# 

# 

# 

# 

# 

# “Lizardbot”

# A reptile-inspired model of a robot optimised for navigating rough terrain

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# Word Count: 4800

# Introduction

Nature can often inspire elegant and efficient solutions to non-natural problems. The tunnel-building behaviours of ants can be used to generate algorithms to manage traffic flow in a city for example. This can be much more efficient than deploying a group of developers to design an elaborate network to structure how the population navigates the city. Alternatively, the nests of ants could be studied and their tunnel-building algorithm applied to the problem. (Plataforma, 2008)

In this project, the characteristics of various reptiles will be applied to a model of a robot - “Lizardbot” - to optimise the physical design and algorithm it uses to navigate rough terrain. For robots used in applications such as bomb disposal or interplanetary exploration, the environments that they face can be highly unpredictable. The margin for error is often low to non-existent, as there can be little physical access to the robot to fix any issues. These robots require a design and an algorithm optimised to enable them to traverse any landscape they are presented with.

On a climbing trip to the Isle of Portland I took a break from repeatedly falling off the cliff to watch a lizard. It was attempting to jump from the path onto a nearby rock. Every time it failed it paused, then tried again with its tail in a completely different position. It was naturally using its tail to counterbalance its body as it jumped and learning from previous attempts. The addition of a tail to a robot can produce an efficient jumping robot with the application of relatively simple maths. A prime example of this is the ‘*UC Berkeley Leaping Lizard & Robot’* video on YouTube*.* (UC Berkeley, 2012) This behaviour can be observed elsewhere in nature; when a praying mantis jumps it swings its body and abdomen to ensure it hits its target, (Burrows *et al*., 2015) or when cats twist their bodies to reorient themselves to land on their feet. (Diamond, 1988)

The lizard’s stabilising tail, alongside other reptilian characteristics, could provide a nature-inspired solution for a robot targeting rough terrain. Since I lack the resources required to test this by building a physical robot, the popular gaming engine Unity (Unity, 2021) will be used to create a virtual model of the robot instead. The Unity engine has built-in physics, collision, and terrain features that bring complex interactions within scope.

# 

# Project Aims

## Primary Objectives

The project can be divided into three areas: the structure of the model, the movement algorithm, and the generation of the terrain.

## Design

This project will focus on determining how various reptilian characteristics influence the success of the model as it navigates an example terrain. The objective is to explore the ‘ideal’ design of a robot, and the relationship between this body and its navigation of a randomly generated terrain. The successor function will be how far it successfully moves across the *z* axis before it gets stuck. This final state may be bouncing back and forth between two points or hitting a section of the terrain that it cannot progress past; the validation function will consider both of these possibilities.

Within this bigger picture goal there will be several smaller experiments into how various features / approaches influence the outcome.

The body of the robot will emulate that of a snake such that it will be constructed of a series of modular sections that will operate independently. This design ‘*provides the ability of traversing in irregular environments, something that surpasses the mobility of the conventional wheeled, tracked and legged types of robots*’. (Kelasidi and Tzes, 2012a)

The body will be generated dynamically to eliminate the assumption that it must have an even number of legs positioned evenly either side of the body and of identical size. Instead, a number of body sections will be created linearly behind the head.

Legs will be added at random positions along the body and rotate 360° to push the body forward. A tail, added behind the body, will rotate to counterbalance the motion of the body and will become particularly relevant when jumping is implemented. Every joint will have a constraint defining how far it is capable of rotating around each axis. The ideal size and weight for each component will be explored.

To simplify the model of the robot it will be constructed of basic shapes. The body will be a string of spheres, whilst the legs will be spheroids. The decision against cubes/cuboids is to prevent the shapes from ‘catching’ on any other RigidBodies that they interact with, as this could interfere with the results. Of course, with a physical robot there will be non-uniform surfaces to contend with but for the purpose of the model it has been assumed that each surface is uniform.

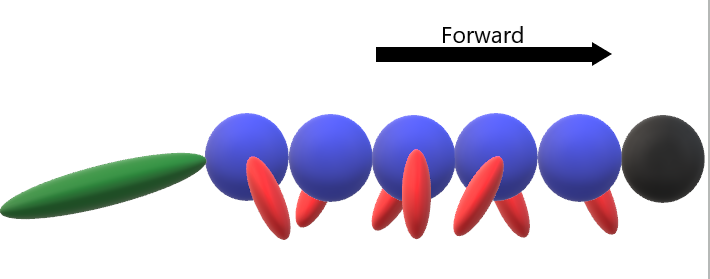


Figure 1: An example of a randomly generated build for Lizardbot (viewed from the side).

The dynamic approach to designing the body of the robot is anticipated to produce greater results as it will, in theory, show that the algorithm is not producing intelligent results as a result of its embodiment. If you take a simple brain (the AI movement algorithm in this case) and provide it with a body then it will produce behaviour. (Brooks, 1991) If the brain can directly influence the body it is provided then, I predict, it will produce *better* behaviour.

To test this, an experiment will be conducted with three different approaches to building the Lizardbot.

1. A pre-designed robot will be used with a fixed number of uniform body sections, each with two identical legs on either side. This body will be unchanged, though the AI controlling the movement of the body will remain.
2. Only the body will be created at first and mutated for a number of generations. At this point the legs will be added and mutated.
3. The full model will be created at the start (such that the body and legs are mutated together).

Comparing the results will explore the relationship between the body and brain of the Lizardbot, with the third option anticipated to result in the robot travelling the furthest across the terrain.

## 

## Movement

### Body

The fundamental movement pattern of the body sections will mimic the serpentine motion of a snake, as shown in figure 2, and will be initialised rotating around the *y* axis only to drive the robot forward in the terrain.

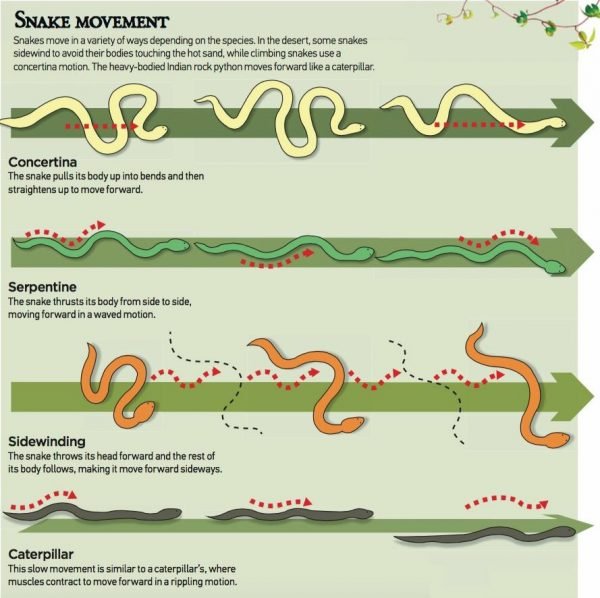


Figure 2: Diagrams of different motions snakes use to move. (Reddy, 2020)

Each section will have the ability to drive forward and/or rotate and will have its own set of parameters (e.g., speed). The first prototype of the model will produce a basic serpentine motion by driving every section whilst rotating alternate sections. The second prototype will aim to improve the fluidity of the motion by using a central pattern generator, as portrayed in figure 4.



Figure 3: An illustration of how the first prototype will drive / rotate each section.

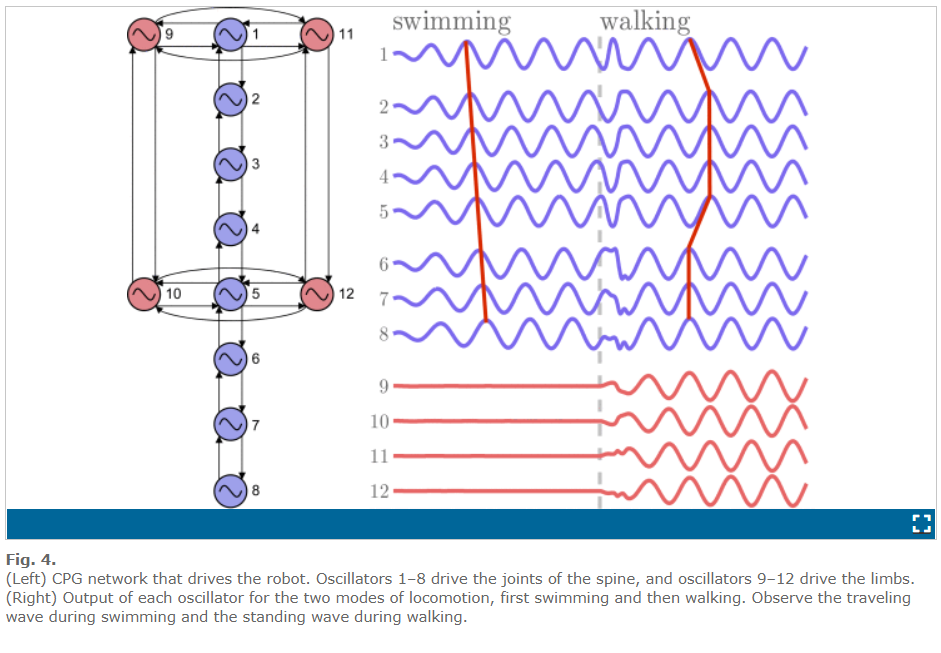


Figure 4: The oscillations created by the central pattern generator used by the Salamandra robot. (Crespi *et al.,* 2013)

Over time a genetic algorithm will mutate the parameters (introducing *x* and *z* rotation) and determine the success of that iteration compared to that of previous generations. This initial setup will be attempting to find the ideal set of parameters to achieve the greatest distance across the terrain.

Due to the physical constraints of the body, it is expected that the resulting motion will still resemble that of a snake. However, the algorithm will not be limited to continuing a serpentine motion.

### 

### The decision to have the body oscillate as opposed to the legs providing the entirety of the motion is supported by *A New Lizard-Inspired Robot With S-Shaped Lateral Body Motions.* (Kim *et al.*, 2020)Due to a lack of degrees of freedom in legged robots a robot may struggle to maintain its posture and direction of movement. *‘A possible solution to such lack of DOFs caused by underactuation may be additional motions of the body’*.

### It is important to note that the referenced study was analysing bipedal robots when this conclusion was drawn. To test the assumption that moving the body with the legs improves performance, another experiment will be conducted to explore the impact of the body remaining static.

### The Salamandra Robotica II used - and inspired - the combination of leg rotation and body oscillation. (Crespi *et al.,* 2013) The Salamandra is covered in more depth later in this report.

### 

### Legs

The approach to the fundamental movement of the legs will be similar to that of the body. As covered in the “*Locomotion of Reptiles*”, (Alexander, 2012) lizards *‘move diagonally opposite feet simultaneously’*. A representation of this is shown in figure 5.

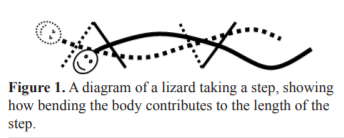


Figure 5: A representation of the relationship between a lizard’s gait and its body motion. (Alexander, 2012)

To implement this motion, diagonally opposite feet will be offset to match the gait of a lizard. While the body continues its own motion, the legs will rotate 360° to push the body forward.

The Salamandra robot (Crespi *et al.,* 2013) inspired the use of a 360° rotation as a simpler mechanism to implement than a biological folding limb. Though the design is somewhat detached from a biological leg, the motion is intended to mimic the gait of a lizard. However, as previously stated for the motion of the body, it will mutate and may ‘evolve’ away from this pattern.

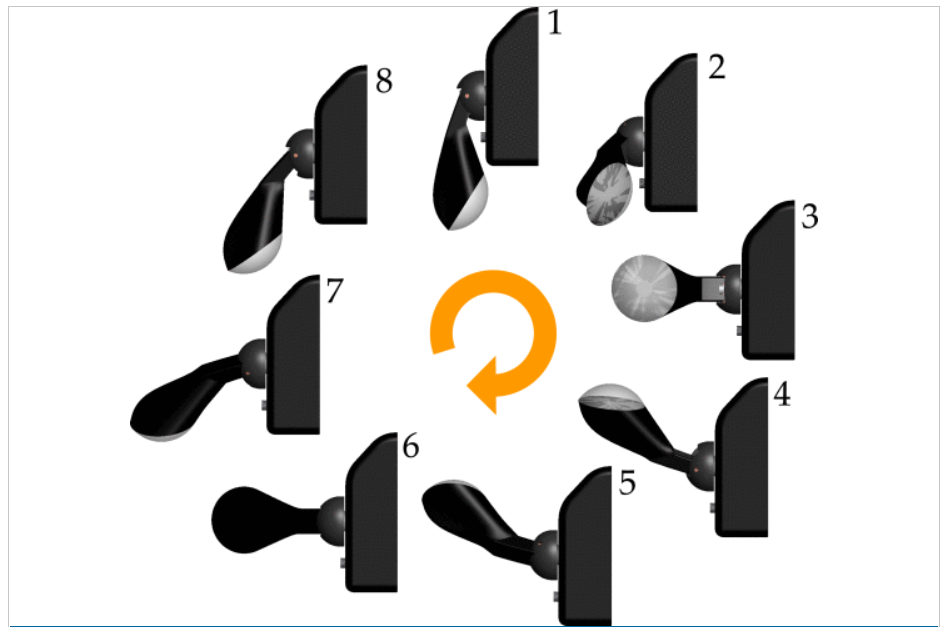


Figure 6: The rotation used by the legs of the Salamandra Robotica II. (Crespi *et al.,* 2013)

### 

### Tail

The motion of the tail will use the same pattern as that of the ‘*Agama robot*’ shown in figure 7, whereby the tail flicks up quickly in the opposite direction of the trajectory of the body. (Libby *et al.,* 2012) The focus of this study was for stabilising a robot as it jumped but it is expected that this counterbalancing of the motion of the body will improve the success of the robot. To test this, an experiment will be performed with and without the tail to determine how it impacts the distance the robot is able to cover.



Figure 7: An agama lizard compared to the agama robot whilst jumping. (Libby *et al.*, 2012)

### 

### Jumping

An additional area that will be explored is a jumping behaviour. The ability to jump would enable the model to explore a much wider range of possibilities within the terrain. It will open up the option to jump across gaps, onto ledges below it (as opposed to falling onto it), and to generally reach higher areas of the terrain.

To accomplish this, spring joints will be added at the boundary between the body section and the base of each leg. When it is determined that the robot has hit a wall that it is unable to progress onto it will angle each ledge at the target and coil the springs. It can then apply increasing force until it is able to land on the ledge - or fail and seek an alternative route. Here the Agama robot’s tail counterbalancing will become particularly relevant to achieve an accurate and balanced jump.

### Improving the AI

Thus far the AI algorithm has been discussed as optimising its parameters to be used across an entire generation, as opposed to changing its behaviour mid-generation to react to a specific circumstance. Once the initial features have been implemented the movement AI will be expanded to use an approach closer to Dynamical Field Theory. (Erlhagen and Schoner, 2002) It will ‘collect’ behaviours. If a specific set of parameters produced a desired behaviour (e.g., *move left*) then this can be activated with increasing force to steer the robot left across the terrain. Figure 8 shows an example of a movement being activated: weakly at first, then again with a stronger activation to produce a greater movement. This approach would produce a behaviour-based approach to navigating the terrain. Theoretically this will improve the distance covered by the robot as it seeks out alternative routes once it becomes stuck.

## 

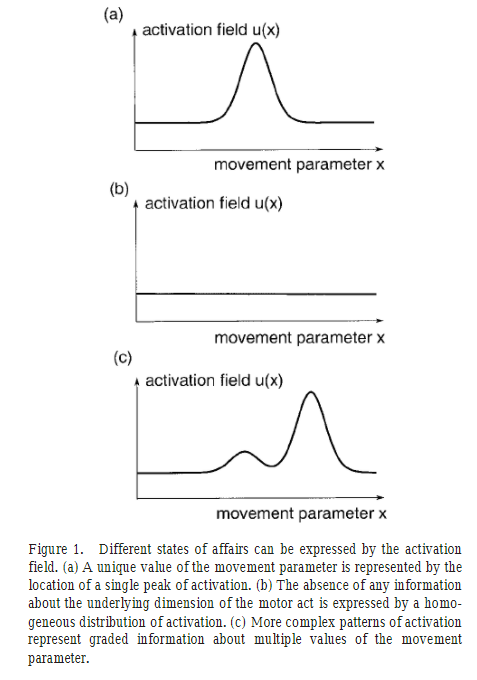


Figure 8: An illustration of how an activation field can be used to produce a desired movement to varying degrees. (Brooks, 1991)

## 

## Terrain

The terrain will be long enough that, even after significant mutation, it is not feasible for the robot to reach the end. The ‘infinite’ terrain will allow the successor function to be continuous rather than a pass / failure.

There are two methods being considered for randomly generating the terrain:

1. Unity’s built-in terrain generator. This would produce realistic terrains that could more easily be adapted to the environment a robot is likely to face. However, it is currently unclear how feasible this method will be within the project timeframe.
2. If option 1 is unsuccessful then another possibility is a 3D map of cubes, cuboids, and square prisms. This approach would be easier to manufacture specific circumstances for the robot to encounter (e.g., a large jump) but is not an accurate depiction of real-world situations.

## 

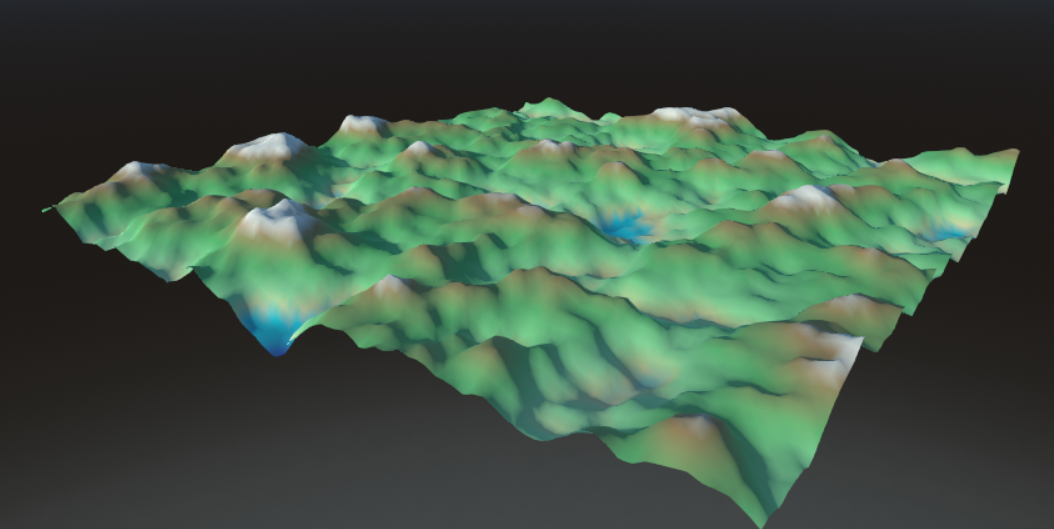


Figure 9: An example of terrain generation using Unity’s terrain functionality. (Popovj, 2018)

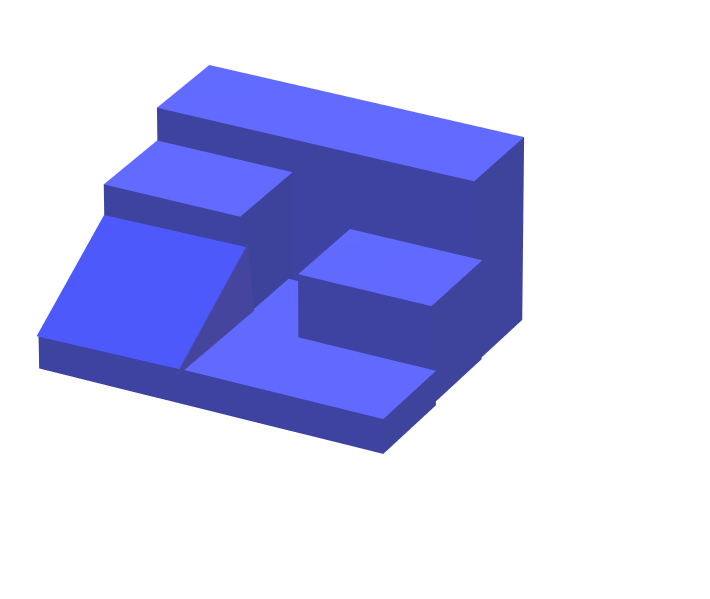


Figure 10: A representation of a terrain generated using basic 3D shapes.

## 

## 

## Extensions

In the instance that the primary objectives of this project are achieved with time remaining then there are some possible extensions that could be implemented:

### Vision

Thus far the Lizardbot has focused on finding the ideal parameters for the overall form and movement of the robot, with additional thought for a dynamic system to allow the robot to react to its situation. There has been little focus on a pre-emptive approach, which has the potential to reduce the risk of damage to a real robot. When a robot becomes stuck it will likely find itself bouncing against the nearby terrain, forcing the prioritisation of durability in the design of the robot. If the robot were given a rudimentary visual system, it could instead anticipate collisions and adjust its course to avoid them.

This extension is the highest priority, as this feature is the most likely of the extensions to satisfy the requirements of a user.

To achieve vision, a camera could be placed in the head of the robot. Unity’s built-in functionality would provide feedback on objects within the range of the camera. This knowledge, when combined with the dynamic systems element of the AI, would allow the robot to make informed ‘decisions’ about how it should approach the terrain it is aware of.

This feature would complement the ability to jump by calculating the point at which it needs to jump. David Lee investigated the factors affecting the specific moment in which an agent will begin reacting to an impending collision. He found that for an agent approaching a stationary object *‘when and how hard he has to brake does not depend simply on his spatial proximity to the vehicle nor simply on his closing velocity and acceleration/deceleration, but on some relationship between these variables. One simple relationship is his temporal proximity to the vehicle—the time-to-collision.’* (Lee, 1976) He found that this relationship could be simplified using the visual system and the relative location of the obstacle in the agent’s visual field.

Using the relative variables at play in the visual field he deduced that the time-to-collision could be calculated using the following equation:

Implementing a visual system to the Lizardbot would unlock a variety of more complex behaviours. Interestingly, the use of this equation would directly relate to the brief of the project as it applies an algorithm derived from nature to a design problem (in this case collision detection and avoidance). ”*Most animals respond avoidantly and directionally to the abstract visual stimulus ... which specifies the approach of an object and impending collision.”* (Schiff, 1965)

Adding a rudimentary vision system would enable the AI to make decisions, as opposed to using the successor function to learn entirely from hindsight. By analysing its surroundings, it could pre-emptively determine which route would have the highest odds of success from previous experience. I anticipate that this feedback loop would drastically increase the distance that the model achieves.

### 

### Terrain Friction

It is important to consider the situatedness of the Lizardbot. A single terrain will be used to test the model and, while it will be generated randomly, the environment that it is presented with will influence the outcome of its cognition. Specifically, it will influence its situated dynamical cognition. This is defined as *“cognition [emerging] from a real-time, continuous and strictly coupled sensorimotor interaction between an unstable subjective experience and an unstable objective world.”* (Da Rold, 2018)

It is not feasible to test the interaction between the Lizardbot and every terrain that a robot could encounter. It is, however, possible to expand the terrain to include common properties. It is known that “*the friction between the snake robot and the ground, affects significantly its motion*” (Kelasidi and Tzes, 2012), hence the coefficient of friction would be a suitable extension to the terrain. By varying the texture of surfaces within the environment the Lizardbot may produce different behaviour.

### Sidewinding vs Serpentine

The base algorithm for the body movement will be established with a serpentine motion. However, Robert Alexander found that *“sidewinding seems to be the most economical of energy of [serpentine, sidewinding and concertina]”.* (Alexander, 2012)

The algorithm could be extended to explore the effect of using a sidewinding motion instead of, or in addition to, serpentine. See figure 2 for a diagram of an example of a sidewinding snake.

# 

# Project Relevance

## Modular Snake Robot

## The work of Ye *et al.* (Ye, 2010) found that a modular snake robot was “*more efficient to get into complicated environments*”. Their aim was to test the abilities of such a robot in environments too harsh for humans. Each module operated independently in a manner similar to the body movement algorithm covered in the project aims. Each module was attached to the adjacent modules via a joint that rotated to create a wave through the body, while a servo drove the module forward. Their robot used a cosine function (ϕi=Acos(2πviT+Δϕi)) to manoeuvre the body, whereas the Lizardbot will make use of a central pattern generator.

The modular snake robot successfully navigated any obstacles with a height less than the height of a module. It also demonstrated the constraints that a physical robot can face (e.g., the error rate in the servos increased with amplitudes over 4cm). Their experiments laid the groundwork for them to explore *“intelligent algorithms for implementing more flexible locomotion and transformation of the modular snake robot to improve its environmental adaptivity.”* These are areas that this project aims to explore without the physical limits that the modular robot faced. The addition of legs and a tail to the modular design is intended to enable the robot to overcome more complex obstacles in an environment. Meanwhile the AI will, in theory, allow the Lizardbot to adapt to its situation more than the modular snake robot was able to.

## Stabilising Robot

Jeongryul Kim *et al.* created a lizard-inspired robot that used a similar snake-like motion with its body to maintain its direction as it drove forward with its two hind legs. (Kim, 2020) The difference with this robot was its treatment of the body and legs as having their own distinct functions: drive versus stability. Testing of this robot was conducted on a treadmill without any obstacles so it is unclear how this robot would perform on an irregular surface.

The Lizardbot will not be aiming to continue in a straight line across a 2D surface - so will not be implementing this stabilisation approach. The movement AI used by the model will instead have a looser link between the legs and the body to react more dynamically to the terrain.

## 

## Agama Robot

The tail-assisted pitch control used to build the Agama lizard-inspired robot will heavily influence the jumping mechanism of the Lizardbot. (Libby *et al.*, 2012) When it is jumping up onto a ledge it will swing its tail downward to counterbalance this force, flipping upwards when jumping down onto a lower surface. Applying this research may also reduce the risk of the robot toppling over if it falls, a situation that could damage a physical robot. The study found that “*the robot with PD feedback tail control maintained a nearly constant body angle by swinging its tail upward and incurred 72% less rotation after a perturbation than did the robot without tail control*”. This is a promising result for stabilising the Lizardbot as it falls.

## Salamandra Robotica II

The relevant goal of the Salamandra Robotica II (Crespi *et al.,* 2013) was to “*advance robotics design for bimodal and efficient locomotion*”. The design of this robot heavily inspired various elements of the design of the Lizardbot: the full rotation of the legs and the central pattern generator for the oscillation of the body.



Figure 11: An image of the Salamandra Robotica II. (Crespi *et al.,* 2013)

One of the key differences between the Salamandra and the Lizardbot is the method for turning. The former used calculated asymmetric oscillations to produce a curving trajectory (shown in figure 12) while the latter will dynamically derive an activation function for a turn.

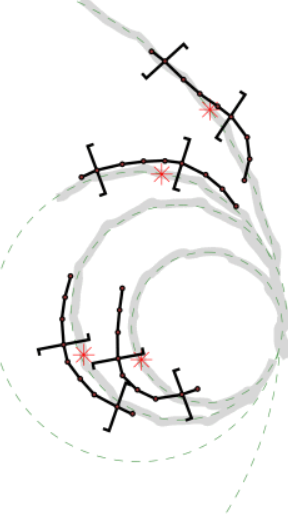


Figure 12: The results of various curved trajectories of the Salamandra Robotica. (Crespi *et al.,* 2013)

Another difference was the Salamandra’s passive tail. The tail fin was only used to test its effect on speed whilst swimming; removing it decreased the speed by 63%. The rigidity of the tail will be matched by this project but, as previously discussed, it will rotate to counterbalance the motion of the rest of the body.

The Lizardbot will incorporate various elements of each of these projects, from the modular design of the body to the tail counterbalancing jumps. The modular snake robot had the same goal as this project: to design a robot that could tackle obstacles in rough terrain. Meanwhile Kim’s robot and the Agama robot were exploring the application of the structure and behaviour of lizards to stabilise a robot. This project will combine the use of a tail for stabilisation (particularly while jumping), the snakelike motion of the body as the legs move, and the modular design to produce a serpentine motion. It will not be constrained by the limits of a physical robot, thus evolving the body of the robot in a manner that these example projects were unable to. The AI will add an element of reactivity to a wider array of situations than these robots, as the model will not be seeking a specific behaviour. Rather, it will determine how the behaviour that it is producing influences its success and absorb this knowledge as it navigates the terrain.

# 

# 

# 

# Requirements Analysis

It is difficult to predict what the specific applications of this project would be and what their range of possible constraints they could have. Are elements of the physical design restricted? What kind of terrain is the robot going to be placed in? Are there other factors that need to be accounted for, e.g., gravity, bodies of water? This project is designed to be broad such that it can be adapted for a narrower use case. It lays the groundwork for research tailored to creating a robot for bomb disposal, planet exploration, or even a robot hoover that can handle stairs!

The features that a target system would include are:

**Testing conducted with a physical robot.**

A model is excellent for testing theories but can overlook the “real physics” that becomes apparent when building a physical robot. For example, the modular snake robot needed to ensure that the robot had “*enough space for installing the joint driving mechanism and the circuit modules*”, settling on each module having size 65mm × 71.5mm × 176mm. (Ye, 2010).

Unfortunately, building a physical robot would be expensive, both in terms of time and the actual cost. It will not be possible to test the model’s performance on an actual robot. This will hopefully be compensated for by having a realistic physics system and a model whose design is directly transferable to the blueprint for a robot.

**Customisable design of the robot.**

A target user may already have known constraints of the design of their physical robot and wish to see this represented in the Lizardbot. If such a constraint is compatible with the existing design of the robot then this will be possible. There will be a section of the code overseeing the generation of the terrain and robot. The user will be able to alter the parameters the controller receives for properties such as each joint’s degrees of freedom or the size of each module.

**Customisable terrain.**

If there were research being conducted into a robot that would encounter steep, mountainous terrain then the terrain generation could be adjusted to account for this. Likewise, if it were designed to traverse flat plains with deep fissures then it would be valuable to test the effectiveness of the Lizardbot in this environment.

The project itself will not provide a UI to customise the terrain. If it is found that Unity’s terrain generation feature is suitable (as discussed in the project aims) then this can very easily be adjusted by the user to suit exactly their requirements. The groundwork will have been completed for them and they will simply need to alter the heightmap and parameters that the terrain generator uses.

**Realistic physics interactions.**

To achieve a suitable successor function, the robot will need to interact realistically with the terrain around it. If it were to remain on a ledge in a precarious position that any physical robot would have immediately fallen off then this is not a realistic model.

This need was a significant factor in the decision to use Unity, as opposed to other modelling software such as MSC Adams. (MSC Adams, 2021) Two basic tests have already been conducted to verify the suitability of both the prototype of the Lizardbot and of Unity’s physics system. The first spawned a prototype of the robot onto an elevated block and its behaviour was exactly as anticipated. It remained on the block until the weight of the hanging body caused it to overbalance and fall off. The second tested its reaction to a collision with a vertical surface. It correctly hit it, bounced off with appropriate force and any force toward the wall resulted in motion parallel to it.

It is expected that collisions and gravity will be accurately modelled, with credit to Unity.

**Robot design is directly transferable.**

It would be beneficial for the target user if there were a direct correlation between the design of the model and the physical structure of a robot based on the model. The model discussed in this paper has a vague shape, but the underlying structure would be transferable to a blueprint for a robot. The joints connecting the modules would provide the constraints that the robot would need to implement. The model will deliberately use realistic constraints to avoid outcomes that cannot be replicated (e.g., the maximum degrees of freedom will be bound).

**Reduction of risk to robot.**

Regardless of the purpose of the robot, any user will want their robot to avoid behaviour that risks damage. This function can be somewhat achieved by having the AI predict the upcoming terrain and proactively decide how it will navigate it. By avoiding collisions and falls, the largest risks can be reduced.

The model could approach this objective in a variety of ways. A form of proactive decision-making has been established as a possible extension to the AI, so will be included if time allows.

Additionally, the behaviour of the tail will theoretically lessen the damage if it successfully levels the robot if it falls. An experiment will need to be conducted to confirm the result of the tail flicking upwards as the robot falls but it is anticipated to somewhat reduce the risk of damage to the robot.

**AI can free the robot when trapped.**

A target user would expect that the robot would attempt to free itself should it become stuck. As discussed in the project aims, a dynamic systems approach will be implemented to trial and error an alternative route. Time permitting, the extension to add a visual system will provide a mechanism to avoid becoming trapped and will better equip it to react to the situation when it does. It would be able to scan its neighbouring terrain and determine the most likely route forward.

**Memory of the terrain.**

If the AI had the ability to remember where in the terrain it had previously encountered then it could avoid moving in circles and backtrack its steps in the instance that it becomes stuck. Due to the time constraints of the project the focus will instead be on which nature-derived characteristics can be applied. It can be argued that memory is natural, and the memory of lizards could be explored, however this will not be the focus of this project.

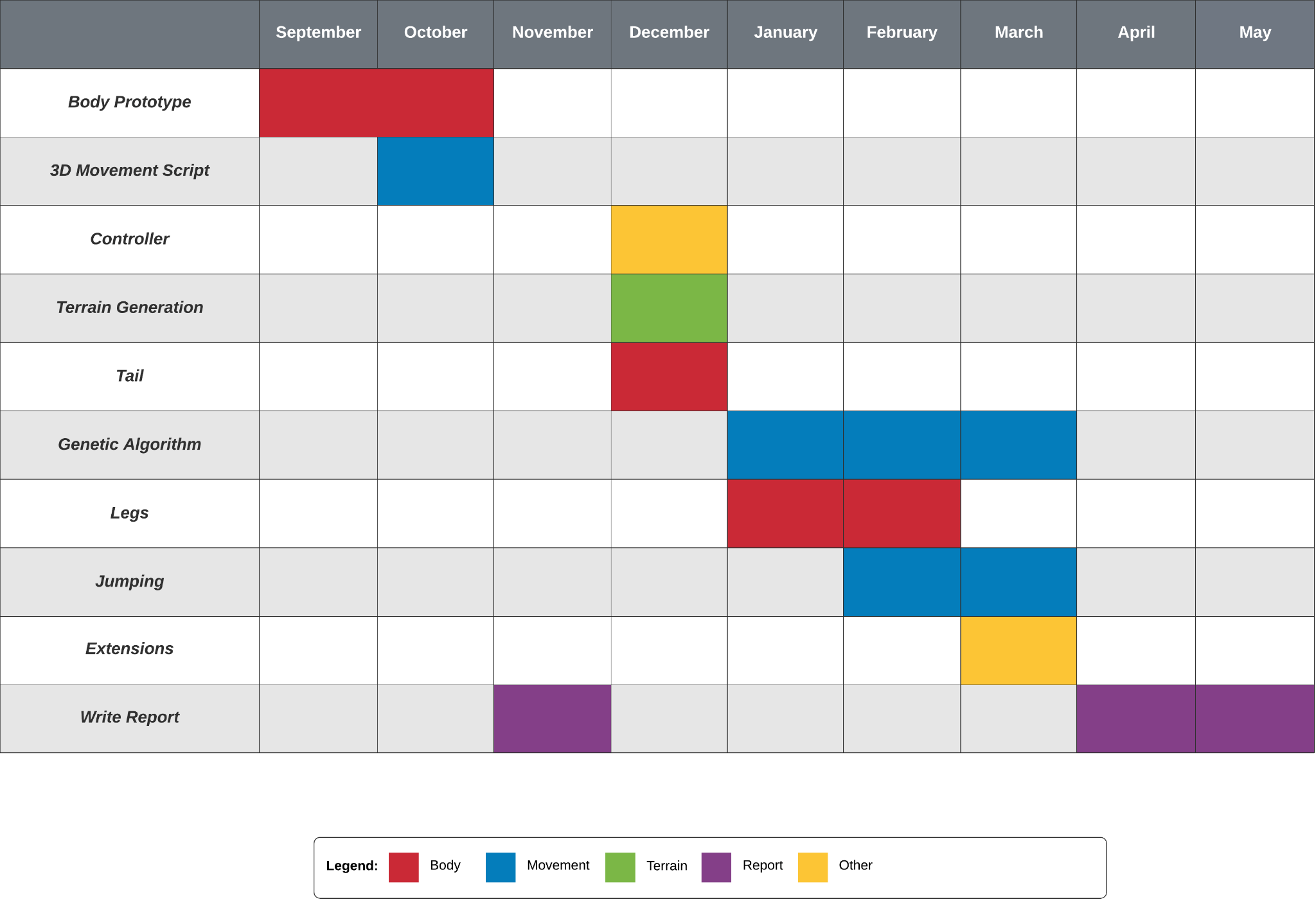
# Professional and Ethical Considerations

This project will be an isolated model without any user testing, personal data, vulnerable people, protected characteristics, or medical data. While it does focus on reptiles there will not be any experimentation on any animals. Referenced studies will have faced their own ethical review and will largely use videos of the animal being studied.

# With these considerations, this project has been classed as low risk. As of 9th November 2021, it has been approved by the project supervisor and is pending approval from the SREO.

# Project Plan

The Gantt diagram outlining the plan for this project is shown below. Currently the controller is in progress while the body prototype and 3D movement script are complete. As outlined in the project aims, the addition of vision-based decision making, differing terrain coefficients of friction, and a sidewinding body motion are the possible extensions.



## 

# References

Alexander, R. (2012) "*Locomotion of Reptiles*". Herpetological Bulletin, Number 121.

Available at: <https://www.thebhs.org/publications/the-herpetological-bulletin/issue-number-121-autumn-2012/3-1-locomotion-in-reptiles/file>

Brooks, Rodney. (1991). *"Intelligence without representation*". Artificial Intelligence 47, 139-159.

Burrows, Malcolm. et al. (2015) "*Mantises Exchange Angular Momentum between Three Rotating Body Parts to Jump Precisely to Targets*". ScienceDirect.

Available at: <https://doi.org/10.1016/j.cub.2015.01.054>

Crespi, Alessandro. (2013) "*Salamandra Robotica II: An Amphibious Robot to Study Salamander-Like Swimming and Walking Gaits*". IEEE.

Available at: <https://ieeexplore.ieee.org/document/6416074>

Da Rold, Federico. (2018) "*Defining embodied cognition: The problem of situatedness*". New Ideas in Psychology: Vol. 51.

DOI: 10.1016/j.newideapsych.2018.04.001

Diamond, 1988.

Diamond, Jared. (1988) "*Why cats have nine lives*". Nature.

DOI: 10.1038/332586a0

Erlhagen, Wolfram. Schoner, Gregor. (2002) "*Dynamic field theory of movement preparation*". Psychological Review.

Available at: <https://www.researchgate.net/publication/11287764_Dynamic_field_theory_of_movement_preparation>

Kelasidi, Eleni. Tzes, Anthony. (2012) "*Serpentine motion control of snake robots for curvature and heading based trajectory - parameterization*". IEEE.

DOI: 10.1109/MED.2012.6265693

Kim, Jeongryul et al. (2020) *"A New Lizard-Inspired Robot With S-Shaped Lateral Body Motions*". IEEE.

DOI: 10.1109/TMECH.2019.2953598

Lee, David. (1976) *"A theory of visual control of braking based on information about time-to-collision*". Perception: 1976 Vol. 5.

DOI: 10.1068/p050437

Libby, Thomas et al. (2012) "*Tail-assisted pitch control in lizards, robots and dinosaurs*". Nature.

Available at: <https://www.nature.com/articles/nature10710>

MSC Adams. (2021) *"Adams - The Multibody Dynamics Simulation Solution*". MSC Software.

Available at: https://www.mscsoftware.com/product/adams

Plataforma SINC. (2008) "*Self-steering Vehicle Designed To Mimic Movements Of Ants.*" ScienceDaily.

Available at: <https://www.sciencedaily.com/releases/2008/09/080917074130.htm>

Popovj, Roman. "romanpopovj" (2018) "*Low Poly Terrain Generator*". Reddit.

Available at: <https://www.reddit.com/r/Unity3D/comments/bbo2d7/low_poly_terrain_generator/>

Image available at: https://i.imgur.com/uCczRmL.png

Reddy, Siddartha. (2020) "*How do Snakes Move?*" Learn Natural Farming.

Available at: <https://learnnaturalfarming.com/how-do-snakes-move/>

Schiff, William. (1965) "*Perception of impending collision: A study of visually directed avoidant behavior*". Psychological Monographs: Vol. 79, Iss. 11.

DOI: 10.1037/h0093887

UC Berkeley. (2012) "*UC Berkeley Leaping Lizard & Robot*". YouTube. Available at: <https://www.youtube.com/watch?v=jOUAIRbrv6s>

Unity. (2021) *"2020.3.19*". Unity Technologies.

Available at: <https://unity3d.com/unity/whats-new/2020.3.19>

Ye, Xiaofeng *et al.* (2010) “*Locomotion control for a modular snake robot over rough terrain”*. 2010 The 2nd Conference on Environmental Science and Information Application Technology.

DOI: 10.1109/ESIAT.2010.5567368

[All links accessed in November 2021]

# 

# Appendices

## Project Proposal

**Lizard Bot**

*Exploring whether a genetic algorithm based on reptiles’ movement can optimise a robot model for movement over rough terrain.*

# Aims

Robots used for purposes such as planet exploration and bomb disposal can face unpredictable or rough terrain. How can such a robot be designed and programmed to enable it to navigate these landscapes?

I aim to explore this problem by modelling a robot in Unity and using genetic algorithms to refine its structure and movement algorithm. The fundamental design of both of these will be heavily influenced by nature - specifically reptiles. Lizards use their tail to counterbalance their motion, snakes move by coiling their body, and geckos use suction to allow them to scale steep surfaces. These features could provide the basis of a nature-inspired simulation of a robot optimised for difficult terrain.

# Objectives

* Design a basic snake-like model of a robot with a movement algorithm that will allow it to coil/uncoil like a snake. The body should be created dynamically and the parameters for its movement generated randomly.
* Create a ‘tail’ for the robot that will mimic how lizards use their tail to counterbalance their motion. This will enable exploration into how a tail feature impacts the success of the robot.
* Extend the robot’s body to include legs. These will also be created dynamically with random variables (e.g. position, degrees of possible rotation, size).
* Generate random terrain for the robot to navigate. This will take the form of a series of cubes / triangular prisms with varying gradients.
* Create a genetic algorithm controller for the robot to experiment which body forms and movement parameters are most successful at moving the furthest in the terrain.

# Extensions

* Explore other factors within the terrain for the robot to interact with. These could include adding varying coefficients of frictions to the terrain, including bodies of water for it to cross, changing the gravity etc.
* Extend the movement algorithm to incorporate different methods snakes use to move. This would extend the foundations of the genetic algorithm and determine how influential the assumptions made within the first version of the script were.
* Add elements like suction to the legs of the robot (based on geckos). This could be used to investigate how the robot could climb steep / overhanging terrain.
* Add ‘eyes’ to the robot - how would knowledge of the upcoming terrain influence the genetic algorithm?

# Timetable

|  |  |  |  |
| --- | --- | --- | --- |
| Basic robot & movement | October 2021 | Genetic Algorithm V1 | January 2022 |
| Terrain generation & tail | November 2021 | Extensions | February 2022 |
| Legs & 3D motion | December 2021 | Extensions & GA V2 | March 2022 |

# 

# Bibliography

Alexander, R. (2012). ‘*Locomotion of Reptiles’.* Herpetological Bulletin, No. 121.

Available at: <https://www.thebhs.org/publications/the-herpetological-bulletin/issue-number-121-autumn-2012/3-1-locomotion-in-reptiles/file>

Fernandez-Rodriguez, Irene; Brana, Florentino. (2020). ‘*The movement dynamics of autotomized lizards and their tails reveal functional costs of caudal autotomy’.* Integrative Zoology.

DOI: *10.1111/1749-4877.12443*

Ye, Xiaofeng; Niu, Yanpeng; Wang, Hui; Meng, Tianhua. (2010). ‘*Locomotion control for a modular snake robot over rough terrain’.* 2010 The 2nd Conference on Environmental Science and Information Application Technology. IEEE.

DOI: *10.1109/ESIAT.2010.5567368*

Belter, Dominik; Łabęcki, Przemysław; Skrzypczyński, Piotr. (2015). ‘*Adaptive Motion Planning for Autonomous Rough Terrain Traversal with a Walking Robot’.* Journal of Field Robotics.

DOI: *10.1002/rob.21610*

[All links accessed in October 2021]

## 

## Progress Log

* *22nd September 2021*. I had my initial meeting with my supervisor, Simon Bowes, whereby we had our initial conversation about the general idea for the project. We discussed which characteristics of reptiles could be implemented, what the potential use cases for the robot could be, and how the project would take shape. This was a productive meeting as Simon questioned any assumptions I had made and pushed me to consider how I would be approaching the coding element of the project.

The outcome of this meeting was that I had a better idea of how to tackle the project and was able to start creating the first design of a snake-like model.

*30th September 2021 - 8th October 2021.* The first form of the robot model was constructed of a series of ‘sections’ and ‘joints’. Each section would be locked into one of the joints adjacent to it and thus would move in parallel.

Each joint would be set to initialise its rotation either clockwise or anticlockwise (around the Y axis only while I established a prototype) until it reached its maximum angle. Each section was set as the child of the joint **in front of it** such that as the joint rotated the section ‘locked to it’ would maintain its relative position and rotation.

Whilst testing the first movement script I used a body with four joint/section pairs, with the joints having respective maximum angles {60c, 0, 60ac, 0} to create a ‘wide coil’ as only half the joints would rotate.

Once the maximum angles were achieved it would switch to ‘uncoiling’. Each section would be locked to the joint **behind it** while the joint rotated to return back to 0°.

The resulting motion was the robot coiling into an S shape, with the head retaining its position and the rest of the body being pulled forward. It would then uncoil by reversing the rotation of the joints to push the body forward.

* *13th October 2021*. I met with Simon to demonstrate the basic prototype of the robot and confirm that there were no issues with my project proposal. In this meeting we discussed how the robot, still using a joint/section rigid motion, would be expanded to mimic the motion of a snake more closely. Simon questioned how I was planning on generating the terrain; did Unity have a built-in feature I could use to create more realistic terrains? How would I be implementing different coefficients of friction in the terrain? These questions gave me some food for thought in the months leading up to beginning work on the terrain.
* *8th October 2021 - 21st October 2021.* I refactored the movement script from one overall script that would serially manipulate each joint to a smaller script for each individual joint to execute in parallel. This was a more elegant structure to the code and also fixed the issue with the robot rotating slightly to one side over time as the first few joints were always rotated prior to the last ones.

The first movement script used *Rotate* which would instantaneously rotate the joint *x°.* The issue with this was that it culminated in ‘jerky’ movement that would cause issues when the robot began interacting with the terrain. I explored the use of *Lerp* to interpolate the overall rotation over a series of frames to create a smooth rotation.

I found a major issue with my approach up until this point: the model was not compatible with Unity’s physics system. Both *Rotate* and *Lerp* directly manipulate the transform of the object, overriding its RigidBody component and causing it to ignore collisions. I recognised that I needed to return to the drawing board and explore other options.

* *21st October 2021 - 27th October 2021.* I researched alternative methods for the design of the robot that would be compatible with Unity’s physics system. This culminated in a redesign of the model to remove the joint objects and replace them with Unity’s configurable joints. Configurable joints behave similarly in terms of locking object’s relative positions together, whilst allowing them to rotate around each other. The implementation more closely models how a physical robot would be built, as the constraints of the joint can be adjusted to prevent rotation past predetermined limits.

The individual sections were changed to rotate using *RotateToward:* a method that adds a vector force to the object’s RigidBody.

The implementation of this new structure provides a much more fluid model that behaves more similarly to a real-world robot. One test saw the model spawn above a floating block, such that its second section hit the block while the rest of it was hanging. The previous prototype would either pass straight through the block, or hit it and behave like a solid object had hit it. In contrast, this improved model hits the block and the other sections dangle off the block. As it begins executing the movement script it moves enough of the body off the block that the centre of gravity drags it off the block and onto the floor.

* *27th October 2021*. Having successfully created a working 3D model, I arranged a meeting with Simon to demonstrate it and discuss how the move to Unity’s configurable joints would affect my project. I had reached this milestone sooner than anticipated; as a result I was confident that I could move some of my project extension ideas into the main project brief. Simon also clarified what I needed to do for the ethics review.

## 