probabilistic approach in generating safer and more optimal posture planning.



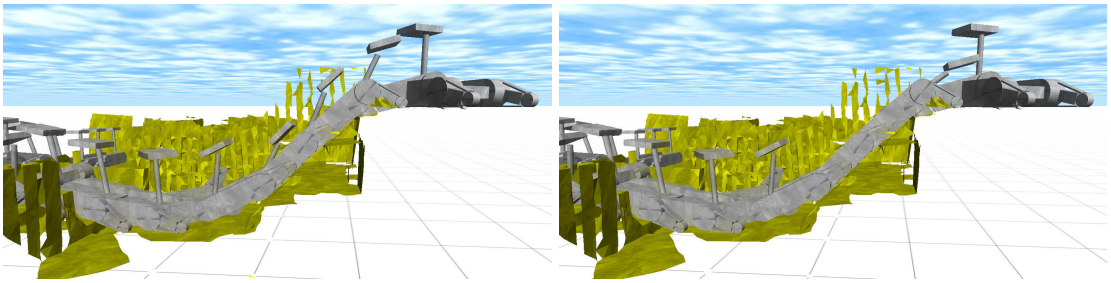
(a) Path where $\beta_{min} = 0.05$.



(b) Path where $SC\_min = 50\%$.

Figure 5.2: Planning based on the minimum safety margin and stability confidence in the USAR arena. Planning based on SC generates safer postures over stairs ($\phi_a = 0^\circ$ in **b**) when compared to the deterministic approach ($\phi_a = 20^\circ$ in **a**).

In the first place the result of planning based on the lowest allowable β_{min} and SC_{min} will be compared to highlight the advantages of the probabilistic approach in generating safer posture routes. Both planners are set to find a path from the top left corner of the USAR arena with a minimum possible $\beta_{min} = 0.05$ and $SC_{min} = 50\%$ to the goal at the bottom right corner. As explained in Section 2.4, the value of $\beta_{min} = 0.05$ was obtained experimentally as the border of stability when the robot was sitting on the 35° ramp of the arena, with the nominal configuration $(\phi_a = 90^\circ, \phi_f = 90^\circ)$. A positive β_μ is the only requirement to achieve $SC_{min} = 50\%$, consequently the minimum allowable safety confidence is assumed to be 50%. While for demonstration purposes, it may be useful to skate along the boundary of tip-over, in the real world (especially in rescue operations) one may need to ensure an SC_{min} significantly larger than 50% due to the cost of losing a robot entirely in a hazardous zone, not apt for subse-



(a) Path where $\beta_{min} = 0.05$.

(b) Path where $SC\_min = 50\%$.

Figure 5.3: Side view of robot poses and postures over the stairs shown in Figure 5.2.

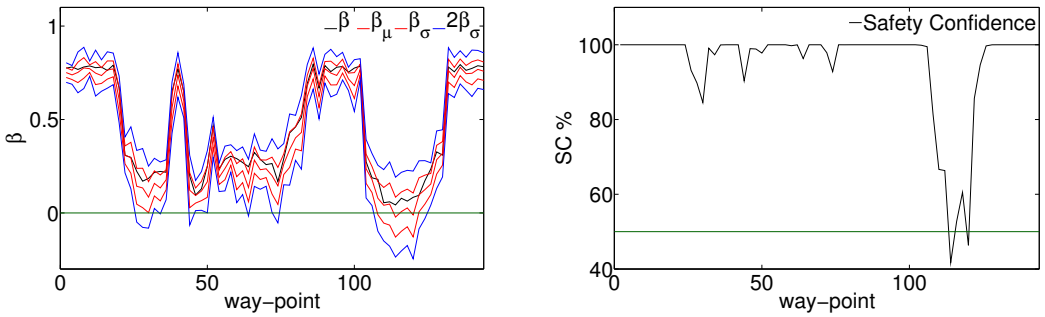
quent retrieval. In fact, one may need to consider the cumulative likelihood of instability across an entire mission and ensure that this is sufficiently small (e.g on Mars, SC would need to be close to 100% for all practical purposes).

The results are depicted in Figure 5.2, where 5.2a and 5.2b illustrate the outcomes of the shortest deterministic and probabilistically stable paths respectively. Only a limited number of the robot poses are shown in the figure for clarity. In both instances the final paths traverse through the step-fields and the stairs, and the robot configurations over both trajectories end up being quite similar (except on the stairs, way-points around $100-130$ in Figure 5.4, discussed below).

The comparison of SC and β over these trajectories are depicted in Figure 5.4. The mean value of the stability measure obtained using the UT transform β_μ at each instant is depicted in red, with the standard deviation σ (68%) and $2\times\sigma$ (95%) around the mean depicted in dashed red and blue in Figure 5.4a and 5.4c. Figure 5.4b and 5.4d illustrate the corresponding SC measures of the resulting two paths. The horizontal dark green dash-dot lines are indicating the reference points where $\beta = 0$ or $SC_{min} = 50\%$.

It can be seen how by setting an arbitrary lower boundary ($\beta_{min} = 0.05$) the deterministic planner's limited concern about the instantaneous value of β results in paths with instances where, although β is computed to be always bigger than $\beta_{min} = 0.05$ as shown in Figure 5.4a, in some places the corresponding β_μ is actually negative ($SC < 50\%$), indicating a high risk for tip-over instability as illustrated in Figure 5.4b. This happens for instance over the stairs (way-points around 117), where β_μ is indeed less than 0.05.

On the other hand, a planner considering an $SC_{min} = 50\%$ as depicted by Figure 5.4d, might end up with instances when β_μ is less than 0.05 in some places (see Figure 5.4c). However, SC



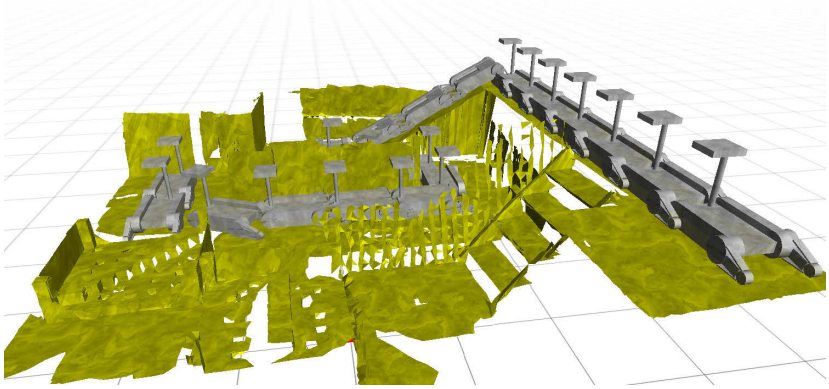
(a) β ($\beta_{min} = 0.05$). **(b)** SC ($\beta_{min} = 0.05$).

(c) β ($SC_{min} = 50\%$). **(d)** SC ($SC_{min} = 50\%$).

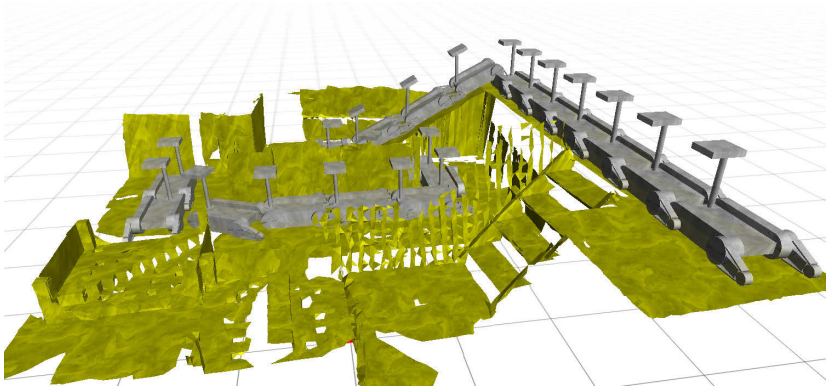
Figure 5.4: Comparison of SC and β over the trajectories depicted in Figure 5.2 where $\beta_{min} = 0.05$ and $SC_{min} = 50\%$ in the USAR arena. The horizontal dark green dash-dot lines are indicating the reference points where $\beta = 0$ or $SC_{min} = 50\%$.

only requires a positive β_μ to remain over the threshold of 50%, which is comfortably achieved by the planner generating postures with lower sensor head heights (e.g. $\phi_a = 0^\circ$ over the stairs section depicted in Figure 5.2b), compared to the resulting postures ($\phi_a = 20^\circ$) of a deterministic planner when $\beta_{min} = 0.05$ (Figure 5.2a). This example clearly shows how the probabilistic approach tends towards more conservative paths stability-wise than a deterministic planner in areas where uncertainty escalates. A side view of robot postures over the stairs in Figure 5.2 is shown in Figure 5.3 for easier comparison.

In the following example the safety margin and stability confidence are increased to $\beta_{min} = 0.20$ and $SC_{min} = 70\%$ respectively as shown in Figure 5.5. Both criteria will now filter out the stairs and step-fields, tending towards a safer but longer path to the goal through the ramp, as shown in Figure 5.5a and 5.5b for $\beta_{min} = 0.20$ and $SC_{min} = 70\%$ respectively. Planning based on $\beta_{min} = 0.20$ has configured the robot where ($\phi_a = 0^\circ$) over the ramp. Yet given



(a) Path where $\beta_{min} = 0.20$.

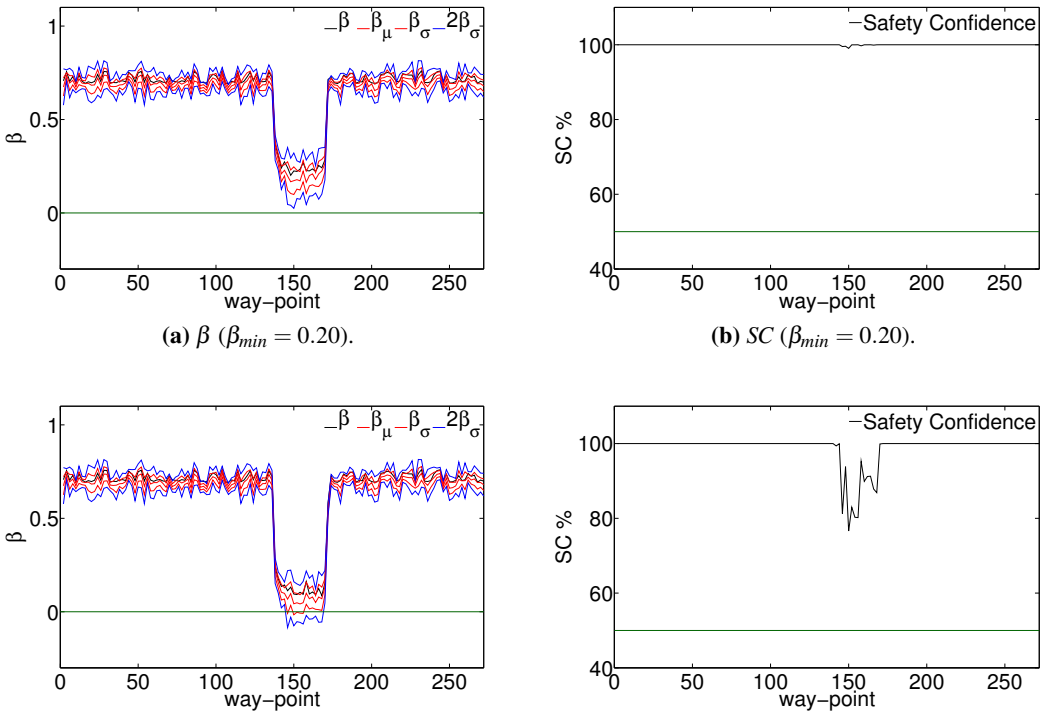


(b) Path where $SC\_min = 70\%$.

Figure 5.5: Planning based on a comfortable safety margin and stability confidence in the USAR arena. Planning based on *SC* generates postures with better visibility over the ramp ($\phi_a = 50^\circ$ in **b**) compared to the deterministic approach ($\phi_a = 0^\circ$ in **a**).

the higher certainty of the map over the ramp (as opposed to more rugged terrain sections), the probabilistic planner where $SC_{min} = 70\%$ can satisfy the stability constraint with a better field of view configuration ($\phi_a = 50^\circ$) for the same area. As with the earlier example, the comparison of *SC* and β over the resulting trajectories are depicted in Figure 5.6.

To fulfil the required $\beta_{min} = 0.20$, the planner based on β_{min} has to configure the sensor head to $\phi_a = 0^\circ$ angle. According to Figure 5.6a, β_μ is around 0.20 which shows a predictable stability confidence as depicted in Figure 5.6b. On the other hand, according to Figure 5.6d, the planner where $SC_{min} = 70\%$ cares about *SC* instead of an instant value of β. This therefore allows the planner to generate arm configurations with better field of view and despite β being less than 0.20 in some points over the ramp in Figure 5.6c, thanks to a small σ, *SC* of the



(a) β ($\beta_{min} = 0.20$). **(b)** SC ($\beta_{min} = 0.20$).

(c) β ($SC_{min} = 70\%$). **(d)** SC ($SC_{min} = 70\%$).

Figure 5.6: Comparison of SC and β over the trajectories depicted in Figure 5.5 where $\beta_{min} = 0.20$ and $SC_{min} = 70\%$ in the USAR arena. The horizontal dark green dash-dot lines are indicating the reference points where $\beta = 0$ or $SC_{min} = 50\%$.

way-points are always bigger than 70%. It can be observed in the ramp area (way-points around $140 - 170$) that uncertainty is very small (β_μ and β_σ around 0.20, Figure 5.6a) and the probabilistic approach is then able to exploit this to generate postures with better visibility than the deterministic planner.

Comparison of Figure 5.5b and 3.20a shows that the outcome of the planner with a constraint of $SC_{min} = 70\%$ resembles the result of planning when the deterministic stability/reconfiguration cost is the objective of the planner and distance is excluded from the cost function. It is worth remembering that for the case of the most stable path a constraint of $\beta_{min} = 0.05$ was introduced for a node to be considered in the search space, and the reconfiguration algorithm will thus specify the posture with the best field of view which satisfies this stability constraint. It is therefore perfectly plausible that the planner may still find a path through a way-points where β is uncomfortably close to $\beta_{min} = 0.05$, consequently resulting

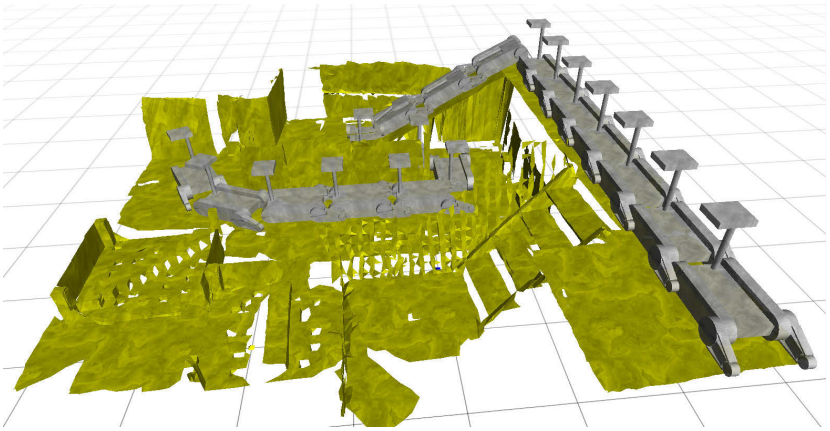


Figure 5.7: Path where $SC\_min = 90\%$ in the USAR arena.

in a negative β_μ ($SC < 50\%$) which indicates a high risk for tip-over instability. In addition, because distance is not considered in the objective function, the overall path would not be guaranteed to be the shortest possible trajectory. As depicted in Figure 5.7, when the safety constraint is increased to higher confidence levels ($SC_{min} = 90\%$), visibility is sacrificed and postures over the ramp exhibit a lowered arm, similar to the postures depicted in Figure 5.5a, in order to fulfil the more restrictive stability confidence criteria.

5.3.2 Results of A* Planner in the UTIAS Arena

In this section the 3D UTIAS terrain data set is used to study the outcomes of planning longer paths with different values for β_{min} and SC_{min}. Comparison of results in Figure 5.8 and 5.9 show how, when the stability constraint is a reasonable medium value, the statistical approach can find a more effective and shorter path than the deterministic technique (the path shown in orange). Moreover statistical analysis on the outcome of planning based on minimum allowable $\beta_{min} = 0.05$ has resulted in a $SC < 50\%$ in some places over the trajectory which indicates a high risk for tip-over instability.

The outcomes of a planner based on different deterministic stability margins are shown in Figure 5.8 where the path with the lowest allowable safety margin $\beta_{min} = 0.05$ is illustrated in black, and paths where $\beta_{min} = 0.10$ and $\beta_{min} = 0.20$ are depicted in orange and yellow respectively. Gray-scale colour coding indicates the height of the terrain from 0 to $2.76m$. A pre-processing algorithm based on terrain gradients was first applied to the model to label out obviously untraversable steep slopes, shown in dark Brown. This effectively filtered out

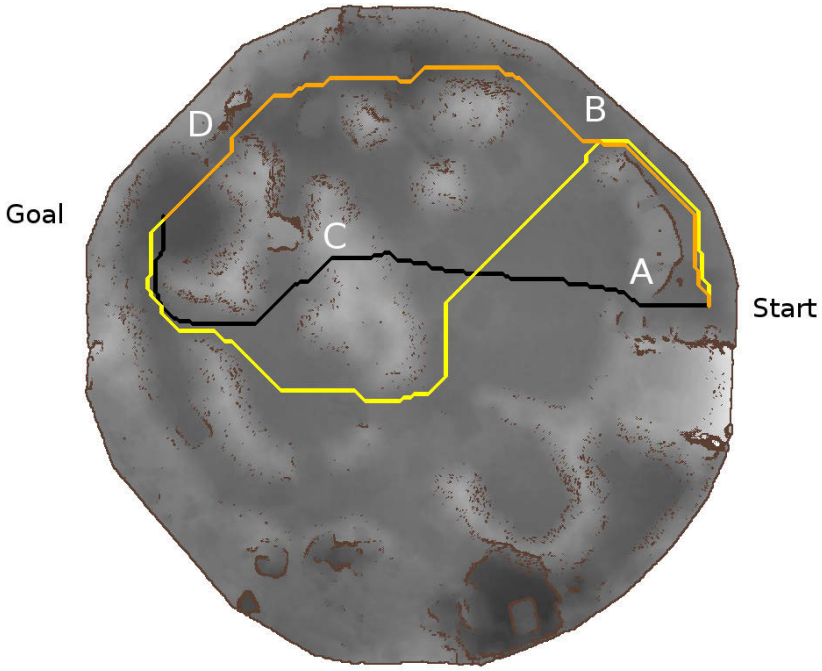


Figure 5.8: Planning based on deterministic stability margin in the UTIAS arena. Path with the lowest allowable safety margin $\beta_{min} = 0.05$ is illustrated in black and the trajectories where $\beta_{min} = 0.10$ and $\beta_{min} = 0.2$ are depicted in orange and yellow respectively.

regions with more than a 56° gradient which corresponds to the critically unstable point in pitch for a robot sitting flat with an arm configuration at 90°, i.e. the highest visible point in flat terrain. In the same way, Figure 5.9 shows the effect of different values of SC_{min} on the planner, where black, orange and yellow illustrate trajectories where $SC\_min = 50\%$, $SC\_min = 70\%$ and $SC\_min = 90\%$ respectively.

For comparison purposes, the result of uncertainty analysis on the trajectories depicted in Figure 5.8 are shown in Figure 5.10, where the mean values of stability measure driven using UT transform at each instant are depicted in red, the standard deviation σ (68%) and $2 \times \sigma$ (95%) around the mean are depicted in dashed red and blue in Figure 5.10a, 5.10c and 5.10e for the trajectories where $\beta_{min} = 0.05$, $\beta_{min} = 0.10$ and $\beta_{min} = 0.20$ respectively. Figure 5.10b, 5.10d and 5.10f are sequentially illustrating the corresponding SC measures during these three trajectories. In the same way the corresponding uncertainty analysis on the trajectories depicted in Figure 5.9 are shown in Figure 5.11.

The result of planning based on the lowest allowable $\beta_{min} = 0.05$ and $SC_{min} = 50\%$ (depicted in black in Figure 5.8 and 5.9 respectively) are found to be quiet coincidental. These

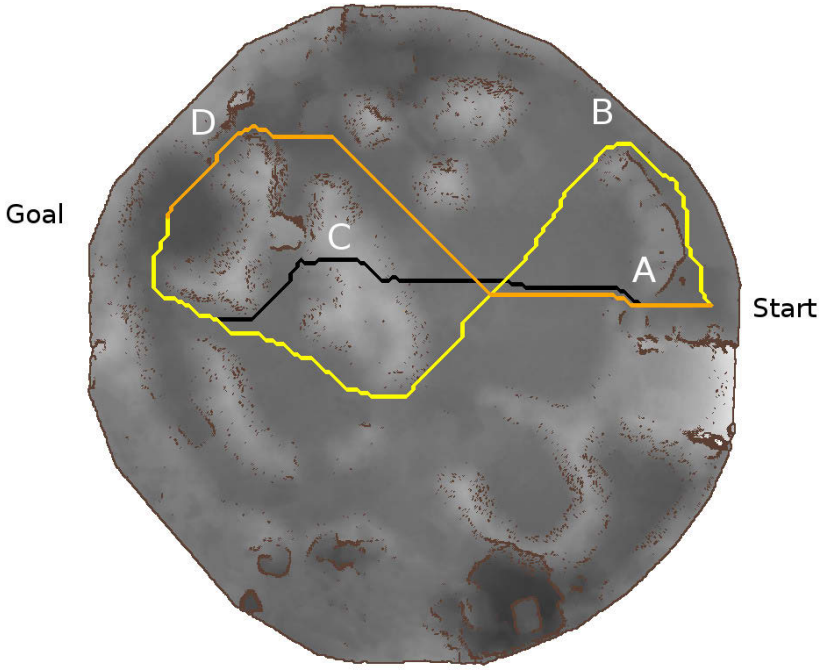
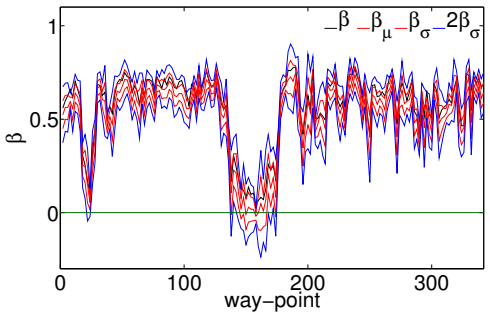


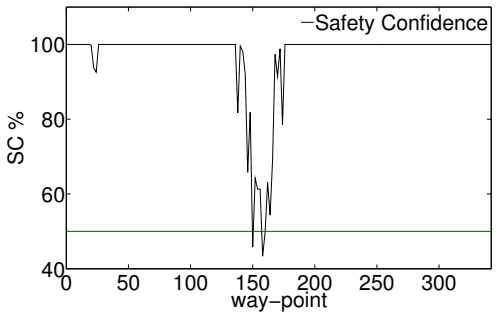
Figure 5.9: Planning according to *SC* measure in the UTIAS arena. Path with the lowest allowable safety confidence $SC_{min} = 50\%$ is illustrated in black and the trajectories where $SC_{min} = 70\%$ and $SC_{min} = 90\%$ are depicted in orange and yellow respectively.

two trajectories are going through (A) and passing directly over the central hill (C). Although the planning based on $\beta_{min} = 0.05$ ensures that instant value of β are always larger than the minimum value, β_{μ} is found to be negative over the more challenging section, hence resulting in an $SC < 50\%$ i.e. a high risk for a tip-over instability as illustrated in the way-points around 150 in Figure 5.10b. This would not represent a dangerous situation when planning is based on an $SC\_min = 50\%$ as the planner will reconfigure the robot so that it fulfils the minimum safety confidence as illustrated in Figure 5.11b. Moreover planning based on more significant stability margins and safety confidence ($\beta_{min} = 0.20$ and $SC_{min} = 90\%$) results in longer routes, depicted in yellow in Figure 5.8 and 5.9 respectively. These trajectories are quite similar to the result of planning when the deterministic stability/reconfiguration cost are the objective of the planner and distance is excluded from the cost function, as illustrated in yellow in Figure 3.23.

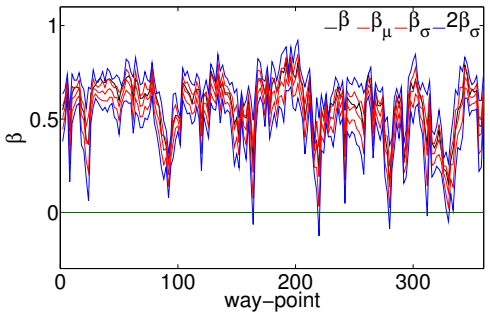
Planning, based on a comfortable stability margin and safety confidence ($\beta_{min} = 0.10$ and $SC_{min} = 70\%$) produced some interesting results. where $\beta_{min} = 0.10$ the planner could not find a trajectory through the front section (A) and resorted to move up towards (B), eventually finding a path via (D) to the goal. On the other hand, the planner where $SC_{min} = 70\%$ considered the



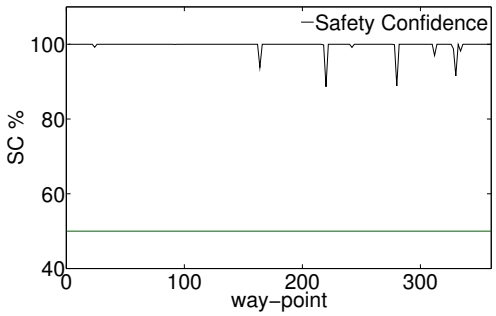
(a) β ($\beta_{min} = 0.05$).



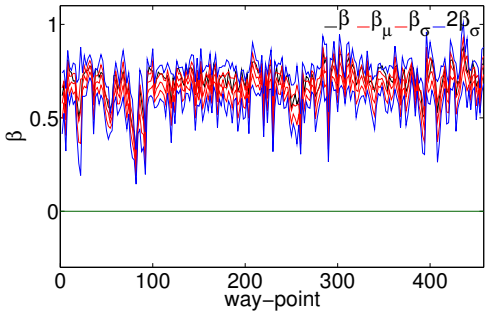
(b) SC ($\beta_{min} = 0.05$).



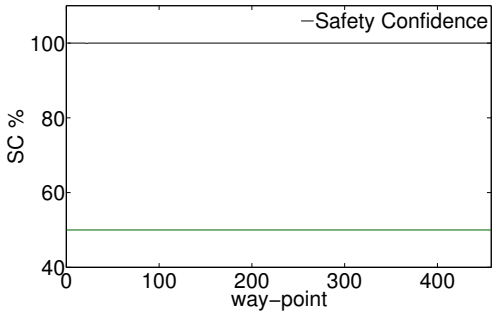
(c) β ($\beta_{min} = 0.10$).



(d) SC ($\beta_{min} = 0.10$).

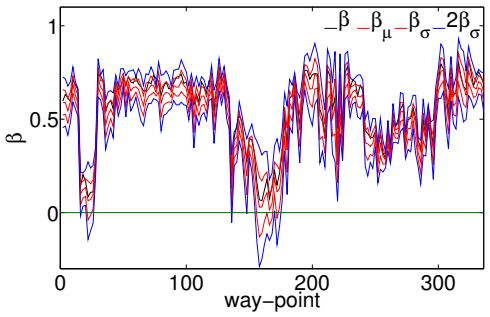


(e) β ($\beta_{min} = 0.20$).

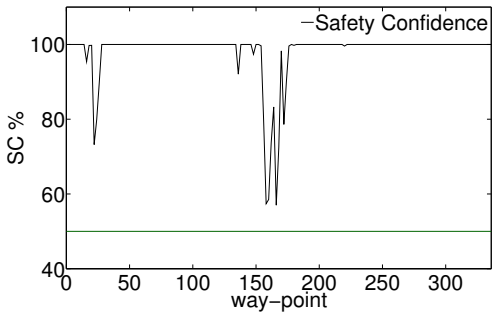


(f) SC ($\beta_{min} = 0.20$).

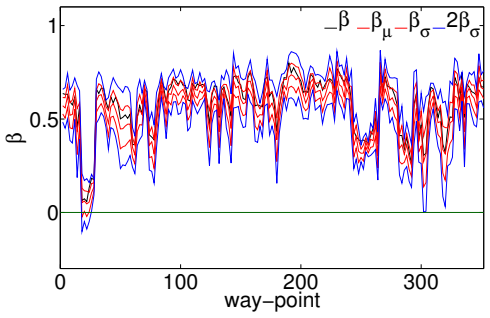
Figure 5.10: Comparison of SC and β over the trajectories depicted in Figure 5.8 where $\beta_{min} = 0.05$, $\beta_{min} = 0.10$ and $\beta_{min} = 0.20$ in the UTIAS arena. The horizontal dark green dash-dot lines are indicating the reference points where $\beta = 0$ or $SC_{min} = 50\%$.



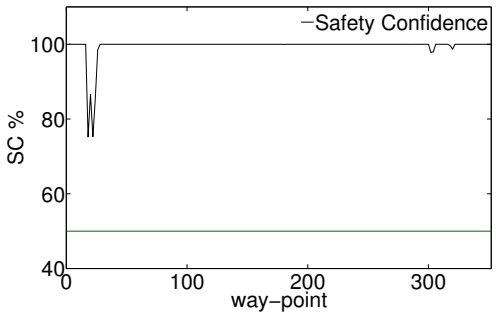
(a) β ($SC_{min} = 50\%$).



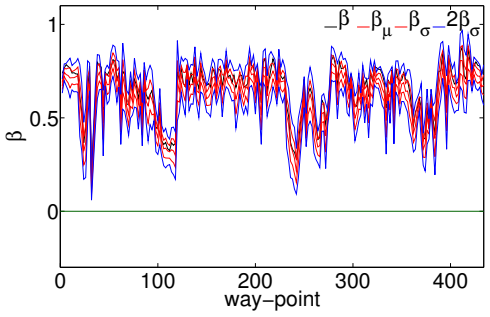
(b) SC ($SC_{min} = 50\%$).



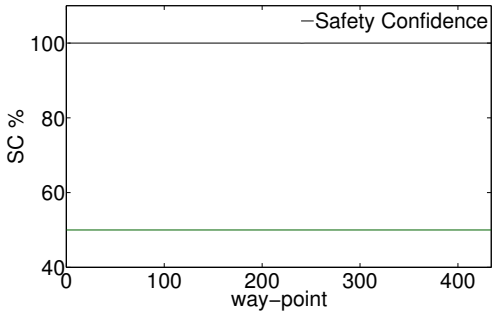
(c) β ($SC_{min} = 70\%$).



(d) SC ($SC_{min} = 70\%$).



(e) β ($SC_{min} = 90\%$).



(f) SC ($SC_{min} = 90\%$).

Figure 5.11: Comparison of SC and β over the trajectories depicted in Figure 5.9 where $SC_{min} = 50\%$, $SC_{min} = 70\%$ and $SC_{min} = 90\%$ in the UTIAS arena. The horizontal dark green dash-dot lines are indicating the reference points where $\beta = 0$ or $SC_{min} = 50\%$.

front section (A) feasible and found a shorter path which goes straight up to the middle of the arena and then coincides with the path where $\beta_{min} = 0.10$ in the final stages in the area labelled as (D). Looking at Figure 5.11c around way-point 25 it is seen how β around (A) is less than $\beta_{min} = 0.10$, revealing the reason why planning based on β_{min} would not consider this area traversable. Looking at the value of *SC* in Figure 5.11d confirms that although β is less than $\beta_{min} = 0.10$ around (A), safety confidence is bigger than 70% and the planner regards this region as comfortably stable to plan over. This example shows how planning based on statistical data instead of the instant values can result in more effective and at the same time safer routes. The overall length of the trajectories illustrated in Figure 5.8 and 5.9 are summarised in Table 5.1.

β_{min}	length (m)	SC_{min}	length (m)
0.05	39.6184	50%	39.0450
0.10	43.4073	70%	41.8475
0.20	54.9470	90%	53.0440

Table 5.1: Overall length of paths shown in Figure 5.8 and 5.9.

5.4 Implementation with RRT planner

The proposed stability analysis approach is applicable to any cost based planner driven by an objective function, e.g. A*, probabilistic roadmap (PRM) [91], rapidly exploring random tree etc. So far in this thesis an implementation with the grid based A* algorithm has been chosen to present the results. In this section an integration of the strategy in the well established sampling based RRT planner is presented for completeness. The RRT planner is chosen here as an example to show the applicability of the approach to a sampling based planner. The main advantage of RRT planner in comparison to grid based A* algorithm is when the full search is fundamentally impractical, or the calculation time is too long. No smoothing algorithms have been applied to the trajectories generated by RRT.

RRTs were originally introduced by Steven M. LaValle in 1998 [86] and have been applied to holonomic, nonholonomic and other problems such as randomized kinodynamic planning [92]. Fundamentally RRT builds a space-filling tree ($\mathcal{T}$) and extends it randomly to efficiently search high-dimensional spaces. As RRT planners can quickly cover an environment

Algorithm 8 The RRT planner algorithm

```
1:  function biuld_RRT(x_init, K)
2:      T.init(x_init)
3:      for k = 1 → K do
4:          x_rand ← random_state
5:          x_near ← nearest_neighbour
6:          if new_state(x_near, x_rand) then
7:              T.add_vertex(x_rand)
8:              T.add_edge(x_near, x_rand)
9:              if (x_rand = x_goal) then
10:                 return T
11:             end if
12:         end if
13:     end for
14:     return trapped
15: end function
```

by the random tree expansion, they have been widely used in autonomous robotics path planning. When extending the tree, it is able to regularly check the collision with obstacles and differential constraints (nonholonomic, kinodynamic etc). In the work presented in this thesis we will add a stability constraint. The main favourable properties of RRTs are:

- The tree expands very fast into unexplored points of the search space.
- The RRT is probabilistically complete, meaning that if a solution exists and an adequate amount of time is given, the algorithm will eventually find it.
- The RRT can be very easily adapted to a whole variety of applications, such as cost based or constrained path finding problems.
- Unlike the PRM, the RRT will always remain connected (regardless of the number of edges), thus it would not spend time to process nodes that may not be used later on.
- When the goal node is added to the tree, the final path would consist of the branches which are observed while back tracking from the goal to start node (similar to parent's chain in A* algorithm).

In spite of the fact that the RRT planner does not need a grid to expand, for simplicity and comparison purposes, lets assume that search space is a 2D grid equal to A* algorithm's

Algorithm 9 The original RRT state check algorithm

```
1: function new_state(x_near, x_rand)
2:     for x_i = x_near → x_rand do // all states along a straight line connection
3:         if x_i = x_obs then
4:             return false
5:         end if
6:     end for
7:     return true
8: end function
```

environment explained in Algorithm 1. The grids of the graph are classified into two sets referred to as $obstacle$ and $free$. The path planning can be viewed as a search in this grid from an initial start node, x_{init} to the goal node x_{goal} while avoiding obstacle nodes x_{obs}. An RRT that is rooted at x_{init} and has K vertices can be summarized as an iterative procedure as illustrated in Algorithm 8.

In beginning, the algorithm initiates RRT tree $\mathcal{T}$ with start node as the first vertex. In each iteration, the algorithm attempts to extend the RRT by adding a random new node x_{rand}. The nearest vertex x_{near} already in the RRT to the given x_{rand} will be chosen according to a metric like Euclidean distance. The function new_state is called in this stage to detect potential collisions to determine whether the x_{rand} (and all intermediate states) satisfies the global constraints as shown for the simple scenario of obstacle avoidance in Algorithm 9. If new_state is successful, the x_{rand} is added as a new vertex to $\mathcal{T}$. An edge from x_{near} to x_{rand} is also added. If the recently added vertex reaches the x_{goal}, the algorithm successfully returns $\mathcal{T}$ and the final path will be the chain of branches from the x_{goal} back to the x_{init} node.

To guarantee the stability of $\mathcal{T}$, the new_state function is modified according to Algorithm 10. For each way-point between x_{near} and x_{rand}, the algorithm calculates the statistical information about the tip-over instance using the $ut\_transform()$ function in the 3D physical simulator. The new branch in the RRT tree would be considered safe only if it is collision-free and also satisfies the corresponding minimum safety confidence. The block diagram of the overall stable RRT algorithm is illustrated in Figure 5.12.

5.4.1 Results of RRT Planner in the USAR Arena

The preference of planning based on a probabilistic margin in comparison with a deterministic stability measure was discussed in Section 5.3. Here we are going to compare the original

Algorithm 10 The stable RRT state check algorithm

```
1: function new_state(x_near, x_rand)
2:     for x_i = x_near → x_rand do // all states along a straight line connection
3:         ut_transform()
4:         if (x_i = x_obs ∨ SC < SC_min) then
5:             return false
6:         end if
7:     end for
8:     return true
9: end function
```

RRT planner with the RRT planner constrained on the safety confidence SC measure. A grid resolution of $5cm$ was assumed for both terrains which resulted in a 2D graph with dimensions of 164×150 and 784×776 for USAR and UTIAS arenas respectively. Some implementations of RRTs limit the length of the connection between the tree and a new state by a growth factor [93]. This forces the random sample to lie within a maximum distance from the tree and limits the size of the incremental growth. In this work, the random sample is uniformly sampled from the entire search space to allow the tree to quickly expand towards large unsearched areas. This freedom in expansion sometimes results in long straight branches (routes) in the tree, but the algorithm will check the feasibility of all intermediate way-points before accepting the new state.

In the first instance the result of the original RRT is compared with the trajectories achieved from planning based on lowest allowable safety confidence, $SC_{min} = 50\%$ and a comfortable margin $SC_{min} = 70\%$, in the USAR arena. The outcomes of the proposed stable RRT planner are illustrated from a top view in Figure 5.13 on the USAR arena in comparison with the standard RRT, where Figure 5.13a is showing all three trajectories simultaneously, and Figure 5.13b presents the RRT tree and trajectory of the ordinary path in a separate figure. A pre-processing algorithm was first applied to the 3D map to determine extreme untraversable areas, e.g. walls and markedly steep slopes. Results in Figure 5.13a show the path derived from the original RRT in blue while the way-points where the robot was not stable for the fixed vertical arm and flipper pose are highlighted in red. The stable path with the the lowest allowable safety confidence $SC_{min} = 50\%$ and the trajectories where $SC_{min} = 70\%$ are depicted in black and yellow respectively.

While ordinary route and stable path where $SC_{min} = 50\%$ may find a way to the goal either from stairs or via the ramp in the top left corner of the arena, the planning with more

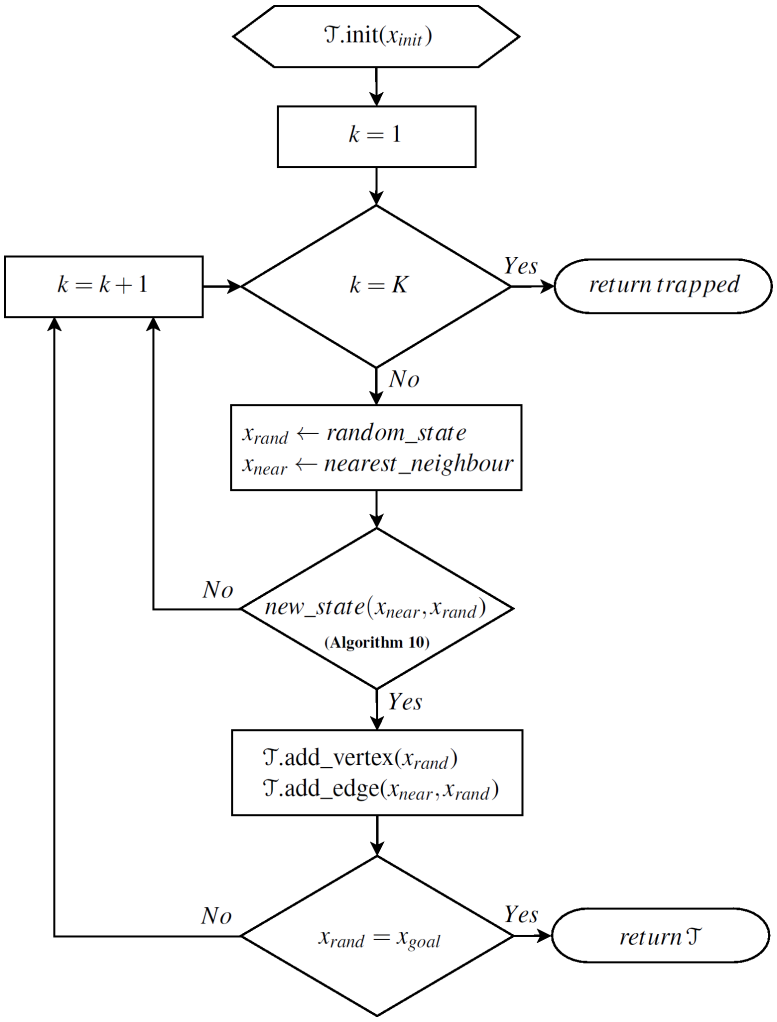
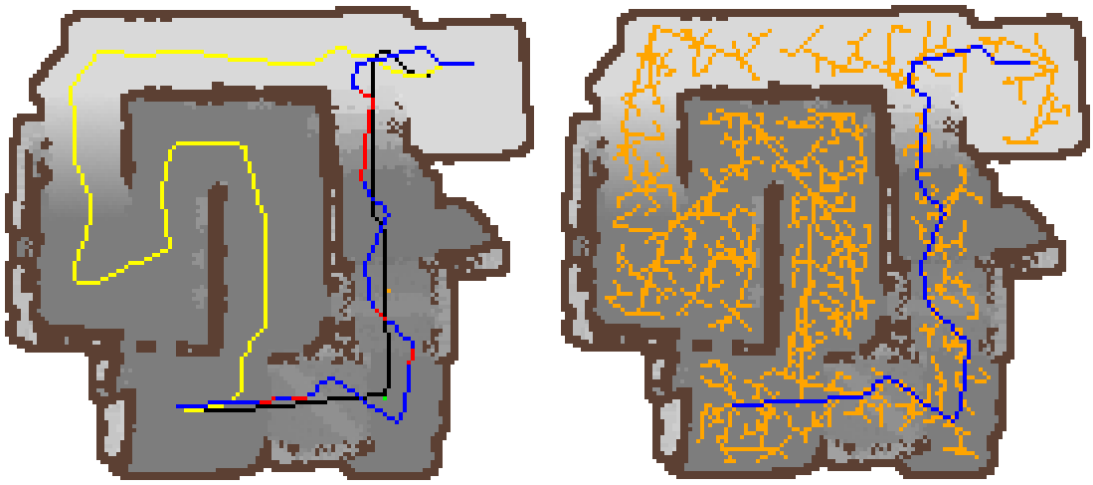


Figure 5.12: The block diagram of the stable RRT algorithm.

conservative stability constraint of $SC_{min} = 70\%$ leaves the ramp the only possible trajectory. As illustrated in Figure 5.13b, the original RRT tree has expanded entire the USAR arena, but most of the time the shorter route via the stairs was chosen as the final trajectory.

The robot configurations along stable trajectories are depicted in Figure 5.14, where Figure 5.14a and 5.14b illustrate the outcomes of the stable paths where $SC_{min} = 50\%$ and $SC_{min} = 70\%$ respectively. Only a limited number of the robot poses are shown in the figure for clarity. The corresponding uncertainty analysis are shown in Figure 5.15. Both planners have handled the corresponding SC_{min} constraint successfully while expanding the RRT trees. To fulfil $SC_{min} = 50\%$, the planner has configured robot to $\phi_a = 0^\circ$ over the stairs section depicted in



(a) The trajectories from the top view.

(b) The RRT tree and trajectory of the ordinary path.

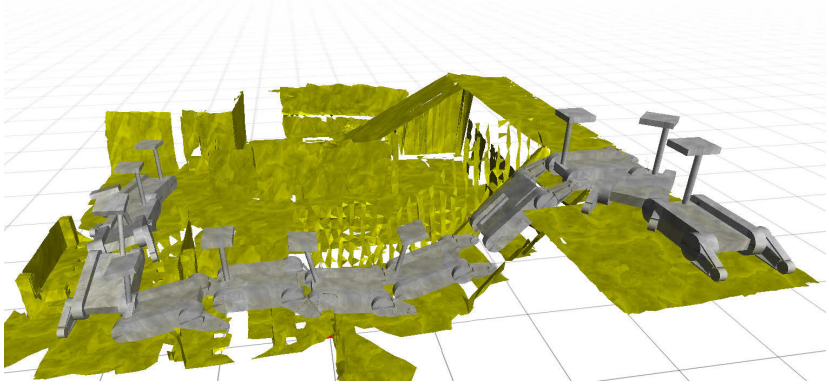
Figure 5.13: Results of stability criterion in the RRT algorithm in the USAR arena. **a)**: Results show the path derived from the original RRT planner in blue with unstable points in red. The stable path with the lowest allowable safety confidence $SC_{min} = 50\%$ and the trajectories where $SC_{min} = 70\%$ are depicted in black and yellow respectively. **b)**: The RRT tree is depicted in orange and has expanded to almost the entire whole 164×150 search space. In the majority of times the algorithm came up with the shorter path via stairs as depicted in blue.

Figure 5.14a, while given the higher certainty of the map over the ramp, the algorithm can satisfy the stability constraint $SC_{min} = 70\%$ with a better field of view configuration ($\phi_a = 50^\circ$), as illustrated in Figure 5.14b.

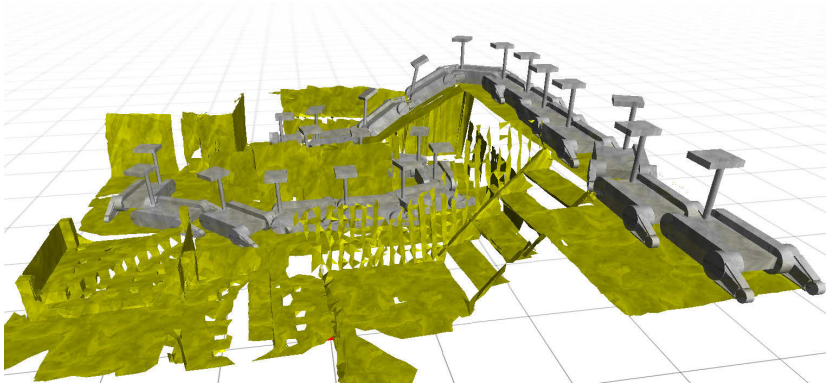
Path (m)	Minimum (A*)	Average RRT over 10 runs	σ
Original	7.9899	9.5692	1.7060
$SC_{min} = 50\%$	8.2485	9.8189	1.6117
$SC_{min} = 70\%$	14.0727	17.2135	1.3460

Table 5.2: Comparison of average length and σ of RRT paths in 10 runs versus the corresponding minimum A* trajectories in the USAR arena.

Table 5.2 summarises the statistical information about average length and standard deviation (σ) of RRT paths over 10 runs versus the corresponding minimum A* trajectories in the USAR arena. Since any increase in the stability constraint will shrink the expansion of the RRT tree, there are less options to choose from for the planner, and over a number of test runs σ will generally decrease as SC_{min} increases. It can be observed how for the original RRT and



(a) Path where $SC\_min = 50\%$.



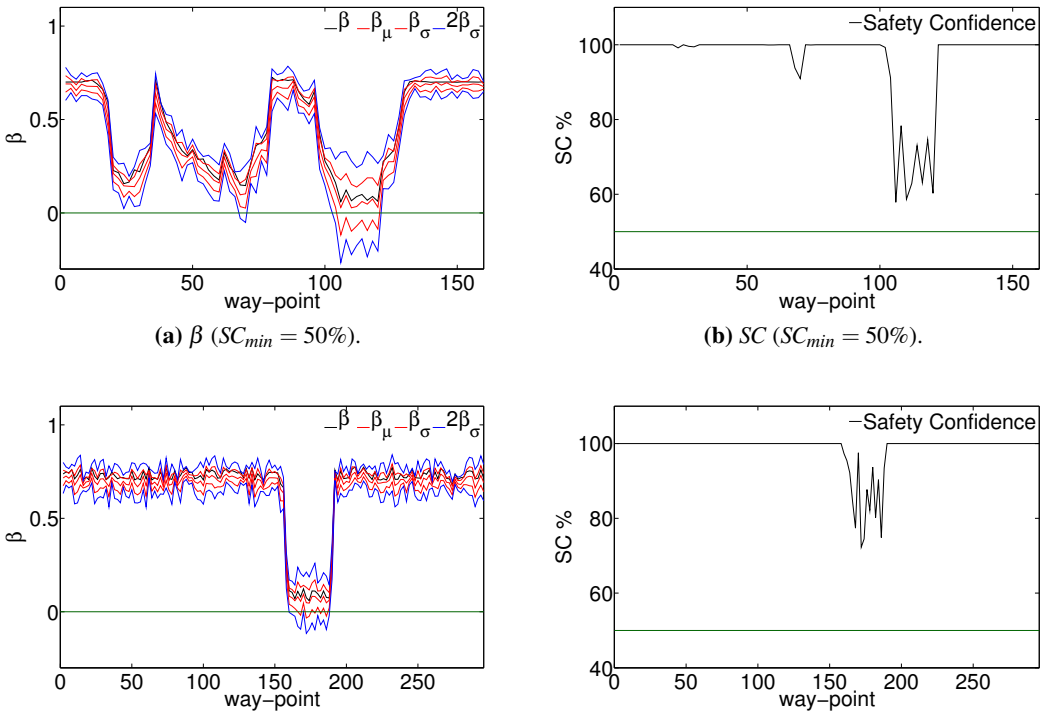
(b) Path where $SC\_min = 70\%$.

Figure 5.14: Results of probabilistic stability criterion on RRT algorithm in the USAR arena.

the case when $SC_{min} = 50\%$, σ values are close together and reasonably larger than the RRT where $SC_{min} = 70\%$. This is because the first two planners have, independently of the adopted configurations, two clear alternatives when it comes to traverse the terrain to go to the goal point, through a ramp or through the stairs, whereas the RRT where $SC_{min} = 70\%$ leaves the ramp as the only possible trajectory. This behaviour will become more apparent in the results of the UTIAS arena as the planner would have a larger search space.

5.4.2 Results of RRT Planner in the UTIAS Arena

In the following example the algorithm is applied to the online available UTIAS datasets. As this terrain mimics an outdoor environment, the comfortable stability confidence is increased to $SC_{min} = 90\%$ when searching for a reliable tip-over margin. In the same way, Figure 5.16



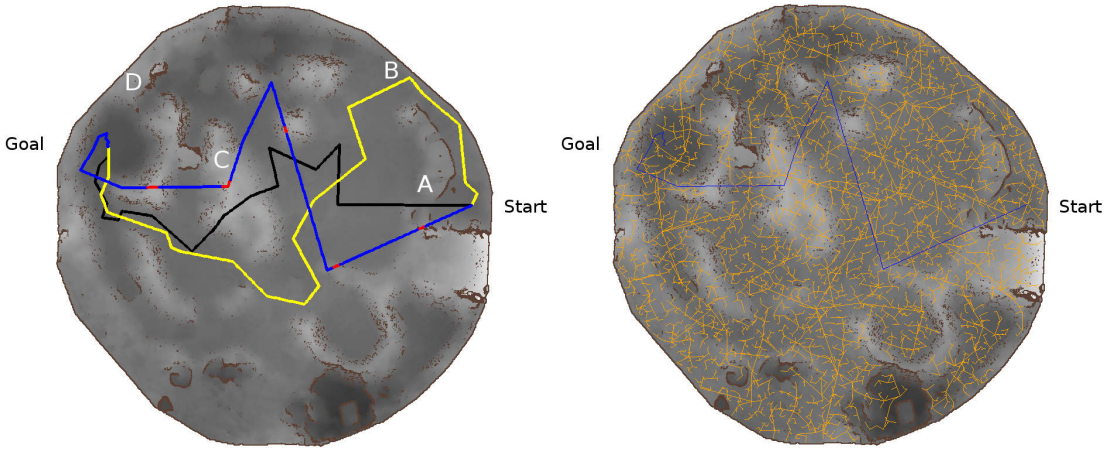
(a) β ($SC_{min} = 50\%$). **(b)** SC ($SC_{min} = 50\%$).

(c) β ($SC_{min} = 70\%$). **(d)** SC ($SC_{min} = 70\%$).

Figure 5.15: Comparison of SC and β over the trajectories depicted in Figure 5.14 in the USAR arena. The horizontal dark green dash-dot lines are indicating the reference points where $\beta = 0$ or $SC_{min} = 50\%$.

shows the result of the stable RRT algorithm in the UTIAS arena, where all three trajectories are depicted in Figure 5.16a for comparison and Figure 5.16b is separately illustrating the expansion of the RRT tree and resulting trajectory for the original planner. Figure 5.16a pictures the original RRT path in blue (with unstable points in red) and compares the effect of different values of SC_{min} on the planner, where black and yellow illustrate trajectories where $SC\_min = 50\%$ and $SC\_min = 90\%$ respectively.

While ordinary and stable RRT planner where $SC_{min} = 50\%$ may find a way to the goal either through (A) or (B), the planning with the highly conservative stability constraint of $SC_{min} = 90\%$ can only go through (B). As illustrated in Figure 5.16b, the original RRT tree has expanded the entire UTIAS arena as well, but mostly the planner came up with a route via (A) and, in this example, eventually found a path crossing from (C) to the goal. In the trials provided in Figure 5.16a, the stable path where $SC_{min} = 50\%$ is going through (A) and passing



(a) The trajectories from the top view.

(b) The RRT tree and trajectory of the ordinary path.

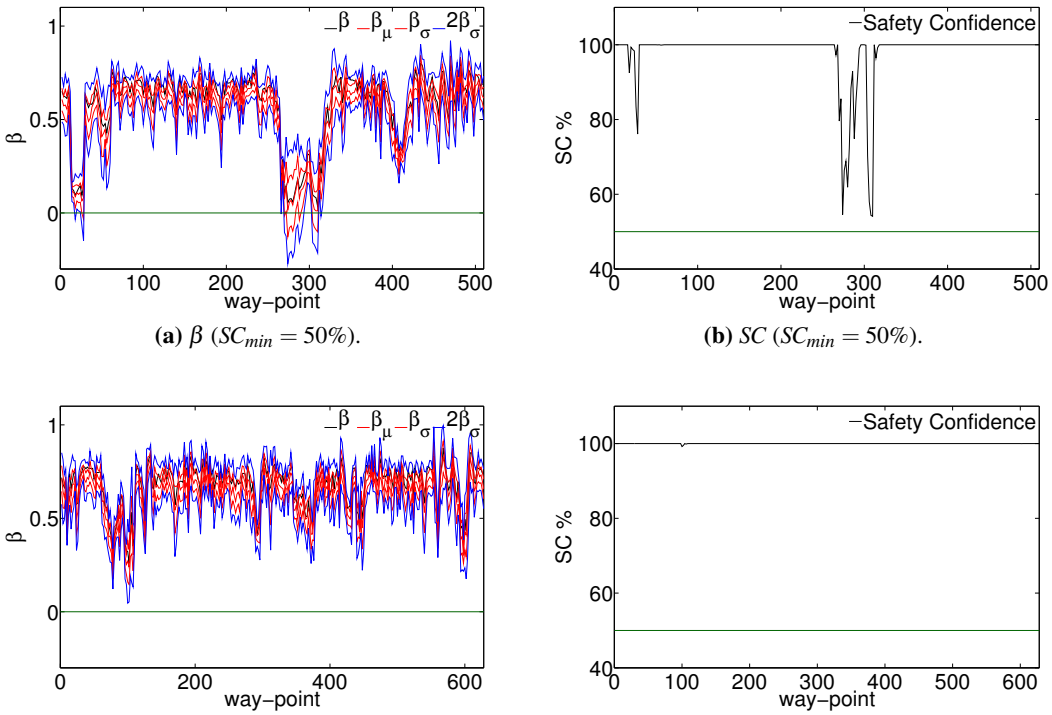
Figure 5.16: Results of stability criterion in the RRT algorithm in the UTIAS arena. **a)**: Results show the path derived from the original RRT planner in blue with unstable points in red. The stable path with the lowest allowable safety confidence $SC_{min} = 50\%$ and the trajectories where $SC_{min} = 90\%$ are depicted in black and yellow respectively. **b)**: The RRT tree is depicted in orange and has expanded to almost the entire whole 784×776 search space. In most instances the planner came up with a route via (A) and in this example eventually found a path crossing from (C) to the goal.

directly over the central hill (C), while the more conservative path where $SC_{min} = 90\%$ avoids both of these regions and moves up towards (B) choosing the longest and safest route which goes around part (C). The corresponding uncertainty analysis for stable routes are shown in Figure 5.17. According to this figure, the SC_{min} over the resulting path and entire RRT tree was effectively satisfied while searching the space for more branches.

Path (m)	Minimum (A*)	Average RRT over 10 runs	σ
Original	33.0823	56.5629	15.3264
$SC_{min} = 50\%$	39.0450	59.6210	11.3975
$SC_{min} = 90\%$	53.0440	73.2896	7.3383

Table 5.3: Comparison of average length and σ of RRT paths in 10 runs versus the corresponding minimum A* trajectories in the UTIAS arena.

In the same way, the statistical information about average length and σ of the paths are collected in Table 5.3. As expected from the previous observations in the USAR arena, σ is continuously descending as more constraints are applied to the planners. Yet given the larger



(a) β ($SC_{min} = 50\%$). **(b)** SC ($SC_{min} = 50\%$).

(c) β ($SC_{min} = 90\%$). **(d)** SC ($SC_{min} = 90\%$).

Figure 5.17: Comparison of SC and β over the trajectories depicted in Figure 5.16 in the UTIAS arena. The horizontal dark green dash-dot lines are indicating the reference points where $\beta = 0$ or $SC_{min} = 50\%$.

path planning search space in the outdoor terrain when compared to the more restrictive mock-up indoor arena, the relative σ of the routes in the UTIAS arena are significantly larger than their USAR arena counterparts.

5.5 Summary

This chapter presents a strategy for motion planning with stability uncertainty over rough terrains. The algorithm is able to exploit information gained from a statistical stability analysis to plan safe and effective routes under the presence of uncertainty in the robot kinematics, terrain model and localisation on the terrain. For completeness, the integration of the strategy with two well studied grid based and sampling based algorithms i.e. A* and RRT planners, are presented.

For completeness the proposed approach has also been embedded within a sampling based planner, consequently proving the ability of the proposed scheme to be nurtured within any cost-based search planner. It has been shown how the planner can accommodate the proposed SC_{min} constraint successfully whilst expanding the RRT trees. The analysis of the stable RRT planner's outcome has indicated how a higher stability constraint associated to the reconfiguration ability of the robot will further shrink the RRT tree's expansion space towards safer traversability areas.

Simulation results in an indoor rescue arena and an outdoor rover testing facility demonstrate the advantages of planning based on statistical stability information when compared with a deterministic approach. The results of path planning based on the lowest allowable safety margin shows that by setting an arbitrary lower boundary, the deterministic planner's limited concern about the instantaneous value of β results in paths with instances where, although β is computed to be always above a certain β_{min}, the corresponding β_{μ} can actually become negative ($SC < 50\%$) at times, indicating an unacceptable high risk of tip-over instability. The contingency of this potentially dangerous situation is minimised when planning is carried out based on *SC_min*, as the planner will reconfigure the robot so that it fulfils the minimum safety confidence at any given time. Moreover, when uncertainty levels are small (on ramps or sloped areas for instance) the probabilistic approach is able to exploit this to generate postures with better visibility than the deterministic planner. Comparison of the resulting trajectories in the outdoor UTIAS arena shows planning based on the proposed statistical stability methodology can result in more effective, and at the same time, safer routes.

6

Conclusions and Future Work

The objective of this thesis was to develop a strategy for stable path planning for reconfigurable robots operating over uneven terrains. This chapter summarises the contributions of this thesis and the several directions to extend the work are presented here.

6.1 Conclusions

The major theoretical and practical achievements of this thesis are:

- An approach to predict the stability of a vehicle based on a mathematical description of a robot's mechanical structure in the ODE dynamic physics simulator is presented. The proposed technique is able to import a 3D model of terrain and simulate the behaviour of a rigid body over the modelled environment and predict robot-terrain contact points required to calculate the FA stability margin. The validity of this approach is verified by the observation of close correlations between simulation experiments of a Packbot robot when contrasted with real data. The necessity of stability computations based on recommended physical simulation for stable path planning have also been illustrated through experiments over two common indoor obstacles i.e. ramps and stairs.

- The proposed stability estimate is employed to develop a strategy for motion planning in challenging environments for reconfigurable robots. This planner is able to minimise the length of the traversed path and optimise over conflicting robot reconfiguration cost between consecutive graph nodes in terms of visibility (i.e arm configurations closer to those orthogonal to the horizontal global plane which can afford a wider sensor view),

traction (i.e. flipper angles that provide the largest track-terrain interaction area) and energy consumption in the presence of stability constraint and limitations in the robot's joint movement. The performance of this influential flipper behaviour is presented by the experimental results over common indoor obstacles in a mock-up USAR arena. Experimental and simulation results with a reconfigurable tracked robot model have been also presented to validate the proposed approach over more simplistic approaches that do not account for reconfiguration in the path estimates.

- In missions over challenging environments, significant uncertainties are associated with the robot model, and localisation and terrain parameters which need to be taken into account for stability prediction. A statistical method for stability prediction which numerically estimates the probability distribution of contact points, CM and FA stability measure has been presented. The well-known conventional standard Monte Carlo scheme is employed to validate the proposed method while in order to speed up the process, a structured UT sampling approach is also presented which results in a significant improvement in computational efficiency. The utility of the proposed statistical stability prediction method is illustrated via experimental results over a ramp and a hill step-field. A novel probabilistic stability criterion derived from the cumulative distribution of a tip-over margin is introduced that allows a safety constraint to be dynamically updated by available sensor data as it becomes available.

- A probabilistic approach to account for robot stability uncertainty when planing motions over uneven terrains is hereby presented. The proposed statistical stability principle is exploited to address the challenging problem of global path planning in rough terrains with stability uncertainty. The proposed safety constraint authorises the planner to generate more conservative motion plans for areas with higher levels of uncertainty, while avoiding unnecessary caution in well-known areas.The proposed systematic approach is particularly applicable to reconfigurable robots that can assume safer postures when required, although it is equally valid for fixed-configuration platforms to choose safer paths to follow. For completeness, the implementations with a grid based A* algorithm as well as a sampling based RRT planner are illustrated. The advantages of planning with the proposed probabilistic stability margin are demonstrated with data collected from an indoor rescue arena, as well as an outdoor rover testing facility.

- The results of path planning based on lowest allowable safety margins confirms that the deterministic planner may generate some way-points where the mean value of β is negative, thereby indicating a high risk for tip-over instability. The probabilistic approach is able to avoid such dangerous situations as the planner will reconfigure the robot so that it fulfils the minimum safety confidence. However for regions with very small uncertainty, the probabilistic planner can effectively employ this confidence to generate postures with better visibility than the deterministic planner. The statistical analysis of outcomes of implementation with the sampling based RRT planner demonstrates that an increment in the stability constraint results in a reduction of the standard deviation of the length of resulting trajectories.

6.2 Future Work

The several potential directions to extend the work presented in this thesis for future work are:

- The proposed robot-terrain interaction prediction algorithm can be improved further to accommodate the full body dynamics and other terrain-interaction parameters that are known to interfere with the stability of the robot, such as friction and slippage. Moreover the initiative behind this approach is to provide solutions to robots entering rigid outdoor or indoor terrains like the urban search and rescue competition arena. As such, the proposed algorithms have not been extended to deal with environments that exceed this assumption such as sandy or muddy terrains. To meet the challenges posed by a general environment, other terrain classification techniques based on visual features such as colour or texture, will have to be incorporated into the existing framework.

- To enhance the locomotion and maximise the track-terrain contact area, the nominal configuration for flippers is assumed to be the angle which keeps them tangent to the terrain. While performance of this influential flippers behaviour is presented by the experimental results over common indoor obstacles in a mock-up USAR arena, the significantly more complex issue of maintaining traction along the path is left to future work.

- This work has not been concerned with the mapping or localisation aspects needed to guarantee a complete navigation solution for safe exploration of rough terrains. The 3D model of terrain is assumed to be provided as an input to the model, yet a practical system should be able to build up the 3D map in fly. These are undoubtedly important

challenges in themselves, and the effect that dynamic map and pose uncertainty will play in the proposed planning process can also be studied within the framework of this work.

- While the proposed *SC* measure accounts for the uncertainty associated with the robot dynamic model, its localisation in the ground and the terrain models, in this work these uncertainties are not directly targeted as an objective for global optimisation purposes. The stochastic stable planner proposed hereby is shown capable to employ *SC* or β as a constraint for suggested routes as well as a cost function for node-to-node optimisation, with the ultimate objective in mind of an iterative process suitable for practical on-line planning in search and rescue scenarios. It has been argued how the ultimate objective has remained to find paths that are stable/confidently stable, over the need to find the most stable/confidently stable path. Looking at global optimisation techniques fall outside the scope of the thesis, in the same manner than the problem of full safe robot navigation and decision-making under stability and state uncertainty does (e.g. POMDP), although it would be an interesting are of further research.

Bibliography

[1] R. B. McGhee and G. I. Iswandhi, "Adaptive locomotion of a multilegged robot over rough terrain," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 4, pp. 176–182, 1979. vii, 10, 11

[2] Y. Okada, K. Nagatani, K. Yoshida, S. Tadokoro, T. Yoshida, and E. Koyanagi, "Shared autonomy system for tracked vehicles on rough terrain based on continuous three-dimensional terrain scanning," *Journal of Field Robotics*, vol. 28, no. 6, pp. 875–893, 2011. vii, 1, 4, 11, 12, 46, 78

[3] Q. Morgan, C. Ken, G. Brian, F. Josh, F. Tully, L. Jeremy, W. Rob, and N. A. Y, "Ros: an open-source robot operating system," in *Proc. ICRA Workshop on Open Source Software*, Kobe, Japan, May 2009, p. 6. vii, 32, 33

[4] J. Pellenz, D. Gossow, and D. Paulus, "Robbie: A fully autonomous robot for robocuprescue," *Advanced Robotics*, vol. 23, no. 9, pp. 1159–1177, 2009. viii, 45, 47

[5] T. Kubota, Y. Kuroda, Y. Kunii, and T. Yoshimitsu, "Path planning for newly developed micro-rover," in *Proc. IEEE International Conference on Robotics and Automation*, Seoul, Korea, May 2001, pp. 3710–3715. viii, 46, 47

[6] K. Iagnemma, A. Rzepniewski, S. Dubowsky, P. Pirjanian, T. Huntsberger, and P. Schenker, "Mobile robot kinematic reconfigurability for rough-terrain," in *Proc. SPIE Symposium on Sensor Fusion and Decentralized Control in Robotic Systems III*, MA, USA, October 2000, p. 8. viii, 48

[7] K. Iagnemma, A. Rzepniewski, S. Dubowsky, and P. Schenker, "Control of robotic vehicles with actively articulated suspensions in rough terrain," *Autonomous Robots*, vol. 14, no. 1, pp. 5–16, 2003. viii, 19, 44, 48

[8] S. Yu, T. Wang, X. Li, C. Yao, Z. Wang, and D. Zhi, "Configuration and tip-over stability analysis for stair-climbing of a new-style wheelchair robot," in *Proc. IEEE International Conference on Mechatronics and Automation*, Xian, China, August 2010, pp. 1387–1392. viii, 49, 50

[9] Y. Liu and G. Liu, "Interaction analysis and online tip-over avoidance for a reconfigurable tracked mobile modular manipulator negotiating slopes," *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 4, pp. 623–635, 2010. viii, 12, 14, 24, 50

[10] G. Ishigami, G. Kewlani, and K. Iagnemma, "Predictable mobility," *IEEE Robotics and Automation Magazine*, vol. 16, no. 4, pp. 61–70, 2009. ix, 1, 78, 79

[11] G. Kewlania, J. Crawfordb, and K. Iagnemma, "A polynomial chaos approach to the analysis of vehicle dynamics under uncertainty," *Vehicle System Dynamics*, vol. 50, no. 5, pp. 749–774, May 2012. ix, 79, 80

[12] C. Fei, C. Ferdinando, C. Carlo, H. Traveler, S. Giuseppe, and C. Darwin, "In-hand precise twisting and positioning by a novel dexterous robotic gripper for industrial high-speed assembly," in *Proc. IEEE International Conference on Robotics and Automation*, Hong Kong, June 2014, pp. 270–275. 1

[13] A. Bonchis, E. Duff, J. Roberts, and M. Bosse, "Robotic explosive charging in mining and construction applications," *IEEE Transactions on Automation Science and Engineering*, vol. 1, no. 1, pp. 1–6, March 2013. 1

[14] F. Z. Robert Grisso, John Perumpral, "An empirical model for tractive performance of rubber-tracks in agricultural soils," *Journal of Terramechanics*, vol. 43, no. 2, pp. 225–236, 2006. 1, 78

[15] E. Papadopoulos and D. A. Rey, "The force angle measure of tipover stability margin for mobile manipulatiors," *Vehicle System Dynamics*, vol. 33, no. 1, pp. 29–48, January 2000. 1, 14, 19, 21, 78

[16] P. R. Roan, A. Burmeister, A. Rahimi, K. Holz, and D. Hooper, "Real-world validation of three tipover algorithms for mobile robots," in *Proc. IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, USA, May 2010, pp. 4431–4436. 1, 9, 13, 38, 78, 98

[17] C. C. Kessens, D. C. Smith, and P. R. Osteen, "A framework for autonomous self-righting of a generic robot on sloped planar surfaces," in *Proc. IEEE International Conference on Robotics and Automation*, Minnesota, USA, May 2012, pp. 4724–4729. 1

[18] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968. 3, 56, 71, 73

[19] K. Pathak, A. Birk, N. Vaskevicius, M. Pfingsthorn, S. Schwertfeger, and J. Poppinga, "Online 3d slam by registration of large planar surface segments and closed form pose-graph relaxation," *Journal of Field Robotics*, vol. 27, no. 1, pp. 52–84, 2010. 3, 24, 46

[20] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping (slam): part i," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 2, pp. 99–110, 2006. 3

[21] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (slam): Part ii," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 3, pp. 108–117, 2006. 3

[22] C. H. Tong and T. D. Barfoot, "A self-calibrating 3d ground-truth localization system using retroreflective landmarks," in *Proc. IEEE International Conference on Robotics and Automation*, Shanghai, China, May 2011, pp. 3601–3606. 3, 70

[23] G. Freitas, G. Gleizer, F. Lizarralde, and L. Hsu, "Kinematic reconfigurability control for an environmental mobile robot operating in the amazon rain forest," *Journal of Field Robotics*, vol. 27, no. 2, pp. 197–216, 2010. 9, 14, 48, 49, 51

[24] R. B. McGhee and A. A. Frank, "On the stability properties of quadruped creeping gaits," *Mathematical Bioscience*, vol. 3, pp. 331–351, August 1968. 10

[25] D. A. Messuri and C. Klein, "Automatic body regulation for maintaining stability of a legged vehicle during rough-terrain locomotion," *IEEE Journal of Robotics and Automation*, vol. 1, no. 3, pp. 132–141, September 1985. 10, 11

[26] S. Hirose, H. Tsukagoshi, and K. Yoneda, "Normalized energy stability margin: generalized stability criterion for walking vehicles," *Proceedings of the International Conference on Climbing and Walking Robots, Brussels*, 1998. 11

[27] J. Morales, J. L. Martinez, A. Mandow, J. Seron, and A. J. Garcia-Cerezo, "Static tip-over stability analysis for a robotic vehicle with a single-axle trailer on slopes based on altered supporting polygons," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 2, pp. 697–705, April 2013. 12

[28] P. Schenker, T. Huntsberger, P. Pirjanian, S. Dubowsky, K. Iagnemma, and V. Sujan, "Rovers for agile, intelligent traverse of challenging terrain," in *Proc. of the 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Nara, Japan, May 2003, pp. 1–10. 11, 44, 49, 50

[29] G. Besseron, C. Grand, F. B. Amar, and P. Bidaud, "Decoupled control of the high mobility robot hylos based on a dynamic stability margin," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, September 2008, pp. 22–23. 11, 14, 49, 50

[30] W. Wang, Z. Du, and L. Sun, "Dynamic load effect on tracked robot obstacle performance," in *Proc. of International Conference on Mechatronics*, Kumamoto, Japan, May 2007, p. 6. 11

[31] P. Ben-Tzvi, S. Ito, and A. A. Goldenberg, "Autonomous stair climbing with reconfigurable tracked mobile robot," in *Proc. IEEE International Workshop on Robotic and Sensors Environments*, Ottawa, Canada, October 2007, pp. 12–13. 11

[32] S. Shoval, "Stability of a multi tracked robot traveling over steep slopes," in *Proc. IEEE International Conference on Robotics and Automation*, LA, USA, April 2004, pp. 4701–4706. 11

[33] D. Inoue, K. Ohno, S. Nakamura, and S. Tadokoro, "Whole-body touch sensors for tracked mobile robots using force-sensitive chain guides," in *Proc. IEEE International Workshop on Safety, Security and Rescue Robotics*, Sendai, Japan, October 2008, pp. 71–76. 12

[34] S. Hirose, H. Tsukagoshi, and K. Yoneda, "Normalized energy stability margin and its contour of walking vehicles on rough terrain," in *Proc. IEEE International Conference on Robotics and Automation*, Seoul, Korea, May 2001, pp. 181–186. 12

[35] M. Vukobratovic, "Zero-moment point-thirty five years of its life," *International Journal of Humanoid Robotics*, vol. 1, no. 1, pp. 157–173, 2004. 13

[36] E. G. Papadopoulos and D. A. Rey, "A new measure of tipover stability margin for mobile manipulators," in *Proc. IEEE International Conference on Robotics and Automation*, vol. 4, Minneapolis, USA, April 1996, pp. 3111–3116. 13, 50, 106

[37] S. A. A. Moosavian and K. Alipour, "Stability evaluation of mobile robotic systems using moment-height measure," in *Proc. IEEE International Conference on Robotics, Automation and Mechatronics*, Bangkok, Thailand, December 2006, pp. 1–6. 13

[38] W. Chonnaparamutt and A. Birk, "A fuzzy controller for autonomous negotiation of stairs by a mobile robot with adjustable tracks," *RoboCup 2007: Robot WorldCup XI*, vol. 5001, no. 1, pp. 196–207, 2008. 14

[39] iRobot Corporation, "irobot 510 packbot," 2015. [Online]. Available: http://www.irobot.com/For-Defense-and-Security/Robots/510-PackBot.aspx 14, 31

[40] A. Chilian and H. Hirschmuller, "Stereo camera based navigation of mobile robots on rough terrain," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, USA, October 2009, pp. 4571–4576. 24, 44

[41] D. Gingras, E. Dupuis, G. Payer, and J. de Lafontaine, "Path planning based on fluid mechanics for mobile robots using unstructured terrain models," in *Proc. IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, USA, May 2010, pp. 1978–1984. 24, 44, 46

[42] M. Corporation, *Kinect sensor device*, 2010. [Online]. Available: http://www.xbox.com/en-US/kinect 24, 82

[43] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Journal of Sensors*, vol. 12, no. 1, pp. 1437–1454, January 2012. 24, 82

[44] F. M. Adolf and H. Hirschmuller, "Meshing and simplification of high resolution urban surface data for uav path planning," *Journal of Intelligent and Robotic Systems*, vol. 61, no. 4, pp. 169–180, January 2011. 24

[45] L. Jaillet, J. Cortes, and T. Simeon, "Sampling-based path planning on configuration-space costmaps," *IEEE Transactions on Robotics*, vol. 26, pp. 635–646, 2010. 24

[46] J. Snyder, "An interactive tool for placing curved surfaces without interpenetration," in *Proc. 22nd annual Conference on Computer Graphics and Interactive Techniques*. ACM, 1995, pp. 209–218. 24

[47] R. Smith, "Open dynamics engine, (http://www.ode.org/)," 2005. [Online]. Available: http://www.ode.org/ 24, 79

[48] A. S. Jacoff, A. Downs, A. Virts, and E. Messina, "Stepfield pallets: Repeatable terrain for evaluating robot mobility," in *Proc. of the Performance Metrics for Intelligent Systems Workshop*, Gaithersburg, MD, USA, December 2008, pp. 29–34. 28

[49] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, February 2007. 33

[50] K. Konolige and E. Marder-Eppstein, "navfn package," 2015. [Online]. Available: http://wiki.ros.org/navfn 34

[51] M. Norouzi, J. V. Miro, and G. Dissanayake, "Planning high-visibility stable paths for reconfigurable robots on uneven terrain," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Portugal, October 2012, pp. 2844–2849. 43, 44, 66, 68

[52] T. M. Howard and A. Kelly, "Optimal rough terrain trajectory generation for wheeled mobile robots," *The International Journal of Robotics Research*, vol. 26, pp. 141–166, 2007. 44

[53] J. Poppinga, A. Birk, and K. Pathak, "Hough based terrain classification for realtime detection of drivable ground," *Journal of Field Robotics*, vol. 25, no. 1-2, pp. 67–88, 2008. 44

[54] N. Vaskevicius, A. Birk, K. Pathak, and S. Schwertfeger, "Efficient representation in 3d environment modeling for planetary robotic exploration," *Advanced Robotics*, vol. 24, no. 8-9, pp. 1169–1197, 2010. 44

[55] D. Bonnafous, S. Lacroix, and T. Simeon, "Motion generation for a rover on rough terrains," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, Hawaeii, USA, October 2001, pp. 784–789. 47

[56] C. Beck, J. V. Miro, and G. Dissanayake, "Trajectory optimization for increased stability of mobile robots operating in uneven terrain," in *Proc. IEEE 7th International Conference on Control and Automation*, Chirstchurch, New Zealand, December 2009, pp. 1913–1919. 47

[57] S. Wirth and J. Pellenz, "Exploration transform: A stable exploring algorithm for robots in rescue environments," in *Proc. IEEE International Workshop on Safety, Security and Rescue Robotics*, Rome, Italy, 2007, pp. 1–5. 47

[58] V. Kumar and K. J. Waldron, "Force distribution in closed kinematic chains," *IEEE Journal of Robotics and Automation*, vol. 4, no. 6, pp. 657–664, December 1998. 51

[59] E. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959. 56

[60] C. H. Tong, D. Gingras, K. Larose, T. D. Barfoot, and E. Dupuis, "The canadian planetary emulation terrain 3D mapping dataset," *The International Journal of Robotics Research*, vol. 32, no. 4, pp. 389–395, 2013. 68, 69

[61] N. Nielsson, *Problem-Solving Methods in Artificial Intelligence*. McGraw-Hill, 1971. 71

[62] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004. 77, 87

[63] K. Iagnemma and S. Dubowsky, *Mobile Robot in Rough Terrain*. Berlin, Germany: Springer Tracts in Advanced Robotics, 2004, vol. 12. 79, 106

[64] S. Isukapalli, S. Balakrishnan, and P. Georgopoulos, "Computationally efficient uncertainty propagation and reduction using the stochastic response surface method," in *Proc. IEEE International Conference on Decision and Control*, Bahamas, December 2004, pp. 2237–2243. 79

[65] Rubinstein and R. Y, *Simulation and the Monte Carlo Method*. New York: Wiley, 1981. 79

[66] M. D. McKay, R. J. Beckman, and W. J. Conover, "Comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979. 79, 86

[67] M. D. Greenberg, *Advanced Engineering Mathematics, 2/E*. Pearson Education India, 1998. 85

[68] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robotos*. The MIT press, 2004. 86

[69] A. F. Smith and A. E. Gelfand, "Bayesian statistics without tears: a sampling-resampling perspective," *The American Statistician*, vol. 46, no. 2, pp. 84–88, 1992. 86

[70] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 477–482, March 2000. 87, 95

[71] G. Welch and G. Bishop, "An introduction to the kalman filter," 1995. 88

[72] H. S. A, V. D. H. G. WAM, V. E. F. K, S. Alfred, and T. B. C. JF, "Laser range finder model for autonomous navigation of a robot in a maize field using a particle filter," *Computers and Electronics in Agriculture*, vol. 100, pp. 41–50, 2014. 104

[73] B. J. Nsasi and P. Vladimir, "Autonomous system for navigation and surveying in underground mines," *Journal of Field Robotics*, vol. 24, no. 10, pp. 829–847, 2007. 104

[74] M. Joshua, B. Timothy, and L. Johan, "Autonomous underground tramming for center-articulated vehicles," *Journal of Field Robotics*, vol. 25, no. 6-7, pp. 400–421, 2008. 104

[75] S. SeungBeum, C. JunHo, C. ChangHyun, J. YeonSub, H. Seung-Yeup, and K. Sungchul, "Mine detecting robot system," in *Field and Service Robotics*. Springer, 2014, pp. 97–109. 104

[76] D. Liang, D. Zongquan, G. Haibo, G. Junlong, Z. Dapeng, and I. K. D, "Experimental study and analysis of the wheels' steering mechanics for planetary exploration wheeled mobile robots moving on deformable terrain," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 712–743, 2013. 104

[77] N. Keiji, K. Seiga, O. Yoshito, O. Kazuki, Y. Kazuya, T. Satoshi, N. Takeshi, Y. Tomoaki, K. Eiji, F. Mineo, *et al.*, "Emergency response to the nuclear accident at the fukushima daiichi nuclear power plants using mobile rescue robots," *Journal of Field Robotics*, vol. 30, no. 1, pp. 44–63, 2013. 104

[78] K. Gupta and A. P. Pobil, *Practical Motion Planning in Robotics: Current Approaches and Future Directions*. John Wiley & Sons, Inc., 1998. 104

[79] L. S. M and S. Michael, *Planning Algorithms*. Cambridge University Press, 2006. 104

[80] T. Sebastian, B. Wolfram, and F. Dieter, *Probabilistic Robotics*. MIT Press, 2005. 104

[81] P. Joelle, G. Geoff, T. Sebastian, *et al.*, "Point-based value iteration: An anytime algorithm for pomdps," in *IJCAI*, vol. 3, 2003, pp. 1025–1032. 104

[82] S. M. TJ and V. N. A, "Perseus: Randomized point-based value iteration for pomdps," *J. Artif. Intell. Res.(JAIR)*, vol. 24, pp. 195–220, 2005. 104

[83] B. Alex, M. Alexei, W. Stefan, and D.-W. Hugh, "Parametric POMDPs for planning in continuous state spaces," *Robotics and Autonomous Systems*, vol. 54, no. 11, pp. 887–897, 2006. 104

[84] C. Salvatore, D. James, and H. Seth, "Exploiting domain knowledge in planning for uncertain robot systems modeled as pomdps," in *Proc. IEEE International Conference on Robotics and Automation*, Alaska, USA, May 2010, pp. 3596–3603. 104

[85] V. D. B. Jur, A. Pieter, and G. Ken, "Lqg-mp: Optimized path planning for robots with motion uncertainty and imperfect state information," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 895–913, 2011. 105

[86] L. S. M, "Rapidly-exploring random trees: A new tool for path planning," Iowa State University, Dept. of Computer Science, Tech. Rep., 1998. 105, 119

[87] D. T. N. E and B. J. W, "Robot motion planning in dynamic, uncertain environments," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 101–115, 2012. 105

[88] C. Paulo and N. Urbano, "Path-following control of mobile robots in presence of uncertainties," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 252–261, 2005. 105

[89] R. Gill, D. Kulic, and C. Nielsen, "Robust path following for robot manipulators," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, Japan, November 2013, pp. 3412–3418. 105

[90] P. Parks, "Liapunov redesign of model reference adaptive control systems," *IEEE Transactions on Automatic Control*, vol. 11, no. 3, pp. 362–367, 1966. 105

[91] L. E. Kavraki, P. Svestka, J. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 556–580, 1996. 119

[92] L. S. M and K. J. J, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001. 119

[93] Z. Liangjun and M. Dinesh, "An efficient retraction-based rrt planner," in *Proc. IEEE International Conference on Robotics and Automation*, Pasadena, USA, May 2008, pp. 3743–3750. 122