Motion Control of a Hexapod Robot Over Uneven Terrain Using Signed Distance Fields

by

Andries Phillipus Lotriet

Thesis presented in partial fulfilment of the requirements for the degree of Master of Engineering (Electronic) in the Faculty of Engineering at Stellenbosch University

Supervisor: Prof. J.A.A. Engelbrecht

March 2023

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: 2023/02/10

Abstract

Motion Control of a Hexapod Robot Over Uneven Terrain Using Signed Distance Fields

A.P. Lotriet

Department of Electrical and Electronic Engineering,
Stellenbosch University,
Private Bag X1, Matieland 7602, South Africa.

Thesis: MEng (EE)

March 2023

In recent times great strides have been made in the field of autonomous robotics, especially with regards to autonomous navigation of wheeled and arial drones. Legged robotics however still face numerous problems before they can become practical to use, the most egregious of these problems being balancing of the robot and optimal foot placement.

This thesis focuses on providing a solution to the latter problem of foot placement. This is achieved by using an depth camera to, in real time, construct a localised map of the environment and subsequently analysing said map for optimal foot placement locations. The system is then tested using a hexapod robot both in simulation and on a physical robot.

Acknowledgments

Dedication

Table of contents

List of figures vii

List of tables viii

List of symbols ix

1 Introduction 1
1.1 Background 1
1.2 Research Goal 1
1.3 Methodology 1
1.4 Scope and Limitations 2
1.5 Thesis Outline 2

2 Literature review 3
2.1 End Effector Placement Method 3
2.2 Localisation and Mapping 3
2.3 Simulation Environment 4

3 Content chapter 5
3.1 Heading level 2 5
3.1.1 Heading level 3 5

4 Modelling 7
4.1 Hexapod Construction 7
4.2 Servo Modelling 7
4.3 Simulation Terrain 7

5 Mapping 8
5.1 Localisation and sparse map 8
5.2 Dense map 8

6 Motion 9
6.1 Gait State Machine 9
6.2 Inverse Kinematics (IK) 9

6.3 Foot Motion 9
6.4 Foot Placement 9

7 End Effector Placement **10**
7.1 Scoring 10
7.1.1 Terrain Proximity 10
7.1.2 Slope 10
7.1.3 Height Delta 10
7.2 Placement Optimisation 10
7.2.1 Cost Function 10
7.2.2 Optimisation Function 10

8 Hardware Implementation **11**

9 Testing **12**

10 Conclusions **13**

A Mathematical proofs **14**
A.1 Euler's equation 14
A.2 Navier Stokes equation 14

B Experimental results **15**

List of references **16**

List of figures

3.1 Water plants 6

List of tables

3.1 Standard ISO paper sizes 6

List of symbols

Constants

$L_0 =$ 300 mm

Variables

Re_{D}	Reynolds number (diameter)	[ ]
x	Coordinate	[m]
$\ddot{x}$	Acceleration	[m/s^2]
θ	Rotation angle	[rad]
τ	Moment	[N·m]

Vectors and Tensors

$\vec{\boldsymbol{v}}$ Physical vector, see equation ...

Subscripts

a Adiabatic

a Coordinate

Abreviations

IK	Inverse Kinematics	v
MuJoCo	Multi-Joint dynamics with Contact	1
GUI	Graphical User Interface	4
ROS	Robot Operating System	4
LiDAR	Light Detection and Ranging	3
RGB-D	Red Green Blue Depth	1
SLAM	Simultaneous Localisation and Mapping	3

Chapter 1

Introduction

1.1 Background

Starting from the big picture, gradually narrow focus down to this project and where this report fits in. Why this specific project/report is worthwhile.

1.2 Research Goal

Broken down into sub objectives

The objectives of the project (in some cases the objectives of the report). If necessary describe limitations to the scope.

1.3 Methodology

When deciding how to determine optimal end effector placement various sensing methods were considered, such as using a Red Green Blue Depth (RGB-D) camera to view the environment, placing force sensors on the robots feet or measuring servo torque to determine when the feet were in contact with a surface. A previous paper by ...... used a RGB-D camera by storing past snapshots to adjust the feet to the optimal height, it was decided that the primary sensing method for this thesis would also be a RGB-D camera but instead of storing snapshots, a height map would be generated of the local environment. This would allow for more advanced methods of placement selection and preliminary collision checking for leg movements.

The first step in realising this system was to construct a accurate simulation of the hexapod. The primary simulation packages that were considered are Gazebo, PyBullet and Multi-Joint dynamics with Contact (MuJoCo). Gazebo was a appealing choice due to the easy integration with ROS, however it was decided to use MuJoCo since it was found to have a far superior contact physics simulation.

Once the hexapod was adequately modelled in MuJoCo a tripod gait state machine, IK system and control interface was implement, at this stage the hexapod was capable of walking on flat terrain.

Next the the system to generate the height map was implemented, this entailed sampling the RGB-D camera and comparing cells in the height map against the depth buffer. Once the height map was implemented it was possible to build the system responsible for end effector placement, this is covered in detail in chapter 7, after which collision checking for the generated end effector motion was implement, ensuring that the hexapod does not get stuck on pieces of terrain.

With this the system was realise in simulation, next the system was implemented and tested on the physical robot, discussed in detail in chapter 8

1.4 Scope and Limitations

1.5 Thesis Outline

Chapter 2

Literature review

This chapter will discuss previous work done regarding various elements of the project, this includes the overarching method of placing end effectors on uneven terrain, simulation environments used, localisation in 3D space, and so forth.

2.1 End Effector Placement Method

Among other research focused on hexapods, many focus on topics such as obstacle avoidance, climbing surfaces, confined surfaces and cargo transportation. When focusing of terrain adaptation most often the use of sensors such as Light Detection and Ranging (LiDAR), torque, or touch are employed. Where usually the height of end effectors are adjusted to the height of the terrain Coelho *et al.* (2021).

Some papers, such as Homberger *et al.* (2017) utilise stereoscopic vision, in addition to end effector height adjustment, also focus on surface material classifications based on which the virtual stiffness of the impedance controller is adjusted.

The focus of this paper will be on end effector height and planar position adaptation through real time walkability classification of the environment. While only utilising an RGB-D camera as sensor

2.2 Localisation and Mapping

This project requires a system that will localise the robot within its environment, as the primary sensor used is an RGB-D camera various visual Simultaneous Localisation and Mapping (SLAM) systems were considered. ORB-SLAM 3, a optimisation-based, sparse map SLAM system was chosen to be used. ORB-SLAM 3 maintains a sparse map, an atlas, of both active and dormant features. This atlas is used to localise in the sparse map (Macario Barros *et al.*, 2022).

The implementation of a dense map to be used for end effector placement is discussed in chapter 5.

2.3 Simulation Environment

The most popular physics simulators for robotics in recent times are Gazebo, MuJoCo and CoppeliaSim (previously V-REP) (Collins *et al.*, 2021). Gazebo and CoppeliaSim both have easy to use Graphical User Interface (GUI) interfaces and easy integration with Robot Operating System (ROS). MuJoCo on the other hand does not have a full GUI interface, only a simulation viewer, and does not have native ROS integration. Having said this MuJoCo was found to be the most accurate and fastest simulator when considering the use case of robotics (Erez *et al.*, 2015).

Considering that the only relevant downside to MuJoCo is the lack of native ROS integration and the lack of a comprehensive GUI, which seeing as MuJoCo has good python bindings, could be seen as a advantage, MuJoCo was chosen as the simulator.

Chapter 3

Content chapter

Unless the chapter heading already makes it clear, an introductory paragraph that explains how this chapter contributes to the objectives of the report/project.

3.1 Heading level 2

3.1.1 Heading level 3

3.1.1.1 Deepest heading, only if you cannot do without it

Equations: An equation must read like part of the text. The solution of the quadratic equation $ax^2 + bx + c = 0$ given by the following expression (note the full stop after the equation to indicate the end of the sentence):

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2b}. \tag{3.1}$$

In other cases the equation is in the middle of the sentence. Then the paragraph following the equation should start with a small letter. Euler's identity is

$$e^{i\pi} + 1 = 0, \tag{3.2}$$

where e is Euler's number, the base of natural logarithms.

The `amsmath` has a wealth of structure and information on formatting of mathematical equations.

Symbols and numbers: Symbols that represent values of properties should be printed in italics, but SI units and names of functions (e.g. sin, cos and tan) must not be printed in italics. There must be a small hard space between a number and its unit, e.g. 120 km. Use the `siunitx` package to typeset numbers, angles and quantities with units:

`\num{1.23e3}`	$\rightarrow$	1.23×10^3
`\ang{30}`	$\rightarrow$	30°
`\qty{20}{N.m}`	$\rightarrow$	20 N·m

Figures and tables: The `graphicx` package can import `PDF`, `PNG` and `JPG` graphic files.

Table 3.1: Standard ISO paper sizes

Paper	Sizes	
	W [mm]	H [mm]
A0	841	1189
A1	594	841
A2	420	594
A3	297	420
A4	210	297
A5	148	210

Figure 3.1: Water plants

Chapter 4

Modelling

This chapter covers the simulation environment and modelling of the hexapod in MuJoCo.

4.1 Hexapod Construction

4.2 Servo Modelling

4.3 Simulation Terrain

Chapter 5

Mapping

In this chapter the localisation and mapping systems will be covered.

5.1 Localisation and sparse map

5.2 Dense map

Chapter 6

Motion

This chapter describes the systems governing the motion of the robot, such as leg motion planning and gait generation.

6.1 Gait State Machine

6.2 Inverse Kinematics (IK)

6.3 Foot Motion

6.4 Foot Placement

Chapter 7

End Effector Placement

A terrain scoring and optimisation function that executes on the height map is used to optimise the anchor points of the three supporting end effectors. This chapter covers the optimisation function and parameters used for this.

7.1 Scoring

7.1.1 Terrain Proximity

7.1.2 Slope

7.1.3 Height Delta

7.2 Placement Optimisation

7.2.1 Cost Function

7.2.2 Optimisation Function

Chapter 8

Hardware Implementation

This chapter describes the process of implementing the system built in previous chapters on the physical robot.

Chapter 9

Testing

This chapter covers all tests performed to validate performance of the system.

Chapter 10

Conclusions

Appendix A

Mathematical proofs

A.1 Euler's equation

Euler's equation gives the relationship between the trigonometric functions and the complex exponential function.

$$e^{i\theta} = \cos\theta + i\sin\theta \tag{A.1}$$

Inserting $\theta = \pi$ in (A.1) results in Euler's identity

$$e^{i\pi} + 1 = 0 \tag{A.2}$$

A.2 Navier Stokes equation

The Navier–Stokes equations mathematically express momentum balance and conservation of mass for Newtonian fluids. Navier-Stokes equations using tensor notation:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}\left[\rho u_j\right] = 0 \tag{A.3a}$$

$$\frac{\partial}{\partial t}\left(\rho u_i\right) + \frac{\partial}{\partial x_j}\left[\rho u_i u_j + p\delta_{ij} - \tau_{ji}\right] = 0, \quad i = 1, 2, 3 \tag{A.3b}$$

$$\frac{\partial}{\partial t}\left(\rho e_0\right) + \frac{\partial}{\partial x_j}\left[\rho u_j e_0 + u_j p + q_j - u_i \tau_{ij}\right] = 0 \tag{A.3c}$$

Appendix B

Experimental results

List of references

Coelho, J., Ribeiro, F., Dias, B., Lopes, G. and Flores, P. (2021). Trends in the control of hexapod robots: a survey. *Robotics*, vol. 10, no. 3, p. 100.

Collins, J., Chand, S., Vanderkop, A. and Howard, D. (2021). A review of physics simulators for robotic applications. *IEEE Access*, vol. 9, pp. 51416–51431.

Erez, T., Tassa, Y. and Todorov, E. (2015). Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4397–4404.

Homberger, T., Bjelonic, M., Kottege, N. and Borges, P.V. (2017). Terrain-dependant control of hexapod robots using vision. In: *2016 International Symposium on Experimental Robotics*, pp. 92–102. Springer.

Macario Barros, A., Michel, M., Moline, Y., Corre, G. and Carrel, F. (2022). A comprehensive survey of visual slam algorithms. *Robotics*, vol. 11, no. 1, p. 24.