Bachelor Project

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Control Engineering

Topological Exploration and On-line Terrain Classification for Hexapod Walking Robot

Vojtěch Kabelka

Supervisor: doc. Ing. Jan Faigl, Ph.D.
May 2016

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Control Engineering

BACHELOR PROJECT ASSIGNMENT

Student: **Vojtěch Kabelka**

Study programme: Cybernetics and Robotics
Specialisation: Systems and Control

Title of Bachelor Project: **Topological Exploration and On-line Terrain Classification for Hexapod Walking Robot**

Guidelines:

1. Familiarize yourself with adaptive motion control [1], and terrain classification method [2] for the hexapod walking robot.2. Develop a wall-following control strategy for the hexapod walking robot.3. Proposed and develop a method for topological exploration of the terrain types based on wall-following and terrain classification.4. Verify the proposed approach experimentally using the real hexapod walking robot.

Bibliography/Sources:

[1] J Mrva, J Faigl: Tactile sensing with servo drives feedback only for blind hexapod walking robot. RoMoCo 2015: 240-245[2] J Mrva, J Faigl: Feature Extraction for Terrain Classification with Crawling Robots. ITAT 2015: 179-185

Bachelor Project Supervisor: doc. Jan Faigl Ing., Ph.D.

Valid until the summer semester 2016/2017

L.S.

prof. Ing. Michael Šebek, DrSc.
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, February 1, 2016

Acknowledgements

I would like to thank my supervisor doc. Ing. Jan Faigl, Ph.D. for his wise and critical comments and consistently pushing the envelope to achieve better results.

My appreciation also goes to other members of the Computational Robotics Laboratory (FEE, CTU in Prague), namely: Bc. Martin Stejskal for the introduction into the hexapod a walking robot; Ing. Jakub Mrva for theoretical background and useful insights; and finally Ing. Petr Čížek for extensive help with the experiments and their evaluation, pinpointing essential pieces of related work and most importantly for reviewing this text in a thorough manner.

Last but not least, I thank my family for their unlimited support.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, May 27, 2016

....
Vojtěch Kabelka

Abstract

In this project, we examine one of desired key skills of mobile robots — topological exploration in an unstructured rough environment. We aim to learn whether the proprioceptive sensing of terrain as the exploration subject is applicable. A novel method for an off-the-shelf six legged robot is proposed, which exploits the adaptive motion gait for rough terrain traversal, the on-line terrain classification, and autonomous behaviour in the form of wall-following. The designed method is extensively validated through series of indoor experiments.

The proposed overall solution shows promising outcomes and proves to be plausible, albeit the results are not faultless. Moreover, the dependency of the terrain classification correctness on the robot's speed is uncovered. For example it has been found that the robot crawling faster classifies ca. 20 % more correctly.

This thesis gains results that provide new insight into the field of proprioceptive sensing, terrain distinguishability and further exhibits the possibilities of the used robotic platforms, e.g., reliable autonomous behaviour in an unknown environment.

Keywords: topological exploration, online terrain classification, wall-following, hexapod

Abstrakt

V této práci se zabýváme jednou ze stěžejních činností mobilních robotů – topologickou explorací v nestrukturovaném prostředí. Cílem práce je zjistit, zda proprioceptivní vnímání terénu může sloužit jakožto předmět topologické explorace. Navrhovaná metoda topologické explorace pro komerčního šestinohého kráčejícího robota využívá adaptivní styl chůze, on-line klasifikaci terénu a autonomní chování ve formě sledování zdi. Výsledné řešení je ověřeno pomocí série experimentů v uzavřeném prostředí.

Navržené řešení produkuje slibné výstupy a může být označeno za validní, byť výsledky nejsou bezchybné. Během exprimentů byla navíc pozorována závislost správnosti terénní klasifikace na rychlosti, se kterou se robot pohybuje. Robot jdoucí rychleji klasifikuje cca o 20 % přesněji.

Práce přináší nové poznatky v oblasti proprioceptivního vnímání, odlišnosti terénů a dále ukazuje na možnosti použitého robota, např. spolehlivé autonomní chování v neznámém prostředí.

Klíčová slova: topologická explorace, online klasifikace terénu, sledování zdi, šestinohý kráčející robot

Překlad názvu: Topologický průzkum prostředí a on-line klasifikace terénu pro šestinohý kráčející robot

Contents

Figures

Tables

Chapter 1

Introduction

Computers reign the world. In pursuit of different kinds of benefits, for example to save time and money, to be more efficient and productive,... mankind passes more and more weekday routines on to the world of **robots**, **agents** and **automation** in general. These routines include heavy industry manufacturing [1], precise manipulation [2], traffic control [3], medical surgeries [4], etc. Based on the activity a human would have to do, appropriate type of "circuitry" is chosen.

1.1 Field of Interest

One of the most rapidly growing and also the most challenging branch of the above mentioned research area is **field robotics**. Although the definition of the term is not determined particularly, the Robotics Institute - Carnegie Mellon University describes it as follows:

> The term "field robotics" was created to distinguish an emphasis on robotics in unconstrained, uncontrived settings, typically outdoors and in the full range of operational and environmental conditions: robotics in the natural world.[5]

Another explanatory description of field robotics can be found in [6]. A field robot can act on its own, or be part of a group; behave autonomously or be controlled by an operator. Related activities — field robotics participate in — extend over mining, agriculture, environmental monitoring and patrolling and rescue missions, beyond the space exploration. Nevertheless, whatever the task is, oftentimes a kind of **environment exploration** is required.

Environment exploration mostly concerns learning an unfamiliar area – with respect to some characteristics (metric dimensions, colour of wall, light intensity, **terrain**,...). The term can be viewed from two standpoints: straightforward and hidden. The former considers exploration as a task or subtask itself, i.e., a discipline. Purposes of carrying out such a task can differ from problem to problem but they can be divided into two main streams: exploration as a building block within a greater task (e.g., rescuing victims of earthquake from demolished buildings,...), or exploration as the main goal (e.g., to create a map of human-inaccessible cave,. . .). On the other hand, the latter takes exploration as a necessary instrument for **place recognition** – the crucial quality for any **autonomous** behaviour.

The quality of place recognition is expressed in the ability of the mobile robot to recognise an already visited place. A robot with such a feature is able to orient itself in an unknown environment, interpret it in the right manner and use the gained knowledge in future. Either metric or **topological** exploration can be utilised in a place recognition problem. The latter does not consider exploring based on metric dimensions, but rather a specific feature/attribute of the given environment.

There are various types of field robots. Taking into account only ground as an environment, the basic division of robots is into wheeled and **legged** ones [7]. As they are, more interest has been given to the area of wheeled robots so far since they are easier to control and faster in general. However, with growing need for the deployment of robots in the rough terrain conditions, wheelers happen to reveal their disadvantages. Conversely, crawlers – e.g., **hexapod** – prove to be suitable for challenging terrain applications mainly because of a higher number of degrees of freedom (DOF).

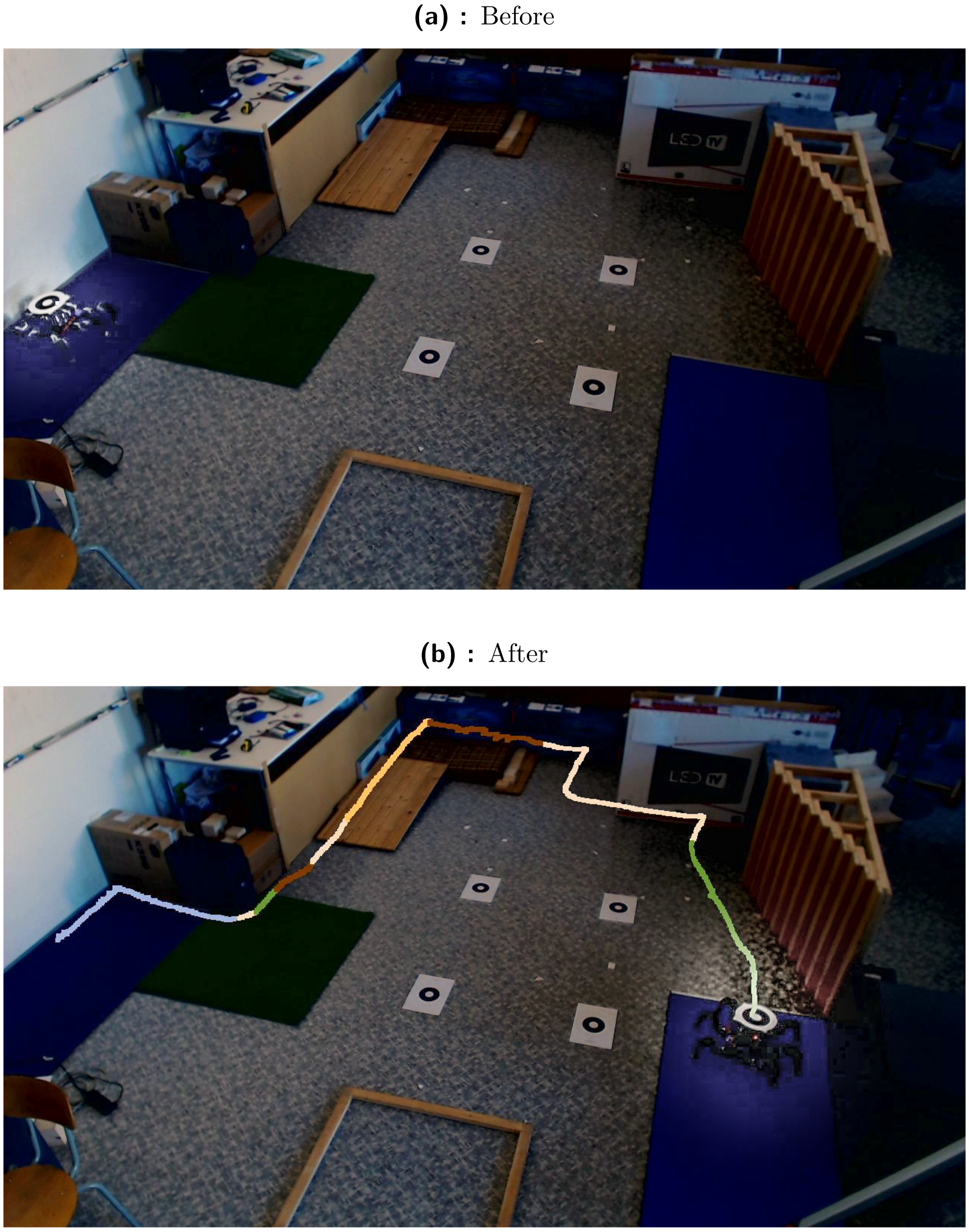
1.2 Goal & Motivation

The highlighted words above determine the content of the following work. The profound goal is to test whether the terrain as an environment feature (defining the subject of topological exploration) can be utilised for autonomous place recognition task using an off-the-shelf low-cost hexapod walking robot in unstructured uneven environment. In order to obtain results on the presented test, we propose to accomplish the following subgoals: online terrain classification, autonomous behaviour of a certain level (i.e., using wall-following) and an ability of rough terrain traversing by the considered walking robot. The modularity is to be utilised, where building blocks contribute to the main task. The next chapters and sections provide with more details and insight into each of the proposed building blocks. The background constraint is to keep the final platform cheap, with as little as possible sensory equipment. Figure 1.1 illustrates the above mentioned.

Whether the approach is successful or not, the procedure of obtaining results can bring new insight into environment topological exploration using solely proprioceptive sensing. Moreover, it can offer a different perspective on benefits of terrain classification performed by an off-the-shelf hexapod walking robot and reveal new potentials of such a platform in general. Last but not least, based on the made effort, an autonomous behaviour of the hexapod can be then further enhanced.

1.3 Thesis Structure

The rest of the thesis is structured as follows. Chapter 2 provides an overview of the related work and approaches to the given topic. Chapter 3 specifies the problem and lists the provided and used resources and equipment. It is followed by Chapter 4 which presents the adopted results which are further extended by this thesis. The proposed solution and its overall concept are presented in Chapter 5 focusing mainly on the design standpoint. Chapter 6 takes a closer look at the implementation of the designed solution. Moreover, practical issues which have been encountered are addressed in this chapter. Chapter 7 includes experiment description and results presentation. The discussion of experiments' results follows in Chapter 8. The connections between the goals which have been stated by the thesis and the obtained results are presented. The chapter also presents possible improvements and development in future work. Chapter 9 summarises and concludes the work.



(a) : Before

(b) : After

Figure 1.1: Thesis Goal

Chapter 2

Related Work

To give broader perspective, a brief survey of other approaches and works related to the presented task follows. If possible, our focus lies on works concerning similar hardware setup and conditions.

2.1 Topological Exploration

Topological exploration/mapping is performed to obtain semantic representation of the examined environment. Such representation comprises information about places, objects, shapes, etc. Therefore, robots utilising topological mapping are able to perceive the world with a more human-like perspective.

An extensive list of more than a hundred approaches to topological and topometrical mapping is presented in article [8]. The approaches are divided into two groups: methods based on place labelling vs. methods based on semantic labelling. The former is addressed in, e.g., [9], [10], [11]. These methods assign semantic labels to places or regions of the accessible work space of the robot.

In [9], the authors present a framework for topometrical mapping of an indoor environment. Paper [10] presents an outdoor topological exploration system based on visual recognition. In their experiments, the mobile robot followed paths in the park environment mapping intersections and building a topological map which can be later used for navigation. Similar approach deployed in an abandoned mine exploration scenario is addressed in [11].

Methods considering semantics of the examined environment assign semantic labels to parts or objects of the perceived structure of the examined environment. Regarding the outdoor exploration, these structures comprise, e.g., traversable terrains, trees or man-made structures. When exploring indoors, robots label walls, ceilings, doors [12] or significant objects appearing in the given environment. Based on the set of pre-learned objects, which are present in the environment, a particular room type can be determined and semantically labeled [13], [14].

However, most of the existing topological exploration techniques utilise exteroceptive[1] sensing (mainly vision and range measurements), our proposed method is based solely on the proprioceptive information.

2.2 Autonomous Behaviour/Wall Following

Wall-following is a well-defined form of autonomous behaviour. Among the frequently used approaches belongs a deployment of fuzzy controllers (FC) – i.e., a set of rules to control the robot based on expected behaviour, in this case wall-following task. This topic is discussed specifically for hexapod walking robots in [15], [16]; and for mobile robots in general in [17]. The FCs do not require a mathematical model in order to be able to control the system. However, they must be learned and tuned using reinforcement-type learning method such as a differential evolution [15] or they can be designed manually [16]. Paper [18] describes the usage of Kalman filter in addition to the FC for the wall-following procedure. The method enables to avoid obstacles and deal with erratic environment changes.

Another controller from the field of evolutionary robotics[2] which determines robot's behaviours is based on neural networks (NN). Work [19] introduces such an approach. Before NN-based controller can be deployed, its parameters (the number of layers and nodes in each layer) must be determined in advance. Controller learning is based on training data from the robot sensory equipment, e.g., 24 ultrasound sensors (creating circle-shape pattern) are exploited in [19].

More conventional methods — using the classic control theory — are presented in [20] and [21]. Within these works, ultrasonic-based and sonar sensory equipment is utilised.

2.3 Rough Terrain Traversing

In principle, three approaches to the rough terrain traversal with the legged robots exists. The simplest one exploits the mechatronic design of the robot as for instance the RHex[3] terrain traversing platform [22].

Secondly, advanced motion planning methods – selecting individual footholds of the walking robot – as addressed in the work by Dominik Belter et al. [23] and [24] develop complex solutions combining sophisticated multi-sensor (mostly exteroceptive) systems with subtle algorithms. Such methods create a representation of the robot's surrounding environment.

Finally, the solutions lying in between the two above utilise a-priori learned motion primitives (motion gait) together with the proprioceptive/tactile sensing to detect

[1] *exteroceptive* = relating to stimuli that are external to an organism, *proprioceptive* = relating to stimuli that are produced and perceived within an organism, especially those connected with the position and movement of the body (Oxford Dictionary of English)

[2] The research area which focuses on creating autonomous behaviours by learning.

[3] `http://www.bostondynamics.com/robot_rhex.html`

the contact point between the leg and the ground. These more low-key solutions utilise either force/touch at the tips of individual legs [25] or the force/torque values provided by the servo drives as the adopted adaptive motion gait [26], [27]. Exploiting the intelligent servo drive feedback offers a solution to avoid increasing DOF by adding passive actuator.

2.4 Terrain Classification

Same as above, the terrain classification procedures for mobile robotics can be divided based on the sensation type being exploited: exteroceptive and proprioceptive. The former is also called non-destructive – i.e., there is no contact between the robot and the terrain being classified. Paper [28] examines the former utilising depth measurements from a laser range finder. Another approach is to use a vision system. Such an approach is utilised in [29]. A promising feature of the method is the variable-length data representation for each terrain type.

The terrain classification methods based on proprioceptive sensing are of our main interest. Authors in [30] utilise additional sensory equipment in the form of ground force sensors. Sensors of similar function are exploited in [31]. In order to prevent mounting extra sensors, intelligent servo drives feedback can be utilised as in [32]. The adopted approach addressed in [33] enhances [32] w.r.t. extending the practicability of the terrain classification procedure to rough uneven environments.

For more advanced applications both types can be combined into one system. Deployment of such a system in a planetary exploration task is introduced in [34]. As an interesting piece of the recent work is mentioned [35], which proposes to use the acoustic emission during the crawling of the robot to distinguish individual terrain types.

Chapter 3

Problem Statement

Even though briefly mentioned in Section 1.2, this chapter is dedicated to detailed description of the given task, assumptions/constraints and resources used in this project.

3.1 Task

The main goal of this work is to allow an off-the-shelf hexapod walking robot to perform autonomously a topological exploration/place recognition based solely on terrain classification. In particular the online terrain classification exploits proprioceptive sensing which utilise the robot servo drives feedback only. During the area exploration, the robot may encounter various rough terrains and, thus it shall be capable of facing such a challenge successfully. For the robot to be able to act autonomously without any external control, a wall-following technique based on infrared (IR) distance/proximity measurement is utilised. When the robot finishes its task, it has obtained topological information about the examined area related to terrain and knowledge of finding the terrain which was given to search for.

Following hypotheses (in rather provisional than rigorous form) are stated as a formulation of motivations and objectives to be reached by this thesis:

H1 **For the given hardware setup, the autonomous behaviour in the form of wall-following enables performing the topological exploration using the given strategy.**

H2 **Successful terrain classification is independent on the speed the given hexapod is moving with.**

H3 **The given hexapod can classify and distinguish terrains of arbitrary qualities and similarities with 90 % accuracy.**

3.2 Assumptions

There are few assumptions and constraints related to the problem presented above.

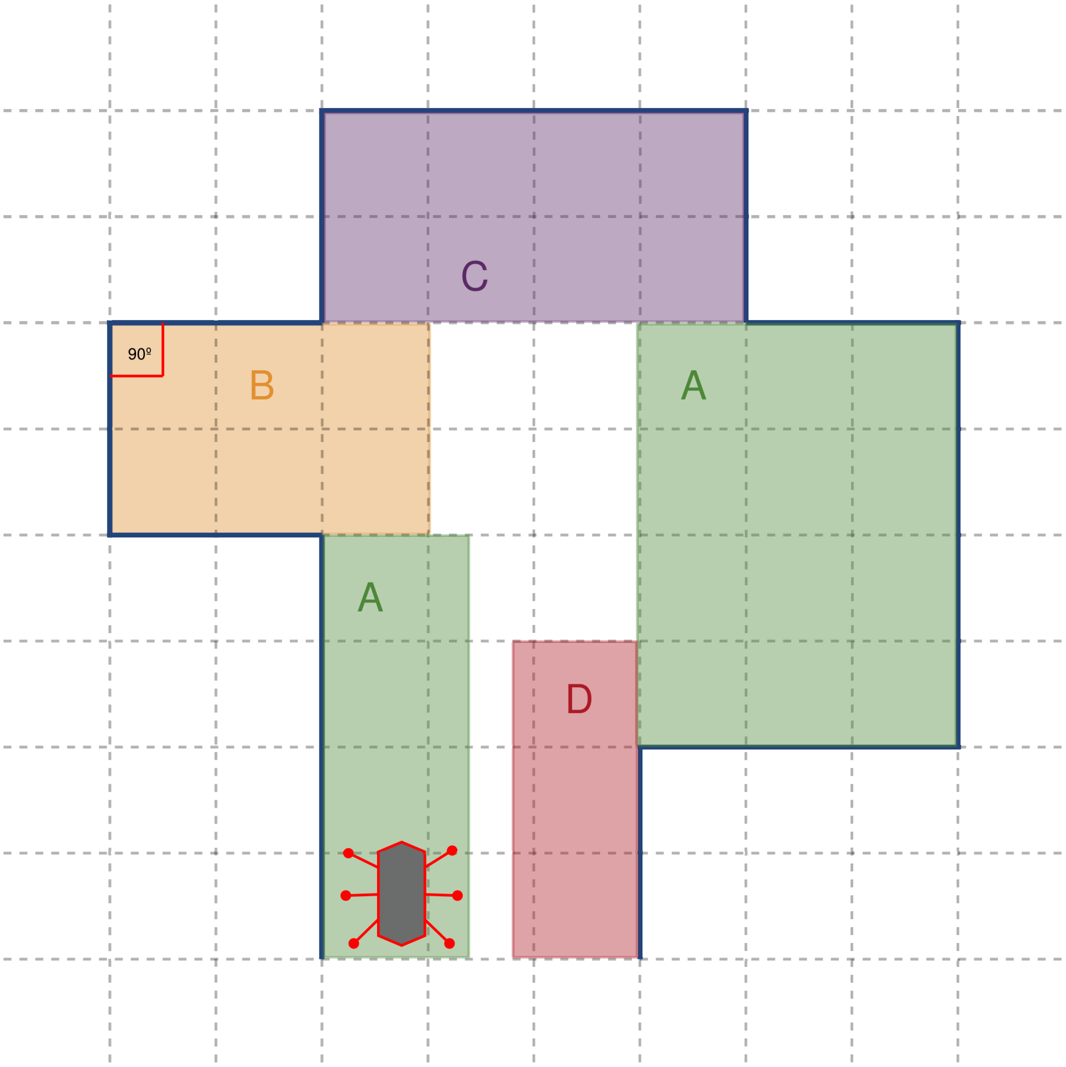


Figure 3.1: The 90° assumption shows a possible environment covered by different terrains A, B, C, D.

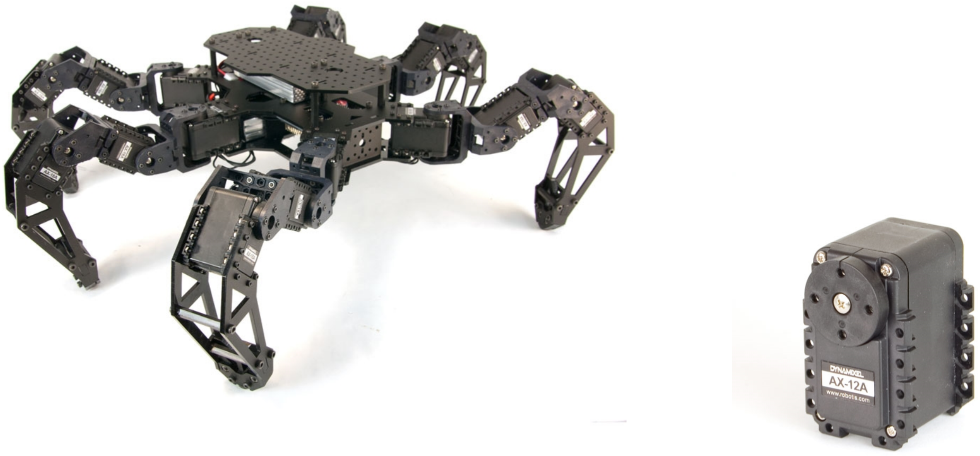
1. It is assumed that the robot has already familiarised itself with terrains it is going to traverse. In other words, the robot is not able to recognise terrain it has never encountered and therefore must be prepared in advance to accomplish its task of the topological exploration of the environment based on terrain classification. The assumption comes from the terrain classification — learning to be more precise — method.

2. For simplicity and to be more focused on the concept of the topological exploration solely based on proprioceptive sensing, the robot operational environment is constrained to be orthogonal (see Figure 3.1). Besides, the constraint is caused also by effort to keep expenses on proximity measurement hardware as low as possible. Even though such an assumption seems to be limiting, there are many practical scenarios when it is completely valid and does not influence the main concept of the proposed exploration procedure.

3.3 Used Equipment

In this section, detailed description of the hardware used for the experimental verification of the stated hypotheses is provided. The section addresses essential principles, which should be well-understood to give the reader a better perspective throughout forthcoming chapters.

3.3.1 Robotic Platform



(a): **(b):**

Figure 3.2: (a):PhantomX AX Hexapod Mark II with (b): Dynamixel AX-12, [36], [37]

A hexapod walking robot presents the main component of hardware. As mentioned above, the commercial platform – PhantomX AX Hexapod II[1] (depicted in Figure 3.2a) worth ca. $1,200 – is deployed for the experimental evaluation in this thesis. The robot consists of a body and six legs. Each leg is composed of three joints and links called the coxa, the femur and the tibia.

The hexapod crawls using 18 Dynamixel AX-12A intelligent servo drives by Robotis[2] (Figure 3.2b) placed in individual joints (i.e., three per leg). These actuators represent the entry-level among others provided by the manufacturer. The servo drives are connected in a daisy chain and communicate with the control unit via a 1 Mbps half-duplex UART interface. Each servo is capable of providing various feedback information such as the current joint position, temperature, load, input voltage, etc. A control table (EEPROM and RAM memory) containing information

[1] `http://www.trossenrobotics.com/hex-mk2`

[2] `http://en.robotis.com/index/product.php?cate_code=101010`

about actuator status and operation is used as an interface between control unit/user and the actuator. By writing to or reading from the control table, the higher-level controller is able to define the servo behaviour (e.g., torque off, write position, read position, etc.). [38]

Some of these features have been utilised within a robot motion planning and behaviour throughout this project and other works [26], [27], [33], [39].

3.3.2 On-board Computing

From physical point of view, controlling a robot can be achieved in two ways: on-board or off-board. Using the former, all computational power is present on the robot itself. On the other hand, the latter suggests using an external PC connected to the robot via a cable. If more power-consuming application is expected, the approaches can be combined. Even though the off-board offers more capabilities regarding computational power, the on-board provides convenience with respect to the mobility. Moreover, the on-board design is essential for autonomous behaviour. Therefore, the on-board approach is utilised in this work.

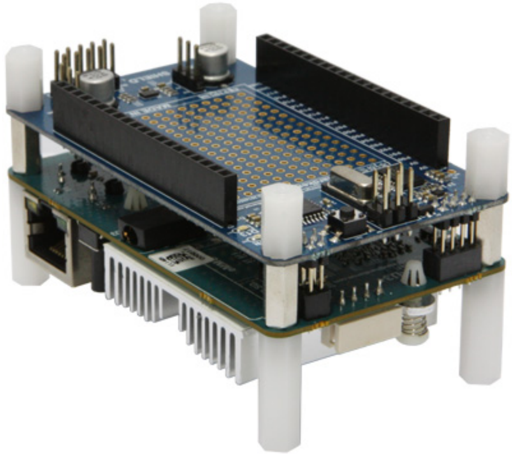


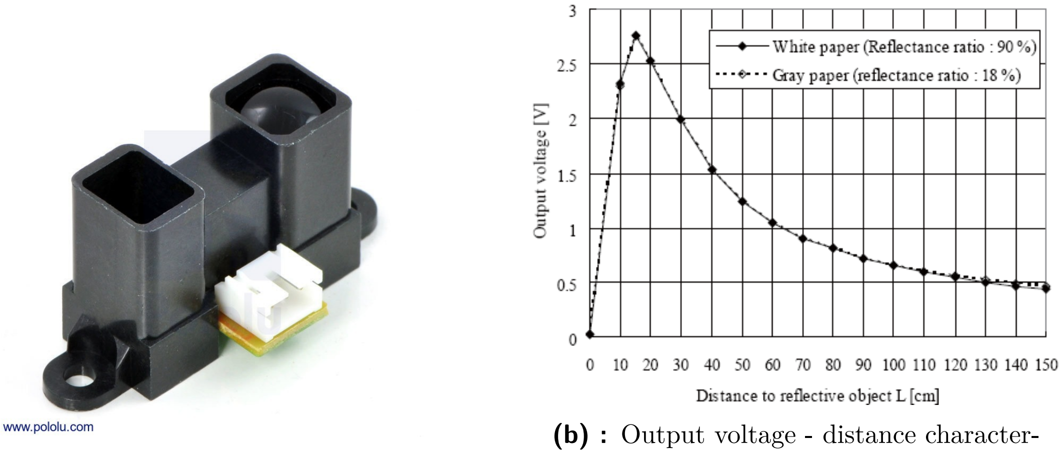
Figure 3.3: Odroid U3 with mounted IO shield, [40].

The Hardkernel[3] Odroid U3 unit together with the corresponding IO shield (both shown in Figure 3.3) have been chosen as the on-board control unit. The U3 board provides several high performance features such as an 1.7 GHz quad-core processor, 2 GB RAM memory or support of different OS (Android, Linux distributions,. . .). It enables various applications covering multimedia, security and robotics, among others. The IO shield used for analog interfacing with the distance measurement hardware has its own micro-controller and is connected to the Odroid unit via UART and I2C ports.

[3] `http://www.hardkernel.com/main/main.php`

3.3.3 IR Sensors

A proximity measurement sensory unit is deployed to make the robot capable of following a wall. Using proximity sensors based on reflecting IR light proves to be a low-cost straightforward solution, though may not be as precise and consistent as more expensive methods (discussed in Section 2.2).



(a) : IR Sensor, [41].

(b) : Output voltage - distance characteristics, [42].

Figure 3.4: Sharp IR Sensor GP2Y0A02YKOF, [43].

In particular, the Sharp[4] GP2Y0A02YKOF units (shown in Figure 3.4a) are utilised within this work. Using information from the official datasheet [44], the sensor produces output voltage corresponding to the distance from a source to the target: ca. 2.5 V for the minimum distance of 20 cm and ca. 0.4 V for the maximum distance of 150 cm (see characteristics in Figure 3.4b). The value is updated with the rate of ca. 26 Hz (i.e., it takes 38.5 ms on average to process a new measurement).

The IR sensor includes two optoelectronic units: position sensitive device (PSD) and infrared emitting diode (IRED), processing and other support circuitry. As seen in Figure 3.5, the performance of the distance measurement is based on the triangulation principle. IRED emits a focused beam of the IR light. A portion of it reflects on an object and returns back to the sensor where it is absorbed by the PSD. This component uses a couple-charged device (CCD) array to determine an angle of incidence and based on that it outputs an analog voltage value related to the distance between the sensor and the object being detected.

The above mentioned principle is very immune to varying ambient light conditions and to colour of the object to be detected. On the other hand, the reflection attribute of the given material plays an essential role when the detection is performed.

[4] `https://www.sharpsde.com/products/optoelectronic-components/sensors/distance-measuring-sensor/#productview`

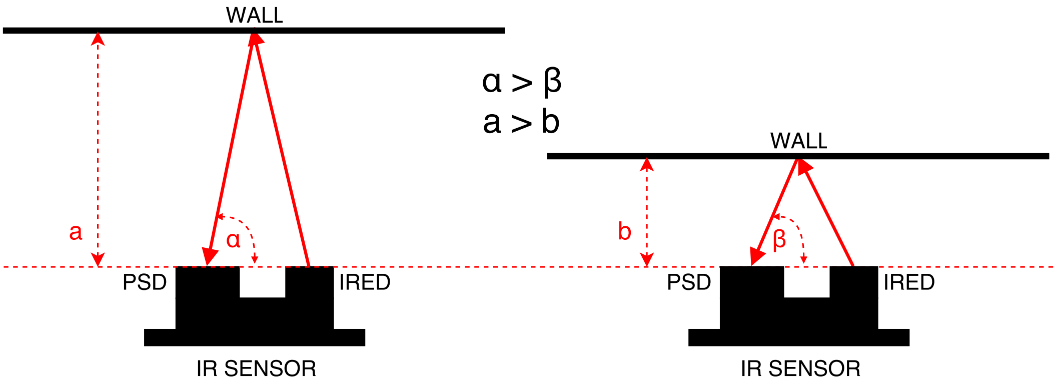


Figure 3.5: Triangulation principle used in the Sharp IR sensors to measure distance from the sensor to an object, e.g., wall. Adopted from [45] and modified for an explanation of the principle.

Chapter 4

Background Knowledge

In the following chapter, adopted works and theories for this thesis are presented and described in further details. The presented background consists of terrain classification and the classifier behind, the default control of the robot, adaptive motion for the given hexapod, and the PID controlling principle.

4.1 Terrain Classification

An appropriate approach must be chosen so that the hexapod is able to classify terrains: this approach must be robust enough to provide reliable results but still sustainable regarding system resources. In particular, the proprioceptive terrain classification utilising the SVM approach is used.

The terrain classification methodology has been adopted - and then implemented, from [33] which evolves mainly from [32]. While the approach presented in [32] does not offer the classification of rough terrains, [33] introduces enhancements such as adaptive gait and corresponding modifications which successfully tackle the issue. As a result, the hexapod is supposed to be able to classify correctly while traversing challenging terrains. If the hexapod's behaviour depends on the terrain classification, the ability to classify terrains on-line is required. This extension is introduced in [39] and utilised in this thesis.

The essential part of the classification is a feature extraction, i.e., the process of selecting key aspects of measured/available data and relating them to the classification procedure. The aspects are supposed to be chosen thoughtfully in the way that enables a reliable terrain type assessment during the classification.

Since the hexapod possesses no other sensory equipment, only the IR distance sensors (which do not participate in the classification procedure), all features emerge only from data provided by the servo drives. In particular, the time course of position error (i.e., differences between the desired servo position and the current one) of the two front legs servos is utilised to capture the influence (ground reaction force - slipperiness, stiffness, etc) of given terrain type.

To obtain sufficient amount of data, the last three consecutive gait cycles are taken into account. As the provided data are too sparse, a mathematical operation of Hermite spline interpolation is carried out to get a denser and more relevant position signal. Because of the belief that each terrain effects a leg/servo differently w.r.t. the phase the leg is currently in, the time domain is transformed into the gait-phase domain. The resulting position signal for a particular cycle segment is then examined for significant values such as the maximum, minimum, mean, median and standard deviation.

Unlike in [32], regular gait assumptions cannot be utilised due to the adaptive gait principle, i.e., phases are not of the same duration (depends on the terrain being traversed). Hence the gait-phase domain is divided not uniformly w.r.t. to time but based rather on the meaning of the phase. Moreover, owing to the fact that the periodicity aspect is missing within the adaptive gait, features from the frequency domain cannot be exploited. The ultimate feature vector comprises 240 attributes. Due to the content of the feature vector, when on-line classification extension is to be used, a gait cycle must be completed before a labelling can be performed.

4.1.1 SVM Overview

The SVM is a classifier with the following properties (assuming a binary problem, i.e., only two classes involved):

- **linear** - geometrically speaking, classifier composes a hyperplane (line, plane,...);
- **discriminative** - statistical/probabilistic model neither known nor estimated, the decision rule based on empirical data;
- **supervised learning/training** - to be trained classifier requires classified/labeled data in advance;
- **structural risk minimisation**;
- **margin between classes maximisation**;
- **support vectors** - only a portion of training data is utilised during the classification.

The approach evolves from- and further enhances the perceptron classifier [46]. Perceptron classifies an input based on the side the data sample is on w.r.t. a hyperplane. The process of training consists of finding such a hyperplane (its normal vector, i.e., a set of weights) that separates training data into mutually distinct sets. The classification is determined computing the following expression:

$$y(\vec{x}) = \begin{cases} \text{class } 1, & \text{if } \operatorname{sign}(\vec{w}\vec{x} + b) < 0 \\ \text{class } 2, & \text{if } \operatorname{sign}(\vec{w}\vec{x} + b) > 0, \end{cases} \tag{4.1}$$

where y is an output label, $\vec{x}$ a measurement, $\vec{w}$ a normal vector (aka a vector of weights) and b is a linear bias. The significant downside of the obtained solution $(\vec{w}, b)$ is that it is not guaranteed to be optimal. On the contrary, the solution computed using SVM classifier – which follows similar learning principle – is. In other words, for perceptron there exists an infinite number of solutions while for SVM only one. Figure 4.1 demonstrates the above mentioned and offers a graphical comparison between these two methods.

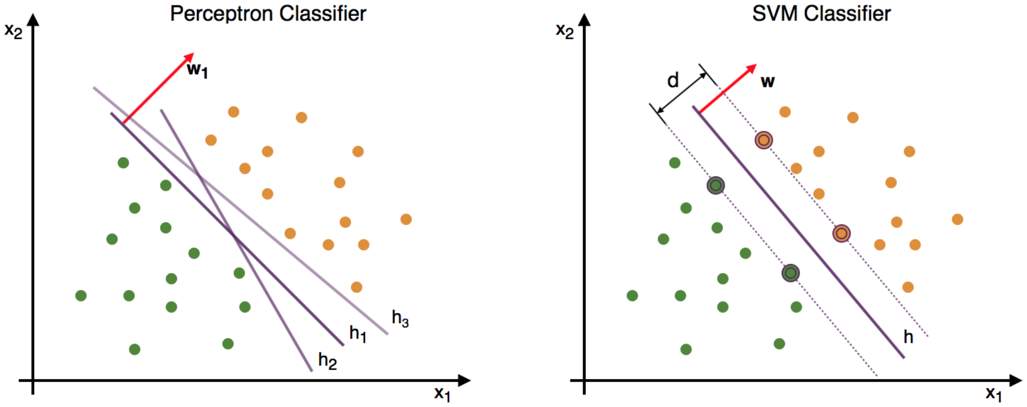


Figure 4.1: Perceptron vs. SVM. Comparison for two-class (yellow and green) problem. Dots represent 2D data samples, purple-coloured lines h_x represent planes and red-coloured arrows $\vec{w}$ planes' normal vectors. In the SVM's part, the data samples with greater size correspond to support vectors. Such vectors hold the equation $\vec{w}\vec{x} + b = \pm 1$. No other data can lie in-between borders (purple dashed lines).

Learning the SVM classifier (finding $\vec{w}$) is thus an optimisation problem. Reaching the optimality is w.r.t. maximising the margin d, i.e., the distance between the classes. The objective function is mathematically expressed as

$$d = \frac{2}{\|\vec{w}\|}. \tag{4.2}$$

Such an objective can be transformed into the field of quadratic optimisation as the minimisation of the expression

$$\frac{1}{2}\|\vec{w}\|. \tag{4.3}$$

However, this objective function is accompanied by constraints forcing the data points to be outside the margin area. A system of inequalities (its size depends on data set cardinality) must be considered and solved.

As a result, the above mentioned — called the primal problem — is converted into the dual problem. Even though it introduces an extra set of variables in a form of Lagrange multipliers, no constraints are present. Thus, in the computational point of view, the procedure is more convenient. The more detailed theory explaining the dual problem is out of the scope of this work and could be found in [47].

Multi-class SVM, One Versus All Principle

If more then two classes are considered for the classification, a slight modification in the classifying strategy must be introduced. One of the most frequently used method is one vs. all principle - also exploited within this project. During the training process, the method creates a number (whatever the number of classes is) of binary SVM classifiers where the first class is a proper one and the second class is the rest of data points. Each binary classifier has its own set of support vectors which are used for classification. An output label is determined comparing results of every classifier for the given measurement.

4.2 Hexapod Control

The hexapod control design from [26] has been exploited throughout this thesis. Using the design, the hexapod crawls based on a periodic motion strategy and thus the fundamental building block is one gait cycle. A gait cycle consists of a well-defined (order of legs, motion parameters) sequence of phases after which the hexapod finishes in the default position (legs positions relative to robot's body are as they started the cycle). A series of cycles comprises the hexapod's crawling.

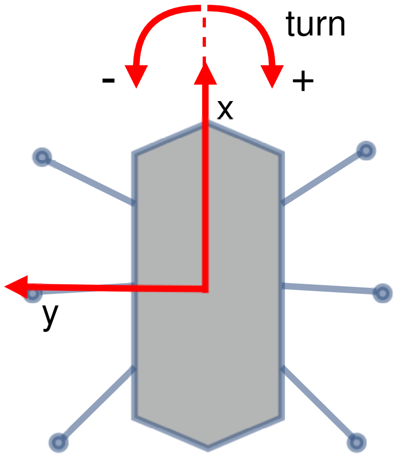


Figure 4.2: Hexapod Motion Orientation.

An interface between a user and the hexapod must be developed. The interface 1) defines a way to command the robot and 2) returns feedback information by the robot to the user. In the solution adopted from [26], a cycle command with a triplet of motion coefficients is given. The coefficients specifying the robot's motion for upcoming cycle are the following (Figure 4.2):

1. **x** - forward stroke,
2. **y** - sideways stroke,
3. **turn**.

These parameters are called primary. There is also a secondary set of parameters establishing the height of leg's lifting up above the ground and the depth of leg's dropping down beneath the ground level. Such a set is defined prior to the start and cannot be changed during the hexapod's crawling.

The primary coefficients represent relative values w.r.t. the maximal ones; hence values between -1 and 1, where the negative sign means a backward direction. The set of maximal values (limits) is defined prior to invoking robot's motion. The list of the limits includes:

- xstroke reach [mm];
- ystroke reach [mm];
- turn angle [-];
- lift-up distance [mm];
- drop-down distance [mm].

As an example, choosing the gait with the limits (50, 50, 0.1, 20, 0)[1] and the gait cycle command triplet (0.5, 0.2, 0) leads to a leg being moved by 25 mm in the x-axis forward direction and by 10 mm in the y-axis sideways direction without any turning. During the motion, the leg lifts up 20 mm and when dropping down, the leg's target is straight the ground without going beneath.

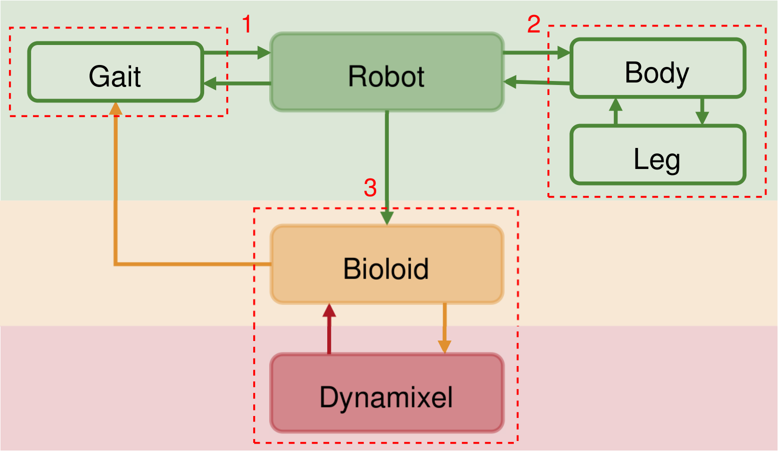


Figure 4.3: Hexapod's Controller Design. Dynamixel and Bioloid controllers are provided from the hexapod's manufacturer. The green-layer controller is the part adopted from [26].

The hexapod's motion controller – defining how the robot moves – is designed into layers. The diagram in Figure 4.3 illustrates the design. The connections between blocks and layers represent commands or information/status passing. In the Robot block, a cycle command with specified motion parameters is invoked. Then, the evolution steps are as follows:

1. the command divided into phases, for the given phase and leg; trajectory points are generated;
2. conversion from cartesian spatial coordinates to servo angles;

[1] The tuple ordering follows the order of the item list above.

3. low-level command containing angles servo drives are supposed to reach by end of the phase.

4.3 Adaptive Motion Gait

A specific motion planning strategy must be designed for a crawling robot to be able to tackle challenging and rough terrains. Different approaches are shortly discussed in Section 2.3. Throughout this project, the adaptive motion gait – for the hexapod a walking robot – developed in [26] and [27] has been utilised.

4.3.1 Gait Overview

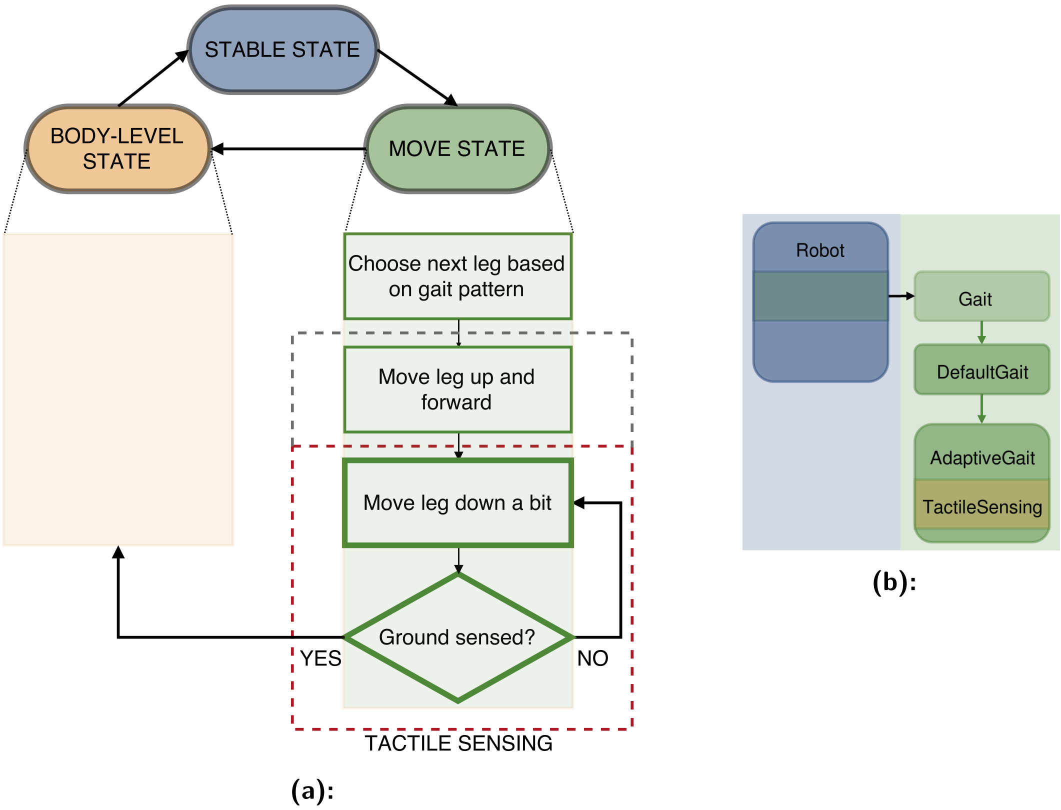


Figure 4.4: Adaptive Gait. Part (a) depicts the gait cycle diagram with the features of adaptive motion. Part (b) illustrates the gait design utilising the extended version of the default motion strategy. Such a design enables extensions for specific behaviours. The extended version contains tactile sensing procedure. Schema adopted from [26].

In Figure 4.4, a simple diagram of the adaptive gait cycle and its design are depicted. One gait cycle includes three main states:

Stable. All legs are on the ground not moving.

Move. First of all a group of legs based on the particular gait pattern is chosen, moved up and forward. The the legs are moved down in individual interpolation steps detecting the surface contact point.

Body-level. Once new footholds found, a transformation of body is computed. Using the obtained transformation, the hexapod is levelled up and next cycle can proceed.

The essential part of the diagram is the red-dashed box representing tactile sensing. In addition to the body levelling phase, it enables the robot to traverse rough and challenging terrains. The grey-dashed extension shows the part of the gait cycle where potential for detecting and hence avoiding an obstacle in a forward direction lies.

4.3.2 Tripod Gait

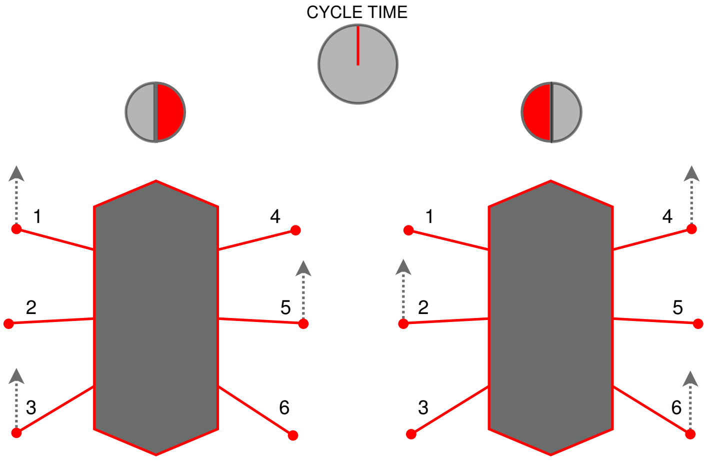


Figure 4.5: Tripod Gait Principle. The red/grey circles in the top of the figure represent cycle parts. The grey-dashed arrows select active legs for the given part of a cycle.

Although several different walking patterns for adaptive gait are possible, in the remaining part of this thesis, the tripod adaptive gait (see Figure 4.5 for graphical description) is considered. A cycle of tripod gait consists of eight phases in total, two of them totally support (all legs on the ground) – devoted to body-levelling. Legs are divided into two groups based on zigzag pattern. The first half of a cycle (not considering the support phase), legs 1, 3, and 5 are moving following the strategy from the green section in Figure 4.4a while legs 2, 4, and 6 are resting and supporting. The second half of the cycle, their roles switch. Once the cycle ends, the next one proceeds. The pattern offers a good balance between motion speed and stability — the crucial aspect for traversing rough terrains.

4.3.3 Tactile Sensing

The tactile sensing procedure lies at the core of the adaptive motion gait. The underlying principle is to approximate ground reaction force using only servo position. Owing to the structure of the leg and its motion, only a femur servo drive is examined. The approach utilises the ability of the used Dynamixel actuator to provide feedback

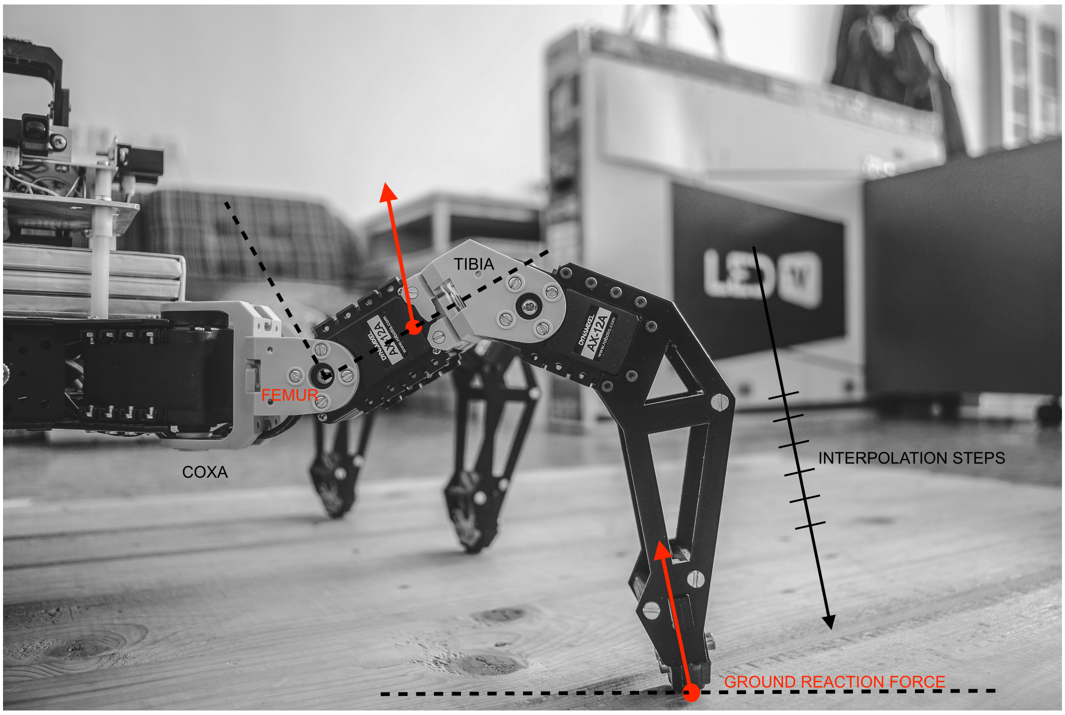


Figure 4.6: Tactile Sensing. Ground/obstacle detection during leg's down-motion. Ground reaction force impacts mainly a femur servo drive - the position error is examined.

information — the current position in particular. Because of the fact that a servo has only proportional controller to follow the reference value — the one given by user/control unit, i.e., the next servo position —, the error between the set point and the current value is present. When such an error reaches and overpasses a specified threshold value, it is assumed that the ground reaction force is responsible for the event and thus, a detection is claimed. To ensure that the described procedure works well, i.e., detects ground/obstacles fast, the sufficient granularity for interpolating must be set. On the other hand, the finer the interpolation is, the slower robot's leg moves. Moreover, static threshold values for the detection prove to be insufficient. Values dynamically changing based on the robot's current position and motion would be more suitable. The principle of tactile sensing is shown using the hexapod's real leg with additional graphics in Figure 4.6.

4.3.4 Robot Odometry

A user exploiting the hexapod control strategy described in Section 4.2 is not able to specify a set of exact coordinates where the robot is supposed to be after the next gait cycle (such information is hidden in lower levels of the control design). However, combining the three motion coefficients in arbitrary form, hexapod's motion behaviour is well-defined (although sometimes not precise) and thus, it can be utilised for higher motion planning.

4.4 PID Controlling Principle

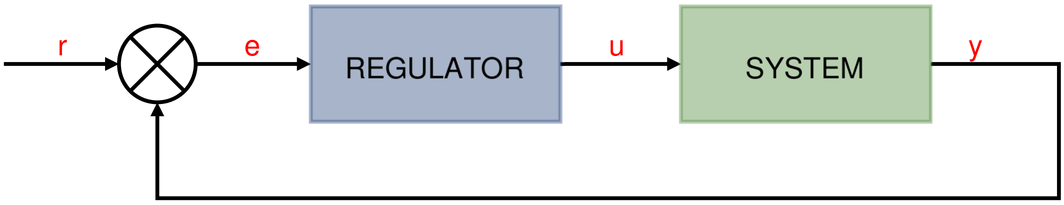


Figure 4.7: Feedback Controlling Scheme.

PID control is one of the most broadly used feedback controlling schemes. It is simple but still efficient and effective. In Figure 4.7, the simplest feedback control diagram is shown. Symbols used in the figure have the following meaning:

- **w** - system current output value;
- **r** - reference value a regulator tries to match with output value;
- **e** - error between reference and current output value;
- **u** - effort/action a regulator generates to fulfil its task of minimising the error.

Using PID controlling principle, the computation of the effort u follows (in continuous form) the equation

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\mathrm{d}\tau + K_d \dot{e}(t), \tag{4.4}$$

where $e(t) = r(t) - w(t)$ is the error at the given time t. The equation consists of three terms: proportional, integral and derivative. Each term is responsible for different behaviour (impact of the particular one is determined by corresponding multiplier K_x, where x serves as a term index):

- **proportional** - K_p, takes into account only the current values - K_p, too big leads to destabilising the system;
- **integral** - K_i, considers all past error up to the time, makes reaching zero error faster but causes system to oscillate;
- **derivative** - K_d, relates to the momentary change of the error, improves and smoothens the system's dynamic behaviour.

In the digital domain, Eq 4.4 can be discretised into

$$u[kT_s] = K_p e[kT_s] + K_i T_s \sum_{\tau=0}^{k} e[\tau T_s] + K_d \frac{e[kT_s] - e[kT_s - T_s]}{T_s}, \tag{4.5}$$

where $e[t] = r[t] - w[t]$, k represents discrete sampling indices and T_s is the sampling period. Operations of integration and derivation are replaced by summation and the backward difference, respectively.

Chapter 5

Proposed Solution

The thesis top-most goal in the form of the topological exploration using the off-the-shelf hexapod and other given resources (hardware and software) is complex and needs a well-thought solution strategy. The following chapter concerns the overall concept and individual designs of modules comprising the proposed solution.

5.1 Overall Concept

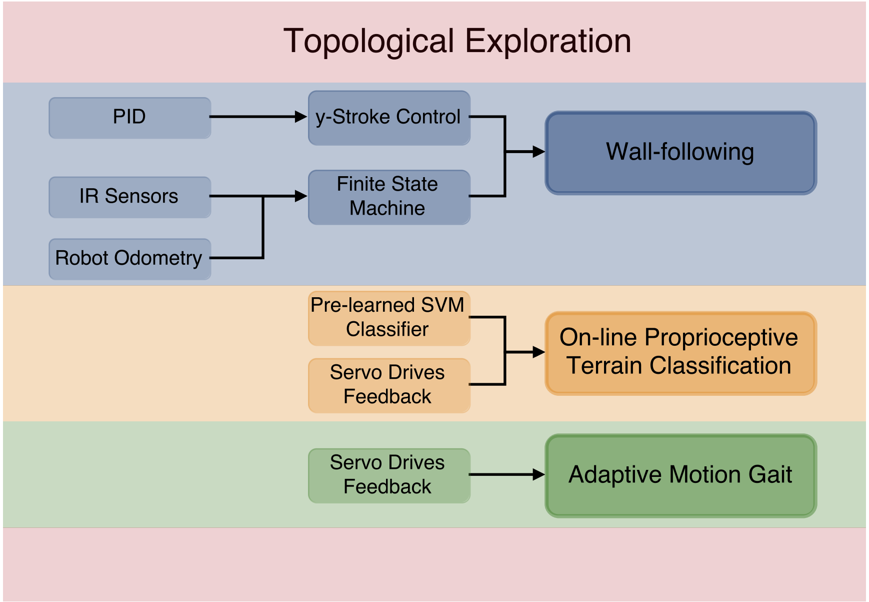


Figure 5.1: Solution Overall Concept

The presented topological exploration depends on mastering autonomous behaviour in a form of being able to follow a wall, adaptive motion gait for rough terrain traversing and on-line proprioceptive-based terrain classification. Figure 5.1 illustrates the proposed solution concept with the cornerstones which together comprise the whole design.

The resulting main application is designed as two-threaded in order to isolate executive part from data preparation part. The executive part includes controlling the hexapod, i.e., where and how it crawls (wall-following, motion planning); and classifying terrains. The data preparation part commands the IO shield (Figure 6.1a), reads and prepares all the data so they can be utilised by the executive thread.

5.2 Wall Following

The wall-following technique is based on information about walls being present in near surroundings. Taking into account the obtained information, the robot decides where to crawl next. Such a decision is made after taking into account several various aspects. A few strategies concerning how to follow a wall are mentioned in Section 2.2. In this work, the following modules for the procedure have been utilised: finite state machine using IR distance sensors (specification in Section 3.3.3) and naive/simple robot odometry; and y-stroke (sideways) control using the PID regulating principle.

5.2.1 Hexapod Control

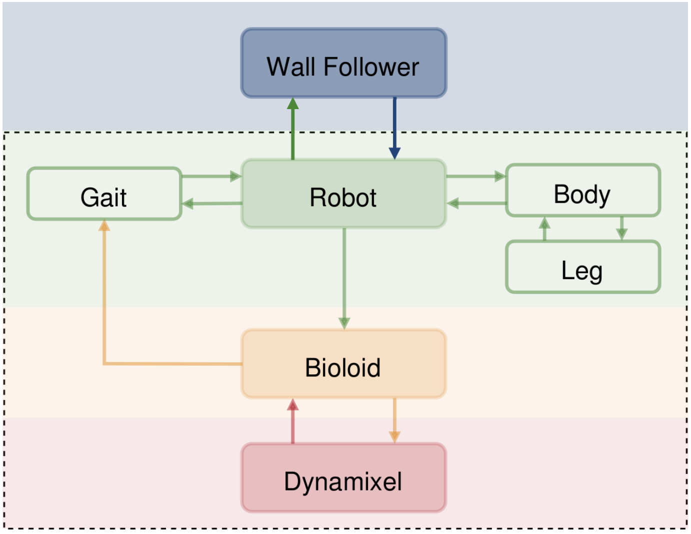


Figure 5.2: Extended Hexapod's Controller Design

Hexapod's controller must be accordingly modified in order to be able to perform autonomous behaviour in form of wall-following. The controller design evolves from the one described in Section 4.2 by adding a new layer on the top and the corresponding interface. Figure 5.2 contains the diagram illustrating the extended controller.

The top-most Wall Follower layer is related to the wall-following procedure and to moving on the 90° grid. It passes commands to the Robot layer defining where the hexapod is supposed to crawl next. The feedback information contains status and error messages if present.

5.2.2 Finite State Machine

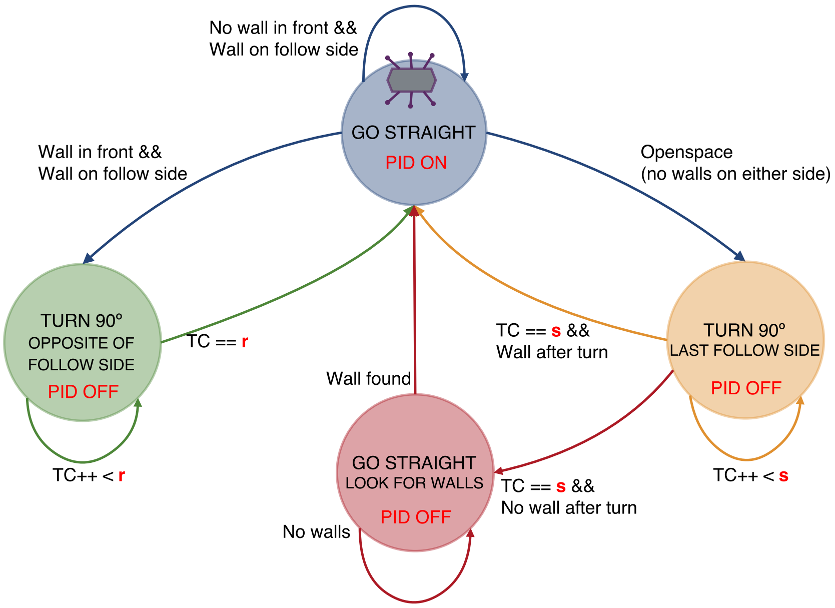


Figure 5.3: Wall-following FSM

The logic of wall-following is based on the finite state machine (FSM) theory. Such a principle is broadly used when the behaviour can be clearly divided into cases. Switching between states is invoked by a well-defined condition or combination of several ones. In Figure 5.3, the automaton for the wall-following procedure is depicted. The FSM consists of four states and corresponding interconnections related to the given conditions. r and s are static values representing how many turn cycles are needed to perform 90° turning. They have been determined partially based on the robot odometry, partially empirically. PID ON and PID OFF denote to whether y-stroke control to keep the same distance is active or idle.

5.2.3 IR Sensors

Using IR distance/proximity sensors for the wall-following task is low-cost, straightforward to implement, and not demanding of computational power. When such sensors have to be utilised, their placement and output data processing must be developed. Both possible usages of IR sensors:

1. distance measurement, i.e., the output value says how far the object is;

2. proximity measurement, i.e., is the object that far hence the output value reaches certain threshold;

are exploited. The former is used as the input for the y-stroke control procedure while the latter serves as the indication whether or not the wall is present.

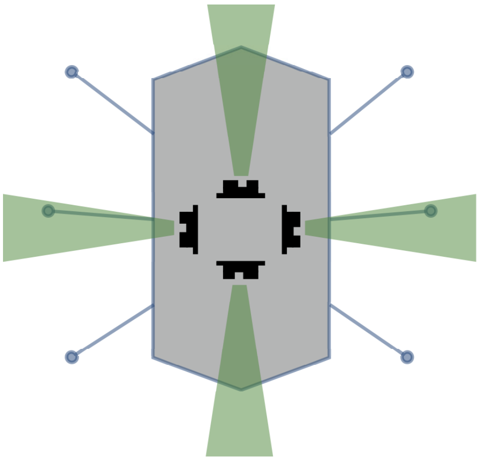


Figure 5.4: IR Sensor Placement. A green trapezium shows a field of view of a sensor. The underlaying condition of sensors having no object (e.g., robot's legs) blocking the view must be fulfilled.

Placement

A possible setup of the mounted IR sensor units can be seen in Figure 5.4. Sensors are placed symmetrically in 4 directions (cardinal directions from now on) to preserve as much versatility as possible. Such a placement enables a robot to follow wall on either left or right side; and to crawl either forwards or backwards.

Data Signal Processing

The IR sensors data processing takes place both in the IO shield (slave) and in the main computing unit Odroid (master). The two parts runs asynchronously. The designed procedure of processing the data comprises the following steps:

- **IO shield:**
  1. choose specified IR sensor;
  2. convert the sensor analog output value using analog-digital converter (ADC) with the resolution of 10 bits (i.e., values between 0 and 1023);
  3. store the digital output;
  4. iterate over all IR sensors.
- **Odroid:**
  1. request IO shield to send the data for all IR sensors
  2. store those which are required (e.g., based on which wall is followed) for upcoming decisions

To provide data only when they are desired, the concept of interrupt service routine (ISR) for the master-slave communication is utilised. Such a routine is fired immediately when the master asks for data. No other slave's instruction can be performed until the data are sent to the master.

5.2.4 Moving On Grid

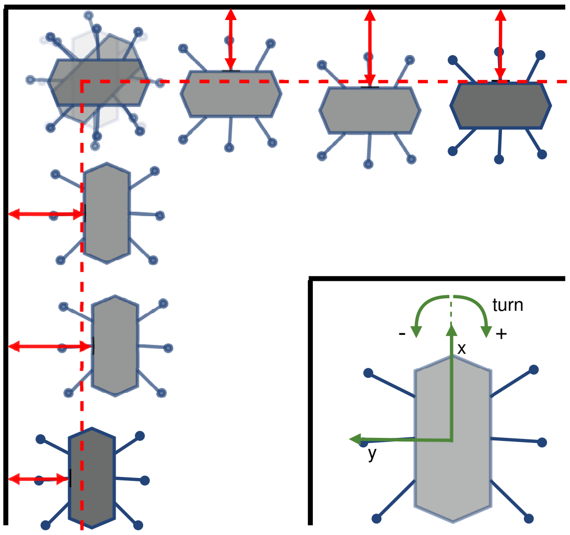


Figure 5.5: Moving On Grid. The robot starts in the left bottom corner and finishes in the upper right one. In the upper left corner a 90° turning is required. The red-dashed line represents reference distance the hexapod tries to keep. On the contrary, red arrows represent current distance between the wall and robot. To capture a real scenario, a slight error is present.

Straight Segments

Control of y-stroke is utilised so that the hexapod is able to reach an acceptable position after long forward crawling. The precision is required for further performance. In Figure 5.5, the function of such a control principle is shown. The controller is used to preserve the same distance from the wall as long as the hexapod is crawling. The design of the controller emanates from PID control theory (Section 4.4). The given hexapod represents the system of the PID scheme (Figure 4.7). For the given problem, the interpretation of symbols is as follows:

- the current output value w as the R sensor distance value;
- the reference r as the distance the hexapod is supposed to be from the wall (same units as the IR sensor value);
- the effort u as motion coefficient in the y direction, aka y-stroke;
- the error e as the difference between the measured and reference distances.

The reference value (aka set point) is static and does not change in time. The y-stroke control module runs periodically together with the IR sensors data processing independently on the main thread. Once new gait cycle starts, it provides the last computed action based on the last set of sensors reading. The module is turned off and on based on the state the hexapod is within the FSM (Figure 5.3).

90° Turnings

When a wall appears in front of the robot while moving in the grid, the hexapod proceeds with the 90° turning. The simple robot odometry for such task is exploited. The hexapod performs the exact number of gait cycles devoted solely to turn motion in order to achieve precise 90° turning – i.e., both the x-stroke and y-stroke are set to zero. The turning coefficient together with the number of cycles (constants r and s depicted in Figure 5.3) must be determined.

5.3 Adaptive Motion Gait

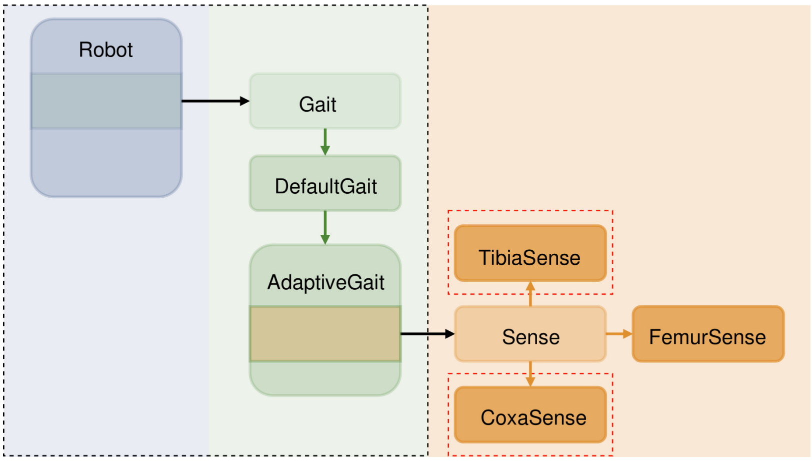


Figure 5.6: Modified & Extended Gait Design. The tactile sensing procedure is separated and placed into the sensing layer. Blocks in red-dashed boxes are not utilised.

The strategy for rough terrain traversing is used in almost the same form as described in Section 4.3. Only a gait design (the former depicted in Figure 4.4b) modification/extension is introduced. While the new design illustrated in Figure 5.6 preserves the same functionality, it establishes a new layer within the gait design devoted to sensing and thus, it broadens the possibilities regarding this matter. Such a design enables sensing not only via the servo drives placed in femur joints but also using other servo drives. Nevertheless, only the block with femur sensing from the extension is utilised throughout this thesis.

Chapter 6

Implementation & Practical Issues

In pursuit of completing the given task, several challenges must be tackled. Both software and hardware issues have emerged. This chapter maps these challenges and issues and discusses corresponding solutions.

6.1 Implementation

6.1.1 Software

Both the hexapod's controller and on-line terrain classification are implemented using object-oriented programming language C++[1]. A few noteworthy third-party libraries are utilised:

- **libdx** - the bottom-most layer of the hexapod's controller provided by servo drive manufacturer;
- **libalglib**[2] - library for linear algebra, interpolation, optimisation, data analysis and processing;
- **libsvm**[3] - library providing SVM algorithm for classification, regression and distribution estimation.

The program performed by the IO board is implemented using the ANSI C programming language. The methodology of accessing peripherals and IO board components via memory mapped IO is used.

6.1.2 Hardware

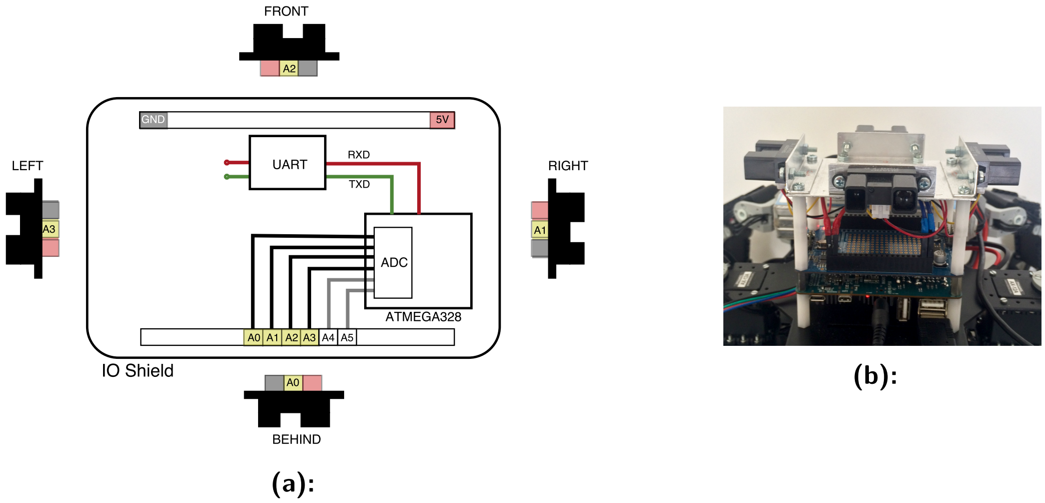
IR Sensors - IO Shield Interface

The IR sensors are mounted on an elevated aluminium platform (Figure 6.1b) to obtain an unblocked view. In Figure 6.1a, the schematic interface between IO shield and sensors is shown. The value wires ($\mathrm{V_{ox}}$) are connected to the IO board using analog pins $A0$ through $A3$. The board also provides GND and 5 V power supply ($\mathrm{V_{cc}}$) electrical potentials.

[1] `http://www.cplusplus.com/info/`

[2] `http://www.alglib.net`

[3] `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`



(a): **(b):**

Figure 6.1: IO Shield Interface & Real IR Sensors

6.2 Practical Issues

6.2.1 Moving On Grid

Moving on the 90^o grid must be precise in order to enable accomplishing the given task. Hence the designed solution must be robust.

PID tuning

The greatest challenge during implementing PID controller is to determine constants K_p, K_i, K_d. Especially when a mathematical model is not present – which is the case with the used hexapod. Therefore, an experimental approach has been utilised. For this project, one of the most naive but also straightforward method called Ziegler-Nichols tuning procedure is exploited [48]. Even though the method has not been rigorously proven, oftentimes it suits the needs. The procedure consists of a couple of steps:[49]

1. Use only proportional term, increase K_p until a system starts oscillating (the hexapod crosses a reference line after every action). Such K_p is called the critical gain and denoted K_c. Find the corresponding period of the oscillation T.

2. Decide whichever combination of terms is to be used. Based on that, determine the constant for each term which participates in the designed controller. For the hexapod performing the wall-following, only the proportional and derivate terms are used, the integral one is omitted.

IR Sensors - Dead Angles

The introduced placement of the IR sensors (Section 5.2.3) together with the narrow field of view cause dead angles to be present (graphically depicted in Figure 6.2).

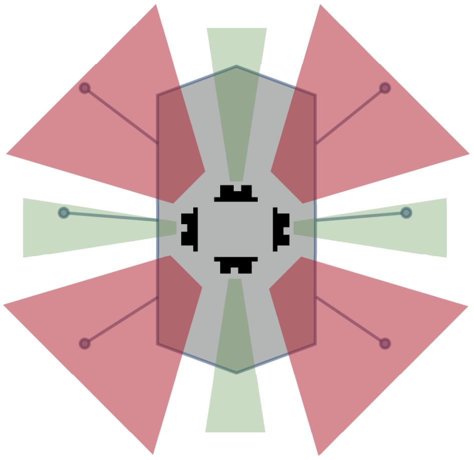


Figure 6.2: IR Sensor Placement - Dead Angles. A red trapezium introduces dead angle zone, where no sensor for the given placement can se.

Unless a radical change in IR placement is done, dead angles cannot be removed or suppressed. Such a phenomenon is not an issue for the overall robot's movement since the hexapod crawls only on the orthogonal grid. However, it complicates the orthogonal turning parts (i.e., the corresponding states in the FSM shown in Figure 5.3). Therefore, other methods are exploited.

IR Sensors - Data Signal Inconsistency

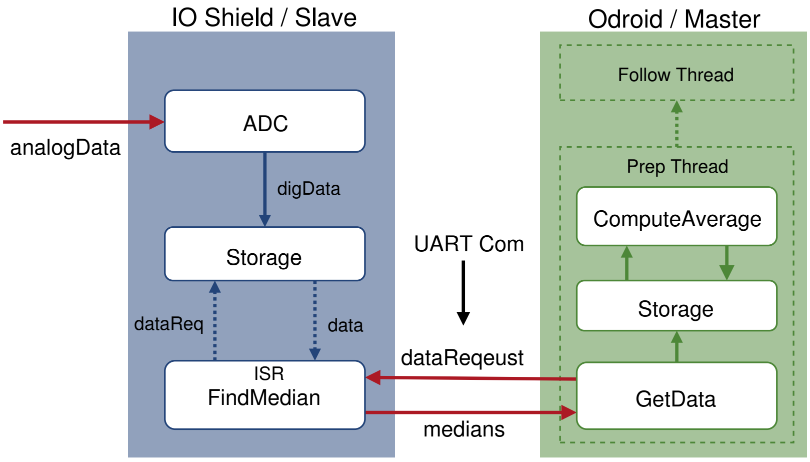


Figure 6.3: IR Sensor Signal Processing

If the data from IR sensors are processed as described in Section 5.2.3 with no additional operations, the resulting output is inconsistent, i.e., rapid changes, outliers are present. Therefore, basic filtering methods are utilised. Figure 6.3 contains the diagram showing the processing procedure with additional information.

The procedure consists of two basic operations: the median out of the last n data samples (carried out in the slave); and the average of the last m medians (in the master). The block FindMedian is fired through ISR, thus it runs asynchronously. On the contrary, the ADC performs conversion periodically at the frequency of ca. 25 kHz. The master's Prep Thread asks for data at the frequency of ca. 4 Hz. The master and the slave communicate over UART at 38,400 bps baud rate.

```
input  : An array of sensor values $senVals$ of size $4 \times n$, where $n \% 2 = 1$
output : Median values
initialise variables;
confirmed ← readVal();
if confirmed then
    for $i \leftarrow 0$ to 3 do
        tmpSenVals ← $senVals[i]$;
        tmpSenVals ← qsort(tmpSenVals);
        median ← tmpSenVals $[\frac{n-1}{2}]$;
        sendVal(median);
    end
end
```

Figure 6.4: Interrupt Service Routine FindMedian. The routine finds median values from the last n samples of the sensory input. The interrupt is fired by receiving a command from the master.

The last n samples of sensor data are stored in the slave's storage. When the master requires the current sensor value, an interrupt is fired. An algorithm depicted in Figure 6.4 describes the interrupt routine. Choosing median suppresses influence of data outliers which are present in the signal. Moreover, computing average of subsequent medians smoothens rapid changes in the data stream.

A solution for moving on the 90° grid using only the IR sensors information has been developed, i.e., the robot crawls straight until a wall appears in front, then it turns until a certain IR sensor value is reached, etc. However, the turning states in the FSM (Figure 5.3) have not been mastered due to the above mentioned issues. Turning parts are thus hard-coded based on the robot odometry.

Turning Parameters

As mentioned above, crawling on the 90° grid using only IR sensor data is not precise enough to successfully accomplish the given task. Thus, abilities of the hexapod's high-level motion controller and robot odometry (Section 4.3.4 and Section 5.2.4) are utilised.

Setting the constants (r and s from Figure 5.3) representing the number of gait cycles devoted to turning is found on the fact that the turning coefficient of value 1 causes the hexapod to turn ca. $\frac{1}{2}$ of a radian during one cycle. Using the turning coefficient of the value 0.5 then makes the need to perform six turning cycles. The turning procedure has been experimentally tuned, where subsequent turning (i.e., commands with different turning coefficients) has been found.

6.2.2 Tactile Sensing

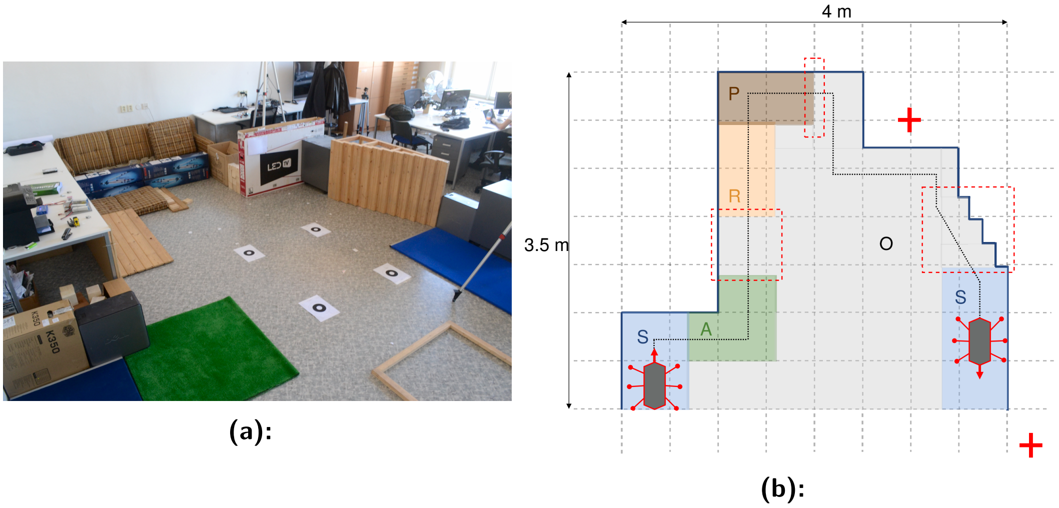
When tactile sensing (Section 4.3.3) is carried out, the detection threshold value plays essential role. Since not every servo drive performs equivalently from the hardware standpoint (may not be perfectly calibrated, etc), the value at which the detection is claimed varies from actuator to actuator. Because of the fact that a wrong ground detection effects the resulting robot's motion, all threshold values are experimentally found using the trial-error approach.

Chapter 7

Experiments & Evaluation

A set of validating experiments has been conducted, in order to test the hypotheses queried in Chapter 3. Hence, an appropriate evaluation technique has been introduced. This chapter describes the experiments and evaluates the corresponding outcomes. The outcomes are in the form of plots, photographs, and videos. Since an external visual localisation system is utilised within the experiments, its brief description is provided.

7.1 Arena Setup



(a):

(b):

Figure 7.1: Experiment Arena - (a): Real, (b) Schema

Due to better convenience regarding resources and equipment (power supply, computers, cables,. . .) and robot's manipulation, an indoor arena is chosen and built up for the proposed experimental verification. Figure 7.1a shows the laboratory experimental setup while Figure 7.1b illustrates its schematic structure. Coloured rectangular areas represent different terrains. Black-dotted line illustrates assumed trajectory and the red-dashed boxes show challenging areas regarding wall-following and adaptive motion gait. Red crosses represent placement of the cameras of the reference localisation system.

In total, five different terrains are present in the experimental field. List of them with description and provisional titles follows:

- **S** - desk covered with soft leather, softblue;
- **A** - artificial grass, artgrass;
- **R** - wooden ramp sloped ca. 15°, ramp;
- **P** - couch parts, pillow;
- **O** - linoleum office floor, office;

A segment with the stair-like pattern is included within the arena to further examine the ability of the wall-following technique with the terrain classification combined. Moreover, in order to demonstrate the technique together with the adaptive motion gait, the ramp and the border between pillows and office floor are set up. While traversing the border, the hexapod descends rapidly (ca. 20 cm vertically in 3 gait cycles) and thus, it is not well-balanced during that time.

7.1.1 WhyCon Localisation System

The external localisation and recording of the robot in the arena is used in order to inspect robot's behaviour, i.e., terrain classification and wall-following. Regarding the former, it allows to compare experiment results with ground truth. In other words, it is possible to contrast the terrain the robot is actually traversing with the terrain the robot thinks is stepping on. Speaking of the latter, the system is able to capture robot's movement hence the validity of the wall-following strategy can be examined.

The system utilised within this work emerges from an open source project called WhyCon[1]. From hardware point of view, USB web cameras Logitech C920[2] are used. The procedure is based on tracking targets (in a form of the specific visual pattern) in camera frame and then transforming tracked position into the user coordinate frame. The principles lying behind are thoroughly described in [50]. The localisation system setup is achieved with the following steps:

1. calibrate the cameras being used;
2. define the user/arena coordinate system using well-placed targets;
3. adjust contrast, brightness and exposition to enable reliable tracking.

Due to the dimensions of the arena, only one camera is not able to capture the environment in such manner that relevant position within the user frame (wherever the robot is crawling) is obtained. Hence two tracking cameras are required in order to provide more precise localisation.

[1] https://github.com/gestom/whycon-orig

[2] http://support.logitech.com/en_us/product/hd-pro-webcam-c920

7.2 Evaluation Methodology

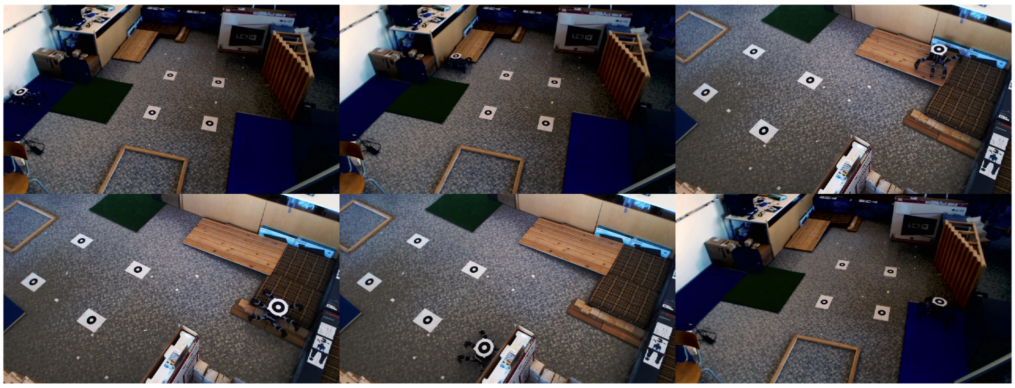


Figure 7.2: Hexapod Moving in Experiment Arena

The basic outline of the experiment is straightforward: while following the arena walls, the hexapod traverses the given terrains and classifies them. When the initial terrain is classified for the second time, the experiment terminates. The hexapod's proposed performance can be seen in Figure 7.2.

The performance of the designed wall-following strategy is shown by plotting the hexapod 3D position from the localisation system and comparing such plot with the scheme of the experiment arena. Besides, the strategy must be successful in order to proceed with the evaluation of the terrain classification procedure and the topological exploration.

Since using the visual localisation system, the ground truth labelling is known. It is obtained after analysing video captured during an experiment run. To thoroughly examine the classification method's abilities and find possible dependencies, the classification window is varied. In other words, a classification is not approved until c consecutive terrain predictions are the same. Parameter c is systematically altered and incidentally determines how many gait cycles from the end impact the classifier. After comparing the true labelling with the modified classification output, the value of c with the fewest misclassifications is evaluated as the best.

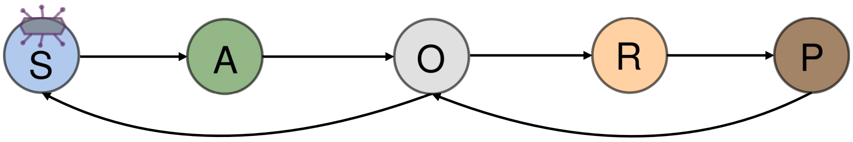


Figure 7.3: Ground-truth Topological Map. Each node represents corresponding terrain. The notation is based on the experiment arena and the list of terrains above. The hexapod icon indicates the initial position (terrain).

Based on the classification list obtained from the above described process, a topological map is created. The created topological map is in a form of directed graph – also expressible in the matrix format. As our map has five nodes (five terrains), 25 unique edges (crossings) are possible. The map is compared with the one emanating from the ground truth classification (illustrated in Figure 7.3). The experimental arena is set up not to have a sequence of two given terrains twice. Thus, the ground-truth topological map does not contain parallel edges. The map for the complete track is expressed in the matrix form as follows:

$$M_{GT} = \begin{array}{c} \\ \text{A} \\ \text{P} \\ \text{O} \\ \text{R} \\ \text{S} \end{array} \begin{array}{c} \begin{array}{ccccc} \text{A} & \text{P} & \text{O} & \text{R} & \text{S} \end{array} \\ \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}. \tag{7.1}$$

Both number of vertices and edges can differ. As the fixed number of terrains is possible to classify, only fewer vertices can be present. On the other hand, there can be either extra false edges or absent ones. Two key values describing the comparison of the created map with the ground-truth one are introduced: hit and false values.

Hit values represent the number of correct node connections. The value is increased if both the ground-truth and the obtained topological map have

1. exactly one directed edge between two given nodes

   OR

2. no directed edge between two given nodes.

On the contrary, the false value is increased with every extra or missing edge. The natural objective is to create such a map with the highest hit value while minimising the false value. The number of explored vertices is taken as the background and primary criterion.

In order to test Hypothesis H2, experiments are carried out at two different speeds: 30 % and 50 % of the robot maximal speed. Even though the speeds lie together in the bottom half of the possible scale, their mutual difference is significant. The difference can be shown on the experiment (when completed) duration time: ca. 13.1 minutes for 0.3 vs. ca. 8.8 minutes for 0.5. Since a labelling is performed once a gait cycle, the numbers of classification samples differ for each speed.

7.3 Results & Remarks

Six runs of experiments (designated as Run x, where x is the number of the particular run) for each speed have been conducted. Due to the challenging parts throughout the track and the termination condition, not all experiments have been completed. The three of them (complete and the most explaining) for each speed have been chosen to be discussed in this section. For both speeds, the tripod gait (Section 4.3.2) with the following parameters has been utilised:

- xstroke reach 100 mm,
- ystroke reach 100 mm,
- turn angle 0.25,
- lift-up distance 50 mm,
- drop-down distance 50 mm.

For each speed, a proper terrain classifier is set up and trained. To prepare the classifier for conditions concerning the given scenario, the wall-following technique is active while gathering the training data. Since the data from turning motions would significantly affect the resulting terrain models and further performance of the on-line classification, the data-gathering is turned off while turning. The results of the classifier learning are illustrated using the confusion matrices/tables. In such a table, sum of the table entries for given column represents the number of feature vectors (Section 4.1) of a corresponding terrain participating in learning procedure.

7.3.1 Fast Speed

Table 7.1: Confusion Matrix for Fast Speed Trial. For two-fold cross-validation 100% accuracy is obtained.

Terrain	Softblue	Artgrass	Ramp	Pillow	Office
Softblue	77	0	0	0	0
Artgrass	0	75	0	0	0
Ramp	0	0	76	0	0
Pillow	0	0	0	73	0
Office	0	0	0	0	75

Bellow are presented the results of experiments for the fast speed. As mentioned above, when the fast speed for crawling is chosen, the robot moves at the speed of 0.5 of the robot maximum – i.e., for given limits it is 0.05 m/s. The confusion matrix for the fast speed can be seen in Table 7.1.

Wall-Following

The results of the wall-following procedure for Run 2 and Run 3 are shown in Figure 7.4. When compared with the shape of the reference trajectory depicted in Figure 7.1b, the mutual resemblance is significant. Notice that the robot is able to follow the stair-like wall segment without any difficulties. Another fact worth mentioning is successfully traversing the border between pillow (brown) and office (grey), while still keeping constant distance from the wall.

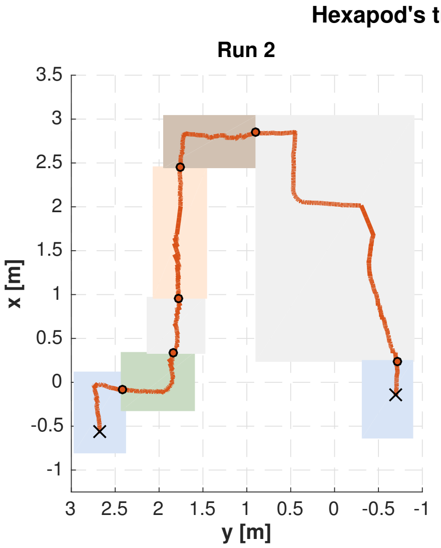


Figure 7.4: Hexapod's Wall-Following Results for Fast Speed Trial. Coloured rectangles represent ground truth of terrains being traversed. Black crosses illustrate the initial and terminal positions.

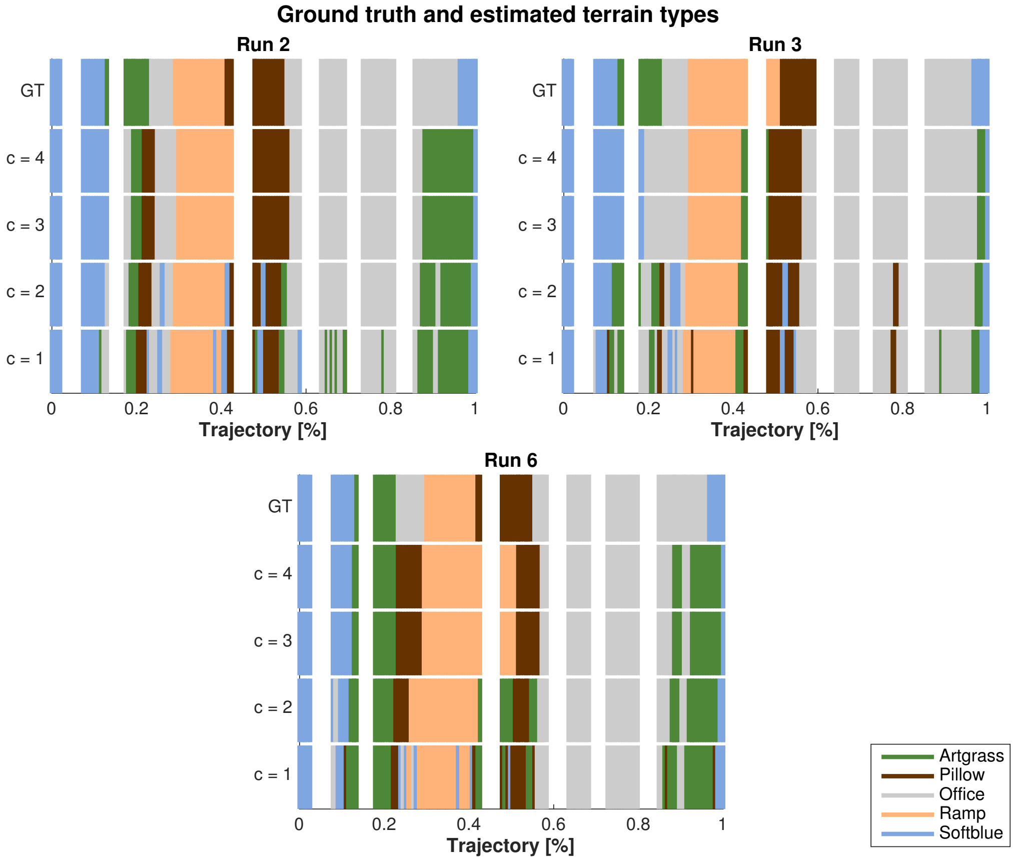


Figure 7.5: Classification Results Compared with Ground-truth for Fast Speed Trial

Terrain Classification & Topological Exploration

In Figure 7.5, a classification diagram for each experiment (Run 2, Run 3, and Run 6) is depicted. The diagrams contain comparison of the ground-truth labelling with the obtained classification. Each row in the diagram represents different size of the classification window. Numerical representation of the graphical comparison follows in Table 7.2. In the right-most column, the total relative error – i.e., the sum of misclassifications divided by the total number of classifications – is presented. As the best size of the classification window proves to be the size of three, albeit the decrease in the relative error is not significant.

Table 7.2: Relative Errors of Classification for Fast Speed Trial

(a) : Run 2: 131 classification samples

c\Terrain	Softblue	Artgrass	Ramp	Pillow	Office	Total
4	0.32	0.64	0.10	0.25	0.34	**0.32**
3	0.27	0.64	0.05	0.19	0.32	**0.29**
2	0.23	0.64	0.00	0.31	0.31	**0.28**
1	0.27	0.64	0.10	0.38	0.41	**0.36**

(b) : Run 3: 129 classification samples

c\Terrain	Softblue	Artgrass	Ramp	Pillow	Office	Total
4	0.32	1.00	0.30	0.36	0.00	**0.26**
3	0.27	1.00	0.30	0.43	0.00	**0.26**
2	0.32	0.50	0.33	0.64	0.16	**0.31**
1	0.41	0.67	0.40	0.79	0.16	**0.37**

(c) : Run 6: 132 classification samples

c\Terrain	Softblue	Artgrass	Ramp	Pillow	Office	Total
4	0.30	0.00	0.00	0.60	0.45	**0.34**
3	0.30	0.00	0.00	0.53	0.45	**0.33**
2	0.39	0.10	0.00	0.53	0.45	**0.36**
1	0.43	0.20	0.16	0.40	0.45	**0.38**

Relative errors for each terrain are given to reveal other possible phenomena. These errors illustrate the so called false negatives, i.e., the terrain is not classified in the way it is supposed to be. It gives the perspective how hard the given terrain is to classify. Even though the errors change from experiment to experiment, it can be concluded that artgrass is the hardest to classify. On the other hand, ramp tends to be the most convenient.

In Figure 7.6, the classification for Run 2 is mapped onto the hexapod's trajectory. All four scenarios (different classification window) are presented. Notice the problematic stair-pattern segment where the office terrain is classified as artgrass.

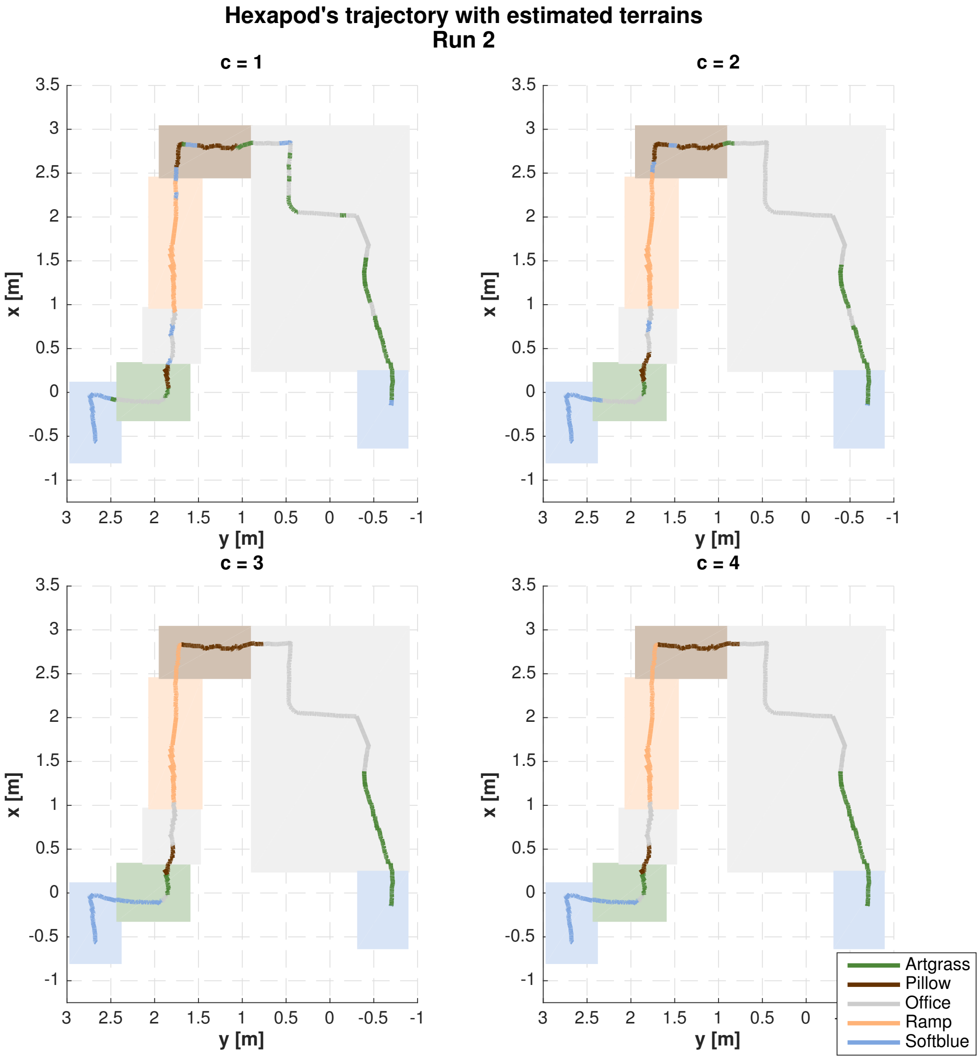


Figure 7.6: Classification Mapped Onto Hexapod's Trajecotry for Fast Speed Trial

Based on the results presented in Figure 7.5, two out of three experiments show this issue. Such misclassifying can be caused by crawling sideways while following the wall. Another trajectory part worthy to mention is the border between artgrass and office. Such a terrain change is likely to be classified as pillow.

Moving a level up to the topological exploration task, the maps created using the classification results are evaluated for each experiment. Such an evaluation reveals whether or not the best classification leads to the best topological map.

All created maps contain five nodes, thus all terrains have been classified. Table 7.3 contains key values (introduced in Section 7.2) describing the comparison of the created topological maps with the ground-truth one for different sizes of the classification window. From the presented results, it can be seen that increasing the

Table 7.3: Topological Map Key Values for Fast Speed Trial. Two consecutive columns represent results for given c – the classification window size.

c	**4**		**3**		**2**		**1**	
	hit	**false**	**hit**	**false**	**hit**	**false**	**hit**	**false**
Run 2	17	9	17	9	14	15	12	32
Run 3	16	9	16	9	15	13	11	26
Run 6	19	7	19	7	14	13	12	32

size of the window leads to rapidly improving both values – for instance considering Run 2 it is 72 % for hit and 42 % for miss. Notice that for all experiment runs the classification window of size three and four give the same results w.r.t. hit and miss values.

Concerning the objective, the best topological map for each experiment follows in Eg 7.2 and Eq 7.3. The window size value is three.

$$M_{R2} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad M_{R3} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \tag{7.2}$$

$$M_{R6} = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{7.3}$$

7.3.2 Slow Speed

Table 7.4: Confusion Matrix for Slow Speed Trial. For two-fold cross-validation 100% accuracy is obtained.

Terrain	Softblue	Artgrass	Ramp	Pillow	Office
Softblue	85	0	0	0	0
Artgrass	0	86	0	0	0
Ramp	0	0	84	0	0
Pillow	0	0	0	88	0
Office	0	0	0	0	89

Results and corresponding commentaries of experiments for the slow speed follow. The hexapod moves at the speed of 0.3 of its maximum forward speed. The SVM model used for the terrain classification emanates from the data presented in Table 7.4. Since some of the results are very similar to the outputs for the fast speed, additional comments and explanations to the given visualisations are provided only where distinct results have been obtained. Because of the presence of problematic area between artgrass and ramp (the red-coloured dashed box on the right in the experiment arena, Figure 7.1b) regarding the terrain classification, the autonomous termination condition is suppressed and the experiment is ended by the human operator. Note that two experiments have not been completed. The ground-truth topological map for incomplete runs lacks the edge going from O to S, i.e., no crossing from office floor to softblue is present. However, taking into account that a portion of one particular terrain (office) is missing, the data still can be analysed.

Wall-Following

The graphical outputs of the wall-following task for Run 5 and Run 6 are presented in Figure 7.7. Run 5 is significantly shorter due to the fact that a mechanical issue promptly appeared during the experiment. Nonetheless, the ability of the hexapod to follow a wall using the designed strategy at slower speed is verified.

Terrain Classification & Topological Exploration

Considering classification diagrams depicted in Figure 7.8 it can be seen that the office terrain is likely to be misclassified for softblue. Especially the part of the arena between artgrass and ramp. Other terrains tend to be better predictable even though the results lack consistency and differ from experiment to experiment, except for artgrass.

The above mentioned remarks are confirmed by numbers presented in Table 7.5. Examining ramp for instance, the numbers in Run 5 and Run 6 show good predictability; however in Run 4, it is complete opposite. A similar phenomenon is present for pillow. On the other hand, values for artgrass are consistent throughout the experiments. Comparing the total relative error of misclassifying for each size of

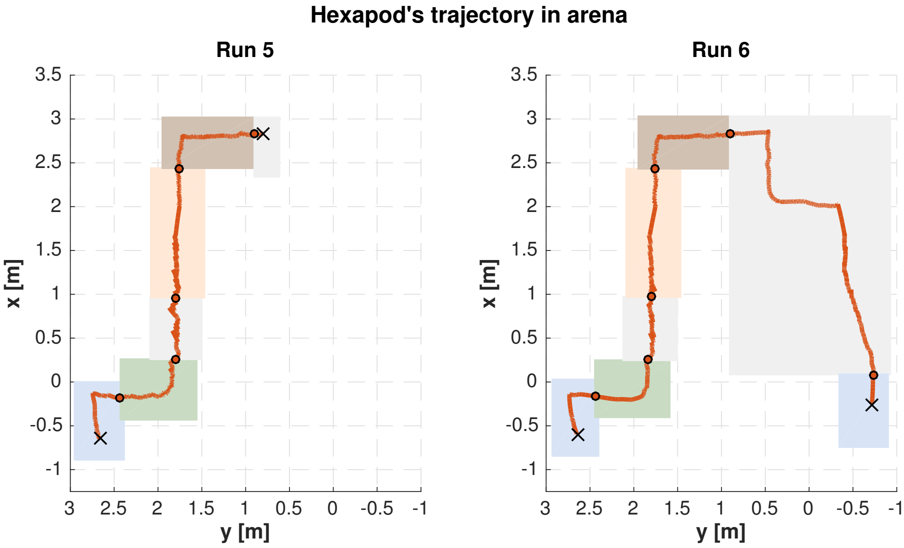


Figure 7.7: Hexapod's Wall-Following Results for Slow Speed Trial.

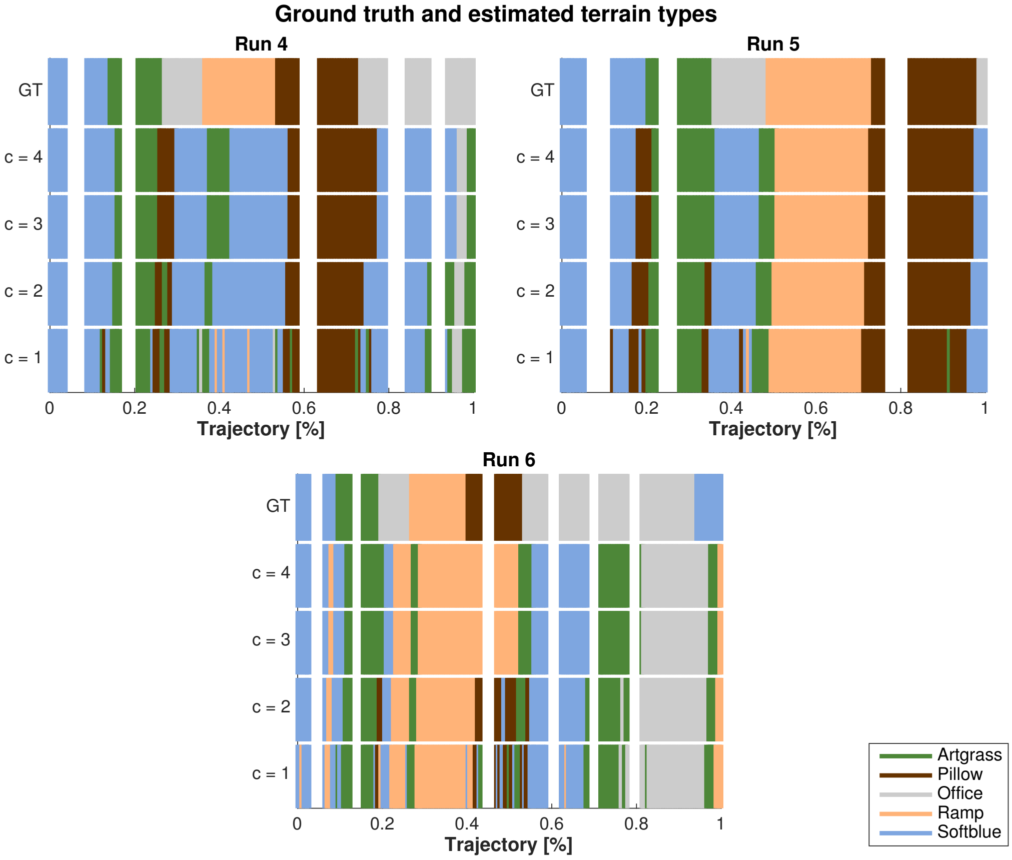


Figure 7.8: Classification Results Compared with Ground-truth for Slow Speed Trial

Table 7.5: Relative Errors of Classification for Slow Speed Trial

(a) : Run 4 Incomplete: 145 classification samples

c\Terrain	Softblue	Artgrass	Ramp	Pillow	Office	Total
4	0.00	0.33	1.00	0.23	0.91	**0.63**
3	0.00	0.33	1.00	0.19	0.91	**0.62**
2	0.00	0.33	1.00	0.15	0.91	**0.61**
1	0.12	0.33	0.90	0.19	0.89	**0.61**

(b) : Run 5 Incomplete: 120 classification samples

c\Terrain	Softblue	Artgrass	Ramp	Pillow	Office	Total
4	0.14	0.19	0.14	0.04	1.00	**0.28**
3	0.19	0.12	0.14	0.08	1.00	**0.29**
2	0.24	0.19	0.14	0.12	1.00	**0.32**
1	0.29	0.19	0.14	0.19	1.00	**0.34**

(c) : Run 6: 210 classification samples

c\Terrain	Softblue	Artgrass	Ramp	Pillow	Office	Total
4	0.62	0.32	0.15	1.00	0.69	**0.60**
3	0.62	0.26	0.15	1.00	0.68	**0.59**
2	0.62	0.26	0.15	0.46	0.65	**0.51**
1	0.65	0.26	0.12	0.62	0.65	**0.53**

the classification window, the size of two is concluded as the best in the overall view.

Figure 7.5 illustrates how the classification results for Run 6 are mapped onto hexapod's trajectory. This particular experiment shows great difficulties with the classification of the part with pillow – as Table 7.5 suggests. Another challenge is the corner part with the office ground-truth where misclassifications for softblue and artgrass appear.

Table 7.6: Topological Map Key Values for Slow Speed Trial

c	4			3			2			1		
	hit	false	v	hit	false	v	hit	false	v	hit	false	v
Run4	14	12	4	14	12	4	15	13	4	11	37	5
Run5	17	8	4	17	8	4	17	9	4	15	19	4
Run6	14	15	4	14	15	4	11	20	5	10	43	5

Unlike Table 7.3 in the previous part, Table 7.6 contains extra type of value representing the number of explored nodes, i.e, terrains. As can be seen from the table, most of scenarios finish with fewer (four) vertices than the ground-truth assumes. Notice that the classification diagrams in Figure 7.5 suggest it is not necessarily

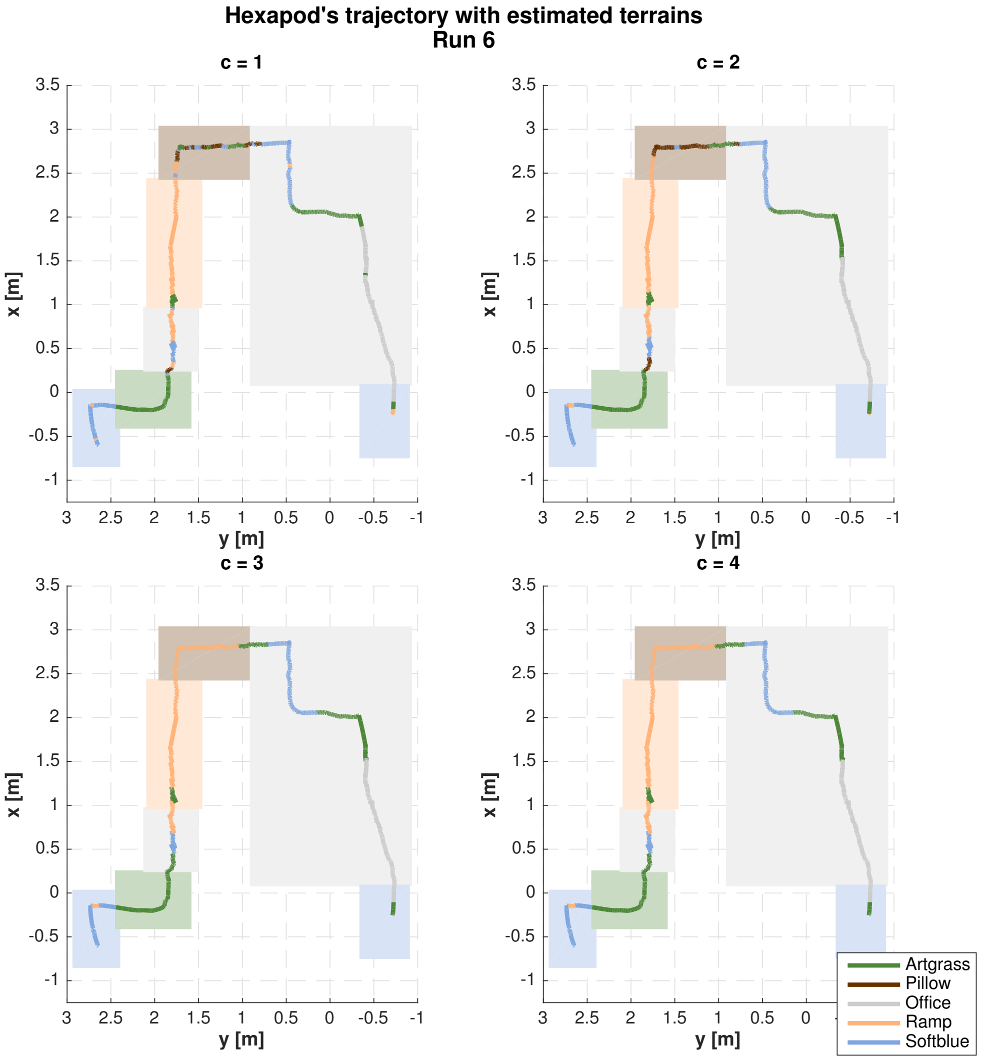


Figure 7.9: Classification Mapped Onto Hexapod's Trajectory for Slow Speed Trial

caused by the length of the experiment. Besides, the hit and false values improve when the window size increases. As it can be seen in Table 7.3, the values are the same whether the size is three or four.

Considering the number of explored vertices as the first criterion when choosing the best window size, the corresponding topological maps in form of matrices are expressed in Eq 7.4 and Eq 7.5. For Run 4, Run 5 and Run 6, the window size values are: 1, 3, and 2. For Run 5, no size leads to explore all five nodes (terrains).

$$M_{R4} = \begin{bmatrix} 0 & 5 & 2 & 0 & 4 \\ 3 & 0 & 0 & 0 & 4 \\ 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 \\ 6 & 2 & 1 & 3 & 0 \end{bmatrix}, \quad M_{R5} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}, \tag{7.4}$$

$$M_{R6} = \begin{bmatrix} 0 & 2 & 2 & 2 & 0 \\ 1 & 0 & 0 & 0 & 3 \\ 2 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 2 & 1 & 0 & 2 & 0 \end{bmatrix}. \tag{7.5}$$

Chapter 8

Discussion

The aim of this work is to develop a solution for topological exploration/place recognition problem based merely on the proprioceptive on-line terrain classification using servo drive feedback only. Such a solution has been supposed to be tested exploiting autonomous behaviour in the form of wall-following technique using solely IR distance/proximity sensors. The designed strategy is constrained to be deployed on an off-the-shelf low-cost hexapod walking robot.

Recapitulation of the achieved results presented in the previous chapter and connecting these results to the thesis goals restated in the last paragraph follow.

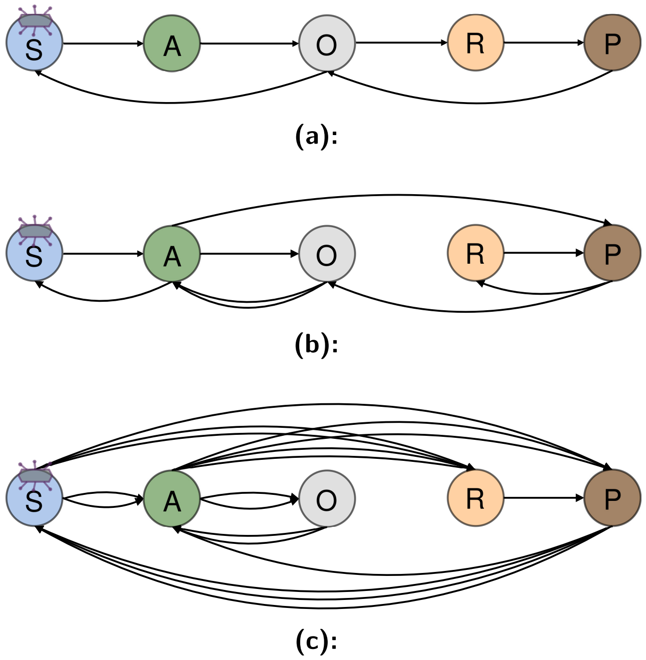


Figure 8.1: Topological Maps. (a): The ground-truth map for complete track. (b): The best map for the fast speed - Run 6. (c): The best map for the slow speed - Run 6.

8.1 Wall-Following

Considering Figure 7.4 and Figure 7.7, the developed wall-following strategy is proved to be satisfactory. The strategy is able to cope with challenges such as rapid height changes and rough terrain traversing. Even though inaccuracies – caused mainly by imprecise turnings – may occur, the procedure allows the main task of topological exploration to be carried out. Hence Hypothesis H1 is approved.

Considering the wall-following technique as an independent subtask on other parts of the designed solution (Figure 5.1), it can be further deployed for other applications where autonomous behaviour is required.

Possible Improvements

The hard-coded turning without versatility represents the greatest deficit within the wall-following strategy. As a solution, the sensory placement should be reconsidered; so the robot would have better perspective about its location. Moreover, mastering the signal processing of the distance sensor measurements in a reliable way would have a significant influence on the robustness of the method.

8.2 Terrain Classification & Topological Exploration

Figure 8.1 introduces the ground-truth and the best topological maps[1] resulting from the performed experimental evaluation described in Chapter 7. When compared with the ground-truth topological map, the obtained maps are distinct w.r.t. the placement of edges and their occurrences.

The topological map structure is directly based on the classification. In order to obtain the same topological map as the ground-truth, the 100% correct classification is required. However, the obtained classification lists are prone to error – ca. 30 % of misclassifications for the best scenarios. Such an incapability is assumed to be caused mainly by an insufficient distinguishability among the given terrains. Hence Hypothesis H3 is considered to be wrong. Another factor worth mentioning is that the experiment trajectory contains frequent turns (the robot has to stop and start again) and the hexapod does not crawl straight for all the time. These phenomena can effect behaviour of the servo drives and thus, the data provided for the classification procedure can differ from the one utilised for learning. Note, the window size ensuring the best classification does not necessarily leads to the best topological map.

To summarise, the topological exploration task using the developed strategy is feasible, and with no doubt worth performing – even though the current classification method is not able to classify flawlessly. When the task is completed, an unknown unstructured environment has been at least partially learned.

[1] They emerge from Eq 7.3 and Eq 7.5.

Possible Improvements

Among enhancements which could improve resulting topological maps is for instance augmented learning of SVM model. Various motion patterns such as going sideways, doing slight turns, stops and immediate starts can be included and emphasised.

Regarding the evaluation of the terrain classification, both false negatives (FN) and false positives (FP)[2] could be taken into account in order to obtain better insight w.r.t. terrain characteristics. Mastering the analysis would allow us to better distinguish among individual terrains. Such a fact could be considered during terrains learning process – e.g., different amount of data for each terrain.

8.3 Speeds Comparison

From the results introduced and commented in the previous parts, it can be concluded that.

- The wall-following task is performed equally whether the speed is slow or fast.
- The considered on-line terrain classification procedure is more suitable for the fast speed, where the relative error of misclassifying is 30 % in average while for the slow speed it is 60 % in average.
- The topological maps created based on the classification are better w.r.t. the given objective (vertices, hit and false edges) for the fast speed.

Such remarks suggest Hypothesis H2 to be wrong. Most likely it is caused by the motion dynamics of the hexapod robot while the faster the robot moves, the more its joints and drives are effected and loaded, resulting in more distinguishable terrain type features. Using an analogy from human world, when one walks slowly either on ice or concrete floor, one feels no difference. However, when a person runs, the traversed terrain is more likely to be distinguished.

8.4 Future Work

Using an off-the-shelf robotic platform, topological exploration and all its components participating in the designed solution (Figure 5.1) offer many possibilities for the future research.

First, an enriching feature would be developing a method for an on-line learning of the terrain classifier. Such a property would lead to the possibility of deploying the hexapod to completely unknown environments. No prior knowledge about terrains would be required. In order to distinguish and categorise unknown terrains, more rigorous and mathematical description regarding a terrain in general should be introduced.

[2] FN = given terrain is supposed to be classified but is not, FP = given terrain is classified but is not supposed to be.

Second, in order to extend operational areas for topological exploration to all erratically unstructured and uncontrived environments – i.e., different obstacles and uneven parts are present in the given setup –; a wall-following approach enabling arbitrary motion should be designed. Such an approach would need to perceive more information about the hexapod's surroundings. Hence a more complex sensing system would need to be deployed. In order to use more powerful controlling principles such as Kalman filter [18], a thorough mathematical model of the robotic platform should be derived.

Note

Since the used localisation system provides the video recording function, all experiments presented in this thesis have been recorded. To provide the reader with the qualitative results, the videos capturing the experiments have been placed on my academic youtube channel[3]. The videos shall present the conducted experiments and illustrate all of the proposed building blocks of this thesis.

[3] https://www.youtube.com/channel/UCZ8KkvHuJrUI_1Ko6o8hIcQ

Chapter 9
Conclusion

Using the given hardware in the form of an off-the-shelf low-cost hexapod walking robot with support of as little as possible sensory equipment, the method for topological exploration has been developed. The unparalleled feature of the proposed solution is the utilisation of the proprioceptive sensing of the terrain for the topological exploration only.

The concept of solution is found on utilising the modularity principle: the essential cornerstones are the wall-following principle and the on-line terrain classification, which has been familiarised and adopted from [33]. The wall-following technique using IR distance sensors and hexapod's motion controller has been successfully developed and deployed.

The overall solution has been experimentally validated in a thorough manner. As the examined environment, an indoor arena — with unstructured and rough features — has been chosen. Two different robot speeds have been examined to see whether a correlation between the used terrain classification method and the robot speed exists. A visual localisation system has been utilised to provide a reliable ground-truth for evaluation of the designed method. The evaluation methodology has been proposed in a way that it can reveal new perspective.

The results bring new insight into the field of on-line terrain classification and mutual terrains' distinguishability. They directly indicate that the topological mapping can be performed using solely the proprioceptive information about the terrain type. Moreover, a direct relationship between the used terrain classification method and the robot's dynamics has been observed. Last but not least, the autonomous behaviour in the form of wall-following has been mastered.

References

[1] R. L. Shell and E. L. Hall, *Handbook of industrial automation.* New York: Dekker, 2000.

[2] E. Yoshida, M. Poirier, J.-P. Laumond, O. Kanoun, F. Lamiraux, R. Alami, and K. Yokoi, "Pivoting based manipulation by a humanoid robot," *Autonomous Robots*, vol. 28, no. 1, pp. 77–88, 2010.

[3] L. D. Baskar, B. De Schutter, J. Hellendoorn, and Z. Papp, "Traffic control and intelligent vehicle highway systems: A survey," *IET Intelligent Transport Systems*, vol. 5, no. 1, pp. 38–52, 2011.

[4] J. A. Boys, E. T. Alicuben, M. J. DeMeester, S. G. Worrell, D. S. Oh, J. A. Hagen, and S. R. DeMeester, "Public perceptions on robotic surgery, hospitals with robots, and surgeons that use them," *Surgical Endoscopy*, vol. 30, no. 4, pp. 1310–1316, 2016.

[5] Collective of Authors, "Field Robotics." `https://www.ri.cmu.edu/research_guide/field_robotics.html`. [Online; webpage; accessed April 25, 2016].

[6] S. Singh and P. Corke, "Editorial for issue number 1, Journal of Field Robotics," *Journal of Field Robotics*, vol. 23, no. 1, pp. 1–2, 2006.

[7] N. Ignell, N. Rasmusson, and J. Matsson, "An overview of legged and wheeled robotic locomotion.," *Available from: Mälardalen University, Web site: http://www. idt. mdh. se/kurser/ct3340/ht12/MINICONFERENCE/FinalPapers/i rcse12_sub mission_21. pdf [Accessed: January 7, 2014]*, 2012.

[8] I. Kostavelis and A. Gasteratos, "Semantic mapping for mobile robotics tasks: A survey," *Robotics and Autonomous Systems*, vol. 66, pp. 86 – 103, 2015.

[9] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation," *Artificial Intelligence*, vol. 99, no. 1, pp. 21 – 71, 1998.

[10] K. Košnar, T. Krajník, and L. Přeučil, "Visual topological mapping," in *European Robotics Symposium 2008*, pp. 333–342, Springer Berlin Heidelberg, 2008.

[11] D. Silver, D. Ferguson, A. Morris, and S. Thayer, "Topological exploration of subterranean environments," *Journal of Field Robotics*, vol. 23, no. 6-7, pp. 395–415, 2006.

[12] A. Nüchter and J. Hertzberg, "Towards semantic maps for mobile robots," *Robotics and Autonomous Systems*, vol. 56, no. 11, pp. 915–926, 2008.

[13] A. Rottmann, Ó. M. Mozos, C. Stachniss, and W. Burgard, "Semantic place classification of indoor environments with mobile robots using boosting," in *AAAI*, vol. 5, pp. 1306–1311, 2005.

[14] S. Ekvall, D. Kragic, and P. Jensfelt, "Object detection and mapping for service robot tasks," *Robotica*, vol. 25, no. 02, pp. 175–187, 2007.

[15] C. F. Juang, Y. H. Chen, and Y. H. Jhan, "Wall-following control of a hexapod robot using a data-driven fuzzy controller learned through differential evolution," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 1, pp. 611–619, 2015.

[16] Z. Y. Yang, C. F. Juang, and Y. H. Jhan, "Hexapod robot wall-following control using a fuzzy controller," in *IEEE International Conference on Control Automation (ICCA)*, pp. 574–578, 2014.

[17] M. Mucientes, D. L. Moreno, A. Bugarín, and S. Barro, "Evolutionary learning of a fuzzy controller for wall-following behavior in mobile robotics," *Soft Computing*, vol. 10, no. 10, pp. 881–889, 2006.

[18] A. Karambakhsh, M. Y. A. Khanian, M. R. Meybodi, and A. Fakharian, "Robot navigation algorithm to wall following using fuzzy kalman filter," in *IEEE International Conference on Control and Automation (ICCA)*, pp. 440–443, 2011.

[19] T. Dash, S. R. Sahu, T. Nayak, and G. Mishra, "Neural network approach to control wall-following robot navigation," in *International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, pp. 1072–1076, 2014.

[20] M. B. Holder, M. M. Trivedi, and S. B. Marapane, "Mobile robot navigation by wall following using a rotating ultrasonic scanner," in *Proceedings of the 13th International Conference on Pattern Recognition*, vol. 3, pp. 298–302 vol.3, 1996.

[21] R. Carelli and E. O. Freire, "Corridor navigation and wall-following stable control for sonar-based mobile robots," *Robotics and Autonomous Systems*, vol. 45, no. 3–4, pp. 235 – 247, 2003.

[22] E. Moore, D. Campbell, F. Grimminger, and M. Buehler, "Reliable stair climbing in the simple hexapod'RHex'," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 3, pp. 2222–2227, IEEE, 2002.

[23] D. Belter, P. Łabęcki, and P. Skrzypczyński, "Adaptive motion planning for autonomous rough terrain traversal with a walking robot," *Journal of Field Robotics*, vol. 33, no. 3, pp. 337–370, 2016.

[24] D. Belter and P. Skrzypczyński, "Rough terrain mapping and classification for foothold selection in a walking robot," *Journal of Field Robotics*, vol. 28, no. 4, pp. 497–528, 2011.

[25] L. Palmer and M. Palankar, "Blind hexapod walking over uneven terrain using only local feedback," in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1603–1608, 2011.

[26] J. Mrva, "Design of motion primitives for a hexapod walking robot operating in a rough environment," Master's thesis, Dept. of Computer Science, FEE, Czech Technical University in Prague, 2014.

[27] J. Mrva and J. Faigl, "Tactile sensing with servo drives feedback only for blind hexapod walking robot," in *International Workshop on Robot Motion and Control (RoMoCo)*, pp. 240–245, 2015.

[28] K. Walas and M. Nowicki, "Terrain classification using laser range finder," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5003–5009, 2014.

[29] A. Angelova, L. Matthies, D. Helmick, and P. Perona, "Fast terrain classification using variable-length representation for autonomous navigation," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007.

[30] X. A. Wu, T. M. Huh, R. Mukherjee, and M. Cutkosky, "Integrated ground reaction force sensing and terrain classification for small legged robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 1125–1132, 2016.

[31] A. Schmidt and K. Walas, "The classification of the terrain by a hexapod robot," in *Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013*, pp. 825–833, Springer International Publishing, 2013.

[32] G. Best, P. Moghadam, N. Kottege, and L. Kleeman, "Terrain classification using a hexapod robot," *Proceedings of the Australasian Conference on Robotics and Automation*, 2013.

[33] J. Mrva and J. Faigl, "Feature extraction for terrain classification with crawling robots," *Proceedings of the 15th conference on Information Technologies–Applications and Theory (ITAT)*, 2015.

[34] C. A. Brooks and K. Iagnemma, "Self-supervised terrain classification for planetary surface exploration rovers," *Journal of Field Robotics*, vol. 29, no. 3, pp. 445–468, 2012.

[35] J. Christie and N. Kottege, "Acoustics based terrain classification for legged robots," 2016.

[36] Trossen Robotics, "PhantomX AX Hexapod Mark II." `http://www.trossenrobotics.com/resize/images/PImages/KIT-RK-PXC-HEX-AX-mk2-c.jpg?bw=1000&bh=1000`. [Online; accessed April 25, 2016].

[37] Trossen Robotics, “Dynamixel AX-12A Robot Actuator.” http://www.trossenrobotics.com/resize/shared/images/PImages/FRS-B-AX-12A-a.jpg?bw=1000&bh=1000. [Online; accessed May 04, 2016].

[38] Robotis, *Dynamixel AX-12A Robot Actuator*. [Online; e-Manual, v1.25.00; accessed May 04, 2016].

[39] M. Stejskal, J. Mrva, and J. Faigl, “Road Following with Blind Crawling Robot,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2016.

[40] HardKernel, “Odroid U3 with Mounted IO Shield.” http://dn.odroid.com/homebackup/U3IOShieldR03AS.jpg. [Online; image; accessed May 04, 2016].

[41] Pololu.com, “Sharp GP2Y0A02YK0F analog distance sensor 20-150 cm.” https://www.pololu.com/picture/view/0J1125. [Online; image; accessed May 04, 2016].

[42] Pololu.com, “Sharp GP2Y0A02YK distance sensor distance-to-voltage graph.” https://www.pololu.com/picture/view/0J1124. [Online; image; accessed May 04, 2016].

[43] Pololu.com, “Sharp GP2Y0A02YK0F Analog Distance Sensor 20-150cm.” https://www.pololu.com/product/1137. [Online; webpage; accessed May 04, 2016].

[44] SHARP Corporation, Electronic Components Group, Opto-electronic Devices Division, *Distance Measuring Sensor GP2Y0A02YK0F*, 2005. [Online; Reference/Specification; accessed May 04, 2016].

[45] Acroname.com, “Sharp Infrared Ranger Comparison.” https://acroname.com/articles/sharp-infrared-ranger-comparison. [Online; webpage; accessed May 04, 2016].

[46] F. Rosenblatt, “The perceptron: a probabilistic model for information storage and organization in the brain.,” *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[47] Carlos Guestrin, “SVMs, Duality and the Kernel Trick.” [Online; lecture; accessed May 25, 2016], 2007.

[48] J. Ziegler and N. B. Nichols, “Optimum settings for automatic controllers,” *Transactions of the ASME*, vol. 64, pp. 759–768, 1942.

[49] J. Sluka, “A PID Controller For Lego Mindstorms Robots.” http://www.inpharmix.com/jps/PID_Controller_For_Lego_Mindstorms_Robots.html. [Online; webpage; accessed May 24, 2016].

[50] T. Krajník, M. Nitsche, J. Faigl, P. Vaněk, M. Saska, L. Přeučil, T. Duckett, and M. Mejail, “A practical multirobot localization system,” *Journal of Intelligent & Robotic Systems*, 2014.

Appendix A
Used Nomenclature and Software

A.1 Abbreviations

DOF	degree of freedom
FC	fuzzy controller
NN	neural networks
IR	infrared
UART	universal asynchronous receiver/transmitter
EEPROM	electrically erasable programmable read-only memory
RAM	random access memory
I2C	inter-integrated circuit
PSD	position sensitive device
IRED	infrared emitting diod
CCD	charge-coupled device
PID	proportional-integral-derivative
SVM	support vector machine
ADC	analog-digital converter
FSM	finite state machine
ISR	interrupt service routine
GND	ground
w.r.t.	with respect to

A.2 Software

MATLAB[1]	computational tool used for the evaluation of the experiment data
draw.io[2]	online diagram software used for all diagrams and graphics
ctuthesis[3]	LaTeX template for theses at Czech Technical University in Prague, developed and administrated by Tomáš Hejda

[1] http://www.mathworks.com/products/matlab/

[2] https://www.draw.io

[3] https://github.com/tohecz/ctuthesis

Appendix B
CD Content

The list with CD content follows:

- evaluation/ - a directory containing essential data logs and MATLAB scripts used to process experimental data;
- hexapod/ - a directory with C++ files comprising an application which runs the robot in the proposed manner, however external libraries are required to successfully compile and run the code – if interested, contact the ComRob Laboratory[1];
- hexplore_bp.pdf - an electronic copy of this thesis;
- specification.pdf - a separate electronic copy of the thesis specification;
- svm/ - a directory with training data (datasets) and resulting classifiers.

[1] http://comrob.fel.cvut.cz