

The Development of Mobile Robot Platform for Urban Search and Rescue Environment

Mohd Ismail Bin Yusof

Submitted for the degree of Doctor of Philosophy

Department of Automatic Control and System Engineering

September 2013

Supervisor: Dr Tony Dodd

The University of Sheffield

Abstract

Search and rescue (SAR) mission always takes place when disasters happen. Disaster could be defined into two categories, namely natural disaster and man-made disaster. Natural disasters normally cover a large area making the SAR mission's team require an aerial view from airplane. This is because it changes the geographical landscape of the affected areas in huge perimeters. The impact is not only changing the whole landscapes, but it also impacts on residences, commercial buildings, transportations and communication infrastructures. This is always the primary reason of choosing an air vehicle as a first respond for any natural disaster. Meanwhile, made-made disasters occur in small areas relative to natural disaster. Terrorist bombing, structural collapse because of human failures or serious accident are some examples of man-made disaster. In addition, the effect from natural disaster such as earthquake also resulting horrendous structural collapse. The challenges for this rescue operation are focused on the interior of the rubble and entire external extent of the damage often not as primary interest because most victims are trapped inside, under the rubbles. Locating, extracting and rescuing any survivors will become the main goal for any rescue mission. Besides, the mission also deals with a lot of potentially dangerous situation such as further collapse, explosions, hazardous gas leaks and fire. Extreme high temperature from fire or explosions prevent rescuer to go down further into rubbles. Urban search and rescue (USAR) is the term that is being used recently for the rescue operation after man-made disaster. Conventionally, dog has been used to identify location of any potential survivors in the rubble. Again, capabilities of dog rescuer are restricted by certain working temperature, uncertainty of void size and fatigue factor. USAR operation is like race against time where trapped survivors cannot wait any longer. Further collapse or explosion may happen anytime. Therefore, the rescue team should have ideal strategy and tactic in order to maximize numbers of survivors being extract from rubble but also minimise the risk face by rescuer. Hazards is everywhere at disaster site. Human rescuers as well as dog are exposed to danger such as further collapse which would trap them in rubble and resulting an increased number of victims. This kind of situation make USAR uncertain. As a result, hazard identification and situation awareness have to be conducted concurrently with finding survivors. Robotic system, in many ways, have shown its versatility in wide range of applications. For instance, a modern and sophisticated automative assembly plant employ robotic systems in the production line in order to fullfil a specific part assembly task. On the other hand, robotic systems also started to be used as exploration vehicle in unknown world such as deep sea and outer space exploration purposes. In many aspects, the implementation of robotic systems in these applications have a significant

impact to the overall process flow of the specific application. Having said that, mobile robots use in many deep sea explorations help scientists to discover the 'underworld' where human cannot explore. Therefore, implementation of robotic system in USAR operation is inevitable. In fact, it has been used in several USAR operation including the 9/11 World Trade Centre tragedy and Fukushima Daiichi Nuclear Power Station. The physical design of mobile robot is one of the main challenges to implement robotic system in USAR operation. The ability to manouver and negotiate with rough terrain is highly essential. In addition, the physical design is also need appropriate sensors in order to sense the environment. Therefore, the overall mechatronic structure must consist a robust platform equipped with sensors and actuators and able to navigate seemlessly on extreme rough terrain as well as perform designated task (e.g., find survivors, clean debris or conduct onsite disaster assessment). In order for a mobile robot to operate in unknown world such as USAR environment, it is crucially important for the mobile robot to have certain level of autonomy to plan a desired behaviour and act according to the surrounding. Eventhough it is very challenging to program a mobile robot for this type of environment, a comprehensive control architecture provide a systematic overview of the overall programming structure whilst simplifiying the programming procedure. On top of that, the ability of the overall robot system to plan and track its mission is clearly present in the control architecture. Navigation problem, which is one of the common problem in any exploration robot, also can be solved systematically. In general navigation task, a mobile robot is required to move according to the prior designated trajectory, normally in 2-dimension (flat surface). However, a mobile robot that is design for the USAR operation should be able work and navigate in unknown, uncertain and complex environment. This thesis describes the development of a mobile robot system motivated by the shape-shifting or variable geometry tracked vehicle (VGTV) configuration. The mobile robot is designed with expectation to be able to traverse on various types of terrain and enhance stability to prevent tip-over mishap. The practical work is evaluated by experimental trials on prepared terrains such as staircase, ramp and curb. On top of that, the control framework is outlined to set the objectives of the mobile robot system based on the control hierarchy. This set of works is further simulated with the aim to solve navigation problems as well as to determine the mobile robot behaviour when it is required to travel on uneven surface.

Acknowledgement

First of all, I would like to thank my supervisor, Dr. Tony Dodd, for giving me the continuous support during my PhD study at The University of Sheffield. I would not be able to finish my work without his support. There are another two people whom I would like to thank for their help and support: Dr. Rene Moreno, who helped me a lot in my first year, and later; Dr. Yuanming Zhang who was a true advisor and helper in all parts of my research. Thanks go also to the other members of the research group with whom I shared a room. I owe my greatest debt to my family in Malaysia, especially my parents, who taught me how to live, and to my wife, my little childrens who have given me love and support throughout my time here in Sheffield.

Contents

LIST OF TABLES iv
LIST OF FIGURES ix
LIST OF ALGORITHMS x
NOMENCLATURE xiii

1 Introduction **1**
1.1 Urban Search and Rescue - The Background 4
1.1.1 Conventional USAR Operation 4
1.1.2 Robotics in USAR 6
1.1.3 The Challenges 9
1.2 Literature Review 11
1.2.1 Mobile Robot Physical Design 12
1.2.2 Control Framework Architecture 13
1.2.3 Motion Planning and Tracking 14
1.3 Aims 15
1.4 List of Novel Contributions 16
1.5 Context and Thesis Outline 17

2 Mechatronic Design of Variable Geometry Tracked Vehicle with Independent Track Control **20**
2.1 Introduction 20
2.1.1 Motivations 21
2.1.2 Aim 25
2.1.3 Methodology 27
2.2 Geometric Formulation 29
2.2.1 Track geometry and configuration control 30
2.2.2 Comparison of track length variation 32
2.2.3 Formulation of the relationship between leadscrew and θ_{cl} 35

2.3 Mechatronic Design . . . 38
2.3.1 Mechanical structure . . . 38
2.3.2 Lowest Control Layer . . . 40
2.4 VGTV Transformation . . . 44
2.5 Discussion and Concluding Remarks . . . 51

3 Control Architecture **56**
3.1 Introduction . . . 56
3.2 Motivations . . . 57
3.3 Aim . . . 58
3.4 Control Structure Requirements . . . 59
3.5 Three-Tiered Layer Control Architecture . . . 59
3.5.1 Highest Control Layer (Mission Control Layer) . . . 60
3.5.2 Intermediate Control Layer (Position Control Layer) . . . 63
3.5.3 Lowest Control Layer (Actuator Control Layer) . . . 67
3.6 Concluding Remarks . . . 69

4 Approximating Kinematics of Mobile Robot Motion on Uneven Surfaces **72**
4.1 Introduction . . . 72
4.1.1 Motivations . . . 73
4.1.2 Aim . . . 75
4.2 Kinematic Model Formulation . . . 76
4.2.1 Mobile robot motion on uneven surface . . . 77
4.2.2 Cartesian coordinate system . . . 79
4.2.3 Trajectory generator . . . 83
4.2.4 Multivariate Gaussian Surface . . . 85
4.2.5 Trajectory on uneven surface . . . 87
4.3 Kinematic Analysis . . . 89
4.3.1 Unit vector acting along the pre-planned trajectory . . . 90
4.3.2 Tangent vector along the pre-planned trajectory . . . 90
4.3.3 Normal unit vector along the pre-planned trajectory . . . 92
4.3.4 Binormal unit vector along the pre-planned trajectory . . . 95
4.3.5 Rolling, Climbing and Turning Angles . . . 97
4.3.6 Kinematic Modeling of a Mobile Robot on Pre-planned Trajectory . . . 100
4.4 Simulation Results . . . 104
4.5 Discussion and Concluding Remarks . . . 106

5 Trajectory Tracking Control of Mobile Robot on Uneven Surface 120

5.1 Introduction . . . 120

5.1.1 Motivations . . . 122

5.1.2 Aim . . . 122

5.2 Problem Statement . . . 123

5.3 Control Strategy . . . 125

5.4 Derivation of Trajectory Tracking Error Model . . . 128

5.4.1 Kalman Filter to estimate mobile robot configuration, q_{est} . . . 130

5.4.2 Error Dynamics . . . 135

5.4.3 Control Based Approximate Linearisation . . . 135

5.5 Simulation Results . . . 137

5.6 Discussion . . . 162

5.7 Concluding Remarks . . . 163

6 Conclusions and Future Work 165

6.1 Summary . . . 166

6.2 Future Work . . . 168

Bibliography 169

List of Tables

2.1 Fixed design parameters for the proposed mobile robot. 29

2.2 The difference between maximum and minimum value of track length for each combination in Figure 2.9. 32

4.1 Comparison between holonomic and non-holonomic mobile robot system. 74

4.2 List of assumptions for high-degree-of-freedom kinematic model derivation. . . . 90

4.3 Quaternion multiplication formula. 97

5.1 List of assumptions for trajectory tracking. 130

5.2 Simulated 2-dimensional pre-planned trajectories used in the simulation environments. 138

5.3 List of figures that correspond to each sub-simulation of trajectory tracking. . . . 139

List of Figures

1.1 Robots on stock for non-industrial applications in year 2006 and new installations between 2007 to 2010 (forecasted). 2

1.2 Professional service robot systems sold in 2010 and 2011. 3

1.3 Professional service robot system unit forecast sale for 2012 to 2015. 3

1.4 A view of a collapsed building following disaster. USAR operation for a big scale disaster site is extremely difficult because of unstructured obstacles. This USAR site was piled-up by steel beams and concrete slabs. 5

1.5 Three types of common rubble and arrangement of obstacles in disaster environment. (Adapted from Murphy et al. (2000, 2001)). 6

1.6 Rescue team for USAR operation of a partially collapsed building consists of several people. Every rescuer has their specific role based on tools and equipment use. Dog is also being use to detect the presence of human under rubble. 7

1.7 Typical workflow during USAR operation. 7

1.8 USAR reference test arenas developed by NIST. 11

1.9 General robot classification. 12

2.1 A basic configuration of double-track VGTV. 23

2.2 PackBot from iRobot is an example of a double-track VGTV and widely use in military operation. 23

2.3 Another double-track VGTV configuration. 24

2.4 Inuktun VGTV is an example of single-track VGTV. 24

2.5 Three-dimensional front view of the VGTV travels along a side slope. 26

2.6 2-dimensional side view of climbing principle of the proposed VGTV mobile robot. 28

2.7 Two-dimensional side views of a VGTV with climbing configuration. 31

2.8 Explanation of track arm with ten unit length. 33

2.9 Comparison of track length changes with different combination of $m : n$. 34

2.10 Climb configuration with parameter of δ. This parameter will be used in order to keep the track length constant. 35

2.11 Track length geometry modeling. 36
2.12 Mechanical structure of the prototype. 39
2.13 Two basic body actuation resulted by the adjustment of the leadscrew mechanism. 41
2.14 Leadscrew mechanism. 42
2.15 Overall mobile robot control architecture to study the behaviour of a mobile robot on an uneven surface. 42
2.16 Actuators, control unit and onboard sensor. 43
2.17 Sample IMU image. 43
2.18 Back panel connector ports. 44
2.19 Experimental setup. 45
2.20 The proposed VGTV mobile robot is ascending a simple ramp with 20 deg slope. 46
2.21 IMU reading related to the ramp obstacle motion in Figure 2.20. 47
2.22 The proposed VGTV mobile robot was overcoming a curb with symmetrical climbing configuration. 47
2.23 IMU reading related to the curb overcoming motion in Figure 2.22. 48
2.24 The proposed VGTV mobile robot was ascending a flight of stair. 49
2.25 IMU reading related to ascending a staircase. 50
2.26 The proposed VGTV mobile robot was descending a staircase. 51
2.27 IMU reading related to descending a staircase. 52
2.28 The proposed VGTV mobile robot was 3-dimensional complex terrain (a rubble). 52
2.29 IMU reading related to three-dimensional complex terrain (a rubble). 53
2.30 The proposed VGTV mobile robot is traveling along a side slope. 53
2.31 IMU reading related to side-slope motion. 54

3.1 General structure of *three-tiered layer control architecture*. 61
3.2 The level of autonomy scale. 62
3.3 An example of a mobile robot required to travel on unknown terrain, given the start and final position. 63
3.4 Mission control block diagram. 64
3.5 The mobile robot detect a survivor trapped inside that rubble and need to approach the survivor before moving towards the final position. 64
3.6 Change of the *global mission objective*. 64
3.7 Position control block diagram. 66
3.8 An extended version of the example given in Figure 3.3. In this example, the mobile robot system is force to think the suitable method to travel from given start and final position. 67

3.9 Flowchart of the *position control layer*. 68
3.10 Actuator control block diagram. 70
3.11 The mobile robot is expected to adjust the θ_{cl} due to side-slope. 71

4.1 A disk rolling without slipping on a flat plane. 74
4.2 Examples of surface condition; flat, incline and uneven surface; within global frame (X_g, Y_g, Z_g). 74
4.3 Overall mobile robot control architecture to study the behaviour of a mobile robot on an uneven surface. 77
4.4 Flowchart of kinematic model derivation of a mobile robot on a surface model. . 78
4.5 A mobile robot traveling on pre-planned path on 3-dimensional surface. ICR is acronym for *instantaneous centre of rotation* 79
4.6 Kinematic analysis of a mobile robot on a flat surface adapted from Fahimi (2009). Subscript g and l denoted 'global' and 'local' axis repectively. 80
4.7 The surface generator models a physical uneven terrain (e.g, stacked-bricks) to an equivalent multivariate Gaussian surface. 85
4.8 The projection of pre-planned path onto the terrain model. 88
4.9 Vector fields along the desired trajectory. 89
4.10 Illustration of a tangent, $\widehat{T}$, normal, $\widehat{N}$, and binormal vector, $\widehat{B}$, on uneven surface model. 91
4.11 Gradient of the surface is the gradient vector that orthogonal to the level curve. . 93
4.12 Unit vectors on 3-dimensional Cartesian space. 98
4.13 Formulation of rolling angle, ψ_g. 98
4.14 Free-body-diagram of mobile robot on 3-dimensional space. Kinematic analysis of mobile robot on x, y, z planes. Subscript G and L are for 'global' and 'local' repectively. Mobile robot is replaced by a 'box' to simplify illustration. 99
4.15 Projection of vector v_p onto $(XY)_g$- axis from Figure 4.14a. 105
4.16 Effect of ψ_i and ψ_f in Cubic Cartesian. 106
4.17 Effect of free parameter, k, in parametric trajectory with $(x_i, y_i) = (0, 0)$ and $(x_f, y_f) = (10, 10)$. 107
4.18 Four random simulated uneven surface model. 108
4.19 Gradient vector on simulated uneven surface model 1. 109
4.20 Gradient vector on simulated uneven surface model 2. 110
4.21 Gradient vector on simulated uneven surface model 3. 111
4.22 Gradient vector on simulated uneven surface model 4. 112
4.23 3-dimensional trajectory on uneven surface models. 113

4.24 Position configuration on surface models. 114
4.25 Generalised vectors on surface. 115
4.26 Rotational motion. 116
4.27 Generalised velocity vectors. 117

5.1 Illustrations of mobile robot motion purely based on the desired velocities control. 121
5.2 General flowchart of mobile robot trajectory tracking problem. 121
5.3 Geometric formulation of trajectory tracking for non-holonomic mobile robot cruising on a flat surface. 123
5.4 Problem statement for trajectory tracking of mobile robot on uneven surface. . . 126
5.5 Mobile robot trajectory tracking control block diagram. 128
5.6 Simulated 2-dimensional pre-planned paths defined in path planner. 140
5.7 Surface models representation of actual terrains. 141
5.8 Noisy measurement of circular path on random generated uneven surfaces. 143
5.9 2-dimensional view of Figure 5.8. 144
5.10 Tracking the circular trajectory on uneven surfaces. 145
5.11 2-dimensional view of Figure 5.10. 146
5.12 Noisy measurement of figure-eight path on uneven surfaces. 147
5.13 2-dimensional view of Figure 5.12. 148
5.14 Tracking the figure-eight trajectory on uneven surfaces. 149
5.15 2-dimensional view of Figure 5.14. 150
5.16 Noisy measurement of spline path on uneven surfaces. 151
5.17 2-dimensional view of Figure 5.16. 152
5.18 Tracking the spline trajectory on uneven surfaces. 153
5.19 2-dimensional view of Figure 5.18. 154
5.20 Comparison of trajectory tracking on an uneven surface with different initial condition configuration 1. 155
5.21 Comparison of trajectory tracking on an uneven surface with different initial condition configuration 2. 156
5.22 Comparison of trajectory tracking on an uneven surface with different initial condition configuration 3. 157
5.23 Trajectory tracking on an uneven surface with $0 < \zeta < 1$. 158
5.24 Trajectory tracking on an uneven surface with $\zeta = 1$. 159
5.25 Trajectory tracking on an uneven surface with $\zeta > 1$. 160
5.26 Error dynamics response (average) vs damping coefficient, ζ. 161

List of Algorithms

5.1 Algorithm for trajectory tracking error control via approximation linearisation with Kalman filter estimation. 129
5.2 Kalman Filter Algorithm as described in Welch and Bishop (1995). 135

Nomenclature

$(x, y, \psi)_{act}$	Actual configuration with respect to global frame.
$(x, y, \psi)_{des}$	Desired configuration with respect to global frame.
$\alpha_x, \alpha_y, \beta_x, \beta_y$	Boundry condition.
$\ddot{x}, \ddot{y}, \ddot{z}$	Second derivative of cubic polynomials.
δ	Adjustment parameter to keep track length constant
δ	Adjustment parameter to keep track length constant
$\dot{\phi}(s)$	First derivative of turning angle
$\dot{\psi}(s)$	First derivative of rolling angle.
$\dot{\theta}(s)$	First derivative of climbing angle.
$\dot{q}(s)$	Mobile robot configuration.
$\dot{q}_d$	Desired mobile robot configuration.
$\dot{q}_m$	Measured mobile robot configuration.
$\dot{q}_{act}$	Actual mobile robot configuration.
$\dot{q}_{des}$	Desired mobile robot configuration.
$\hat{N}_x, \hat{N}_y, \hat{N}_z$	Unit normal vector interm of xyz direction.
$\hat{T}$	Unit tangent vector.
$\hat{T}_\perp$	Result of rotated tangent vectors which leads to rolling angle parameter, ψ.
$\|T_{2d}\|$	Length of the tangent vector $\vec{T}_{2d}$.
$\|T_{3d}\|$	Length of the tangent vector $\vec{T}_{3d}$.
ϕ_{3d}	The rolling angle of mobile robot on uneven surface.

ψ_f	Final heading angle.
ψ_i	Initial heading angle.
e	Exponential.
σ_x, σ_y	Spread of the Gaussian bump.
θ_g	Gaussian bump orientation.
θ_{3d}	The climbing angle of mobile robot on uneven surface.
θ_q	Quaternion rotate angle [rad].
$\upsilon_{d_{r,l}}$	Desired velocities of mobile robot for right and left track.
$\upsilon_{m_{r,l}}$	Measured velocities of mobile robot for right and left track.
$\vec{N}$	The unit normal vector
$\vec{N}_x$, $\vec{N}_y$ and $\vec{N}_z$	The normal vectors in the direction of x, y, z
$\vec{T}_{2d(x,y)}$	Tangent vector components in 2-dimensional Cartesian space in X and Y-direction respectively.
$\vec{T}_{2d}$	Tangent vector in 2-dimensional Cartesian space.
$\vec{T}_{3d(x,y,z)}$	Tangent vector components of $\vec{T}_{3d}$ in 3-dimensional Cartesian space in X, Y and Z-direction respectively.
$\vec{T}_{3d}$	Tangent vector in 3-dimensional Cartesian space.
$\|\vec{N}\|$	Unit normal vector length.
$\widehat{B}$	Binormal vector of uneven surface at an arbitrary point.
$\widehat{N}$	Normal vector of uneven surface at an arbitrary point.
$\widehat{T}$	Tangent vector of uneven surface at an arbitrary point.
$\widehat{T}_{2d(x,y)}$	Unit tangent vector for $\left(\vec{T}_{2d}\right)_x$ and $\left(\vec{T}_{2d}\right)_y$ respectively.
$\widehat{T}_{3d(x,y,z)}$	Unit tangent vector for $\left(\vec{T}_{2d}\right)_x$ and $\left(\vec{T}_{2d}\right)_y$ respectively.
A	Amplitude/peak of the Gaussian bump.
a_{tp}, b_{tp} and c_{tp}	The unit normal vector in (x, y, z)-direction
$e, \dot{e}$	Error and first derivative of error of mobile robot posture configuration.

e_q	Configuration error.
$e_{x,y,\psi}$	Configuration error with respect to global frame.
i	*for* loop to generate representation of obstacles as multi-Gaussian terrain.
j	*for* loop to compute geometric path based on number of sample, N.
k	Free parameter that precisely influence the pre-planned path. If $k \geqslant 0$, mobile robot is in forward motion and if $k < 0$ mobile robot is in backward motion.
N	Numbers of sampling.
$numG$	Maximum numbers of Gaussian bump.
P	A unit vector
P_{rot}	Rotated vector by quaternion
Q	Quaternion.
$q(s)$	Pre-planned geometric path.
$Q*$	Quaternion conjugate.
$Q_{\hat{N}}$	Unit quaternion for normal vector.
$Q^*_{\hat{N}}$	Unit quaternion conjugate for normal vector.
R_{T_i}	Rotated tangent vector in complex term i.
R_{T_j}	Rotated tangent vector in complex term j.
R_{T_k}	Rotated tangent vector in complex term k.
s	Timing law of sampling time.
X, Y	Global coordinate on x, y-axis.
x_f, y_f	Final value in [meter] of coordinate (x, y)
x_i, y_i	Initial value in [meter] of coordinate (x, y)
x_0, y_0	Coordinate of the centre of the Gaussian bump.

Chapter 1

Introduction

For the last four decades, robotic systems have evolved and attracted many researchers around the world to study a wide range of robotics problems and their potential applications. As found in Garcia et al. (2007), robotics research covers a wide range of topics such as mechanics, electrical, electronics, computer systems, robot perception and even bio-mechanics. Moreover, the real-world application of robotic systems has evolved from "traditional" industrial robot manipulator arms in assembly plants (e.g. car manufacturing plant) to high precision robot arms for delicate applications such as surgery. On the other hand, the evolution has also impacted the application of non-industrial robotic systems (also called service robots) such as explorer mobile robot, which ranges from sophisticated exploration rovers to small and portable household vacuum cleaner mobile robots.

According to Prassler et al. (2000) and Wyrobek et al. (2008), non-industrial robotic systems can be further identified into two categories: (1) professional service robots; and (2) household/domestic service robots. For professional service robots applications cover a wide area such as underwater, cleaning, defense, rescue and security, construction and demolition, hostile fields, medical, logistic and others. Meanwhile, examples for household/domestic service robot are automatic lawnmower and vacuum cleaner mobile robot. The versatility of non-industrial robotic systems applications and their evolution have a very high impact in human societies. Moreover, it is becoming increasingly difficult to avoid the importance of mobile robots in human tasks (Dillmann, 2004). In this thesis, the possible physical design of professional service robot systems especially for search and rescue operations is investigated. In addition, mathematical model and control framework for mobile robot traversing on rough terrain is presented later in the thesis.

According to the statistics reported by International Federation of Robotics (Bob Struijk, 2012), there are at least 110,000 professional service robots used around the world in the period 1998 to 2011. This estimate, however, includes units that are no longer usable (damaged) and

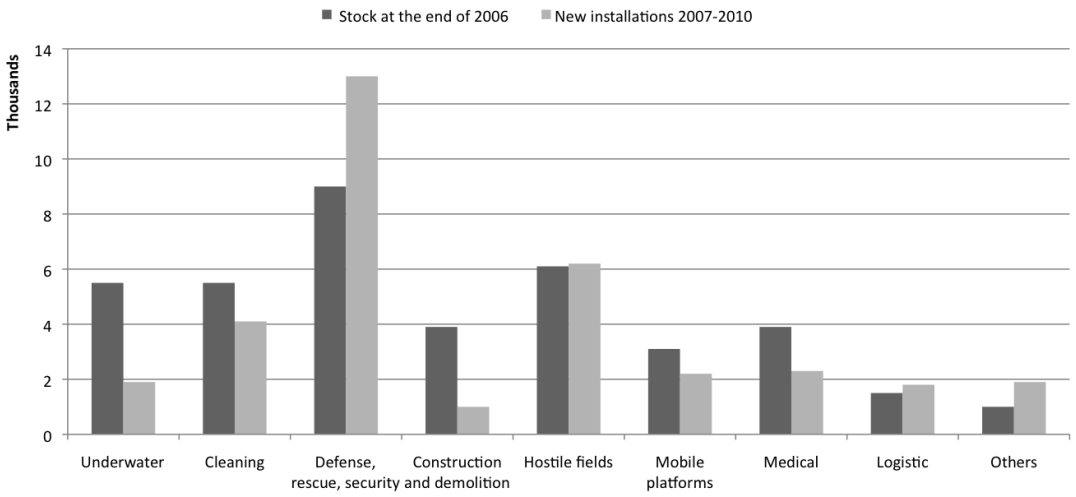


Figure 1.1: Robots on stock for non-industrial applications in year 2006 and new installations between 2007 to 2010 (forecasted). Reproduced from Siciliano et al. (2009).

outdated. Moreover, it is very difficult to count the number of service robots that are still in operation from this estimate because of the diversity of service robot applications. For instance, a service robot used in defense operation (military) is expected to have shorter lifespan due to the nature of its operation. On top of that, rapid technology development makes it very difficult and not cost effective to integrate latest technology with older military robot system (Bekey, 2012).

In addition to the previous statistic, the International Federation of Robotics also reported that the sales value of professional service robot systems is increasing every year (Siciliano et al., 2009). Figure 1.1 illustrates the number of service robots in operation at the end of 2006 (darker grey). Moreover, the chart also shows the number of service robots that was projected to be installed in 2007 to 2010 (lighter grey). It is very clear that service robots for defense, rescue and security operations dominate the data. Meanwhile, the latest report indicates that the sales value in year 2011 has increased by six percent (USD 3.6 billion) from year 2010. The number of service robots sold between 2010 and 2011 is depicted in Figure 1.2. Again, with about forty percent of the total sales in year 2011, service robots in defense, rescue and security operation has the highest number of service robots sold among other applications.

It is estimated that about 28,000 units robot system for defense, rescue and security application is forecasted to be sold for the period 2012 to 2015 (chart in Figure 1.3). Therefore, it is very important to design and develop new mobile robot systems for this operation to fulfill the high demand. Coincidently, designing mobile robots for this particular operation requires additional capability to negotiate with rough terrain that is also one of the aims in this thesis.

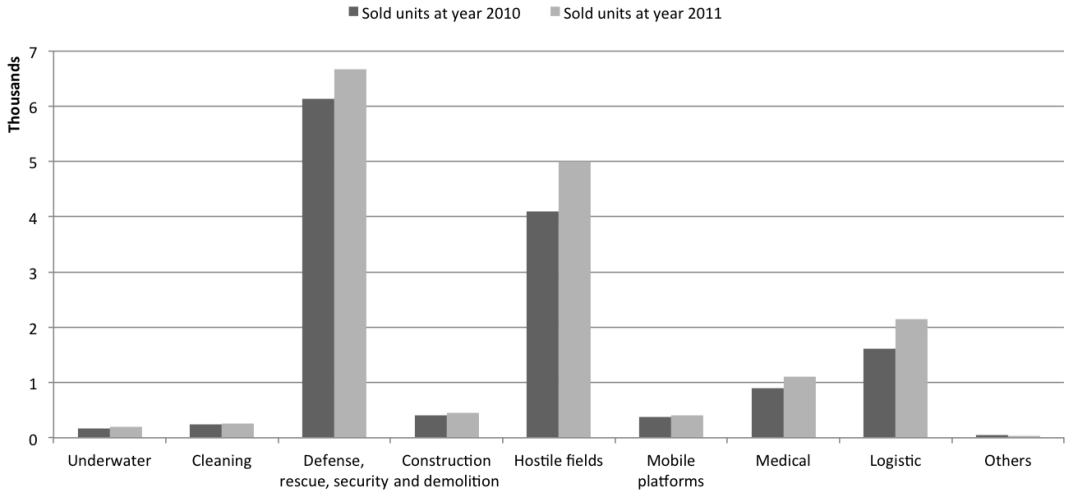


Figure 1.2: Professional service robot systems sold in 2010 and 2011. Reproduced from International Federation of Robotics (2012).

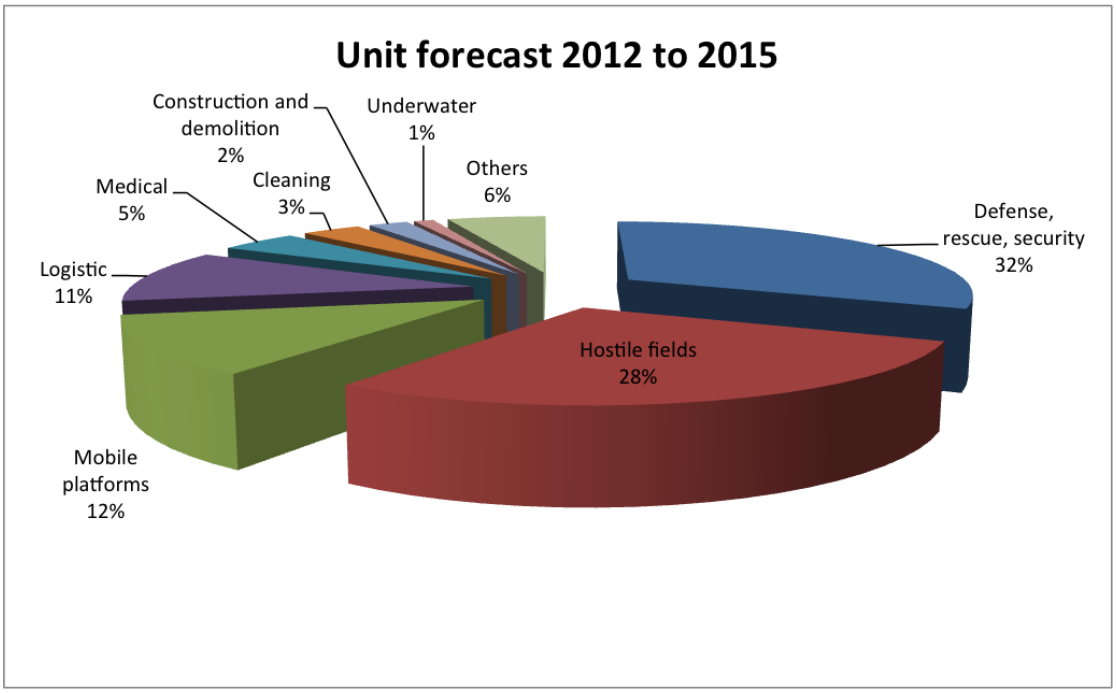


Figure 1.3: Professional service robot system unit forecast sale for 2012 to 2015. Reproduced from International Federation of Robotics (2012).

1.1 Urban Search and Rescue - The Background

it was noted earlier in this chapter that the demand for service robots for defense, rescue and security applications has increased since the year 2006. Moreover, the demand is projected to increase by at least another thirty-two percent from the total projections of 2012 to 2015 (International Federation of Robotics, 2012). In this thesis, however, the focus is to investigate the potential design of a mobile robot system to negotiate with rough terrains especially for search and rescue operation. It is very common for both applications (defense and search & rescue) to use similar mobile robot systems due to numerous similarities of working environment (Niu et al., 2013). Mobile robots such as Unmanned Ground Vehicle (UGV), Unmanned Air Vehicle (UAV), Unmanned Underwater Vehicle (UUV) or Unmanned Surface Vehicle (USV) are being used as a tool for many specific tasks in defense applications. In contrast, the selection of mobile robot for search and rescue operation mainly depends on the nature of disaster (Lima, 2012).

According to Kitano et al. (1999), a search and rescue operation can be described as locating and extracting any survivor after disaster. In general, disaster can be divided into two major areas which are natural disaster and man-made disaster. Search and rescue operation for natural disaster such as earthquake and typhoon require UAV as a first responder. This is because the impact of natural disaster could change the geographical landscape at huge perimeter and UAVs are able to map the whole area (exterior mapping) as well as to locate survivors. Meanwhile, the primary interest for man-made disaster, such as collapsed building or fire situation, is towards the interior of rubble or building (Figure 1.4). This is because most of the victims trapped inside buildings are under rubble. As a result, UGVs are a common mobile robotics platform used by rescue community as soon as disaster happens (Carlson and Murphy, 2005). Additionally, search and rescue operation for man-made disaster is widely known as Urban Search and Rescue (USAR) (Casper and Murphy, 2003). This section will review a conventional method for USAR activity and the potential to implement mobile robot system in USAR operation.

1.1.1 Conventional USAR Operation

Search and rescue operation for USAR environment can also be defined as locating a trapped survivor and then extricate them to safer location for further medical attention (Barbera and Lozano, 1993). Conventionally, rescue personnel employ specially trained dogs and/or detection equipment to sense any sign of human presence under identified collapsed structure. Then, rubble is carefully moved to provide access for a rescue team to move survivors out of the rubble. Even though the whole process sounds quite simple, the actual search and rescue operation is a “race



Figure 1.4: A view of a collapsed building following disaster. USAR operation for a big scale disaster site is extremely difficult because of unstructured obstacles. This USAR site was piled-up by steel beams and concrete slabs. Reproduced from Rijans (2013) under Creative Common license.

against the time" (Murphy et al., 2009). This means that the rescue team has to act as fast as possible to locate survivors under rubble because survivors cannot wait any longer. At the same time, they also need to take several precautions to avoid any accident; such as gas explosion or rescuer trap under rubble due to further collapse, that will result in severe injury or even fatality amongst rescuer team (Jacoff et al., 2002).

In general, according to Murphy et al. (2000, 2001), rubble can be categorised in three categories as depicted in Figure 1.5. Even though it is not compulsory to follow this classification, at least it provides motivation and general guidance to develop an appropriate mobile robot system for USAR application. In the next section, the existing design of mobile robot system used for USAR operation is presented in detail. Furthermore, the possible control framework as well as the control algorithm is investigated that will lead to fully autonomous ground rescue mobile robot.

Figure 1.6 shows a conventional and basic approach to locate and extricate process which is currently being practised. In the operation, a rescue team consists of a number of personnel with different functions and capabilities. Besides, specifically trained rescue dogs are used as their immediate responder. These specifically trained rescue dogs will use their powerful scent capability to detect any presence of human life. They will sniff between rubble and sometimes go



(a) Pancake. (b) Lean-to. (c) V-shape.

Figure 1.5: Three types of common rubble and arrangement of obstacles in disaster environment. (Adapted from Murphy et al. (2000, 2001)).

deeper through void and bark in case there is a sign of trapped victim. Instead of using 'body-sniffing dog' approach, rescue personnel use equipment such as listening and imaging devices to locate trapped victims. Listening devices such as seismic waves are able to detect presence of human life by matching 'banging fingerprint' wave produce by trapped victims provided that the victim produces a series of banging sound. Besides that, imaging devices such as camera or thermal imaging sensor is attached on a probe and then inserted into rubble to provide some form of visual display. Both devices provide an important set of data on the exact location of trapped victims. This data is critically important for further extrication procedure. The workflow of conventional search and rescue operation is summarised in Figure 1.7. The implementation of mobile robot system into conventional USAR operation will be define further in this section. Moreover, challenges to integrate mobile robot system in actual disaster area and initiatives effort by research community to increase USAR mobile robot efficiency will be discussed.

1.1.2 Robotics in USAR

To date, many types of robot have been proposed for USAR (Osuka et al., 2002). Robots have started to be used in USAR to assist human rescuers to search for trapped survivors, search any possible path through rubble for excavation, structural inspection and detection of hazardous material (Murphy, 2004). Smaller robots could enter small size void that human or search dog cannot access. Besides that, mobile robot could explore high temperature area that exceeds biological rescuer's maximum working temperature, provided that the robot's components have a high working temperature. The use of robots also minimises the risk of human rescuer being trapped if there is any further collapse.

The integration of mobile robotic systems into USAR operations presents significant challenges for the search and rescue community (Murphy et al., 2000; Murphy and Burke, 2005). Amongst the most important criteria for mobile robot in USAR application are capabilities to

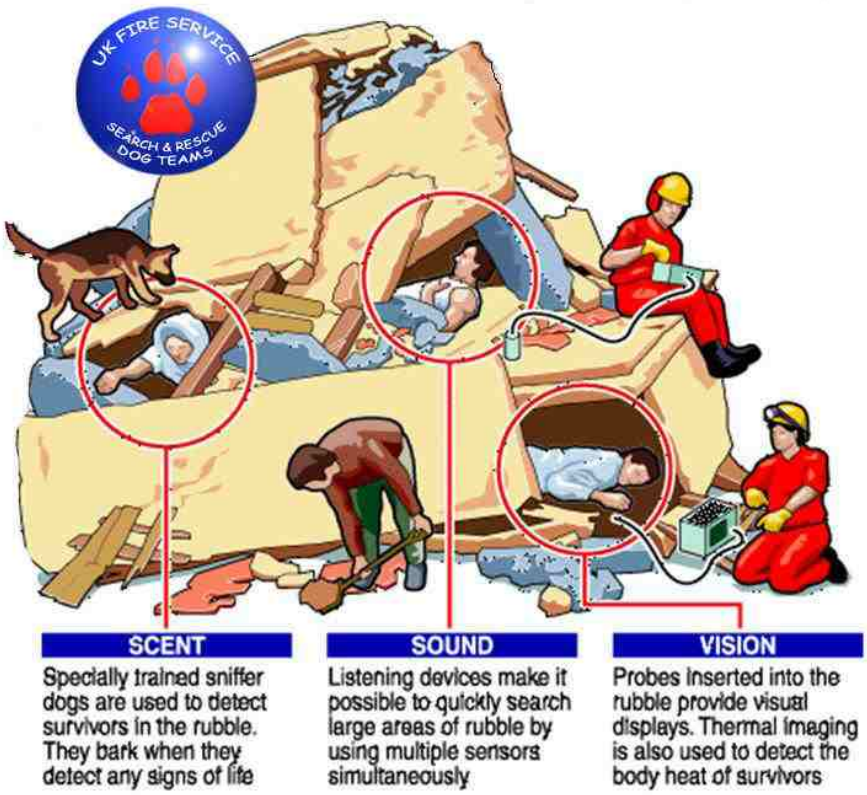


Figure 1.6: Rescue team for USAR operation of a partially collapsed building consists of several people. Every rescuer has their specific role based on tools and equipment used. Dog is also being used to detect the presence of human under rubble. Reproduced from http://www.ukfssartdogteams.org.uk/operational_information.htm

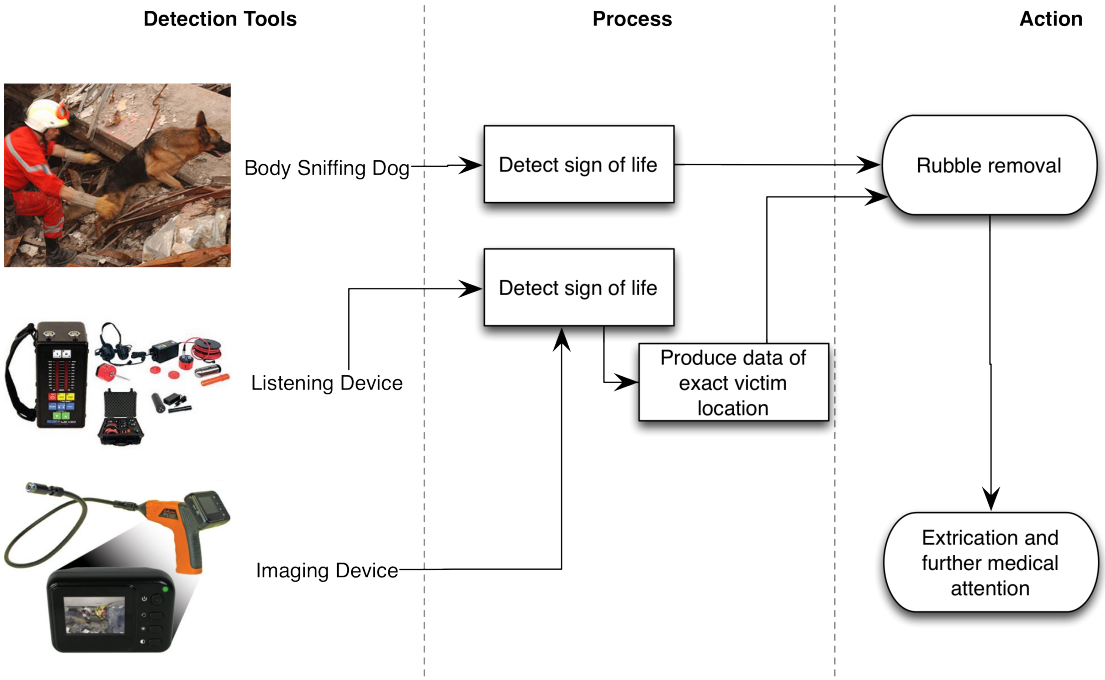


Figure 1.7: Typical workflow during USAR operation.

explore in very unstructured, uneven, and uncertain environments. For example, the disaster area shown on Figure 1.4 is piled-up by steel beams and concrete chunks making search and rescue operation extremely difficult and dangerous. Risks such as further collapse can happen anytime if rescue personnel accidentally cut unstable steel beam during extrication process. Furthermore, explosions can also happen in case of fuel and fire factor. As a result, integration of mobile robotic systems has been chosen to reduce those risks. However, the main idea of this integration is to increase the efficiency of overall operation rather than to totally replace the human rescuer.

It is possible to identify the key challenges and open issues in this application by looking at the application of mobile robot in USAR operation mission broadly. A general set of missions for mobile robots in USAR operation can be defined into ten tasks (Murphy et al., 2008):

1. *Search*: It is well known that the primary task of any USAR operation is to search for trapped victim by detecting any sign of life.

2. *Reconnaissance and mapping*: It is broader than search where the robot should penetrate deeper into rubble and map the interior environment of collapsed structure.

3. *Structural inspection*: This task is done by providing visual information and image processing data so that human operator knows stability level of collapsed structure.

4. *Rubble removal*: As a consequence to the *structural inspection* task, this task is to remove rubble for extrication work once structural stability level is determined.

5. *Medical assessment and intervention*: The rescue process flow continues by mobile robot with special equipment and medical sensors approaching a trapped victim. This type of mobile robot is capable of immediate medical assessment and intervention (e.g. pain relief) before further assessment by medical personnel. Additionally, the mobile robot acts as a communication relay between medical personnel and the victim.

6. *Medically sensitive extrication and evacuation*: For instance, in case of chemical or radiological disaster, danger areas can extend for kilometers. Mobile robot is required to transport victims to medical zone which is outside the affected area.

7. *Beacon/repeater*: Instead of single mobile robot platform, multiple mobile robot network acting as a mobile *beacon/repeater* to extend the wireless communication range.

8. *Serving as a surrogate for a team member*: Mobile robot may work side-by-side with human rescuer and serving as a surrogate for a team member. It might be located between surface rescuers and onsite rescuers (rescuers inside rubble) and provide an update on

status information. This task may be introduced mainly when there are communication difficulties.

9. *Adaptive shoring*: Mobile robots which are specifically designed with heavy duty mechanism would be able to provide an adaptive shoring to prevent secondary collapse that might injure rescuers and survivors.

10. *Logistics support*: Transporter robot act as logistics support by conveying equipment and supplies from storage to disaster site.

It is very difficult to design a mobile robot in order to fulfill all tasks. In the event of disaster, it is very ideal to have several mobile robot systems that may work co-operatively to achieve the mission goals. However, Murphy and Burke (2005) reported that mobile robots work individually to achieve the objectives of each task. This is because, each task requires different mobile robot capability which reflects the dimension, sensors and hardware features. In this thesis, the writer attempt to solve tasks 1 and 2 by proposing a novel mobile robot design. Furthermore, a control method is proposed in order to address the fundamental issue of interaction of mobile robot with rough terrain.

1.1.3 The Challenges

The research topic in USAR, in particular implementation of mobile robot systems is huge (Murphy et al., 2008). Furthermore, the best approach to test USAR mobile robot systems is to involve the actual search and rescue operation (Jacoff et al., 2001). However, at this stage, the actual implementation of USAR mobile robot systems is considered as the secondary response units due to lack of training and awareness amongst rescue pesonnel (Micire and Murphy, 2002). For that reason, National Institute of Science and Technology (NIST) developed reference test arenas (as shown in Figure 1.8) to increase awareness amongst research community as well as the rescue department authority. In these arenas, rescue robot has to perform various tasks from actual USAR operation especially search, mapping and structural inspection. Having said that, the challenges and research questions can be defined by addressing problems arising within tasks defined in the previous section. Mainly, there are five major research areas which cover the aspects of (Casper et al., 2000; Jacoff et al., 2001):

1. design and configuration;

2. navigation and mapping;

3. control architecture;

4. communication; and

5. detection method.

The design and configuration of mobile robotics is actually based on the nature and level of impact of disaster (Murphy et al., 2008). The level of impact might be different from one disaster to another. On the other hand, a collapsed building may pile-up by concrete chunks and steel beams. Both situations require a robust platform that is able to fulfill the tasks explained above. Robot's dimension and weight are directly related to each task. For example, the first three tasks may only need a small and portable platform.

Meanwhile, tasks 4 and 9 may need bigger and heavy-duty platform. Thus, the possibility of developing a single generic platform for all tasks is an open issue. In other words, the research community in USAR has to answer the question "Is my mobile robot design generic for any disaster?". In actual USAR operation at disaster site, rescuers need to communicate with trapped survivors either by verbal or audio devices. However, in the event of mobile robot integration, communication becomes another problem because the environment is full of noise and selection of communication media becomes more difficult. Undoubtedly, a small mobile robot can reach deeper into rubble than human or dog rescuer.

However, in some circumstances, mobile robots may experience loss of communication signal because of the limitation of radio communication media (Nourbakhsh et al., 2005). Another important research question is how robots interact and react in an unknown, uncertain and unstructured disaster world. Control architecture is an essential element for every mobile robot regardless of its functions. A complete structure of control architecture determines the level of behaviour for a specific function. In brief, control architecture is a sequence of robot's behaviours when it moves from the start point until it reaches its target. During that period, a robot may require to overcome static obstacles, experience sudden drop, trapped inside rubble or even react to sudden obstacle in front of it. For every occasion, the robot has to answer the question "What should I do?".

Navigation and mapping problem for disaster environment is one of the main interests in USAR research community. One of the solutions is to use Simultaneous Localisation and Mapping (SLAM) methodology. Although SLAM for unknown structured environment is considered solved (Durrant-Whyte and Bailey, 2006; Bailey and Durrant-Whyte, 2006), the unknown and unstructured world remains an open problem. Even though SLAM is not compulsory to solve navigation problem, this solution is known to be a 'holy grail' towards fully autonomous rescue robot system. In the perspective of control architecture, any solution for mobile robot navigation will work closely with decision-making module in order to decide on the right direction and orientation. At the same time, decision-making module provides information for localization and mapping module.



Figure 1.8: USAR reference test arenas developed by NIST. (Reproduced from Standards and Technology (2012)).

Detection method is also a module in overall control architecture which determines the robots' behaviour whenever it detects signs of life. Multiple sensors are attached onto the robots' platform for this purpose. However, detection of victims is extremely difficult where robots should be able to differentiate sensing input from one entity to another. The major issue for detection is sensors' sensitivity and how effective the sensing algorithm is compared with rescue dog or human perception. Therefore, a sensible victim detection algorithm is needed in order for conventional rescue community to fully accept the integration or intervention of mobile robotics in USAR operation. This thesis, however, will focus on the first three issues which are design and configuration, control architecture, and navigation and mapping.

1.2 Literature Review

Earlier in this chapter, the writer have discussed that there is a serious demand for the development of mobile robot systems for rescue application for the next coming years. Moreover, the development of mobile robot system, particularly for USAR operation, is required to address specific mobile robot tasks as described in Section 1.1.2. On top of that, the outcome of the intervention of mobile robot system in actual USAR operation is expected to increase the overall operation efficiency.

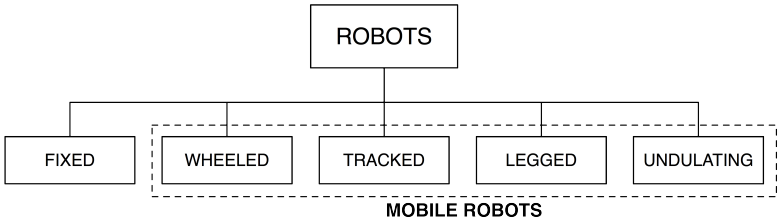


Figure 1.9: General robot classification.

Figure 1.9 shows a general classification of robot systems where four robot categories grouped inside the dashed box can be classified as mobile robot system. Each mobile robot category is named based on the type of locomotion which are wheeled, tracked, legged and undulation (Hirose, 1991). Based on this classification, the suitability of each category of mobile robot for USAR operation will be reviewed. On top of that, the existing rescue mobile robot platforms as well as their control strategies and the influence in the proposed mobile robot system will also be reviewed in this thesis.

1.2.1 Mobile Robot Physical Design

In order to design a mobile robot system for USAR operation, one must identify the design specification based on the mobile robot task explained in Section 1.1.2. In brief, mobile robot tasks can provide general guidance in designing the physical mobile robot system, especially its size, dimension and mechanical capability. For example, a mobile robot designed for *search* and *reconnaisance* task should be relatively smaller size that will allow it to penetrate through rubble as well as move over obstacle (Morrey et al., 2003). In contrast, mobile robot for *logistic support* and *adaptive shoring* requires larger and heavier machinery in order to achieve the design objectives (Murphy, 2002).

A considerable amount of literature related to the mobile robot physical design has been published in Hirose (1991), Silva and Machado (2007) and Hemes et al. (2011). In addition to the mobile robot tasks, these studies provide fundamental design characteristics and selection of mobile robot configuration restricted to the terrain. However, the development of generic design for unstructured environment is still a major research question. Previous studies have suggested that the design of a mobile robot for USAR operation, in particular within *search* and *reconnaisance*, has to be small enough to fit through voids (Murphy et al., 2000), but much larger to overcome surrounding obstacles (Jinguo et al., 2007). Moreover, it is also suggested that the USAR mobile robot is highly mobile and flexible (Micire and Murphy, 2002). Additionally, the choice of mobile robot also depends on the logistic support at disaster site (Murphy, 2004). This is because a large and heavy mobile robot platform requires heavy machinery to transport from one place to the other compared to small and light weight which can be carried by one or two

people (Murphy, 2002).

Among the mobile robot classification shown in Figure 1.9, legged configuration (particularly biped) and undulation mobile robot (i.e. snake-like configuration) are argued to be highly flexible in terms of terrain adaptation due to the high number of degree-of-freedom (Ostrowski and Burdick, 1998; Gomi et al., 2003). However, Huang et al. (2001) proved that the biped mechanism is not yet suitable for unstructured environment because of two reasons: (1) higher centre of gravity (Matsumoto et al., 1998); and (2) large clearing capability (Morrey et al., 2003). Nevertheless, Saranli et al. (2001) and Taylor et al. (2007) have demonstrated RHex (six-legged hexapod robot) and Whegs (wheel-leg robot) are suitable for rough terrain applications. Meanwhile, Choset (2002) argued the mobile robot with undulation motion (e.g. hyper-redundant mechanism) is difficult to program resulting in longer training duration (Choset et al., 2000).

Besides legged and hyper-redundant mechanism configurations, wheeled and tracked mechanisms are the most common type of mobile robot configuration (Martínez et al., 2005; Dudek and Jenkin, 2010). However, according to Siegwart et al. (2002) and Lee et al. (2003), these type of mobile robot require larger wheel (or track's sprocket) diameter by at least double of the obstacle height. Nevertheless, it might not be very useful as the height of obstacle in unstructured environment varies (Davids, 2002). Therefore, Chiu et al. (2005), Kim and Lee (2007) and Clement and Villedieu (1987) implied additional assisted climbing mechanism to enhance obstacle overcoming ability even with wheel (or track's sprocket) diameter which is less than obstacle height. However, a single-tracked articulated body crawler demanded track's tension control mechanism due to variation of track length during transformation (Iwamoto and Yamamoto, 1990).

1.2.2 Control Framework Architecture

In the previous subsection, we have reviewed mobile robot classifications which motivated the proposed mobile robot design in this thesis. In general, it is compulsory for any mobile robot system to contain a diverse range of sensors and actuators which are controlled through certain level of complexity software architecture (Brooks, 1986; Shafer et al., 1986). Having said that, it is very helpful for a mobile robot developer to answer several questions in order to establish or to construct a system architecture for their mobile robot system, such as:

- What is the global objective of the mobile robot system?
- How will the mobile robot monitor the mission?
- What should the mobile robot react to the actual mission?
- How should the mobile robot systems behave in case of unforeseen circumstances?

According to Kortenkamp and Simmons (2008) there are several methods to solve all the above questions, either concurrently or asynchronously. However, solving these concurrently will increase the complexity of the architecture structure. Despite that no one single mobile robot architecture is considered the best (Kortenkamp and Simmons, 2008), Brooks et al. (2005) suggested a well-organised mobile robot architecture which often helps to reduce complexity of the architecture structure for mobile robot system. Meanwhile, Simmons et al. (1997a) offers additional advantages especially in composition of several abstraction layers to increase reliability and proven by Simmons et al. (1997b) through office delivery task (i.e. office environment navigation) problem.

In depth, robot architecture (industrial and service robot system) is always associated with task planners and controllers components and often arranged in forms of algorithm (programming language) (Alili et al., 2009). Additionally, a complete robot architecture system can be grouped in several subsystems which interact with each other in order to accomplish the global mission (Stoytchev and Arkin, 2001). Moreover, this approach can be improved by separating the subsystem according to robot behaviour as suggested by Woolley et al. (2011). In USAR perspective, mobile robot system architecture often requires the subsystem to interact asynchronously because the robot's sensors and actuator actively deal with many uncertainty, in unknown and unstructured environment (Proetzsch et al., 2010). Therefore, this thesis will look at potential for a framework architecture in order to demonstrate the proposed mechatronic system in actual setting.

1.2.3 Motion Planning and Tracking

Apart from the physical mobile robot configuration and the control architecture framework, navigation and mapping problem is also an important element to take into consideration to ensure adequate implementation of mobile robot in USAR environment. Again, it is convenient to answer several questions to establish the direction of navigation and mapping solution, such as:

- What is the global objective of the mobile robot mission?
- How to achieve the global objective?

To date, there are various methods to produce a world map (i.e. structural map modeling for robot environment) for unknown environment (Burgard and Hebert, 2008). The development of this particular map typically applies several types of sensors where each type of sensor delivers certain advantages and disadvantages (Manikas et al., 2007). Moreover, the selection of appropriate sensors is crucial to model and construct terrain (or obstacle) representation which

is more important than typical mapping.

In general, the development of world map for any unknown, unstructured environment is considered as probabilistic robotics problem (Thrun et al., 2005), whereby the primary solution is mainly dominated by employing the Kalman filter and its derivatives (Dissanayake et al., 2000, 2001; Hahnel et al., 2003; Leonard and Durrant-Whyte, 1991). Additionally, Durrant-Whyte and Bailey (2006) and Bailey and Durrant-Whyte (2006) have suggested to utilise the Kalman filter to solve *SLAM* in order to build a consistent map whilst localise a mobile robot in world environment. Meanwhile, there are still some demands in solving mapping problem with less computational power (i.e less sensor payload) as proposed in Moravec and Elfes (1985). The focus in this method is emphasising on modeling the terrain/obstacles has been proven by Daud et al. (2013) and Husain et al. (2013).

Concurrent to the environment map modeling, mobile robot navigation, in particular planning and tracking, are also important in order to derive the global mission in terms of mobile robot motion (i.e. desired travelling pattern and tracking the desired travel path). Even though building a world map, planning and tracking of a mobile robot motion can be solved using various *SLAM* methodology (Thrun et al., 2005; Thrun and Leonard, 2008; Durrant-Whyte and Bailey, 2006; Bailey and Durrant-Whyte, 2006), planning and tracking their motion on an unknown, uncertain and unstructured terrain can be influenced by the obstacle (terrain) shape and dimension (Howard and Kelly, 2007). Therefore, Kanayama et al. (1991) suggested Fahimi (2009) approach to determine the mobile robot motion behaviour using the derivation of mobile robot mechanic over specific time interval. This derivation is very important and has been proven by Liu and Liu (2009b) in particular in a structured two-dimensional environment.

1.3 Aims

Earlier in this chapter, the writer have described the future potential of this research area based on the sales projection of mobile robot in the application of defense, rescue and security for the next coming years. Furthermore, a number of existing research activities have motivated this work towards developing a complete search and rescue mobile robot system. One of the aims of this research is to design and develop a mobile robot platform for USAR operation which is highly mobile, flexible, and has to be in appropriate dimension to deal with uncertainty in unstructured world whilst understanding its surrounding.

In addition to the physical design, another aim is to propose a comprehensive control framework in order to fully utilise the novel features in the physical design. On top of that, the work also aims to study the motion of the mobile robot traversing on rough terrain and its behaviour

when instructed to undertake specific mission plan. The objectives of this thesis are summarised as follows:

- To design a mobile robot system that is able to traverse on rough terrain;
- To evaluate the proposed mechatronic design on uneven surfaces;
- To develop the track-angle mathematical model for the mobile robot when it requires to vary its geometry;
- To develop a mathematical model as the representation of an uneven terrain;
- To develop the desired three-dimensional path on terrain model;
- To investigate the desired behaviour of mobile robot in order to fulfill certain mission;
- To simulate the mobile robot behaviour associated with the terrain model;
- To implement the conventional method of two-dimensional trajectory tracking model on an uneven terrain model; and
- To evaluate the performance of trajectory tracking controller.

In order to evaluate the proposed mechatronic design, the prototype is tested on several type of surfaces. Furthermore, computer simulation were used to model the terrain surface and evaluate the controller performance. Moreover, results are presented in each chapter in order to evaluate research goals described earlier.

1.4 List of Novel Contributions

There are several novel aspects contributed whilst solving the issues addressed in Section 1.1.3 as well as to bridge the research gaps (discussed Section 1.2). The main contributions of this thesis are summarised as follows:

1. The development of a new and novel design of a mobile robot that is able to adapt to various types of terrain by employing the single-tracked VGTV design. Due to the single-tracked design and non-deformable track material properties, a novel track tension control method is applied. Moreover, both side of the track angle can be adjusted independently to enhance stability of the platform especially on a complex terrain. (Chapter 2)
2. The development of the overall control framework based on *three-tiered layer control architecture*, in which the *highest control layer* defines the desired mission objective. On the other hand, the *intermediate control layer* translates this mission into actual mobile robot

motion behaviour. Additionally, the *lowest control layer* emphasises on track motion with respect to the angle required in order to overcome specific obstacle or to enhance physical body stability to prevent tip-over. In the proposed control framework, 'intervention' block is introduced to represent the external input that may overwrite the initial mission. (Chapter 3)

3. The development of mathematical model to represent the actual terrain is one of the novel contributions discussed in this thesis. The surface model which is generated to visualise actual terrain is modeled by utilising summation of several Gaussian functions. This is one of the key elements of this research which initiate the evaluation of mobile robots behaviour along the desired path on the surface. Concurrently, a two-dimensional path/trajectory is generated as the navigation guidance (travel plan) for the mobile robot, implementing a cubic Cartesian parametric equations. Then, the two-dimensional path/trajectory and the surface model are synthesis to get a new three-dimensional navigation guidance for the mobile robot. (Chapter 4)

4. Further, the development of full and comprehensive kinematic model for mobile robots that are required to travel on complex and uneven terrain. The kinematic model derived in this thesis is in a complete six-degree-of-freedom which is very useful for three-dimensional problem. In brief, the kinematic model is very beneficial to evaluate or predict the behaviour of a mobile robot when traversing on uneven terrain. (Chapter 4)

5. Finally, the development of trajectory tracking model to ensure the mobile robot is following the designated three-dimensional path. Similarly, this model is develop by implementing the conventional two-dimensional solution whilst extend it into a comprehensive tracking model for three-dimensional problem. (Chapter 5)

The details of each contribution, including the mathematical derivation and results, will be discussed in respective chapters.

1.5 Context and Thesis Outline

In the next chapter, Chapter 2, the proposed design is presented which utilises the VGTV configuration and proposes a new method to control the variation of track's tension. This chapter begins with a brief introduction (Section 2.1) with provides a general overview of USAR mobile robot in terms of dimension, portability and driving configurations. Furthermore, a comprehensive literature review, aims and proposed methodology are discussed in this section. Section 2.2 accommodates a comprehensive initial analysis of the relationship between track

length and geometric transformation, describing mathematical justification of tracks' varying mechanism to control the track length by identifying one side of the track relation with the climb angle because the other side is identical. The mechatronic design of the proposed mobile robot system is explained in Section 2.3. Further, Section 2.4 provides the assessment from the actual physical transformation of the proposed VGTV design over several obstacles including ramp, curb, stair and ruble. The last two sections provide some discussions and concluding remarks (Section 2.5) related to the observation of actual transformation of VGTV presented in Section 2.4.

The control architecture framework for the proposed mechatronic system is discussed in Chapter 3. This chapter begins with a brief introduction which addresses the importance of a control architecture for systematic approach of mobile robot programming. Then, the motivation is given in Section 3.2, which provides a comprehensive literature review about the growing body of mobile robot control architectures. Additionally, this section emphasises the systematic control architecture methodology which motivates the control framework proposed in this thesis. Section 3.3 defines the primary objectives of the control architecture and Section 3.4 formulates the control framework in terms of the degree of control hierarchy. Describing this further, Section 3.5 describes the *three-tiered layer control architecture* as well as the control schemes embedded in each layer. Meanwhile, Section 3.6 provides some concluding remarks.

In order to demonstrate the capability of the control architecture framework proposed in Chapter 3, Chapter 4 highlights the development of a high degree-of-freedom kinematic model for uneven surface. This chapter begins with a comprehensive introduction (Section 4.1) addressing the distinction between the holonomic and non-holonomic mobile robot system. Additionally, a comprehensive literature review and aims are given in this section. In Section 4.2, a comprehensive initial analysis of kinematic formulation for 3-dimensional environment is given, describing utilitisation of a two-dimensional formulation on a 3-dimensional environment. In addition, the mathematical derivations consisting of the generation of the pre-planned path, and modeling of an uneven terrain are explained in this section. Furthermore, Section 4.3 formulates the generalised vectors along the desired trajectory for both linear and rotational unit vector components associated with mobile robot motions, as well as the desired kinematic behaviour (linear and angular velocities) of mobile robot along the desired trajectory. Using a computer simulated program, the results are presented in Section 4.4. Section 4.5, provides some discussion related to the results of the kinematic simulation and some concluding remarks.

Following the kinematic model discussed in Chapter 4, Chapter 5 focusses on the control method of mobile robot trajectory tracking, modeling and control under the same mechanical constraint explained in Chapter 4, while focusing on the utilisation of the classical control

technique of a flat surface for an uneven surface problem. This chapter begins with a brief introduction (Section 5.1) highlighting the demands of tracking capability for a mobile robot systems, reviews of present methods and the aim of this section. Furthermore, in Section 5.2, comprehensive problem statements for mobile robot motion on flat and uneven terrain are given, describing utilitisation of flat surface formulation on an uneven terrain environment. Section 5.3 explains the control strategy. The mathematical derivation of error tracking model based on Kalman filter position estimate is derived in Section 5.4.1 and 5.4.2. Meanwhile, Section 5.4.3 describes the control laws and the controller performance and simulation results are presented in Section 5.5. The last two sections (Section 5.6 and Section 5.7) provide some discussion and concluding remarks related to the simulation results and controller performance.

The last chapter, Chapter 6, provides the conclusion of the research work discussed in this thesis. Futhermore, some potential research work related to the subject of this research area is discussed in this chapter.

Chapter 2

Mechatronic Design of Variable Geometry Tracked Vehicle with Independent Track Control

2.1 Introduction

Over the past two decades, there has been increasing interest in research activities emphasising the application of mobile robot systems in uncertain, unstructured and dangerous environments. The application of mobile robot systems, for example in urban search and rescue (USAR) operation, requires a platform that is capable to handle and negotiate many uncertainties, in particular the terrain's structure. It is also difficult to ignore the need for the mobile robot platform to have a certain level of autonomy in order to adapt itself to various types of terrain and work robustly in complex and highly unstructured enviroments. On top of that, it is challenging to design and develop a generic and versatile mobile robot system that is able to adapt and perform in many types of environment and operating missions. In this chapter, the main focus is to propose a novel design of a mobile robot that is able to adapt to various types of terrain. Furthermore, the proposed mechatronic system is tested in several terrain setups including ramp, curb, staircase and rubble.

In order to design a mobile robot system that is able to negotiate with uneven terrain, one must identify his/her mobile robot design associated with the mobile robot classification. In general, mobile robot system can be classified according to the physical size and portability. A common classification approach, which is presented in Casper and Murphy (2003) divides mobile robots into three categories:

- *Man-packable* is the smallest mobile robot in terms of physical size. For that reason, a mobile robot belonging to this category should be able to be carried safely by one man. In other words, the entire mobile robot system including batteries, remote control, spare-parts and etc can fit in one or two backpacks.

- *Man-portable* is slightly larger than *man-packable*. In terms of portability, a mobile robot in this category should be able to be transported by all-terrain vehicle and can be carried by one or two men for a short distance.

- *Maxi* is the largest category in terms of mobile robot physical size. It requires special transportation such as trailer in order to transit from the mobile robot storage area to designated operation sector.

In addition to the classification of a mobile robot system, another key aspect in designing a versatile mobile robot that is suitable for uneven terrain is the selection of the driving mechanism system. A typical driving mechanism can be broadly categorised over many mobile robots' locomotion in various applications, which include *wheeled*, *crawler-tracked*, *legged* and *hybrid* mechanism. However, in USAR mission perspective, a mobile robot is required to achieve stability whilst moving in robust and uncertain environment. On top of that, it is critically important to ensure sufficient clearing capability that allows it to pass over "small" voids or explore narrow paths. For these reasons, *legged* mechanism such as biped human-like mobile robot and RHex, (Moore et al., 2002), are inappropriate for USAR operation. However, it is difficult to ignore the capability of biological inspired *legged* robot such as ASIMO in USAR operation. This is because their ability to be programmed to provide human-like motion which is highly useful in USAR operation. Therefore, the adaptation of *legged* mechanism robot in the near future is inevitable. Meanwhile, with reference to classification given earlier, the design of the mobile robot should be considered as *man-packable* in order to move through "small" void and narrow paths. Another important design characteristic is the ability of mobile robot to pass or climb obstacles (e.g. staircase). In this chapter, the main research interest is to investigate the existing mobile robot configuration that is suitable to perform motion in complex and uneven terrain. Additionally, the most fundamental problem in design process is addressed by improving existing design to accomplish desired motion on specific obstacles.

2.1.1 Motivations

To date, there has been a wealth of research activities on mobile robot driving mechanism. As mentioned earlier, currently, *legged* mechanism in particular biped human-like mobile robot is clearly not designed for a highly unstructured environment application due to fabrication cost and programming complexity. However, despite being developed for educational and entertainment purposes, biped mobile robot (e.g. ASIMO) has proven highly adaptable when moving on prepared terrain, in particular staircase (Hirai et al., 1998). However, implementation in complex and uncertain environment is still highly debated especially after 2011 Tohoku earthquarke that

severely damaged Fukushima Daiichi Nuclear Power Plant (Kaneko, 2012).

Meanwhile, *hybrid* design such as *hyper-redundant* mechanism is reported to be able to enter narrow space and traverse in highly complex environment. Nevertheless, it is difficult to control these motion because it consists of more then two flexible joints. On the other hand, *wheeled* mechanism requires larger wheel diameter to extend the climbing limit to at least half of its wheel diameter (Moosavian et al., 2006). Similarly, a typical *crawler-tracked* mechanism is only able to overcome an obstacle that has at most half of the track's height. However, in contrast to *wheeled* mechanism, a *crawler-tracked* mechanism can achieve good stability by optimising track-to-ground contact surface.

In spite of that, a promising *crawler-tracked* configuration that is capable to climb over an obstacle which is larger than the track's height is called variable geometry tracked vehicle or VGTV. It can be further identified based on its track arrangement: either single-track or double-track. A double-track VGTV, Figures 2.1-2.3, normally consists of two tracks on each side of its body where the longer track is used to drive the platform whilst the other one, arranged at the front of the platfrom and have shorter length, is actuated via separate track arm to perform various tasks such as climbing (Lee et al., 2003). Another variation of double-track VGTV is presented by Schempf et al. (1999) and Lee et al. (2004), where the actuated tracks are configured in triangular shape bringing an unadjustable climbing angle. Both double-track VGTVs are designed to prevent track material deformation particularly during climbing motion.

On the other hand, a single-track VGTV, Figure 2.4, has a simpler track arrangement where each side of the mobile robot platform only employs one track to perform climbing motion. The first single-track VGTV was introduced by Iwamoto and Yamamoto (1984) and it was further improved by Goldenberg et al. (2009). Both designs consisted of three wheels on each side and a set of planetary wheels to ensure a constant track length whilst changing the configuration of track. Meanwhile, Kim and Lee (2007) proposed a different approach to address the track length problem by investigating the distance between rotation axis (of track arm) and both ends of the arms.

Both single and double-track VGTV arrangements were designed to achieve specific climbing motion whilst maintaining a constant track length. Motivated by Kim and Lee (2007) and iRobot (2010), this chapter proposes a new VGTV configuration with two independent track arms. The prototype is able to modify its geometrical configuration based on sensor input whilst the independent track arm enhances stability of its body when traversing along uneven surfaces. For example, consider a single-track VGTV traversing along a side slope as depicted in Figure 2.5. Without independent track control feature (Figure 2.5a), the mobile robot tends to roll-

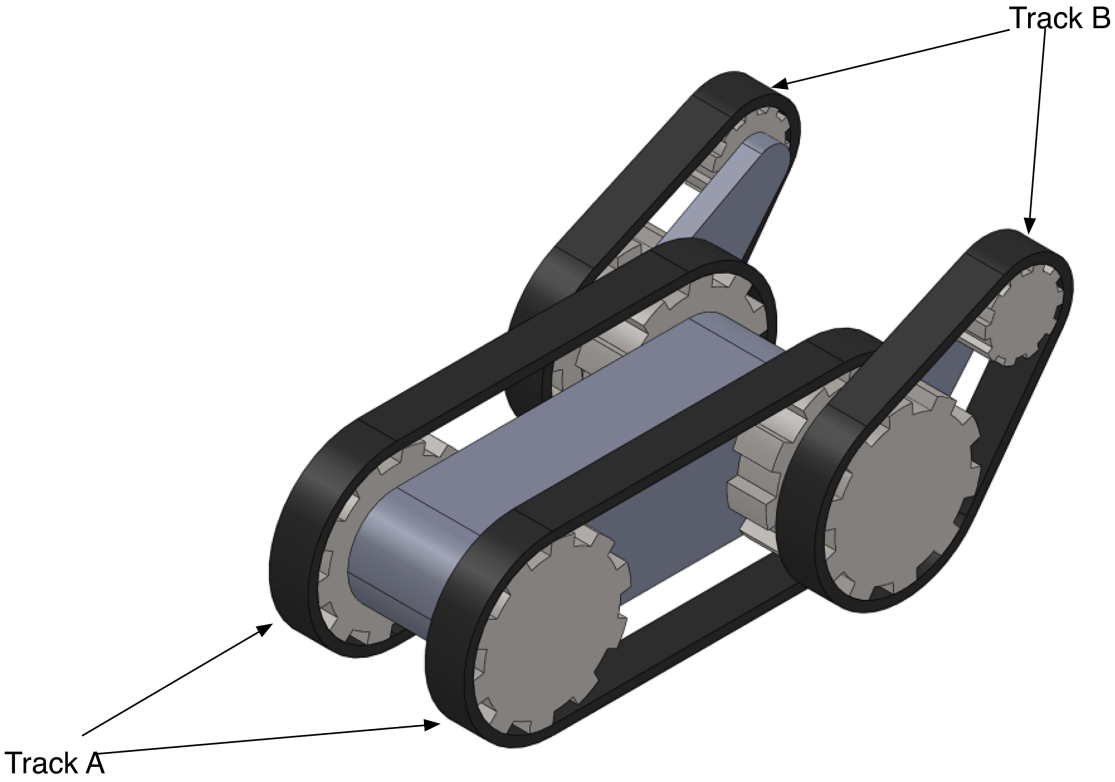


Figure 2.1: A basic configuration of double-track VGTV. Image is adapted from iRobot (2010).

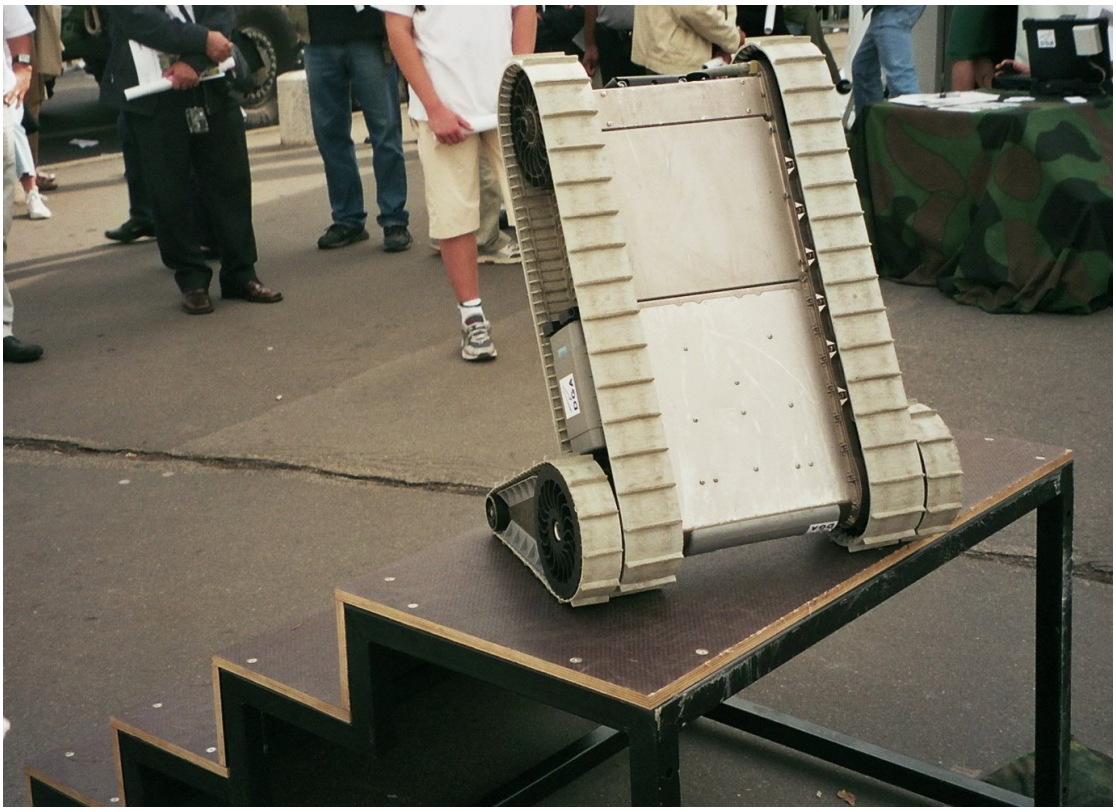


Figure 2.2: PackBot from iRobot is an example of a double-track VGTV and widely use in military operation (iRobot, 2010). Image is free to share, under the licensed of Creative Commons.

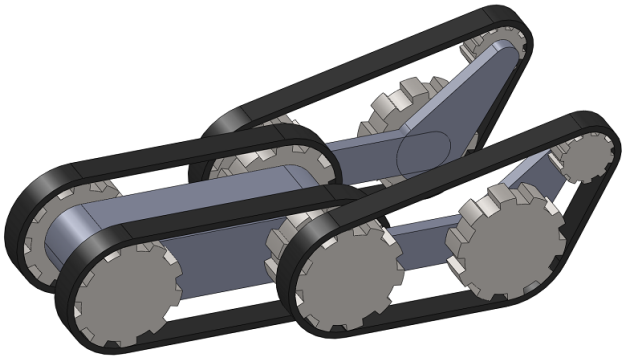
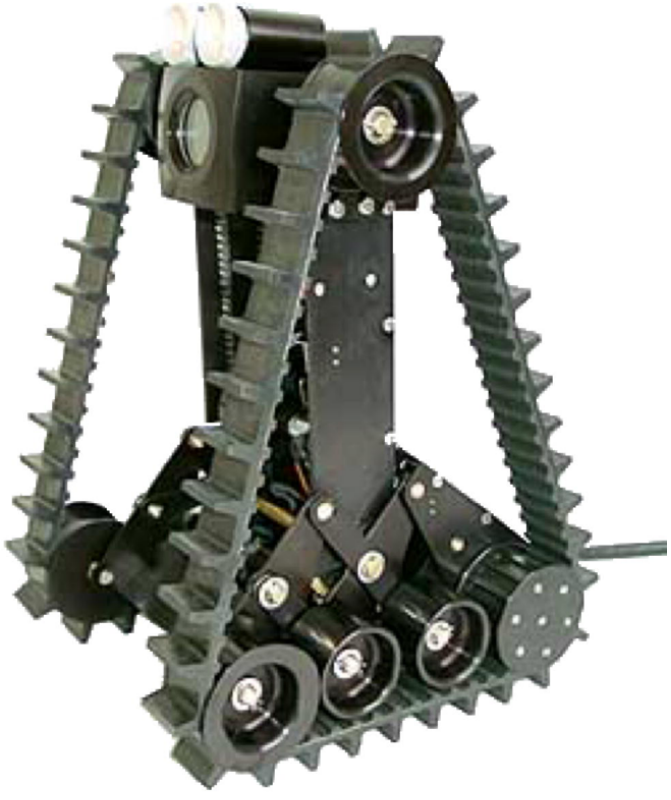


Figure 2.3: Another double-track VGTV configuration. Adapted from Woosub et al. (2005).



Source: Inukton (www.inukton.com)

Figure 2.4: Inuktun VGTV is an example of single-track VGTV. Adapted from Wang and Gu (2007).

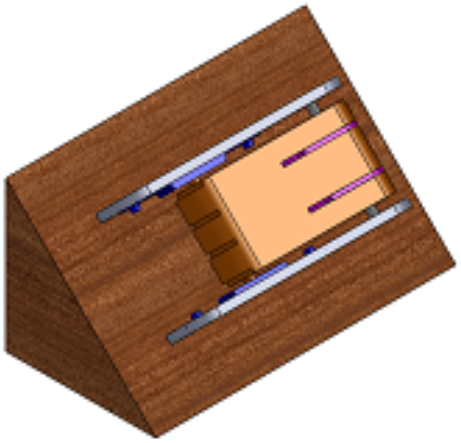
over if the side-slope angle exceeds the angle that the mobile robot can compensate. In contrast, Figure 2.5c illustrates the proposed design in this chapter executing transformation on one side of its track to overcome potential mishap. Additionally, the track's tension variation is controlled by a simple leadscrew mechanism and has self-righting capability due to a symmetrical design. In this thesis, the term VGTV is refers to the single VGTV arrangement.

2.1.2 Aim

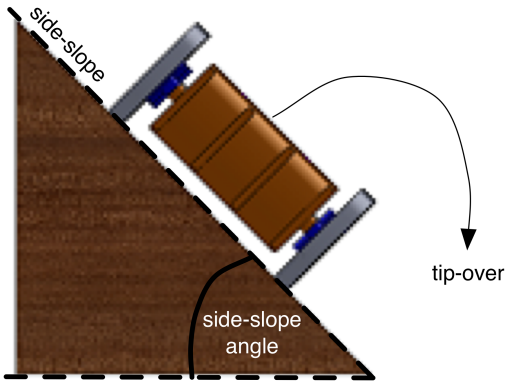
The aim of this chapter is to focus on developing a mobile robot platform that is able to travel on various types of terrain. The proposed design, which is motivated by Goldenberg et al. (2009) and Paillat et al. (2008) is a new VGTV configuration that has actuated body mechanisms on both sides of the track. The actuated part which is called track arm, can perform identical motion for climbing purpose or controlled independently to ensure stability of overall physical body. Due to the single-track design and non-deformable track material properties, a novel track tension control method is applied. Unlike Goldenberg et al. (2009), the prototype of VGTV mobile robot can achieve a constant track length throughout geometric transformation by employing a simple leadscrew mechanism. The track length geometry formulation, which is motivated by Kim and Lee (2007) is important to determine the design parameters prior to the actual physical design. In summary, certain improvements are made over the prototype proposed by Kim and Lee (2007) and Goldenberg et al. (2009) which are:

- linear leadscrew mechanism is employed to control the track's tension;
- the track angle on both sides of mobile robot can be controlled independently;
- the prototype is designed in symmetrical manner whereby;
  - the left side is identical to the right side; and
  - the upper side is identical with the bottom side, to achieve self-righting capability.
- unlike design parameters determined by Kim and Lee (2007), the distance between both track's arm end and the rotation axis is equal; and
- the track geometric modeling results the relationship between leadscrew mechanism and track angle.

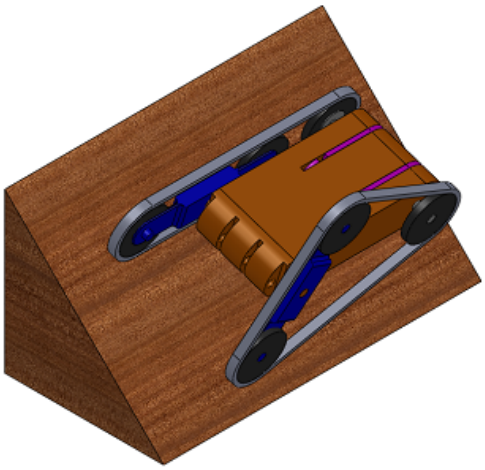
Furthermore, the result from track geometric modeling which is motivated from Kim and Lee (2007) is important to minimise the track material deformation. The function of the leadscrew mechanism is not only useful to achieve a constant track length throughout geometric transformation, but also beneficial to adjust the track's tension. For instance, the leadscrew can reduce



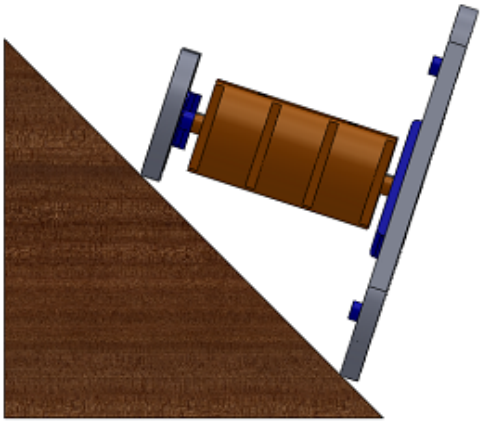
(a) Fixed-geometry tracked mobile robot.



(b) Front-view of fixed-geometry tracked mobile robot.



(c) Proposed VGTV mobile robot on side-slope.



(d) Front-view of the proposed VGTV mobile robot on side-slope.

Figure 2.5: Three-dimensional and two-dimensional (front-view) of the VGTV traveling along a side slope. Figure 2.5a and 2.5b illustrate a conventional fixed-type tracked mobile robot with possibility to experience tip-over with excessive side-slope angle (rolling angle, ϕ). The proposed VGTV mobile robot in this chapter (Figure 2.5c and 2.5d) is capable to negotiate with side-slope to prevent mishap overturn. .

the track tension to get better grip on terrain. Additonally, it also has a stereo vision camera system (SVS) to capture video image of its surrounding, as well as an inertial measurement unit (IMU) to detect body orientation during transformation.

2.1.3 Methodology

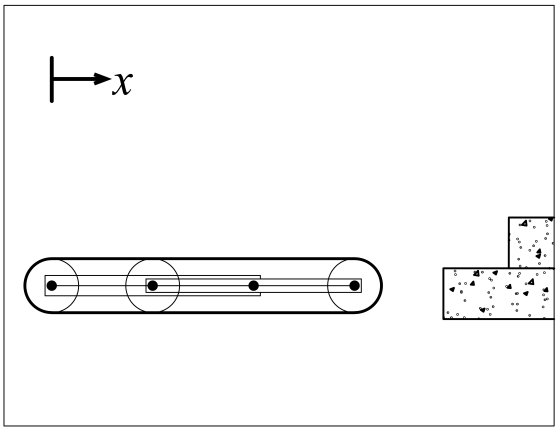
Generally speaking, any mobile robot travelling on unknown and uneven terrain is required to have certain level of ability to adapt with various terrain structure. In case of the VGTV proposed in this chapter, it utilises the variation of geometrical configuration in order to adapt with various obstacle features such as ramp, curb, rubble and stair. In this chapter, the problem of mobile robot (in particular VGTV) dealing with uneven surface is classified into two categories:

1. overcome obstacle in forward direction (climbing motion); and

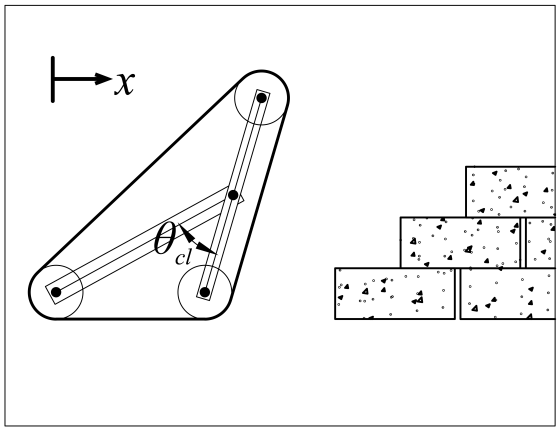
2. comprise a side-slope to prevent tip-over.

In Figure 2.6, an example of a VGTV overcoming obstacle principle is illustrated. Let the arrow labelled with x denote the forward direction, the basic command movement of the VGTV is controlled by a local control unit, such as microcontroller along side with motor drivers. As soon as the mobile robot encounters an obstacle, it quickly varies its configuration to adapt with the obstacle with an appropriate climbing angle, θ_{cl}, instead of avoiding the particular obstacle. In the second scenario as depicted in Figure 2.5a, consider the VGTV moving along a side-slope. If the side-slope angle exceeds the allowable limit that the mobile robot can handle, there is some possibility for the mobile robot to slide along the slope or even worse, accidentally tip-over. In contrast, a VGTV with independent tracked control is able to adjust its orientation by varying only one side of the track configuration to enhance platform stability to prevent mishap as illustrated in Figure 2.5c.

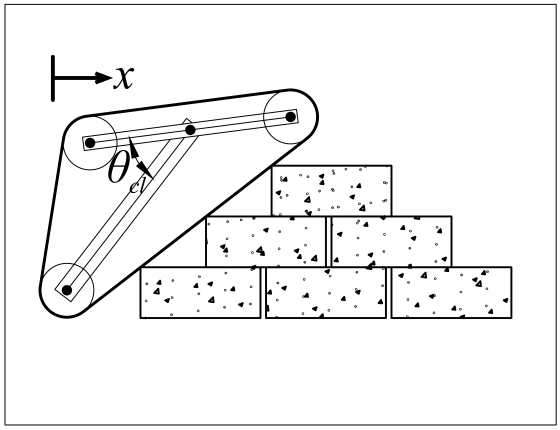
This chapter is hereinafter organised as follows; in Section 2.2, a comprehensive initial analysis of relationship between track length and geometric transformation are given, describing mathematical justification of tracks' varying mechanism to control the track length by identifiying one side of the track relation with the climb angle because the other side is identical. Furthermore, the mechanical structure, control and sensing capabilities are explained in Section 2.3. Section 2.4 demonstrates the actual physical transformation of the proposed VGTV design over several obstacles including ramp, curb, stair and rubble. The last section provide some discussion and concluding remarks (Section 2.5) related to the observation of actual tranformation of VGTV presented in Section 2.4.



(a) VGTV traverse on a flat surface.



(b) VGTV approaches an inclination.



(c) VGTV climbs an obstacle.

Figure 2.6: 2-dimensional side view of climbing principle of the proposed VGTV mobile robot. Figure 2.6b illustrates VGTV configuration before transforming to a climbing configuration as shown in Figure 2.6c.

Table 2.1: Fixed design parameters for the proposed mobile robot.

	Symbol	Value (mm)
Total length	L	530
Total height	H	80
Total width	W	150
Arm length	x_2	300
Chasis length exclude wheel	x_1	450
Wheel diameter	D_w	80
Track width	tl_w	40

2.2 Geometric Formulation

In Section 2.1.1, several examples of VGTV configuration and methods to control the track length problem have been presented. Meanwhile, in Section 2.1.2, a new approach to control the track variation is proposed by employing a linear leadscrew mechanism in order to achieve a constant track length throughout geometrical transformation. The next step is to describe the physical design. However, it is very important to investigate the track geometry formulation related to the geometrical transformation before describing the mechatronic design of the proposed VGTV mobile robot in detail. This is because, without any track's tension control mechanism such as leadcrew, the total track length will increase as the θ_{cl} increases. In contrast, with the implementation of track's tension control mechanism on the design, it is not only able to achieve a constant track length as θ_{cl} increases, but also able to vary the track's grip by slightly reducing the track's tension especially during climbing.

As mentioned in the previous section, the proposed VGTV design in this chapter uses a single-track configuration and utilises a leadscrew mechanism to adjust the track's tension. Now, let the VGTV traverse on a flat surface before it encounters an obstacle. The behaviour of the VGTV when it encounters any obstacles is illustrated in Figure 2.6. The control principles are based on the obstacle's slope parameter, then a suitable geometric transformation is initiated for climbing purpose. In order to investigate the relation between track length and geometric transformation, the analysis of the track length is devised in two conditions which are:

1. track length during VGTV in flat position, and

2. track length during VGTV in any climbing position (at $0 \leqslant \theta_{cl} \leqslant \frac{\pi}{2}$ rad).

Because the proposed VGTV is designed identical on left and right side, the calculation of the desired track length, tl_{des}, is done only at one side of the mobile robot. Therefore, the related mathematical computation in this section is done by assuming both tracks vary with the same direction. Additionally, the design parameters for the proposed VGTV are summarised in Table 2.1, considering design requirement and specification determined in Moosavian et al. (2006).

2.2.1 Track geometry and configuration control

Consider the parameter θ_{cl} denoted as the angle required to climb an obstacle. This angle is constructed between the track arm and the VGTV's main body. The desired track length, tl_{des}, is given when the mobile robot platform is in initial posture where the initial posture is given by when the geometric configuration is fully flat. Meaning, most of the tracks' surface on both sides of the platform are in contact with the terrain surface as shown in Figure 2.7a. Moreover, the climbing parameter, θ_{cl}, is equal to zero for the initial posture. With reference to Figure 2.7a, the track length can be derived as

$$\begin{aligned} tl_{des} \quad = tl_{flat} \quad &= x_1 + (x_1 - nx_2) + nx_2 + s_2 + s_3 \\ &= 2 \cdot x_1 + s_2 + s_3 \end{aligned} \tag{2.1}$$

where

x_1 is the length of track surface which is in contact with the terrain surface within flat configuration, which is also equal to the length of CE,

s_2 and s_3 are the track length (curvature) around front idler and back driving wheels.

However, in order to overcome an obstacle which is higher than half of its track height, the VGTV is required to transform its geometric configuration by actuating both track arms at certain angle resulting in changes in θ_{cl}. Let the track arms' operating angle be set to $0 < \theta_{cl} < \frac{\pi}{2}$ rad. Investigating Figure 2.7b, the variation of track length value within this limit results in the new track length equation. The new track length and angle relationship can be written as

$$tl_{0<\theta<\frac{\pi}{2}} \quad = \quad x_2 + a + b + s_1 + s_2 + s_3 \tag{2.2}$$

With reference to Figure 2.7b, the parameters of a, b, s_1, s_2 and s_3 in equations 2.1 and 2.2 can be derived using *Law of Cosines*, and given as

$$a = \sqrt{(mx_2)^2 + (x_1 - nx_2)^2 - 2\,(mx_2)\,(x_1 - mx_2)\cos(\theta_{cl})} \tag{2.3}$$

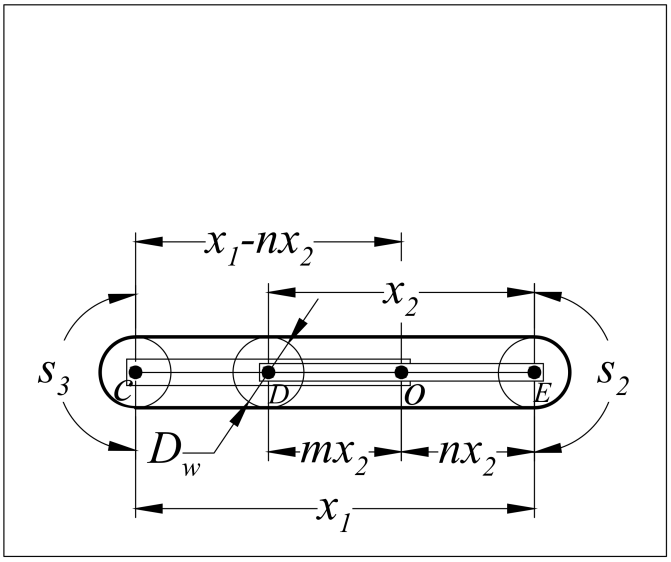
$$b = \sqrt{(nx_2)^2 + (x_1 - nx_2)^2 - 2\,(nx_2)\,(x_1 - nx_2)\cos(\pi - \theta_{cl})} \tag{2.4}$$

$$s_1 = \frac{D_w}{2}\left\{\pi - \cos^{-1}\left[\frac{a^2 + (mx_2)^2 - (x_1 - nx_2)^2}{2amx_2}\right]\right\} \tag{2.5}$$

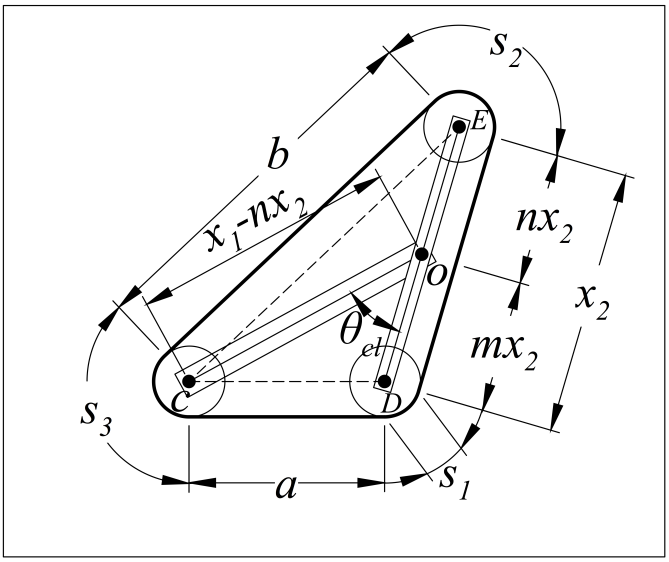
$$s_2 = \frac{D_w}{2}\left\{\pi - \cos^{-1}\left[\frac{b^2 + nx_2^2 - (x_1 - nx_2)^2}{2bnx_2}\right]\right\} \tag{2.6}$$

$$s_3 = \frac{D_w}{2}\left[\pi - \cos^{-1}\left(\frac{a^2 + b^2 - x_2^2}{2ab}\right)\right] \tag{2.7}$$

where



(a) Flat ($\theta = 0$).



(b) climb $\left(0 < \theta < \frac{\pi}{2}\right)$.

Figure 2.7: Two-dimensional side views of a VGTV with climbing configuration. On a flat surface, the proposed VGTV does not require to vary its geometry, as shown in Figure 2.7a. In contrast, it is required to transform its geometric shape in order to deal with obstacles. However, the tracks length parameter changes when the mobile robot varies its geometry. Figure 2.7b shows all the parameters to be considered in order to analyse the track length problem.

Table 2.2: The difference between maximum and minimum value of track length for each combination in Figure 2.9.

Ratio	1 : 9	2 : 8	3 : 7	4 : 6	5 : 5	6 : 4	7 : 3	8 : 2
Difference	93.02	62.69	28.34	27.50	70.82	127.04	187.85	252

x_2 is the track length which is parallel to track's arm, DE,

s_1 is the track length (curvature) of second idler wheel,

a is the track length during climbing configuration, CD,

b is the track length during climbing configuration, CE,

θ_{cl} is angle constructed between main chassis and track arm in order to adapt with climbing slope, $\angle COD$,

m is the distance between centre of idler wheels with the rotation axis of track arm, OD,

n is the distance between centre of idler wheels with the rotation axis of track arm, OE, and

D_w is diameter of the wheels.

Note that the computation distance rate of EO and DO in Figure 2.7b is important to determine the total track length as θ_{cl} increase. On top of that, a desired total track length at any value of θ_{cl} is determined by substituting design parameters in Table 2.1 into equation (2.1). Further computation related to the distance rate of EO and DO and θ_{cl} will be analysed in the next section.

2.2.2 Comparison of track length variation

It has been suggested that the distance ratio between both track's arm end and rotation axis, EO and DO is $1 : 1.310$ (Kim and Lee, 2007). However, a simpler method to analyse the effect of the total track length and the distance ratio of EO and DO is by solving equation (2.2) at several distance ratios. Now, consider the track arm length, DE, equally divided into ten small units as characterised in Figure 2.8. Additionally, the placement of the rotation axis is illustrated by any of the solid grey dots along the centre of the arm (dotted line). Let the rotation axis fit at one of the grey dots, the distance from the rotation axis towards the left end of the track arm is denoted by m (or inner arm distance) and the distance from the rotation axis towards the right end of the track arm is denoted by n (or inner arm distance). In this section, the analysis of the track length is achieved using the distance rate of m and n combination.

Substituting each combination of m and n into equations (2.3), (2.4), (2.5) and (2.6), results in a set of track length variations as depicted in Figure 2.9. The results show that the physical track length is fixed to desired track length, tl_{des}, but as the physical track angle, θ_{cl}, varies, the required total track length varies. Table 2.2 summarises the results from the calculation for each combination of m and n in terms of the minimum and the maximum length of required track as

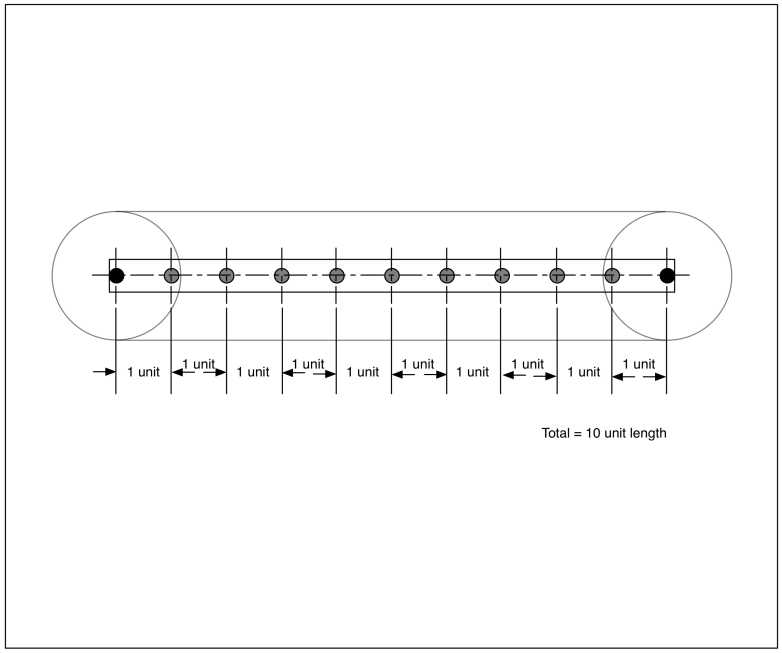
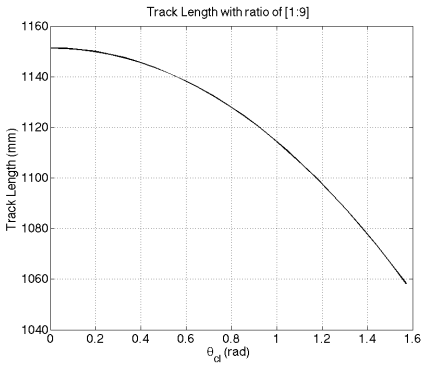


Figure 2.8: Explanation of track arm with ten unit length. The dashed line indicates the horizontal centre line component of the arm and the grey-dot denotes possible location of the rotation axis.

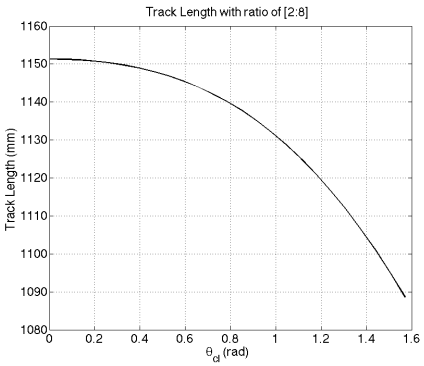
well as the difference between these two values. This difference implies that the track's material must be subject to deformable material properties. However, it is stated in Section 2.1.2, the track used on the proposed VGTV is non-deformable.

From Table 2.2, the combination of m and n with value $(4:6)$ has the least difference between the maximum and minimum length of the track. However, observing the $(m:n)$ relation with track length given in Figure 2.9d, the maximum track length is given at $\theta_{cl} \approx 1.2\,\text{rad}$ before reduces the track length at $1.2\,\text{rad} < \theta_{cl} \leqslant \frac{\pi}{2}\,\text{rad}$. The second least difference is given by the combination of $(3:7)$ but it slightly increases the track length from initial position (flat with $\theta_{cl} = 0$) to $0.7\,\text{rad}$ before drastically reducing the length at $0.7\,\text{rad} < \theta_{cl} \leqslant \frac{\pi}{2}\,\text{rad}$. In these two events, it is difficult for any actuator to compensate with the sudden changes of the track length because the track length does not monotonically increase with θ_{cl}. The third least difference between the minimum and maximum length is with combination of $(2:8)$ where the inner arm distance is shorter than the outer arm distance. In mechanical design point of view, the big difference between inner arm distance and outer arm distance results in poor torque distribution because load (track tension) are located at both ends of the arm. For this reason, the combination of $(5:5)$ seems to be the most appropriate and easier to control because the track length gradually increase $0\,\text{rad} < \theta_{cl} \leqslant \frac{\pi}{2}\,\text{rad}$.

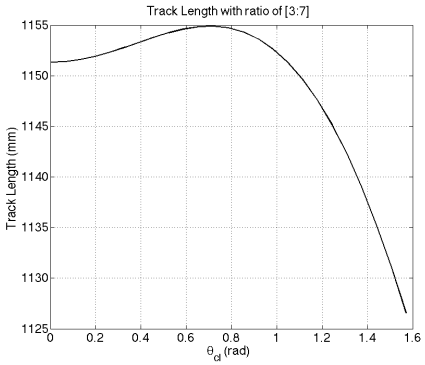
With the combination of m and n at $(5:5)$, the rotation axis is located exactly at the centre of the track's arm. Substituting the parameters of x_1, x_2 and D_w as defined in Table 2.1 into equation 2.2, the relation between track length and $(m:n)$ within $0\,\text{rad} < \theta_{cl} \leqslant \frac{\pi}{2}\,\text{rad}$ is



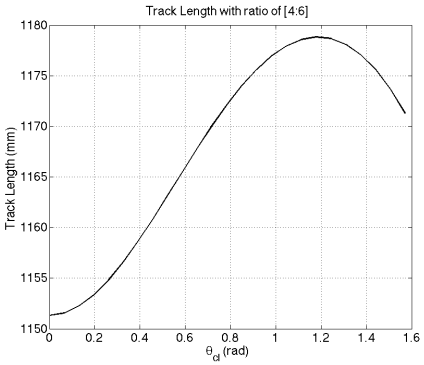
(a) 1 : 9



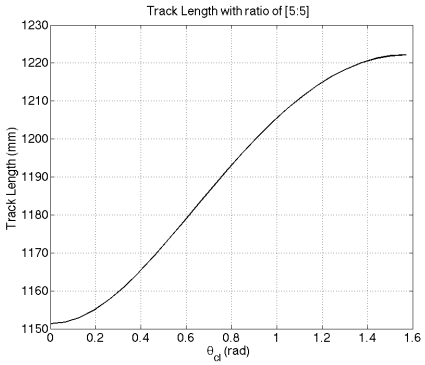
(b) 2 : 8



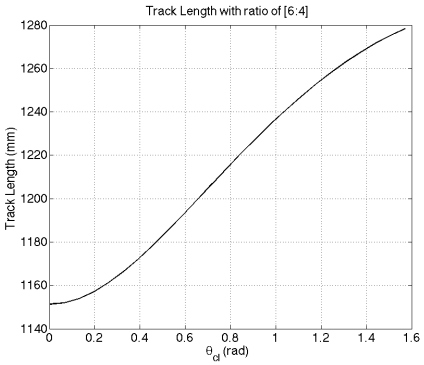
(c) 3 : 7



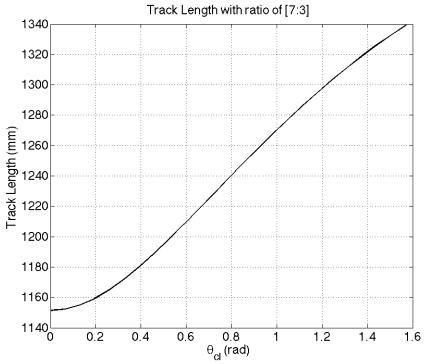
(d) 4 : 6



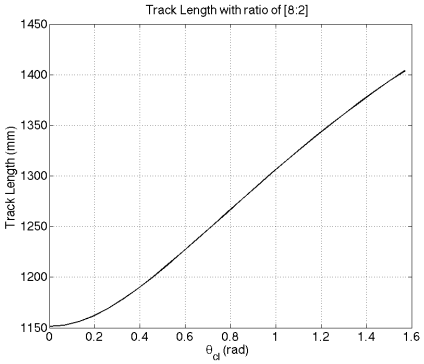
(e) 5 : 5



(f) 6 : 4



(g) 7 : 3



(h) 8 : 2

Figure 2.9: Comparison of track length changes with different combination of $m : n$. The analysis is done without consideration of parameter δ in order to study the appropriate combination of $m : n$.

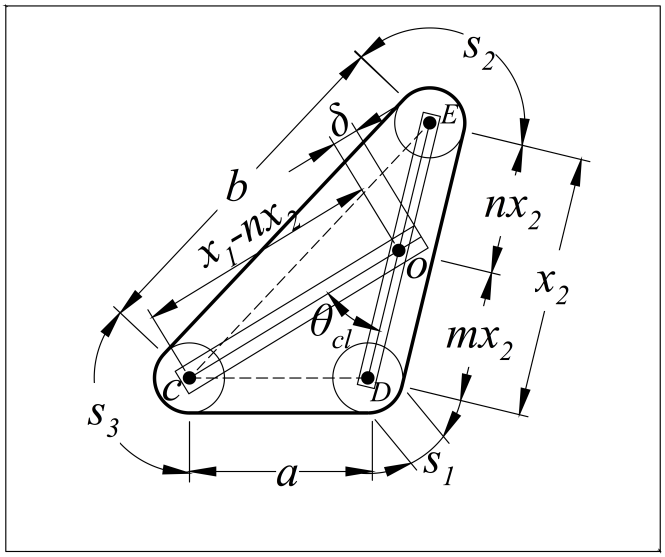


Figure 2.10: Climb configuration with parameter of δ. This parameter will be used in order to keep the track length constant.

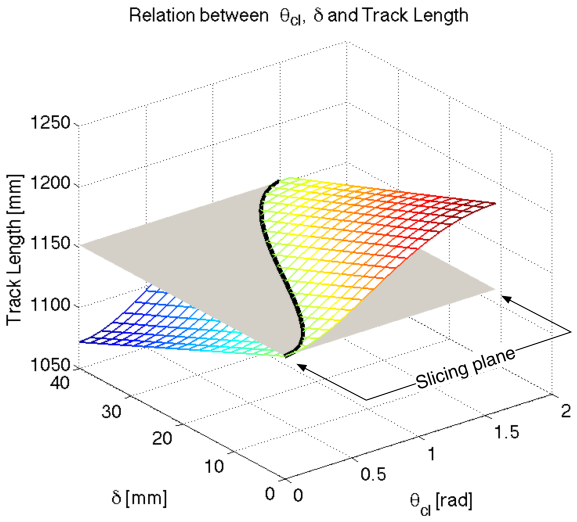
characterised in Figure 2.9e. Clearly, the track length monotonically increases with θ_{cl} as the VGTV varies its geometric configuration.

2.2.3 Formulation of the relationship between leadscrew and θ_{cl}

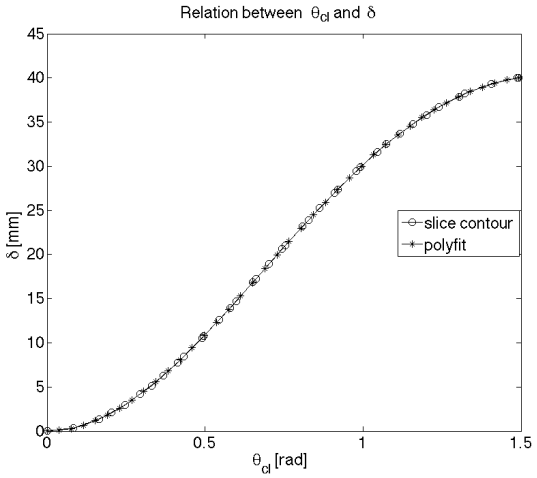
As described earlier in Section 2.1, the track material used for the proposed VGTV design is not capable to tolerate the track's elongation. For that reason, an additional parameter is introduced to control the required track length to within acceptable bounds to avoid undue tension on the physical track. In other words, as the physical VGTV varies its configuration, the track length must be equal to the desired track length given in equation (2.1). Figure 2.10 shows the geometric formulation of the VGTV in climbing configuration with an additional track tension control distance, δ.

From Figure 2.10, the new formulation of the track length within the range $0\,\text{rad} < \theta_{cl} \leqslant \frac{\pi}{2}\,\text{rad}$ is given by

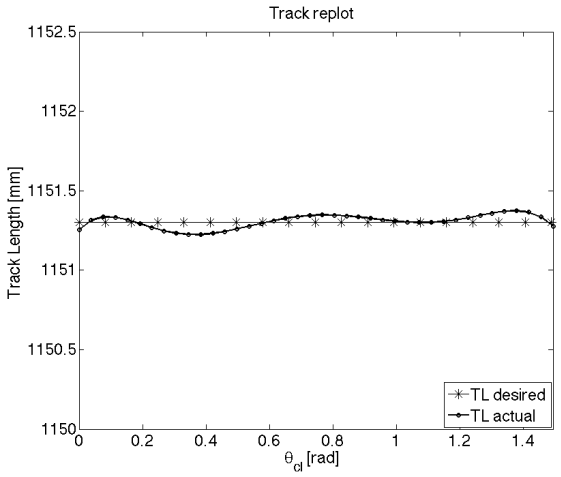
$$tl_\delta \quad = \quad x_2 + a_\delta + b_\delta + s_{1_\delta} + s_{2_\delta} + s_{3_\delta}$$



(a) Relationship of θ, δ and track length variations.



(b) Relationship θ and δ at a constant track length value.



(c) Track replot.

Figure 2.11: Track length geometry modeling. The mesh in Figure 2.11a illustrates the relationship between θ_{cl} and δ with variations of track length value. Slicing through the mesh at tl_{des} gives relationship of θ_{cl} and δ at tl_{des}. This relation is important to keep the track length constant (Figure 2.11c).

where parameters a, b, s_1, s_2 and s_3 are derived using law of cosine

$$\begin{aligned}
a_\delta &= \sqrt{(mx_2)^2 + (x_1 - nx_2 - \delta)^2 - 2(mx_2)(x_1 - nx_2 - \delta)\cos(\theta_{cl})} \\
b_\delta &= \sqrt{(nx_2)^2 + (x_1 - nx_2 - \delta)^2 - 2(nx_2)(x_1 - nx_2 - \delta)\cos(\pi - \theta_{cl})} \\
s_{1_\delta} &= \frac{D_w}{2}\left\{\pi - \cos^{-1}\left[\frac{a_\delta^2 + (mx_2)^2 - (x_1 - nx_2 - \delta)^2}{2a_\delta m x_2}\right]\right\} \\
s_{2_\delta} &= \frac{D_w}{2}\left\{\pi - \cos^{-1}\left[\frac{b_\delta^2 + (nx_2)^2 - (x_1 - nx_2 - \delta)^2}{2bnx_2}\right]\right\} \\
s_{3_\delta} &= \frac{D_w}{2}\left[\pi - \cos^{-1}\left(\frac{a_\delta^2 + b_\delta^2 - x_2^2}{2a_\delta b_\delta}\right)\right]
\end{aligned}$$

As mentioned before, the total track length is fixed to the desired track length, tl_{des}, therefore

$$\begin{aligned}
tl_\delta &= tl_{des} \\
tl_{des} &= x_2 + a_\delta + b_\delta + s_{1_\delta} + s_{2_\delta} + s_{3_\delta}
\end{aligned} \tag{2.8}$$

With the value of desired track length set by equation (2.1), $tl_{des} = 1150\,\text{mm}$, the variable parameter δ is given by substituting x_1, x_2 and D_w defined in Table 2.1 into equation (2.8). In the computer simulation, δ_{max} is calculated as

$$\delta_{max} \approx 40.20\,\text{mm}$$

when $\theta_{cl} = 90\,\text{deg}$.

Further simulations were used to determine the required relationship between θ_{cl} and δ. Simulation result shows that a constant value of track length is achievable to satisfiy the value of tl_{des} throughout track transformation, within the range of $0\,\text{rad} < \theta_{cl} \leqslant \frac{\pi}{2}\,\text{rad}$. In order to eliminate the unwanted track's material deformation and achieved tl_{des}, δ needs to be tuned automatically. Mesh plot in Figure 2.11a shows the variation of track length required for each combination of θ_{cl} and δ. From the knowledge given by computation in equation (2.1), slicing through the mesh horizontally along $tl_{des} = 1150\,\text{mm}$, results in a 2-dimensional plot (Figure 2.11b) of the relation between θ_{cl} and δ. Then, an approximate curve fitting method is used to obtain a polynomial relation that is used to determine the tuning parameter of δ when flipper angle is θ_{cl}. The final result of the simulation is shown in Figure 2.11c which clearly shows that tl_{act} is approximated as tl_{des} with error $\pm 0.1\text{mm}$. The function $\delta(\theta_{cl})$ is given by

$$\delta(\theta_{cl}) \approx 4.12\theta_{cl}^5 - 8.93\theta_{cl}^4 - 22.68\theta_{cl}^3 + 58.69\theta_{cl}^2 - 1.08\theta_{cl} + 0.04 \tag{2.9}$$

Note that equation (2.9) is not unique because it is dependent on the setting of polynomial order in the computation. Substituting θ_{cl} (where $0 \leqslant \theta_{cl} \leqslant \frac{\pi}{2}$ rad) into equation (2.9), then plotting on same plot of Figure 2.11b results in identical plot where curve with "o" denotes the plot resulting from slicing the mesh while "solid-dot" denotes the plot resulted from substituting θ_{cl} into equation (2.9).

In this section, the mathematical derivation of VGTV transformation associated with track length was investigated. Parameter δ was introduced to keep the track length constant at any value of θ_{cl} where the range of θ_{cl} is between zero rad to $\frac{\pi}{2}$ rad. In the next section, the actual physical design of the VGTV prototype is presented focusing on mechanical structure and general control framework.

2.3 Mechatronic Design

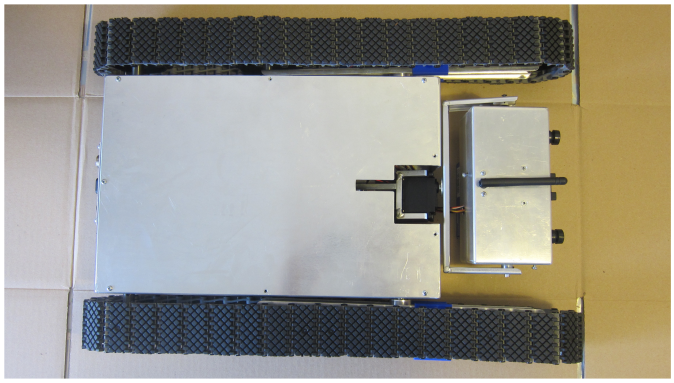
Previous analysis in Section 2.2 provides the relationship between δ and the climbing angle which is a very important consideration in order to proceed with actual mechatronic design. In the beginning of the actual mechatronic design phase, several objectives have been defined based on the tasks and mobile robot types described in Murphy et al. (2008). The primary design objective is to adapt with the *search* and *reconnaisance* task which clearly involves negotiation with various types of terrain and obstacles. In addition, size and mobility of the mechanical design must impact the primary design objectives. In this section, the design of the proposed VGTV mobile robot is divided into two parts. One is the mechanical structure and the other is the control unit and sensing capabilities.

2.3.1 Mechanical structure

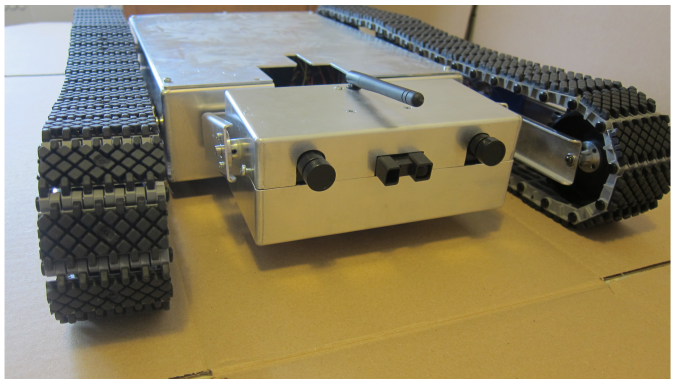
As defined in Casper and Murphy (2003), the proposed VGTV in this chapter is considered as a *man-packable* mobile robot. Illustrated in Figure 2.12, the proposed VGTV mobile robot is designed in compact size, 300 mm ×150 mm ×50 mm, and composed of three main parts: (1) main body, provides enclosure for internal devices; (2) left and right identical track arms; and (3) stereo vision system (SVS) casing, enclosure for SVS unit.

The key design criteria are:

- Unlike existing VGTV, the prototype is capable to construct different θ_{cl} on both sides of the track arms to avoid mishap due to tip-over (Figure 2.13b).
- Similarly, it is also able to perform symmetrical climbing angle, θ_{cl}, on both sides of track arms (Figure 2.13a).



(a) Top view.



(b) Front view.

Figure 2.12: Mechanical structure of the prototype.

- In order to achieve specific θ_{cl} and control the track's tension, two sets of leadscrew mechanisms (Figure 2.14) are used whenever the track arms change their angles to adapt with terrain. Figure 2.16 shows the arrangement of actuators inside the main chassis.

Instead of using variable planetary wheel system to control the track length as found in Goldenberg et al. (2009), the prototype utilises a simple but unique linear leadscrew mechanism to control the track's tension. The variable track geometry is the result of the track arm transformation at the specified θ_{cl} (Figure 2.13a). On top of that, a constant total track length (1150 mm) is achievable throughout geometric transformation and control by leadscrew mechanism. The rotating motion of the track arm which is actuated by servo drive is attached on the leadscrew mechanism. Having said that, as the servo drive rotates the track arm at the specified θ_{cl}, the leadscrew moves or adjusts the position of the servo drive by adjusting the leadscrew (Figure 2.14).

Additionally, two rechargeable twelve volt batteries connected in series to supply power to all the actuators and electronic boards and provides one hour overall endurance. The prototype employs eight motor actuators: (1) two twelve volt DC motors (motor 1 and 2) located at rear of the main body chassis to drive both tracks; (2) two twelve volts DC motors (motor 3 and 4) located at left and right inside the main chassis to drive the leadscrew parts; (3) two heavy

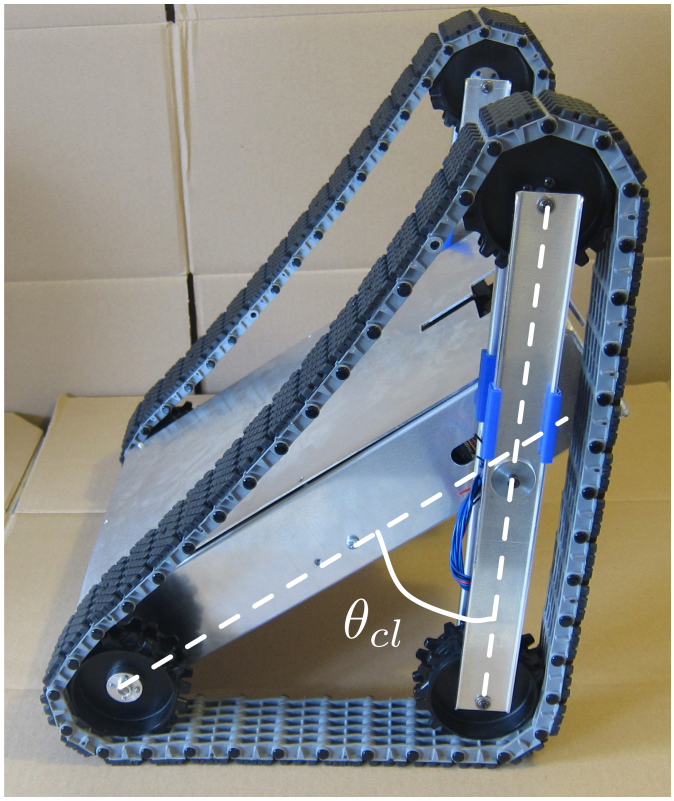
duty servo motors (motor 5 and 6) to construct appropriate θ_{cl} for both track arms; and (4) two standard servo motors to adjust the orientation of the SVS enclosure.

2.3.2 Lowest Control Layer

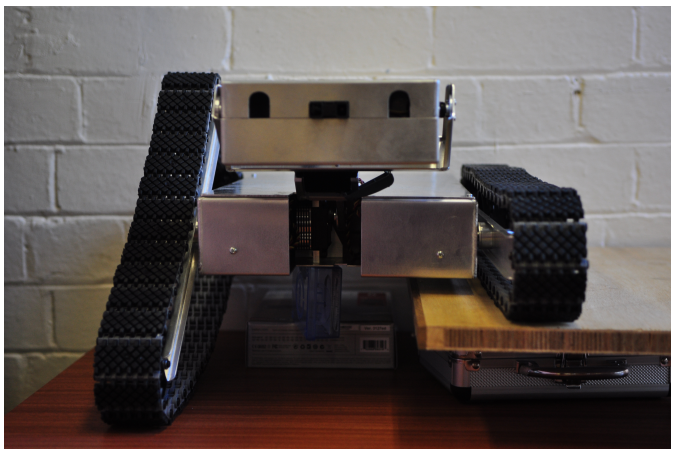
Besides the actuators explained in previous section, another fundamental issue in mechatronic design is to determine the control framework based on the level of autonomy. In this chapter, the VGTV is assumed to be remotely-operated. The command that indicates basic movements (e.g., forward-reverse and left-right) are under minimal control by an operator. In order to minimise the operator load, VGTV will autonomously adapt the track configuration to the terrain. To fullfill this requirement, this chapter proposes a three layer control architecture as depicted in Figure 2.15.

The control architecture itself is a top-down approach in which the highest level control layer defines the global mission, the intermediate level control layer focuses on the VGTV's configuration control and the low level control layer executes instructions from the intermediate level control layer and translates them into the track speed. Consider, for example, the VGTV is required to travel on designated path with certain type of obstacle. In order to climb the obstacle, the VGTV is required to transform its configuration and adjust the track's tension. For that particular motion, the VGTV needs to execute two rotational motion simultanously: (1) rotate motor 3 and 4; and (2) construct θ_{cl} via rotation of motor 5 and 6. On the other hand, if the VGTV is required to enhance stability along a side-slope (Figure 2.5c), only one side of the track arm is actuated.

As mentioned before, the low-level control is described as the local control system which processes input from sensors (or manual controller) and develops appropriate command signals and transmits them to the designated actuators. In general, the low-level control system for the proposed VGTV mobile robot is dedicated to actuator motion control. From Figure 2.16, all DC motor actuators are connected to the designated motor drivers and controlled by a central control unit employed by an *Arduino MEGA 2560* board. Meanwhile, servos are directly connected to the output pins of the *Arduino MEGA 2560*. The posture and orientation of the prototype is determined by a *Sparkfun 9-degree-of-freedom Razor Inertial Measurement Unit (IMU)*. Figure 2.17 illustrates the posture of the VGTV prototype identified by these sensors. The reading of the sensor indicated by the value of roll, pitch and yaw with body orientation and heading direction are visualised by the red square and green arrow respectively. Additionally, the back plane connector port shown in Figure 2.18 provides connection interface for (1) manual controller; and (2) external computer for serial access and programming mode via two USB ports.



(a) Symmetrical climbing.



(b) Independent track arm.

Figure 2.13: Two basic body actuation resulted by the adjustment of the leadscrew mechanism.

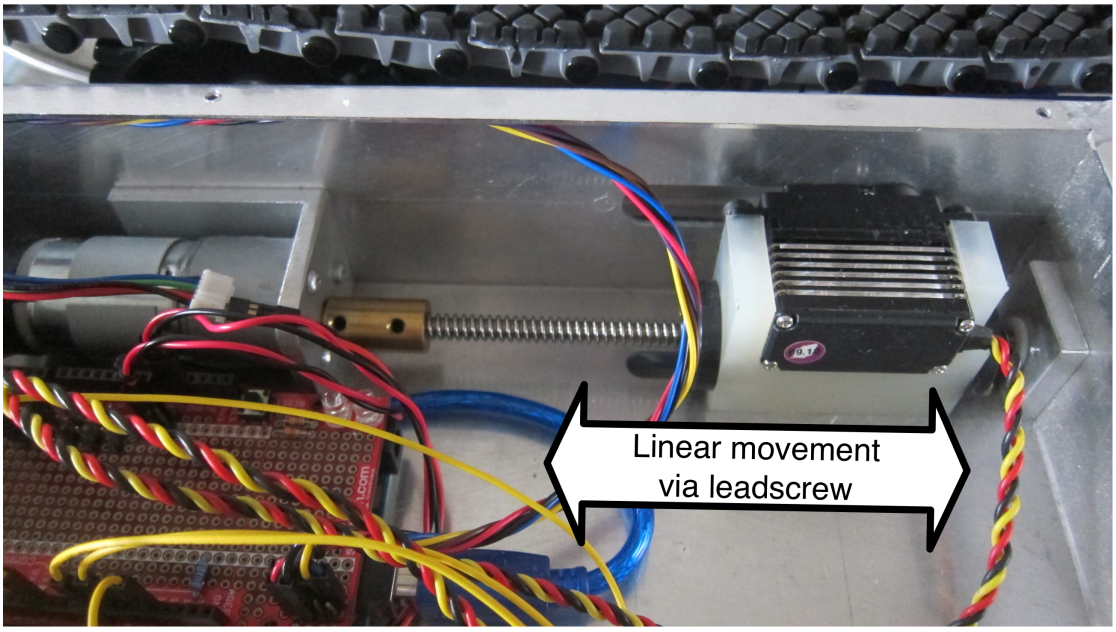


Figure 2.14: Leadscrew mechanism.

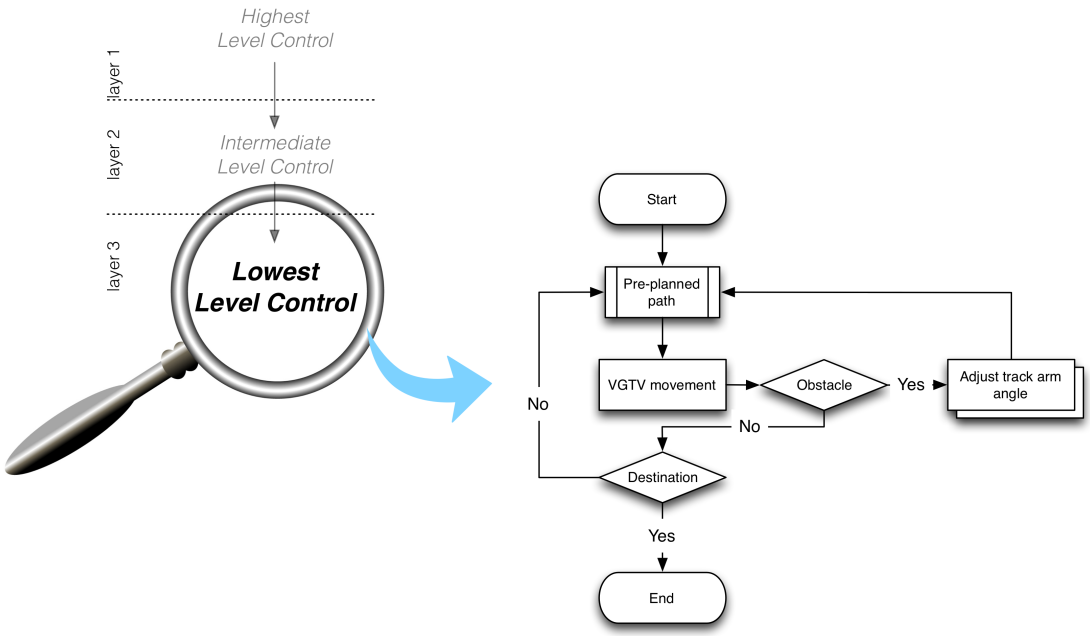


Figure 2.15: The flowchart is the mobile robot control architecture with the magnifier indicating that the lowest level control layer is the focus in this chapter. The internal operation logic in low-level control layer is illustrated by the right hand side flowchart.

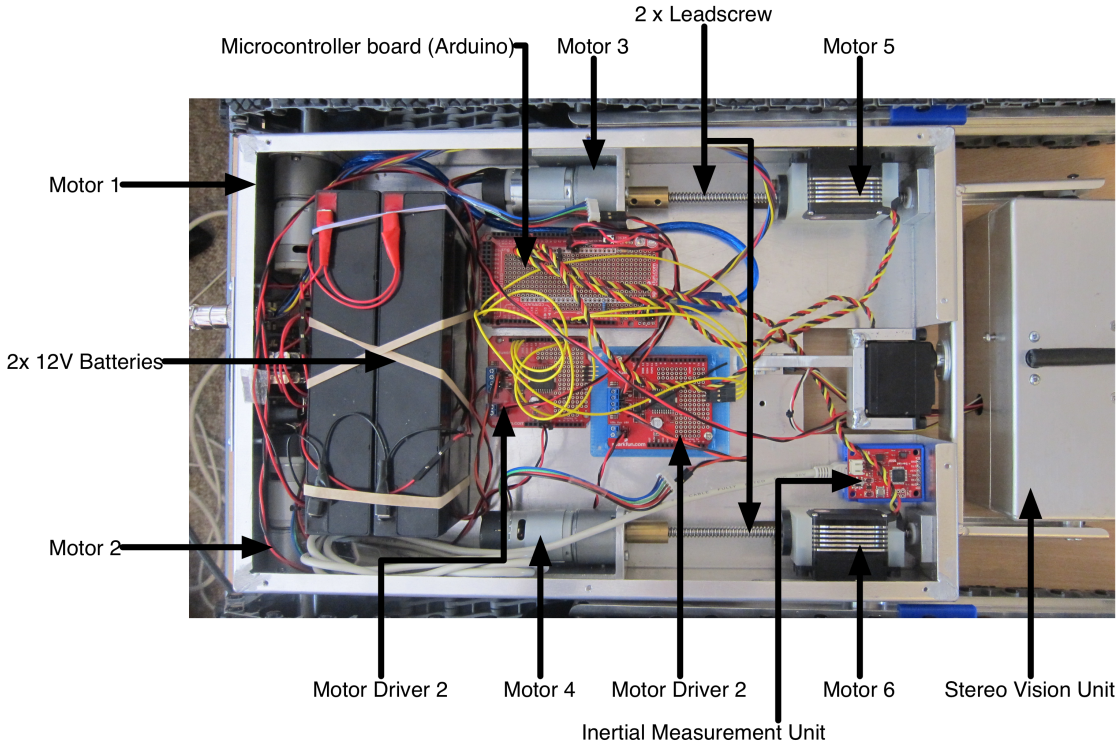


Figure 2.16: Actuators, control unit and onboard sensor.

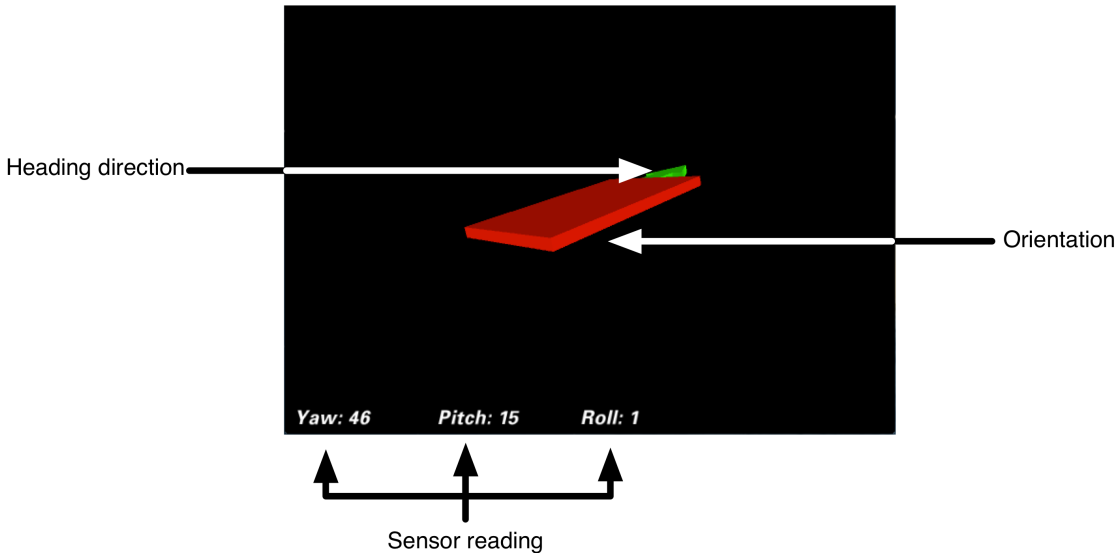


Figure 2.17: Sample IMU image. Software interface is developed by Peter Bartz and Sascha Spors, released under GNU GPL (General Public License) Bartz (2012).

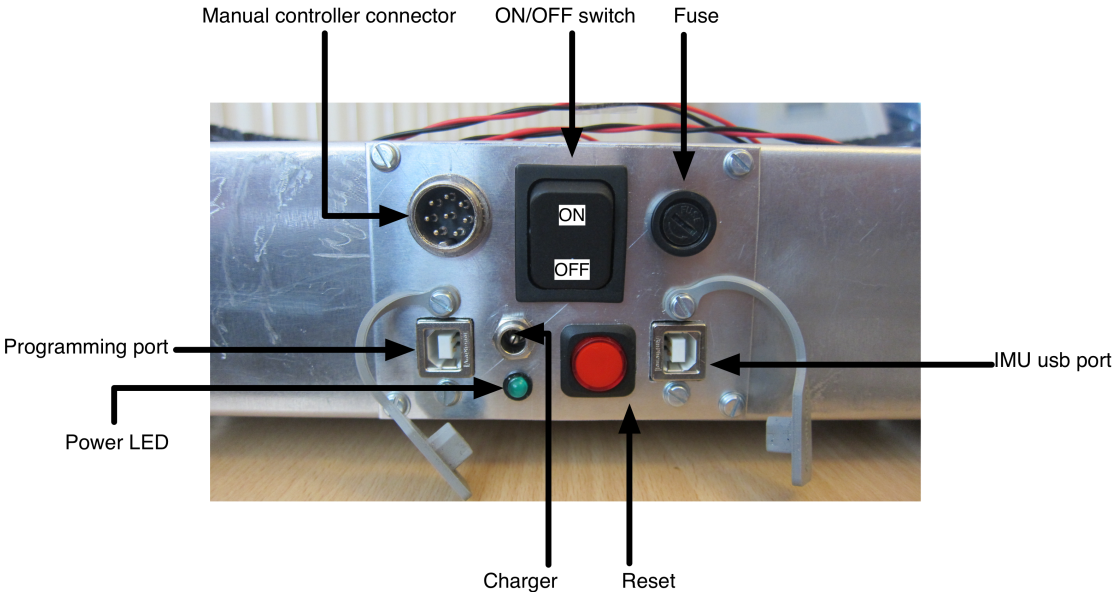


Figure 2.18: Back panel connector ports.

2.4 VGTV Transformation

In order to evaluate the proposed VGTV design, several experiments were conducted. The objective of the experiment is to observe the behaviour of the proposed VGTV on several types of obstacles/terrains. Moreover, the orientation of the main chassis was recorded using *Sparkfun 9-degree-of-freedom Razor IMU AHRS*. The obstacles/terrains used in this experiment were:

- *Ramp*: The simplest obstacle to represent flat inclination with specific slope angle.
- *Curb*: Vertical inclination where the obstacle height is higher than height of the VGTV.
- *Stair*: The most tested obstacle feature in indoor and outdoor environment. VGTV is required to ascend and descend the staircase.
- *Rubble*: Represents a complex terrain where VGTV is tested on two aspects; (1) ascend/descend; and (2) stability on side-slope.

A computer was used in the experiment to display the IMU measurement as depicted in Figure 2.19. The sensor which was directly connected to a computer measures the orientation of the prototype when it travel along the obstacles.

The VGTV was configured into flat configuration as the initial posture (start position). Note that the SVS unit was detached from the mobile robot platform. This is because the function of this device is purely for capturing live video of its surrounding. Nonetheless, in this experiment, non-video input was utilised as an input to the VGTV. Despite that, the command for the VGTV's movement is set by an external handheld controller. The results were organised in terms of VGTV's transformation pictures according to the experiment's difficulty level described



Figure 2.19: Experimental setup.

earlier. For convenience, each experimental result will be presented in two figure groups. The first group is photographs of the physical transformation. Meanwhile, the second group is the sensor reading in the form of visualisation of main chassis orientation measured by the *Sparkfun 9-degree-of-freedom Razor IMU AHRS*.

Photos shown in Figure 2.20 illustrate the behaviour of the prototype when it traveled on a ramp. The ramp which is made by a flat panel with 0.8 metre length and elevated by 0.3 metre height (at one side) resulting in approximate 20 degree inclination. Within this experiment setup, the VGTV was not required to construct θ_{cl} via its track arms. On the other hand, in the second experiment, the VGTV was expected to overcome a simple curb with the dimension of 0.17 metres height. The motion of the VGTV and behaviour of the track arms is shown in Figure 2.22. In order to conquer the curb, the VGTV is required to adjust its θ_{cl} to approximately 25 deg (0.4363 rad). Substituting 0.3491 rad into equation (2.9) yielding

$$\begin{aligned}\delta\left(0.4363\right) &\approx 4.12\left(0.4363\right)^5 - 8.93\left(0.4363\right)^4 - 22.68\left(0.4363\right)^3 + 58.69\left(0.4363\right)^2 \\ &\quad - 1.08\left(0.4363\right) + 0.04 \\ &\approx 8.6\,\text{mm}\end{aligned}$$

The result from the above solution means that both sides of the leadscrews were adjusted by approximately 20 mm in linear motion resulting in symmetrical climbing motion (Figure 2.22b).

Inspite of *ramp* and *curb*, another obstacle feature involved in this experiment was a staircase. In general, staircase is the most common obstacle tested on many mobile robots with a goal



(a) Before climbing. (b) Climbing.

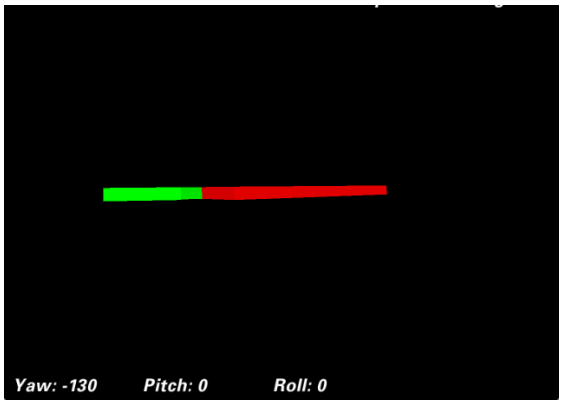
Figure 2.20: The proposed VGTV mobile robot was ascending a simple ramp with 20 deg slope. It was not required to change its configuration due to the flatness of the slope.

to investigate the stair climbing capability. Even though the staircase structure is physically complex compared to the curb, the VGTV was expected to behave similarly to the curb testing. Such experiments were performed and shown in Figure 2.24 and Figure 2.26. Additionally, the IMU readings are recorded in Figure 2.25 and Figure 2.27 for ascending and desending stair accordingly. In comparison between these scenarios, descending a staircase is an easy task for VGTV due to the the robustness of the design where it can simply descend in flat configuration. In constrast, it was required to transform its configuration to overcome the staircase, particularly at the beginning of the climbing. The stair climb rules of the prototype are summarised as follows:

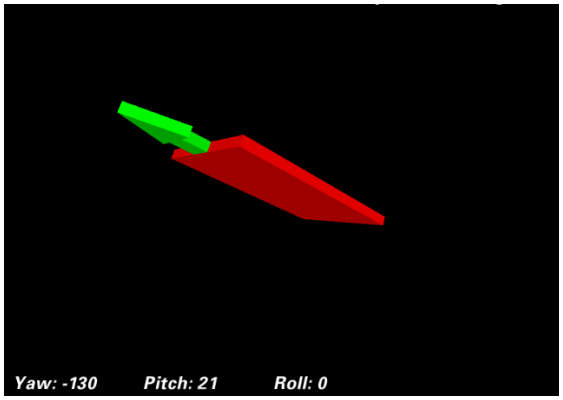
- flat to climb with $\theta_{cl} \approx 0.6109\,\text{rad}$ (Figure 2.24a), then
- gradually reducing θ_{cl} (Figure 2.24b) and continue ascending, and finally
- transform to flat configuration when it reaches the top (Figure 2.24c).

From the results (photos), both track arms are transformed in symmetrical manner to conquer the staircase.

The final experiment was the rubble where the VGTV is required to traverse on that particular setup (Figure 2.28). There were two elements tested in this experiment. First, the VGTV was tested on climbing ability, and secondly it must regain stabilty to prevent tip-over. Similar to the staircase experiment, VGTV was started in a flat configuration. Then, it transformed its shape via track arm angle of $\theta_{cl} \approx 0.4363\,\text{rad}$ to start climbing before slowly reconfigured itself into flat shape. Due to the complexity of the rubble setup, the VGTV was forced to move along a side-slope. For this type of obstacle feature, the left track was adjusted by $\theta_{cl} \approx 0.4363\,\text{rad}$ in order to prevent tip-over as depicted in Figure 2.28. IMU readings related to rubble experiment are recorded in Figure 2.29.



(a) IMU reading before climbing.



(b) IMU reading while climbing the ramp.

Figure 2.21: IMU reading related to the ramp obstacle motion in Figure 2.20.



(a) Before climbing.



(b) Ready to climb.



(c) Start to overcome.

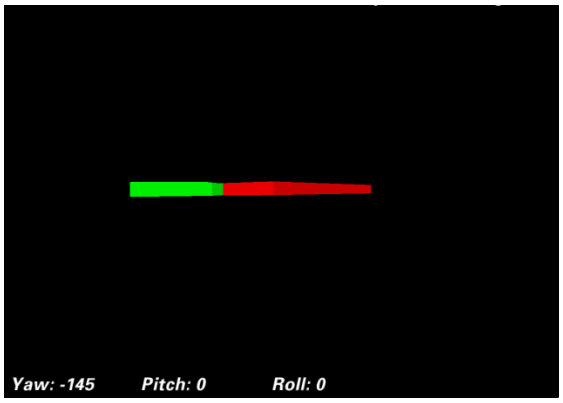


(d) Climbing the curb.

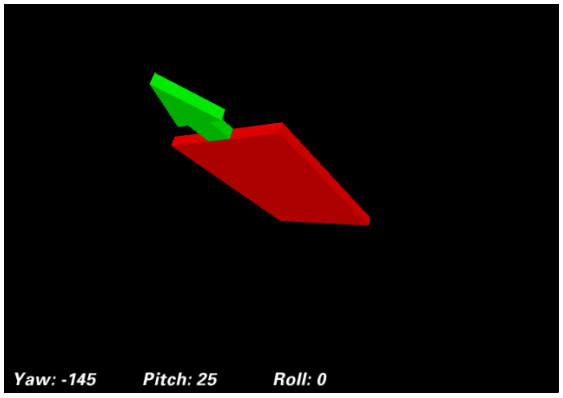


(e) After climbing.

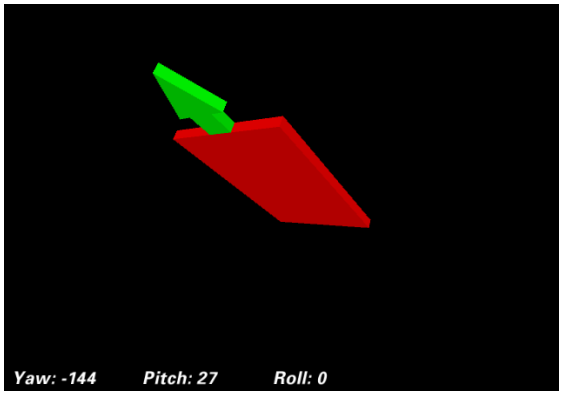
Figure 2.22: The proposed VGTV mobile robot overcoming a curb with symmetrical climbing configuration. The curb is 0.17 metres height which is higher than the VGTV track's height.



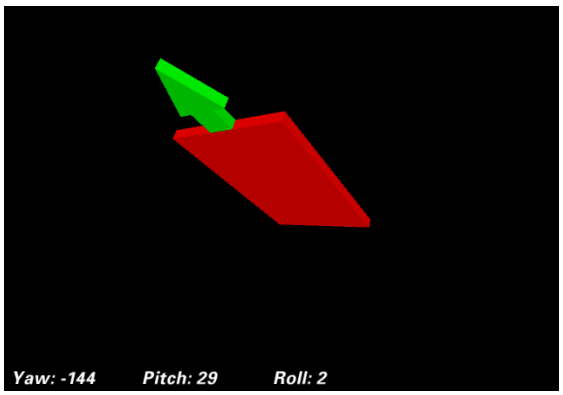
(a) Before climbing.



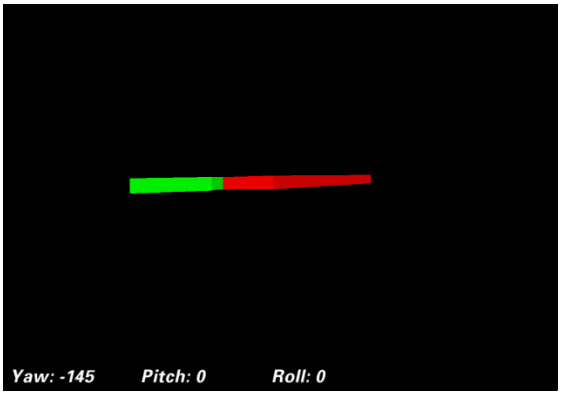
(b) Ready to climb.



(c) Start to overcome.



(d) Climbing the curb.



(e) After climbing.

Figure 2.23: IMU reading related to the curb overcoming motion in Figure 2.22.



(a) Before climbing.

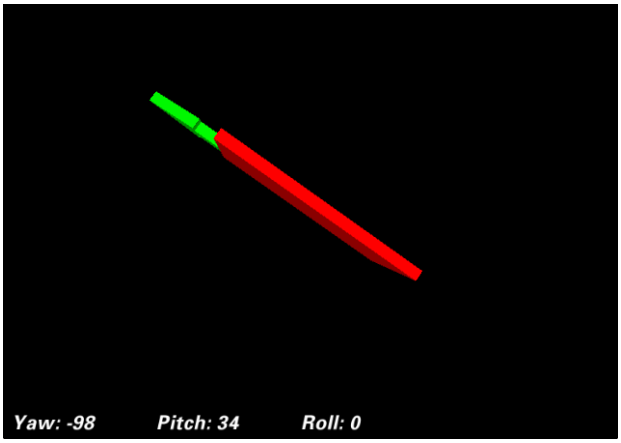


(b) Climbing.

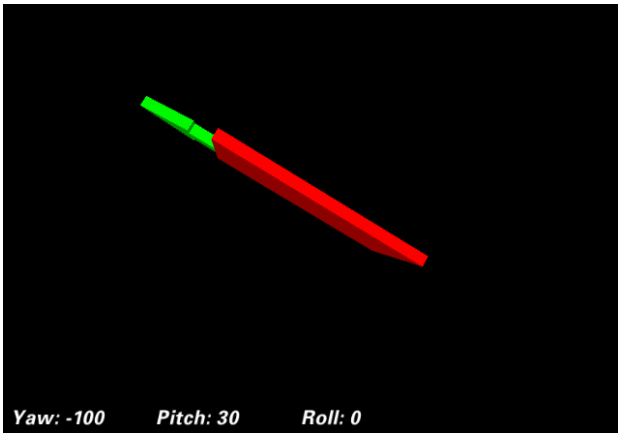


(c) After climbing.

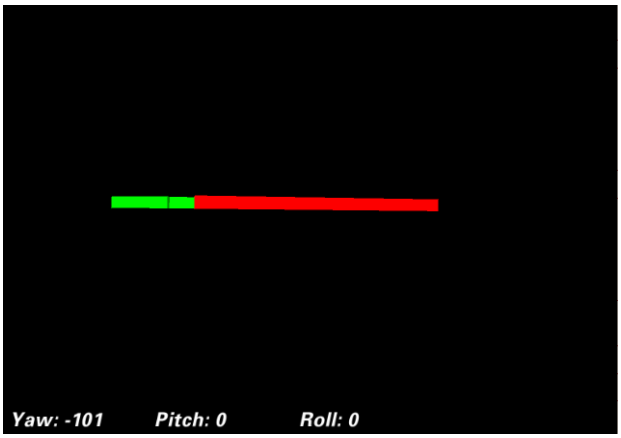
Figure 2.24: The proposed VGTV mobile robot ascending a flight of stairs. Started from a flat configuration, the VGTV was required to change its shape into climbing configuration then gradually reconfigured to flat configuration as soon as its finished climbing.



(a) Before climbing.



(b) Climbing.



(c) After climbing.

Figure 2.25: IMU reading related to ascending a staircase.



(a) Before descending.

(b) Start to descend.

(c) Descending.

(d) After descending.

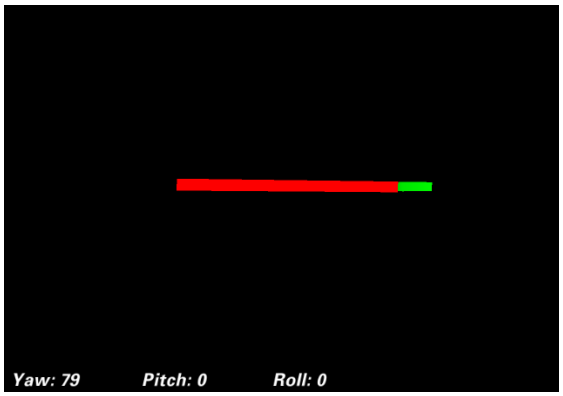
Figure 2.26: The proposed VGTV mobile robot was descending a staircase. In contrast to climbing the staircase, descending motion is simpler because it can travel down with single configuration, in this demonstration flat configuration was used.

2.5 Discussion and Concluding Remarks

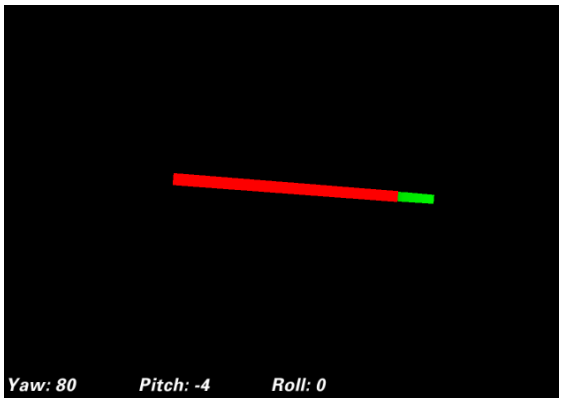
This chapter was written with the objective to develop a new mobile robot platform that is able to travel on various types of terrain motivated by literature of single-track VGTV configuration described in Section 2.1.1. As mentioned in Section 2.1.1, the fundamental problem of single-track VGTV is to maintain a constant total track length throughout geometric transformation. Prior studies also indicated the importance of keeping the track's tension within transformation. However, very little was found in the literature review on the question of how to increase track's grip (traction) when traveling along an obstacle. Moreover, none of the literature address the problem of VGTV moving along a side-slope and overcoming tip-over mishap.

Instead of developing a new mobile robot system, this study is also set out with the aim of enhancing VGTV capability by introducing independent track arm control. Unlike the track's tension control mechanism proposed by Paillat et al. (2008), the prototype in this chapter employs two leadscrew mechanisms mounted on both sides of the platform in order to independently control the track's tension during the transformation. On top of that, a constant total track length is achievable at any value of θ_{cl} by adjusting the linear position of servo actuators which are also part of the leadscrew assembly.

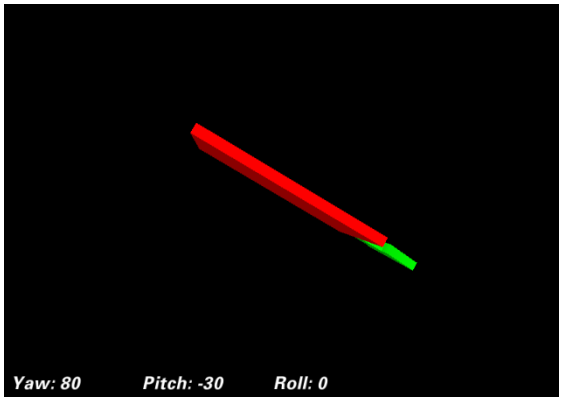
Motivated by the track length analysis in Kim and Lee (2007), the simulation result in Section



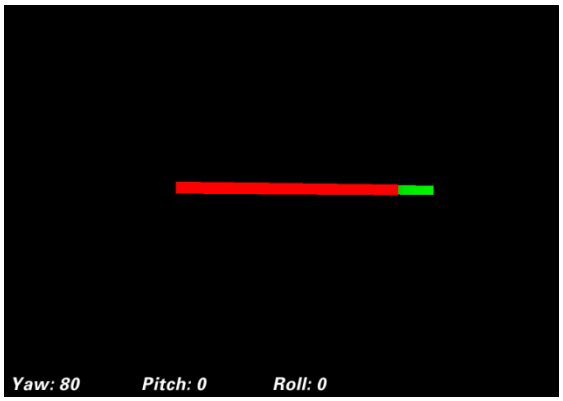
(a) Before descending.



(b) Start to descend.



(c) Descending.



(d) After descending.

Figure 2.27: IMU reading related to descending a staircase.



(a) Before climbing.



(b) Ready to overcome.

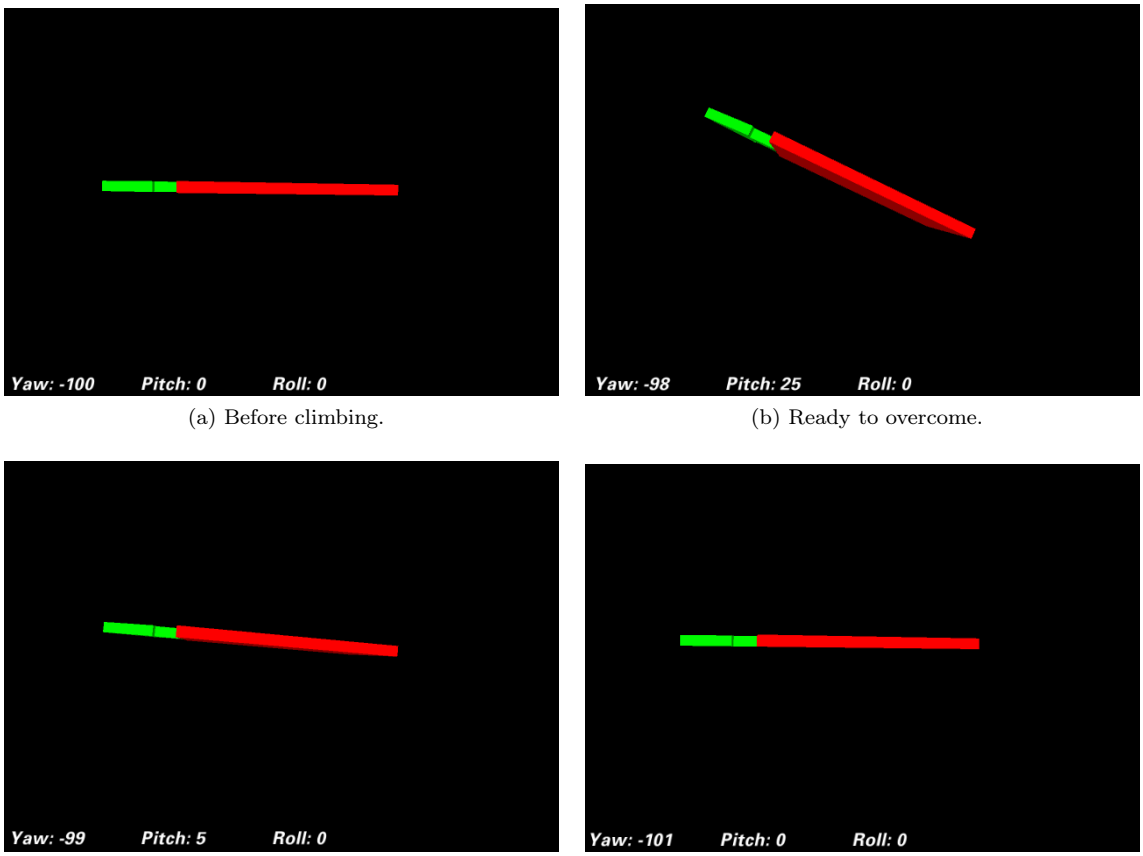


(c) Climbing the rubble.



(d) Complete climbing the rubble.

Figure 2.28: The proposed VGTV mobile robot was 3-dimensional complex terrain (a rubble).



(a) Before climbing.

(b) Ready to overcome.

(c) Climbing the rubble.

(d) Complete climbing the rubble.

Figure 2.29: IMU reading related to three-dimensional complex terrain (a rubble).



Figure 2.30: The proposed VGTV mobile robot is traveling along a side slope of the rubble. The prototype was managed to prevent tip-over mishap utilising independent track arm control.

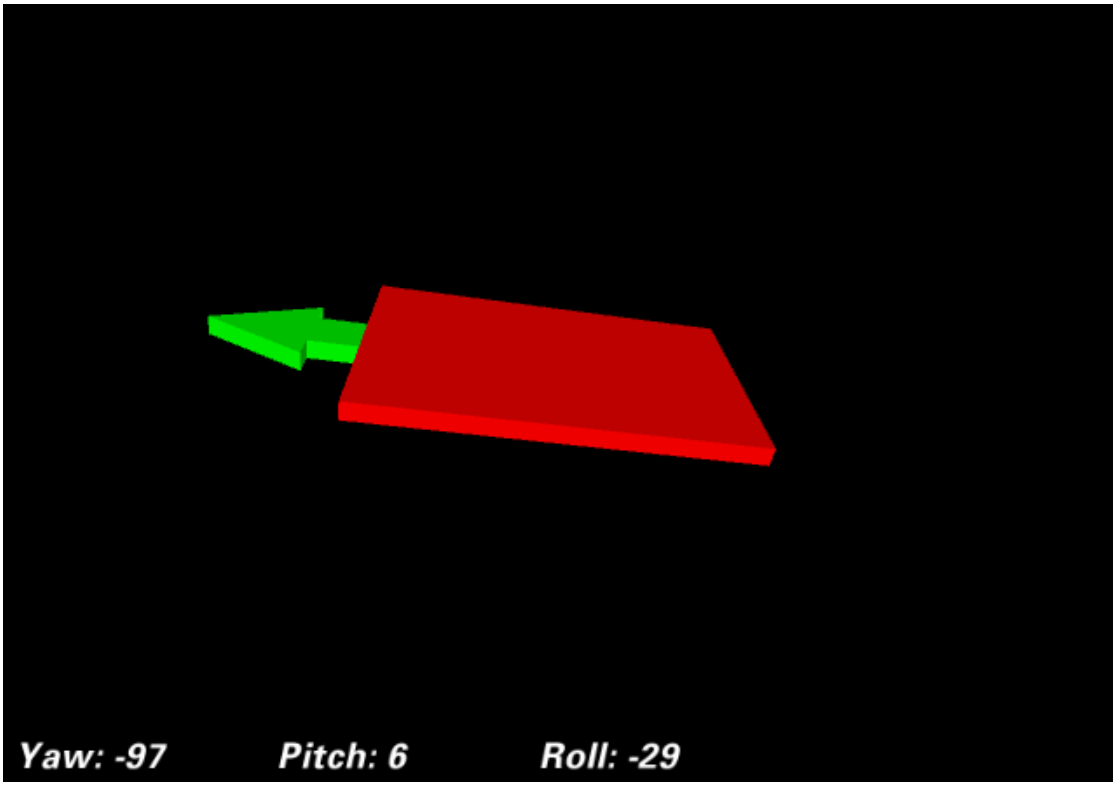


Figure 2.31: IMU reading related to side-slope motion.

2.2 shows that the actual total track length is approximately close to the desired track length with small error of ± 0.1 mm. This error, however, can be considered as an additional traction to enhance grip on the surface. Moreover, the results from the physical experiments in Section 2.4 show that the prototype is able to negotiate with various types of obstacles. In contrast to Paillat et al. (2008), the VGTV is able to enhance body stability in case of side-slope (Figure 2.30) to prevent tip-over mishap.

However, more research on this topic needs to be undertaken especially on the selection of the appropriate hardware and actuators. There were several issue raised from the experiment such as:

- chassis-to-terrain surface clearance was too small resulting in the mobile robot getting stuck on obstacles;
- rear driving motors are not powerful enough to drive and turn the platform especially if it gets stuck; and
- servo mechanism used on the prototype does not deliver enough stall torque to resist track's tension.

Some suggestions can be made to overcome these problems such as:

- increase the wheel diameter as well as optimising the chassis dimension to ensure sufficient

chassis-to-terrain surface clearance;

- utilise a full-drive-system (or four-wheel-drive system) on the prototype by implementing four motors to drive four track sprockets;
- replace the servo mechanism with another motor with higher stall torque capability; and

In the next chapter, we will discuss the overall control architecture with adoption of the proposed VGTV mobile robot configuration. The discussion will also include a framework dedicated for independent track control of the VGTV. The knowledge from this chapter, in particular the ability of a mobile robot to vary its geometric configuration will be utilised. The future framework will include the decision-making process in case the mobile robot required to operate in complex, unstructured and unforeseen environments.

Chapter 3

Control Architecture

3.1 Introduction

In the previous chapter, the mechatronic design of a variable geometry tracked vehicle (VGTV) mobile robot platform has been presented. It has the ability to vary its geometrical configuration to adapt to uneven terrain based on the adjustment of independent leadscrew mechanisms installed on both sides of the track. Moreover, the physical behaviour of the VGTV over certain types of predefined uneven surfaces is also presented. In this chapter, a complete control system architecture for the VGTV which travels on an unknown and unstructured environment is addressed. In order to define the overall control structure for the proposed mechatronic systems in Chapter 2, the VGTV is assumed to be able to negotiate with various type of obstacles.

As mentioned in the previous chapter, there are many challenges in order to develop a complete robotic system for urban search and rescue (USAR). Defined as one of the challenges, the mobile robot control architecture brings the important role to ensure the mobile robot utilised in USAR operation is able to deliver a given mission (Section 1.1.2) successfully. For instance, a mobile robot which is assigned to a *reconnaissance and mapping* task must be able to provide an accurate map of the interior environment of collapsed structures. On the other hand, a *search* mobile robot must be able to negotiate various types of terrain whilst traversing and searching for trapped survivors. It is the responsibility of the overall control architecture of the specific mobile robot platform to process a designated task in order to accomplish the designated mission.

Currently, most mobile robot systems available for USAR operation are required to communicate with an operator, especially for decision-making scenario (Murphy, 2002). One example, is reported by Marques et al. (2007), where a *search* mobile robot named as 'RAPOSA' depended on commands from an operator to travel between rubble. Nevertheless, it is expected that mobile robot system for USAR operation becomes more independent particularly in decision-making scenario (Murphy, 2004). Having said that, a mobile robot system may have a certain level of decision-making capability to establish autonomous adequacy, regardless of the auton-

omy level, especially in unknown and unstructured environments. In this chapter, we present an overarching control architecture for USAR mobile robots.

The remainder of this chapter is arranged as follows: in Section 3.2, a comprehensive literature review about the growing body of mobile robot control architecture. Additionally, the literature emphasises on systematic control architecture methodology which motivates the control framework proposed in this thesis. On top of that, Section 3.3 defines the primary objectives of the control architecture. Furthermore, Section 3.4 formulates the control framework in terms of the degree of control hierarchy. Describing this further, Section 3.5 describes the *three-tiered layer control architecture* as well as the control schemes embedded in each layer. Finally, Section 3.6 provides concluding remarks.

3.2 Motivations

In general, the control architecture is highlighted as the central operation to the entire discipline of robotic system, regardless of their classification. Moreover, the control scheme for completely autonomous robot systems must consist of several complex and complicated control processes (Bruemmer et al., 2002). Therefore, there is a serious demand to propose a systematic control approach to resolve the fundamental control architecture issue (Jacoff et al., 2002). From a USAR mobile robot perspective, it is really obvious that they are required to have a systematic control architecture because they always deal with many uncertainties such as unexpected danger, various obstacles features and sudden changes/motion of their surrounding (Liu et al., 2007). Consequently, it is worth to investigate the existing mobile robot architectures proposed for various types of applications in order to motivate a more generic control framework for USAR mobile robots.

A large and growing body of literature has investigated the impact of certain designs of control architectures onto the mobile robot systems. One of the earliest control architectures introduced is the *sense-plan-act* (SPA) concept which has been widely used in many applications (Nilsson, 1969; Simmons et al., 1997b). However, this architecture demanded higher computing power to increase the computation speed in real-time processing due to longer planning time (Miller and Stein, 2001). Therefore, Brooks (1986) and Brooks et al. (2005) suggested the *subsumption* control architecture, which normally consists of many 'processing' blocks in a single control framework. The advantage of the *subsumption* control architecture is that a 'processing' block is allowed to communicate (via connection) with another 'processing' block to achieve certain robot behaviour. In fact, the 'processing' blocks are authorised to communicate with more that one 'processing' block (Toal et al., 1996). Moreover, the capability of this control architecture

is successfully demonstrated in Brooks (1986) and Brooks et al. (2005) on their *MIT AI Lab* mobile robot.

Another systematic mobile robot control architecture is proposed by Prescott et al. (1999), utilising several layers control architecture. In comparison to SPA and *subsumption* control architecture, the layered control architecture is arranged according to the organisational hierarchy to ensure that each layer can interact properly. In general, a basic prototype of layered control architecture is organised in three-tiered architecture where the higher layers have higher organisational level of abstraction whereby the lower layer is associated with the actual behaviour of the mobile robot (Coste-Manière and Simmons, 2000). In addition, Murphy et al. (2000) suggested that mobile robot system for USAR application would apply an adjustable autonomy due to the nature of the USAR environment where interaction between mobile robot systems and operator is very important.

3.3 Aim

The aim of this chapter is to focus on developing a control architecture for mobile robot platform that is able to travel on various types of terrain, in particular the proposed mechatronic design described in Chapter 2. The proposed architecture, which is motivated by Gat et al. (1998); Ridao et al. (2000) and Williamson and Carnegie (2007), consists of three-tier control layers where the highest layer defines the global mission whereas the lower layer implements the mission in terms of robot control. In order to comply with the requirements discussed in Casper et al. (2000) and Murphy et al. (2000), each control layer is composed of several control schemes. The control schemes are engaged to process instructions from higher layer as well as to interact between layers.

Unlike the *subsumption* architecture, the control schemes embedded in the proposed control architecture introduced the *selector* function block to enhance the interaction of mobile robot systems with external entity (i.e. operator). In summary, the proposed control architecture is designed based on:

- the systematic approach of three-tiered layer control;
- the organisation of layers is according to control hierarchy;
- the dynamic autonomy between layers; and
- the requirement of interaction with selector function block to highlight external component.

Furthermore, the description of the proposed control architecture in this chapter provides some general introduction for remaining chapters in this thesis.

3.4 Control Structure Requirements

Consider the VGTV mobile robot proposed in Chapter 2 is utilised in this chapter. Apart from the ability to change its geometrical configuration, the VGTV is also able to control the track geometry independently. This ability highlights very useful feature for the mobile robot to enhance stability when traveling along a side-slope. As a result, there is a need to design a complete control scheme for the mobile robot system to control the independent track control mechanisms. On top of that, the mobile robot system is expected to have certain knowledge of the terrain/obstacle features prior to any geometric transformation.

In this chapter, a generic control architecture is proposed for USAR mobile robot, particularly for *search*, *reconnaisance* and *mapping* task. As described in Chapter 1, these three mobile robot tasks can be classified as the mobile robot navigation mission. In terms of robot architecture, the layers are arranged according to the degree of control hierarchy. Motivated by Kortenkamp and Simmons (2008) and Woodman et al. (2010), there are three important layers:

- the highest control layer - executive layer;
- the intermediate control layer - sequential layer; and
- the lowest control layer - reactive layer.

The detailed definition of each layer is defined according to their degree of control hierarchy where the higher degree of control hierarchy is described as the global control. Meanwhile, the lower degree of control hierarchy applies to the local control of the mobile robot. In short, we can assume that the global control is the mobile robot position with respect to the world environment whereby the actuators control command is the local control elements which reflect the physical behaviour of the mobile robot.

3.5 Three-Tiered Layer Control Architecture

As mentioned in the previous section, one of the aims of this chapter is to provide a control framework particularly for independent track control vehicle. In general, this thesis is focused on the implementation of *three-tiered layer control architecture* as briefly explained earlier in Chapter 2 (Figure 2.15). The obvious advantage of this control architecture is the organisation of control scheme based on the level of taxonomy.

The proposed control architecture as shown in Figure 3.1 is arranged according to the hierarchy, where the top layer (*highest control layer*) determines the *global mission objective* of the mobile robot system. Meanwhile, the lower control layers define the mobile robot behaviour (*intermediate control layer*) and desired actuators motion (*lowest control layer*). The first two

control layers, *highest control layer* and *intermediate control layer*, are responsible for defining the higher degree control algorithm which is more generic compared to the *lowest control layer*. Meanwhile, the *lowest control layer* is proposed to adapt to the specific mobile robot system. In addition, the subsumption-like control scheme, as proposed in Prescott et al. (1999) and Ridao et al. (2000), is embedded in each layer in order to arrange the control block diagrams systematically for ease of programming/simulations. Before we further describe this control architecture, it is convenient to define the overall control strategy based on Sholes (2007).

According to Sholes (2007), the autonomy level of a specific mobile robot system (UAV, UGV, USV or UUV) can be defined based on the level of taxonomy. Figure 3.2 describes the general approach for all mobile robot systems adapted from Huang et al. (2005) and Sholes (2007). Nonetheless, the autonomy level for a mobile robot system could evolve according to the control scheme. In other words, a remotely operated mobile robot system may have a certain level of autonomy that will allow it to react autonomously based on sensor(s) input. For example, a remotely operated mobile robot assigned for a reconnaissance task may reroute its predefined path if the onboard sensor detects any danger (or obstacle) on its path (e.g. fire, extreme heat, extreme high incline, etc).

The capability for the mobile robot to reroute its predefined path due to the danger (or obstacle) in this example makes the autonomy level increase from level 1 to level 6. In this section, we propose the anatomy of the *three-tiered layer control architecture* with respect to the two elements:

1. dependencies between each control layers; and

2. evolution of the autonomy level within the *three-tiered layer control architecture* proposed in this thesis.

In addition, it is convenient to explain the proposed control architecture based on the simple example shown in Figure 3.3. In short, the example addresses a navigation problem whereby the mobile robots are required to travel along an unknown and unstructured terrain. In this example, the mobile robot starts at a designated start position and must finish at allocated final position.

3.5.1 Highest Control Layer (Mission Control Layer)

In Figure 3.1, it is clear that the mobile robot control architecture is defined as a top-down approach. Having said that, a mobile robot which is controlled by this control architecture is dependent on mission specification defined from the *highest control layer* in order to execute *lowest control layer*. From Figure 3.1, we can assume that the mission specification is the start

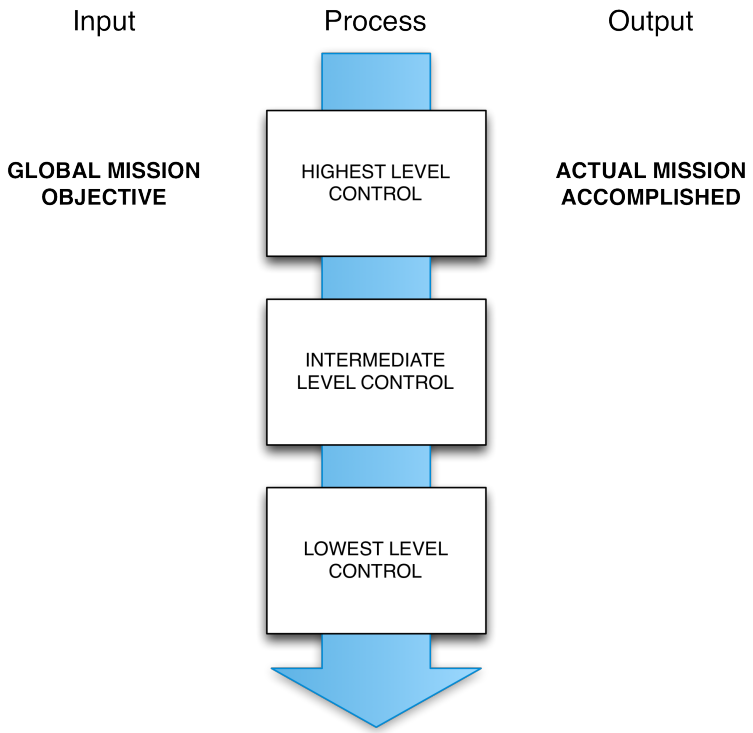


Figure 3.1: General structure of *three-tiered layer control architecture*. The arrow indicates that the control architecture is a top-down approach.

and final position of the mobile robot. Also called as *mission control layer*, the mission specified in this control layer is monitored by a *mission monitor* to ensure the mobile robot system achieves the *global mission objective*.

Consequently, the *mission monitor* is highlighted as the key feature of this control layer. The function of the *mission monitor* is to compare the actual mission with the desired mission. On top of that, it is very important to define the control scheme embedded inside the *mission control layer* which consists of several blocks as shown in Figure 3.4. As mentioned earlier, the input to this control layer is the *global mission objective* whilst the output is the *actual mission accomplished*. In precise, this particular control layer addressed the first research question outlined in Section 1.2.2 (Chapter 1).

Consider the example illustrated in Figure 3.3, let the *global mission objective* be the mobile robot travelling between the predefined start and final position. In this particular control layer, the important mission specification is the start and final position. The terrain structure and obstacle feature are not modeled in this layer. However, the *surface generator* block, which is intermediary operation between highest and intermediary control layer, is employed to model the terrain surface. Meanwhile, the *path/trajectory planner* (which is also intermediary operation between highest and intermediary control layer) generates a two-dimensional path without considering the surface and obstacle structure. It means, the mobile robot is always assumed to

more autonomy

Level	Level Description	Observation	Decision Making	Capability	Example
10	Fully autonomous	Data fusion from all sources, onboard and offboard	Totally independence to plan and reroute in order to achieve the mission objective(s)	Execute complex mission with no operator intervention	Fully autonomous mobile robot with minimal guidance from operator
9	Collaborative operations	Perception from two or more mobile robot systems	Tactical collaborative reasoning	Execute complex mission with some operator oversight	More on tactical to accomplish mission objective(s)
8	Cooperative operations	Sensor fusion from external, such as another mobile robot system	Advance decision making according to information from another systems	Rapid motion execution	Decision making based on shared data from another similar mobile robot
7	Vehicle movement detection and tracking	Actual movement of mobile robot based on the preplanned path	Robust planning and negotiation with complex terrain	Terrain negotiation	Full mobility mobile robot with limited operator intervention
6	Terrain analysis - complex obstacle detection	Model representation of the environment	Automatic reroute path	Terrain negotiation	Semi-autonomous with limited operator intervention
5	Simple obstacle detection	Data from onboard onboard obstacle detection sensor(s)	Manually reroute the path	Semiautonomous navigation	Semi-autonomous with operator intervention at any time
4	Onboard processing of sensory images	Simple obstacle or terrain surface and shape	Negotiate with terrain	Semiautonomous navigation	Follow predefined static landmarks along the path
3	Preplanned mission	Waypoint (preplanned path) to cover most of the area	Compute desired actuators motion	Basic path following under operator control	Two-dimensional preplanned path such as waypoint selection
2	Remote control with vehicle posture knowledge	Local and global configuration of mobile robot are display for operator	Mobile robot configuration (position and orientation)	Remote operation in relatively complex and known environment	Basic teleoperation where the operator have knowledge about the environment.
1	Full remotely operated	Surrounding images is view by camera mounted on the mobile robot	None	Remote operation in non-complex and known environment	Basic teleoperation, similar to demote control toys

less autonomy

Figure 3.2: The level of autonomy scale, adapted from Huang et al. (2005); Sholes (2007). Note that the 'obstacle' terminology in level 4 to level 6 could be physical object such as rubble or danger (heat, fire, explosive gas, etc).

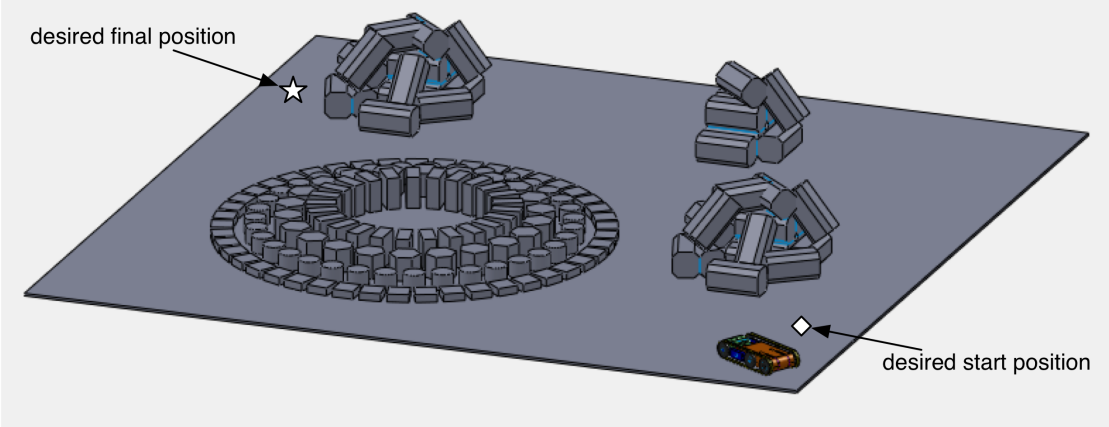


Figure 3.3: An example of a mobile robot required to travel on unknown terrain, given the start and final position.

be travelling on a planar environment according to the information given in this layer.

In addition, the dashed block labeled as *intervention* represents the external input, e.g. operator overwrite, where it can exploit the current mission objective as requires. Note that the connection from the *intervention* block to the control scheme (in this layer) is through a selector which means this layer only executes either predefined mission objective or overwrite mission instruction from external input. The selector function can be illustrated by extending example in Figures 3.5 and 3.6 in the actual USAR environment.

In Figure 3.5, the mobile robot detects the presence of a survivor which is slightly off the predefined travel path. Consequently, the selector allows the prior mission to be overwritten by activating the external input line whilst deactivating the *global mission objective* connection. The system is required to return to the prior mission mode (via selector) as soon as external mission completed. In contrast, Figure 3.6 illustrates the 'intervention' block overwrite the prior *global mission objective* to a new *global mission objective* where this particular block redefine the desired final position of the mobile robot.

In this layer, it is difficult to define the autonomy level, however, if we assume the mobile robot is initially set as level 1 (full remote control), the mission planner block increases its autonomy to level 3 whilst the selector (from 'intervention' block) may allow the autonomy level to evolve to level 5 and 6.

3.5.2 Intermediate Control Layer (Position Control Layer)

In the previous control layer, we have seen that the *global mission objective* is defined as the global input to the control architecture. Furthermore, the example given in the previous section (Section 3.5.1) illustrates the function of *intervention* block in case of any demand to redefine the mission. In order to fulfill the desired mission objective, the *intermediate control layer* is

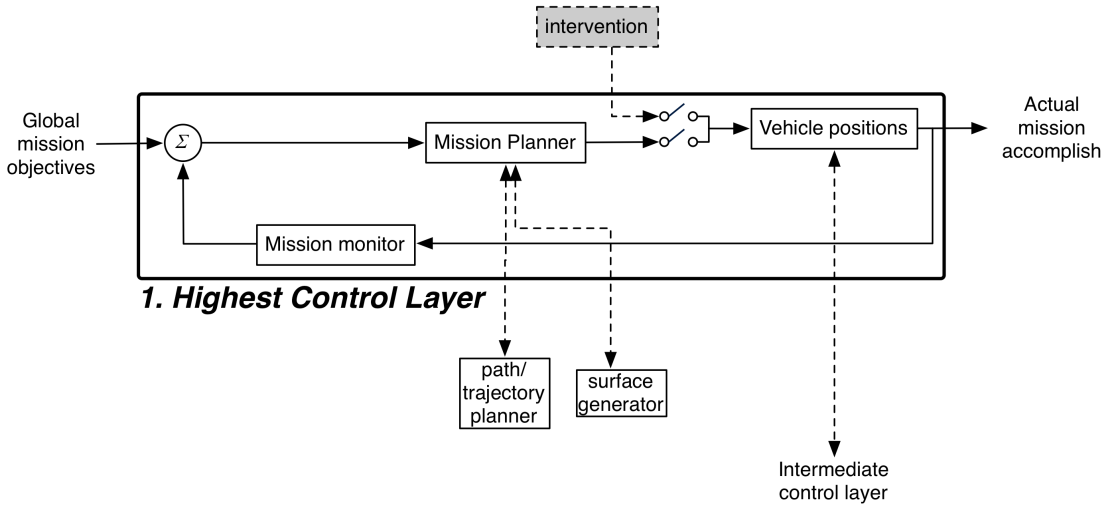


Figure 3.4: Mission control block diagram.

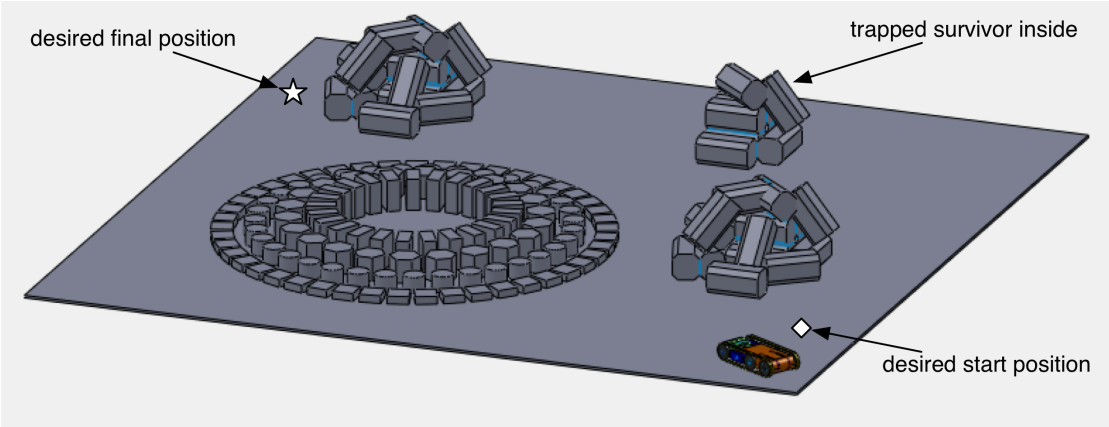


Figure 3.5: The mobile robot detect a survivor trapped inside that rubble and need to approach the survivor before moving towards the final position.

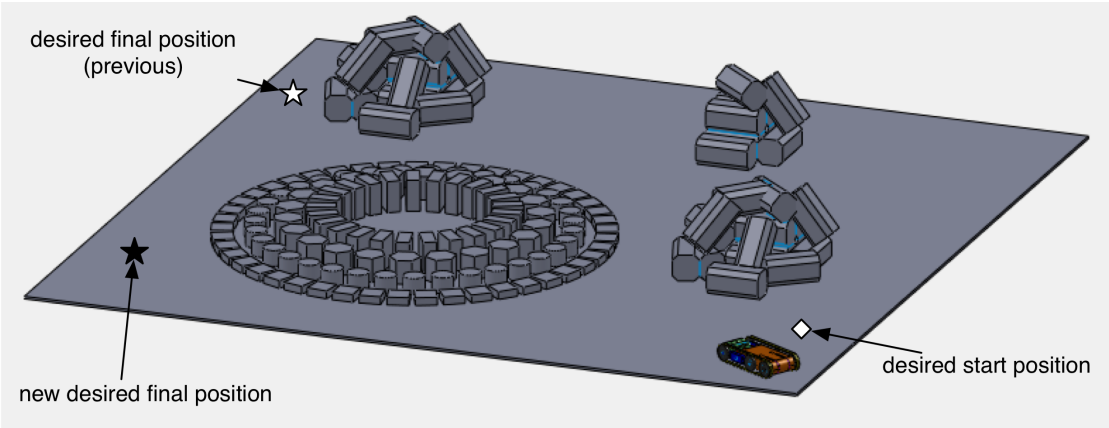


Figure 3.6: Change of the *global mission objective*.

required to derive *global mission objective* in terms of mobile robot behaviour based on the world environment. In other words, the *intermediate control layer* is the central operating layer which determines mobile robot behaviour with respect to the actual environment. In this particular control layer, the mobile robot behaviour is derived according to the desired position of the mobile robot on the terrain whereby its motion is represented by a set of velocity vectors. It is also convenient to assume that the surface structure of the world environment is modeled (and computed) within this architecture by employing a surface generator block.

The key features highlighted in this control layer are:

- the generation of the desired path/trajectory;
- the modeling of terrain/obstacle representation;
- the derivation of the desired position on the modeled surface along the predefined desired path;
- the definition of the behaviour of the mobile robot system on terrain; and
- the control scheme to ensure that the mobile robot is travelling as desired.

These key features address the second and third research questions outlined in Section 1.2.2 (Chapter 1). In order to arrange the key features and integrate inside the *position control layer*, it is useful to transform these key features into a control block diagram. Figure 3.7 shows the control block diagram embedded in the *position control layer*. As a sequence from the *mission control layer*, the input for this layer is connected from *mission planner* block. As mentioned previously, even though the *mission planner* block is shown as located on *mission control layer*, the actual operation of *mission planner* is located at intermediary position between *mission control layer* and *position control layer*.

The *path/trajectory planner* (which is a part of the *mission planner* operation) is utilised to generate a two-dimensional preplanned path. It is critically important to define the desired path without any knowledge of the actual terrain. However, the preplanned path must cover specified area of the world environment. Therefore, it is assumed that a two-dimensional path is sufficient to simplify the complexity of the architecture prior to further operation to model the terrain. In the real-world, the path generator operation can be done by various methods such as map-based pin point location or even more complex onboard preload path (Elfes, 1987; Fujimura and Samet, 1989; Edlinger and von Puttkamer, 1994). In this chapter, we assume that the desired two-dimensional path is generated via map-based pin point location.

Concurrent to the *path/trajectory planner*, the *surface generator* block is employed to model a three-dimensional surface that represents the actual terrain/obstacle based on multiple sensor

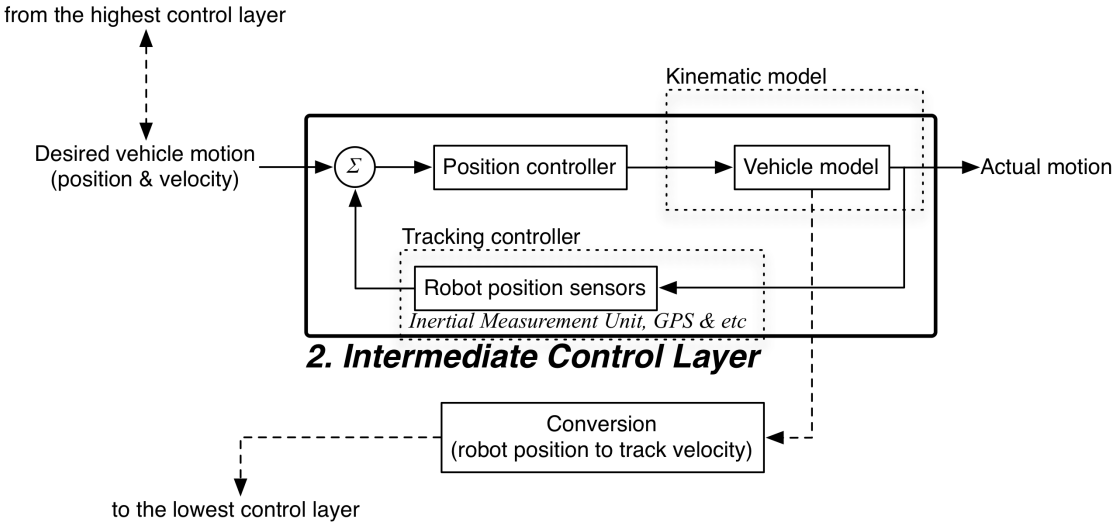


Figure 3.7: Position control block diagram.

inputs. Further to this operation, additional control scheme denoted as *kinematic model* block is applied to derive:

1. the position of the mobile robot on the terrain model along the desired path with respect to the time (third key feature); and

2. the motion behaviour of the mobile robot on the surface addressed by the fourth key feature.

Meanwhile, the responsibility of the *tracking controller* in the control block diagram is to ensure the mobile robot is travelling on the desired path as addressed on the fifth key feature.

Figure 3.9 explains the flowchart of the full operation in this control layer. Moreover, it is very convenient to extend the example illustrated by Figure 3.3 in terms of the mobile robot thought with regards to the *path/trajectory planner*, *surface generator* and *kinematic model* as shown in Figure 3.8. Consider similar problem in Figure 3.3, now, the *path/trajectory planner* generated the best possible path to cover most of the specified area. Meanwhile, the *surface generator* produces the terrain model based on multiple sensor inputs. Because this control layer highly depends on the *global mission objective* and *mission planner*, the internal control loop is utilised to track the position of the mobile robot operating until the robot reaches the desired final point, as shown in Figure 3.9. Similar to the previous section, assume the mobile robot is initially set as level 1 (full remote control), the mission planner block increases its autonomy to level 3 whilst the selector (from 'intervention' block) may allow the autonomy level to evolve to level 5 and 6.

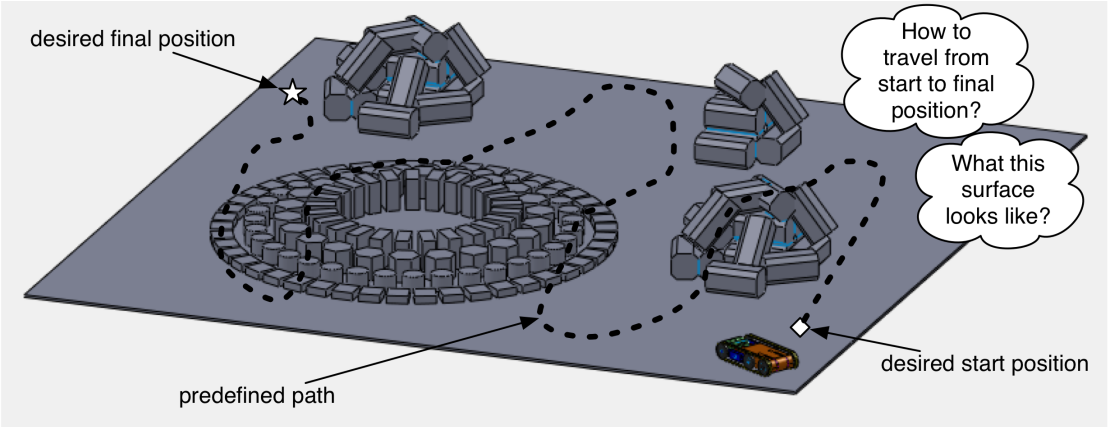


Figure 3.8: An extended version of the example given in Figure 3.3. In this example, the mobile robot system is force to think the suitable method to travel from given start and final position.

3.5.3 Lowest Control Layer (Actuator Control Layer)

We have seen that the higher degree control hierarchy is defined in the first two control layers. In short, higher degree control hierarchy, which is described in Section 3.4, are the layers that process more generic control inputs such as *global mission objective* and the desired position of mobile robot along the path. Having said that, the higher degree control hierarchy would be suitable to various types of mobile robot systems regardless of their driving mechanism. For instance, a UAV mobile robot may apply similar *global mission objective* (the desired final position) utilised in this chapter whilst its position is measured with respect to the altitude. It is the responsibility of the *lowest control layer*, which is introduced in this section, to provide the specific control scheme with respect to the hardware (actuators) configuration. In this section, the *lowest control layer* is emphasised on actuators control for the proposed mechatronic system described in Chapter 2.

In Figure 3.1, we can find that the *lowest control layer* is introduced as the bottom control layer in the *three-tiered layer control architecture*. Also called the *actuator control layer*, it is responsible to translate the *global mission objective* in terms of actuators command signal in order to achieve the desired position of mobile robot along the path. Rather than directly translate the *global mission objective* into appropriate actuators signal, the *global mission objective* is elaborated into: (1) travel pattern; (2) terrain mapping; and (3) kinematic model, as discussed in Section 3.5.2. Pursuing this further, the *actuator control layer* is responsible to synthesise commands from the *position control layer* into appropriate actuators signal which can be defined as the key control features. The key control features for the *actuator control layer* are:

- both tracks' speed;
- servo angle; and

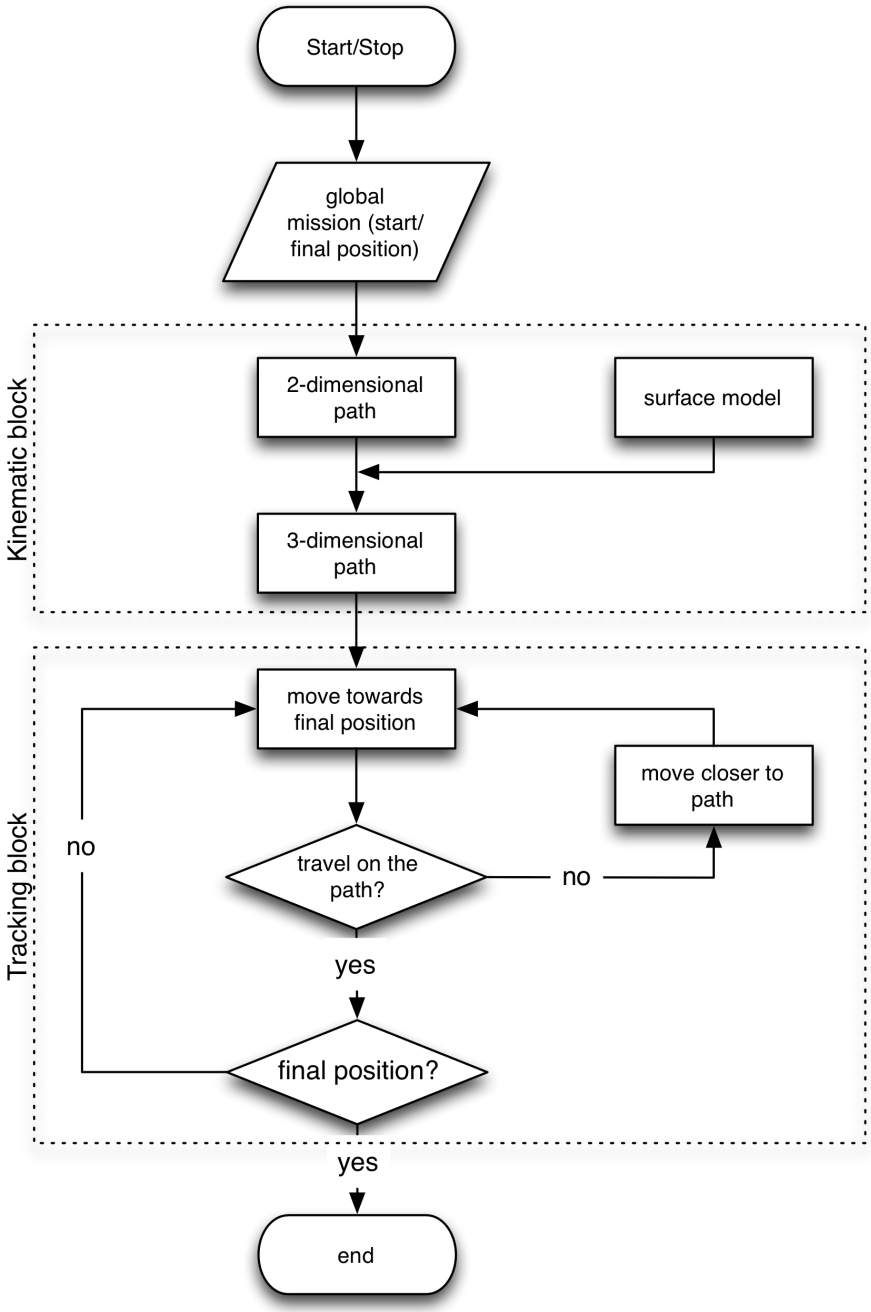


Figure 3.9: Flowchart of the *position control layer*.

- leadscrews' displacement.

Moreover, the control scheme of the *actuator control layer* is not generic because it has to be designed according to the mechatronic system.

Recall the mechatronic systems in Chapter 2, the mobile robot system consists of two tracks on both sides of its body which can vary independently via servo-leadscrew combination mechanism. In this Chapter, we will simplify and refer the term θ_{cl} as the servo-leadscrew combination mechanism. Figure 3.10 shows the control block diagram embedded in this control layer. Reinstate the example shown in Figure 3.8, this control layer will purely convert control commands from *position control layer*, particularly from the *kinematic model* block, into actuators motion.

Additionally, it is very important to highlight that one of the novel aspects in this thesis is covered in this control layer. In order to explain this novel element, it is convenient to adapt previous example shown in Figure 3.3. Redraw Figure 3.3 emphasising the novel element, Figure 3.11 illustrates the VGTV path along a side-slope. As mentioned in Chapter 2, without the independent track angle control mechanism, the VGTV will accidentally tip-over if the side-slope along the path is exceeding the operating angle that the VGTV can adapt. Therefore, the *position control layer* via its *kinematic model* computes and predicts the side-slope angle along the path. As a result, the servo-leadscrew combination mechanisms which are responsible to construct θ_{cl} will adjust according to the suitable parameters. Similar to the previous section, we assume the mobile robot is initially set as level 1 (full remote control), the *kinematic model* increase its autonomy from level 3 whilst the selector (from 'intervention' block) may allow the autonomy level to evolve to level 5 and 6.

3.6 Concluding Remarks

In this chapter, we proposed the overall control framework proposed for USAR mobile robot, particularly the VGTV proposed in Chapter 2. In summary, the proposed control framework consists of three layers control architecture where each layer is designated according to the control hierarchy. Briefly, the highest control layer defines the global mission objective whilst the lowest control layer is associated with actuator behaviour. The sequence between these layers is fulfilled by the intermediate layer control.

In the next chapter, we will formulate the mobile robot kinematic behaviour related to obstacle/terrain structure. The knowledge from this chapter, in particular the intermediary control block between highest layer control and intermediate layer control (Figure 2.20) will be extended in the form of mathematical model. On the other hand, more complex obstacle such as staircase and rubble will be converted into equivalent mathematical model. The future

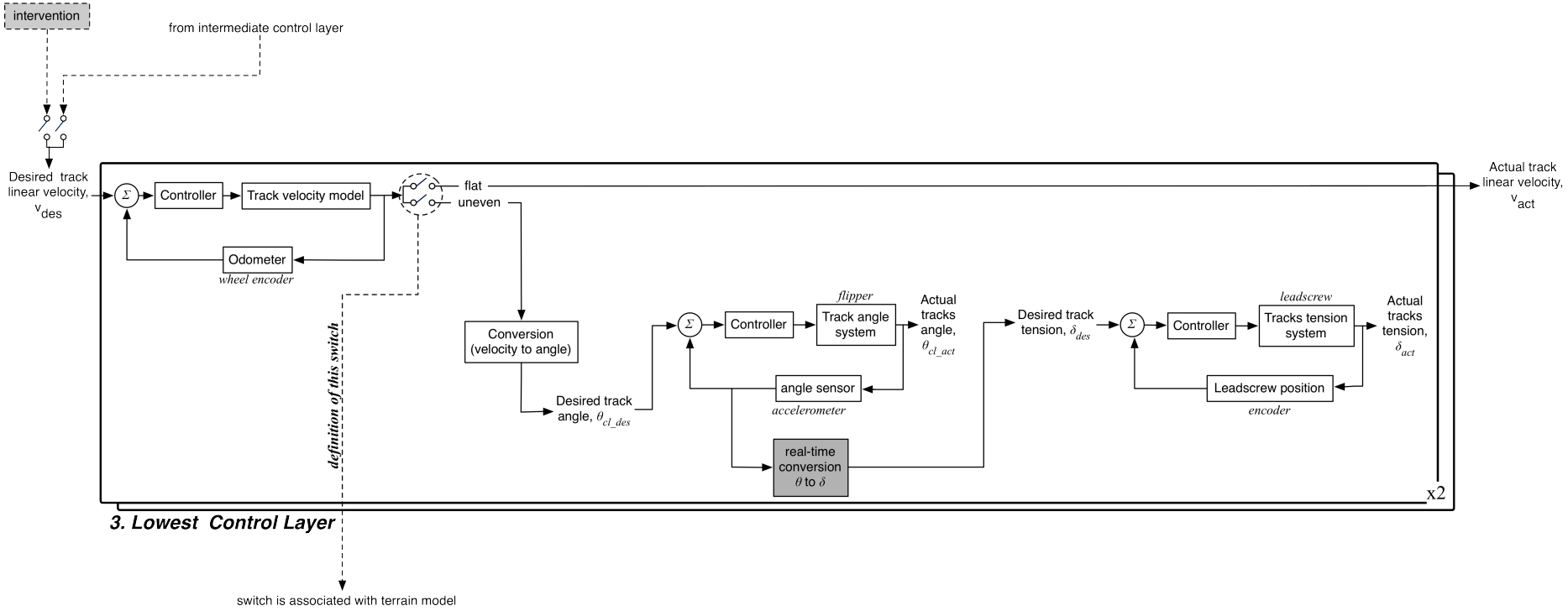


Figure 3.10: Actuator control block diagram.

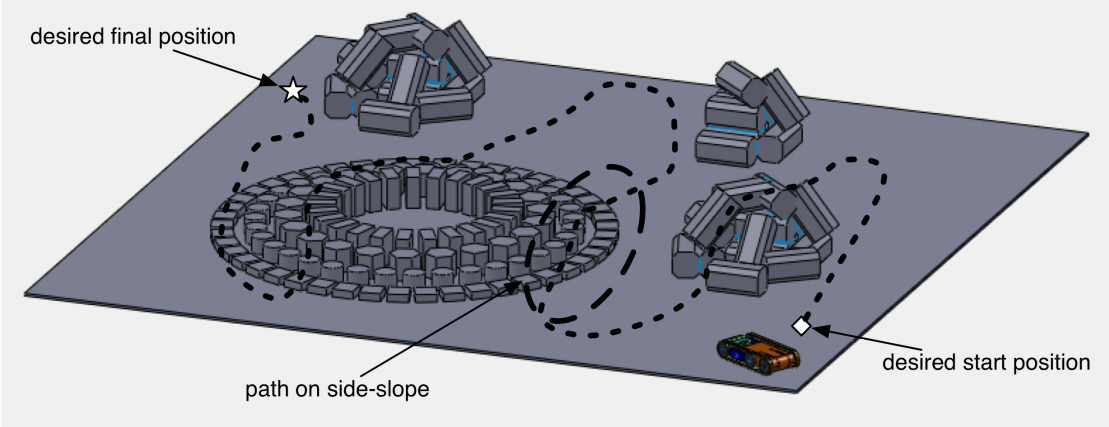


Figure 3.11: The mobile robot is expected to adjust the θ_{cl} due to side-slope.

framework will include the path planner and obstacle modeling formulation in order to investigate the mobile robot behaviour in particular its motion on uneven surface.

Chapter 4

Approximating Kinematics of Mobile Robot Motion on Uneven Surfaces

4.1 Introduction

In the previous chapter, the overall control architecture for the proposed mechatronic design has been discussed. The architecture consists of three control layers arranged according to the degree of control hierarchy. In brief, the higher degree of control hierarchy is described as the global control while the lower degree of control hierarchy applies to the local control of the mobile robot. In this chapter, we will focus on the higher degree control by highlighting the theoretical analysis of kinematics behaviour of the VGTV which travels on an uneven surface is addressed. However, in order to derive the kinematic model of the mobile robot in complex environment, such as uneven surface, the further analysis in this chapter will adapt the fixed tracked mobile robot locomotion in order to simplify the prototype proposed in Chapter 2. Consequently, the kinematics analysis of fixed configuration tracked locomotion has a similar kinematic model to a differential drive considering an assumption made by Martínez et al. (2005), if the tracks are replaced with imaginary wheels.

In the past thirty years, there has been increasing interest in studying kinematic control for a mobile robot traveling on unprepared terrain. Particular work in this research area has contributed significant progress on mobile robot exploration related applications such as search and rescue, mining exploration, inspection and many more. On top of that, many simulations and physical experiments have been reported in order to investigate the mobile robot behaviour on unprepared terrain (De Luca et al., 2001). However, most solutions for outdoor, unstructured and uneven terrain tend to converge to the complex *Simultaneous Localisation and Mapping (SLAM)* methodologies (Nuchter et al., 2007). For that reason, a simpler approach to solve the mobile robot navigation problem for uneven surface is the focus in this chapter. The emphasis of

the model derivation technique is utilising and extending the conventional kinematic modeling for a flat surface to get the approximate mobile robot behaviour (in terms of velocity vector) for uneven surface. The detailed derivation of mobile robot motion on uneven surface is described and simulated.

In general, even though the kinematic model for a specific locomotion type of mobile robot is not unique, it becomes very important to describe the desired robot behaviour when it traverses on the desired trajectory (Siciliano et al., 2009). Additionally, the kinematic model is beneficial to provide an appropriate velocity vector to drive a mobile robot to the desired configuration. Prior to this, it is very important to identify the classification of the mobile robot in order to formulate the specific kinematic model. In general, the formulation of kinematic model of any mechanical system is based on the classification of the holonomy properties, where in this thesis in particular, the tracked locomotion mobile robot is considered as a non-holonomic mobile robot. It is known by the fact that the classification of the kinematic model (holonomic or non-holonomic) is associated with rolling contact between two rigid bodies that determine the admissible motions under certain constraints (DeVon and Bretl, 2007). Consider a disk rolling in forward direction on a flat plane as depicted in Figure 4.1. In the absence of slipping (to the left or right), the disk is defined as a non-holonomic system because it keeps its direction on its sagittal plane. In contrast, a motion of a sphere on a flat plane (where the disk in Figure 4.1 is replaced with a sphere) is described as holonomic because it does not limit its motion on a single direction. In simpler words, a holonomic system is able to change its direction instantaneously whilst a non-holonomic system has its mobility reduced by at least one constraint (Siciliano et al., 2009). The comparison between holonomic and non-holonomic mechanical systems is summarised in Table 4.1.

As mentioned earlier, the tracked adjustment mechanism is assumed to be fixed at initial position and considered as a non-holonomic mobile robot where the fundamental of the non-holonomic system is explained by its constraint in Figure 4.1. Despite the kinematic behaviour associated with non-holonomic constraint on a flat surface, it is also necessary to study the kinematic behaviour of a mobile robot traversing on an uneven surface. Figure 4.2 shows three general types of surface classification with respect to the generalised global frame (fixed reference frame).

4.1.1 Motivations

Amongst the earliest theoretical work (Kitano and Jyozaki, 1976), and practical implementation (Kitano and Kuma, 1977), on kinematic model is derived from the steerability of a non-holonomic mobile robot on a flat surface with consideration of slippage, inertial force and moment of

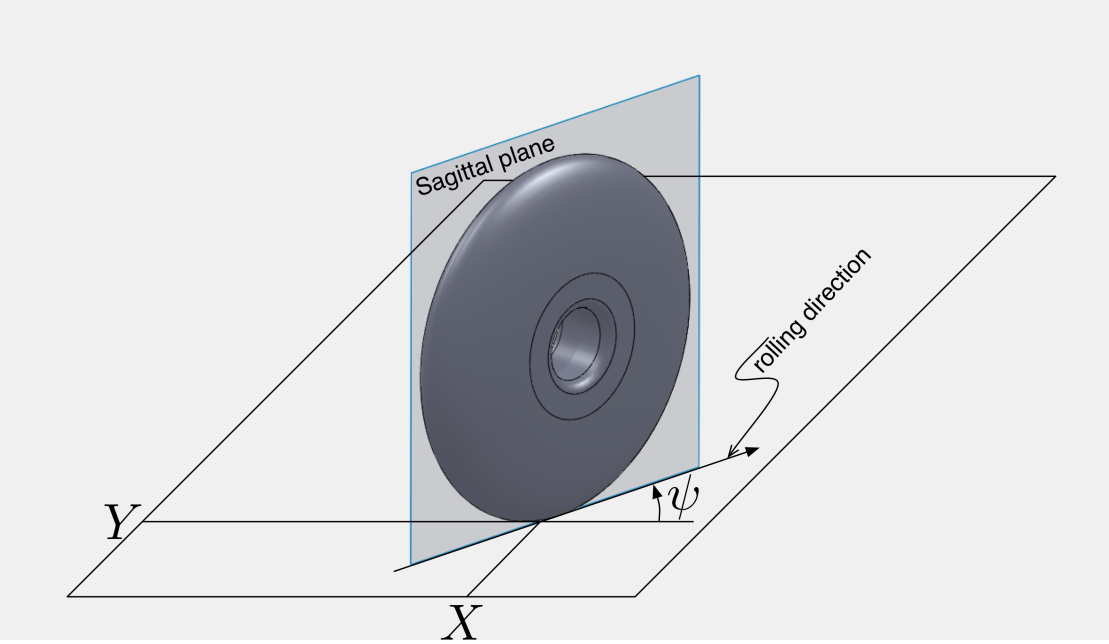
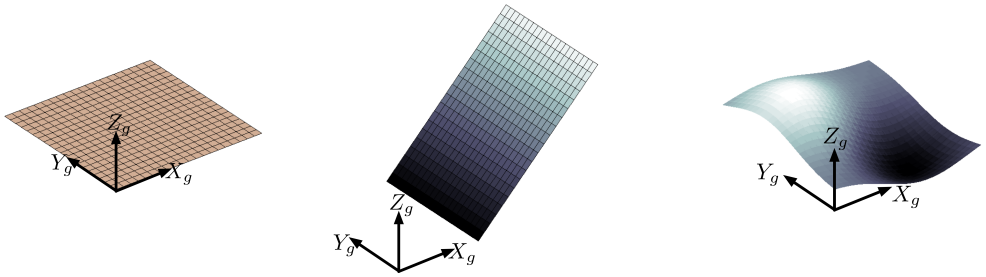


Figure 4.1: A disk rolling without slipping on a flat plane. (X, Y) is the position on the plane whilst ψ is the heading direction. The disk is rolling on its sagital plane without slipping in lateral direction. Adapted from DeVon and Bretl (2007).



(a) Flat surface (horizontal). (b) Flat surface (incline). (c) Uneven surface.

Figure 4.2: Examples of surface condition; flat, incline and uneven surface; within global frame (X_g, Y_g, Z_g).

Table 4.1: Comparison between holonomic and non-holonomic mobile robot system.

	Holonomic	Non-holonomic
Definition	Mobility of the system is not limited in any direction.	Mobility of the system is reduced by at least one constraint.
Example	Mobile robot/vehicle in service industry which utilises wheel configuration such as: • meccanum, or • omni directional can achieve instantaneous, fast and smooth turn direction motion (Holmberg and Khatib, 2000).	In parallel parking problem, the motion of the front wheel of the car is required to allow parking (Paromtchik and Laugier, 1996). Common driving configurations are: • wheeled vehicle (e.g., unicycle, car-like, differential drive), • tracked vehicle.

inertia. The relationship between generalised global frame and velocity vector leading to a more reliable model has been widely studied (Micaelli and Samson, 1993; Fierro and Lewis, 1995; Luca et al., 1998). Another solution of a related problem has also been presented by Martínez et al. (2005), employing bounded position approximation of the instantaneous centres of rotation (ICRs) on both sides of the driving mechanism. Additionally, Liu and Liu (2009a) presented a systematic approach to obtain the direct and inverse kinematic model of a mobile robot on flat and firm ground. Nonetheless, it is increasingly difficult to ignore the need to study the kinematic behaviour in complex uneven terrain.

In recent years, there has been increasing interest to develop kinematic model of non-holonomic mobile robot for complex environment. A kinematic model which has been derived from the Lagrangian dynamics equation has been presented by Chakraborty and Ghosal (2005). In extension to this paper, Auchter et al. (2009) proposed a detailed simulation of a mobile robot traversing on an uneven surface utilising passive variable camber (PVC) configuration, but both works are limited to torus wheel-shape. In this chapter, a novel kinematic model for complex and uneven terrain is presented. The modeling of the uneven terrain used in this chapter is motivated by Muniz et al. (1995). The terrain model is generated using simulation by assuming the sensory information given by the mobile robot sensor(s) is sufficent to compose a terrain model (Gaussian bumps) which is approximately equivalent to the actual terrain.

4.1.2 Aim

The aim of this chapter is to investigate the linear and angular velocities of the non-holonomic mechanical system along a pre-planned path of an uneven surface. A higher degree-of-freedom kinematic model, which is proposed in this chapter will examine the desired velocities behaviour corresponding to the desired configuration at each time interval. Nonetheless, even though a kinematic model of motion on flat surface is not unique (Siciliano et al., 2009), the proposed model is derived from the knowledge of flat surface kinematic model presented in Fahimi (2009). Meanwhile, the surface modeling process utilises a simple multi-dimensional Gaussian function due to the complexity of an uneven (rough) terrain in real world environment. In this chapter, the problem of deriving the mobile robot behaviour on an uneven surface is presented. In addition, we will present our analysis to the case of three different examples of surface model.

There are some initial assumptions defined as:

- the pre-planned path and surface model are defined offline;
- the actual terrain is highly negotiable and free from obstacles;
- the higher degree-of-freedom kinematic model is derived for a simple mobile robot mechan-

ics; and

- no slippage on driving mechanism.

The remainder of this chapter is arranged as follows; in Section 4.2, a comprehensive initial analysis of kinematic formulation for 3-dimensional environment is given, describing utilitisation of a two-dimensional formulation on a 3-dimensional environment. In addition, the mathematical derivations consisting of the generation of the pre-planned path, and modeling of an uneven terrain are explained in this section. Furthermore, Section 4.3 formulates the generalised vectors along the desired trajectory for both linear and rotational unit vector components associated with mobile robot motions, as well as the desired kinematic behaviour (linear and angular velocities) of mobile robot along the desired trajectory. Using a computer simulated program, the results are presented in Section 4.4. Section 4.5, provides some discussion related to the results of the kinematic simulation and some concluding remarks.

4.2 Kinematic Model Formulation

Consider the three layer mobile robot control architecture framework in Figure 4.3 as proposed in Chapter 2. The control architecture itself is a top-down approach in which the highest level control layer defines the global mission, the intermediate level control layer focuses on the mobile robot's configuration control and the low level control layer executes instructions from the intermediate level control layer and translate it into the actuators' motion (e.g., track speed). For example, a mobile robot is required to traverse an unknown terrain with a specific travel plan (initial and final position) defined by the global mission in the highest level control. This specific initial and final position is described in the global frame with respect to the Cartesian coordinate. Then, the trajectory planner (block component 1) develops an appropriate two-dimensional pre-planned trajectory with consideration of area coverage, however without any information of the surface condition. On the other hand, the surface generator (block component 2) analyses and develops a model of the actual terrain using an approach motivated by Muniz et al. (1995). Furthermore, the kinematic model (block component 3) synthesises the surface model and the pre-planned path to determine the desired behaviour of the mobile robot on the terrain with respect to the local frame. Analysis from the kinematic model is very crucial for further computation indicated by dotted grey arrow outside the magnifier.

The flowchart in Figure 4.4 summarises the operations involved in order to formulate the kinematic model in this chapter. This flowchart, which combines both flat surface and uneven surface terrain type, describes the step-by-step procedure on deriving the model based on the surface condition. Note that the left-half of the flowchart explains the procedure for flat surface

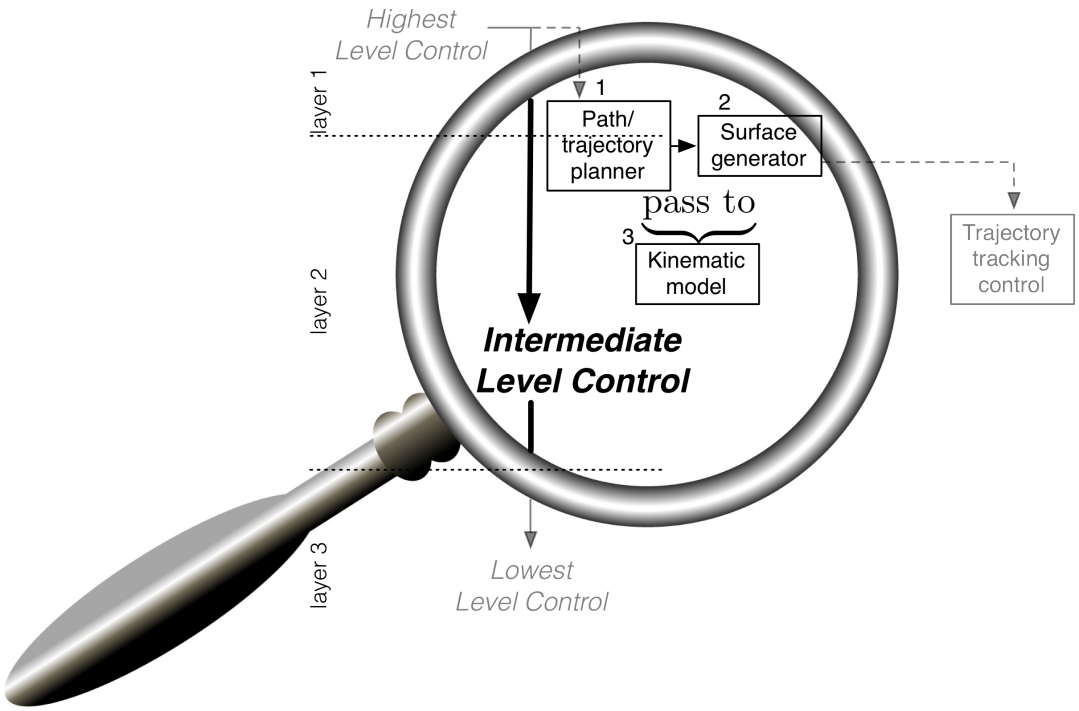


Figure 4.3: The flowchart is the mobile robot control architecture with the magnifier indicating that the intermediate level control layer is the focus in this chapter. Path/trajectory planner and surface generator are intermediary operation between Layer 1 and Layer 2. Kinematic model computes desired parameters (mobile robot configuration parameters) based on generalised velocity vector along the trajectory of surface function. Dotted solid black arrow lines with closed arrow head (- -►) denotes the topic discussed only in this chapter. Results from these operations are fed to trajectory tracking control for further operation.

modeling whereas the other-half is the focus in this chapter. The mathematical formulation utilises similar method of kinematic model of mobile robot travel on flat surface, with certain extension to solve the kinematic model for uneven surface. On top of that, the model of the uneven surface is constructed by an addition of several Gaussian functions. But, only single multivariate Gaussian function is presented for the analysis in order to simplify the derivation process. Nevertheless, results of the mobile robot kinematic behaviour in Section 4.4 are shown in a complete summation of several Gaussian function.

4.2.1 Mobile robot motion on uneven surface

Consider the definition of kinematic constraint of a non-holonomic mechanical system as discussed in Section 4.1. The mobile robot kinematic model of an uneven surface as depicted in

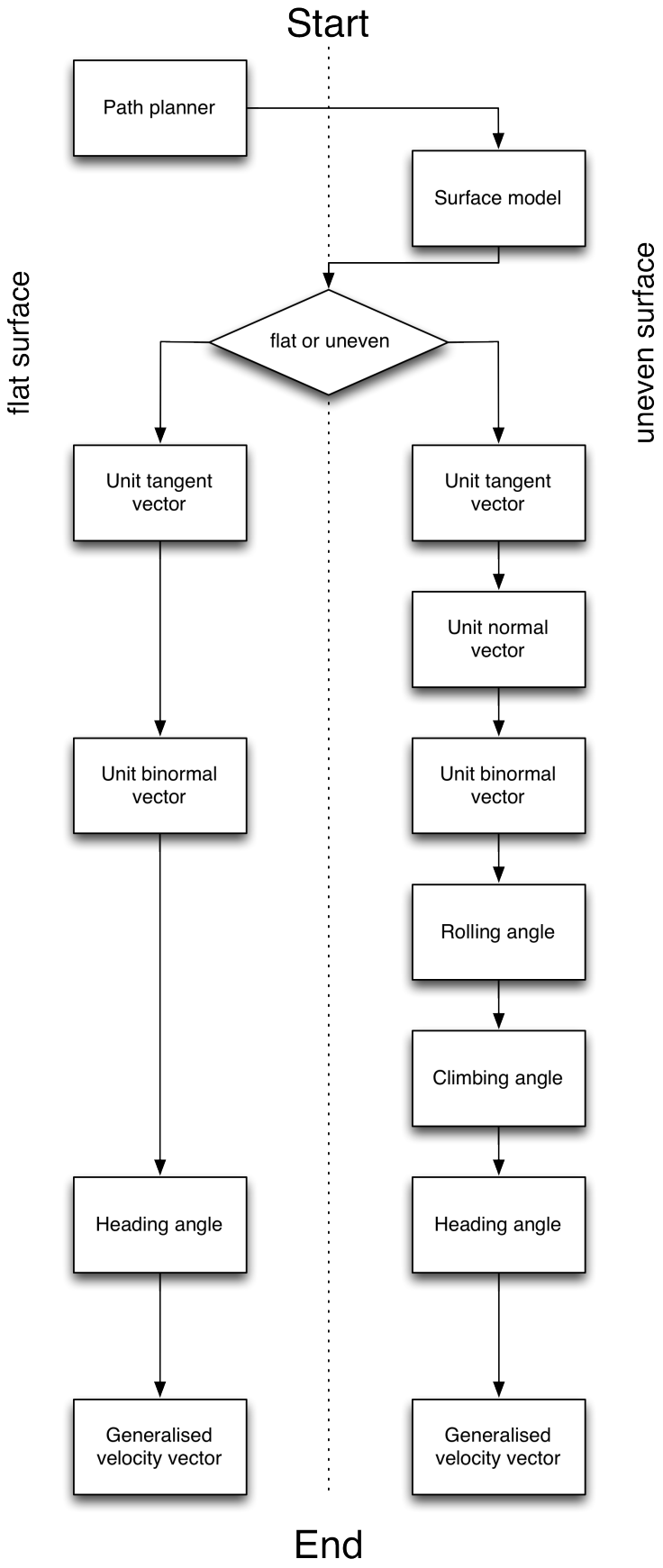


Figure 4.4: Flowchart of kinematic model derivation of a mobile robot on a surface model.

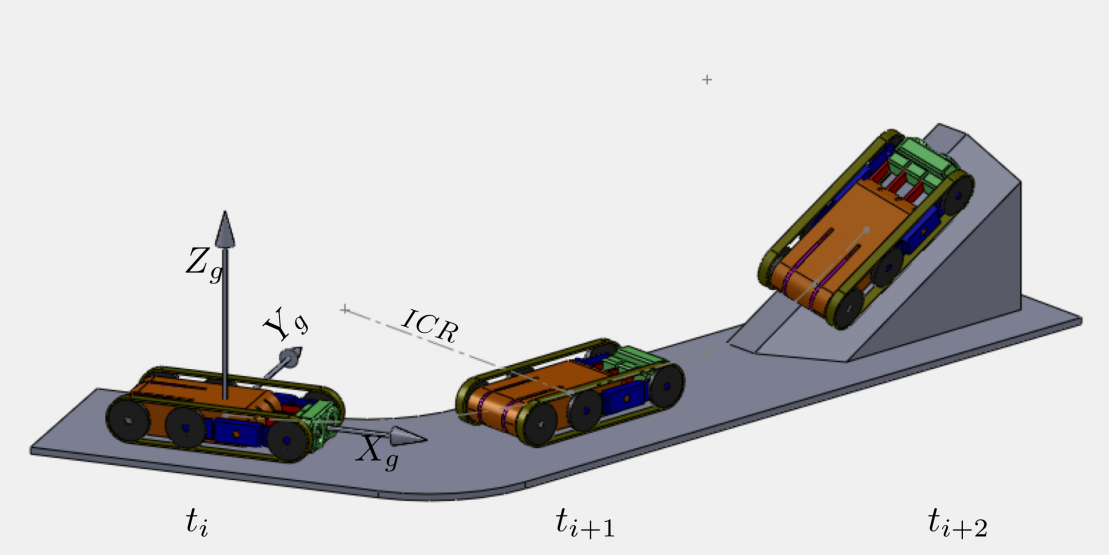


Figure 4.5: A mobile robot traveling on pre-planned path on 3-dimensional surface. ICR is acronym for *instantaneous centre of rotation*

Figure 4.5 is investigated. Observe that the configurations of the mobile robot at t_{i+2} consists of climbing and rolling motion whereby the heading angle changed at t_{i+1} relative to the initial configurations. The orientation of the local axis changes according to the mobile robot posture, with x_l-axis pointing in the driving direction of the mobile robot regardless of the rotational motion.

Note that the desired pre-planned trajectory is generated by a two-dimensional parametric path considering the initial and final positions of the mobile robot but without any information of the terrain condition. Nonetheless, it is crucial for the mobile robot to have a certain level of perception about the terrain along the trajectory. For instance, some part of the desired trajectory indicates that the mobile robot requires to negotiate with extreme inclination or side-slope. Nevertheless, in this chapter, the terrain is assumed to be highly negotiable.

4.2.2 Cartesian coordinate system

In order to formulate the kinematic model of a non-holonomic mobile robot that travels on uneven surface, it is very useful to distinguish the local frame and the global frame Cartesian coordinate system. The simplest way to explain these frames is by revisiting the formulation of the kinematic model of a non-holonomic mobile robot on flat surface (De Luca et al., 2001). With reference to Figure 4.6, the Cartesian coordinate system is divided into two frames which are:

1. The global frame, (X_g, Y_g), is the fixed frame on the surface that describes the reference configuration for the mobile robot. Even though it is not necessary, it is common that the fixed frame refers to the initial configuration of the mobile robot. In contrast, Figure

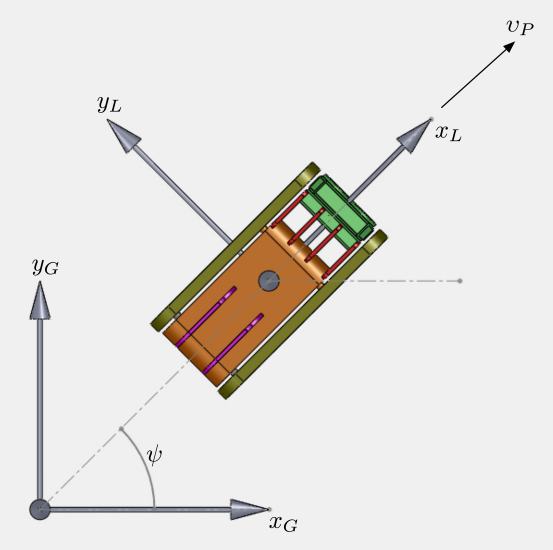


Figure 4.6: Kinematic analysis of a mobile robot on a flat surface adapted from Fahimi (2009). Subscript g and l denoted 'global' and 'local' axis repectively.

4.2b shows that the global frame is not fixed along the inclined plane to indicate the initial configuration.

2. The local frame denoted by (x_l, y_l) is the moving frame along the trajectory but fixed on the mobile robot centre of mass. On the flat surface example in Figure 4.6, the direction of the x_l-axis denotes the forward driving direction of the mobile robot with generalised velocity v_p. Meanwhile, y_l-axis indicates the mobile robot in lateral direction that is orthogonal to the forward direction.

For the rest of this thesis, the global frame is considered as the initial configuration of the mobile robot. The general equation of kinematic model for a non-holonomic mobile robot as shown in De Luca et al. (2001) gives the desired mobile robot configuration relative to the global frame and is described by

$$\dot{q} \quad = \quad G\left(q\right) u \tag{4.1}$$

where

$\dot{q}$ is the mobile robot configuration with respect to the global frame,

$G\left(q\right)$ is the kinematic matrix coefficient, and

u is the control input vector field, in this case the driving velocities.

Besides, it is appropriate to rearrange equation (4.1) in terms of the desired driving velocities and u is computed via pseudo-velocities as explained in Siciliano et al. (2009). Therefore u is

given by

$$\begin{aligned} u &= G^{\dagger}(q)\,\dot{q} \\ &= \left(G^{T}(q)\,G(q)\right)^{-1} G^{T}(q)\,\dot{q} \end{aligned} \tag{4.2}$$

With reference to Figure 4.6, the kinematics of the mobile robot platform on the flat surface are determined by position, $(\dot{x}_{2d}, \dot{y}_{2d})$, and heading angle, $\dot{\psi}_{2d}$, hence

$$\dot{x}_{2d} = \upsilon_{2d} \cdot \cos\psi_{2d} \tag{4.3}$$

$$\dot{y}_{2d} = \upsilon_{2d} \cdot \sin\psi_{2d} \tag{4.4}$$

$$\dot{\psi}_{2d} = \omega_{2d} \tag{4.5}$$

where

$(\dot{x}, \dot{y})_{2d}$ is the forward (driving) velocity vector of the mobile robot corresponding to the global frame,

$\dot{\psi}_{2d} = \omega_{2d}$ denotes the turning velocity or the rate of turning angle, and

ψ_{2d} is the heading angle composed by x_l and X_g.

On top of that, the heading angle, ψ_{2d}, is calculated as

$$\psi_{2d} = \tan^{-1}\left(\frac{y_{2d}}{x_{2d}}\right) \tag{4.6}$$

Rearranging equations (4.3), (4.4) and (4.5) into similar form of equation (4.1)

$$\underbrace{\begin{bmatrix} \dot{x}_{2d} \\ \dot{y}_{2d} \\ \dot{\psi}_{2d} \end{bmatrix}}_{\dot{q}} = \underbrace{\begin{bmatrix} \cos\psi_{2d} & 0 \\ \sin\psi_{2d} & 0 \\ 0 & 1 \end{bmatrix}}_{G(q)} \times \underbrace{\begin{bmatrix} \upsilon_{2d} \\ \omega_{2d} \end{bmatrix}}_{u} \tag{4.7}$$

Then, comparing equation (4.7) with equation (4.2) results in the pseudo-velocity of the mobile robot, hence

$$\underbrace{\begin{bmatrix} \upsilon_{2d} \\ \omega_{2d} \end{bmatrix}}_{u} = \underbrace{\left(\begin{bmatrix} \cos\psi_{2d} & 0 \\ \sin\psi_{2d} & 0 \\ 0 & 1 \end{bmatrix}^{T} \begin{bmatrix} \cos\psi_{2d} & 0 \\ \sin\psi_{2d} & 0 \\ 0 & 1 \end{bmatrix}\right)^{-1}}_{(G^{T}(q)G(q))^{-1}} \times \underbrace{\begin{bmatrix} \cos\psi_{2d} & 0 \\ \sin\psi_{2d} & 0 \\ 0 & 1 \end{bmatrix}^{T}}_{G^{T}(q)} \underbrace{\begin{bmatrix} \dot{x}_{2d} \\ \dot{y}_{2d} \\ \dot{\psi}_{2d} \end{bmatrix}}_{\dot{q}} \tag{4.8}$$

Expanding equation (4.8) with consideration of trigonometric identity, $\left(\cos^2\psi_{2d} + \sin^2\psi_{2d} = 1\right)$,

$$\begin{bmatrix} \upsilon_{2d} \\ \omega_{2d} \end{bmatrix} = \begin{bmatrix} \cos^2\psi_{2d} + \sin^2\psi_{2d} & 0 \\ 0 & 1 \end{bmatrix}^{-1} \times \begin{bmatrix} \dot{x}_{2d}\cos\psi_{2d} + \dot{y}_{2d}\sin\psi_{2d} \\ \dot{\psi}_{2d} \end{bmatrix} \tag{4.9}$$

Let $\Upsilon = \begin{bmatrix} \cos^2\psi_{2d} + \sin^2\psi_{2d} & 0 \\ 0 & 1 \end{bmatrix}$, computing the determinant then solving equation (4.9) yields

$$\begin{aligned} \begin{bmatrix} \upsilon_{2d} \\ \omega_{2d} \end{bmatrix} &= \frac{1}{\det(\Upsilon)} \begin{bmatrix} 1 & 0 \\ 0 & \cos^2\psi_{2d} + \sin^2\psi_{2d} \end{bmatrix} \times \begin{bmatrix} \dot{x}_{2d}\cos\psi_{2d} + \dot{y}_{2d}\sin\psi_{2d} \\ \dot{\psi}_{2d} \end{bmatrix} \\ &= \frac{1}{\cos^2\psi_{2d} + \sin^2\psi_{2d}} \begin{bmatrix} 1 & 0 \\ 0 & \cos^2\psi_{2d} + \sin^2\psi_{2d} \end{bmatrix} \times \begin{bmatrix} \dot{x}_{2d}\cos\psi_{2d} + \dot{y}_{2d}\sin\psi_{2d} \\ \dot{\psi}_{2d} \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{\cos^2\psi_{2d}+\sin^2\psi_{2d}} & 0 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} \dot{x}_{2d}\cos\psi_{2d} + \dot{y}_{2d}\sin\psi_{2d} \\ \dot{\psi}_{2d} \end{bmatrix} \end{aligned} \tag{4.10}$$

Applying trigonometric identity $(\cos^2\psi_{2d} + \sin^2\psi_{2d} = 1)$ in equation (4.10),

$$\begin{aligned} \begin{bmatrix} \upsilon_{2d} \\ \omega_{2d} \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} \dot{x}_{2d}\cos\psi_{2d} + \dot{y}_{2d}\sin\psi_{2d} \\ \dot{\psi}_{2d} \end{bmatrix} \\ &= \begin{bmatrix} \dot{x}_{2d}\cos\psi_{2d} + \dot{y}_{2d}\sin\psi_{2d} \\ \dot{\psi}_{2d} \end{bmatrix} \end{aligned} \tag{4.11}$$

Substituting $\cos\psi_{2d} = \frac{\dot{x}_{2d}}{\upsilon_{2d}}$ and $\sin\psi_{2d} = \frac{\dot{y}_{2d}}{\upsilon_{2d}}$ into linear velocity vector part of equation (4.11)

$$\upsilon_{2d} = \pm\sqrt{\dot{x}_{2d}^2 + \dot{y}_{2d}^2} \tag{4.12}$$

On the other hand, with reference to the differentiation rule, $\left(\frac{d}{dx}\tan^{-1}u = \frac{1}{1+u^2}\frac{du}{dx}\right)$, differentiating equation (4.6) yields

$$\dot{\psi}_g = \frac{\ddot{y}_{2d}\dot{x}_{2d} - \ddot{x}_{2d}\dot{y}_{2d}}{\dot{x}_{2d}^2 + \dot{y}_{2d}^2} \tag{4.13}$$

From equation (4.9), $\omega_{2d} = \dot{\psi}_g$. Therefore it is convenient to change the notation of equation (4.13)

$$\omega_{2d} = \frac{\ddot{y}_{2d}\dot{x}_{2d} - \ddot{x}_{2d}\dot{y}_{2d}}{\dot{x}_{2d}^2 + \dot{y}_{2d}^2} \tag{4.14}$$

The final result for this particular kinematic problem indicates that υ_{2d} and ω_{2d} are written with respect to the global frame. On the other hand, the local frame denoted by x_l and y_l in Figure 4.6 is associated with the desired trajectory.

4.2.3 Trajectory generator

The kinematic model for the mobile robot on a flat surface has been described in Section 4.2.2. The model explains the boundary of the admissible motion for the mobile robot in the global frame. Moreover, the model results in the desired generalised linear and angular velocities with respect to the global frame. In this section, the main goal is to formulate the desired trajectory of the mobile robot in order to explain the local frame convention in Cartesian coordinate system. In addition, with reference to Figure 4.3, the trajectory generator block is characterised by a two-dimensional trajectory based on the global objective defined by the mission control block in the highest control layer. Assume that the global navigation objective (defined in mission control) for the rest of this thesis is described as the initial and final configurations of the desired mobile robot motion.

The trajectory generator is then derived by this global objective into a series of desired configurations along the trajectory with expectation that the surface is: (1) a flat surface; and (2) absence from any obstacles. In this chapter, a two-dimensional spline trajectory is utilised to investigate the kinematic behaviour of the mobile robot on a surface.

Timing-law and cubic Cartesian parametric equations

Consider the desired initial, $q(t_i)$, and final configurations, $q(t_f)$, are given by

$$q(t_i) = q_i \tag{4.15}$$

$$q(t_f) = q_f \tag{4.16}$$

resulting in the potential derivation of the desired trajectory written in timing law (Siciliano et al., 2009), where the parameter of s is a function of time, therefore

$$s(t_i) = s_i \tag{4.17}$$

$$s(t_f) = s_f \tag{4.18}$$

The timing law in equations (4.17) and (4.18) will be used for the rest of this thesis.

Meanwhile, a trajectory generator utilising cubic polynomials to develop the desired trajec-

tory and given by Siciliano et al. (2009)

$$x(s) = s^3 x_f - (s-1)^3 x_i + \alpha_x s^2 (s-1) + \beta_x s (s-1)^2 \quad (4.19)$$

$$y(s) = s^3 y_f - (s-1)^3 y_i + \alpha_y s^2 (s-1) + \beta_y s (s-1)^2 \quad (4.20)$$

where

x_i is the desired initial x-position with reference to X_g in global frame,

x_f is the desired final x-position with reference to X_g in global frame,

y_i is the desired initial y-position with reference to Y_g in global frame,

y_f is the desired final y-position with reference to Y_g in global frame, and

α and β are related to the initial and final orientation (posture) of the trajectory and derived as

$$\begin{bmatrix} \alpha_x \\ \alpha_y \end{bmatrix} = \begin{bmatrix} k\cos\psi_f - 3x_f \\ k\sin\psi_f - 3y_f \end{bmatrix} \quad (4.21)$$

$$\begin{bmatrix} \beta_x \\ \beta_y \end{bmatrix} = \begin{bmatrix} k\cos\psi_i + 3x_i \\ k\sin\psi_i + 3y_i \end{bmatrix} \quad (4.22)$$

where

ψ_i is the initial orientation in radian,

ψ_f is the final orientation in radian, and

k is a "free parameter" that highly influences the generated desired trajectory (Siciliano et al., 2009).

A desired trajectory with a higher value of k will have maximum curve (bend). The sign of k (positive or negative) determines the driving direction of the mobile robot. A positive k means the mobile robot is required to start and finish with forward motion whereby a negative k means that the mobile robot is moving in backward (reverse) direction.

Generalised velocity vector along the trajectory

It is convenient to derive the first and second derivatives of equations (4.19) and (4.20) for future vector field analysis. Therefore, the first derivatives of the parametric trajectory which

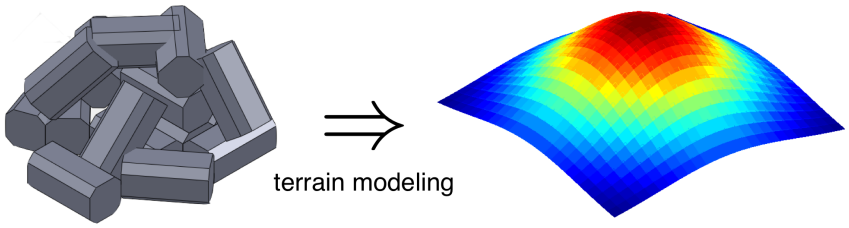


Figure 4.7: The surface generator models a physical uneven terrain (e.g, stacked-bricks) to an equivalent multivariate Gaussian surface.

corresponds to the generalised velocity vector are given by

$$\dot{x}(s) = \begin{bmatrix} -3(s-1)^2 \\ 3s^2 \end{bmatrix}^T \begin{bmatrix} x_i \\ x_f \end{bmatrix} + \begin{bmatrix} 2s(s-1)+s^2 \\ 2s(s-1)+(s-1)^2 \end{bmatrix}^T \begin{bmatrix} \alpha_x \\ \beta_x \end{bmatrix} \tag{4.23}$$

$$\dot{y}(s) = \begin{bmatrix} -3(s-1)^2 \\ 3s^2 \end{bmatrix}^T \begin{bmatrix} y_i \\ y_f \end{bmatrix} + \begin{bmatrix} 2s(s-1)+s^2 \\ 2s(s-1)+(s-1)^2 \end{bmatrix}^T \begin{bmatrix} \alpha_y \\ \beta_y \end{bmatrix} \tag{4.24}$$

and the second derivatives are given as

$$\ddot{x}(s) = 6\begin{bmatrix} -(s-1) \\ s \end{bmatrix}^T \begin{bmatrix} x_i \\ x_f \end{bmatrix} + 2\begin{bmatrix} 3s-1 \\ 3s-2 \end{bmatrix}^T \begin{bmatrix} \alpha_x \\ \beta_x \end{bmatrix} \tag{4.25}$$

$$\ddot{y}(s) = 6\begin{bmatrix} -(s-1) \\ s \end{bmatrix}^T \begin{bmatrix} y_i \\ y_f \end{bmatrix} + 2\begin{bmatrix} 3s-1 \\ 3s-2 \end{bmatrix}^T \begin{bmatrix} \alpha_y \\ \beta_y \end{bmatrix} \tag{4.26}$$

Note that the $\dot{x}(s)$ and $\dot{y}(s)$ are the tangential components of the desired trajectory. In local frame point of view, the tangential components is local x-axis, x_l, whereas y_l is always orthogonal to the x_l.

4.2.4 Multivariate Gaussian Surface

The next operation which is defined by component block 2 in Figure 4.3 is to determine the surface model of the actual terrain condition. As mentioned at the beginning of Section 4.2, the construction of the terrain model is simplified by a single Gaussian model whilst a multi-Gaussian terrain model can be developed by the summation of several Gaussian functions. Allow the surface generator to perform a surface modeling procedure of the uneven terrain as depicted

on right-hand-side in Figure 4.7. Assume that the modeling procedure utilises a multivariate Gaussian function, Muniz et al. (1995). As a result, a smooth surface model is produced which is approximately equivalent to the actual terrain.

A general formula for the n-dimension multivariate Gaussian function given by

$$f_x(x_1, ..., x_n) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} \times \exp\left\{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right\} \quad (4.27)$$

where

Σ is the variance-covariance matrix of x, and

μ is the mean of the function.

In case of bivariate, a two-dimensional Gaussian function is written as

$$f(x_1, x_2) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left\{-\frac{\left[\left(\frac{x_1-\mu_1}{\sigma_x}\right)^2 - 2\rho\left(\frac{x_1-\mu_1}{\sigma_x}\right)\left(\frac{x_2-\mu_2}{\sigma_y}\right) + \left(\frac{x_2-\mu_2}{\sigma_y}\right)^2\right]}{2(1-\rho^2)}\right\} \quad (4.28)$$

where

σ_x is the width of the Gaussian bump in X_g-axis,

σ_y is the width of the Gaussian bump in Y_g-axis,

ρ is the parameter that measured the angle orientation of the Gaussian, and

$\mu_{1,2}$ is the mean of the Gaussian function.

Consider the mobile robot employs a sensor (e.g., range finder) that is capable to determine the elevation of the terrain denoted with the parameter A. Therefore, the generalised two-dimensional Gaussian function in equation (4.28) can be rewritten corresponding to the global frame convention as

$$f(X,Y)_g = A\exp\left\{-\left(\frac{(X_g - x_o)^2}{2\sigma_x^2} + \frac{(Y_g - y_o)^2}{2\sigma_y^2}\right)\right\} \quad (4.29)$$

where

$f(X,Y)_g$ is a bivariate Gaussian function corresponding to the global frame,

X_g and Y_g are the variables along X and Y global axes,

x_o and y_o are the means and

A is the elevation of the surface.

However, in real-world application, the actual terrain is not always modeled by a single Gaussian function. Therefore, a higher complexity surface model employs a multiple Gaussian model given

by,

$$f_{mul}(X,Y)_g \quad = \quad f_1(X,Y)_g + f_2(X,Y)_g + f_3(X,Y)_g + ... + f_{\text{numG}}(X,Y)_g \qquad (4.30)$$

where

numG is the maximum number of Gaussian function, and

$f_i\left(X,Y\right)_g$ is a single Gaussian function corresponding to the global frame with $1 \leqslant i \leqslant$ numG. Nevertheless, for the ease of further derivation, the Gaussian function in equation (4.29) is utilised as the surface model. As mentioned earlier, even though single Gaussian is shown in the derivation, the generation of the surface model in computer simulation is done by the summation of several Gaussian function (equation (4.30)) with different parameters A, μ and σ.

4.2.5 Trajectory on uneven surface

In Section 4.2.3, we have defined the pre-planned trajectory for the two-dimensional frame described by equations (4.19) and (4.20). Meanwhile, Section 4.2.4 determined the surface model of the actual terrain condition. The objective in this section is to extend the formulation of the two-dimensional pre-planned trajectory for the uneven surface problem. Figure 4.8 describes the proposed method where the operations decomposed of the projection of two-dimensional pre-planned trajectory onto the surface model. Now, consider the two-dimensional pre-planned spline trajectory as depicted by the top picture in Figure 4.8 whilst the surface generator generates the surface model of the actual terrain (middle picture). The projection of the desired trajectory onto the surface model results in a third parametric equation which corresponds to the height of the surface along the desired trajectory. In order the compute this projection, one must consider to apply equations (4.19) and (4.20) into equation (4.29).

Substituting equations (4.19) and (4.20) into equation (4.29) yields the trajectory which is bounded to the Z-axis direction of the global frame, hence

$$z\left(s\right) \quad = \quad A\exp\left\{-\left(\frac{\left(x\left(s\right)-x_o\right)^2}{2\sigma_x^2}+\frac{\left(y\left(s\right)-y_o\right)^2}{2\sigma_y^2}\right)\right\} \qquad (4.31)$$

The generalised velocity vector for the desired trajectory on an uneven surface is given by the first derivative of the trajectory. Therefore, differentiating equation (4.31) yields

$$\dot{z} \quad = \quad A\left(-\frac{\dot{x}x-\dot{x}x_o}{\sigma_x^2}-\frac{\dot{y}y-\dot{y}y_o}{\sigma_y^2}\right)\times\exp\left\{-\frac{1}{2}\left(\frac{x-x_o}{\sigma_x}\right)^2-\frac{1}{2}\left(\frac{y-y_o}{\sigma_y}\right)^2\right\} \qquad (4.32)$$

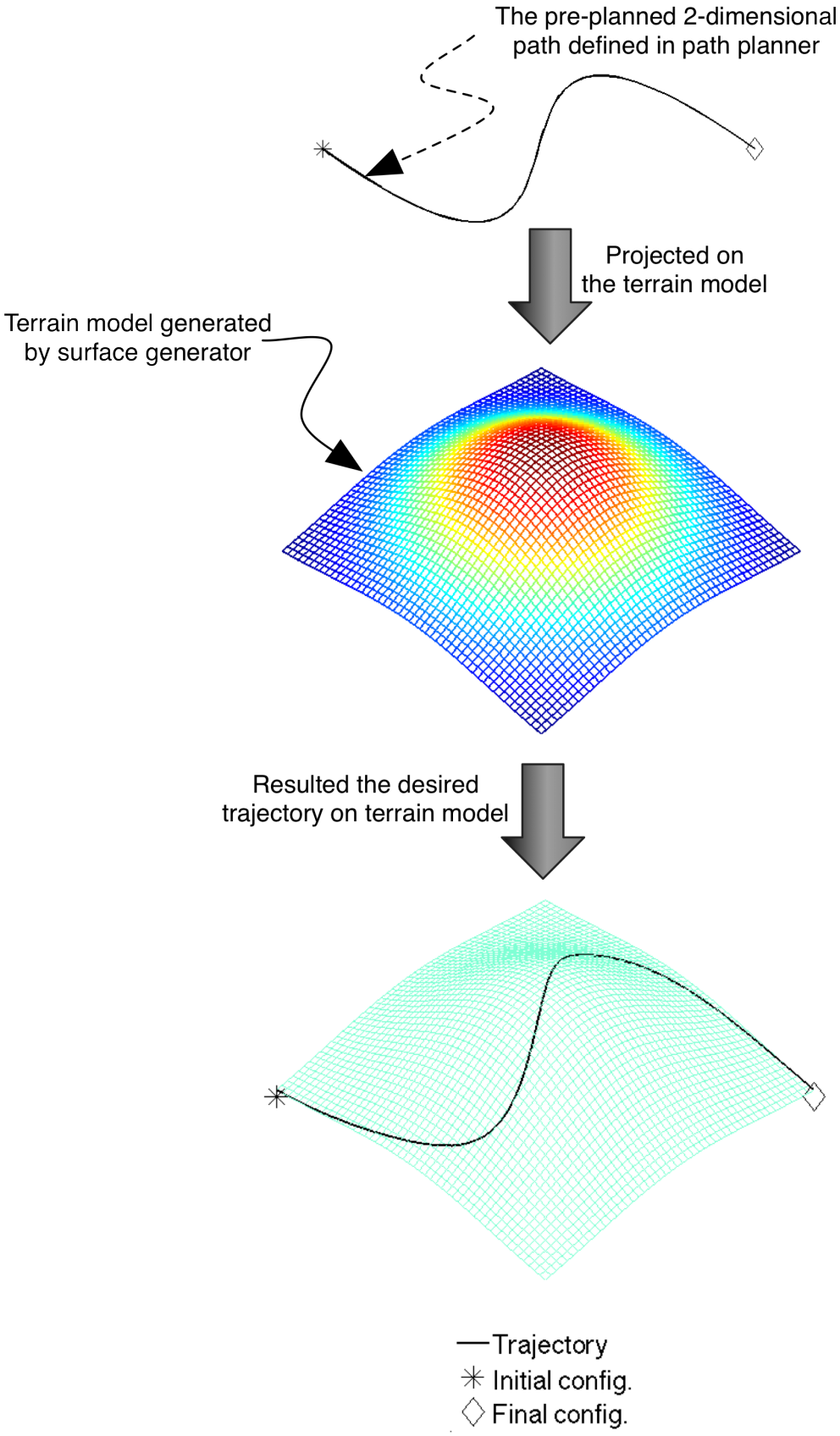


Figure 4.8: The projection of pre-planned path onto the terrain model.

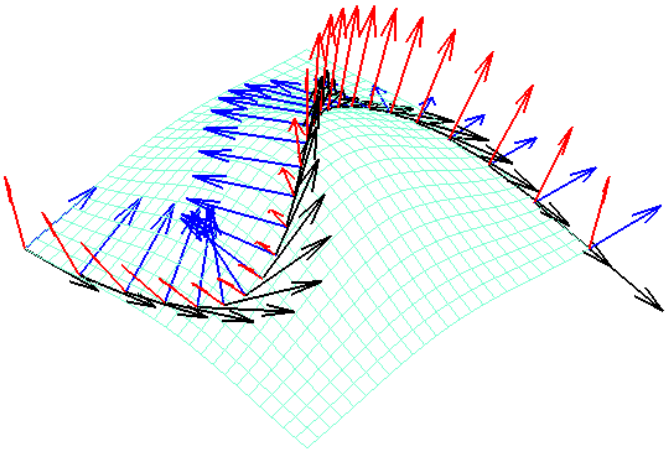


Figure 4.9: Vector fields along the desired trajectory.

and the second order differentiation of the desired trajectory is given by

$$\begin{aligned}\ddot{z} &= A\left(-\frac{\dot{x}^2+\ddot{x}x-\ddot{x}x_o}{\sigma_x^2}-\frac{\dot{y}^2+\ddot{y}y-\ddot{y}y_o}{\sigma_y^2}\right)\times\exp\left\{-\frac{1}{2}\left(\frac{x-x_o}{\sigma_x}\right)^2-\frac{1}{2}\left(\frac{y-y_o}{\sigma_y}\right)^2\right\}\\ &+ A\left(-\frac{\dot{x}x-\dot{x}x_o}{\sigma_x^2}-\frac{\dot{y}y-\dot{y}y_o}{\sigma_y^2}\right)\times\exp\left\{-\frac{1}{2}\left(\frac{x-x_o}{\sigma_x}\right)^2-\frac{1}{2}\left(\frac{y-y_o}{\sigma_y}\right)^2\right\} \qquad (4.33)\end{aligned}$$

Subsequently, the desired trajectory for an uneven surface is given by equations (4.19), (4.20) and (4.31). For simplification, $q = q(s)$, $x = x(s)$, $y = y(s)$, $z = z(s)$, $\dot{q} = \dot{q}(s)$, $\dot{x} = \dot{x}(s)$, $\dot{y} = \dot{y}(s)$, $\dot{z} = \dot{z}(s)$, $\ddot{q} = \ddot{q}(s)$, $\ddot{x} = \ddot{x}(s)$, $\ddot{y} = \ddot{y}(s)$ and $\ddot{z} = \ddot{z}(s)$ will be used for the rest of the thesis. Note that the projection of the two-dimensional trajectory onto a multiple Gaussian can be formulated by applying equations (4.19) and (4.20) into equation (4.30).

4.3 Kinematic Analysis

The proposed kinematic analysis in this chapter comprised of the derivation of the generalised velocity vector fields along the desired trajectory, as illustrated in Figure 4.9. In contrast to the global frame explained in Section 4.2.2, each vector field for a specific time-interval represents the

Table 4.2: List of assumptions for high-degree-of-freedom kinematic model derivation.

No	Assumptions
1	Pre-planned path, surface model are defined offline.
2	Actual terrain is negotiable.
3	Model derived is for a simple mobile robot with non-holonomic constraint.
4	No slippage on driving mechanism.
5	Actual terrain is modeled and generated by a multivariate Gaussian function.
6	Rolling and climbing motion is not defined in higher control layer.

local frame of the mobile robot. Prior to this, the desired two-dimensional trajectory is projected onto the surface to visualise appropriate travel plan along the surface. The assumptions made earlier are summarised in Table 4.2. The goal of this section is to formulate the generalised velocity vector from the inverse kinematic model for the mobile robot travel on uneven surface as illustrated in Figure 4.5. On top of that, the linear unit vector components (tangent, normal and binormal) and rotational angular components (roll, climb and turn) are derived in order to investigate the kinematic behaviour of the mobile robot. As mentioned in Section 4.2, the derivation operation is illustrated by the flowchart in Figure 4.4.

4.3.1 Unit vector acting along the pre-planned trajectory

In this section, the derivation of all vector fields along the pre-planned trajectory incorporated with surface condition (flat or uneven) will be normalised into unit vector fields. The unit vector fields along the pre-planned trajectory represent the local frame acting on the mobile robot where:

- the tangent vector is the forward direction of the mobile robot as well as local x_l-axis;
- the normal vector corresponds to the direction of local z_l-axis; and
- the binormal vector is perpendicular to the tangent vector as well as the normal vector according to the right-hand-rule depicted in Figure 4.10.

4.3.2 Tangent vector along the pre-planned trajectory

Consider the previous two-dimensional pre-planned spline trajectory illustrated by the top picture of Figure 4.8, the tangent vector on two-dimensional Cartesian space, $\vec{T}_{2d}$, along this trajectory is given by the first derivative of the parametric trajectory in equations (4.19) and (4.20). Considering equations (4.23) and (4.24), the tangent vector components, $\vec{T}_{2d(x)}$ and $\vec{T}_{2d(y)}$, are

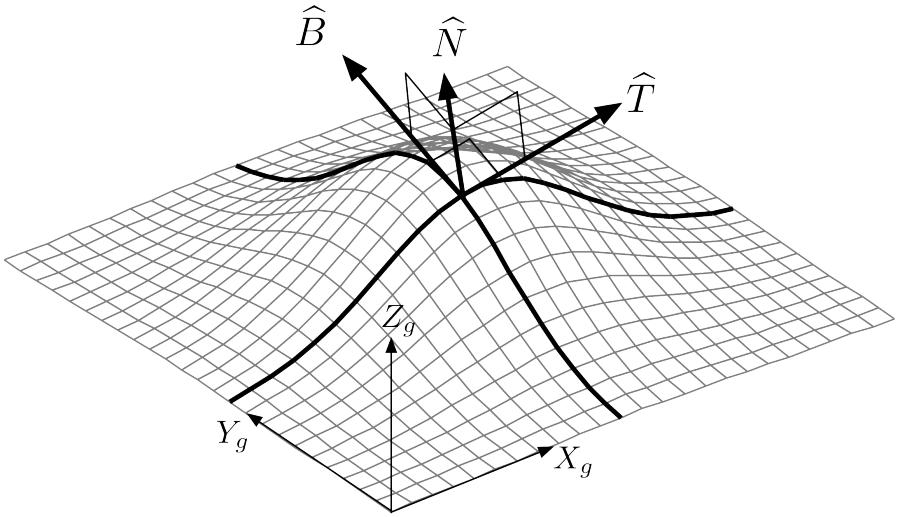


Figure 4.10: Illustration of a tangent, $\widehat{T}$, normal, $\widehat{N}$, and binormal vector, $\widehat{B}$, on uneven surface model.

given by

$$\vec{T}_{2d(x)} = \dot{x} \tag{4.34}$$

$$\vec{T}_{2d(y)} = \dot{y} \tag{4.35}$$

where $\vec{T}_{2d(x)}$ and $\vec{T}_{2d(y)}$ are associated with the tangent vector in the direction of X_g and Y_g in the global frame. The length of $\vec{T}_{2d}$ is the resultant of $\vec{T}_{2d(x)}$ and $\vec{T}_{2d(y)}$, therefore

$$\|T_{2d}\| = \sqrt{\left(\vec{T}_{2d(x)}\right)^2 + \left(\vec{T}_{2d(y)}\right)^2}$$

The tangent vector, $\vec{T}_{3d}$, along the trajectory is given by the first derivative of projected trajectory. Note that this trajectory is expressed by equations (4.19), (4.20) and (4.31). The tangent vector components can be written as $\vec{T}_{3d(x)}$, $\vec{T}_{3d(y)}$ and $\vec{T}_{3d(z)}$, where

$$\vec{T}_{3d(x)} = \vec{T}_{2d(x)} \tag{4.36}$$

$$\vec{T}_{3d(y)} = \vec{T}_{2d(y)} \tag{4.37}$$

Meanwhile, $\vec{T}_{3d(z)}$ is computed by the first derivative of the parametric trajectory projected on the multivariate Gaussian surface which is expressed by

$$\vec{T}_{3d(z)} = \dot{z} \tag{4.38}$$

Therefore, the length of $\vec{T}_{3d}$ is the resultant of $\vec{T}_{3d(x)}$, $\vec{T}_{3d(y)}$ and $\vec{T}_{3d(z)}$, yielding

$$\|T_{3d}\| = \sqrt{\left(\vec{T}_{3d(x)}\right)^2 + \left(\vec{T}_{3d(y)}\right)^2 + \left(\vec{T}_{3d(z)}\right)^2} \tag{4.39}$$

Transforming equations (4.34) to (4.38) in terms of unit vector by dividing every vector component with the corresponding vector length, yields

$$\widehat{T}_{2d(x)} = \frac{\vec{T}_{2d(x)}}{\|T_{2d}\|} \tag{4.40}$$

$$\widehat{T}_{2d(y)} = \frac{\vec{T}_{2d(y)}}{\|T_{2d}\|} \tag{4.41}$$

$$\widehat{T}_{3d(x)} = \frac{\vec{T}_{3d(x)}}{\|T_{3d}\|} \tag{4.42}$$

$$\widehat{T}_{3d(y)} = \frac{\vec{T}_{3d(y)}}{\|T_{3d}\|} \tag{4.43}$$

$$\widehat{T}_{3d(z)} = \frac{\vec{T}_{3d(z)}}{\|T_{3d}\|} \tag{4.44}$$

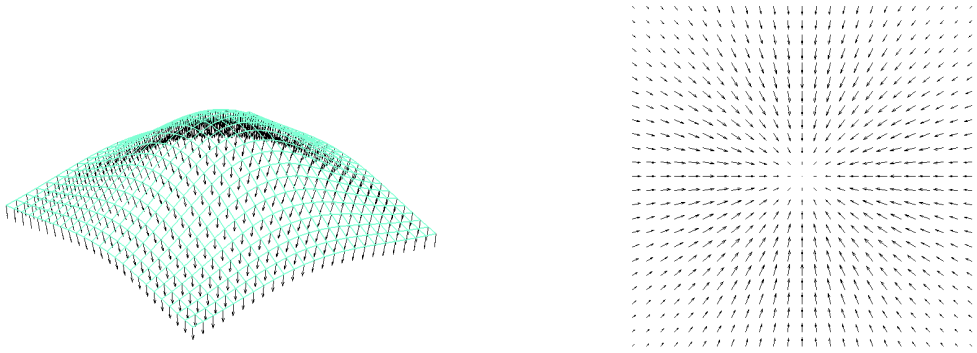
As mentioned in Section 4.2.2, the local frame axis that corresponds to x_l is given by this tangent vector where the unit tangent vectors are given by equations (4.42) to (4.44). The direction of x_l denotes the forward driving direction of the mobile robot along the trajectory. The next step is to derive the unit normal vector which represent the z_l-axis.

4.3.3 Normal unit vector along the pre-planned trajectory

In order to construct the unit normal vector along the trajectory, it is convenient to visualise all normal vectors acting on the surface by employing gradient of the surface, ∇f. By definition, the gradient of a surface, ∇f, is a gradient vector that is orthogonal to the level curve (normal plane) as illustrated in Figure (4.11). On the other hand, directional derivative, $D_u f$, gives the rate of change at any arbitrary point at fixed direction (Kreyszig, 2010). Consider the tangent plane equation (4.45) at any arbitrary point, (x_o, y_o, z_o), on the multivariate Gaussian surface

$$a_{tp}\,(x - x_o) + b_{tp}\,(y - y_o) + c_{tp}\,(z - z_o) = 0 \tag{4.45}$$

where



(a) 3-dimensional view.

(b) Top view.

Figure 4.11: Gradient of the surface is the gradient vector orthogonal to the level curve. The 3-dimensional view depicted in Figure 4.11a shows the arrows are orthogonal (perpendicular) to the level curve at any point on the surface. Meanwhile, Figure 4.11b shows the gradient vector field form the top view.

$\{a, b, c\}_{tp}$ are coefficients of the tangent plane.

The gradient vector of the surface, $\nabla f = \langle f_x, f_y, f_z \rangle$, always produces a normal vector which indicates a line that is perpendicular to the tangent plane given by

$$f_x(x_o, y_o, z_o)(x - x_o) + f_y(x_o, y_o, z_o)(y - y_o) + f_z(x_o, y_o, z_o)(z - z_o) = 0 \quad (4.46)$$

where

$f_x(x_o, y_o, z_o)$ is the partial derivative of equation (4.29) with respect to x, $\frac{\partial f}{\partial x}$,

$f_y(x_o, y_o, z_o)$ is the partial derivative of equation (4.29) with respect to y, $\frac{\partial f}{\partial y}$, and

$f_z(x_o, y_o, z_o)$ is the partial derivative of equation (4.29) with respect to z, $\frac{\partial f}{\partial z}$.

Comparing equations (4.45) and (4.46), we can obtain the coefficients of a tangent plane as

$$\begin{aligned} a_{tp} &= f_x(x_o, y_o, z_o) \\ b_{tp} &= f_y(x_o, y_o, z_o) \\ c_{tp} &= f_z(x_o, y_o, z_o) \end{aligned}$$

Note that $(f)_{xyz}$ is the partial derivative of multivariate Gaussian surface. Meanwhile, $(a, b, c)_{tp}$ is the normal vector, $\vec{N}$, and can be written as

$$\vec{N} = \langle a_{tp}, b_{tp}, c_{tp} \rangle \quad (4.47)$$

where

a_{tp} is $\vec{N}$ in x-direction, $\vec{N}_x$,

b_{tp} is $\vec{N}$ in z-direction, $\vec{N}_y$,

c_{tp} is $\vec{N}$ in y-direction, $\vec{N}_z$.

As mentioned in equation (4.46), $(a, b, c)_{tp}$ is the vector of partial derivatives of the multivariate Gaussian surface, therefore

$$\begin{pmatrix} a_{tp} \\ b_{tp} \\ c_{tp} \end{pmatrix} = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{pmatrix}$$

Inspecting the elements of the multivariate Gaussian function (equation (4.29)) used in this study, the uneven surface is given by the function $f(x, y)$. Therefore, the partial derivatives of equation (4.47) can be rewritten as (Kreyszig, 2010)

$$\vec{N}_x = \frac{\partial f(x, y)}{\partial x} \tag{4.48}$$

$$\vec{N}_y = \frac{\partial f(x, y)}{\partial y} \tag{4.49}$$

$$\vec{N}_z = -1 \tag{4.50}$$

where the partial derivatives in equations (4.48), (4.49) and (4.50) with respect to surface level are given by

$$\vec{N}_x = -\frac{A(x - x_o)}{\sigma_x^2} \times \exp\left\{-\left(\frac{(x - x_o)^2}{2\sigma_x^2} + \frac{(y - y_o)^2}{2\sigma_y^2}\right)\right\} \tag{4.51}$$

$$\vec{N}_y = -\frac{A(y - y_o)}{\sigma_x^2} \times \exp\left\{-\left(\frac{(x - x_o)^2}{2\sigma_x^2} + \frac{(y - y_o)^2}{2\sigma_y^2}\right)\right\} \tag{4.52}$$

$$\vec{N}_z = -1 \tag{4.53}$$

It is also convenient to write $\vec{N}$ in terms of the unit vector by dividing equations (4.51), (4.52) and (4.53) by the magnitude length of $\vec{N}$,

$$\widehat{N}_x = \frac{\vec{N}_x}{\|N\|} \tag{4.54}$$

$$\widehat{N}_y = \frac{\vec{N}_y}{\|N\|} \tag{4.55}$$

$$\widehat{N}_z = \frac{\vec{N}_z}{\|N\|} \tag{4.56}$$

where $\|N\|$ is magnitude length of the vector and given by

$$\|N\| = \sqrt{\vec{N}_x^2 + \vec{N}_y^2 + \vec{N}_z^2} \tag{4.57}$$

Note that the derivation of the normal vector in this section is purely based on 3-dimensional Cartesian space whereas in two-dimensional Cartesian space, the normal vector field is pointing upward which is in the same direction as the global Z_g-axis. This is true because the tangent plane created on the flat surface is the $(XY)_g$-plane itself. The next step is to derive the vector field that is binormal to the tangent and normal vector.

4.3.4 Binormal unit vector along the pre-planned trajectory

Earlier in Section 4.3.2 and 4.3.3, two of the local axes component which are x_l (tangent vectors along the trajectory) and z_l (normal vector along the trajectory) have been described. Moreover, another vector field is required to complete the local axes system which is y_l. Using *right-hand-rule* convention, y_l is given by the vector which is binormal to the tangent, $\vec{T}$, and the normal vector, $\vec{N}$. Rotating $\vec{T}$ 90 deg around $\vec{N}$ results in a vector field that is binormal to $\vec{T}$ and $\vec{N}$. Besides solving this procedure using Euler angle method, the unit quaternions rotation method is beneficial to find an arbitrary rotation of a vector field around another vector field regardless of the angle between those vectors. The quaternion rotation is written as

$$P_{\text{rotated}} = Q \cdot P \cdot Q^* \tag{4.58}$$

where

P is the unit vector field before rotation,

$P_{rotated}$ is the unit vector field as a result from quaternions rotation,

Q is the unit quaternions of the rotation axis,

Q^* is unit quaternions conjugate of the rotation axis.

The Q and Q^* are expressed as

$$Q = \cos\left(\frac{\theta_q}{2}\right) + (a_x i + a_y j + a_z k)\sin\left(\frac{\theta_q}{2}\right) \tag{4.59}$$

$$Q^* = \cos\left(\frac{\theta_q}{2}\right) - (a_x i + a_y j + a_z k)\sin\left(\frac{\theta_q}{2}\right) \tag{4.60}$$

where

θ_q is rotation transformation in radians,

a_x, a_y and a_z are rotation axes in terms of unit vectors in the local frame.

Let $\widehat{T}_{3d(xyz)}$ in equations (4.42) to (4.44) represent P, whereas $\widehat{N}_{3d(xyz)}$ in equations (4.54) to (4.56) represents the rotation axis. The $\widehat{T}_{3d(xyz)}$ and $\widehat{N}_{3d(xyz)}$ in unit vector form can be written

as

$$\widehat{T}_{3d(xyz)} = i\,\widehat{T}_{3d(x)} + j\,\widehat{T}_{3d(y)} + k\,\widehat{T}_{3d(z)} \quad (4.61)$$

$$\widehat{N}_{3d(xyz)} = i\,\widehat{N}_{3d(x)} + j\,\widehat{N}_{3d(y)} + k\,\widehat{N}_{3d(z)} \quad (4.62)$$

Substituting equation (4.62) into equations (4.59) and (4.60) yields

$$Q_{\widehat{N}} = \cos\left(\frac{\theta_q}{2}\right) + i\,\widehat{N}_{3d(x)}\sin\left(\frac{\theta_q}{2}\right) + j\,\widehat{N}_{3d(y)}\sin\left(\frac{\theta_q}{2}\right) + k\,\widehat{N}_{3d(z)}\sin\left(\frac{\theta_q}{2}\right) \quad (4.63)$$

$$Q^*_{\widehat{N}} = \cos\left(\frac{\theta_q}{2}\right) - i\,\widehat{N}_{3d(x)}\sin\left(\frac{\theta_q}{2}\right) - j\,\widehat{N}_{3d(y)}\sin\left(\frac{\theta_q}{2}\right) - k\,\widehat{N}_{3d(z)}\sin\left(\frac{\theta_q}{2}\right) \quad (4.64)$$

where

$Q_{\widehat{N}}$ denotes unit quaternions of unit normal vector, and

$Q^*_{\widehat{N}}$ denotes unit quaternions conjugate of unit normal vector.

Let $\theta_q = \frac{\pi}{2}$radian, applying equations (4.61), (4.63) and (4.64) into (4.58) yields

$$\begin{aligned}\widehat{B} &= Q_{\widehat{N}} \times \widehat{T} \times Q^*_{\widehat{N}} \\ &= \left\{\cos\left(\frac{\theta_q}{2}\right) + i\,\widehat{N}_{3d(x)}\sin\left(\frac{\theta_q}{2}\right) + j\,\widehat{N}_{3d(y)}\sin\left(\frac{\theta_q}{2}\right) + k\,\widehat{N}_{3d(z)}\sin\left(\frac{\theta_q}{2}\right)\right\} \\ &\quad \times \left\{i\,\widehat{T}_{3d(x)} + j\,\widehat{T}_{3d(y)} + k\,\widehat{T}_{3d(z)}\right\} \\ &\quad \times \left\{\cos\left(\frac{\theta_q}{2}\right) - i\,\widehat{N}_{3d(x)}\sin\left(\frac{\theta_q}{2}\right) - j\,\widehat{N}_{3d(y)}\sin\left(\frac{\theta_q}{2}\right) - k\,\widehat{N}_{3d(z)}\sin\left(\frac{\theta_q}{2}\right)\right\} \quad (4.65)\end{aligned}$$

where

$\widehat{B}$ is the result of the quaternions rotation of $\widehat{T}$ around $\widehat{N}$.

Furthermore, substituting all $\widehat{N}$'s (equations (4.54) to (4.56)) and $\widehat{T}$'s (equations (4.42) to (4.44)) terms into equation (4.65), then expanding the result with consideration of quaternions multiplication rule depicted in Table (4.3) yields

$$\widehat{B} = i\left\{\widehat{R}_{Ti}\right\} + j\left\{\widehat{R}_{Tj}\right\} + k\left\{\widehat{R}_{Tk}\right\} \quad (4.66)$$

where

$\widehat{R}_{Ti}$ is the magnitude of perpendicular unit vector in x_l-direction,

$\widehat{R}_{Tj}$ is the magnitude of perpendicular unit vector in y_l-direction, and

$\widehat{R}_{Tk}$ is the magnitude of perpendicular unit vector in z_l-direction.

Expanding equation 4.65, then simplify the result with Quaternion rules depicted in Table 4.3.

Table 4.3: Quaternion multiplication formula.

×	**1**	**i**	**j**	**k**
1	1	i	j	k
i	i	-1	k	-j
j	j	-k	-1	i
k	k	j	-i	-1

Rearranging the result in terms $\widehat{R}_{Ti}$, $\widehat{R}_{Tj}$ and $\widehat{R}_{Tk}$ yields

$$\widehat{R}_{Ti} = \begin{bmatrix} \cos^2\left(\frac{\theta_q}{2}\right) + \left(\widehat{N}_{3d(x)} - \widehat{N}_{3d(y)} - \widehat{N}_{3d(z)}\right)\sin^2\left(\frac{\theta_q}{2}\right) \\ 2\sin\left(\frac{\theta_q}{2}\right)\left[\sin\left(\frac{\theta_q}{2}\right)\widehat{N}_{3d(x)}\widehat{N}_{3d(y)} - \cos\left(\frac{\theta_q}{2}\right)\widehat{N}_{3d(z)}\right] \\ 2\sin\left(\frac{\theta_q}{2}\right)\left(\sin\left(\frac{\theta_q}{2}\right)\widehat{N}_{3d(x)}\widehat{N}_{3d(z)} + \cos\left(\frac{\theta_q}{2}\right)\widehat{N}_{3d(y)}\right) \end{bmatrix}^T \begin{bmatrix} \widehat{T}_{3d(x)} \\ \widehat{T}_{3d(y)} \\ \widehat{T}_{3d(z)} \end{bmatrix} \quad (4.67)$$

$$\widehat{R}_{Tj} = \begin{bmatrix} 2\sin\left(\frac{\theta_q}{2}\right)\left[\sin\left(\frac{\theta_q}{2}\right)\widehat{N}_{3d(x)}\widehat{N}_{3d(y)} + \cos\left(\frac{\theta_q}{2}\right)\widehat{N}_{3d(z)}\right] \\ \cos^2\left(\frac{\theta_q}{2}\right) + \left(\widehat{N}_{3d(y)} - \widehat{N}_{3d(z)} - \widehat{N}_{3d(x)}\right)\sin^2\left(\frac{\theta_q}{2}\right) \\ 2\sin\left(\frac{\theta_q}{2}\right)\left(\sin\left(\frac{\theta_q}{2}\right)\widehat{N}_{3d(y)}\widehat{N}_{3d(z)} - \cos\left(\frac{\theta_q}{2}\right)\widehat{N}_{3d(x)}\right) \end{bmatrix}^T \begin{bmatrix} \widehat{T}_{3d(x)} \\ \widehat{T}_{3d(y)} \\ \widehat{T}_{3d(z)} \end{bmatrix} \quad (4.68)$$

$$\widehat{R}_{Tk} = \begin{bmatrix} 2\sin\left(\frac{\theta_q}{2}\right)\left[\sin\left(\frac{\theta_q}{2}\right)\widehat{N}_{3d(x)}\widehat{N}_{3d(z)} - \cos\left(\frac{\theta_q}{2}\right)\widehat{N}_{3d(y)}\right] \\ 2\sin\left(\frac{\theta_q}{2}\right)\left(\sin\left(\frac{\theta_q}{2}\right)\widehat{N}_{3d(y)}\widehat{N}_{3d(z)} + \cos\left(\frac{\theta_q}{2}\right)\widehat{N}_{3d(x)}\right) \\ \cos^2\left(\frac{\theta_q}{2}\right) + \left(\widehat{N}_{3d(z)} - \widehat{N}_{3d(y)} - \widehat{N}_{3d(x)}\right)\sin^2\left(\frac{\theta_q}{2}\right) \end{bmatrix}^T \begin{bmatrix} \widehat{T}_{3d(x)} \\ \widehat{T}_{3d(y)} \\ \widehat{T}_{3d(z)} \end{bmatrix} \quad (4.69)$$

Furthermore, the magnitude length of $\widehat{B}$ is given as

$$\left\|\widehat{B}\right\| = \sqrt{\left(\widehat{R}_{Ti}\right)^2 + \left(\widehat{R}_{Tj}\right)^2 + \left(\widehat{R}_{Tk}\right)^2} \quad (4.70)$$

Returning to the derivation operation flowchart in Figure 4.4, now we have a complete local axes Cartesian system where tangent vector, binormal vector and normal vector denote the x_l, y_l and z_l respectively.

4.3.5 Rolling, Climbing and Turning Angles

In the previous section, vector fields associated with the desired mobile robot motion along the desired trajectory have been investigated. Additionally, from Figure 4.5, it has been observed that there are three potential rotational motions on an uneven surface which are: (1) rolling angle; (2) climbing angle; and (3) turning angle. It is very important to analyse these angles because it will reflect the physical behaviour of the mobile robot in order to negotiate with the terrain. For a mobile robot such as VGTV proposed in Chapter 2, it will configure one side of its track configuration to enhance stability in case of very step side-slope. Moreover, the mobile robot can also replan its trajectory in case of climbing angle exceeding its climbing limit.

The rotational motions of mobile robot along the desired trajectory are referred as the differ-

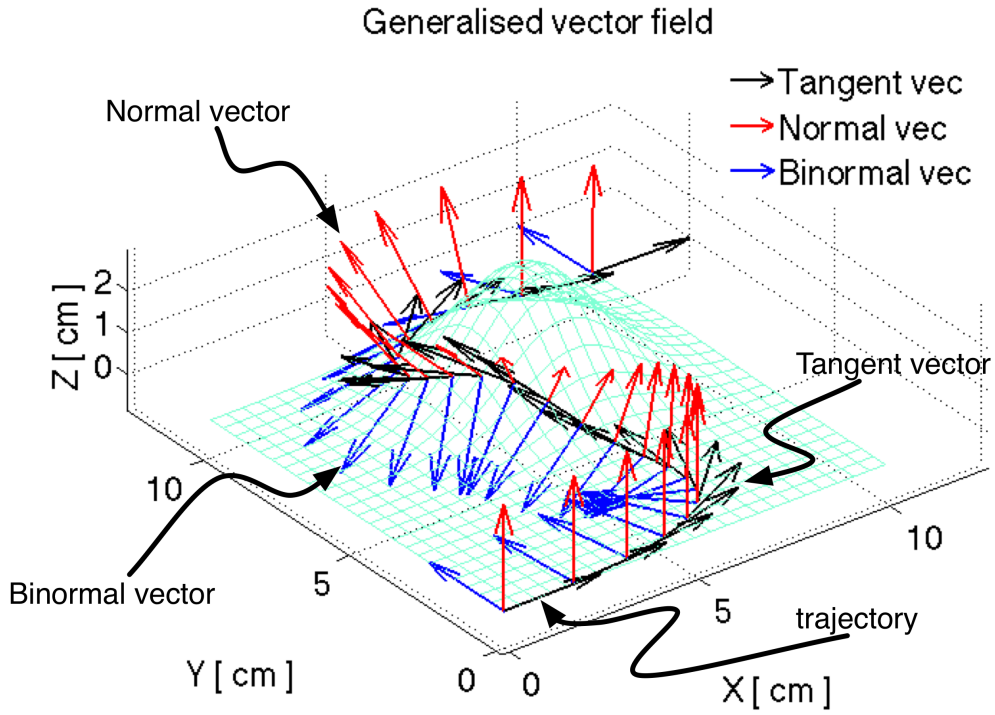
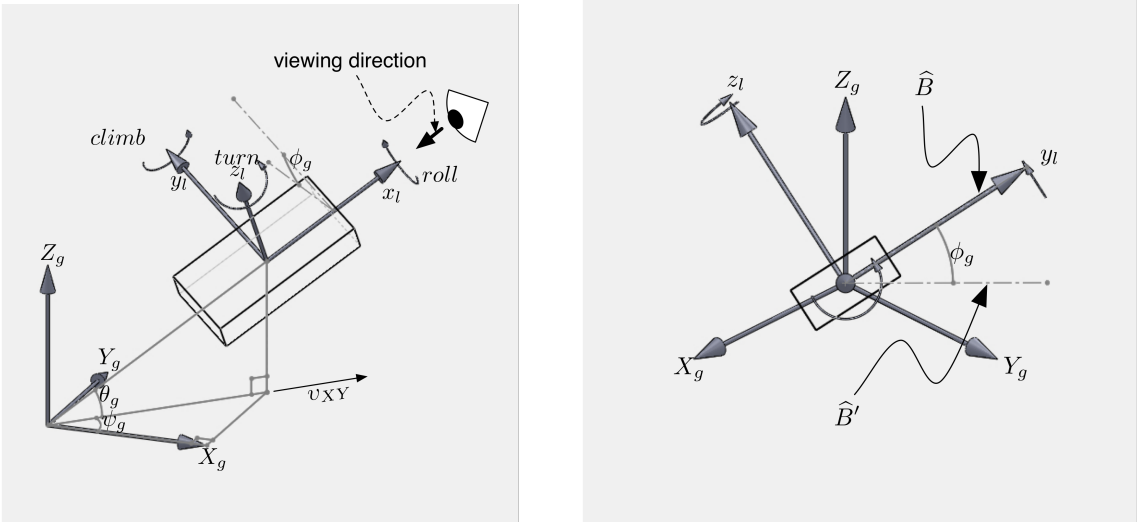


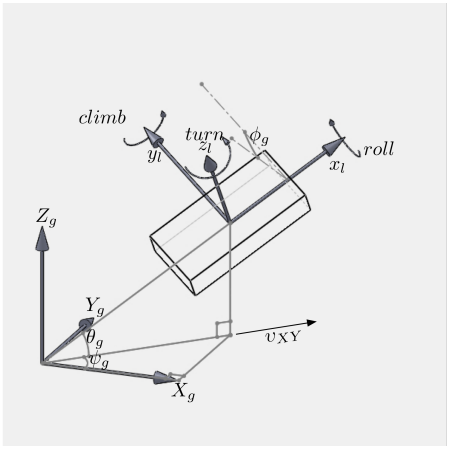
Figure 4.12: Illustration of the unit tangent vector, $\left\|\widehat{T}\right\|$, the unit normal vector, $\left\|\widehat{N}\right\|$, and the unit binormal vector, $\left\|\widehat{P}\right\|$ on 3-dimensional Cartesian space.



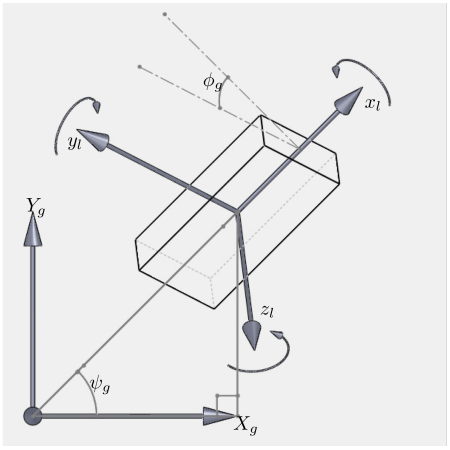
(a) 3-dimensional view.

(b) Front view.

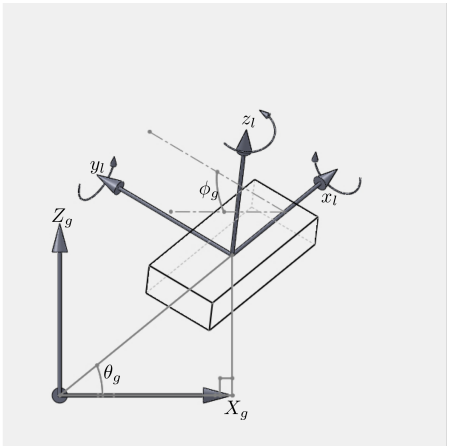
Figure 4.13: Formulation of rolling angle, ϕ_g. The geometric diagram on the right-hand-side visualise the two-dimensional view of ϕ_g construction. Note that the arrow indicates the direction of viewing.



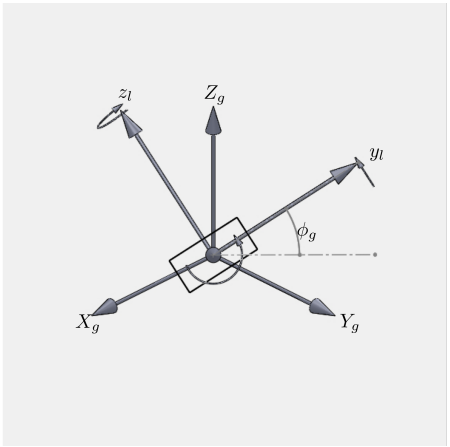
(a) 3-dimensional view.



(b) Top view $(XY)_g$-axis.



(c) Side viewXZ-axis.



(d) View from front of mobile robot.

Figure 4.14: Free-body-diagram of mobile robot on 3-dimensional space. Kinematic analysis of mobile robot on x, y, z planes. Subscript G and L are for 'global' and 'local' repectively. Mobile robot is replaced by a 'box' to simplify illustration.

ence between the desired local coordinate frame and the global frame. Therefore, the derivations of each angle associated with the rotational motions are derived with respect to the global frame. The definition of each rotational motion are graphically explained in Figure 4.14. Let the rolling angle be denoted as ϕ_g, consider the projection of unit binormal vector, $\widehat{B}$, onto $(XY)_g$-plane as depicted in Figure 4.13 as $\widehat{B}'$. Therefore, ϕ_g is an angle enclosed between $\widehat{B}$ and $\widehat{B}'$. Figure 4.13b shows a two-dimensional graphical explanation of ϕ_g formulation and the mathematical relationship is given by

$$\phi_g = \tan^{-1}\left(\frac{R_{Tk}}{\sqrt{R_{Ti}^2 + R_{Tj}^2}}\right) \tag{4.71}$$

On the one hand, Figure 4.14c explains the geometrical formation between $\vec{T}_{3d}$ and $\vec{T}_{2d}$ which produces an angle of climbing designated as θ_g. Meanwhile, the turning angle, ψ_g, is generated

through an angle enclosed by $\vec{T}_{3d(x)}$ and $\vec{T}_{3d(y)}$. Considering equations (4.42) and (4.43), it can be seen that ψ_g is constant regardless of the type surface (flat or uneven). Therefore, θ_g and ψ_g are given by

$$\theta_g = \tan^{-1}\left(\frac{\widehat{T}_{3d(z)}}{T_{2d}}\right) \tag{4.72}$$

$$\psi_g = \tan^{-1}\left(\frac{\widehat{T}_{2d(y)}}{\widehat{T}_{2d(x)}}\right) \tag{4.73}$$

In this section, the rolling, climbing and turning angles have been defined in order to complete the derivation process for uneven surface as described in Figure 4.4. The next step is to derive the kinematic model of the mobile robot for uneven surface as illustrated in Figure 4.5 with consideration of assumptions in Table 5.1.

4.3.6 Kinematic Modeling of a Mobile Robot on Pre-planned Trajectory

Now, consider the mobile robot configurations on an uneven surface as depicted in Figure 4.5. At position 3, observe that the mobile robot produces rolling, climbing and turning motion as well as changes in coordinates, (x, y, z), relative to the position 1 (initial position). Again, using Figure 4.14 to illustrate free-body-diagrams associated only with the rotational motions. The rolling motion is a rotational motion of the mobile robot when it traverses on a side slope. Figure 4.14d explains the rolling angle, ϕ_{3d}, is an angle developed by the rotational motion around x_l-axis. Additionally, the climbing angle, θ_{3d}, as illustrated in Figure 4.14c, is an angle when the mobile robot is ascending or desending a slope. Furthermore, the turning angle, ψ_{3d}, is an angle constructed between x_l and X_g, as described in Figure 4.14b. Examining Figure 4.6 and Figure 4.14b, $\psi_{3d} = \psi_{2d}$ regardless of surface condition. Therefore, mobile robot configurations for an uneven surface, $\dot{q}_{3d}$, is given by

$$\dot{q}_{3d} = \begin{bmatrix} \dot{x}_{3d} & \dot{y}_{3d} & \dot{z}_{3d} & \dot{\phi}_{3d} & \dot{\theta}_{3d} & \dot{\psi}_{3d} \end{bmatrix}^T$$

Now, let $\vec{v}_p$ denote the velocity vector in forward direction of the mobile robot whereas the projection of $\vec{v}_p$ onto $(XY)_g$-plane denoted by $\vec{v}_{p_{XY}}$ and explained in Figure 4.15, hence

$$\vec{v}_{p_{XY}} = \vec{v}_p \cos\theta_{3d} \tag{4.74}$$

Assume the mobile robot does not change forward direction whilst climbing the inclination, $\dot{x}_{3d}$ and $\dot{y}_{3d}$ with reference to $(XY)_g$-plane are given by

$$\dot{x}_{3d} = \vec{v}_{p_{XY}} \cos\psi_{3d} \tag{4.75}$$

$$\dot{y_{3d}} = \vec{v}_{p_{XY}} \sin\psi_{3d} \tag{4.76}$$

Note that deriving $\dot{x}_{3d}$ and $\dot{y}_{3d}$ with reference to $(XY)_g$-plane are identical to equations (4.3) and (4.4) because the projection $\vec{v}_p$ onto $(XY)_g$-plane is very much alike kinematic analysis in Figure 4.6. Furthermore, substituting equation (4.74) into (4.75) and (4.76) yields

$$\dot{x}_{3d} = \vec{v_p} \cos\theta_{3d} \cos\psi_{3d} \tag{4.77}$$

$$\dot{y_{3d}} = \vec{v}_p \cos\theta_{3d} \sin\psi_{3d} \tag{4.78}$$

whereby

$$\dot{z}_{3d} = \vec{v}_p \sin\theta_{3d} \tag{4.79}$$

The linear velocity components of mobile robot configuration, $\dot{q}_{3d}$, with respect to the global frame has been described in equations (4.77), (4.78) and (4.79). Similar to derivation of kinematic model of mobile robot on flat surface in Section (4.2.2), the relation between angular velocity and mobile robot motion is required to complete the formulation of kinematic model of mobile robot on uneven surface.

For a flat surface, we have seen that the turning angle rate is given by the angular velocity with respect to the instantenous centre of rotation (ICR). In contrast, the angular rate for mobile robot motion on uneven surface is more complex because the angular motion is decomposed of rolling, climbing and turning motion. In order to solve the angular velocities part of $\dot{q}_{3d}$, assume the rotational motions of the mobile robot immitates the roll, pitch and yaw motion of an aircraft vehicle. Consider Euler rate in terms of Euler angles and body angular velocity in flight stability study (Nelson (1998)), the angular velocities are written as

$$\dot{\phi} = \omega_x + \omega_y \sin\phi \tan\theta + \omega_z \cos\phi \tan\theta \tag{4.80}$$

$$\dot{\theta} = \omega_y \cos\phi - \omega_z \sin\phi \tag{4.81}$$

$$\dot{\psi} = (\omega_y \sin\phi + \omega_z \cos\phi) \sec\theta \tag{4.82}$$

where

$\dot{\phi}$ is the angular rate for roll angle,

$\dot{\theta}$ is the angular rate for pitch angle, and

$\dot{\psi}$ is the angular rate for yaw angle.

However, in mobile robot motion on uneven surface, it is more accurate to define these angular rates as:

$\dot{\phi}$ is the rolling angle rate resulted by a side-slope,

$\dot{\theta}$ is the climbing angle rate resulted from climbing an inclination motion, and

$\dot{\psi}$ is the turning angle rate resulted from turning motion.

Rearranging equations (4.77) to (4.82) in the form of equation (4.1) yields

$$\underbrace{\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}_{3d}}_{\dot{q}} = \underbrace{\begin{bmatrix} \cos\theta\cos\psi & 0 \\ \cos\theta\sin\psi & 0 \\ \sin\theta & 0 \\ 0 & \cos\phi\tan\theta \\ 0 & -\sin\phi \\ 0 & \cos\phi\sec\theta \end{bmatrix}}_{G(q)} \underbrace{\begin{bmatrix} \upsilon_p \\ \omega_z \end{bmatrix}}_{u} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & \sin\phi\tan\theta \\ 0 & \cos\phi \\ 0 & \sin\phi\sec\theta \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix}}_{\text{unwanted motion(s)}} \tag{4.83}$$

where

ω_z is the angular velocity referring to the turning motion, and

ω_x and ω_y are the angular velocities with respect to rolling and climbing motion.

As described in Section 4.2, the nonlinear surface properties of an uneven surface model are "unwanted" parameters in desired mobile robot motion. This means, ω_x and ω_y are inevitable disturbances of kinematic model in equation 4.83 correspond to parameters ϕ and θ. Therefore, ϕ and θ are undesirable but measureable.

Eliminating terms ω_x and ω_y yields

$$\underbrace{\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}_{3d}}_{\dot{q}} = \underbrace{\begin{bmatrix} \cos\theta\cos\psi & 0 \\ \cos\theta\sin\psi & 0 \\ \sin\theta & 0 \\ 0 & \cos\phi\tan\theta \\ 0 & -\sin\phi \\ 0 & \cos\phi\sec\theta \end{bmatrix}}_{G(q)} \underbrace{\begin{bmatrix} \upsilon_p \\ \omega_z \end{bmatrix}}_{u} \tag{4.84}$$

and the pseudo-velocity is given by equation (4.2), hence

$$\underbrace{\begin{bmatrix} \upsilon_p \\ \omega_z \end{bmatrix}}_{u} = \underbrace{\left(\begin{bmatrix} \cos\theta\cos\psi & 0 \\ \cos\theta\sin\psi & 0 \\ \sin\theta & 0 \\ 0 & \cos\phi\tan\theta \\ 0 & -\sin\phi \\ 0 & \cos\phi\sec\theta \end{bmatrix}^T \begin{bmatrix} \cos\theta\cos\psi & 0 \\ \cos\theta\sin\psi & 0 \\ \sin\theta & 0 \\ 0 & \cos\phi\tan\theta \\ 0 & -\sin\phi \\ 0 & \cos\phi\sec\theta \end{bmatrix} \right)^{-1}}_{(G^T(q)G(q))^{-1}} \times \underbrace{\begin{bmatrix} \cos\theta\cos\psi & 0 \\ \cos\theta\sin\psi & 0 \\ \sin\theta & 0 \\ 0 & \cos\phi\tan\theta \\ 0 & -\sin\phi \\ 0 & \cos\phi\sec\theta \end{bmatrix}^T}_{G^T(q)} \underbrace{\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}}_{\dot{q}} \tag{4.85}$$

Expanding equation (4.85) yields

$$\begin{bmatrix} \upsilon_p \\ \omega_z \end{bmatrix} = \begin{bmatrix} \left(\cos^2\theta\cos^2\psi + \cos^2\theta\sin^2\psi + \sin^2\theta\right) & 0 \\ 0 & \left(\cos^2\phi\tan^2\theta + \sin^2\phi + \frac{\cos^2\phi}{\cos^2\theta}\right) \end{bmatrix}^{-1} \times \begin{bmatrix} \dot{x}\cos\theta\cos\psi + \dot{y}\cos\theta\sin\psi + \dot{z}\sin\theta \\ \dot{\phi}\cos\phi\tan\theta - \dot{\theta}\sin\phi + \dot{\psi}\cos\phi\sec\theta \end{bmatrix} \tag{4.86}$$

Calculating the determinant of $\begin{bmatrix} \left(\cos^2\theta\cos^2\psi + \cos^2\theta\sin^2\psi + \sin^2\theta\right) & 0 \\ 0 & \left(\cos^2\phi\tan^2\theta + \sin^2\phi + \frac{\cos^2\phi}{\cos^2\theta}\right) \end{bmatrix}^{-1}$, then solving equation (4.86) yields

$$\begin{bmatrix} \upsilon_p \\ \omega_z \end{bmatrix} = \begin{bmatrix} \frac{\dot{x}\cos\theta\cos\psi + \dot{y}\cos\theta\sin\psi + \dot{z}\sin\theta}{(\cos^2\theta\cos^2\psi + \cos^2\theta\sin^2\psi + \sin^2\theta)} \\ \frac{\dot{\phi}\cos\phi\tan\theta - \dot{\theta}\sin\phi + \dot{\psi}\cos\phi\sec\theta}{\left(\cos^2\phi\tan^2\theta + \sin^2\phi + \frac{\cos^2\phi}{\cos^2\theta}\right)} \end{bmatrix} \tag{4.87}$$

Consider the "unwanted" terms described earlier, terms $\dot{\phi}$'s and $\dot{\theta}$'s in equation (4.87) are cancelled. Substituting equation (4.14) into equation (4.87) to compute the approximation of the

generalised input velocity vectors. Therefore

$$\begin{bmatrix} \upsilon_p \\ \omega_z \end{bmatrix} \approx \begin{bmatrix} \frac{\dot{x}\cos\theta\cos\psi+\dot{y}\cos\theta\sin\psi+\dot{z}\sin\theta}{(\cos^2\theta\cos^2\psi+\cos^2\theta\sin^2\psi+\sin^2\theta)} \\ \frac{\left(\frac{\ddot{y}_{2d}\dot{x}_{2d}-\ddot{x}_{2d}\dot{y}_{2d}}{\dot{x}_{2d}^2+\dot{y}_{2d}^2}\right)\cos\phi\sec\theta}{\left(\cos^2\phi\tan^2\theta+\sin^2\phi+\frac{\cos^2\phi}{\cos^2\theta}\right)} \end{bmatrix} \tag{4.88}$$

For a mobile robot that is supposed to cruise on horizontal flat surface, the parameters ϕ and θ are not measureable. Therefore, consider and apply $\left(\cos^2+\sin^2=1\right)$ into equation (4.88) yielding

$$\begin{bmatrix} \upsilon_p \\ \omega_z \end{bmatrix} = \begin{bmatrix} \dot{x}\cos\psi+\dot{y}\sin\psi \\ \frac{\ddot{y}_{2d}\dot{x}_{2d}-\ddot{x}_{2d}\dot{y}_{2d}}{\dot{x}_{2d}^2+\dot{y}_{2d}^2} \end{bmatrix} \tag{4.89}$$

which is identical to equations (4.12) and (4.14). Therefore, the inverse kinematic model described by equation (4.89) is reliable based on assumptions made earlier.

Even though rolling and climbing motion are not considered in the higher level control layer, the parameters associated with these motions are possible to measure via an appropriate sensor. Consequently, equations (4.12) and (4.14) are applicable for the mobile robot in Figure 4.5 with assumptions:

- high level control layer defines a two-dimensional pre-planned trajectory,
- the mobile robot traverse at a constant speed, and
- uneven surface is negotiable, meaning the mobile robot does not require any additional mechanisms (e.g., stabiliser or transformable mechanism ability) to negotiate with inclination and side-slope.

As a summary for Section 4.3, we have described a complete mathematical derivation for mobile robot kinematic behaviour especially associated with its motion on uneven terrain. Given a pre-planned trajectory and a surface model (Section 4.2) the mobile robot is able to forecast the appropriate velocity behaviour in order to negotiate with specific type of terrain. In the next section, comprehensive computer programmed simulations is done to study the behaviour of non-holonomic mobile robot moves along the desired trajectory on uneven surfaces.

4.4 Simulation Results

In order to assess the proposed direct and inverse kinematic model in equations (4.84) and (4.89), the mobile robot motion is simulated along the desired trajectory on the uneven surface. The

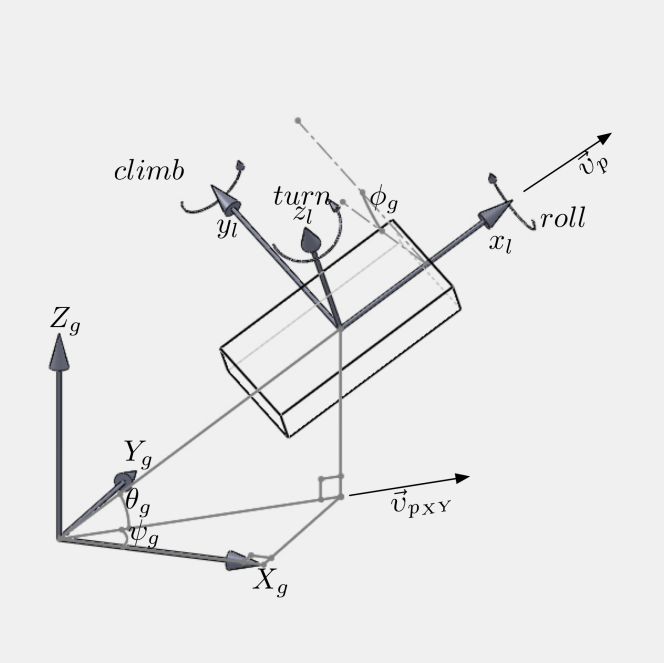


Figure 4.15: Projection of vector v_p onto $(XY)_g$- axis from Figure 4.14a.

objective of the computer simulation is to study the behaviour of the mobile robot that has to travel on an uneven surface. In this section, the simulation is divided into three parts which are:

1. investigate the effect of initial, $(x_i,\ y_i,\ \psi_i)$, final, $(x_f,\ y_f,\ \psi_f)$, configuration parameters and "free parameter", k, on the pre-planned path,

2. generate surface models of random uneven terrains utilising normal distribution function defined in equation (4.29), then compute the vector fields over the surface to get the gradient of the surfaces.,

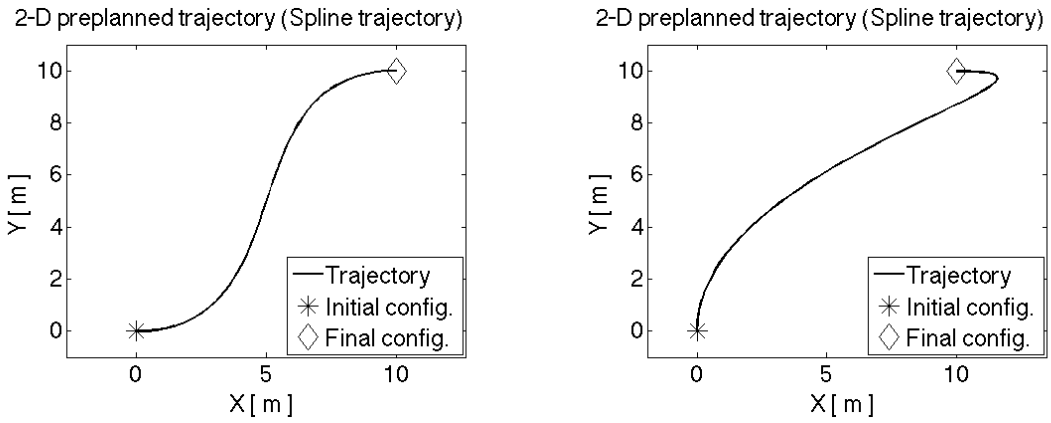
3. synthesise the two-dimensional pre-planned path and the surface model in order:

   (a) to compute linear and angular positions of along the path on the surfaces,

   (b) to plot the local axes, $(x_l,\ y_l,\ z_l)$, with respect to right-hand-rule.

   (c) to study the behaviour in term of generalised velocity (v and ω), of the mobile robot on the surfaces.

In Figure 4.16, the spline curvature of the two-dimensional pre-planned paths is studied with respect to the timing-law, s, and a constant free parameter, $k = 20$. The first simulated path, shown in Figure 4.16a is a spline with initial configuration $(x_i, y_i, \psi_i) = (0, 0, 0)$ and final configuration $(x_f, y_f, \psi_f) = (10, 10, 0)$. With the same free parameter, $k = 20$, the second path in Figure 4.16b is set to initial configuration $(x_i, y_i, \psi_i) = \left(0, 0, \frac{\pi}{2}\right)$ and final configuration $(x_f, y_f, \psi_f) = (10, 10, \pi)$. On the other hand, Figure 4.17a to 4.17d distinguish the



(a) $\psi_i = 0$, $\psi_f = 0$, $k = 20$.

(b) $\psi_i = \frac{\pi}{2}$, $\psi_f = \pi$, $k = 20$.

Figure 4.16: Effect of ψ_i and ψ_f in Cubic Cartesian.

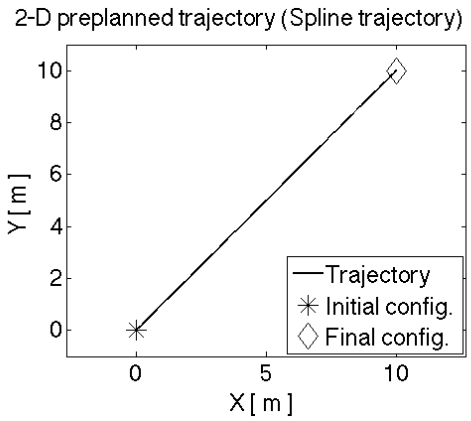
two-dimensional pre-planned paths with different "free parameter" values. Now, initial and final configurations are fixed at $(x_i, y_i, \psi_i) = (0, 0, 0)$ and $(x_f, y_f, \psi_f) = (10, 10, 0)$. The deeper spline curvature is a result from increasing value of k as shown in Figure 4.17d (with $k = 50$) compared to Figure 4.17a (with $k = 0$).

Further analyses are employed to the simulated surface model as depicted in Figures 4.18a to 4.18d as well as two-dimensional path in Figure 4.17d. The simulated surface models are generated via equation (4.29). Meanwhile, it is beneficial to compute the gradient of these surfaces which gives orthogonal vector fields to the level curves as illustrated from Figure 4.19 to Figure 4.22. The gradient vector fields formulation in equation (4.46) is useful to derive the normal and binormal vector of the path.

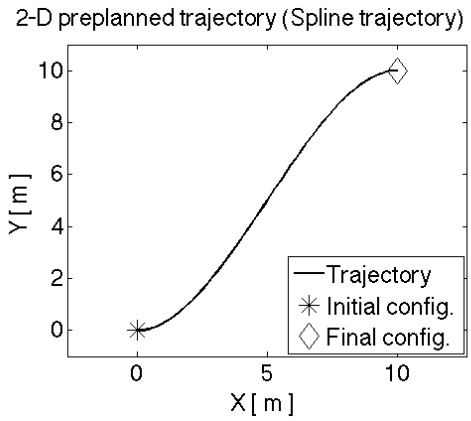
Kinematic synthesis between the surface models and the two-dimensional path produces realistic 3-dimensional path to the surfaces curve where results are shown in Figures 4.23a to 4.23d. The calculation of tangent vectors along the 3-dimensional path resulting generalised velocity vectors along the path. This calculation is important to compute normal and binormal vectors. The plot of these vector fields is functional for the computation mobile robot rotational behaviour along the path. As mentioned in previous assumptions, wheel slippage and angular rate for rolling and climbing is beyond of the scope of this thesis. Therefore, the generalised linear and angular velocity vectors (equation (4.89)) for the mobile robot are described in Figure 4.27.

4.5 Discussion and Concluding Remarks

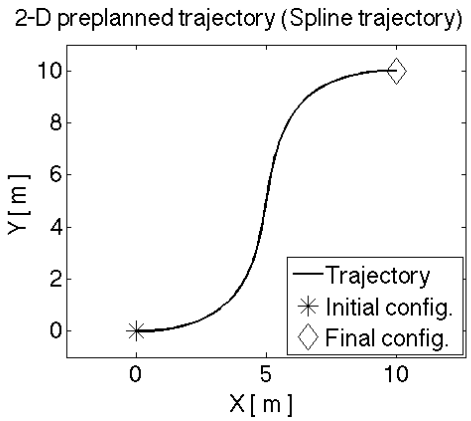
This chapter was written with the objective to derive a kinematic model for a mobile robot that can traverse on an uneven surface. This study set out with the aim of evaluating the mobile



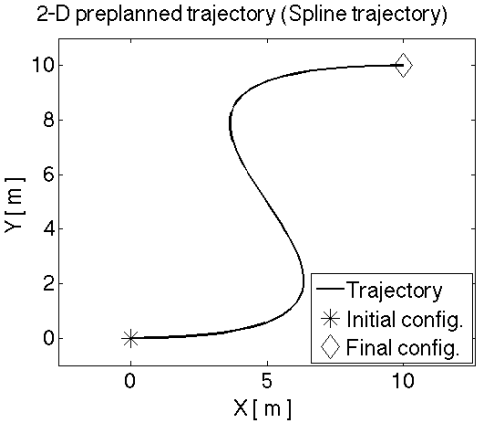
(a) $k = 0$.



(b) $k = 10$.

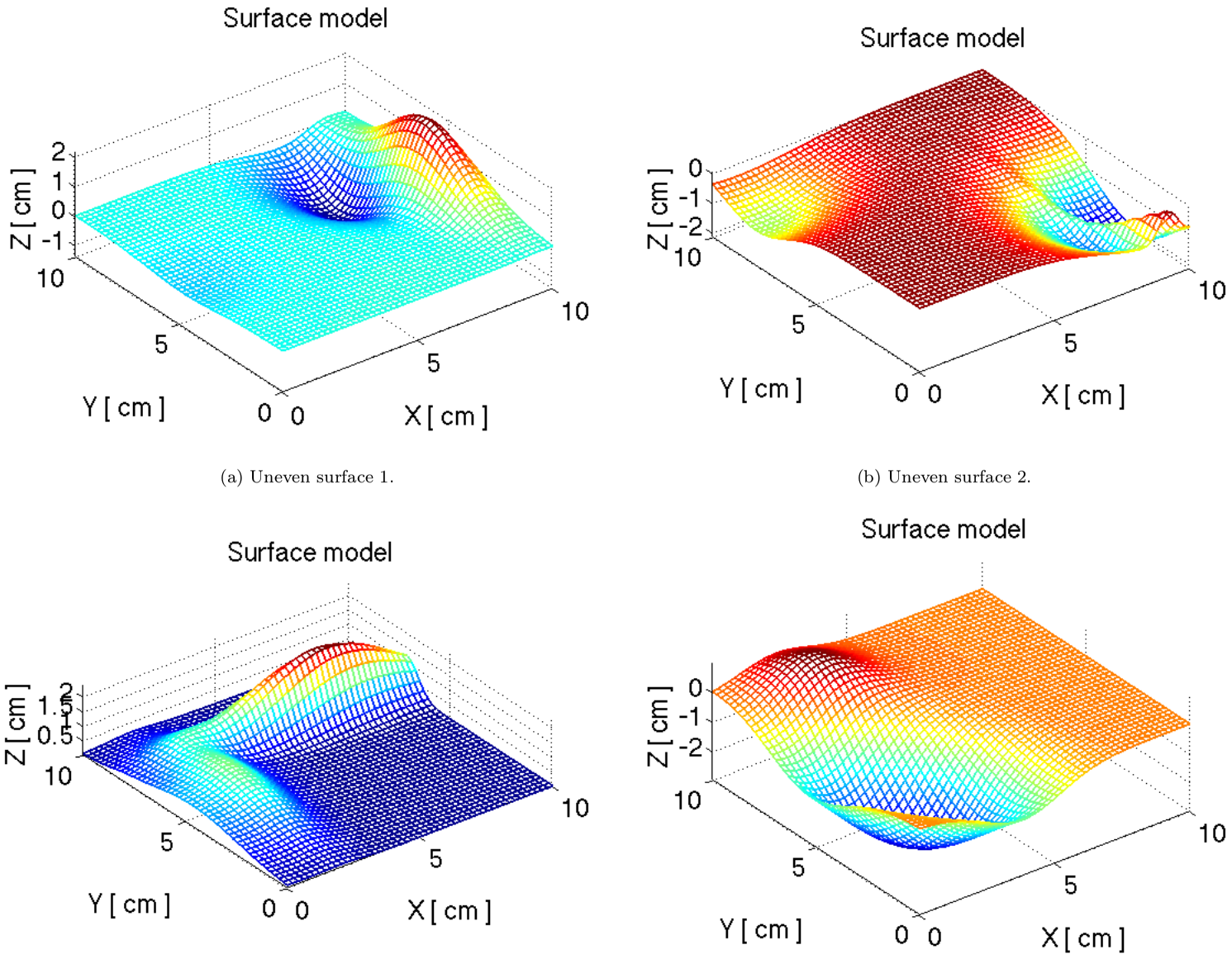


(c) $k = 25$.



(d) $k = 50$.

Figure 4.17: Effect of free parameter, k, in parametric trajectory with $(x_i, y_i) = (0, 0)$ and $(x_f, y_f) = (10, 10)$.



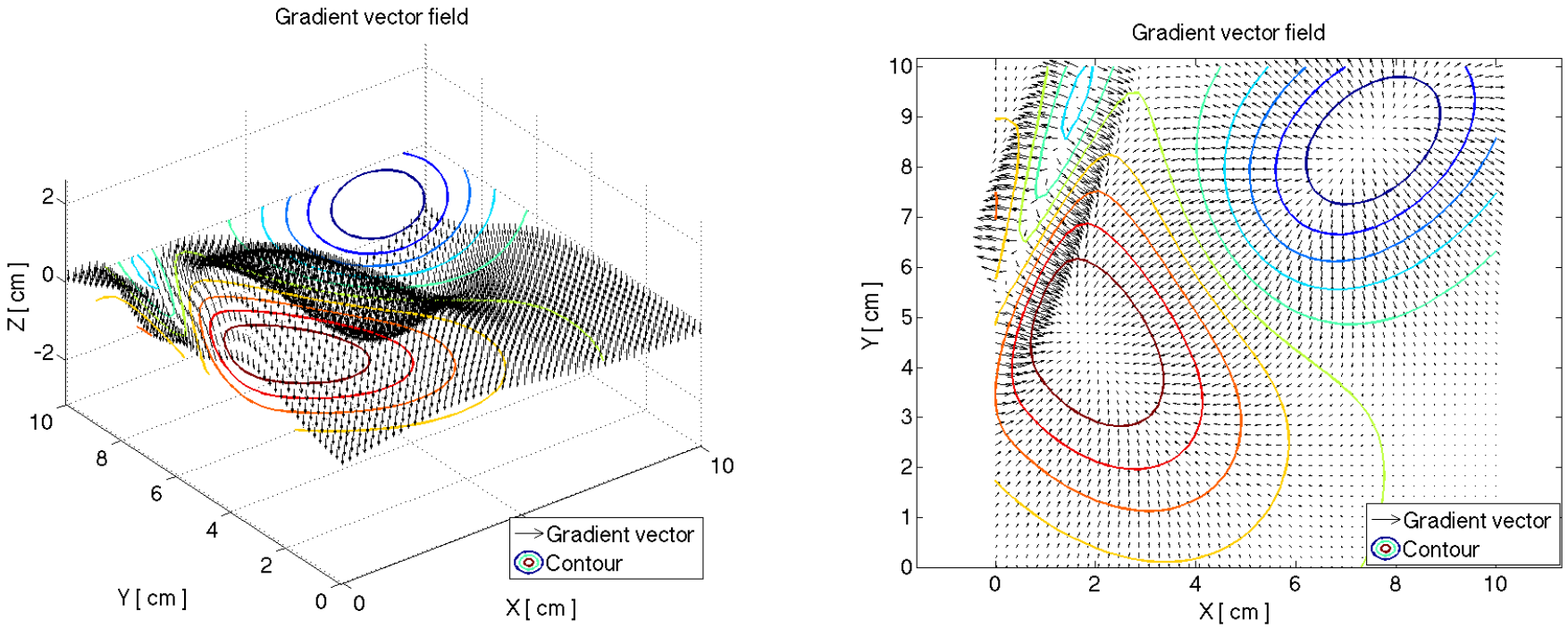
(a) Uneven surface 1.

(b) Uneven surface 2.

(c) Uneven surface 3.

(d) Uneven surface 4.

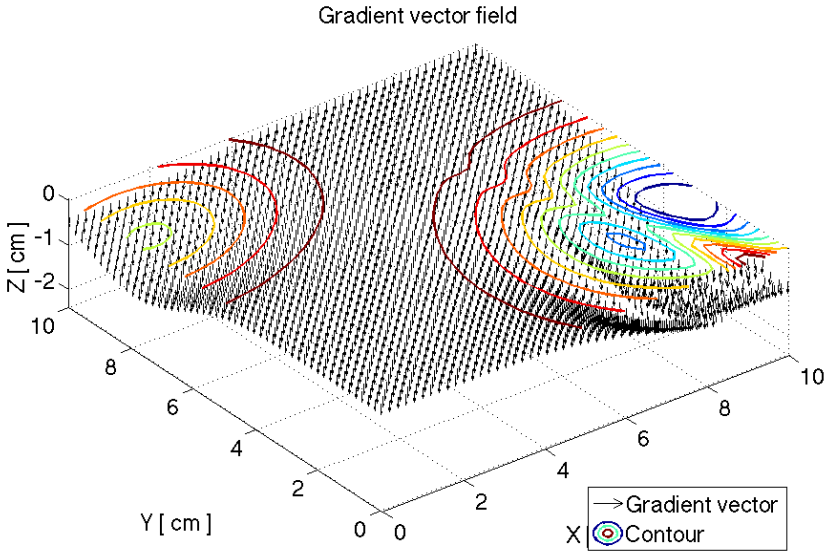
Figure 4.18: Four random simulated uneven surface model.



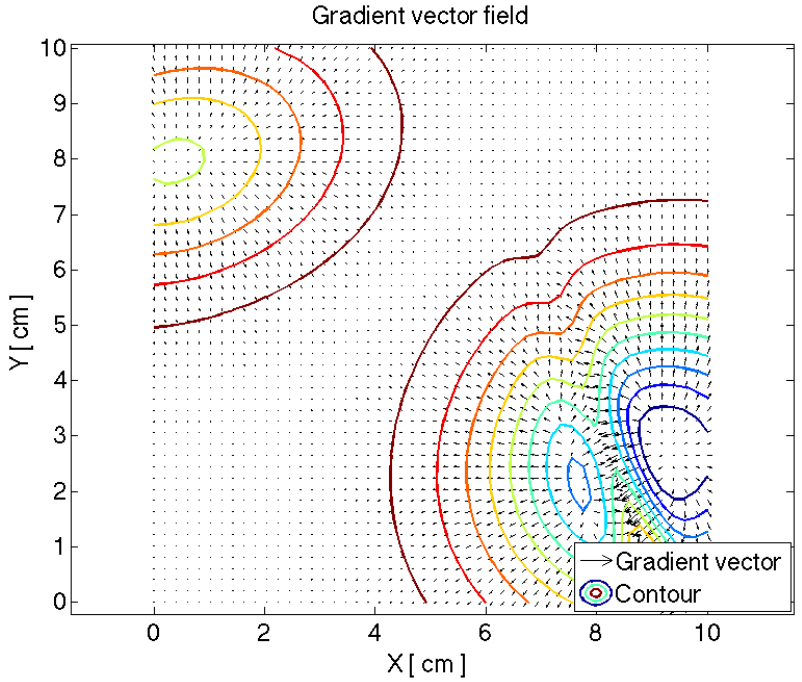
(a) 3-dimensional view of gradient vector on surface model 1.

(b) Two-dimensional view of gradient vector on surface model 1.

Figure 4.19: Gradient vector on simulated uneven surface model 1.

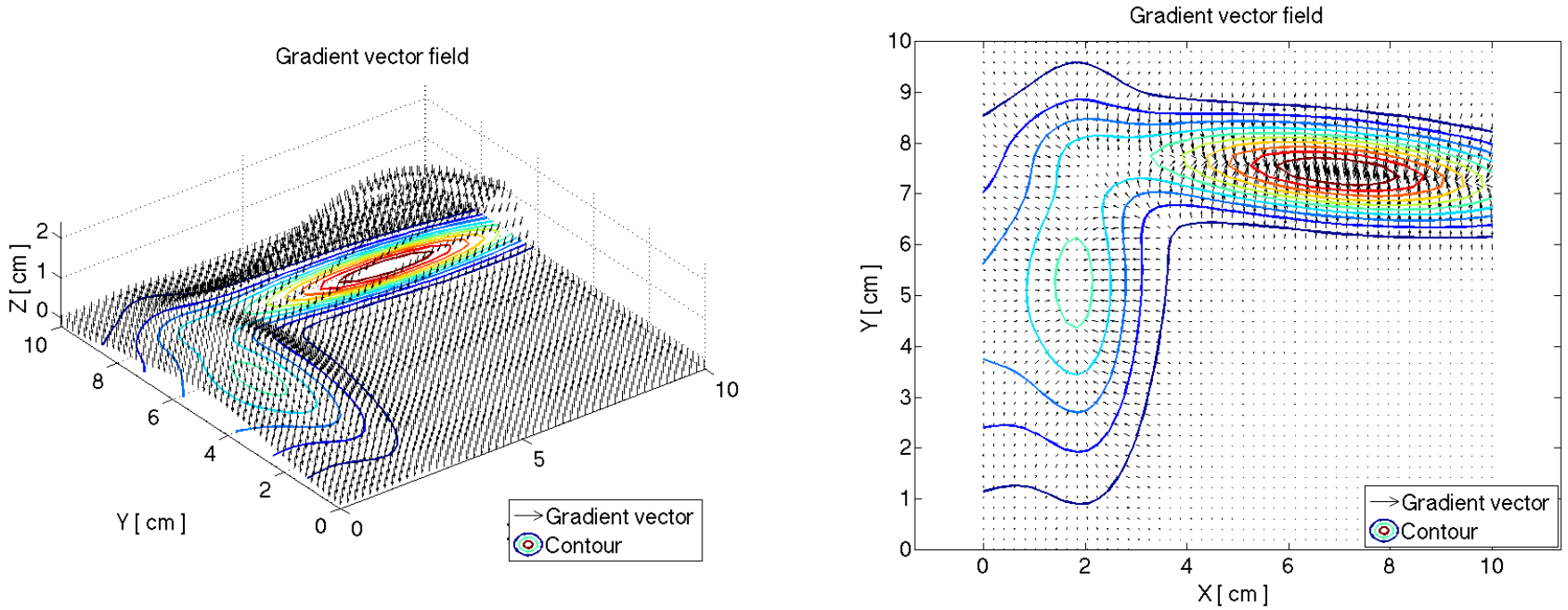


(a) 3-dimensional view of gradient vector on surface model 2.



(b) Two-dimensional view of gradient vector on surface model 2.

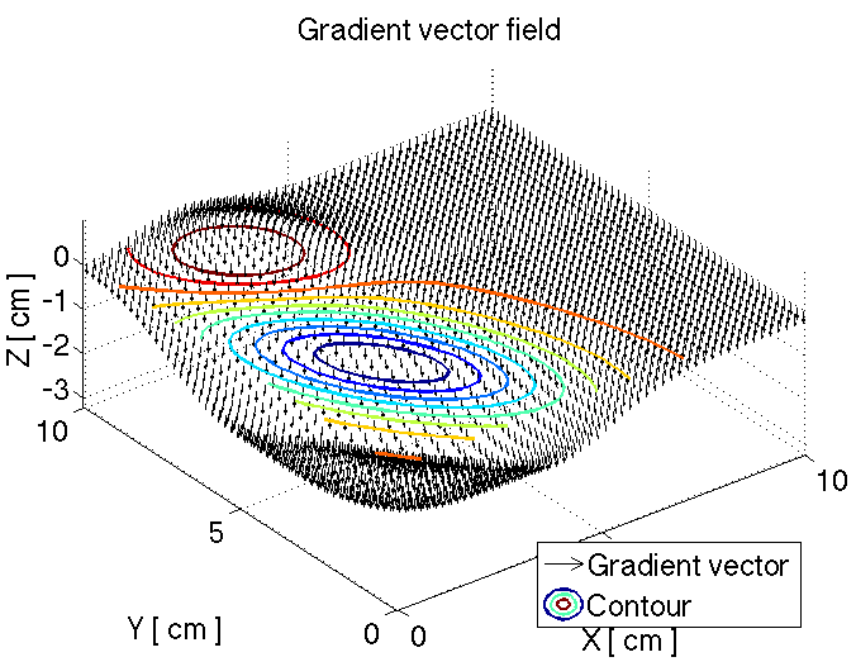
Figure 4.20: Gradient vector on simulated uneven surface model 2.



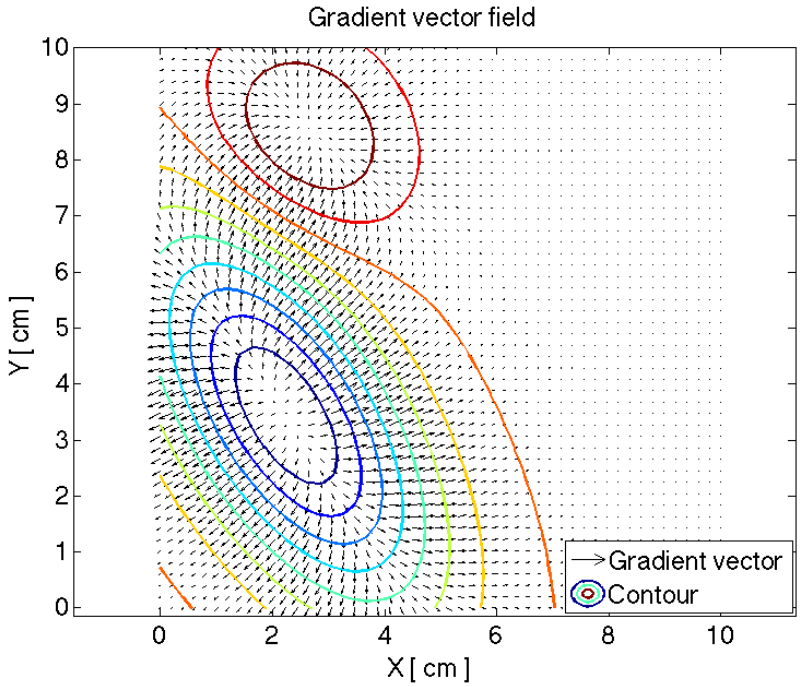
(a) 3-dimensional view of gradient vector on surface model 3.

(b) 3-dimensional view of gradient vector on surface model 3.

Figure 4.21: Gradient vector on simulated uneven surface model 3.

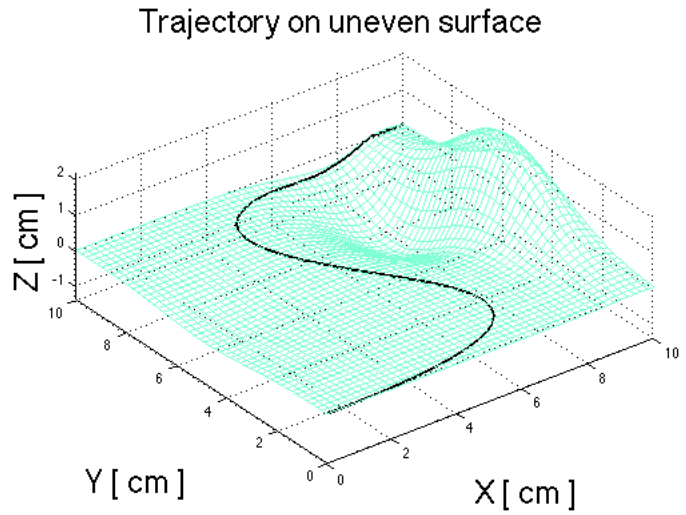


(a) 3-dimensional view of gradient vector on surface model 4.

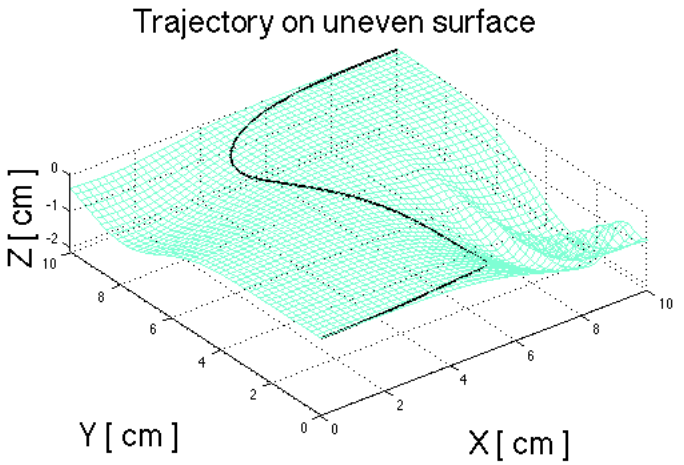


(b) 3-dimensional view of gradient vector on surface model 4.

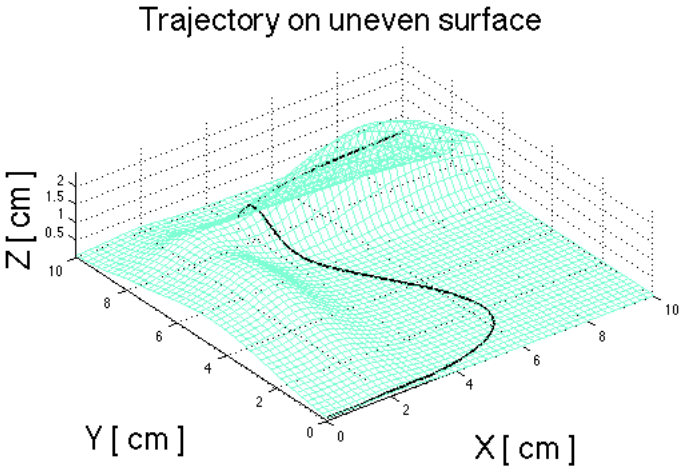
Figure 4.22: Gradient vector on simulated uneven surface model 4.



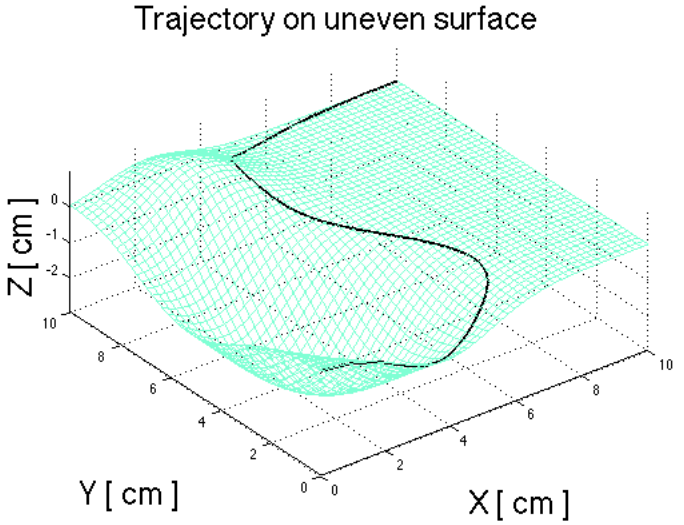
(a) 3-dimensional trajectory of surface 1.



(b) 3-dimensional trajectory of surface 2.

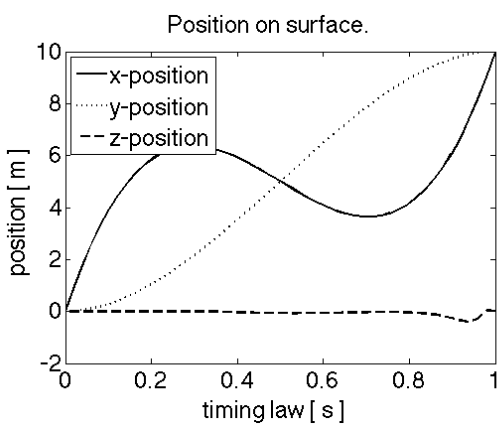


(c) 3-dimensional trajectory of surface 3.

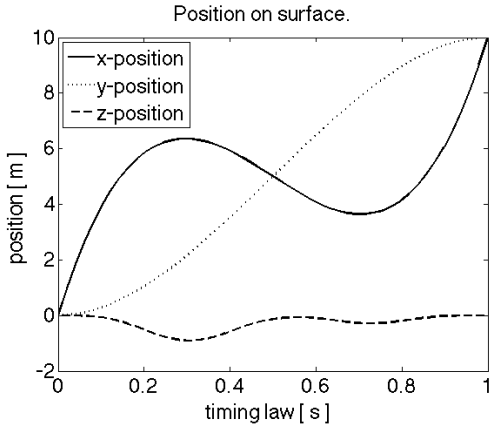


(d) 3-dimensional trajectory of surface 4.

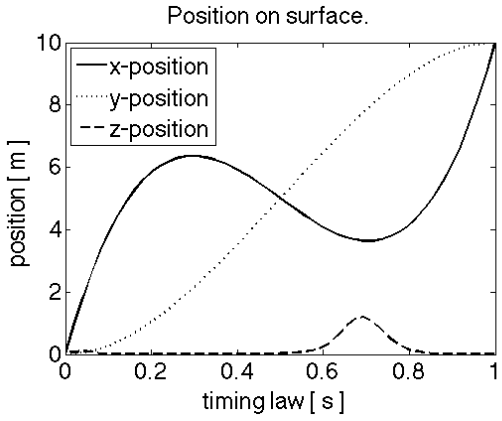
Figure 4.23: 3-dimensional trajectory on uneven surface models.



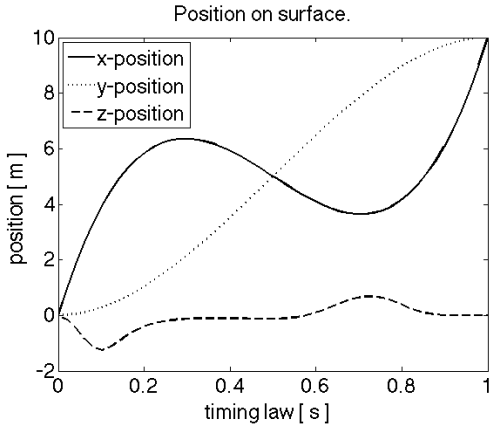
(a) Position in xyz-direction for surface 1.



(b) Position in xyz-direction for surface 2.

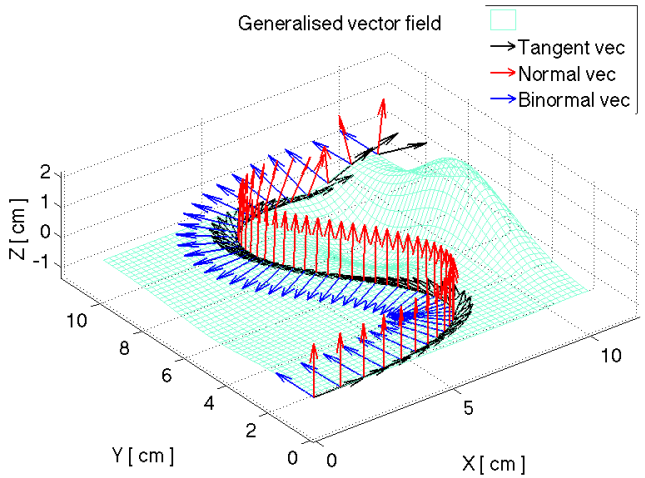


(c) Position in xyz-direction for surface 3.

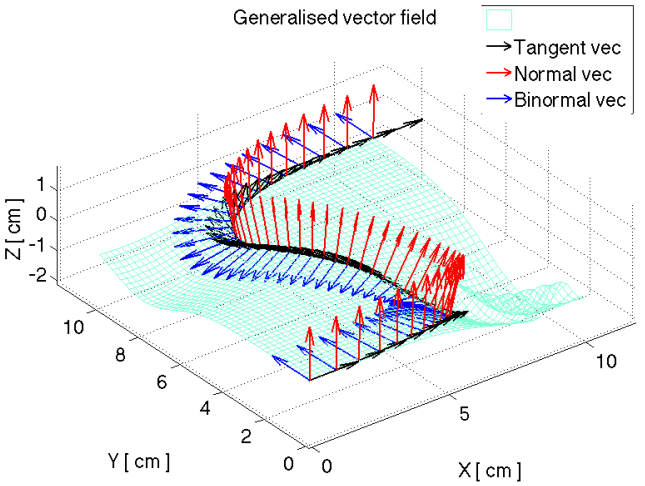


(d) Position in xyz-direction for surface 4.

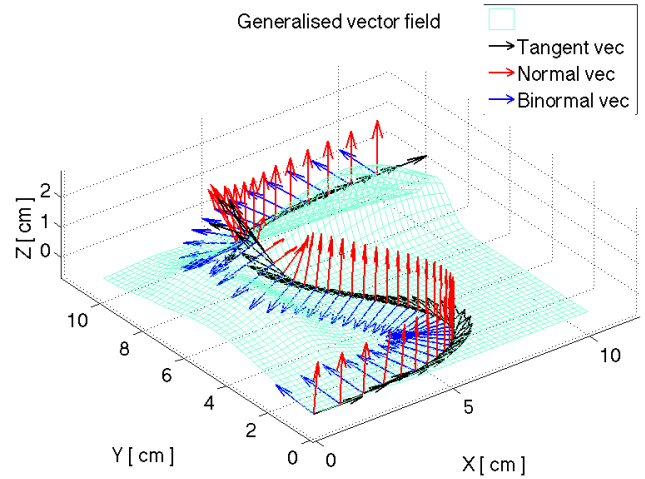
Figure 4.24: Position configuration on surface models.



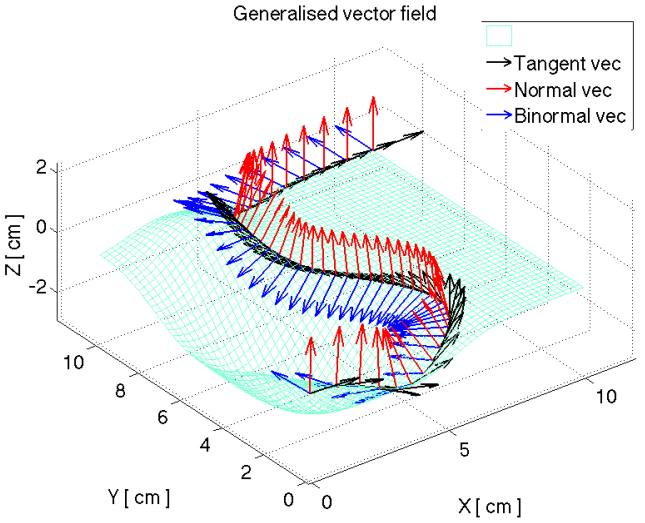
(a) Generalised vectors on surface 1.



(b) Generalised vectors on surface 2.

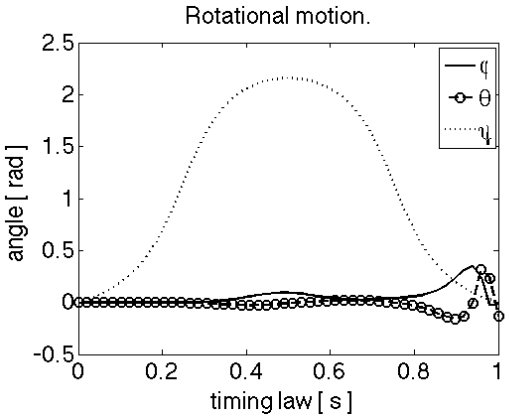


(c) Generalised vectors on surface 3.

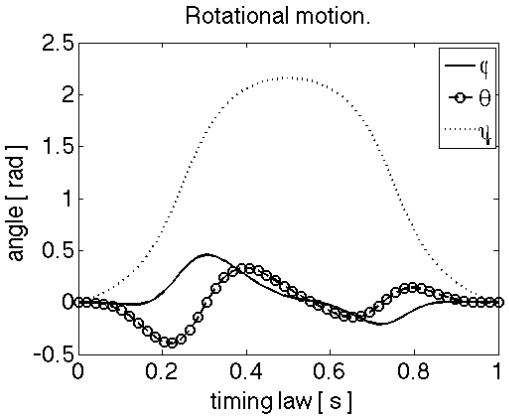


(d) Generalised vectors on surface 4.

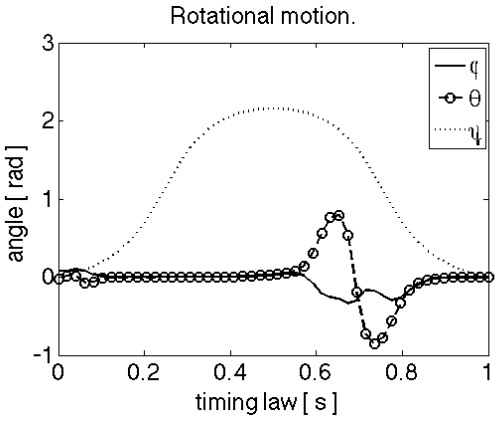
Figure 4.25: Generalised vectors on surface.



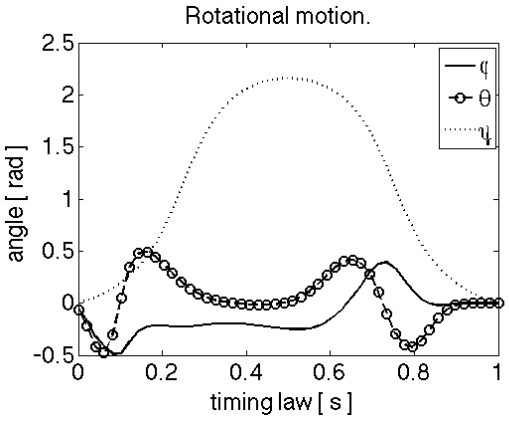
(a) Rotational motion of surface 1.



(b) Rotational motion of surface 2.

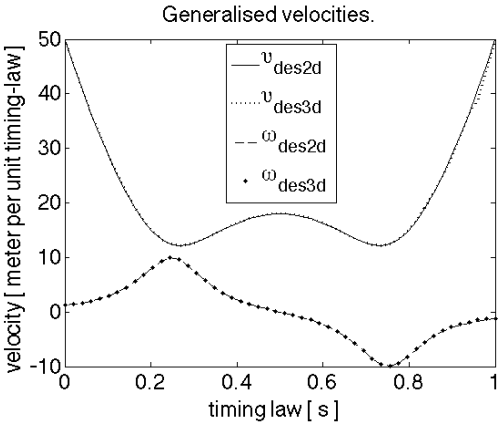


(c) Rotational motion of surface 3.

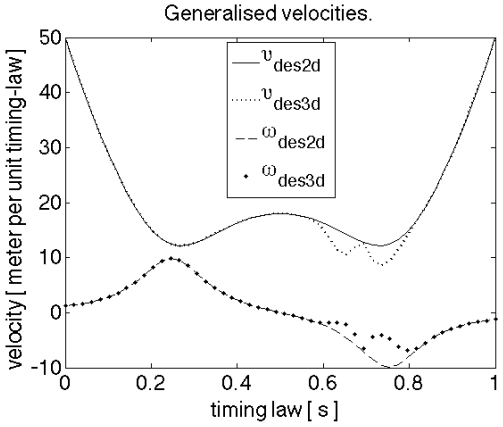


(d) Rotational motion of surface 4.

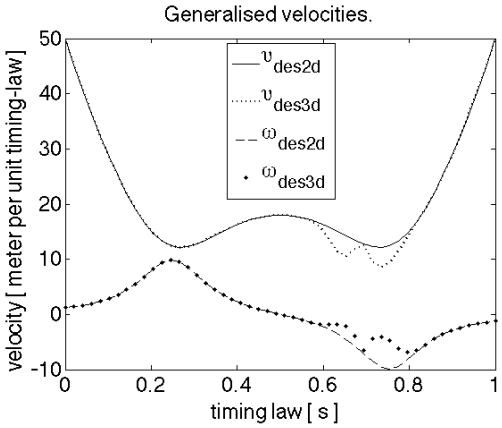
Figure 4.26: Rotational motion.



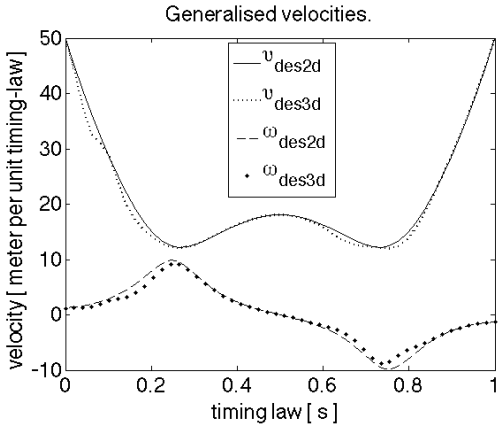
(a) Velocity vectors, (v_{3d}, ω_{3d}),for surface model 1.



(b) Velocity vectors, (v_{3d}, ω_{3d}),for surface model 2.



(c) Velocity vectors, (v_{3d}, ω_{3d}),for surface model 3.



(d) Velocity vectors, (v_{3d}, ω_{3d}),for surface model 4.

Figure 4.27: Generalised velocity vectors.

robot behaviour in terms of linear and angular velocities along a pre-planned path of an uneven surface with the knowledge of two-dimensional kinematic problem laid by De Luca et al. (2001). Even though the kinematic model for non-holonomic mobile robot is not unique by definition, the proposed generalised model constructs a comprehensive and solid foundation of the framework to study kinematic problem in real-world environment. The result in this study indicates:

- uneven terrain can be modelled by a the summation of several multivariate Gaussian functions;
- surface model is beneficial for mobile robot to deal with terrain or obstacles rather than avoiding it;
- unit quaternion rotation formulation is useful as an alternative to Euler angle; and
- input vector, u, for uneven surface can be reduced to two-dimensional case considering the elimination of undesired rotational element from the equation.

It is interesting to note that in all four simulated uneven surface model used of this study, the input velocity vectors are nearly equivalent (or approximate) to two-dimensional kinematic problem.

This chapter proposed an extension of the existing solution for the problem of deriving kinematic model for a mobile robot that is supposed to traverse on a flat surface as defined in higher level control layer. However, due to the uncertain and highly complex surface of the real-world environment, the mobile robot has to deal with an uneven terrain. With some assumptions described earlier in this chapter, the real-world terrain is simulated into a surface model before it is synthesised with the pre-defined path resulting in a new realistic path-on-terrain information for the mobile robot to predict its motions along the surface. The proposed model has certain advantages over existing two-dimensional kinematic model because:

1. it is capable to express mobile robot position in terms of altitude changes relative to zero-ground level[1];
2. it is capable to determine its orientation with respect to side-slope and inclination is observed.

The kinematic model is tested on a two-dimensional pre-planned path with "free parameter" $k = 50$ onto four random generated surfaces. Precisely, the direct kinematic model is relied on angular orientation (ϕ, θ, ψ) as well as input vectors to achieve the particular configuration

[1]Zero-ground level is a level that is defined by the zero height. This can be visualise by observing Z_g-axis at value zero ($Z_g = 0$) in Figure 4.18 and constructing imaginary horizontal plane cutting through $Z_g = 0$ at any value of X_g and Y_g.

along the pre-planned trajectory. On the other hand, the inverse kinematic model is derived by utilising pseudo-inverse method from Siciliano et al. (2009) to compute the generalised velocity vectors (υ and ω) along the surfaces. Returning to the objective posed in the beginning in this chapter, it is now possible to state that input vector for the proposed kinematic solution is approximately equivalent to the two-dimensional case. Therefore, one can employ the input vectors for a two-dimensional model on a further analysis of navigation problem such as tracking the desired trajectory.

In the next chapter, we will again discuss the control of mobile robot navigation problem in unknown, complex and unstructured environment. The knowledge from this chapter, in particular the approximate input vectors, will become the reference input in the navigation tracking problem. In two-dimensional domain, the tracking problem is solvable and converges to the desired trajectory if one can identify the appropriate gains for the controller to drive the mobile robot closer to the desired trajectory. Assumptions made in this chapter will be forwarded to next chapter.

Chapter 5

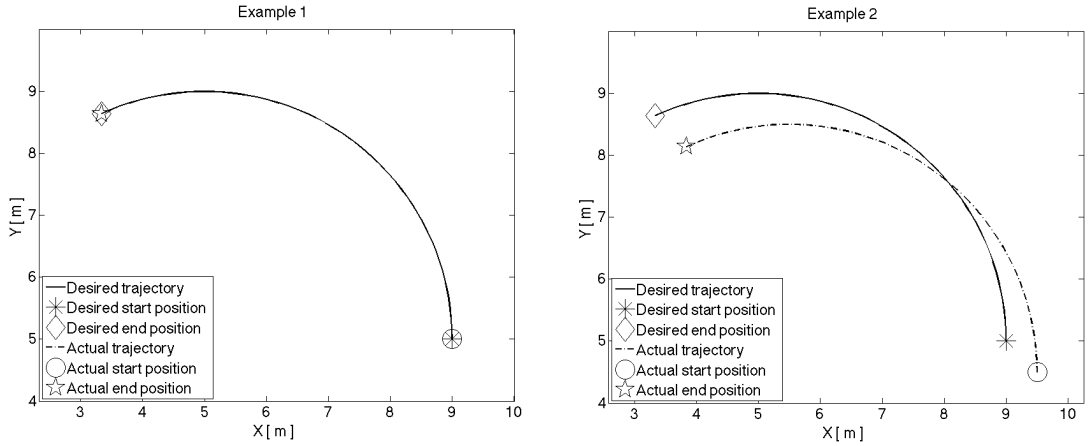
Trajectory Tracking Control of Mobile Robot on Uneven Surface

5.1 Introduction

In Chapter 4, we investigated the kinematic behaviour of a mobile robot subject to the terrain condition (flat or uneven surface) in which the solution of the kinematic model represented the fundamental solution of the mobile robot navigation problem. This chapter presents a control method of mobile robot trajectory tracking, modeling and control under the same mechanical constraint explained in Chapter 4, while focusing on the utilisation of the classical control technique of a flat surface for an uneven surface problem.

The basis of the navigation problem for mobile robot system is derived either from ***path following*** or ***trajectory tracking***. Path following, which emphasises the term ***following***, by definition is the competency of a mobile robot to follow the designated path and to keep a certain distance with the path. This can be illustrated by a curb following mobile robot where it cruises along the curb's curvature and at the same time keeps a certain distance from the curb's edge (Samson, 1995). In contrast, trajectory tracking which emphasises the term ***tracking***, is the ability of a mobile robot to achieve convergence to the designated path/trajectory. This chapter focusses on the trajectory tracking problem to achieve convergence to a desired trajectory.

Assume a pre-planned trajectory generated using an offline "trajectory generator", leading to a series of desired velocity and acceleration vectors along the desired trajectory. The vectors are fed into the "trajectory following algorithm" resulting in onboard computations to calculate the appropriate velocities of the driving mechanism with respect to the generated trajectory. Now, consider a mobile robot controlled purely based on these desired velocities. It will start to cruise and produce an actual trajectory which is close to the desired trajectory. Imagine if the mobile robot's actual start position is accurately placed on the desired start point, the actual trajectory will coincide with the desired trajectory as illustrated in Figure 5.1a. However,



(a) Initial actual configurations are equal to the desired initial configurations.

(b) Initial actual configurations are different from the desired initial configurations.

Figure 5.1: Illustration of mobile robot motions with open-loop tracking control scheme. Figure 5.1a shows the actual trajectory overlapped with the desired trajectory when the actual initial configurations matches the desired initial configurations. Meanwhile, in Figure 5.1b, the possible actual trajectory is a reproduction of the desired trajectory. However, the actual trajectory does not converge to the desired trajectory because the actual initial configurations are different with the desired configurations.



Figure 5.2: General flowchart of mobile robot trajectory tracking problem.

if the mobile robot starts further away from the desired start point, the actual trajectory will only replicate but not converge to the desired trajectory as depicted in Figure 5.1b. Meanwhile, mobile robot posture is being update between a constant time-interval in order to observe the deviation between the desired and actual configuration. The sequence of this generic mobile robot trajectory following process is illustrated in Figure 5.2.

Nonetheless, in the trajectory tracking problem, the mobile robot is required to cruise along the desired trajectory regardless of the starting point or trajectory offset due to disturbances (noise). In order to derive the most suitable tracking controller, it is convenient to derive the trajectory tracking problem of a non-holonomic mechanical system on flat surface and then extend the solution to the uneven surface. In the environment of flat surface, the mobile robot must follow the pre-planned Cartesian trajectory with a specified timing-law[1]. Although it is not necessary, it is convenient to define the pre-planned path in a form of parametrised geometrical path.

[1]See Section 4.2 in Chapter 4 for explanation about timing-law.

5.1.1 Motivations

For many years, there has been a considerable amount of research interest in studying various control methods for the trajectory tracking problem for non-holonomic mobile robot system. A mobile robot (or vehicle) can be classified as a non-holonomic mechanical system when its mobility is reduced by at least one mechanical constraint (Siciliano et al., 2009). A simple explanation of non-holonomic mechanical system can be illustrated with the observation of a car-like vehicle, where an appropriate turn angle via front-axle is required when the vehicle needs to turn right (or left). Limitation to turn instantenously is an example of non-holonomic mechanical constraint while turning angle developed via front-axle is associated with the vehicle's kinematics.

In general, the research activities have been motivated to achieve the desired motion behaviour associated with above mentioned constraint of specific kinematic model as explained in Chapter 4. The first control of trajectory tracking problem for a non-holonomic mobile robot has been proposed by Kanayama et al. (1988) using decomposition of errors between reference and current posture and became a fundamental solution to this problem. The stability of tracking control rule in this proposal is analysed using Lyapunov function while approximate linearisation approach is used to solve the nonlinear issue. In addition, the correction of mobile robot's posture errors are controlled through the desired linear and angular velocities (Kanayama et al., 1990, 1991). The motivation has also contributed various control techniques derived from the highly nonlinear tracking error dynamics. Some advanced control methods such as neural network (Fierro and Lewis, 1995), fuzzy control (Blazic, 2010), adaptive control (Martins et al., 2008), and model-based predictive control (Klancar and Skrjanc, 2007), also demonstrated the capability to solve the nonlinear issue in mobile robot trajectory tracking problem. In case of rough terrain (uneven surface), research activities are focussed on navigation problem rather than trajectory tracking with the basis of visual odometry (Konolige et al., 2011). Therefore, trajectory tracking of mobile robot motion on uneven surface remains unsolved.

5.1.2 Aim

The aim of this chapter is to solve the control problem of trajectory tracking motion for non-holonomic mobile robot on uneven surface using particular feedback technique. Due to the nonlinearity in error dynamics, a classical approximate linearisation technique as proposed in Kanayama et al. (1990, 1991) is utilised. In this chapter, we make similar assumptions to Chapter 4 where the workspace on the uneven surface is obstacle-free. This means that the mobile robot is able to cruise around the surface and is not required to configure its geometrical transformation in case of negotiating obstacles. In addition, we will present our analysis to the case of 3 different

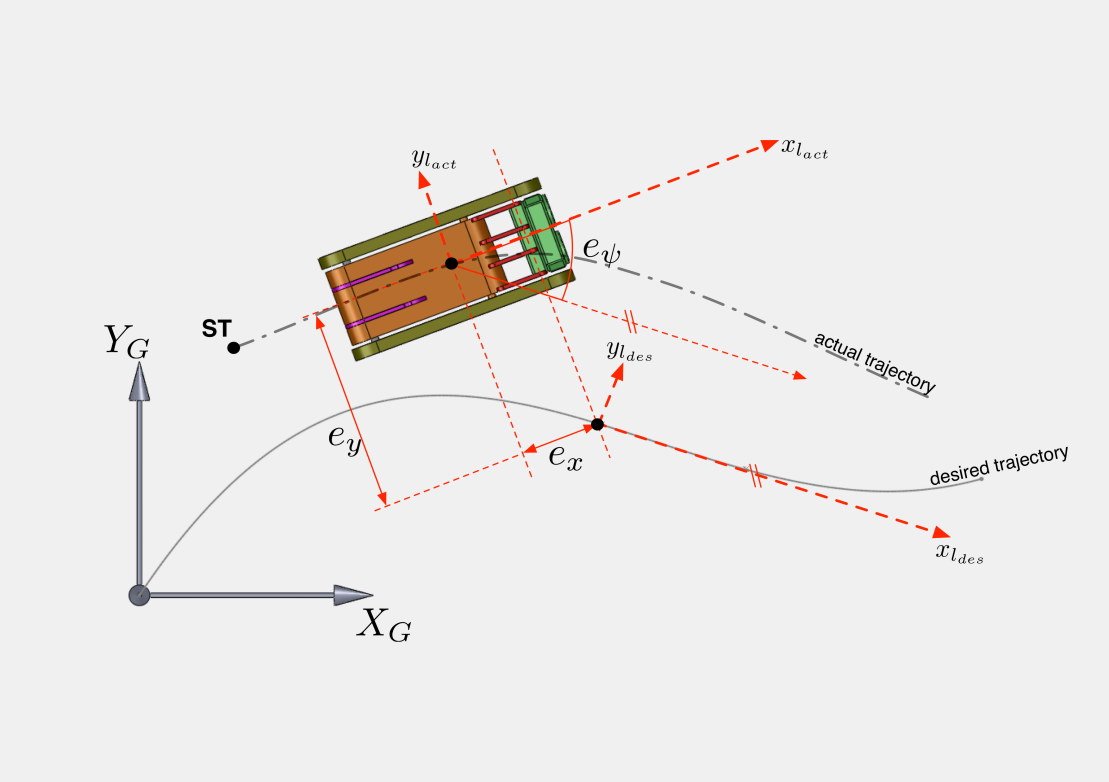


Figure 5.3: Geometric formulation of trajectory tracking for non-holonomic mobile robot cruising on flat surface. The desired, (-), and actual trajectory, (-.-), are represented by spline-curves. The global reference axis is denoted by (X_G, Y_G) while the local axis acting on mobile robot's body and along the desired trajectory are denoted by subscript l. The origin of global axis is the start point for the desired trajectory whereas the actual start point is label as ST. Configuration errors of mobile robot's posture with respect to the local axis are denoted as e_x, e_y and e_ψ. Additionally, the heading direction of the mobile robot is indicated by the axis $x_{l_{act}}$.

examples of pre-defined trajectory.

The rest of this chapter is arranged as follows; in Section 5.2, comprehensive problem statements for mobile robot motion on flat and uneven terrain are given, describing utilitisation of flat surface formulation on an uneven terrain environment. Section 5.3 explains the control strategy. The mathematical derivation of error tracking model based on Kalman filter position estimate is derived in Section 5.4.1 and 5.4.2. Furthermore, Section 5.4.3 describes the control laws and the controller performance and simulation results are presented in Section 5.5. The last two sections (Section 5.6 and Section 5.7) provide some discussion and concluding remarks related to the simulation results and controller performance.

5.2 Problem Statement

Assume a non-holonomic mobile robot exhibits a motion on a flat surface as depicted in Figure 5.3, where subscripts G and l denote the global and local axes acting on the centre of mass of the mobile robot respectively. The orientation of the mobile robot is given by the angle of

global axes and desired local axis (i.e., angle developed between X_G and $x_{l_{act}}$ represents the yaw angle, ψ_{act}, at specific time-interval). Therefore, configuration of mobile robot is given by the parameters of $x_{l_{act}}$, $y_{l_{act}}$ and ψ_{act}. In spite of that, the Cartesian and orientation errors are formulated through geometric difference between the actual local coordinate, $(x_{l_{act}}, y_{l_{act}})$, and the desired local coordinate, $(x_{l_{des}}, y_{l_{des}})$. These errors are denoted by e_x, e_y and e_ψ. Let mobile robot configuration be denoted as q, the desired, actual and configuration error are given by

$$q_{des} = \begin{bmatrix} x_{des} & y_{des} & \psi_{des} \end{bmatrix}^T \tag{5.1}$$

$$q_{act} = \begin{bmatrix} x_{act} & y_{act} & \psi_{act} \end{bmatrix}^T \tag{5.2}$$

$$e_q = \begin{bmatrix} e_x & e_y & e_\psi \end{bmatrix}^T \tag{5.3}$$

Let the similar mobile robot cruise on an uneven surface as shown in Figure 5.4. Here, multivariate Gaussian function is utilised to generate the uneven surface as discussed in Chapter 4. Three important assumptions are made: (1) the uneven surface is negotiable; (2) a simple mobile robot is used; and (3) the surface is obstacle-free. Assumptions 1 and 2 describe that the mobile robot unit does not require to vary its geometrical shape to negotiate the terrain. Furthermore, assumption 3 explains that an "obstacle avoidance" algorithm is not required as the mobile robot can roam freely on the surface.

Similar to the flat surface motion described in Figure 5.3, the desired and actual heading direction of mobile robot are illustrated by the directions of $x_{l_{des}}$ and $x_{l_{act}}$ respectively. Literally, these axes are the tangential components of the trajectories considering positive (ascending slope) and negative (decending slope) direction. Due to the variation of surface height, there is a need to consider mobile robot's position in z-axis direction. Additionally, rotational motions due to surface's slope can be identified as rolling and climbing which derive roll angle, ϕ, and climb angle, θ. Therefore, the desired and actual configuration parameters of mobile robot on an uneven surface are x, y, z, ϕ, θ and ψ (equations (5.4) and (5.5)). Furthermore, configuration errors of trajectory tracking on uneven surface can be denoted as $(e_x, e_y, e_z, e_\phi, e_\theta, e_\psi)$ as shown in equation (5.6).

$$q_{des} = \begin{bmatrix} x_{des} & y_{des} & z_{des} & \phi_{des} & \theta_{des} & \psi_{des} \end{bmatrix}^T \tag{5.4}$$

$$q_{act} = \begin{bmatrix} x_{act} & y_{act} & z_{act} & \phi_{act} & \theta_{act} & \psi_{act} \end{bmatrix}^T \tag{5.5}$$

$$e_q = \begin{bmatrix} e_x & e_y & e_z & e_\phi & e_\theta & e_\psi \end{bmatrix}^T \tag{5.6}$$

However, it is very useful to observe and transform the actual mobile robot configuration

parameters, q_{act}, into estimated parameters for further trajectory tracking derivation. This is a more natural approach due to presence of noise especially in position's sensor reading. Having said that, q_{mea} is the measurement data from position sensor, $\left(q_{mea} \overset{sensor(s)}{\Longleftarrow} q_{act} \right)$, and can be written as

$$\begin{bmatrix} x_{mea} \\ y_{mea} \\ z_{mea} \\ \phi_{mea} \\ \theta_{mea} \\ \psi_{mea} \end{bmatrix} \overset{sensor(s)}{\Longleftarrow} \begin{bmatrix} x_{act} \\ y_{act} \\ z_{act} \\ \phi_{act} \\ \theta_{act} \\ \psi_{act} \end{bmatrix} \tag{5.7}$$

Further computation of q_{mea} will lead to the estimated configuration of mobile robot and one could write the estimated configuration as

$$q_{est} = \begin{bmatrix} x_{est} & y_{est} & z_{est} & \phi_{est} & \theta_{est} & \psi_{est} \end{bmatrix}^T \tag{5.8}$$

Note that there are three separate algorithms before trajectory tracking problem can be solved. The first algorithm is the development of a 2-dimensional pre-planned path that is generated using parametric equations given by $x(s)$ and $y(s)$ in equations (4.19) and (4.20), and may vary depending on the shape of the desired trajectory as well as the timing-law. Then, a surface (terrain) generator that computes and generates an appropriate surface to visualise terrain condition, either flat or uneven surface. It employs a multivariate Gaussian function as described by equation (4.29) in Chapter 4 in case of an uneven surface. The last process is to find a set of appropriate desired configurations of the mobile robot (equation (5.4)) using kinematic model with consideration of the pre-planned path and surface parameter. In the same process, a series of desired mobile robot velocities (linear and angular) along the desired path is calculated using either one from these two methods which are: (1) the pseudo-inverse-kinematic model; or (2) reference velocity input equations in Siciliano et al. (2009) with of assumption constant acceleration. These three algorithms are discussed in Chapter 4.

5.3 Control Strategy

A three layer control architecture is proposed as a framework for mobile robot control as described in Chapter 4. The control architecture itself is a top-down approach in which the highest level control layer defines the mobile robot's mission, the intermediate level control layer focusses on the mobile robot's configuration control and the low level control layer executes instructions from

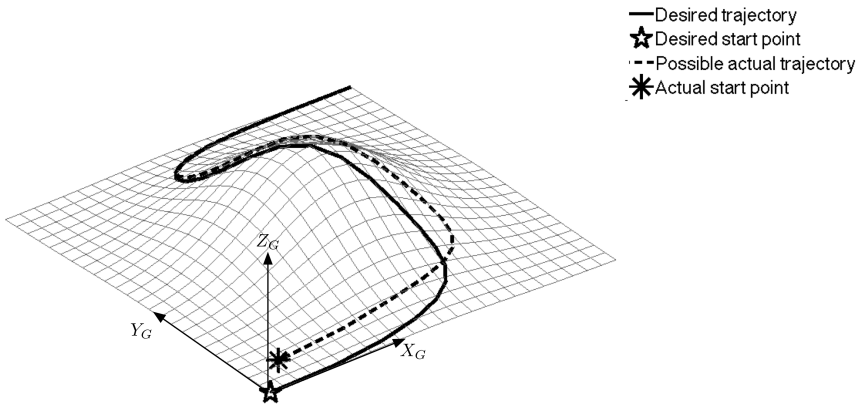


Figure 5.4: Problem statement for trajectory tracking of mobile robot on uneven surface. The solid spline-curve (-) is the desired trajectory while dashed spline-curve (- -) is the actual trajectory. Consider similar mobile robot in Figure 5.3 travelling on this uneven surface. There are three additional configurations added with respect to the global frame when the mobile robot cruises towards the centre of the surface which are: (1) change in altitude due to elevation of surface; (2) climbing motion; and (3) rolling motion. Even though mobile robot's configuration is computed with the reference of global coordinate system, the derivation of configuration error is computed in local coordinate system regardless of the surface condition.

the intemediate level control layer and translates it into the actuators' motion (i.e. wheel speed). This can be illustrated briefly by the motion of a mobile robot on a flat surface. In this example, the high level control layer only defines two important information which are the desired start position and desired end position. The desired traveling time between start to end position is not important. Then, an external path/trajectory planner suggests an appropriate travel plan (2-dimensional pre-planned path) before the surface generator models a surface with respect to the actual terrain (as described in Chapter 4). In the intermediate level control layer, the pre-planned path is derived into a series of desired mobile robot configurations with respect to the global coordinate, q_{des}, within a constant time-interval and considering the surface condition. In the same control layer, q_{des} is set as the control setpoint for the trajectory tracking algorithm. In Figure 5.5, the flowchart on the left-hand-side illustrates the hierarchical control layer in mobile robot control framework while the control block diagram on the right-hand-side explains the operations in the intermediate level control layer. The magnifier icon on the flowchart indicates that this chapter will focus on the intermediate level control layer.

Now, let a mobile robot prepare to cruise on an uneven surface, Figure 5.4, where q_{des} has been defined and set as the setpoint in the control block diagram as shown in Figure 5.5. The actual start position is assumed to be different to the desired start position. Additionally, the desired velocities, v_{des} and ω_{des}, are derived from the parametric equations of 2-dimensional

pre-planned path (Siciliano et al., 2009),

$$\upsilon_{des} = \pm\sqrt{\dot{x}_{des}^2 + \dot{y}_{des}^2} \tag{5.9}$$

$$\omega_{des} = \frac{\ddot{y}_{des}\dot{x}_{des} - \ddot{x}_{des}\dot{y}_{des}}{\dot{x}_{des}^2 + \dot{y}_{des}^2} \tag{5.10}$$

Note that a Kalman filter (block component 5 in Figure 5.5) is utilised to estimate the configuration of the mobile robot, q_{est}. The input to the Kalman filter is q_{mea}, which is the measurement of q_{act} through position sensors (block component 4 in Figure 5.5). However, due to measurement noise, the unwanted sensor noise should be filtered to get a more accurate estimate of the mobile robot's configuration, in this case q_{est}.

At every time-interval, q_{est} and q_{des} are compared, but rather than directly compute the difference between q_{est} and q_{des}, the configuration error, e_q, can be computed as the transformation of q_{est} onto q_{des} in local coordinate system. Therefore, the transformation exercise is the multiplication of the rotational matrix, R, (block component 1 in Figure 5.5) with the difference between q_{est} and q_{des}, given by equation (5.11), as mentioned in Kanayama et al. (1990).

$$e_q = R \times (q_{des} - q_{est}) \tag{5.11}$$

Configuration error, e_q, the required velocities to drive the mobile robot to the next point are calculated with consideration of the desired velocities (υ_{des} and ω_{des}) as well as e_q. This operation is shown in component 2 in Figure 5.5 with a goal to drive the mobile robot closer to the desired trajectory at every time-interval. Then, the configuration of the mobile robot is determined using multiplication of the calculated velocities and matrix coefficient of the kinematic model ($\dot{q} = G(q)\,u$), where u is the input transformation related to the velocities calculated in component 2 and $G(q)$ is the matrix coefficient of kinematic model in component 3. In the same component block, integral operation of $\dot{q}$ results in the actual configuration of mobile robot, q_{act}. However, rather than feeding back q_{act} and comparing to the setpoint, q_{des}, it is more convenient to improve the method by transforming q_{act} into an estimate of the parameters by considering Kalman filter block in feedback line.

Algorithm 5.1 describes the method of utilising the Kalman filter in mobile robot trajectory tracking error control. Firstly, it is necessary to initialise υ_{des}, ω_{des}, q_{est}, state vector, $\hat{X}(0|0)$, and covariance matrix, $P(0|0)$, of the Kalman filter. Then, measurement of mobile robot's position is observed at every iteration (k denotes the iteration index of discrete time-interval) before executing the Kalman filter algorithm. In this algorithm, q_{est} at every time-interval is calculated via prediction and estimation operation, then the configuration error is computed using the multiplication of rotation matrix, R, and the difference of q_{est} and q_{des}. The quantity of error at

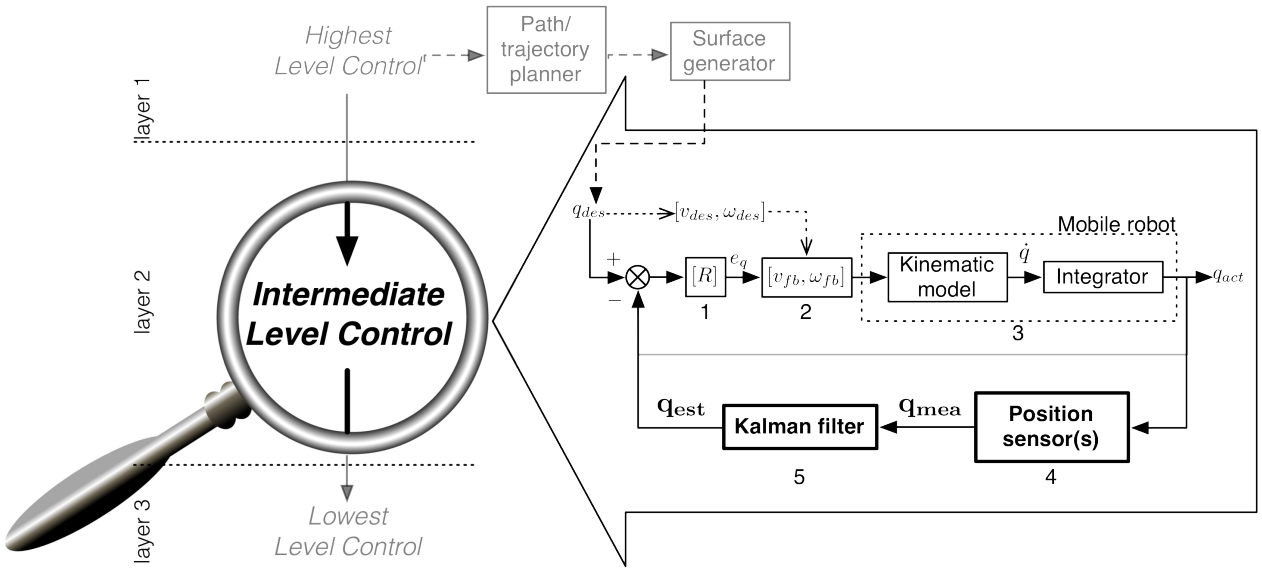


Figure 5.5: The flowchart on the left-hand-side is the mobile robot control architecture with the magnifier indicating that the intermediate level control layer is the focus of this chapter. Meanwhile, the block diagram inside the big arrow on the right-hand-side explains the control operations embedded in the intermediate level control layer. In the control block diagram, the grey line is the feedback line connected to control input as proposed by Kanayama et al. (1990, 1991) and Sun (2005). However, in the events of disturbance, Kalman filter is needed to filter noise. Dotted black/grey arrow lines with closed arrow head (- -▶) denote separate processes outside intermediate level control layer to define q_{des} while dotted arrow lines with opened arrow head (- ->) denote the application of desired velocities into equations of input transformation. Each component in control block diagram is numbered and explained accordingly in Section 5.3.

each time-interval is very important to determine the parameter of v_{fb} and ω_{fb}. Meanwhile, the control inputs denoted by u_1 and u_2, are defined via the control gains of approximate linearisation of error dynamics. The loop continues until the last iteration number and as a result, the actual trajectory will converge to the desired trajectory. The parameters of u_1, u_2, v_{fb} and ω_{fb} are utilised from Siciliano et al. (2009).

5.4 Derivation of Trajectory Tracking Error Model

The assumptions made in this chapter are summarised in Table 5.1. The trajectory tracking error model encompasses operations in block components 1 and 2 as depicted in Figure 5.5. With reference to Figure 5.3, equation (5.11) can be written in vector matrix form as shown in equation (5.12) (Kanayama et al., 1990; Siciliano et al., 2009).

$$\begin{bmatrix} e_x \\ e_y \\ e_\psi \end{bmatrix} = \begin{bmatrix} \cos\psi_{est} & \sin\psi_{est} & 0 \\ -\sin\psi_{est} & \cos\psi_{est} & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x_{des} - x_{est} \\ y_{des} - y_{est} \\ \psi_{des} - \psi_{est} \end{bmatrix} \tag{5.12}$$

Algorithm 5.1 Algorithm for trajectory tracking error control via approximation linearisation with Kalman filter estimation.

$k = 1$
  Initialise $q_{est}\,(k)$
  Initialise v_{des} and ω_{des}
  Initialise Kalman filter: $(\hat{X}\,(0|0))$, $(P\,(0|0))$
for $k \longrightarrow N$
  Position measurement, $q_{mea}\,(k)$
  $\Rightarrow$Go to Kalman filter algorithm in Algorithm 5.2.
    q_{mea} --$\rightarrow$Prediction, $\hat{X}\,(k+1|k)$, $\hat{Z}\,(k+1)$
    q_{est} $\leftarrow$--Estimation, $\hat{X}\,(k+1|k+1)$, $P\,(k+1|k+1)$
  Compute configuration error, $e_q = R \times (q_{des} - q_{est})$
  Approximate linearisation feedback of u_1 and u_2
  Input transformation, $v_{fb}\,(k)$ and $\omega_{fb}\,(k)$
  $\dot{q}_{act}\,(k+1) = G\,(q) \times \left[\begin{array}{cc} v_{fb}\,(k) & \omega_{fb}\,(k) \end{array}\right]^T$
  $q_{act}\,(k+1) = \int (\dot{q}_{act}(k+1))$
end

note : k is iteration index, N is numbers of iteration.

Based on the description in Sections 5.2 and 5.3, equation (5.12) only applies to a flat surface. Meanwhile, the equation of e_q for uneven surface can be derived by substituting equations (5.4), (5.8) and (5.6) into (5.11) and multiplying by the six-degree-of-freedom rotation matrix, R_{6dof}.

$$
\begin{aligned}
e_q &= R_{6dof} \times (q_{des} - q_{est}) \\
\begin{bmatrix} e_x \\ e_y \\ e_z \\ e_\phi \\ e_\theta \\ e_\phi \end{bmatrix} &= \begin{bmatrix} \cos\psi_{est}\cos\theta_{est} & A & C & 0 & 0 & 0 \\ -\sin\psi_{est}\cos\theta_{est} & B & D & 0 & 0 & 0 \\ -\sin\theta_{est} & -\cos\theta_{est}\sin\phi_{est} & \cos\theta_{est}\cos\phi_{est} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \\
&\times \begin{bmatrix} x_{des} - x_{est} \\ y_{des} - y_{est} \\ z_{des} - z_{est} \\ \phi_{des} - \phi_{est} \\ \theta_{des} - \theta_{est} \\ \psi_{des} - \psi_{est} \end{bmatrix}
\end{aligned}
\tag{5.13}
$$

with R_{6dof} given by the Euler angle rotation formulation multiplication[2]. Notation of A, B, C and D in R_{6dof} in given by,

$$A = \sin\psi_{est}\cos\phi_{est} - \cos\psi_{est}\sin\theta_{est}\sin\phi_{est}$$

$$B = \cos\psi_{est}\cos\phi_{est} + \sin\psi_{est}\sin\theta_{est}\sin\phi_{est}$$

[2]The R_{6dof} derivation is given by the multiplication of the Euler angles $R\,(\psi)$, $R\,(\theta)$ and $R\,(\phi)$ as explained in Siciliano et al. (2009).

Table 5.1: List of assumptions for trajectory tracking.

No.	Assumption
1	A simple non-holonomic mobile robot is used.
2	No slippage of wheels.
3	A pre-defined surface model is generated offline.
4	Surface model is an uneven type but highly negotiable.
5	Surface is obstacle free, so that mobile robot can roam freely on the surface.
6	Trajectory tracking model derived with reference of local coordinate (Kanayama et al., 1991).

$C = \sin\psi_{est}\sin\phi_{est} + \cos\psi_{est}\sin\theta_{est}\cos\phi_{est}$

$D = \cos\psi_{est}\sin\phi_{est} - \sin\psi_{est}\sin\theta_{est}\cos\phi_{est}$

For any ground vehicle, changes in the Cartesian coordinate, in particular Z-axis direction, are strictly bounded to surface's contour (variation of height). Mobile robot's motion associated with Z-axis direction are climbing and rolling motions. This means, parameters of e_z, e_θ and e_ϕ in equation (5.13) are strictly non-controllable. Therefore, all the error terms related to z, ϕ and θ can be cancelled. As a result, equation (5.13) is reduced to the error configuration of a flat surface equivalent (equation (5.12)) and valid for an uneven surface (θ and ϕ bounded) provided a simple non-holonomic mobile robot is used. Mobile robot with "terrain negotiable" capabilities such as Variable Geometry Tracked Vehicle is not considered in this solution.

The error configuration, e_q, employs q_{est}, which is determined from measurement of q_{act} and resulting q_{mea}. This recursive computation is undertaken using a Kalman filter before further derivation of trajectory tracking error model.

5.4.1 Kalman Filter to estimate mobile robot configuration, q_{est}

Kalman filter (Welch and Bishop, 1995), is a recursive algorithm and very useful to solve many tracking problems (Blackman and House, 1999). A standard Kalman filter consists of a plant model with state vector, $\underline{X}(k+1)$, (equation (5.14)) and an observation model with observer state vector, $\underline{Z}(k)$, (equation (5.15)). For a general position tracking problem, the plant model defines the Cartesian position of the system at each time-interval in response to control input (i.e., velocity or acceleration) and measurement noise. Meanwhile, the observation model describes a position measurement via a sensor with certain quantity of measurement noise. In trajectory tracking problem, state vector in equation (5.14) and observer state vector in equation (5.15) is written with the reference mobile robot configuration and sensor observation respectively.

$$\underline{X}(k+1) = A(k)\,\underline{X}(k) + B(k)\,\underline{u}(k) + \Gamma(k)\,\underline{\varepsilon}(k) \tag{5.14}$$

$$Z(k) = C(k)\,\underline{X}(k) + \underline{\eta}(k) \tag{5.15}$$

where

$\varepsilon(k) \sim N(0, Q(k))$, and

$\underline{\eta}(k) \sim N(0, R(k))$

are uncertainties in probability density function (pdf).

Additionally, the interest of the control solution as discussed in Section 5.3 is to achieve convergence to the desired trajectory with the basis of two conditions: (1) mobile robot cruises at constant linear velocity; and (2) control input vector, $\underline{u}(k)$, is absent. The first condition is made with the basis that the mobile robot travels at an adequate speed to deal with terrain variation at every time-step along the trajectory. As a result, the acceleration or rate of velocity changes is disregarded. Meanwhile, the second condition is made with the basis of the solution for tracking problem is not necessarily required external control input resulting in, $\underline{u}(k) = 0$. Therefore, equation (5.14) can be rewritten as

$$\underline{X}(k+1) = A(k)\underline{X}(k) + \Gamma(k)\underline{\varepsilon}(k) \tag{5.16}$$

Consider the same conditions, and with reference to Figures 5.3, 5.5 and Algorithm 5.1, mobile robot configuration and velocities (linear and angular) are described by state vector in equations (5.17) at discrete time-interval k,

$$\underline{X}(k) = \begin{bmatrix} P_x(k) \\ V_x(k) \\ P_y(k) \\ V_y(k) \\ \psi(k) \\ \omega(k) \end{bmatrix} \tag{5.17}$$

$$\underline{X}(k+1) = \begin{bmatrix} P_x(k+1) \\ V_x(k+1) \\ P_y(k+1) \\ V_y(k+1) \\ \psi(k+1) \\ \omega(k+1) \end{bmatrix} \tag{5.18}$$

and observation state vector

$$\underline{Z}(k) = \begin{bmatrix} Z_x(k) \\ Z_y(k) \\ Z_\psi(k) \end{bmatrix} \tag{5.19}$$

The state comprises the Cartesian position and heading angle relative to initial configuration (start point). For a constant velocity system, we can write the changes of velocity with respect to time, in the direction of driving, lateral and turning as

$$\begin{aligned} \dot{V}_x(k) &= 0 \\ \dot{V}_y(k) &= 0 \\ \dot{\omega}(k) &= 0 \end{aligned}$$

However, a more "natural" approach especially in real environment, is to assume some slight variation of the velocities. Therefore, rate of velocity change at every time-interval can be expressed as

$$\dot{V}_x(k) = \varepsilon(k) \tag{5.20}$$

$$\dot{V}_y(k) = \varepsilon(k) \tag{5.21}$$

$$\dot{\omega}(k) = \varepsilon(k) \tag{5.22}$$

where $\underline{\varepsilon}$ is uncertainty of state, model uncertainty given by $\underline{\varepsilon}(k) \sim N(0, Q(k))$.

Additionally, at each constant time-step, Δ, velocities of every direction are given by adding the velocity's error to the previous velocity, leading to

$$V_x(k+1) = V_x(k) + \Delta\varepsilon(k) \tag{5.23}$$

$$V_y(k+1) = V_y(k) + \Delta\varepsilon(k) \tag{5.24}$$

$$\omega(k+1) = \omega(k) + \Delta\varepsilon(k) \tag{5.25}$$

where a first order Euler approximation has been asssumed.

Meanwhile, Cartesian and angular position at each time-interval are derived considering

velocity between k and $k+1$ is equal to average,

$$P_x(k+1) = P_x(k) + \Delta\left(\frac{V_x(k+1) + V_x(k)}{2}\right) \tag{5.26}$$

$$P_y(k+1) = P_y(k) + \Delta\left(\frac{V_y(k+1) + V_y(k)}{2}\right) \tag{5.27}$$

$$\psi(k+1) = \psi(k) + \Delta\left(\frac{\omega(k+1) + \omega(k)}{2}\right) \tag{5.28}$$

Substituting equations (5.26), (5.27) and (5.28) with equations (5.23), (5.24) and (5.25) respectively results in the Cartesian and angular position at each time-step, Δ, in terms of velocities disturbances,

$$P_x(k+1) = P_x(k) + \Delta V_x(k) + \frac{\Delta^2 \varepsilon(k)}{2} \tag{5.29}$$

$$P_y(k+1) = P_y(k) + \Delta V_y(k) + \frac{\Delta^2 \varepsilon(k)}{2} \tag{5.30}$$

$$\psi(k+1) = \psi(k) + \Delta\omega(k) + \frac{\Delta^2 \varepsilon(k)}{2} \tag{5.31}$$

Rearranging equations (5.29), (5.30), (5.31), (5.23), (5.24) and (5.25) with consideration of equations (5.15), (5.16), (5.17), (5.18) and (5.19), the plant and observation model (equations (5.32) and (5.33)) can be rewritten as

$$\underbrace{\begin{bmatrix} P_x(k+1) \\ V_x(k+1) \\ P_y(k+1) \\ V_y(k+1) \\ \psi(k+1) \\ \omega(k+1) \end{bmatrix}}_{\underline{X}(k+1)} = \underbrace{\begin{bmatrix} 1 & \Delta & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{A(k)} \underbrace{\begin{bmatrix} P_x(k) \\ V_x(k) \\ P_y(k) \\ V_y(k) \\ \psi(k) \\ \omega(k) \end{bmatrix}}_{\underline{X}(k)} + \underbrace{\begin{bmatrix} \frac{\Delta^2}{2} \\ \Delta \\ \frac{\Delta^2}{2} \\ \Delta \\ \frac{\Delta^2}{2} \\ \Delta \end{bmatrix}}_{\Gamma} \underline{\varepsilon}(k) \tag{5.32}$$

$$\underbrace{\begin{bmatrix} Z_x(k) \\ Z_y(k) \\ Z_\psi(k) \end{bmatrix}}_{Z(k)} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}}_{C(k)} \underbrace{\begin{bmatrix} P_x(k) \\ V_x(k) \\ P_y(k) \\ V_y(k) \\ \psi(k) \\ \omega(k) \end{bmatrix}}_{\underline{X}(k)} + \underbrace{\begin{bmatrix} \eta_x(k) \\ \eta_y(k) \\ \eta_\psi(k) \end{bmatrix}}_{\eta(k)} \tag{5.33}$$

where

$A(k)$ is dynamics relation of $\underline{X}(k+1)$ to $\underline{X}(k)$,

$C(k)$ is matrix coefficient of state $\underline{X}(k)$ related to outputs,

$\Gamma(k)$ is matrix coefficient of model uncertainty, $\underline{\varepsilon}(k)$,

$\underline{\varepsilon}(k)$ is uncertainty of state, model uncertainty, and

$\underline{\eta}(k)$ is measurement uncertainty given by $\underline{\eta}(k) \sim N(0, R(k))$.

The next step is to utilise the Kalman filter algorithm (Algorithm 5.2) with the aim to get robot position estimation for every time-interval while "smooth"ing the actual trajectory. After initialising the state and state covariance, a cyclic process of predict next state and covariance, compute gain and estimate occurs. The filter is very robust, therefore the ratio between plant model uncertainty covariance matrix, $Q(k)$, and observation model uncertainty covariance matrix, $R(k)$, can be accomplished through "trial and error" method. Covariance matrices for Kalman filter are shown by equations (5.34), (5.35) and (5.36).

$$P(k) = |0.1 \times \text{PRNG}| \times \mathbb{I}_6 \tag{5.34}$$

$$Q(k) = 9555 \times |\mathbb{I}_6 \times \text{PRNG}| \tag{5.35}$$

$$R(k) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.36}$$

where

$P(k)$ is state covariance matrix,

$Q(k)$ is plant model uncertainty,

PRNG is normally distributed pseudorandom number generator,

$\mathbb{I}_6$ is 6×6 identity matrix,

$R(k)$ is observation model uncertainty.

As a result of the Kalman filter algorithm, the estimated configuration, q_{est}, of a mobile robot is given by equation (5.37). At each time-interval, q_{est} is computed and fed to the tracking controller in order to get "smooth" estimate trajectory over a complete simulation time with respect to the sensor observation.

$$\begin{bmatrix} x_{est} \\ y_{est} \\ \psi_{est} \end{bmatrix} = \begin{bmatrix} P_x \\ P_y \\ \psi \end{bmatrix} \tag{5.37}$$

Algorithm 5.2 Kalman Filter Algorithm as described in Welch and Bishop (1995).

for $k \longrightarrow N$

Start

Initial estimates $\hat{X}(0|0), P(0|0)$

Predict

$\hat{X}(k+1|k) = A(k)\hat{X}(k|k)$

$P(k+1|k) = A(k)P(k|k)A^T(k) + \Gamma(k)Q(k)\Gamma^T(k)$

$\hat{Z}(k+1) = C(k)\hat{X}(k+1|k)$

$S(k+1) = C(k)P(k+1|k)C^T(k) + R(k+1)$

Gain

$G(k+1) = P(k+1|k)C^T(k)S^{-1}(k+1)$

Estimate

$\hat{X}(k+1|k+1) = \hat{X}(k+1|k) + G(k+1)\left[Z(k+1) - \hat{Z}(k+1)\right]$

$P(k+1|k+1) = P(k+1|k) - G(k+1)S(k+1)G^T(k+1)$

end

5.4.2 Error Dynamics

In Section 5.4, it has been described that the error configuration in equation (5.12) is valid for uneven surface because the mobile robot's motion is bounded to the height of the surface with consideration of assumptions depicted in Table 5.1. Therefore, as stated in Siciliano et al. (2009) the error dynamics, $\dot{e}_q$, of trajectory tracking problem are given by

$$\dot{e}_q = \begin{bmatrix} 0 & \omega_{des} & 0 \\ -\omega_{des} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} e + \begin{bmatrix} 0 \\ \sin e_\psi \\ 0 \end{bmatrix} v_{des} + \begin{bmatrix} 1 & -e_y \\ 0 & e_x \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{5.38}$$

where u_1 and u_2 are the control inputs. The input transformation (block component 2 in Figure 5.5) shown in equation (5.39) and (5.40) depends on e_q as well as u_1 and u_2

$$v_{fb} = v_{des}\cos e_\psi - u_1 \tag{5.39}$$

$$\omega_{fb} = \omega_{des} - u_2 \tag{5.40}$$

Input transformation, v_{fb} and ω_{fb}, are set of appropriate velocities to drive the mobile robot while converging to the desired trajectory.

5.4.3 Control Based Approximate Linearisation

Approximate linearisation as stated in Siciliano et al. (2009) is the simplest yet most efficient approach to solve the nonlinearity in the error dynamics in equation (5.38). Inspecting the terms in the error dynamics of equation (5.38), there is a demand to solve the nonlinearity in the error dynamics due to presence of a trigonometric term, "$\sin e_\psi$", e_x and e_y. As stated in Siciliano

et al. (2009), by eliminating "$\sin e_\psi$" in equation, the linearised closed-loop error dynamics can be written as

$$\dot{e}_q = \begin{bmatrix} 0 & \omega_{des} & 0 \\ -\omega_{des} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} e + \begin{bmatrix} 0 \\ e_\psi \\ 0 \end{bmatrix} \upsilon_{des} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{5.41}$$

The closed-loop error dynamics in equation (5.41) is still time-varying. Therefore, the linear feedback with sign function, u_1 and u_2, are derived in equations (5.42) and (5.43) as suggested by Siciliano et al. (2009)

$$u_1 = -k_x e_x \tag{5.42}$$

$$u_2 = -k_y \cdot \text{sign}\left(\upsilon_{des}\right) e_y - k_\psi e_\psi \tag{5.43}$$

where, the feedback gains are defined by

$$\begin{aligned} k_x &= 2\zeta\sqrt{\omega_{des}^2 + b\upsilon_{des}^2} \\ k_y &= b\left|\upsilon_{des}\right| \\ k_\psi &= k_x \end{aligned}$$

Note that the input transformation is based on parameters u_1 and u_2 in equations (5.39) and (5.40). The actual configuration, q_{act}, for specific time-interval is the result from integral operation of multiplication of kinematic model, $G\left(q\right)$, and input tranformation whilst the estimated configuration, q_{est} , is the result of Kalman filter operation. Both q_{act} and q_{est} is summarised by equations (5.44) and (5.45). The arrow in equation (5.45) indicates the measurement of q_{act}.

$$q_{act} = \int \left(G\left(q\right) \times \begin{bmatrix} \upsilon_{fb} & \omega_{fb} \end{bmatrix}^T \right) \tag{5.44}$$

$$q_{est} = \text{Kalman Filter} \left\{ q_{mea} \overset{sensor(s)}{\Longleftarrow} q_{act} \right\} \tag{5.45}$$

In this section, a comprehensive and systematic approach to derive the trajectory tracking model for a mobile robot systems that is required to travel on an uneven surface has been discussed. Describing this further, the simulations related to the tracking model is provided in the next section.

5.5 Simulation Results

In this section, the trajectory tracking controller as described in Section 5.4.3 is implemented on three different pre-planned paths[3] as characterised in Table 5.2. Prior to this, four random generated uneven surfaces are defined by multivariate Gaussian functions as illustrated in Figure 5.7. The objective of the computer simulation is to investigate the convergence to zero of the closed-loop linearised error dynamics of trajectory tracking controller for non-holonomic mobile robot system travel on an uneven surface. The simulation results for each type of simulated pre-planned paths, are split into three sub-simulation environments which are:

1. trajectory tracking of three different pre-planned paths on ***four random generated surfaces*** consisting of:
   (a) noisy measurement data,
   (b) mobile robot position estimate using Kalman filter without tracking controller, and
   (c) mobile robot position estimate using Kalman filter with tracking controller.
2. trajectory tracking of three different pre-planned paths on ***three different initial configurations***, and
3. trajectory tracking of three different pre-planned paths based on ***damping coefficient,*** ζ, value.

For convenient, the results of these sub-simulation environments are summarised in separate tables as follows:

- Sub-simulation 1 (based on four random generated surfaces) is summarised in Table 5.3a,
- Sub-simulation 2 (based on three different initial conditions) is summarised in Table 5.3b, and
- Sub-simulation 2 (based on ζ condition) is summarised in Table 5.3c.

[3]These parametric paths are written in timing-law, s, as explained in Chapter 4

Table 5.2: Simulated two-dimensional pre-planned trajectories used in the simulation environments. x_c and y_c is the centre Cartesian coordinate for circular and figure-eight trajectory in metres. R and ω refer to radius [metres] and angular velocity [radian per unit timing-law] respectively. For spline trajectory, x_i, y_i, x_f and y_f are initial and final position of the trajectory.

(a) Circular and figure-eight trajectory.

Path type	**Figure no.**	**Parametric equation**	x_c	y_c	R	ω
circular	Figure 5.6a	$x_{cir} = x_c + R_{cir} \cdot \cos(\omega_{cir} \cdot s)$ $y_{cir} = y_c + R_{cir} \cdot \sin(\omega_{cir} \cdot s)$	5	5	4	6.2
figure-eight	Figure 5.6b	$x_{f8} = x_c + R_{f8} \cdot \sin(2\omega_{f8} \cdot s)$ $y_{f8} = y_c + R_{f8} \cdot \sin(\omega_{f8} \cdot s)$	5	5	4	$\frac{1}{10}$

(b) Spline trajectory.

Path type	**Figure no.**	**Parametric equation**	x_i	y_i	x_f	y_f
spline	Figure 5.6c	$x_{sp} = \begin{pmatrix} s^3 x_f - (s-1)^3 x_i \\ +\alpha_x s^2 (s-1) \\ +\beta_x s (s-1)^2 \end{pmatrix}$ $y_{sp} = \begin{pmatrix} s^3 y_f - (s-1)^3 y_i \\ +\alpha_y s^2 (s-1) \\ +\beta_y s (s-1)^2 \end{pmatrix}$	0	0	10	10

Table 5.3: List of figures that correspond to each sub-simulation of trajectory tracking.

(a) Sub-simulation 1.

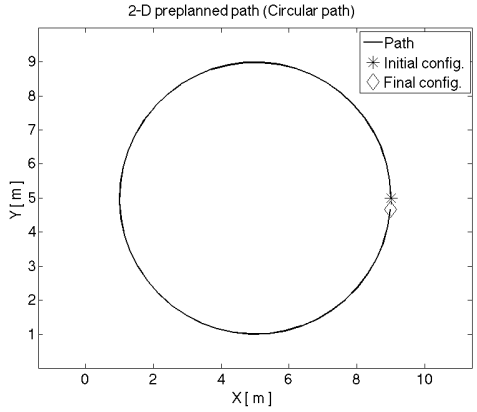
		noisy measurement		**tracking performance** $\zeta = 0.7$	
		3D view	**2D view**	**3D view**	**2D view**
Surface 1	**circular**	Figure 5.8a	Figure 5.9a	Figure 5.10a	Figure 5.11a
	figure-eight	Figure 5.12a	Figure 5.13a	Figure 5.14a	Figure 5.15a
	spline	Figure 5.16a	Figure 5.17a	Figure 5.18a	Figure 5.19a
Surface 2	**circular**	Figure 5.8b	Figure 5.9b	Figure 5.10b	Figure 5.11b
	figure-eight	Figure 5.12b	Figure 5.13b	Figure 5.14b	Figure 5.15b
	spline	Figure 5.16b	Figure 5.17b	Figure 5.18b	Figure 5.19b
Surface 3	**circular**	Figure 5.8c	Figure 5.9c	Figure 5.10c	Figure 5.11c
	figure-eight	Figure 5.12c	Figure 5.13c	Figure 5.14c	Figure 5.15c
	spline	Figure 5.16c	Figure 5.17c	Figure 5.18c	Figure 5.19c
Surface 4	**circular**	Figure 5.8d	Figure 5.9d	Figure 5.10d	Figure 5.11d
	figure-eight	Figure 5.12d	Figure 5.13d	Figure 5.14d	Figure 5.15d
	spline	Figure 5.16d	Figure 5.17d	Figure 5.18d	Figure 5.19d

(b) Sub-simulation 2.

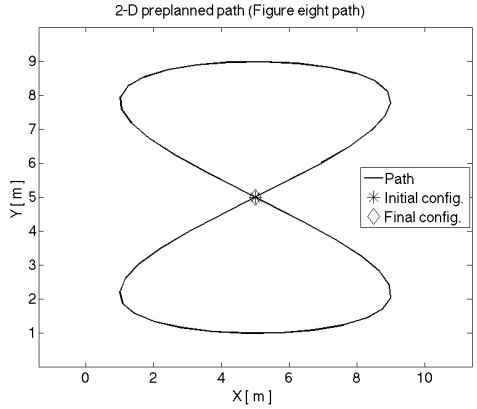
		tracking performance $\zeta = 0.7$	
		3D view	**2D view**
initial condition 1	**circular**	Figure 5.20a	Figure 5.20b
	figure-eight	Figure 5.20c	Figure 5.20d
	spline	Figure 5.20e	Figure 5.20f
initial condition 2	**circular**	Figure 5.21a	Figure 5.21b
	figure-eight	Figure 5.21c	Figure 5.21d
	spline	Figure 5.21e	Figure 5.21f
initial condition 3	**circular**	Figure 5.22a	Figure 5.22b
	figure-eight	Figure 5.22c	Figure 5.22d
	spline	Figure 5.22e	Figure 5.22f

(c) Sub-simulation 3.

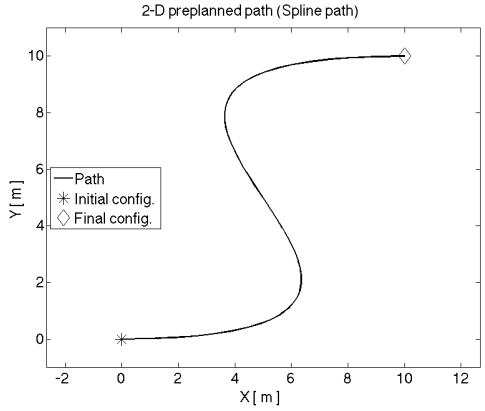
		tracking performance	
		3D view	**2D view**
$0 < \zeta < 1$	**circular**	Figure 5.23a	Figure 5.23b
	figure-eight	Figure 5.23c	Figure 5.23d
	spline	Figure 5.23e	Figure 5.23f
$\zeta = 1$	**circular**	Figure 5.24a	Figure 5.24b
	figure-eight	Figure 5.24c	Figure 5.24d
	spline	Figure 5.24e	Figure 5.24f
$\zeta > 1$	**circular**	Figure 5.25a	Figure 5.25b
	figure-eight	Figure 5.25c	Figure 5.25d
	spline	Figure 5.25e	Figure 5.25f



(a) Circular path.

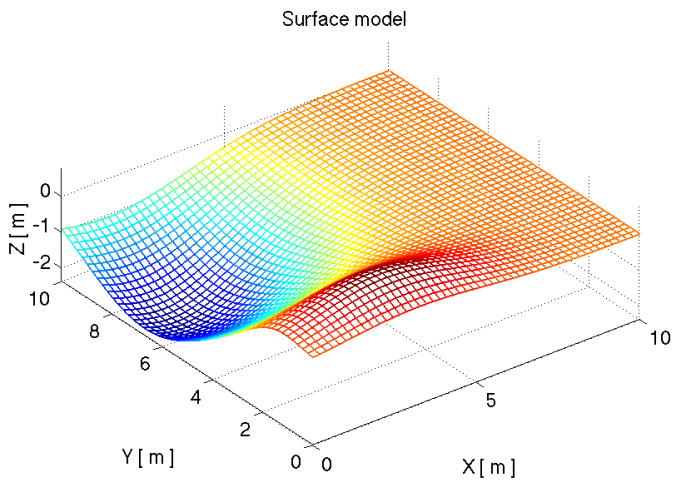


(b) Figure-eight path.

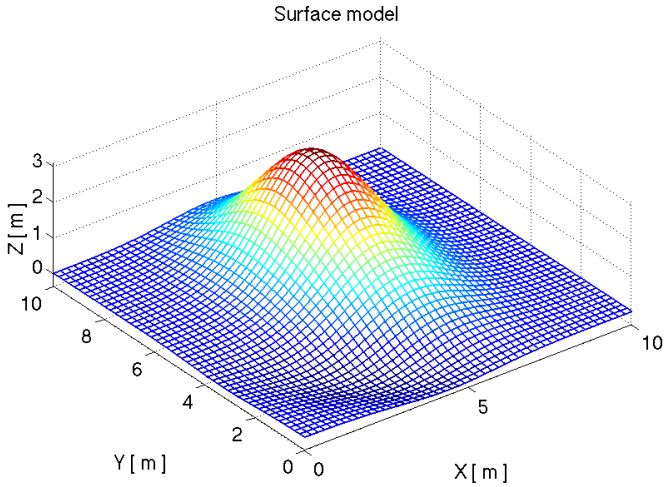


(c) Spline path.

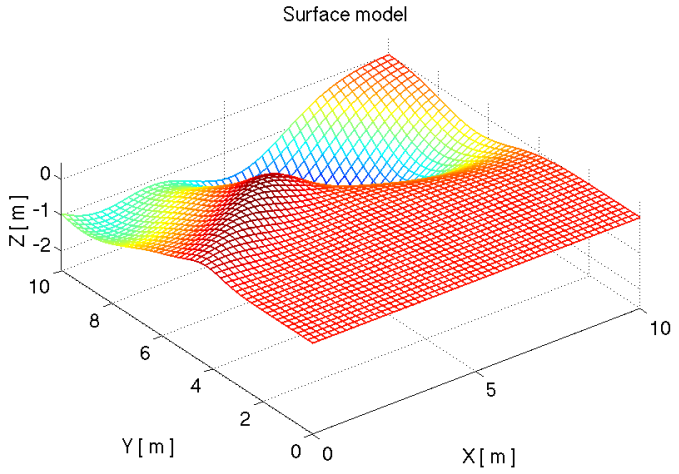
Figure 5.6: Simulated 2-dimensional pre-planned paths defined in path planner.



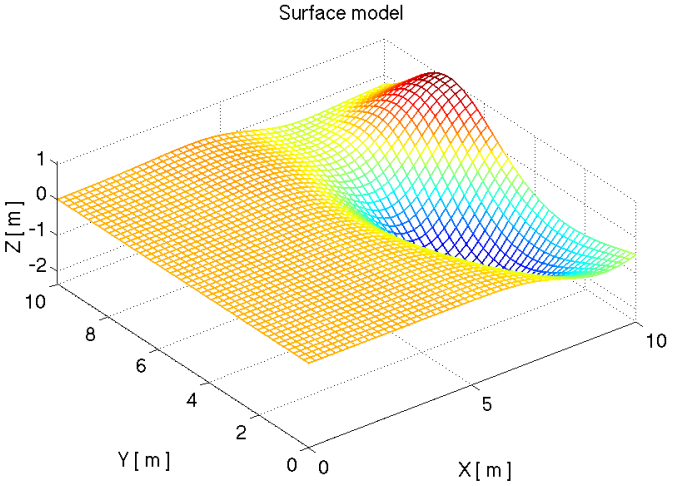
(a) Uneven surface 1.



(b) Uneven surface 2.



(c) Uneven surface 3.



(d) Uneven surface 4.

Figure 5.7: Surface models representation of actual terrains.

Sub-simulation 1 The ζ and natural frequency, b, is set to be constant for all trajectory tracking simulations ($\zeta = 0.7$ and $b = 10$). Figures 5.8, 5.12 and 5.16 show the simulation results of a mobile robot with noisy position measurement (black dot is noisy data) while Figures 5.9, 5.13 and 5.17 show the same results in 2-dimensional viewpoint. These noisy data is filtered using Kalman filter recursive algorithm, resulting mobile robot system estimate its configuration, q_{est}, (dotted-solid line) to track the desired trajectory as plotted in Figures 5.10, 5.14 and 5.18. Improved results (dashed-line) are also plotted on the same figures considering the orientation error between q_{des} and q_{est} and linearised feedback. Again, for convenience, these results are represented in two-dimensional viewpoint in Figures 5.11, 5.15 and 5.19.

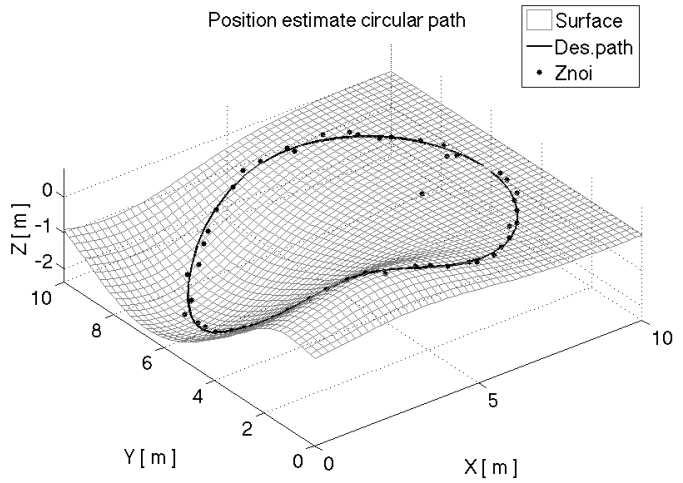
Sub-simulation 2 To assess the performance of the controller, three different initial configurations are used in this sub-simulation. Figures 5.20, 5.21 and 5.22 are the trajectory tracking response on three different initial configurations on a similar random generated surface. Results show that the convergence to the desired trajectory are accomplished regardless of the initial configurations.

Sub-simulation 3 In order to compare the tracking performance of the controller, the last simulation is done using different ζ value which corresponds to each damping coefficient. Figures 5.23, 5.24 and 5.25 are the trajectory tracking response on three different ζ values on a random generated uneven surface. In addition, results in Figure 5.26 show a better comparison of the average error dynamics response for different ζ values where:

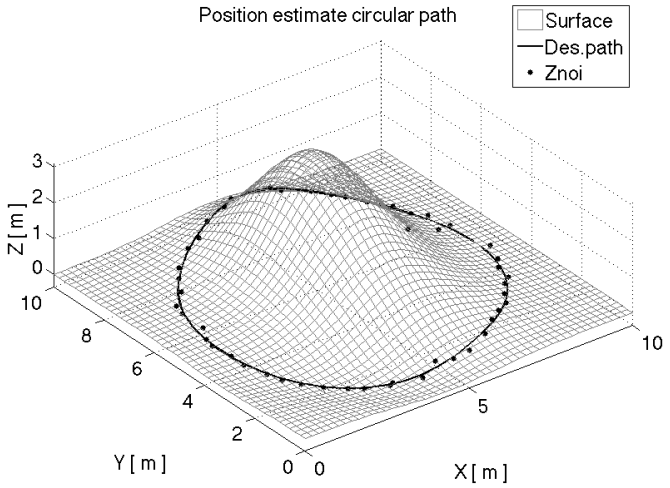
- Figure 5.26a plots the average error dynamics response for tracking in Figure 5.23,
- Figure 5.26b plots the average error dynamics response for tracking in Figure 5.24, and
- Figure 5.26c plots the average error dynamics response for tracking in Figure 5.25.

The results from sub-simulations 1 and 2 show the convergence to zero error is guaranteed for linear time-invariant system with the choice of $\zeta = 0.7$ and $b = 10$. For "poor" initial configurations (Figure 5.20 to 5.22) resulting certain amount of initial configuration errors, the Kalman filter and control law keep pushing the robot to desired trajectory and the tracking performance is better. Although, this is not guaranteed for a time-varying trajectory tracking.

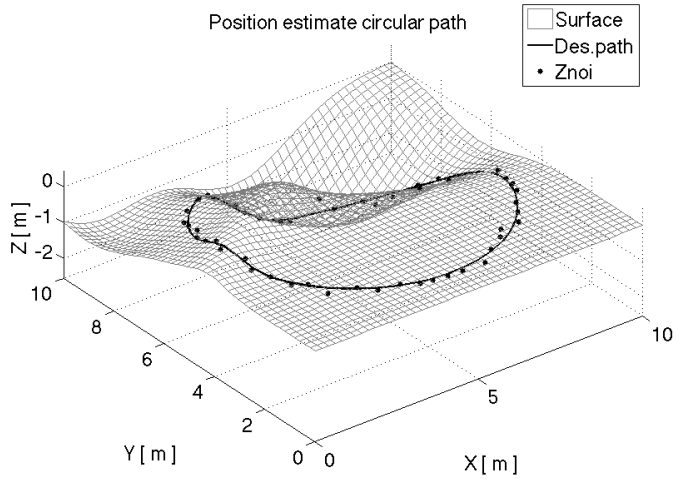
The error dynamics shown in Figure 5.26 are calculated by taking the average of error dynamics response for circular, figure-eight and spline trajectory for each ζ condition. The result illustrates the zero convergence of spline path on comparison based on ζ value. An undamped ζ shows faster zero convergence approximately at sample point, $s \approx 0.2$.



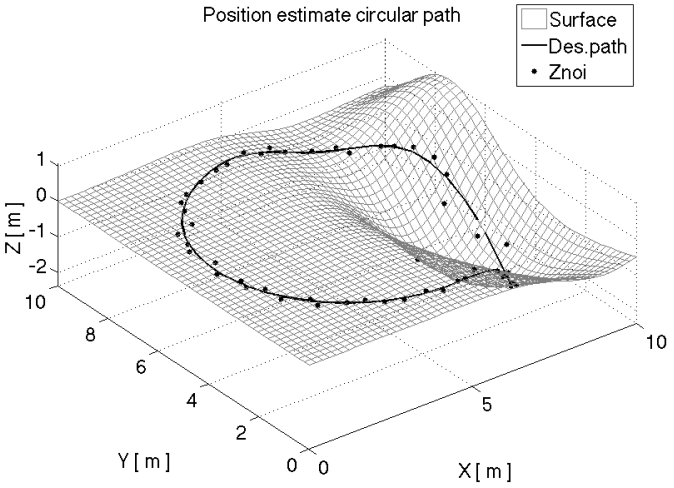
(a) Uneven surface 1.



(b) Uneven surface 2.

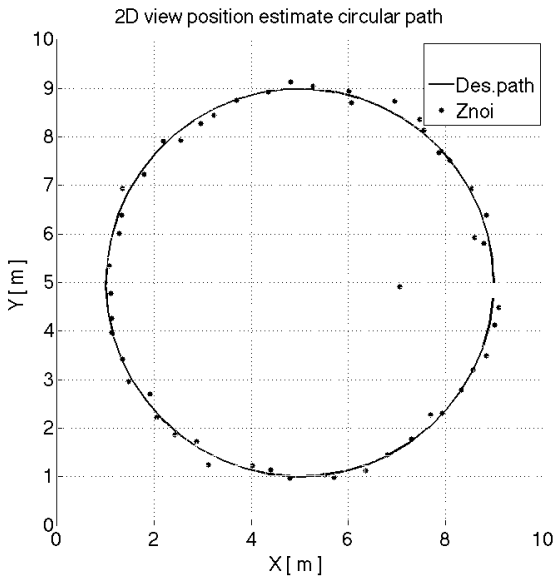


(c) Uneven surface 2.

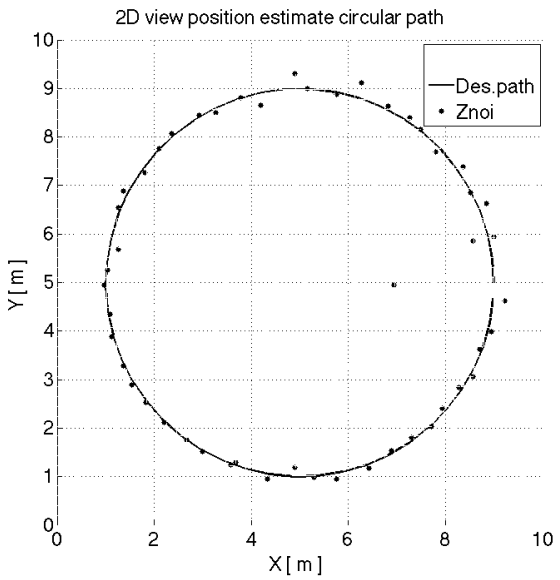


(d) Uneven surface 4.

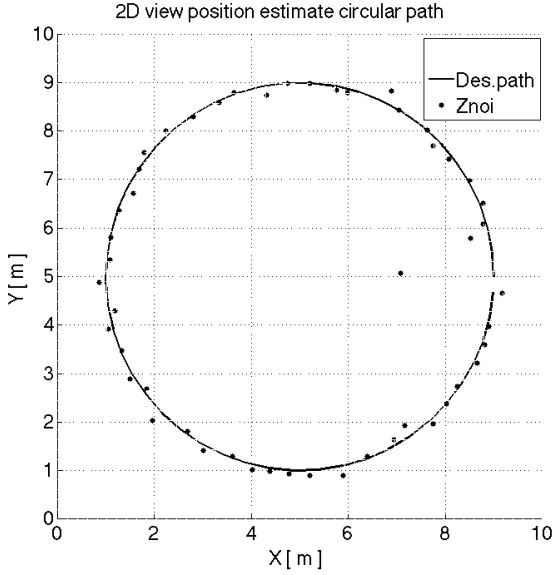
Figure 5.8: Noisy measurement of circular path on random generated uneven surfaces. [Legend: *Surface* is surface, *Des.path* is desired path/trajectory and *Znoi* is measurement noise.]



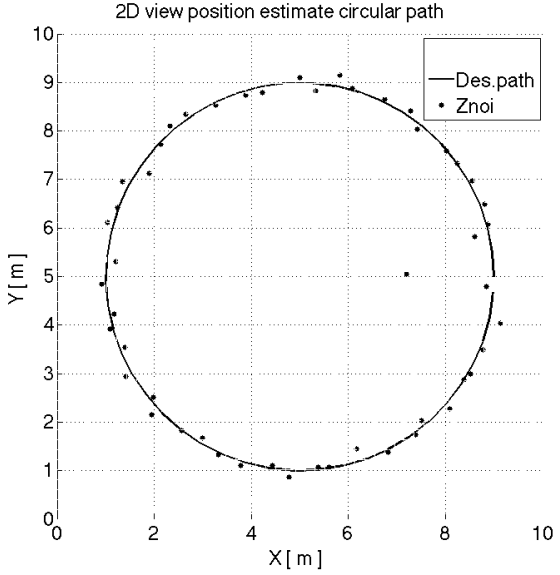
(a) 2D view of uneven surface 1.



(b) 2D view of uneven surface 2.

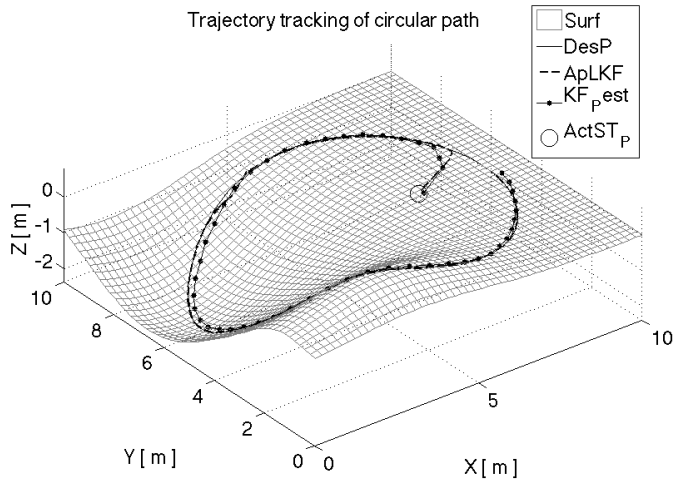


(c) 2D view of uneven surface 1.

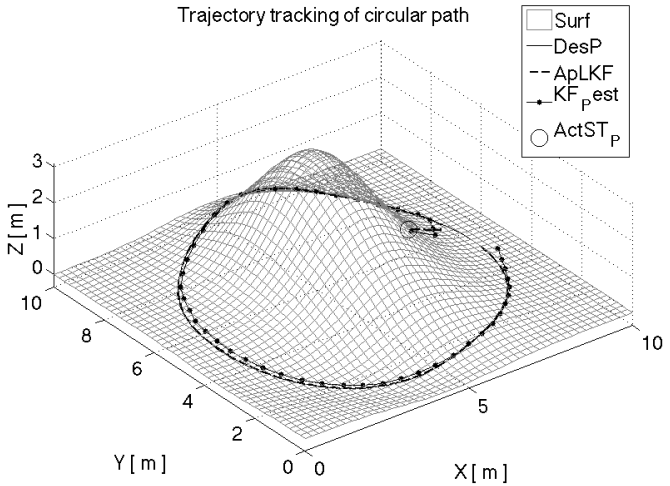


(d) 2D view of uneven surface 1.

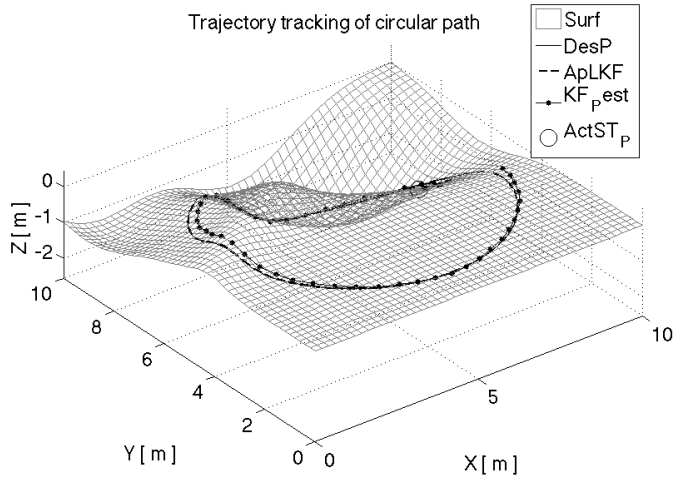
Figure 5.9: 2-dimensional view of Figure 5.8. [Legend: *Des.path* is the desired trajectory/path and *Znoi* is measurement noise.]



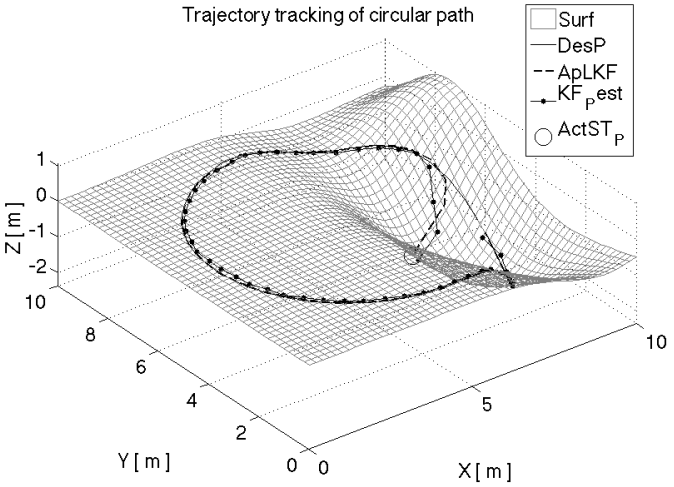
(a) Tracking on uneven surface 1.



(b) Tracking on uneven surface 2.

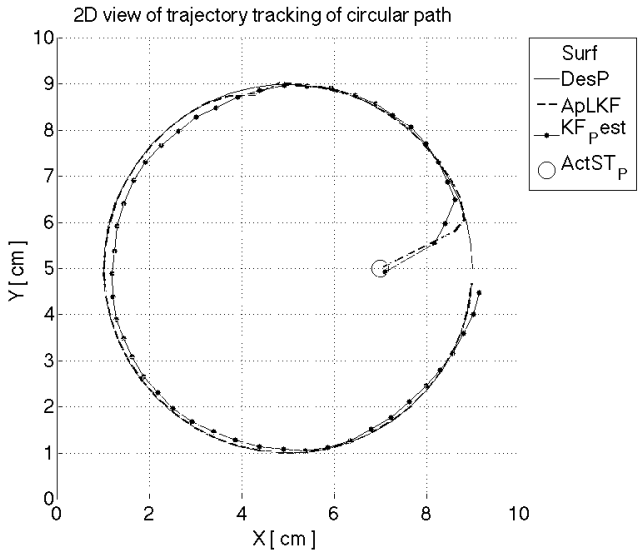


(c) Tracking on uneven surface 3.

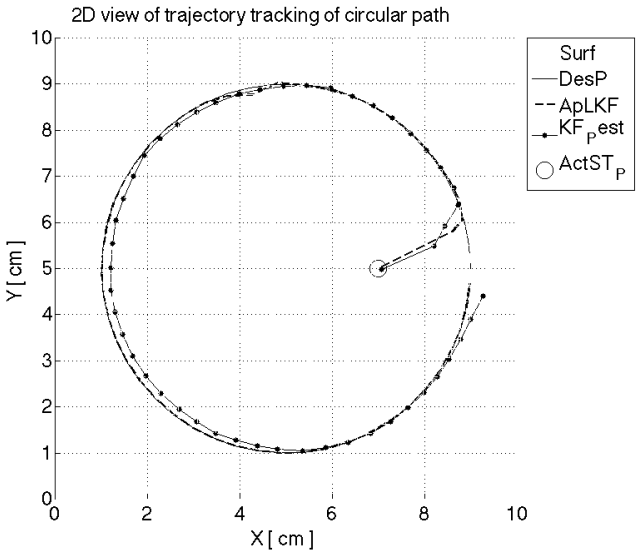


(d) Tracking on uneven surface 4.

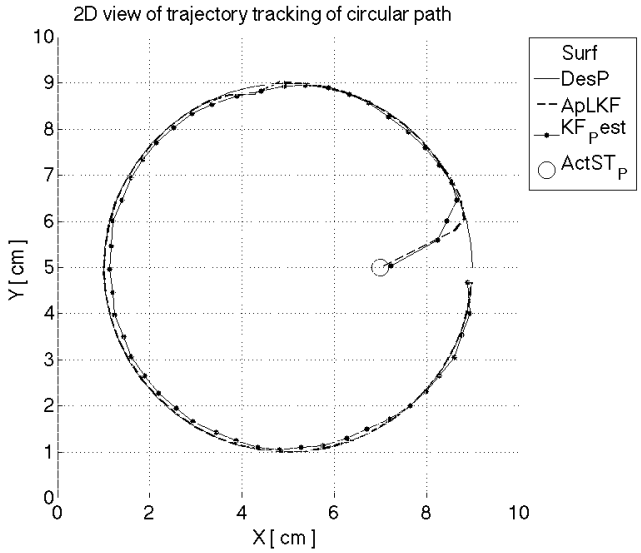
Figure 5.10: Tracking the circular trajectory on uneven surfaces. [Legend: *Surf* is surface, *DesP* is desired path, *ApLKF* is approximate linearisation with Kalman Filter, KF_pest is Kalman filter position estimate and $ActST_p$ is Actual start position.]



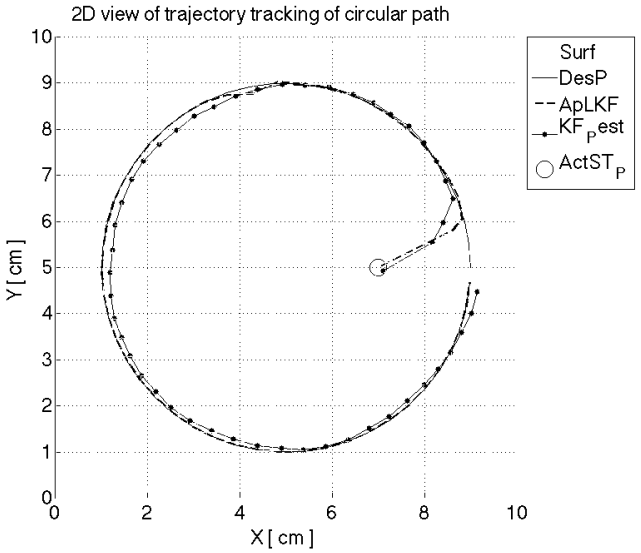
(a) 2D view tracking on uneven surface 1.



(b) 2D view tracking on uneven surface 2.

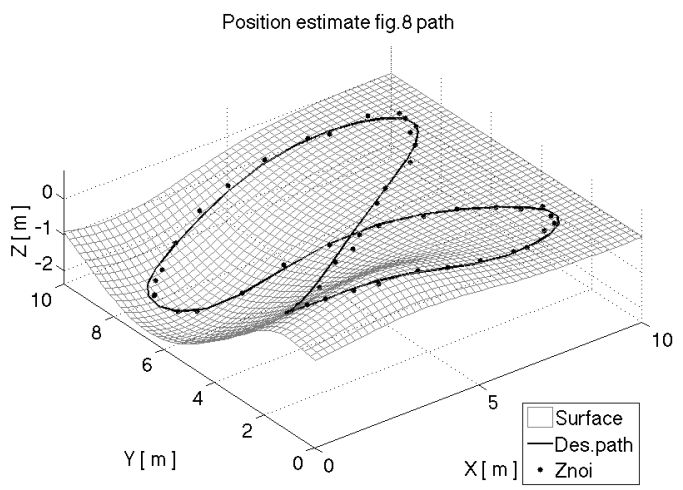


(c) 2D view tracking on uneven surface 3.

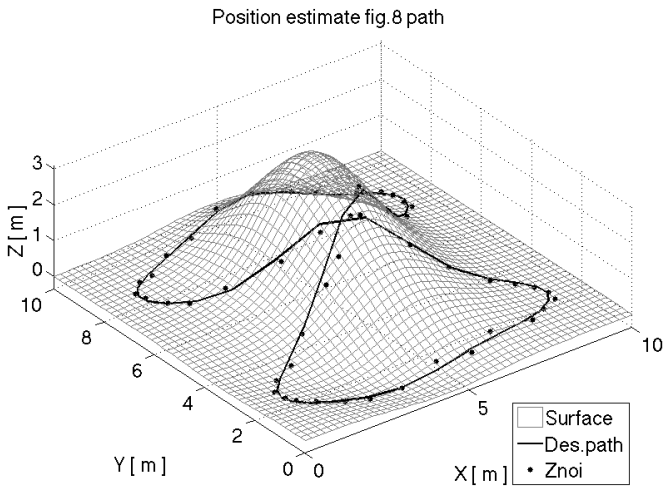


(d) 2D view tracking on uneven surface 4.

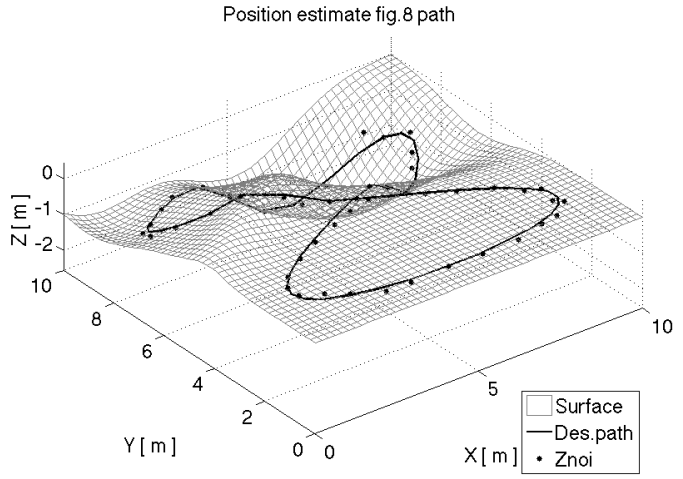
Figure 5.11: 2-dimensional view of Figure 5.10. [Legend: *Surf* is surface, *DesP* is desired path, *ApLKF* is approximate linearisation with Kalman Filter, KF_pest is Kalman filter position estimate and $ActST_p$ is Actual start position.]



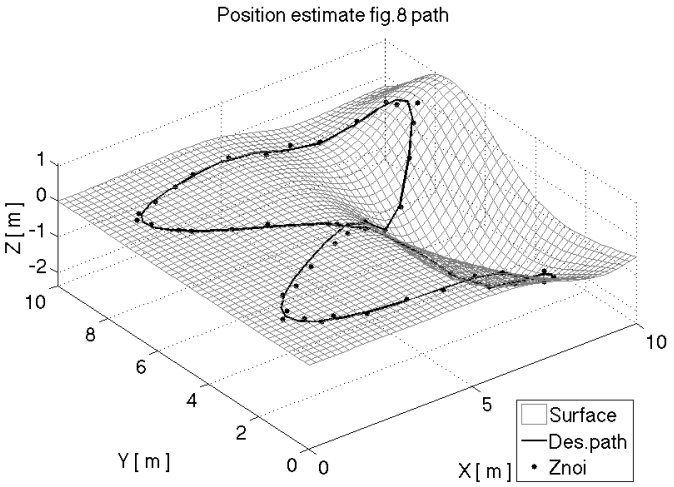
(a) Uneven surface 1.



(b) Uneven surface 2.

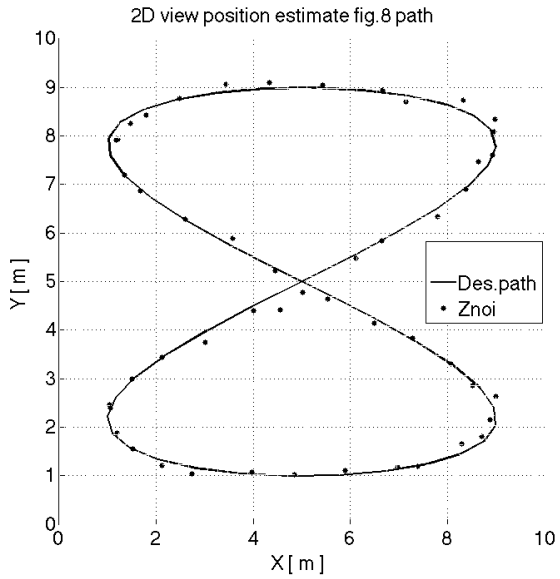


(c) Uneven surface 2.

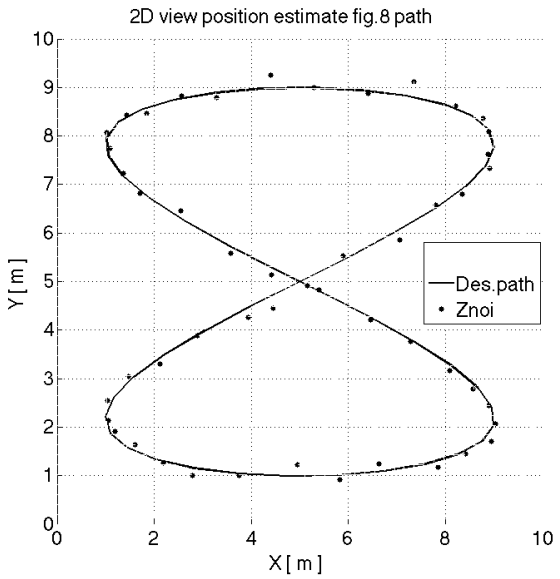


(d) Uneven surface 4.

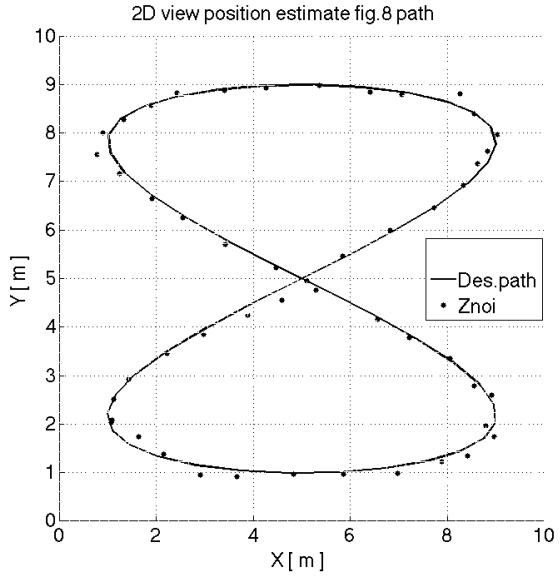
Figure 5.12: Noisy measurement of figure-eight path on uneven surfaces. [Legend: *Surface* is surface, *Des.path* is desired path/trajectory and *Znoi* is measurement noise.]



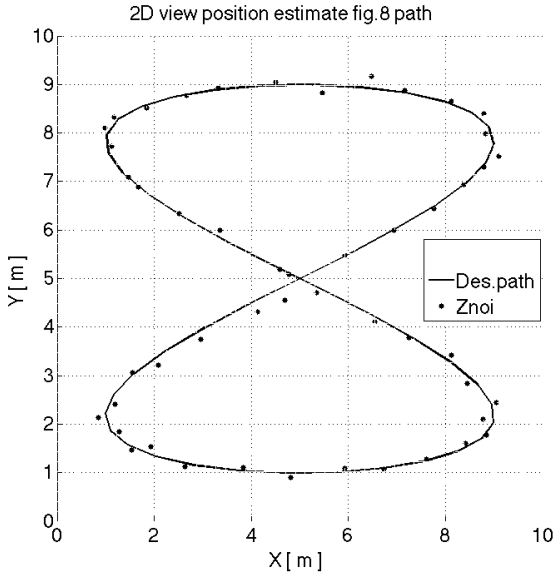
(a) 2D view of uneven surface 1.



(b) 2D view of uneven surface 2.

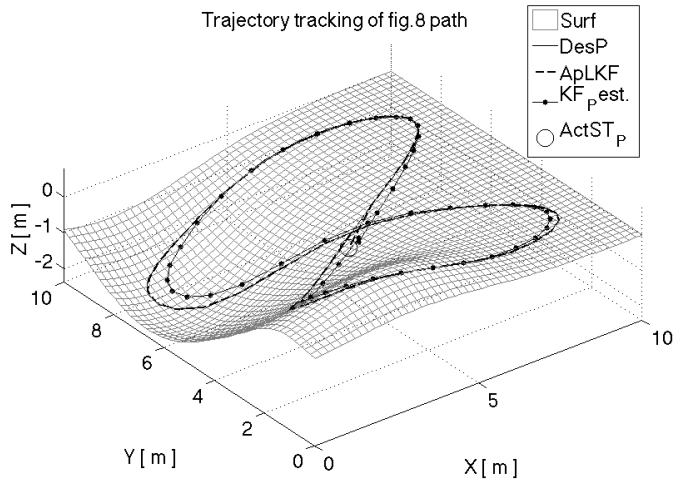


(c) 2D view of uneven surface 1.

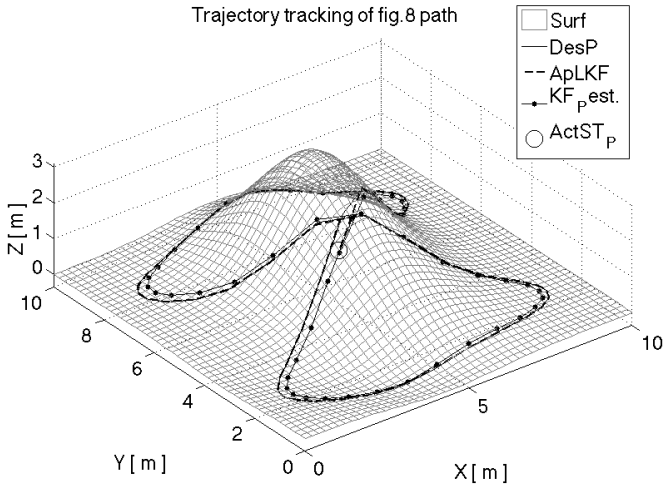


(d) 2D view of uneven surface 1.

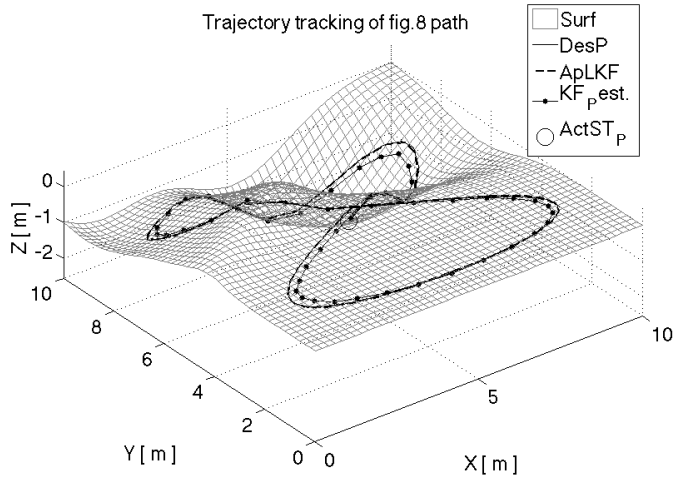
Figure 5.13: 2-dimensional view of Figure 5.12. [Legend: *Des.path* is the desired trajectory/path and *Znoi* is measurement noise.]



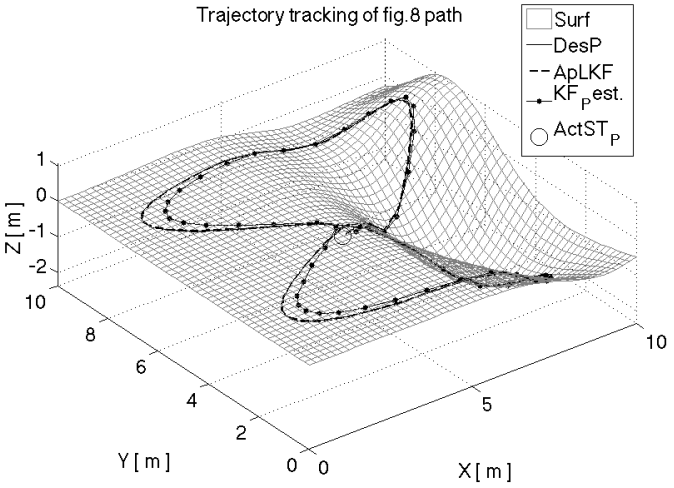
(a) Tracking on uneven surface 1.



(b) Tracking on uneven surface 2.

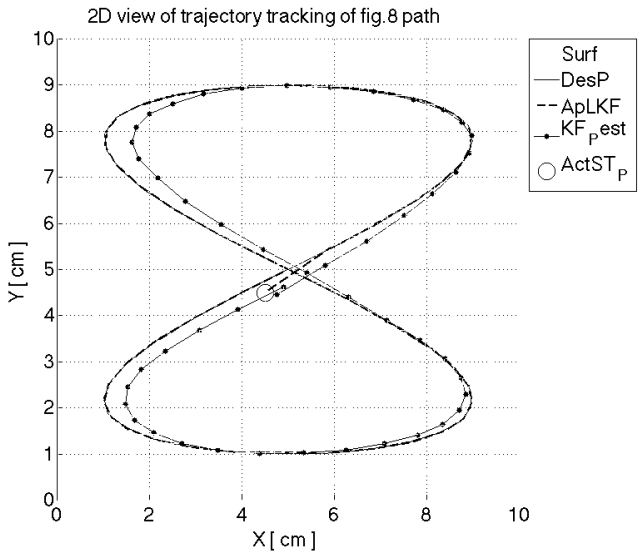


(c) Tracking on uneven surface 3.

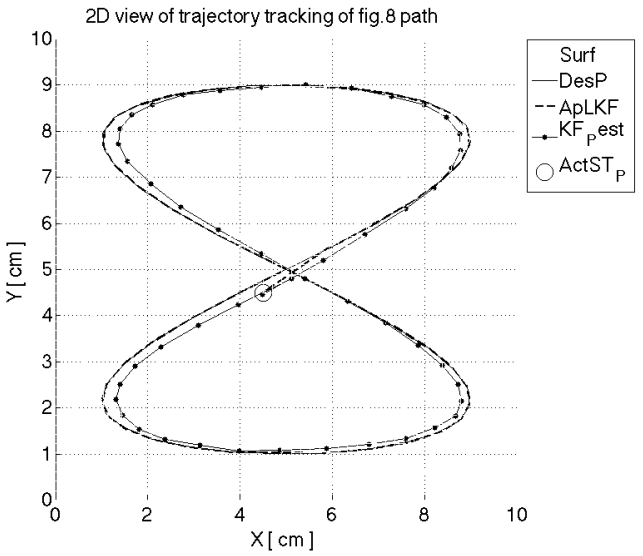


(d) Tracking on uneven surface 4.

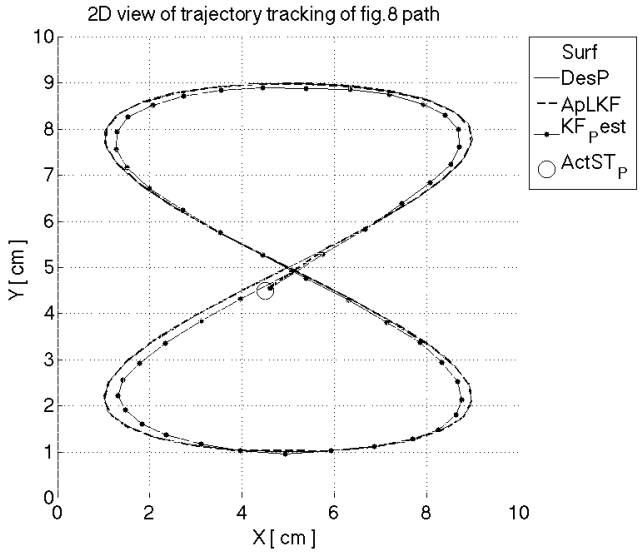
Figure 5.14: Tracking the figure-eight trajectory on uneven surfaces. [Legend: *Surf* is surface, *DesP* is desired path, *ApLKF* is approximate linearisation with Kalman Filter, $KF_p est$ is Kalman filter position estimate and $ActST_p$ is Actual start position.]



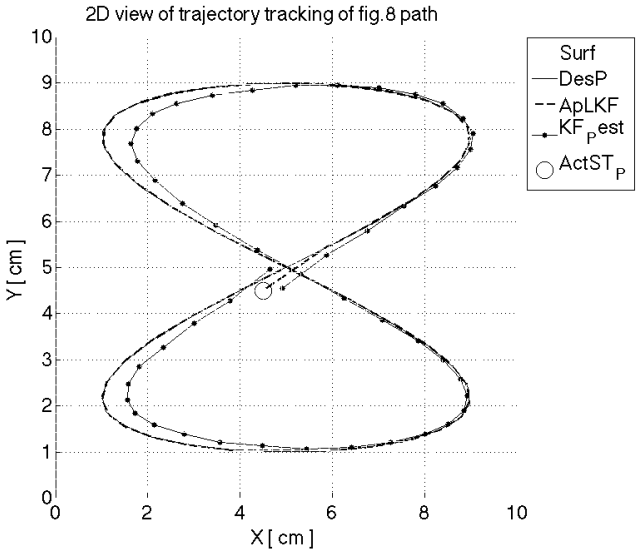
(a) 2D view tracking on uneven surface 1.



(b) 2D view tracking on uneven surface 2.

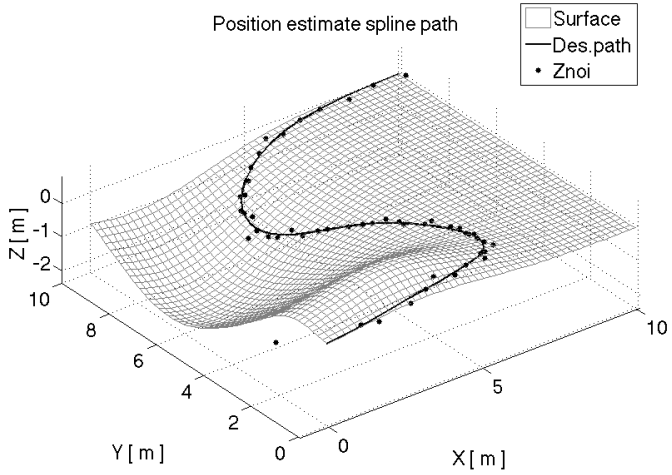


(c) 2D view tracking on uneven surface 3.

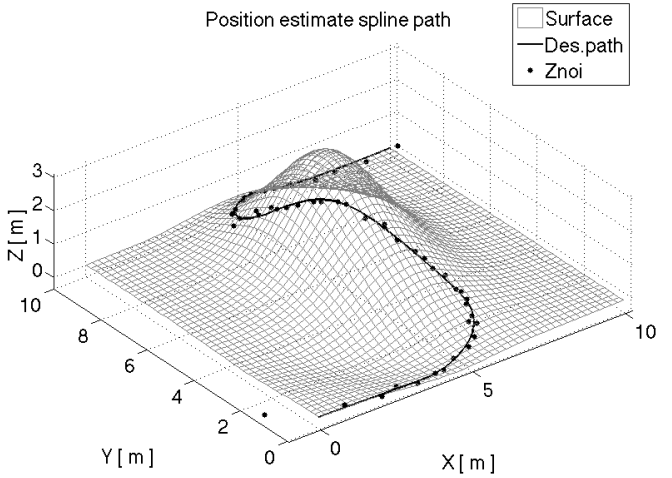


(d) 2D view tracking on uneven surface 4.

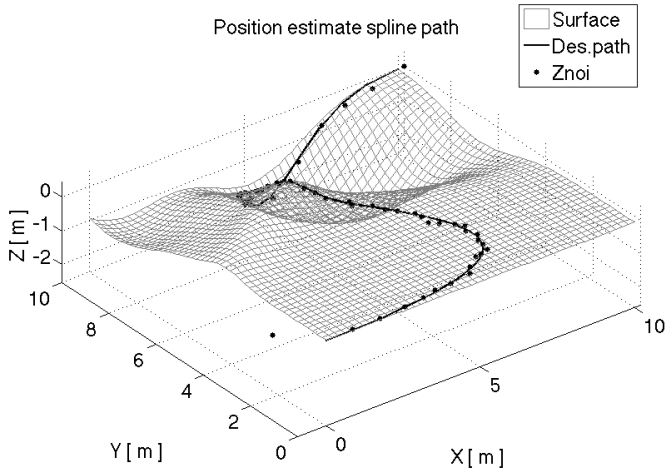
Figure 5.15: 2-dimensional view of Figure 5.14. [Legend: *Surf* is surface, *DesP* is desired path, *ApLKF* is approximate linearisation with Kalman Filter, KF_pest is Kalman filter position estimate and $ActST_p$ is Actual start position.]



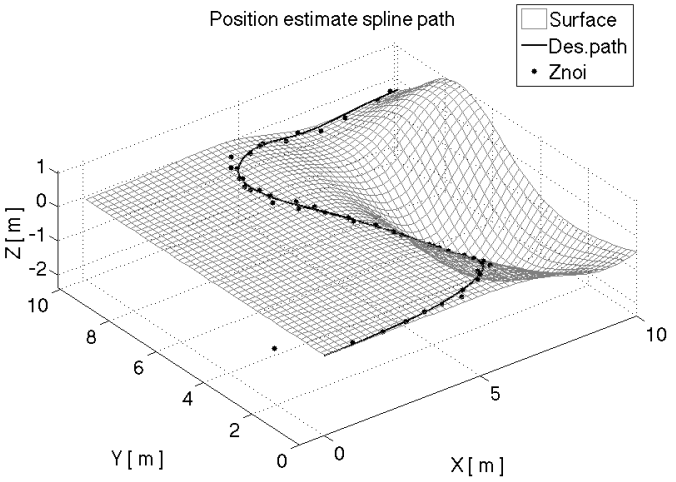
(a) Uneven surface 1.



(b) Uneven surface 2.

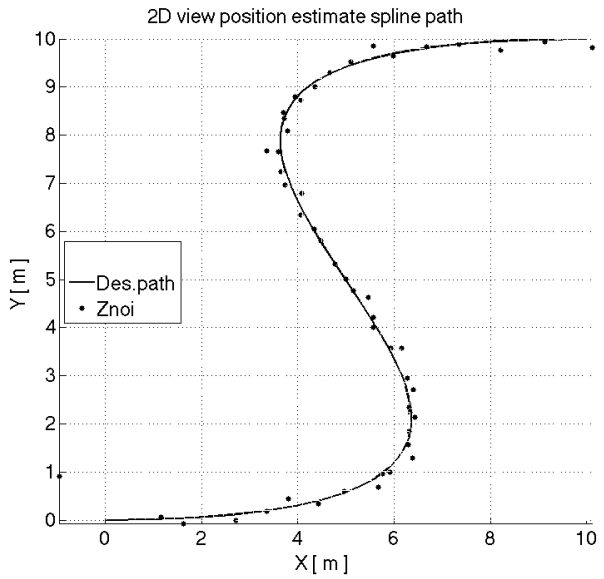


(c) Uneven surface 3.

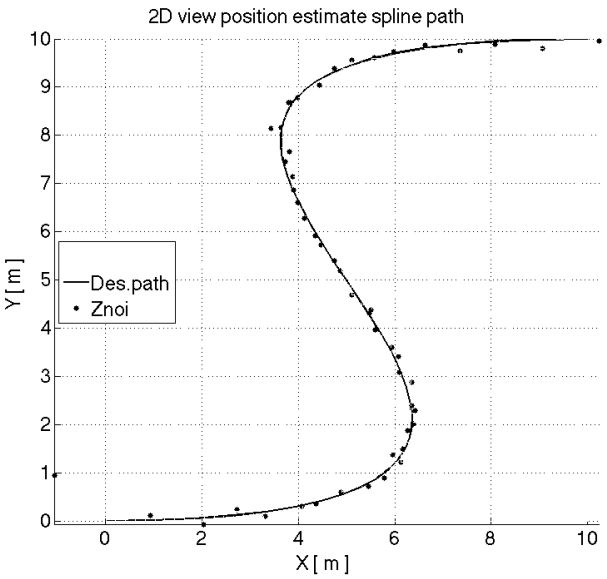


(d) Uneven surface 4.

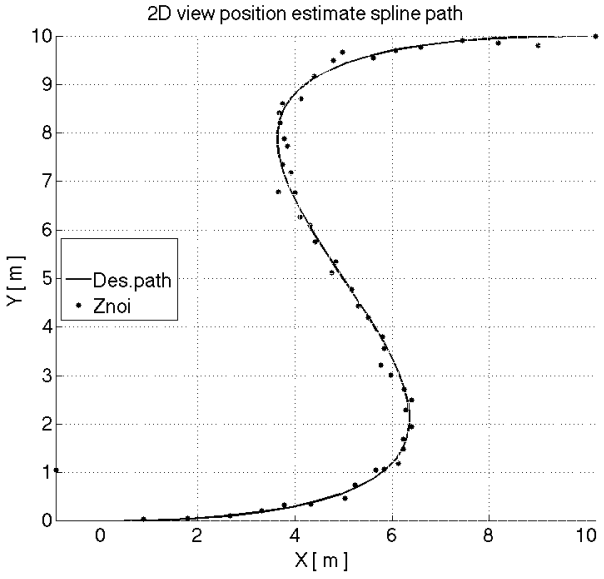
Figure 5.16: Noisy measurement of spline path on uneven surfaces. [Legend: *Surface* is surface, *Des.path* is desired path/trajectory and *Znoi* is measurement noise.]



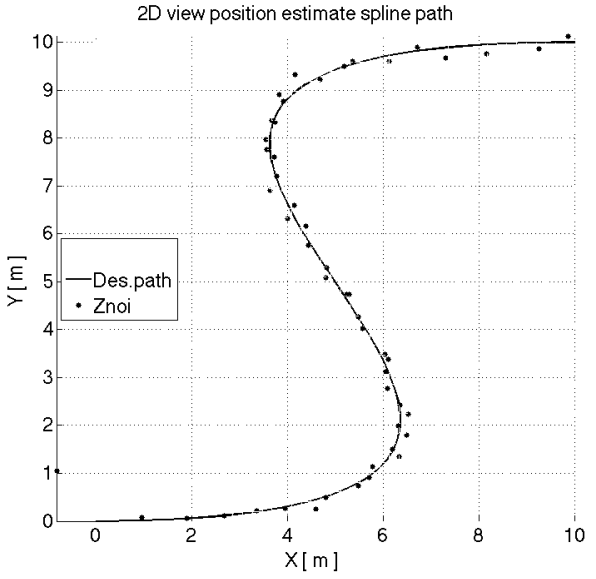
(a) 2D view of uneven surface 1.



(b) 2D view of uneven surface 2.

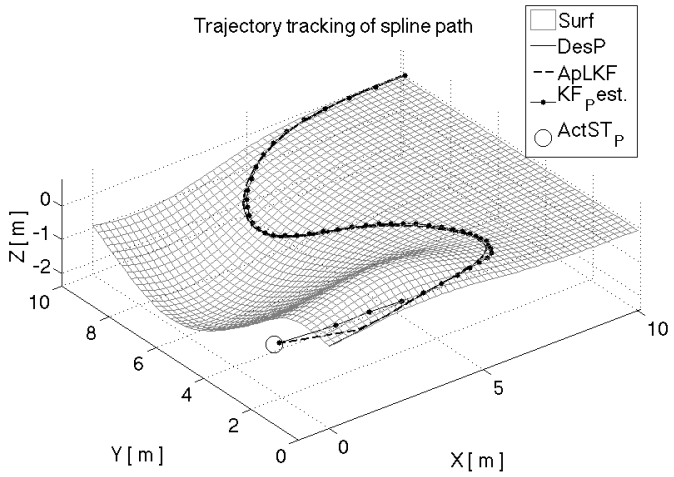


(c) 2D view of uneven surface 3.

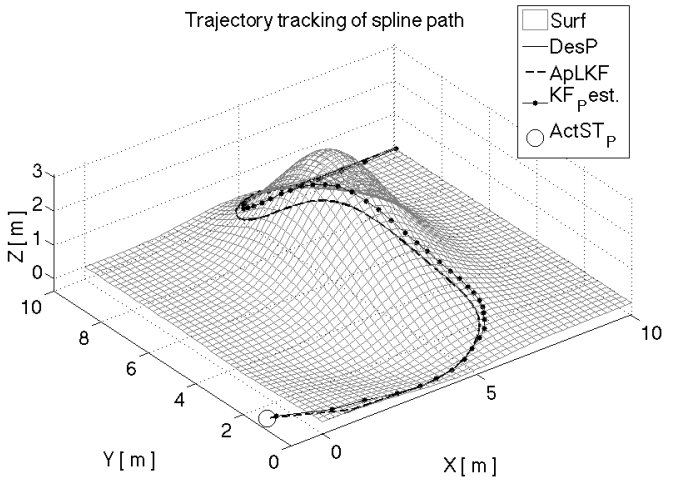


(d) 2D view of uneven surface 4.

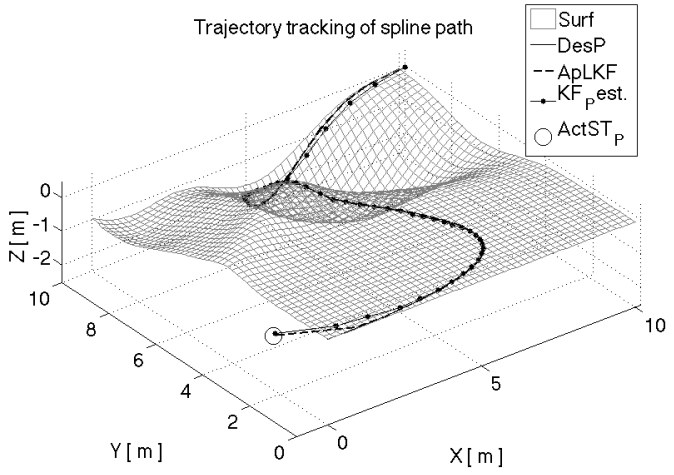
Figure 5.17: 2-dimensional view of Figure 5.16. [Legend: *Des.path* is the desired trajectory/path and *Znoi* is measurement noise.]



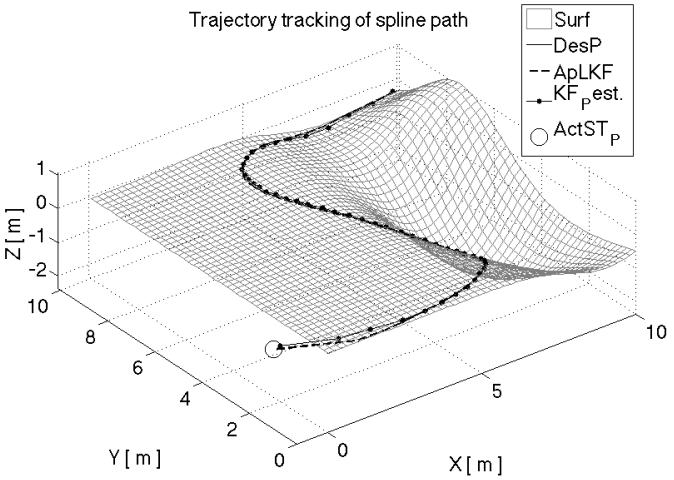
(a) Tracking on uneven surface 1.



(b) Tracking on uneven surface 2.

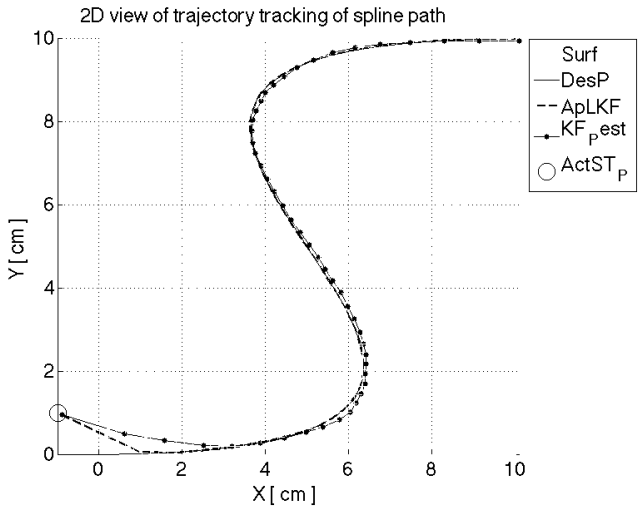


(c) Tracking on uneven surface 3.

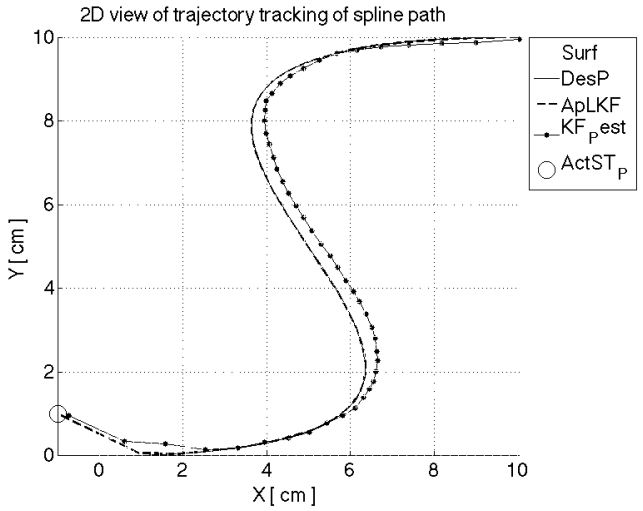


(d) Tracking on uneven surface 4.

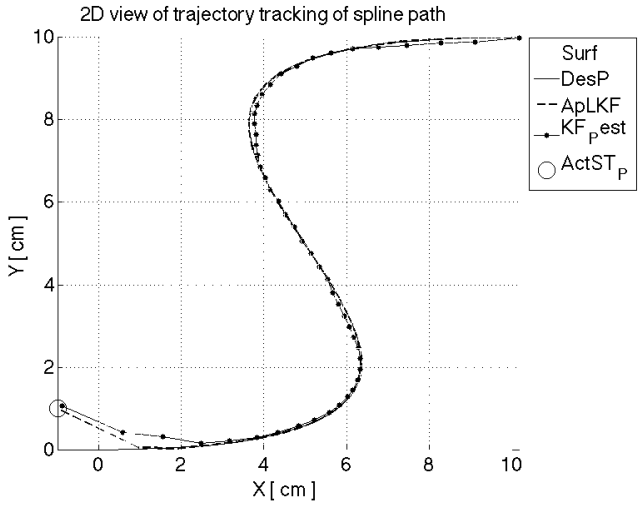
Figure 5.18: Tracking the spline trajectory on uneven surfaces. [Legend: *Surf* is surface, *DesP* is desired path, *ApLKF* is approximate linearisation with Kalman Filter, KF_pest is Kalman filter position estimate and $ActST_p$ is Actual start position.]



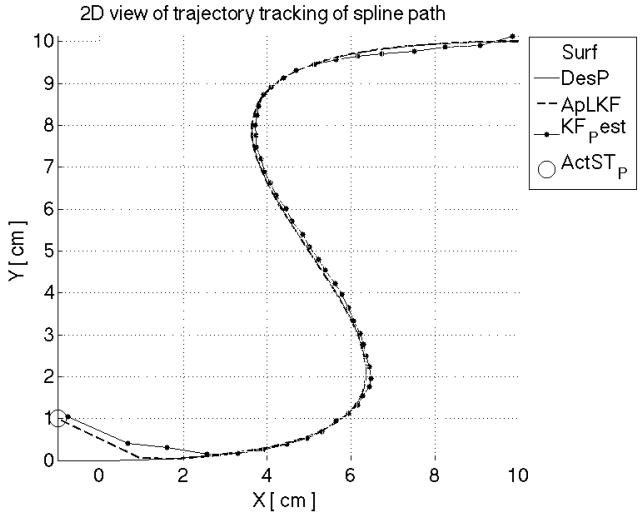
(a) 2D view tracking on uneven surface 1.



(b) 2D view tracking on uneven surface 2.

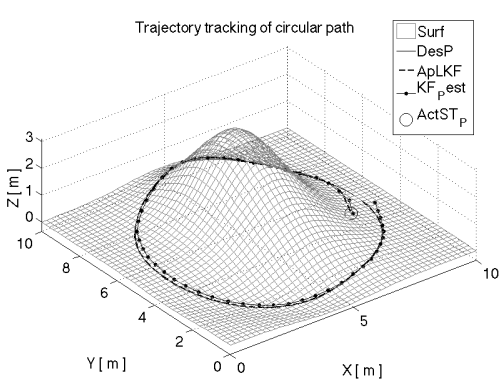


(c) 2D view tracking on uneven surface 3.

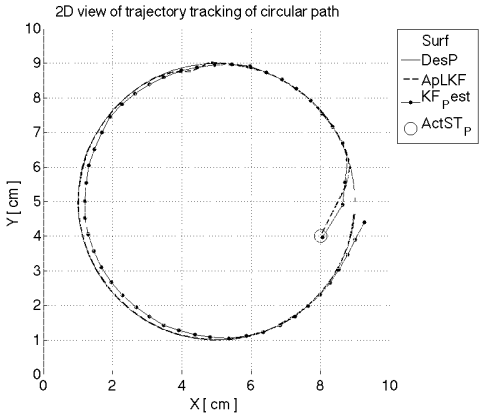


(d) 2D view tracking on uneven surface 4.

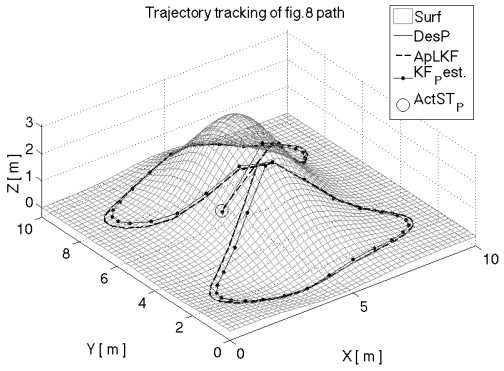
Figure 5.19: 2-dimensional view of Figure 5.18. [Legend: *Surf* is surface, *DesP* is desired path, *ApLKF* is approximate linearisation with Kalman Filter, KF_pest is Kalman filter position estimate and $ActST_p$ is Actual start position.]



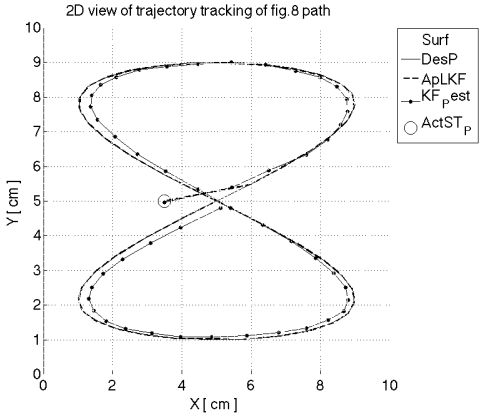
(a) Circular 3D.



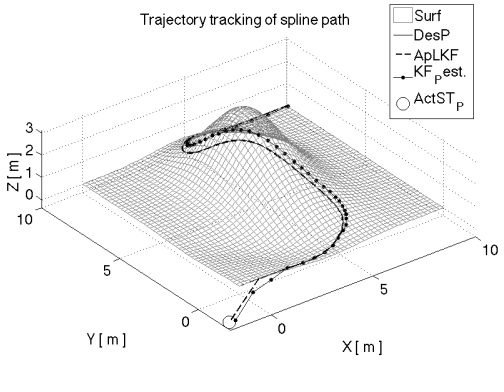
(b) Circular 2D.



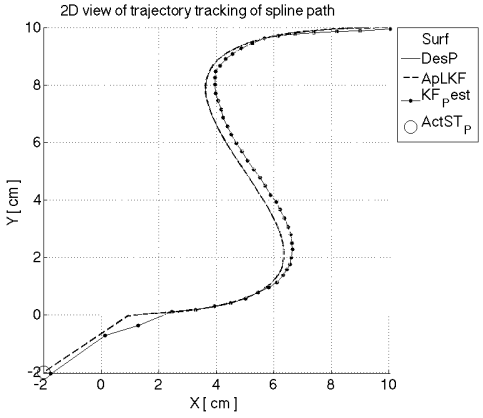
(c) Figure-eight 3D.



(d) Figure-eight 2D.

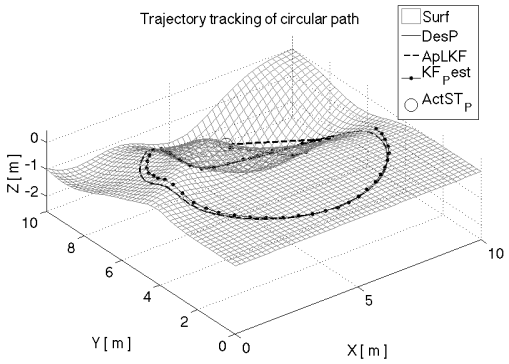


(e) Spline 3D.

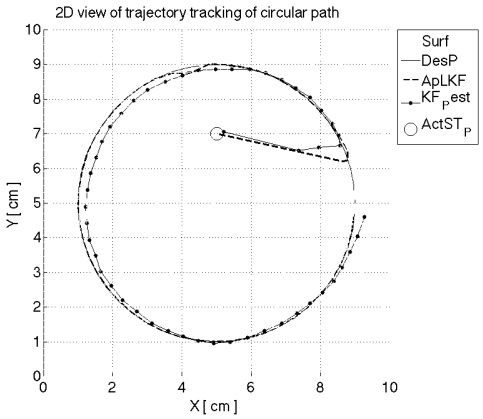


(f) Spline 2D.

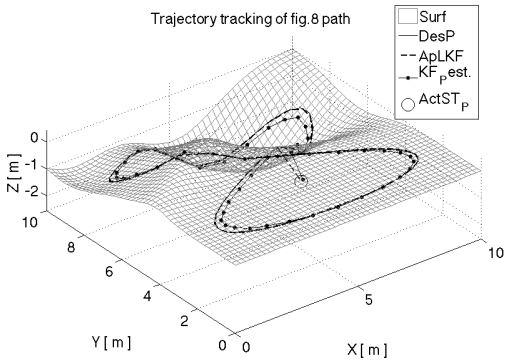
Figure 5.20: Comparison of trajectory tracking on an uneven surface with different initial condition configuration 1. The initial configurations are given as follow: (1) Figure 5.20a and 5.20b $\rightarrow (x_i, y_i, \psi_i) = \left(8, 4, \frac{\pi}{2}\right)$; (2) Figure 5.20c and 5.20d $\rightarrow (x_i, y_i, \psi_i) = \left(3.5, 5, \frac{\pi}{2}\right)$; and (3) Figure 5.20e and 5.20f $\rightarrow (x_i, y_i, \psi_i) = \left(-2, -2, \frac{\pi}{2}\right)$. [Legend: *Surf* is surface, *DesP* is desired path, *ApLKF* is approximate linearisation with Kalman Filter, $KF_p est$ is Kalman filter position estimate and $ActST_p$ is Actual start position.]



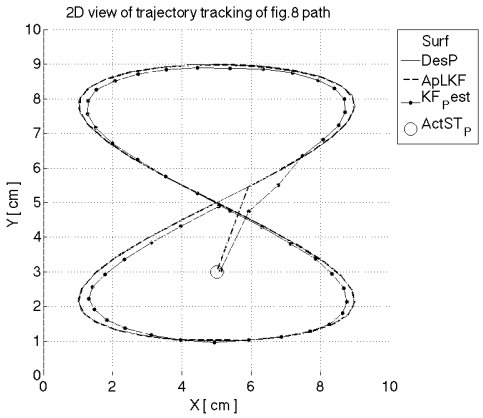
(a) Circular 3D.



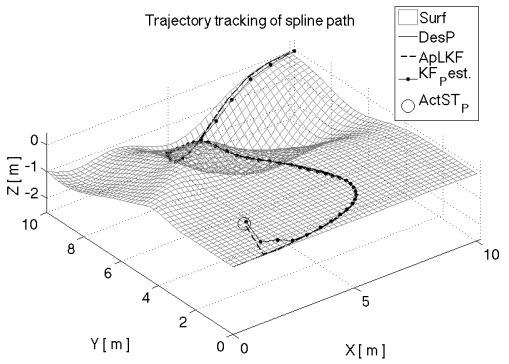
(b) Circular 2D.



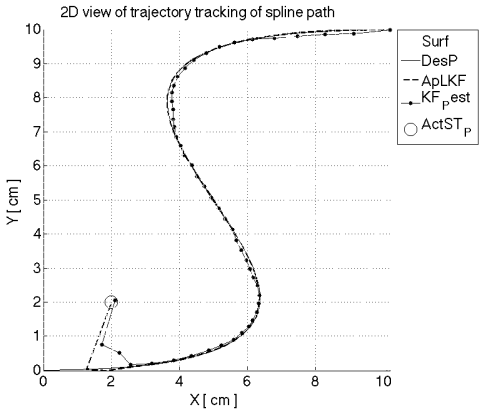
(c) Figure-eight 3D.



(d) Figure-eight 2D.

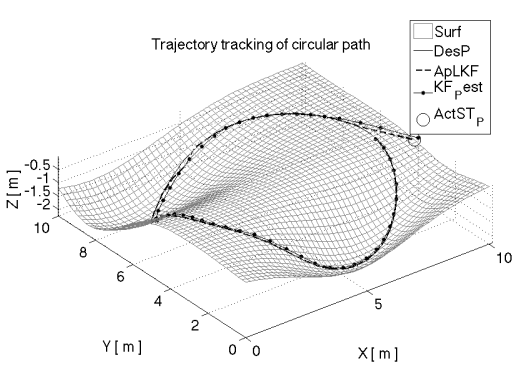


(e) Spline 3D.

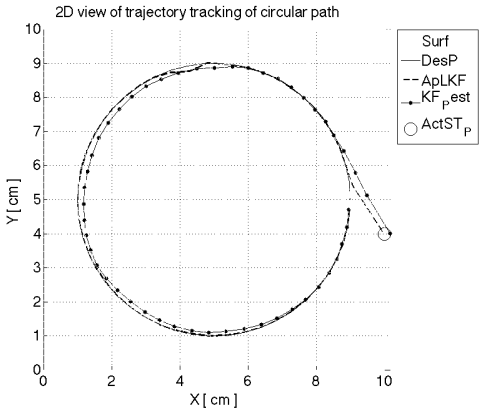


(f) Spline 2D.

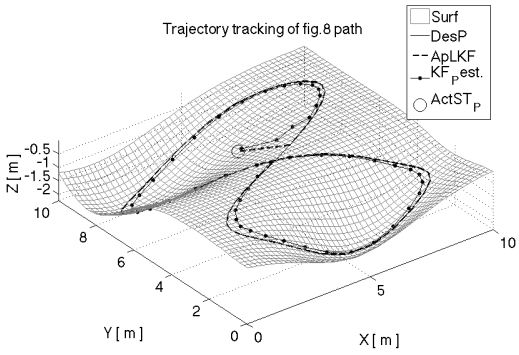
Figure 5.21: Comparison of trajectory tracking on an uneven surface with different initial condition configuration 2. The initial configurations are given as follow: (1) Figure 5.21a and 5.21b $\rightarrow (x_i, y_i, \psi_i) = (5, 7, 0)$; (2) Figure 5.21c and 5.21d $\rightarrow (x_i, y_i, \psi_i) = (5, 3, 0)$; and (3) Figure 5.21e and 5.21f $\rightarrow (x_i, y_i, \psi_i) = (2, 2, 0)$. [Legend: *Surf* is surface, *DesP* is desired path, *ApLKF* is approximate linearisation with Kalman Filter, $KF_p est$ is Kalman filter position estimate and $ActST_p$ is Actual start position.]



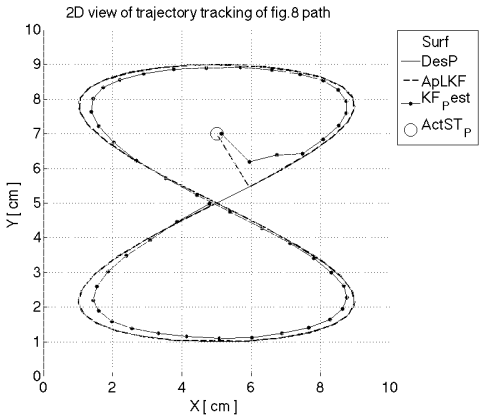
(a) Circular 3D.



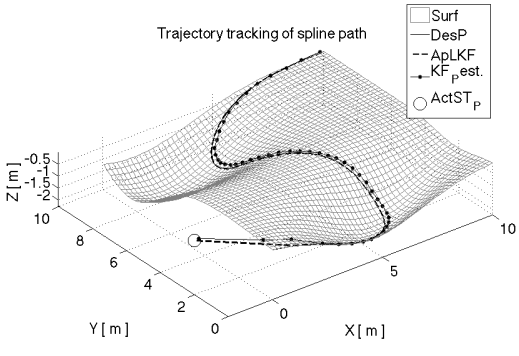
(b) Circular 2D.



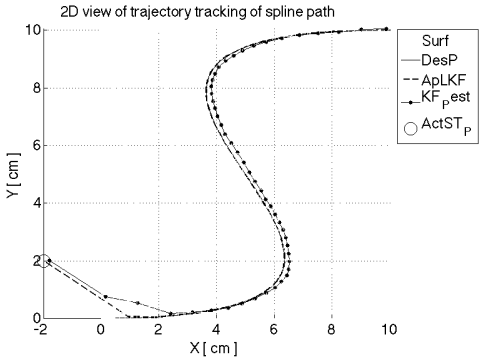
(c) Figure-eight 3D.



(d) Figure-eight 2D.

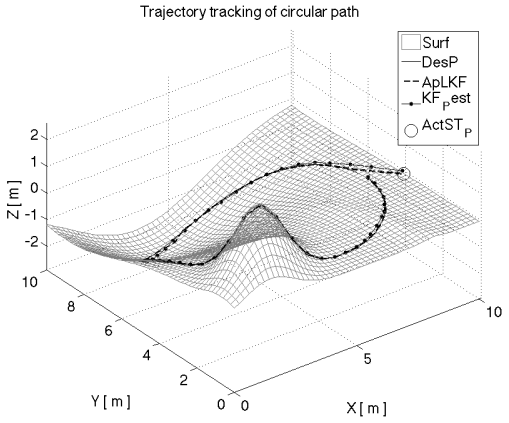


(e) Spline 3D.

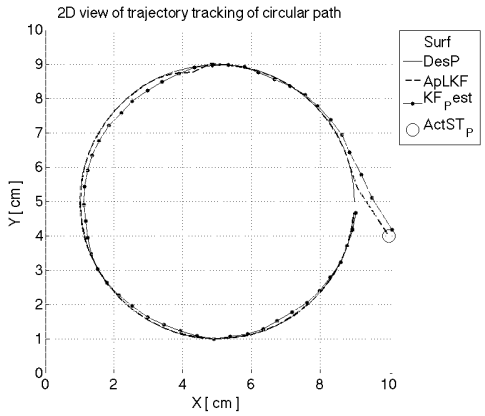


(f) Spline 2D.

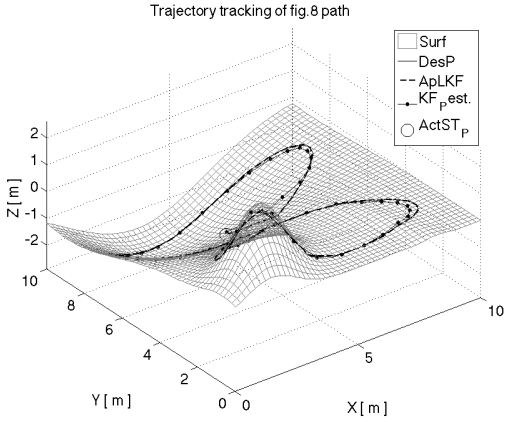
Figure 5.22: Comparison of trajectory tracking on an uneven surface with different initial condition configuration 3. The initial configurations are given as follow: (1) Figure 5.22a and 5.22b $\rightarrow (x_i, y_i, \psi_i) = \left(10, 4, 3\frac{\pi}{2}\right)$; (2) Figure 5.22c and 5.22d $\rightarrow (x_i, y_i, \psi_i) = \left(5, 7, -\frac{\pi}{2}\right)$; and (3) Figure 5.22e and 5.22f $\rightarrow (x_i, y_i, \psi_i) = \left(-2, 2, -\frac{\pi}{2}\right)$. [Legend: *Surf* is surface, *DesP* is desired path, *ApLKF* is approximate linearisation with Kalman Filter, $KF_p est$ is Kalman filter position estimate and $ActST_p$ is Actual start position.]



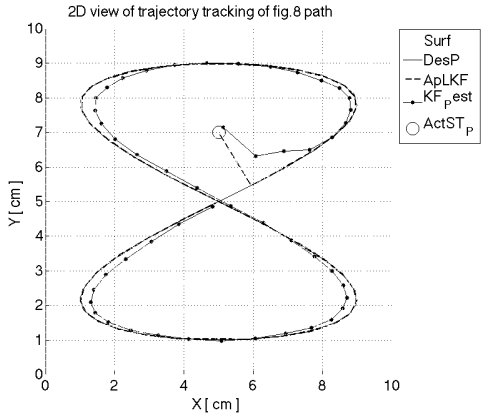
(a) Circular 3D.



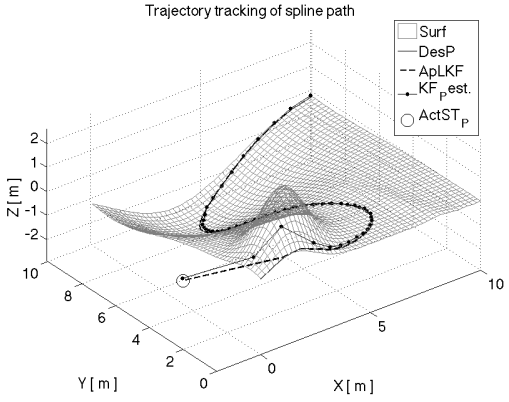
(b) Circular 2D.



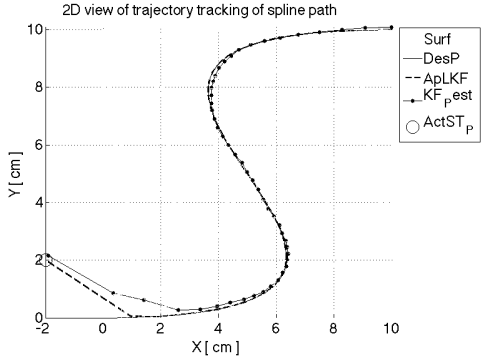
(c) Figure-eight 3D.



(d) Figure-eight 2D.

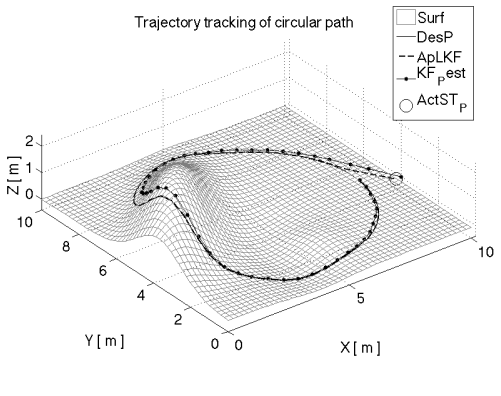


(e) Spline 3D.

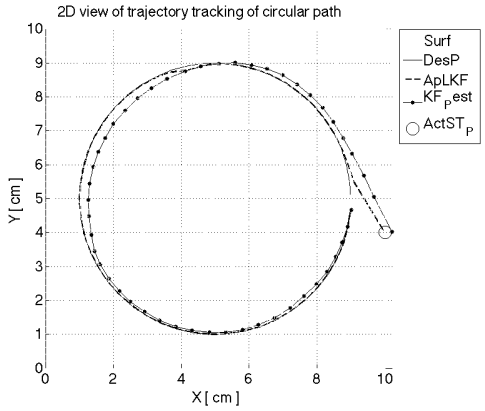


(f) Spline 2D.

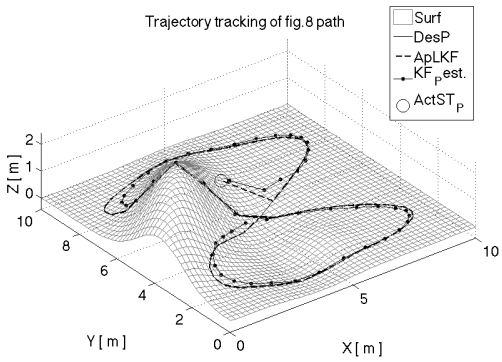
Figure 5.23: Trajectory tracking on an uneven surface with $0 < \zeta < 1$. [Legend: *Surf* is surface, *DesP* is desired path, *ApLKF* is approximate linearisation with Kalman Filter, $KF_p est$ is Kalman filter position estimate and $ActST_p$ is Actual start position.]



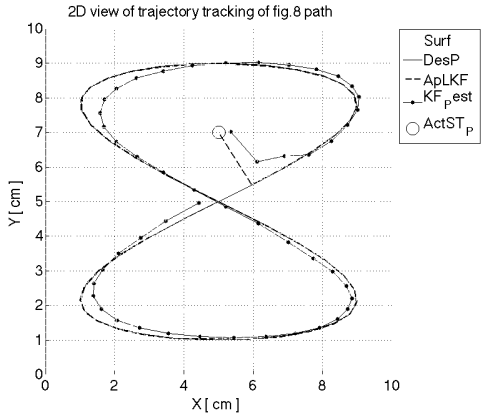
(a) Circular 3D.



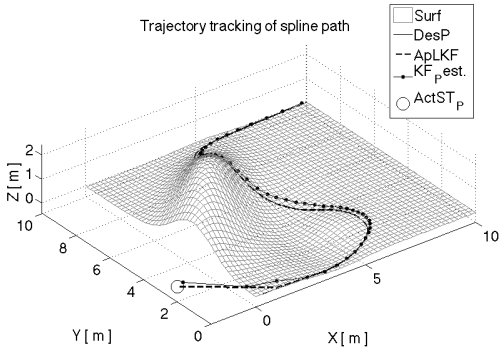
(b) Circular 2D.



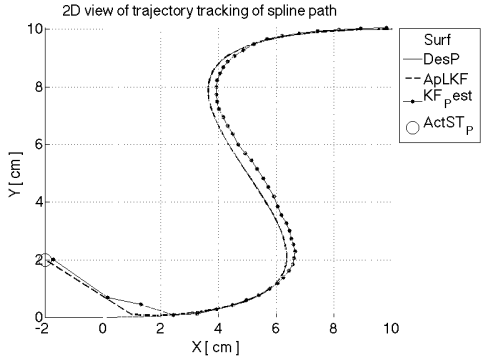
(c) Figure-eight 3D.



(d) Figure-eight 2D.

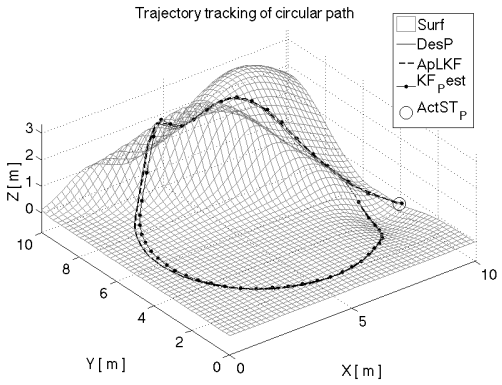


(e) Spline 3D.

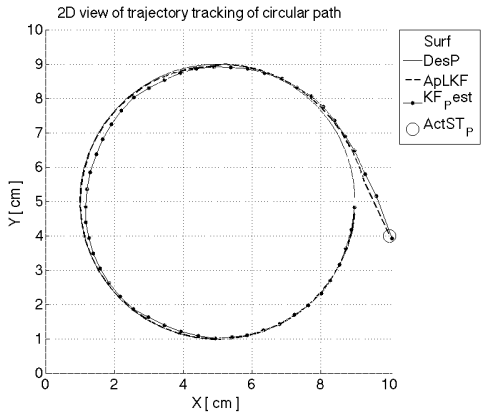


(f) Spline 2D.

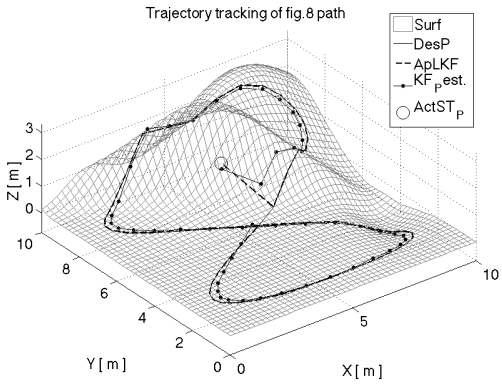
Figure 5.24: Trajectory tracking on an uneven surface with $\zeta = 1$. [Legend: *Surf* is surface, *DesP* is desired path, *ApLKF* is approximate linearisation with Kalman Filter, $KF_p est$ is Kalman filter position estimate and $ActST_p$ is Actual start position.]



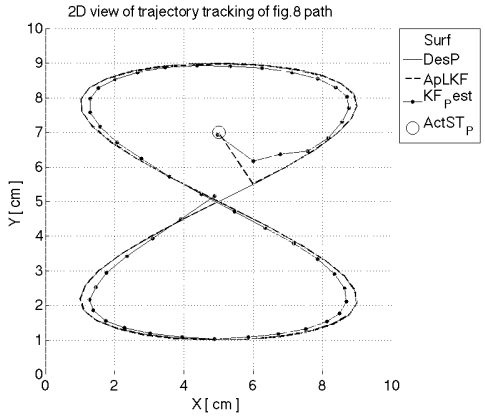
(a) Circular 3D.



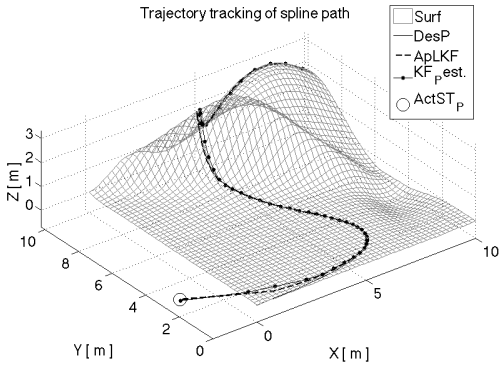
(b) Circular 2D.



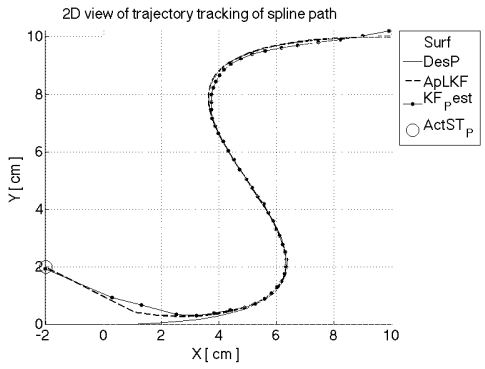
(c) Figure-eight 3D.



(d) Figure-eight 2D.

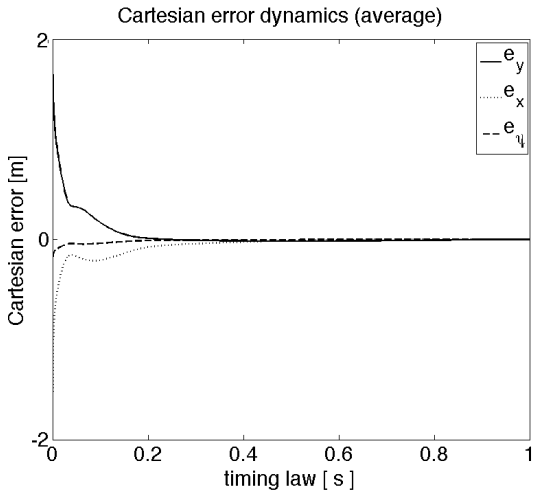


(e) Spline 3D.

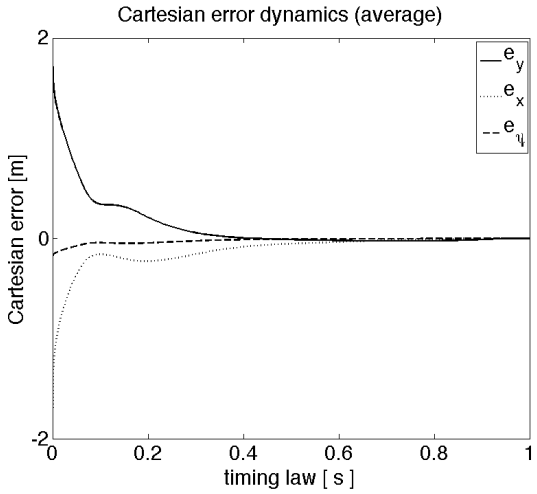


(f) Spline 2D.

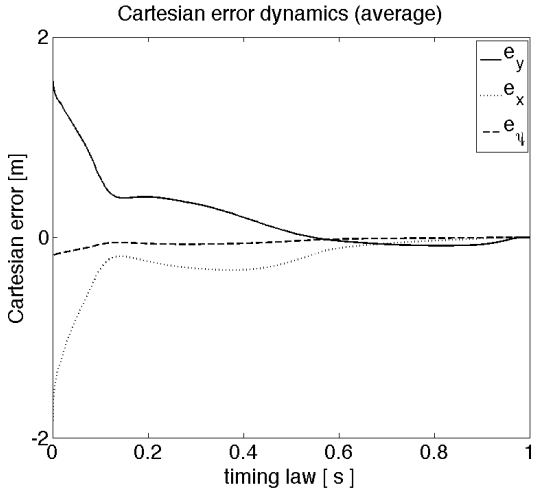
Figure 5.25: Trajectory tracking on an uneven surface with $\zeta > 1$. [Legend: *Surf* is surface, *DesP* is desired path, *ApLKF* is approximate linearisation with Kalman Filter, $KF_p est$ is Kalman filter position estimate and $ActST_p$ is Actual start position.]



(a) Undamped, $0 < \zeta < 1$.



(b) Critical damp, $\zeta = 1$.



(c) Overdamp, $\zeta > 1$.

Figure 5.26: Error dynamics response vs damping coefficient, ζ. Figure 5.26a, 5.26b and 5.26c correspond to the average error dynamics from underlying computation of trajectory tracking results shown by Figure 5.23, 5.24 and 5.25.

5.6 Discussion

As mentioned in Section 5.1.1, the study of mobile robot motion on uneven surface focusses on global navigation problem rather than solving a trajectory tracking problem. This study set out with the aim of assessing the convergence to zero error of the trajectory tracking of mobile robot motion on an uneven surface through utilisation of existing flat surface motion solution laid out in Kanayama et al. (1990), Sun (2005) and Siciliano et al. (2009). The zero convergence is guaranteed for time-invariant case and stability is bounded via Lyapunov analysis (Zhang et al., 2007).

The results in this study show that:

- zero convergence of trajectory tracking error was performed successfully on an uneven surface as well as flat surface; and
- an improved mobile robot configuration along the desired trajectory benefited from the underlying Kalman filter algorithm through an acceptable ratio between covariance matrix of $Q(k)$ and $R(k)$.

The most interesting finding was the accomplishment of zero convergence of trajectory tracking error through utilisation of a simple approximate linearisation method to solve nonlinearity in error dynamics in the event of mobile robot motion on an uneven surface environment. In equation (5.35), $Q(k)$ was defined through "trial and error" method and from result in Section (5.5) we have seen a big value of $Q(k)$ coefficient guarantees zero convergence. Comparing the results in Section (5.5) with related studies done in flat surface environment, the findings are in agreement with Siciliano et al. (2009) where selection of control parameters (in particular, control gain, k, and damping coefficient, ζ) are essential to improve convergence to the desired trajectory. In spite of that, a mobile robot operating in unstructured environment, especially in urban search and rescue (USAR) operation, often requires to replan the desired trajectory in case of:

- avoiding unexpected hazard or non-negotiable obstacles such as big boulders or extreme heat on the desired trajectory; or,
- locating and approaching potential sign of survivor but off the desired trajectory for further rescue action.

On the one hand, it is very common for any uneven surface to have a steep slope where a variable geometry tracked vehicle (VGTV) mobile robot is required to vary its shape in order to negotiate with the terrain to prevent mishap overturn. Therefore, further work is required to establish these limitations.

5.7 Concluding Remarks

In this chapter, an extension of a solution for the trajectory tracking problem of a non-holonomic mobile robot by extending the tracking error model of a flat surface to an uneven surface is proposed. The proposed controller has certain advantages over existing trajectory tracking controllers based on two reasons:

1. Kalman filter algorithm is used to compute the estimate of the mobile robot's configuration instead of direct integral operation of the dynamic part, $\dot{q}$, before comparing with the desired state of the global control input; and

2. Tracking controller is capable to execute mission on uneven surfaces as well as flat surfaces.

The proposed tracking control algorithm is tested on three examples of 2-dimensional pre-planned trajectories projected onto uneven surfaces. The controller design relies on linear and angular velocities (υ_{des} and ω_{des}) as input reference signals. This input reference signal concisely updated the input transformation (υ_{fb} and ω_{fb}) part which improves the trajectory tracking performance. Convergence to the desired trajectory was analysed and simulations were performed to illustrate the behaviour of the proposed control scheme. The above comprehensive analysis shows that the approximate linearisation control scheme based tracking controller is a good trajectory tracking controller considering:

- The convergence to the desired trajectory (with zero tracking error or very small tracking error) will be achieved regardless of:
  - the type of pre-defined trajectory;
  - the type of surface (flat or uneven); and
  - mobile robot start configuration.

- A smooth actual trajectory of a mobile robot will be produced as a result of using the Kalman filter algorithm in this control scheme which makes it relevant in practical applications due to sensor's noisy measurements.

- A shorter settling time (time required to achieve convergence to desired trajectory) can be accomplished with appropriate parameter selection of damping coefficient, ζ, which reflects the controller's gain (k_x, k_y and k_ψ).

Simulation results verify the approximate linearisation approach yields global convergence of desired trajectory of the closed-loop tracking system even with very poor initial start configuration. Furthermore, the results show that the control objectives were accomplished and provide substantial guidance to develop an actual trajectory tracking controller for practical systems.

The next chapter will dedicated for the conclusion of this thesis. Additionally, recommendation and potential further research work related to this research topis will be provided.

Chapter 6

Conclusions and Future Work

This thesis has investigated possible solution to the problems of implementing mobile robot systems for a complex, uneven and unstructured environment, such as in USAR operation. The results from this research work make several contributions to the current literature.

- First, the development of a new and novel design of a VGTV mobile robot configuration has been made, by introducing the independent track control mechanisms. The effects of these independent track control mechanisms is associated with the mobile robot behaviour, particularly when platform stability is required.

- Second, the development of a control framework which is arranged according to the degree of control hierarchy, where the control flow is applied in the concept of a *three-tiered layer control architecture*. It is shown that the outline of this control framework is beneficial for systematic programming, simulation and modeling.

- Third, the development of a mathematical model to represent an actual terrain feature is made, generated using summation of several Gaussian functions. The visualisation of the terrain structure is incorporated with the interaction between the *highest control layer* and *intermediate control layer* in the *three-tiered layer control architecture*.

- Fourth, the development of a comprehensive, full three-dimensional kinematic model for mobile robots that are required to travel on complex and uneven terrain is derived. It is shown that this model is sufficiently concise as to provide the evaluation and prediction of mobile robot behaviour when traversing on uneven terrain.

- Fifth, the development of trajectory tracking model to ensure the mobile robot is following the designated three-dimensional path which is defined in *three-tiered layer control architecture*, extending the implementation of conventional two-dimensional trajectory tracking solution onto three-dimensional problem.

Prior to these, in Chapter 1, we have identified some of the key areas to focus in order to motivate the methodologies employed in this thesis. In this chapter, we summarised and concluded the research works presented thoughout this thesis. The structure of this chapter is arranged as follows: in Section 6.1, we summarise and conclude all the main areas covered in this thesis. The final section, Section 6.2, discusses some potential future work on this research topic.

6.1 Summary

Chapter 2 was written with the objective to develop a new mobile robot that is able to travel on various types of terrain. Influenced by the Variable Geometry Tracked Vehicle configuration proposed in Kim and Lee (2007) and iRobot (2010), the proposed mobile robot in this thesis is driven by a pair of caterpillar tracks which can be controlled independently. Moreover, a fundamental problem related to the track's tension properties is discussed. A mathematical formulation of the track tension model was also presented. This work contributes to existing knowledge of VGTV mobile robot design by providing the independent track control method which is very beneficial in complex terrain negotiation. The chapter then discussed the proposed mechatronic design and at the same time previewed the control architecture by emphasising the lowest control layer. This control architecture is important for the rest of the modelling and simulation in this thesis, presented in Chapter 3.

Chapter 3 explained the importance of a control framework in defining the mobile robot behaviour. This chapter began with the objective to develop a generic mobile robot control architecture by focusing on the terrain negotiation scenarios. One of the significant advantages of the proposed control framework is the arrangement of the *three-tiered layer control architecture* which is based on the degree of control hierarchy. Moreover, the relevence of the layer architecture system is improved by implementing the 'switch'ing ability which is responsible to dictate any intervention over the prior defined global mission objective. Furthermore, the 'switch'ing ability provides deeper understanding of the role of mission planner in the *highest control layer*. The outcome of the proposed control framework supports the idea of a systematic control approach and further formulation in the next chapter.

Chapter 4 was written with the objective to derive a higher degree-of-freedom kinematic model for a mobile robot that is supposed to traverse on an uneven surface. This higher-degree-of-freedom kinematic model is derived by extending the existing two-dimensional kinematic model into a full three-dimensional kinematic model. This study set out with the aim of evaluating the mobile robot behaviour in terms of linear and angular velocities along a pre-planned path of an uneven surface with the knowledge of two-dimensional kinematic problem. Even though

the kinematic model for non-holonomic mobile robot is not unique by definition, the proposed generalised model constructs a comprehensive and solid foundation of the framework to study kinematic problem in real-world environment. The study has gone some way towards enhancing our understanding of mobile robots behaviour when it is required to traverse on uneven terrain. The following conclusions can be drawn from the present study:

- uneven terrain can be modelled by a the summation of several multivariate Gaussian functions;
- surface model is beneficial for mobile robot to deal with terrain or obstacles rather than avoiding it;
- unit quaternion rotation formulation is useful as an alternative to Euler angle; and
- input vector, u, for uneven surface can be reduced to two-dimensional case considering the elimination of undesired rotational element from the equation.

The outcome of this is significant for the derivation of tracking model derived in Chapter 5.

Chapter 5 has investigated a solution for the trajectory tracking problem of a non-holonomic mobile robot by extending the tracking error model of a flat surface to an uneven surface. The present study is designed to determine the effect of the Kalman filter algorithm to estimate the mobile robot configuration along the predefined path. In addition, this study also aims to assess the capability of the tracking controller in mission execution especially on an uneven surface. This work contributes to existing knowledge of conventional trajectory tracking method by providing a solution for a mobile robot systems travel on complex terrain. Moreover, the current findings add substantially to our understanding of Kalman filter ability in order to get a smooth tracking trajectory whilst converging to the desired path. The most obvious finding to emerge from this study are that the convergence to the desired trajectory can be achieved regardless of:

- the type of pre-defined trajectory;
- the type of surface (flat or uneven); and
- mobile robot start configuration.

The results also shown that a smooth trajectory of a mobile robot will be produced as a result of using the Kalman filter algorithm in this control scheme which makes it relevant in practical applications due to sensor's noisy measurements.

6.2 Future Work

The work discussed in this thesis has identified and confirmed the feasibility of: (1) independent track control mechanism on VGTV mobile robot to negotiate with various types of terrain; and (2) control architecture framework classification in order to systematically simulate (and program) the kinematic and tracking behaviour of mobile robot system on complex terrain. Moreover, this research has thrown up many questions in need of further investigation. Further experimental investigations are needed to improve the mechatronic design of the proposed mobile robot system. More broadly, research is also needed to expand the scope of control architecture in order to increase the mobile robot level of autonomy, towards more autonomous capability.

It is recommended that further research on the **mechatronic design** needs to be undertaken in order to improve the competency of the mobile robot platform on complex terrain. Further research might explore:

- the optimisation of the chassis dimension to ensure sufficient chassis and terrain surface clearance;
- the utilisation of full-drive-system (or four-wheel-drive system) on the prototype by implementing four motors to drive four track sprockets; and
- the capability of standard DC motor to construct track angle, θ_{cl}, in order to overcome servo mechanism limitations.

It would be more interesting to assess the effects of surface side-slope angle onto the mobile robot behaviour, which is associated with the mobile robot decision making ability. Furthermore, more research is required on real-world implementation, particularly to investigate the effectiveness of mechatronic design on actual disaster site trials. However, this assessment is entirely depends on the customisation of the control architecture.

Additionally, considerably more work on **control architecture** will need to be done to upgrade to existing prototype into another level of autonomy. A further study on sensor fusion topic could apply the surface model methodology explained in Chapter 4. It is suggested to employ a Microsoft Kinect sensor instead of stereo vision camera in order to model the obstacle feature. A future study investigating the incorporation of sensor fusion module embedded in the higher control layers would be very interesting, particularly to determine the appropriate behaviour of mobile robot on terrain model. It is forecasted that the future research will emerge into *SLAM* domain.

Bibliography

Samir Alili, Rachid Alami, and Vincent Montreuil. A task planner for an autonomous social robot. In *Distributed Autonomous Robotic Systems*, pages 335–344. Springer, 2009.

Joseph Auchter, Carl A. Moore, and Ashitava Ghosal. A novel kinematic model for rough terrain robots. In *Advances in Computational Algorithms and Data Analysis*, volume 14 of *Lecture Notes in Electrical Engineering*, pages 215–234. Springer Netherlands, 2009. ISBN 978-1-4020-8918-3.

T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (slam): part ii. 13(3): 108–117, 2006. ISSN 1070-9932.

Joseph A Barbera and Michael Lozano. Urban search and rescue medical teams: Fema task force system. *Prehospital and Disaster Medicine*, 8(04):349–356, 1993.

Peter Bartz. Building an ahrs/head-tracker using the "9dof razor imu" or the "9dof sensor stick" by sparkfun, 2012. URL `https://dev.qu.tu-berlin.de/projects/sf-razor-9dof-ahrs/wiki/Tutorial`.

G Bekey. Current trends in robotics: technology and ethics. *Robot ethics: the ethical and social implications of robotics. MIT Press, Cambridge*, pages 17–34, 2012.

Samuel Blackman and Artech House. Design and analysis of modern tracking systems, 1999.

Saso Blazic. Takagi-sugeno vs. lyapunov-based tracking control for a wheeled mobile robot. *WSEAS Transactions on Systems and Control*, 5(8):667–676, 2010. ISSN 1991-8763.

ZM NDU Bob Struijk. Robot production volume data trends and analysis. 2012.

Alex Brooks, Tobias Kaupp, Alexei Makarenko, Stefan Williams, and Anders Oreback. Towards component-based robotics. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005.(IROS).*, pages 163–168. IEEE, 2005.

Rodney Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation,*, 2(1):14–23, 1986.

David J Bruemmer, Donald D Dudenhoeffer, and Julie L Marble. Dynamic-autonomy for urban search and rescue. In *AAAI Mobile Robot Competition*, pages 33–37, 2002.

Wolfram Burgard and Martial Hebert. World modeling. In Bruno Siciliano and Oussama Khatib, editors, *Springer Handbook of Robotics*, pages 853–869. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-23957-4.

Jennifer Carlson and Robin R Murphy. How ugvs physically fail in the field. *IEEE Transactions on Robotics,*, 21(3):423–437, 2005.

Jennifer Casper and Robin R. Murphy. Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. *IEEE Transactions on Systems, Man, and Cybernetics*, 33(3):367–385, 2003.

Jennifer L. Casper, Mark Micire, and Robin R. Murphy. Issues in intelligent robots for search and rescue. volume 4024, pages 292–302, Orlando, FL, USA, 2000. SPIE. URL `http://link.aip.org/link/?PSI/4024/292/1`.

Nilanjan Chakraborty and Ashitava Ghosal. Dynamic modeling and simulation of a wheeled mobile robot for traversing uneven terrain without slip. *Journal of Mechanical Design*, 127: 901–909, 2005.

Y. Chiu, N. Shiroma, H. Igarashi, N. Sato, M. Inami, and F. Matsuno. Fuma: environment information gathering wheeled rescue robot with one-dof arm. pages 81–86, 2005.

Howie Choset. *Bridge inspection with serpentine robots.* Transportation Research Board, IDEA Program, 2002.

Howie M Choset, Jonathan E Luntz, Ellie Shammas, Tarek Rached, Douglas Hull, and Christina C Dent. Design and motion planning for serpentine robots. In *SPIE's 7th Annual International Symposium on Smart Structures and Materials*, pages 148–155. International Society for Optics and Photonics, 2000.

Gilles Clement and Eric Villedieu. Variable geometry track vehicle, 1987.

Ève Coste-Manière and Reid Simmons. Architecture, the backbone of robotic systems. In *IEEE International Conference on Robotics and Automation, 2000. Proceedings. ICRA'00.*, volume 1, pages 67–72. IEEE, 2000.

M.R. Daud, K. Nonami, and A. Irawan. Lrf assisted autonomous walking in rough terrain for hexapod robot comet-iv. In Kenzo Nonami, Muljowidodo Kartidjo, Kwang-Joon Yoon, and Agus Budiyono, editors, *Autonomous Control Systems and Vehicles*, volume 65 of *Intelligent*

Systems, Control and Automation: Science and Engineering, pages 237–249. Springer Japan, 2013. ISBN 978-4-431-54275-9.

Angela Davids. Urban search and rescue robots: from tragedy to technology. *Intelligent Systems, IEEE*, 17(2):81–83, 2002.

Alessandro De Luca, Giuseppe Oriolo, and Marilena Vendittelli. Control of wheeled mobile robots: An experimental overview. In *Ramsete*, pages 181–226. Springer, 2001.

D. DeVon and T. Bretl. Kinematic and dynamic control of a wheeled mobile robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS.*, pages 4065–4070, 2007.

Rüdiger Dillmann. Teaching and learning of robot tasks via observation of human performance. *Robotics and Autonomous Systems*, 47(2):109–116, 2004.

Gamini Dissanayake, Hugh Durrant-Whyte, and Tim Bailey. A computationally efficient solution to the simultaneous localisation and map building (slam) problem. In *IEEE International Conference on Robotics and Automation, 2000.*, volume 2, pages 1009–1014. IEEE, 2000.

M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. 17(3):229–229, 2001.

Gregory Dudek and Michael Jenkin. *Computational principles of mobile robotics*. Cambridge university press, 2010.

H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part i. 13(2):99–110, 2006. ISSN 1070-9932.

Thomas Edlinger and Ewald von Puttkamer. Exploration of an indoor-environment by an autonomous mobile robot. In *Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems' 94.' Advanced Robotic Systems and the Real World', IROS'94.*, volume 2, pages 1278–1284. IEEE, 1994.

Alberto Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation,*, 3(3):249–265, 1987.

Farbod Fahimi. Mobile robots. In *Autonomous Robots*, pages 1–58. Springer US, 2009.

R. Fierro and F. L. Lewis. Control of a nonholonomic mobile robot: backstepping kinematics into dynamics. In *Proceedings of the 34th IEEE Conference on Decision and Control*, volume 4, pages 3805–3810, 1995.

Kikuo Fujimura and Hanan Samet. A hierarchical strategy for path planning among moving obstacles [mobile robot]. *IEEE Transactions on Robotics and Automation,*, 5(1):61–69, 1989.

E. Garcia, M. A. Jimenez, P. G. De Santos, and M. Armada. The evolution of robotics research. 14(1):90–103, 2007. ISSN 1070-9932.

Erann Gat et al. On three-layer architectures. *Artificial intelligence and mobile robots*, pages 195–210, 1998.

Andrew A Goldenberg, Jun Lin, et al. Variable configuration articulated tracked vehicle, October 13 2009. US Patent 7,600,592.

Hiroshi Gomi, Kazushi Hamaya, and Takashi Matsumoto. Biped mobile robot, May 20 2003. US Patent 6,564,888.

D. Hahnel, R. Triebel, W. Burgard, and S. Thrun. Map building with mobile robots in dynamic environments. 2003.

B. Hemes, D. Canelon, J. Danes, and N. Papanikolopoulos. Robotic tumbling locomotion. In *2011 IEEE International Conference on Robotics and Automation (ICRA),*, pages 5063–5069, 2011. doi: 10.1109/ICRA.2011.5980330.

K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka. The development of honda humanoid robot. In *IEEE International Conference on Robotics and Automation, 1998.*, volume 2, pages 1321–1326 vol.2, 1998.

S. Hirose. Three basic types of locomotion in mobile robots. In *Fifth International Conference on Advanced Robotics, 1991. 'Robots in Unstructured Environments',*, pages 12–17 vol.1, 1991.

Robert Holmberg and Oussama Khatib. Development and control of a holonomic mobile robot for mobile manipulation tasks. *The International Journal of Robotics Research*, 19(11):1066–1074, 2000.

Thomas M Howard and Alonzo Kelly. Optimal rough terrain trajectory generation for wheeled mobile robots. *The International Journal of Robotics Research*, 26(2):141–166, 2007.

Hui-Min Huang, Kerry Pavek, Brian Novak, James Albus, and E Messin. A framework for autonomy levels for unmanned systems (alfus). *Proceedings of the AUVSIÕs Unmanned Systems North America*, pages 849–863, 2005.

Qiang Huang, Kazuhito Yokoi, Shuuji Kajita, Kenji Kaneko, Hirohiko Arai, Noriho Koyachi, and Kazuo Tanie. Planning walking patterns for a biped robot. *IEEE Transactions on Robotics and Automation,*, 17(3):280–289, 2001.

Ammar Husain, Heather Jones, Balajee Kannan, Uland Wong, Tiago Pimentel, Sarah Tang, Shreyansh Daftry, Steven Huber, and William L Whittaker. Mapping planetary caves with an autonomous, heterogeneous robot team. In *2013 IEEE Aerospace Conference,*, pages 1–13, 2013.

Statistical Department International Federation of Robotics. Professional service robots, 2012. URL `http://www.worldrobotics.org/index.php?id=home&news_id=262`.

iRobot. irobot packbot, 2010. URL `http://www.irobot.com/gi/ground/510_PackBot`.

Taro Iwamoto and Hiroshi Yamamoto. Variable configuration track laying vehicle, November 20 1984. US Patent 4,483,407.

Taro Iwamoto and Hiroshi Yamamoto. Mechanical design of variable configuration tracked vehicle. 112(3):289–294, 1990. URL `http://link.aip.org/link/?JMD/112/289/1`.

Adam Jacoff, Elena Messina, and John Evans. A standard test course for urban search and rescue robots. *NIST Special Publication SP*, pages 253–259, 2001.

Adam Jacoff, Elena Messina, and John Evans. Performance evaluation of autonomous mobile robots. 29(3):259–267, 2002.

Liu Jinguo, Wang Yuechao, Li Bin, and Ma Shugen. Current research, key performances and future development of search and rescue robots. *Journal of Mechanical Engineering*, 2:4, 2007.

Y. Kanayama, A. Nilipour, and C. A. Lelm. A locomotion control method for autonomous vehicles. In *Proceedings, IEEE International Conference on Robotics and Automation*, volume 2, pages 1315–1317, 1988.

Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi. A stable tracking control method for an autonomous mobile robot. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 384–389, 1990.

Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi. A stable tracking control method for a non-holonomic mobile robot. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, pages 1236–1241, 1991.

Kenji Kaneko. Towards emergency response humanoid robots. In *2012 13th Int'l Workshop on Mechatronics, 2012 9th France-Japan & 7th Europe-Asia Congress on and Research and Education in Mechatronics (REM),*, pages 504–511, 2012.

J. Kim and C. Lee. Variable transformation shapes of single-tracked mechanism for a rescue robot. In *International Conference of Control, Automation and System*, pages 1057–1061, Seoul, Korea, 2007.

Hiroaki Kitano, Satoshi Tadokoro, Itsuki Noda, Hitoshi Matsubara, Tomoichi Takahashi, Atsuhi Shinjou, and Susumu Shimada. Robocup rescue: Search and rescue in large-scale disasters as a domain for autonomous agents research. In *1999 IEEE International Conference on Systems, Man, and Cybernetics, 1999. IEEE SMC'99 Conference Proceedings.*, volume 6, pages 739–743. IEEE, 1999.

M. Kitano and H. Jyozaki. A theoretical analysis of steerability of tracked vehicles. 13(4): 241–258, 1976. ISSN 0022-4898.

M. Kitano and M. Kuma. An analysis of horizontal plane motion of tracked vehicles. 14(4): 211–225, 1977. ISSN 0022-4898.

Gregor Klancar and Igor Skrjanc. Tracking-error model-based predictive control for mobile robots in real time. 55(6):460–469, 2007. ISSN 0921-8890. URL `http://www.sciencedirect.com/science/article/pii/S0921889007000140`.

Kurt Konolige, Motilal Agrawal, and Joan Sola. Large-scale visual odometry for rough terrain. In *Springer Tracts in Advanced Robotics*, volume 66, pages 201–212. Springer Berlin Heidelberg, 2011. URL `http://dx.doi.org/10.1007/978-3-642-14743-2_18`.

David Kortenkamp and Reid Simmons. Robotic systems architectures and programming. In Bruno Siciliano and Oussama Khatib, editors, *Springer Handbook of Robotics*, pages 187–206. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-23957-4. doi: 10.1007/978-3-540-30301-5_9. URL `http://dx.doi.org/10.1007/978-3-540-30301-5_9`.

Erwin Kreyszig. *Advanced engineering mathematics*. Wiley.com, 2010.

C. H. Lee, S. H. Kim, S. C. Kang, M. S. Kim, and Y. K. Kwak. Double-track mobile robot for hazardous environment applications. 17:447–459, 2003.

Woosub Lee, Sungchul Kang, Munsang Kim, and Mignon Park. Robhaz-dt3: teleoperated mobile platform with passively adaptive double-track for hazardous environment applications. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004.(IROS 2004).*, volume 1, pages 33–38. IEEE, 2004.

John J Leonard and Hugh F Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(3):376–382, 1991.

Pedro U Lima. Search and rescue robots: The civil protection teams of the future. In *Third International Conference on Emerging Security Technologies (EST), 2012*, pages 12–19. IEEE, 2012.

Jinguo Liu, Yuechao Wang, Bin Li, and Shugen Ma. Current research, key performances and future development of search and rescue robots. *Frontiers of Mechanical Engineering in China*, 2(4):404–416, 2007.

Y. Liu and G. Liu. Interaction analysis and online tip-over avoidance for a reconfigurable tracked mobile modular manipulator negotiating slopes. PP(99):1–13–, 2009a. ISSN 1083-4435. URL `10.1109/TMECH.2009.2031174`.

Yugang Liu and Guangjun Liu. Modeling of tracked mobile manipulators with consideration of track-terrain and vehicle-manipulator interactions. 57(11):1065–1074, 2009b. ISSN 0921-8890.

A. Luca, G. Oriolo, and C. Samson. Feedback control of a nonholonomic car-like robot. In *Lecture Notes in Control and Information Sciences*, pages 171–253. Springer Berlin Heidelberg, 1998.

Theodore W Manikas, Kaveh Ashenayi, and Roger L Wainwright. Genetic algorithms for autonomous robot navigation. *Instrumentation & Measurement Magazine, IEEE*, 10(6):26–31, 2007.

Carlos Marques, João Cristóvão, Paulo Alvito, Pedro Lima, Joao Frazao, Isabel Ribeiro, and Rodrigo Ventura. A search and rescue robot with tele-operated tether docking system. *Industrial Robot: An International Journal*, 34(4):332–338, 2007.

JL Martínez, Anthony Mandow, Jesús Morales, S Pedraza, and A García-Cerezo. Approximating kinematics for tracked mobile robots. *The International Journal of Robotics Research*, 24(10): 867–878, 2005.

Felipe N. Martins, Wanderley C. Celeste, Ricardo Carelli, Mario Sarcinelli-Filho, and Teodiano F. Bastos-Filho. An adaptive dynamic controller for autonomous mobile robot trajectory tracking. 16(11):1354–1363, 2008. ISSN 0967-0661.

Osamu Matsumoto, Shuuji Kajita, Muneharu Saigo, and Kazuo Tani. Biped-type leg-wheeled robot. *Advanced Robotics*, 13(3):235–236, 1998.

A. Micaelli and Claude Samson. Trajectory tracking for unicycle-type and two-steering-wheels mobile robots, 1993. URL `http://hal.inria.fr/inria-00074575/en/`.

Mark Micire and Robin Murphy. Analysis of the robotic-assisted search and rescue response to the world trade center disaster. 2002.

David P Miller and Cathryne Stein. Creating autonomous roboticists. *Intelligent Systems, IEEE*, 16(2):20–23, 2001.

E. Z. Moore, D. Campbell, F. Grimminger, and M. Buehler. Reliable stair climbing in the simple hexapod 'rhex'. In *IEEE International Conference on Robotics and Automation, 2002. Proceedings. ICRA '02.*, volume 3, pages 2222–2227, 2002.

S. A. A. Moosavian, H. Semsarilar, and A. Kalantari. Design and manufacturing of a mobile rescue robot, 2006. URL `10.1109/IROS.2006.281835`.

Hans Moravec and Alberto Elfes. High resolution maps from wide angle sonar. In *1985 IEEE International Conference on Robotics and Automation. Proceedings.*, volume 2, pages 116–121. IEEE, 1985.

Jeremy M Morrey, Bram Lambrecht, Andrew D Horchler, Roy E Ritzmann, and Roger D Quinn. Highly mobile and robust small quadruped robots. In *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings.*, volume 1, pages 82–87. IEEE, 2003.

F Muniz, E Zalama, P Gaudiano, and J Lopez-Coronado. Neural controller for a mobile robot in a nonstationary environment. *CAS/CNS Technical Report Series*, (006), 1995.

R. Murphy, J. Casper, J. Hyams, M. Micire, and B. Minten. Mobility and sensing demands in usar. 1, 2000.

R Murphy, Jeffery Kravitz, S Stover, and Rahmat Shoureshi. Mobile robots in mine rescue and recovery. *Robotics & Automation Magazine, IEEE*, 16(2):91–103, 2009.

Robin Murphy, Jenn Casper, and Mark Micire. Potential tasks and research issues for mobile robots in robocup rescue. pages 339–344. 2001. URL `http://dx.doi.org/10.1007/3-540-45324-5_36`.

Robin R Murphy. Trial by fire [rescue robots]. *Robotics & Automation Magazine, IEEE*, 11(3): 50–61, 2004.

Robin R Murphy and Jennifer L Burke. Up from the rubble: Lessons learned about hri from search and rescue. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 49, pages 437–441. SAGE Publications, 2005.

Robin R. Murphy, Satoshi Tadokoro, Daniele Nardi, Adam Jacoff, Paolo Fiorini, Howie Choset, and Aydan M. Erkmen. Search and rescue robotics. pages 1151–1173. 2008.

Robin Roberson Murphy. Rats, robots, and rescue. *Intelligent Systems, IEEE*, 17(5):7–9, 2002.

Robert C Nelson. *Flight stability and automatic control*, volume 2. WCB/McGraw Hill, 1998.

Nils J Nilsson. A mobile automaton: An application of artificial intelligence techniques. Technical report, DTIC Document, 1969.

Fu Niu, Wei Hua Su, Jing Gong Sun, and Xin Yue Xu. The human-robot interaction: An investigation of rescue robot. *Advanced Materials Research*, 711:523–528, 2013.

Illah R Nourbakhsh, Katia Sycara, Mary Koes, Mark Yong, Michael Lewis, and Steve Burion. Human-robot teaming for search and rescue. *Pervasive Computing, IEEE*, 4(1):72–79, 2005.

Andreas Nuchter, Kai Lingemann, Joachim Hertzberg, and Hartmut Surmann. 6d slam - 3d mapping outdoor environments, 2007. ISSN 1556-4967. URL `http://dx.doi.org/10.1002/rob.20209`.

Jim Ostrowski and Joel Burdick. The geometric mechanics of undulatory robotic locomotion. *The international journal of robotics research*, 17(7):683–701, 1998.

Koichi Osuka, Robin Murphy, and Alan C Schultz. Usar competitions for physically situated robots. *Robotics & Automation Magazine, IEEE*, 9(3):26–33, 2002.

J. L. Paillat, P. Lucidarme, and L. Hardouin. Variable geometry tracked vehicle, description, model and behavior. 2008.

I.E. Paromtchik and C. Laugier. Autonomous parallel parking of a nonholonomic vehicle. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 13–18, 1996. doi: 10.1109/IVS.1996.566343.

Erwin Prassler, Arno Ritter, Christoph Schaeffer, and Paolo Fiorini. A short history of cleaning robots. *Autonomous Robots*, 9(3):211–226, 2000.

Tony J Prescott, Peter Redgrave, and Kevin Gurney. Layered control architectures in robots and vertebrates. *Adaptive Behavior*, 7(1):99–127, 1999.

Martin Proetzsch, Tobias Luksch, and Karsten Berns. Development of complex robotic systems using the behavior-based control architecture ib2c. *Robotics and Autonomous Systems*, 58(1): 46–67, 2010.

P Ridao, J Yuh, J Batlle, and K Sugihara. On auv control architecture. In *2000 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings.*, volume 2, pages 855–860. IEEE, 2000.

Rijans. 2013 savar building collapse, 2013. URL `http://en.wikipedia.org/wiki/2013_Savar_building_collapse`.

C. Samson. Control of chained systems application to path following and time-varying point-stabilization of mobile robots. 40(1):64–77, 1995. ISSN 0018-9286.

Uluc Saranli, Martin Buehler, and Daniel E Koditschek. Rhex: A simple and highly mobile hexapod robot. *The International Journal of Robotics Research*, 20(7):616–631, 2001.

Hagen Schempf, Edward Mutschler, Colin Piepgras, J Warwick, Brian Chemel, Scott Boehmke, William Crowley, Robert Fuchs, and Joshua Guyot. Pandora: autonomous urban robotic reconnaissance system. In *1999 IEEE International Conference on Robotics and Automation, 1999. Proceedings.*, volume 3, pages 2315–2321. IEEE, 1999.

Steve Shafer, Anthony Stentz, and Charles Thorpe. An architecture for sensor fusion in a mobile robot. In *1986 IEEE International Conference on Robotics and Automation. Proceedings.*, volume 3, pages 2002–2011. IEEE, 1986.

Eric Sholes. Evolution of a uav autonomy classification taxonomy. In *Aerospace Conference, 2007 IEEE*, pages 1–16. IEEE, 2007.

Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. Mobile robots. In *Advanced Textbooks in Control and Signal Processing*, pages 469–521. Springer London, 2009. URL `http://dx.doi.org/10.1007/978-1-84628-642-1_11`.

Roland Siegwart, Pierre Lamon, Thomas Estier, Michel Lauria, and Ralph Piguet. Innovative design for wheeled locomotion in rough terrain. *Robotics and Autonomous systems*, 40(2): 151–162, 2002.

Manuel F Silva and JA Tenreiro Machado. A historical perspective of legged robots. *Journal of Vibration and Control*, 13(9-10):1447–1486, 2007.

Reid Simmons, Richard Goodwin, Karen Zita Haigh, Sven Koenig, and Joseph O'Sullivan. A layered architecture for office delivery robots. In *Proceedings of the first international conference on Autonomous agents*, pages 245–252. ACM, 1997a.

Reid G Simmons, Richard Goodwin, Karen Zita Haigh, Sven Koenig, Joseph O'Sullivan, and Manuela M Veloso. Xavier: Experience with a layered robot architecture. *ACM Sigart Bulletin*, 8(1-4):22–33, 1997b.

National Institute Standards and USA Technology. Performance metrics and test arenas for autonomous mobile robots, 2012. URL `http://www.nist.gov/el/isd/testarenas.cfm`.

Alexander Stoytchev and Ronald C Arkin. Combining deliberation, reactivity, and motivation in the context of a behavior-based robot architecture. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation, 2001.*, pages 290–295. IEEE, 2001.

Shuli Sun. Designing approach on trajectory-tracking control of mobile robot. 21(1):81–85, 2005. ISSN 0736-5845. URL http://www.sciencedirect.com/science/article/pii/S0736584504000444.

BK Taylor, S Balakirsky, E Messina, and RD Quinn. Design and validation of a whegs robot in usarsim. In *Proceedings of the 2007 Workshop on Performance Metrics for Intelligent Systems*, pages 105–112. ACM, 2007.

Sebastian Thrun and John J. Leonard. Simultaneous localization and mapping. pages 871–889. 2008. URL http://dx.doi.org/10.1007/978-3-540-30301-5_38.

Sebastian Thrun, Wolfram Burgard, Dieter Fox, et al. *Probabilistic robotics*, volume 1. MIT press Cambridge, 2005.

Daniel Toal, Colin Flanagan, Caimin Jones, and Bob Strunz. Subsumption architecture for the control of robots. In *Proc. of the IMC*, volume 13, 1996.

Zhelong Wang and Hong Gu. A review of locomotion mechanisms of urban search and rescue robot. 34:400–411, 2007.

Greg Welch and Gary Bishop. An introduction to the kalman filter, 1995.

David A Williamson and Dale A Carnegie. Toward hierarchical multi-robot urban search and rescue: Development of a ÔmotherÕagent. In *Autonomous Robots and Agents*, pages 1–7. Springer, 2007.

Roger Woodman, Alan Winfield, Chris Harper, and Mike Fraser. Safety control architecture for personal robots: Behavioural suppression with deliberative control. In *The Seventh IARP Workshop on Technical Challenges for Dependable Robots in Human Environments*, 2010.

Brian G Woolley, Gilbert L Peterson, and Jared T Kresge. Real-time behavior-based robot control. *Autonomous Robots*, 30(3):233–242, 2011.

Lee Woosub, Kang Sungchul, Kim Munsang, and Shin Kyungchul. Rough terrain negotiable mobile platform with passively adaptive double-tracks and its application to rescue missions. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005.*, pages 1591–1596, 2005.

Keenan A Wyrobek, Eric H Berger, HF Machiel Van der Loos, and J Kenneth Salisbury. Towards a personal robotics development platform: Rationale and design of an intrinsically safe personal robot. In *IEEE International Conference on Robotics and Automation, 2008.*, pages 2165–2170. IEEE, 2008.

Huai-Xiang Zhang, Guo-Jun Dai, and Hong Zeng. A trajectory tracking control method for nonholonomic mobile robots. In *International Conference on Wavelet Analysis and Pattern Recognition, ICWAPR '07.*, volume 1, pages 7–11, 2007.