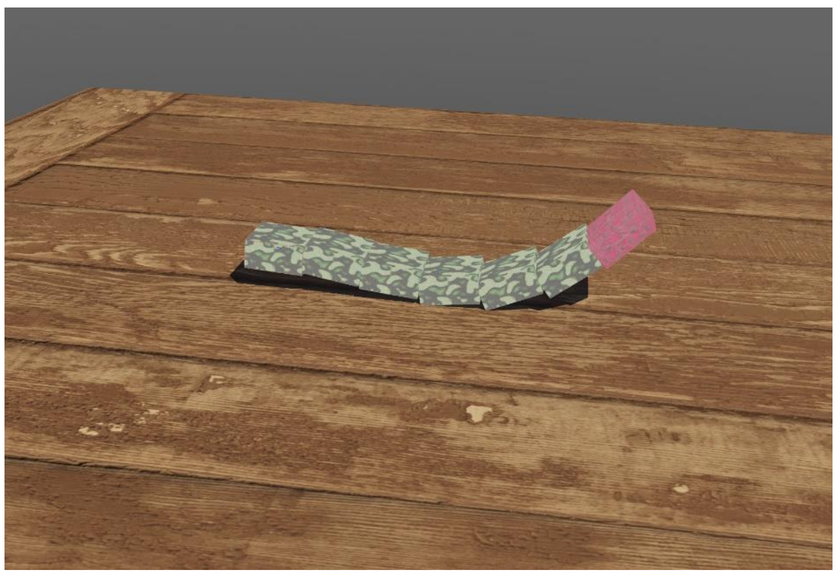
Serpentine Locomotion in Difficult Terrain

Project Report



This work was done for the course "Mobile Robots" lectured at faculdade de ciências de Lisboa in Portugal during the 2019-2020 academic year.

Martim Viana nº 55099

João Pinto nº 55018

1) Introduction

The topic of our work stems from our curiosity in serpentine robots and how they can be used to transverse difficult terrain. Serpentine robots are not new, they have been researched thoroughly by many scientists including Dr Hirose and his team (M. Saito) (Masaki Yamakita) whose designs are today build commercially by the British company Robot Center (Robot Center, n.d.).

Nevertheless, we feel it's an area with high potential that we desired to study. Our project aim was to build a simple robot, from the ground, that mimics the movement of a serpentine and therefore can move in a scenario. We also had the ambition to build into the robot some semi-controlled behavior like for example chasing another robot or find his way in the scenario.

Due to the nature of the topic and the global health emergency that turned 2020 on its heels, keeping us away from the campus, we projected to do our work in a simulator. Choosing a simulator was tricky as we did not find many that allowed to do what we wanted to accomplish and is the nature of course work to be time constraint. This does not allow much breathing room to learn the ins and outs of a complicated simulator and develop the project. In the end, we decided to take a leap of faith and choose the simulator *webot*s from *Cyberbotics*, hoping that the effort needed to learn how to use can be reflected in the quality and originality of our project.

2) Approach

When we started the project, we immediately faced the question on how to use a simulator that we know little to nothing about? by studying it of course. We are not going to detail the process and struggles we faced in learning to use. But we do want to credit the makers of the tutorials both in the website (Cyberbotics, n.d.) and in YouTube (Soft illusion, n.d.) as essential in our ability to understand the basics of this complex simulator.

We divided the work, as Fig1 shows, in two major areas of concern and two minor ones. The first major one was building the robot itself and all the components needed for it to work, the second major was simulating the serpentine movement and all the math involved in its wonders. In the minor ones we placed building the environment of the simulation and implementing a minor control mechanism in the robot.

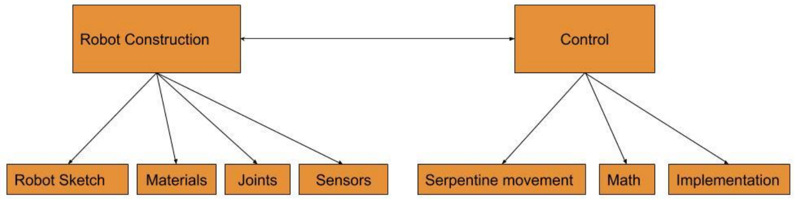


Fig1: The two major concerns

Building the robot

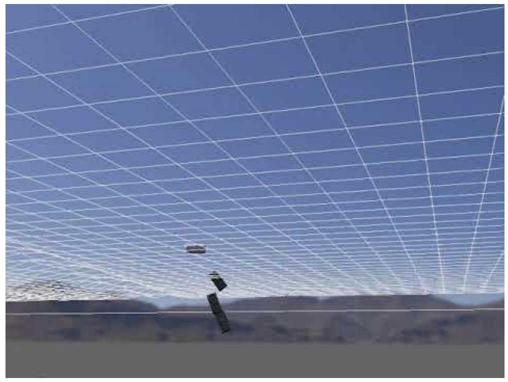
We began by drawing a sketch of our idea for the robot as Fig2 shows. We knew we wanted rounded geometric forms somehow connected to each other, we figured that five would probably be enough (we were wrong).

Fig2: Initial Sketch

The webots simulator present us with several choices for the geometric forms namely boxes, capsules, cones and pipes. We tried to build the robot with pipes first (shown in fig3) as it seemed to us the best choice, alas when we tried a basic movement it didn't work as we thought it should. We reckon that the physics of the pipe may not be implemented properly but we found no documented proof of our claims. By a method of trial and error we ended using boxes to build the robot as they appear to work best in our initial tests.

Fig3: First attempt using pipes...

Or second issue was how to connect the geometric pieces together in a way that they can have enough degrees of freedom to be able to make the serpentine movement. Joints are used for this purpose and here, like previously, Webots simulator offers multiple choices: SliderJoints simulating a translation motion offering one degree of freedom, *Hidgejoints* simulation a rotation motion offering two degrees of freedom and *balljoints* simulating a rotation motion with three degrees of freedom. We choose to use the *hidgejoint* as it the one that fitted or needs. However, we made the initial mistake off connecting all the joints to the “head” of the robot instead to each geometric box in successive order and the result was less than ideal as show in video 1.



Video1: Mistakes were made...

<u>Simulating the serpentine movement</u>

After the robot structure was built, we needed to tackle the main issue of our project: The movement of the robot. The serpentine movement has been studied in detail and its locomotion formulas calculated in different ways by different people. It was not our intention to re-invent the wheel but to successfully apply them to our project.

We started out by using the serpentinoid curve equation, which was introduced by Hirose. Despite the robot assuming a sinusoidal curvature, it failed to take time into consideration, and therefore not describing an accurate serpentine locomotion.

$$\rho(s) = \frac{-2K_n \pi \alpha}{L} \sin\left(\frac{2K_n \pi s}{L}\right),$$

Equation 1

In the equation 1, L is the whole length of the snake, Kn is the number of wave shapes, α is the initial winding angle of the curve and s is the body length along the body curve.

Next, we tried using the acceleration equation 2 [8] which, despite showing some initial promises, was scraped as we found out what webots doesn’t allow setting negative accelerations nor speeds, and our homebrew simulations showed no promises.

$$\ddot{s} = \begin{cases} a & 0 \leqslant t < T/10 \\ 0 & T/ \leqslant t < 9T/10 \\ -a & 9T/10 \leqslant t < T, \end{cases}$$

Equation 2

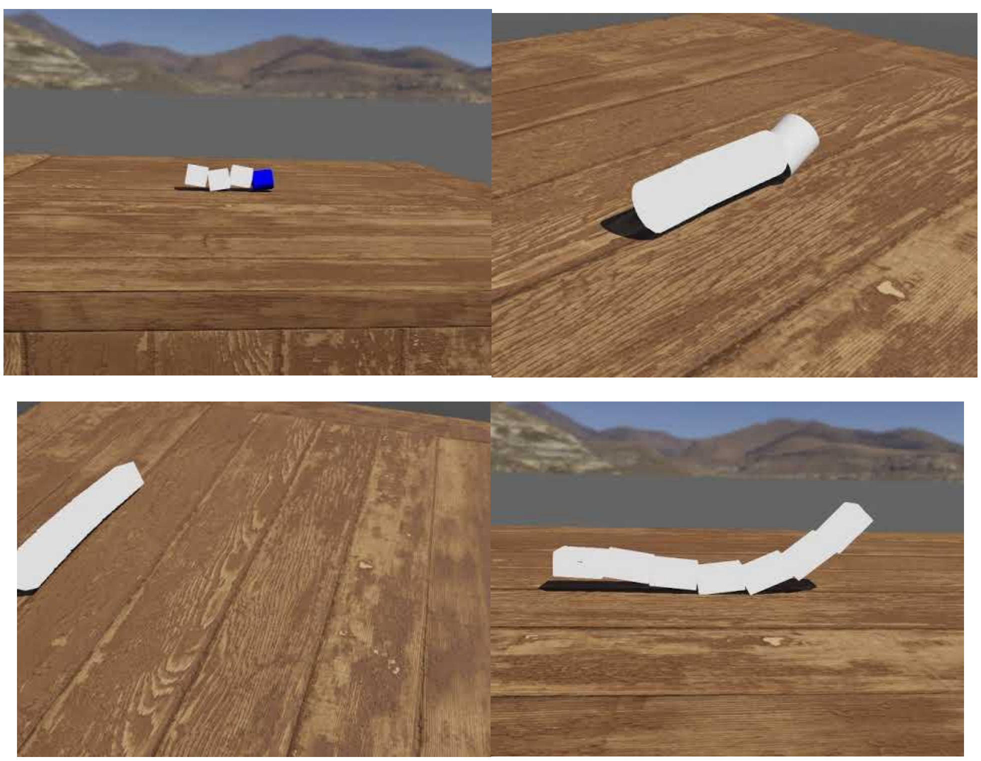
Finally, we've settled on equation 3 [10]:

$$\kappa(s,t) = \alpha \cos(2\pi(s + t/T))$$

Equation 3

Where α is the amplitude of the movement, s is the normalized arc length, which is between 0 and 1 and T is the wave period.

Having the equations is one thing, implementing them is quite another and in our way to implement the robot we had many failed interactions like the video 2 to 5 can attest:



Video 2-5: failed attempts in serpentine movement

In the end we realized that we needed to opt for a centralized control philosophy for it to work. We also needed to add many more connecting boxes and joints to the robot that we initially though for it to have enough "strength" to move.

3) Implementation

After all our work during the previous phase implementation was a tad less complicated, it had however a few challenged to be overcome in order to be able to put all the pieced together.

<u>The robot:</u>

The robot was built like figure 4 shows. While the robot works perfectly in the simulator, we are unsure of this design viability in the real world as serpentines living tissue tends to be malleable and solid boxes are not. Most "real world" robot that we found, while researching this project, seem to have rubber tips around the contact areas. We reckon that this is would be an inevitability if we attempted to transport the robot from the virtual to the real world.

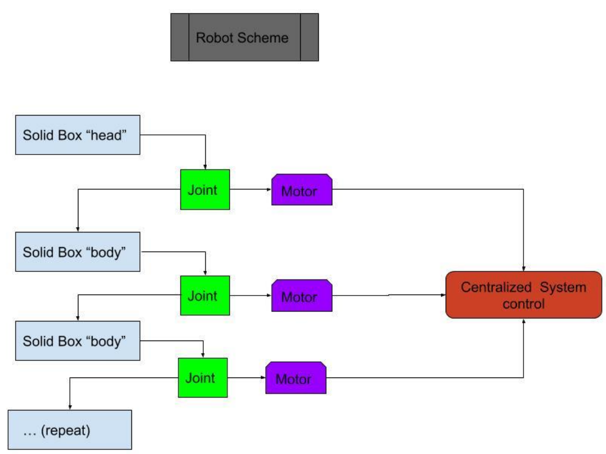


Figure 4: Schematic of the robot

<u>Terrain:</u>

One of our goals for this project was to build and outdoor environment as the navigation through unregular terrain is one of the main advantages of Serpentine robots. This is shown by the "real" robot ACM-R5H from robot center (Robot Center, n.d.). To build this we used the solid "pit", seen in figure 5, that Randomly generates uneven terrain based on Perlin noise. (Cyberbotics, n.d.) We tweaked and studied it so it could fit our needs and goals.

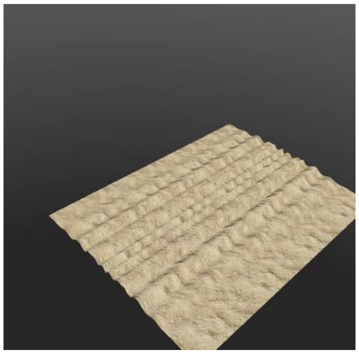
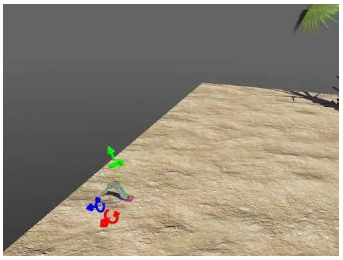


Figure 5: Building an uneven terrain

<u>Sensors and control:</u>

Our last goal to this project was to build control and/or a purpose for the robot. For example, to follow another robot or find an object in the environment. To accomplish this, we implemented in the robot a Lidar sensor. Lidar is a method for measuring distances (ranging) by illuminating the target with laser light and measuring the reflection with a sensor. Differences in laser return times and wavelengths can then be used to make digital 3-D representations of the target. (Wikipedia, n.d.)

With this sensor we though to be able to measure the distance to an object and make decisions with that information. Unfortunately, we couldn't make any lidar model available to webots to work. Since all of them cover a 270º angle and, as such, they registered the floor behind the robot as the closest distance between an object, therefore decreasing the robot's amplitude until it stops as is shown in Video 6. To solve this problem, we switched to a standard proximity sensor using lasers that produced the results we were aiming.

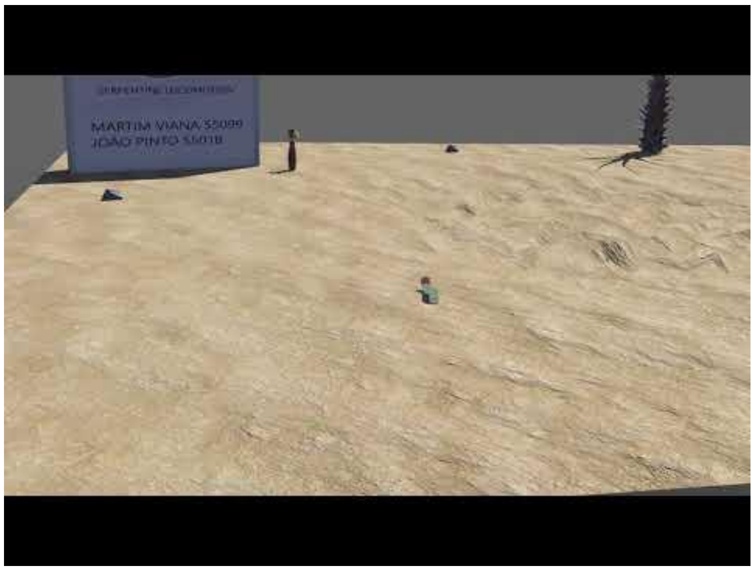


Video 6: Lidar problems

Unfortunately, we were unable to achieve our goal of following another robot as this proved far too complex for the time we allotted to this project but we were able to stop and avoid an obstacle if it presented itself in front of its head.

4) Results:

In the end of the project we were able to build a simple serpentine robot that moves in an uneven terrain as video 7 shows (although we admit it moves more like a worm than a snake). While we are overall pleased with the results he still does not move flawlessly in the terrain if its to rugged or has dramatic features (like High climbs or descents).



Video 7: Final result

5) Final Comments and future work

Our goal was to build a snake like robot that could move in an uneven terrain and we feel that we achieved that goal. It was an ambitious goal due to special circumstances of this year 2020, the short time we had to develop the project and the need to learn a complex tool like Webots from ground zero.

Due to this a considerable amount of work was left to be done in correcting bugs and implementing new features, we will discuss some of future work in the next few paragraphs.

i. Lateral movement and turning: Serpentine movement are very complex. One thing is to move forward, and another is the lateral moves that we see snakes do in real life and is a testament to the wonders of nature and its evolution. Implementing that kind of movement is a challenge for future work, It’s a feature that could prove particularly useful to avoid obstacles.

ii. Turning 180 degrees: Like lateral movement turning 180 degrees and go the opposite direction the robot is it is complicated due do the length of the snake, especially if the space the robot has to operate is tight.
iii. Control and goals: Like we admitted previously this is one of the features that we were unable to complete due to time constrains. We would like to improve on this making the robot perfectly avoiding objects in the environment (we would need the two features already discussed) and to be able to find some kind of object in it. As this robot are particularly suitable to search in dangerous and not easily accessible places.

iv. Building a physical version: The ultimate goal would be to transport the simulated robot to the real world. We saw many examples in YouTube and other media of this kind of robots made by fellow students and researchers with impressive results.

6) Bibliography

[1] Cyberbotics. (n.d.). Retrieved from Webots User Guide: https://cyberbotics.com/doc/guide/tutorials

[2] Faura-Molina, J. A. (n.d.). *Snake-Like Locomotion using Smart Materials*. Retrieved from http://academic.uprm.edu/pcaceres/Undergrad/Snake/id3.htm

[3] M. Saito, M. F. (n.d.). Modeling, Analysis and Synthesis of Serpentine Locomotion with a Multilink Robotic Snake.

[4] Masaki Yamakita, N. K.-W. (n.d.). A Snake-Like Swimming Robot Using IPMC Actuators and verification of Doping Effect.

[5] Robot Center. (n.d.). *ACM-R5H*. Retrieved from https://www.youtube.com/watch?v=o2qm7-HFHt4

[6] Soft illusion. (n.d.). *Soft illusion Youtube channel*. Retrieved from https://www.youtube.com/channel/UCrl9pLcAAKy8wuXkN-on3xQ/videos

[7] Wikipedia. (n.d.). *Lidar*. Retrieved from https://en.wikipedia.org/wiki/Lidar

[8] Akbarzadeh e H. Kalani, «Design and Modeling of a Snake Robot Based on Worm-Like Locomotion», Advanced Robotics, vol. 26, n. 5–6, pp. 537–560, Jan. 2012, doi: 10.1163/156855311X617498.

[9] J. Gonzalez-Gomez, E. Aguayo, e E. Boemo, «Locomotion of a Modular Worm-like Robot Using a FPGA-based Embedded MicroBlaze Soft-processor», em Climbing and Walking Robots, Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 869–878.

[10] C. D. Onal e D. Rus, «Autonomous undulatory serpentine locomotion utilizing body dynamics of a fluidic soft robot», *Bioinspir. Biomim.*, vol. 8, n. 2, p. 026003, Mar. 2013, doi: 10.1088/1748-3182/8/2/026003.