"Lizardbot"

A reptile-inspired model of robots optimised to navigate rough terrain

Abstract
Insert abstract here

Contents

1   Introduction

Add in the intro pretty much directly from the interim report here

2   Project Aims

Why is the robot being modelled instead of physically built?
Why did I choose to use Unity?

2.1   Primary Objectives

2.1.1   Robot Design

Get from interim report - explain overall design and why those decisions were made e.g. simplistic design

2.1.2   Robot Movement

Basic overview of why each component will move the way it does. Tie each point back to how they are founded (or not founded) in natural algorithms.
Include jumping here

2.1.3   Terrain Generation

The terrain will be static - why?
Why will I have three separate terrains? - Octopus
What is the importance of having a smooth terrain?

2.1.4   AI

How will the AI work? Genetic algorithm outline
Dynamic systems theory
Damage minimisation
How do I want the AI to manipulate the relationship between the body and movement?
Why do I want there to be a relationship between the two? - article Simon sent
How am I going to test the relationship?
How will the robot be measured?

2.2   Extension Objectives

2.2.1   Vision

How would a rudimentary visual system reduce damage to the robot?
How would this move the AI from a reactive to proactive mechanism?

2.2.2   Terrain Friction

How do snakes work with different frictions?

2.2.3   UI

How could a UI help lower the threshold to the project and make it easier to 'work with' the AI?

2.2.4   Flexible Tail

What are the advantages of having a flexible tail?

3   Project Relevance

3.1   Salamandra Robotica II

Insert from interim report

3.2   Agama Robot

Insert from interim report

3.3   tbc

Find a team that have modelled a robot vs building one

4   Requirements Analysis

Insert from interim report - needs some work
Add section on the constraints of this project

5   Professional and Ethical Considerations

Insert from interim report with more reference to code of conduct

6   Implementation

6.1   Terrain

Three terrains were generated using Procedural Toolkit [2] to test the performance of the robot across various environments: rough, uneven, and smooth. These categories were inspired by the those used by the Octopus robot [1].
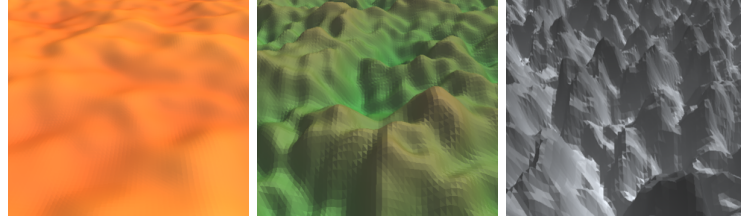

Figure 1: Examples of the three terrain types. From left to right: Smooth, Uneven, Rough.

At one point the height of the terrain was proportional to the number of sections of the robot, a similar method to that of the Octopus robot. However, as the terrain is a control variable the heights were switched to a static value: $Smooth = 8, Uneven = 16, Rough = 24$.

Overall, the rougher the terrain, the higher and more closed in it is. Most of the development of the robot was conducted on the smooth and uneven terrains, as the rough terrain aims to provide a more extreme environment with which to test the efficacy of the AI.

The smooth terrain is deliberately featureless to test the behaviour of the robot in a simple environment. Herbert Simon provided an elegant example of the importance of this consideration: an ant is observed making its way back to its nest across a beach.

Its route is 'a sequence of irregular, angular segments' that suggests some level of complexity in the ant's behaviour. However, the beach for the ant is a much harsher environment than it is for a human. It is more likely that 'its complexity is really a complexity in the surface of the beach, not a complexity in the ant.' Thus, the situatedness of the robot could culminate in behaviours that are not of its own making and are instead caused by its relationship with the terrain. The smooth terrain should reduce the role of the environment and allow for emergent behaviours to be prescribed to the robot itself. It is worth noting that the robot is still being tested on three terrains with some common properties (e.g. gravity) and these factors may introduce bias in the AI. This is a reasonable situation as long as applications of the Lizardbot are further modelled on encounterable terrains to allow the AI to adapt the robot accordingly. For proof of concept the sample set of terrains is sufficient.

6.2   Body

Insert from progress log & CPG log

The velocity that should be applied to this section is calculated using

$$\overrightarrow{v_i} = \overrightarrow{v_{i-1}} + \frac{m}{2}\overrightarrow{w}$$

For rotating sections $i = 0, ..., m$, where $m \leq n$, the value of w will be calculated using S or C as specified.

$$S : \overrightarrow{w} = sin\overrightarrow{\theta_{i-1}} + sin\overrightarrow{\theta_i}$$
$$C : \overrightarrow{w} = cos\overrightarrow{\theta_{i-1}} + cos\overrightarrow{\theta_i}$$

6.3   Tail

Insert from tail log

$$r = \left| x_i - \frac{1}{n} \sum_i^n m_i x_i \right|$$

$$L = \sum_i^n r_i m_i v_i$$

$$v_t = -\frac{L}{r_t m_t}$$

6.4   Legs

$$P = D + V cos\theta + U cos\theta$$

$$v_i = g(P_{i+1} - P_i)$$

6.5   Performance

Insert from trapped algorithm log - get performance from GA

6.6   Genetic Algorithm

Insert from GA log

$$G_i = R^{[0,1]} < m \longrightarrow (max(G_i) - min(G_i))R^{[0.01,0.1]}G_i$$

$$G(1)_i = R^{[0,1]} < r \longrightarrow \begin{array}{l} R^{[0,1]}<0.5 \longrightarrow G(1)_i \\ R^{[0,1]}\geq 0.5 \longrightarrow G(2)_i \end{array}$$

where R denotes a randomly generated number. G(1) refers to the input robot, whilst G(2) is the selected robot. For *Triad* recombination G(2) is randomly chosen from either of the two selected robots, with equal probability.

6.7   Dynamic Systems

Insert from DST log

7   Results

How does mutating the body / movement independently work?
How does the addition of the tail help?
How does initiating the body with a serpentine motion established affect the outcome?
What happens when the body is set as static?
What is the outcome when the legs are out of sync from the body?
Starting with defaults, what parameters does the AI mutate to? What is the corresponding behaviour for this?
Does the AI converge on the same parameters when started with different defaults?

8   Conclusion

9   References

Primary References

[1] Cianchetti, M. (2015). Bioinspired locomotion and grasping in water: the soft eight-arm octopus robot. *Bioinspiration & Biomimetics.*
https://iopscience.iop.org/article/10.1088/1748-3190/10/3/035003
Accessed: 2022-04-10.

Software References

[2] Syomus (2021). Proceduraltoolkit. *GitHub.*
https://github.com/Syomus/ProceduralToolkit
Accessed: 2022-04-10.

Code References

[3] tomvds (2008). Euler, quaternions, radians, degrees... huh? *Unity Forums.*
https://forum.unity.com/threads/euler-quaternions-radians-degrees-huh.53407/
Accessed: 2022-04-11.

10   Appendices

10.1   Code of Conduct