Increasing the autonomy of planetary exploration robots

Politecnico
di Bari

Department of Mechanics, Mathematics and Management

Aerospace Science and Engineering

Ph.D. Program

SSD: ING-IND/13–APPLIED MECHANICS

Final Dissertation

Increasing the autonomy of planetary exploration robots

by

Ugenti Angelo

Supervisors:

Prof. Giulio Reina

Prof. Giacomo Mantriota

Coordinator of Ph.D. Program:

Prof. Marco Donato De Tullio

Course n°36, 01/11/2020-31/10/2023

Politecnico
di Bari

LIBERATORIA PER L'ARCHIVIAZIONE DELLA TESI DI DOTTORATO

Al Magnifico Rettore
del Politecnico di Bari

Il/la sottoscritto/a Angelo Ugenti nato/a a Bitonto il 17/04/1993

residente a Bitetto in via Papa Pio IX, 2 e-mail angelo.ugenti@poliba.it

iscritto al 3° anno di Corso di Dottorato di Ricerca in Ingegneria e Scienze Aerospaziali ciclo XXXVI

ed essendo stato ammesso a sostenere l'esame finale con la prevista discussione della tesi dal titolo:

INCREASING THE AUTONOMY OF PLANETARY EXPLORATION ROBOTS

DICHIARA

1) di essere consapevole che, ai sensi del D.P.R. n. 445 del 28.12.2000, le dichiarazioni mendaci, la falsità negli atti e l'uso di atti falsi sono puniti ai sensi del codice penale e delle Leggi speciali in materia, e che nel caso ricorressero dette ipotesi, decade fin dall'inizio e senza necessità di nessuna formalità dai benefici conseguenti al provvedimento emanato sulla base di tali dichiarazioni;
2) di essere iscritto al Corso di Dottorato di ricerca in Ingegneria e Scienze Aerospaziali ciclo XXXVI, corso attivato ai sensi del "*Regolamento dei Corsi di Dottorato di ricerca del Politecnico di Bari*", emanato con D.R. n.286 del 01.07.2013;
3) di essere pienamente a conoscenza delle disposizioni contenute nel predetto Regolamento in merito alla procedura di deposito, pubblicazione e autoarchiviazione della tesi di dottorato nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica;
4) di essere consapevole che attraverso l'autoarchiviazione delle tesi nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica del Politecnico di Bari (IRIS-POLIBA), l'Ateneo archivierà e renderà consultabile in rete (nel rispetto della Policy di Ateneo di cui al D.R. 642 del 13.11.2015) il testo completo della tesi di dottorato, fatta salva la possibilità di sottoscrizione di apposite licenze per le relative condizioni di utilizzo (di cui al sito http://www.creativecommons.it/Licenze), e fatte salve, altresì, le eventuali esigenze di "embargo", legate a strette considerazioni sulla tutelabilità e sfruttamento industriale/commerciale dei contenuti della tesi, da rappresentarsi mediante compilazione e sottoscrizione del modulo in calce (Richiesta di embargo);
5) che la tesi da depositare in IRIS-POLIBA, in formato digitale (PDF/A) sarà del tutto identica a quelle **consegnate**/inviate/da inviarsi ai componenti della commissione per l'esame finale e a qualsiasi altra copia depositata presso gli Uffici del Politecnico di Bari in forma cartacea o digitale, ovvero a quella da discutere in sede di esame finale, a quella da depositare, a cura dell'Ateneo, presso le Biblioteche Nazionali Centrali di Roma e Firenze e presso tutti gli Uffici competenti per legge al momento del deposito stesso, e che di conseguenza va esclusa qualsiasi responsabilità del Politecnico di Bari per quanto riguarda eventuali errori, imprecisioni o omissioni nei contenuti della tesi;
6) che il contenuto e l'organizzazione della tesi è opera originale realizzata dal sottoscritto e non compromette in alcun modo i diritti di terzi, ivi compresi quelli relativi alla sicurezza dei dati personali; che pertanto il Politecnico di Bari ed i suoi funzionari sono in ogni caso esenti da responsabilità di qualsivoglia natura: civile, amministrativa e penale e saranno dal sottoscritto tenuti indenni da qualsiasi richiesta o rivendicazione da parte di terzi;
7) che il contenuto della tesi non infrange in alcun modo il diritto d'Autore né gli obblighi connessi alla salvaguardia di diritti morali od economici di altri autori o di altri aventi diritto, sia per testi, immagini, foto, tabelle, o altre parti di cui la tesi è composta.

Bari, 15/01/2024 Firma Angelo Ugenti

Il/La sottoscritto, con l'autoarchiviazione della propria tesi di dottorato nell'Archivio Istituzionale ad accesso aperto del Politecnico di Bari (POLIBA-IRIS), pur mantenendo su di essa tutti i diritti d'autore, morali ed economici, ai sensi della normativa vigente (Legge 633/1941 e ss.mm.ii.),

CONCEDE

- al Politecnico di Bari il permesso di trasferire l'opera su qualsiasi supporto e di convertirla in qualsiasi formato al fine di una corretta conservazione nel tempo. Il Politecnico di Bari garantisce che non verrà effettuata alcuna modifica al contenuto e alla struttura dell'opera.
- al Politecnico di Bari la possibilità di riprodurre l'opera in più di una copia per fini di sicurezza, back-up e conservazione.

Bari, 15/10/2024 Firma Angelo Ugenti

Politecnico
di Bari

Department of Mechanics, Mathematics and Management

Aerospace Science and Engineering

Ph.D. Program

SSD: ING-IND/13–APPLIED MECHANICS

Final Dissertation

Increasing the autonomy of planetary exploration robots

by

Ugenti Angelo

Angelo Ugenti

Referees:

Prof. Carlos J. Pérez del Pulgar

Prof. Giovanni Boschetti

Supervisors:

Prof. Giulio Reina

Giulio Reina

Prof. Giacomo Mantriota

Coordinator of Ph.D Program:

Prof. Marco Donato De Tullio

Marco Donato de Tullio *firma*

Course n°36, 01/11/2020-31/10/2023

Acknowledgements

Vorrei esprimere sincera gratitudine nei confronti dei miei supervisori, prof. Giulio Reina e prof. Giacomo Mantriota. La loro fiducia nei miei confronti e la loro passione per la ricerca e la didattica sono state fonti di ispirazione per questo percorso. Questo lavoro di tesi non sarebbe stato possibile senza la loro conoscenza e disponibilità.

I'd like to aknowledge prof. Hamid Taghavifar, for sharing his expertise with me during my stay at Concordia University in Montreal.

Infine, una dedica a María, alla mia famiglia e ai miei amici. A loro devo la mia serenità.

ABSTRACT

Mobile robots designed for planetary exploration are essential for advancing our understanding of celestial bodies, enabling scientists to collect valuable data and explore areas beyond the reach of human astronauts. These robots are equipped with advanced sensors, cameras, scientific instruments, and communication systems, allowing them to navigate autonomously, analyze the environment, and transmit data back to Earth. This thesis focuses on increasing the degree of autonomy of planetary exploration robots. The research objectives are divided into three main topics:

- Terrain Awareness: This research leverages machine learning algorithms to classify terrain types using proprioceptive data modulated by rover-environment interactions. By selecting informative subsets of this data, the aim is to improve terrain classification for both generalization and extrapolation, crucial for long-range navigation and safety.
- Innovative Suspension Systems: a novel off-road tracked robot is introduced in this study, and its suspension system is evaluated through analytical and multibody models. By analyzing its performance on challenging terrains this research contributes to improving the design of robotic systems for planetary exploration.
- Path Planning: This work delves into reactive computing path planning, focusing on the use of Harmonic Artificial Potential Fields (HAPF). The aim is to enhance local and global planning while addressing typical reactive computing limitations, such as local minima and sub-optimality.

By addressing these research objectives, this thesis seeks to advance the capabilities of mobile robots for planetary exploration, contributing to safer and more efficient exploration of celestial bodies, such as Mars and the Moon.

Contents

List of Figures

List of Tables

Acronyms

ADE Autonomous DEcision making in very long traverses. 2, 17, 19

APF Artificial Potential Field. 7, 8, 15, 16, 125

BFA Bogie Front Arm. 42, 43, 47

BRA Bogie Rear Arm. 42, 43, 47

CNN Convolutional Neural Network. 4, 11, 20, 21, 23, 68, 69, 71, 72, 74, 75, 122, 123

DFKI Deutsches Forschungszentrum für Künstliche Intelligenz. 17, 18

DMS Degrees Minutes Seconds. 19

DOF Degrees Of Freedom. 17, 18

DTM Digital Terrain Model. 53, 54, 118, 125

FFT Fast Fourier Transform. 21

FSA Front Swing Arm. 36, 42, 43, 47, 77, 85, 86

FTS Force-Torque Sensor. 17

GPS Global Positioning System. 10, 19, 96, 101, 104

HAPF Harmonic Artificial Potential Field. 7, 8, 14–16, 54, 59, 122, 125

HiRISE High Resolution Imaging Science Experiment. 53

IMU Inertial Measurement Unit. 10, 18, 25, 96, 101

JT Joint Telemetry. 18, 25, 26

KPI Key Performance Indicator. 83, 85, 86, 89

LC Load Cell. 18, 25, 26, 28

MB MultiBody. 5, 6, 14, 92, 95, 96, 104, 107, 124, 125

NN Neural Network. 11

PCA Principal Component Analysis. 10

PDE Partial Differential Equation. 7, 15

PER Planetary Exploration Robot. 1–5

PWM Pulse Width Modulation. 18, 25, 26

RELU REctified Linear Unit. 22

RMS Root Mean Square. 99, 101, 104, 124, 125

RSA Rear Swing Arm. 36, 42, 43, 47

SVM Support Vector Machine. 4, 10, 11, 19–21, 23, 29, 32, 34, 67–69, 71, 72, 74, 75, 122, 123

Chapter 1

Introduction

1.1 Mobile robots for planetary exploration

Mobile robots for planetary exploration are designed to operate on the surfaces of celestial bodies, and they play a crucial role in space exploration by enabling scientists to gather valuable data, conduct experiments, and explore areas that are difficult or unsafe for human astronauts to reach. Mobile robots are equipped with advanced sensors, cameras, and scientific instruments, allowing them to navigate, collect samples, and analyze the environment.

The key features of a Planetary Exploration Robot (PER) are:

- **Autonomous Decision Making**: advanced algorithms allow PERs to plan a trajectory, move independently, avoid obstacles and make decisions based on their surroundings;

- **Robust Mobility**: PERs are designed to navigate challenging terrains such as rocky surfaces, steep slopes, and sandy dunes;

- **Scientific Instruments**: PERs are equipped with a variety of scientific instruments such as spectrometers, cameras, and sensors to analyze the surrounding environment;

- **Communication Systems**: Mobile robots are equipped with communication systems to transmit data back to Earth. These systems are essential for sending scientific data, images, and status reports to mission control;

- **Energy Sources**: PERs rely on various energy sources such as solar panels or radioisotope thermoelectric generators to power their systems;
- **Extreme Environment Adaptability**: PERs must withstand extreme temperatures, radiation, and dust storms prevalent on other planets.

This thesis focuses on the first two key features, i.e. autonomous decision making and robust mobility.

1.2 Research objectives

As stated in the previous section, this work will focus on autonomous decision making and robust mobility. During the PhD program, three main topics were investigated within these two macro research areas:

- terrain awareness;
- study, modeling and validation of an innovative suspension system for mobile robots in unstructured environments;
- path planning for autonomous vehicles.

These three topics are highly correlated. Terrain awareness and dynamic models can be used by an autonomous robot to decide the best locomotion strategy in each moment of the mission. Also, an autonomous path planner can use this information to plan the optimal path for the rover.

The following sections present the rationale behind the interest for these topics and the intended outcomes of this work.

1.2.1 Terrain awareness

This section and all those related to this topic are based on the published paper [1] and it has been developed as part of the research activity for the project Autonomous DEcision making in very long traverses (ADE) [2], funded by the European Union's Horizon 2020 research and innovation programme. The main goal of ADE was to develop and test a rover system capable to achieve autonomous long-range navigation in hostile environments, while guaranteeing consistent data collection. The mobility range of PERs has been up to date limited to few hundreds of meters per sol [3, 4, 5]. From a purely technical point of view, this limitation has both hardware and software sources. The former is the most important and consists in the finite power storage of a rover locomotion system, that is fixed given a robot design. The software-related source corresponds to the limited skills in terms of autonomous decision-making, that can be improved by artificial intelligence. Improving these capabilities extends the autonomy of the rover across multiple geographical areas and therefore expands opportunities of data collection. Directly related to long-range navigation is also the safety issue. The importance of sensing hazards was highlighted, for example, in April 2005, when the Mars exploration rover Opportunity embedded itself in a dune of loosely packed drift material [6]. The terrain geometry as reconstructed from a distance via stereovision did not indicate any hazard. However, the high compressibility of the loose drift material caused the wheels to sink deeply into the surface. The combination of the drift's low internal friction and the motion resistance due to sinkage prevented the rover from producing sufficient thrust to travel up the slope. Opportunity's progress was delayed for more than a month while engineers worked to find a way out. A similar embedding event led to the end of operations for the twin rover Spirit in 2010. Therefore, future generations of PERs will require key technologies suitable to overcome these limitations, performing long traverses while guaranteeing fast reaction, mission reliability and safety, and optimal exploitation of the robot's resources within reasonable costs. In this context, the ability to sense and characterize the incoming terrain would represent an enabling technology towards long-term autonomy and potential hazard avoidance [7]. In this thesis, the objective is to demonstrate the potential of terrain classification via learning algorithms that are trained on proprioceptive features. Here, proprioceptive features refer to statistics that are extracted from the measurement of a physical variable pertaining to the robot-environment interaction, e.g., wheel velocity, forces, body linear and angular accelerations. The hypothesis is that, being modulated by the terrain properties, these features are a rich source of information from which the specific terrain type can be inferred via learning approaches [8, 9].

One of the contributions of this research refers to the selection of the most informative subset of proprioceptive features derived from the sensor suite integrated onboard of PERs. A range of aspects is addressed that includes feature extraction, feature ranking, multivariate feature selection and efficient feature space construction. While feature selection has been largely investigated in other domains e.g., image processing, text processing and gene expression analysis [10], it remains largely under investigated for the terrain classification problem of PER, and rough-terrain robots, in general. In contrast to other areas of applications where datasets with tens or hundreds of thousands of variables are available forming a statistically significant population, data acquired by a rover driving over natural terrain present many challenges such as sparseness, presence of unknown and uncontrolled disturbances, dependence on the specific time and site of the acquisition. The objectives pursued by feature selection include improvement in the prediction performance, reduction in training time, computational burden and memory usage of the algorithm and facilitation of understanding the underlying process that generated the data.

The other contribution of this research is the adoption of a suitable learning algorithm to infer the type of terrain from the selected feature set. This algorithm will have to look for patterns in the data to construct the mapping from the proprioceptive measurements to the corresponding terrain type. The well-known Support Vector Machine (SVM) is contrasted with a deep Convolutional Neural Network (CNN). While SVM requires in input hand-crafted features that are selected during a pre-processing stage, CNN uses learned features that are extracted automatically form the signal time histories. An important goal of the proposed approach is to improve the performance of terrain classifiers for two use cases: generalization and extrapolation. Generalization is defined as the performance of an algorithm on previously unseen observations (test set) that is extracted from the same distribution as the data in the training set, e.g., the same test run. The error measured on the test set corresponds to the on-line performance of the model and depends on the operating conditions included in the training set. The second use case, extrapolation, is even more challenging since, in general, learning algorithms are known to perform poorly outside the training data population. We compare the performance of the two terrain classifiers (SVM and CNN) for both generalization and extrapolation.

1.2.2 Innovative suspension systems

This section and all those related to this topic are based on the published papers [11] and [12]. A critical aspect of PERs (and in general for every off-road robot) is locomotion design and, depending on the specific application, researchers can choose between wheeled, legged, tracked or hybrid robots. The locomotion type for off-road robots is designed according to the required performance in terms of maximum speed, energetic efficiency, and mobility in unstructured environments. In general, wheeled robot excel in speed and energetic efficiency, while legged platforms perform better than the others in unstructured environments [13]. Tracked robots represent a middle point, more efficient than legged robots, and with better mobility than wheeled platforms. For this reason, and thanks to their large contact area with the ground, tracked robots have gained a lot of interest for off-road applications, especially on soft and yielding terrains. Although there are a few concepts of tracked PER in the literature (e.g. the one presented in [14]), the use of tracks remains under-investigated in this sector.

The purpose of this part of the thesis is to introduce and study a novel architecture of an off-road tracked robot named Polibot. The robot is shown in Fig. 1.1(a) along with a CAD rendering with the main dimensions (Fig. 1.1(b)). On each side, the robot features a rubber track wrapped around four independently suspended road wheels, an idler wheel, and the sprocket. An accurate evaluation of the performance of the suspension system of a PER allows to understand what kind of obstacles can be overcome by the robot. This study included a complete kinematic and dynamic analysis of the Polibot, thanks to the use of two models: a simple analytical model built in MATLAB®, and a more complex MultiBody (MB) model built in MSC Adams®. These simulations will then be validated against experimental tests on the real robot. The analytical model is a tool for the inverse kinematics of the proposed design and predicts the system configuration given the terrain geometry. The analytical model can be useful as well to evaluate the influence of the many design parameters on the robot behaviour.

Moreover, this work presents the development of the Polibot MB model and its validation against the experimental prototype in both static and dynamic tests and the performance evaluation on various challenging scenarios, including positive and negative obstacles, and stochastic uneven terrain. Tracked robots are examples

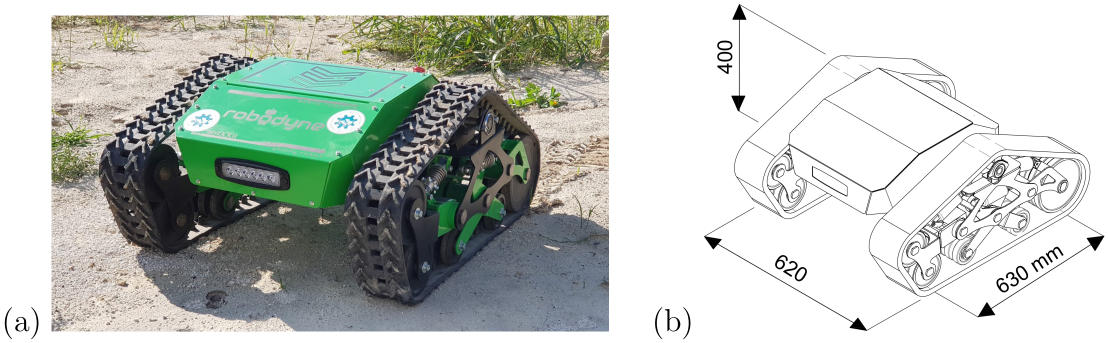


Figure 1.1: The all-terrain rover Polibot: (a) prototype, and (b) isometric CAD view with the indication of the main dimensions.

of flexible MB system with challenges that include the presence of many friction contacts, track modeling, and track-terrain interaction. In this research, the Polibot digital twin is developed within MSC Adams [15] using the toolkit Adams Tracked Vehicle (ATV) [16] that is under testing and development at the Robotic Mobility Lab of Politecnico di Bari. Following this approach, many system configurations can be evaluated from the early design phase and a wide number of operating conditions can be simulated with high fidelity without performing expensive and time-consuming field experiments.

1.2.3 Path planning

Path planning plays a vital role in enabling mobile robots to autonomously navigate and complete tasks in a wide range of environments, such as warehouses, factories, or outdoor unstructured spaces, such as those commonly found in space applications [17]. Over the recent years, planetary exploration related projects are aiming to achieve a rover autonomy of 1000 km per day [2]. For these applications, path planning involves determining a safe and efficient path for a robot to traverse from its current location to a desired destination while avoiding obstacles and considering other constraints, such as highly deformable terrains and scientific return requirements. There are several common approaches to path planning for mobile robots: grid-based and sampling-based approaches, reactive methods, Voronoi diagrams, optimization methods and machine-learning algorithms [18]. This thesis focuses on reactive computing, because these algorithms are capable of quickly incorporating new information, such as newly discovered obstacles, and react while moving in unstructured environments, with limited prior information.

The most common approach to reactive computing is Artificial Potential Field (APF), which was first introduced by Khatib [19]. In Khatib's work, the APF is a continuous function with its lowest value in correspondence of the goal location, simulating a force that attracts the robot to the target. Khatib also included high values of the APF close to obstacles, simulating a repulsive force that pushes the robot farther away. By following the gradient descent of the APF, the robot should be able to reach its final destination [20]. The main features of the APF approach are the low computational requirements needed to produce the potential field and find a path and the ability to quickly react to uncertain or dynamic environments[21]. However, these algorithms are usually not complete, meaning that they are not always able to find a path to the goal even if one exists, due to possible local minima in the potential function. Also, the path found via gradient descent is usually sub-optimal [21], because it is difficult to include all the criteria for optimal planning in the definition of the potential function.

One of the strategies to avoid the local minima problem is the use of Harmonic Artificial Potential Field (HAPF) functions, demonstrated for the first time by Connolly et al. [22]. Harmoinc functions are those obtained as a solution to Laplace's equation:

$$\nabla^2 \phi = 0 \tag{1.1}$$

where ϕ is the potential function and ∇^2 is the Laplacian operator, which corresponds to the divergence of the gradient of the function. This elliptic Partial Differential Equation (PDE) is important in multiple branches of physics, as its solution describes potential flows, stady-state heat conduction and other phenomena. To apply these functions to the APF path planning, it is necessary to make a physical analogy to one of these phenomena, for example to potential flow. If a mass source and a mass sink are located at the start and goal positions respectively, the resulting potential field will generate many streamlines flowing between the two locations following the gradient descent. Each of these streamlines can be considered as a possible path for the robot. Outer boundaries, inner walls and obstacles are considered as impenetrable surfaces to which the gradient of the potential is parallel.

The purpose of this section of the thesis is to introduce a novel approach to HAPF with the following objectives:

- exploit the properties of APF for local planning;
- exploit the completeness of harmonic potentials for global planning;
- exploit the multi-path capability of HAPF to set an optimization problem and limit the sub-optimality of APF approaches;
- consider the terrain estimation capabilities of the robot (Section 1.2.1)
- include aspects related to the dynamics of off road tracked systems, such as terramechanics, non-holonomic constraints and complex suspensions (Section 1.2.2).

1.3 Thesis outline

- Chapter 2: Literature Review, which presents a survey of related research, highlighting the novel aspects of this thesis
- Chapter 3: Materials and Methods, which presents the algorithms developed, the robotic platforms used as test beds and the general methodology for each step of this work
- Chapter 4: Results and Discussion
- Chapter 5: Conclusions

Chapter 2

Literature Review

2.1 Machine learning algorithms for terrain awareness

Solving terrain-related challenges such as soil identification is an important research area in autonomous robots, alongside trajectory planning, localization and obstacle avoidance [7]. The latest developments in terrain classification strategies show that researchers have been focusing on two main categories: visual (or exteroceptive) and visual-independent (or proprioceptive) methods. In both approaches, data collected from sensors are used to train machine or deep learning-based classifiers that enable identification of the traversed terrain. The sensors used for visual perception include RGB cameras [23, 24], RGB-D cameras [25], LiDARs [23], visual cameras [26] and monocular cameras [27]. Although visual-based approaches are more common than proprioceptive-based ones, they have limitations as well. The performance of RGB cameras is limited by difficult environmental conditions (e.g., low, or direct lighting and surface reflectivity). LiDARs struggle to capture the fine texture of objects and terrains, and they also perform poorly in compromised environment conditions (e.g., in presence of dust, hail and smog). Furthermore, vision-based rovers are not able to navigate in unfamiliar surroundings because observing distant terrain patches does not provide information about the mechanical properties that directly impact on vehicle mobility. Therefore, researchers have investigated methods that use proprioceptive sensing for terrain classification. In this case, the sensors used to perceive the incoming terrain include Inertial Measurement Unit (IMU), force-torque sensors, microphones, and wheel encoders. As an example, Hishikawa et al. [28] used microphones to support an RGB camera in dark conditions. Brooks and Iagnemma [8] measured vibrations via accelerometers, analyzed them in the frequency domain and implemented an online classifier that relies on Principal Component Analysis (PCA) for feature reduction. DuPont et al. [29] presented a method based on frequency response and vibration-based transfer function. Giguire and Dudek [30] used a tactile probe combined with accelerometers to account for inertial effects. Dutta and Dasgupta [31] pursued a low cost approach using a multi-sensor platform fitted with GPS, IMU and metal detector. A model-based observer grounded in the Cubature Kalman filter was also proposed in [32] to predict terrain deformability using vertical acceleration measurements. The above works based on visual-independent approaches represent a step forward in the direction of providing a mobile robot with information about the mechanical properties of the terrain. Although they achieved high confidence levels, little effort was spent on feature selection as a mean to reduce the computational burden of the model without penalties in performance. Ultimately, the objective of researchers that work on robot-terrain interaction is to develop an accurate algorithm that runs online while the robot is moving. This algorithm must comply with the limited resources of an autonomous vehicle in terms of processing power and memory. A reduction in the number of features used to train and test a machine learning classifier would lead to a lighter computational burden in terms of feature extraction time, testing time and memory usage. One of the contributions of this thesis is to develop a feature selection algorithm demonstrating that these benefits can be achieved without compromising the accuracy of the model. A body of research has been devoted to the feature extraction process, as the quality of the feature space directly affects the accuracy of the associated classifier. The feature extraction strategy depends on the machine learning approach chosen for terrain classification. Traditionally, for a supervised machine learning algorithm such as SVM, an extraction stage is required where features are hand-crafted by experts based on their knowledge in the specific application domain. Attempts have been made in various research fields to find effective features, for example in image-processing-related applications [33]. However, this approach is not always possible for classifiers and it is often practically difficult, for instance when the relationship between input measurements and user-defined classes is extremely complex or even completely unknown beforehand. Additionally, features that are crafted manually may be not optimal. For this reason, finding more systematic ways to get good features has drawn an increasing research interest [34]. Notable progress has been done recently to find learning techniques that allow models to learn features automatically from data with minimal manual input. Solutions using deep Neural Network (NN) have especially attracted much attention. The effectiveness of deep NNs has been demonstrated in many fields other than image classification, such as audio and natural language processing or transfer learning. The adoption of Recurrent and Convolutional NNs was discussed in [35], in the context of terrain classification using an agricultural robot equipped only with inertial and electrical current sensors. However, although the promising results, it remains challenging to evaluate the effectiveness of learned features contrasted with expert-designed ones. The complexity of this comparison resides in the difficulty of determining the descriptive power of hand-crafted features. For this reason, this thesis presents a fair comparison between hand-crafted and learned features through a rigid feature scoring and selection process. In a previous work by the research group [36], a preliminary attempt was presented to select a subset of optimal proprioceptive features to train an SVM-based ground classifier then tested over only two terrain types, e.g. rock and sand. In this work, several novel additions are made. First, a whole new signal engineering stage is introduced to improve the overall information content. The signal selection strategy is formalized and reflected in an explanatory block diagram. Improved robustness has been achieved by increasing the number of training repetition for each candidate feature set. Then, the importance of feature selection for terrain classification is shown by comparing a machine learning approach (SVM) with a deep Convolutional Neural Network (CNN) in terms of model complexity, computational burden, and prediction accuracy over a larger terrain set (3 types of terrain against 2 of the previous work). Finally, the system is evaluated not only in a standard generalization problem but as well as in two more challenging extrapolation contexts that are seldom described in the Literature.

2.2 Innovative suspension systems for mobile tracked robots in unstructured environments

Tracked locomotion systems can be classified in many different ways. However, considering the body architecture only, tracked robots can have a non-articulated body or an articulated body. The first category includes tracked robots in which the axes of rotation of the road wheels are fixed with respect to the robot frame. This architecture is very widespread among commercial tracked robots. An example is the Trackbot by Inspectorbots [37], a symmetric platform with tracks thicker than the robot body and without external payload that can also operate after a capsize. Another example of a commercial non-articulated tracked robot is provided in [38]. Although non-articulated robots are extremely simple mechanically, they offer poor obstacle negotiation capability.

On the other hand, tracked robots with articulated bodies can be actively articulated, if the degrees of freedom of the body are actuated or passively articulated, when the degrees of freedom of the body are passive. In general, actively articulated tracked robots are more promising in terms of mobility in unstructured environments [37]. Examples are snakelike robots [39], modular tracked robots [40], hybrid wheel-track robots [41], hybrid leg-track robots [42] and hybrid leg-wheel-track robots [43]. However, the mechanical complexity of these designs and the requirement for accurate control of the body usually lead to high manufacturing costs.

Passively articulated tracked robot represent a trade-off between the mechanical simplicity of non-articulated bodies and the mobility of actively articulated platforms. The architecture proposed in this thesis and described in Section 3.2.1 belongs to this last category.

The simplest passive architecture for tracked rovers usually presents road wheels connected to the body through torsional or linear spring-damper elements of one degree of freedom. For example, Tao et al. [44] developed a stair-climbing tracked robot, in which each side presented four fixed wheels and a suspended wheel with an elastic cushioning device. To improve mobility and obstacle avoidance capabilities, more complex passively articulated tracked designs are required. However, limited work is available in the literature. Kim et al. [45] presented a mobile robot with four independent driving tracks that are connected using two rocker links and four pitch-roll passive joints of two degrees of freedom with the objective of avoiding incomplete contact between the driving tracks and the ground. Sun and Jing [46] proposed a tracked robot with passive bio-inspired suspension, with the objective to improve loading capacity and vibration isolation performance. The bio-inspired suspension is obtained by connecting each loading wheel to an X-shaped structure with springs installed horizontally.

The novelty of the architecture proposed in this thesis compared to other existing passively articulated robots is the use of an articulated suspension system, in which each road wheel is suspended independently to adapt to terrain irregularities and distribute pressure evenly in all conditions. The architecture (detailed in Section 3.2.1) provides the robot with improved mobility, with lower mechanical complexity compared to actively articulated robots. The vehicle passive suspension system performs multiple tasks such as maintaining the contact between the rubber tracks and the terrain surface, providing the vehicle stability, and protecting the vehicle frame from all the shocks generated by the terrain unevenness. This is especially important for autonomous off-road robots that carry vision sensors for localization or data acquisition. A functional comparison between Polibot and an existing robot is presented in section 4.2.3. Among the works mentioned above, the suspended tracked vehicle proposed by Sun and Jing [46] is found to be more similar and comparable with the architecture proposed in this work. On the contrary, the robot proposed in [45] is hardly comparable to Polibot, as it comprises four independent tracks, and each track is rigid, meaning that the relative position of the wheels of each track does not change.

The second contribution of this section of the thesis consists of the definition of an inverse kinematic model for the proposed architecture using a quasi-static force approach. One of the assets available to mobile robots to achieve autonomous off-road mobility is on-board navigation. An effective kinematic model is necessary to perform real-time computations for autonomous navigation. The literature presents numerous works on the subject. However, very little effort has been made to include suspension kinematics into the model, mainly via computationally expensive multi-body approaches. The reason behind this lack of tracked suspension analytical models is that, unlike wheeled robots [47], tracked rovers usually present mechanically simple suspensions, where the road wheels are fixed to the vehicle frame or connected through torsional or linear spring-damper elements. Galati and Reina [48] proposed an approach to model kinematics for tracked vehicles by obtaining a geometric analogy with a wheeled differential drive model. Sun and Jing [46] proposed a tracked mobile robot with passive bio-inspired suspension, and modeled it via a multi-body software. Ata and Salem [49] adapted the half-car model to tracked vehicles, considering only the pitch and bounce of the vehicle. Guo et al. [50] developed a universal skid steering model based on terramechanics for both wheeled and tracked mobile robots. The novelty of the proposed model in comparison with the limited related research available is that it is capable, given the shape of the supporting surface, of computing the pose of the rover and the complete configuration of its suspension system considering the constant-lentgh constraint imposed by the presence of the track.

The third contribution is the development of the Polibot MB model and its validation against the experimental prototype in both static and dynamic tests and the performance evaluation on various challenging scenarios, including positive and negative obstacles, and stochastic uneven terrain.

Taking advantage of the increasing availability of computational resources, numerical simulations have become an important tool for predicting the behavior of complex mechanical systems [51, 52]. Tracked robots are examples of flexible MB system with challenges that include the presence of many friction contacts, track modeling, and track-terrain interaction. In this research, the POLIBOT digital twin is developed within MSC Adams [15] using the toolkit Adams Tracked Vehicle (ATV) [16] that is under testing and development at the Politecnico of Bari's Robotic Mobility Lab. Following this approach, many system configurations can be evaluated from the early design phase and a wide number of operating conditions can be simulated with high fidelity without performing expensive and time-consuming field experiments.

2.3 HAPF path planning for autonomous robots

As stated in the Introduction, this thesis deals with Harmonic Artificial Potential Field (HAPF) for path planning.

Early efforts on HAPF ([22], [53], [54]) proved that these potential functions have properties that can be useful for robotics applications. They are complete, meaning that they guarantee to always provide a path to the goal. In fact, the difference to standard APF methods is that the potential field deriving from Laplace's equation does not present any local minima, but only one global minimum in correspondence of the destination. Therefore, the gradient descent cannot get stuck in any point other than the final goal. Also, paths derived from HAPF are generally smooth, meaning that they are free of sharp turns, which cause the robot to slow down. However, Laplace's equation can only be solved analytically for only a limited set of boundary conditions. In general, in path planning applications it is necessary to obtain numerical solutions, which may be computationally expensive.

To the best of the author's knowledge, no research work in the literature has applied HAPF path planning to off-road robotics. The interest of the author in this topic arises from the main features of these algorithms (as described in the Introduction), but also from their versatility, that allows HAPF path planning to be easily coupled with recent developments in vehicle dynamics, control [55] and autonomous terrain estimation [1].

Saudi and Sulaiman [56] applied HAPF to a simulation of path planning for indoor robots and focused on complexity reduction for the numerical solution of Laplace's equation. They proposed a faster iterative method based on the 9-Point Laplacian discretization scheme but with a four points half-sweep iteration and managed to speed up the computation of HAPF. Although complexity reduction is a fondamental aspect of the numerical solution of PDEs, the idea of re-compute the HAPF every time there is a change in the boundary conditions is impractical for outdoor dynamic environments. In the work presented in this thesis, Laplace's equation is solved only once by the global planner to define the initial optimal path. To deal with changes in the environment (such as new or dynamic obstacles), the local planner uses the more classical approach of adding a repulsive force to the potential. The reader is invited to refer to section 3.3.5 for details.

Falcó et al. [57] applied proper generalized decomposition to obtain parametric harmonic potential functions, with the main appeal being that the solution is obtained once per mission, including all the possible parameters. They assumed that all possible obstacle configurations can be modeled by a single parameter, therefore the resulting potential function models all of them. Also, they derived all of the possible paths obtained from the harmonic potential by considering various initial orientations of the robot. Although modelling all possible obstacle configurations cannot easily be applied to off-road applications, such as agriculture and planetary exploration, the idea of considering all the possible paths is an interesting aspect of their work. However, they choose the optimal path only based on the initial heading angle of the robot. In this thesis, the multi-path capability of harmonic potentials is exploited by setting a path optimization problem to limit the sub-optimality of APFs.

Rousseas et al. [58] combined HAPF and C-space search to infer to their approach the strong traits of both techniques, that are completeness and optimality respectively. However, their criterion for optimal planning is related to optimal and safe control, which might be enough for indoor applications. For unstructured off-road scenarios, other criteria should be considered for optimal planning, such as terrain characteristics and irregularities.

Based on this literature review and the objectives presented in the Introduction, the following chapter presents the materials and methods used to conduct this research.

Chapter 3

Materials and Methods

3.1 Terrain learning

The first part of this section (3.1.1) briefly presents the experimental planetary rover used for data gathering, describing the onboard sensor suite and the datasets collected during the field trials for developing terrain classification models. Then, the learning algorithms for terrain classification are presented, providing insights into the theoretical background. Finally, the signal engineering problem is addressed, including signal augmentation, feature extraction and feature selection.

3.1.1 The rover SherpaTT

The experimental test bed used in the ADE project is the SherpaTT rover (Fig. 3.1) built by the Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI) [59]. SherpaTT is a hybrid four-wheeled-leg rover, where the wheel-on-leg design constitutes an actively articulated suspension system. Flexible metal wheels provide a passive ground adaption on a small scale, while the active suspension fits the wheel positions to larger ground irregularities [60]. Each of the four legs of SherpaTT's suspension has five Degrees Of Freedom (DOF): the rotation of the whole leg about the pan axis with respect to the robot body, the two rotations of the inner and outer leg parallelograms, the steer and drive angle of the wheel. A unique feature of Sherpa is a 6-axis Force-Torque Sensor (FTS) mounted on the flange of each wheel-drive actuator, providing direct measurement of the force

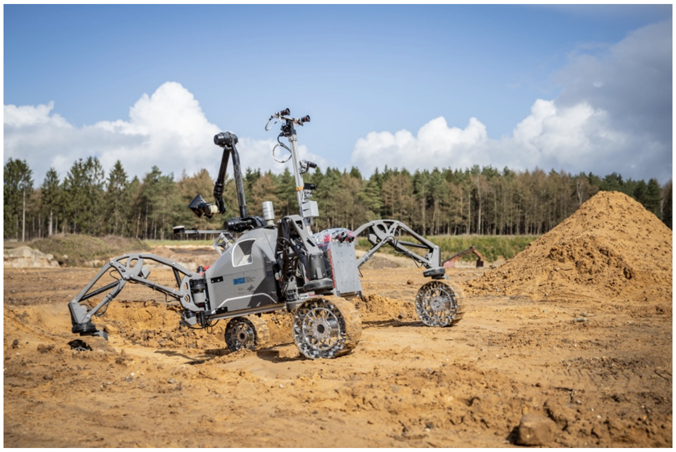


Figure 3.1: SherpaTT in a sandy trench during the ADE final field tests in spring 2021.

system exchanged with the ground. The rover also features a six-DOF manipulation arm. The arm is designed to withstand a good portion of the rover's weight to support it during locomotion. However, for the experiments described in this thesis, the arm was not involved in locomotion testing.

The logging system provides data at a rate of 100 Hz and comprises the following main proprioceptive blocks:

- Inertial Measurement Unit (IMU).
- Wheel-mounted 6-axis Load Cell (LC). In this study, solely the LC mounted on the front left wheel is adopted.
- Joint Telemetry (JT). Each of the 20 actuated joints of the suspension system delivers telemetry such as supply voltage, supply current, temperatures, PWM duty cycle, position (relative and absolute), and velocity.

The main data set used for this work was generated at the DFKI premises in Bremen, Germany. SherpaTT was remotely controlled to move for approximately 10 m in a straight line over three types of terrain: sand, gravel, and paved ground. This represents a varied dataset with a high traction, low deformability surface (paved ground) at one end, and a surface with low traction and high deformability (sand) on the other end, with gravel in the middle of the two (Fig. 3.2). For each

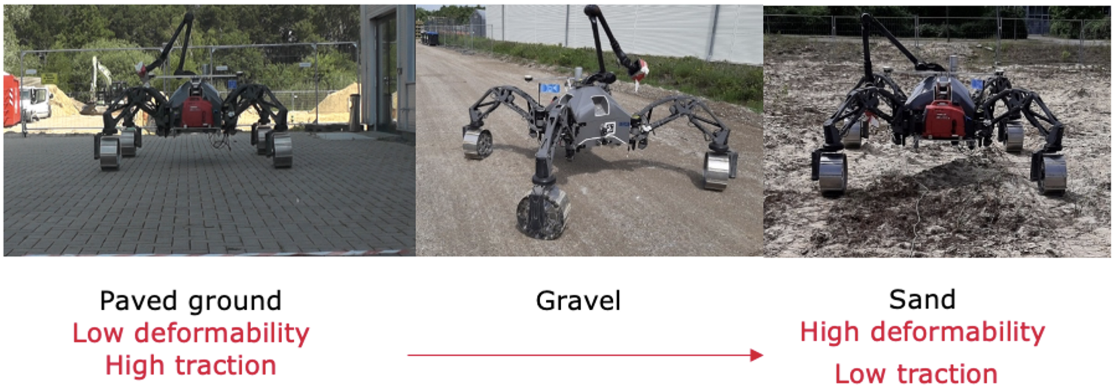


Figure 3.2: Types of surfaces traversed by SherpaTT during the test and development of the system.

terrain, five runs were repeated in forward and reverse drive, except for gravel for which only four runs are available. Two different drive speeds of the rover were used, namely 0.1 m/s and 0.15 m/s. A second data set was generated in a sand mine close to Bremen (please refer again to Fig. 3.1, GPS coordinates (DMS format): 53°18' 54.9" N, 8°41' 17.3" E) during the ADE's final testing in April 2021. This independent data set is used to predict terrain labels for observations outside the training data population. In this last environment, the surface traversed was somewhat like the sand case of the previous settings but the terrain was more compact and wetter. It can be directly observed in Fig. 3.1 how humid sand got matted to the wheels while traversing, unlike in the previous environments (Fig. 3.2).

3.1.2 Learning algorithms

3.1.2.1 Support Vector Machine

SVM is a well-established machine learning solution for soil classification problems [9, 61, 62]. This section will present a summary of the theory behind SVM classification. For a detailed description of SVM algorithm please refer to [63] and [64]. An SVM problem is composed of two stages: training and testing. Given two classes A and B (binary classifier), an input training set S composed of p samples and n features can be defined as:

$$S = \{(\mathbf{x}_i, y_i) : \mathbf{x}_i \in \mathbb{R}^n, y_i \in \{-1, 1\}, i = 1, 2, \ldots, p\} \tag{3.1}$$

with

$$\begin{cases} y_i = 1 \quad if \qquad \mathbf{x}_i \in A \\ y_i = -1 \quad if \quad \mathbf{x}_i \in B \end{cases} \tag{3.2}$$

where $\mathbf{x}_i$ are referred as predictors and y_i represents the response variable. The purpose of the linear SVM algorithm is to find a decision function D that allows, in the testing phase, to classify any new sample $\mathbf{x} \in \mathbb{R}^n$ according to the sign of $D(\mathbf{x})$. This is done by finding the hyperplane that maximizes its distance to the support vectors (i.e., the predictors closest to the hyperplane), while minimizing the loss due to misclassification. The Lagrangian dual of this optimization problem can be formulated as:

$$\max_{\alpha} \left(\sum_{i=1}^{p} \alpha_i - \sum_{i=1}^{p} \sum_{j=1}^{p} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \right) \tag{3.3}$$

subject to

$$\sum_{i=1}^{p} y_i \alpha_i = 0 \tag{3.4}$$

$$0 \leq \alpha_i \leq C \tag{3.5}$$

where α_i are Lagrangian multipliers and C is a parameter called box constraint.

The dominant approach for multi-class applications is to reduce the single problem into multiple binary classification problems [65]. One of the most common methods for such reduction is the Error-Correcting Output Codes model [66]. The most important parameter for this method is the coding design, a matrix where elements indicate which classes are trained by each binary learner, reducing the multiclass problem to a series of binary problems. In this research, SVM is considered as the benchmark approach that is compared against other alternatives as a deep CNN.

3.1.2.2 Convolutional neural network

In contrast to SVM that uses handcrafted features manually engineered by data analysts, CNN derives features automatically from inputs throughout a training process, searching for those that better characterize each terrain. However, as input, CNN takes an image-like observation, therefore a first practical issue to solve is how to derive a 3D object from several signals. One possible solution, proposed in this research, is to resort to Fast Fourier Transform (FFT) to construct magnitude spectrograms of the signals then appended into a multichannel object forming the input for the net. So, sensory data can be assembled in 3D shape, namely height, width and depth. The height corresponds to the frequencies (nF) analyzed by the FFT, the width corresponds to the number of time windows (nW) adopted in the spectrogram, and the depth is the number of signals (nCh).

The architecture of the CNN is shown in Fig. 3.3 where the neural dimensions and the learnable variables of each layer are indicated. The first layer takes as input the multichannel spectrogram, next, the batch-normalization layer normalizes inside the mini-batch the value kept by each input neuron. The normalization process follows equation 3.6, where x_n and y_n are respectively the input and output values of neuron n of this layer, batch mean μ_B and standard deviation σ_B are computed during training, while learnable parameters offset γ and bias β are searched through optimization across the whole training set. Computational constant ϵ can improve numerical stability when variance σ_B^2 is small.

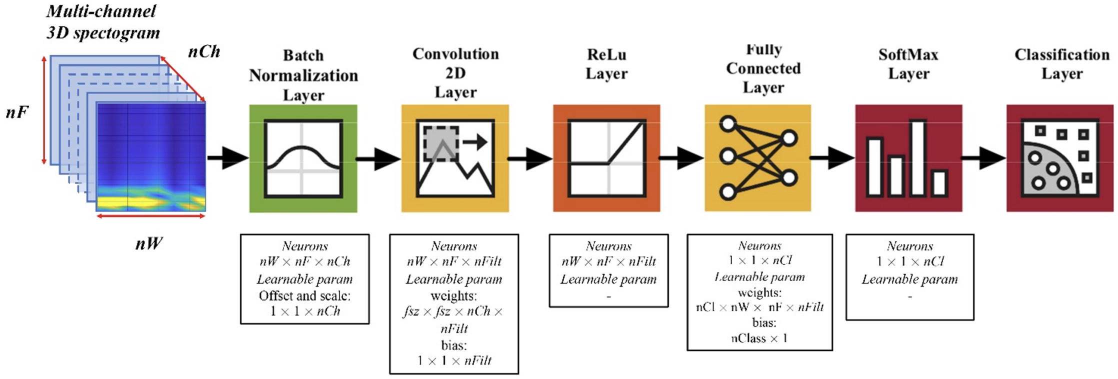


Figure 3.3: Architecture of the convolutional neural network.

$$y_n = \gamma \frac{x_n - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta \quad \forall n = 1, \ldots, (nW \cdot nF \cdot nCh) \tag{3.6}$$

The following 2D convolution layer spans the output across time and frequency domain convoluting the $nW \times nF \times nFilt$ batch-normalized spectrograms into $nFilt$ objects of dimensions $nW \times nF$. A user-specified number of square filters $nFilt$ with size fsz are here used to perform convolution process briefly described in equation 3.7 where X is the zero-padded neural grid after batch-normalization and Y the output of the convolution process. The learnable parameters of this layer are the weights of matrix ${}_m\omega$, kernel of filter m, and ${}_m\beta$ the corresponding bias.

$$Y_{w,f,m} = \sum_{c=1}^{nCh} \sum_{i,j=-\frac{fsz}{2}}^{\frac{fsz}{2}} {}_m\omega_{i,j,c} \cdot X_{w+i,f+j,c} + {}_m\beta$$

$$\forall w = 1, \ldots, nW \quad \forall f = 1, \ldots, nF \quad \forall m = 1, \ldots, nFilt \tag{3.7}$$

The output of the convolution process is passed to the REctified Linear Unit (RELU) activated neurons in a grid $nW \times nF \times nFilt$, fully connected to nCl neurons where nCl is the number of terrain classes considered. Compared to other activation functions such as the sigmoidal function, RELU helps in preventing the exponential growth in the neural network computation and the "vanishing gradient" problem that is the tendency for the gradient of a neuron to approach zero for high values of the input [67]. The two following layers SoftMax and Classification are standard as output layers for classification networks. The function SoftMax is defined in equation 3.8 where x_n is the n^{th} input neuron and y_n is the corresponding output of this layer.

$$y_n = \frac{\exp(x_n)}{\sum_{i=1}^{nCl} \exp(x_i)} \quad \forall n = 1, \ldots, nCl \tag{3.8}$$

The output layer of this network is the classification layer that computes the cross-entropy loss for classification among terrains.

3.1.2.3 Parameters of the learning algorithms

In this section, the values assigned to the parameters of the learning algorithms are highlighted.

The parameter set of the SVM-based classifier is indicated in Table 3.1. It was found empirically to give the best balance of sensitivity and specificity [68].

Parameter	Value
C (Box Constraint)	1
Standardize	True
Coding design	one-versus-one

Table 3.1: Parameters of the SVM classifier.

As for CNN, during the training stage the learnable parameters are updated at each iteration, whereas the hyper-parameters are defined by the user to govern the training process. In one iteration, the network analyses the samples contained in the mini-batch. One epoch consists in the number of iterations necessary to review the entire training dataset. The training stage stops after the network has passed through the entire dataset the number of times specified as the maximum number of epochs. It is usually preferred to stop the training before this number has been reached, not only because it shortens the time required for training, but also because it prevents overfitting on the training set. Therefore, a percentage of the training data is kept apart as validation set, and the network evaluates its loss after the number of iterations specified as validation frequency. The validation patience is the number of times that this loss can be smaller or equal to the previously smallest loss before the training stage stops. The initial learning rate drops by a factor (learn drop factor) after a given number of iterations (learn drop period). Part of the hyperparameters is set according to the Literature, e.g. the solver and the gradient threshold follows the value suggested in [67]. The remaining parameters have been selected empirically through grid-search and they are reported in Table 3.2. Note that for a fair comparison with SVM, the magnitude spectrograms of the signals used as input to CNN are obtained from a time window $ws = 2s$ (please refer to Section 3.1.3.2).

Parameter	Value
Filter size (fsz)	[5, 5]
Number of filters ($nFilt$)	9
Mini-batch size	160
Maximum number of epochs	150
Validation percentage	15%
Validation frequency	20
Validation patience	15
Initial learning rate	0.005
Learn drop factor	0.2
Learn drop period	10

Table 3.2: Hyper-parameters of the CNN classifier.

3.1.3 Signal engineering

A list of measurements available from the SherpaTT's sensor suite is shown in Table 3.3, with corresponding sensorial group and Signal ID. From a first analysis of Table 3.3, some of the signals may appear seemingly correlated. However, if we consider, for example, body acceleration and wheel force, these signals are actually uncorrelated through the flexibility of the suspension system, and therefore they are both relevant for the proposed analysis. Signals that are directly derived from measurements are referred to as direct signals. Conversely, signals engineered with expert knowledge combining direct signals are referred to as indirect, as explained in the next section. Figure 3.4 shows a sample time history of the vertical acceleration (gravity-compensated) and drive torque experienced by SherpaTT on different terrains. As seen from this figure, signals show a signature that seems to change according to the specific surface. The goal of this research is to learn this signature to gain terrain awareness. To this aim, it is necessary to select the most relevant signals for building an accurate predictor.

3.1.3.1 Signal augmentation

To improve the information content, an augmentation engine combines multiple direct measurements based on our understanding of the physical mechanisms underlying the wheel-terrain interaction. These are few of the many possible signal combinations that can be implemented, and they are chosen following a

Signal	Symbol	Sensor	Signal ID
Longitudinal Force	F_x	LC	S1
Vertical Force	F_z	LC	S2
Drive Torque	T_d	LC	S3
Drive electrical current	C_d	JT	S4
Drive PWM duty cycle	PWM_d	JT	S5
Longitudinal acceleration	a_x	IMU	S6
Lateral acceleration	a_y	IMU	S7
Vertical acceleration	a_z	IMU	S8
Gyro roll rate	$gyro_x$	IMU	S9
Gyro pitch rate	$gyro_y$	IMU	S10
Gyro yaw rate	$gyro_z$	IMU	S11

Table 3.3: List of available proprioceptive signals.

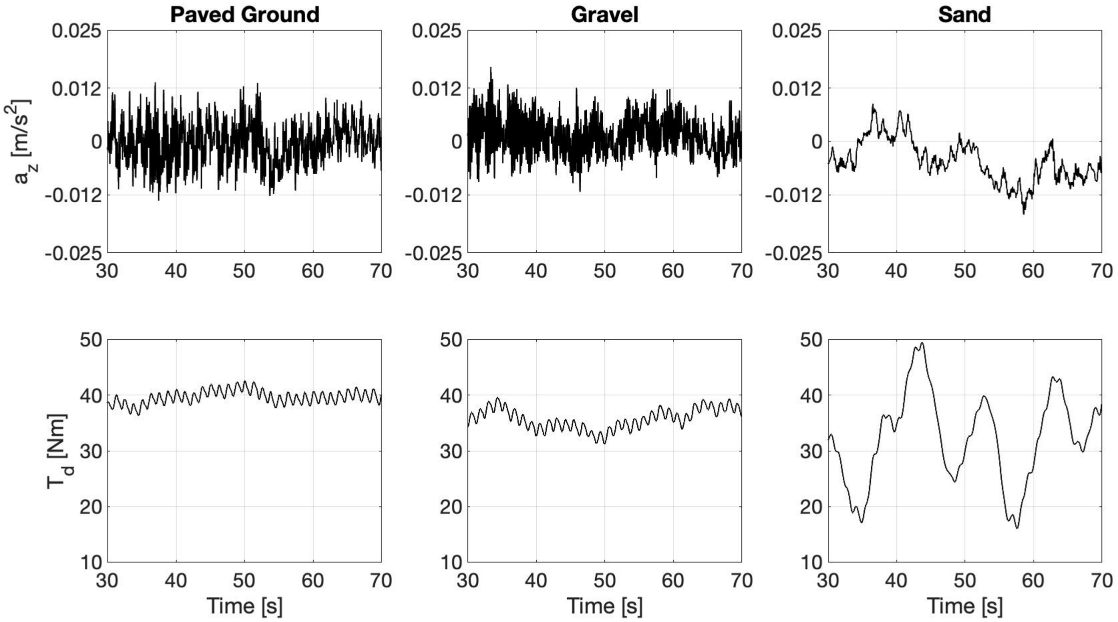


Figure 3.4: Vertical acceleration and drive torque (wheel front left) measured while SherpaTT driving straight on different terrains.

trial-and-error approach to provide the best performance over other alternatives. In this way, nine more indirect signals can be obtained (Table 3.4). The derivation of these signals is detailed in this section, and the rationale behind the choice of these entities is also explained. Two main motivations support the proposed augmentation stage. First, two or more signals that are not relevant by themselves can be useful when combined. Then, noise reduction and consequently better class separation may be achieved by adding variables that are seemingly redundant [10]. This explains why we resort to indirect or combined signals and include redundant measurements of the same physical quantity.

Signal	Symbol	Sensors	Signal ID
Mechanical Power	P_M	LC, JT	S12
Electrical Power	P_E	JT	S13
Vertical force offset	dx	LC	S14
Friction coefficient 1	μ_1	LC	S15
Friction coefficient 2	μ_2	LC	S16
Friction coefficient 3	μ_3	LC, JT	S17
Speed deviation	SD	JT	S18
Normalised speed deviation	SD_n	JT	S19
Sinkage	z	LC	S20

Table 3.4: List of indirect signals.

The first indirect signal is the power loss due to the wheel traction on given terrain. It can be derived from a “mechanical” or “electrical” analysis. The mechanical power can be estimated as follows:

$$P_M = T_d \cdot \omega \tag{3.9}$$

where ω is the rotational speed of the wheel. Conversely, the electrical power consumption can be obtained as:

$$P_E = \eta \cdot V_d \cdot PWM_d \cdot C_d \tag{3.10}$$

where V_d is the drive voltage, C_d is the wheel drive current, PWM_d is the duty cycle of the wheel drive Pulse Width Modulation, and η is the efficiency of the electric motor, assumed to be constant and approximately equal to 0.85.

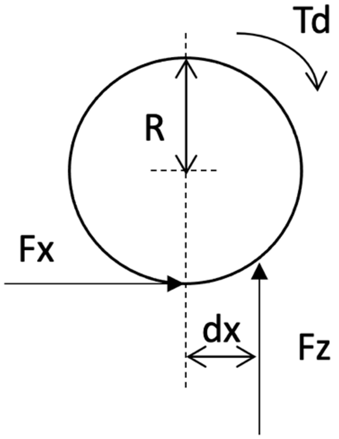


Figure 3.5: Definition of vertical force offset (dx).

Due to the rolling resistance, the direction of the resultant vertical force F_z might not pass through the centre of the wheel, with an offset in the direction of the movement (Fig. 3.5). This is especially true for soft terrain where the impact of rolling resistance is larger. Therefore, we can define the vertical force offset dx from the equilibrium of moments around the centre of the wheel, neglecting the contribution of rotational inertia:

$$dx = \frac{T_d - F_x \cdot R}{F_z} \tag{3.11}$$

where R is the loaded wheel radius defined as:

$$R = R_N - \frac{F_z}{k_z} \tag{3.12}$$

being R_N (=200 mm) the nominal wheel radius, and k_z the vertical stiffness of the SherpaTT wheel that was experimentally estimated as 69 N/mm.

The friction coefficient is an important entity related with the traction ability over the traversed surface. In this work, it is estimated in three different ways:

$$\mu_1 = \frac{F_x}{F_z} \tag{3.13}$$

$$\mu_2 = \frac{T_x}{F_z \cdot R} \tag{3.14}$$

$$\mu_3 = \frac{C_d \cdot k_T}{F_z \cdot R} \tag{3.15}$$

where k_T (17.4 Nm/A in this case) is the scale factor taking into account the torque constant of the electric motor and the transmission ratio of the motor reducer.

Speed deviation is the difference between the angular speed of each wheel ω and the average angular speed of the four wheels $\overline{\omega}$. In this work, speed deviation was estimated in two ways:

$$SD = |\omega - \overline{\omega}| \tag{3.16}$$

$$SD_{normalized} = \frac{\omega - \overline{\omega}}{\overline{\omega}} \tag{3.17}$$

Wheel sinkage is another critical parameter related to rough terrain mobility that can be approximated as suggested in [69]:

$$z = R\left(1 - \cos\left(\frac{2 \cdot dx}{R}\right)\right) \tag{3.18}$$

One important aspect is the general data consistency. As an example, Fig. 3.6 shows the drive torque delivered by the left wheel drive motor, measured by three different sensors. Direct torque measurement from the wheel-mounted LC is denoted by a solid grey line, whereas indirect estimation via the associated electric current drawn by the motor is marked by a black solid line. Finally, an alternative indirect measurement via the LC-derived longitudinal force is also plotted using a dashed black line. As seen in this figure, all three measurements show a good agreement. Similar results were observed on different surfaces.

3.1.3.2 Feature extraction

First, each sensory signal is divided in time windows, and then, for each window features are extracted as the four main statistical moments. The size of the window, ws is a design parameter. It is set as $ws = 2s$ corresponding to a traversed terrain patch of about 20 cm (comparable with the wheel radius) at an average travel speed of 0.1 m/s. In previous works by the resarch group [35], it was found that this value of window size represents a good trade-off between informative content and spatial resolution.

The four statistical moments are mean E, variance σ, skewness Sk and kurtosis

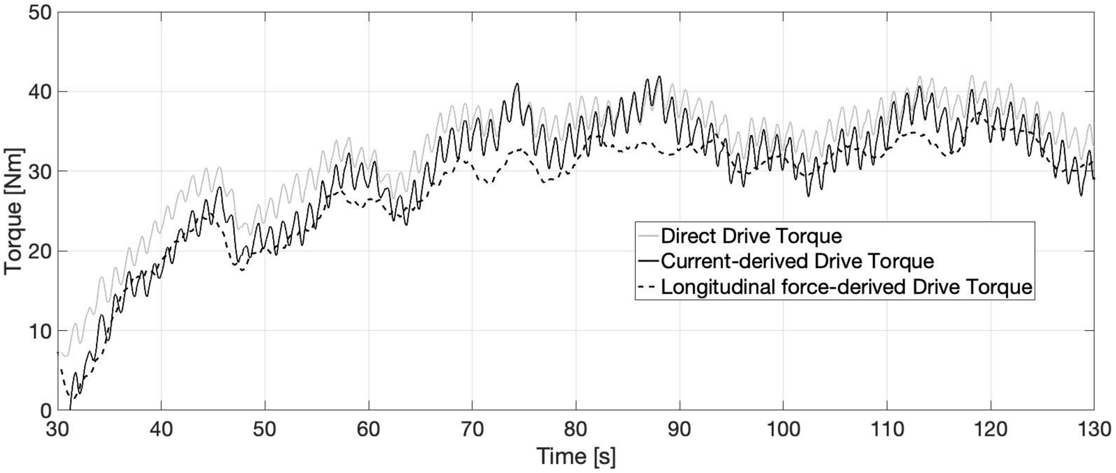


Figure 3.6: Torque applied by the left front wheel of SherpaTT as obtained from: direct measurement of the load cell (solid grey line), indirect measurement via the electric current drawn by the motor (solid black line), or alternatively via the longitudinal force provided by the load cell (dashed black line).

Ku and are defined as follows:

$$E_i = \frac{1}{N}\sum_{n=1}^{N} x_n \tag{3.19}$$

$$\sigma_i^2 = \frac{1}{N}\sum_{n=1}^{N} \left(x_n - E_i\right)^2 \tag{3.20}$$

$$Sk_i = \frac{1}{N}\frac{\sum_{n=1}^{N} \left(x_n - E_i\right)^3}{\left(\sqrt{\sigma_i^2}\right)^3} \tag{3.21}$$

$$Ku_i = \frac{1}{N}\frac{\sum_{n=1}^{N} \left(x_n - E_i\right)^4}{\left(\sqrt{\sigma_i^2}\right)^4} \tag{3.22}$$

where x_n is the value of the signal at the n^{th} time step and N is the total number of time-steps for the i^{th} window. The extraction of the statistical features brings the size of the SVM-feature space to 80 (20 signals multiplied by their 4 statistical moments). The generic feature will be indicated as $SiMj$, where i represents the signal ID ($i = 1, \ldots, 20$), whereas j represents the statistical moment ($j = 1, \ldots, 4$).

3.1.3.3 Feature selection

Retaining only the features with the highest information content reduces the computational cost while preserving the accuracy of the model. The selection process can be performed via feature scoring using appropriate validity indices. Then, an iterative search algorithm can be followed to select a reduced best feature space.

A validity index can be assigned to each feature. This index represents a measure of the information content of the feature. In this work, two validity indices are considered: the Pearson Coefficient (PC) [63], and the WB index [70]. The PC index can be computed through linear regression of a feature against the 3 classes of terrain, e.g., sand, gravel, and paved ground. The higher the PC, the larger the information content of the feature. Although this index can be successfully used for 2-class classification problems [36], it might be difficult to implement it for multi-class cases like the one presented in this work, because the number assigned to each type of terrain is arbitrary. To overcome this issue, first, the PC index is computed for each terrain pair (e.g., sand-gravel, gravel-paved ground, and sand-paved ground), and then averaged. For example, the PC index of the feature $SiMj$ against the classes 1 and 2 (sand and gravel) can be calculated as [10]:

$$ {}_{2}^{1}PC_{SiMj} = \frac{\text{cov}\left({}_{2}^{1}F_{SiMj}, {}_{2}^{1}y\right)}{\sqrt{\text{var}\left({}_{2}^{1}F_{SiMj}\right)\text{var}\left({}_{2}^{1}y\right)}} \tag{3.23} $$

where ${}_{2}^{1}F_{SiMj}$ is a vector containing all values of the feature $SiMj$ for terrains 1 and 2, whereas ${}_{2}^{1}y$ contains class values (1 or 2) for each element of ${}_{2}^{1}F_{SiMj}$. Similarly, ${}_{3}^{2}PC_{SiMj}$ (PC index of feature $SiMj$ against the classes gravel and paved ground) and ${}_{3}^{1}PC_{SiMj}$ (PC index of feature $SiMj$ against the classes sand and paved ground) follow the same principle. The overall PC index for feature $SiMj$ can be now computed as follows:

$$ PC_{SiMj} = \frac{{}_{2}^{1}PC_{SiMj} + {}_{3}^{2}PC_{SiMj} + {}_{3}^{1}PC_{SiMj}}{3} \tag{3.24} $$

In addition, the WB index can be computed for feature $SiMj$:

$$WB_{SiMj} = m\frac{SSW_{SiMj}}{SSB_{SiMj}} \tag{3.25}$$

where SSW is the sum of square within classes and SSB is the sum of squares between classes, computed as follows:

$$SSW_{SiMj} = \sum_{k=1}^{nCl}\sum_{s=1}^{n_k}\left(x_s - \mu_k\right)^2 \tag{3.26}$$

$$SSB_{SiMj} = \sum_{k=1}^{nCl} n_k\left(\mu_k - \mu\right)^2 \tag{3.27}$$

where x_s is the s^{th} sample of feature $SiMj$, μ_k is the class k centroid value, μ is the overall dataset centroid value, n_k is the number of samples in class k and nCl (=3) is the number of classes. A low value of WB_{SiMj} indicates that classes form compact and distant clusters relatively to feature $SiMj$. Therefore, the score assigned to each feature will be WB^{-1}: the higher the WB^{-1}, the better the feature for classification purposes.

The rationale behind using two validity indices is that the WB and PC have two different statistical meanings: the former describes the compactness of classes, the latter shows the correlation between a given feature and the type of terrain. One may think that a feature with a low value of PC index will also have a relatively low value of WB^{-1} index. However, this is not always true, and exceptions do occur. For example, Fig. 3.7 shows the distribution of PC and WB indices for the 25 features with the highest scores. $S6M2$ is the feature with the third highest value of PC index, but it is only the 21^{st} feature in terms of WB^{-1}. Similarly, $S16M2$ is the feature with the second highest value of WB^{-1} index, but it is only the 14^{th} in terms of PC. This shows that the two indices rank the features in different ways, therefore they complement each other very well.

The proposed feature selection approach is based on the iterative search scheme presented in the block diagram of Fig. 3.8. The input to the algorithm is the full set of n_{feat} (=80) features. These features are then ranked using the output of one of the two validity indices (PC or WB) as a score. The best feature set is

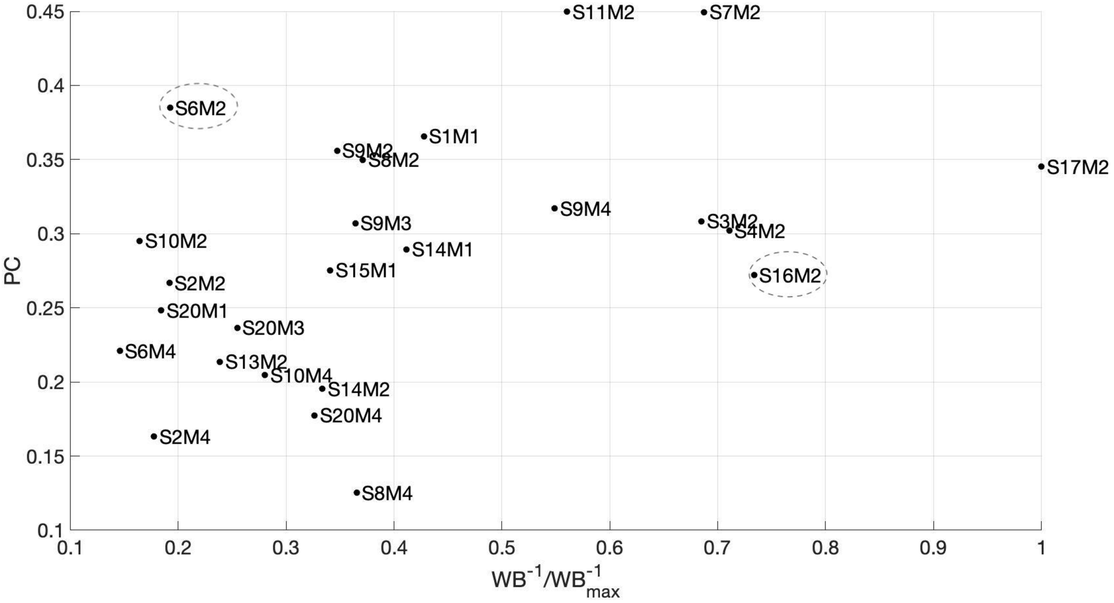


Figure 3.7: PC and WB indices distribution for the most relevant features.

initialized with the first $n_{min} - 1$ (=2) features of the ranking. At this point, the objective is to iterate on all the remaining features to find those which provide better classification performance. In each iteration, identified with the index i that varies from n_{min} to n_{feat}, the i^{th} feature in the ranking is added provisionally to the best feature set. Then, an SVM-based classifier is trained and evaluated in terms of $F1$ score via 5-fold cross validation. The k-fold cross validation process partitions data into k randomly chosen subsets (or folds) of roughly equal size. Therefore, to improve the robustness of the feature selection algorithm, the training phase is repeated n_{train} (=10) times and the final $F1$ score is computed as the average of the scores obtained at each training phase. If the final $F1$ score is sufficiently higher than the best $F1$ score obtained so far, the i^{th} feature is kept in the best feature set, and the best $F1$ score is updated. Otherwise, the i^{th} feature is discarded from the best set and not considered for training purposes.

In order to facilitate the reading of the block diagram in Fig. 3.8, the meaning and the numerical values of the parameters involved in the selection process are collected in Table 3.5.

The selection process discussed in Fig. 3.8 can be repeated for each one of the two validity indices. Eventually, two best reduced feature spaces will be obtained:

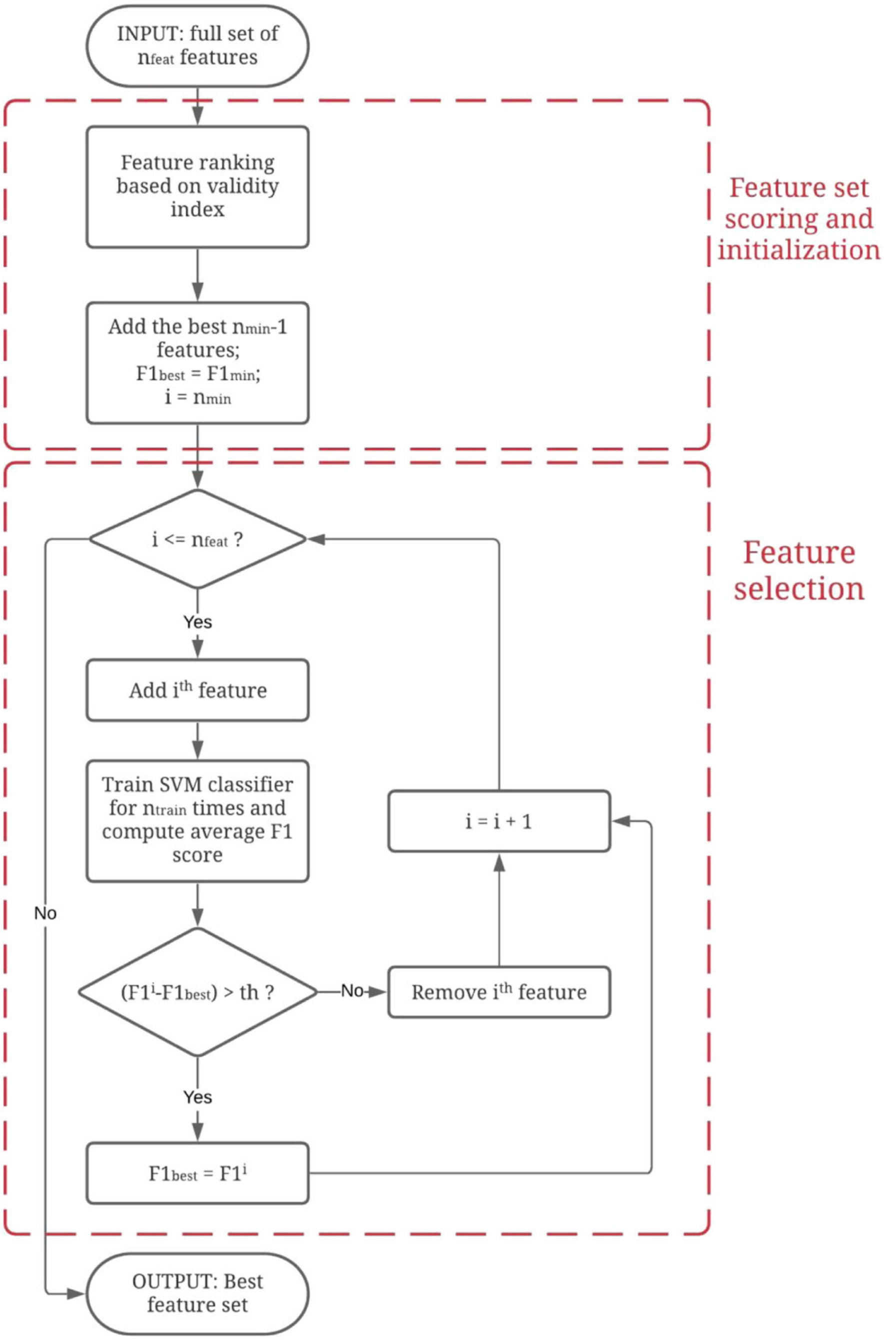

INPUT: full set of n_{feat} features
Feature ranking based on validity index
Add the best n_{min}-1 features; $F1_{best}$ = $F1_{min}$; i = n_{min}
Feature set scoring and initialization
i <= n_{feat} ?
Feature selection
Yes
Add i^{th} feature
Train SVM classifier for n_{train} times and compute average F1 score
i = i + 1
No
($F1^{i}$-$F1_{best}$) > th ?
No
Remove i^{th} feature
Yes
$F1_{best}$ = $F1^{i}$
OUTPUT: Best feature set

Figure 3.8: PC and WB indices distribution for the most relevant features.

Parameter	Description	Value
n_{min}	Minimum number of features	3
$F1_{min}$	Minimum F1 score	60%
th	Accepted improvement (threshold) in the F1 score	5%
n_{train}	Number of trainings for each new best feature set	5
n_{feat}	Number of features in the initial full feature set	80

Table 3.5: List of parameters involved in the feature selection approach.

one associated with the PC and the other with the WB index. To further improve the robustness of the selection algorithm, the union of these two sets is chosen as the best for SVM training purposes. The 18 selected features are listed in Table 3.6. It is worth noting that three features extracted from indirect signals are included as well, thus, proving the utility of the signal augmentation phase. A 3D plot of the three most relevant features in terms of WB index is shown in Fig. 3.9 to help the reader to easily visualize the result of the whole selection process. As shown in this figure, the sand data form a quite compact cluster, with relatively low values of all three features. Conversely, gravel and paved ground data show higher values of $S7M2$ (variance of a_y) than sand and mainly differentiate in values of $S17M2$ (variance of μ_3).

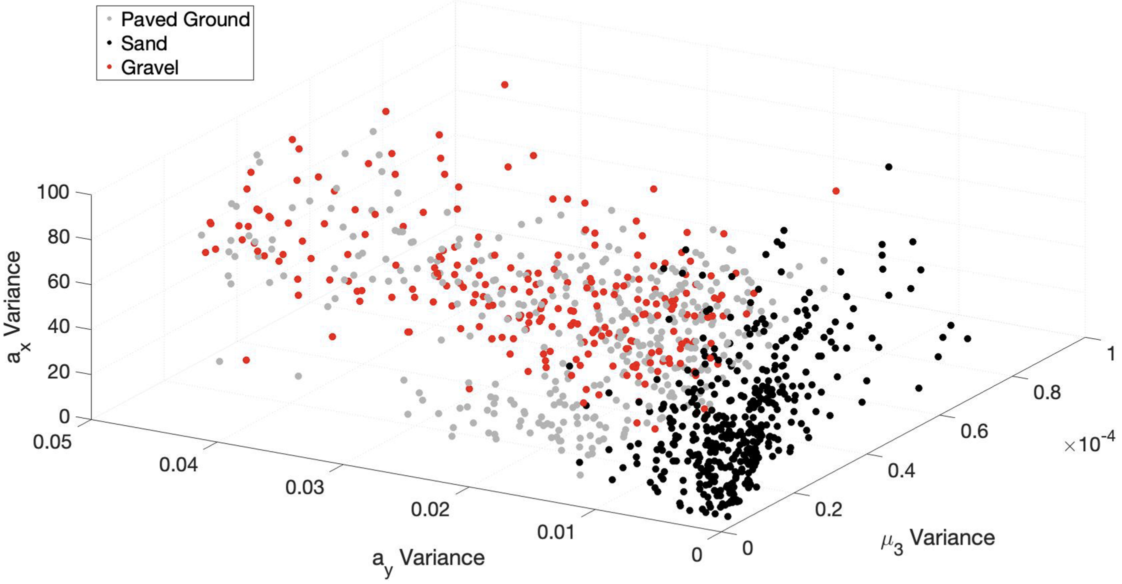


Figure 3.9: 3D plot of the first three features with the highest score of WB^{-1}.

Signal	Moment	Signal type	Feature ID	WB^{-1}/WB^{-1}_{max}	PC
μ_3	Variance	Indirect	$S17M2$	1.00	0.345
a_y	Variance	Direct	$S7M2$	0.691	0.449
T_d	Variance	Direct	$S3M2$	0.685	0.308
$gyro_z$	Variance	Direct	$S11M2$	0.561	0.450
$gyro_x$	Kurtosis	Direct	$S9M4$	0.549	0.317
F_x	Mean	Direct	$S1M1$	0.428	0.366
a_z	Variance	Direct	$S8M2$	0.371	0.350
$gyro_x$	Skewness	Direct	$S9M3$	0.364	0.307
$gyro_x$	Variance	Direct	$S9M2$	0.348	0.356
μ_1	Mean	Indirect	$S15M1$	0.342	0.275
z	Kurtosis	Indirect	$S20M4$	0.327	0.177
a_x	Variance	Direct	$S6M2$	0.192	0.385
F_z	Variance	Direct	$S2M2$	0.192	0.267
$gyro_y$	Variance	Direct	$S10M2$	0.164	0.295
F_x	Variance	Direct	$S1M2$	0.143	0.253
a_y	Kurtosis	Direct	$S7M4$	0.048	0.115
PWM_d	Mean	Direct	$S5M1$	0.013	0.062
$gyro_z$	Mean	Direct	$S11M1$	0.011	0.064

Table 3.6: Best feature set.

3.2 Analysis of an all-terrain tracked robot with innovative suspension system

3.2.1 Polibot

The fully functioning prototype of the all-terrain rover Polibot is shown in Fig. 1.1(a) along with a CAD rendering with indication of the overall dimensions (Fig. 1.1(b)). A side view of the real robot is shown in Fig. 3.10. The name derives from an acronym for Politecnico of Bari robot. It comprises a central body and two side tracks. A rubber track is wrapped around the drive sprocket (W_1), the idler wheel W_2, and the four ground wheels (W_i, $i = 3, \ldots, 6$). What is unique about this tracked locomotion system is the presence of an articulated passive suspension system on either side that allows each ground wheel to move independently with respect to the vehicle body providing remarkable adaptability to irregular terrain. Referring to the left track suspension (Fig. 3.10), a detailed numbered schematic of the system is shown in Fig. 3.11. The meaning of the numbers is explained in

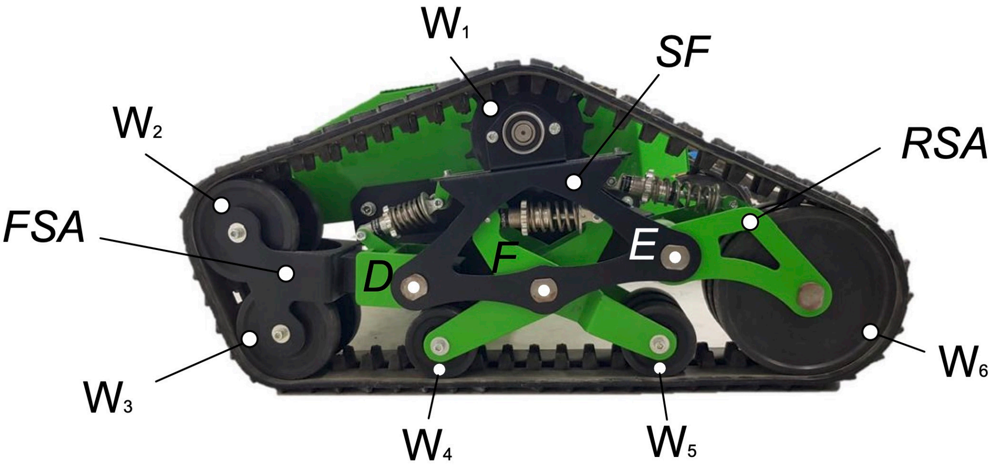


Figure 3.10: Detailed left side view of Polibot.

Table 3.7. One can note that the vehicle weight is distributed over the contact patch via the four road wheels. Each wheel is suspended with respect to the mid subframe SF, attached to the robot body and accommodating the drive sprocket W_1, using a dedicated swing arm and a spring-damper element. For example, the front (FSA) and rear (RSA) swing arm allow, respectively, W_3 and W_6 to rotate about the hinges D and E in order to conform to the given terrain geometry. In addition, the two swing arms bringing W_4 and W_5 are hinged in F to form a middle bogie-like sub-suspension. The track tension can be adjusted by controlling the length of the FSA using a male screw rod and a nut-shaped rotation element. The tension adjustment unit is not visible in Fig. 3.10. The FSA also brings the idler wheel W_2 forming an attack angle with the road wheel W_3 that facilitates the climb of obstacles.

Polibot is a ROS-based skid steer vehicle equipped with two 350W 24VDC brush motors, each of which is coupled with an angular gearbox with a ratio $\tau = 30$. The tracked undercarriage is made of two track assemblies rigidly attached to the main body frame by two brackets. All powertrain components are installed inside the body frame while the drive sprocket is directly coupled to the gearbox output shaft. The belts are made of fiberglass reinforced bands and an inner steel cord all embedded in melted natural rubber. The drive sprockets and the idle wheels are made of UHMW, which is a high density polyethylene with very high wear resistance. Each belt has a length of about 2000 mm and a width of about 127

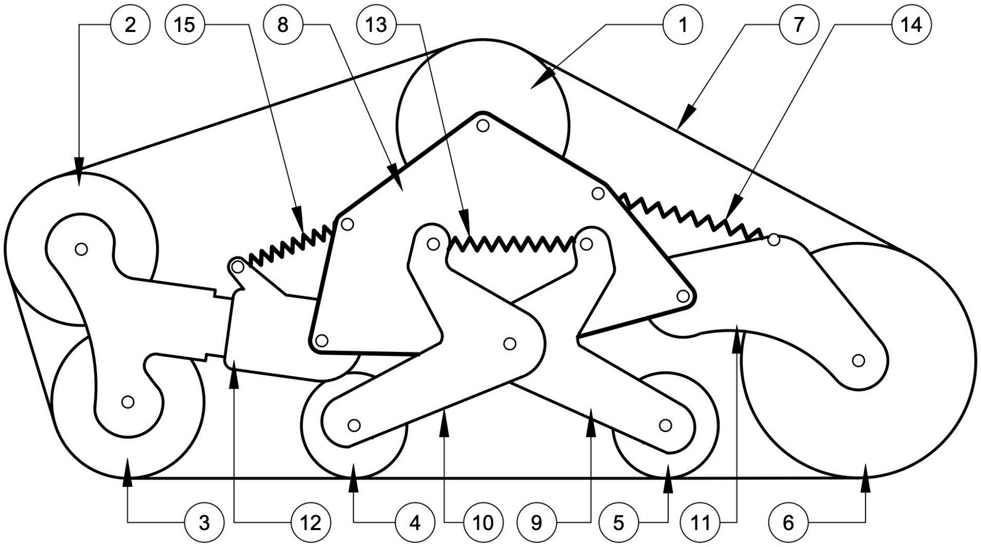


Figure 3.11: Suspension components

Item number	Description	Symbol
1	Sprocket	W_1 (wheel 1)
2	Idler wheel	W_2
3	Ground wheel	W_3
4	Ground wheel	W_4
5	Ground wheel	W_5
6	Ground wheel	W_6
7	Track	-
8	Subframe	SF
9	Bogie rear arm	BRA
10	Bogie front arm	BFA
11	Rear swing arm	RSA
12	Front swing arm	FSA
13	Spring-damper	S_1
14	Spring-damper	S_2
15	Spring-damper	S_3

Table 3.7: List of components.

mm while the height of the treads is 20 mm placed on a support base having a thickness of 8 mm. It should be noted that the tread height has been designed to be 20 mm to provide better grip on agricultural terrain and to prevent the rubber track from slipping when in contact with soft ground or unpaved surfaces. The sprockets have a total of 17 teeth and a diameter of 200 mm with a thickness of 20 mm and engage the rubber tracks around their bushings that have a pitch of 26 mm to provide traction. In the proposed design, the drive sprocket is placed on top to prevent the power transmission elements to wear out faster due to environmental contamination from dust, mud and granular terrain or get damaged as a result of collisions with rocks and obstacles. This solution partially limits the extension of the wrapping angle. However, attention has been paid that at least three teeth are simultaneously engaged, which guarantee a correct power transmission. To further mitigate system sensitivity to detracking issues, higher teeth have been cut for the sprocket that exceed the rubber belt thickness.

The electronics installed on board the robot integrate current and voltage sensors to measure the power drawn by the motors in addition to two quadrature optical encoders with 1024 pulses that are mounted on the rear shaft of both motors in order to implement a closed-loop speed control. The vehicle has been designed to provide an upper flat surface that can be used to place additional devices and sensors like laptops, LiDARS, IMUs or cameras. Polibot features an embedded industrial computer with Intel i7 CPU, 16GB RAM DDR and 256GB SSD providing wireless connectivity with Wi-Fi and Bluetooth interfaces. The main operating system installed on the computer is Ubuntu and it is used to run ROS and to generate locomotion commands over a RS232 serial port directly connected to the motors controller. The main power supply consists of a 24VDC 30Ah LiPo battery package providing 30Ah with an overall standard autonomy of about 3 hours.

The rover Polibot was designed and developed within the collaboration between the Robotic Mobility Lab of Politecnico di Bari and the start-up Robodyne. This thesis work focuses instead on the development of an analytical quasi-static model of the inverse kinematics of Polibot and on a multi-body digital twin in MSC Adams environment.

3.2.2 Polibot analytical modeling

This section presents the inverse kinematic model that involves computing the pose of the rover and the configuration of its suspension system, given the shape of the supporting surface. To this aim, a contact model must be first defined. For tracked vehicles, a contact model may be difficult to define. In first approximation, it is assumed that the track has negligible thickness and conforms perfectly to the terrain geometry. Then, each road wheel is in contact with the supporting surface at its lowest point, denoted by $CP_i, i = 3, \ldots, 6$ to match the numbering of the road wheel, as explained in Fig. 3.12. This is a reasonable assumption when moving over firm terrain. For vehicles moving on soft terrain, distributed track-terrain contact stresses should be considered and resolved to resultant forces at a single point. In the case of negotiation of a rock, the contact point depends on the obstacle geometry and can be defined referring to the wheel-terrain contact plane, as detailed, for example, in [47].

The input to the analytical model is the geometric properties of the robot and the elevation map of the terrain. The outputs are the position and tilt of the vehicle body (i.e., mid subframe SF) along with the suspension configuration.

In the remainder of this Section, the Polibot model is developed as follows: Section 3.2.2.1 deals with the degrees of freedom of the system, while Sections 3.2.2.2 and 3.2.2.3 describe the equations required to solve the inverse kinematic problem.

3.2.2.1 Degrees of freedom

By referring again to the left suspension of Polibot (Fig. 3.11), and considering only the points of the rigid bodies that interact with other components, the system can be further schematised as shown in Fig. 3.13. For clarification, the rigid bodies that compose the kinematic scheme (wheels and spring excluded) are listed in Table 3.8, and their geometry is detailed in Appendix A. It is important to highlight that the dimension d_{11} (shown in Fig. A.1 in Appendix A) can be changed by the tension adjustment unit and therefore it is considered as an input parameter.

To study the motion of the suspension system, a global reference frame XYZ is defined in Fig. 3.12. Please note that for simplicity sake, a half-symmetry model

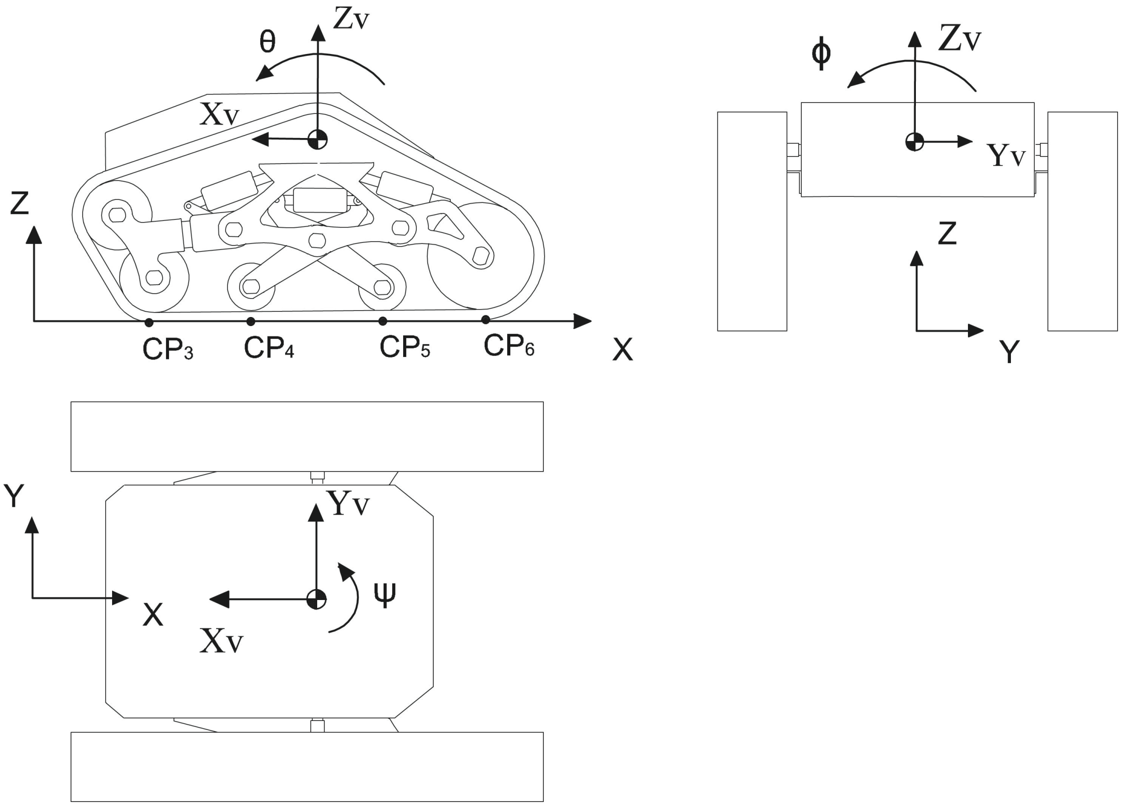


Figure 3.12: Polibot coordinate frames

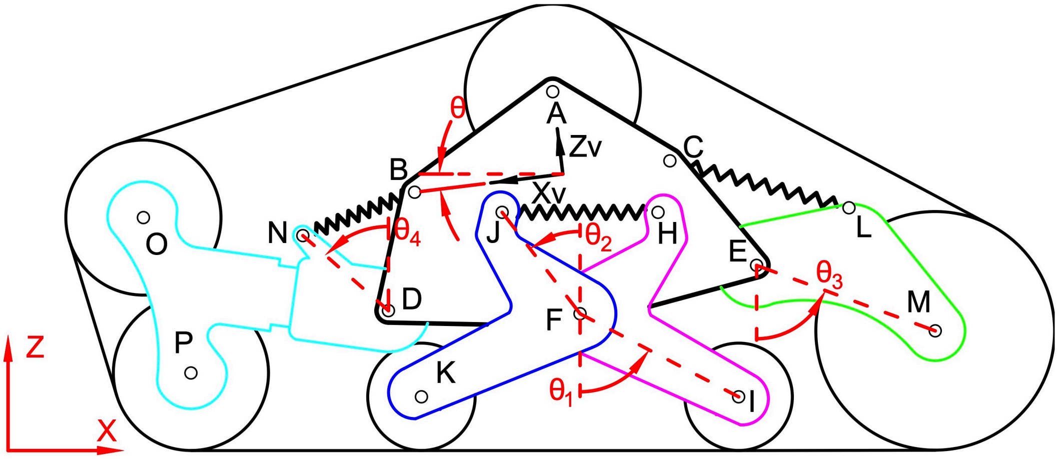


Figure 3.13: Left suspension schematics

Body number	Points	Color
1	ABDFEC	Black
2	HFI	Magenta
3	JFK	Blue
4	ELM	Green
5	DNOSP	Cyan

Table 3.8: List of rigid bodies.

is assumed, i.e., the vehicle center of mass is constrained to the X-Z plane and the chassis can only pitch (θ) whereas roll and yaw rotations (ϕ and ψ) are not taken into account.

Under this assumption, the system of Fig. 3.13 consists of five rigid bodies connected by three revolute joints (D, E, F). The seven resulting degrees of freedom are listed in Table 3.9.

3.2.2.2 Constraints

The terrain elevation map that is assumed to be known in this analysis can be represented by the following expression:

$$Z = f_{te}(X) \tag{3.28}$$

where f_{te} is the terrain elevation function, that gives the height of the terrain (Z) for any value of X. For example, the terrain elevation function for a flat horizontal

Number	Symbol	Description
1	X_A	SF vertical position
2	Z_A	SF horizontal position
3	θ	SF Pitch
4	θ_1	BRA orientation
5	θ_2	BFA orientation
6	θ_3	RSA orientation
7	θ_4	FSA orientation

Table 3.9: List of degrees of freedom as shown in Fig. 3.13.

surface is $Z = h_0$, with h_0 a constant.

Referring to Fig. 3.13, given the X-coordinate of the first contact point X_{CP_3} and assuming that wheels 3 to 6 make contact with the ground with their lowest point, the following constraint equations can be written:

$$X_P = X_{CP_3} \tag{3.29}$$

$$Z_P - r_3 = f_{te}(X_P) \tag{3.30}$$

$$Z_K - r_4 = f_{te}(X_K) \tag{3.31}$$

$$Z_I - r_5 = f_{te}(X_I) \tag{3.32}$$

$$Z_M - r_6 = f_{te}(X_M) \tag{3.33}$$

where r_i is the radius of wheel i. Given the geometry of the suspension (please refer to Fig. A.1 in Appendix A), the Z-coordinate of the wheel centers (P, K, I and M) can be expressed as a function of the degrees of freedom of Table 3.9 via loop closure equations. For conciseness, these expressions are omitted here and reported in Appendix B.1.

Another constraint equation is derived from the adoption of a steel-core rubber track, whose length can be considered reasonably constant, thus imposing the following geometric condition that must be satisfied during the relative motion of the suspension elements:

$$L_{track} = L_{nom} \tag{3.34}$$

where L_{nom} is the nominal length of the track. The derivation of L_{track} as a function of the degrees of freedom is reported in Appendix B.3.

The system of non linear equations (3.29)-(3.34) consist of six equations in seven unknowns, which are the seven degrees of freedom. Thus, at this stage the inverse kinematic problem is under constrained and there exists an infinite set of solutions. For this reason, it is necessary to consider the quasi-static equilibrium equations, as explained in the next section.

3.2.2.3 Quasi-static analysis

If a vehicle has more than two contact points as Polibot, it will be dynamically indeterminate and the normal forces under the wheels cannot be obtained by global equilibrium equations. For these two reasons it is necessary to consider the elastic element deflection. To do so, the equilibrium of moments acting on the BRA, BFA, RSA and FSA around joints D, F and E should be added to the problem. The rover weight and the position of the center of gravity in the vehicle coordinate frame (Xv_G,Zv_G) have been experimentally estimated with the use of scales (Refer to section 4.2.4.1).

The robot is assumed to move at a constant speed, therefore the only external forces acting in the longitudinal direction are the tangential forces under each wheel (F_i, for $i = 3, \ldots, 6$) and the motion resistance of the running gear (F_v), as indicated in Fig. 3.14. The motion resistance of the running gear accounts for obstacle resistance, internal resistance of the running gear and the resistance due to interaction with the terrain. Figures 3.15 and 3.16 show the internal forces acting on Polibot suspension system.

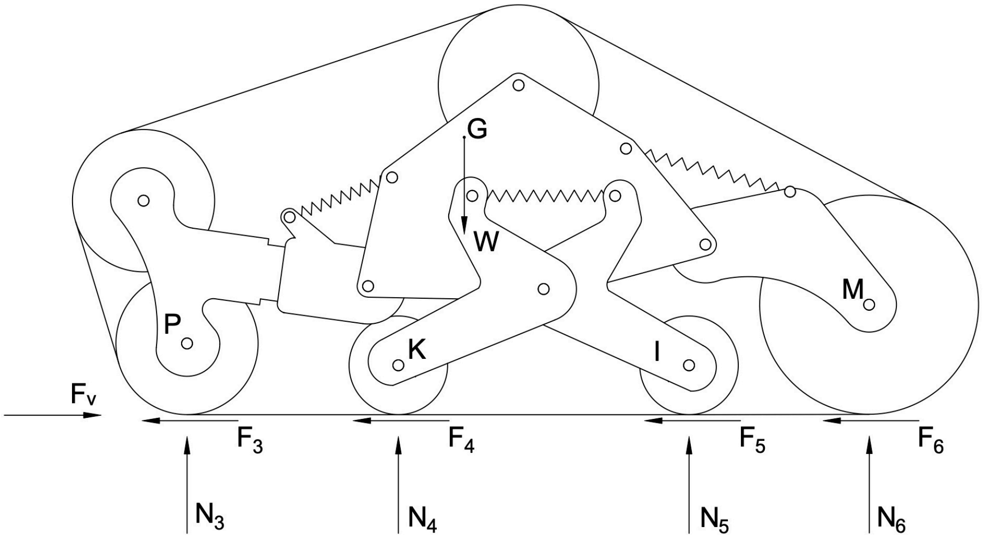


Figure 3.14: Global free body diagram of Polibot.

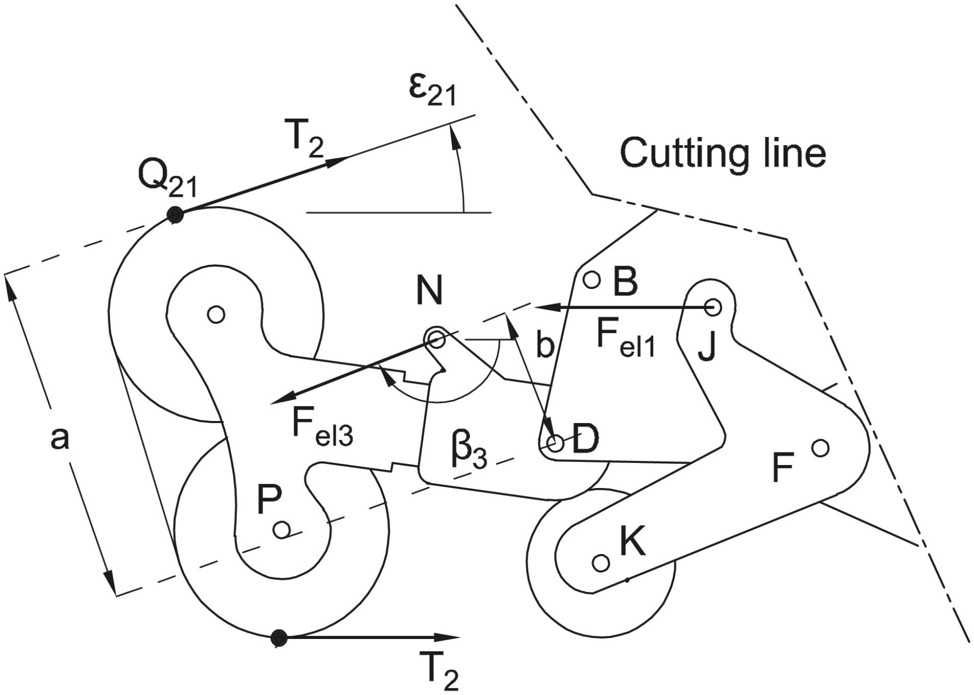


Figure 3.15: Internal forces acting on Polibot suspension system: left half.

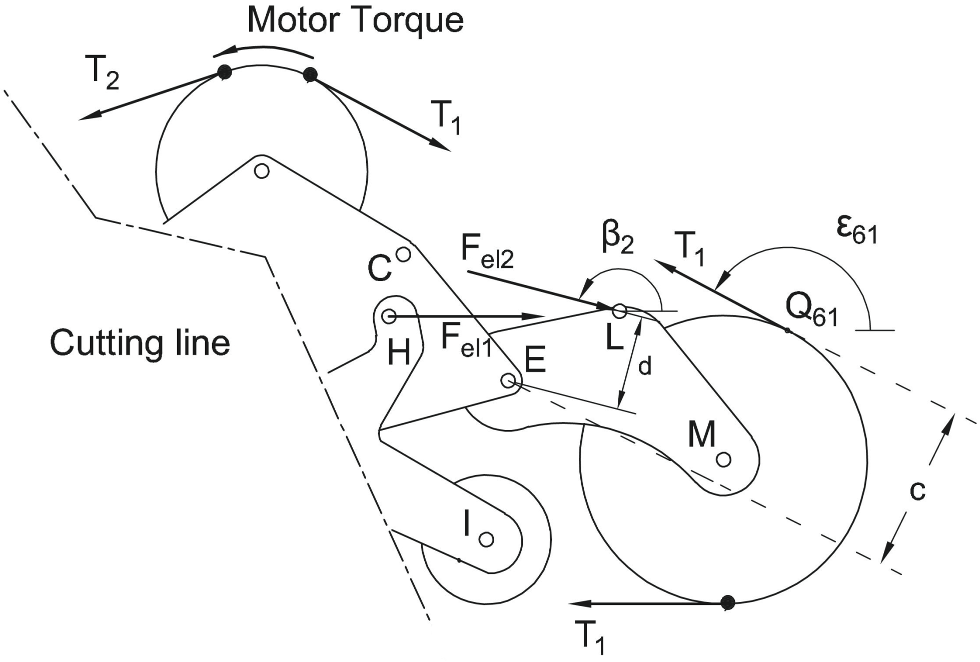


Figure 3.16: Internal forces acting on Polibot suspension system: right half.

The required equilibrium equations can be easily derived from Figures 3.14, 3.15 and 3.16 and they can be written in matrix form as follows:

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & CP_3^x & 0 & CP_4^x & 0 & CP_5^x & 0 & CP_6^x \\ 0 & CP_3^x - X_D & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & CP_4^x - X_F & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & CP_5^x - X_F & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & CP_6^x - X_E \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & -k_c & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -k_c & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -k_c & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -k_c \end{bmatrix} \cdot \begin{bmatrix} F_3 \\ N_3 \\ F_4 \\ N_4 \\ F_5 \\ N_5 \\ F_6 \\ N_6 \end{bmatrix} +$$

$$+ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & CP_3^z - Z_D - a & 0 & 0 & b \\ 0 & 0 & Z_J - Z_F & 0 & 0 \\ 0 & 0 & Z_F - Z_H & 0 & 0 \\ Z_E - CP_6^z + c & 0 & 0 & d & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} T_1 \\ T_2 \\ F_{el,1} \\ F_{el,2} \\ F_{el,3} \end{bmatrix} = \begin{bmatrix} F_v \\ W \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.35)$$

which in compact form can be written as:

$$G_c \cdot f_c + G_{int} \cdot f_{int} = f_s \quad (3.36)$$

where f_c includes all the forces arising from wheel-ground interaction (i.e. normal and tangential forces), f_{int} represents the internal forces (i.e. track tension and elastic forces), f_s includes the summed effects of gravitational forces, inertial forces, and forces due to interaction with the environment. The geometric parameters a, b, c and d are indicated in Figures 3.15 and 3.16. The system (3.36) consists in a set of 12 equations in 14 unknowns. The twelve equations include, in order:

- global force balance along X
- global force balance along Z
- global torque balance around the center of gravity
- torque balance on BFA around point F
- torque balance on BRA around point F
- torque balance on RSA around point E
- torque balance on FSA around point D
- longitudinal force balance of the track element that exchanges forces with the ground
- proportionality between F_i and N_i, one equation per wheel (i.e. for $i = 3, \ldots, 6$)

The last four equations (proportionality between F_i and N_i) are required to distribute the tractive effort to each wheel proportionally to the terrain normal reaction.

The 14 unknown parameters are described in Table 3.10, and they also include the proportionality constant between tangential and normal wheel-terrain forces k_c defined as follows:

$$k_c = \frac{F_i}{N_i}, \quad for \quad i = 3, \ldots, 6 \tag{3.37}$$

To close the system of equations, it is necessary to consider the deflection of the elastic elements. Each spring has a pre-load and behaves as a rigid body if any force lower than the pre-load is applied. For example, the force-length characteristic of a spring with stiffness 12 N/mm, pre-load 50 N and maximum length 150 mm is shown

Unknown force	Description
T_1	Tension of the track, tight side
T_2	Tension of the track, slack side
F_3	Tangential traction force for W_3
F_4	Tangential traction force for W_4
F_5	Tangential traction force for W_5
F_6	Tangential traction force for W_6
N_3	Normal reaction force for W_3
N_4	Normal reaction force for W_4
N_5	Normal reaction force for W_5
N_6	Normal reaction force for W_6
k_c	Proportionality constant between F_i and N_i
$F_{el,1}$	Force applied to S_1
$F_{el,2}$	Force applied to S_2
$F_{el,3}$	Force applied to S_3

Table 3.10: List of force parameters.

in Fig. 3.17, where the force applied to the spring is considered in compression. If a traction force is applied to the spring, the elastic element behaves as a rigid body and the length is equal to its maximum length. This characteristic can be generalised and applied to the three elastic elements of the suspension:

$$L_i = \begin{cases} L_{max,i}, & \text{if } F_{el,i} < F_{pre,i} \\ L_{max,i} - \frac{F_{el,i} - F_{pre,i}}{k}, & \text{otherwise} \end{cases} \qquad i = 1,2,3 \tag{3.38}$$

where L_i is the length of spring i when a force $F_{el,i}$ is applied to its ends, $F_{pre,i}$ and $L_{max,i}$ are the pre-load and the maximum length of spring i respectively, k is the elastic stiffness. The length of the elastic elements (L_i for $i = 1,2,3$) is derived as a function of the degrees of freedom in B.2. Pre-loads and maximum lengths are input parameters.

The system of equations (3.36) and (3.38) consists of 15 equations in 14 unknowns. However, it cannot be solved independently of the constraint equations shown in section 3.2.2.2. In fact, the matrices G_c and G_{int} are functions of the seven degrees of freedom of the system, as shown in C. For these reasons, the force analysis conducted in this section must be coupled with the constraint equations 3.29-3.34, resulting in a system of 21 equations in 21 unknowns. The unknowns are

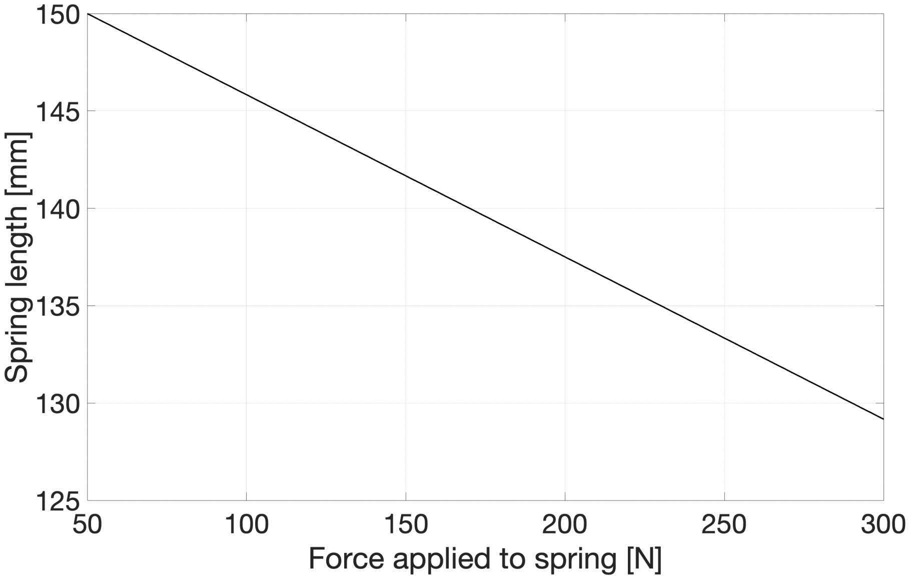


Figure 3.17: Force-length characteristic of a spring with 12 N/mm stiffness, 50 N pre-load, and maximum length of 150 mm.

the seven degrees of freedom of Table 3.9 plus the fourteen unknown parameters of Table 3.10.

3.2.3 Polibot multibody model

3.2.3.1 Multibody modeling

In this work, the robot Polibot is considered as a MB system. The position problem corresponds to solving the constraint equations:

$$\phi(q,t) = 0 \tag{3.39}$$

where q is the vector of generalised system coordinates and t is the time. By differentiating Equation 3.39, it is possible to obtain the velocity kinematic constraint:

$$\phi_q \dot{q} = 0 \tag{3.40}$$

where ϕ_q is the Jacobian matrix of the constraint equations with respect to q, and $\dot{q}$ is the vector of generalised velocities. By taking the first derivative again, the acceleration kinematic equations are obtained:

$$\phi_q \ddot{q} = -\dot{\phi_q}\dot{q} = c \tag{3.41}$$

where $\ddot{q}$ is the vector of generalised accelerations. For a consistent time evolution of the modeled system, the genaralised coordinates q must satisfy the kinematic Equations 3.39 to 3.41.

Dynamics must be included in the model by applying Newton's law and by taking into account constraints, leading to the following set of differential equations:

$$M\ddot{q} + \phi_q^T \lambda = Q \tag{3.42}$$

where M is the mass matrix, Q is the vector of external forces and λ is the vector of Lagrangian multipliers. The objective of a multibody simulation software is to solve Equations 3.41 and 3.42 simultaneously. If n and m refer, respectively, to the dimension of the generalised coordinates and constraint equations, Equation 3.42 represents a set of n equations in $n+m$ variables. By adding the m Equations 3.41, the following system can be obtained and solved for simultaneous solution of the accelerations and Lagrange multipliers [71]:

$$\begin{bmatrix} M & {\phi_q}^\top \\ \phi_q & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \lambda \end{bmatrix} = \begin{bmatrix} Q \\ c \end{bmatrix} \tag{3.43}$$

3.2.3.2 POLIBOT digital twin

In this work, the multibody model of POLIBOT was built with a methodology based on templates, which is the standard design methodology of MSC Adams/ATV: the vehicle is considered as a set of interacting subsystems, and each subsystem is modeled independently and later integrated to form the assembly to simulate. Figure 3.18 shows the vehicle model that comprises six subsystems: the body, the drive sprocket, the suspension units (front, middle and rear, each with one or two road wheels) and the track belt. The geometry of all the parts of the robot are imported into MSC Adams as Parasolid files extracted from 3D CAD elements. This strategy is preferred to using the elementary geometries provided by the ATV toolkit to improve fidelity.

The robot main body is represented by the subsystem hull, and its inertial

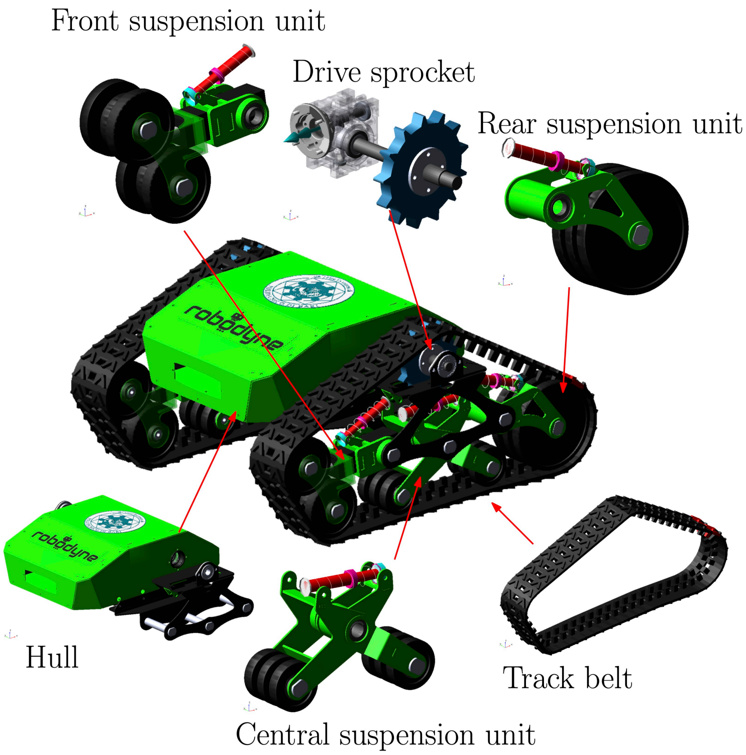


Figure 3.18: The multibody model of POLIBOT

properties are consistent with those of the real prototype. A hull property file provides the software with the coordinates of the lower and upper profiles of the body. To place the other subsystems (shown in Fig. 3.18) correctly in the final assembly with respect to the body, the hull template needs output communicators. The total mass of the tracked vehicle assembly and the position of the center of gravity are derived with static tests and compared against experimental data in Section 4.2.4.1. The powertrain engages with the track belts through shafts, gears and sprockets on each side. A revolute joint connects each sprocket to the hull to achieve purely rotational relative motion about the drive axle. To impose motion, the user can provide to the powertrain subsystem either a prescribed angular velocity or torque.

Each of the four road wheels of the tracked vehicle are suspended to the robot chassis using a trailing arm suspension type. The templates of the central and front suspension units are generated especially to include features that are not available in the ATV default databases. The subsystem of the front suspension unit, shown on the top left corner of Fig. 3.18, is made up of four rigid bodies: two road wheels and two components that constitute the swing arm. The first part of the swing arm is attached to the robot body through a revolute joint, allowing it to rotate about the hinge point. This pivoting motion is resisted by the action of a spring-damper element designed to emulate the shock absorber used in the prototype between the chassis and the arm. The second rigid body forming the front trailing arm is connected to the first one through a translational joint that allows it to change the overall length of the arm and adjust the track belt tension to an optimal value for better traction performance, similarly to the screw-nut tensioning unit of the actual robot. For this reason, the tensioner, which is an important entity of the ATV toolkit, is defined between the two sliding bodies that constitute the front trailing arm. The front suspension unit carries two road wheels that are connected via revolute joints to the terminal part of the swing arm.

The central suspension unit is shown at the bottom of Fig. 3.18. It comprises two swing arms that carry a road wheel each and that are connected to the hull through revolute joints. The relative rotation between the two swing arms is opposed by the action of another linear spring-damper force. In such way, when one wheel travels upwards because of the terrain geometry, the other is pushed against the ground, with consequent high adaptability to uneven terrains and obstacles.

The POLIBOT prototype features rubber tracks reinforced with steel cables. In the model, based on the ATV toolkit design methodology, the single track is divided into 56 discrete segments, wrapped around the road wheels and the sprocket. Each one of these segments consists of a segment part and a box-shaped connector part that are linked to one another through a 6×6 force field element, modeled as a Timoshenko beam. The elastic and damping forces and torques at the segment interfaces therefore depend on the relative displacement and velocity of the local segment frames [72], as well as on the coefficients in the stiffness and damping matrices: these are calculated by referring to the Young modulus and shear modulus of the rubber of which the track is made, i.e., $E = 5.5$ MPa and $G = 20$ MPa respectively, together with the dimensions of the cross section, namely 120 mm width and 9 mm height. A multiplier equal to 100 is applied to account for the stiffening along the longitudinal direction of the track due to the presence of the steel cables. The values obtained are summarized in Table 3.11. As for

	x	y	z
Translational stiffness (N/mm)	3.24×10^4	72.45	917.57
Translational damping (Ns/mm)	324.20	0.72	9.17
Rotational stiffness (Nmm/deg)	2.48×10^4	8.13×10^5	144.32
Rotational damping (Nmms/deg)	248.29	8.13×10^3	1.44

Table 3.11: Properties of the rubber field used for the track segment connections

the damping matrix, this is calculated as a fraction of the stiffness matrix. The damping coefficients reported in Table 3.11 correspond to a damping ratio equal to 0.01: such ratio is chosen by performing a drop simulation of the belt model alone and comparing its oscillations and final deformed shape with that of the real rubber track, as suggested in [73].

3.3 HAPF path planning algorithms for tracked robots

Simulations have been carried out with a model of the all-terrain rover Polibot. More details on Polibot's design and its suspension modelling can be found in section 3.2.

3.3.1 Realistic scenarios

To test the algorithms developed in this thesis, it was decided to use realistic planetary exploration scenarios. For example, Fig. 3.19 shows a Digital Terrain Model (DTM) of a zone of Mars (latitude 76.2°, longitude 95.44°), which is one of the many made available on the High Resolution Imaging Science Experiment (HiRISE) website by The University of Arizona [74]. HiRISE is a powerful camera onboard the Mars Reconnaissance Orbiter, launched in 2005, arrived at Mars in 2006 and been imaging ever since. HiRISE DTMs are made from two images of the same area on the ground, taken from different look angles. For a detailed explanation of the process used to create HiRISE DTMs, the reader is invited to refer to [75]. The map scale of the DTM of Fig. 3.19 is 0.98 m per pixel, and the vertical precision is in the tens of centimeters.

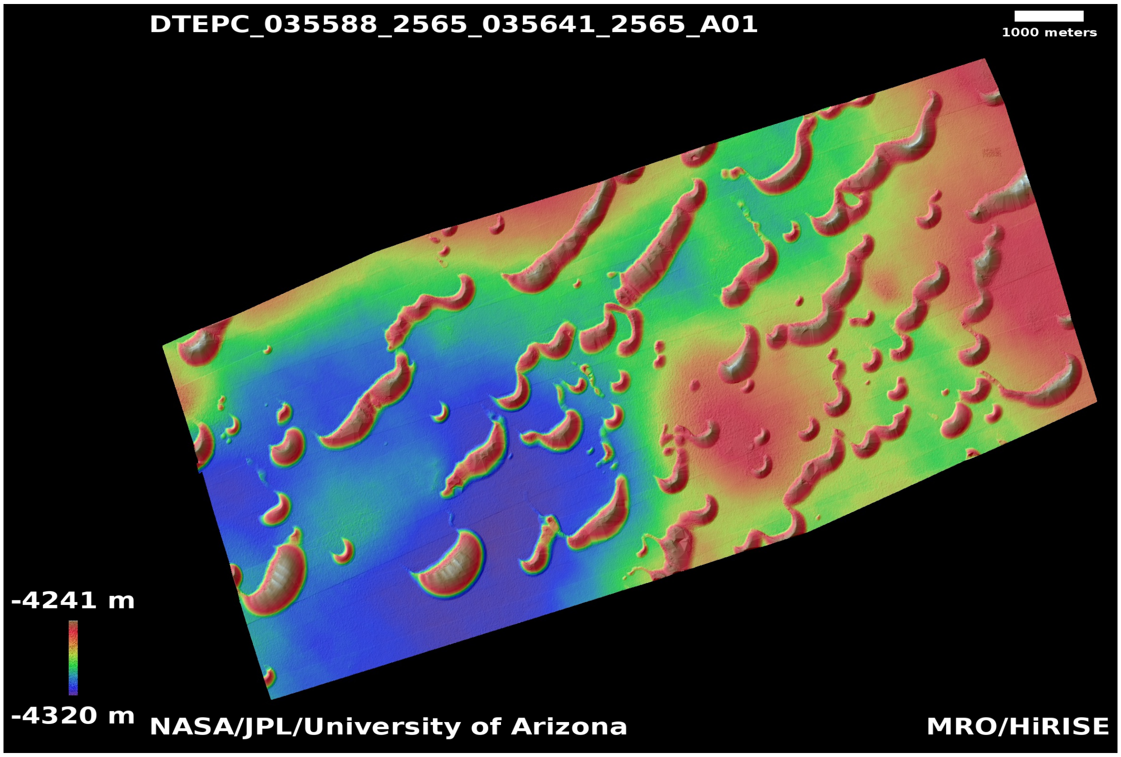

DTEPC_035588_2565_035641_2565_A01
1000 meters
-4241 m
-4320 m
NASA/JPL/University of Arizona
MRO/HiRISE

Figure 3.19: HiRISE Digital Terrain Model of a zone of Mars (latitude 76.2°, longitude 95.44°).

To use such a DTM for HAPF path planning, it is necessary to extract an obstacle map from it. To do so, first, a smaller 2000 m x 2000 m square section of the original DTM is selected and it is shown in Fig. 3.20. As indicated in the contour plot, X and Y represent the global coordinates used for this study, and their origin is in the bottom left corner of the DTM of Fig. 3.19. Also, Z is the elevation as obtained from the DTM.

Next, a down-sample of the original DTM is performed, tripling its scale from 0.98 per pixel to 2.94 per pixel. There are two advantages to start with a coarser map: first, it is a way to simulate a realistic planetary exploration scenario in which a detailed map is not available at the beginning of the mission, and second, it speeds up the numerical solution of Laplace's equation as the size of the computational domain is reduced by a factor of 9. At this point, the sectioned and down-sampled DTM is ready to be converted into an obstacle map. The approach used in this work is to calculate the absolute slope in each point of the DTM and mark as

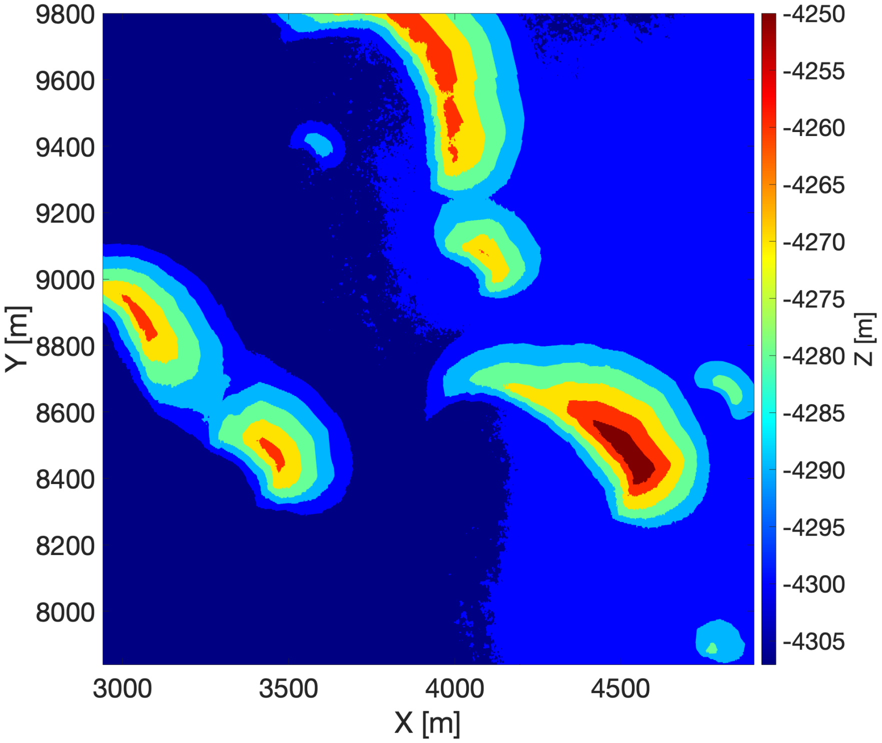


Figure 3.20: Section of the original DTM of Fig. 3.19 considered for path planning.

obstacles all the points that have a value greater than the maximum safe slope that can be overcome by the robot. For each point (i, j) in the computational grid, the absolute slope can be calculated as combination of the two components along X and Y:

$$Sl_{i,j} = \sqrt{(SlX_{i,j})^2 + (SlY_{i,j})^2} \tag{3.44}$$

$$SlX_{i,j} = \frac{Z_{i+1,j} - Z_{i-1,j}}{a} \tag{3.45}$$

$$SlY_{i,j} = \frac{Z_{i,j+1} - Z_{i,j-1}}{a} \tag{3.46}$$

where a is the grid spacing. All the points that satisfy the following condition are marked as obstacles:

$$Sl_{i,j} > \tan^{-1} \alpha \tag{3.47}$$

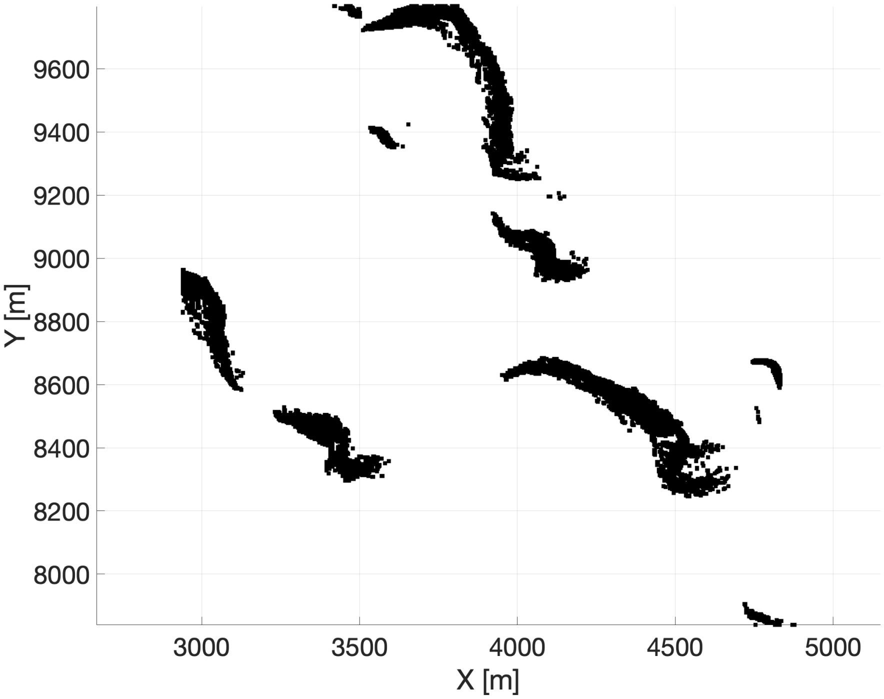


Figure 3.21: Obstacle map correspondent to the DTM of Fig. 3.20.

where tan^{-1} is the inverse tangent function and α is the maximum safe slope angle for the robot. By setting $\alpha = 20°$, the resulting obstacle map is shown in Fig. 3.21.

3.3.2 Numerical solution of Laplace's equation

The 2D version of Laplace's equation is:

$$\nabla^2\phi = \frac{\partial^2\phi}{\partial X^2} + \frac{\partial^2\phi}{\partial Y^2} = 0 \tag{3.48}$$

where ϕ is the harmonic potential function, X and Y are the 2D coordinates. To solve it numerically, it is necessary to choose a discretization scheme. The most common approximation of the 2D laplacian is the 9-point laplacian [76]. Following this scheme, the 2D Laplace's equation discretized on a squared uniform grid (i.e. with equal constant grid spacing along the two directions) becomes:

$$\begin{aligned}\phi_{i+1,j+1} + \phi_{i-1,j+1} + \phi_{i+1,j-1} + \phi_{i-1,j-1} + \\ +4\left(\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1}\right) - 20\phi_{i,j} = 0\end{aligned} \tag{3.49}$$

where the coefficients are obtained analytically as shown in [76]. Equation 3.49 can be used to set up an iterative algorithm in which the value of the potential in each point (i, j) at the iteration k is given by:

$$\begin{aligned}\phi^k_{i,j} = \frac{1}{20}(\phi^{k-1}_{i+1,j+1} + \phi^{k-1}_{i-1,j+1} + \phi^{k-1}_{i+1,j-1} + \phi^{k-1}_{i-1,j-1} + \\ +4\left(\phi^{k-1}_{i+1,j} + \phi^{k-1}_{i-1,j} + \phi^{k-1}_{i,j+1} + \phi^{k-1}_{i,j-1}\right))\end{aligned} \tag{3.50}$$

where $k = 1, 2, 3, \ldots$ is the iteration number. The iterative algorithm of Eq. 3.50 requires an initial guess ϕ^0. When the residuals are low enough, that is when the difference between the potential at two consecutive iterations is lower then a certain threshold, the iterative algorithm can be stopped and the final ϕ^k is the numerical solution of the 2D Laplace's equation.

To apply Eq. 3.50 to the computational grid of Fig. 3.21, it is necessary to define the boundary conditions first. Boundary conditions are required for the start position, the goal position, obstacles and outer limits of the grid. A Dirichlet condition (prescribed value of potential) is applied to the start and goal to represent a mass source and a mass sink. Usually, the potential has an infinite value (positive and negative respectively) for these two conditions. For numerical purposes, prescribing a high (and low) value is enough, for example:

$$\phi_{start} = 10^3 \tag{3.51}$$

$$\phi_{goal} = -10^3 \tag{3.52}$$

A Neumann condition (prescribed value of derivative of potential) is applied to obstacles and outer boundaries as they have to act as an impenetrable wall. Thus, the normal velocity at the wall is set to zero as follows:

$$\left(\frac{\partial \phi}{\partial n}\right)_{wall} = 0 \tag{3.53}$$

where n is the wall normal. In this research, obstacles are constructed as an irregular set of squares. Therefore, the "walls" of the obstacle are locally parallel to the X or Y axis. Equation 3.53 can be discretized using a two-point scheme. For brevity, in this document only the discetization in case of a wall parallel to the X axis with free space below (Fig. 3.22) is presented. The other cases are similar. For any pair of points (i, j) and $(i, j+1)$ that are next to a wall parallel to the X axis and located in point $(i, j+2)$, the discretized Neumann condition (Eq. 3.53) is:

$$\phi_{i,j+1} = \phi_{i,j} \tag{3.54}$$

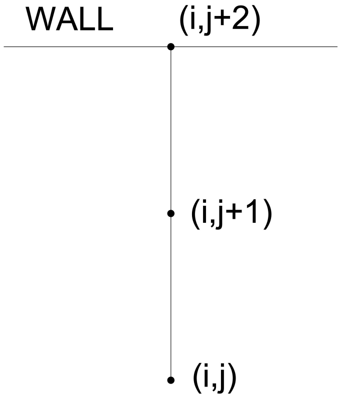


Figure 3.22: Representation of a wall parallel to the X axis in the computational grid with free space below.

The accuracy and speed of the numerical solution of Laplace's equation depends on the chosen type of discretization and iterative schemes. There are other iterative schemes proposed in the literature that have much faster convergence. However, this is outside the scope of this work, as Laplace's equation is only solved once at the beginning of the mission. The solution obtained with equations 3.50, 3.51, 3.52 and 3.53 is accurate enough to produce smooth and collision-free paths, as shown in the results section 4.3.

3.3.3 Path optimization problem

After the computation of the HAPF by numerically solving Laplace's equation, many paths can be derived starting from the start location and following the gradient descent. The gradient of the potential is not defined at the start position, because it is a point on the boundary of the computational domain. Therefore, the initial direction can be arbitrarily chosen, as tracked robots can easily rotate on the spot. For any different initial direction, a different path can be derived via gradient descent. The purpose of this section is to describe the path optimization problem used in this research to choose the optimal path and, consequently, the optimum initial direction.

Let P be a set of N suitable paths found via gradient descent, and p_j a generic path of this set, with $j = 1, \ldots, N$. The optimum path is chosen by minimizing a cost function $C(p_j)$ given by a weighted sum of four different costs:

$$C(p_j) = \sum_{i=1}^{4} w_i c_i(p_j) \tag{3.55}$$

$$\sum_{i=1}^{4} w_i = 1 \tag{3.56}$$

$$0 \leq w_i \leq 1 \tag{3.57}$$

where w_i are the weights. Each cost c_i is a function of the generic path p_j and represents a specific optimality criterion. The costs are normalized to have always values between 0 and 1.

The first cost is related to the distance to obstacles. Let $d(p_j)$ be a function that calculates the minimum distance between any point of path p_j and any obstacle. For this criterion, the optimum path is the one with the highest value of d, and c_1 can be calculated as follows:

$$c_1(p_j) = \frac{max(d) - d(p_j)}{max(d) - min(d)} \tag{3.58}$$

The second cost is related to path length. Let $L(p_j)$ be a function that calculates the length of path p_j. For this criterion, the optimum path is the one with the lowest value of L, and c_2 can be calculated as follows:

$$c_2(p_j) = \frac{L(p_j) - min(L)}{max(L) - min(L)} \tag{3.59}$$

The third cost is related to the absolute slope (Eq. 3.44) of the terrain encountered along the path. Let $sl(p_j)$ be a function that gives the average absolute slope encountered along path p_j. For this criterion, the optimum path is the one with the lowest value of sl, and c_3 is given by:

$$c_3(p_j) = \frac{sl(p_j) - min(sl)}{max(sl) - min(sl)} \tag{3.60}$$

The fourth and final cost is related to the sinkage of the robot over deformable terrain. Let $Za(p_j)$ be a function that gives the average sinkage of the robot along path p_j. For this criterion, the optimum path is the one with the lowest value of Za, and c_4 is given by:

$$c_4(p_j) = \frac{Za(p_j) - min(Za)}{max(Za) - min(Za)} \tag{3.61}$$

For details about sinkage estimation, please refer to the following section.

3.3.4 Sinkage estimation

A nonholonomic skid-steering tracked vehicle with independently driven tracks is represented in Fig. 3.23, along with the global and local reference systems OXY and Gxy, respectively. X_G and Y_G are the global coordinates of the center of gravity G. The heading angle θ is the rotation angle between the two reference systems. X_G, Y_G and θ are the generalized coordinates of the robot. In general, the velocity v of the center of gravity G has two components in the local reference system:

$$v_x = v\cos\beta \tag{3.62}$$

$$v_y = -v\sin\beta \tag{3.63}$$

where β is the sideslip angle, which is not zero every time the vehicle steers, because the center of gravity and the center of turn are not aligned on the same transversal line [77].

The velocity of the vehicle, its rate of turn r and the curvature radius R_c are

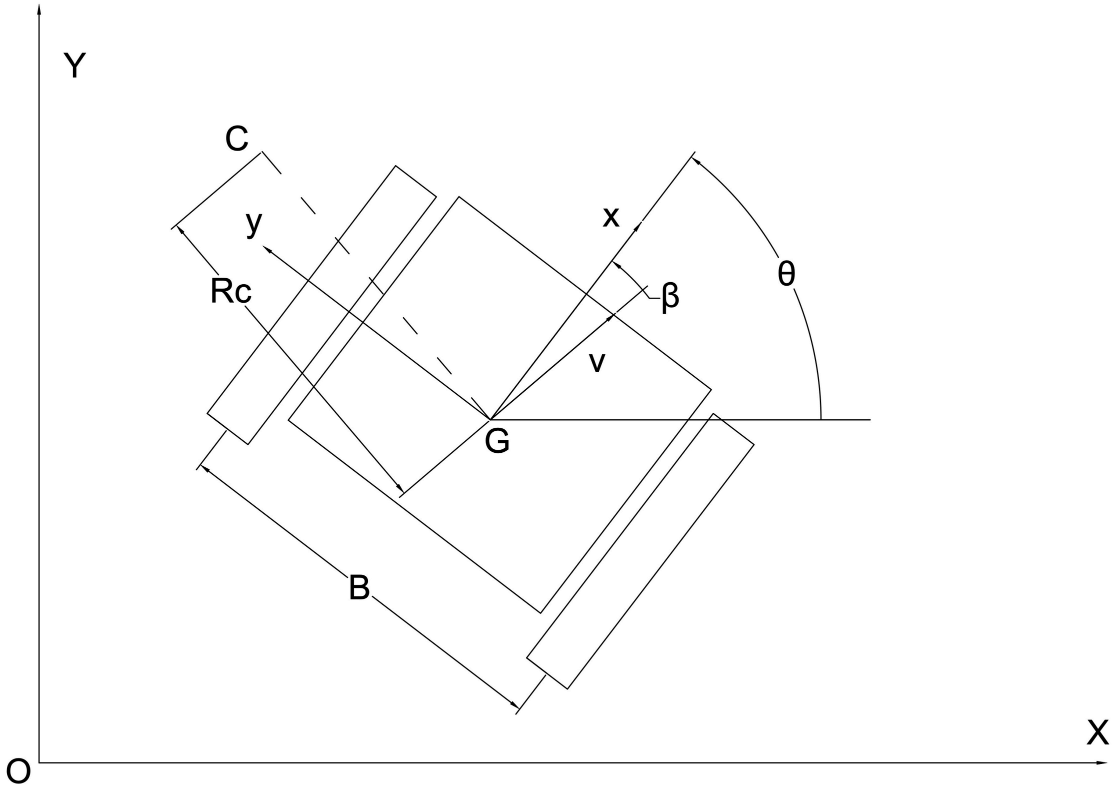


Figure 3.23: Representation of a tracked vehicle with global and local reference systems.

related by:

$$v = R_c r \tag{3.64}$$

The kinematic model of the tracked robot of Fig. 3.23 is:

$$\dot{X} = v_x \cos\theta - v_y \sin\theta \tag{3.65}$$

$$\dot{Y} = v_x \sin\theta + v_y \cos\theta \tag{3.66}$$

$$\dot{\theta} = r \tag{3.67}$$

The longitudinal speed of the vehicle v_x and its rate of turn r can be expressed as a function of the longitudinal speed of the two tracks v_{xl} and v_{xr}:

$$v_x = \frac{v_{xr} + v_{xl}}{2} \tag{3.68}$$

$$r = \frac{v_{xr} - v_{xl}}{B} \tag{3.69}$$

$$v_{xr} = \frac{R\omega_r}{\tau}(1 - S_r) \tag{3.70}$$

$$v_{xl} = \frac{R\omega_l}{\tau}(1 - S_l) \tag{3.71}$$

where R is the sprocket radius, ω is the rotational speed of the motors (right and left), τ is the transmission ratio between the motor and the sprocket, S is the slippage of the tracks (right and left) and B is the distance between the two tracks.

The load acting on each track of the vehicle is influenced by the slope of the terrain. The slope angle in the transversal direction γ causes a load transfer across the tracks. The slope angle in the longitudinal direction δ, affects the pressure distribution below each track. Referring to the slopes calculated along X and Y (Eq. 3.45 and 3.46), these angles can be derived as follows:

$$\delta = \tan^{-1}\left(SlX\cos\theta + SlY\sin\theta\right) \tag{3.72}$$

$$\gamma = \tan^{-1}\left(-SlX\sin\theta + SlY\cos\theta\right) \tag{3.73}$$

The forces acting in the transversal plane are shown in Fig. 3.24. For simplicity, the effect of the slope angle in the transversal direction (γ) is represented by changing the direction of the weight.

From the equilibrium of forces along the z axis and the equilibrium of moments around the x axis, it is possible to derive an expression for the two loads:

$$W^r = m\left[\frac{1}{2}g\cos\gamma + \frac{h}{B}\left(rv_x + \dot{v}_y - g\sin\gamma\right)\right] \tag{3.74}$$

$$W^l = m\left[\frac{1}{2}g\cos\gamma - \frac{h}{B}\left(rv_x + \dot{v}_y - g\sin\gamma\right)\right] \tag{3.75}$$

where m is the robot mass, g is the gravitational acceleration, h is the height of the center of gravity. From the two equations above, it is easy to verify that the two loads are equal when the transversal slope is null ($\gamma = 0$) and the vehicle is not

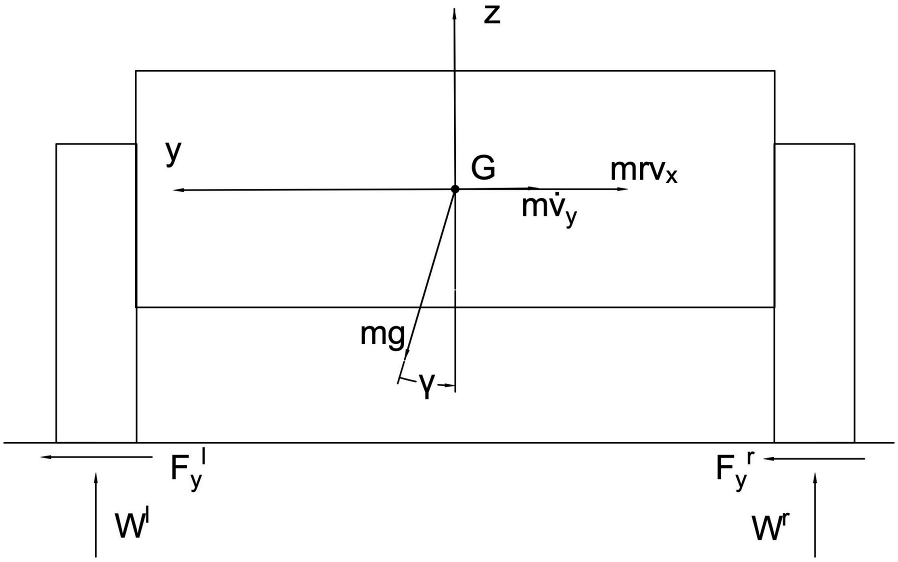


Figure 3.24: Forces acting on the robot on the xy (top) and yz (bottom) planes when steering.

turning ($r = 0$) and not starting to turn ($\dot{v}_y = 0$).

After the loads are known, it is necessary to consider how the suspension distributes the load across the four road wheels. The analytical model of the suspension described in section 3.2.2 is used for this scope. Referring to the schematics of the suspension shown in Fig. 3.25, the inputs to the model are the load acting on the track W and the slope angle in the longitudinal direction δ, while the outputs are the four loads acting on the road wheels N_i (with $i = 1, \ldots, 4$).

Finally, it is possible to compute the pressure distribution below the track. In reality, pressure distribution below a track depends on many factors, and the most important are the pressure-sinkage relationship, the response of the terrain to repetitive loads and the deformation of the flexible track due to terrain pressure [78]. However, it is time consuming to perform pressure calculations that include all these aspects for every position in the path. In this work, a simpler approach is chosen, where the track is considered as rigid and the pressure depends only on the load on each wheel. One of the possible idealized normal pressure distribution types is the sinusoidal [79], in which the pressure varies from peak values in correspondence of each road wheel and zero at the mid-points between each pair of road wheels, as shown in Fig. 3.26. The values of the peaks are chosen to satisfy the condition that

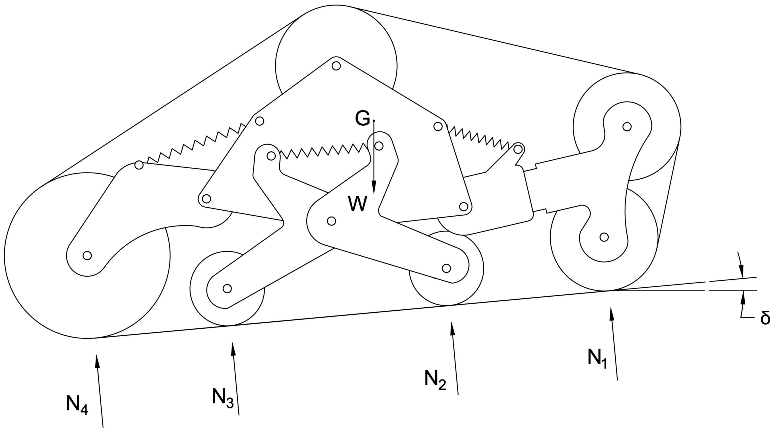


Figure 3.25: Schematics of the suspension with indication of the loads acting on each road wheel.

the integral of the normal pressure between two consecutive zero-pressure points is equal to the load of the correspondent road wheel.

This pressure distribution can now be used to estimate the sinkage used in Eq. 3.61 via Bekker's pressure-sinkage equation [80]:

$$z = \left(\frac{p}{\frac{k_c}{b} + k_\phi} \right)^{\frac{1}{n}} \tag{3.76}$$

where z is the sinkage, p is the pressure, b is the width of the contact patch, n, k_c, and k_ϕ are terrain-specific pressure-sinkage parameters.

3.3.5 Global and local planning

The objective of this section is to wrap up the path planning methodology presented in this thesis. The planning framework is summarized in the flow chart of Fig. 3.27. The initial obstacle map (Fig. 3.21) is provided to the global planner, together with the robot start position and the goal location. The global planner solves Laplace's equation as described in section 3.3.2 and choses the optimal path (section 3.3.3). At this stage, the motion is imposed to the robot, and the kynematic model of section 3.3.4 is used to calculate rate of turn and heading angle at each point of the path. If a change in the environment requires a deviation from the global optimal path, the local planner kicks in.

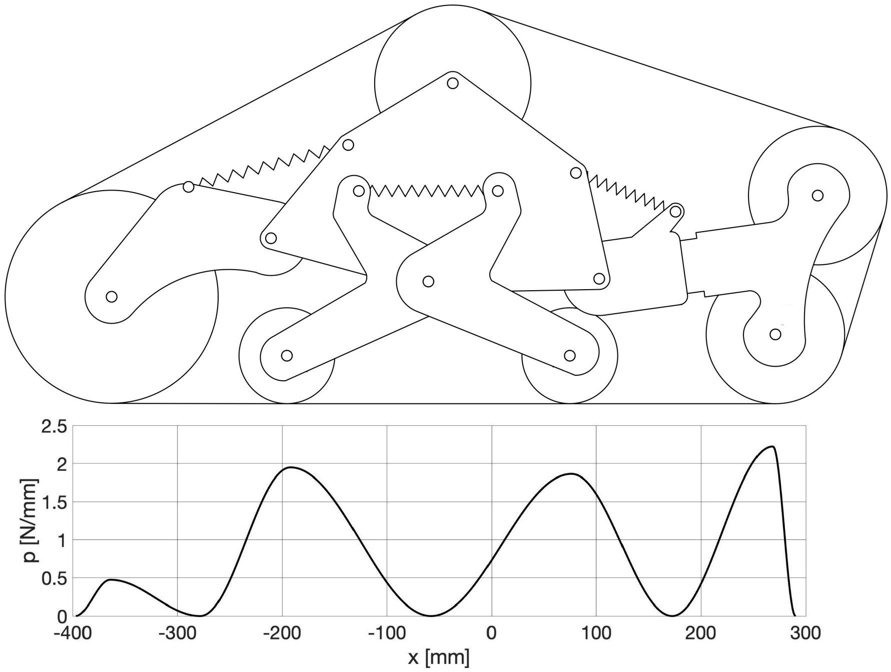


Figure 3.26: Pressure distribution below each track.

The strategy adopted by the local planner to avoid new detected obstacles is based on adding a provisional local potential source to the existing potential field which acts as a repulsive force that pushes the robot away. After the obstacle is avoided, the local planner tries to return to the global optimal path, by provisionally adding a local sink in correspondence of the closest and safest point of the optimal path, which acts as an attractive force. Results obtained with the proposed path planning framework are presented and discussed in the next chapter. The function used to add these provisional potential sources and sinks is the following:

$$\phi_{local} = \begin{cases} \nu \log\left(\frac{d_0}{d}\right) & if \quad d < d_0 \\ 0 & if \quad d \geq d_0 \end{cases} \tag{3.77}$$

where ϕ_{local} is the potential associated with every provisional source or sink, ν is a constant (positive for sources and negative for sinks), d is the distance from the source (or sink), d_0 is a constant that defines how big is the area influenced by the new local potential, to ensure that the new disturbance does not affect zones of the potential that are far from the new obstacles.

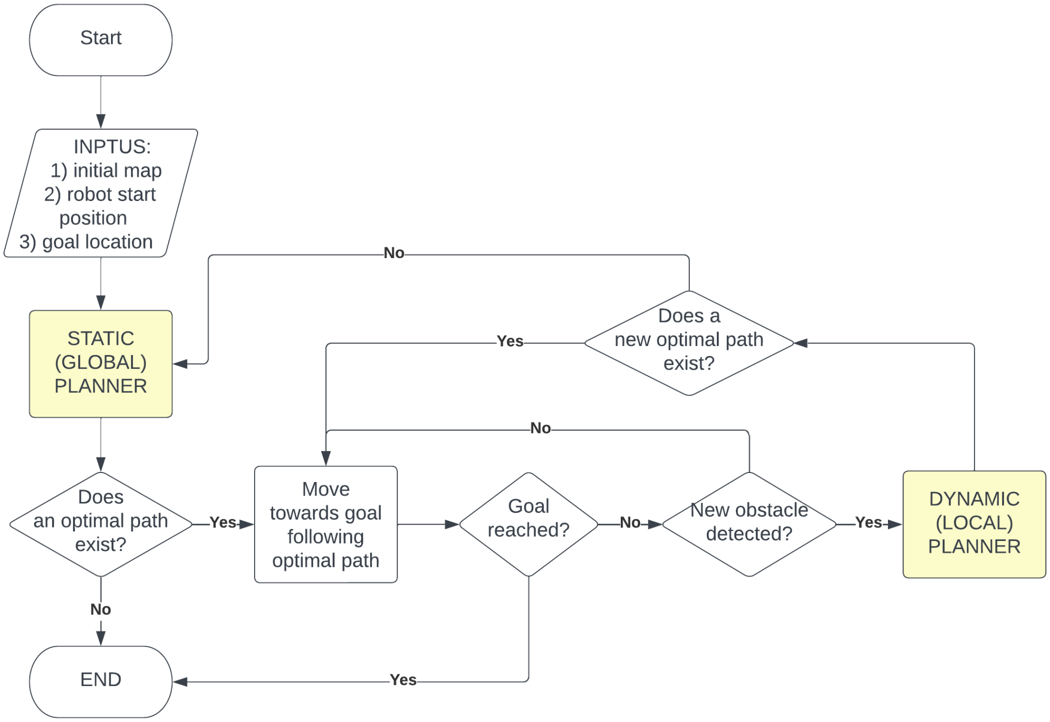


Figure 3.27: Path planning framework.

Chapter 4

Results

4.1 Role of feature and signal selection for terrain learning

In this section, the results of the proposed terrain algorithms are presented. First the generalization problem is applied to the main dataset. Next, results for two extrapolation cases are presented.

4.1.1 Generalization

In the generalization problem, only the main data set is used (i.e., experiments on paved ground, gravel, and sand). The algorithms are tested via 5-fold cross validation. The data set comprises of 1204 samples, where a sample corresponds to a 2-second time window. Of these 1204 samples, 443 are collected on paved ground, 338 on gravel and 423 on sand.

One of the objectives of this thesis is to demonstrate how a proper feature selection algorithm can reduce the computational and memory cost of the model, while maintaining a similar accuracy in prediction. Table 4.1 shows comparison between the two machine learning algorithms in terms of accuracy and computational burden. Moreover, SVM is tested with three different feature sets:

- Direct feature set (44 features)
- Full feature set (80 features)

- Best feature set (18 features)

while CNN is tested with three different signal sets:

- Direct signal set (11 signals)
- Full signal set (20 signals)
- Best signal set (13 signals)

	SVM			CNN		
Feature and signal sets	Direct	Full	Best	Direct	Full	Best
Accuracy [%]	89.8	90.8	90.9	95.6	96.4	96.2
Model memory usage [kB]	547.6	996.9	228.0	44.9	53.2	45.8
Training time [ms]	118.9	157.7	148.0	1.07 e4	1.46 e4	1.06 e4
Testing time [ms]	17.4	29.8	21.2	153.0	119.4	114.7
Feature extraction time [ms]	0.6	0.9	0.6	2.6	4.6	2.7

Table 4.1: Performance comparison between terrain classifiers trained on different feature sets: direct, full, best feature set.

The signals used for training CNN correspond to those used to compute SVM features. In fact, the 44 direct features are the 4 statistical moments of the 11 direct signals and the full 80-feature set is composed by the 4 statistical moments of the full 20-signal set. Furthermore, the training set for CNN includes the signals used to derive the features in the best feature set. Namely, the 13 best signals are: friction coefficients 1 and 3, longitudinal, lateral and vertical accelerations, drive torque, yaw, pitch and roll rates, longitudinal and vertical forces, sinkage, drive PWM.

The accuracy of the SVM model trained with the direct and full feature sets is 89.8% and 90.8%, respectively. With the full feature set, more samples are correctly classified by SVM, but memory usage has increased by 82%, training time by 32%, testing time by 71% and feature extraction time by 50%. This proves the effectiveness of the signal augmentation in terms of accuracy and shows the drawbacks in terms of computational burden. The purpose of feature selection is to reduce the computational cost, without losing classification accuracy. The results presented for SVM trained with the best feature set, prove that the feature selection algorithm proposed in this work is effective. In fact, the accuracy reaches

90.9% and, when compared to the SVM trained on the full feature set, the model memory usage is reduced by 77%, training time by 6%, testing time by 29%, feature extraction time by 33%.

The effectiveness of both input signal augmentation and feature selection is also confirmed by the results presented for CNN. This deep learning algorithm gains in terms of accuracy from signal augmentation reaching 96.4%. Using the full signal set still results for CNN in the same drawbacks presented for SVM: model memory usage increased by 18%, training time by 36%, feature extraction time by 77%. In contrast with SVM, testing time for CNN with full signal set is reduced by 22%. Training CNN with the best signals resulting from feature selection leads to an accuracy of 96.2% and when compared to the full-signal CNN, the model memory usage is reduced by 14%, training time by 27%, testing time by 4%, feature extraction time by 41%. Feature extraction times presented in the last row of Table 4.1 are suitable for online application for both SVM and CNN, even if construction of multichannel spectrograms from best signals for CNN takes about 2.1 ms more than construction of best features for SVM. It should also be noted that feature extraction time for both SVM and CNN can be further improved by optimizing the current MATLAB code using vectorization or processing the data directly with a C++ code. Note that at the time of writing of the thesis, the algorithms and the dataset are under revision in a private Github repository that will be made available to the interested readers upon publication.

Confusion matrices for both SVM and CNN are shown in Fig. 4.1 only for best feature and best signal sets. Sensitivity results for each class are contained in the diagonal elements of each confusion matrix. The performance of both models in terms of precision, recall and $F1$ score are shown in Table 4.2. Both models perform good in generalization of data, with CNN being slower but significantly more accurate. This increase in classification accuracy is not the main advantage for CNN classification model with respect to SVM. Where the two models show the greatest difference in classification performance is indeed extrapolation, as shown in the following section.

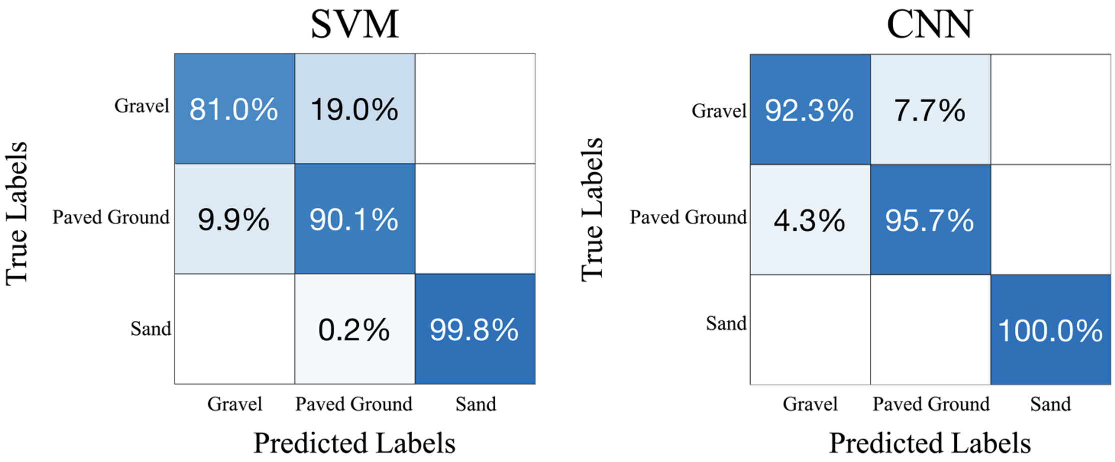


Figure 4.1: Generalization Results for best features SVM and best signals CNN.

	SVM			CNN		
Class	Gravel	Paved Ground	Sand	Gravel	Paved Ground	Sand
Precision [%]	89.1	82.4	100	80.3	82.2	100
Recall [%]	81.0	90.1	99.8	92.3	95.7	100
$F1$ score [%]	84.9	86.1	99.9	85.9	88.4	100

Table 4.2: Accuracy, Precision, Recall and F1 score for SVM and CNN in generalization.

4.1.2 Extrapolation

In the extrapolation problem, the operating conditions of training and testing sets are different, therefore these sets do not come from the same population. In this work two extrapolation cases are presented. The first one deals with varying rover speed, whereas the second one assesses the performance of the algorithms on a terrain unseen in the training phase.

4.1.2.1 Testing on a new vehicle velocity

During the experiments with SherpaTT, the rover was controlled at two different speeds: 0.1 m/s and 0.15 m/s. Of the 14 runs, 7 were conducted at low speed (0.1 m/s) and 7 at high speed (0.15 m/s). Data collected at low-speed form the low-speed distribution, whereas data collected at high-speed belong to the high-speed distribution. In the extrapolation problem presented here, low-speed data are used as training set, while high-speed data are used as testing set. Both sets belong to the main dataset (paved ground, gravel, and sand). Proprioceptive data are very useful for terrain classification but also show a strong dependency from traversing speed [81]. Most terrain classification algorithms analyse and classify proprioceptive data acquired at constant traversing velocity on different terrains. Studies have been also conducted to show dependency of terrain classification performances from rover's traversing speed, searching for the velocity that maximizes classification performance. To be able to classify the traversed terrain at any travelling speed, a rover should be equipped with a model trained on a vast variety of possible traversing speeds or could only use speed-independent features that are difficult to construct and may not be well suited for terrain classification. Another way of achieving the goal of sensing and classifying the terrain at any travelling speed is using a model that shows good results when tested on data acquired at a traversing velocity different from the one used for training. Figure 4.2 contains the confusion matrices for both SVM and CNN when trained on low-speed data and tested on high-speed ones. As can be seen, despite both models showed good results in generalization only CNN is also capable of extrapolating the information of the traversed terrain from data acquired at a different speed. The two models were still trained and tested using only best feature set for SVM and corresponding signal

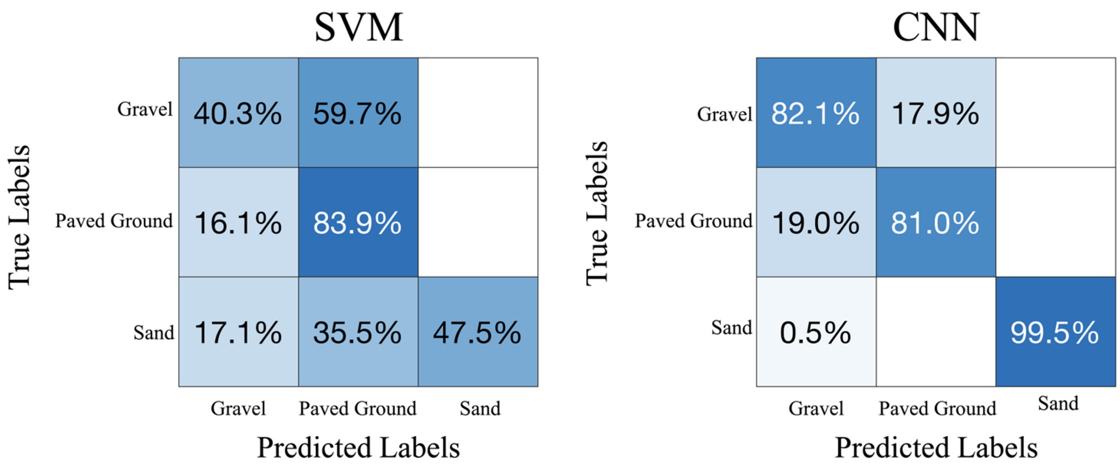


Figure 4.2: Extrapolation results for best features SVM and corresponding signals CNN.

set for CNN. While CNN keeps classification accuracy as high as 89.5%, SVM becomes unreliable achieving only 55.7% of correctly classified data samples. The performances of both models in terms of precision, recall and $F1$ score are shown in Table 4.3.

	SVM			CNN		
Class	Gravel	Paved Ground	Sand	Gravel	Paved Ground	Sand
Precision [%]	54.8	46.8	100	80.3	82.2	100
Recall [%]	40.3	83.9	47.5	82.1	81.0	99.5
$F1$ score [%]	46.4	60.1	64.4	81.2	81.6	99.7

Table 4.3: Precision, Recall and F1 score for SVM and CNN in extrapolation using varying velocity.

It should also be pointed out that high-speed data used as testing constitute 50% of available data, representing therefore a testing set larger than the ones usually used (20-30%). The robustness of the classification performance of CNN on a large testing set composed by data acquired at a different speed suggests that this model is well suited for terrain classification purposes. Moreover, the features automatically learned from signal spectrograms appear to be more reliable than statistic ones and represent a better choice to be able to classify the traversed terrain at various travelling velocities.

Similar results are obtained when trained on high-speed data and tested on low-speed data, and they are omitted for brevity sake.

4.1.2.2 Testing on an independent dataset

The second extrapolation use case aims to evaluate the system response when labeling observations collected on a terrain different from those used in training (independent dataset). To this aim, the ground classifier previously trained on the main data set (formed by paved ground, gravel, and sand) is further validated on a representative dataset gathered from a second field test campaign run in a planetary-analogue terrain in a sand mine near Bremen (Fig. 4.3).

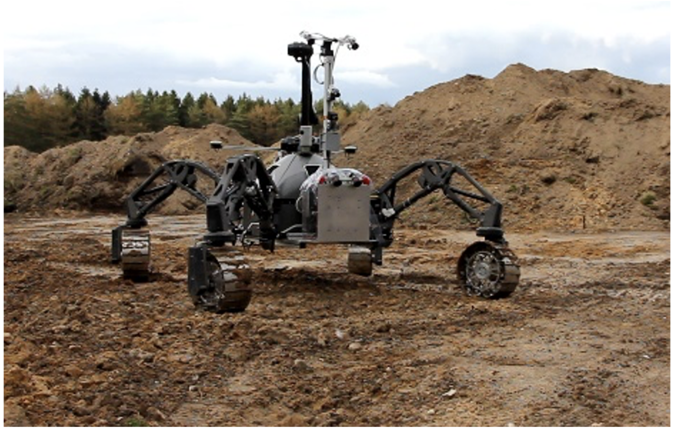


Figure 4.3: Sherpa TT during the sand mine testing.

For this extrapolation challenge, we have tried to solve the classification problem at hand by referring to terrain difficulty labels rather than specific terrain classes, as explained in Table 4.4. Adopting the proposed terrain difficulty scale, paved ground and sand can be seen as the opposite extremes. Firm ground offers better traction and less compressibility, therefore a low difficulty label can be assigned to it. Conversely, soft ground poses more challenges, and it is scored as a highly difficult surface. Then, the difficulty degree associated with an unknown observation can be considered as inversely proportional to the distance from the class sand. One should note that such a generalization effort can be useful or necessary for the practical implementation of planetary exploration terrain classifiers that can be only trained on Earth using representative analogue surfaces, and then applied to unknown planetary surfaces via extrapolation.

Terrain type	Equivalent Category of Terrain difficulty
Sand	High
Gravel	Medium
Paved Ground	Low

Table 4.4: Category of difficulty assigned to each terrain type of the training set.

The sand mine independent dataset consists of 302 samples, where, again, a sample corresponds to a 2-second window. It should be also underlined that, although ground-truth data is not available for this extrapolation problem, the terrain in the sand mine can be expected as a surface with medium-high difficulty, like the sand type of the main dataset (Fig. 3.2) but somewhat more compact and humid. As an indicative measure, sample tracks left by the wheels on the sand mine terrain are shown in Fig. 4.4.

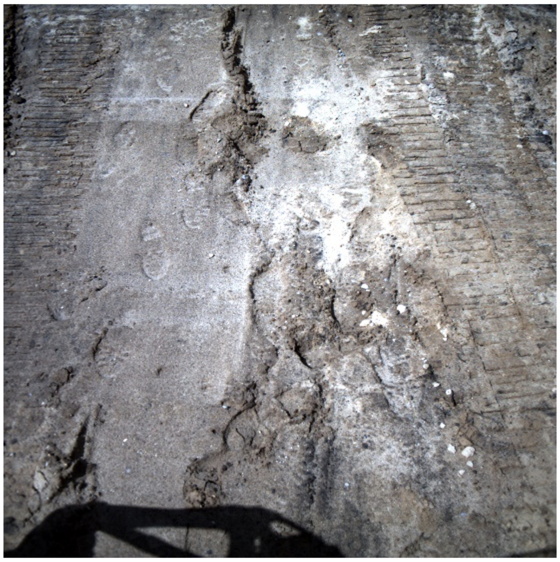


Figure 4.4: Close up of the tracks left by the wheels of SherpaTT during the sand mine testing.

The classification results obtained from SVM and CNN are collected in Table 4.5 showing predicted labels of terrain difficulty. Out of the 302 samples, the SVM-based algorithm classifies 71.2% as high difficult terrain, 17.2% as medium and 11.6% as low. CNN performs similarly, classifying 69.9% of the new terrain samples as highly difficult, 24.2% as medium and 5.9% as low. A relatively low

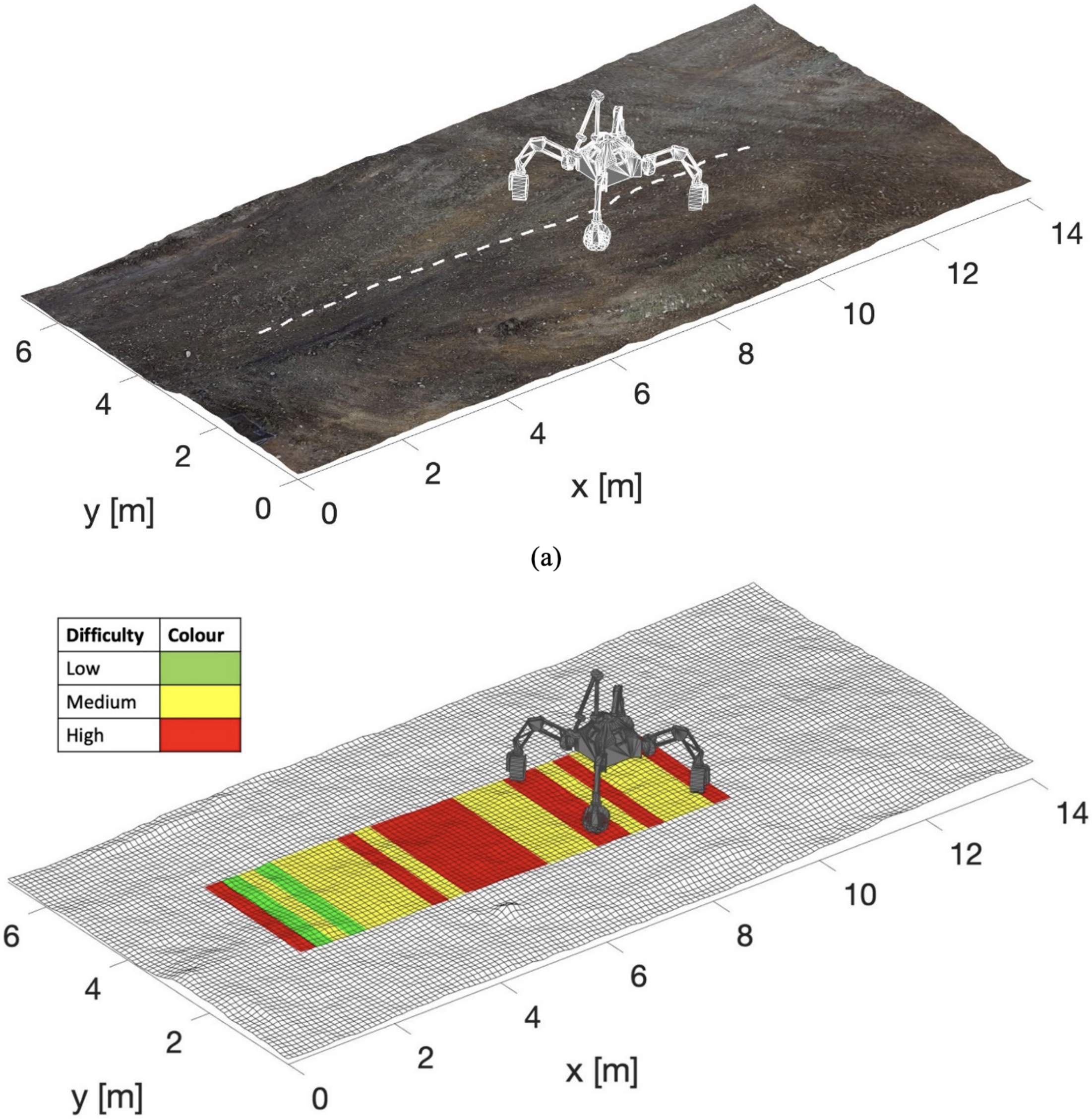
Terrain Difficulty labels	SVM	CNN
High	215	211
Medium	52	73
Low	35	18

Table 4.5: Terrain difficulty predictions as obtained from SVM and CNN in the sand mine test.

percentage of the test samples (about 12% for SVM and 6% for CNN) is classified as hard soil.

For an easier visualization, the results obtained from the CNN-based classifier are presented in Fig. 4.5 during a sample straight run using a semantic labelling where the successive terrain patches traversed by the rover are marked according to a color map that reflects the terrain difficulty scale of Table 4.4 (see also to the inset of Fig. 4.5(b)). We recall that three discrete levels of terrain difficulty are considered: low, medium, and high.

Figure 4.5(a) shows the 3D stereo-generated map of the environment with overlaid a CAD model of SherpaTT and the path followed by the rover denoted with a dashed white line, whereas in Fig. 4.5(b) the corresponding terrain labeling is reported with terrain patches marked respectively in red, yellow, and green, for high, medium, and low difficulty. In this test that was performed on fairly homogeneous terrain, the system mostly classifies the sand mine surface as of medium-high difficulty with two erroneous predictions (low difficulty) between 2 and 3 m.



(a)

(b)

Figure 4.5: Semantic labeling using discrete terrain difficulty categories: (a) 3D stereo-generated map of the environment with overlaid the path (dashed white line) followed by the rover, (b) corresponding terrain difficulty visualization. Terrain patches are marked respectively in red, yellow, and green, for high, medium, and low difficulty.

4.2 Assessment of Polibot suspension system

4.2.1 Analytical model validation

The ability of the proposed analytical model to predict the configuration of the real Polibot is evaluated in various scenarios. First, the static or baseline configuration of Polibot is considered. Then, the influence of the weight change on the system configuration is investigated. Finally, two conditions of obstacle climb are evaluated. In all tests, the robot configuration as predicted by the quasi-static kinamatic model is compared with that of the real prototype.

4.2.1.1 Static configuration and impact of payload

The static or baseline configuration refers to the robot at rest on a flat surface (see, for example, Fig. 3.10). The actual robot parameters are used in the analytical model, as shown in Table 4.6.

Parameter	Description	Value
X_{CP_3}	X-coordinate of contact point for W_3	0
f_{te}	Terrain elevation function	Z = 0
L_{nom}	Nominal track length	2026 mm
m_{nom}	Robot's mass	96.2 kg
(X_G^v, Z_G^v)	Center of gravity in vehicle coordinates	(50, -100) mm
F_{res}	Motion resistance	0 N
$L_{max,1}$	Maximum length of spring 1	150 mm
$L_{max,2}$	Maximum length of spring 2	150 mm
$L_{max,3}$	Maximum length of spring 3	130 mm
$F_{pre,1}$	Pre-load of spring 1	60 N
$F_{pre,2}$	Pre-load of spring 2	0 N
$F_{pre,3}$	Pre-load of spring 3	120 N
d_{11}	Length of FSA (Fig. A.1)	198 mm

Table 4.6: Input parameters for the static configuration

The results obtained from the analytical model compared with the experimental measurements taken manually using a measuring tape and a digital inclinometer are collated in Table 4.7. The average relative percentage error in predicting the real robot configuration resulted in about 4% with a worst case scenario of 10% for the chassis pitch angle estimation.

Variable	Analytical Model	Real robot	Relative error [%]
X_A [mm]	311.0	303.9	2.3
Z_A [mm]	303.3	309.4	-1.9
θ [deg]	4.41	4.02	10.0
θ_1 [deg]	61.6	59.1	4.3
θ_2 [deg]	38.3	40.8	-6.1
θ_3 [deg]	71.6	72.2	-0.9
θ_4 [deg]	52.2	53.5	-2.5

Table 4.7: Polibot configuration expressed in the global reference frame as obtained from the quasi-static model and measured from the real robot in the static configuration

It should be noted that a nominal pitch angle is required for the system: the payload will be added to the robot in such a way that the chassis will tend to rotate clockwise. In addition, a positive nodding angle is sought to keep the line of sight and look-ahead distance in a range of about 1-3 meters away from the robot. Therefore, a positive pitch angle is generally considered positive and a requirement of the robot.

When the robot weight increases by 83% simulating the addition of a 80 kg payload, all suspension elements are compressed, leading to a decrease of the center of mass height of 12 mm and a rotation of the chassis of 1 degree, as shown in Fig. 4.6. The comparison of the analytical model with the real robot configuration is reported in Table 4.8. Again, the largest error results in the picth angle prediction (13%), whereas the center of mass height is estimated with a 3% accuracy.

Configuration	Variable	Analytical Model	Measurement	Relative error [%]
Added mass (80 kg)	Z_A [mm]	291.2	300.5	-3.1
	θ [deg]	2.61	3.04	-13.3

Table 4.8: Results of 80 kg payload configuration.

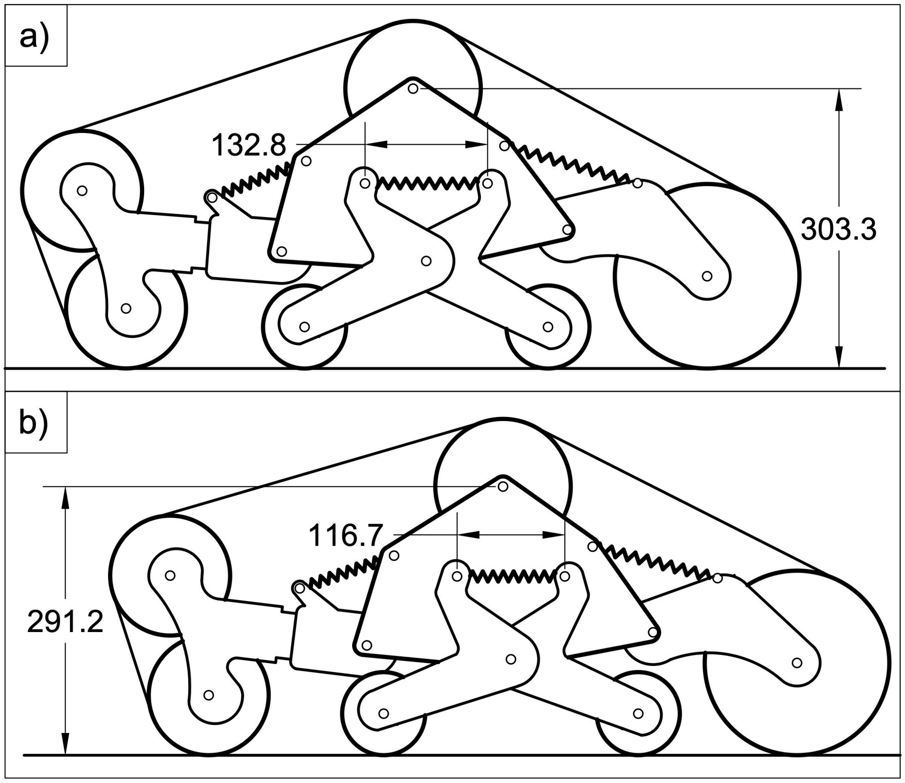


Figure 4.6: Comparison of baseline (a) and 80 kg payload (b) configurations.

4.2.1.2 Obstacle negotiation

Two configurations are evaluated for the case of a 55-millimeter-high obstacle below the front and rear wheels respectively. The comparison between simulated and real suspension for each of these scenarios is shown in Fig. 4.7, and the numerical results are collected in Table 4.9. When the robot is negotiating the obstacle with its front wheel, the chassis rotates clockwise, resulting in a negative pitch angle (-2.98 deg). When the obstacle is moved below W_6, the pitch angle increases of 3.3 degrees when compared to the baseline static configuration. In both cases, the error in the estimation of the pitch angle via the analytical model is not higher than 10%, while the maximum error for the center of gravity height is 3%.

Configuration	Variable	Analytical Model	Measurement	Relative error [%]
Obstacle W_3 (55 mm)	Z_A [mm] θ [deg]	307.9 -2.69	318 -2.98	-3.2 10.0
Obstacle W_6 (55 mm)	Z_A [mm] θ [deg]	307.8 7.31	313 8.07	-1.7 -8.8

Table 4.9: Results of obstacle negotiation configurations.

To further validate the model, a second obstacle negotiation scenario is considered, in which the robot negotiates a vertical wall. The real robot is commanded to move against a wall, Fig. 4.8 (b), and the manoeuvre is stopped at different checkpoints to take manual measurements used for comparison with the model. The results are shown in Fig. 4.9, where the simulated and real robot pitch is plotted against the vertical displacement of the idler wheel (W_2) along the wall. Each black dot represents a measurement checkpoint during the experiment. For safety reasons, the experiment is stopped when the pitch angle reaches -40 degrees, to avoid damage to the prototype. The simulation, however, carries on up to almost -90 degrees, where the only wheel in contact with the ground is the rear road wheel (W_6) and the simulated robot is about to turn over. In the range covered by the experiments, the average relative error between the simulated and measured pitch angle is 4.7%.

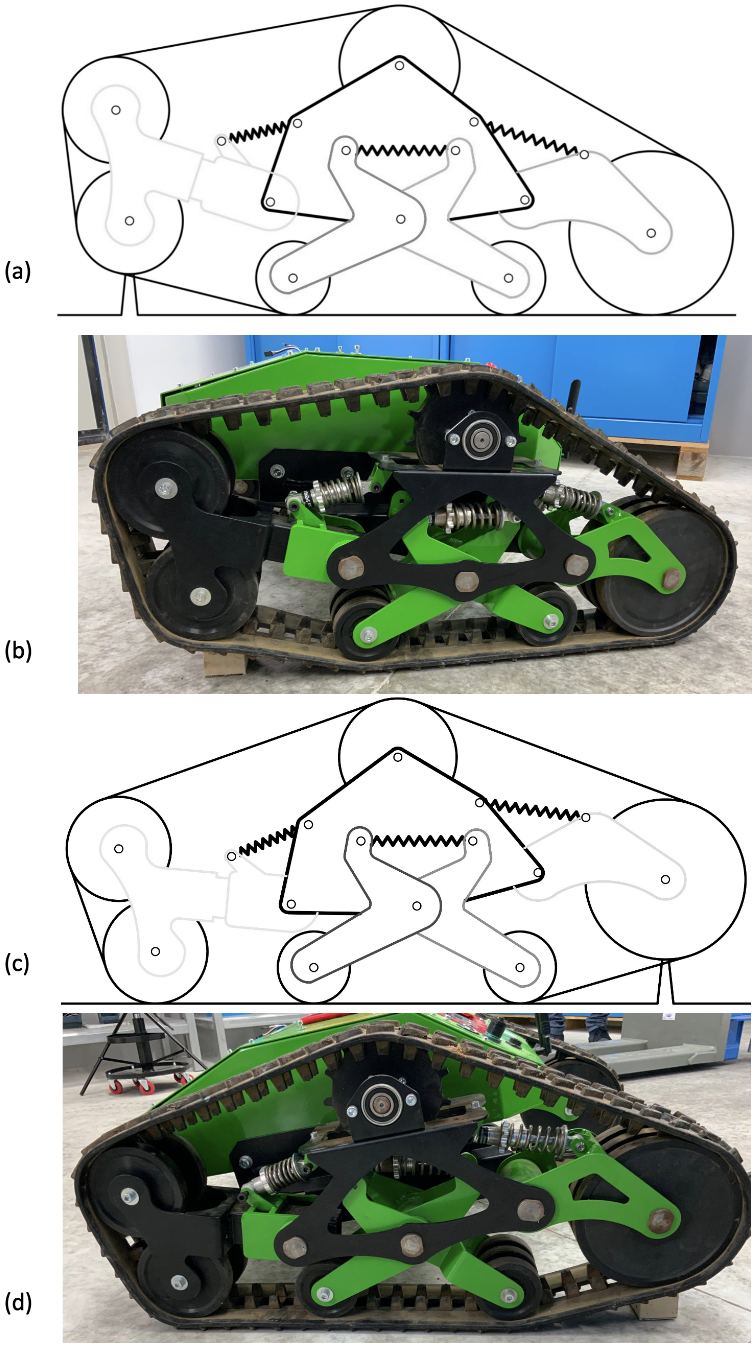


Figure 4.7: Comparison between simulated (a) and real (b) obstacle negotiation for the front wheel W_3 and between simulated (c) and real (d) obstacle negotiation for the rear wheel W_6. The obstacle is 55-mm high in all cases.

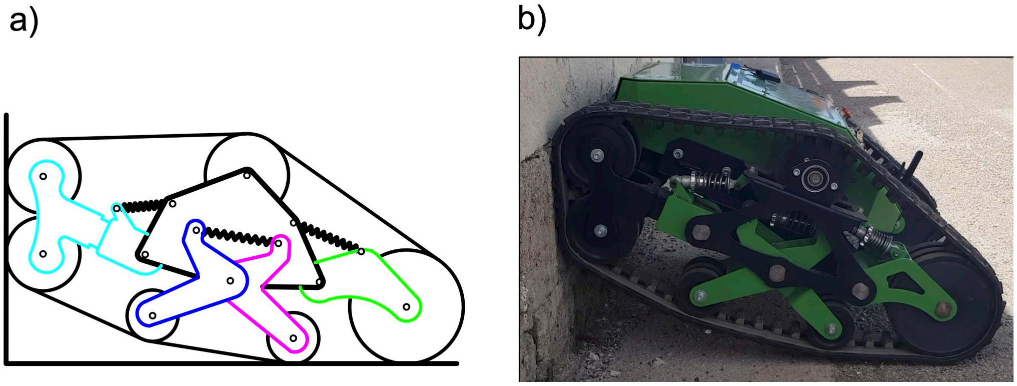


Figure 4.8: Comparison between the analytical model (a) and the real robot (b) in a wall climbing scenario.

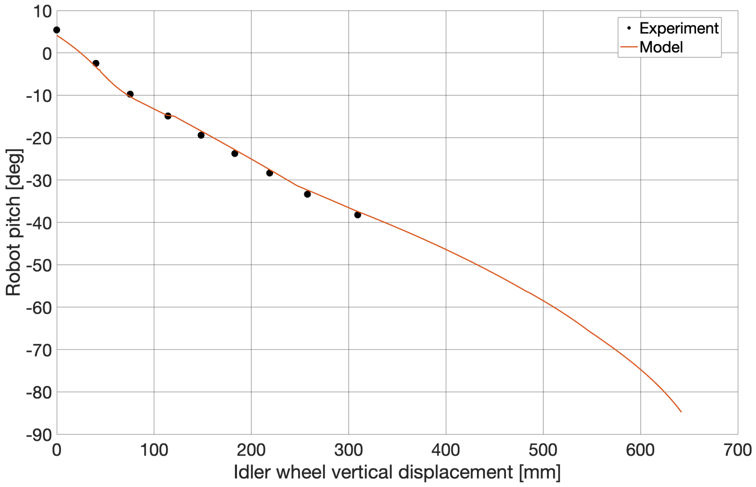


Figure 4.9: Robot pitch as a function of the vertical displacement of the idler wheel (W_2) in a wall climbing scenario: experimental measurements taken from the real robot (black dots) compared with the analytical model (red line)

4.2.2 Polibot setup analysis

Once the analytical model has been validated, it can be used in a number of ways. For example, to analyse the system behaviour in the neighborhood of the baseline configuration. In this context, three Key Performance Indicators (KPIs) are defined in this work and listed in Table 4.10. The first KPI is the pitch of the robot θ. It can be considered as a performance indicator because the objective of a suspension is to keep the sprung mass stable and as close as possible to its initial configuration. Moreover, a mobile robot might carry instrumentation for data collection that need the robot frame to be aligned with the gravity field.

The second and third KPIs are related to the capability of the suspension to distribute the robot weight across the four ground wheels. The KPI indicated with n_{cont} represents the number of wheels that make contact with the ground simultaneously. This is an important parameter to evaluate, because the suspension implemented in Polibot was designed to guarantee contact with the ground for all four road wheels to improve traction performance on all kind of terrains. Ideally, n_{cont} should always be equal to 4, but changes in design parameters or operating conditions can cause it to drop to 3. Finally, the third KPI (σ_N) is the normalised standard deviation of ground normal forces. A low value of σ_N indicates that the weight of the robot is well distributed across the four road wheels, with beneficial effects in terms of lower terrain compaction and better traction. This final KPI can be computed as follows:

$$\sigma_N = \frac{\sqrt{\frac{1}{4}\sum_{i=3}^{6}(N_i - \overline{N})^2}}{\overline{N}} \tag{4.1}$$

with:

$$\overline{N} = \frac{1}{4}\sum_{i=1}^{4} N_i \tag{4.2}$$

where N_i $(i = 3, \ldots, 6)$ is the normal reaction force below each road wheel.

First, the effect of adding mass to the system is analyzed, assuming that the position of the center of gravity is not affected. The results are shown in Fig. 4.10.

As the robot mass increases, the pitch (θ) decreases slightly. For example, an increase of 20% in robot mass causes a reduction of only half a degree in pitch

KPI	Description
θ	Robot pitch
n_{cont}	Number of wheels that make contact with the ground
σ_N	Normalised standard deviation of ground normal forces

Table 4.10: Key Performance Indicators

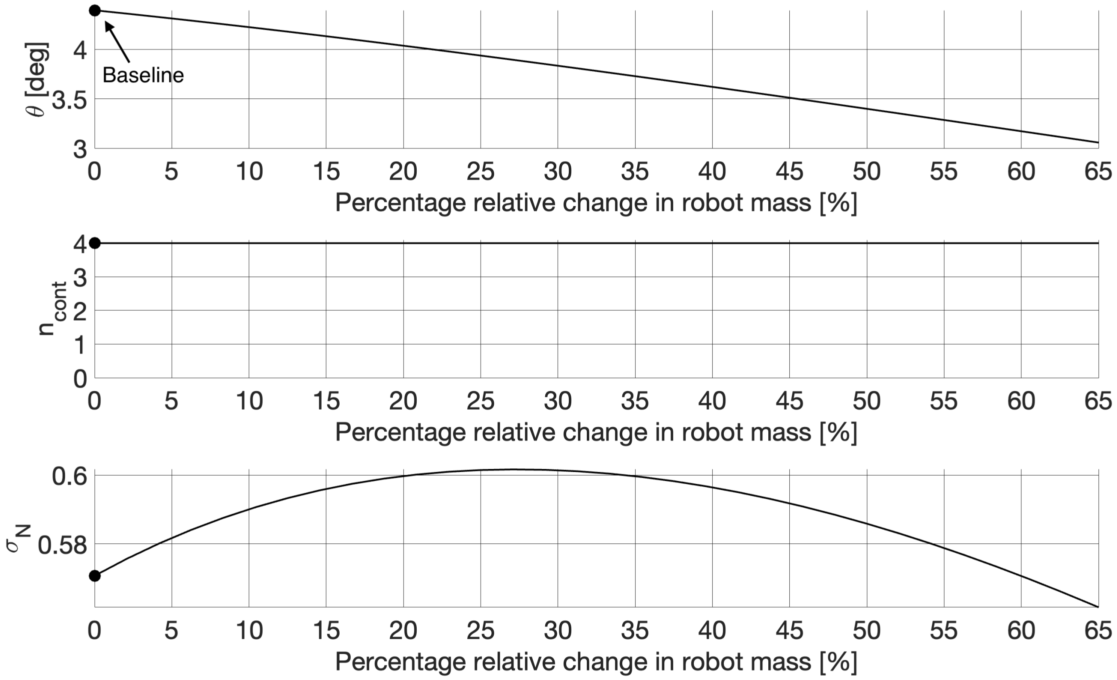


Figure 4.10: Effect of robot mass on KPIs: robot pitch (top), number of wheels in contact with ground (middle) and normalised standard deviation of ground normal forces (bottom).

angle. This result is important for the repeatability of the experiments, because the look-ahead distance of exteroceptive sensors will not be influenced by fluctuations of the robot mass that can be caused by different test setups. However, although all four wheels are always in contact with the ground, N_{std} initially increases with the robot mass, reaches a maximum, and then starts to decrease. This can be explained by the middle elastic element being more affected by the reduced robot body height associated with the increase in mass. An increase in robot mass also causes a reduction of track tension, as shown in Fig. 4.11. For a robot mass 65% higher than the baseline value, the track tension drops almost to zero, which is detrimental for traction. However, in this case, the track tension can be brought back up with the tension adjustment unit, as explained in the remainder of this section.

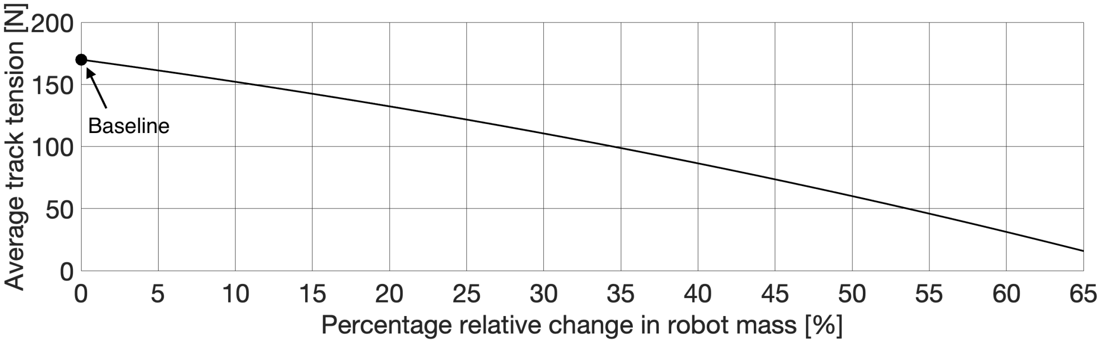


Figure 4.11: Effect of robot mass on track tension.

Two other strictly related design parameters are analyzed in this section: the track length (L_{nom}) and the length of the Front Swing Arm (FSA), indicated with d_{11} (as show in Fig. A.1 in A). The length of the FSA can be changed by the tension adjustment unit. A reduction of track length and an increase in FSA length have the same common effect to increase the tension of the track, as shown in Figures 4.12 and 4.13. Usually, the track tension is adjusted to the minimum possible to avoid power loss due to friction. However, in off-road applications the track tension should be high enough to maximize the tractive force. The baseline values of d_{11} and L_{nom} for Polibot are chosen to achieve an even weight distribution across the road wheels while keeping an adequate level of tension. Figures 4.14 and 4.15 shows the effect on the KPIs of changing track length and FSA length, respectively. As

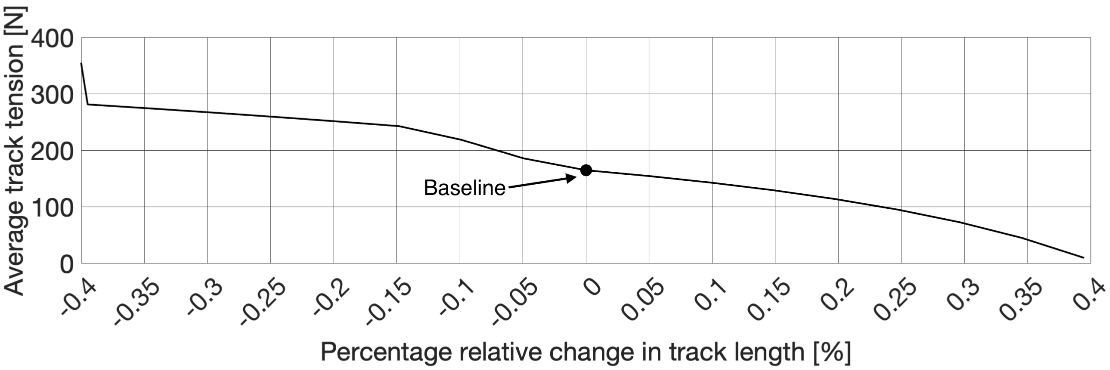


Figure 4.12: Effect of track length on track tension.

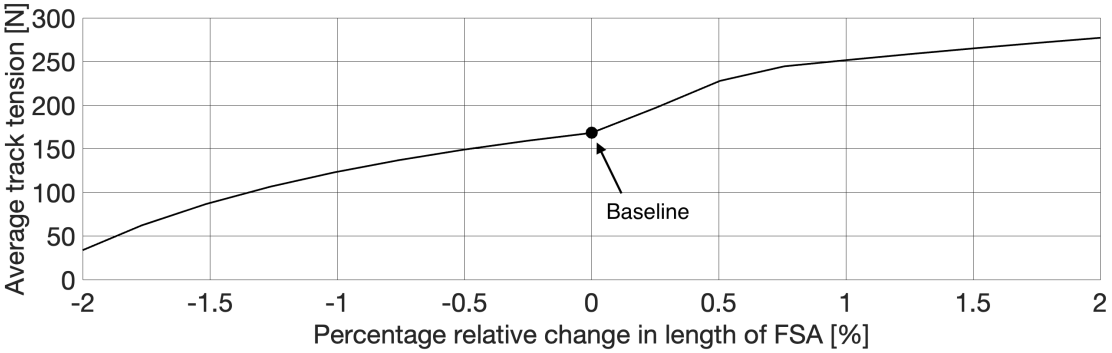


Figure 4.13: Effect of FSA length on track tension.

expected, the behaviour of the KPIs when increasing the track length is the same obtained by reducing the length of the FSA: N_{std} decreases, indicating a more uniform distribution of the normal forces below the road wheels. However, this benefit comes to the cost of reducing the track tension (Figures 4.12 and 4.13), which affects the performance of the vehicle. On the other hand, reducing the track length (or increasing the length of the FSA) has the negative effect of a less uniform weight distribution across the four road wheels. When the track length is reduced by 0.15% (or the length of the FSA increased by 0.7%) the rear wheel (W_6) is lifted from the ground due to the increased track tension, making n_{cont} drop from 4 to 3.

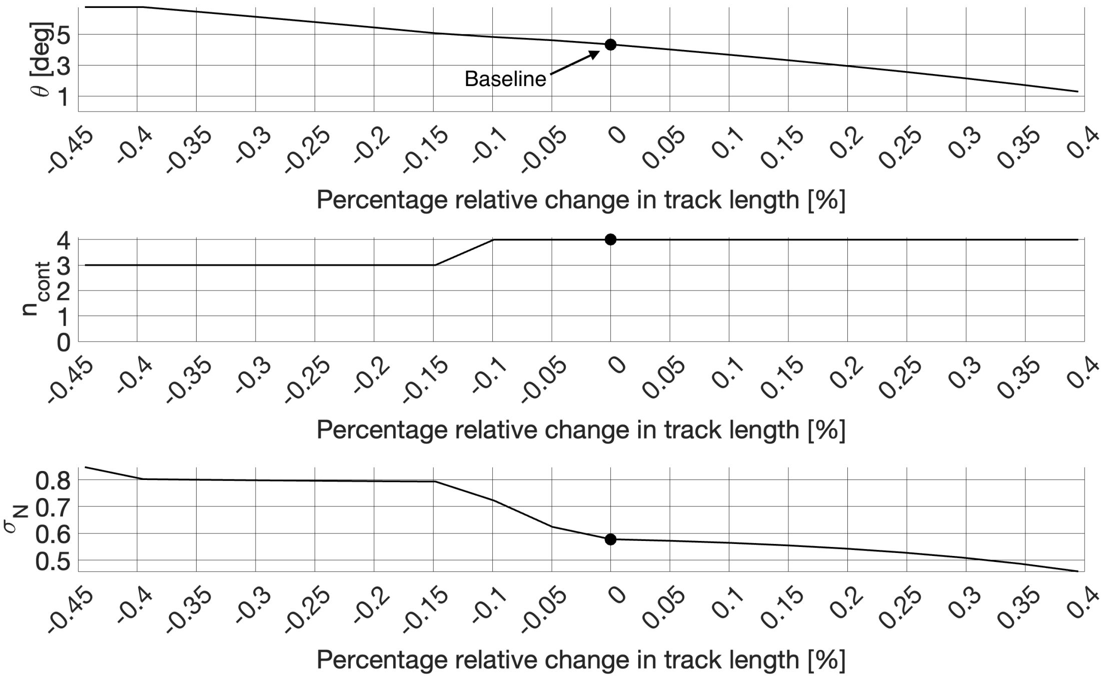


Figure 4.14: Effect of track length on KPIs: robot pitch (top), number of wheels in contact with ground (middle) and normalised standard deviation of ground normal forces (bottom).

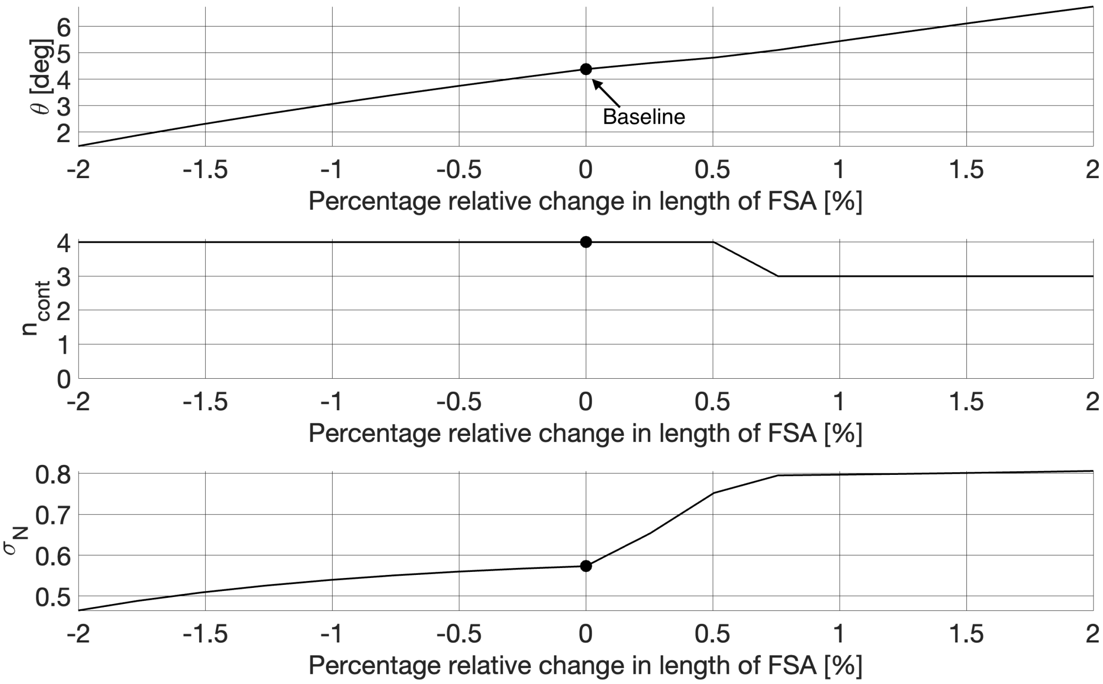


Figure 4.15: Effect of FSA length on KPIs: robot pitch (top), number of wheels in contact with ground (middle) and normalised standard deviation of ground normal forces (bottom).

4.2.3 Comparison with existing platforms

In this section, the robot proposed in [46] is considered for comparison. The suspension schematics using a bio-inspired X-shaped architecture is shown in Fig. 4.16(a) along with the corresponding kinematically equivalent model (Fig. 4.16(b)). For a fair comparision, the distance between the first and last road wheel is set to 630 mm (to match Polibot longitudinal wheelbase) and the radius of all wheels is assumed equal to the radius of Polibot first road wheel. In addition, the parameters of the springs and the track length are chosen to match Polibot initial track tension. Both types of suspension adopt an equal constant-length tensioner. The analytical model for the X-shaped suspension is discussed in D.

The two suspensions are compared in a scenario of obstacle negotiation. A step-like object is positioned below the first road wheel and its height is changed from zero to 65 mm (equal to the wheel radius). The results in terms of KPIs are shown in Fig. 4.17. As seen form this figure, the black dashed curves that represent the response of the X-shaped robot interrupt when the step height reaches approximately 35% of the wheel radius. This is because the track tension nulls out at that point, as shown in Fig. 4.18. Conversely, the tension in Polibot track is well above zero, even when overcoming an obstacle whose height matches the size of the radius of the first road wheel.

Referring again to the KPIs of Fig. 4.17, it can be noted in the middle graph (n_{cont}) that Polibot is able to keep all four wheels pushed against the ground, even for high obstacles. This is also visible in the bottom graph that shows how the two architectures distribute the vertical forces acting on the road wheels. For low obstacles (lower than about 28% of the wheel radius) the X-shaped suspension performs better, with a lower σ_N. However, for more challenging and higher obstacles, Polibot outperforms its counterpart.

As a final remark, it should be noted that Polibot offers greater mechanical simplicity. In fact, the bioinspired suspension requires seven revolute joints and six rigid links per road wheel, which means that overall the robot of Fig. 4.16 would require twenty eight revolute joints and thirty links per track, against the nine joints and five links used by Polibot.

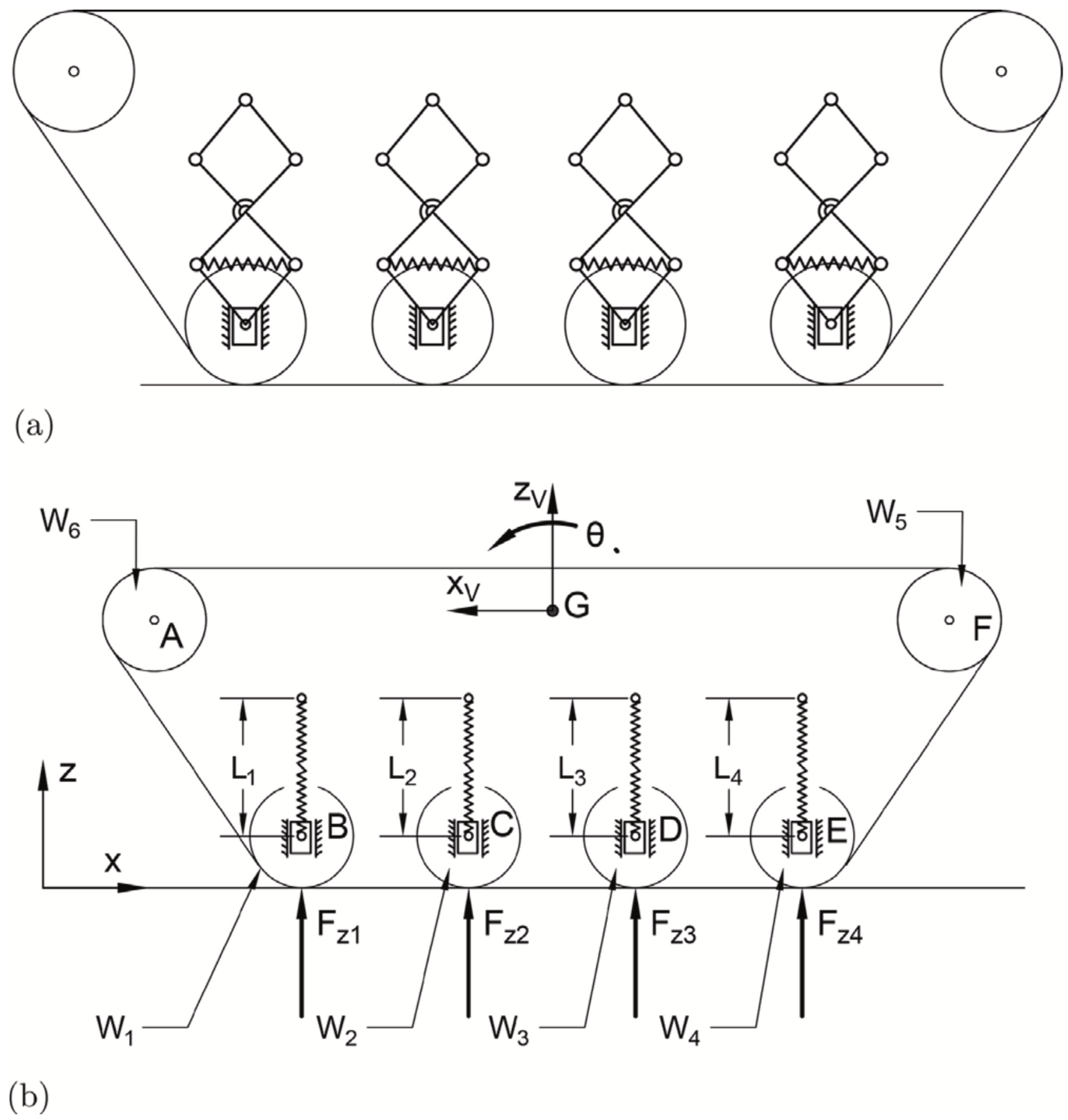


Figure 4.16: The existing X-shaped passive suspension proposed by Sun et al. used for comparison with Polibot: (a) original scheme, (b) kinemetically equivalent model.

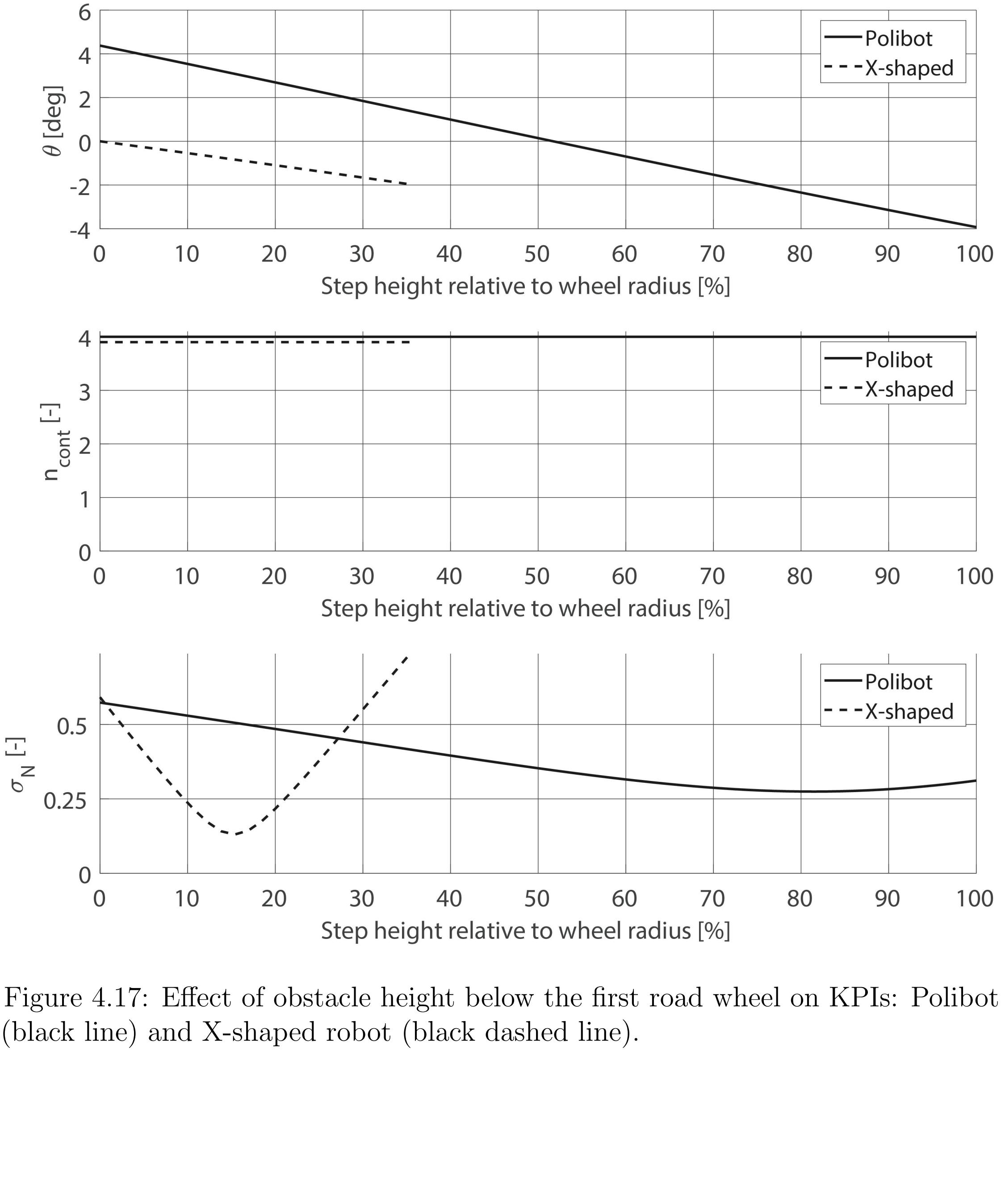


Figure 4.17: Effect of obstacle height below the first road wheel on KPIs: Polibot (black line) and X-shaped robot (black dashed line).

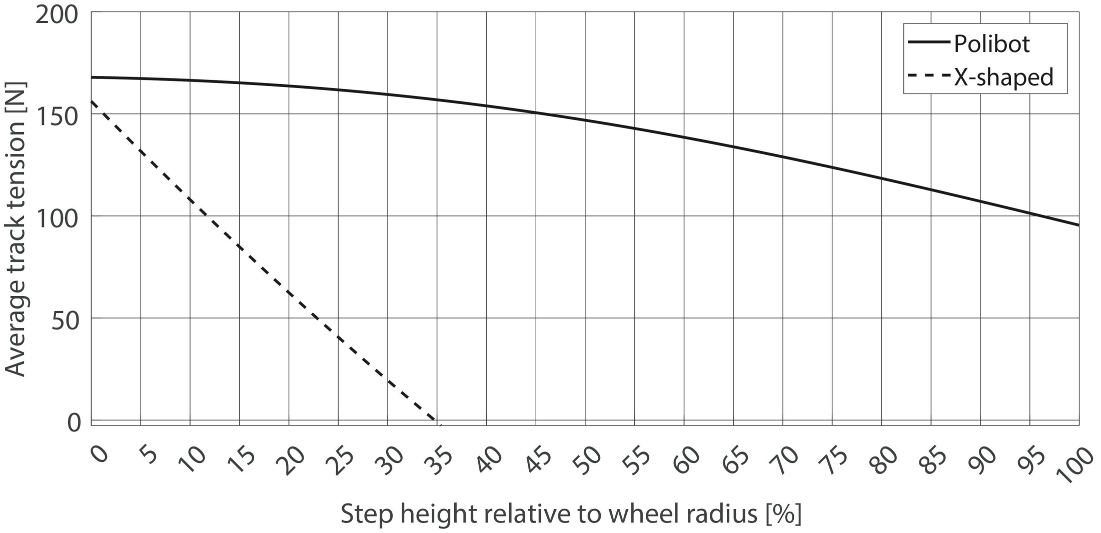


Figure 4.18: Effect of obstacle height below the first road wheel on track tension: Polibot (black line) and X-shaped robot (black dashed line).

4.2.4 Multibody model validation

The simulations to validate the MB model and the comparison against experiments on the actual prototype are presented in this section. First, static tests are performed to evaluate the inertial properties of Polibot. Then, the dynamic response of the model in different tests is compared against the experimental data gathered in the field.

4.2.4.1 Static tests

This section describes the procedure followed to identify the position of the center of gravity with respect to a reference system located at the center point of an imaginary line connecting the centers of the two sprockets, as indicated in Figures 4.19(a) and 4.20(a).

A first test is designed to estimate x_G, that represents the x coordinate of the center of gravity, using the suspension and ground reaction method. Note that the procedure here described is usually applied to wheeled vehicles, while being controversial for tracked vehicles, for which the ISO Standard 789-6 prescribes the use of decking and knife edges. The objective is to suspend the robot in such a way that only the four front and rear wheels are exchanging forces with the ground, to

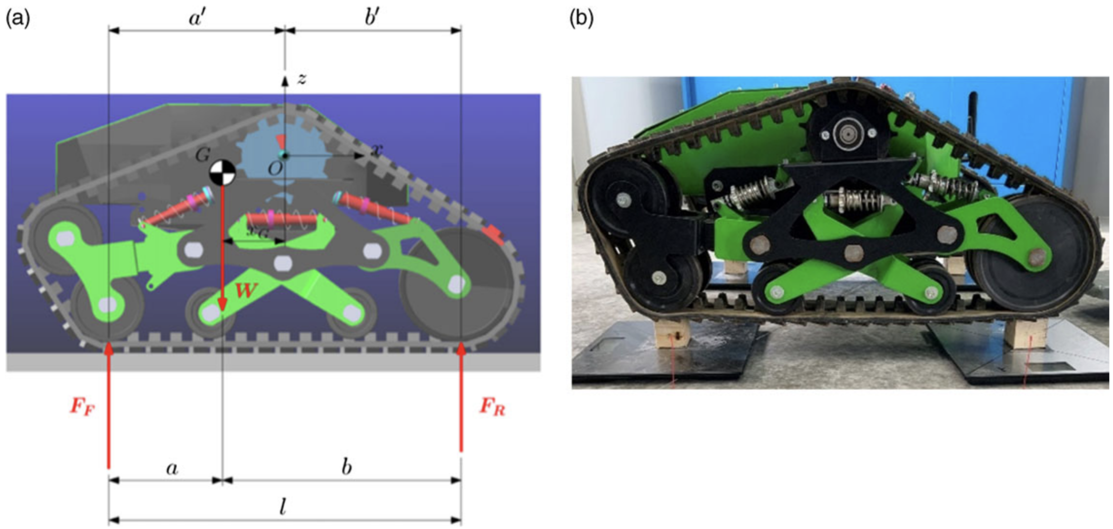


Figure 4.19: Procedure for the determination of the x coordinate of the center of gravity: model (a) and real robot (b).

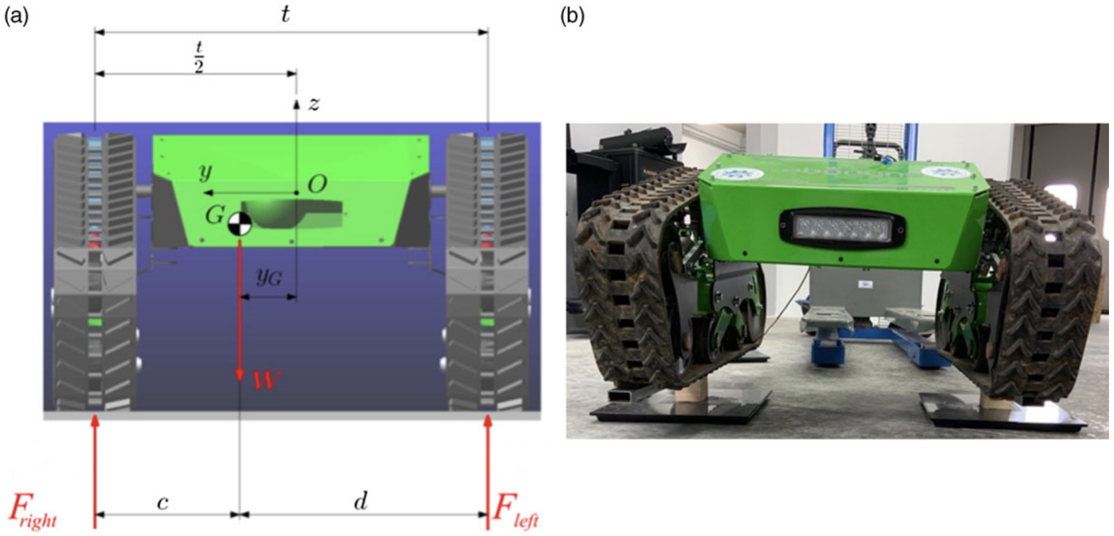


Figure 4.20: Procedure for the determination of the y coordinate of the center of gravity: model (a) and real robot (b).

measure F_F and F_R, that indicate the fractions of the rover weight that loads the front and rear wheels respectively, as shown in Fig. 4.19(a).

To achieve this result, after locking the suspension to avoid internal configuration variations, the rover is lifted onto two bars, with the front (left and right) wheels resting on the first bar and the rear wheels on the second one, to emulate the front and rear axle of a vehicle. Each bar is placed on a scale, and once F_F and F_R are known, x_G can be estimated as follows:

$$x_G = b' - \frac{F_F \cdot l}{F_F + F_R} \tag{4.3}$$

where W is the robot weight, b' and l are geometric parameters as indicated in Fig. 4.19(a).

A similar procedure is implemented for the estimation of the y coordinate of the center of gravity, y_G. The robot is settled onto two bars, but in this case the bars are placed parallel and underneath each track, to measure F_{left} and F_{right}, that represent the fractions of the rover weight that loads the left and right tracks respectively, as shown in Fig. 4.20(a). Each bar is placed on a scale, and once F_{left} and F_{right} are known, y_G can be estimated as follows:

$$y_G = \left(\frac{1}{2} - \frac{F_{left}}{F_{left} + F_{right}}\right) t \tag{4.4}$$

where t is the distance between the tracks, as shown in Fig. 4.20(a).

A final procedure is designed to estimate z_G, that represents the z coordinate of the center of gravity. With the suspension motion locked up, the robot front axle is jacked up so that the lower parts of the tracks are at an angle, α, with the horizontal. The rear wheels are positioned on scales, while the front axle is raised as shown in Fig. 4.21. Once the load on the rear wheels is measured, z_G can be measured as follows:

$$z_G = R_F + \frac{\frac{F_r \cdot l_1}{\cos\alpha} - W \cdot a}{W \tan\alpha} - h_O \tag{4.5}$$

where W is the rover weight previously measured, F_R is the weight on the rear wheels with front elevated, h_O is the distance between the center of the sprocket and the bottom part of the track, while a and l_1 are geometric parameters, as

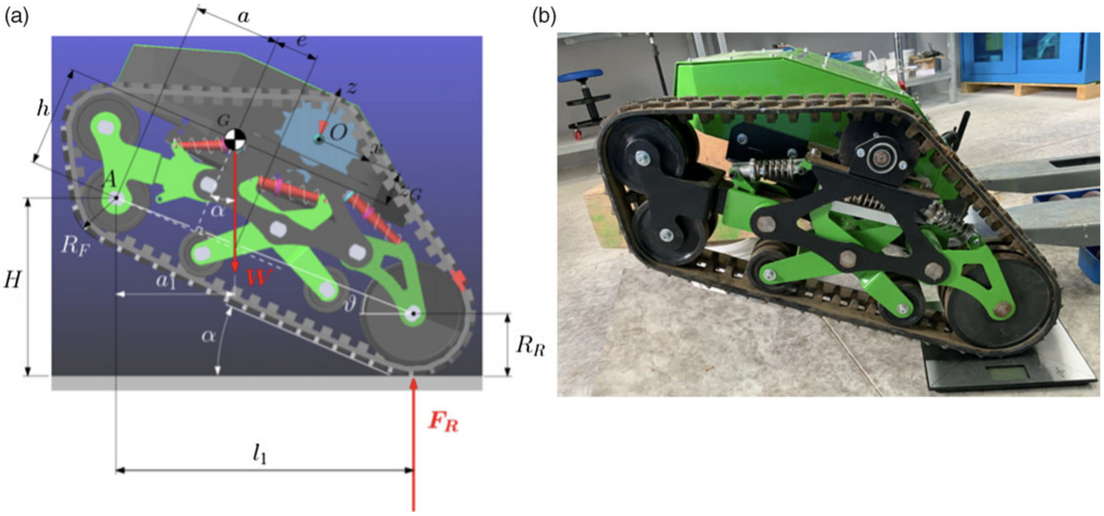


Figure 4.21: Procedure for the determination of the z coordinate of the center of gravity: model (a) and real robot (b).

indicated in Fig. 4.21(a).

The procedures described in this section have been applied to Polibot prototype, leading to the results shown in the second column of Table 4.11. Also, the overall robot weight is 97 kg. The resulting y coordinate is not equal to zero, implying a slight asymmetry of the system with respect to the xz plane. However, 8 millimeters correspond to only 1.3% of the distance between the tracks, and it could be justified with manufacturing tolerances and inaccuracy of the measurements of the scales.

Coordinate	Empirical value [mm]	Simulated value [mm]	Error [%]
x_G	-52	-56	7
y_G	8	0	-
z_G	-100	-99	1

Table 4.11: Empirically estimated and simulated position of the center of gravity of Polibot.

Similar tests are now performed on the MB model of Polibot as shown in Figs. 4.19(b), 4.20(b) and 4.21(b). Note that each subsystem has been designed to closely match the prototype counterpart in terms of geometry and materials. This modelling effort leads to the results shown in the third column of Table 4.11. As expected, the simulated y coordinate of the center of gravity is zero, as the model is perfectly symmetric. The x and y coordinates differ by 7% and 1% respectively between the empirical and simulated values. The overall mass of the modeled rover is 97 kg, which exactly matches the actual weight of the prototype.

To complete the set of inertial properties, the MB model is used to compute the mass inertia tensor. These values are not validated against experimental estimations because of the difficulty of measuring moments of inertia. However, as illustrated in Fig. 3.18, special care has been devoted to importing into MSC Adams the Parasolid files of the real CAD geometries and assigning them to the corresponding rigid parts of the different subsystems: this not just for visualization purposes, but also, most importantly, as a way to place the parts, together with their material and density, in the correct relative position between each other. Consequently, not only the model mass and position of the center of gravity are, as experimentally observed, close to those of the real prototype, but also the mass inertia tensor is expected to be by only a few percentage points separated from the true one. The simulated moments of inertia are shown in Table 4.12.

The results presented in this section demonstrate that the inertial properties of the modeled robot are satisfactory, and the validation of the MB model can move to the next phase with dynamic tests.

Moment of inertia	Value [kg m^2]
I_{xx}	8.1
I_{yy}	6.3
I_{zz}	10.1

Table 4.12: Empirically estimated and simulated position of the center of gravity of Polibot

4.2.4.2 Dynamic tests

Field tests were conducted in Candiolo, close to Turin in the North of Italy, to validate the dynamics of the MB model of Polibot. The prototype was fitted with a sensor frame that mounted GPS receivers, IMU, and stereocameras, as shown in Fig. 4.22. Sensory data are recorded and stored by the robot. During field tests, the robot was teleoperated to move across different types of surface. It is commanded by means of PWM signals that modulate the torque delivered by the two electric motors (one per each sprocket). The actual rotational speed of the

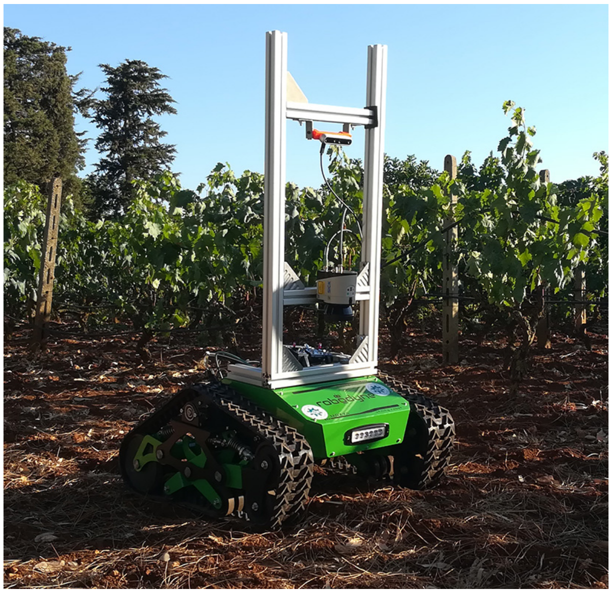


Figure 4.22: Polibot in action in the test site of Candiolo, Italy.

motors is then measured during the tests with the use of encoders. The measured rotational speeds are finally provided as input to the multibody model for the simulation, while the rest of the data gathered is used for validation purposes.

It should be noted that the virtual shock absorbers of the digital twin are set to replicate the exact properties of the real devices. Therfore, the linear stiffness coefficient is set to 12 N/mm, while the linear damping coefficient is equal to 0.8 Ns/mm. The preloads of the front, middle and rear shock absorbers are, respectively, 180 N, 130 N and 0 N.

The first surface considered for validation is a track of rigid strips, shown in Fig. 4.23. It was chosen to evaluate the vertical dynamic response of the robot. The simulation is run on a hard soil that matches the geometry of the actual strips, which are of different heights, ranging between 65 and 125 mm. The strips are 60 mm wide (in the direction of motion), and the space between two consecutive strips is 100 mm. Figure 4.23 describes the geometry of the first five strips. In the case of hard soil simulation, the forces that arise from the contact between the track segments and the soil are calculated according to a penalty-based method as:

$$F_n = K \cdot \delta^e - STEP(0, 0, \delta_{max}, c_{max}) \cdot \dot{\delta} \tag{4.6}$$

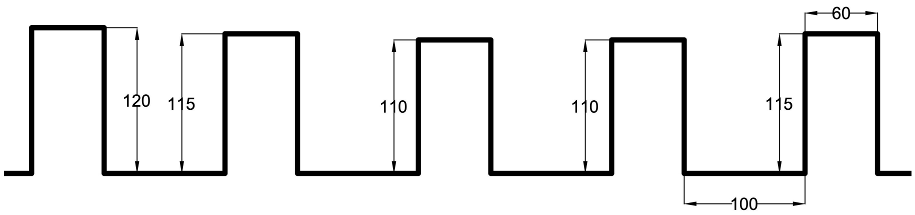


Figure 4.23: Track of strips used for dynamic validation of the model: detail of the first five strips with dimensions in millimeters.

where K is the stiffness modeling the elasticity of the surfaces of contact, δ represents the penetration between the colliding bodies, e is a positive real value denoting the force exponent. The second addend of the right hand side of Eq. 4.6 accounts for the energy dissipation that is a function of penetration velocity $\dot{\delta}$. The step function is defined as follows:

$$STEP(0, 0, \delta_{max}, c_{max}) = \begin{cases} 0 & \text{if } \delta \leq 0 \\ c_{max}\left(\frac{\delta}{\delta_{max}}\right)^2\left(3 - \frac{2\delta}{\delta_{max}}\right) & \text{if } 0 < \delta < \delta_{max} \\ c_{max} & \text{if } \delta \geq \delta_{max} \end{cases} \tag{4.7}$$

and serves to smoothly ramp up the damping coefficient from zero, when the penetration is zero, to c_{max} when the penetration reaches the boundary value δ_{max}, thus preventing the damping force from being discontinuous at the onset of the contact. A high stiffness K for the contact between the track segments and the ground allows excessive interpenetration to be avoided between the road and the robot, but at the same time can lead to integrator convergence issues. This parameter usually ranges from 200 N/mm up to 2.0×10^8N/mm: a value $K = 2.0 \times 10^5$ N/mm is chosen as a satisfactory trade-off between accuracy and simulation time. It is advisable the force exponent to be different from 1.0, preferably higher than 1.5: $e = 2.0$ is set to have a sufficiently smooth function that is with a continuous first-order derivative. The damping coefficient is assumed to be equal to one percent of the stiffness coefficient, i.e., $c_{max} = 2.0 \times 10^3$ Ns/mm, while the reasonable value $\delta_{max} = 1$ mm is assumed for the penetration depth at which full damping is applied.

For the tangential interaction between the track segments and the ground a modified Coulomb friction model [82] is adopted that requires to specify the static and the dynamic friction coefficients, μ_s and μ_d, together with the stiction and friction transition velocities, v_s and v_d. The static friction coefficient is set to 0.9, while the dynamic one is equal to 0.7. These values have been shown to minimize the error between experimental testing and numerical simulation, thus confirming the expected high adherence of the rubber track on dry concrete and asphalt on which the test campaign was conducted. To avoid integrator difficulties, the stiction transition velocity should preferably be higher than the integrator error used for the solution, while the friction transition velocity should be higher than five times the error. As reducing both parameters shows to have negligible impact on the results in the face of a considerable increase in computation effort, the default values v_s=0.1 m/s and v_d=0.5 m/s are adopted.

The results of the simulation are compared against the corresponding field test and discussed in this section. Fig. 4.24 shows the angular velocities of the motors measured by the encoders while Polibot was moving over the track of rigid strips. The test was repeated twice at two different speeds (0.1 m/s and 0.2 m/s). The results are presented in terms of measured and simulated vertical acceleration in Fig. 4.25. It can be noted that the simulated vertical acceleration is qualitatively similar to the real one for both speeds. Quantitatively, the Root Mean Square (RMS) of the simulated and experimental vertical accelerations is shown in Fig. 4.26. In the low-speed case, the simulated acceleration has an RMS of 0.049 m/s^2, against a measured value of 0.052 m/s^2, with an error of 5.7%. In the high-speed case, the root mean square of the simulated vertical acceleration is 0.100 m/s^2, while the measured value is 0.103 m/s^2, with an error of only 3%.

The second test considered for validation is a manoeuvre on asphalt. It should be noted that in practice the motion primitives of Polibot are limited to straight line driving and turn-on-the-spot steering. This choice was made considering the large amount of slippage incurred by the robot during steering that makes its control extremely difficult for general turning maneuvers. Therefore, the test manouvre consists in a straight line movement followed by two 90-degree turns. Due to the nature of the terrain traversed, it was chosen to run the simulation of the model on flat, hard soil. Figure 4.27 shows the angular velocities of the motors measured by the encoders during the test. The Polibot prototype is not fitted with a load cell

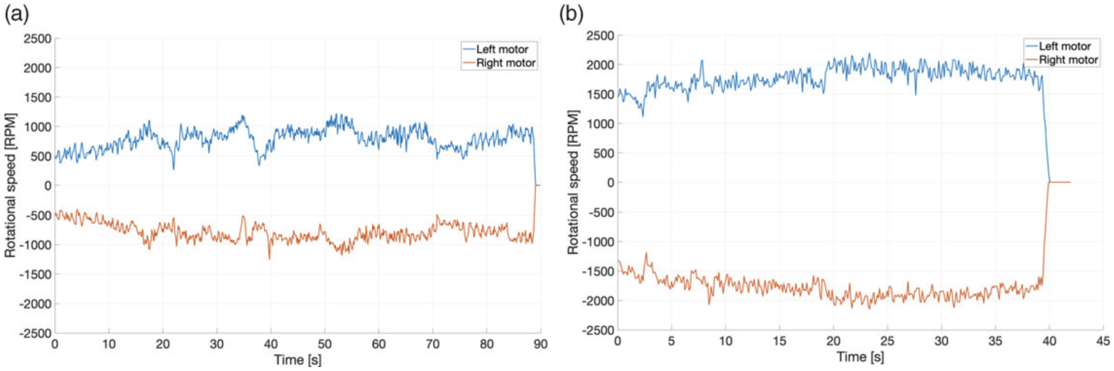


Figure 4.24: Angular velocities measured by the encoders of the two motors while Polibot is moving over the track of rigid strips at 0.1 m/s (a) and 0.2 m/s (b).

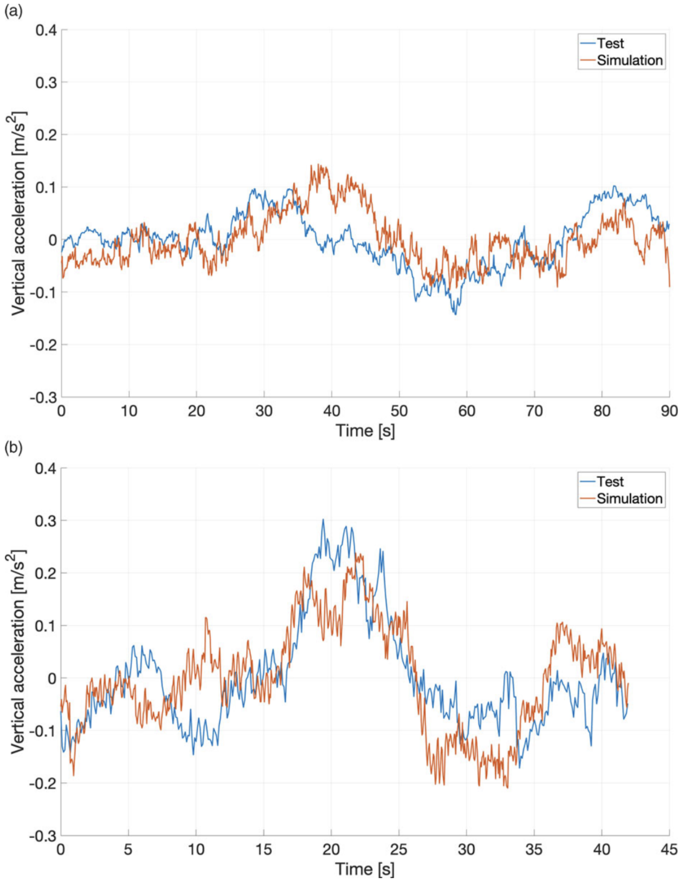


Figure 4.25: Vertical acceleration measured and simulated while Polibot is moving over the track of rigid strips at 0.1 m/s (a) and 0.2 m/s (b).

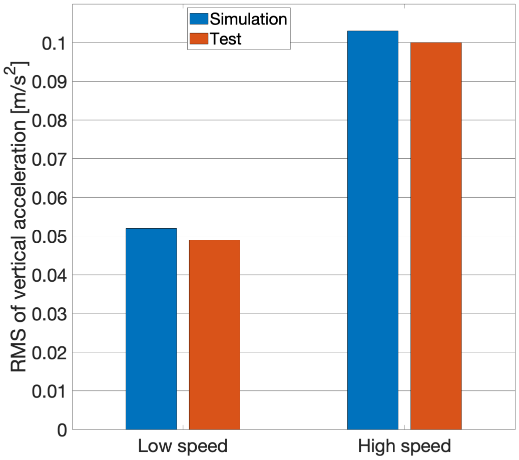


Figure 4.26: Comparison of root mean square of simulated and experimental vertical accelerations of Polibot while moving over the track of rigid strips.

to measure the drive torque of the two motors. However, for validation purposes, the sum of the two driving torques during the straight line manoeuvre is estimated from the measured overall battery drain current as follows:

$$T_{sum} = \eta \frac{C \cdot V}{\omega_{mean}} \tag{4.8}$$

where C is the measured overall battery drain current, V is the battery voltage, ω_{mean} is the average between left and right motor speeds, η is the average motor efficiency provided by the manufacturer. This estimated overall torque is compared against the sum of the simulated right and left motor torques in Fig. 4.28, showing good agreement with a RMS error of 0.0684 Nm and a RMS percentage error of 8.02%.

The corresponding trajectory, as measured by the GPS, is shown by the red line in Fig. 4.29 and it is compared against the simulated trajectory, in blue. The initial absolute heading is measured by the IMU and it is also provided as input to the model. The overall distance covered by the robot during the test on asphalt, taken from the origin of the frame of Fig. 4.29, is $d_{mea} = 23.88$ m. The percentage deviation of the simulation from the measured data, in terms of final position, is calculated as:

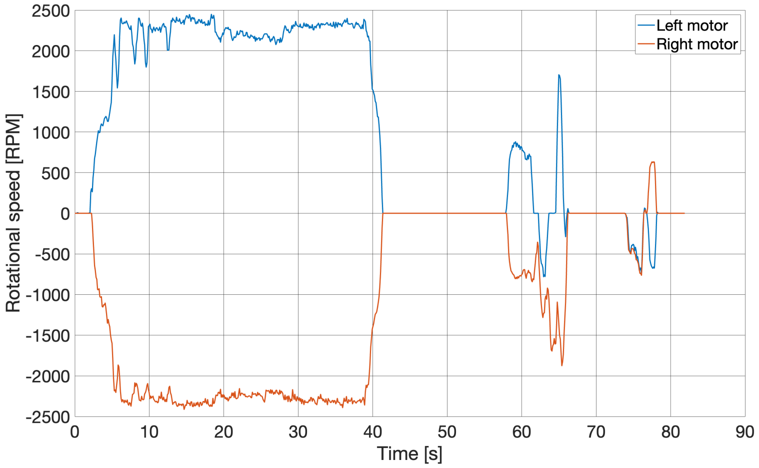


Figure 4.27: Angular velocities measured by the encoders of the two motors while Polibot is moving on asphalt.

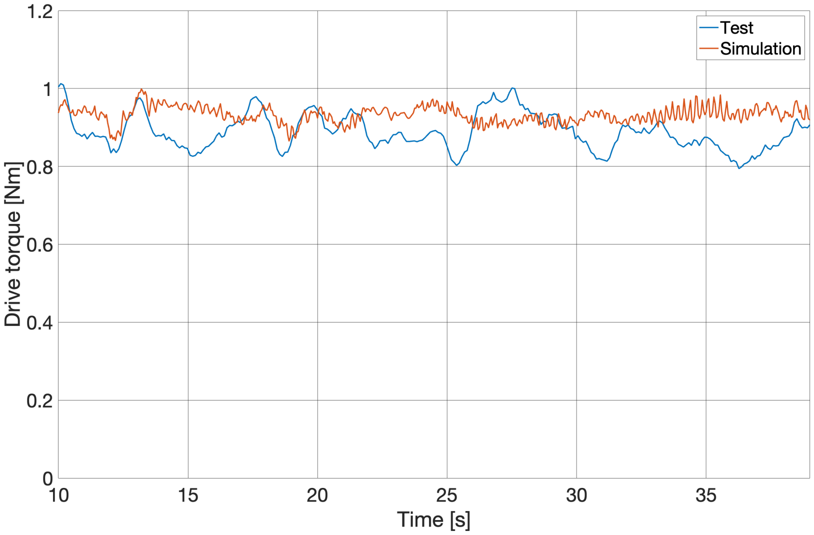


Figure 4.28: Comparison between estimated and simulated overall motor torque during straight line.

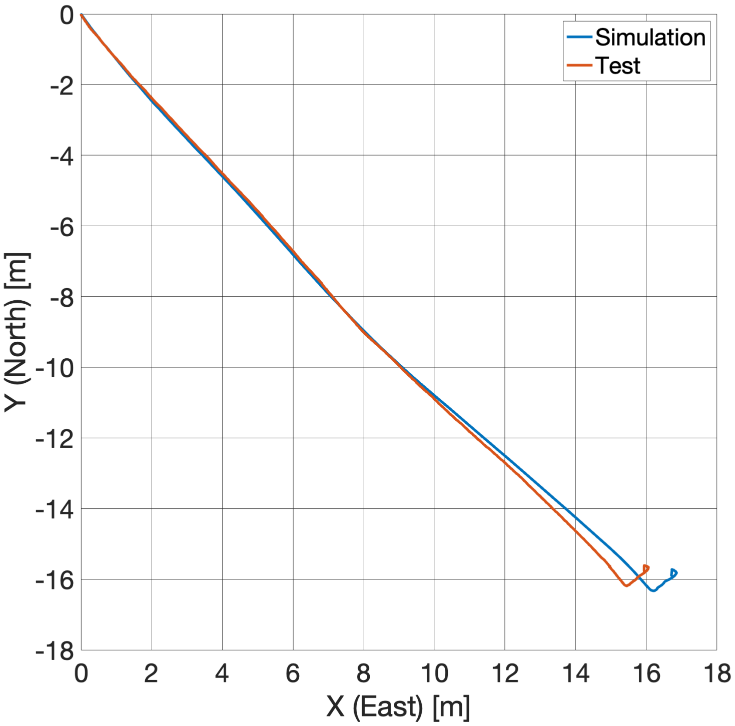


Figure 4.29: Comparison between measured and simulated path followed by the robot during the validation test on asphalt.

$$E = 100\frac{d}{d_{mea}} \tag{4.9}$$

where d is the distance between the final position of the real and the simulated path. Similarly, the following metric is adopted to evaluate the average local position relative percentage error between the real and the simulated robot:

$$E_p = 100\sqrt{\frac{1}{n_t}\sum_{k=1}^{n_t}\frac{|r_{mea,k} - r_{sim,k}|^2}{|r_{mea,k}|^2}} \tag{4.10}$$

where $r_{mea,k}$ is the measured position vector at time step k, $r_{sim,k}$ the simulated position vector at time step k, n_t the number of time steps, and $|\cdot|$ represents the magnitude of the corresponding vector.

The final metric considered for validation is the root-mean-square error of the simulation from the measured data in terms of absolute heading of the robot, calculated as:

$$E_{\psi} = \sqrt{\frac{1}{n_t}\sum_{k=1}^{n_t}\left(\psi_{mea,k} - \psi_{sim,k}\right)^2} \tag{4.11}$$

where $\psi_{mea,k}$ is k-th absolute heading angle measured by the real robot and $\psi_{sim,k}$ is the corresponding simulated value. The resulting value for the final position deviation is $E = 3.25\%$, while the average local position relative error is $E_p = 6.23\%$. The heading deviation is $E_{\psi} = 2.30°$.

Finally, a curved path on asphalt is considered where Polibot is commanded to follow a circular path of 30 m radius with a travel speed of about 0.65 m/s. The experimental path as obtained from the GPS is overlaid over the Google Earth® map of the test track in Fig. 4.30 as a dashed red line. Again, the experimental angular speed of the two drive sprockets is used as input to the MB model. The corresponding simulated path is marked with a solid blue line. As seen from this figure, the experimental and virtual paths are in good agreement with an average local position relative error and heading RMS error, respectively, of $E_p = 7.41\%$ over 65 m of total travel distance and $E_{\psi} = 5.66°$.

As a final remark, Fig. 4.31 clarifies how the contact forces are estimated in the MB model. For a given integration step, forces are calculated at each contact point. Individual contributions distributed over the contact area are then summed up to compute the net normal and tangential force originated by the contact event, as shown in Fig. 4.31.

The results presented in this section demonstrate that the behavior of Polibot can be replicated by the MB model with a good level of accuracy. This terminates the validation phase of the model, that can now be used to simulate challenging scenarios, avoiding wear of the real Polibot and costly and time-consuming field experiments.

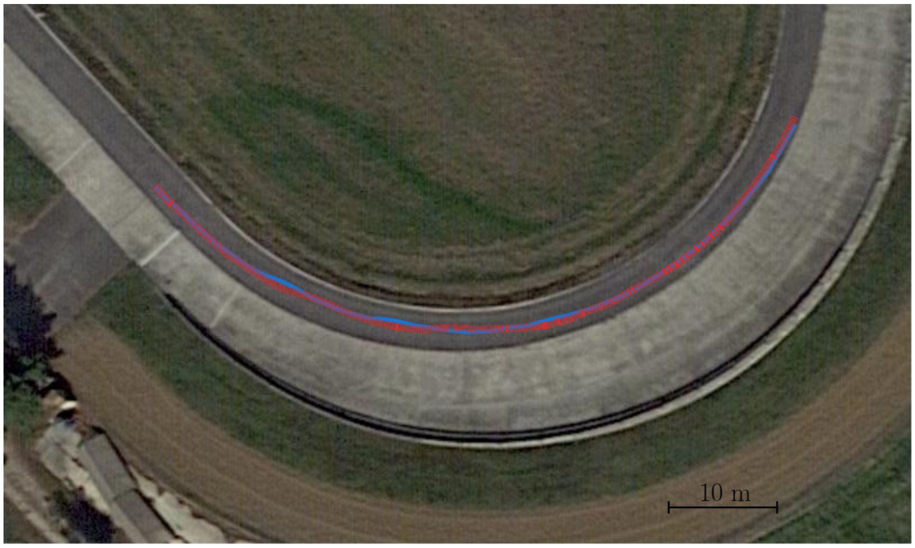


Figure 4.30: Comparison between measured (dashed red line) and simulated path (solid blue line) followed by the robot during the validation test on a curved path. GPS coordinates of the test site (DMS format): 44° 57' 24.5988" N, 7° 33' 25.956" E).

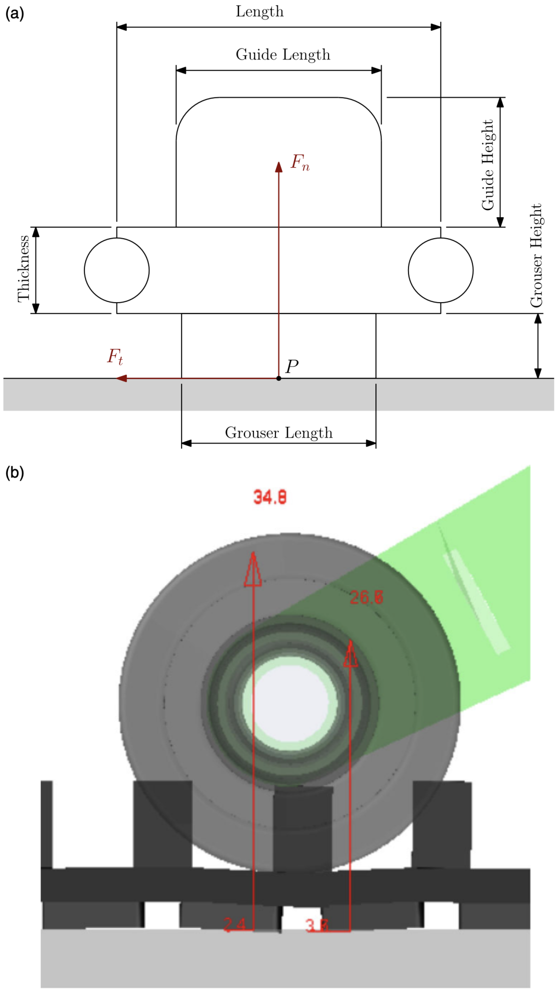


Figure 4.31: Contact forces between track belt elements and ground: (a) schematic of a single track belt segment, (b) calculation example of contact forces (expressed in N) under the left bogie wheel.

4.2.5 Simulation of highly challenging environments

In this section, the validated MB model is used to simulate four challenging environments and to test the effectiveness of the innovative suspension of Polibot. In the challenging scenarios shown in Table 4.13, the robot will try to negotiate a sinusoidal bump (Section 4.2.5.1), overcome a ditch (Section 4.2.5.2), fall from a step (Section 4.2.5.3) and traverse uneven terrain (Section 4.2.5.4).

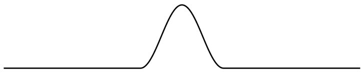
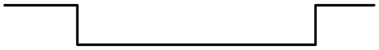
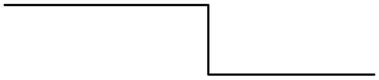
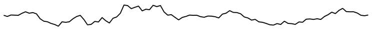
Environment	Sketch
Sinusoidal bump negotiation	
Ditch negotiation	
Falling from a step	
Stochastic uneven	

Table 4.13: Environments considered for testing the multibody model.

Each environment is simulated in half vehicle mode to take advantage of the longitudinal symmetry and reduce the computational time.

For each scenario, the simulation is repeated in two conditions: one in which the robot is not suspended and the road wheels are fixed with respect to the chassis (locked suspension case); the second, in contrast, with the passive suspension regularly in operation. To make a fair comparison between the two cases of activated and deactivated suspension, the dynamic simulations are preceded by a static analysis to evaluate the tensioner setup length that achieves an initial track tension of 150 N which is considered optimal for traction and similar to the real initial track tension of the prototype.

4.2.5.1 Sinusoidal bump

It is simulated the negotiation of a sinusoidal bump, whose profile is defined as:

$$z(x) = \frac{H}{2}\left[1 + cos\left(2\pi\frac{x - x_0}{W}\right)\right] \tag{4.12}$$

where $H = 35$ mm and $W = 70$ mm refer, respectively, to the width and height of the obstacle, and x_O locates the position of the center of the obstruction along the direction of travel x. The 3D geometry of the bump is provided to the software through a Road Data File (.rdf) that contains the points and nodes of the road profile (see Fig. 4.32).

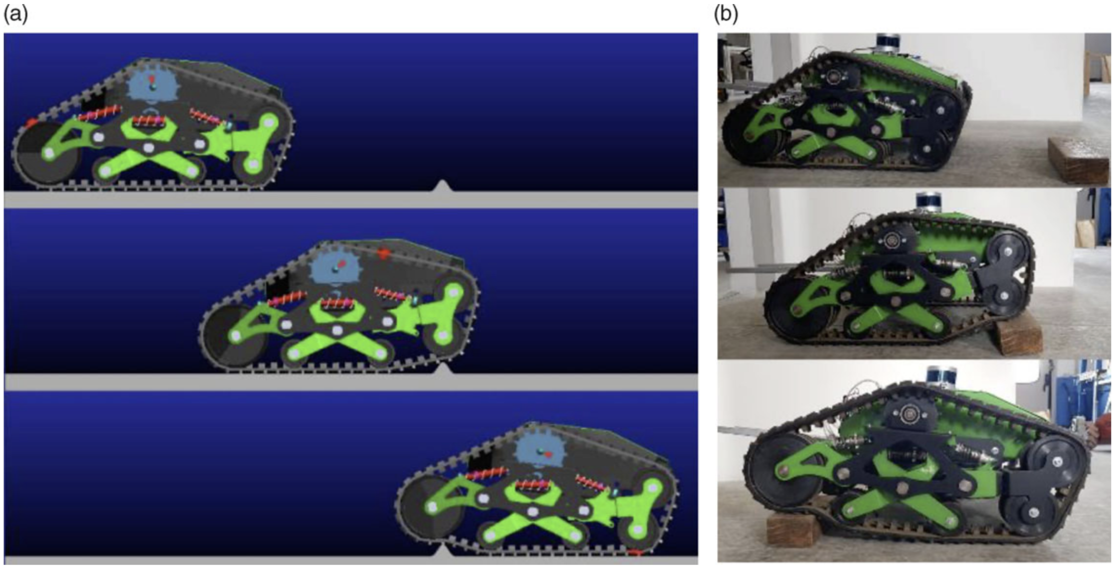


Figure 4.32: Bump negotiation: (a) simulated Polibot, (b) real prototype.

As an initial test, the results obtained from the real robot and the multibody model are compared as shown in Fig. 4.33 considering a constant speed of 0.25 m/s. The agreement of the two curves is fairly good. When looking at the acceleration root-mean-square values, the relative error results of 4.9%, further attesting to the validation of the digital twin. Afterwards, a simulation is performed with the vehicle that moves at a constant velocity of 0.55 m/s, corresponding to an imposed angular velocity to the sprocket of 376 deg/s. The simulation starts with the robot and the obstacle separated by 900 mm of flat road; the impact happens after about 1.5 s and the vehicle starts its climb.

Figure 4.34 depicts the displacement and acceleration of the hull in the bounce (vertical) direction with unlocked and locked suspension for the case of forward speed of 0.55 m/s. In Table 4.14, it can be clearly seen that, when the suspension is locked, the vehicle experiences higher bounce vibrations, as highlighted by the acceleration root-mean-square values of 2.67 m/s^2 and 1.36 m/s^2 for the configurations with

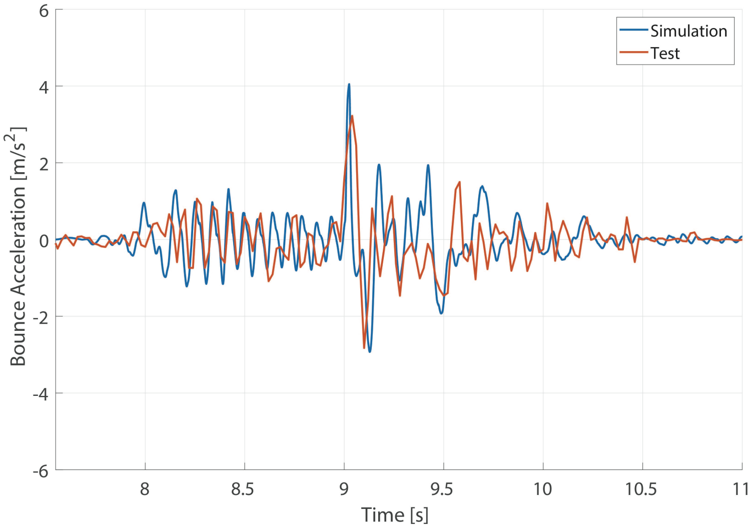


Figure 4.33: Comparison between measured and simulated bounce response for the bump negotiation test with a travel speed of 0.25 m/s.

locked and active suspension, respectively. In Fig. 4.35 the pitch motion is analyzed, from which we see that the chassis experiences significantly reduced angular acceleration when the suspension system is unlocked: the peak-to-peak and root-mean-square values are 2.52E+03 deg/s^2 and 211.80 deg/s^2 respectively, as opposed to 7.27E+03 deg/s^2 and 385.73 deg/s^2 obtained from the ride simulation with locked suspension (see Table 4.14).

Peak-to-peak				
Suspension	BV [m/s]	BA [m/s^2]	AV [°/s]	AA [°/s^2]
Locked	0.41	35.81	91.79	7.27E+03
Unlocked	0.40	21.30	93.54	2.52E+03
Root-mean-square				
Suspension	BV [m/s]	BA [m/s^2]	AV [°/s]	AA [°/s^2]
Locked	5.07E-02	2.67	11.22	385.73
Unlocked	4.66E-02	1.36	10.94	211.80

Table 4.14: Results obtained from the bump negotiation simulation (0.55 m/s). BV = Bounce Velocity, BA = Bounce Acceleration, AV = Angular Velocity, AA = Angular Acceleration.

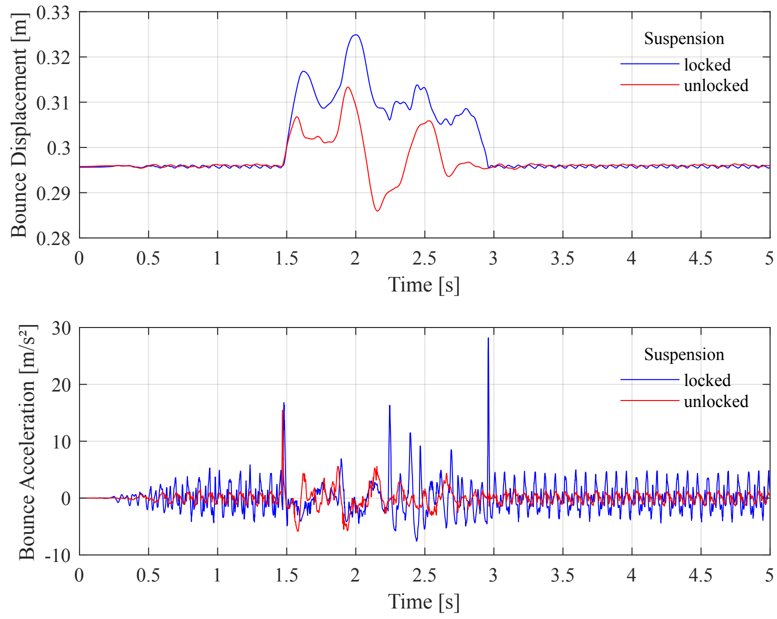


Figure 4.34: Results obtained from the bump negotiation simulation (0.55 m/s): (top) bounce displacement, (bottom) bounce acceleration.

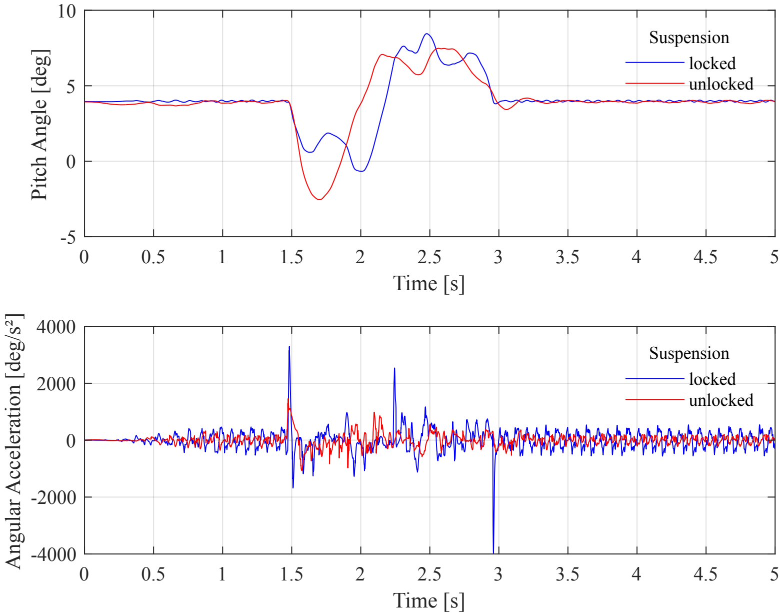


Figure 4.35: Results obtained from the bump negotiation simulation (0.55 m/s): (top) pitch angle, (bottom) angular acceleration.

4.2.5.2 Ditch

In this section a negative obstacle (i.e. a ditch) is considered. The geometry of the ditch is shown in Fig. 4.36. The height of the ditch is chosen to match the radius of the first road wheel, while its length is shorter than the longitudinal wheelbase of the robot. In this way, the robot must start climbing out of the ditch while the last road wheel is still on top of the first edge. In the simulated scenario, the robot is moving from left to right with a speed of 0.1 m/s.

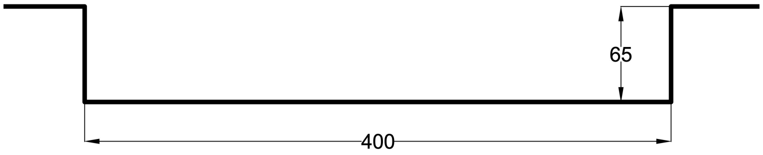


Figure 4.36: Geometry of the ditch with dimension in millimeters.

The results of the simulation are shown in Figs. 4.37 and 4.38 in terms of vertical and longitudinal acceleration respectively. Although the accelerations show a similar pattern in both cases of unlocked and locked suspension, the locked case presents bigger spikes, which reflects on the peak-to-peak root-mean-square values, as shown in Table 4.15. When the suspension is active, the peak-to-peak value of the vertical acceleration is 55% lower than the locked case (4.055 m/s^2 against 9.040 m/s^2), while the root-mean-square is 30% lower (0.559 m/s^2 against 0.799 m/s^2). Similarly the peak-to-peak value of the longitudinal acceleration is 47% lower (7.830 m/s^2 against 14.690 m/s^2), while the root-mean-square is 32% lower (0.772 m/s^2 against 1.137 m/s^2).

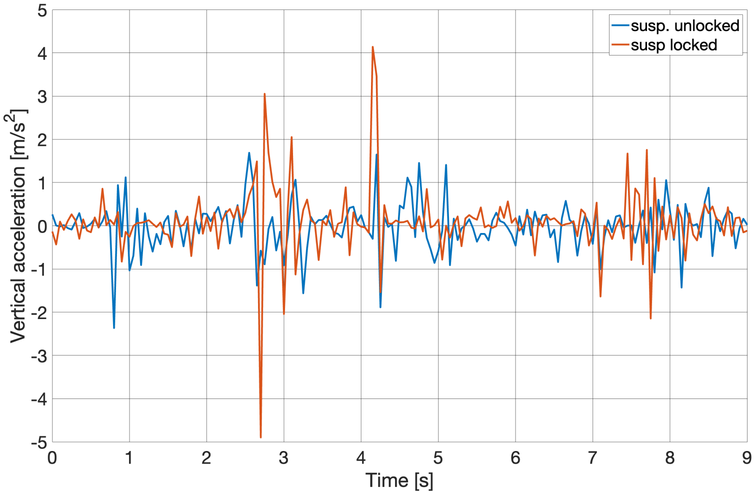


Figure 4.37: Vertical acceleration, negotiation of a ditch.

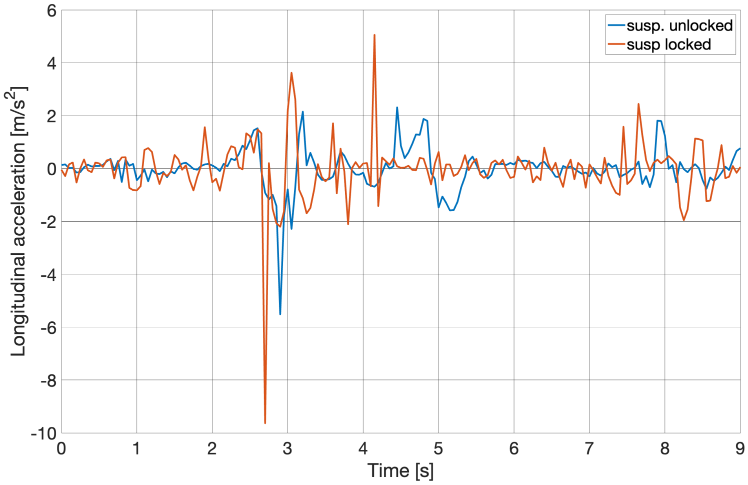


Figure 4.38: Longitudinal acceleration, negotiation of a ditch.

Peak-to-peak		
Suspension	VA [m/s^2]	LA [m/s^2]
Locked	9.040	14.690
Unlocked	4.055	7.830
Root-mean-square		
Suspension	VA [m/s^2]	LA [m/s^2]
Locked	0.799	1.137
Unlocked	0.559	0.772

Table 4.15: Results obtained from the ditch negotiation simulation. VA = Vertical Acceleration, LA = Longitudinal Acceleration.

4.2.5.3 Falling from a step

The scenario considered in this subsection is a fall from a step. The geometry of the step is shown in Fig. 4.39. The height of the step is chosen to match the diameter of the first road wheel. In the simulation, the robot is moving from left to right with a speed of 0.5 m/s.

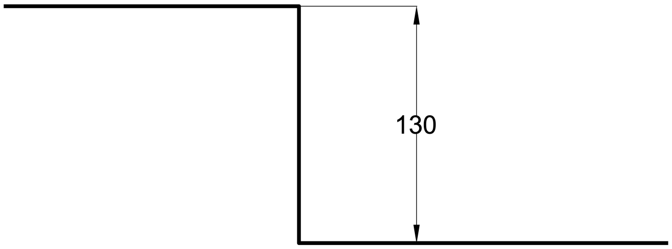


Figure 4.39: Geometry of the step with dimension in millimeters.

The results of the simulation are shown in Fig. 4.40 in terms of vertical acceleration and in Table 4.16 in terms of peak-to-peak and root-mean square values. When the suspension is active, the peak-to-peak value of the vertical acceleration is 25% lower than the locked case (8.294 m/s^2 against 11.074 m/s^2), while the root-mean-square is 32% lower (1.756 m/s^2 against 2.600 m/s^2).

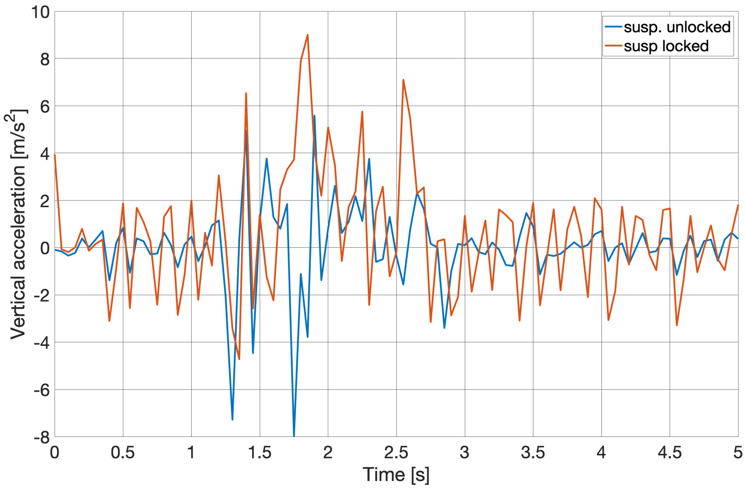


Figure 4.40: Vertical acceleration, falling from a step.

Peak-to-peak	
Suspension	VA [m/s^2]
Locked	11.074
Unlocked	8.294
Root-mean-square	
Suspension	VA [m/s^2]
Locked	2.600
Unlocked	1.756

Table 4.16: Results obtained from the step simulation. VA = Vertical Acceleration.

4.2.5.4 Stochastic uneven

The last scenario considered for testing the multibody model is the negotiation of a stochastic uneven surface modeled according to International Organization for Standardization [83] as an ISO F-profile, corresponding to a very poor surface profile with an RMS of 0.0383 m [84]. The surface is shown in Fig. 4.41. In the simulation, the robot moves from left to right with a speed of 0.5 m/s.

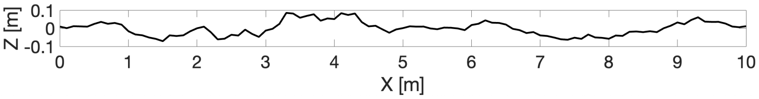


Figure 4.41: Representation of the stochastic uneven surface, ISO F-profile.

The results of the simulation are shown in Fig. 4.42 in terms of vertical acceleration and in Table 4.17 in terms of peak-to-peak and root-mean square values. When the suspension is active, the peak-to-peak value of the vertical acceleration is 16% lower than the locked case (30.166 m/s^2 against 35.901 m/s^2), while the root-mean-square is 30% lower (2.609 m/s^2 against 3.753 m/s^2).

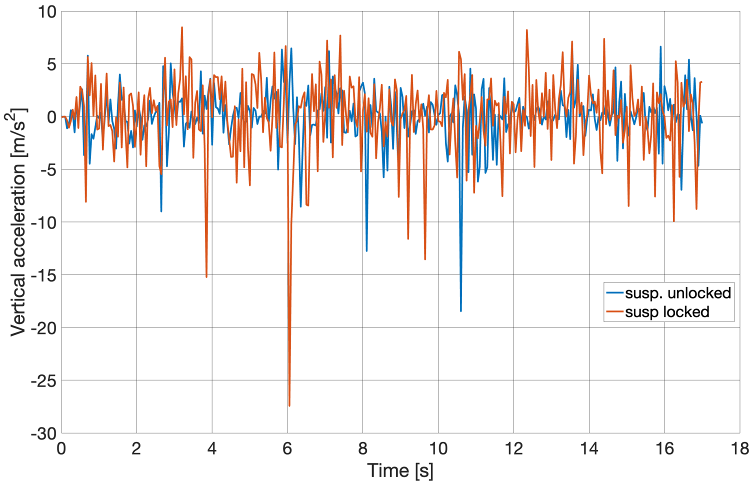


Figure 4.42: Vertical acceleration, stochastic uneven ISO F-profile.

Peak-to-peak	
Suspension	VA [m/s^2]
Locked	35.901
Unlocked	30.166
Root-mean-square	
Suspension	VA [m/s^2]
Locked	3.753
Unlocked	2.609

Table 4.17: Results obtained from the stochastic uneven simulation. VA = Vertical Acceleration.

4.3 Application of HAPF path planning to a planetary exploration scenario

Given the obstacle map of Fig. 3.21, a starting point located at coordinates (3200, 8000) and a goal located at coordinates (4600, 9400), the resulting potential field obtained with the iterative scheme described in section 3.3.2 is shown in Figures 4.43 and 4.44. The global maximum and minimum correspond to the start and goal positions respectively, and their values depend on the boundary conditions (Eq. 3.51 and 3.52). In the computational domain, the potential varies smoothly between these two values. The black lines of Fig. 4.43 are iso-potential lines and the gradient of the potential is always perpendicular to these lines. Because of the Neumann boundary conditions (Eq. 3.53), the iso-potential lines are perpendicular to obstacles and outer boundaries, which guarantees obstacle avoidance.

For demonstration purposes, it is assumed that the terrain presents a patch of regolith at the center of the map, as shown in Fig. 4.45, while all the rest of the surface is hard ground. For sinkage calculation (Eq. 3.76), the properties of the regolith terrain are assumed $n = 1$, $k_c = 1.4 \quad kN/m^{(n-1)}$ and $k_\phi = 820 \quad kN/m^{(n-2)}$ [85]. The results of the path optimization problem described in section 3.3.3 are shown in Fig. 4.46 and Table 4.18. By varying the initial heading angle and following the gradient descent, twelve different paths are found. The best paths for each individual cost are highlighted in Fig. 4.46 in white (best c_1), black (best c_2), gray (best c_3) and yellow (best c_4), while the best overall path (assuming equal weights $w_i = 0.25, for \quad i = 1, \ldots, 4$) is blue. All the other paths are in red. The

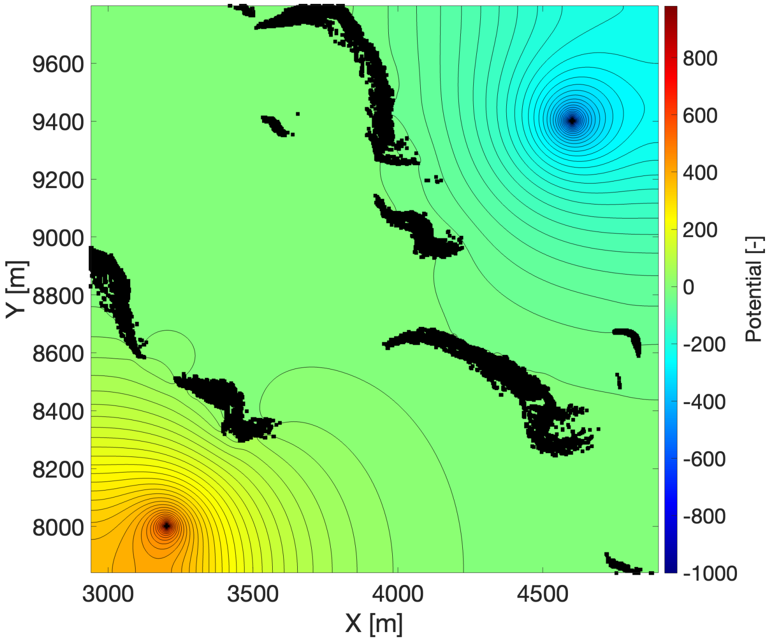


Figure 4.43: Contour plot of the initial potential field.

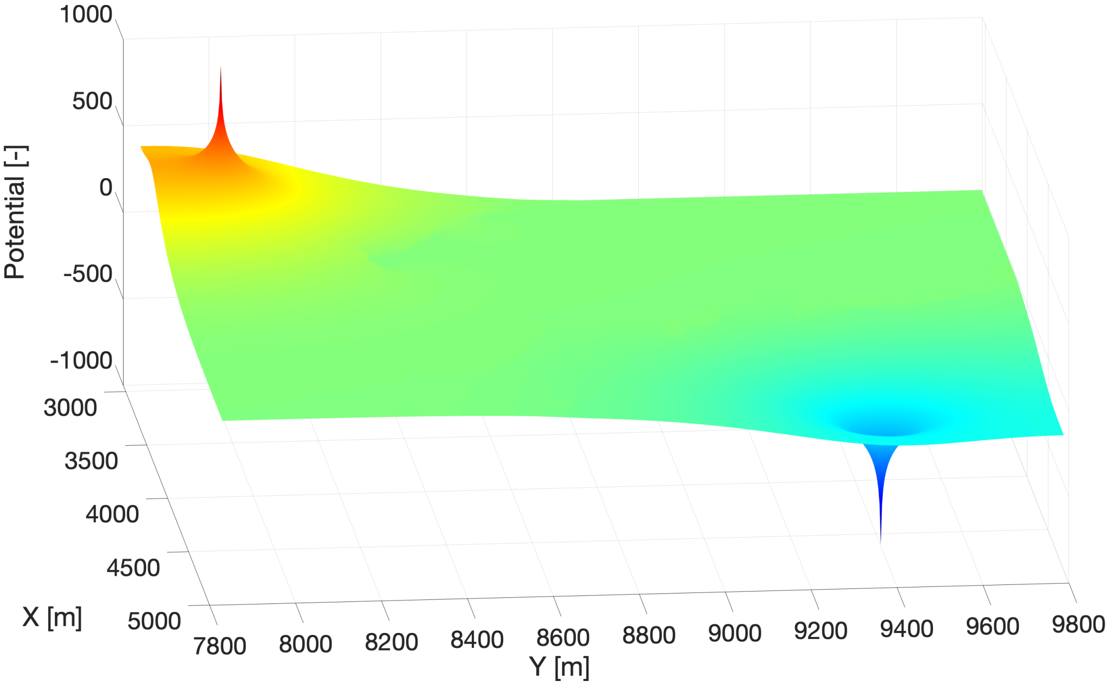


Figure 4.44: 3D surface plot of the initial potential field.

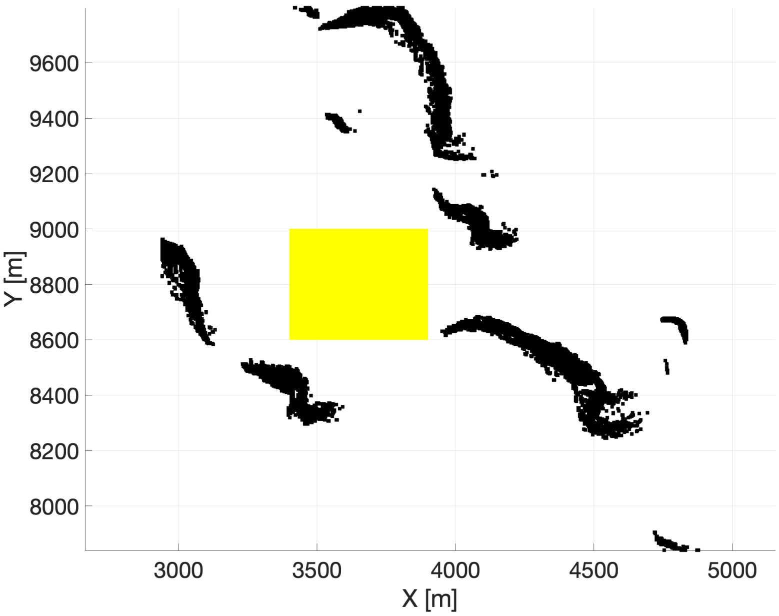


Figure 4.45: Obstacle map with a patch of regolith yellow.

gray path (number 11 in Table 4.18) is not visible because it coincides with the blue path (best overall). The white path (number 10) is the path that optimizes the distance from the obstacles, the black path (number 8) is the shortest, while the yellow path (number 7) is the best in terms of terramechanics (c_4) and indeed it avoids the central portion of the map, where the robot would find the patch of regolith. The blue path (number 11) is the best overall, with a minimum distance to obstacles of 78.45 m, a lenght of 2160 m, an average slope of 0.0528 (corresponding to about 3 degrees) and an average sinkage of 0.019 mm.

After the optimal path is selected, the robot can start following it. In this work, it assumed that the robot follows perfectly the gradient of the potential. Path tracking will be addressed in future works. During the motion, new obstacles are detected. For brevity, obstacle detection is addressed in the following way: it is assumed that the robot is able to perfectly reconstruct the terrain in a squared area of 40 m x 40 m centered on itself. This reconstruction is actually taken from the original DTM (note that the original DTM was down-sampled before feeding it to the path planner). Therefore this new map has a resolution of 0.98 m, and new high-slope areas may appear because of this. Note that robot localization and mapping is out of the scope of this work. Therefore, the new refined map is

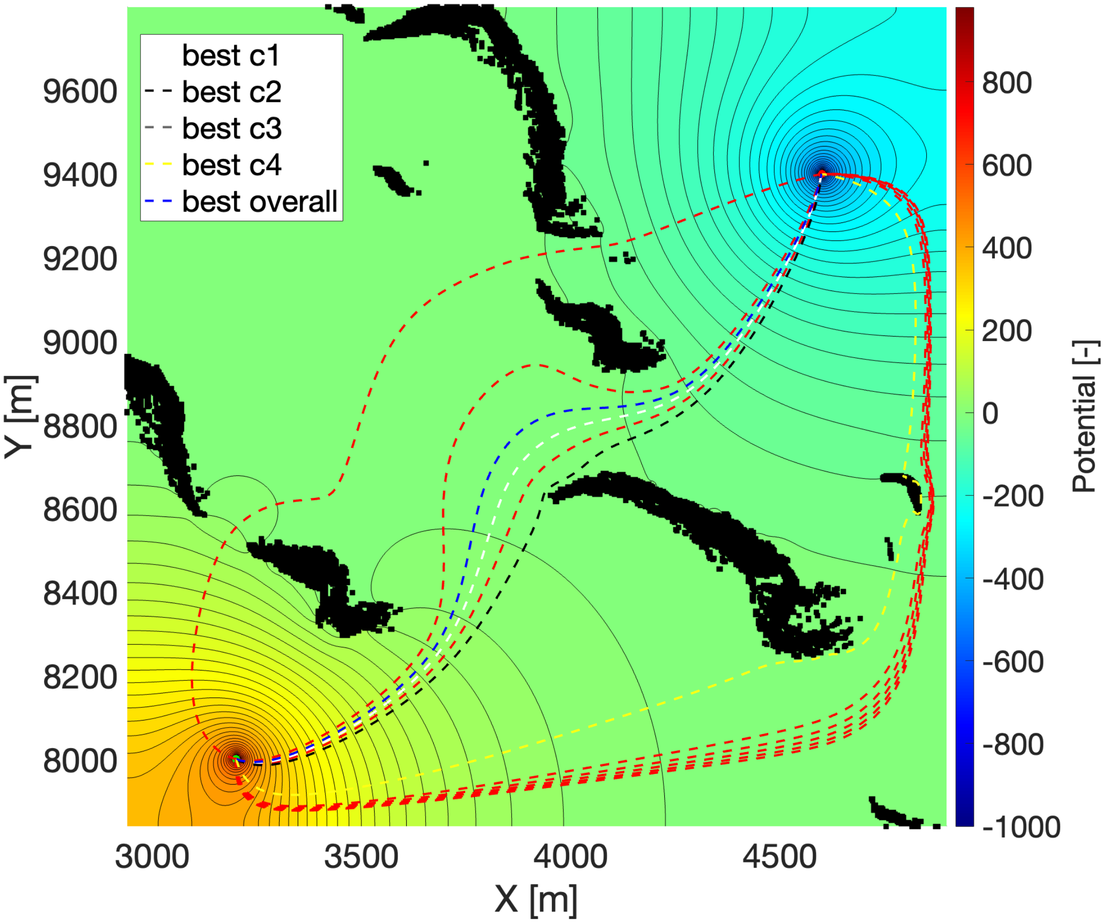


Figure 4.46: Results of the path optimization problem.

Path number	C_1	C_2	C_3	C_4	Total cost
1	0.73996	0.26628	0.76059	8.4858e-05	0.44173
2	0.66189	1	0.14748	3.7877e-05	0.45235
3	0.67495	0.98542	0.21135	4.2432e-05	0.46794
4	0.70102	0.96696	0.096677	4.0192e-05	0.44117
5	0.72414	0.94266	0.14817	4.4251e-05	0.45375
6	0.77247	0.90671	0.094264	4.865e-05	0.44337
7	1	0.80175	1	0	0.70044
8	0.88714	0	0.51367	0.86357	0.5661
9	0.41921	0.00097182	0.53514	1	0.48883
10	0	0.016521	0.43395	0.90589	0.33909
11	0.16517	0.047619	0	0.73455	0.23683
12	0.57628	0.17784	0.18603	0.19723	0.28435

Table 4.18: Costs of each path found.

automatically overlapped to the existing low-resolution map.

The resulting trajectory is plotted with a blue dashed line in Fig. 4.47 and compared against the original optimal path (in red). Every time a new obstacle is detected, the local planner generates a local source potential (Eq. 3.77). The location of these repulsive potential sources is indicated with red dots in Fig. 4.47. On the other hand, every time the robot deviates too much from the initial optimal path, the planner generates temporary sink (attractive) potentials, which location is indicated with blue dots. In the zoomed detail of Fig. 4.48 it is possible to appreciate how the robot avoids the newly encountered obstacles (red dots) and tries to converge to the optimal path when the danger is overcome.

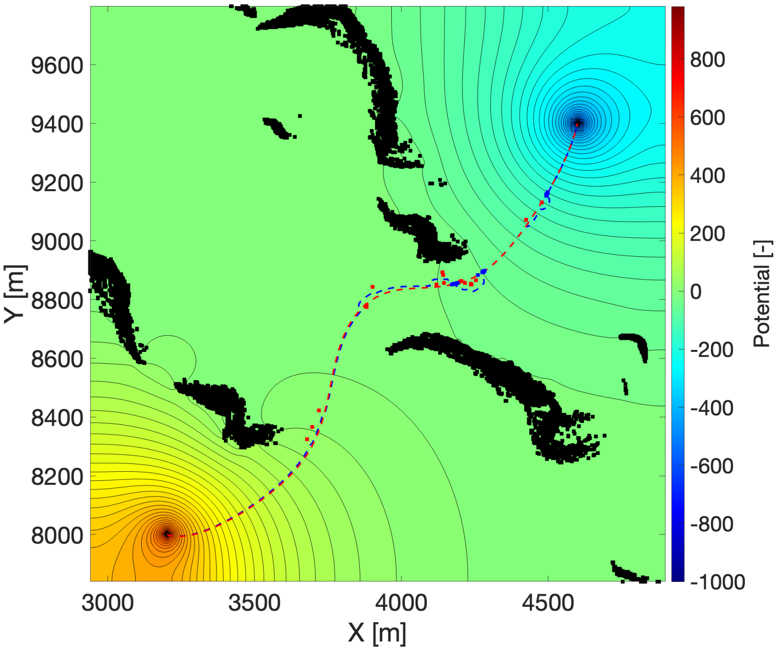


Figure 4.47: Resulting trajectory.

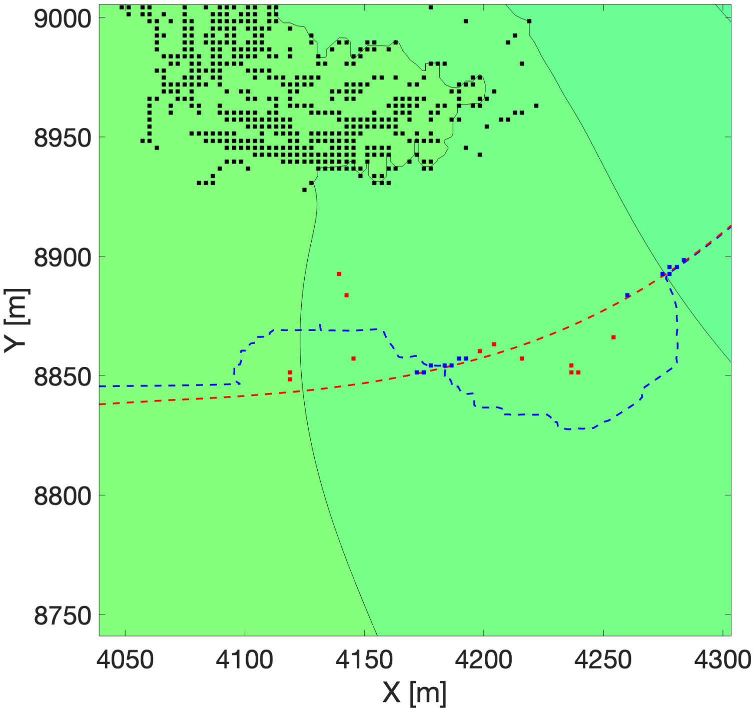


Figure 4.48: Detail of the resulting trajectory highlighting the obstacle avoidance capability of the algorithm.

Chapter 5

Conclusions

In conclusion, this thesis has delved into the realm of mobile robots for planetary exploration, focusing on two key features: autonomous decision making and robust mobility. These attributes are fundamental in extending the horizons of space exploration, allowing for extended mobility, increased safety, and improved data collection capabilities. Throughout this work, three interrelated research objectives were addressed: machine learning algorithms for terrain awareness based on proprioceptive sensing, the validation of an innovative suspension system, and a novel approach to HAPF path planning for autonomous vehicles.

The following sections draw the conclusions for each research objective, highlighting main results, limitations and future works.

5.1 Proprioceptive sensing and feature engineering for terrain learning

This work presented an approach to soil classification that relies on proprioceptive sensing only, e.g. accelerations, forces, torques, and electrical currents. The algorithms developed are validated on data collected during tests performed with the hybrid wheeled-legged rover SherpaTT. The physics-based signal augmentation process presented in this paper uses 11 proprioceptive measurements to produce a large set of 80 features for SVM and 20 signals for CNN. This improved the information content as proved by the high classification accuracy obtained in generalization (90.8% for SVM and 96.4% for CNN). The proposed feature selection algorithm allows SVM to retain a high classification accuracy with only a portion of the full set (18 features), with successful reductions in memory usage (-77%) and required time for training (-6%), testing (-29%) and feature extraction (-33%). The same benefits also apply to CNN when using a reduced set of 13 signals related to the 18 best SVM features, improving memory usage (-14%), training time (-27%), testing time (-4%) and feature extraction time (-41%). The comparison between SVM and CNN shows good capabilities of both models in generalization, with accuracy higher than 90%. More challenging extrapolation problems have been tackled as well to evaluate the impact of varying operating conditions and site of the acquisition. In these tests, CNN outperformed the SVM counterpart. When tested on a new vehicle velocity, CNN reached an accuracy of 89.5%, against 55.7% held by SVM. When tested on a new terrain, CNN recognized its deformability class more frequently than SVM, correctly classifying 6% more of the available samples. Based on these results, the proposed CNN qualifies as a good algorithm for soil classification even in the presence of disturbances and unknown conditions.

This thesis proved that is possible to use only proprioceptive features to infer the signature of a particular surface via learning algorithms. Moreover, the presented promising results suggest the possibility to extend rover travelling distance thanks to on-board integration of the developed learning algorithms.

Future developments of this research refer to:

- continuous training of the system by incorporating instances of “new terrain” classes during normal operations, therefore making the system adaptive;
- augmenting the classifier with new special classes; for example, instances of excessive wheel slippage (close to 100%) can be used to train a hazard class to inform the rover of impending immobilization conditions;
- combining the proposed framework using proprioceptive signals with exteroceptive signals.

The latter would enable the vehicle to predict hazards or trapping conditions before driving through the ground, e.g., based on non-contact information coming from vision sensors. An initial attempt was proposed in [86].

5.2 Polibot and its innovative suspension system

This thesis presented a novel off-road tracked robot named Polibot with an innovative passively articulated suspension system, in which each road wheel is independently suspended to adapt to terrain irregularities and distribute pressure evenly in all conditions. The proposed architecture has greater mobility when compared with non-articulated robots or other passively articulated solutions, but it is still mechanically simpler then actively articulated suspensions.

An analytical modeling tool for the inverse kinematics of the proposed design is also presented. The model is represented by a system of 21 equations that includes constraints and a quasi-static analysis. The unknowns of the problem are the seven degrees of freedom of the suspension and the fourteen unknown forces, including contact and internal forces. The validation against experimental results shows good agreement with the results of the model, which is able to predict the system configuration given the terrain geometry. The model is also used to evaluate the impact of suspension design parameters and operating conditions, proving that it is a useful tool to optimize the current prototype of Polibot, or to design the future generations of the rover. The current model will be further developed in future work to include a full dynamic analysis of the suspension and to account for the terramechanics at the track-ground interface. This tool can be implemented in the real robot for model-based control and estimation. As a further research, a model-based optimization of Polibot design parameters can be developed.

This thesis also presented the MultiBody model of the tracked robot Polibot. The performance of the suspension in various challenging scenarios was evaluated with both active and locked modes. The multibody model was design using MSC Adams and its ATV (Adams Tracked Vehicle) toolkit. Specific templates were developed for the middle and front units of the innovative passive suspension implemented on the prototype of Polibot. The simulations results proved that the suspension system is capable to mitigate the vibrations of the chassis in response to various surfaces. In case of bump negotiation, the RMS value of the bounce (vertical) acceleration and of the pitch acceleration are reduced by 49% and 45%, respectively. While overcoming a ditch, the RMS of the vertical acceleration and longitudinal acceleration are reduced by 30% and 32% respectively. In the case of the robot falling from a step, the RMS of the vertical acceleration is 32% lower than in the locked suspension case. Finally, also for the test on the irregular surface with an ISO F profile, the suspension performs well, with a reduction of 30% in the RMS of vertical acceleration.

Experimental tests and simulations have been performed on hard terrain. This is one limitation of the proposed research. Future efforts will be devoted to extend the analysis to deformable and loose soils drawing on Terramechanics theory. As additional developments, the authors will focus on the fine-tuning of the model parameters, with the objective to replicate more closely the data gathered from the real prototype during tests in the field and in the laboratory. In addition, an investigation of novel suspension architectures can be carried out thanks to the novel MultiBody model.

5.3 A novel HAPF path planning approach for planetary exploration

This thesis effectively demonstrated the application of HAPF to planetary exploration robots. A path planning approach based on Harmonic Artificial Potential Field was applied to a real-case planetary exploration scenario. A DTM of a portion of Mars was used as input map and converted to an obstacle map, which is then used as a computational domain to solve Laplace's equation. The resulting potential is then used to generate various suitable paths, among which the optimal path is chosen via a path-optimization problem. The proposed algorithm was able to generate an optimal path according to four criteria: distance to obstacles, length, average slope and average sinkage. Also, the local planner was able to avoid unexpected hazards encountered by the robot along the path.

The proposed algorithm preserves the same features of traditional APF approaches, while eliminating the typical local minima issue. The local planning strategy is able to cope with new information about the surrounding environment. Therefore, the proposed HAPF algorithm can be easily coupled with a terrain classifier such as the ones developed in this thesis.

One of the limitations of this research is that only the vertical (quasi-static) dynamics of the robot is considered, with the use of the analytical model of Polibot developed in this thesis. Future efforts will focus on including longitudinal and lateral dynamics, slippage estimation and other terramechanics-related phenomena.

Appendix A

Geometric parameters of Polibot suspension

Figure A.1 presents a summary of the main geometric parameters of the bodies that make up the suspension.

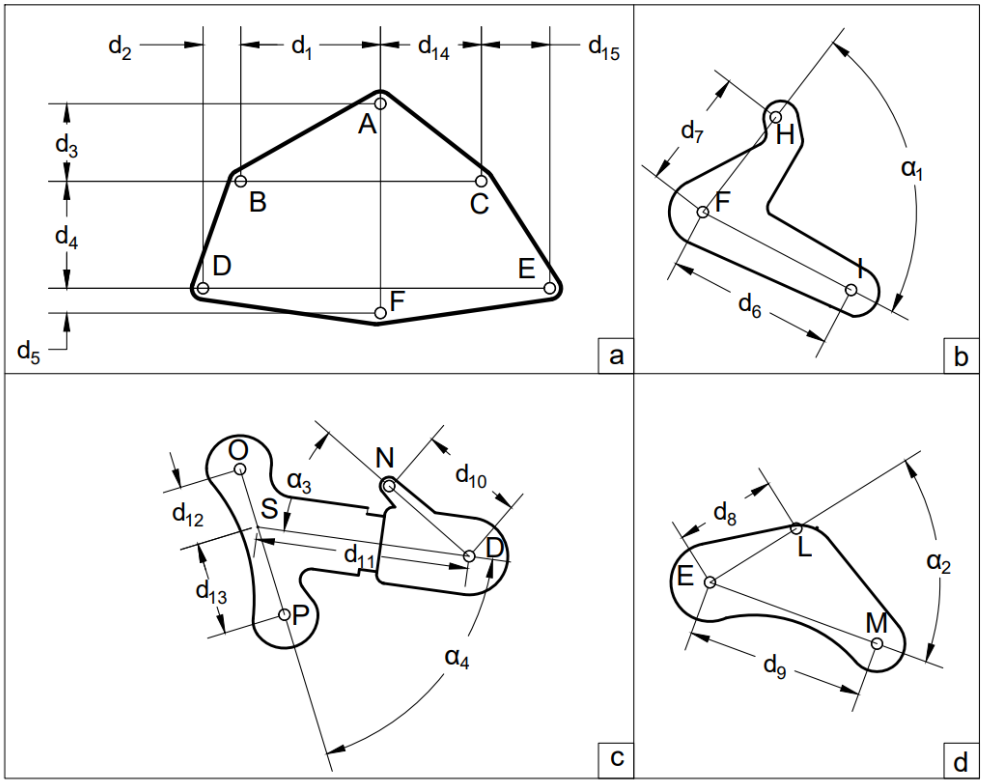


Figure A.1: Geometry of the suspension bodies: a) SF, b) BRA and BFA, c) FSA, d) RSA.

Appendix B

Analytical derivation of geometric entities of Polibot suspension

B.1 Derivation of wheel centers coordinates as a function of the degrees of freedom of the suspension

Given the geometry of the suspension (Figures 3.13 and A.1), the X and Z coordinates of the wheel centers (P, K, I and M) can be expressed as a function of the degrees of freedom of Table 3.9 as follows:

$$\begin{aligned} X_P = X_A + (d_3 + d_4) \cdot \sin\theta - (d_1 + d_2) \cdot \cos\theta - d_{11} \cdot \sin(\theta_4 + \alpha_3) + \\ + d_{13} \cdot \sin(\theta_4 + \alpha_3 - \alpha_4) \end{aligned} \tag{B.1}$$

$$X_K = X_A + (d_3 + d_4 + d_5) \cdot \sin\theta - d_6 \cdot \sin(\theta_2 + \alpha_1) \tag{B.2}$$

$$X_I = X_A + (d_3 + d_4 + d_5) \cdot \sin\theta + d_6 \cdot \sin\theta_1 \tag{B.3}$$

$$X_M = X_A + (d_3 + d_4) \cdot \sin\theta + (d_{14} + d_{15}) \cdot \cos\theta + d_9 \cdot \sin\theta_3 \tag{B.4}$$

$$Z_P = Z_A - (d_3 + d_4) \cdot \cos\theta - (d_1 + d_2) \cdot \sin\theta + d_{11} \cdot \cos(\theta_4 + \alpha_3) + \\ -d_{13} \cdot \cos(\theta_4 + \alpha_3 - \alpha_4) \qquad \text{(B.5)}$$

$$Z_K = Z_A - (d_3 + d_4 + d_5) \cdot \cos\theta + d_6 \cdot \cos(\theta_2 + \alpha_1) \qquad \text{(B.6)}$$

$$Z_I = Z_A - (d_3 + d_4 + d_5) \cdot \cos\theta - d_6 \cdot \cos\theta_1 \qquad \text{(B.7)}$$

$$Z_M = Z_A - (d_3 + d_4) \cdot \cos\theta + (d_{14} + d_{15}) \cdot \sin\theta - d_9 \cdot \cos\theta_3 \qquad \text{(B.8)}$$

B.2 Derivation of length and direction of springs as a function of the degrees of freedom

The length of the three springs (S_1, S_2 and S_3) can be obtained as a function of the degrees of freedom of Table 3.9 as follows:

$$L_1 = d_7 \cdot [\sin\theta_2 + \sin(\theta_1 + \alpha_1)] \qquad \text{(B.9)}$$

$$L_2 = \sqrt{(X_L - X_C)^2 + (Z_L - Z_C)^2} \qquad \text{(B.10)}$$

$$L_3 = \sqrt{(X_B - X_N)^2 + (Z_B - Z_N)^2} \qquad \text{(B.11)}$$

where:

$$X_L - X_C = d_4 \cdot \sin\theta + d_{15} \cdot \cos\theta + d_8 \cdot \sin(\theta_3 + \alpha_2) \qquad \text{(B.12)}$$

$$Z_L - Z_C = -d_4 \cdot \cos\theta + d_{15} \cdot \sin\theta - d_8 \cdot \cos(\theta_3 + \alpha_2) \qquad \text{(B.13)}$$

$$X_B - X_N = d_2 \cdot \cos\theta - d_4 \cdot \sin\theta + d_{10} \cdot \sin\theta_4 \qquad \text{(B.14)}$$

$$Z_B - Z_N = d_2 \cdot \sin\theta + d_4 \cdot \cos\theta - d_{10} \cdot \cos\theta_4 \qquad \text{(B.15)}$$

For springs 2 and 3 it is also necessary to define their directions, indicated with β_2 and β_3 in Figures 3.15 and 3.16:

$$\beta_2 = \text{atan2}\left(Z_C - Z_L, X_C - X_L\right) \tag{B.16}$$

$$\beta_3 = \text{atan2}\left(Z_N - Z_B, X_N - X_B\right) \tag{B.17}$$

where atan2 is the four-quadrant inverse tangent.

B.3 Derivation of track length as a function of the degrees of freedom of the suspension

In this appendix, the expression of the length of the track as a function of the degrees of freedom of Table 3.9 is derived. The length can be calculated as the sum of line segments and arcs, where the track wraps around the wheels. For each wheel, the extension of the arc of contact with the track can be expressed as a function of the relative position of the following and precedent wheels. For example, the length of the section of the track that wraps around wheel 1 (Fig. B.1) delimited by the points Q_{12} and Q_{16} can be calculated as follows:

$$\overset{\frown}{Q_{12}Q_{16}} = R_1 \cdot (2\pi + \delta_{fol} - \gamma_{fol} - \delta_{pre} - \gamma_{pre}) \tag{B.18}$$

where R_1 is the radius of wheel 1, δ_{fol} is the orientation of the line $\overline{AO}$ that points to the center of the following wheel, δ_{pre} is the orientation of the line $\overline{AM}$ that points to the center of the precedent wheel, γ_{fol} is the angle that defines the position of the tangency point Q_{12} with respect to $\overline{AO}$, γ_{pre} is the angle that defines the position of the tangency point Q_{16} with respect to $\overline{AM}$. These angles can be computed as follows:

$$\delta_{fol} = \text{atan2}(Z_O - Z_A, X_O - X_A) \tag{B.19}$$

$$\delta_{pre} = \text{atan2}(Z_M - Z_A, X_M - X_A) \tag{B.20}$$

$$\gamma_{fol} = \arccos \frac{R_1 - R_2}{\overline{AO}} \tag{B.21}$$

$$\gamma_{pre} = \arccos \frac{R_1 - R_6}{\overline{AM}} \tag{B.22}$$

where atan2 is the four-quadrant inverse tangent, R_2 and R_6 are the radii of wheels 2 and 6 respectively, $\overline{AO}$ and $\overline{AM}$ represent the length of the segment of the correspondent line.

The next entity required for the calculation of the track length is the line segment delimited by the two tangency points Q_{12} and Q_{21}. Its length can be computed as follows:

$$\overline{Q_{12}Q_{21}} = \sqrt{\overline{AO}^2 - (R_1 - R_2)^2} \tag{B.23}$$

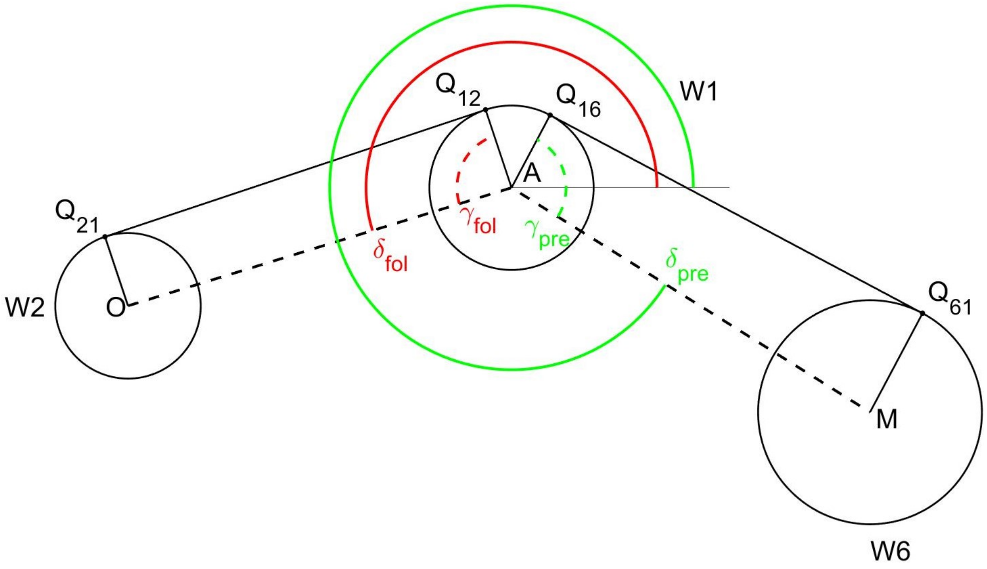


Figure B.1: Geometric construction for the calculation of the length of the contact arc between track and wheel 1.

Finally, an expression for the overall track length can be derived from the generalisation of equations B.18 and B.23:

$$L_{track} = \sum_{i=1}^{6} R_i \cdot (2\pi + \delta_{fol,i} - \gamma_{fol,i} - \delta_{pre,i}{}^{*} - \gamma_{pre,i}) + \sqrt{\overline{C_i C_{i+1}}^2 - (R_i - R_{i+1}{}^{\dagger})^2} \tag{B.24}$$

where C_i is the center of the wheel i and R_i is the radius of the wheel i.

Given the geometry of the suspension, all the terms in B.24 can be expressed as a function of the degrees of freedom of Table 3.9. The final track length expression is omitted for brevity. A numerical approach is adopted for the solution of the equation set (1)-(8), using the trust region algorithm. To this end, the problem can be formulated as:

$$F(x) = 0 \tag{B.25}$$

where x is the array of unknowns and F is the array of equations. The trust region algorithm is based on Newton's method, with improved robustness in the case of starting point far from the solution or in case of singular Jacobian. It can be proved that the trust region sub-problem is given by:

$$\min_{d}[\frac{1}{2}F(x_k)^T F(x_k) + d^T J(x_k)^T F(x_k) + \frac{1}{2}d^T J(x_k)^T J(x_k) d] \tag{B.26}$$

where x_k is the array of variables at iteration k, J is the Jacobian of equation B.25 and d is the search direction (i.e. the quantity to add to x_k to obtain x_{k+1}). For more details on trust region methods, the interested readers can refer to [87].

Finally, in the case of concave track profile, the formulas for the wheel on the outside of the track should be updated as follows, with respect to the same geometry of Fig. B.1:

$$\gamma_{fol} = -\arccos\frac{R_1 + R_2}{\overline{AO}} \tag{B.27}$$

$$\gamma_{pre} = -\arccos\frac{R_1 + R_6}{\overline{AM}} \tag{B.28}$$

[*] For the formula of equation B.24 to work, δ_{pre} must be numerically grater than δ_{fol} for all wheels. It may happen that this condition is not verified for some wheels. In those situations, it is sufficient to add 2π to δ_{pre}.

[†] For wheel 6, R_{i+1} is replaced by R_1

Appendix C

Equilibrium equations for the analytical model of Polibot

Referring to Figures 3.14, 3.15 and 3.16, seven equilibrium equations can be obtained (global vertical forces, global moments around point P, moments on BRA around F, moments on BFA around F, moments on RSA around E and moments on FSA around D):

$$F_{tra} = F_{res} \tag{C.1}$$

$$W = N_3 + N_4 + N_5 + N_6 \tag{C.2}$$

$$N_4 \cdot (X_K - X_P) + N_5 \cdot (X_I - X_P) + N_6 \cdot (X_M - X_P) = W \cdot (X_G - X_P) \tag{C.3}$$

$$N_5 \cdot d_6 \cdot \sin \theta_1 + F_{el,1} \cdot d_7 \cdot \cos(\theta_1 + \alpha_1) = 0 \tag{C.4}$$

$$N_4 \cdot d_6 \cdot \sin(\theta_2 + \alpha_1) + F_{el,1} \cdot d_7 \cdot \cos \theta_2 = 0 \tag{C.5}$$

$$
\begin{aligned}
N_6 \cdot d_9 \cdot \sin\theta_3 - T_1 \cdot (d_9 \cdot \cos\theta_3 + r_6) - F_{el,2} \cdot d_8 \cdot \cos(\theta_3 - \beta_2 + \alpha_2) + \\
-T_1 \cdot \cos\epsilon_{61} \cdot (Z_{Q_{61}} - Z_E) + T_1 \cdot \sin\epsilon_{61} \cdot (x_{Q_{61}} - x_E) = 0
\end{aligned}
\tag{C.6}
$$

$$
\begin{aligned}
N_3 \cdot [d_{11} \cdot \sin(\theta_4 + \alpha_3) - d_{13} \cdot \sin(\theta_4 + \alpha_3 - \alpha_4)] + \\
-F_{el3} \cdot d_{10} \cdot \cos(\theta_4 - \beta_3) + \\
+T_2 \cdot [d_{13} \cdot \cos(\theta_4 + \alpha_3 - \alpha_4) - d_{11} \cdot \cos(\theta_4 + \alpha_3) + r_3] + \\
-T_2 \cdot \cos\epsilon_{21} \cdot (Z_{Q_{21}} - Z_D) - T_2 \cdot \sin\epsilon_{21} \cdot (x_D - x_{Q_{21}}) = 0
\end{aligned}
\tag{C.7}
$$

The directions of the elastic forces (indicated in Figures 3.15 and 3.16 with β_2 and β_3), are derived as a function of the degrees of freedom of Table 3.9 in B.2. The directions of the track tensions (ϵ_{21} and ϵ_{61}) are derived in B.3.

Finally, the last equation is given by the longitudinal equilibrium of the track element that exchanges forces with the ground:

$$
F_{tra} = T_1 - T_2 \tag{C.8}
$$

Appendix D

Analytical model for the X-shaped suspended robot

Referring to the suspension proposed by Sun et al. [46] and to the corresponding scheme of Fig. 4.16(b), the following equations can be written:

$$X_B = X_{CP_3} \tag{D.1}$$

$$Z_B - r = 0 \tag{D.2}$$

$$Z_C - r = 0 \tag{D.3}$$

$$Z_D - r = 0 \tag{D.4}$$

$$Z_E - r = 0 \tag{D.5}$$

$$L_{track} = L_{nom} \tag{D.6}$$

$$F_{z1} \cdot cos(\theta) - F_{el,1} + T \cdot [cos(\epsilon_{21} - \theta - \pi/2) - sin(\theta)] = 0 \tag{D.7}$$

$$F_{z2} \cdot cos(\theta) - F_{el,2} = 0 \tag{D.8}$$

$$F_{z3} \cdot cos(\theta) - F_{el,3} = 0 \tag{D.9}$$

$$F_{z4} \cdot cos(\theta) - F_{el,4} + T \cdot [cos(\pi/2 - \epsilon_{56} + \theta) + sin(\theta)] = 0 \tag{D.10}$$

$$F_{z1} + F_{z2} + F_{z3} + F_{z4} - Mg = 0 \tag{D.11}$$

$$F_{z2} \cdot (X_C - X_B) + F_{z3} \cdot (X_D - X_B) + F_{z4} \cdot (X_E - X_B) - Mg \cdot (X_G - X_B) = 0 \tag{D.12}$$

where D.1 to D.6 represent the constraint equations, which are the same used for Polibot model (Section 3.2.2.2), D.7 to D.10 are the equilibrium equations for each suspended wheel, D.11 is the global equilibrium of external forces in the Z direction, D.12 is the global equilibrium of external moments around point B. These 12 equations constitute a system in 12 unknowns: the position of the central frame (X_G, Z_G and θ), the length of each elastic element (L_i, with $i = 1, \ldots, 4$), the normal reaction forces below each road wheel ($F_{z,j}$, with $j = 1, \ldots, 4$) and the average track tension T.

Bibliography

[1] Angelo Ugenti, Fabio Vulpi, Raúl Domínguez, Florian Cordes, Annalisa Milella, and Giulio Reina. On the role of feature and signal selection for terrain learning in planetary exploration robots. *Journal of Field Robotics*, 39(4):355–370, 2022.

[2] Jorge Ocón, Iulia Dragomir, Florian Cordes, Raúl Dominguez, Robert Marc, Vincent Bissonnette, Raphael Viards, Anne Claire Berthet, Giulio Reina, Angelo Ugenti, et al. Ade: Enhancing autonomy for future planetary robotic exploration. In *IAF Space Exploration Symposium 2021 at the 72nd International Astronautical Congress, IAC 2021.* International Astronautical Federation, IAF, 2021.

[3] ESA. Robotic exploration of Mars, 2021. `http://exploration.esa.int/mars/` [Last accessed October 2023].

[4] NASA Jet Propulsion Laboratory. Mars Exploration Rovers, 2021. `http://marsrovers.jpl.nasa.gov/home/index.html` [Last accessed October 2023].

[5] NASA. Mars 2020, 2020. `https://mars.nasa.gov/mars2020/` [Last accessed October 2023].

[6] R Cowen. Opportunity rolls out of purgatory. *Science News*, 167(26):413, 2005.

[7] MG Harinarayanan Nampoothiri, B Vinayakumar, Youhan Sunny, and Rahul Antony. Recent developments in terrain identification, classification, parameter estimation for the navigation of autonomous robots. *SN Applied Sciences*, 3:1–14, 2021.

[8] Christopher A Brooks and Karl Iagnemma. Vibration-based terrain classification for planetary exploration rovers. *IEEE Transactions on Robotics*, 21(6):1185–1191, 2005.

[9] Ramon Gonzalez, Samuel Chandler, and Dimi Apostolopoulos. Characterization of machine learning algorithms for slippage estimation in planetary exploration rovers. *Journal of Terramechanics*, 82:23–34, 2019.

[10] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[11] Angelo Ugenti, Rocco Galati, Giacomo Mantriota, and Giulio Reina. Analysis of an all-terrain tracked robot with innovative suspension system. *Mechanism and Machine Theory*, 182:105237, 2023.

[12] Andrea Grazioso, Angelo Ugenti, Rocco Galati, Giacomo Mantriota, and Giulio Reina. Modeling and validation of a novel tracked robot via multibody dynamics. *Robotica*, 41(10):3211–3232, 2023.

[13] Luca Bruzzone and Giuseppe Quaglia. Locomotion systems for ground mobile robots in unstructured environments. *Mechanical Sciences*, 3(2):49–62, 2012.

[14] Sachiko Wakabayashi, Hitoshi Sato, and Shin-Ichiro Nishida. Design and mobility evaluation of tracked lunar vehicle. *Journal of Terramechanics*, 46(3):105–114, 2009.

[15] Adams Development Team. MSC Adams/View Manual, 2022. `https://help.hexagonmi.com/en-US/bundle/Adams_2021.3_Adams_View_User_Guide/resource/Adams_2021.3_Adams_View_User_Guide.pdf` [Last accessed 05 December 2022].

[16] Adams Development Team. MSC Adams/Adams tracked vehicle (atv) toolkit, 2022. `https://help.hexagonmi.com/it-IT/bundle/adams_2020/page/adams_help/Adams_Car_Package/tire/tire_softsoil_model/tire.softsoil.model.xhtml` [Last accessed 05 December 2022].

[17] Róbert Marc and Piotr Weclewski. Capabilities of long range autonomous multi-mode rover navigation system-ergo field trials and planned evolution.

In *Proceedings of the 15th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, 2019.

[18] Alessandro Gasparetto, Paolo Boscariol, Albano Lanzutti, and Renato Vidoni. Path planning and trajectory planning algorithms: A general overview. *Motion and Operation Planning of Robotic Systems: Background and Practical Approaches*, pages 3–27, 2015.

[19] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, 5(1):90–98, 1986.

[20] Daniel Koditschek. Exact robot navigation by means of potential functions: Some topological considerations. In *Proceedings. 1987 IEEE international conference on robotics and automation*, volume 4, pages 1–6. IEEE, 1987.

[21] Jose Ricardo Sanchez-Ibanez, Carlos J Perez-del Pulgar, and Alfonso García-Cerezo. Path planning for autonomous mobile robots: A review. *Sensors*, 21(23):7898, 2021.

[22] Christopher I Connolly, J Brian Burns, and Rich Weiss. Path planning using laplace's equation. In *Proceedings., IEEE International Conference on Robotics and Automation*, pages 2102–2106. IEEE, 1990.

[23] Lei Tai, Shaohua Li, and Ming Liu. Autonomous exploration of mobile robots through deep neural networks. *International Journal of Advanced Robotic Systems*, 14(4):1729881417703571, 2017.

[24] Lorenz Wellhausen, Alexey Dosovitskiy, René Ranftl, Krzysztof Walas, Cesar Cadena, and Marco Hutter. Where should i walk? predicting terrain properties from images via self-supervised learning. *IEEE Robotics and Automation Letters*, 4(2):1509–1516, 2019.

[25] Roberto Manduchi, Andres Castano, Ashit Talukder, and Larry Matthies. Obstacle detection and terrain classification for autonomous off-road navigation. *Autonomous robots*, 18:81–102, 2005.

[26] Kyohei Otsu, Masahiro Ono, Thomas J Fuchs, Ian Baldwin, and Takashi Kubota. Autonomous terrain classification with co-and self-training approach. *IEEE Robotics and Automation Letters*, 1(2):814–819, 2016.

[27] Dan Barnes, Will Maddern, and Ingmar Posner. Find your own way: Weakly-supervised segmentation of path proposals for urban autonomy. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 203–210. IEEE, 2017.

[28] Reina Ishikawa, Ryo Hachiuma, and Hideo Saito. Self-supervised audio-visual feature learning for single-modal incremental terrain type clustering. *IEEE Access*, 9:64346–64357, 2021.

[29] Edmond M Dupont, Carl A Moore, Emmanuel G Collins, and Eric Coyle. Frequency response method for terrain classification in autonomous ground vehicles. *Autonomous Robots*, 24:337–347, 2008.

[30] Philippe Giguere and Gregory Dudek. A simple tactile probe for surface identification by mobile robots. *IEEE Transactions on Robotics*, 27(3):534–544, 2011.

[31] Ayan Dutta and Prithviraj Dasgupta. Ensemble learning with weak classifiers for fast and reliable unknown terrain classification using mobile robots. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(11):2933–2944, 2016.

[32] Giulio Reina, Antonio Leanza, and Arcangelo Messina. Terrain estimation via vehicle vibration measurement and cubature kalman filtering. *Journal of Vibration and Control*, 26(11-12):885–898, 2020.

[33] Dengsheng Lu and Qihao Weng. A survey of image classification methods and techniques for improving classification performance. *International journal of Remote sensing*, 28(5):823–870, 2007.

[34] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[35] Fabio Vulpi, Annalisa Milella, Roberto Marani, and Giulio Reina. Recurrent and convolutional neural networks for deep terrain classification by autonomous robots. *Journal of Terramechanics*, 96:119–131, 2021.

[36] Mauro Dimastrogiovanni, Florian Cordes, and Giulio Reina. Terrain estimation for planetary exploration robots. *Applied Sciences*, 10(17):6044, 2020.

[37] Luca Bruzzone, Shahab Edin Nodehi, and Pietro Fanghella. Tracked locomotion systems for ground mobile robots: A review. *Machines*, 10(8):648, 2022.

[38] Satoshi Morita, Toshifumi Hiramatsu, Marta Niccolini, Alfredo Argiolas, and Matteo Ragaglia. Kinematic track modelling for fast multiple body dynamics simulation of tracked vehicle robot. In *2018 23rd International Conference on Methods & Models in Automation & Robotics (MMAR)*, pages 910–915. IEEE, 2018.

[39] Takafumi Haji, Tetsuya Kinugasa, Koji Yoshida, Hisanori Amano, and Koichi Osuka. Experiment of maneuverability of flexible mono-tread mobile track and differential-type tracked vehicle. *Industrial Robot: An International Journal*, 2010.

[40] Zhao Gong, Fugui Xie, Xin-Jun Liu, and Shuzhan Shentu. Obstacle-crossing strategy and formation parameters optimization of a multi-tracked-mobile-robot system with a parallel manipulator. *Mechanism and Machine Theory*, 152:103919, 2020.

[41] Zirong Luo, Jianzhong Shang, Guowu Wei, and Lei Ren. A reconfigurable hybrid wheel-track mobile robot based on watt ii six-bar linkage. *Mechanism and Machine Theory*, 128:16–32, 2018.

[42] Wenzhi Guo, Jiandu Qiu, Xinrui Xu, and Juan Wu. Talbot: A track-leg transformable robot. *Sensors*, 22(4):1470, 2022.

[43] Paul M Moubarak, Eric J Alvarez, and Pinhas Ben-Tzvi. Reconfiguring a modular robot into a humanoid formation: A multi-body dynamic perspective on motion scheduling for modules and their assemblies. In *2013 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 687–692. IEEE, 2013.

[44] Weijun Tao, Yi Ou, and Hutian Feng. Research on dynamics and stability in the stairs-climbing of a tracked mobile robot. *International Journal of Advanced Robotic Systems*, 9(4):146, 2012.

[45] Jihoon Kim, Jongwon Kim, and Donghun Lee. Mobile robot with passively articulated driving tracks for high terrainability and maneuverability on unstructured rough terrain: Design, analysis, and performance evaluation. *Journal of Mechanical Science and Technology*, 32(11):5389–5400, 2018.

[46] B. Sun and X. Jing. A tracked robot with novel bio-inspired passive ”legs”. *Robotics and biomimetics*, 4(1):1–14, 2017.

[47] G. Reina and M. Foglia. On the mobility of all-terrain rovers. *Industrial Robot: An International Journal*, 40(2):121–131, 2013.

[48] R. Galati and G. Reina. Terrain awareness using a tracked skid-steering vehicle with passive independent suspensions. *Frontiers in Robotics and AI*, 6:46, 2019.

[49] W. G. Ata and A. M. Salem. Semi-active control of tracked vehicle suspension incorporating magnetorheological dampers. *Vehicle System Dynamics*, 55(5):626–647, 2017.

[50] Tianyou Guo, Junlong Guo, Bo Huang, and Huei Peng. Power consumption of tracked and wheeled small mobile robots on deformable terrains–model and experimental validation. *Mechanism and Machine Theory*, 133:347–364, 2019.

[51] M. Balena, G. Mantriota, and G. Reina. Dynamic handling characterization and set-up optimization for a formula sae race car via multi-body simulation. *Machines*, 9(6):126, 2021.

[52] Jose-Luis Blanco-Claraco, Antonio Leanza, and Giulio Reina. A general framework for modeling and dynamic simulation of multibody systems using factor graphs. *Nonlinear Dynamics*, 105:2031–2053, 2021.

[53] S Akishita, S Kawamura, and K-i Hayashi. Laplace potential for moving obstacle avoidance and approach of a mobile robot. In *Japan-USA Symposium on flexible automation, A Pacific rim conference*, pages 139–142, 1990.

[54] S Sasaki. A practical computational technique for mobile robot navigation. In *Proceedings of the 1998 IEEE International Conference on Control Applications (Cat. No. 98CH36104)*, volume 2, pages 1323–1327. IEEE, 1998.

[55] Hamid Taghavifar, Subhash Rakheja, and Giulio Reina. A novel optimal path-planning and following algorithm for wheeled robots on deformable terrains. *Journal of Terramechanics*, 96:147–157, 2021.

[56] Azali Saudi and Jumat Sulaiman. Indoor path planning using harmonic functions via half-sweep arithmetic mean method. *International Journal of Computing and Digital Systems*, 8(5):524–528, 2019.

[57] Antonio Falcó, Lucía Hilario, Nicolás Montés, Marta C Mora, and Enrique Nadal. A path planning algorithm for a dynamic environment based on proper generalized decomposition. *Mathematics*, 8(12):2245, 2020.

[58] Panagiotis Rousseas, Charalampos Bechlioulis, and Kostas J Kyriakopoulos. Harmonic-based optimal motion planning in constrained workspaces using reinforcement learning. *IEEE Robotics and Automation Letters*, 6(2):2005–2011, 2021.

[59] Florian Cordes, Frank Kirchner, and Ajish Babu. Design and field testing of a rover with an actively articulated suspension system in a mars analog terrain. *Journal of Field Robotics*, 35(7):1149–1181, 2018.

[60] Florian Cordes and Ajish Babu. Sherpatt: A versatile hybrid wheeled-leg rover. In *Proceedings of the 13th International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS 2016)*, 2016.

[61] Mauro Bellone, Giulio Reina, Luca Caltagirone, and Mattias Wahde. Learning traversability from point clouds in challenging scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 19(1), 2018.

[62] Giulio Reina, Annalisa Milella, and Rocco Galati. Terrain assessment for precision agriculture using vehicle dynamic modelling. *Biosystems engineering*, 162:124–139, 2017.

[63] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[64] Vladimir Vapnik. *The Nature of Statistical Learning Theory.* Springer Science & Business Media, 2013.

[65] Kai-Bo Duan and S Sathiya Keerthi. Which is the best multiclass svm method? an empirical study. In *International workshop on multiple classifier systems*, pages 278–285. Springer, 2005.

[66] Thomas G Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of artificial intelligence research*, 2:263–286, 1994.

[67] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[68] Yi Lin, Yoonkyung Lee, and Grace Wahba. Support vector machines for classification in nonstandard situations. *Machine learning*, 46:191–202, 2002.

[69] Junlong Guo, Tianyou Guo, Ming Zhong, Haibo Gao, Bo Huang, Liang Ding, Weihua Li, and Zongquan Deng. In-situ evaluation of terrain mechanical parameters and wheel-terrain interactions using wheel-terrain contact mechanics for wheeled planetary rovers. *Mechanism and Machine Theory*, 145:103696, 2020.

[70] Qinpei Zhao and Pasi Fränti. Wb-index: A sum-of-squares based index for cluster validity. *Data & Knowledge Engineering*, 92:77–89, 2014.

[71] Javier Garcia De Jalon and Eduardo Bayo. *Kinematic and dynamic simulation of multibody systems: the real-time challenge.* Springer Science & Business Media, 2012.

[72] Justin Madsen, Toby Heyn, and Dan Negrut. Methods for tracked vehicle system modeling and simulation. *University of Wisconsin, Madison*, 2010.

[73] Andrea Nicolini, Francesco Mocera, and Aurelio Soma. Multibody simulation of a tracked vehicle with deformable ground contact model. *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, 233(1):152–162, 2019.

[74] The University of Arizona. Hirise, digital terrain models. `https://www.uahirise.org/dtm/` [Last accessed: 16 October 2023].

[75] Randolph L Kirk, Elpitha Howington-Kraus, Mark R Rosiek, Jeffery A Anderson, Brent A Archinal, Kris J Becker, DA Cook, Donna M Galuszka, Paul E Geissler, Trent M Hare, et al. Ultrahigh resolution topographic mapping of mars with mro hirise stereo images: Meter-scale slopes of candidate phoenix landing sites. *Journal of Geophysical Research: Planets*, 113(E3), 2008.

[76] Randall J LeVeque. *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems.* SIAM, 2007.

[77] Jo Yung Wong. *Theory of ground vehicles.* John Wiley & Sons, 2022.

[78] Jo Yung Wong. *Terramechanics and off-road vehicle engineering: terrain behaviour, off-road vehicle performance and design.* Butterworth-heinemann, 2009.

[79] BM Wills. The measurement of soil shear strength and deformation moduli and a comparison of the actual and theoretical performance of family of rigid tracks. *J Agric. Engng Res.*, 8(2):115–131, 1963.

[80] Mieczyslaw Gregory Bekker. *Introduction to terrain-vehicle systems. part i: The terrain. part ii: The vehicle.* University of Michigan Press, 1969.

[81] Chengchao Bai, Jifeng Guo, Linli Guo, and Junlin Song. Deep multi-layer perception based terrain classification for planetary exploration rovers. *Sensors*, 19(14):3102, 2019.

[82] C Vasileiou, A Smyrli, A Drogosis, and E Papadopoulos. Development of a passive biped robot digital twin using analysis, experiments, and a multibody simulation environment. *Mechanism and Machine Theory*, 163(1):104346, 2021.

[83] ISO-8608:1995. Mechanical vibration-road surface profiles reporting of measured data. 1995.

[84] Giulio Reina, Antonio Leanza, and Arcangelo Messina. On the vibration analysis of off-road vehicles: Influence of terrain deformation and irregularity. *Journal of Vibration and Control*, 24(22):5418–5436, 2018.

[85] Giancarlo Genta. *Introduction to the mechanics of space robots*, volume 26. Springer Science & Business Media, 2011.

[86] Angelo Ugenti, Fabio Vulpi, Annalisa Milella, and Giulio Reina. Learning and prediction of vehicle-terrain interaction from 3d vision. In *Multimodal Sensing and Artificial Intelligence: Technologies and Applications II*, volume 11785, pages 167–173. SPIE, 2021.

[87] Andrew R Conn, Nicholas IM Gould, and Philippe L Toint. Trust-region methods. *Society for Industrial and Applied Mathematics*, 2000.