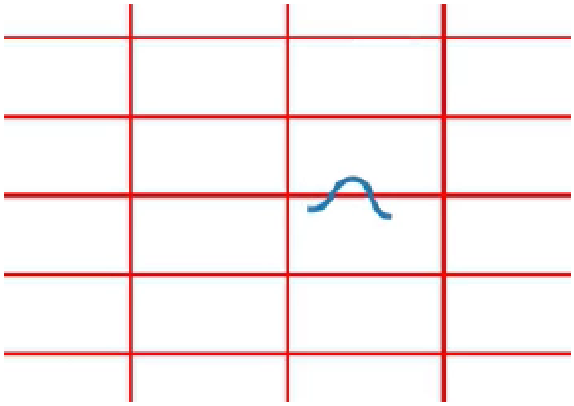
Assignment: Slithering Snake Motion

Biologically Inspired Autonomous Systems

Karina Bay Bloch (Kablo17) & Nicoline L. Thomsen (nthom18)

18/05-2022



Contents

1 Introduction **2**

2 Task 1 - Changing Friction coefficients **2**

3 Task 2 - Changing magnitudes forces applied to the body **3**

4 Task 3 - Changing rod parameters **4**

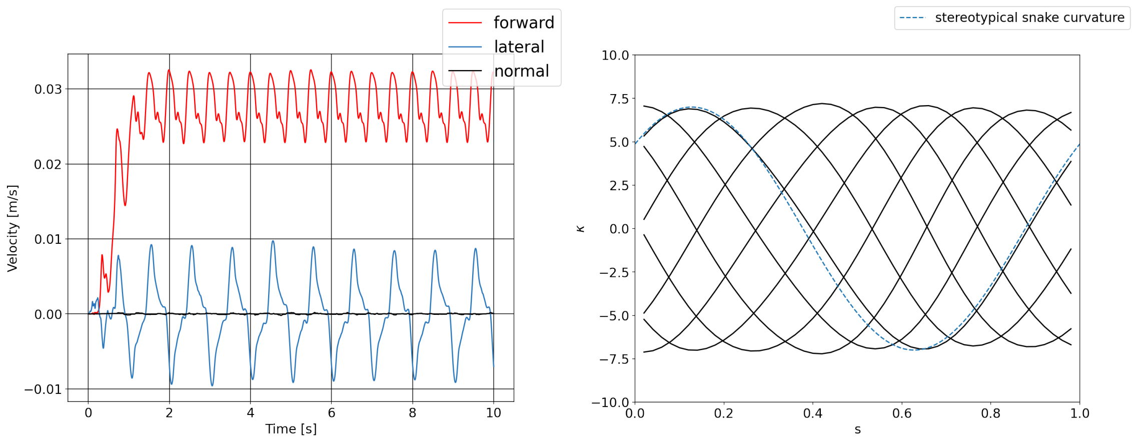
5 Task 4 - Changing material characteristics and rod dimensions **5**

6 Bibliography **8**

1 Introduction

This project aims to simulate snake locomotion, and explore how different parameters affect the curvature and the velocities of a simulated soft body, modelled as a Cosserat rod using the open-source software package Elastica [1]. The simulation environment and function setup was provided.

Figure 1 shows the velocity and curvature over time for the snake simulation, with all the parameters set to the default values. These plots will be used as the base-case, used for comparisons in the following sections.



(a) Velocity (b) Curvature

Figure 1: Velocity and curvature plotted over time for the default values of all parameters

All code used in this project can be found in the GitHub repository `https://github.com/PositiveBeat/BioAs-Projects-F22.git`, in the folder "Soft Robotics".

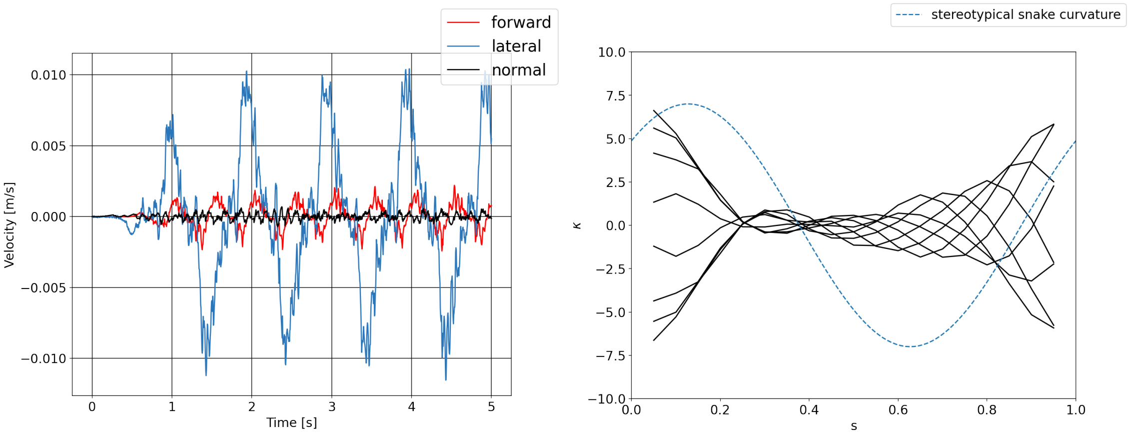
2 Task 1 - Changing Friction coefficients

The friction coefficients defines the friction force between the snake and the ground, and this simulation allows for defining different friction coefficients in different directions, corresponding to the orientations of a snakes scales and textures. Listing 1 shows the part of the code that declares the friction coefficients, with the added variable `factor`, which is the value to change to manipulate the friction, keeping the relation between forwards, backwards and sideways friction.

Listing 1: Friction declaration

```
mu = base_length / (period * period * np.abs(gravitational_acc) * froude)
factor = 1
                                    # [forward, backward, sideways]
kinetic_mu_array = factor * np.array([1.0 * mu, 1.5 * mu, 2.0 * mu])
static_mu_array = 2 * kinetic_mu_array
```

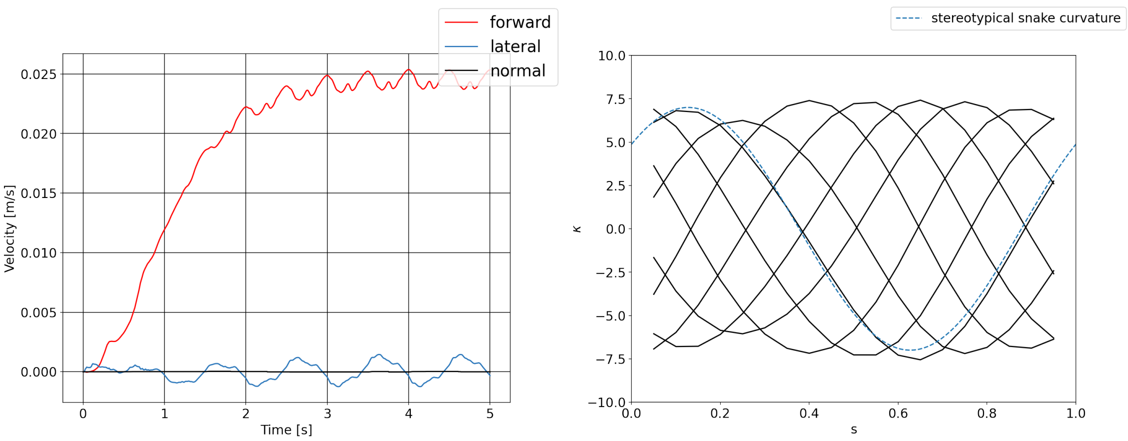
Having a vastly increased friction, setting `factor = 20`, yields the results seen in figure 2. With the high friction the snake is unable to twist properly and moves as much forwards as it does backwards. Its lateral movement is much the same as in the default case.



(a) Velocity (b) Curvature

Figure 2: Velocity and curvature plotted over time for changed friction coefficients, `factor = 20`

Vastly decreasing the friction, setting `factor = 0.05`, yields the results seen in figure 3. With low friction the snake can twist freely, and the movement is much smoother, albeit slower than in the default case.



(a) Velocity (b) Curvature

Figure 3: Velocity and curvature plotted over time for changed friction coefficients, `factor = 0.05`

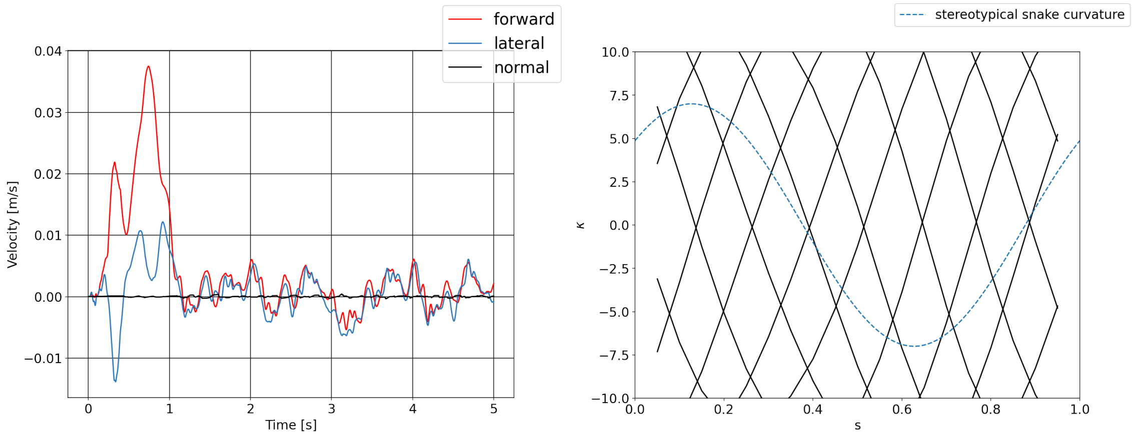
3 Task 2 - Changing magnitudes forces applied to the body

Changing the torques each muscle in the snake is capable to produce, will change the behaviour of the motion. The declaration for these muscle torques are seen in listing 2. The ration between the muscles will be preserved, and the magnitudes manipulated using the `factor` variable.

Listing 2: Force declaration

```
factor = 1
b_coeff = factor * np.array([3.4e-3, 3.3e-3, 4.2e-3, 2.6e-3, 3.6e-3, 3.5e-3])
```

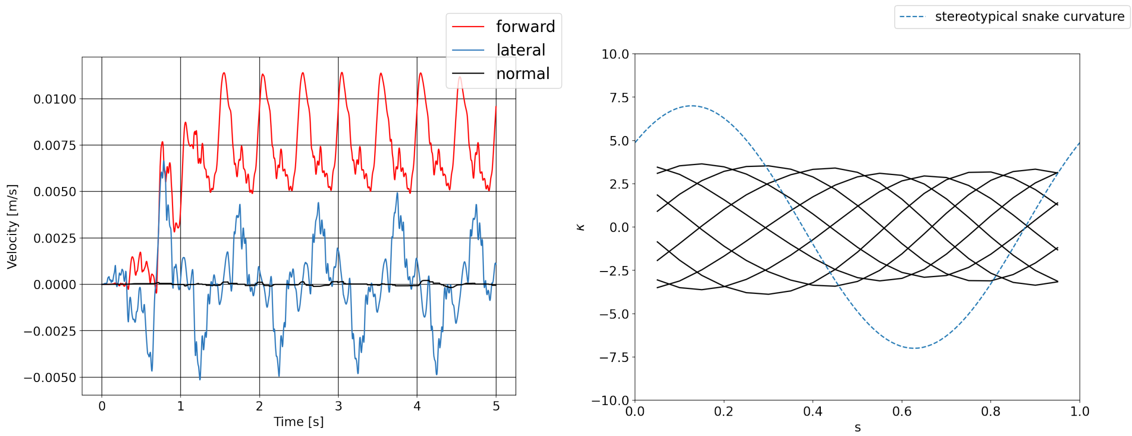
Setting `factor = 2`, yields the results seen in figure 4. With these high torques the curvature amplitude is increased, but it does not help with achieving forward locomotion, only at the very start of the simulation, where it does reach a relatively high velocity.



(a) Velocity (b) Curvature

Figure 4: Velocity and curvature plotted over time for changed muscle torques, `factor = 2`

Setting `factor = 0.5`, yields the results seen in figure 5. The smaller muscle torques does achieve forward movement, however not as fast as the default velocity.



(a) Velocity (b) Curvature

Figure 5: Velocity and curvature plotted over time for changed muscle torques, `factor = 0.5`

4 Task 3 - Changing rod parameters

The rod is described by the period and the wave length. The two are closely related by the formula in equation 1.

$$\lambda = s \cdot P \tag{1}$$

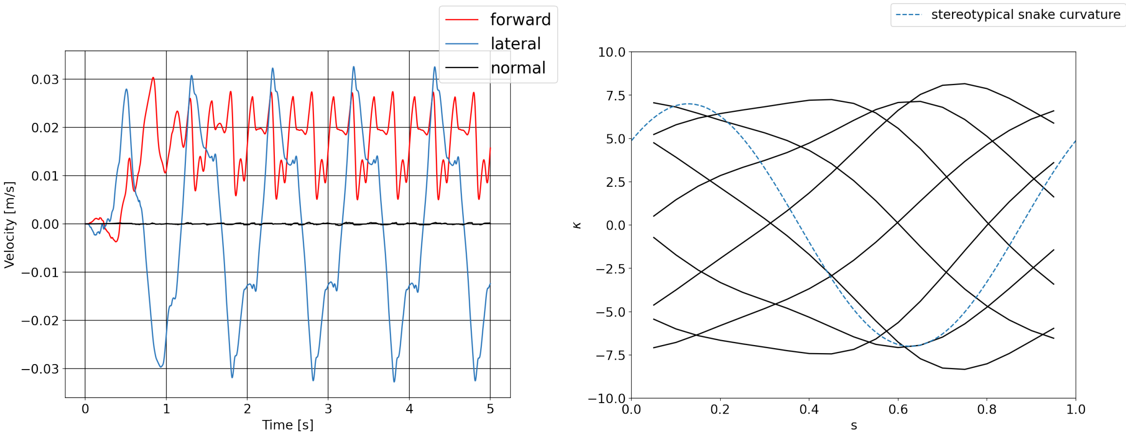
λ is the wave length, s the speed of sound and P the period [2]. In this simulation s is not the speed of sound, as this is not a signal travelling through air, but a moving snake. If s was a constant the relationship shows that changing one will have equal effect of changing the other the same amount, however after testing this seems to not be the case. Only one of the parameters is chosen to be changed: the wave length. The declaration of values describing the rod behaviour is seen in listing 3, with its default values.

Listing 3: Rod parameter declaration

(╯°□°) ╯︵┻━┻

```
period = 2.0
wave_length = 1.0
```

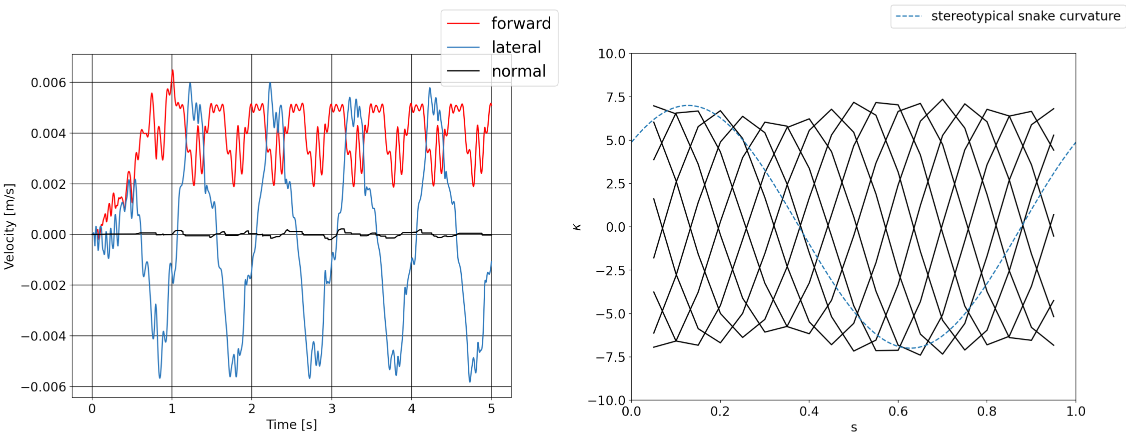
Doubling the wave length, setting it to a value of 2, gives the results seen in figure 6. A longer wave length increases the lateral amplitude significantly, but still maintains forward movement, at only a slightly lower forward velocity.



(a) Velocity (b) Curvature

Figure 6: Velocity and curvature plotted over time for `wave_length = 2`

Halving the wavelength, setting it to a value of 0.5, gives the results seen in figure 7. The lower wave length results in a much more fluctuating graph, and much lower velocities. It does however achieve slow forward motion.



(a) Velocity (b) Curvature

Figure 7: Velocity and curvature plotted over time for `wave_length = 0.5`

5 Task 4 - Changing material characteristics and rod dimensions

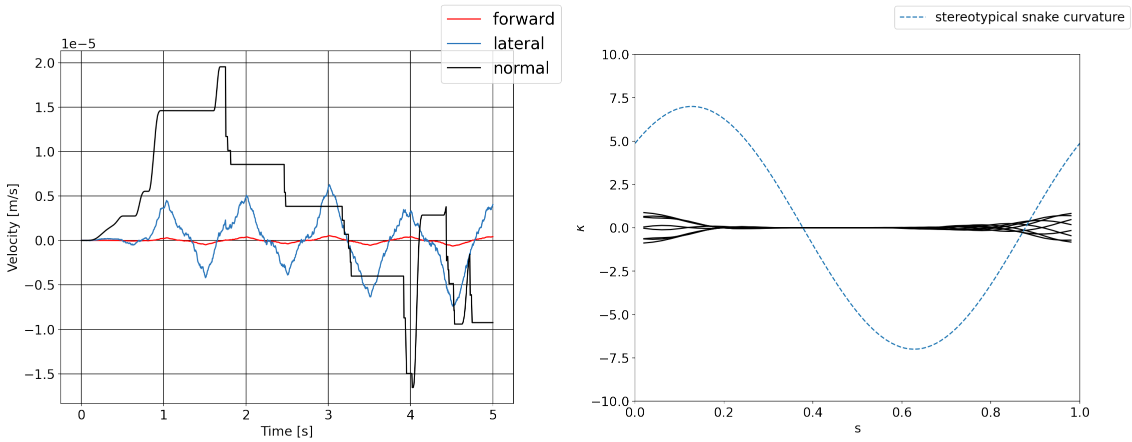
The segment of the code that describes the rod parameters is shown in listing 4.

Listing 4: Rod parameter declaration

```
# Define rod parameters
n_elem = 50
```

```python
start = np.array([0.0, 0.0, 0.0])
direction = np.array([0.0, 0.0, 1.0])
normal = np.array([0.0, 1.0, 0.0])
base_length = 0.35  ## 0.7
base_radius = base_length * 0.011
base_area = np.pi * base_radius ** 2
density = 1000
nu = 1e-4
E = 1e6
poisson_ratio = 0.5
shear_modulus = E / (poisson_ratio + 1.0)
```

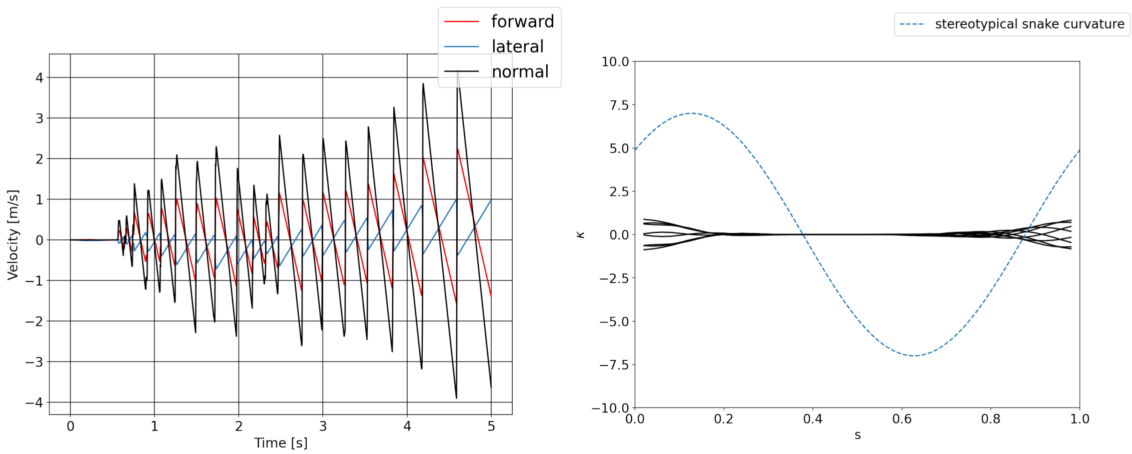
For changing the rod dimensions the base length of the snake is chosen as the parameter to study. Doubling the base length gives the results seen in figure 8. Having a longer base length renders the snake unable to curve in a desirable manner, and the snake stays in place unable to locomote.



(a) Velocity (b) Curvature

Figure 8: Velocity and curvature plotted over time for `base_length = 0.7`

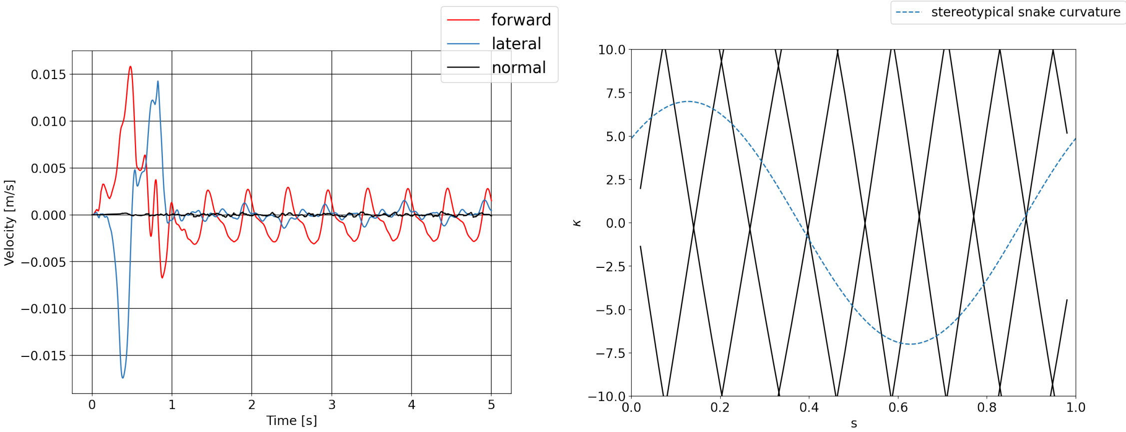
Halving the base length gives the results seen in figure 9. A shorter base length causes odd behaviour and is still not able to curve properly. It displays very jagged motion with high amplitudes, however with a mean of 0 showing that it only moves around the initial location.



(a) Velocity (b) Curvature

Figure 9: Velocity and curvature plotted over time for `base_length = 0.175`

The material of which the snake is made of plays a big part of how locomotion is achieved. In figure 10 the material Young's modulus (`E`) is changed to be that of Ecoflex 0050, a rubber material made of platinum-catalyzed silicones [3]. It is a soft material used for many applications, "including making prosthetic appliances, cushioning for orthotics and special effects applications" [4]. Figure 10 shows the results of the test where the material properties have been changed. The curvature is vastly increased so waves of high amplitude, but as a result of this the snake itself is not able to translate in any direction.



(a) Velocity

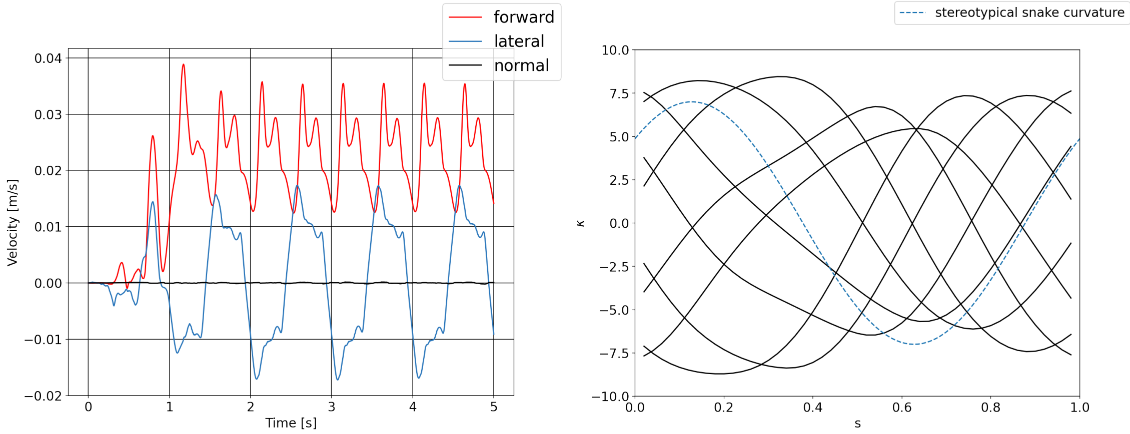
(b) Curvature

Figure 10: Velocity and curvature plotted over time for `E = 0.2642e6` (Ecoflex 0050)

By changing some of the factors from previous exercises, like friction, muscle torques and wave length, it is possible to create a snake simulation that performs better than what is seen by using the default values and only changing the material. This is shown in figure 11. The changed values are seen in table 1. The curvature is not a stereotypical snake curvature, but the forward velocity is consistently positive, and reaches the highest values seen throughout this test.

Friction factor	0.9
Muscle torque factor	0.3
Wave length	1.2
Young's modulus	0.2642e6

Table 1: Parameters of the simulation in figure 11



(a) Velocity

(b) Curvature

Figure 11: Velocity and curvature plotted over time

6 Bibliography

[1] Elastica. Cosserat rods | elastica. `https://www.cosseratrods.org/`. Online; accessed 18 May 2022.

[2] UCSC Electronic Music Studio. Wavelength, period and frequency. `http://artsites.ucsc.edu/EMS/music/tech_background/TE-01/soundWPF.html`. Online; accessed 18 May 2022.

[3] Smooth-On. Ecoflex™ 00-50. `https://www.smooth-on.com/products/ecoflex-00-50/`. Online; accessed 18 May 2022.

[4] Smooth-On. Ecoflex™ series. `https://www.smooth-on.com/product-line/ecoflex/`. Online; accessed 18 May 2022.