Motion Planning for an Amphibious Snake-like Robot

Matthew Prior

Aberystwyth University

Introduction

This project focuses on motion planning for an amphibious biologically inspired snake-like robot. The aim of this project is to create a motion planning algorithm for [1] to achieve goal navigation or environmental mapping in amphibious environments.

There have been many advancements in this particular field, from developing newer more sophisticated biologically inspired robots [1] to exploring different motion-planning methods [2], I intend to utilize and build upon these existing developments in the field to achieve the projects goal.

Currently, the state of the project consists of a snake-like robot with a testing scene in Coppelia where I've implemented a bug-like algorithm to navigate towards a goal, this goal-driven state also has 2 undulating locomotion methods, one being the classic snake-like wave and the other more eel-focused with the head remaining in front.

As well as this, I have also written extensive documentation [3] during an exploration phase describing how the example control script works, allowing you to understand how the robot senses, actuates in order to achieve it's goal.

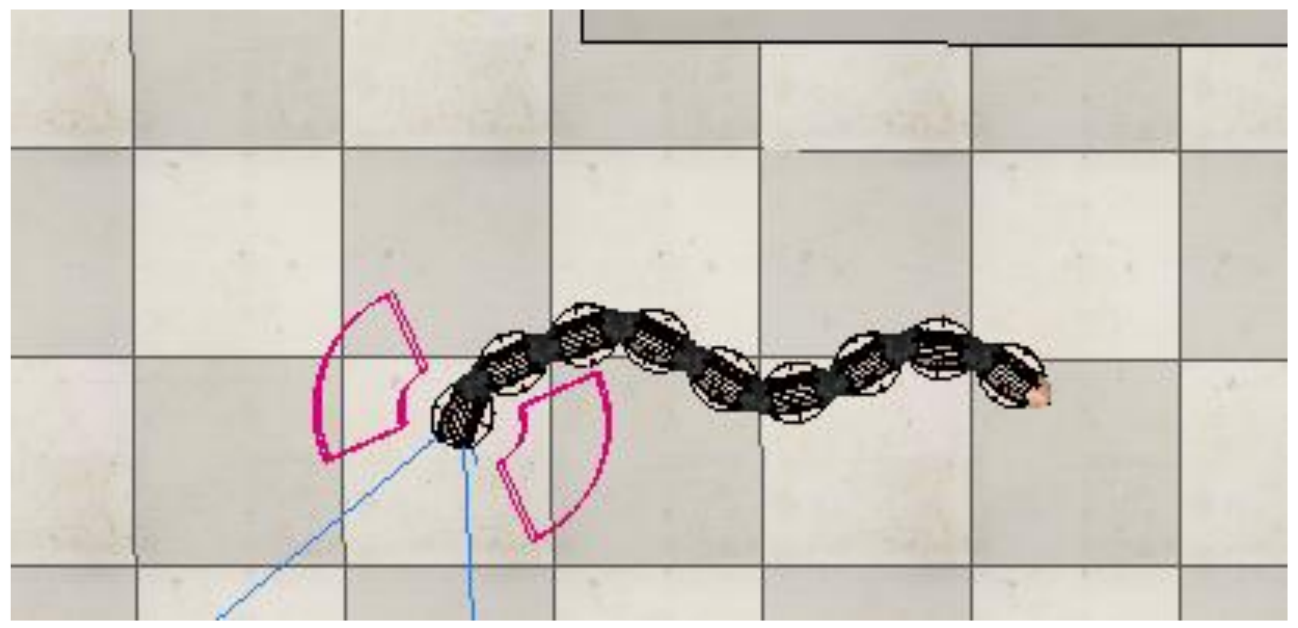

Figure 1: Oscillating snake-like

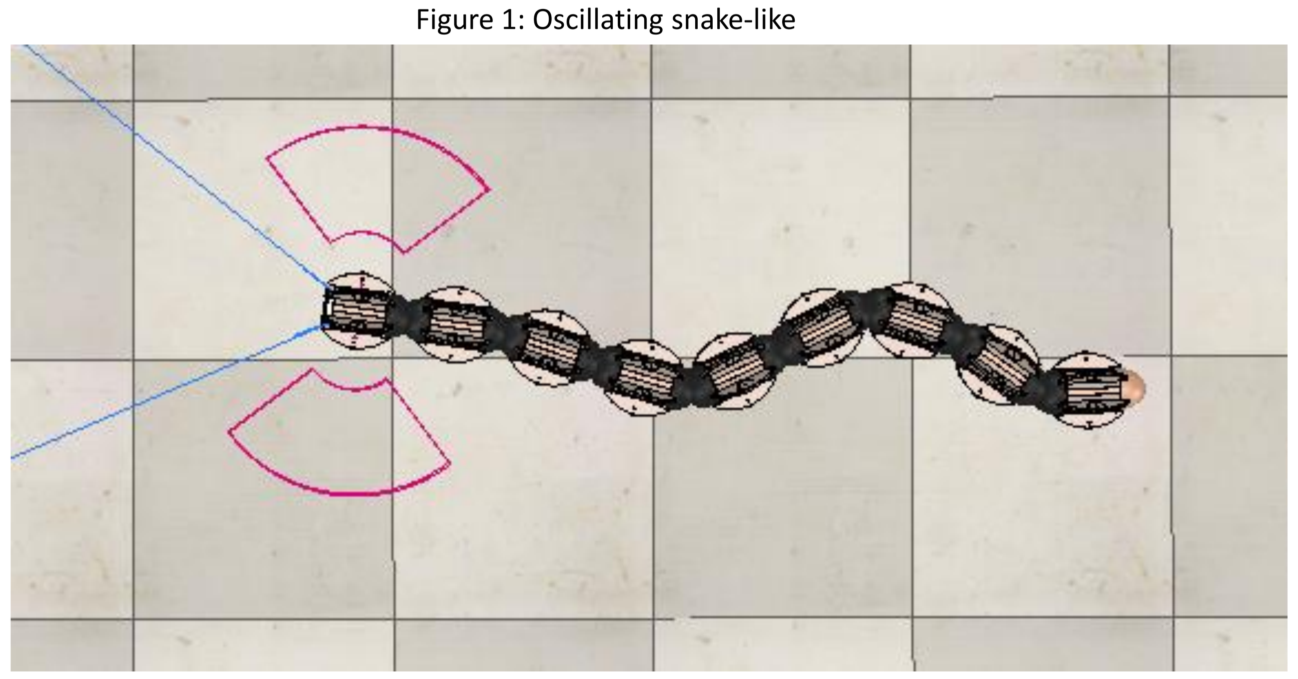

Figure 2: Eel-like pattern

Technical Information

This project is built using a simulator called CoppleiaSim with project files available in my GitHub repository [3], with the control scripts currently written in Lua.

How locomotion is achieved

Locomotion for the robot was quite a challenge as there were many different factors to take into account, however in the end I decided upon an undulatory locomotion style where the robot would undulate back and forth creating a wave. This was achieved through having phases of motion, where one would oscillate left until a certain threshold and the other

Eel-like:

An Eel-like locomotion was achieved through staggering the motion across each of the links, currently the robot uses a respondable setup in which each link will mimic the head on a delay; I reduced the impact of oscillation the closer the link was to head creating an eel-like effect where the head would stay in front during travel.

Goal navigation

Goal navigation is implemented in a way that we take the distance between the robot and it's goal (A pre-defined object in the simulator) which is calculated by the square root of the difference in X and Y between the two; with the distance we can periodically store snapshots of how far the robot is and use that to determine if we need to move left or right in order to navigate towards the goal.

Issues

Initially the main problem was trying to figure out what the robot already did, since the robot came pre-packed into the software it already had some basic control scripts which I had to figure out before I could control the robot. I also learnt around this time that all movements were randomised so I added some additional sensing widgets to the robot to make controlling it easier.

Goal navigation posed the biggest problem, having initially just implemented a coordinate based system I had problems with it ruining the snakes locomotion style, so I added an additional distance-based check.

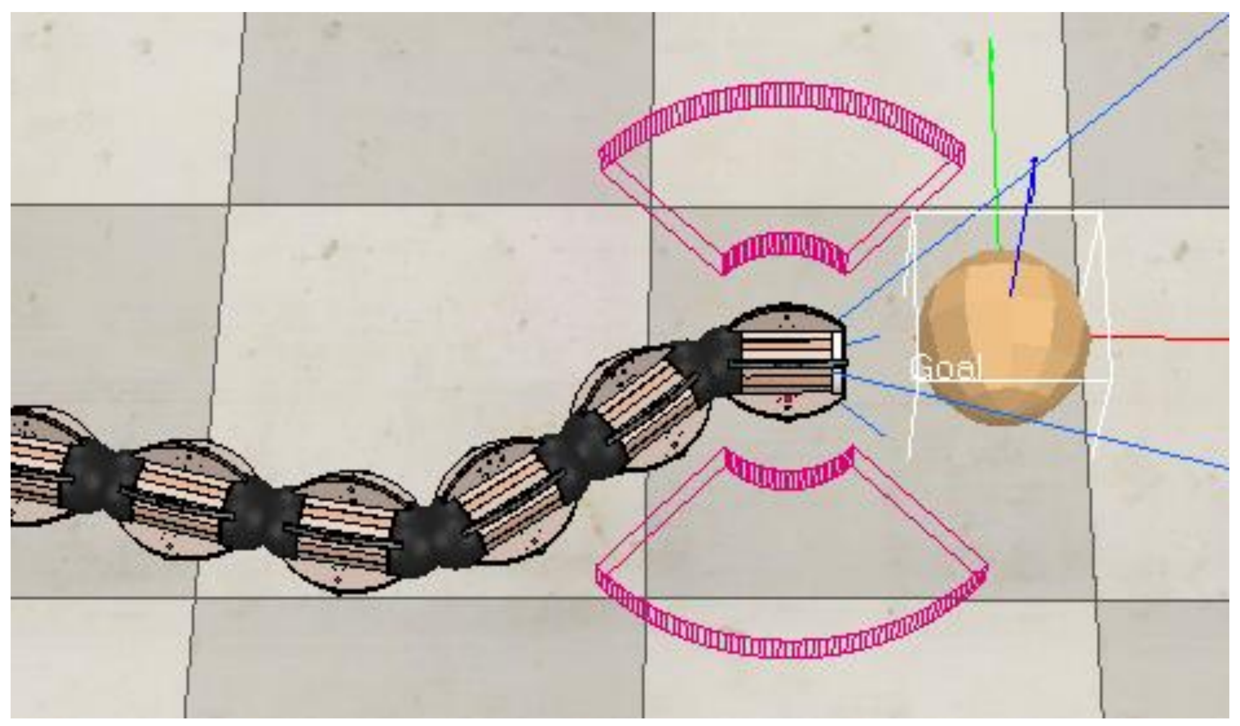

Figure 3: Snake navigating towards goal

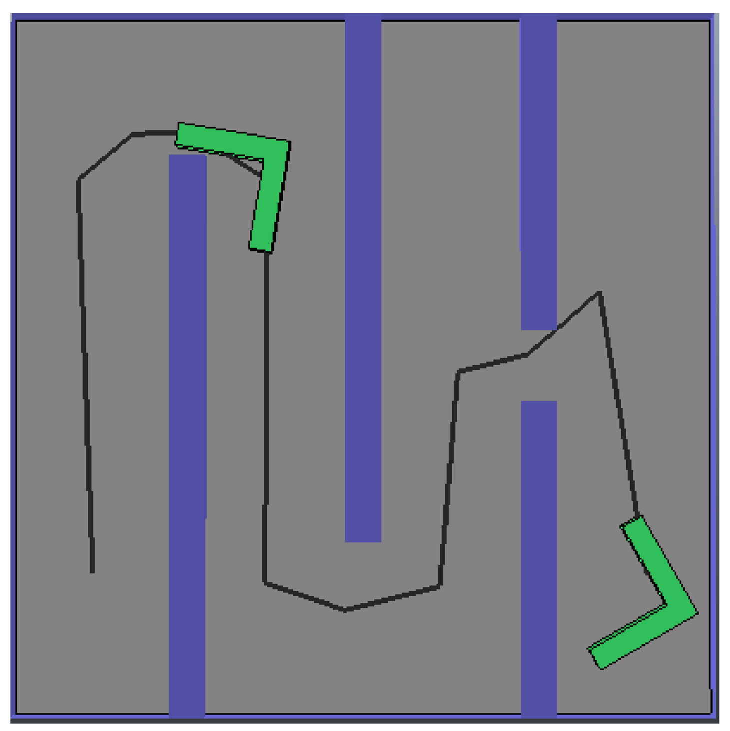

Figure 4: Example of path planning in Coppelia

Remaining & Future Work

To complete this project I need to implement a more robust version of goal navigation, ideally one involving path planning, see figure 4, which will allow the robot to navigate via a pre-defined path at the start of the simulation, making motion-planning easier.

A major requirement is also to properly calculate the hydrodynamics of the robot so that it mimics the properties of water given certain situations. This will also include extending the current goal navigation to allow for goals to be underwater for this to be tested properly, at the moment the robot is restricted on land.

An additional goal was to extend the goal navigation so that the robot wouldn't have to have information about the goal, this would require the front sensor to be used in conjunction with other sensors to recognise and navigate towards the goal. It would also require a lot of research into recognition under different proposed environments using the Vision Sensor offered by Coppelia.

In order to further extend this project I would have to implement a more sophisticated mapping algorithm (SLAM etc) which will provide a more extendable solution over the current X/Y cartesian based implementation.

I would also like to implement a proper motion control system instead of the current version I have, preferably with a voting-based system that can influence the current angle instead of an underlying control system with an overhead influencing factor such as goal or object detection.

Further Information

[1] Robot Center. The ACM-R5 robot. (2013). [Online]. Available: https://www.robotcenter.co.uk/products/acm-r5h. [Accessed 07/03/2020].

[2] E. Kelasidi; K., Y., Pettersen; Jan, Tommy, Gravdahl. (2014). A waypoint guidance strategy for underwater snake robots. 22nd Mediterranean Conference on Control and Automation, Palermo, DOI: 10.1109/MED.2014.6961590

[3] M. Prior. (2020). Github repository of the project. [Online]. Available: https://github.com/EPend/MajorProject [Accessed 07/03/2020].