Traversability Estimation Techniques for Improved Navigation of Tracked Mobile Robots

Bijo Sebastian

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
In
Mechanical Engineering

Pinhas Ben-Tzvi, Chair
Corina Sandu
Andrew J. Kurdila
Ryan K. Williams

September 19, 2019
Blacksburg, Virginia

Keywords: Traversability Estimation, Disturbance Rejection Control, Terrain Estimation, Tele-operation, Tracked Robot, Vehicle Slip

TRAVERSABILITY ESTIMATION TECHNIQUES FOR IMPROVED NAVIGATION OF TRACKED MOBILE ROBOTS

Bijo Sebastian

ABSTRACT

The focus of this dissertation is to improve autonomous navigation in unstructured terrain conditions, with specific application to unmanned casualty extraction in disaster scenarios. Robotic systems are being widely employed for search and rescue applications, especially in disaster scenarios. But a majority of these are focused solely on the search aspect of the problem. This dissertation proposes a conceptual design of a Semi-Autonomous Victim Extraction Robot (SAVER) capable of safe and effective unmanned casualty extraction, thereby reducing the risk to the lives of first responders. In addition, the proposed design addresses the limitations of existing state-of-the-art rescue robots specifically in the aspect of head and neck stabilization as well as fast and safe evacuation.

One of the primary capabilities needed for effective casualty extraction is reliable navigation in unstructured terrain conditions. Autonomous navigation in unstructured terrain, particularly for systems with tracked locomotion mode involves unique challenges in path planning and trajectory tracking. The dynamics of robot-terrain interaction, along with additional factors such as slip experienced by the vehicle, slope of the terrain, and actuator limitations of the robotic system, need to be taken into consideration. To realize these capabilities, this dissertation proposes a hybrid navigation architecture that employs a physics engine to perform fast and accurate state expansion inside a graph-based planner.

Tracked skid-steer systems experience significant slip, especially while turning. This greatly affects the trajectory tracking accuracy of the robot. In order to enable efficient trajectory tracking in varying terrain conditions, this dissertation proposes the use of an active disturbance rejection controller. The proposed controller is capable of estimating and counter acting the effects of slip in real-time to improve trajectory tracking. As an extension of the above application, this dissertation also proposes the use of support vector machine architecture to perform terrain identification, solely based on the estimated slip parameters.

Combining all of the above techniques, an overall architecture is proposed to assist and inform tele-operation of tracked robotic systems in unstructured terrain conditions. All of the above proposed techniques have been validated through simulations and experiments in indoor and simple outdoor terrain conditions.

TRAVERSABILITY ESTIMATION TECHNIQUES FOR IMPROVED NAVIGATION OF TRACKED MOBILE ROBOTS

Bijo Sebastian

GENERAL AUDIENCE ABSTRACT

This dissertation explores ways to improve autonomous navigation in unstructured terrain conditions, with specific applications to unmanned casualty extraction in disaster scenarios. Search and rescue applications often put the lives of first responders at risk. Using robotic systems for human rescue in disaster scenarios can keep first responders out of danger. To enable safe robotic casualty extraction, this dissertation proposes a novel rescue robot design concept named SAVER. The proposed design concept consists of several subsystems including a declining stretcher bed, head and neck support system, and robotic arms that conceptually enable safe casualty manipulation and extraction based on high-level commands issued by a remote operator.

In order to enable autonomous navigation of the proposed conceptual system in challenging outdoor terrain conditions, this dissertation proposes improvements in planning, trajectory tracking control and terrain estimation. The proposed techniques are able to take into account the dynamic effects of robot-terrain interaction including slip experienced by the vehicle, slope of the terrain and actuator limitations.

The proposed techniques have been validated through simulations and experiments in indoor and simple outdoor terrain conditions. The applicability of the above techniques in improving tele-operation of rescue robotic systems in unstructured terrain is also discussed at the end of this dissertation.

DEDICATION

This dissertation is dedicated to

My parents: Sebastian Joseph and Baby Sebastian

My siblings: Bincy Sebastian and Joseph Sebastian

and

My wife: Nadiya Philip

ACKNOWLEDGEMENTS

It brings me great joy to present my doctoral dissertation titled “Traversability estimation techniques for improved navigation of tracked mobile robots”. I express my deepest gratitude to my advisor, Dr. Pinhas Ben-Tzvi, for his support and guidance throughout the duration of my research at the Robotics and Mechatronics Lab (RML), Virginia Tech. I would like to thank the members of my doctoral committee Dr. Corina Sandu, Dr. Andrew J. Kurdila, and Dr. Ryan K. Williams, for their constructive comments and input to my research. I would like to thank all of the faculty and staff of Virginia Tech who helped me in my work, directly and indirectly over the last four years.

I would also like to thank my current and former colleagues Dr. Anil Kumar, Eric Refour, Vinaykarthik Kamidi, Hailin Ren, Adam Williams, Shubhdildeep Sohal, Teja Vanteddu, Shumin Feng, Raghuraj Chauhan, Daniel Budolak, and all other current and previous lab members for their support and help during my stay at RML. I would like to thank all my friends, especially Dr. Savio Periera and Dr. Prasanth Valayamkunnath for making the last four years at Virginia Tech memorable. I would like to thank the Newman Catholic Ministry, especially Fr. David Sharland, for making me a part of the community. I want to thank my loving family, especially my wife Nadiya Philip; everything I have, I owe to your prayers, support and great love.

This work was supported by the US Army Medical Research & Material Command’s Telemedicine & Advanced Technology Research Center (TATRC), under Contract No. W81XWH-16-C-0062. The views, opinions, and/or findings contained in this report are those of the authors and should not be construed as an official Department of the Army position, policy, or decision unless so designated by other documentation.

TABLE OF CONTENTS

ABSTRACT .... **ii**

GENERAL AUDIENCE ABSTRACT .... **iv**

DEDICATION .... **v**

ACKNOWLEDGEMENTS .... **vi**

TABLE OF CONTENTS .... **vii**

LIST OF FIGURES .... **xii**

LIST OF TABLES .... **xvi**

LIST OF ABBREVIATIONS .... **xvii**

NOMENCLATURE .... **xix**

CHAPTER 1: INTRODUCTION **1**

1.1. BACKGROUND 1

1.2. SUMMARY OF CONTRIBUTIONS 2

1.3. DISSERTATION STRUCTURE 3

1.4. SELECTED PUBLICATIONS 5

CHAPTER 2: LITERATURE REVIEW **7**

2.1. BACKGROUND ON ROBOTIC SYSTEMS FOR SEARCH & RESCUE 7

2.1.1. Search robotic systems 9

2.1.2. Extraction robotic systems 12

2.1.3. Evacuation robotic systems 15

2.1.4. Robotic rescue competitions *17*

CHAPTER 3: PROBLEM STATEMENT AND PROPOSED SOLUTION............ 20

3.1. NOVEL ROBOTIC SOLUTION FOR CASUALTY EXTRACTION 20

3.1.1. Need for a rescue robotic system *20*

3.1.2. SAVER... *21*

3.1.3. Capabilities of SAVER *23*

3.1.4. Need for improving tracked locomotion in rough terrain *24*

3.2. TRACKED LOCOMOTION 25

3.2.1. Background on tracked locomotion *26*

3.2.2. Capabilities and challenges of tracked locomotion....................................... *26*

3.2.3. Summary of contributions towards autonomous navigation *28*

3.2.3.1. Physics based planning 28

3.2.3.2. Slip rejection control.. 28

3.2.3.3. Real-time terrain estimation... 28

3.2.3.4. Improving tele-operation of tracked mobile robots 29

CHAPTER 4: PHYSICS BASED PLANNING FOR TRACKED MOBILE ROBOTS.. 30

4.1. INTRODUCTION.. 30

4.2. REVIEW OF ROUGH TERRAIN PLANNING AND CONTROL TECHNIQUES 32

4.3. PROPOSED METHOD.. 36

4.3.1. Novelty of the proposed approach *39*

4.3.2. High-level planner *40*

4.3.3. Kino-dynamic aspect of planning *41*

4.3.4. Low-level controller 44

4.3.5. Implementation details 47

4.4. SIMULATION 49

4.4.1. Results and discussion 50

4.5. EXPERIMENTAL VALIDATION 52

4.5.1. Results and discussion 54

4.6. NEURAL NETWORK BASED TRAVERSABILITY ESTIMATION 57

4.6.1. Proposed technique 58

4.6.2. Data collection in simulated environment 60

4.6.3. Neural Network (NN) Architecture 63

4.6.4. Experimental Validation 64

4.6.5. Path planning application 66

4.7. CONCLUSION AND FUTURE WORK 68

CHAPTER 5: SLIP REJECTION CONTROL FOR TRACKED MOBILE ROBOTS 71

5.1. INTRODUCTION 71

5.2. REVIEW OF SLIP REJECTION CONTROLLERS 74

5.3. PROPOSED APPROACH 79

5.3.1. Generalized model 81

5.3.2. Proposed Estimation method 82

5.3.3. Low-level controller 84

5.3.4. Modification to the control effort 85

5.3.5. Novelty of the proposed approach 85

5.4. EXPERIMENTAL VALIDATION 86

5.5. RESULTS AND DISCUSSION 89

5.5.1. Improvement in path following 92

5.5.2. Estimation of the augmented parameters 93

5.6. CONCLUSION AND FUTURE WORK 97

CHAPTER 6: REAL-TIME TERRAIN ESTIMATION FOR TRACKED MOBILE ROBOTS 101

6.1. INTRODUCTION 101

6.2. REVIEW OF TERRAIN ESTIMATION METHODS 103

6.3. PROPOSED REAL-TIME TERRAIN ESTIMATION METHOD 106

6.3.1. Identifying the features of interest 107

6.3.2. Real-time estimation of weights 113

6.3.3. Application to terrain estimation 114

6.4. VALIDATION OF THE PROPOSED APPROACH 119

6.4.1. Experimental validation 120

6.5. APPLICATION TO AUTONOMOUS NAVIGATION 127

6.6. CONCLUSION 129

CHAPTER 7: IMPROVING TELE-OPERATION FOR TRACKED MOBILE ROBOTS 131

7.1. INTRODUCTION 131

7.2. REVIEW OF TELE-OPERATION TECHNIQUES 132

7.3. PROPOSED APPROACH 136

7.4. EXPERIMENTAL VALIDATION 140

7.5. RESULTS 141

7.6. FUTURE WORK 142

CHAPTER 8: CONCLUSION AND FUTURE WORK 144

8.1. SUMMARY 144

8.2. DIRECTIONS FOR FUTURE RESEARCH 144

8.2.1. Improvements in physics based planning 145

8.2.2. Improvements in slip rejection control 146

8.2.3. Improvements in terrain estimation 147

8.2.4. Improvements in tele-operation 148

8.2.5. Realizing safe and efficient casualty extraction in disaster scenarios 149

REFERENCES 150

LIST OF FIGURES

Figure 2.1: Search and rescue robots-timeline 8

Figure 2.2: Search robots: (A) Remotec Wolverine, (B) iRobot PackBot, (C) Quince, (D) Soryu III, (E) NIFTi UGV, (F) Foster-Miller Solem, (G) Inkutun VGTV-Xtreme, (H) Inkutun micro-VGTV and micro-tracks 10

Figure 2.3: Casualty extraction robots: (A) iRobot Valkyrie, (B) REX, (C) BEAR, (D) cRONA, (E) Modular rescue robot (Traction Robot), (F) Modular rescue robot............ 14

Figure 2.4: Casualty evacuation robots: (A) REV, (B) LSTAT on REV, (C) LSTAT with Snakebot manipulator, (D) Lockheed SMSS, (E) Qinetiq Titan, (F) HDT global protector 16

Figure 3.1: Subsystems of SAVER 21

Figure 3.2: Casualty extraction procedure using SAVER 22

Figure 3.3: SAVER carrying casualty 24

Figure 3.4: Failure of tele-operated tracked robotic systems. K. Massey, “Squad Mission Equipment Transport (SMET): Lessons Learned for Industry,” 2016 27

Figure 4.1: Block diagram representation of the proposed planner................................. 37

Figure 4.2: Flow chart showing the working of the proposed planning algorithm........... 38

Figure 4.3: Simulated scene showing the motion of tracked vehicle on rough terrain..... 42

Figure 4.4: Schematic showing the states of the robot, goal and the control input given to the system 45

Figure 4.5: Simulation results (a) Moderate terrain (b) Rough terrain 51

Figure 4.6: Experimental setup (a) Terrain with start and goal positions marked (b) STORM fitted with LOSA marker 53

Figure 4.7: Performance comparison of the proposed planner with kinematic planner over the experimental setup 55

Figure 4.8: Terrain topography map showing the 3D path followed by the robot under both the planners over the experimental setup.. 56

Figure 4.9: Overall layout of the proposed neural network architecture 58

Figure 4.10: Data collection through simulation: (A) Tracked platform with Kinect on simulated outdoor terrain, (B) Topography map of the terrain used to generate the dataset. Simulated indoor scenarios (C) Facing a wall, (D) Facing edge of platform, (E) Facing incline (F) Facing bumps 60

Figure 4.11: Variation of the training and validation accuracy over multiple epochs...... 64

Figure 4.12: Experimental validation in structured and unstructured terrain 65

Figure 4.13: Variation of traversability estimates over time in structured and unstructured terrains: (A) Facing a wall at an angle (Indoor), (B) Facing an edge of the platform (Indoor), (C) Going downhill (Outdoor), (D) Facing a wall (Outdoor). T denotes traversable and N denotes non-traversable 66

Figure 4.14: Path planning in unstructured terrain condition. The start and goal locations are marked in blue and green dots, respectively. The path followed by the robot is marked in a solid red line.. 68

Figure 5.1: Block diagram representation of traditional path tracking controllers........... 72

Figure 5.2: Block diagram representation of the proposed active disturbance rejection controller for path following in AGVs.. 81

Figure 5.3: Experimental setup STORM mounted with POZYX unit 87

Figure 5.4: Experimental validation STORM on different terrains. (a) Vinyl flooring, (b) Asphalt (c) Artificial Turf (d) Grass and gravel 88

Figure 5.5: Path tracking results on asphalt 90

Figure 5.6: Path tracking results on vinyl concrete flooring 91

Figure 5.7: Variation in value of the added parameters when the robot follows the path on asphalt under the action of the ADRC 94

Figure 5.8: Performance comparisons of both controllers over different terrain conditions 96

Figure 6.1: The absolute weights corresponding to each feature as obtained from the least squares fit 111

Figure 6.2: The first two PCs of the estimated weights. The figure shows (a) ground truth and (b) the classification results from the SVM. The magnitude of the first PC is along the X axis and the second along the Y axis 116

Figure 6.3: Normalized confusion matrix obtained from the trained SVM terrain classification 117

Figure 6.4: Flow chart showing the working of the proposed terrain estimation algorithm 118

Figure 6.5: Validation of the trained SVM over simulated terrain transition data with (a) estimated values of the PCs used by the SVM and (b) the terrain estimation results from the SVM 120

Figure 6.6: Experimental setup for the validation of the proposed terrain estimation technique on two different cases: (a) asphalt and vinyl concrete floor, (b) vinyl concrete flooring and grass-gravel 121

Figure 6.7: Validation of the trained SVM through experiments on asphalt and vinyl flooring with (a) PCs of the estimated values of the weights used by the SVM and (b) terrain estimation results along with ground truth 122

Figure 6.8: Validation of the trained SVM through experiments on vinyl flooring and grass-gravel with (a) PCs of the estimated values of the weights used by the SVM and (b) terrain estimation results along with ground truth 123

Figure 7.1: Overall layout of the proposed approach 135

Figure 7.2: Controller used to obtain user commands 137

Figure 7.3: Estimating traversability using physics simulation, the simulated robot model is shown in inset... 138

Figure 7.4: Terrain map being used for the validation... 139

Figure 7.5: The objects of interest used in the simulated validation (a) Human dummy (b) Fire (c) Medicine box... 140

Figure 7.6: Overall architecture to assist and inform teleoperation of tracked robotic modules 143

LIST OF TABLES

Table 5.1: Review of existing state-of-the-art controllers for handling slip..................... 76

Table 6.1: Model fitting result on each terrain condition 110

Table 6.2 Comparison of the proposed approach with existing state-of-the-art terrain estimation techniques... 126

Table 7.1: Quantitative results from simulation 142

LIST OF ABBREVIATIONS

SAVER	Semi-Autonomous Victim Extraction Robot
ADRC	Active Disturbance Rejection Controller
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
AUV	Autonomous Underwater Vehicle
BEAR	Battlefield Extraction-Assist Robot
REX	Robotic Extraction Vehicle
CASEVAC	Casualty Evacuation
REV	Robotic Evacuation Vehicle
LSTAT	Life Support for Trauma and Transport
S-MET	Squad Multipurpose Equipment Transport
ELROB	European Land Robot Trial
MedEvac	Medical Evacuation
IMU	Inertial Measurement Unit
DGC	DARPA Grand Challenge
PCA	Principal Component Analysis
DAE	Differential Algebraic Equations
URDF	Universal Robot Description Format
LOSA	Linear Optical Sensors Array
PID	Proportional-Integral-Derivative Controller
NN	Neural Network
ICR	Instantaneous Center of Rotation

SVM	Support Vector Machine
PC	Principal Component
EKF	Extended Kalman Filter
MPC	Model Predictive Control
OOI	Objects Of Interest

NOMENCLATURE

x, y, θ	Position (m) and orientation (rad) of the Robot
x_g, y_g	Position (m) of goal
V	Linear velocity of the robot (m/s)
ω	Angular velocity of robot (rad/s)
ω_l, ω_r	Left and right wheel angular velocity of the robot (rad/s)
D	Wheel diameter (m)
L	Width of the robot /axle track (m)
$\alpha_\mathrm{i}, \beta_\mathrm{i}$	Augmented parameters to account for slip
{R}	Robot fixed frame
{G}	Global inertial frame
$\mu_1 - \mu_{15}$	Additive zero-mean Gaussian noise
$\boldsymbol{K}$	Kalman gain
$\boldsymbol{\mathcal{X}}$	State vector
$\boldsymbol{\mathcal{Y}}$	Measurement vector
$\boldsymbol{S}$	State covariance matrix
$\boldsymbol{Q}$	Covariance matrix for process noise
$\boldsymbol{R}$	Covariance matrix for measurement noise
k_p, k_d	Proportional and Derivative gain of PID controller
V_{max}	Tunable parameter in velocity controller (m/s)
$\boldsymbol{U}$	Control input from PID controller
$\boldsymbol{U'}$	Modified control input from ADRC controller

CHAPTER 1

INTRODUCTION

1.1. Background

The focus of this dissertation is to improve autonomous navigation in unstructured terrain conditions, with specific applications to unmanned casualty extraction in disaster scenarios. Over the past few decades robotic systems have been used extensively for search and rescue applications, but the majority of the deployments were aimed for the search aspect of the problem. Very few systems have been proposed to address human rescue from disaster scenarios, and none of them have been tested in real field conditions. To this extent, this dissertation proposes a novel rescue robot design concept that aims to address some of the issues pertaining to existing state-of-the-art military and civilian rescue systems.

One of the primary capabilities needed for effective casualty extraction is reliable navigation in unstructured terrain. Autonomous navigation in unstructured terrain conditions, particularly systems with tracked skid-steer locomotion mode, offers unique challenges in path planning and trajectory tracking. To this extent, this dissertation proposes novel techniques to improve navigation of tracked robotic systems in challenging terrain conditions; specifically through the use of physics engines to account for the dynamics of robot-terrain interactions, use of active disturbance rejection techniques to counter the effects of slip, and use of trained support vector machine architectures to perform real-time terrain identification.

Even though autonomous operation of robotic systems in unstructured conditions has been an active field of research for the past few decades, it is rarely employed in disaster scenarios when human life is at risk. Based on the deployment history of search and rescue robotic systems it can be inferred that human guided or semi-autonomous operation based on high-level commands issued by a human tele-operator would be the preferred operation mode for the near future. The proposed use of physics engines for traversability estimation can also be applied to tele-operating robotic vehicles in challenging terrain conditions while reducing the cognitive load on the human operator. In addition, an overall architecture to employ the proposed slip rejection and terrain estimation techniques to aid tele-operation is also described at the end of the dissertation.

1.2. Summary of Contributions

The major contributions of this dissertation can be broadly summarized as follows:

1. A detailed review of existing search and rescue robotic systems is presented and the major limitations and remaining challenges are identified. In order to address the shortcomings of existing state-of-the-art robotic systems, a novel rescue robot design concept, the Semi-Autonomous Victim Extraction Robot (SAVER) is presented. An outline of the proposed conceptual victim extraction procedure is also described.
2. A detailed discussion on the popularity of tracked robotic systems for applications in challenging environments especially for search and rescue in disaster scenarios. The various challenges associated with tracked locomotion and the need for accurate traversability estimation and its improvement are described in detail.

3. The following techniques to improve autonomous navigation in challenging terrain conditions are proposed, specifically in path planning, trajectory tracking and terrain estimation.
    a. The use of a physics engine for accurate traversability estimations taking into account the dynamic-robot terrain interactions is explored.
    b. Design of an Active Disturbance Rejection Controller (ADRC) to estimate and counteract the effects of varying slip to improve trajectory tracking accuracy.
    c. The use of trained Support Vector Machine to preform terrain estimation solely based on the state evolution of the robot under known control inputs.
    d. The use of the above developed techniques to inform and assist in tele-operation of tracked robotic systems.

The contributions to improving autonomous navigation in challenging terrain conditions are discussed in detail in Section 3.2.3.

1.3. Dissertation Structure

The rest of the dissertation is organized as follows:

Chapter 2 presents a detailed review of existing state-of-the-art search and rescue robotic systems. Existing state-of-the-art rescue robotic systems are discussed in detail, focusing on their capabilities and shortcomings. The need for a novel casualty extraction robot is presented at the end of the chapter.

Chapter 3 presents a novel conceptual design of the Semi-Autonomous Victim Extraction Robot (SAVER) along with a conceptual description of the proposed casualty extraction method using SAVER. This chapter details the advantages of tracked locomotion techniques for robotic systems such as SAVER along with the various challenges that need to be addressed. The major contributions made by this dissertation towards improving autonomous tracked locomotion in unstructured terrain are also summarized.

Chapter 4 describes the need to account for the dynamic vehicle terrain interactions during path planning. The use of physics engines to model vehicle terrain interactions in order to enable reliable motion planning is described in detail. The proposed technique is validated through simulations and experiments and the results are presented. The use of physics engine simulations to create training data for neural network based traversability estimation is also presented.

Chapter 5 discusses the adverse effects of slip during trajectory tracking for tracked robotic systems. The need for estimating and compensating for the effects of slip to improve trajectory tracking is presented. An active disturbance rejection controller is presented to address the effect of slip by modifying the output of a low-level controller. The proposed technique is presented in detail along with the results of experimental validation from four common terrain conditions.

Chapter 6 presents the need for real-time terrain estimation to improve tele-operated and autonomous navigation of tracked robotic systems in challenging terrain conditions. Trained support vector machine architecture to perform terrain identification solely based on the state evolution of the robot for known control inputs is presented. The proposed technique is experimentally validated on two simple terrain transition cases. The results are discussed and the potential of using the above technique to improve planning, control and fault detection in autonomous tracked robots are also discussed.

Chapter 7 explores the applicability of above mentioned techniques to assist and inform tele-operation of tracked robotic vehicles. Specifically the applicability of physics based simulations to inform the tele-operator about possible failure of the robotic system in the near future is presented.

Chapter 8 concludes the dissertation by providing a summary of the presented work and a discussion about potential directions for future research.

1.4. Selected Publications

Disclosure: Content from these publications has been used throughout this dissertation.

Journal Articles

1. Sebastian, B., Ben-Tzvi, P., "Physics Based Path Planning for Autonomous Tracked Vehicle in Challenging Terrain," *Journal of Intelligent and Robotic Systems*, Springer, Vol. 95, Issue 2, pp. 511-526, August 2019.
2. Sebastian, B., Ben-Tzvi, P., "Active disturbance rejection control for handling slip in tracked vehicle locomotion", *Journal of Mechanisms and Robotics,* Transactions of the ASME, Vol.11, Issue 2, pp. 021003:1-12, April 2019.
3. Sebastian, B., Ben-Tzvi, P., “Support Vector Machine Based Real-time Terrain Estimation for Tracked Robots”, *Mechatronics*, Elsevier, Vol. 62, pp. 102260. October 2019.
4. Williams, A., Sebastian, B., Ben-Tzvi, P., "Review and Analysis of Search, Extraction, Evacuation, and Medical Treatment Field Robots", *Journal of*

Intelligent and Robotic Systems, Springer, February 2019. Available online: https://doi.org/10.1007/s10846-019-00991-6

5. Williams, A., Sebastian, B., Ben-Tzvi, P., "A Robotic Head Stabilization Mechanism for Medical Transport", *Robotics*, Multidisciplinary Digital Publishing Institute, Vol. 8, Issue 1, pp. 23, March 2019.

Conference Articles

1. Sebastian, B., Williams, A., Ben-Tzvi, P., “Gaussian Kernel Controller for Path Tracking in Mobile Robots”, Proceedings of the 2018 ASME IDETC/CIE, 42nd Mechanisms & Robotics Conference, Quebec City, Canada, Aug. 26-29, 2018.
2. Sebastian, B., Williams, A., Ben-Tzvi, P., “Control of a Head Stabilization System for Use in Robotic Disaster Response”, Proceedings of the 2017 ASME International Mechanical Engineering Congress and Exposition (IMECE 2017), Tampa, Florida, Nov. 3–9, 2017.
3. Sebastian, B., Hailin, R., Ben-Tzvi, P., “Neural network based heterogeneous sensor fusion for robot motion planning”, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2019), Macau, China, Nov. 3- 8, 2019.

Patents

1. Ben-Tzvi, P., Williams, A., Sebastian, B., Kumar, A., Saab, W., "Semi-Autonomous Victim Extraction Robot (SAVER)”, U.S. Provisional Patent Application No. 62/836,915, April 22, 2019.

CHAPTER 2

LITERATURE REVIEW

2.1. Background on Robotic Systems for Search & Rescue

One of the most impactful and exciting applications of robotic technology, is in the field of search and rescue. Many fatalities in the aftermath of disasters and combat are due to treatable traumatic injuries that can be avoided by timely medical treatment [1], [2]. This notion of time-sensitive treatment is represented by “The Golden Hour of Trauma” [3]–[6], the theory that if medical assistance is provided within a short time following traumatic injuries, the survival rate of the injured person rises appreciably. While the debate on the exact definition and duration of this critical period is unresolved in the medical literature, a mandate from the Secretary of Defense in 2009 to prioritize transporting military causalities in an hour or less resulted in a significant decline in mortality due to traumatic injuries, especially those requiring blood transfusions [7]. As hemorrhage due to major trauma has been found to be the cause of death in up to 80% of potentially survivable wounds in the U.S. military, timely evacuation and transportation must be emphasized when improving medical care [8]. This emphasis led to the U.S. Army Medical Research and Material Command reopening investigation in this field [9]. While rapid medical assistance dominates the focus in the reduction of traumatic field injuries, the risks associated with first responders involved in victim handling procedures must not be ignored [10]. In disaster or combat scenarios, deployment of a medic or other rescue personnel into a dangerous area risks the lives of both the rescuer and the injured. Furthermore, during terror attacks or military operations there may be “leave behind” explosives or enemy troops targeting first responders [11]. Moreover, manmade catastrophic events often occur in remote locations, making it difficult to send qualified personnel. In the above cases, robots can make significant contributions to saving the lives of both the injured personnel and responders.

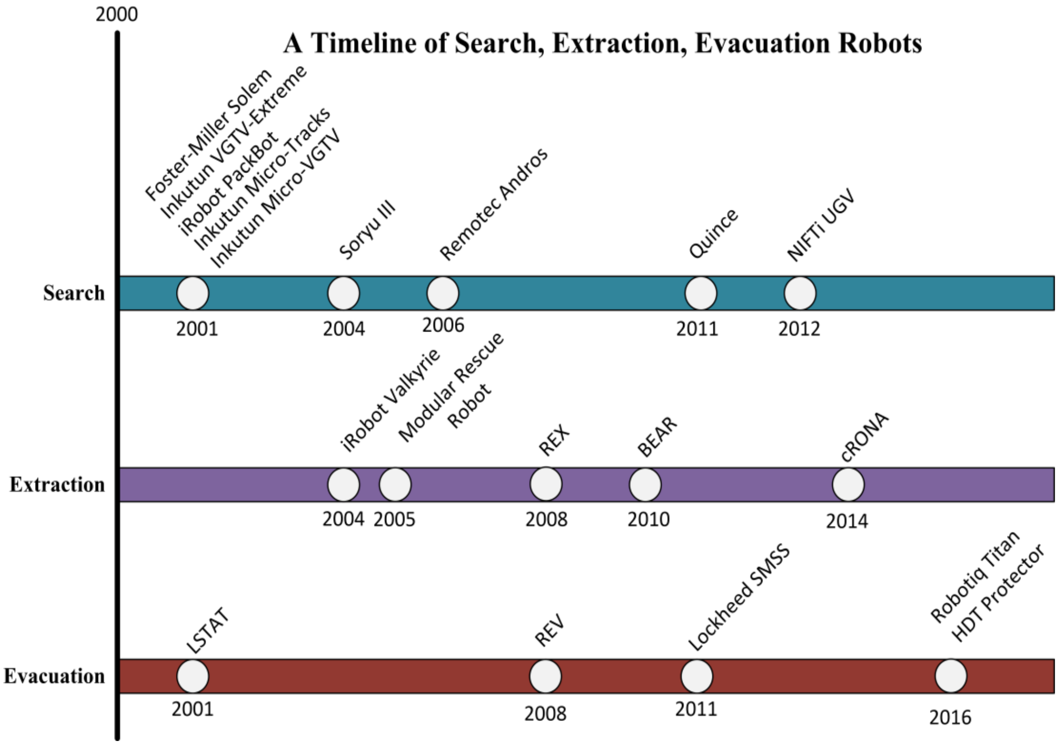


Figure 2.1: Search and rescue robots-timeline

Robots present an opportunity to go where rescuers cannot, keep responders out of danger, work indefatigably, and augment the capabilities of the humans who put their lives at risk while helping others. Over the last few decades a lot of research is being done towards search and rescue applications of robotic systems. Despite the general categorization of existing state of the art systems as “search and rescue”, not all of them are equipped to conduct human rescue operations independently. Instead a majority of it is aimed specifically at the search aspect of the problem, where robotic systems attempt to find and report the location of any injured personnel. Very few systems have been designed with the capability of extraction and evacuation of human casualty. A timeline of the implementation and/or testing of these robots as described in the literature are shown in Fig. 2.1. The following section provided a summary of state-of-the-art systems in each of the individual categories along with the aspects that need further improvement before safe and efficient human casualty rescue can be realized.

2.1.1. Search robotic systems

Search robotics is a mature field, and robots have been actively incorporated into search procedures in numerous disasters as far back as the September 11 attacks in 2001 [12]. In this section, I provide a brief review of some of the most notable, existing search robots to provide a frame of reference for the reader. For the sake of brevity, only few successfully implemented and field-tested ground robotic systems shown in Fig. 2.2 are reviewed here. For a detailed review of search robots, the reader can refer to [13], [14]. Even though most of the systems discussed below are classified as “search and rescue” systems in literature, here they only occupy the search category, as they alone cannot facilitate the extraction and evacuation of an injured personal from a disaster scenario.

Historically, a large amount of attention has been to the robotic systems that belong to the search class. Spurred by the close succession of the catastrophic 1995 Oklahoma City bombing and Kobe earthquake [15], unmanned aerial vehicles (UAVs), unmanned ground vehicles (UGVs), and autonomous underwater vehicles (AUVs) with search capabilities have all been deployed with some capacity in response to well-known disasters. Robot-human teams were deployed to probe the rubble of the World Trade Center following the attacks on September 11th in 2001, UAVs were used to assist in the search for those trapped by the flooding resulting from Hurricane Katrina in 2005, and an

AUV surveyed the damage to the Rollover Pass Bridge caused by Hurricane Ike in 2008. While in Japan, mobile robots such as Quince were utilized to measure the radiation in the aftermath of the Fukushima nuclear disaster in 2011 [14], [16]–[18].

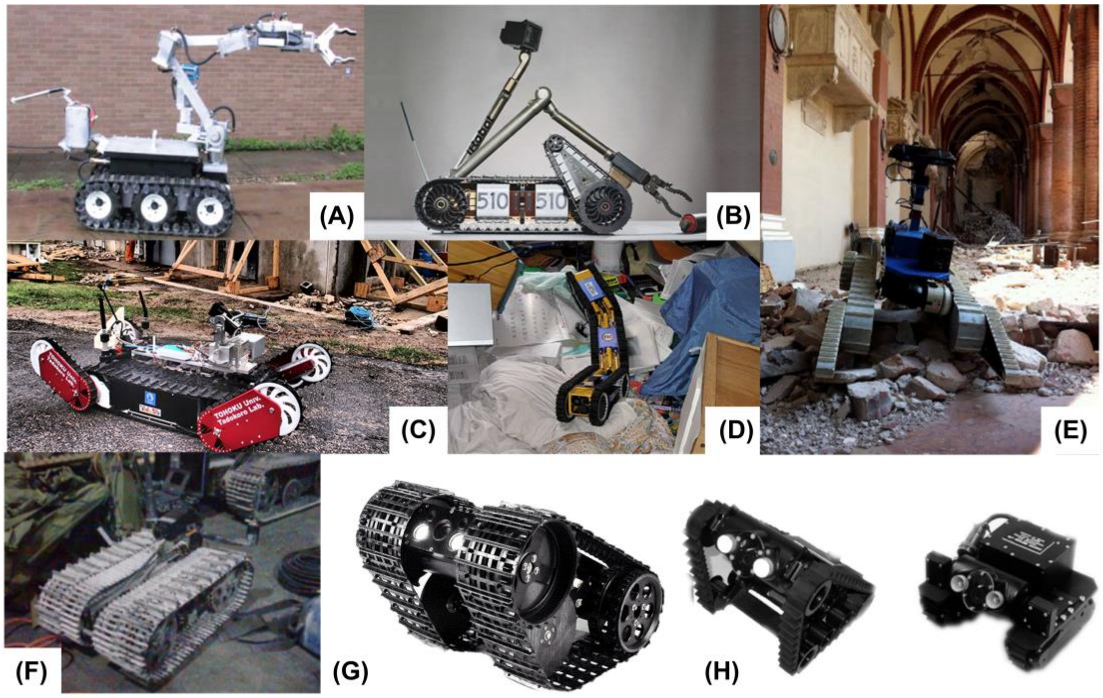


Figure 2.2: Search robots: (A) Remotec Wolverine, (B) iRobot PackBot, (C) Quince, (D) Soryu III, (E) NIFTi UGV, (F) Foster-Miller Solem, (G) Inkutun VGTV-Xtreme, (H) Inkutun micro-VGTV and micro-tracks

Robots have the ability to play an integral part in surveying affected regions and locating people in distress during the aftermath of disasters or combat. Search robots are generally designed to act as mobile sensory platforms that perform small crucial tasks which enable the use of sophisticated detection equipment in spaces that may be unsafe or unreachable for humans [19]. Furthermore, with proper design, robots can run continuously, with just a momentary stop for refueling, facilitating nonstop search efforts and allowing human team members to divide shifts more effectively. This would mitigate the risks of sleep depreciation related errors in the operation of complex technology, a major issue in search and rescue efforts [20].

Based on the deployment history, operation modes of the search robotic systems can be divided into three major categories: surveying the scenario to estimate the extent of damage and the stability of structures, collecting data for post processing (such as 3D maps of the interiors of buildings), and looking for potentially injured persons. In order to perform the above functions effectively, the robots are designed to be small (man-packable), agile, and requiring only a small degree of supervision from the human operator. Most of the commonly used UGV systems were initially developed for military purposes such as EOD. However, these robots have been modified for search and rescue to become much smaller than their corresponding military systems, so that they can fit into openings that people and dogs cannot enter.

Robots can now provide a remote presence for rescuers in areas that are physically inaccessible or unsafe, while also allowing the rescuers to “sense and act at a distance” [13]. In comparison to existing active or semi-active articulated cameras used for similar tasks, robots travel further into the rubble while also interacting with the environment such as by taking samples or closing valves via a manipulator. Additional capabilities, such as the ability to work indefinitely without tiring and the use of a wide array of sensors to detecting toxic or explosive gasses in the environment, make robots better equipped for search activities than humans or animals. Above all, robots should be used in scenarios where there is a risk to the life of the rescuers. Thus, key factors in the advantage of robots over other search systems are their terrain adaptability and ability to interact with the environment. With the advancement of research, we can have robot-human teams that allow for faster coverage of the disaster environment, allowing for better allocation of resources and a collaborative system that performs better than the sum of its parts. With regards to search, an effective mobile search robot is one that possesses the ability to adapt to the necessary terrain. This adaptability lends the robots the flexibility required for successful field deployment. Ongoing research in this area focuses on the use of these robots in autonomous and semi-autonomous multi-robot teams, in order to effectively search over a large area. Further work is required on methodologies and machine intelligence required for the robots to not only operate in tandem with other robots, but alongside search and rescue personnel with minimal training. Human-robot cooperation could vastly increase the usability and benefit imparted by search robots, and would help to further their implementation by search and rescue teams.

2.1.2. Extraction robotic systems

Extraction systems are robots capable of physically maneuvering or carrying the injured person out of the disaster zone. These ‘rescue’ robots are by necessity larger and at times, more complex, than their search-focused brethren.

Extraction of a wounded person using a robot is a complex task due to the necessary interaction between a robot and an injured or possibly incapacitated person. Recent advancements in sophisticated actuation and control systems over the last 10-15 years have led to expanded efforts into robotic extraction. This area is less mature than search robotics and is not widely discussed in literature. Some of the state-of-the-art casualty extraction robotic systems are shown in Fig. 2.3.

Systems such as the Battlefield Extraction-Assist Robot (BEAR) and the Robotic Extraction Vehicle (REX) are indicative of these types of robots [21], [22]. iRobot

Valkyrie: One of the earliest solutions to the robotic Casualty Evacuation (CASEVAC) question was created by iRobot in 2004, called the Valkyrie [23]. Funded by the Army Telemedicine and Advanced Technology Research Center (TATRC), it had evolved from iRobot’s earlier medic robot, Bloodhound. Essentially a modification of the company’s man-packable UGV Packbot, it consisted of a flexible stretcher, called a Sked, which was tethered to the robot. The intent was that when rescuing a casualty in a combat zone, a medic could remotely operate the Packbot to drive out to the casualty, where the injured soldier could roll onto the stretcher and be pulled to safety. The Robotic Extraction and Evacuation platform is a “marsupial” robotic system. The larger and faster Robotic Evacuation Vehicle (REV) transported a smaller Robotic Extraction (REX) rover that would then be deployed near the extraction site [22]. The system was designed by Applied Perceptions Inc. in collaboration with TATRC.

One of the most promising CASEVAC robots to date is the Battlefield Extraction Assist Robot, built by Vecna Robotics [21], [24]. It is a semi-anthropomorphic tracked robot designed with an emphasis on agility and maneuverability. The extraction procedure is simple: the BEAR approaches the wounded soldier, slides its arms under the wounded soldier, and then carries them to safety. In a similar vein as BEAR, cRONA is a humanoid robot that uses two arms to lift up an injured person and carry them to safety while utilizing tracks for locomotion [25].

In addition to the designs considered above, there have been several other notable attempts at creating a casualty extraction robot. One such design was a modification of Foster-Miller’s widely used military robotic platform TALON [26]. The concept was to use an attachment to the robot consisting of an arm terminating in a conveyor belt meant to slide under a wounded person, then lift and transport them [27]. However, no further work was performed beyond the initial report. Another notable design was a modular patient transport system designed to help those effected in case of nuclear emergency [28].

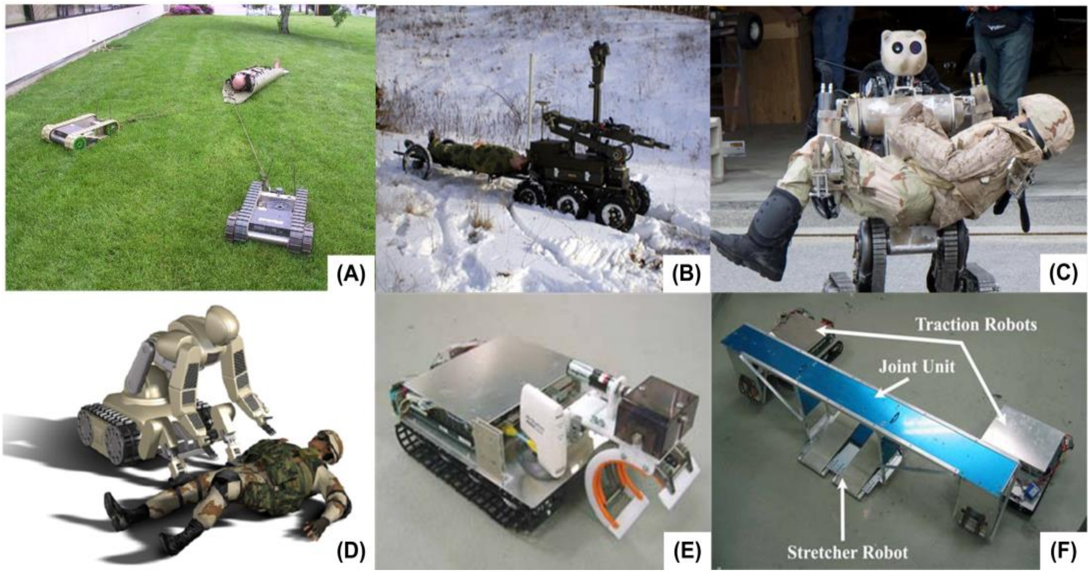


Figure 2.3: Casualty extraction robots: (A) iRobot Valkyrie, (B) REX, (C) BEAR, (D) cRONA, (E) Modular rescue robot (Traction Robot), (F) Modular rescue robot

In the area of casualty extraction robots, one of the critical challenges is keeping the injured person safe and secure during transport. One of the more difficult tasks in this operation occurs when attempting to transfer the injured or incapacitated person from the ground to the transport platform. By necessity, patient transfer requires some combination of lifting, dragging, or sliding, and current designs fail to place sufficient emphasis on maintaining a stable transfer mode. Further attention to this problem could reduce the danger of exacerbating any existing injuries or causing new ones. In addition, the existing systems all rely on direct, continuous operator control, which may face challenges when operating in remote locations with poor network infrastructure. A robotic platform with semi-autonomous communications and lag-compensating control could help ensure the safety of the injured person when communications are intermittent.

2.1.3. Evacuation robotic systems

Once an injured person has been safely extracted from the point-of-injury the next step is often to transport them to a more secure medical station for first aid and/or triage or more in-depth medical attention. To provide an improvement on the existing manual approaches, such a robotic system can be used with some degree of autonomy while providing feedback on the injured person's current state. Research and development in this area has largely been focused on the creation of a larger, multi-purpose, mobile ground vehicle that has configurable modules to facilitate the evacuation of injured personnel and the peripheral systems intended to provide onboard patient monitoring in such operations [29]. The robots that are designed to perform this task include the Life Support for Trauma and Transport (LSTAT), a stretcher with a full set of sensory equipment and a robotic snake-like manipulator, [27], and the Robotic Evacuation Vehicle, a mobile patient transport robot [22], as shown in Fig. 2.4.

Even though not a mobile robot, Integrated Medical Systems Inc.'s LSTAT patient care platform [27] is major effort in this direction. While appearing to be simply a stretcher, it possesses enough capabilities to be a mobile Intensive Care Unit. In addition to the stretcher itself the LSTAT consists of a ventilator, a defibrillator, a suction pump, a fluid and drug infusion pump, and a blood chemistry analyzer [30]. It also carries sensors that monitor blood pressure, pulse oximetry, end-tidal CO_2, temperature, oxygen flow, and electrocardiography. The patient data is shown on a display mounted on the stretcher, and broadcast to a hand-held monitor or available wireless networks. As previously described, the REV, is the larger transport half of the marsupial pair REX and REV.

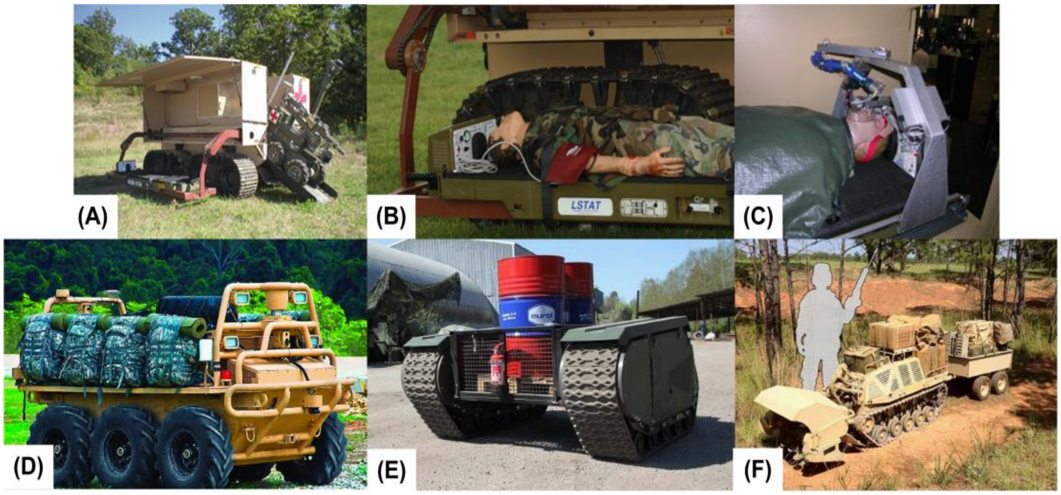


Figure 2.4: Casualty evacuation robots: (A) REV, (B) LSTAT on REV, (C) LSTAT with Snakebot manipulator, (D) Lockheed SMSS, (E) Qinetiq Titan, (F) HDT global protector

Upon reaching a combat zone, REV would deploy a ramp and send REX into the field to extract a wounded solider. Once retrieved, REV would act as an autonomous, reconfigurable transport vehicle equipped with two LSTAT stretchers and ballistic armor, in order to safely evacuate the wounded soldiers [22]. The Squad Multipurpose Equipment Transport (S-MET) program is a U.S. Army initiative intended to drive development of an autonomous or semi-autonomous mobile robot that transports the supplies required by an infantry squad to operate for 72 hours and provides a mobile power source to recharge the electronics carried by the soldiers [31]. These mobile robots would have manual operation, follow-the-leader, and autonomous navigation capabilities. In addition to the increased load carrying capabilities afforded by the S-MET, they must also be reconfigurable into casualty evacuation platforms, either through attachment points for a standard stretcher or through inherent medical transport capabilities [32].

In disaster situations, the medical personnel generally respond after the primary danger has subsided, and thus require shorter operating ranges for their equipment as medical treatment centers can be located near the disaster zone. A purpose built evacuation platform would not be used often enough to justify the inclusion of such a large piece of equipment in a squad loadout. Therefore, the overly specific REV has been supplanted by the more versatile pack mule-like S-METs. This provides operational flexibility while still providing evacuation capabilities if necessary. However, this removes some of the patient-care specific benefits that REV incurred through the incorporation of the LSTAT into its design. The desire for a more compact and modular solution led to the creation of the LS-1.

2.1.4. Robotic rescue competitions

Many major robotics competitions have featured “medical assistance and extraction” as the central theme, a part of their overall challenge, or as an event for demonstration purposes. Some of these competitions include The European Land Robot Trial (ELROB), euRathlon, RoboCup Rescue, and the Darpa Robotic challenge, with the ELROB being the closest towards replicating the real life search, extraction, evacuation and treatment challenges.

The ELROB is a robotics competition that has been conducted every alternate year since 2006, focusing on military and civilian applications of advanced robotic systems. [33]. In the recent competitions (starting in 2014), search and rescue scenarios such as locating injured personnel inside collapsed structures and performing medical evacuations (MedEvac) have been included in ELROB. For the MedEvac challenge, two dummies representing wounded soldiers are hidden in non-urban terrain. Their approximate location was supplied to the team. The participant then had to locate the wounded ‘soldier’ and extract them to a base location, within a specified time limit. During the 2014, 2016 and 2018 ELROB, many major institutions proposed innovative solutions to the above challenges.

The RoboCup Rescue competitions were initiated as a part of the worldwide RoboCup competition in 2000 [34]. These include both the Rescue Robot League and the Rescue Simulation League [35], [36]. The Rescue Robot League involves exploring and searching for simulated casualties within an arena, including subtasks such as mapping, remote manipulation, and autonomous operations. The tasks, including the test environment, are based on the standard test methods for emergency response robots developed by the U.S. National Institute of Standards and Technology [37]. DARPA started robotics competitions in 2004 with the Grand Challenge [38]. Inspired by the Fukushima Daiichi nuclear disaster, the DARPA Robotics Challenge (DRC) had Urban Search and Rescue as the core theme fostering research on robots capable of assisting humans in response to natural and manmade disasters. A major focus of the DRC was to develop ways to combine the complementary strengths and weaknesses of the robot system and human operator(s). Even though the competitions required humanoid robots to perform complex tasks like driving a utility vehicle, opening a door, handling valves [39]–[42] etc., it did not involve any direct casualty extraction or evacuation challenges.

The EU-FP7 euRathlon project was a three-year initiative funded by the European Commission, started in 2013. As an international competition, it welcomes universities, industries, and independent teams from any EU country. The Grand Challenge, conducted on September 2015 [43], [44], was inspired by the Fukushima accident of 2011, providing real world challenges focused on outdoor robotics.

CHAPTER 3

PROBLEM STATEMENT AND PROPOSED SOLUTION

3.1. Novel Robotic Solution for Casualty Extraction

Although no single system or group of systems exists to date that can do robotic search, extraction, and evacuation, as depicted by the progression of the systems above, the global research community is moving towards integrating co-robotic teams with the aim of achieving this ultimate goal, in tandem with human searchers as well as multimodal robots.

3.1.1. Need for a rescue robotic system

In addition to the above improvements in each of the individual facets of search, extraction and evacuation performed by robotic systems, several additional optimization points present themselves when considering the design of a rescue robot as a whole. An important one is to emphasize the stabilization of the head and neck in transport to minimize further injury to the cervical spine. In the robots reviewed above, this is not addressed purposefully in any design. At best, the placement of a cervical collar by the robot is mentioned in passing in the patents. A second area in which focus can be directed towards is the issue of creating a well-balanced all terrain mobility platforms. BEAR was one of the most complete of the designs discussed, but the tracked system coupled with the height of the robot adds complexity. Finally, simplified operational complexity should be a key goal without leaving out functionality, such as the simplified function offered with Valkyrie. These robots would be deployed in some of the most dangerous areas on the planet, whether in a war zone or a disaster area, and as such should be easy and intuitive to operate. Scaling back on the degrees of freedom and making the operator control unit straightforward to control could save precious time when attempting a rescue.

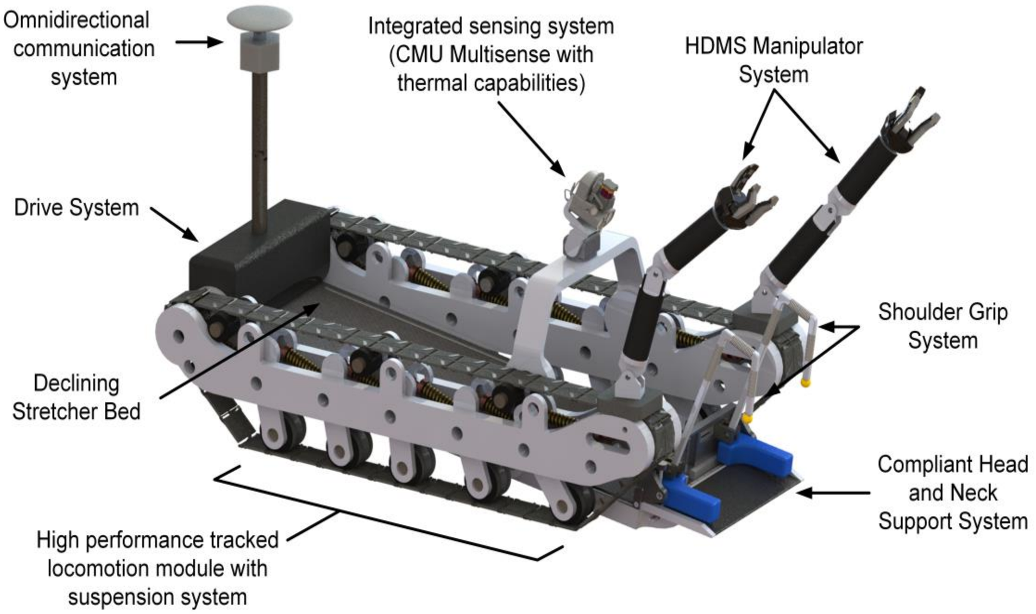

Omnidirectional communication system
Integrated sensing system (CMU Multisense with thermal capabilities)
HDMS Manipulator System
Drive System
Shoulder Grip System
Declining Stretcher Bed
Compliant Head and Neck Support System
High performance tracked locomotion module with suspension system

Figure 3.1: Subsystems of SAVER

3.1.2. SAVER

Based on the review of the state-of-the-art systems for robotic rescue and the analysis of their shortcomings [45], a novel rescue robot design was conceived: the Semi-Autonomous Victim Extraction Robot (SAVER)[46]. The proposed conceptual design was developed at the Robotics and Mechatronics Lab at Virginia Tech, in collaboration with RE^2 Inc. funded by the U.S. Army Telemedicine and Advanced Technology Research Center. The proposed system was designed for an average soldier in full battle gear weighing around 135 kg (300 lbs), with a height of approximately 6 feet. The system is designed for an overall size of 2.21m x 1.2m x 1.25m (L x W x H), with a total curb weight of less than 180 kg. The proposed conceptual design of SAVER along with its subsystems is shown in Fig. 3.1.

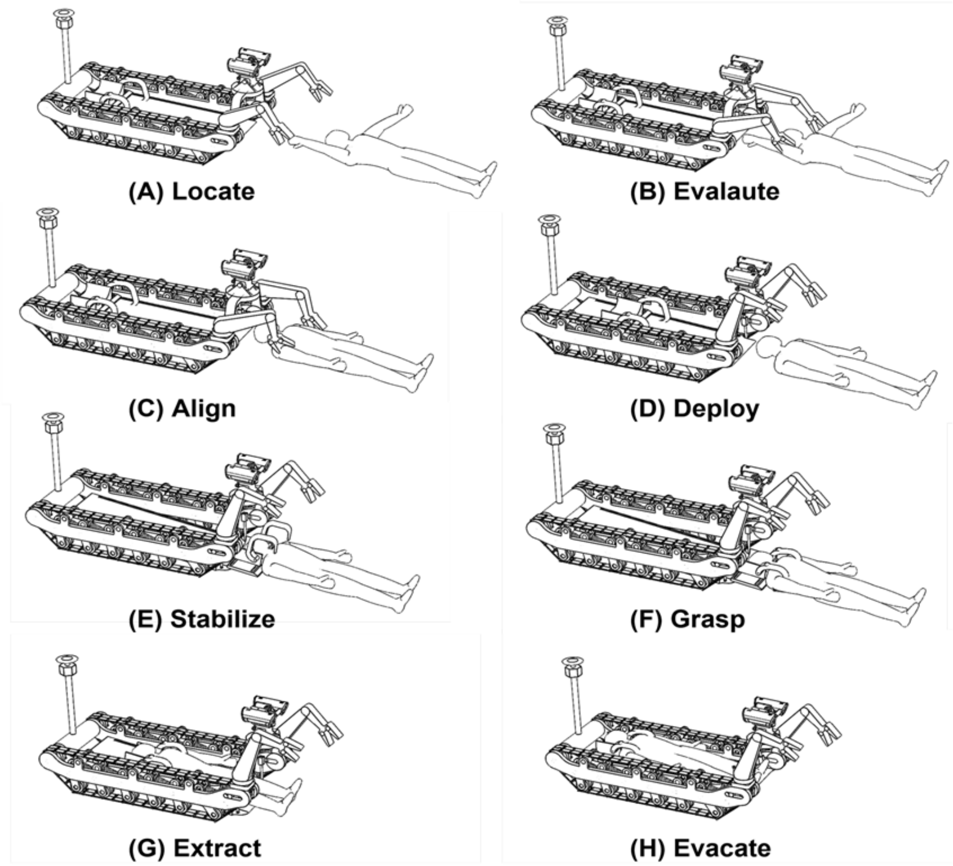


Figure 3.2: Casualty extraction procedure using SAVER

The casualty extraction procedure using SAVER can be summarized as follows: the rescue robot will be brought in to the scene using external means like the SMSS or air dropped into the disaster scenarios using a helicopter, similar to the marsupial concept used by REX/REV systems. Launched within range of the disaster scenario, the SAVER system will locate the injured, drive up to the person, estimate the posture and then align the person so that he/she can be easily transferred on to the stretcher. The head support system slides down the stretcher and stabilizes the head and neck of the person and then engages the shoulder support hooks. The injured person is then slowly pulled on to the declined stretcher. The overall procedure is depicted below in Fig. 3.2.

3.1.3. Capabilities of SAVER

In order to successfully execute the above-mentioned procedures, the robot is designed to be a semi-autonomous system. Navigating towards the already located injured personnel and then evacuating the extracted casualties to the triage zone will be done autonomously. This require the robot to be able to navigate autonomously in rugged terrain, taking into account the challenges introduced by dynamic robot-terrain interactions, initial efforts in this direction show promising results [47]. In addition the project will explore ways to enable the robot follow a field medic so as to enable co-operative behaviors with the system. Handling injured personnel fully autonomously in unstructured terrain is still beyond the state-of-the-art in robotics. As such, the SAVER system is designed to do this with the help of a remote operator using the HDMS developed by RE^2. With an extensive sensor suite providing real time visual and force feedback, the operator will be able to successfully manipulate the injured person into the right posture using the dual arm manipulation system. In addition a robotic head stabilization system that autonomously stabilizes the head and neck of a patient has been fully designed, built and tested to guarantee desired degree of performance [48]. A simulated casualty extraction scenario using SAVER is shown in Fig. 3.3.

Conducting the human rescue process in a fully autonomous manner will require further development in the fields of machine intelligence and human robot interactions. Instead, advancement human-robot cooperative teams that employ human in the loop control, where the human operator makes high-level decisions and the robotic system interprets the high-level commands to perform the dangerous rescue operation is a more effective and feasible solution.

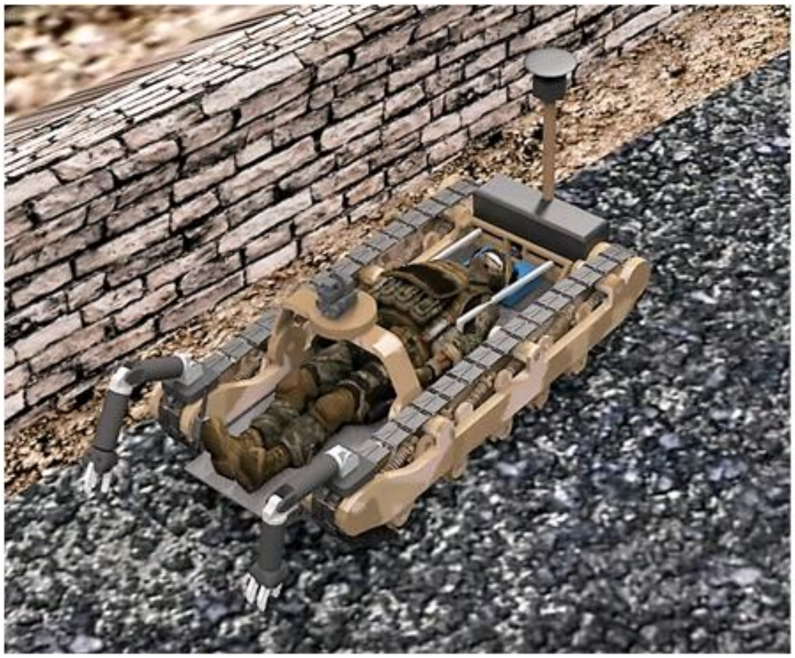


Figure 3.3: SAVER carrying casualty

3.1.4. Need for improving tracked locomotion in rough terrain

In addition to developing the detailed conceptual design of SAVER, ongoing research focuses on developing and testing each of the subsystems and their capabilities. To this extent a prototype model of the proposed head support system was built and tested on a human mannequin [45], [48], [49]. In addition, sensing techniques have been developed and tested for enabling obstacle avoidance to aid in autonomous navigation [50] and human pose estimation to aid in casualty rescue [51].

As mentioned in the previous section, one of the major milestones on the road to safe and efficient robotic casualty extraction is reliable autonomous navigation in challenging terrain conditions. The ensuing conditions following natural or manmade disasters often involve poor communication channels between the affected region and the outside world, with limited bandwidth and increased latency. The use of tethered rescue systems in the past, allowing for a robust communication channel, has had varying degrees of success [12], [15], [19], [52]. Unfortunately, the use of a tether limited the mobility of the robots and introduced the risk of the tether becoming stuck in the rubble. Owing to the critical nature of rescue missions, remote operation of a rescue robotic system in an unstructured volatile environment is a challenging task, even with a communication tether. This drives the need for higher-level autonomy in rescue robots, such as the ability to navigate on their own over challenging terrain conditions in a reliable manner. The recent call by the US Army [9], [53] for unmanned casualty evacuation platforms and the introduction of CasEvac/MedEvac scenarios in robotic benchmarking competitions like El-Rob [13] reinforce the growing need for such autonomous systems.

Assuming the location of the injured casualty is available, the robotic platform must be able to navigate to the desired location relying only on high level commands issued by the remote operator or field medic. The major focus of my dissertation has been to estimate and improve traversability of tracked robotic systems including SAVER in challenging terrain conditions.

3.2. Tracked locomotion

Autonomous navigation is a ubiquitous task for mobile robots. Great advances have been made in this domain in the past few decades, as demonstrated by the recent developments in self-driving vehicles, warehouse automation, and even smart vacuum systems that are now a common part of the households. Despite these advances, autonomous navigation in its true sense is still an open research problem for many challenging conditions such as tracked locomotion in unstructured terrain.

3.2.1. Background on tracked locomotion

Tracked vehicles were first created to facilitate navigation over a variety of ground conditions such as snow, loose sand, mud, steep slopes, terrain cluttered with rubble or any combination of these (from here on referred to as rough terrain) that is otherwise not feasible for conventional wheeled vehicles. Such vehicles are often the best choice for applications such as hauling heavy military equipment or agricultural operations that require a significant amount of traction. The superiority of tracked locomotion over wheeled systems in such scenarios is due to its increased traction and comparatively lower ground pressure. Based on the above factors, tracked locomotion is often considered best suited for search and rescue applications [54] where terrain conditions are often treacherous and the environment very unstable. This is demonstrated in practice as well, as the majority of the search and rescue robots that have been deployed in the field over the past few decades use tracks as their primary method of locomotion [13], [14], [16], [17].

3.2.2. Capabilities and challenges of tracked locomotion

Tracked locomotion systems come with unique challenges of their own. Owing to their inherent mechanical advantage, tracked systems can go over many obstacles. On the other hand there are cases where the robot can tip over or get stuck. Several factors such as the characteristics of the terrain in terms of slip, slope, and soil properties, characteristics of the robot in terms of weight and moment of inertia, actuator limitations, and the nature of the track profile all play major role in traversability. As such trivial techniques on estimating traversability based on presence or absence of obstacles will be overly conservative for tracked robots. In other words, Vehicle-terrain dynamic interaction needs to be taken into consideration in order to accurately estimate and improve tracked locomotion.

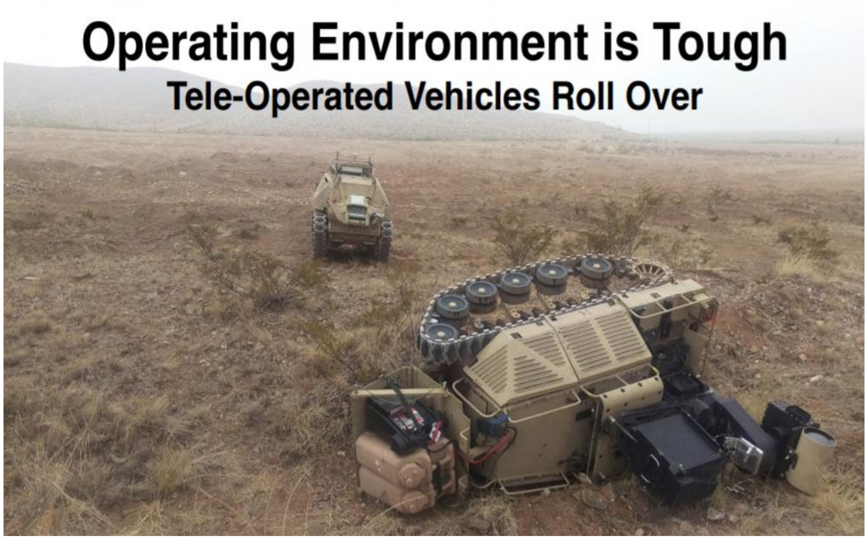

Operating Environment is Tough
Tele-Operated Vehicles Roll Over

Figure 3.4: Failure of tele-operated tracked robotic systems. K. Massey, "Squad Mission Equipment Transport (SMET): Lessons Learned for Industry," 2016

The most widely used navigation modes for search and rescue system especially for military application are tele-operation, semi-autonomous leader-follower navigation and fully autonomous waypoint navigation. Among these modes the most widely used is the fully autonomous waypoint navigation. Even with the recent advancements in autonomous capabilities of robotic systems, state-of-the-art robotic systems cannot match the decision making capabilities of a human operator especially for applications in disaster scenarios. But even tele-operated systems can fail under rough terrain conditions as shown in Fig. 3.4. This dissertation aims to improve traversability of tracked robots on challenging terrain conditions under both autonomous and semi-autonomous operating modes.

3.2.3. Summary of contributions towards autonomous navigation

The specific contributions of this dissertation in improving autonomous tracked navigation in challenging terrain can be summarized as follows:

3.2.3.1. Physics based planning

The use of physics engines for accurate traversability estimations taking into account the dynamic-robot terrain interactions is explored. Improvement in path planning brought out by the traversability estimations is validated through experiments and simulations. The use of physics engine simulations to generate training data for neural network based traversability estimations is also explained in detail.

3.2.3.2. Slip rejection control

An Active Disturbance Rejection Controller (ADRC) has been designed to estimate and counteract the effects of varying slip to improve trajectory tracking. The performance of the proposed controller was compared to that of a tuned PD controller on four different terrain conditions. Quantitative improvement in performance realized by the proposed controller design is also summarized and discussed.

3.2.3.3. Real-time terrain estimation

The use of trained Support Vector Machine to preform terrain estimation solely based on the state evolution of the robot under known control inputs is presented. Possible application of real-time terrain estimation to improve path planning and control for autonomous navigation of tracked robots is also presented.

3.2.3.4. Improving tele-operation of tracked mobile robots

In addition to improving autonomous navigation, the above mentioned techniques can also reduce the risk of mission failure associated with tele-operation. Tele-operated systems often fail due to roll/tip over and collisions with the environment resulting from limited environmental awareness on the part of the tele-operator. Physics engine based traversability estimation, slip rejection and terrain estimation techniques can be used to assist and inform tele-operation, reducing risk of failure.

Each of the above contributions is discussed in detail in the following chapters.

CHAPTER 4

PHYSICS BASED PLANNING FOR TRACKED MOBILE ROBOTS

4.1. Introduction

Determining a feasible path that will take a robot from the starting point to a goal location is a major step in successful autonomous navigation. Much of the existing research towards autonomous ground robotic systems has focused on wheeled mobile robots navigating flat structured terrain conditions, mostly in indoor or urban scenarios [55], [56]. Reliable autonomy in challenging terrain conditions remains an open research problem. The two distinguishing factors that make rough terrain autonomy significantly more difficult as compared to flat/structured terrain autonomy are listed below:

Characteristics of terrain: For a structured terrain, a 2D occupancy grid is sufficient to plan an optimal path, but even obstacle free regions with significant slip or sinkage can prevent the robot from making progress in rough terrain. For challenging terrain conditions, it is not enough to have a 2D map of the environment. Additional factors including characteristics of the terrain in terms of slip, slope, and soil properties, terrain topology, variations in slip due to changing soil makeup and terrain stability conditions need to considered in order to plan a feasible path. Terrain topology refers to the shapes and features that describe the three-dimensional nature of the surface. In this work, a 3D elevation map is used to represent terrain topology.

Characteristics of the vehicle: For car-like or Hilare [57] type robots moving in structured terrain, the presence or absence of obstacles can be used to determine traversability in a trivial manner. But in the case of robots that are designed specifically for rough terrain locomotion such as SOLERO[58], the JPL Sojourner Rover [59], or hybrid tracked-wheeled multi-directional mobile robot like STORM [60] and HMMR [61]–[65], the vehicle characteristics such as weight and moment of inertia of the platform, actuator limitations, and the nature of the track profile need to be taken into consideration for planning. In addition to the above factors, tracked robotic systems are capable of navigation over most obstacles owing to their inherent mechanical advantages. As such, trivial estimation of traversability based on the presence or absence of obstacles will be an overly conservative approach for tracked systems.

All of the above factors make real-time traversability estimation using onboard sensor data a non-trivial problem for tracked systems. At the same time, estimating traversability is particularly important when it comes to reliable motion planning. For applications such as planetary exploration, it is mission critical to enable the system to obtain, understand, and utilize terrain information in real-time [66]. In summary, the path planning algorithms need to take into account the dynamic interactions between the robot and the terrain in detail in order to achieve reliable path planning. For a robotic system trying to navigate in previously unseen terrain, this requires continuous estimation of terrain properties such as topology and slip while progressively exploring the terrain and updating the map. All of this should be performed while keeping in mind the limited computational power available onboard.

Traditional techniques as mentioned above require the researcher to explicitly define the relationship between the detected features from the sensor data and the traversability of the terrain. This is difficult due to the fact that the dynamic interactions between the robot and the terrain are too complex to accurately model. A majority of the existing traversability estimation methods involve heuristic based techniques that rely on a variety of factors such as the maximum height of features in the terrain, slope, roughness, or even a combination of these [2]–[4] as obtained from onboard sensors such as LIDAR, camera, Inertial Measurement Units (IMU), etc. These techniques are often specifically designed for the application at hand and do not generalize well for different environmental conditions. Others have used a full dynamic model of the robot along with the terrain map to determine the same [69]. Even though they provide reliable results, full dynamic simulation of robot motion could be computationally intensive for the limited resources available onboard the robots. A detailed survey of existing geometry based and vision based terrain traversability methods is presented by Papadakis in [70].

In this chapter, a novel technique for estimating traversability using physics engine simulations capable of taking into account all of the above mentioned requirements is presented. The applicability of the proposed technique to improve path planning in challenging conditions is then validated through simulations and experiments strategy that account the above-mentioned requirements. Finally the use of physics engine simulations to generate training data for neural network based real-time traversability estimations is presented. For the purpose of the analysis presented in this chapter it is assumed that the robot knows its pose information in all six dimensions.

4.2. Review of rough terrain planning and control techniques

Robot path planning is a very well researched field, where many algorithms with specific advantages for a variety of applications have been proposed over the last few decades. A survey of recent advances in the domain of path planning techniques can be found in [55], [71]. In addition to mobile robotics systems, advanced planning techniques could also help modular robotic systems in self-assembly [72]. One of the major events that resulted in significant advances in autonomous vehicle research was the DARPA Grand Challenge (DGC) series in 2004 and 2005 [73]–[76]. The challenges focused for the most part on cars traversing "largely uniform and unchallenging" terrain, as mentioned in [69]. While the path planning strategies used by some of the teams considered terrain information, advanced vehicle dynamics were mostly ignored. Much of the self-driving car research that has followed draws on the foundation provided by the DGC, as such these are mostly targeted towards flat/structured terrain motion, even when focused towards driving in off-road conditions [77], [78].

Rough terrain path planning, as mentioned above, requires a more serious consideration of the robot and terrain characteristics as compared to general planar (x, y, θ) robot path planning [56], [79]–[81]. The majority of the strategies for rough terrain path planning use terrain maps classified into "occupied" and "unoccupied "cells based on the presence or absence of certain terrain features characteristic of the problem at hand. The work presented in [82] provides a method for estimating traversibility of unknown terrain using 3D vision sensors. The work describes an Unevenness Point Descriptor, computed from point cloud normal vectors with Principal Component Analysis (PCA). It showed good performance in terms of ground detection but the effectiveness of the same in robot path planning has not been demonstrated. On a similar note, [83] has proposed a method to estimate the roughness of the terrain using normal vector deviation based on data obtained from a 3D LIDAR. The terrain roughness information is then integrated into a Traversable Region Model, which uses a 2D Voronoi-based map to segment the XY region into cells and then assigns terrain traversibility based on roughness and slope as the cost of each cell. Others have used a terrain height map as obtained from 3D LIDAR, stereo Vision or structured light sensors to compute some form of terrain characterization based on slope, roughness and slip parameters. The above information is then used to compute artificial potential fields in order to determine the optimal path for the robot [84]–[87] . For instance, [86] has focused on casting the feasibility and cost of robot motion over a terrain as an optimization problem. They used the height map of the terrain along with a Fast Marching Method in order to come up with a potential field free of local minima. Using state lattice planner with primitive trajectories for path planning of a large tracked vehicle in open terrain is mentioned in [88]. However, the work assumes the terrain is assumed to be open and no consideration is provided for any terrain features like slope or actuator limitations of the robot. This allows the planner to assume that the robot is capable of executing the primitive trajectories at all times and is therefore not applicable to the challenges being addressed in this chapter. Approaches that involve finding the smoothest path in a given terrain map or looking for the minimum artificial potential energy assume that the path with these characteristics to be easily traversable. As mentioned previously, smoothness of a terrain does not always relate to traversability, especially in the presence of loose sand, ice, or mud. In other cases, looking for the smoothest terrain might prove to be a conservative approach as this ignores the ability of the platform to traverse challenging terrain as offered by its mechanical design.

The one common drawback of the above approaches is not taking into account the dynamics of the robot. Even though terrain topology plays a major role, ignoring or simplifying the dynamics of the vehicle and its actuator limitations can lead to failure of the planner through collisions or the vehicle's inability to execute the planned maneuvers. In contrast with the above approaches, [89]–[92] have used a simplified dynamic model of the vehicle along with the terrain elevation map to ensure that the vehicle does not tip over while traveling the path provided by the planner. Another similar work, [93], used a simplified model of the vehicle and terrain to check for stability at intermediate waypoints along the proposed path. Recent work by Currier and Wicks [73] proposed analytical methods for real-time estimation of Instantaneous Maneuvering Manifolds for large autonomous vehicles in order to predict their allowable dynamic operating ranges. Their method takes into consideration the uncertain and dynamic nature of payloads on autonomous vehicles as well as the varying frictional coefficient of the terrain, as applied to Ackerman steered vehicles. A two phase rough terrain path planning for actively reconfigurable robots is proposed in [94]. An initial path is obtained from a graph search, followed by identifying the rough regions on the path using vision data. Biased RRT* in the continuous state space of the robot is then used to refine the path on the rough regions, thereby taking into account the actuator limitations on the robot.

Another major factor to be taken into account for rough terrain path planning is terrain slip. Existing methods [58], [93] handle slip by means of robust trajectory tracking controllers while executing the planned trajectory. However, the above-mentioned approaches do not consider terrain slip during the planning stage. This is an inefficient strategy, as there will be cases where the robot cannot go over a slope or travel along the sides of a ravine due to significant slip. The planner then provides an un-traversable path that can result in failure, as even the best trajectory-tracking controller cannot overcome such significant slip events.

In summary, inaccuracies and simplifications in modeling the robot-terrain interactions lead to cases in which the feasible path reported by the planner results in collisions in the real world, or the robot not being able to execute the path at all. While some of the existing works may incorporate one or two of the above-mentioned factors during the planning stage, there exists no comprehensive solution to this problem.

4.3. Proposed method

In order to meet the requirements as discussed above, I propose a novel path planning architecture that consists of a high-level planner that takes into account the dynamic robot-terrain interactions by simulating closed loop motion of the robot with a low-level controller on a realistic terrain model inside a physics engine [47]. Once a feasible path to goal is obtained, the same low-level controller is used to execute the proposed path on the actual robot. The overall working of the proposed planner can be explained as follows: The high-level planner starts with a 2D grid map of the region in which the terrain topology is initially flat. The robot then obtains information about the nearby static obstacles using onboard sensors such as LIDAR, and updates the 2D occupancy grid to include them. In addition, the robot obtains the terrain topology for the robot's current position and nearby cells (at minimum the immediate eight neighboring cells). Once this information is obtained, for every obstacle-free neighboring cell that the robot can move into, the feasibility of the path is further validated using the physics engine to account for the dynamics of the robot-terrain interactions. A block diagram representation of the proposed navigation technique is presented in Fig. 4.1.

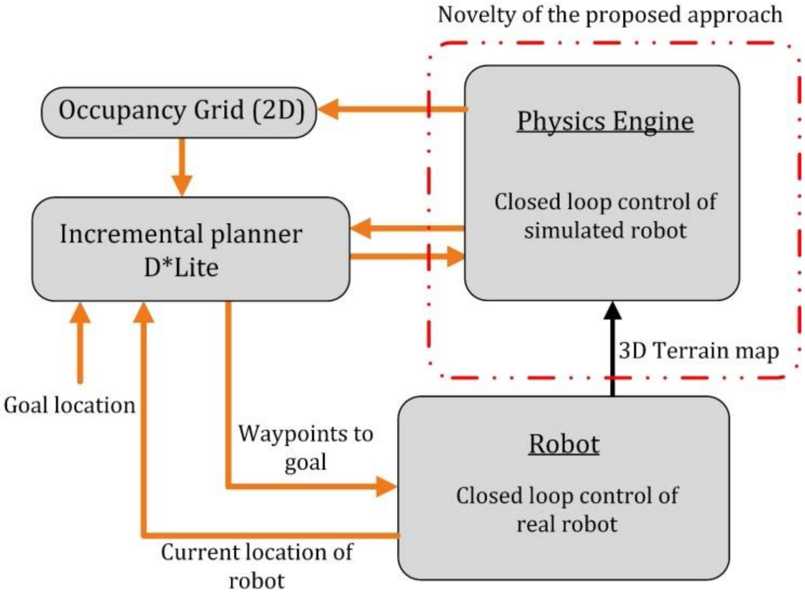

Novelty of the proposed approach
Occupancy Grid (2D)
Physics Engine
Closed loop control of simulated robot
Incremental planner D*Lite
3D Terrain map
Goal location
Waypoints to goal
Robot
Closed loop control of real robot
Current location of robot

Figure 4.1: Block diagram representation of the proposed planner

The physics engine realistically models the terrain topology and the actuator limitations of the physical robot. The simulated robot inside the physics engine is directed to travel from the current location to the desired neighboring cell by means of the closed loop low-level controller. If the simulated robot successfully travels to the neighboring cell, the cost of the cell can be kept as a tunable combination of time taken and expended control effort. If the robot is not able to travel to the neighboring cell within a specified period, that neighboring cell is marked as unreachable. This can occur due to terrain conditions, obstacles, robot dynamics, or actuation limits of the robot. The simulated feasibility of motion is checked only for cells that already have a 3D terrain map generated, using the information obtained from the actual robot. For any cells that does not yet have this information, the high-level planner assumes the motion is feasible. This allows the system to come up with an initial path to goal extending beyond the range of the real robot's sensors.

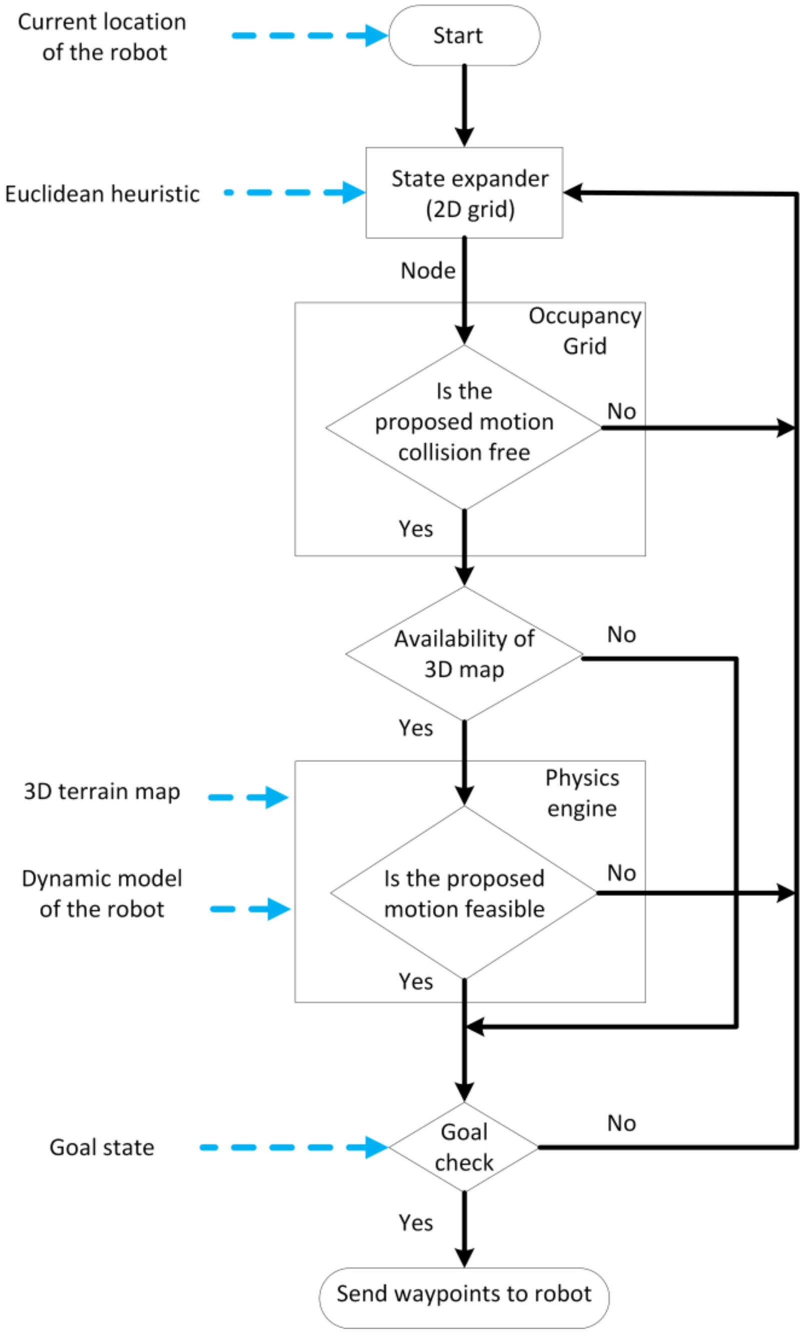


Figure 4.2: Flow chart showing the working of the proposed planning algorithm

Based on the results from the physics engine, the initial path generated by the high-level planner is passed to the physical robot as a list of waypoints. The low-level controller running on the physical robot then guides it to the next waypoint (local goal), based on the current position. Once the robot moves to the next waypoint within the vicinity of previously unexplored cells, it collects more sensory data. The updated sensor information is passed to the high-level planner, which then replans if necessary and outputs the new path. This process is iterated until the physical robot arrives at the goal or determines there is no path available. A flow chart explaining the working of the proposed method is shown in Fig. 4.2. A major advantage of this method is that only intermediate goals are passed on to the physical robot, rather than the actual actuator commands. This allows the low-level controller onboard to inherently correct for any deviations in the mass, inertia or other physical parameters between the physical robot and the simulated robot, thereby providing a better guarantee of reaching the goal location without colliding with obstacles.

4.3.1. Novelty of the proposed approach

The major contribution of this work lies in exploring the use of physics engines for state expansion of a mobile robotic system inside a high-level planner**.** In addition, a detailed description on how to create a complete architecture for the autonomous navigation of a ground robot in rough terrain is also presented. The use of a physics engine to handle the kino-dynamic aspects of the planning such as actuator limitations, slip and other terrain conditions allows the planner to generate reliable paths while keeping the computational complexity low enough for real-time operation. In addition, the ability of the physics engine to simulate the motion of the robot under the action of the low-level controller on the simulated terrain allows the planner to take into account the effects of the hybrid controller on the motion of the robot. All of these factors demonstrate improvements on the existing work in this domain.

4.3.2. High-level planner

Previous approaches in path planning for robotic systems with complex dynamics, significant drift, and limited sensing mostly relied on kino-dynamic versions of stochastic methods like RRT, RRT*, or its variants [95]–[99]. The general approach when utilizing kino-dynamic RRT* is to plan in higher dimensional state space, where the planner randomly samples states, followed by either explicitly calculating or randomly guessing actions that will take the robot from the current state to the target state. Once a feasible path is calculated from start state to the goal state, the corresponding set of actions is then performed by the robot in the physical space. The success of these methods thereby strongly depends on the accuracy of the modelling. The presence of significant un-modeled dynamics can result in the robot being unable to reach the goal. However, performing such random sampling while using a physics engine in order to expand the states leads to high computational cost. This makes the approach unsuitable for real-time planning in dynamic environments.

Since the current application is focused towards motion planning in a previously unseen environment, a more appropriate high-level planner is D* Lite, due to its efficient re-planning capabilities that allow it to handle changes in the environment [100]. D* Lite is more computationally efficient when compared to repeated A*, while being easier to implement as compared to original D* [101]. As a grid based planner, D* Lite is made to work on the discretized work space (x, y plane in this case) rather than in the higher dimensional discretized state space $(x, y, \theta, \dot{x}, \dot{y}, \dot{\theta})$, while using the Euclidean distance from the goal as a heuristic. The total length of the path taken by the robot from start to goal is the cost function. In order to reduce complexity, the grid-based algorithm utilizes a robot model that can move from any given cell to the eight neighboring cells, provided the neighboring cell does not have an obstacle in it. The high-level planner is initialized with a uniformly discretized 2D model of the environment (occupancy grid), with all unknown edge costs set to minimum value. This allows it to develop initial guesses on the path to the goal without having the full terrain map to begin with.

4.3.3. Kino-dynamic aspect of planning

In the case of robotic systems with complicated dynamics involving non-holonomic constraints, finding a feasible trajectory between the initial and final/desired states is a non-trivial problem. This is particularly true with systems involving significant interaction with the working environment, such as walking robots, dexterous manipulation and tracked vehicles of significant mass traversing through rough terrain while under the action of a low-level feedback controller. In order to accurately model the motion of the physical robot, the planner must also take into consideration the low-level controller that drives the physical robot [95]. Most of the existing approaches ignore these aspects or model them using simplified kinematic equations, which may result in unrealistic estimates thus leading to failures in path planning.

Existing works in this domain have tried to first model the motion of tracked vehicles on rough terrain as a set of analytical equations and then solve these equations. However, as the complexity of the system increases the computational cost associated with the modelling increases, leading prior work to focus only on certain specific aspects of the challenge such as slope of the terrain or the effect of slip. As a result, the accuracy of the prior methods is impacted by focusing on a certain specific aspect of the system and neglecting others.

In contrast with existing literature in this domain, I propose the use of physics engines to model the kino-dynamic aspect of planning. A physics engine is software that provides a simulation of physical systems, capable of simulating rigid body dynamics (including collision detection), soft body dynamics, and fluid dynamics. A snapshot of the motion of tracked vehicle on rough terrain as simulated by a physics engine is shown in Fig. 4.3. These are primarily used in the domains of computer graphics, video games, movies and high-performance scientific simulation. A physics engine calculates the forces that arise between bodies when they interact with each other, with the goal of preventing bodies from inter-penetrating. These forces are then used to derive the motion of the bodies, using multi-body dynamic equations.

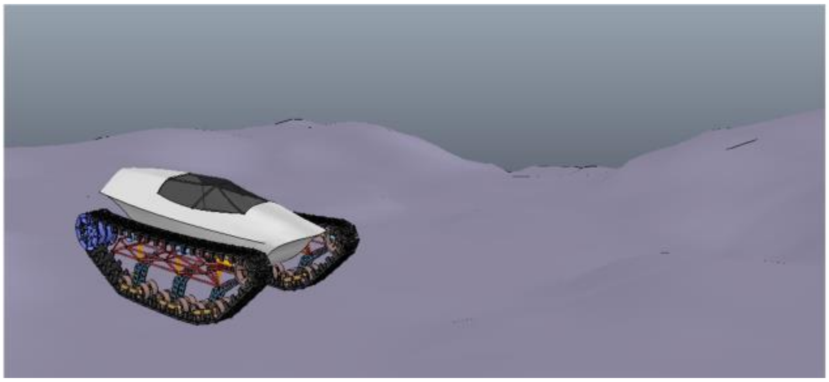


Figure 4.3: Simulated scene showing the motion of tracked vehicle on rough terrain

The engine also models various joints, such as revolute or prismatic joints, as sets of algebraic constraint equations. The effects of friction that arise between the bodies while they interact are also taken into consideration. In order to solve for the motion of the objects within the simulation, a time stepping approach is used wherein the differential algebraic equations (DAE) from the dynamics and constraints are solved simultaneously to obtain the state of the system subsequent to the time step. For more details regarding the working of physics engines refer to [102]–[104].

Compared to the existing analytic modeling approaches, the use of a physics engine offers a more computationally efficient and practical solution to model the robot-terrain interactions, low-level control algorithms and the sensor models that the actual robot possesses. The use of a physics engine allows for the state evolution of these systems forward in time while taking into account motion dynamics, gravity, friction and other aspects of ground interaction. The robot model can be initialized in most existing physics engines by importing the robot in Universal Robot Description Format (URDF). The URDF consists of a set of files describing the physical attributes of the robot such as mass, inertia, and the dimensions of the various links as well as how they connect together. Provided the robot-terrain system is correctly initialized, the future states can be obtained with a reasonable degree of accuracy through simulation without having to derive and solve the equations that govern the evolution of the system.

The engine needs only the current state of the system and the control inputs in order to estimate the new state, much like the actual physical robot itself. In other words, the high-level planner can treat the physics engine as a “black box”. By simulating the motion of the robot from one waypoint to another, the planner can take into account the dynamics of robot-terrain interactions, including slip, in a realistic manner during the planning stage itself and thereby accurately chart a feasible path. The above reasoning justifies the use of physics based robotic simulators for state expansion, rather than simplified dynamic equations, at the cost of marginal increase in computational load.

A comparison of some of the existing state of the art physics engines can be found in [105]. For the purposes of this work, I will be using Bullet Physics in headless mode to reduce the computational overhead [106]. Bullet was chosen due to the superior performance of its friction model and the excellent documentation available online. Moreover, Bullet falls under the category of real-time physics engine as it uses approximate calculations in order to produce accurate real time results. While Bullet is capable of modelling soft terrain such as in the case of loose sand or mud, the simulations used in this work treat both the robot and the terrain as rigid bodies. The approximation of rigid bodies allows for faster simulation based on the capabilities of current state of the art physics engines. As physics engines improve in their ability to model soft deformable objects and their interaction, simulations that are more realistic can be incorporated for better representation of the motion of a tracked vehicle on deformable terrain such as loose sand or mud, increasing the accuracy of the proposed approach. The hardware and software improvements necessary for incorporating these features are part of ongoing work.

4.3.4. Low-level controller

The path computed by the high-level planner (a set of waypoints ending at the goal) will be executed on the physical robot by a low-level controller. The low-level controller, running at a higher frequency than the high-level planner, continuously monitors the state of the robot and the environment through sensors in order to generate control inputs (right and left track velocities for a differential drive robot) to ensure stable navigation from the current state to the next waypoint. For the proposed approach, a hybrid automaton is used as the low-level closed loop controller.

Navigating successfully in a dynamic environment is a challenging task, especially from a control system design perspective. The system should be capable of reacting to a variety of situations that may occur at unknown time instants. One of the ways to approach this problem is through the design of behavior-based control, where separate controllers are formulated for handling different scenarios such as going towards the goal, avoiding an obstacle, and stopping the robot upon reaching goal.

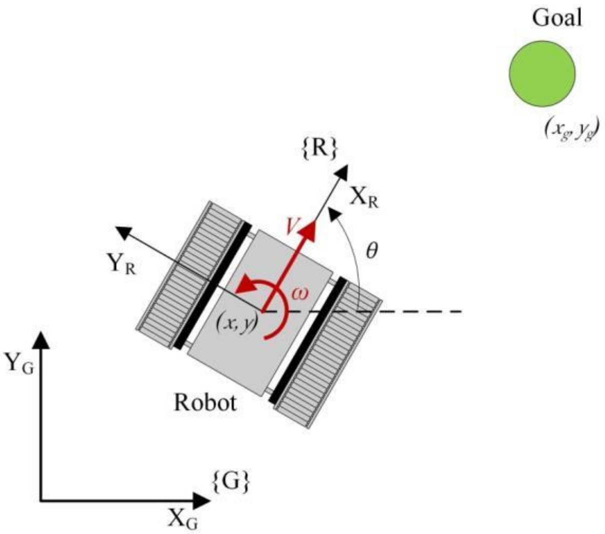


Figure 4.4: Schematic showing the states of the robot, goal and the control input given to the system

For instance, The Go-to-goal controller is essentially a PID controller that drives the vehicle to the nearest waypoint as provided by the high-level planner. In order to drive the robot from current location (x, y) to a goal location (x_g, y_g), the robot has to first align towards the goal and then drive forward till it reaches a location within the threshold distance to the goal point. A schematic showing the stats of the robot, goal and the control inputs given to the robot is given in Fig. 4.4. The slope of the line connecting the robot and the goal position, θ^*, is the desired orientation of the robot. Based on the heading error the angular velocity control input (ω) to the robot can be determined from the equations given below:

$$\theta^* = tan^{-1}(\frac{y_g - y}{x_g - x})$$

$$e = \theta^* \ominus \theta \tag{4.1}$$

$$\omega = k_p e + k_d \dot{e}$$

where, k_p, and k_d are the proportional, and derivative gains respectively, e is the error limited between [–π, π) in order to account for the wraparound of orientation

The linear velocity of the robot is scaled non-linearly based on the angular velocity as given by:

$$V = V_{max}\left(1 - \frac{2tan^{-1}(|\omega|)}{\pi}\right) \tag{4.2}$$

This allows the robot to slow down before taking turns.

A higher-level state machine is made responsible for switching between the behaviors based on the external inputs, such as sensor information from proximity sensors or laser scanners, thereby ensuring that the robot always operates within one of the finite states (the behavior-based controllers) at any given point of time. Guard conditions inside the state machine designed for each behavior enable switching between the behaviors. The individual behaviors, along with the state machine and the guard conditions, together form a hybrid automaton based closed loop controller for autonomous mobile robot navigation [107].

While this approach simplifies the controller design, the presence of hard switches in the behavior could introduce chatter due to scenarios that require infinite switches in finite time (Zeno phenomenon) and thereby detrimentally affect the performance of the overall system. As mentioned in [107] this can be solved by adding regularization controllers to fuse the behavior of the two conflicting nodes in the system. The design of a hybrid automaton based closed loop controller for low-level mobile robot navigation is described in [108]. A simplified version of the same as the low-level controller was used for the proposed approach.

4.3.5. Implementation details

The following section provides further details on the proposed approach to provide the reader with a deeper understanding for the application of the method.

Computational time: For the purpose of the simulation and the experimental validation described in the following sections, the 2D grid resolution for the high-level planner is kept equal to 0.5m along both the axis. With the fixed grid resolution, the time taken by the physics engine to simulate the motion of the robot between two adjacent cells is at max 0.8 seconds (while running on an HP laptop with a 2.6 GHz Intel processor and 8GB RAM). Once the robot reaches the neighboring cell, the simulation is terminated and the motion is reported feasible to the high-level planner along with the cost. The minor variation in time taken is caused by variations in the nature of the terrain corresponding to the cells. In cases where the robot fails to reach the goal, the simulation could run infinitely long. To prevent this, the simulation is forcibly terminated after one second and the motion is reported not feasible. Under the above approach, one second is the worst-case time taken to simulate the motion of the robot between two adjacent cells.

Optimality of path: The proposed approach is capable of finding the optimum path by using an optimal high-level planner, D* Lite in this case, after a subset of feasible paths are provided through the physics engine. Like any other implementation, the definition of optimality depends on the cost function assigned to the planner. For the sake of demonstrating the functionality of the proposed implementation, the length of the path traversed by the robot to reach the goal is optimized. As indicated by previous works [99], [109], [110] the shortest path may not be optimal in terms of time taken by the robot to reach the goal or the energy spend by the robot in doing so. By modifying the cost function on the high-level planner, one can enable the proposed approach to optimize for time, energy or any combination of these, since all of the costs associated with the motion of the robot can be obtained from the physics engine.

Handling slip: The kinematic model of the differential drive robot described in Section 3.3 is solely used for designing the low-level controller. This is done without taking into account the effects of slip. The physics engine in the proposed architecture is responsible for handling the longitudinal and lateral slip experienced by the robot. The coefficient of friction of the terrain modelled inside the physics engine is kept low such that the robot experiences more slip than it would in reality. By simulating robot motion on such terrain, the proposed navigation architecture can detect possible maneuvers by the robot that may fail in the real world during the planning stage itself. Even though conservative in nature, this approach allows the system to handle longitudinal and lateral slip by avoiding such maneuvers while evaluating the path to goal. This is demonstrated in the experimental trials, which were performed in a high slip condition as described in the following section (Section 5). This validates the fact that the proposed planner takes into account the slip of the robot during the planning stage itself, unlike existing approaches. Additional improvements such as estimating slip while traversing through the unknown terrain and updating the physics engine in real time will be considered as part of the future work.

4.4. Simulation

In order to validate the feasibility of the proposed planning architecture, it was initially tested using simulation. The simulations were performed using the V-REP robotic simulator provided by Coppelia Robotics [111]. The terrain maps were generated using the ANT landscape add on [112] in BLENDER [113], which was then imported into V-REP for simulation. The motion of the robot was simulated in two different terrain maps differentiated by increasing frequency of variations in the terrain height map, characterizing a moderate and extreme level of roughness. The simulated terrain domain in both cases was a square with sides four meters in length. For the moderate terrain case, the simulated terrain height varies from a minimum of -0.5m to a maximum of 0.3m. For the extreme terrain case, the simulated terrain height varies from a minimum of -0.5m to a maximum of 0.1m, but with high frequency variations in terrain height as compared to the moderate terrain case. For both the terrain maps, the starting position of the robot is considered as ground level or zero elevation. The robot used for the simulation is the caterpillar model provided in V-REP with a nodding SICK LMS 300 LIDAR for obtaining the terrain map and five ultrasonic sensors (two on each side and one in the front) for the low-level Avoid-obstacle controller. The simulated robot is 0.5m in length, 0.5m in width and 0.3m in height, with a total mass of 16Kg.

The proposed planning architecture, including the D* Lite high-level planner and the low-level hybrid automaton controller, were implemented in MATLAB which then communicates with V-REP through the remote API interface. During the simulation the position and orientation (all six dimensions) of the robot was read off directly from the simulator by the low-level controller running in MATLAB. Bullet Physics running in headless mode was used to model the dynamic aspects of the planning problem. For both the simulation cases, the results of the proposed planner were compared with the outputs of a kinematic planner. The only difference between the proposed planner and the kinematic planner is that the kinematic version does not check the validity of the proposed path using the physics engine, ignoring the dynamic aspects of the problem. This allows the kinematic planner to produce the optimal shortest path in all cases, with no regard to the feasibility of the proposed path. Both the simulated terrains have bounding walls to prevent the robot from going outside the simulated region. Other than the bounding walls, no other static obstacles were present in both the terrain maps. For the purpose of the simulation, the low-level controller was set such that the simulated robot would not detect the terrain features as obstacles.

4.4.1. Results and discussion

The proposed planning architecture was able to find feasible paths in both moderate and extreme terrain cases, whereas the kinematic planner failed in the second case. In addition, the proposed planner matched the results of the kinematic planner in producing the shortest feasible (optimal) path for the moderate terrain case. This signifies the improvements of the proposed method over the existing artificial potential field methods and others that do not consider the dynamics of the robot or terrain topography.

Based on the results, in both the simulation cases, the proposed planning architecture met the requirements that were described in the previous sections. The results of the simulation are shown in the topographical plots in Fig. 4.5, with the outcome of the kinematic planner in red and the proposed planner in blue. The density of the terrain lines denotes relative steepness of the terrain. For both cases, the waypoints provided by the planner are shown as dots and the path followed by the simulated robot in dashed lines. In both scenarios, the kinematic planner outputs the shortest path to the goal, which is a straight line. For the moderate terrain case, the proposed planner matched the result of the kinematic planner showing that it is optimal in terms of path length.

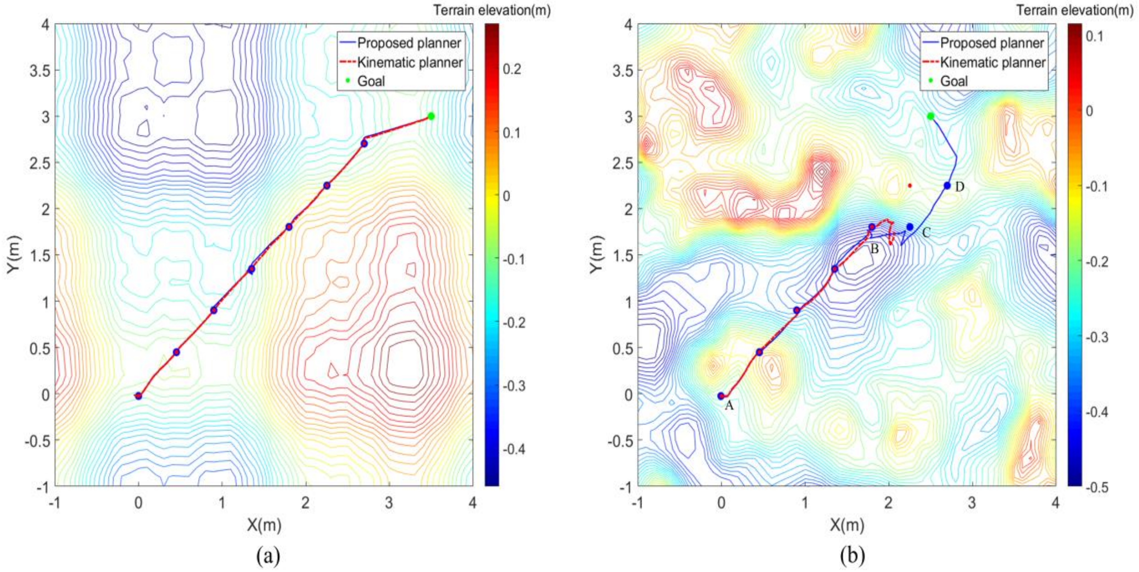


Figure 4.5: Simulation results (a) Moderate terrain (b) Rough terrain

For the extreme terrain case, the shortest path is not feasible since at point B, as shown in Fig 4.5 (b), the terrain is too steep for the robot to climb. The kinematic planner fails at this point whereas the proposed planner detects the failure inside the physics engine and therefore takes a right turn in order to ensure the feasibility of the proposed path, while still trying to minimize the overall path length. In addition, the proposed method shows better performance as compared to artificial potential field methods that use the height map of the terrain to develop with feasible paths. At point A as shown in Fig. 4.5 (b), artificial potential field methods and methods looking for smooth regions in the terrain map would have traveled in the reverse direction in order to avoid the steep climb. The proposed method instead checks whether the robot can execute the climb inside the physics engine and based on the result decides to go forward, resulting in a shorter path to goal. The deviations of the robot from the desired path at points B, C and D as shown in Fig. 4.5 (b), are due to the longitudinal and lateral slip experienced by the simulated robot in V-REP.

4.5. Experimental Validation

The proposed planning architecture was experimentally validated using STORM, a mobile robot platform developed in the Robotics and Mechatronics Lab [60]. STORM has multi-directional mobility capabilities enabled via hybrid combination of tracks and wheels that operate independently of each other. It also has ultrasonic sensors on all four sides for obstacle avoidance applications. Each STORM module is developed to be capable of docking with other robotic modules to form a larger system with better locomotion and manipulation capabilities [114]–[116]. For the purpose of this chapter, the robot was operated only in tracked differential drive mode. The robot is 0.41m in length, 0.3m in width and 0.12m in height, and weighs a total of 9Kg. The robot possesses ultrasonic sensors on all four sides for obstacle avoidance and utilizes an ODROID XU4 computer for onboard processing. 3D scans of the experimental terrain (height map) were obtained using a Kinect XBOX One. Due to the limited battery power and computational capability available onboard the robot, a complete 3D scan of the terrain was obtained beforehand for the experiment rather than in real-time.

For the duration of the experiment, positional information of the robot was obtained using Linear Optical Sensor Arrays (LOSA) tracking system, an opto-inertial motion tracking device developed in the Robotics and Mechatronics Lab [117]. The LOSA system is capable of providing 3D position and orientation information at very high accuracy, up to 1mm in position and 1 degree in orientation. The near-infrared LOSA marker was mounted on the robot and the LOSA tracker was connected to a laptop. Based on the current location of the robot as obtained from LOSA, the provided location of goal and the terrain map, the planner computes a feasible path to reach the goal. The proposed planner running on the laptop sends the path as waypoints to the robot's onboard computer through ROS. The low-level hybrid automaton controller running onboard the robot takes the waypoints as intermediate goals along with the positional information in order to drive the robot. As the infrared sensors on LOSA and Kinect XBOX One used in this experiment are limited to indoor use, the proposed planning architecture was tested on a rough terrain model created indoors using a tarp-covered rubble pile to create a varied, high slip environment. The experimental terrain had an overall length of 5.5m and width of 5.5m. The terrain height varied from 0m to a maximum of 0.25m.

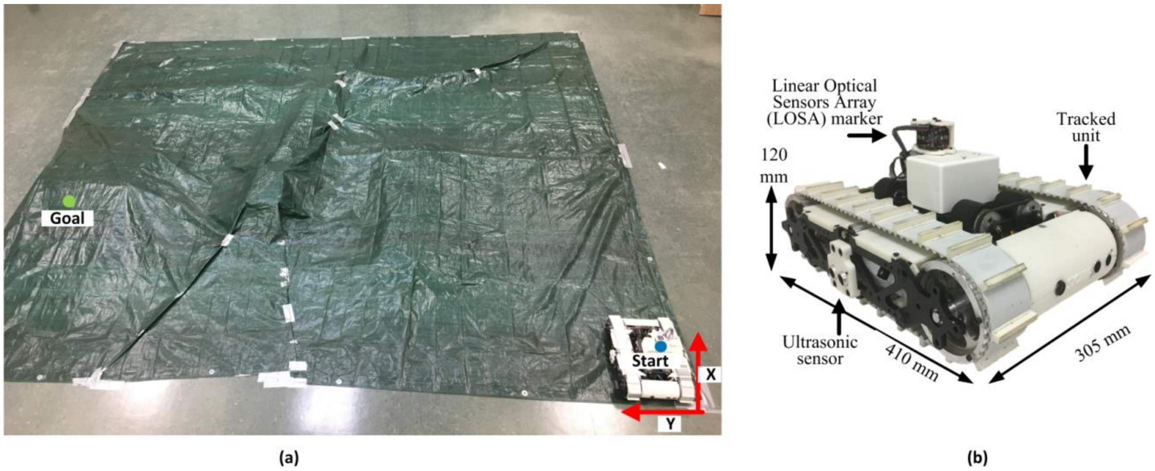


Figure 4.6: Experimental setup (a) Terrain with start and goal positions marked (b) STORM fitted with LOSA marker

A picture of the experimental setup, showing the start and goal positions, are shown in Fig. 4.6 (a). The STORM module fitted with the LOSA marker is shown in Fig. 4.6 (b). As in the case of simulation, a virtual bounding wall was provided inside the planner to prevent the robot from going outside the limits of the experimental setup. In addition, the low-level controller was set such that STORM would not detect the terrain features as obstacles. For the purpose of the experiment, the 2D grid resolution for both the proposed and kinematic planner was kept equal to 0.5m along both the axis. Due to this resolution, the path proposed by the kinematic planner is not a straight line. Reducing the grid size would improve the resolution of the planner thereby making the path proposed by the kinematic planner closer to the ideal straight-line path at the cost of increased computation.

4.5.1. Results and discussion

For the purpose of experimental validation, the robot was required to go from one end of the terrain to the other while crossing a steep ridge in the middle. As in the case of the simulations, both the kinematic planner and the proposed planner were used to plan the path of the robot. The kinematic planner proposed the shortest path, which required the robot to cross the steep central ridge. The robot was unable to cross the ridge during the experiment and thus the kinematic planner failed. The proposed planner was provided with the complete map of the terrain within the physics engine, from which it evaluated that the robot would not be able to cross the central ridge. Based on this information, the proposed planner developed a longer but feasible path that successfully guides the robot to the desired goal location. The path proposed by the kinematic planner and the proposed planner are shown as waypoints in Fig. 4.7. The actual path followed by the robot in both cases is also shown.

The deviation of the actual path followed by the robot from the waypoints specified by the planners was mainly due to the significant slip encountered by the system on the tarp. For the purpose of the experiment, the d_{thresh} value was set to 200mm. This means that the low-level controller running on the robot assumed that the goal point was reached as soon as it was within 200mm distance from the point. Fig. 4.8 shows the path followed by the robot under the action of both planners in 3D for better clarity.

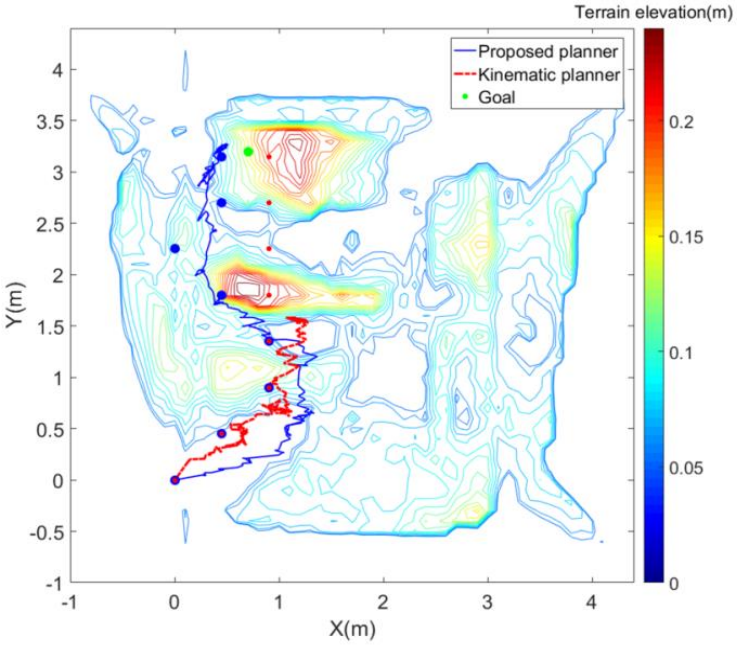


Figure 4.7: Performance comparison of the proposed planner with kinematic planner over the experimental setup

The high frequency oscillations in the robots motion were mainly due to the slipping of the robot tracks on tarp. On an outdoor terrain with less slip, the robot should be able to achieve a smoother trajectory. The experiments thus demonstrated that the proposed planner was more effective in providing a feasible path than existing planners. Coupled with the online physics check, the proposed planner presented clear advantages for rough, unstructured terrain planning.

The time taken by the robot to reach the goal under the action of the proposed navigation architecture was 1 minute and 47 seconds. As the high-level planner was implemented based on D* Lite, the proposed architecture dynamically re-planed at every waypoint based on the updated map. Therefore, the above-mentioned time depended not only on the planner but also on the average speed of the robot, type of terrain and even on the communication delay between the proposed planner running on the laptop and the low-level controller running on board the robot.

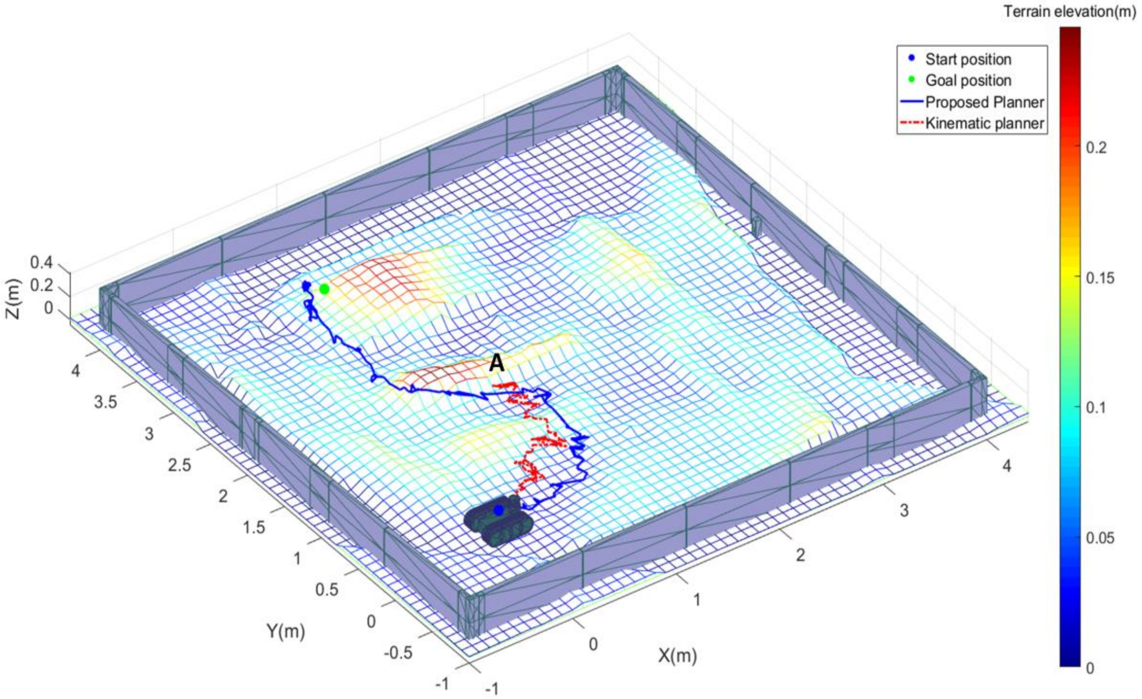


Figure 4.8: Terrain topography map showing the 3D path followed by the robot under both the planners over the experimental setup

4.6. Neural network based traversability estimation

In addition to their use in state expansion for motion planning, physics engine simulations can also be used for generating training data for neural network based real-time traversability estimation as explained in the following section. As mentioned in Section 4.2, traditional traversability estimation techniques require explicit definition of the relationship between the detected features from the sensor data and the traversability of the terrain. This is often difficult due to the fact that the dynamic interactions between the robot and the terrain are too complex to accurately model. To address this issue, recently there has been more interest in learning based traversability estimation techniques that are capable of deriving the relationship from the given data. Based on the diversity of the collected data, these techniques are capable of generalizing to environments with varied features and characteristics. In addition, even though the training process itself is computationally intensive, the trained machine learning architecture could be deployed with minimum hardware requirements, thereby improving the real-time applicability of these techniques.

In this regard, Murphy et al. [118] presented a Gaussian process technique to predict the traversability of unknown locations based on traversability estimates garnered from onboard sensing along with a-priori available overhead color images of the regions. Chavez-Garcia et al. [119] presented traversability estimation as a heightmap classification problem that could be solved using a trained convolutional neural network based on terrain patches represented as images. Unsupervised classification of outdoor 3D LIDAR data for future terrain traversability estimations was presented by Maligo and Lacroixin [120]. A semi supervised learning approach, where the robot learns its traversability capabilities from a human operator was presented by Suger et al. in [121]. An approach towards correlating exteroceptive (terrain observations or images) and proprioceptive (acceleration signals) information to assess terrain manoeuvrability for mobile robots was presented by Bekhti et al. in [122], whereas proprioceptive sensing information including wheel slip, vehicle orientation, vibration, and power consumption were used to map instantaneous traversability in [123]. An approach towards combining separate traversability maps generated from colour images and point clouds using Bayes' rule was presented by Sock et al. in [124]. A detailed survey of recent learning based techniques towards estimating terrain traversability is given by Wong et al. in [66].

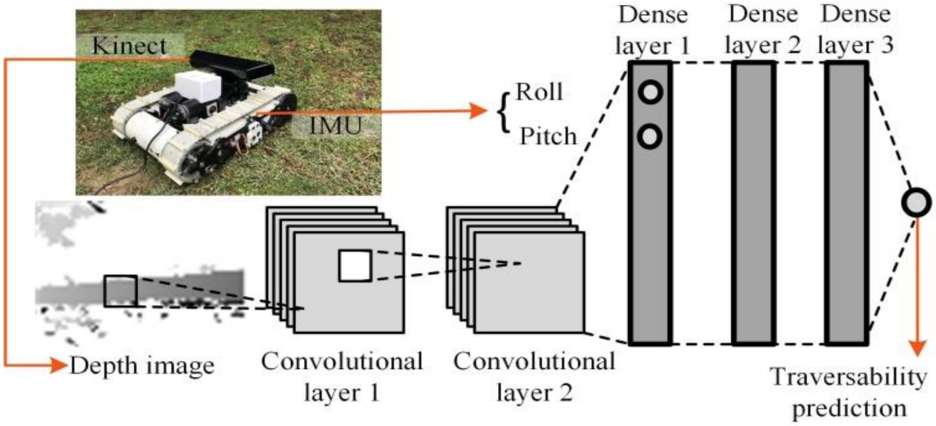


Figure 4.9: Overall layout of the proposed neural network architecture

4.6.1. Proposed technique

The proposed neural network architecture [125] is capable of combining heterogeneous sensor information to analyze terrain traversability in real-time. The proposed architecture consists of two convolutional layers and three dense layers as shown in Fig. 4.9. The convolutional layers operate on the depth image obtained from the Kinect sensor, while the dense layers operate on the output of the convolution layers stacked with the roll and pitch angles of the robot to generate traversability estimation. In comparison to existing state of the art techniques, the proposed work offers the following novelties and advantages:

(1) The ability to combine heterogeneous sensor data: The proposed approach uses depth data of the robot's surroundings along with roll and pitch of the platform as input to the NN architecture. Even though there are previous approaches towards combining separate traversability maps from different sensing modalities [124] and also for correlating different sensor inputs [122], to the authors knowledge this work is the first attempt towards training a single neural network to use heterogeneous sensor data for estimating terrain traversability.

(2) Prior knowledge of the environment is not required: Existing learning based terrain traversability approaches in the literature require prior knowledge of the environment in some form, such as 3D maps [119], or overhead images [118]. There are many real-life applications of field robotic systems where prior information is not available. As mentioned by Shan et al. [68], assuming the prior availability of a 3D map can severely limit the reliability of these techniques when it comes to field robotic applications. The proposed approach on the other hand only relies on real-time sensor data using onboard sensors.

(3) Use of simulations for generating training data: Unlike existing approaches [10],[11] that use experimental data for training the traversability estimation techniques, the proposed approach uses training data generated from a high fidelity simulation. A robotic simulator capable of accurately modelling the dynamic motion of a mobile robot over challenging terrain conditions along with the sensor outputs was used. This allowed

generating a diverse training dataset without any damage to the real robot, while ensuring that the trained policies are applicable to real-life conditions.

4.6.2. Data collection in simulated environment

One of the major difficulties with the use of ML based methods in robotics is the need for extensive and accurate datasets. In order to accurately judge whether or not a particular maneuver can be performed, the robot must actually perform the maneuver. Since the dataset will need samples of both feasible and non-feasible cases, the robot will have to perform maneuvers that could cause potential damage. In order to create an extensive and accurate dataset without damaging the robot, a robotic simulator, V-REP (Virtual Robotics Experimentation Platform) [111] was used to perform the desired maneuvers and record the results along with the simulated sensor data.

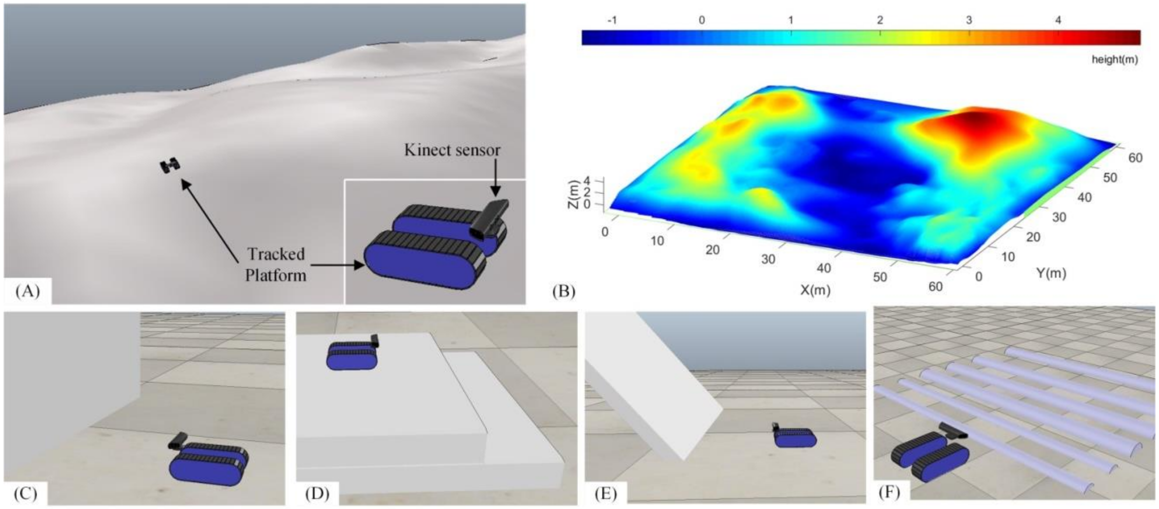


Figure 4.10: Data collection through simulation: (A) Tracked platform with Kinect on simulated outdoor terrain, (B) Topography map of the terrain used to generate the dataset. Simulated indoor scenarios (C) Facing a wall, (D) Facing edge of platform, (E) Facing incline (F) Facing bumps

V-REP is a robotic simulator by Coppelia Robotics. This cross platform software allows simulating robotic platforms including a variety of sensors and actuators, along with realistic physics engines that support dynamic simulations of robot motion. A tracked robotic platform with a Kinect depth sensor attached to its front was made to travel along a simulated terrain condition consisting of hills, valleys, and structured terrain consisting of walls, ramps, and bumps as shown in Fig. 4.10. The virtual terrain of size 60x60 m for the simulation was created in Blender using the Ant plugin [113]. Due to the varying terrain conditions, the frictional resistance experienced by the tracked robotic vehicle is not the same on both sides. As such, providing the same velocity command to both tracks does not guarantee that the robot will drive straight, even in the simulator. This makes it necessary to have a closed-loop controller running during the data collection simulations. Moreover, the same low-level controller is used on the actual robot for the experimental validation. As such, simulating the controller during data collection makes the data more representative of the actual motion of the robot in real-life conditions.

The maximum linear and angular velocity commands from the controller were limited to 0.03 m/s and 1.0 rad/s, respectively. Once the desired linear and angular velocities have been calculated using the low level controller, they can be transformed into the desired left and right track velocities using the transformation equations between differential drive and unicycle robot models before they are applied to the tracked robot.

It should be noted that even though the low-level controller presented here is based on robot kinematics, V-REP performs a full dynamic simulation of the robot motion at each instant. Before conducting the data collection simulation, the weight, size and actuation limits of the simulated robot were adjusted to be the same as that of the hardware used for experimental validation, STORM [126]. The simulated robot was 0.41 m in length, 0.3 m in width and 0.12 m in height with a total weight of 9 kg, similar to the actual robot. The torques applied on the tracks were limited to a maximum of 10 Nm with a PID (Proportional-Integral-Derivative) speed controller simulating the performance of the motor driver on the actual robot. This further reduced the possible mismatches between the behavior of the simulated and real robot.

For the purpose of data collection, the low-level controller was implemented in MATLAB and was made to communicate with V-REP. Based on the simulated robot pose (x, y, θ) as obtained from V-REP, the desired control commands were sent back from MATLAB during the simulation.

In order to create the training instances, the simulated robot was placed at random poses on the simulated terrain and checked for static stability. Once the robot was stable, a depth image of the area directly in front of the robot was obtained from the simulated Kinect sensor, along with the roll and pitch of the robot. This data was sent to MATLAB and stored. The robot was then required to move towards a goal location directly in front of the robot at a fixed distance of 1m using the closed-loop controller. If the robot was able to reach within 0.1m of the goal location within 75 seconds of simulation time, the motion was recorded as successful, and if not, as a failure.

Ten thousand instances of robot motion were simulated to create an extensive and accurate dataset. This dataset was used as training and test data for the proposed NN architecture. Each training instance consists of a depth image (48x64 array) of the region in front of the robot, roll and pitch (normalized values) of the robot before the motion was started, and the result of whether the motion was successfully completed.

4.6.3. Neural Network (NN) Architecture

This section explains the proposed NN architecture which incorporates the depth image and the IMU data to predict traversability of the terrain. The proposed architecture consists of convolutional layers and dense fully connected layers.

The convolutional layers process the depth images to extract features that correspond to the traversability of the terrain. Each convolutional layer consists of 32 filters with 2x2 kernel size followed by a max pooling layer with the same kernel size. IMU data is fed into the network through the first fully connected layer. This is done by concatenating the IMU data with the flattened output of the second convolutional layer in order to improve the accuracy of the final prediction. Each fully connected layer consists of 256 neurons except for the last one, which consists of only one neuron to perform the traversability prediction. A sigmoid activation function is used in the prediction layer while all the other layers use leaky ReLU activation function [127], as given by

$$f(x) = \begin{cases} x & x > 0 \\ 0.02x & otherwise \end{cases} \tag{4.3}$$

The neural network is built and trained in Keras with TensorFlow [128] as the backend using Adam optimizer [129].

In order to prevent scaling of the weights based on the varying units of the incoming heterogeneous data, normalization was performed on the raw input data before being fed into the NN. The raw depth image was clipped to measure only between 0m to

4m and then normalized between [0,1]. The IMU data was clipped between [-40, 40] degrees and normalized within [0, 1].

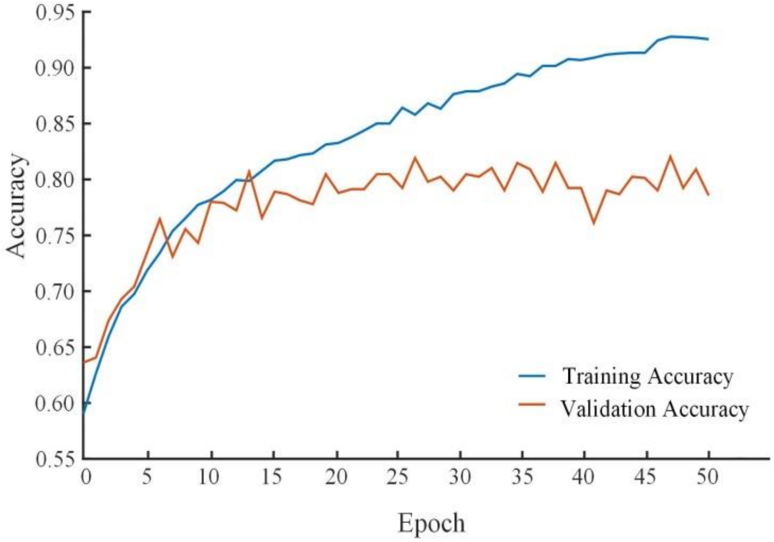


Figure 4.11: Variation of the training and validation accuracy over multiple epochs

To train the NN, the dataset was split randomly: 9,000 samples were used for training and 1,000 samples for validation. Both training and validation datasets contained approximately half traversable and half non-traversable samples. As shown in Fig. 4.11, the proposed architecture gave an average validation accuracy of 80% after ten epochs. To avoid overfitting, the best weights of the NN were saved by monitoring the minimum loss of validation during the training process. The trained architecture provided 91.6% accuracy on detecting true positives (traversable) and 72.0% accuracy on detecting true negatives (non-traversable).

4.6.4. Experimental Validation

One of the inherent issues with training ML architectures inside simulations is that the real-world scenario could be different from the math model underlying the physics simulation. Even though the trained architecture may work well in simulation, model mismatch issues can make transfer of knowledge from the physics simulator to the real-world difficult. As such, it is important to validate the performance of the trained architecture in real-world conditions with real robot and sensor data.

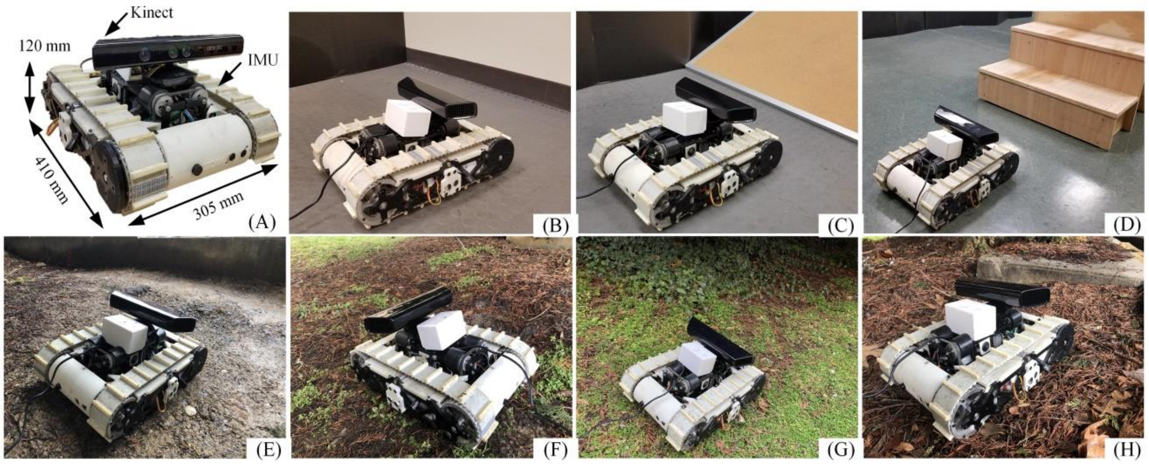


Figure 4.12: Experimental validation in structured and unstructured terrain

The experimental validation involved testing the proposed NN architecture in five structured indoor and four unstructured outdoor scenarios, some of which are shown in Fig. 4.12. Fig. 4.12 (B)-(D) denote indoor conditions and (E)-(G) denotes outdoor conditions. Similar to the simulated scenario, a tracked robotic system, STORM, fitted with a Kinect sensor and IMU, was used for the experimental validation. As in the case of data collection simulations, a depth image of the terrain in front of the robot, along with the orientation of the robot was sent to the trained NN architecture. It should be noted that the trained architecture was deployed on the hardware for experimental validation without any additional tuning. Moreover, the outdoor experiments were conducted after sunset to reduce the infrared interference from sunlight with the Kinect.

In each case the robot was placed at a distance from possible obstacles and the low level controller on the robot was used to drive the robot straight with a constant linear velocity of 0.01 m/s. Throughout the process, the onboard sensor data was used to estimate traversability in real-time.

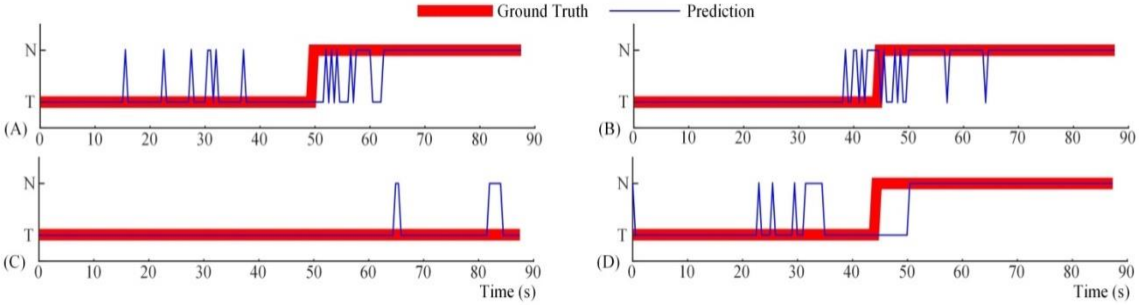


Figure 4.13: Variation of traversability estimates over time in structured and unstructured terrains: (A) Facing a wall at an angle (Indoor), (B) Facing an edge of the platform (Indoor), (C) Going downhill (Outdoor), (D) Facing a wall (Outdoor). T denotes traversable and N denotes non-traversable

The real-time traversability estimates from four different scenarios are shown in Fig. 4.13. For non-traversable cases the ground truth shifts from traversable to non-traversable when the robot is 1m away from the obstacle. As evidenced from Fig. 4.13, the proposed technique works well in both indoor and outdoor scenarios, even though it experiences false positives and negatives close to the transition point.

4.6.5. Path planning application

The feasibility of using real-time traversability predictions as obtained from the trained architecture for incremental path planning applications was validated through simulations. The same tracked robotic vehicle model used in the data collection simulations was also used for the path planning simulation.

The traversability estimations from the trained architecture were used to update an incremental planner, D* Lite [100] to enable an autonomous tracked vehicle to successfully navigate to a goal location without any prior maps. The overall navigational architecture consisted of D*Lite as the high-level planner which starts off with a grid map (each unit being 0.5x0.5 m) having the start and goal locations, the low level PID controller that drives the robot to each waypoint as dictated by the planner, and the NN architecture that informs the planner about possible obstacles in front of the robot. In order to reject false detections, the map was updated only when the NN predicted non-traversability continuously for five iterations. The planner then updates its map based on the detected obstacles. The simulation was performed in V-REP, with the navigation architecture implemented in python. For the duration of the simulation, the x, y position and roll, pitch, yaw orientations of the robot were given to the navigation architecture along with the depth images, which in turn returned the control inputs to V-REP.

The path planning simulation was performed on unstructured terrain map (3.5x3.5 m), outside of the training data. Under the proposed navigation architecture, the robot avoids the steep hill in the middle and goes around as shown by the topography map in Fig. 4.14. It can be seen clearly that the robot successfully modifies its plan in real-time based on the traversability estimates during navigation. It should be noted that the robot did not possess any other sensors to detect obstacles and the path planning algorithm was solely informed by the trained NN architecture.

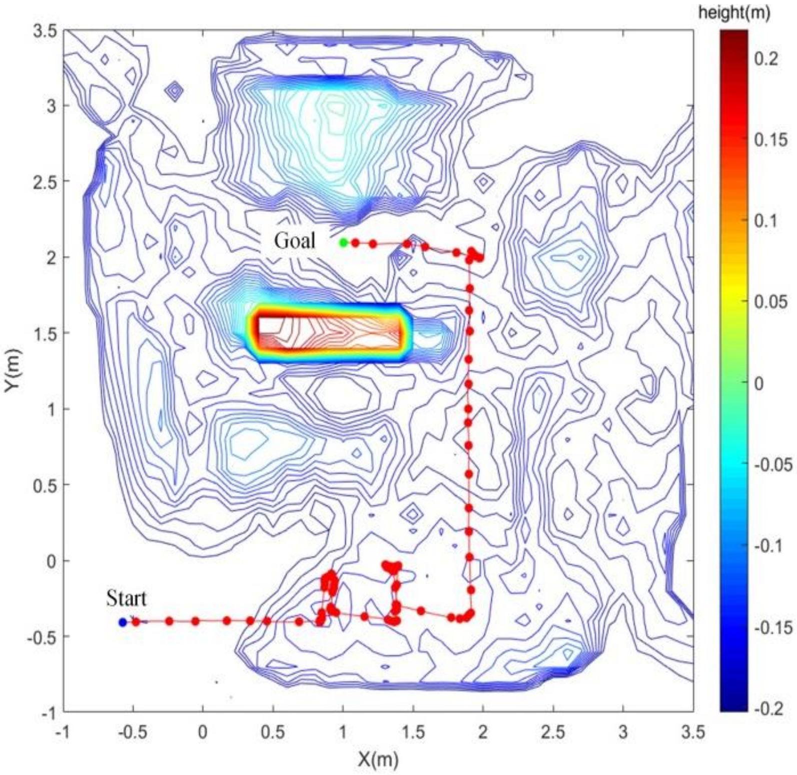


Figure 4.14: Path planning in unstructured terrain condition. The start and goal locations are marked in blue and green dots, respectively. The path followed by the robot is marked in a solid red line

4.7. Conclusion and Future work

This chapter proposed the use of a novel planning architecture to overcome the shortcomings of existing kinematic and artificial potential field based path-planning methods for mobile ground robot navigation in high dynamic terrains. The proposed planning architecture involved the use of a grid based high-level planner along with a physics engine to take into consideration of the kino-dynamic aspects of robot-terrain interactions. The path proposed by the high-level planner as a set of waypoints to the low-level hybrid automaton controller was checked for feasibility inside the physics engine. The same hybrid automaton was then used to execute the proposed path on the real robot. The proposed method was validated through both simulations and experimental results. In both, the performance of the proposed planner was tested against a kinematic planner that does not take into account the dynamics of the robot or terrain. In both simulation and experiment, the proposed planning architecture has shown clear improvement in performance by developing feasible paths in challenging terrain conditions where the kinematic planner failed. Furthermore, in moderate terrain conditions the proposed planner produced paths as efficient as those of the kinematic planner. These results demonstrate that the proposed planner is optimal in terms of path length, while taking into account the feasibility of the proposed path.

The proposed navigation architecture was capable of handling static obstacles by means of the D* Lite high-level planner. By assuming that all the obstacles are static, the high-level planner can take into account the possibility of collisions while the robot moves to the goal without taking into account the actions of the robot in the velocity space or adding time as a dimension to the planning space. However, there exist planners that can handle moving obstacles by adding time as a dimension to the planning space [130], [131]. By replacing D* Lite with these planners, along with providing estimated velocity of the moving obstacle inside the physics engine, the proposed approach will be able to handle moving obstacle cases efficiently. A drawback to this is that as the number of moving obstacles increases, the computational overhead of the physics engine also increases. In order to incorporate dynamic obstacle planning, additional hardware and software computational optimizations will be required.

The planning architecture described in this chapter assumed constant slip over the full duration of navigation. The dynamic model of the robot was assumed constant as well. In real-life conditions, these assumptions may not hold. Slip can vary based on the type of terrain the robot is operating on, while factors affecting the dynamic model of the robot such as total mass of the robot and positon of the center of mass could change based on the nature of the mission performed by the robot. Future work will aim to take into account these cases as well.

CHAPTER 5

SLIP REJECTION CONTROL FOR TRACKED MOBILE ROBOTS

5.1. Introduction

A majority of the applications that involve Autonomous Ground Vehicles (AGVs), such as performing reconnaissance and rescue operations in rough terrain or autonomous package deliveries in an urban setting, require the system to follow a path predetermined by a high-level planner or based on a predefined task [132]–[134]. The path, usually defined by series of waypoints, will be taken as an input to a low-level path following controller that then generates the control inputs for the robot. For applications involving differential drive or skid-steer robots, the low-level controller most often leverages a unicycle model, as given in Eq. 5.1, to drive the robot along the desired path. Since the control inputs for the unicycle model are the linear (V) and angular (ω) velocity of the robot, it is easier to come up with reliable and intuitive control laws using this model as compared to the differential drive or skid-steer model. The unicycle model is given below:

$$\begin{aligned}\dot{x} &= Vcos\theta \\ \dot{y} &= Vsin\theta \\ \dot{\theta} &= \omega\end{aligned} \tag{5.1}$$

where (x, y) is the 2D position robot fixed frame {R} with respect to the global inertial frame {G} and θ represents the orientation {R} with respect to {G} about the Z-axis. The above model can be summarized into the following form:

$$\dot{\boldsymbol{X}} = \boldsymbol{CU} \tag{5.2}$$

where, $\boldsymbol{X} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$, $\boldsymbol{C} = \begin{bmatrix} cos\theta & 0 \\ sin\theta & 0 \\ 0 & 1 \end{bmatrix}$, $\boldsymbol{U} = \begin{bmatrix} V \\ \omega \end{bmatrix}$

Once the control inputs are determined based on the states (x, y, θ) of the robot they are transformed into the left and right wheel angular velocity (ω_l, ω_r) before they can be applied on the differential drive robot, using the following equation:

$$\begin{bmatrix} \omega_l \\ \omega_r \end{bmatrix} = \begin{bmatrix} \dfrac{2V - \omega L}{D} \\ \dfrac{2V + \omega L}{D} \end{bmatrix} \tag{5.3}$$

where D and L are the wheel diameter and the width of the robot (axle track) respectively. The above equations are valid for a differential drive robot with two narrow wheels on both sides and a castor wheel at the base, such as the Pioneer P3DX platform, moving at a low speed without slip. For such a system, the two wheels can be assumed to make point contact with the ground and the axle track of the platform can be measured as the distance between the point contacts. The traditional navigation architecture as described above is shown in Fig. 5.1.

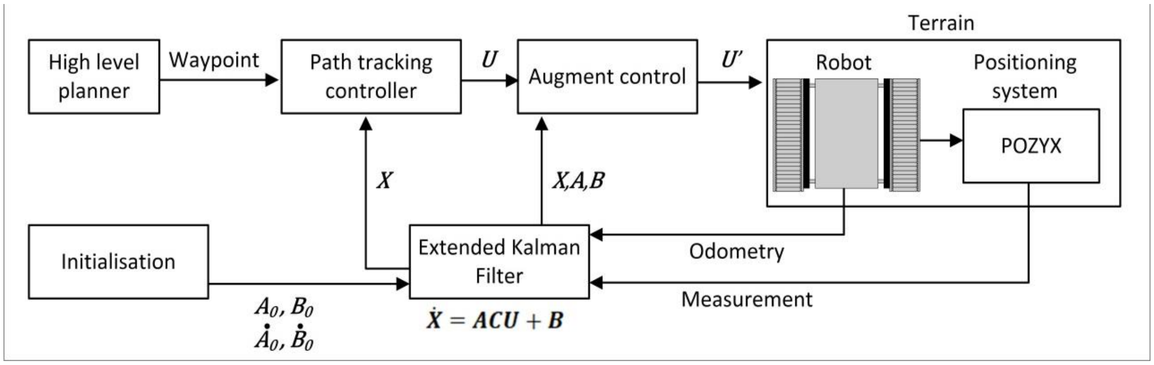


Figure 5.1: Block diagram representation of traditional path tracking controllers

The increased ground contact offered by tracked locomotion mode becomes a disadvantage in this case. Unlike wheeled or legged locomotion, which can assume no slip under ideal conditions, the tracked locomotion requires a skidding turn, where a large portion of the track must slide against the terrain. Therefore, tracked locomotion is also called slip/skid-steer locomotion. The large amount of skidding involved during a turn makes it difficult to accurately estimate the center of rotation of the robot and thereby the future position and orientation of the robot for given left and right track velocities [132], [133]. Moreover, the motor torque required to perform skid-steer varies greatly depending on the amount of friction on the terrain, geometry, material of track, etc. The above factors make it difficult to obtain a reliable model for tracked vehicle locomotion, which in turn affects the accuracy in position estimation, model predictive control, and motion planning. As such, model based control techniques have limited performance when it comes to path following application for tracked robots on challenging terrain conditions. A detailed review of some of the existing methods and their limitations is given in Section 5.2. While the major factors that affect the degree of slip experienced by a system can be broadly generalized as the weight of the vehicle, nature of terrain, wheel velocity, material of wheel, etc., the exact relationship between these parameters and their effect on the motion of the robot is still unknown.

In this chapter, I propose an active disturbance rejection based method to handle the effect of slip in an online manner. The proposed approach uses an Extended Kalman Filter based observer to estimate the varying slip parameters online. The estimated parameters are then used to compensate for the effects of slip at each iteration by modifying the control actions of the low-level controller. The proposed approach is validated through experiments over the flat and uneven terrain conditions of asphalt, vinyl flooring, artificial turf, grass, and gravel using the tracked skid-steer mobile robot, STORM [126].

5.2. Review of slip rejection controllers

In the case of skid-steer robots, especially tracked platforms, the assumptions of no slip and point contact with the ground are not valid. This, in turn, causes the linear and angular velocities, V and ω, experienced by the robot to be significantly different from the values expected by the closed loop controller. This effect, which can be represented as noise in the actuator, becomes particularly significant when the robot moves over challenging terrain. Depending on the size and inertia of the robot and the type of terrain, this can lead to cases where the effect leads to inefficient navigation profiles, jerky motion, and may even be significant enough to prevent the robot from reaching the desired goal. A recent review by Gonzalez and Iagnemma [135] states that reliable slip estimation and compensation strategies play a major role in enabling safe and efficient navigation. Even though the review focuses mainly on extraterrestrial rovers, the limitations of current techniques and the persisting challenges as reviewed by the authors are relevant to autonomous ground vehicles in general.

In general, longitudinal slip is defined as the difference between the velocity measured at the wheel and the linear velocity at the center of the wheel [135]. The velocity measured at the wheel is given by Ωr, where Ω is the rotational speed and r is the wheel radius. This is usually the desired velocity from the controller perspective. Provided there is a way to estimate the linear velocity of the wheel's center, v_{linear}, the percentage of motion lost to slip, s, can be calculated as

$$s = 100 * \begin{cases} \dfrac{\Omega r - v_{linear}}{\Omega r}, (driving) \\ \dfrac{\Omega r - v_{linear}}{v_{linear}} (braking) \end{cases} \tag{5.4}$$

Earlier works to improve path following in AGV's assumed that once slip is estimated it can be easily compensated for by adjusting the control efforts, V and ω, to account for the effect of slip while using Eq. 5.2, 5.3, and 5.4. With this assumption, the focus was directed towards estimating the exact amount of slip experienced by the robot through estimation of the linear and angular velocities of the robot. The major problem with this approach is that majority of the existing platforms use Inertial Measurement Units (IMUs) which directly measure the acceleration of the robot, and therefore provide noisy velocity estimates. In this regard, an adaptive approach towards slip estimation was presented by Burke [136], to estimate slip solely based off the angular velocity of the robot, which can be measured with a high degree of accuracy from the IMU. Other approaches [137]–[140] involved the use of optical flow sensors, RTK GPS, and fusion of multiple sensing modalities to accurately estimate slip. Even though some of these works succeeded in accurately estimating slip, they did not result in significant improvement in path following because a majority of these approaches relied on the traditional differential drive model to compensate for the effect of slip and assumed that the underlying path following controller would be able to handle the model mismatches as well. As mentioned in [136], model-plant mismatch is a particular concern when model predictive controllers are applied.

Recently, many research groups have tried to formulate better models to accurately predict the motion of AGVs in the presence of slip. Based on a review of the available literature, the methods can be broadly classified into two groups, one that uses the full dynamic model of the system including the robot-terrain interactions and the second that relies on the kinematic model of the robot, while ignoring the dynamic effects.

Approach	Limitation
Estimate slip: IMU, optical flow sensors, RTK GPS, and multi- sensor fusion [127], [128]–[130]	Underlying model is limited
Full dynamic model of vehicle terrain interaction [63], [131]–[133]	Requires real-time terrain property estimates, additional sensors and computing capability
Pure pursuit generates desired control commands & LQR handles slip as disturbance [134]	Requires robot and terrain parameters in advance
Online parameter estimation & ICR based tracked robot model [135]–[138].	Small amount of longitudinal slip ➔ large ICR values ➔ controller saturates
Proposed approach: Using augmented kinematic models with online parameter estimation method to account for the effect of slip in real time	

Table 5.1: Review of existing state-of-the-art controllers for handling slip

The approaches that use the full dynamic model [67], [141]–[143] usually start from the first principles to estimate slip based on one or more features characterizing the robot terrain interactions such as the effect of slope, tire/track forces, nature of the terrain, etc. or through physics engines that use multi-body dynamics method to model the motion of the robot over challenging terrain [47]. Even though the use of physics engines allow for easy modelling while ensuring reliable estimates for the motion of the robot, this method does not directly provide an estimate for the slip experienced by the robot and is therefore not serviceable for controller design. On the other hand, using detailed analytic models to estimate slip over varying terrain conditions requires real-time estimation of terrain geometry, track tension, forces exerted by the track on the ground, and soil properties including consistency and compressibility [144]. Estimating these parameters in real-time and solving the model is not feasible due to the computational limits of the systems onboard AGVs.

In the case of using the kinematic models to estimate and compensate for slip, the approach has been focused towards using additional parameters in the ideal differential drive robot model as given by Eq. 5.1, to account for the effects of robot-terrain interactions. The effective wheelbase model [145] and the general kinematic slip model [146] fall under this category. A recent comparison of some of these models can be found in [147]. One notable work in this direction includes a model predictive optimal control presented by Rajagopalan et al. [148]. The approach essentially used a Pure Pursuit Path Follower to generate the desired control commands along with an LQR tracker to handle the effects of slip as disturbance by modifying the control inputs. The disadvantage here is that this method needs an estimated value of the slip parameters and powertrain dynamics of the robot in advance to tune the LQR gains. This is infeasible for field robots specifically in cases where the terrain data is either unknown beforehand or varies drastically. The solution is to use online parameter estimation. Martinez et al. [149], [150] proposed an Instantaneous Center of Rotation (ICR) based tracked robot model. Real-time estimation of the ICR has shown to improve robot localization [151] and trajectory tracking [152]. However, the fact that even a small amount of longitudinal slip can result in very large ICR values when the robot is moving in a straight line render it unsuitable for path following controllers. This causes the path following controller to saturate frequently leading to jerky motion of the robot. Helmick et al. proposed a Kalman Filter based method to calculate slip [153]. The method, specifically designed for the Rocky 8 Mars Rover, used the estimated slip in an inverse kinematic model to modify the wheel velocities and steering angles to better follow the path.

In addition to improvements in trajectory tracking, slip estimation has also been performed to get better estimates of the power consumption by robots. Recent work by Canfield, Hill and Zuccaro [154], [155] has introduced a novel method to calculate slip parameters along with torque applied and path followed by the robot, based on an ICR based dynamic model of the robot. The estimated slip parameters were then used to evaluate the power consumption of the skid-steer mobile robot, which could in turn be used to find optimal trajectories that minimize power consumption. The proposed method was practically validated on two different manufacturing applications using a skid steer robot with magnetic-tracks. On a similar note, Gupta, Ordonez and Collins [156] has proposed methods to perform dynamically feasible energy efficient motion planning for skid steer robots, taking into account torque limitations. The proposed method takes into account the payload of the robot along with terrain conditions and slip experienced by the robot, in order to come up with minimum turn radius constraints. These constraints are then used in a sampling based model predictive optimization technique to generate energy efficient trajectories for skid-steered autonomous ground vehicles.

Based on a survey of existing work in this domain, the ideal approach towards handling the effect of slip in AGV path following would be to use an augmented kinematic model that can take into account the time-varying effects along with an online parameter estimation method to accurately predict the values in real time. In this regard, Active Disturbance Rejection Controller (ADRC), which is capable of handling modelling uncertainty as external disturbance, satisfies all of the requirements. The ADRC can act as a high-level controller that estimates the disturbance signal online that then modifies the control action provided by a low-level path following controller to cancel out the effects of slip.

5.3. Proposed approach

This section describes the design of an Active Disturbance Rejection Controller (ADRC) to handle the effect of slip and thereby improve path following performance for ground robots [157]. The seminal paper that introduced ADRC [158] highlighted four major contributions to tackle the shortcomings of the traditional PID framework: the use of a simple differential equation as a transient trajectory generator, a noise-tolerant tracking differentiator, nonlinear control laws, and the concept and method of total disturbance estimation and rejection. Among the improvements brought out by ADRC, its ability to estimate and reject disturbances in an online fashion has generated great interest in the method [159]. In comparison to existing classical and state-space control techniques, including model predictive control, ADRC does not require an explicit model of the process it is trying to control. Instead it can work with a simple canonical model by taking into account all of the modelling uncertainties as external disturbances. This characteristic makes ADRC an ideal choice for handling systems with process parameter variations and disturbances that cannot be easily modelled. As mentioned in the previous sections, disturbances, modelling uncertainties, and parameter variations are the major factors that make the control of AGV on rough terrain difficult and as such, ADRC is perfectly suited for this application.

ADRC is usually designed as a high-level controller acting on top of a stable low-level controller designed for an ideal, known, model of the system. The primary idea behind active disturbance rejection is to lump the modelling uncertainties and parameter variations into disturbance signals that are then formulated as additional states resulting in an augmented system. An online observer is then designed for the augmented system to estimate the additional states based on the output or feedback from the system. The estimated value of the disturbances, in turn, modifies the control action produced by the low-level controller to account for the disturbances or modelling errors present in the real system.

Assuming the estimator is designed well, ADRC effectively converts the real system into the ideal model for which the low-level controller is guaranteed to perform well. In other words, the entire ADRC architecture can be wrapped as an additional layer on top of the existing low-level controller. It is important to note the underlying assumption that the disturbance experienced by the real system can be considered to be piecewise constant. Taking into account the high update rate achievable on the computing architectures available onboard modern AGV's, this assumption holds well for practical implementation. A block diagram of the proposed ADRC architecture is given in Fig. 5.2. The three essential parts of the ADRC architecture are the model, estimator, and the modification to the control law.

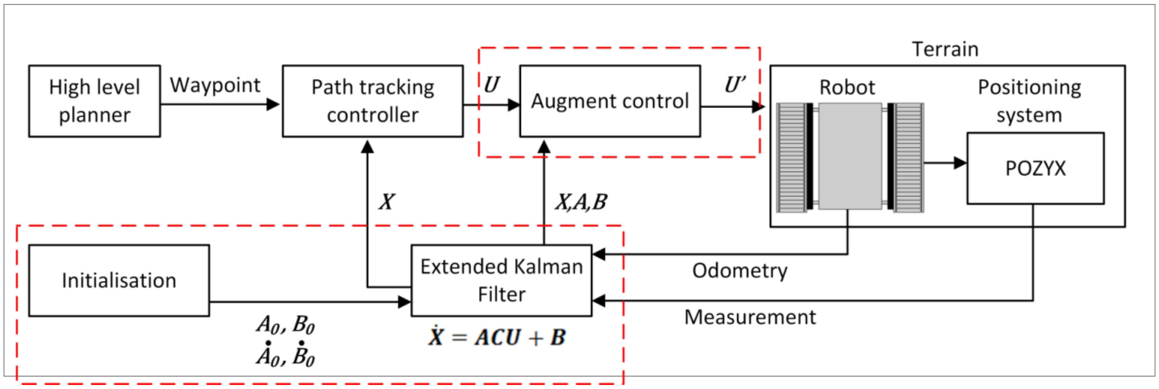


Figure 5.2: Block diagram representation of the proposed active disturbance rejection controller for path following in AGVs

5.3.1. Generalized model

As mentioned in Section 5.2, many different models have been proposed over the years to account for the effect of slip in AGV motion. Unlike the existing models that have added parameters to account for specific factors such as the variation in the wheelbase of the robot, a generic model that takes into account the scaling and shift produced in the robot states as a result of slip is proposed. The proposed generic model is given below as a modification to Eq. 5.1.

$$\dot{x} = \alpha_1 V cos\theta + \beta_1$$

$$\dot{y} = \alpha_2 V sin\theta + \beta_2 \qquad (5.5)$$

$$\dot{\theta} = \alpha_3 \omega + \beta_3$$

The α_i parameters are added to address the effect of slip through scaling the rate of change of robot states and the β_i parameters are added to address the same through shift in the rate of change of states. Eq. 5 can be summarized as:

$$\dot{\boldsymbol{X}} = \boldsymbol{ACU} + \boldsymbol{B} \tag{5.6}$$

where, $\boldsymbol{A} = \begin{bmatrix} \alpha_1 & 0 & 0 \\ 0 & \alpha_2 & 0 \\ 0 & 0 & \alpha_3 \end{bmatrix}$ and $\boldsymbol{B} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}$

5.3.2. Proposed Estimation method

The augmented model ADRC requires an observer to estimate the value of the added parameters. The proposed approach will use an Extended Kalman Filter (EKF) as the observer. The prediction model for the EKF can be obtained by discretizing the kinematic model given in Eq. 6 using the Euler method. The augmented parameters are also updated in the EKF prediction model as per the ADRC approach. The complete prediction model used in this chapter is given below:

$$\begin{gathered}
x_k = x_{k-1} + (\alpha_{1,k-1} V_{k-1} cos\theta_{k-1} + \beta_{1,k-1})\Delta t + \mu_1 \Delta t \\
y_k = y_{k-1} + (\alpha_{2,k-1} V_{k-1} sin\theta_{k-1} + \beta_{2,k-1})\Delta t + \mu_2 \Delta t \\
\theta_k = \theta_{k-1} + (\alpha_{3,k-1} \omega_{k-1} + \beta_{3,k-1})\Delta t + \mu_3 \Delta t \\
\alpha_{i,k} = \alpha_{i,k-1} + \dot{\alpha}_{i,k-1}\Delta t + \mu_{3+i}\Delta t \\
\beta_{i,k} = \beta_{i,k-1} + \dot{\beta}_{i,k-1}\Delta t + \mu_{6+i}\Delta t \\
\dot{\alpha}_{i,k} = \dot{\alpha}_{i,k-1} + \mu_{9+i}\Delta t \\
\dot{\beta}_{i,k} = \dot{\beta}_{i,k-1} + \mu_{12+i}\Delta t
\end{gathered} \tag{5.7}$$

where, i varies from 1 to 3, $\mu_1 - \mu_{15}$ denotes the process (additive zero-mean Gaussian) noise, k is the discrete time index and Δt is the time step between each update. Note that the resulting state vector has 15 elements. As denoted by the model, the order of the augmented system is higher than that of the traditional model given in Eq. 2. However, this is a necessary trade-off in the implementation of the ADRC. In accordance with the classical EKF implementation Eq. 7 can be summarized as:

$$\widehat{\boldsymbol{\mathcal{X}}}_k = \boldsymbol{\mathcal{F}}(\boldsymbol{\mathcal{X}}_{k-1}, \boldsymbol{U}_{k-1}) \tag{5.8}$$

As per the classical EKF notation, variables with '^' are predicted values and those without are corrected values. The state covariance matrix, $\boldsymbol{S}$, is 15x15 matrix that is updated as:

$$\widehat{\boldsymbol{S}_k} = \boldsymbol{G}_{k-1}\boldsymbol{S}_{k-1}\boldsymbol{G}_{k-1}^T + \boldsymbol{V}_{k-1}\boldsymbol{Q}\boldsymbol{V}_{k-1}^T \tag{5.9}$$

where $\boldsymbol{Q}$ is the covariance matrix for process noise and $\boldsymbol{G}$ and $\boldsymbol{V}$ are the partial derivatives of $\boldsymbol{\mathcal{F}}$ w.r.t $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{U}$ respectively. The sensor measurements for the EKF are obtained from an absolute positioning system that provides noisy updates of states (x_k, y_k, θ_k) collected in vector $\boldsymbol{\mathcal{Y}}_{\mathrm{k}}$. The measurement equation for the EKF is therefore given by:

$$\boldsymbol{\mathcal{M}}(\boldsymbol{\mathcal{X}}_k) = \begin{Bmatrix} x_k + \vartheta_1 \\ y_k + \vartheta_2 \\ \theta_k + \vartheta_3 \end{Bmatrix} \tag{5.10}$$

where $\vartheta_1 - \vartheta_3$ denote the measurement (additive zero-mean Gaussian) noise. The Kalman gain is calculated as:

$$\boldsymbol{K}_k = \widehat{\boldsymbol{S}_k}\boldsymbol{M}_k^T(\boldsymbol{M}_k\widehat{\boldsymbol{S}_k}\boldsymbol{M}_k^T + \boldsymbol{R}_k)^{-1} \tag{5.11}$$

where $\boldsymbol{R}$ is the covariance matrix for measurement noise and $\boldsymbol{M}$ denotes the partial derivative of $\boldsymbol{\mathcal{M}}$ w.r.t $\boldsymbol{\mathcal{X}}$. The state and the covariance matrices are updated using the canonical EKF update equation as given by:

$$\boldsymbol{\mathcal{X}}_k = \widehat{\boldsymbol{\mathcal{X}}}_k + \boldsymbol{K}_k\left(\boldsymbol{\mathcal{Y}}_k - \boldsymbol{\mathcal{M}}(\boldsymbol{\mathcal{X}}_k)\right)$$

$$\boldsymbol{S}_k = (\boldsymbol{I}_{15x15} - \boldsymbol{K}_k\boldsymbol{M}_k)\,\widehat{\boldsymbol{S}_k}. \tag{5.12}$$

The values for the $\boldsymbol{G}$, $\boldsymbol{V}$, and $\boldsymbol{M}$ for the proposed model are provided in the appendix. The numerical values chosen for the $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices are given in Section 4. In order to improve the accuracy of the prediction step in the EKF, instead of using the V and ω provided by the controller in the calculation, the linear and angular velocity of the robot calculated from the wheel mounted encoders were used.

5.3.3. Low-level controller

A simple “Go-to-Goal” behavior is used as the low-level controller. Assuming a unicycle robot model as given by Eq. 2, this controller guides the robot to reach a given planar goal point from any given starting planar position and orientation [134]. This behavior consists of a PD controller that determines the angular velocity, ω, of the robot with a nonlinear scaling applied to the linear velocity, V, based on the calculated ω. Assuming the robot to be at coordinates (x, y) with an orientation of θ and the goal location to be at coordinates (x_g, y_g), the desired orientation of the robot is given by:

$$\theta^* = tan^{-1}(\frac{y_g - y}{x_g - x}) \tag{5.13}$$

The error in orientation is given by, $e = \theta^* \ominus \theta$ where $\ominus$ denotes difference taking into account the wraparound of angles. Based on the error, a PD controller can be written for the angular velocity of the robot as, $\omega = k_p e + k_d \dot{e}$; where k_p and k_d are the proportional and derivative gains respectively. The linear velocity of the robot is scaled based on ω as follows:

$$V = V_{max}\left(1 - \frac{2tan^{-1}(|\omega|)}{\pi}\right) \tag{5.14}$$

where $V_{max}$is a tunable parameter. The above controller scales down the forward velocity of the robot from V_{max} to 0 for non-zero ω in a nonlinear fashion. This allows the robot to slow down before making sharp turns allowing for smoother navigation while ensuring that the control output stays within the actuation limits of the robot. The above controller is asymptotically stable provided k_p, k_d, and V_{max} are greater than zero.

5.3.4. Modification to the control effort

Based on the estimated value of the augmented parameters $\widehat{\boldsymbol{A}}$ and $\widehat{\boldsymbol{B}}$ and the control action, $\boldsymbol{U}$, provided by the low-level controller, the modified control actions can be obtained as follows:

$$\boldsymbol{U}' = \boldsymbol{C}^{-1}\widehat{\boldsymbol{A}}^{-1}(\boldsymbol{C}\boldsymbol{U} - \widehat{\boldsymbol{B}}) \tag{5.15}$$

Note that $\boldsymbol{C}$ is not a square matrix and as such a Moore-Penrose pseudo inverse is calculated to obtain $\boldsymbol{C}^{-1}$**.** The updated control law, when applied to the augmented system as given by Eq. 6, converts it into the ideal system given by Eq. 2. Analogously, the ADRC takes care of the disturbances with the low-level controller essentially acting on the ideal model, which is a proven stable system.

5.3.5. Novelty of the proposed approach

Even though Extended Kalman Filter based methods have been used in the past to estimate the effects of slip in AGV motion towards improving localization [160], none of the prior work has used it towards estimation and compensation of the effects of slip.

Additionally, in contrast with the previous methods, the proposed method can account for the effect of time varying slip as well as nonlinear actuator/drive train dynamics, the resistance offered by the terrain on the robot, and other factors that can vary with time through the use of a generalized model. Another major advantage of the proposed method is that due to its hybrid architecture the overall approach can be used with other state of the art path following or trajectory tracking controllers. Even though the proposed work uses a simple "Go-to-Goal" behavior as the low-level controller, any stable path following controller mentioned in literature [161]–[164] can be used instead.

5.4. Experimental validation

In order to validate the performance improvement provided by the proposed ADRC architecture as compared to using the low-level controller alone, here after referred to as PD controller, path following trials were conducted. The experimental validation was performed using the hybrid mobility platform STORM, developed in the Robotics and Mechatronics Lab at Virginia Tech [126]. The system, weighing around 9 kg, has a combination of tracks and wheels that allow it to move in all directions along with ultrasonic sensors and cameras that allow for obstacle avoidance and visual odometry applications. For the experiments described in this chapter, only the tracked locomotion mode was used. The robot is fitted with an ODROID XU4 single board computer with Robot Operating System (ROS) Indigo [165] running onboard.

For the ADRC implementation, encoders on the left and right tracks of the robot were used to calculate the forward and angular velocity executed by the robot as input for the prediction step of EKF. The state feedback, (x, y, θ), for the correction step of the EKF was obtained using the POZYX positioning system [166] fitted on the robot. POZYX is an ultra-wideband (UWB) based positioning system that uses four anchors placed on the perimeter of the experimental area along with a tag placed on the robot. POZYX provides three dimensional position and orientation of the tag with an accuracy of up to one centimeter in positioning and one degree in orientation. Even though this chapter describes the use of POZYX, the proposed architecture could work with any absolute positioning system such as the LOSA[117], VICON, or D-GPS. The STORM module mounted with the POZYX tag along with the dimensions of the robot is shown in Fig. 5.3.

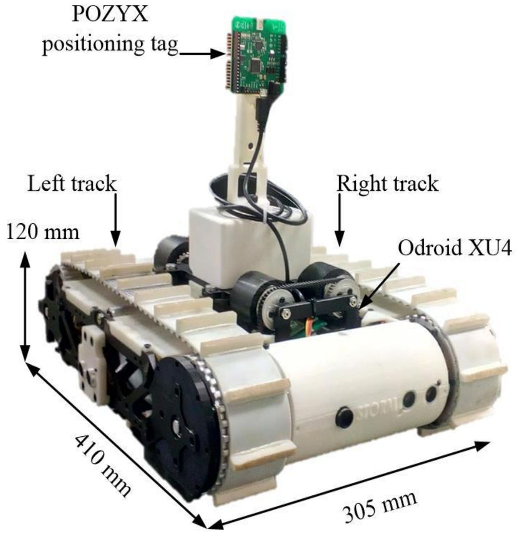


Figure 5.3: Experimental setup STORM mounted with POZYX unit

The entire ADRC architecture including the EKF was implemented as a series of ROS nodes running onboard the robot for the duration of the experiments. Separate ROS nodes were created for the EKF, low-level controller, ADRC, and hardware interfaces for obtaining sensor data and dissemination of motor commands. The EKF and the POZYX positioning node were run at 50Hz, while the ADRC architecture consisting of the low-level controller along with the modification to the controls was run at 10Hz.

The values of the $\boldsymbol{Q}$ matrix and $\boldsymbol{R}$ matrix used in the EKF are given below;

$$\boldsymbol{Q} = \begin{bmatrix} 0.015 & 0.0 \\ 0.0 & 0.015 \end{bmatrix}, \boldsymbol{R} = \begin{bmatrix} 0.001 & 0.0 & 0.0 \\ 0.0 & 0.001 & 0.0 \\ 0.0 & 0.0 & 0.00012 \end{bmatrix} \qquad (5.16)$$

Note that for the above values, $\boldsymbol{Q}$ and $\boldsymbol{R}$ are assumed to be uncorrelated. The augmented parameters in the EKF were initialized as $\boldsymbol{A}_0 = 1.0, \boldsymbol{B}_0 = 0.0, \dot{\boldsymbol{A}}_0 = 0.0,$ and $\dot{\boldsymbol{B}}_0 = 0.0,$ as shown in Fig. 5.2. The initial value of the covariance matrix was set to be a non-zero diagonal matrix with 0.01 for the diagonal elements corresponding to the states and 1.0 corresponding to each of the augmented parameters. All of the above values were chosen heuristically.

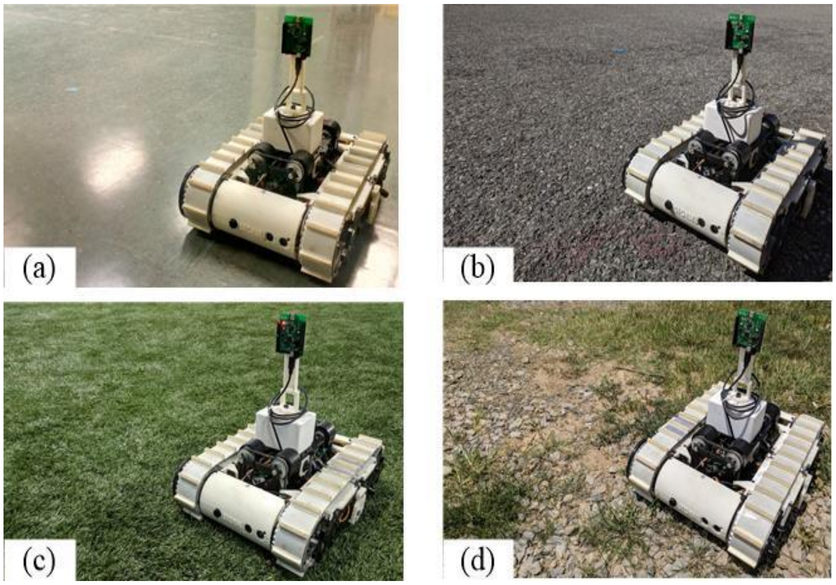


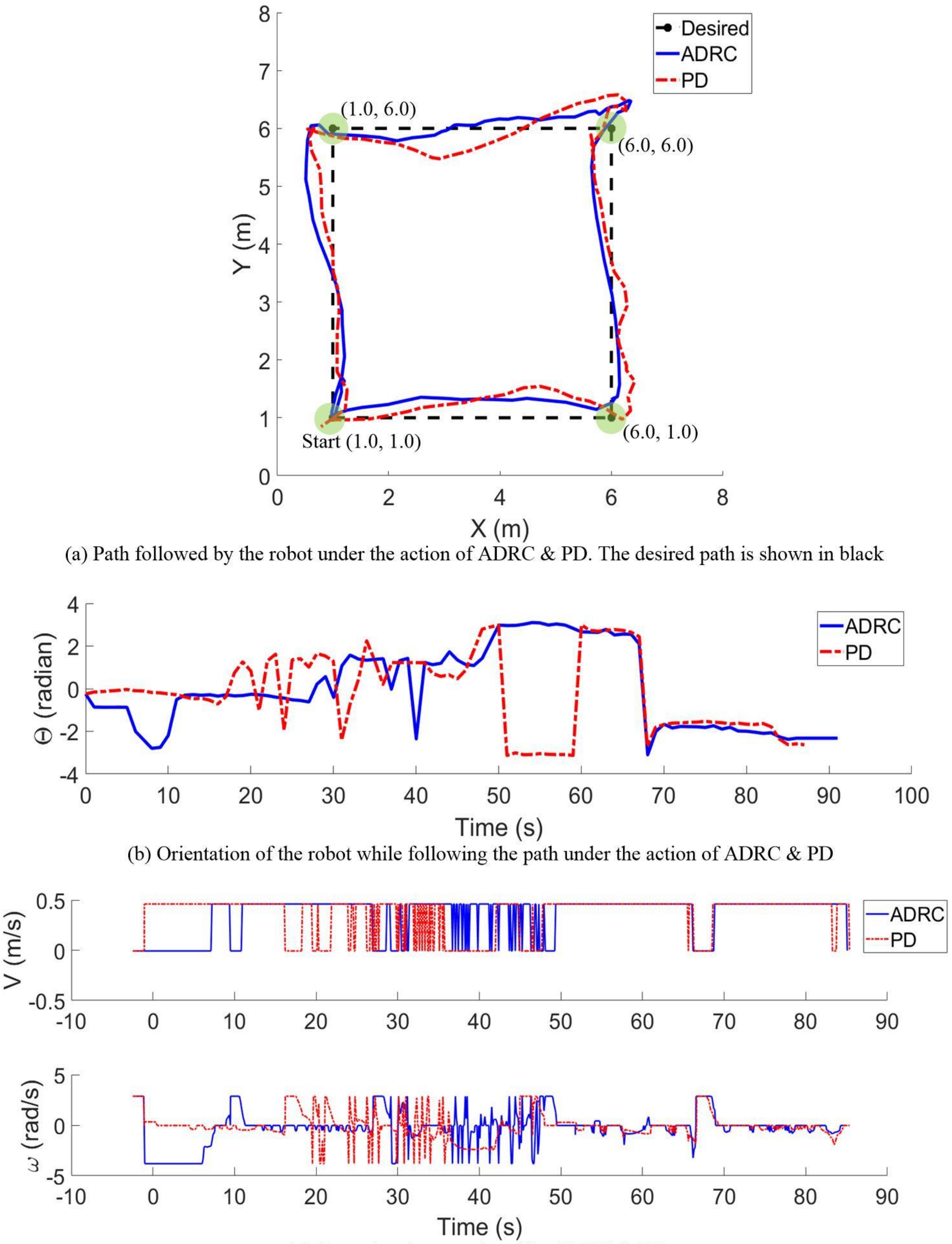
Figure 5.4: Experimental validation STORM on different terrains. (a) Vinyl flooring, (b) Asphalt (c) Artificial Turf (d) Grass and gravel

The experimental trials were performed over four different terrain conditions: vinyl flooring, asphalt, artificial turf, grass, and gravel terrain as shown in Fig. 5.4. The experiments over vinyl flooring and artificial turf were conducted indoors whereas asphalt, grass and gravel terrains were tested outdoors. For the grass and gravel case, the terrain was uneven and the ground sloped in both X and Y directions. In addition, the terrain varied from grass to gravel over the length of the path followed by the robot. On each terrain, five trials with the proposed ADRC architecture and five trials with the simple PD control were performed.

It is important to note that the low-level controller and the waypoints are kept the same for both the cases. The EKF for the PD implementation used the ideal model given in Eq. 2. The ADRC, on the other hand, used the augmented model and the estimated values were used to modify the control inputs. For each terrain, four waypoints laid out as a square are given to the low-level controller. The size of the square is different in case of each terrain based on the amount of space available. In addition, the POZYX data proved to be too noisy for a square perimeter larger than 10 m on each side.

5.5. Results and discussion

For the square waypoints given in each trial, the "Go-to-goal" behavior used as the low-level controller should cause the robot to make a perfect square, intersecting each of the waypoints, provided there is no slip. The desired perfect square path along with the actual path followed by the robot under the action of the ADRC and simple PD control is shown for asphalt and vinyl are shown in Figs. 5.5 and 5.6. As shown by the results, the varying amount of slip on each of the terrains causes the robot to deviate from the square path. The variation in orientation of the robot with respect to time and the control actions, V and ω, applied to the robot under the ADRC and simple PD are also shown in Figs. 5.5 and 5.6. The position and orientation of the robot as demonstrated in the plots are from the EKF running on board the robot.

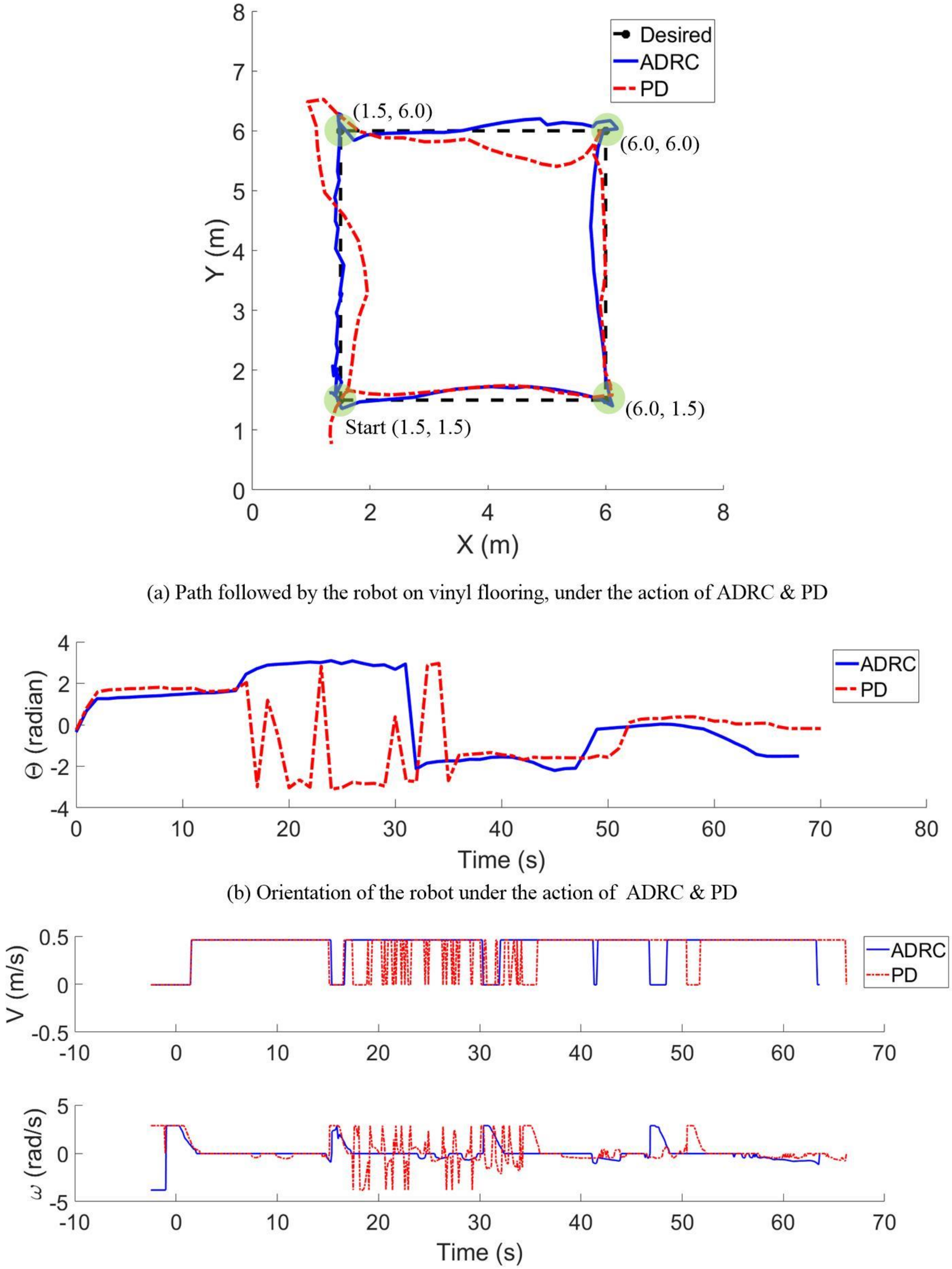


(a) Path followed by the robot under the action of ADRC & PD. The desired path is shown in black

(b) Orientation of the robot while following the path under the action of ADRC & PD

(c) Control actions produced by ADRC & PD

Figure 5.5: Path tracking results on asphalt



(a) Path followed by the robot on vinyl flooring, under the action of ADRC & PD

(b) Orientation of the robot under the action of ADRC & PD

(c) Control actions produced by ADRC & PD

Figure 5.6: Path tracking results on vinyl concrete flooring

5.5.1. Improvement in path following

As mentioned in the introduction, the performance of the skid-steer approach over different terrain conditions is greatly dependent on the amount of resistance offered by the terrain. The STORM module that was used for testing is fitted with tracks that have protruding treads as was shown in Fig. 5.3. When the platform moves over a non-deformable flat terrain such as the case of the vinyl flooring, the tips of the treads are the only part of the robot that makes contact with the ground. This causes excessive slipping of the robot, often resulting in the controller over correcting for the errors. In the case of artificial turf and grass or gravel on the uneven terrain, the treads dig into the terrain due to the weight of the robot and thereby offer additional resistance to the motion, especially while turning. This is true even in case of asphalt where the small cracks in the surface act as gripping points for the treads. The difference is that the terrain is deformable in case of artificial turf, grass and gravel as compared to the case of asphalt. As such, asphalt offers the maximum resistance to the skid-steer motion of the robot. The effect of the varying amount of resistance with the fixed maximum torque on the motors results in varied path following performance for the robot on each terrain, particularly under the action of the PD controller.

The ADRC, on the other hand, captures the varying nature of the terrain in the augmented parameters and thereby provides similar path following performance on different terrain conditions. This is evident from the smoother path tracked by the ADRC for each of the terrain conditions. Since the low-level controller used here adjusts the ω based on the error in orientation, the jerky motion produced by the PD control is particularly evident from the oscillatory nature of the plots showing the orientation of the robot. This is also observable in the plot of commanded velocity in each case. The ADRC produces relatively smooth motion command as compared to PD in each case.

Similar performance may be obtained by a properly tuned PID controller. For instance, in Fig. 5.5(a) and 5.6(a) the deviation of the PD controller from the desired path especially at the corners could be due to the overshoot of the controller rather than slip. This could be avoided by retuning the controller, but it will have to be re-tuned for each new terrain condition or whenever there is a significant change to the hardware of the robot such as the type of tracks being used. In this regard, the proposed approach can be considered to be an adaptive self-tuning controller. The major factor that distinguishes the proposed controller design from existing adaptive controllers is that the ADRC achieves self-tuning through the augmented parameter values in the modified control law and the augmented parameters themselves are treated as a disturbance being estimated inside the EKF. This allows the entire approach to be designed without an accurate model, and remain agnostic toward the noise in sensor measurements. Note that the applied low-level controller reduces the forward velocity of the robot whenever the desired ω is high as explained in Section 5.3. This is evident from the results as the V applied is zero whenever the ω is high.

5.5.2. Estimation of the augmented parameters

Figure 5.7 shows the estimated value of the augmented parameters $\boldsymbol{A}$ and $\boldsymbol{B}$, for all five trials on asphalt. The average nature is shown in red for each parameter. As shown by the plots, the time evolution of the parameters follows a general trend. The fact that the parameters change with time for any given trial, indicates that the nature of slip varies over time even for a specific terrain. It should be noted that, even though for a given terrain condition, the overall nature of the terrain remains the same such as asphalt or artificial turf, there were variations encountered by the robot that were not uniform across the trials. Examples include but are not limited to, one of the tracks going over a large enough rock or even a patch of sand, the treads getting caught on patches of grass or turf, pebbles getting stuck between the track and the driving sprocket.

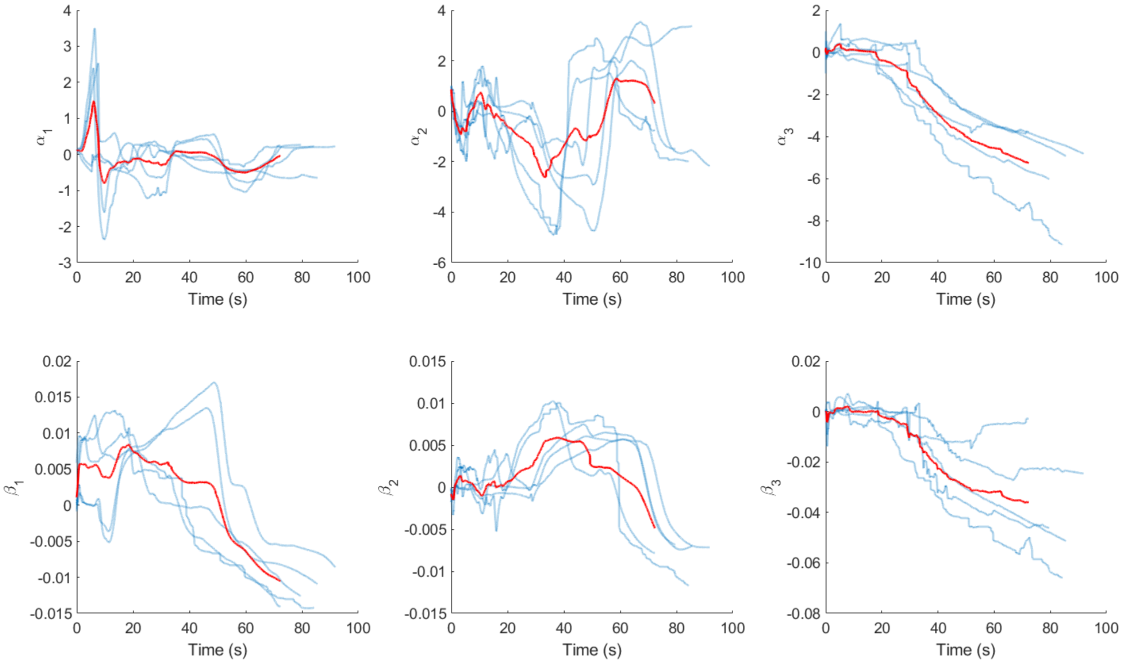


Figure 5.7: Variation in value of the added parameters when the robot follows the path on asphalt under the action of the ADRC

Once the robot encounters the above mentioned variations in the terrain, the overall behavior of the system changes, which causes the low level controller to output different commands and the states of the robot, including the augmented parameters evolves differently. This is evident in the plots where for a given terrain condition some trials produced estimated values far different from the average, while still following the overall trend. In addition, the command to start the low level controller was given manually to the robot at the start of each experimental trial. This caused the robot motion and thereby the parameter evolution, to start at different time instances across the trials which introduced the minor shifts in the evolution of the parameters as seen from the figure. Tuning the $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices further could enable the EKF to maintain the general trend better despite the variations encountered for a given terrain. The fact that the augmented parameters follow a general trend for multiple trials in a given terrain, while varying significantly across different terrain conditions, opens up interesting possibilities as discussed in Section 6.

The evolution of the covariance terms corresponding to the augmented parameters for each of the terrain condition filter converges quickly. In order to quantify the performance for each trial, energy spent, time taken by the robot to complete the path, and the mean cross-track error (MCTE) for each controller in all of the terrain cases are compared. The mean values for each of the parameters over five trials for each terrain and their standard deviation (value given in parenthesis) are shown in Fig. 5.8. The total energy spent completing the path is proportional to:

$$E = mV^2 + I\omega^2 \tag{5.17}$$

where m is the total mass of the robot and I is the rotational inertia of the main body about the axis normal to the plane of its movement. Exact measurements of the total energy spent by the robot rely on the terrain interaction forces, which in turn depend on the properties of the terrain, and individual inertia values for each of the moving components of the robot. This is beyond the scope of this work.

Since the onboard controller taking the measurements is discrete, the value of E for each trial was obtained by numerically integrating the value of E for each iteration of the controller and then averaging over the total time taken to complete the trial. The forward and angular velocity executed by the controller as measured by the encoders are used to calculate E.

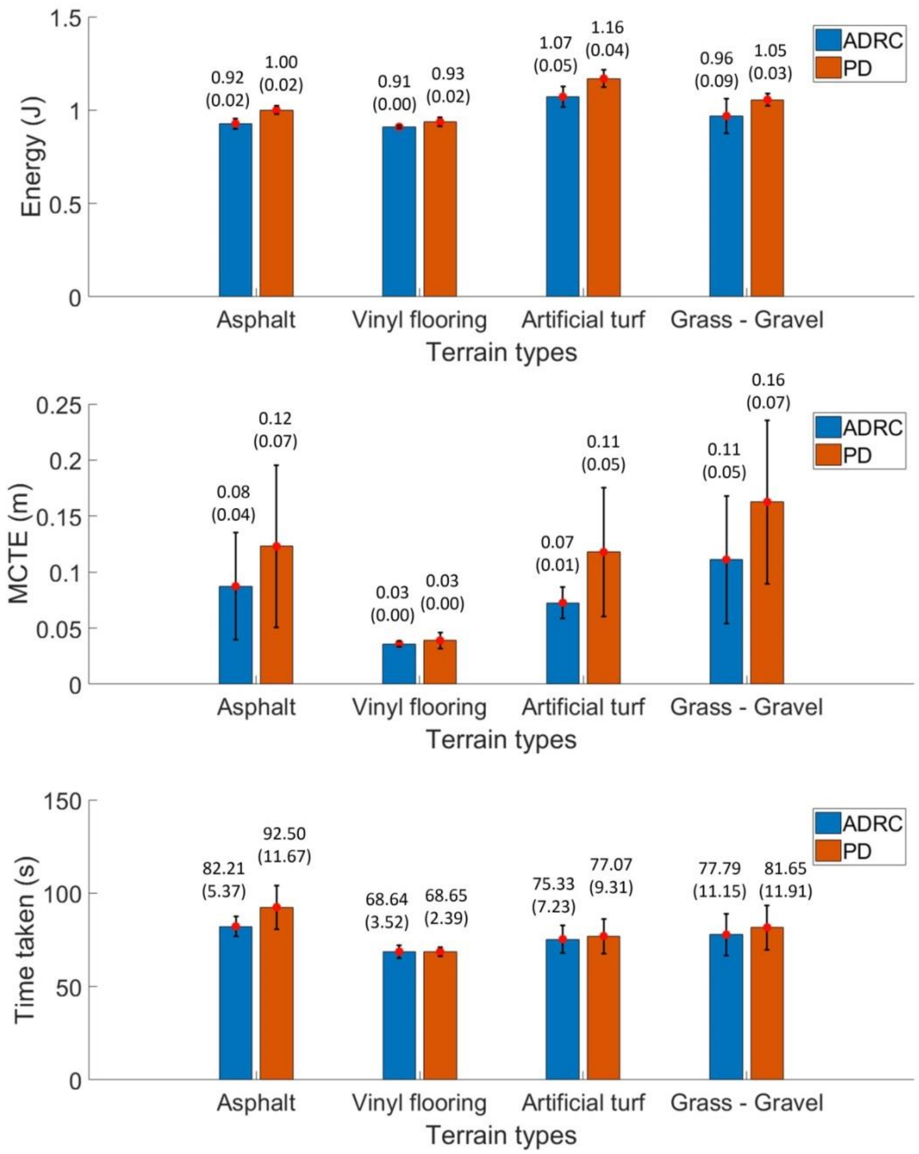


Figure 5.8: Performance comparisons of both controllers over different terrain conditions

The cross-track error is the Euclidean distance between the robot and the closest point on the path (in this case the straight line joining the previous and the next waypoint) computed during each iteration of the controller. The mean cross-track error (MCTE) is the average of the cumulative cross track error over the total number of trials. Based on the results as shown in Fig. 5.8, the proposed controller performs better than PD in each case.

5.6. Conclusion and future work

This chapter proposed the design of ADRC to estimate and compensate for the effects of slip during path following on challenging terrain conditions using autonomous tracked vehicles. The proposed ADRC architecture used a generic model that can account for the scaling and shift in the states of the system due to the effects of slip through augmented parameters. An EKF observer was used to estimate the value of the augmented parameters for any given terrain condition. The estimated value of the parameters was then used to modify the control outputs from a low-level controller in order to compensate for the effects of slip which resulted in better path following performance. Even though this chapter described the use of a simple “Go-to-goal” behavior as the low-level controller, the proposed approach could work with a more complicated behavior that also controls the heading and curvature of the path followed by the robot in relevant applications.

The performance of the proposed ADRC architecture was compared to PD controller through experimental trials involving four different terrain conditions (vinyl flooring, asphalt, artificial turf, grass, and gravel), with ten trials on each terrain (five under ADRC and five under PD). The path followed by the robot change in orientation with respect to time and the final control actions applied to the robot under the action of the ADRC and simple PD control were compared in each case. The time taken by the robot, energy consumed, and Mean Cross Track Error averaged over five trials under the influence of the ADRC and PD control was also compared for each of the terrains.

As shown by Figs. 5.5, 5.6, and 5.8, the ADRC consistently showed better performance compared to the low level PD controller alone. The path tracked by the robot is closer to the desired path in case of ADRC and the motion is smooth as shown by the variation in orientation and the applied control actions. While the ADRC provided smooth corrections to the robot, the PD often overcorrected on the account of slip and resulted in a jerky motion. As shown in Fig. 5.8, ADRC provided significant improvement in MCTE; 33.33% on asphalt, 36.36% on artificial turf and 31.25% on grass-gravel. In addition, the use of ADRC improved the energy consumption by approximately 8% on asphalt, artificial turf and grass-gravel, and the time taken by about 11% on asphalt. Except on vinyl flooring where the performances of both controllers were similar, experimental results on all other terrain conditions showed significant improvement with the use of ADRC. It should be noted that the ADRC architecture used the same low-level PD controller, with the only difference being that the ADRC used the augmented parameters to modify the control input that emanated from the PD controller to additionally account for the effect of slip. Taking into account the above factors, any improvement in trajectory tracking could only be attributed to the improved performance exhibited by the ADRC architecture.

Even though this chapter assumed slip to be the major cause for the disturbance experienced by the robot based on existing literature, it could be due to other unknown or un-modeled kinematic and dynamic properties of the system. The fact that it could be from other sources can be considered as another advantage of the proposed control method in terms of improvement in trajectory tracking. Detailed analysis into whether the variations captured by the model is solely due to slip will require accurate slip estimation using additional sensors on the robot such as a free wheel with an encoder. Such an analysis is beyond the scope of this work and will be a part of future work.

The performance metrics as reported in the literature for any path following controller implementation depends greatly on the platform being used for testing, accuracy and precision levels of the sensors, and the type of terrain over which the system is being tested. Majority of the existing techniques for slip estimation rely on precise measurements of the velocity of the robot. As stated before, the major advantage of the proposed method is to use even noisy position estimates from the POZYX system to provide reliable path following performance. The noisy position data provided by the POZYX system onboard STORM makes reliable velocity estimation and subsequent slip estimation difficult. Even though this highlights the capability of the proposed method to provide reliable trajectory tracking performance in presence of noisy data, this prevents implementation of other existing techniques on the available platform. In this regard, a detailed performance comparison between the proposed method and existing state of the art methods is beyond the scope of this work. Future testing will rely on using different sensors including RTK GPS or vision-based tracking systems to obtain reliable slip estimations for performance comparison. In addition, the developed architecture will be further tested on larger robots such as the HMMR [167]–[173] on more challenging outdoor terrain conditions including sand and mud. The path following trials described in this chapter was conducted over a small region due to the limited range of the POZYX positioning systems. The use of RTK GPS system will allow for longer trials in outdoor scenarios. The improvement in trajectory tracking performance with the ADRC is expected to be more pronounced in the case of longer trials, which will be analyzed as a part of future work.

CHAPTER 6

REAL-TIME TERRAIN ESTIMATION FOR TRACKED ROBOTS

6.1. Introduction

While navigating in unstructured terrain conditions, through autonomous or tele-operated manner, performing high level planning and control requires detailed knowledge of the terrain. Vision based methods can be used for terrain estimation, but they are highly susceptible to changes in ambient light and other environment conditions such as presence of smoke, fog, or dust. Other sensor based approaches including LIDAR and Inertial Measurement Units (IMU) are also not robust enough. For example, while a rocky slope and an expanse of loose sand at same gradient may look similar in a 3D scan, the robot can fail to climb or become entrenched on the sandy slope depending on a number of factors. Currently, for systems that are deployed in the field, high level navigation decisions are made by a human operator based on feedback from the camera or LIDAR systems. The most recent DARPA Subterranean (SubT) Challenge [174] requires roboticists to address the problem of terrain estimation in order to enable fully autonomous navigation in unstructured conditions.

In addition to improving the autonomous navigation capabilities of terrestrial rovers, and search and rescue robots, real-time terrain estimation can also benefit commercial vehicles such as cars and trucks through improvements in advanced driver assistance systems (ADAS) which in turn leads to fully autonomous driving. Generating accurate predictions about how the vehicle is going to respond to various maneuvering commands from the onboard planning system is vital for safe autonomous driving. In this regard, ascertaining road terrain parameters is crucial, as vehicle behavior and the safe limits, including maximum safe acceleration and braking limits, vary drastically with changes in the terrain. There exists studies by different groups in using a wide variety of sensing modalities mounted on vehicles including radar [175], images from a monocular camera [176], and RGBD point clouds obtained from a stereo camera pair [177] to estimate the terrain conditions. In most cases, the goal was long range binary classification of the terrain into navigable and non-navigable areas aimed at both on-road and off-road autonomous navigation.

In comparison to the existing research, this work proposes a method for real-time classification of different terrain types based on the state evolution of a ground robot, specifically the measured change in pose of the robot for a known control input. An offline technique to determine the features of interest from the collected training data is presented. A Support Vector Machine (SVM) classifier is trained on the principal components (PCs) of the weights associated with the features of interest to perform terrain estimation. The online moving window least squares estimation technique used to update the weights in real-time is also presented along with the SVM architecture.

By using a trained SVM to perform terrain estimation based on the collected state evolution data, the proposed method does not require dedicated sensing modalities solely for terrain estimation. As such, this method is generally applicable in all conditions where the robot can traverse. The training data was obtained from four different terrain conditions including vinyl flooring, asphalt, artificial turf, and grass-gravel. The proposed technique is validated using a skid-steer tracked robot, where the response to control inputs is significantly affected by terrain characteristics, over multiple simulated and real terrain transitions cases.

It should be noted that the proposed technique enables the robot to identify the terrain it is operating on, based on previously learned terrain types. This requires the robot to travel over a given terrain condition and collect data regarding how the robot pose changes for a given control input. As such it cannot be used to classify terrain into navigable versus non-navigable as the robot cannot collect motion data on non-navigable terrain. Readers interested in navigable versus non-navigable classification of terrain should refer to prior work by the authors that specifically addressed this issue [47].

6.2. Review of terrain estimation methods

Planetary exploration missions are a major application area where online terrain parameter estimation is used to improve navigation. Terrain estimation allows the rovers to operate with minimal human supervision in terrain conditions that are often challenging and previously unseen. To this extent, Iagnemma et al. [178] used a simplified form of classical terramechanics equations along with a linear-least squares estimator to compute terrain parameters in real-time. Since their approach relied on terramechanics equations, it assumed that the vertical load on the robot, torque applied on motors, sinkage of the robot, wheel angular speed, and wheel linear speed could be measured or estimated. This requires additional sensors on the robot. On a similar note, Reina et al. [179] proposed methods for wheel slippage and sinkage detection as applied to planetary rovers by comparing wheel encoder readings with gyro readings and motor current in order to detect slip. They also proposed a novel vision based method to detect wheel sinkage.

One of the major disadvantages of vision based methods is that they are susceptible to variations in ambient light and the presence of dust, smoke, or other environmental conditions. In addition, vision and range based methods are not good at detecting non-geometric hazards, such as in cases when the topmost layer of terrain is different from the underlying load bearing surface. To handle these issues Iagnemma and Brooks [180] have explored terrain classification based on the vibrations induced on the rover by the wheel-terrain interaction during driving. Their approach aimed at creating a stand-alone classifier by performing standard signal processing techniques on the data collected from a vibration sensor mounted on the wheel of the rover. On a similar note, Giguere and Dudek [181] attached an accelerometer at the end of a metallic rod to form a tactile probe. The acceleration patterns induced at the tip of the rod were analyzed to estimate the terrain conditions using a trained neural network. Similarly, Park et al. [182] used peak variances extracted from contact sensor data to perform terrain classification. Wolf and Sukhatme proposed the use of Hidden Markov Models and SVMs for semantic terrain mapping[183]. They used data provided by range sensors and odometers to classify the terrain into navigable and non-navigable regions. Weiss et al. [184] proposed the use of an SVM trained over raw vibration sensor data collected by the robot to perform terrain classification. They proposed that based on the terrain estimation data, the vehicle could adapt its driving style to better match the terrain being traversed. Their method was compared to five other existing vibration-based terrain classification methods in [185] and the results showed that significantly better performance. An interesting example of the application of environment identification to improve autonomous navigation was presented by Giguere et al. in [186]. They performed environment identification based on actuator and inertial sensor data to autonomously switch between walking and swimming for their amphibious hexapod, AQUA.

In addition to rovers, real-time estimation of terrain parameters has been explored in the context of commercial vehicles including cars and trucks. As mentioned by Wang et al. in [187], real-time terrain estimation can improve the performance of driver assistance systems by specifying terrain specific driving strategies aimed at improving factors like fuel efficiency. They proposed a two-stage road terrain identification approach for land vehicles using feature-based and Markov random field algorithms. As per their approach, feature-based identification results obtained using an accelerometer, camera, and LRF was improved upon by using a Markov Random Field to get optimal identification results. A more recent work by Khaleghian and Taheri looks at terrain classification methods using an intelligent tire [188]. The intelligent tire setup essentially consists of a tri-axial accelerometer attached to the tire inner liner, wheel speed sensors, and an additional accelerometer on the vehicle chassis along with a data acquisition system. In addition, the robot also had free wheels for accurately measuring slip. Based on the accelerometer readings and the wheel slip ratio, a fuzzy logic algorithm was used to perform terrain classification.

The survey by Khaleghian, Emami, and Taheri [189], lists recent advances in tire-road friction estimation towards improving vehicle’s stability, traction, and ABS controller performance for cars. This work provides a detailed summary of existing approaches including optical, acoustic, and tire tread sensors in addition to model based methods towards estimating tire-road friction. Even though commercial vehicles could also benefit greatly from terrain estimation and road parameter estimation, the rest of the chapter will focus on these techniques as applied to the motion of tracked skid steer robots in varying terrain conditions. The applicability to commercial vehicles will be further addressed as part of future work.

6.3. Proposed real-time terrain estimation method

A main point to notice based on the above review is that almost all existing works use dedicated sensing systems solely for estimating the terrain type or to estimate the terrain characteristics like roughness, friction, etc. A majority of the terrain estimation methods use contact sensors, non-contact sensors, or a combination of both. As such these methods are susceptible to the failure modes on each sensing type. On the other hand this work tries to explore the idea of using the robot itself as the sensor. This work is based on the hypothesis that the state evolution of the robot on any given terrain contains sufficient information to accurately perform terrain estimation, provided the control inputs applied to the robot are known [190]. This eliminates the need for dedicated sensors, which removes the associated limitations and failure modes, while also making the method generally applicable in all possible terrain conditions, where the robot can navigate. A similar approach towards estimating wind parameters using an RC helicopter has been explored before [191]. In addition to the real-time terrain estimation method, this work proposes techniques to improve navigation, especially in the areas of control, fault monitoring, autonomous path planning, and improving localization of mobile robotic systems traversing over varying terrain conditions.

The rest of this section is divided into three subsections, the first subsection explores the use of linear regression techniques to identify the features of interest that capture the effects of the terrain over which the robot is moving, the second subsection introduces methods for real-time estimation of the weights associated with the features of interest and the final subsection introduces PCA to reduce the dimensionality of the data, along with the SVM classifier to perform terrain classification based on the PCs of the estimated weight values.

6.3.1. Identifying the features of interest

Recently there has been significant interest towards using machine learning based model fitting to identify the features of interest that represent a physical process [192], [193]. The overall approach can be summarized as follows: experimental data is used to create an over-complete library of possible features of interest, consisting of the variables involved in the process, nonlinear functions of these variables and even their products. A regression method is then used to find the best fitting model over the experimental data. Based on the best fitting model, features that show high weights can be easily identified as the features of interest. These would be the features that contribute the most towards the governing equations that represent the physical process.

The above approach was used to identify the features of interest for the motion of a tracked skid steer robot over varying terrain conditions. Experimental data was obtained by driving the robot STORM [126] over four different terrain conditions with ten trails on each terrain. The terrain conditions consisted of asphalt, vinyl flooring, artificial turf, and a combination of grass and gravel. Each trial consisted of driving the robot in a square over the terrain with side length of four to six meters.

For the data collection trials the robot was manually driven over regions where the terrain conditions were uniform within the POZYX sensing limits. The path followed by the robot for the duration of the trial was recorded using the absolute positioning system

POZYX [166]. The control inputs were recorded using the wheel encoders on the robot. Each instance of the recorded data consists of previous 2D pose of the robot including position (x_t, y_t) and yaw angle θ_t, control inputs given to the robot in terms of linear and angular velocity (V_t, ω_t), along with the current position and orientation of the robot $(x_{t+1}, y_{t+1},\ \theta_{t+1})$. The experimental platform and the terrain over which it was run are the same as that in Chapter 3. Since the experimental data was collected with a light weight robot moving at low speeds, with negligible inertial effects, a kinematic analysis of the motion was considered sufficient.

The motion of differential drive platforms with one or more supporting castor wheels can be modeled accurately with a simple unicycle robot model, as given below.

$$\begin{aligned} x_{t+1} &= x_t + V_t \cos\theta_t \Delta t \\ y_{t+1} &= y_t + \ V_t \sin\theta_t \Delta t \\ \theta_{t+1} &= \theta_t + \ \omega_t\, \Delta t \end{aligned} \tag{6.1}$$

where (x, y) is the 2D position robot fixed frame {R} with respect to the global inertial frame {G}, and θ represents the orientation of {R}, with respect to {G} about the Z-axis. In Eq. 1 Δt denotes the time gap between the current robot pose (x_t, y_t, θ_t) and the pose at the next time step $(x_{t+1}, y_{t+1}, \theta_{t+1})$.

Existing studies [147], [149] have shown that the motion of skid steer platforms, especially in varying terrain conditions, deviate considerably from the ideal model given by Eq. 1. Modification to the ideal model have been suggested recently, including instantaneous center of rotation (ICR) based kinematics models [149], [150], to better approximate the motion of skid steer robots on varying terrain conditions. In contrast to existing approaches, linear regression on the collected data is used in this work to determine the features of interest and thereby formulate a more accurate governing equation for the motion of tracked skid steer robots in a data driven manner. The linear regression model used on the data takes the form of:

$$\mathbb{Y} = \mathbb{w}\mathbb{x} + \mathbb{b} \tag{6.2}$$

where the feature vector and the output vector are given by

$$\begin{aligned} \mathbb{x} = [\, & x_t, y_t, \theta_t, V_t\Delta t, \omega_t\Delta t, V_t\cos\theta_t\Delta t, \ldots \\ \ldots\ & V_t\sin\theta_t\Delta t, \omega_t\cos\theta_t\Delta t, \omega_t\sin\theta_t\Delta t]^T \\ \mathbb{Y} = \ & [\, x_{t+1}, y_{t+1}, \theta_{t+1}]^T \end{aligned} \tag{6.3}$$

For Eq. 2, $\mathbb{w}$ is the matrix of weights that linearly map the feature vector $\mathbb{x}$ to output vector $\mathbb{Y}$**,** k is the total number of features in the over complete library, and $\mathbb{b}$ contains the constant bias term for each equation. For the chosen feature vector $\mathbb{x}$ shown in Eq. 3, the total number of features k is nine. Therefore $\mathbb{w}$ is a 3x9 matrix, with w_{ij} being each element where i varies from 1 to 3 corresponding to linear regression models for $x_{t+1}, y_{t+1}, \theta_{t+1}$and j varies from 1 to k. The feature vector, $\mathbb{x}$ was chosen based on the existing research that suggested possible modifications to the unicycle model to better approximate the motion of skid steer vehicle on varying terrain [147]. To verify the feasibility of the proposed method, the ideal unicycle model, as given by Eq. 1, was simulated in MATLAB, without any added noise. The state variables (x, y,θ) were initialized at zero and control inputs (V, ω) were applied to the model to make the robot move in a circle, straight line, sinusoidal path, and also turn in place. The linear regression method, applied to the experimental data, was applied to the simulated ideal data as well. Performing least square regression on the experimental data allows us to determine the weights associated with each feature given in Eq. 3 for the governing equations corresponding to x_{t+1}, y_{t+1}, and θ_{t+1}. The number of data points collected for each terrain condition and the R squared test score for the best fit model are given in Table 6.1.

Terrain condition		Number of data points	R squared test score
Simulated model	x_{t+1}	7885	0.999
	y_{t+1}		0.999
	θ_{t+1}		0.989
Asphalt	x_{t+1}	7596	0.999
	y_{t+1}		0.999
	θ_{t+1}		0.992
Grass - gravel	x_{t+1}	7059	0.982
	y_{t+1}		0.967
	θ_{t+1}		0.993
Artificial turf	x_{t+1}	6673	0.994
	y_{t+1}		0.998
	θ_{t+1}		0.997
Vinyl flooring	x_{t+1}	6000	0.999
	y_{t+1}		0.998
	θ_{t+1}		0.996

Table 6.1: Model fitting result on each terrain condition

As per the R squared test score, the models given by the regression algorithm provide a good fit on the experimental data. Figure 6.1 shows the absolute value of the weights corresponding to each feature as obtained from the best fit model, for the different data sets.

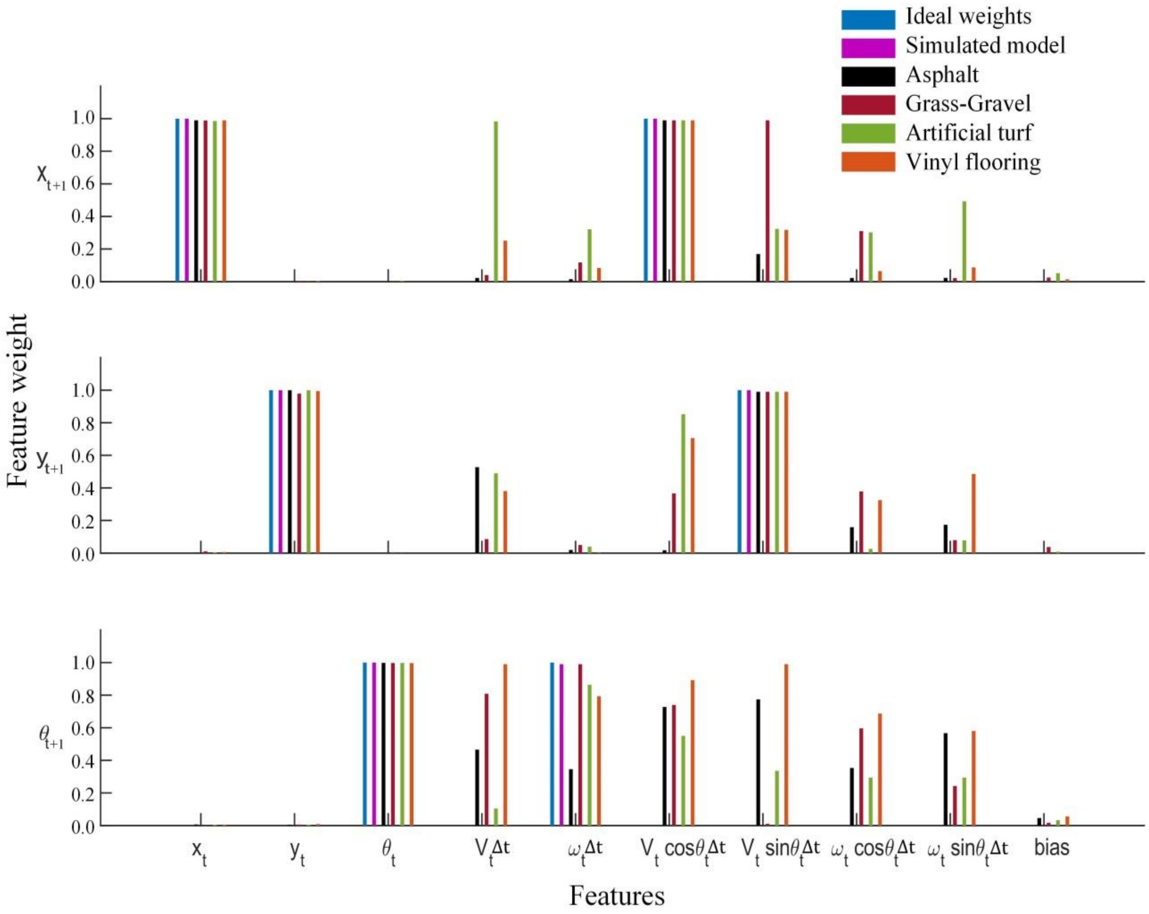


Figure 6.1: The absolute weights corresponding to each feature as obtained from the least squares fit

In Fig. 6.1, the features that show higher weights are the ones that contribute most towards the motion of the robot. By identifying the features that have high weights (>0.5) in at least one terrain condition, a more accurate governing equation representing the evolution of the robot's states can be formed:

$$
\begin{aligned}
x_{t+1} &= w_{11}x_t + w_{12}V_t\Delta t + w_{13}V_t\cos\theta_t\Delta t + w_{14}V_t\sin\theta_t\Delta t \\
y_{t+1} &= w_{21}y_t + w_{22}V_t\Delta t + w_{23}V_t\cos\theta_t\Delta t + w_{24}V_t\sin\theta_t\Delta t
\end{aligned}
\tag{6.4}
$$

$$\theta_{t+1} = w_{31}\theta_t + w_{32}V_t\Delta t + w_{33}V_t\cos\theta_t\Delta t + w_{34}V_t\sin\theta_t\Delta t + w_{35}\omega_t\Delta t + w_{36}\omega_t\cos\theta_t\Delta t + w_{37}\omega_t\sin\theta_t\Delta t$$

The advantage of the above approach is that it allows us to pick out the features that are most relevant to the physical phenomenon, while ignoring all of the non-relevant ones, completely based on the collected experimental data. It should be noted that limiting the loop back condition on θ between $[-\pi, \pi)$, will result in the linear regression failing to converge due to the inherent non linearity. To counter this, the collected data was adjusted to remove the loop back of θ, allowing it to continuously increase or decrease without limits.

In addition to deciding upon the features of interest Fig. 6.1 also offers some additional insight worth discussing. The ideal weights for each feature as per Eq. 1 are given in blue. The weights predicted by least square fit for the simulated data is shown in magenta. As seen from the figure the ideal weights and the simulated data match perfectly as expected. This validates the fact that the linear regression fitting works as expected. On the other hand, the experimental data collected from each terrain gives a different set of weights as compared to the ideal model. For example, as per the ideal model x_{t+1} should depend only on x_t and $V_t\cos\theta_t\Delta t$, with weights equivalent to 1.0. This is true for the data generated from the ideal model, but the experimental data obtained from most of the terrain conditions depend on $V_t\sin\theta_t\Delta t$ as well. Similarly, the equations for y_{t+1} and θ_{t+1} deviate from the ideal model for the experimental data. Another important factor to note is that each terrain shows different optimal value of weights for a given feature of interest. It can be inferred that the variations in the optimal value of weights is characteristic to the terrain over which the robot is moving. In other words, the weights associated with the features of interest holds characteristic information about the terrain.

It is possible that some of the above features may not contain characteristic information about the terrain, and as such may not contribute much for the terrain estimation application. It is also possible that by choosing all features with weights > 0.5 in at least one terrain condition, some of the weights could be redundant. However, these problems will be addressed through the use of PCA over the estimated weights, as explained later in Section 3.3.

The model given in Eq. 4 can be used in the real-time estimation of the weights, provided the pose data (x, y, θ) of the robot at time t and $t + \Delta t$, along with the control input (V, ω) is available. The estimated value of the weights in turn can be used to identify the terrain over which the robot moves. These points will be explored in greater detail in the following section.

6.3.2. Real-time estimation of weights

Based on the generalized model given in Eq. 4, the following section will explain the use of a moving window least squares estimator to obtain the best fitting value of the weights corresponding to each feature in real-time. As mentioned in the previous section, at the least, a noisy estimate of the robot's pose is necessary to perform a least square estimation of the weights. This is inherently necessary information for any autonomous navigation application and can be obtained from different pose estimation sensors such as the POZYX[166], LOSA[117], Real-time kinematic GPS, visual odometry etc. For the purpose of this work data obtained from the POZYX system was used.

Based on Eq. 4, Eq. 2 can be modified as;

$$\mathbb{Y} = \mathbb{w}\mathbb{x} \tag{6.5}$$

It should be noted that while $\mathbb{w}$ is still a 3x9 matrix, only 15 weights are non-zero (four for x_{t+1}, four for y_{t+1}, and seven for θ_{t+1}) as denoted by Eq. 4. In theory, all of the unknown weights could be estimated accurately with just 15 measurements, but there are some practical limitations to this. The POZYX system or any other positioning system working in outdoor environments is susceptible to noise. As such, using the minimum number of measurements will result in a noisy estimation of the unknown weights. Therefore, a large number of measurements needs to be used with a moving window least squares estimator to reduce the effect of noise and improve accuracy. On the other hand using a very large window will require a large set of prior measurements before the weights can be estimated. It will also lead to high computational cost. Moreover, with a large enough window it is possible to filter out the unique effects of the terrain, which could adversely affect the terrain estimation applications that will be discussed later. For the purpose of this work the filter size was manually tuned to 400 samples to yield the best performance. It should be noted that the moving window starts the estimation only after receiving the first 400 samples which takes about four seconds based on the current setup.

6.3.3. Application to terrain estimation

This section describes the use of Support Vector Machine (SVM) [194] in performing terrain classification based on the estimated value of the weights. As given by Eq. 4, the estimator returns 15 weights at each instance, but it is possible that the some of them are interrelated, resulting in redundancy of information.

Principal component analysis is a popular method used in machine learning community to remove redundancy of information. The principal components are a linear combination of the original variables that are orthogonal to each other. For the proposed application, even though the total number of PCs is the same as the number of estimated weights, the first ten PCs are sufficient to capture more than 98% of the data. It should be noted that, PCA is essentially eigenvalue decomposition and therefore the chosen PCs are the first ten eigenvectors of the covariance matrix of the estimated weights, in decreasing order of eigenvalue. Using the orthogonal transformation given by the first ten PCs, the estimated values of the 15 weights w_{ij} as given by Eq. 4 were transformed into reduced dimensional data along the PCs $\mathrm{w}_{\mathrm{pi}}\,(\mathrm{i}=1-10)$. This reduced dimensional data transformed along the PCs, hereafter referred to as the PCs of the estimated weights, was used in performing terrain estimation. By using the PCs of the estimated weights for the terrain estimation, the computational load in performing the estimation is reduced while maintaining an acceptable amount of accuracy.

SVMs have been used previously in terrain estimation, specifically on vibration based terrain estimation techniques [184], [185]. They allow for nonlinear classification based on the kernel function used inside the SVM. For the proposed terrain estimation technique an SVM with a Radial Basis kernel function was used on the reduced dimensional data. For the SVM implementation the Statistics and Machine Learning Toolbox in MATLAB [195] was used. A *one- versus-rest classification* was implemented using multiclass error-correcting output codes model containing multiple SVM binary learners as provided by the MATLAB Toolbox. Before training the SVM, the collected data was randomly divided into training and test data sets in 3:1 ratio.

Figure 6.2 shows the first two PCs of the collected data with ground truth marking and the SVM terrain estimation over the whole range of values. Using ten PCs gives 85.27% accuracy on the training data and 84.3% accuracy on the test data.

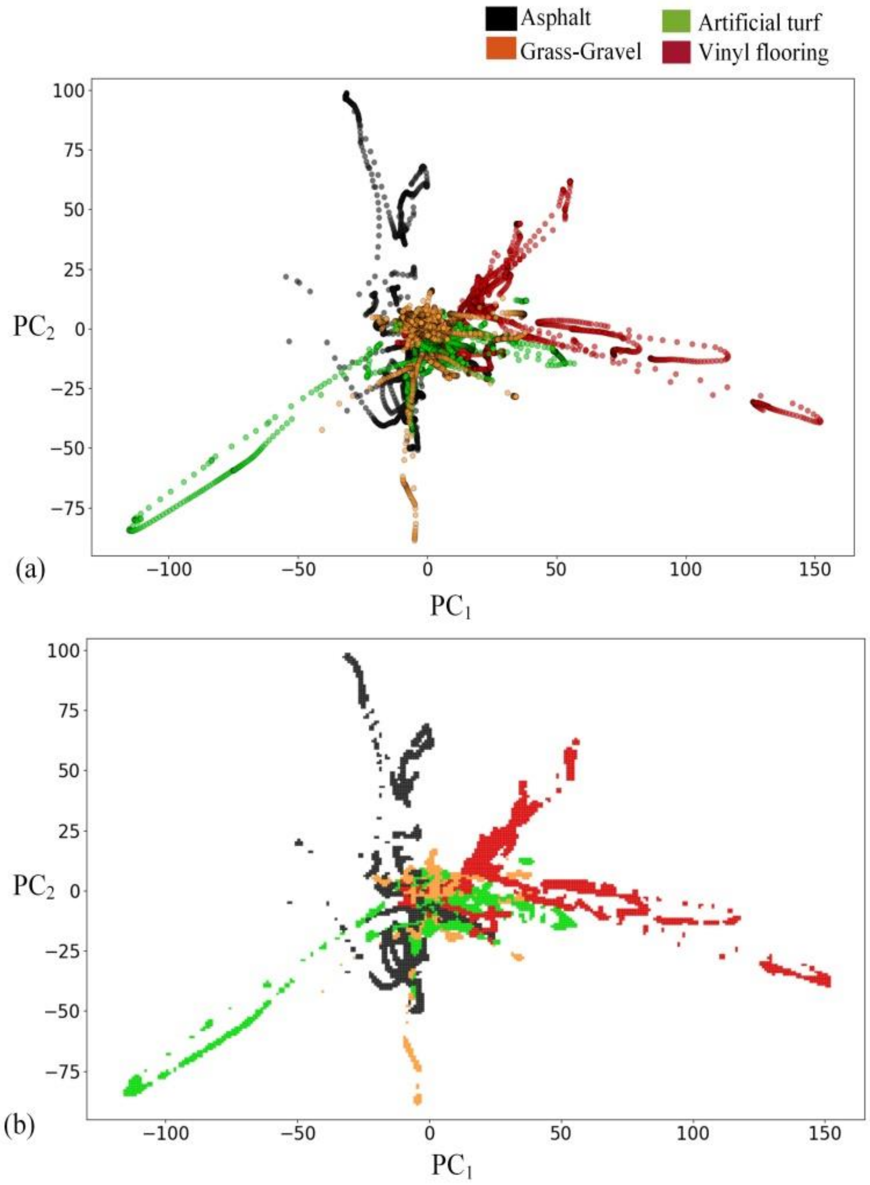


Figure 6.2: The first two PCs of the estimated weights. The figure shows (a) ground truth and (b) the classification results from the SVM. The magnitude of the first PC is along the X axis and the second along the Y axis

The confusion matrix showing the performance of the trained SVM is shown in Fig. 6.3.

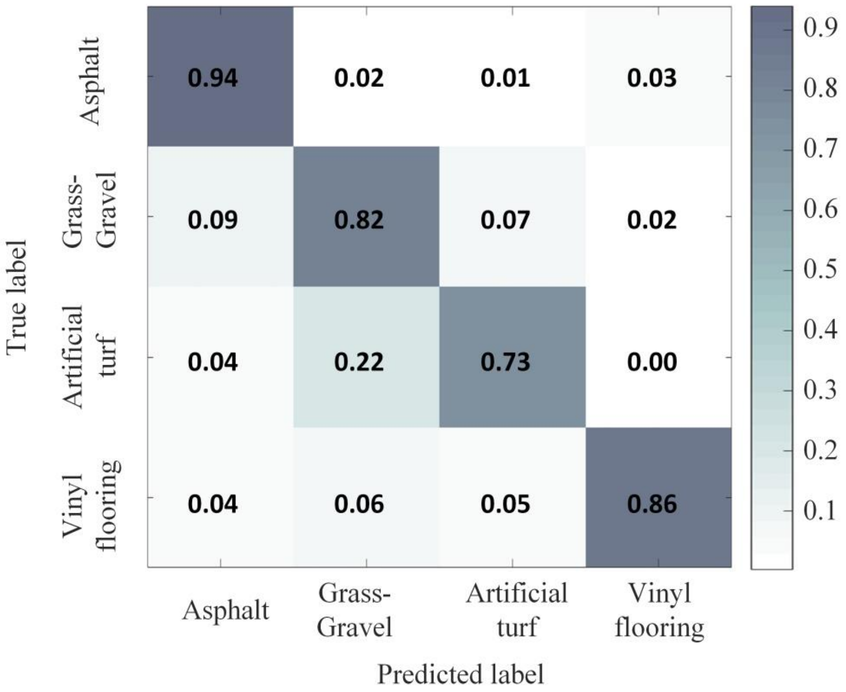


Figure 6.3: Normalized confusion matrix obtained from the trained SVM terrain classification

Based on the results provided, the proposed method performs well for terrain estimation application. It is important to note that, even though the exact value of the weights themselves could depend on the terrain as well as on the characteristics of the robot including the type of tracks, treads used on the track, presence or absence of suspension system and their layout. By using labeled data collected with the same robotic platform over different terrain conditions, the trained SVM is able to disregard the common factors namely the characteristics of the robot, and instead focus on the terrain relevant features for performing better estimation. Figure 6.4 summarizes the overall approach, including all the steps for the online terrain estimation and the offline training that was described in this section, along with the results that are obtained at each intermediate step.

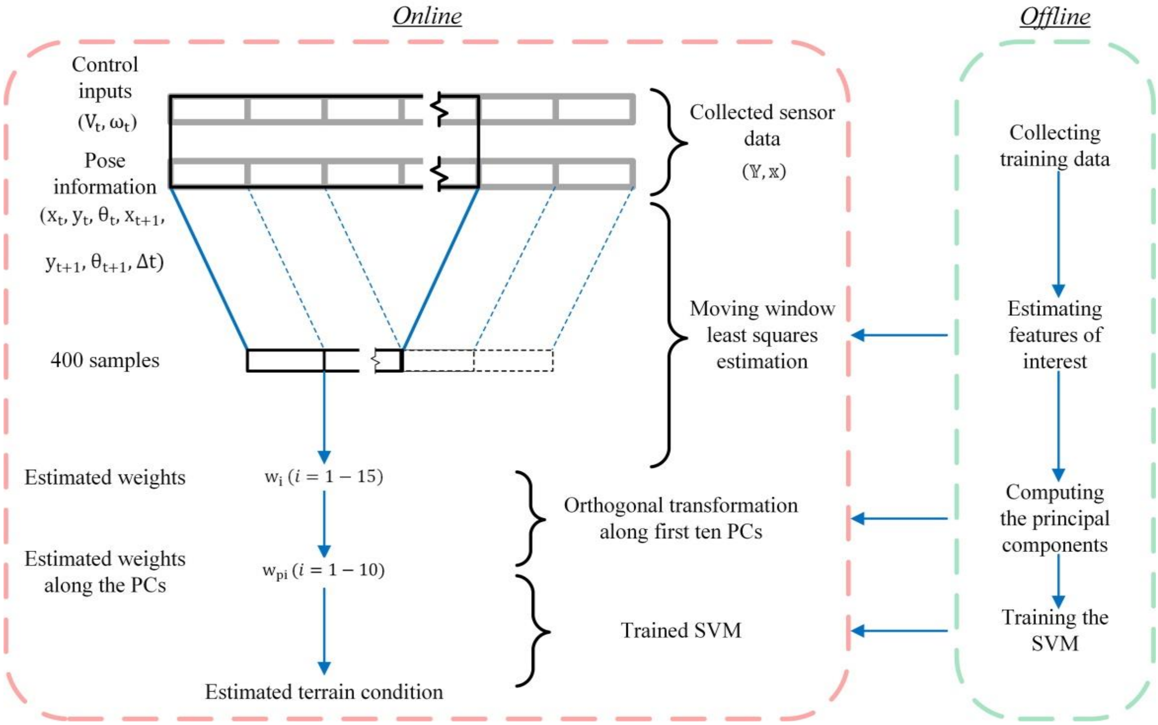

Online
Offline
Control inputs
(V_t, ω_t)
Pose information
$(x_t, y_t, \theta_t, x_{t+1}, y_{t+1}, \theta_{t+1}, \Delta t)$
Collected sensor data
$(\mathbb{Y}, \mathbb{x})$
400 samples
Moving window least squares estimation
Estimated weights
$w_i\ (i = 1 - 15)$
Orthogonal transformation along first ten PCs
Estimated weights along the PCs
$w_{pi}\ (i = 1 - 10)$
Trained SVM
Estimated terrain condition
Collecting training data
Estimating features of interest
Computing the principal components
Training the SVM

Figure 6.4: Flow chart showing the working of the proposed terrain estimation algorithm

The offline steps for collecting the training data, estimating the features of interest, estimating the principal components from the training data, and the training of the SVM itself needs to be performed only once. These steps have to be repeated only if the locomotion model of the robot changes, such as in the case of using a car-like robot or if additional terrain categories need to be added. In contrast to solid mechanics or finite-element modeling methods the data driven approach presented here allows the SVM to learn distinct terrain types based on the labeled data provided during the offline training phase. Even though the a-priori training is computationally intensive, it is performed offline, whereas the computationally efficient estimation can be run in real-time.

6.4. Validation of the proposed approach

The performance of the trained SVM for real-time terrain estimation applications was validated using simulated terrain transition data, generated from the collected real-world training data. Even though the training data was collected by driving the robot on four uniform terrain conditions separately, the collected data samples from different terrain conditions could be stacked together to simulate robot motion over varying terrain conditions.

The moving window least squires estimation of the weights requires 400 consecutive samples. Stacking random samples from different terrain conditions could result in unrealistic estimates of the weights. The SVM on the other hand looks at each individual sample of the ten PCs to produce terrain estimations. In order to validate the performance of the proposed technique in detecting terrain transitions, artificially generated test cases simulating terrain transitions was formed using the real-world data. The PCs of the estimated weights for the entire dataset was generated separately for each terrain condition and 500-600 samples from each individual terrain conditions were stacked together in a random order such that it would appear as if the robot was transitioning from one terrain condition to another for the trained SVM. The trained SVM was used to estimate the terrain condition from the stacked PCs of the estimated weights and compared with the ground truth marking on the dataset. The results are shown in Fig. 6.5.

It is clear from the results that the proposed approach has the ability to recognize terrain conditions with a high degree of accuracy, 93.84%, with fast detection of terrain transitions. It should be noted that instead of state evolution data, the PCs of the estimated weights were given directly to the SVM. As such, it is able to detect change in terrain conditions immediately, without the delay caused by the moving filter.

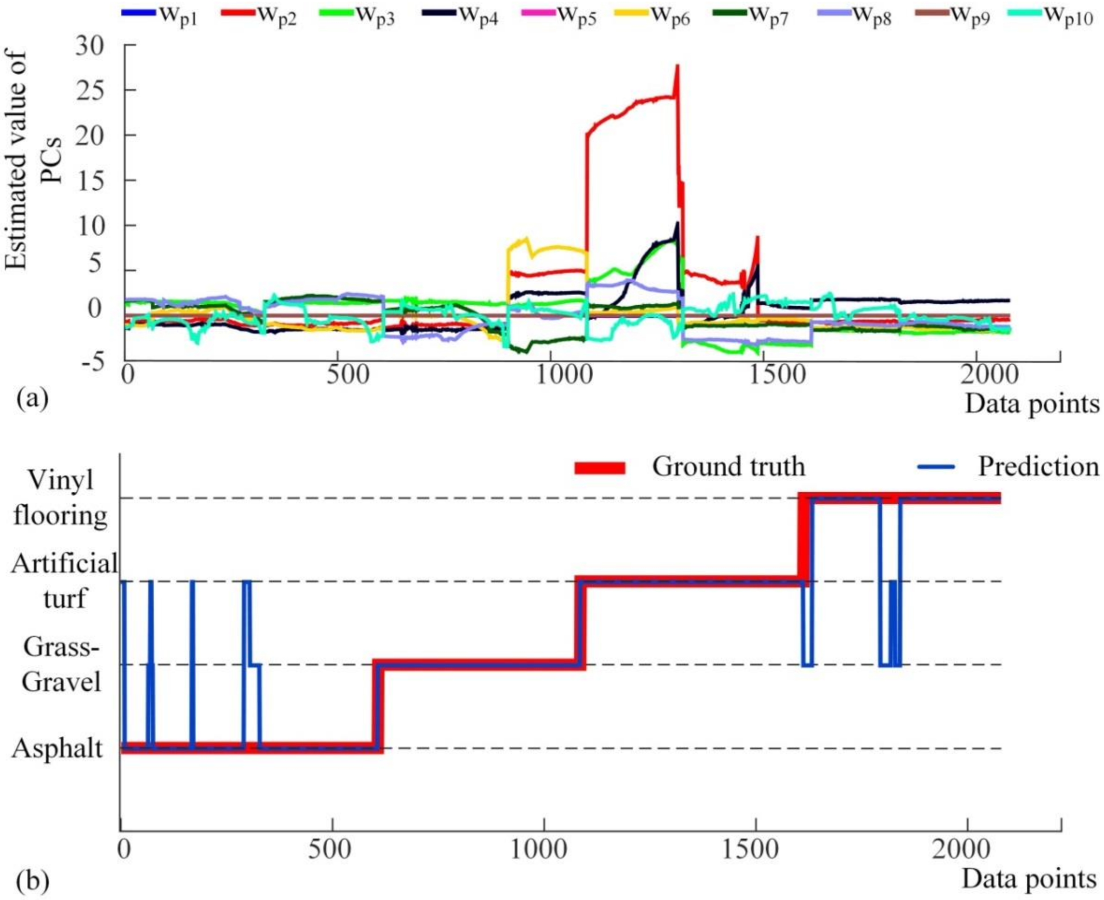


Figure 6.5: Validation of the trained SVM over simulated terrain transition data with (a) estimated values of the PCs used by the SVM and (b) the terrain estimation results from the SVM

6.4.1. Experimental validation

In order to validate the proposed terrain estimation technique, two different experimental cases were considered. In each case, the robot was made to run on a region consisting of two different types of terrain. The first case being asphalt and vinyl concrete flooring and the second vinyl concrete flooring and grass-gravel. It should be noted that although the experimental validation was performed on the same types of terrain as in the data collection, it was not performed in the same location. The experimental validation required regions where two different terrain conditions were present within the six meter square sensing limits of the POZYX positioning system. On the other hand, training data was collected in regions where the terrain conditions were uniform throughout. The two experimental setups are shown in Fig. 6.6.

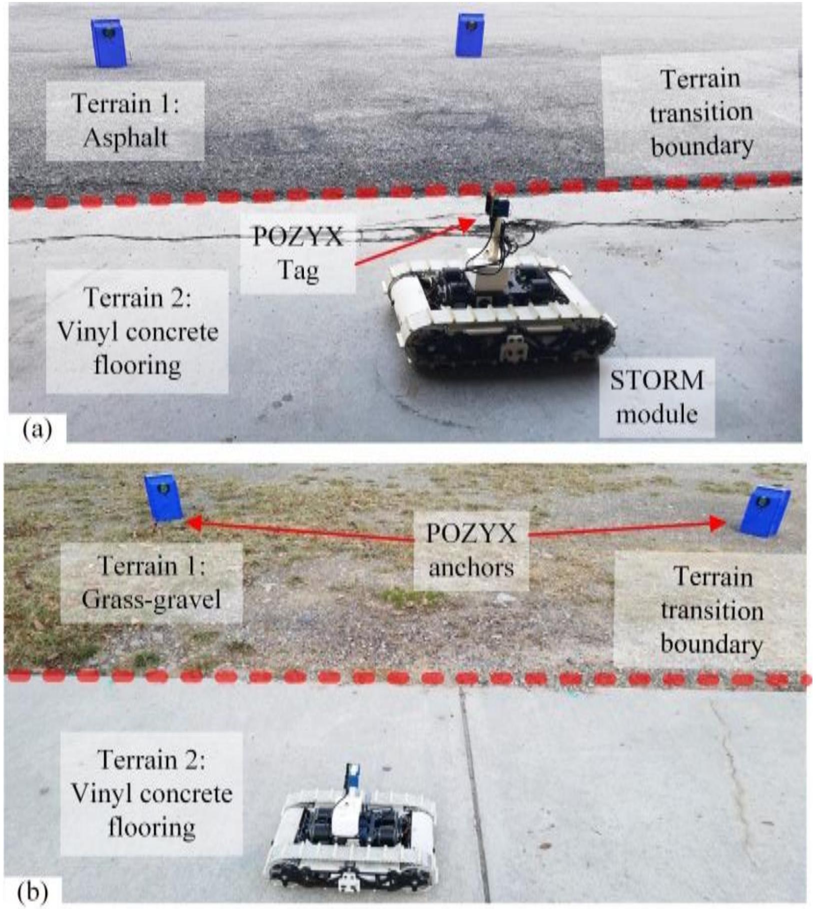

Terrain 1:
Asphalt
Terrain transition boundary
POZYX Tag
Terrain 2: Vinyl concrete flooring
STORM module
(a)
POZYX anchors
Terrain 1: Grass-gravel
Terrain transition boundary
Terrain 2: Vinyl concrete flooring
(b)

Figure 6.6: Experimental setup for the validation of the proposed terrain estimation technique on two different cases: (a) asphalt and vinyl concrete floor, (b) vinyl concrete flooring and grass-gravel

For the experimental validation described here, the robot was manually driven between the different terrains at relatively low velocity of 0.2-0.5 m/s. As in the data collection phase, the POZYX system was set up to record the pose of the robot at 100Hz sampling rate. The manual control inputs along with the state evolution information were used to estimate the weights. Based on the PCs of the estimated value of the weights the terrain classification was performed in real-time. The experimental platform, STORM, has an ODROID XU4 computer for onboard processing. Due to the limited computational capability of the onboard computer, the collected data was sent via ROS [165] to an HP laptop with a 2.6 GHz Intel processor and 8GB RAM. The real-time estimation of the weights and the terrain classification based on the PCs of the estimated weights were performed on the laptop. Figures 6.7 and 6.8 shows the PCs of the estimated value of the weights used by the SVM along with the estimation results.

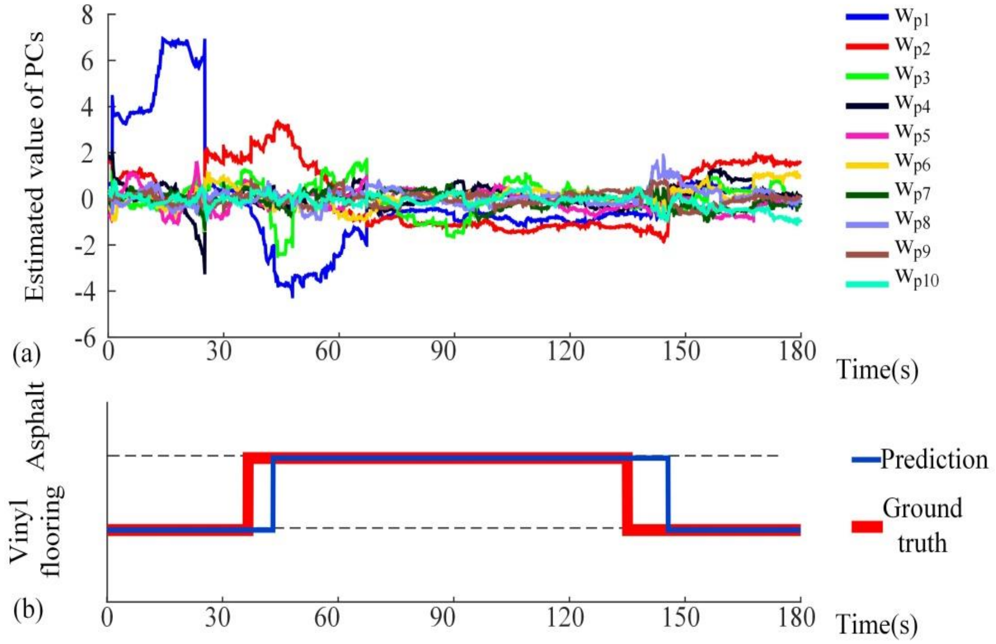


Figure 6.7: Validation of the trained SVM through experiments on asphalt and vinyl flooring with (a) PCs of the estimated values of the weights used by the SVM and (b) terrain estimation results along with ground truth

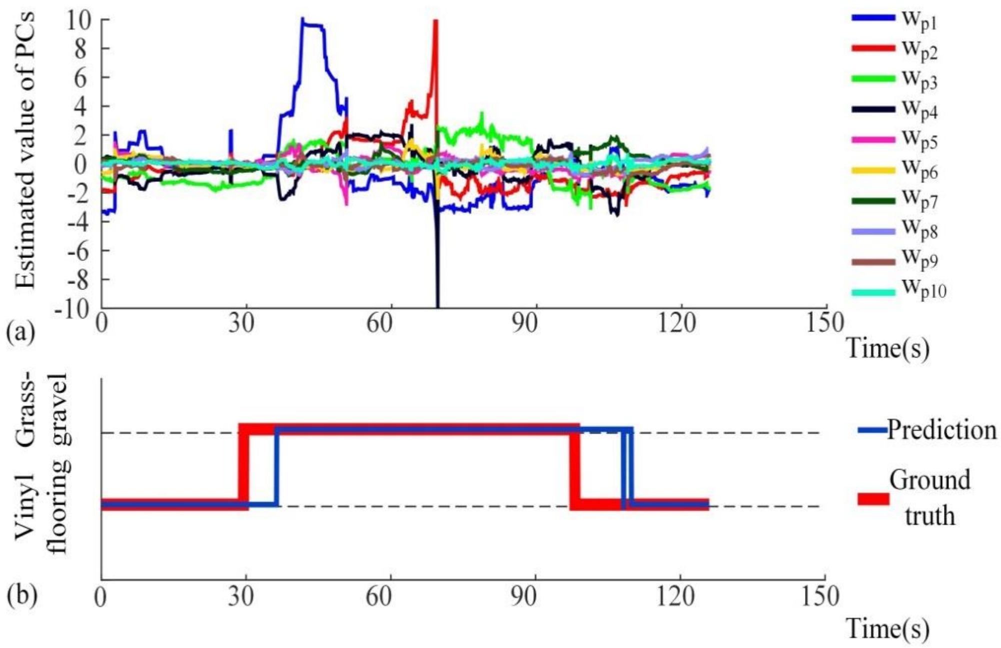


Figure 6.8: Validation of the trained SVM through experiments on vinyl flooring and grass-gravel with (a) PCs of the estimated values of the weights used by the SVM and (b) terrain estimation results along with ground truth

It should be noted that for both experiments the transition boundary is a straight line, as shown in Fig. 6.6. The POZYX absolute positioning system was setup such that the world coordinate system had one axis parallel to the terrain transition boundary. This allows for easier estimation of ground truth terrain condition of the robot.

The position coordinates of the boundary was measured and recorded prior to the experiment. For cases when the robot position was less than the recorded value, the ground truth was marked as terrain 1. Similarly, when the robot position was greater than the recorded value, the ground truth was marked as terrain 2. It should be noted that the above setup was used solely for ease of marking the ground truth and is not a limitation of the proposed terrain estimation method, as it can detect terrain transition boundaries of any shape. The ground truth, marked based on the position of the robot, is also overlaid with the estimation results. Both the ground truth and the estimations have been shown only after the first 400 samples have been taken by the estimator after the robot started moving.

For the experimental data, the SVM accuracy came out to be 90.53% on asphalt and vinyl pair, 85.33% accuracy on vinyl and grass-gravel pair. The lower accuracy of the SVM over the experimental data could be attributed to the fact that it is operating on the PCs of the weights that are being estimated in real-time using the moving window approach, as compared to the simulation case where the computed PCs were directly passed into the SVM. As mentioned before, at any given instant, the moving window estimator looks at 400 samples from the past for estimating the weights. This means that it takes at least four seconds (time taken to collect 400 data samples) after the robot has moved into the new terrain condition for the estimated weights to reflect the change accurately. This is evident in Fig. 6.7, when the SVM does not immediately recognize when the robot goes back from the asphalt into vinyl flooring condition. Similar inference can be drawn from Fig. 6.8. In both cases it takes about 8-15 seconds for the SVM to accurately predict the terrain condition after the robot has made the transition. Even after the robot has moved onto the new terrain, it has to move around and collect sufficient data before the estimated weights reflect the change to the extent that the SVM architecture can make accurate predictions. The rest of the time delay could be associated with the time taken for the value of the estimated weights to change, which is dependent on the terrain condition, the robot, and the motion commands being sent to the robot.

Prior knowledge about the region having only two kinds of terrain was leveraged by constraining the classification results to only the candidate regions for the duration of the experiment. For example in the vinyl and grass-gravel case, the SVM only chose between vinyl and grass-gravel based on whichever had the best score. It should be noted that since only the PCs of the estimated weights are passed into the SVM and not the x and y coordinates of the robot, the classifier does not learn the coordinates of the regions that correspond to each terrain condition. This ensures that the learned architecture is generalized such that it could be applied when the robot moves over previously unseen terrain conditions.

A summary of state-of-the-art techniques using different sensing modalities for performing terrain classification and traversability estimation is given in Table 2. The output from each proposed technique, their reported accuracy, and possible disadvantages are also given along with the proposed method for comparison. A quantitative comparison of the proposed technique with existing state-of-the-art terrain estimation techniques is not provided as it depends on many additional factors including the specifications of the sensor used for collecting data, computational capabilities of the experimental setup, etc.

As the proposed technique does not require any dedicated sensor, a direct quantitative comparison of accuracy is not possible. From Table 2 it can be inferred that the proposed technique performs at least as good as the state-of-the-art techniques in terms of the number of detected terrain conditions and accuracy. As mentioned previously, the major advantage of the proposed technique is that it does not require any dedicated sensing modality which in turn removes associated failure modes, costs, and computational overload. Multi sensor techniques could be used for overcoming the limitations of each individual sensing modality, but this comes with the added cost of sensors and computational overhead. On the other hand, the proposed technique requires only the state evolution of the robot along with the control commands performed by the robot, both readily available on all autonomous or semi-autonomous mobile robots.

Sensing modality	Output	Accuracy	Disadvantage
Radar [158]	Binary detection of obstacles	100% when used in combination with LIDAR	Cannot be used as a standalone sensor
Monocular camera [159]	Binary detection of traversability	98.53% at pixel level prediction	Subject to failure of vision sensors such as in presence of fog, smoke or dust
Stereo camera [160]	Binary detection of traversability	92% classification accuracy	
Measured soil and wheel parameters with classical terra-mechanics equations [161]	Terrain parameter estimation for binary detection of traversability	-	Need additional sensors for estimating normal force and torque on the wheel, wheel sinkage etc
Vibration induced through wheel-terrain interaction [163]	Online identification of terrain: sand, gravel, and clay	96% on Sand, 78% on Dirt, and 82% on gravel	May not be applicable to tracked vehicles as the vibration experienced by the tracks are influenced by weight of vehicle and properties of suspension system
Intelligent tire [171]	Classification between asphalt, concrete, soil, and grass	-	
Accelerometer attached on a tactile probe [164]	Classification between ten different indoor and outdoor surfaces	94.6% success	Limits mobility of the vehicle
Actuator and inertial sensor data from vehicle [169]	Classification between snow linoleum, ice, deep water	90%	Designed specifically for rotating leg mechanisms
Proposed technique of using state evolution for known control commands	Classification between asphalt, grass-gravel, vinyl and artificial turf	90.53% on asphalt and vinyl pair, 85.33% accuracy on vinyl and grass-gravel pair.	No dedicated sensors needed. Capable of working on any traversable terrain

Table 6.2 Comparison of the proposed approach with existing state-of-the-art terrain estimation techniques

It should be noted that while this work reports proof of concept validation of the proposed method, its performance in terms of accuracy and number of detected terrain conditions can be improved with more data.

6.5. Application to autonomous navigation

Real-time terrain estimation can be used to improve autonomous navigation, specifically in control, autonomous path planning, and in improving robot localization. Varying nature of terrain, specifically slip can result in varying trajectory tracking performance of the robot, unless the trajectory tracking controller adjusts the gains according to the nature of the terrain. Previously an extended Kalman filter based approach towards handling this issue was proposed in [157]. Assuming optimal set of gains for each terrain condition is available in the form of a look up table, real-time terrain estimation can enable the robot to choose the optimal gains for each terrain condition and thereby guarantee optimal performance. Since the terrain estimation approach only relies on the control output (V_t, ω_t) and the state evolution of the robot, the changing gain values on the trajectory tracking controller will not affect the terrain estimation process.

The proposed real-time terrain estimation techniques can be used to improve path planning for field robotic applications such as in search and rescue scenarios. As of today, the reported field trails [52], [196] have primarily relied on the intuition of the remote operator, based on visual feedback, to decide whether a robot should favor a particular terrain over another. This could be automated by monitoring the estimated values of the weights associated with the features of interest. Based on the deviation of the estimated weights from the ideal unicycle model, the relative traversability of multiple terrain conditions could be compared. Terrains that deviate more would be less traversable as compared to terrains that produce closer to ideal weights. For instance, note that in Fig. 6.1 the weights corresponding to features having control inputs (V_t and ω_t terms) are high for x_{t+1}, and y_{t+1}. But for θ_{t+1}, the weights are relatively low. Based on this, it can be inferred that it is difficult to turn the robot on asphalt. This was noticed during the experiment as the minute cracks on the asphalt provide greater resistance to turning as the wedges on the track become entrenched in them. Similar inferences can be drawn for other terrain conditions as well. This information could then be used in autonomous path planning applications by assigning a relatively high cost for less traversable terrain conditions. This allows for a more complete sense of autonomy for rescue robotic systems, with lesser reliance on human input. Even in the absence of absolute positioning sensors, such as GPS or POZYX, which is usually the case with disaster scenarios, the weights can be estimated by relying on visual odometry. Moreover, since this does not require classifying the terrain into previously learned terrain types using the SVM, the proposed techniques could be applied on previously unseen terrain conditions. It should be noted that for the proposed technique to estimates the weights, the robot must be able to traverse the terrain for a period of time. If the terrain condition causes the robot to immediately get stuck, the proposed technique would fail.

In addition to the above mentioned applications, terrain estimation can also be used in localization and mapping, during remote sensing or while surveying disaster scenarios. This could be of particular importance when the robot is moving over unstructured terrain conditions such as underground mines, as in the new DARPA challenge [174]. These regions have no GPS reception, and relying purely on visual odometry methods could lead towards accumulating drift error. On the other hand with real-time terrain recognition, the terrain itself could be used as landmarks, such that coupled with visual odometry methods precise localization of the robot can be done.

6.6. Conclusion

This chapter aimed at using state evolution of a robot moving over varying terrain conditions, under known control inputs, to perform real-time terrain estimation. The weights in the governing equation for robot motion were estimated from the state evolution data which was then transformed along the PCs and passed on to a trained SVM to generate the real-time terrain estimates. The proposed approach was validated over experimental data collected from four different terrain conditions. The trained SVM reported accuracy over 80% for simulated and actual testing.

This chapter presented a proof-of-concept validation of the idea that terrain estimation can be performed solely based on the state evolutions of the robot. Although the experimental validity of the proposed work was demonstrated using the POZYX system, it should be noted that the proposed method does not depend on any specific sensing technique. In fact, the experimental validation showed that the method works well with the limited range and noisy output of POZYX. Using a better pose estimation system such as the LOSA[117], RTK GPS or Differential GPS is expected to significantly improve the performance of the proposed technique. Detailed analysis on the effect of the various features of interest and their associated weights on the proposed terrain classification technique will be analyzed as part of future work along with detailed experimentation using various position sensing modalities over a wide variety of terrain conditions. Further testing with different kernels for the SVM or using other machine learning techniques to improve the estimation performance will also be explored in the future. In addition, as mentioned in Section 4, the robot velocity was kept low for the data collection as well as the experiments. Recording robot motion with a wider range of velocities on different terrain conditions can result in a richer dataset, allowing the trained SVM to perform better with various terrain conditions.

CHAPTER 7

IMPROVING TELE-OPERATION FOR TRACKED MOBILE ROBOTS

7.1. Introduction

This chapter proposes a novel architecture to inform and assist a remote human operator in tele-operating a tracked robotic vehicle moving over challenging terrain conditions. Majority of the robotic systems employed for search and rescue applications are designed to be tele-operated by first responders, who may not have prior experience or specialized training in operating these systems. Requiring a human operator to navigate a tracked robotic system in challenging terrain condition relying majorly on camera feed often leads to scenarios where the robot rolls over or gets stuck.

To address the above issue and thereby enable reliable tele-operation using tracked robotic systems, reliable traversability estimation is necessary. As mentioned in the previous chapters this is a particularly challenging problem in the case of tracked robotic systems. Owing to their inherent mechanical advantages these systems have the ability to go over obstacles, which makes trivial traversability estimation based solely on the presence or absence of obstacles an overly conservative approach for such systems. Even though a challenging problem, accurate traversability estimation could significantly improve the situational awareness of a tele-operator particularly in the challenging terrain conditions. Informing and assisting teleoperation based on accurate traversability estimations can effectively improve tele-operation of mobile robots in real life search and rescue scenarios, while reducing the cognitive load on the human tele-operator.

7.2. Review of tele-operation techniques

Existing literature has explored the use of model based techniques to improve tele-operation. These approaches use simple analytical models of tracked robots along with obstacle information obtained from onboard LIDAR or other proximity sensors. But the applicability of the existing techniques is limited to robotic systems moving on flat terrain conditions in presence of obstacles having a well-defined shape. To this extent, the proposed work aims to use physics engines to predict the motion of a robot forward in time, based on the current user input, while taking into account the complex terrain geometry along with the dynamic effects of robot terrain interactions. Predicted motion of the robot based on the user command and possible failure in robot motion are then conveyed to the tele-operator through visual and haptic feedback. The proposed approach is tested through human-in-the-loop simulations using a tracked robotic system in a simulated disaster scenario. Quantitative measurements regarding the number of failures encountered by the system, and time taken to successfully explore the terrain along with qualitative measurements on the ease of operation brought about by the proposed architecture as reported by the users are analyzed to validate improvement in performance as compared to state-of-the-art techniques.

Mobile robots are now widely being used in applications that are not suitable for human operators, such as search and rescue and surveillance applications in natural or manmade disaster scenarios [52], [196], [197]. Almost all field deployments of search and rescue robotic systems in the recent years involved human tele-operators either fully controlling the robotic system (rate control) or giving high level commands (waypoint control) while actively monitoring the system behavior [16], [20], [198], [199]. The major reasons limiting the use of fully autonomous operation are:

(1) Need for further development in the domain of artificial intelligence and machine learning before robots are capable of making safe and intelligent decisions on their own.

(2) Lack of trust on the part of human first responders regarding the capabilities of robotic systems in scenarios where minor mistakes could put the expensive robotic system and human lives in danger.

(3) Intuition of an experienced human operator is particularly valuable especially in scenarios where time is limited, such as while looking for casualties in disaster zones.

It can be inferred that human guided or semi-autonomous operation based on high level commands issued by a human tele-operator will be used for the near future until full autonomy can be achieved. As mentioned in the introduction, teleoperation in unstructured environments is often subject to limited bandwidth, degraded signals, and communication delays all leading to limited environmental awareness on the part of the tele operator, which in turn leads to unstable or oscillatory robot motion, resulting in unintended collisions, systems getting stuck, and roll/tip over.

The major challenges involved in assisting tele-operation through traversability estimation can be described as follows:

- Sensing the terrain condition: Majority of the existing tele-operated systems use monocular video feeds to inform the remote operator. In addition to limited field of view and lack of depth information, these are often low resolution due to the limited communication bandwidth and thereby results in a poor understanding of the environment on the part of the remote operator.

Addressing the lack of environmental awareness on the part of the remote operator, often referred to as telepresence, is a major factor towards improving tele-operation.

- Estimating user input for a forward horizon: For tele operated exploration of unstructured environments, the control inputs from the human operator cannot be predicted beyond a limited time horizon. The human operator and thereby the tele operated robotic system can change trajectory as the environment is being explored, or even based on the traversability predictions obtained from the system. Instead of trying to predict the user input for extended periods of time, prior work by Chipalkatty et al., [200] has shown that a zero-order hold of the human input for a short duration of time is a more effective approach. Zero-order hold refers to holding the current user input constant for the duration of prediction.
- Predicting the motion of the robotic system: Existing model predictive control (MPC) based techniques to improve reliability of tele-operation has mainly relied on analytical models to predict robot motion. Majority of these techniques were tested in simple scenarios involving robot motion over flat terrain with obstacles of defined shape and size. This limitation comes from the need for explicit mathematical models governing the motion of the tracked vehicle in forward time for any candidate control action.

  In most real life applications, modelling robot motion at high fidelity would result in non-linear formulations with non-convex safe regions, difficult to be solved using MP, even with the advanced computational power available

[201]. This in turn limits the applicability of these methods in the unstructured environments encountered in real-life applications. A good review of some of these techniques are given in [201].

- Conveying information to the tele-operator: Existing studies have explored the use of conveying traversability estimations and raw sensor information to the user to improve tele-presence, resulting in mixed results. Even though they improve the environmental understanding, it comes at the cost of additional cognitive workload. A notable approach has been to convert range information into force feedback on haptic controllers to augment the user's experience [202], [203]. This approach kept a minimal cognitive load while significantly improving the performance of the tele-operator.

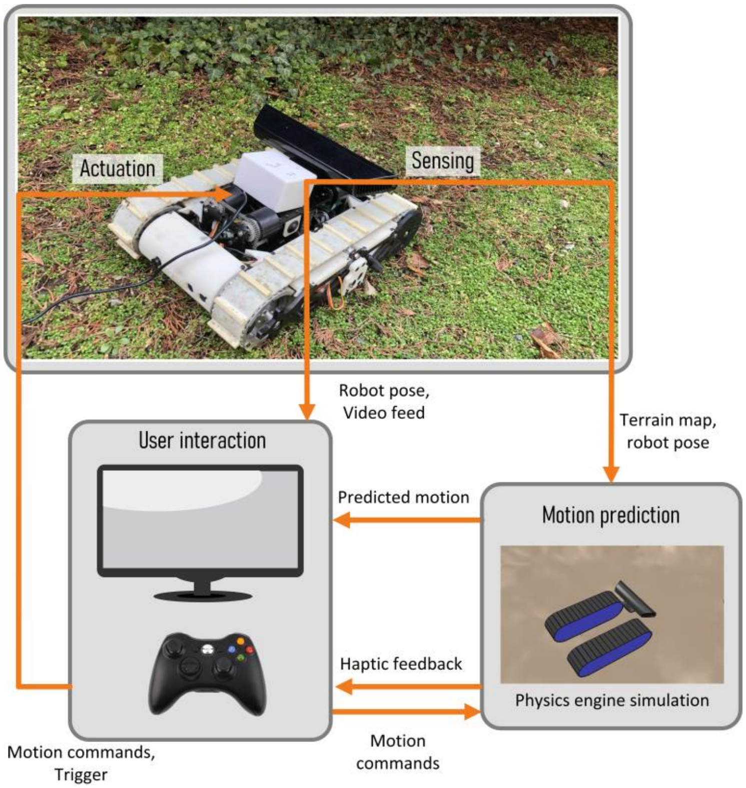


Figure 7.1: Overall layout of the proposed approach

7.3. Proposed approach

The proposed approach aims to address the above mentioned challenges associated with existing state-of-the-art tele-operation assist systems as described below:

- Sensing the terrain condition: A 3D map of the environment is obtained in real-time and used to perform traversability estimations as described below. This could be obtained using any 3D LIDAR or Stereo camera system or even an Xbox Kinect. Similar to prior implementation, the tele-operator is provided with a monocular feed from a camera mounted on the robot. In addition, a map of the terrain is also provided to the user showing the boundaries of the region with the robot positon and orientation since the start of tele-operation.
- Estimating user input for a forward horizon: Unicycle model was used to tele-operate the robotic system, where the user provides linear and angular velocity commands to the robot, (v, ω). The user commands are then converted to left track velocity and right track velocity, (v_l, v_r) before being applied to the actual robot. The user input is taken using an Xbox joystick module, with separate axis being used for linear and angular velocity commands, as shown in Fig. 7.2. Using the joystick the tele-operator can provide linear velocity commands varying between (-1m/s to 1m/s) and angular velocity commands varying between (-1rad/s to 1rad/s).

  In order to estimate traversability for a forward horizon, the user inputs should be estimated for the forward horizon, based on the current user input. Inspired by the success of prior work [200], [201], a zero-order hold is used for predicting the user input for the forward horizon. This essentially means that each time the forward prediction of the robot motion needs to be done, the current human inputs are held constant for the duration of the prediction horizon. As mentioned in the previous studies, for low time delay systems with short prediction horizons this is the most effective approach.

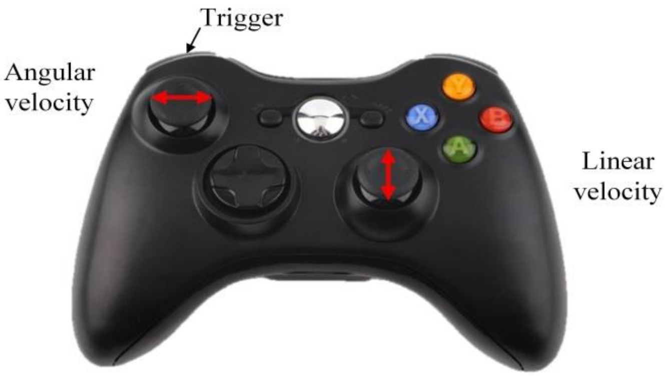


Figure 7.2: Controller used to obtain user commands

- Predicting the motion of the robotic system: In order to estimate traversability of the robotic system forward in time based on the predicted user inputs we use physics engines. Physics engines are capable of estimating motion of the robotic system based on the dynamic model of the robot along with the terrain topography map obtained from sensor information. The possible failure modes that could be detected using a physics simulation include failure due to slip, roll and pitch angles of the robot exceeding safe limits, and collisions with the environment. The physics engine used in this study was Bullet [106]. The physics engine simulation of the robot motion on a terrain map is shown in Fig. 7.3

Recent advancements in computational capabilities and physics engine libraries have enabled faster than real-time dynamic simulations. For the purpose of this work, Bullet physics engine running in headless mode was used to reduce the computational load. Based on the prediction results, if the user approves of the motion of the robotic system, he/she could initiate the motion.

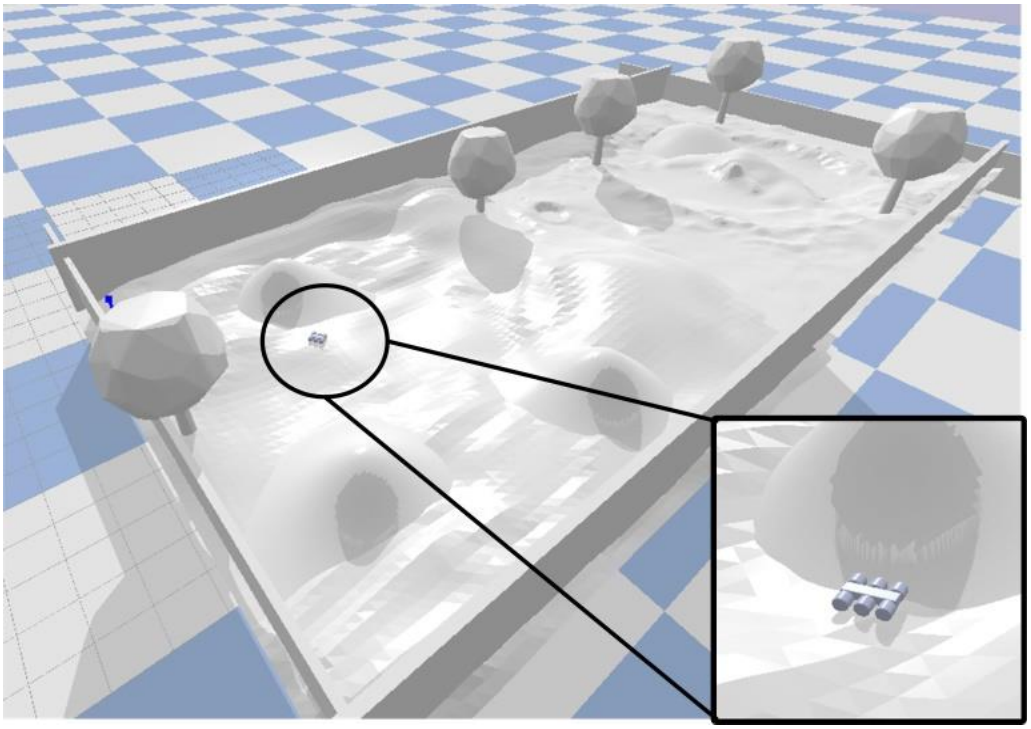


Figure 7.3: Estimating traversability using physics simulation, the simulated robot model is shown in inset

- <u>Conveying information to the tele-operator</u>: The results of the traversability estimations are conveyed to the tele-operator through visual cues and haptic feedback. The predicted path for the robot and final position at the end of the prediction horizon is also displayed on the topography map. Cases where the user input could lead to collisions or failure of mission is indicated through visual cues as well as haptic feedback. The visual cues denote the path leading to failure and the location of failure. Based on the real-time visual and haptic feedback the tele-operator could modify the control inputs and thereby prevent possible failure.

It should be noted that while the proposed architecture informs the user about possible mission failures, the control commands to the robot are relayed by the human operator. In other words, the proposed architecture assists and informs the tele-operator but the navigation decisions themselves are made by the human operator based on his/her intuition. Based on the author’s knowledge, this is the first work towards exploring the use of physics engines in improving teleoperation in unstructured environments.

It should be noted that this work does not explicitly handle the effect of time delay. The simulated and experimental validation presented in this work involved very small delays in the order of milliseconds. A detailed analysis of the effects of larger or smaller time delays on the performance of the proposed methods is beyond the scope of this work.

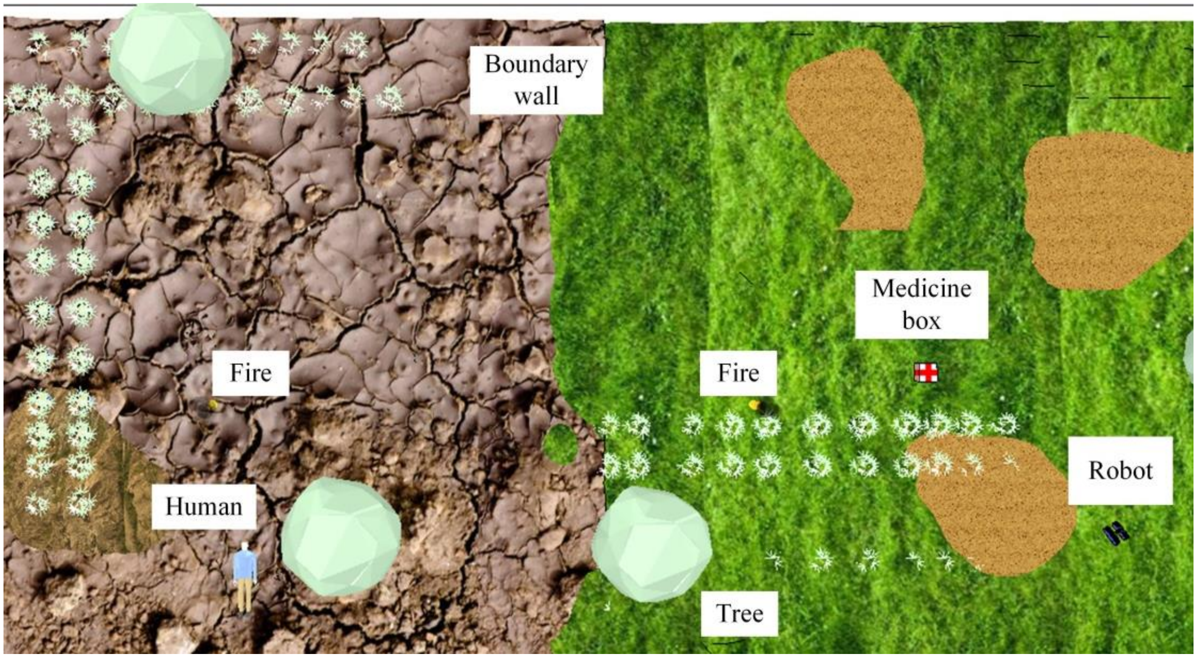


Figure 7.4: Terrain map being used for the validation

7.4. Experimental validation

In order to experimentally validate the improvements in tele-operation based on the proposed architecture, a simple and brief user simulation was performed. A simulated disaster scenario, as shown in Fig. 7.4, was created with objects of interest (OOIs) including a human dummy, fire, and medicine box placed at random locations. Six users were required to tele-operate a tracked robotic system inside the simulated environment to locate all three OOIs in the least possible amount of time. The OOIs used for this simulation are shown in Fig.7.5.

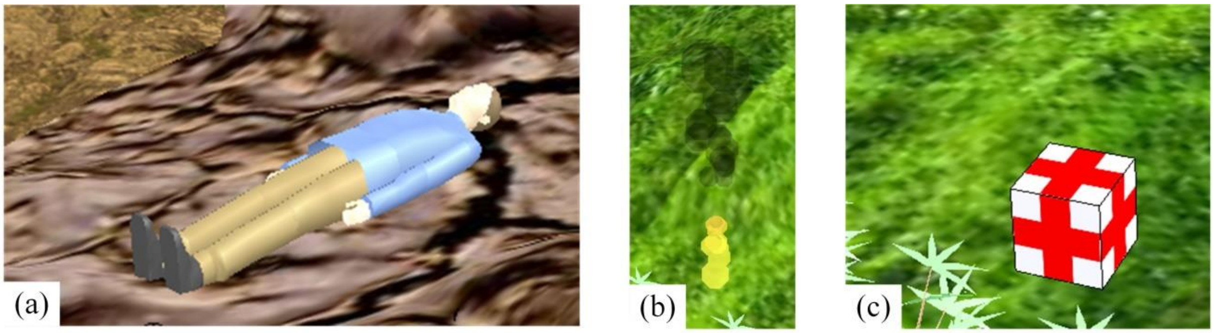


Figure 7.5: The objects of interest used in the simulated validation (a) Human dummy (b) Fire (c) Medicine box

The user/operator was required to use the Xbox Joystick to operate the robot. Each operator performed two different trails, one with the traditional tele-operation architecture, using only video feed from the robot, and a map showing position of the robot. For the second simulation the proposed architecture for assisting and informing the tele-operator was provided. This involved showing the estimated future motion of the robot on the map. In cases where there was no collision, the predicted path of the robot was shown in green. For the cases where the predicted motion resulted in mission failure, the predicted path was shown in red, ending on the location of predicted failure. In addition, a haptic feedback was given to the operator using the joystick to indicate imminent failure.

The operators were trained to operate the robotic system in the simulated environment prior to the start of the simulations. For both simulations the time taken by the operator to find all three objects of interest was noted. The number of times the robot collided with the environment and the number of times the system failed completely, due to getting stuck or flipping over was also recorded. In cases where the robot failed, it was reset at the start position immediately and the operator was asked to continue with the mission, with the timer running continuously. In addition the operator was asked to rate the improvement in tele-operation by the haptic feedback and the predictive display on a scale of 0 to 5 with 0 being no improvement and 5 being significant improvement. The operators were also asked about the relative advantages and disadvantages of the predictive display and the haptic feedback.

7.5. Results

The quantitative results of the simulations are shown in Table 7.1. The mean values for the time taken to find all OOI’s, number of collisions with the environment, number of times the system completely failed and the robot had to start from the initial location again is given. The improvement score assigned by the operators to the proposed tele-operation architecture is also given. The standard deviations for all the above values are shown in parenthesis.

On a qualitative note, most of the operators commented that the proposed architecture greatly improved the tele-operation task. None of the operators faced even a single collision or failure while using the proposed architecture as the system always informed the operator beforehand about imminent collision/failure. With the traditional architecture operators faced many collision and failures. One of the operators noted that he or she was ready to give up the mission after three failures and subsequent restarts.

	Traditional tele-operation architecture	Proposed tele-operation architecture
Time taken to find the OOI	2 min 48sec (58 sec)	1 min 21sec (43 sec)
Number of collision with environment	2.5 (0.5)	0(0)
Number of time system failed	1.33(1.03)	0(0)
Improvement score for proposed architecture (0-5)	-	4.75(0.22)

Table 7.1: Quantitative results from simulation

Majority of the operators stated that the haptic feedback was more useful as compared to the predictive display. Most of the operators focused on the camera feed from the robot and did not pay much attention to the predictive display. As such the haptic feedback was better preferred due to the reduced cognitive load on the operator. The results show that the proposed architecture significantly improves the tele-operation of robotic systems in challenging environments.

7.6. Future work

In addition to physics engine based traversability estimation, the other techniques proposed in the previous chapters such as active disturbance rejection control to handle slip and terrain estimation could also be used to improve tele-operation. An overall architecture employing above techniques to inform the user about the conditions in tele-operated terrain is shown in Fig. 7.6. In addition to using physics engine simulation to perform traversability estimation based on predicted user input, the proposed architecture

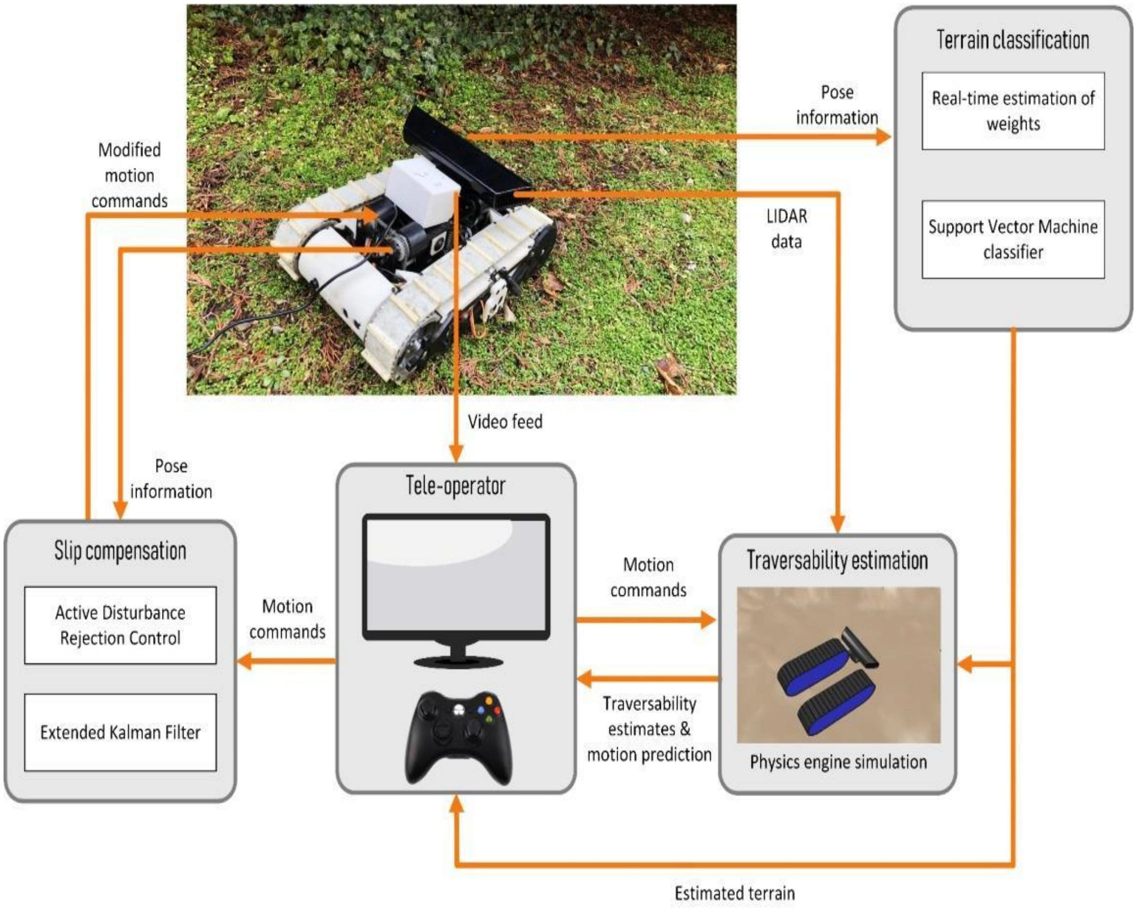


Figure 7.6: Overall architecture to assist and inform teleoperation of tracked robotic modules

also employs previously developed ADRC techniques to handle the effects of slip during teleoperation.

The motion commands generated by the user are passed through the ADRC to allow for counteracting the effect of slip, thereby providing similar performance on varying terrain conditions. In addition, terrain classification techniques are used to inform the user about the terrain type. This further improves the environmental awareness of the tele-operator, thereby improving his/her telepresence.

CHAPTER 8

CONCLUSION AND FUTURE WORK

This chapter concludes the dissertation with a summary of the current work as well as providing directions for future research.

8.1. Summary

The major contributions of this dissertation are two-fold: (1) a conceptual design for novel rescue robot system that could effectively perform casualty extraction and evacuation from disaster scenarios, and (2) novel techniques to improve tracked robot motion in challenging terrain conditions.

A detailed survey of existing rescue robot systems was presented and the specific shortcomings of these systems which limit them from deployment in field conditions were identified. The proposed Semi-Autonomous Victim Extraction Robot (SAVER) was conceptually designed to address some of the identified shortcomings and thereby help realize safe and effective casualty rescue in disaster scenarios.

This dissertation made novel contributions in planning, control and terrain estimation for improving tracked robot motion. An overall architecture to employ the above developed techniques to aid and improve tele-operation of tracked robots using the above techniques was also presented.

8.2. Directions for future research

Even though all of the different techniques proposed in this dissertation have been validated through simulation and experiments, they can be further improved. This section discusses some of the shortcomings of the proposed techniques and provides directions for further research.

8.2.1. Improvements in physics based planning

Even though the effectiveness of the proposed navigational architecture along with physics based traversability estimation was validated through simulations and experiments, a detailed stability analysis remains to be done. In addition, calculating the uncertainty associated with the traversability estimations also needs to be explored. This would extend the applicability of the proposed technique towards probabilistic planning applications. While the performance of the proposed planner was shown to be better in comparison with a naive kinematic planner, detailed performance comparisons with state-of-the-art planning techniques is future work.

In addition to traversability estimation robotic simulator can also be used to make intelligent decisions regarding the behavior of robotic systems in other domains as well. Even though this work explains the application of the novel path planning method for a tracked differential drive robot, the overall approach used here can be applied to autonomous navigation of any robotic system. For instance, physics engines are capable of modeling the motion of quadcopters, including the execution of extreme maneuvers. For path planning applications involving quadcopters, a high-level planner will have to evaluate the feasibility of such maneuvers under a given environment, such as while flying inside a collapsed building structure for search and rescue operations. The proposed approach using physics engines can provide better results in such cases. The major difference in applying the proposed framework to other platforms, including quadcopters and AUVs will be in designing stable closed-loop controllers that can work at the lowest level for these highly nonlinear systems. The overall approach of using a high-level planner along with a physics based simulation, including the robot model with the low-level controller, can remain the same.

In addition to the use of physics engine for path planning applications, its use for generating training data sets for NN based real-time traversability estimation was also explored in this dissertation. Different NN architectures such as Long Short Term Memory (LSTM) [204] can further improve the performance of the learning based traversability estimation. Color images of the environment as obtained from cameras could be easily integrated into the proposed NN architecture to improve the estimation accuracy. Most robotic simulators including V-REP and Gazebo have photorealistic rendering modes with accurate camera models that could be used to generate the training data in such cases. Similarly, slip experienced by the vehicle is another major factor that could affect the traversability of a given region. Including slip and RGB images to the NN architecture and analyzing their effect on traversability predictions will be explored as part of future work.

8.2.2. Improvements in slip rejection control

The proposed ADRC implementation used a random walk model to estimate the augmented parameters and thereby improve path tracking under varying terrain conditions. As mentioned in Section 5.3.2, the proposed approach assumes that the augmented parameters are piece wise constant. Based on the experimental results it is clear that the above assumption holds true for the proposed application under a sampling frequency of 50Hz. While the experimental validation is performed as a part of this dissertation, a detailed mathematical analysis into the nature of the augmented parameters and their relationship to physical attributes such as terrain friction coefficient, or slip remains to be analyzed.

Even though the working of the proposed slip rejection controller was validated with a simple PD low-level controller, it could work effectively with other low-level path tracking controllers. For instance pure-pursuit or Gaussian kernel controllers could also be used for path tracking at the lowest level. As the name indicates a Gaussian kernel controller uses of gradients of moving Gaussian kernels for path tracking in mobile robots [161]. The controller was designed to address shortcomings in existing pure pursuit implementations in order to produce smooth motions while ensuring accurate tracking of the provided reference path.

8.2.3. Improvements in terrain estimation

While the terrain estimation technique proposed in this dissertation works well enough for classification of the four terrain conditions that it was tested upon, further research into using additional features of the data including the variance of the collected parameters, skewness, kurtosis etc., could provide more information about the terrain. Adding these additional features is expected to improve the accuracy of the estimation method based on a preliminary analysis of the collected data. It should be noted that the as per the above analysis, the estimated value of the weights can depend on the features of the mobility platform, as much as they depend on the terrain over which the robot is moving. This in turn opens up possibilities for using the above method as a diagnostic tool in order to gauge the performance of the robot. Assuming that the terrain conditions remain the same, changes in the estimated value of the weights can be used to determine the presence of faults in the robot. For all of the tests conducted as part of this work the performance of the mobility platform was assumed to be same, but future analysis will focus on using the proposed method for fault detection in the robot.

Even though this work explored the use of terrain estimation methods as applied to a skid steer robot, ongoing work aims at applications on commercial vehicles such as cars and trucks. The proposed technique can be used to estimate driving conditions such as presence of snow or ice on the road and also towards monitoring the conditions of the vehicle in real-time. The presence of suspension systems in these vehicles can adversely affect the performance of terrain estimation methods as it reduces the effect of the terrain on the motion of the vehicle. This may require attaching sensors directly to the wheels of the vehicle, before the suspension system, as explored in [205].

8.2.4. Improvements in tele-operation

The proposed work explored the use of physics engine based traversability estimation in improving the ease and reliability of mobile robot tele-operation in challenging terrain conditions. While the proposed work was evaluated in a simulated scenario, detailed evaluation with mobile robot tele-operation in real-world scenario will be the next step. This would require use of 3D LIDAR or stereo camera system to generate the terrain map which would then be used for traversability estimation inside the physics engine. In addition, the overall architecture as given in Fig. 7.6, where ADRC is used to improve trajectory tracking and a trained SVM is used for terrain estimation also remains to be tested in real-world conditions. Updating the parameters inside the physics engine based on the estimated nature of the terrain, such as soil type and compactness, presence or absence of vegetation etc., would further improve the accuracy of the traversability estimation. Simulating soft terrain locomotion inside physics engines is another major direction to be explored.

In addition to the improvement brought out by the traversability estimation, a high level planner such as kino-dynamic RRT* could be used to further assist the tele-operator. Based on the terrain information being used for traversability estimation, the planner could come up with alternate collision free paths, deviating minimally from the predicted motion of the robot. The predicted path could be reported back to the tele-operator who could then proceed with the mission without failure.

8.2.5. Realizing safe and efficient casualty extraction in disaster scenarios

The use of robotic systems to augment the efforts of search, rescue, and medical response teams has the potential to improve the efficiency of these humanitarian efforts and save lives. Many innovative systems have demonstrated effective solutions to the problems presented by rescue and medical response. With further advancement, in the future the search, extraction, evacuation, and treatment of disaster victims and military casualties may someday be carried out by a single cooperative robotic team. The development of SAVER forms the basis for such further advancements in the robotics field, advances that can lead to robots that save lives and better prepare humanity to respond to catastrophic and disastrous situations.

REFERENCES

[1] N. W. Dickey, “Combat Trauma Lessons Learned from Military Operations of 2001-2013,” Defense Health Board, 2015.

[2] International Federation of Red Cross and Red Crescent Societies (IFRC), *World disasters report 2009: Focus on Early Warning, Early Action*. Geneva.

[3] M. M. Dinh, K. Bein, S. Roncal, C. M. Byrne, J. Petchell, and J. Brennan, “Redefining the golden hour for severe head injury in an urban setting: The effect of prehospital arrival times on patient outcomes,” *Injury*, vol. 44, no. 5, pp. 606–610, 2013.

[4] A. M. K. Harmsen, G. F. Giannakopoulos, P. R. Moerbeek, E. P. Jansma, H. J. Bonjer, and F. W. Bloemers, “The influence of prehospital time on trauma patients outcome: A systematic review,” *Injury*, vol. 46, no. 4, pp. 602–609, 2015.

[5] E. B. Lerner and R. M. Moscati, “The Golden Hour: Scientific Fact or Medical ‘Urban Legend’?,” *Acad. Emerg. Med.*, vol. 8, no. 7, pp. 758–760, 2001.

[6] C. D. Newgard *et al.*, “Emergency Medical Services Intervals and Survival in Trauma: Assessment of the ‘Golden Hour’ in a North American Prospective Cohort,” *Ann. Emerg. Med.*, vol. 55, no. 3, pp. 235–246, 2010.

[7] R. S. Kotwal *et al.*, “The effect of a golden hour policy on the morbidity and mortality of combat casualties,” *JAMA Surg.*, vol. 151, no. 1, pp. 15–24, 2016.

[8] B. J. Eastridge *et al.*, “Died of Wounds on the Battlefield: Causation and Implications for Improving Combat Casualty Care,” *J. Trauma Inj. Infect. Crit. Care*, vol. 71, pp. S4–S8, Jul. 2011.

[9] U.S. Army Medical Research and Material Command, “Unmanned Systems

Teaming for Semi-Autonomous Casualty Extraction," *SBIR-STTR*, 2017. [Online]. Available: https://www.sbir.gov/sbirsearch/detail/1319095. [Accessed: 11-Feb-2017].

[10] P. L. Chapman *et al.*, "Training, Deployment Preparation, and Combat Experiences of Deployed Health Care Personnel: Key Findings From Deployed U.S. Army Combat Medics Assigned to Line Units," *Mil. Med.*, vol. 177, no. 3, pp. 270–277, Mar. 2012.

[11] Department of Homeland Security, "First Responder Guide for Improving Survivability in Improvised Explosive Device and/or Active Shooter Incidents," 2015. [Online]. Available: https://www.dhs.gov/sites/default/files/publications/First Responder Guidance June 2015 FINAL 2.pdf. [Accessed: 01-Dec-2018].

[12] R. G. Snyder, "Robots assist in search and rescue efforts at WTC," *IEEE Robot. Autom. Mag.*, vol. 8, pp. 26–28, 2001.

[13] R. R. Murphy, *Disaster Robotics*. MIT Press, 2014.

[14] R. R. Murphy *et al.*, "Search and Rescue Robotics," in *Springer Handbook of Robotics*, Springer, 2008, pp. 1151–1173.

[15] A. Davids, "Urban Search and Rescue Robots: From Tragedy to Technology," *IEEE Intelligent Systems*, vol. 17, no. 2. pp. 81–83, 2002.

[16] R. R. Murphy, "A decade of rescue robots," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 5448–5449.

[17] K. Nagatani *et al.*, "Emergency response to the nuclear accident at the Fukushima Daiichi Nuclear Power Plants using mobile rescue robots," *J. F. Robot.*, vol. 30,

no. 1, pp. 44–63, Jan. 2013.

[18] K. Nagatani *et al.*, “Redesign of rescue mobile robot Quince,” in *IEEE International Symposium onSafety, Security, and Rescue Robotics (SSRR)*, 2011, pp. 13–18.

[19] R. Murphy, J. Casper, J. Hyams, M. Micire, and B. Minten, “Mobility and sensing demands in USAR,” in *IECON Proceedings (Industrial Electronics Conference)*, 2000, vol. 1, pp. 138–142.

[20] J. Casper and R. R. Murphy, “Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center,” *IEEE Trans. Syst. Man, Cybern. Part B*, vol. 33, no. 3, pp. 367–385, Jun. 2003.

[21] T. Atwood and J. Klein, “VECNA’s Battlefild Extraction-Assist Robot BEAR,” *Robot Magazine*, 2007. [Online]. Available: https://web.archive.org/web/20101023081400/http://www.vecna.com/robotics/solutions/bear/index.shtml. [Accessed: 01-Dec-2018].

[22] R. Watts, P. Rowe, and G. Gilbert, “TATRC and TARDEC collaborative robots program.” 2004.

[23] B. M. Yamauchi, “PackBot: a versatile platform for military robotics,” in *Unmanned Ground Vehicle Technology VI*, 2004, vol. 5422, pp. 228–237.

[24] D. Theobald, “Mobile Extraction-Assist Robot,” US Patent 7,719,222 B2, 2010.

[25] J. Hu and Y.-J. Lim, “Robotic First Responder System and Method,” US Patent US20140150806A1, 2014.

[26] P. Wells and D. Deguire, “TALON: a universal unmanned ground vehicle platform, enabling the mission to be the focus,” in *Unmanned Ground Vehicle*

Technology VII, 2005, vol. 5804.

[27] G. Gilbert, T. Turner, and R. Marchessault, “Army medical robotics research.” 2007.

[28] Y. Iwano, K. Osuka, and H. Amano, “Development of stretcher component robots for rescue activity,” in *IEEE Conference on Robotics, Automation and Mechatronics, 2004.*, 2005, vol. 2, pp. 915–920.

[29] N. Fisher and G. R. Gilbert, “Unmanned Systems in Support of Future Medical Operations in Dense Urban Environments,” *Small Wars J.*, 2016.

[30] K. Johnson *et al.*, “Clinical evaluation of the Life Support for Trauma and Transport (LSTAT^{TM}) platform,” *Crit. Care*, vol. 6, no. 5, pp. 439–446, 2002.

[31] K. Thompson, “Squad Multipurpose Equipment Transport (SMET),” 2015. [Online]. Available: https://ndiastorage.blob.core.usgovcloudapi.net/ndia/2016/GRCCE/Thompson.pdf. [Accessed: 01-Dec-2018].

[32] K. Massey, “Squad Mission Equipment Transport (SMET): Lessons Learned for Industry,” 2016. [Online]. Available: https://ndiastorage.blob.core.usgovcloudapi.net/ndia/2016/GRCCE/Massey.pdf. [Accessed: 01-Dec-2018].

[33] F. E. Schneider, D. Wildermuth, B. Brüggemann, and T. Röhling, “European land robot trial (elrob) towards a realistic benchmark for outdoor robotics,” in *1st international Conference on Robotics in Education (RiE2010)*, 2010, pp. 65–70.

[34] R. Sheh, S. Schwertfeger, and A. Visser, “16 Years of RoboCup Rescue,” *KI - Künstliche Intelligenz*, vol. 30, no. 3–4, pp. 267–277, Oct. 2016.

[35] H. Kitano, “RoboCup Rescue: A grand challenge for multi-agent systems,” in *4th International Conference on MultiAgent Systems, ICMAS 2000*, 2000, vol. 22, no. 1, pp. 5–12.

[36] S. Tadokoro *et al.*, “The RoboCup-Rescue project: a robotic approach to the disaster mitigation problem,” in *IEEE International Conference on Robotics and Automation (2000 ICRA Millennium Conference.)*, 2000, vol. 4, pp. 4089–4094.

[37] A. Jacoff, E. Messina, and J. Evans, “A Standard Test Course for Urban Search and Rescue Robots,” in *Performance Metrics for Intelligent Systems Workshop (NIST SP 970)*, 2000, pp. 253–259.

[38] “DARPA Grand Challenge, Official Website.” [Online]. Available: http://archive.darpa.mil/grandchallenge/. [Accessed: 02-Dec-2018].

[39] G. C. Haynes *et al.*, “Developing a robust disaster response robot: CHIMP and the robotics challenge,” in *Springer Tracts in Advanced Robotics*, vol. 121, no. 2, 2018, pp. 103–144.

[40] S. Kim *et al.*, “Approach of Team SNU to the DARPA Robotics Challenge finals,” in *IEEE-RAS International Conference on Humanoid Robots*, 2015, vol. 2015-Decem, pp. 777–784.

[41] S. Kohlbrecher *et al.*, “Human-robot teaming for rescue missions: Team vigir’s approach to the 2013 darpa robotics challenge trials,” *J. F. Robot.*, vol. 32, no. 3, pp. 352–377, May 2015.

[42] S. J. Yi *et al.*, “Team THOR’s Entry in the DARPA Robotics Challenge Trials 2013,” *J. F. Robot.*, vol. 32, no. 3, pp. 315–335, May 2015.

[43] A. Matos *et al.*, “Multiple robot operations for maritime search and rescue in

euRathlon 2015 competition," in *OCEANS 2016 - Shanghai*, 2016, pp. 1–7.

[44] A. F. T. Winfield *et al.*, "euRathlon 2015: A multi-domain multi-robot grand challenge for search and rescue robots," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9716, no. September 2014, pp. 351–363.

[45] A. Williams, B. Sebastian, and P. Ben-Tzvi, "Review and Analysis of Search, Extraction, Evacuation, and Medical Field Treatment Robots," *Journal of Intelligent & Robotic Systems*. Springer Netherlands, Available Online, DOI: 10.1007/s10846-019-00991-6, Feb-2019.

[46] P. Ben-Tzvi, A. Williams, B. Sebastian, A. Kumar, and W. Saab, "Semi-Autonomous Victim Extraction Robot (SAVER)," U.S. Provisional Patent Application No. 62/836,915 April 22, 2019.

[47] B. Sebastian and P. Ben-Tzvi, "Physics Based Path Planning for Autonomous Tracked Vehicle in Challenging Terrain," *J. Intell. Robot. Syst.*, vol. 95, no. 2, pp. 511–526, Aug. 2019.

[48] B. Sebastian, A. Williams, and P. Ben-Tzvi, "Control of a Head Stabilization System for Use in Robotic Disaster Response," *Proceedings of the 2017 ASME International Mechanical Engineering Congress and Exposition (IMECE 2017)*. ASME, Tampa, Florida, 2017.

[49] A. Williams, B. Sebastian, and P. Ben-Tzvi, "A Robotic Head Stabilization Device for Medical Transport," *Robotics*, vol. 8, no. 1, p. 23, Mar. 2019.

[50] A. Kumar, H. Ren, and P. Ben-Tzvi, "Obstacle Identification for Vision Assisted Control Architecture of a Hybrid Mechanism Mobile Robot," *Proceedings of the*

2017 ASME Dynamic Systems and Control Conference (DSCC 2017). Tysons Corner, Virginia, USA. October 11–13, 2017.

[51] H. Ren, A. Kumar, X. Wang, and P. Ben-Tzvi, “Parallel Deep Learning Ensembles for Human Pose Estimation,” *Proceedings of the ASME 2018 Dynamic Systems and Control Conference (DSCC 2018)*. Atlanta, Georgia, USA. September 30–October 3, 2018.

[52] R. Murphy, J. Kravitz, S. Stover, and R. Shoureshi, “Mobile robots in mine rescue and recovery,” *IEEE Robot. Autom. Mag.*, vol. 16(2), no. 2, pp. 91–103, Jun. 2009.

[53] M. K. Beebe and G. R. Gilbert, “Robotics and Unmanned Systems – ‘Game Changers’ for Combat Medical Missions,” *NATO RTO-HFM 182 Symp. Adv. Technol. New Proced. Med. F. Oper.*, 2010.

[54] S. Odedra, S. Prior, and M. Karamanoglu, “Investigating the mobility of unmanned ground vehicles,” in *Proceedings of the International Conference on Manufacturing and Engineering Systems, Huwei, Yunlin, Taiwan*, 2009.

[55] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli, “A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles,” *IEEE Trans. Intell. Veh.*, vol. 1, no. 1, pp. 33–55, Mar. 2016.

[56] S. M. LaValle, *Planning Algorithms*. Cambridge: Cambridge University Press, 2006.

[57] F. Fahimi and SpringerLink (Online service), “Autonomous robots - kinematics, path planning, and control.” Springer, 2008.

[58] P. Lamon, *3D-Position Tracking and Control for All-Terrain Robots*, 1st ed., vol. 43. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[59] M. Tarokh and G. J. McDermott, “Kinematics modeling and analyses of articulated rovers,” *IEEE Trans. Robot.*, vol. 21, no. 4, pp. 539–553, Aug. 2005.

[60] P. Ben-Tzvi and W. Saab, “A Hybrid Tracked-Wheeled Multi-Directional Mobile Robot,” *J. Mech. Robot.*, vol. 11, no. 4, Aug. 2019.

[61] P. Ben-Tzvi, A. A. Goldenberg, and J. W. Zu, “Articulated hybrid mobile robot mechanism with compounded mobility and manipulation and on-board wireless sensor/actuator control interfaces,” *Mechatronics*, vol. 20, no. 6, pp. 627–639, Sep. 2010.

[62] P. Ben-Tzvi, S. Ito, and A. A. Goldenberg, “A mobile robot with autonomous climbing and descending of stairs,” *Robotica*, vol. 27, no. 2, pp. 171–188, Mar. 2009.

[63] P. Ben-Tzvi and P. M. Moubarak, “A mobile Robot with Hybrid Traction and Mobility Mechanism,” U.S. Patent 9,004,200, Issued April 14, 2015.

[64] P. Ben-Tzvi, “A Mobile Robot with Symbiosis of Locomotion and Manipulation,” U.S. Patent 8,225,892, Issued July 24, 2012.

[65] P. Ben-Tzvi and A. A. Goldenberg, “Hybrid Mobile Robot,” U.S. Patent 7,874,386, Issued January 25, 2011.

[66] C. Wong, E. Yang, X.-T. Yan, and D. Gu, “Adaptive and intelligent navigation of autonomous planetary rovers — A survey,” in *2017 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, 2017, pp. 237–244.

[67] T. M. Howard and A. Kelly, “Optimal Rough Terrain Trajectory Generation for Wheeled Mobile Robots,” *Int. J. Rob. Res.*, vol. 26, no. 2, pp. 141–166, Feb. 2007.

[68] T. Shan, J. Wang, B. Englot, and K. Doherty, “Bayesian Generalized Kernel

Inference for Terrain Traversability Mapping," in *Proceedings of The 2nd Conference on Robot Learning*, 2018, vol. 87, pp. 829–838.

[69] P. N. Currier, "A method for modeling and prediction of ground vehicle dynamics and stability in autonomous systems," Doctoral Dissertation, Virginia Tech, 2011.

[70] P. Papadakis, "Terrain traversability analysis methods for unmanned ground vehicles: A survey," *Eng. Appl. Artif. Intell.*, vol. 26, no. 4, pp. 1373–1385, Apr. 2013.

[71] W. Rone and P. Ben-Tzvi, "Mapping, localization and motion planning in mobile multi-robotic systems," *Robot. J.*, vol. 31, no. 1, pp. 1–23, Jan. 2013.

[72] P. Moubarak and P. Ben-Tzvi, "Modular and reconfigurable mobile robotics," *Robotics and Autonomous Systems*, vol. 60, no. 12. pp. 1648–1663, 2012.

[73] P. N. Currier and A. L. Wicks, "A novel method for prediction of mobile robot maneuvering spaces," *J. Terramechanics*, vol. 50, no. 2, pp. 85–97, Apr. 2013.

[74] S. Thrun *et al.*, "Stanley: The robot that won the DARPA Grand Challenge," *J. F. Robot.*, vol. 23, no. 9, pp. 661–692, Sep. 2006.

[75] A. Bacha *et al.*, "Odin: Team VictorTango's entry in the DARPA Urban Challenge," *J. F. Robot.*, vol. 25, no. 8, pp. 467–492, Aug. 2008.

[76] C. P. Urmson *et al.*, "High Speed Navigation of Unrehearsed Terrain : Red Team Technology for Grand Challenge 2004," 2004.

[77] K. Chu, M. Lee, and M. Sunwoo, "Local Path Planning for Off-Road Autonomous Driving With Avoidance of Static Obstacles," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1599–1616, Dec. 2012.

[78] A. Goswami, "Hierarchical Off-Road Path Planning and Its Validation Using a

Scaled Autonomous Car ’,” Clemson University, 2017.

[79] S. Thrun, “Probabilistic robotics,” *Communications of the ACM - Robots: intelligence, versatility, adaptivity*, vol. 45, no. 3. pp. 52–57, 2002.

[80] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Springer Berlin Heidelberg, 2008.

[81] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, “The Office Marathon: Robust navigation in an indoor office environment,” in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 300–307.

[82] G. Reina, M. Bellone, L. Spedicato, and N. I. Giannoccaro, “3D traversability awareness for rough terrain mobile robots,” *Sens. Rev.*, vol. 34, no. 2, pp. 220–232, 2014.

[83] C. Castejón, B. L. Boada, D. Blanco, and L. Moreno, “Traversable region modeling for outdoor navigation,” *J. Intell. Robot. Syst. Theory Appl.*, vol. 43, no. 2–4, pp. 175–216, 2005.

[84] S. Garrido, L. Moreno, F. Martín, and D. Álvarez, “Fast marching subjected to a vector field–path planning method for mars rovers,” *Expert Syst. Appl.*, vol. 78, no. February, pp. 334–346, 2017.

[85] R. Raja, A. Dutta, and K. S. Venkatesh, “New potential field method for rough terrain path planning using genetic algorithm for a 6-wheel rover,” *Rob. Auton. Syst.*, vol. 72, pp. 295–306, 2015.

[86] D. Amorim and R. Ventura, “Towards Efficient Path Planning of a Mobile Robot on Rough Terrain,” *2014 IEEE Int. Conf. Auton. Robot Syst. Compet.*, pp. 22–27, 2014.

[87] K. Konolige, “A gradient method for realtime robot control,” in *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113),* 2000, vol. 1, pp. 639–646.

[88] Xiuyi Fan *et al.*, “Integrated planning and control of large tracked vehicles in open terrain,” in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 4424–4430.

[89] A. Kelly and A. Stentz, “Rough Terrain Autonomous Mobility Part 2 : An Active Vision , Predictive Control Approach,” *Auton. Robots*, vol. 5, pp. 163–198, 1998.

[90] H. Seraji and A. Howard, “Behavior-based robot navigation on challenging terrain: A fuzzy logic approach,” *IEEE Trans. Robot. Autom.*, vol. 18, no. 3, pp. 308–321, Jun. 2002.

[91] A. Howard, B. Werger, and H. Seraji, “Integrating terrain maps into a reactive navigation strategy,” in *2003 IEEE International Conference on Robotics and Automation*, 2003, vol. 2, pp. 2012–2017.

[92] M. Cherif and C. Laugier, “Using Physical Models to Plan Safe and Executable Motions for a Rover Moving on a Terrain,” in *Int. Workshop on Intelligent Robotic Systems*, pp. 57–66.

[93] K. Iagnemma and S. Dubowsky, “Mobile Robots in Rough Terrain,” in *Springer Tracts in Advanced Robotics*, vol. 12, no. 8, Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. XII, 111.

[94] M. Brunner, B. Bruggemann, and D. Schulz, “Hierarchical rough terrain motion planning using an optimal sampling-based method,” *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 5539–5544, 2013.

[95] Y. Kuwata, G. A. Fiore, J. Teo, E. Frazzoli, and J. P. How, “Motion planning for urban driving using RRT,” in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 1681–1686.

[96] R. Pepy, A. Lambert, and H. Mounier, “Path Planning using a Dynamic Vehicle Model,” in *2006 2nd International Conference on Information & Communication Technologies*, 2006, vol. 1, no. 1, pp. 781–786.

[97] S. M. LaValle and J. J. K. Jr., “Randomized Kinodynamic Planning.,” *I. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, 2001.

[98] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How, “Real-Time Motion Planning With Applications to Autonomous Urban Driving,” *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 5, pp. 1105–1118, Sep. 2009.

[99] D. J. Webb and J. van den Berg, “Kinodynamic RRT*: Optimal Motion Planning for Systems with Linear Differential Constraints,” *CoRR*, vol. abs/1205.5, 2012.

[100] S. Koenig and M. Likhachev, “D* Lite,” in *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, 2002, pp. 476–483.

[101] A. (Tony) Stentz, “The Focussed D* Algorithm for Real-Time Replanning,” in *Proceedings of the International Joint Conference on Artificial Intelligence*, 1995.

[102] E. J. Haug, *Computer Aided Kinematics and Dynamics of Mechanical Systems*. Allyn and Bacon, 1989.

[103] A. M. Ladd, “Motion Planning for Physical Simulation,” Rice University, 2006.

[104] M. G. Coutinho, *Guide to Dynamic Simulations of Rigid Bodies and Particle Systems*. London: Springer London, 2013.

[105] A. Roennau, F. Sutter, G. Heppner, J. Oberlaender, and R. Dillmann, “Evaluation

of physics engines for robotic simulations with a special focus on the dynamics of walking robots," in *2013 16th International Conference on Advanced Robotics (ICAR)*, 2013, pp. 1–7.

[106] "Bullet Physics Engine Ver. 2.87." [Online]. Available: http://bulletphysics.org/wordpress/. [Accessed: 11-Feb-2017].

[107] M. Egerstedt, K. Johansson, J. Lygeros, and S. Sastry, "Behavior based robotics using regularized hybrid automata," in *Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No.99CH36304)*, 1999, vol. 4, pp. 3400–3405 vol.4.

[108] M. Egerstedt, "Controls for the Masses [Focus on Education]," *IEEE Control Syst.*, vol. 33, no. 4, pp. 40–44, Aug. 2013.

[109] P. C. P. Cheng and S. M. LaValle, "Reducing metric sensitivity in randomized trajectory design," *Proc. 2001 IEEE/RSJ Int. Conf. Intell. Robot. Syst. Expand. Soc. Role Robot. Next Millenn. (Cat. No.01CH37180)*, vol. 1, pp. 43–48, 2001.

[110] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Rob. Res.*, vol. 30, no. 7, pp. 846–894, 2011.

[111] Coppelia Robotics, "V-REP," 2010. [Online]. Available: http://www.coppeliarobotics.com/. [Accessed: 11-Feb-2017].

[112] J. Hazevoet, M. Anders, and I. Huish, "ANT Landscape Extension for Blender 2.6." [Online]. Available: https://wiki.blender.org/index.php/Extensions:2.6/Py/Scripts/Add_Mesh/ANT_Landscape. [Accessed: 01-Jan-2017].

[113] Blender Foundation, "Blender v2.7." [Online]. Available:

https://www.blender.org/. [Accessed: 01-Jan-2017].

[114] S. S. Sohal, W. Saab, and P. Ben-Tzvi, “Improved Alignment Estimation for Autonomous Docking of Mobile Robots,” *Proceedings of 2018 ASME IDETC/CIE, 42nd Mechanisms and Robotics Conference*, vol. 5A-2018. American Society of Mechanical Engineers, Quebec City, Canada, Aug. 26-29, 2018., Aug-.

[115] W. Saab and P. Ben-Tzvi, “Development of a Novel Coupling Mechanism for Modular Self-Reconfigurable Mobile Robots,” *Proceedings of 2015 ASME IDETC/CIE, 39th Mechanisms and Robotics Conference*. Boston, MA, Aug. 2–5, 2015.

[116] P. M. Moubarak, P. Ben-Tzvi, Z. Ma, and E. J. Alvarez, “An active coupling mechanism with three modes of operation for modular mobile robotics,” *Proceedings of 2013 IEEE International Conference on Robotics and Automation (ICRA 2013)*. IEEE, Karlsruhe, Germany, May 6-10, 2013, pp. 5469-5474.

[117] A. Kumar and P. Ben-Tzvi, “Spatial Object Tracking System Based on Linear Optical Sensor Arrays,” *IEEE Sens. J.*, vol. 16, no. 22, pp. 7933–7940, Nov. 2016.

[118] L. Murphy, S. Martin, and P. Corke, “Creating and using probabilistic costmaps from vehicle experience,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 4689–4694.

[119] R. O. Chavez-Garcia, J. Guzzi, L. M. Gambardella, and A. Giusti, “Learning Ground Traversability from Simulations,” *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1695–1702, Sep. 2017.

[120] A. Maligo and S. Lacroix, “Classification of Outdoor 3D Lidar Data Based on Unsupervised Gaussian Mixture Models,” *IEEE Trans. Autom. Sci. Eng.*, vol. 14,

no. 1, pp. 5–16, Jan. 2017.

[121] B. Suger, B. Steder, and W. Burgard, “Traversability analysis for mobile robots in outdoor environments: A semi-supervised learning approach based on 3D-lidar data,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 3941–3946.

[122] M. A. Bekhti, Y. Kobayashi, and K. Matsumura, “Terrain traversability analysis using multi-sensor data correlation by a mobile robot,” in *2014 IEEE/SICE International Symposium on System Integration*, 2014, pp. 615–620.

[123] S. Martin and P. Corke, “Long-term exploration &amp;amp; tours for energy constrained robots with online proprioceptive traversability estimation,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 5778–5785.

[124] J. Sock, J. Kim, J. Min, and K. Kwak, “Probabilistic traversability map generation using 3D-LIDAR and camera,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 5631–5637.

[125] B. Sebastian, H. Ren, and P. Ben-Tzvi, “Neural Network Based Heterogeneous Sensor Fusion for Robot Motion Planning,” *Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2019), Macau, China*. 2019.

[126] P. Kumar, W. Saab, and P. Ben-Tzvi, “Design of a Multi-Directional Hybrid-Locomotion Modular Robot With Feedforward Stability Control,” *Proceedings of 2017 ASME IDETC/CIE, 41st Mechanisms & Robotics Conference*. Cleveland, Ohio, Aug. 6-9, 2017.

[127] D. D. Hoffman and W. A. Richards, “Parts of recognition,” *Cognition*, vol. 18, no. 1–3, pp. 65–96, Dec. 1984.

[128] M. Abadi *et al.*, “TensorFlow: A system for large-scale machine learning,” *Proc. 12th USENIX Conf. Oper. Syst. Des. Implement.*, p. 44, 2016.

[129] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *3rd Int. Conf. Learn. Represent.*, Dec. 2014.

[130] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, “Randomized Kinodynamic Motion Planning with Moving Obstacles,” *Int. J. Rob. Res.*, vol. 21, no. 3, pp. 233–255, Mar. 2002.

[131] S. H. Tang, F. Kamil, W. Khaksar, N. Zulkifli, and S. . Ahmad, “Robotic motion planning in unknown dynamic environments: Existing approaches and challenges,” *2015 IEEE Int. Symp. Robot. Intell. Sensors*, pp. 288–294, 2015.

[132] F. Rodríguez, *Autonomous Tracked Robots in Planar Off -Road Conditions*, vol. 6. Springer International Publishing, 2014.

[133] R. Siegwart, *Introduction to Autonomous Mobile Robots*. MIT Press, 2004.

[134] P. Corke, *Robotics, Vision and Control*. Springer International Publishing, 2011.

[135] R. Gonzalez and K. Iagnemma, “Slippage estimation and compensation for planetary exploration rovers. State of the art and future challenges,” *J. F. Robot.*, vol. 35, no. 4, pp. 564–577, Jun. 2018.

[136] M. Burke, “Path-following control of a velocity constrained tracked vehicle incorporating adaptive slip estimation,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2012, pp. 97–102.

[137] L. Ding, H. Gao, Z. Deng, and Z. Liu, “Slip-ratio-coordinated control of planetary

exploration robots traversing over deformable rough terrain," in *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, 2010, pp. 4958–4963.

[138] J. Kim and J. Lee, "A kinematic-based rough terrain control for traction and energy saving of an exploration rover," in *IEEE International Conference on Intelligent Robots and Systems*, 2016, vol. 2016-Novem, pp. 3595–3600.

[139] J. D. Terry and M. A. Minor, "Traction estimation and control for mobile robots using the wheel slip velocity," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2008, pp. 2003–2009.

[140] P. Moubarak and P. Ben-Tzvi, "A globally converging algorithm for adaptive manipulation and trajectory following for mobile robots with serial redundant arms," *Robotica*, vol. 31, no. 8, pp. 1299–1311, 2013.

[141] C. C. Ward and K. Iagnemma, "A dynamic-model-based wheel slip detector for mobile robots on outdoor terrain," *IEEE Trans. Robot.*, vol. 24, no. 4, pp. 821–831, Aug. 2008.

[142] K. Naveed, Z. H. Khan, and A. Hussain, "Adaptive Trajectory Tracking of Wheeled Mobile Robot with Uncertain Parameters," in *Computational Intelligence for Decision Support in Cyber-Physical Systems. Studies in Computational Intelligence*, Springer, Singapore, 2014, pp. 237–262.

[143] J. Taheri-Kalani and M. J. Khosrowjerdi, "Adaptive trajectory tracking control of wheeled mobile robots with disturbance observer," *Int. J. Adapt. Control Signal Process.*, vol. 28, no. 1, pp. 14–27, Jan. 2014.

[144] L. Auersch, "VEHICLE-TRACK-INTERACTION AND SOIL DYNAMICS,"

Veh. Syst. Dyn., vol. 29, no. S1, pp. 553–558, 1998.

[145] W. Yu, O. Y. Chuy, E. G. Collins, and P. Hollis, “Analysis and experimental verification for dynamic modeling of a skid-steered wheeled vehicle,” *IEEE Trans. Robot.*, vol. 26, no. 2, pp. 340–353, Apr. 2010.

[146] N. Seegmiller, F. Rogers-Marcovitz, G. Miller, and A. Kelly, “A Unified Perturbative Dynamics Approach to Online Vehicle Model Identification,” in *International Symposium on Robotics Research*, 2011, pp. 1–16.

[147] J. R. Fink and E. A. Stump, “Experimental analysis of models for trajectory generation on tracked vehicles,” in *IEEE International Conference on Intelligent Robots and Systems*, 2014, pp. 1970–1977.

[148] V. Rajagopalan, C. Mericli, and A. Kelly, “Slip-aware Model Predictive optimal control for Path following,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2016, vol. 2016-June, pp. 4585–4590.

[149] J. L. Martínez, A. Mandow, J. Morales, S. Pedraza, and A. García-Cerezo, “Approximating kinematics for tracked mobile robots,” *Int. J. Rob. Res.*, vol. 24(10), no. 10, pp. 867–878, Oct. 2005.

[150] A. Mandow, J. L. Martínez, J. Morales, J. L. Blanco, A. García-Cerezo, and J. González, “Experimental kinematics for wheeled skid-steer mobile robots,” in *IEEE International Conference on Intelligent Robots and Systems*, 2007, pp. 1222–1227.

[151] J. Pentzer, S. Brennan, and K. Reichard, “Model-based prediction of skid-steer robot kinematics using online estimation of track instantaneous centers of rotation,” *J. F. Robot.*, vol. 31, no. 3, pp. 455–476, May 2014.

[152] J. Pentzer, S. Brennan, and K. Reichard, “The use of unicycle robot control strategies for skid-steer robots through the ICR kinematic mapping,” in *IEEE International Conference on Intelligent Robots and Systems*, 2014, pp. 3201–3206.

[153] D. M. Helmick, S. I. Roumeliotis, Y. Cheng, D. S. Clouse, M. Bajracharya, and L. H. Matthies, “Slip-compensated path following for planetary exploration rovers,” *Adv. Robot.*, vol. 20, no. 11, pp. 1257–1280, Jan. 2006.

[154] S. L. Canfield, T. W. Hill, and S. G. Zuccaro, “Prediction and Experimental Validation of Power Consumption of Skid-Steer Mobile Robots in Manufacturing Environments,” *J. Intell. Robot. Syst.*, vol. 94, no. 3–4, pp. 825–839, Jun. 2019.

[155] S. L. Canfield, T. W. Hill, and S. G. Zuccaro, “Modeling Power Requirements for Skid-Steer Mobile Robots in Manufacturing Environments,” in *ASME. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2016.

[156] N. Gupta, C. Ordonez, and E. G. Collins, “Dynamically feasible, energy efficient motion planning for skid-steered vehicles,” *Auton. Robots*, vol. 41, no. 2, pp. 453–471, Feb. 2017.

[157] B. Sebastian and P. Ben-Tzvi, “Active Disturbance Rejection Control for Handling Slip in Tracked Vehicle Locomotion,” *J. Mech. Robot.*, vol. 11, no. 2, p. 021003, Apr. 2019.

[158] J. Han, “From PID to active disturbance rejection control,” in *IEEE Transactions on Industrial Electronics*, 2009, vol. 56, no. 3, pp. 900–906.

[159] G. Herbst, “A Simulative Study on Active Disturbance Rejection Control (ADRC) as a Control Tool for Practitioners,” *Electronics*, vol. 2, no. 4, pp. 246–279, 2013.

[160] N. Seegmiller, F. Rogers-Marcovitz, G. Miller, and A. Kelly, “Vehicle model identification by integrated prediction error minimization,” *Int. J. Rob. Res.*, vol. 32, no. 8, pp. 912–931, Jul. 2013.

[161] B. Sebastian, A. Williams, and P. Ben-Tzvi, “Gaussian Kernel Controller for Path Tracking in Mobile Robots,” *Proceedings of the 2018 ASME IDETC/CIE, 42nd Mechanisms & Robotics Conference*. Quebec City, Canada, Aug. 26–29, 2018.

[162] J. J. Park and B. Kuipers, “A smooth control law for graceful motion of differential wheeled mobile robots in 2D environment,” *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 4896–4902, 2011.

[163] J. Wit, C. D. Crane, and D. Armstrong, “Autonomous ground vehicle path tracking,” *J. Robot. Syst.*, vol. 21, no. 8, pp. 439–449, 2004.

[164] R. C. Coulter, “Implementation of the Pure Pursuit Path Tracking Algorithm,” 1992.

[165] Open Source Robotics Foundation, “Indigo - ROS Wiki,” 2014. [Online]. Available: http://wiki.ros.org/indigo. [Accessed: 07-Jul-2018].

[166] Pozyx Labs, “Pozyx Accurate Positioning,” 2015. [Online]. Available: https://www.pozyx.io/. [Accessed: 25-Jun-2018].

[167] P. M. Moubarak and P. Ben-Tzvi, “Adaptive manipulation of a Hybrid Mechanism Mobile Robot,” *Proceedings of 2011 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*. Montreal, Quebec, Canada, pp. 113-118, September 17-18, 2011.

[168] P. Ben-Tzvi, “Experimental validation of a hybrid mobile robot mechanism with interchangeable locomotion and manipulation,” *Proceedings of 2009 IEEE/RSJ*

International Conference on Intelligent Robots and Systems (IROS 2009). St. Louis, Missouri, pp. 420-421, October 11-15, 2009.

[169] P. Ben-Tzvi, C. Raoufi, A. A. Goldenberg, and J. W. Zu, “Virtual Prototype Development and Simulations of a Tracked Hybrid Mobile Robot,” *Proceedings of MSC. Software 2007 Virtual Product Development Conference*. Detroit, Michigan, October 10-12, 2007.

[170] P. Ben-Tzvi, A. A. Goldenberg, and J. W. Zu, “Design, simulations and optimization of a tracked mobile robot manipulator with hybrid locomotion and manipulation capabilities,” *Proceedings of 2008 IEEE International Conference on Robotics and Automation (ICRA 2008)*. Pasadena, California, pp. 2307-2312, May 19-23, 2008.

[171] P. Ben-Tzvi, A. A. Goldenberg, and J. W. Zu, “Implementation of Sensors and Control Paradigm for a Hybrid Mobile Robot Manipulator for Search and Rescue Operations,” *Proceedings of 2007 International Workshop on Robotic and Sensors Environments (ROSE 2007)*. Ottawa, Ontario, Canada, pp. 92-97, October 12-13, 2007.

[172] P. Ben-Tzvi, A. A. Goldenberg, and J. W. Zu, “Design and Analysis of a Hybrid Mobile Robot Mechanism With Compounded Locomotion and Manipulation Capability,” *J. Mech. Des.*, vol. 130, no. 7, pp. 1–13, 2008.

[173] P. Ben-Tzvi, “Experimental validation and field performance metrics of a hybrid mobile robot mechanism,” *J. F. Robot.*, vol. 27, no. 3, pp. 250–267, 2010.

[174] “DARPA Subterranean (SubT) Challenge,” 2017. [Online]. Available: https://www.darpa.mil/program/darpa-subterranean-challenge. [Accessed: 06-Nov-

2018].

[175] Johnston, “Off-highway obstacle detection,” *IEEE Instrum. Meas. Mag.*, vol. 9, no. 6, pp. 16–24, Oct. 2006.

[176] J. Mei, Y. Yu, H. Zhao, and H. Zha, “Scene-Adaptive Off-Road Detection Using a Monocular Camera,” *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 242–253, Jan. 2018.

[177] M. Bellone, G. Reina, L. Caltagirone, and M. Wahde, “Learning Traversability From Point Clouds in Challenging Scenarios,” *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 296–305, Jan. 2018.

[178] K. Iagnemma, S. Kang, H. Shibly, and S. Dubowsky, “Online Terrain Parameter Estimation for Wheeled Mobile Robots With Application to Planetary Rovers,” *IEEE Trans. Robot.*, vol. 20, no. 5, pp. 921–927, Oct. 2004.

[179] G. Reina, L. Ojeda, A. Milella, and J. Borenstein, “Wheel slippage and sinkage detection for planetary rovers,” *IEEE/ASME Trans. Mechatronics*, vol. 11, no. 2, pp. 185–195, Apr. 2006.

[180] C. A. Brooks and K. Iagnemma, “Vibration-based terrain classification for planetary exploration rovers,” *IEEE Trans. Robot.*, vol. 21, no. 6, pp. 1185–1191, Dec. 2005.

[181] P. Giguere and G. Dudek, “A Simple Tactile Probe for Surface Identification by Mobile Robots,” *IEEE Trans. Robot.*, vol. 27, no. 3, pp. 534–544, Jun. 2011.

[182] B. Park, J. Kim, and J. Lee, “Terrain Feature Extraction and Classification for Mobile Robots Utilizing Contact Sensors on Rough Terrain,” *Procedia Eng.*, vol. 41, pp. 846–853, Jan. 2012.

[183] D. F. Wolf and G. S. Sukhatme, “Semantic Mapping Using Mobile Robots,” *IEEE Trans. Robot.*, vol. 24, no. 2, pp. 245–258, Apr. 2008.

[184] C. Weiss, H. Frohlich, and A. Zell, “Vibration-based Terrain Classification Using Support Vector Machines,” in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems,* 2006, pp. 4429–4434.

[185] C. Weiss, N. Fechner, M. Stark, and A. Zell, “Comparison of Different Approaches to Vibration-based Terrain Classification.,” in *3rd European Conference on Mobile Robots,* 2007, pp. 7–12.

[186] P. Giguere, G. Dudek, S. Saunderson, and C. Prahacs, “Environment Identification for a Running Robot Using Inertial and Actuator Cues,” in *Robotics: Science and Systems II,* 2006.

[187] S. Wang, S. Kodagoda, L. Shi, and X. Dai, “Two-Stage Road Terrain Identification Approach for Land Vehicles Using Feature-Based and Markov Random Field Algorithm,” *IEEE Intell. Syst.*, vol. 33, no. 1, pp. 29–39, Jan. 2018.

[188] S. Khaleghian and S. Taheri, “Terrain classification using intelligent tire,” *J. Terramechanics*, vol. 71, pp. 15–24, Jun. 2017.

[189] S. Khaleghian, A. Emami, and S. Taheri, “A technical survey on tire-road friction estimation,” *Friction*, vol. 5, no. 2, pp. 123–146, Jun. 2017.

[190] B. Sebastian and P. Ben-Tzvi, “Support vector machine based real-time terrain estimation for tracked robots,” *Mechatronics J.*, vol. 62, p. 102260, Oct. 2019.

[191] A. Kumar and P. Ben-Tzvi, “Novel wireless sensing platform for experimental mapping and validation of ship air wake,” *Mechatronics J.*, vol. 52, pp. 58–69, Jun. 2018.

[192] A. Karpatne *et al.*, “Theory-Guided Data Science: A New Paradigm for Scientific Discovery from Data,” *IEEE Trans. Knowl. Data Eng.*, vol. 29(10), no. 10, pp. 2318–2331, Oct. 2017.

[193] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, “Data-driven discovery of partial differential equations,” *Sci. Adv.*, vol. 3(4), no. 4, p. e1602614, Apr. 2017.

[194] C. Cortes and V. Vapnik, “Supprot-Vector Networks,” *Mach. Learn.*, vol. 297(20), no. 20, pp. 273–297, 1995.

[195] MathWorks, “Statistics and Machine Learning Toolbox, MATLAB,” 2006. [Online]. Available: https://www.mathworks.com/help/stats/index.html. [Accessed: 10-Jan-2019].

[196] R. R. Murphy, S. Tadokoro, and A. Kleiner, “Disaster Robotics,” in *Springer Handbook of Robotics*, Springer International Publishing, 2016, pp. 1577–1604.

[197] R. R. Murphy, “Trial by fire,” *IEEE Robot. Autom. Mag.*, vol. 11, no. 3, pp. 50–61, Sep. 2004.

[198] S. Tadokoro, “DDT project on rescue robots and systems,” in *2006 SICE-ICASE International Joint Conference*, 2006, pp. 3429–3434.

[199] F. Matsuno and S. Tadokoro, “Rescue Robots and Systems in Japan,” in *2004 IEEE International Conference on Robotics and Biomimetics*, 2004, pp. 12–20.

[200] R. Chipalkatty, G. Droge, and M. B. Egerstedt, “Less Is More: Mixed-Initiative Model-Predictive Control With Human Inputs,” *IEEE Trans. Robot.*, vol. 29, no. 3, pp. 695–703, Jun. 2013.

[201] J. Storms, K. Chen, and D. Tilbury, “A shared control method for obstacle avoidance with mobile robots and its interaction with communication delay,” *Int.*

J. Rob. Res., vol. 36, no. 5–7, pp. 820–839, Jun. 2017.

[202] Z. Ma and P. Ben-Tzvi, “An Admittance Glove Mechanism for Controlling a Mobile Robot,” *Proceedings of the ASME 2012 IIDETC/CIE, 36th Mechanisms & Robotics Conference*. Chicago, IL, Aug. 12-15, 2012.

[203] Z. Ma and P. Ben-Tzvi, “RML glove-an exoskeleton glove mechanism with haptics feedback,” *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 2, pp. 641–652, Apr. 2015.

[204] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[205] K. B. Singh, M. Ali Arat, and S. Taheri, “An Intelligent Tire Based Tire-Road Friction Estimation Technique and Adaptive Wheel Slip Controller for Antilock Brake System,” *J. Dyn. Syst. Meas. Control*, vol. 135, no. 3, p. 031002, Feb. 2013.