University of Southern Denmark

Master Thesis

Classification of terrain based on proprioception sensing for multi-legged walking robot

Author:
Bc. Martin Bulín

Supervisors:
Dr. Tomas Kulvicius
Dr. Poramate Manoonpong

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Embodied AI & Neurorobotics Lab
Faculty of Engineering

April 28, 2016

Declaration of Authorship

I, Bc. Martin BULÍN, declare that this thesis titled, “Classification of terrain based on proprioception sensing for multi-legged walking robot” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"*Favorite quotation.*"

Quotation Author

UNIVERSITY OF SOUTHERN DENMARK

Abstract

Faculty of Engineering

Embodied AI & Neurorobotics Lab

Master of Science

Classification of terrain based on proprioception sensing for multi-legged walking robot

by Bc. Martin Bulín

The abstract is a concise and accurate summary of the research described in the document. It states the problem, the methods of investigation, and the general conclusions, and should not contain tables, graphs, complex equations, or illustrations. There is a single abstract for the entire work, and it must not exceed 350 words in length.…

Acknowledgements

Students may include a brief statement acknowledging the contribution to their research and studies from various sources, including (but not limited to)

Their research supervisor and committee, Funding agencies, Fellow students, and Family.

The acknowledgments and the people to thank go here, don't forget to include your project advisor…

Contents

List of Figures

List of Tables

List of Algorithms and Code Parts

Chapter 1

Introduction

The thesis must clearly state its theme, hypotheses and/or goals (sometimes called “the research question(s)”), and provide sufficient background information to enable a non-specialist researcher to understand them. It must contain a thorough review of relevant literature, perhaps in a separate chapter.

1-2 pages intro

1.1 Problem Formulation

1 page Motivation and Research Questions

1.2 Motivation for Chosen Methods

motivation for using proprioception sensing motivation for using a neural net as a classifier

1/2 page

1.3 Hypotheses

1/2 page

1.4 Thesis Outline

1/2 page

Chapter 2

State of the Art

chapter intro

2.1 Machine Learning and Classification

Machine Learning and Classification in general, different classifiers (SVM, k-NN, RandomForest, Bayes...)

2-3 pages

2.2 Introduction to Neural Networks

neural networks from the beginning, network types, principles its usage for classification

4-5 pages

2.2.1 Pruning Algorithms

based on the paper Pruning Algorithms - A Survey: a summary of what has been already done, principles 1-2 pages

2.3 Terrain Classification for Legged Robots

based on the literature : a summary of what has been already done in terrain classification, summary of different methods (visual, laser, haptic, proprioception, ...)

5-8 pages

Chapter 3

Master Thesis Objectives

objectives (goals) 1/2 page

Chapter 4

Neural Network Implementation

The account of the research should be presented in a manner suitable for the field. It should be complete, systematic, and sufficiently detailed to enable a reader to understand how the data were gathered and how to apply similar methods in another study. Notation and formatting must be consistent throughout the thesis, including units of measure, abbreviations, and the numbering scheme for tables, figures, footnotes, and citations. One or more chapters may consist of material published (or submitted for publication) elsewhere. See “Including Published Material in a Thesis or Dissertation” for details.

chapter intro

overall kitt_nn framework diagram

1 page

4.1 Structural Elements

kitt_net.py, kitt_neuron.py, kitt_synapse.py

structure diagram

1-2 pages

4.2 Learning Algorithm

Backpropagation implementation in python

algorithm

1-2 pages

4.3 Graphical User Interface

GUI description and its usage

printscreen

1 page

4.4 Pruning Algorithm

This is the novelty of the work, detailed description

algorithm

2 pages

4.4.1 General Validation

Information on the statistics and form of evaluation

XOR Dataset

evaluation on XOR dataset

MNIST Dataset

evaluation on MNIST dataset

further MNIST analysis

figures, tables

4-5 pages

Chapter 5

Terrain Classification for AMOS II

Classification, one of the most widely used areas of machine learning, has a broad array of applications (see chapter 1). To fit a classifier to a problem, one needs to define a problem data structure. Data consists of samples and discrete targets, often called classes. The samples are sooner or later converted into so called feature vectors of a fixed length. The lenght of feature vectors usually determines an input of chosen classifier and number of classes sets an output.

The classification problem in this thesis relates to AMOS II, an open-source multi sensori-motor robotic platform (see fig. 5.2). The task is to classify various terrain types, while the only input comes from proprioceptive sensors. The overall process is based on simulation data and as chapter 4 reveals, feedforward neural networks are involved.

5.1 Overall Process Summary

The very first step is to make the AMOS II simulation run (section 5.2.2). Then a simple tripod gait controller is implemented (section 5.2.3). To generate various terrain types, the number of variable terrain quailities and their ranges are determined (section 5.3.1). Based on these qualities (parameters), a number of virtual terrains is defined (section 5.3.2) and an optimality of these parameters is briefly analysed (section 5.3.3).

Next, AMOS II (its simulation alternative) is forced to walk on every defined terrain type for a sufficiently long period of time and data from all proprioceptors are saved. This data is then verified and failing experiments are removed. The data acquisition step is parameterized by a standard deviation of additive (Guassian) terrain noise and is run for several values.

Having a clean simulation data from all sensors, a feature vector structure is determined. Then a Gaussian signal noise is added. Finally, a dataset is created by splitting all the data into training, validation and testing sets. As it is indicated on fig. 5.1, several datesets and several classifiers are generated during the process.

An optimal neural network classifier is found. The optimal network is then pruned by the algorithm developed in section X. Classification performance of developed tools are compared to a *Scikit-learn* network classification library sknn [].

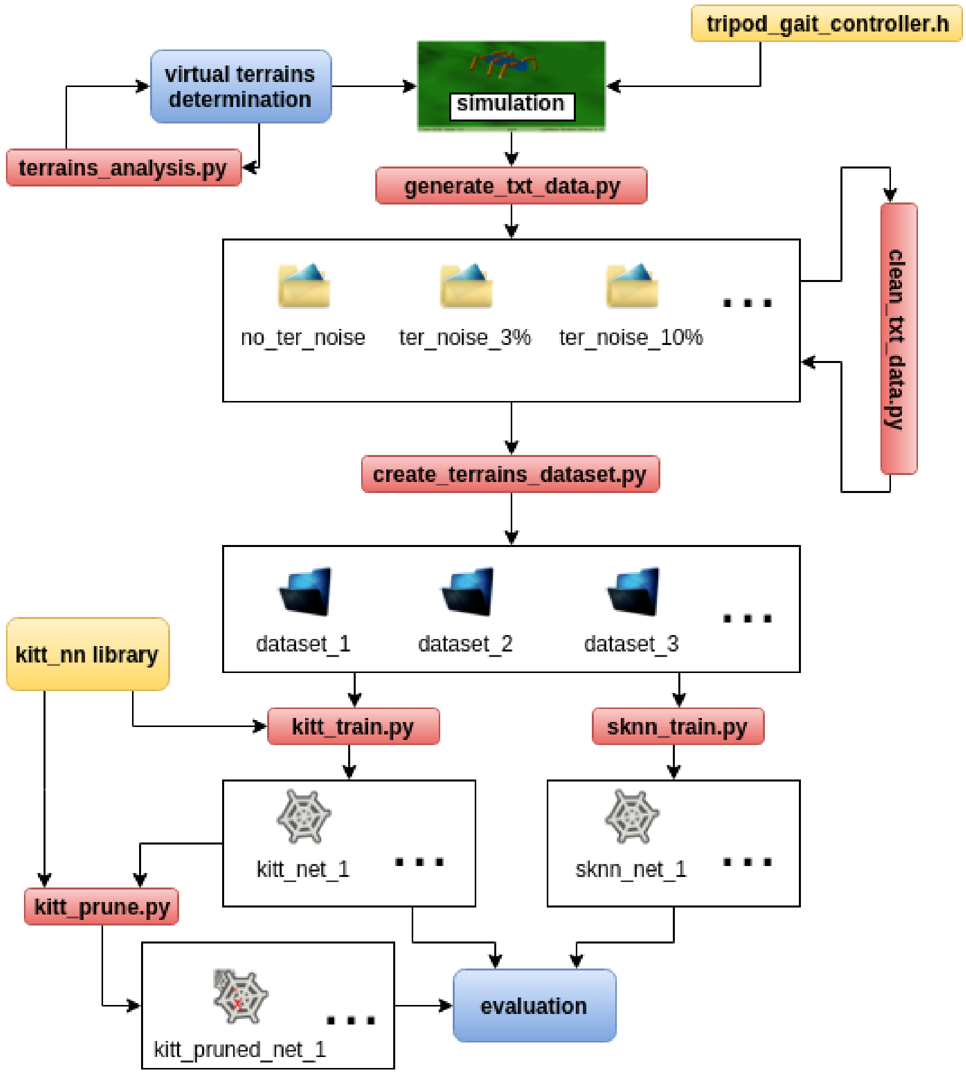

FIGURE 5.1: Terrain classification process - overall diagram.

The dataset packages may differ in these parameters:

- terrain types included (-> number of classes)
- sensors on input
- samples length (number of simulation timesteps)
- terrain noise and signal noise
- number of samples

The trained networks may differ in following parameters:

- neural network structure
- accuracy on training/validation/testing sets

5.2 Experimental Environment Specification

Naturally, the idea of the research is to implement an online terrain classifier on the real machine. Therefore the target robot is described in the following section (5.2.1).

Nevertheless, it is usually a good idea to base the reasearch on simulation data if a satisfactory simulator is available. In this case, *LPZ Robots* (*Research Network for Self-Organization of Robot Behavior*) is used (section 5.2.2).

5.2.1 Hexapod Robot AMOS II

The *AMOS II* abbreviation stands for Advanced Mobility Sensor Driven-Walking Device - version II. It is a biologically inspired hardware platform of size 30x40x20 cm and weight 5.8 Kg (see fig. 5.2). It is mainly used to perform experiments with neural control, memory and learning on a device with many degrees of freedom and to study the coordination of it (*Open-source multi sensori-motor robotic platform AMOS II*).

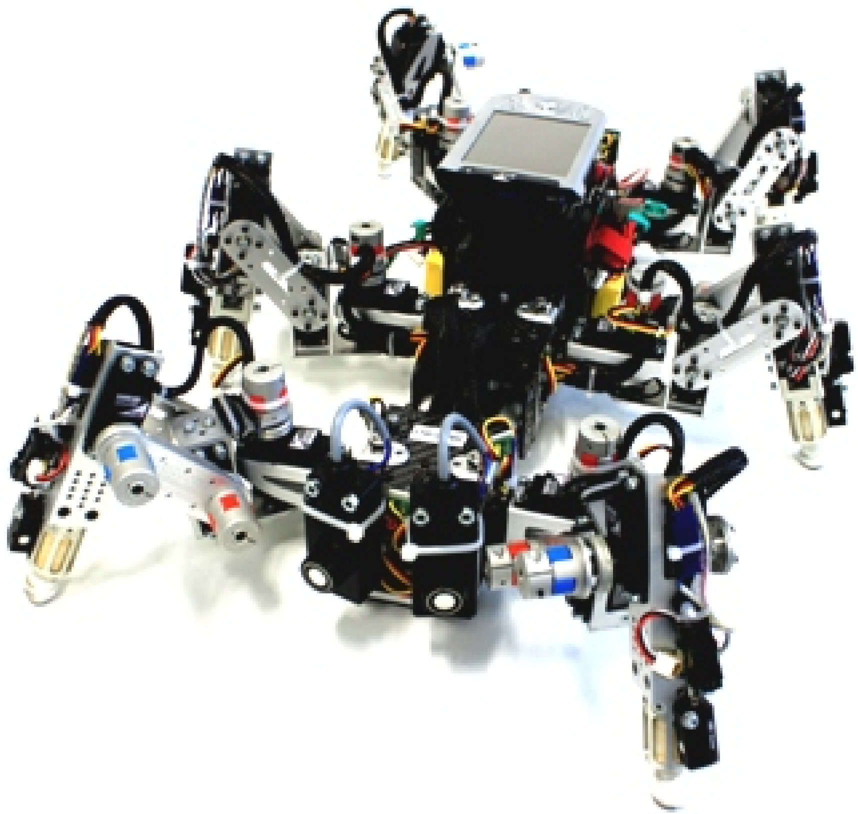

FIGURE 5.2: AMOS II. (*Open-source multi sensori-motor robotic platform AMOS II*)

In general, the robot serves as a hardware platform for neural perception-action systems experiments. The body parts are modeled on the basis of robot's biological inspiration - a cockroach.

A wide range of sensors allows AMOS II to perform several autonomous behaviours. However, only the proprioceptive sensors are important for this research, therefore, we focus on angle sensors and foot contact sensors. All of them are located on robot's legs, so the leg structure is shown on fig. 5.3.

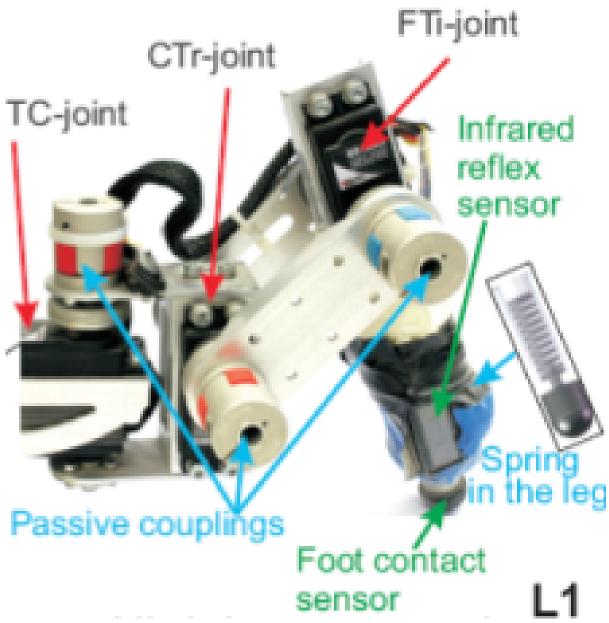

Figure 5.3: AMOS II. (*Open-source multi sensori-motor robotic platform AMOS II*)

As figures 5.2 and 5.3 reveal, the robot has six foot contact sensors in total, one on each leg. Each of them returns a value from range $[0.0, 1.0]$ depending on how strong the foot contact is - it is equal 1.0 if the robot stands on the leg with its full weight and it equals 0.0 when the leg is in the air.

There are three joints on each of robots legs. The thoraco-coxal (TC-) joint is responsible for forward/backward movements. The coxa-trochanteral (CTr-) joint enables elevation and depression of the leg and the last one, femur-tibia (FTi-) joint is used for extension and flexion of the tibia.

These joints are physically actuated by standard servo motors. Having the servos positions, angles of the joints are known and are also considered as propriceptive sensors. As AMOS II has six legs and there are three joints on each leg, there are **18 angle sensors** in total. There is also one backbone joint angle, however, as this one is not implemented in the simulation (see section 5.2.2), it is omitted in this research.

In table 5.1 there are all the propriceptors, their shortcuts and original ranges listed. The ranges are based on the individual servos locations and are explicitly set up to avoid collisions. In section 5.4 a normalization of these ranges is discussed.

Regarding robots actuators, the servo motors can produce variably compliant motions as if each of them were driven by a pair of agonist and antagonist muscles (see *Open-source multi sensori-motor robotic platform AMOS II* for details).

TABLE 5.1: AMOS II - Proprioceptive sensors

shortcut	*sensor description*	*original range*
ATRf	Angle sensor, Thoraco joint, Right front leg	
ATRm	Angle sensor, Thoraco joint, Right middle leg	
ATRh	Angle sensor, Thoraco joint, Right hind leg	
ATLf	Angle sensor, Thoraco joint, Left front leg	
ATLm	Angle sensor, Thoraco joint, Left middle leg	
ATLh	Angle sensor, Thoraco joint, Left hind leg	
ACRf	Angle sensor, Coxa joint, Right front leg	
ACRm	Angle sensor, Coxa joint, Right middle leg	
ACRh	Angle sensor, Coxa joint, Right hind leg	
ACLf	Angle sensor, Coxa joint, Left front leg	
ACLm	Angle sensor, Coxa joint, Left middle leg	
ACLh	Angle sensor, Coxa joint, Left hind leg	
AFRf	Angle sensor, Femur joint, Right front leg	
AFRm	Angle sensor, Femur joint, Right middle leg	
AFRh	Angle sensor, Femur joint, Right hind leg	
AFLf	Angle sensor, Femur joint, Left front leg	
AFLm	Angle sensor, Femur joint, Left middle leg	
AFLh	Angle sensor, Femur joint, Left hind leg	
FRf	Foot contact sensor, Right front leg	[0.0, 1.0]
FRm	Foot contact sensor, Right middle leg	[0.0,1.0]
FRh	Foot contact sensor, Right hind leg	[0.0, 1.0]
FLf	Foot contact sensor, Left front leg	[0.0, 1.0]
FLm	Foot contact sensor, Left middle leg	[0.0, 1.0]
FLh	Foot contact sensor, Left hind leg	[0.0, 1.0]

For purposes of this thesis, it is enough to know that it is possible to generate various gaits using the joints actuators and robots neural locomotion control. The gait controller used for this research is described in section 5.2.3.

5.2.2 LPZ Robots Simulation

The *lpzrobots* project, developed by a research group at the University of Leipzig (*Research Network for Self-Organization of Robot Behavior*) under GPL license, contains many subprojects. For purposes of this thesis, the most important ones are:

selforg : homeokinetic controllers implementation framework

ode_robots : a 3D physically correct robot simulator

The project is implemented in *C++* and needs an Unix system to be run. It consists of two main GIT repositories to be forked - lpzrobots and go_robots. The overall software architecture is shown on fig. 5.4.

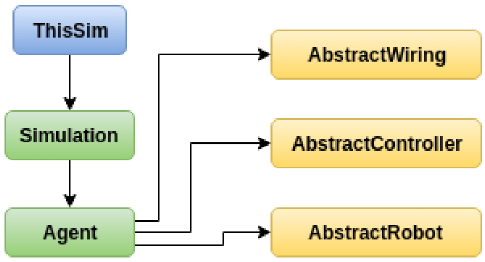

FIGURE 5.4: Software architecture for LPZRobots and GoRobots. (*Research Network for Self-Organization of Robot Behavior*)

To introduce the elements in fig. 5.4, *ThisSim* is an inherited class of another class called *Simulation* and is initialized everytime the simulation is launched. It integrates all elements together, controls the environment as well as the robot and sets up initial parameters.

An instance of the *Agent* class integrates all components of the agent (robot) by using the shown classes.

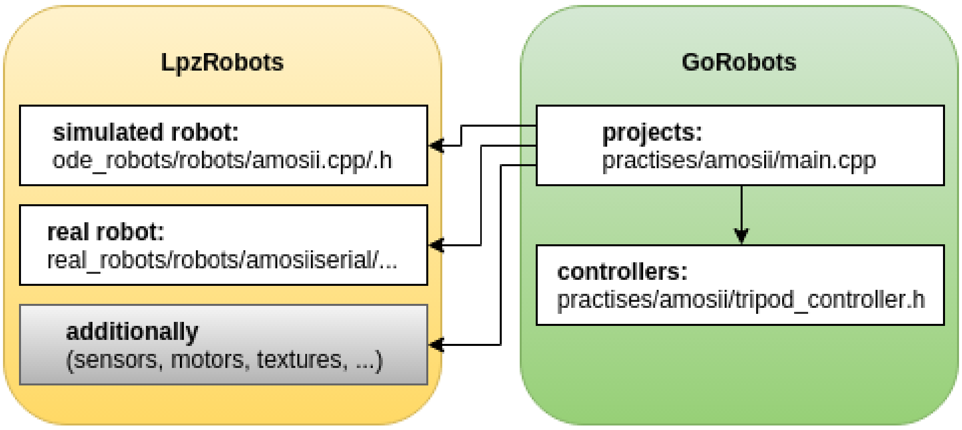

FIGURE 5.5: Structure of the two repositories (LPZRobots and GoRobots). (*Research Network for Self-Organization of Robot Behavior*)

On fig. 5.5 the cooperation of the two repositories is illustrated. With reference to appendix A, one can call the *main.cpp* file from *root/simulation/mbulinai22015-gorobots_edu-fork/practices/amosii* directory as the main simulation file for purposes of the thesis. It sets up the environment with initial parameters $controlinterval = 10$ and $simstepsize = 0.01$, which means the simulation sensitivity is *10 steps* per second.

It also sets the initial camera and robot position in the map. The robot position is chosen randomly and the reason for that is described in section 5.5. The robot fixator, which is originally implemented for AMOS II is removed, so the robot starts walking right after the simulation is launched.

The *main.cpp* file contains all terrain types parameters introduced in section 5.3. The required terrain to be simulated is then passed to this file as an argument. Additionally, the standard deviation value of Gaussian terrain noise (details in section 5.5.1) is set as another argument. Finally, the file is ready to take one more argument, which is a simulation noise represented by a float number. In this research it is fixed to zero however and only the terrain noise combined with a signal noise are used.

The virtual vizualization of AMOS II is illustrated on fig. 5.6.

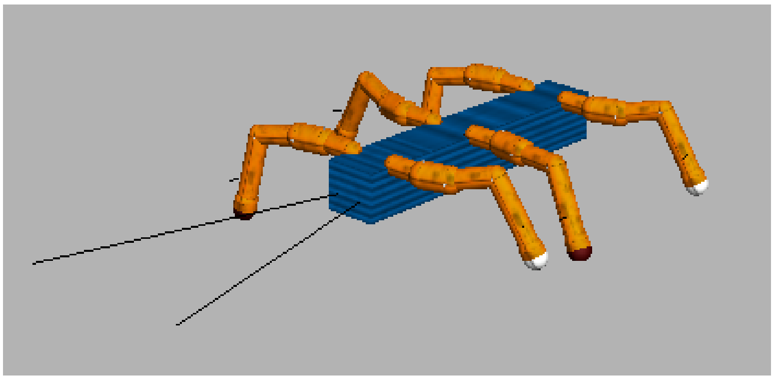

FIGURE 5.6: Simulation alternative for AMOS II.

5.2.3 Tripod Gait Controller

5.3 Virtual Terrain Types

Since the verification is based on the simulation only, the goal is to design an authentical virtual environment. For this purpose various terrain types need to be virtually imitated.

Luckily, the **LpzRobots** AMOS II simulator supports some terrain settings. In the main simulation file (*main.cpp* - see A), a *'rough terrain'* substance is being initialized and passed through a handle to a *TerrainGround* constructor.

PART OF CODE 5.1: Setting a terrain ground in main.cpp

```
Substance roughterrainSubstance(terrain_roughness, terrain_slip,
                        terrain_hardness, terrain_elasticity);
oodeHandle.substance = roughterrainSubstance;
TerrainGround* terrainground = new TerrainGround(oodeHandle,
                        osgHandle.changeColor(terrain_color),
                        "rough1.ppm", "", 20, 25, terrain_height);
```

As part of code 5.1 shows, the terrain substance is defined by four parameters: **roughness**, **slipperiness**, **hardness** and **elasticity**.

Besides the substance handle, the *TerrainGround* constructor takes six more arguments.

terrain_color : simulation ground color

"rough1.ppm" : an image in the .ppm format, a lowest common denominator color image file format (*PPM Format Specification*), a bitmap height file

"" : texture image (not used)

20 : walking area x-size

25 : walking area y-size

terrain_height : maximum terrain height

5.3.1 Terrain Qualities

Out of the listed ground parameters, some of them are picked up and being called *terrain qualities*, as they define a specific terrain type.

It has been decided not to change the *.ppm* image for various terrains and so *rough1.ppm* is fixed. Also the walking area is set to (big enough) final size of *20x25*. The color is variable, however, besides the simulation graphics it does not have any effect on results.

Therefore, a virtual terrain type is defined by five qualitites. Each of them is a float number from an empirically stated range [1]. (table 5.2).

TABLE 5.2: Terrain qualities and their ranges

	min value	max value
roughness	0.0	10.0
slipperiness	0.0	100.0
hardness	0.0	100.0
elasticity	0.0	2.0
height	0.0	0.1

5.3.2 Terrains Parameters Determination

To determine a terrain type, one has to come up with the five parameters from table 5.2.

First, number of identifiable virtual terrain types needs to be determined. For purposes of this thesis, it has been decided to create **14 terrain types**. Their parameters (showed in table 5.3) have been set up intuitively, based on the AMOS II simulated behaviour. With respect to the qualities ranges from table 5.2, the values have been normed to (0, 1).

[1]The upper range limits have been set up based on significant changes in robot behaviour for various parameter values.

TABLE 5.3: Virtual terrain types parameters.

#	*terrain title*	*roughness*	*slipperiness*	*hardness*	*elasticity*	*height*
1	**carpet**	0.3	0.0	0.4	0.15	0.2
2	**concrete**	1.0	0.0	1.0	0.0	0.0
3	**foam**	0.5	0.0	0.0	1.0	0.7
4	**grass**	0.5	0.0	0.3	0.3	0.5
5	**gravel**	0.7	0.001	1.0	0.0	0.3
6	**ice**	0.0	1.0	1.0	0.0	0.0
7	**mud**	0.05	0.05	0.005	0.25	0.2
8	**plastic**	0.1	0.02	0.6	0.5	0.0
9	**rock**	1.0	0.0	1.0	0.0	1.0
10	**rubber**	0.8	0.0	0.8	1.0	0.0
11	**sand**	0.1	0.001	0.3	0.0	0.2
12	**snow**	0.0	0.8	0.2	0.0	0.2
13	**swamp**	0.0	0.05	0.0	0.0	1.0
14	**wood**	0.6	0.0	0.8	0.1	0.2

Colors linked to the terrains in table 5.3 are used in the simulation as well as in the figures in Results section.

5.3.3 Analysis of Chosen Parameters

In general, proper data preparation is an important part of classification tasks, hence a brief analysis is presented.

The goal is to imitate real terrains authentically as possible and at the same time to generate such terrains, that are clearly distinguishable from each other. The more two terrains differ the better classification results are expected.

Having five terrain qualities calls for a 5-D space, which is difficult to illustrate or even imagine. Therefore, formula 5.1 is used to compute a similarity factor of two terrain types (the five qualities are listed in table 5.2 and table 5.3).

$$SF_{t_1,t_2} = \sum_{i=1}^{5} \left| quality(i,t_1) - quality(i,t_2) \right| \tag{5.1}$$

Naturally, equation 5.1 ends up with $SF_{similar} = 0.0$ for two terrains with exactly same parameters and $SF_{different} = 5.0$ for two terrains differing most possibly.

The following figure (5.7) shows the variability (similarity factors) of generated terrains.

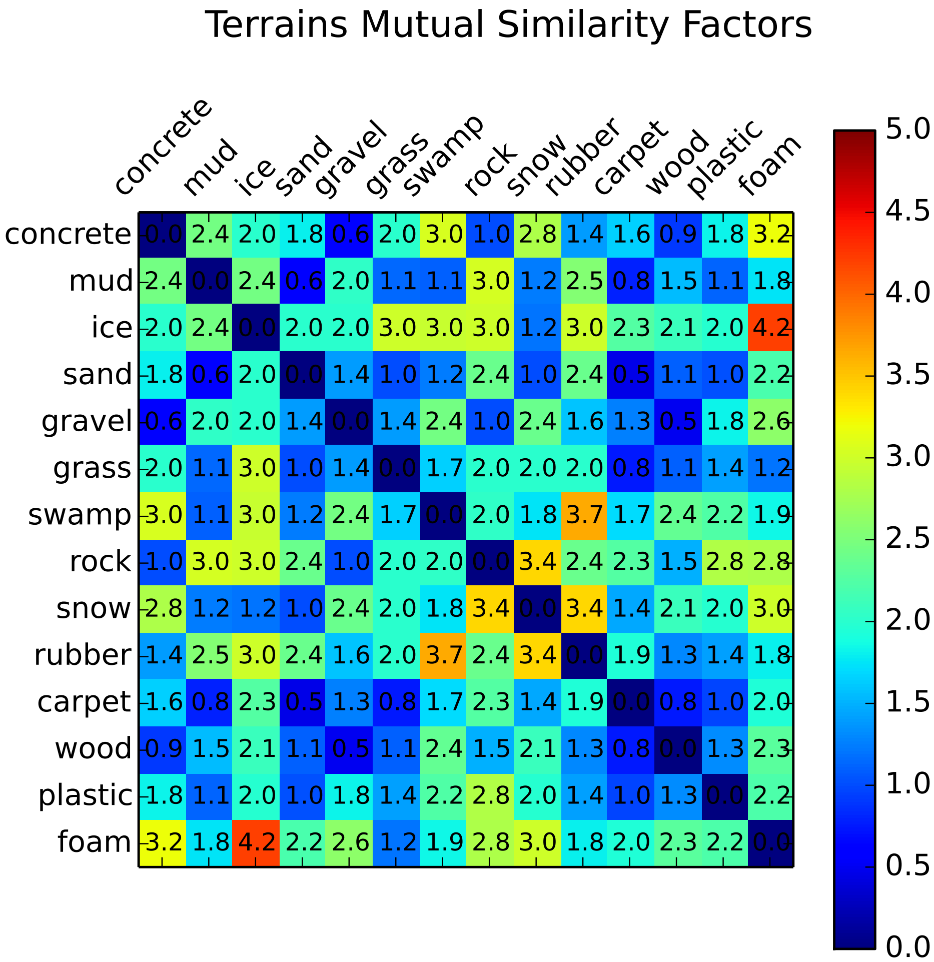

FIGURE 5.7: Variability of generated terrain types.

5.4 Feature Vector Building

Determination of sensors to be used and its transformation into a feature vector

2-3 pages

5.5 Data Acquisition

Description of how the data has been acquired from the simulation and saved as .txt, adding terrain noise

2 pages

5.5.1 Terrain Noise

5.6 Data Processing

Cleaning the data (deleting incomplete ones), adding signal noise, transformation into datasets, splitting into training-validation-testing sets

2-3 pages

5.6.1 Signal Noise

5.7 Training and Classification

Neural net training with several parameters and comparison with training with scikit-neuralnetwork library

2-3 pages

5.7.1 Scikit-neuralnetwork library

brief description of the library and its usage 1/2 pages

1 page

Chapter 6

Experimental Evaluation

The account of the research should be presented in a manner suitable for the field. It should be complete, systematic, and sufficiently detailed to enable a reader to understand how the data were gathered and how to apply similar methods in another study. Notation and formatting must be consistent throughout the thesis, including units of measure, abbreviations, and the numbering scheme for tables, figures, footnotes, and citations. One or more chapters may consist of material published (or submitted for publication) elsewhere. See "Including Published Material in a Thesis or Dissertation" for details.

evaluation (tables and figures) of classification:

- various terrain noise standard deviation values
- various signal noise standard deviation values
- various sensors on network input (only foot, only angle...)
- various timesteps used as one sample (-> time needed for detection)
- various number of detected terrains as outputs
- various network structures
- various training parameters (epochs, learning rate, batch size...)

evaluation of neural nets as a classifier:

- comparison to other classifiers on the same data, classifiers are ready provided by sknn library

evaluation of proprioception sensing against other methods (visual, haptic, laser...):

- comparison to the results from the literature

evaluation of the pruning algorithm:

- various starting structures, ends up with the same minimal-optimal structure?
- various noise types, same minimal structure?
- speed comparisons of the fully-connected vs. pruned structure
- further analysis:
  - which sensors are redundant/crucial

- which sensors are important for which terrain
- comments on the minimal structure and benefits of having it

10-15 pages (many figures, tables)

6.1 Discussion

2-3 pages, maybe together with the figures and tables

Chapter 7

Conclussion

In this section the student must demonstrate his/her mastery of the field and describe the work's overall contribution to the broader discipline in context. A strong conclusion includes the following:

Conclusions regarding the goals or hypotheses presented in the Introduction, Reflective analysis of the research and its conclusions in light of current knowledge in the field, Comments on the significance and contribution of the research reported, Comments on strengths and limitations of the research, Discussion of any potential applications of the research findings, and A description of possible future research directions, drawing on the work reported. A submission's success in addressing the expectations above is appropriately judged by an expert in the relevant discipline. Students should rely on their research supervisors and committee members for guidance. Doctoral students should also take into account the expectations articulated in the University's "Instructions for Preparing the External Examiner's Report".

2-3 pages

7.1 Future Work

All references:

(Zenker et al., 2013) and (Kesper et al., 2012) and (Xiong, Worgotter, and Manoonpong, 2014) and (Mou and Kleiner, 2010) and (Coyle, 2010) and (Hoepflinger et al., 2010) and (Ahmed, 2015) and (Ordonez et al., 2013) and (Bermudez et al., 2012) and (Reed, 1993) and (Spenneberg and Kirchner, 2007) and (Belter, 2011)

Bibliography

[Ree93] R. Reed. “Pruning Algorithms - A Survey”. In: *IEEE Transactions on Neural Networks (Volume:4 , Issue: 5)* (Sept. 1993), pp. 740–747. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=248452.

[SK07] D. Spenneberg and F. Kirchner. *The Bio-Inspired SCORPION Robot: Design, Control and Lessons Learned, Climbing and Walking Robots: towards New Applications*. ISBN 978-3-902613-16-5, 2007.

[Coy10] E. Coyle. “Fundamentals and Methods of Terrain Classification Using Proprioceptive Sensors”. PhD thesis. Florida State University Tallahassee, 2010.

[Hoe+10] M. A. Hoepflinger et al. “Haptic terrain classification for legged robots”. In: *Robotics and Automation (ICRA), IEEE International Conference* 3 (May 2010), pp. 2828–2833. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5509309.

[MK10] W. Mou and A. Kleiner. “Online learning terrain classification for adaptive velocity control”. In: *Safety Security and Rescue Robotics* 26 (July 2010), pp. 1–7. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=5981563.

[Bel11] Dominik Belter. “Gait control of the six-legged robot on a rough terrain using computational intelligence learning and optimization methods”. PhD thesis. Poznan University of Technology, Nov. 2011.

[Ber+12] F. L. G. Bermudez et al. “Performance analysis and terrain classification for a legged robot over rough terrain”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems* 7 (Dec. 2012). URL: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6386243.

[Kes+12] P. Kesper et al. “Obstacle-Gap Detection and Terrain Classification of Walking Robots based on a 2D Laser Range Finder”. In: *Nature-inspired Mobile Robotics* (2012), pp. 419–426. URL: http://manoonpong.com/paper/2013/CLAWAR2013_Kesper.pdf.

[Ord+13] C. Ordonez et al. “Terrain identification for RHex-type robots”. In: *Unmanned Systems Technology XV* 17 (May 2013). URL: http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=1689675.

[Zen+13] S. Zenker et al. “Visual terrain classification for selecting energy efficient gaits of a hexapod robot”. In: *International Conference on Advanced Intelligent Mechatronics* 12 (July 2013), pp. 577–584. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6584154&tag=1.

[XWM14] X. Xiong, F. Worgotter, and P. Manoonpong. "Neuromechanical control for hexapedal robot walking on challenging surfaces and surface classification". In: *Robotics and Autonomous Systems* 7 (Aug. 2014), pp. 1777–1790. URL: www.elsevier.com/locate/robot.

[Ahm15] Mohammed Nour Abdel Gwad Ahmed. "An Intelligent Architecture for Legged Robot Terrain Classification Using Proprioceptive and Exteroceptive Data". PhD thesis. University of Bremen, June 2015.

[Misa] *Open-source multi sensori-motor robotic platform AMOS II*. http://manoonpong.com/AMOSII.html.

[Misb] *PPM Format Specification*. http://netpbm.sourceforge.net/doc/ppm.html. Updated: 02 November 2013.

[Misc] *Research Network for Self-Organization of Robot Behavior*. http://robot.informatik.uni-leipzig.de/software/. last modified: 06. July 2015.

Appendix A

Code Documentation

Write your Appendix content here.

Appendices must be limited to supporting material genuinely subsidiary to the main argument of the work. They must only include material that is referred to in the document.

Material suitable for inclusion in appendices includes the following:

Additional details of methodology and/or data Diagrams of specialized equipment developed Copies of questionnaires or surveys used in the research Do not include copies of the Ethics Certificates in the Appendices.