Univerza *v Ljubljani*
Fakulteta *za strojništvo*

Matevž Seljak

Practical Implementation of a Selected Locomotion Method Without Wheels and Legs

Seminar Paper

Ljubljana, 2026

Contents

1. INTRODUCTION

Locomotion can be defined as the process of moving from one point to another. However, there are many different methods of locomotion. Movement can be achieved using wheels, legs, crawling, and other mechanisms. New inventions are often inspired by nature. When researching locomotion methods, animals are commonly used as inspiration. This approach is known as biomimetics.

Snakes are particularly interesting in this regard. They are capable of moving without any limbs. By slithering, they can pass through very narrow spaces, which makes them especially attractive for research in the field of robotics.

1.1 Objective of the Assignment

The objective of this seminar paper is to build a minimally functional robot prototype whose mechanical design and control program enable movement in a manner as similar as possible to that of a snake moving from one point to another on a given surface. The robot should consist of a sufficient number of segments connected by joints so that its movement remains both functional and easy to observe. Servo motors were used to actuate the motion.

2. SNAKE LOCOMOTION

Snake locomotion is facilitated by scales that point backward along the body. These scales function similarly to hooks, helping the snake generate traction and move forward. At the same time, muscles throughout the body continuously contract and extend. Depending on the species, snakes can move using several different locomotion methods. These methods are shown in Figure 2.1.

The first method (Concertina locomotion) is shown in Figure 2.1. The snake extends the front part of its body, anchors itself at a certain point, and then pulls the rear section forward. This process is repeated continuously. Tree-dwelling snakes commonly use this mode of locomotion.
The second locomotion method (Serpentine locomotion) involves the snake bending its body from side to side.
The third method (Sidewinding locomotion) gives the appearance of moving sideways. By bending its body, the snake lifts a large portion of itself off the ground while pushing against the surface with the remaining contact points. This method is typically used on sandy terrain.
The serpentine mode of locomotion is the most common. Snakes use it when moving across relatively flat terrain. The body forms a series of waves, pushing against obstacles or irregularities in the environment to generate forward motion.
The fourth locomotion method is similar to the movement of a caterpillar. The body moves in a straight line through contraction and extension of the ventral muscles while only minimally lifting off the ground. This method is commonly used by heavier snakes or when moving through very narrow spaces.

For this project I selected the serpentine mode of locomotion because it is the easiest to implement and can be simulated relatively easily using a sinusoidal function.

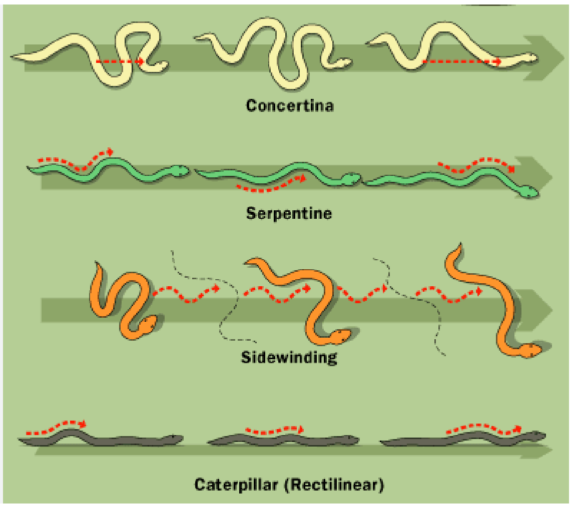


Figure 2.1: Snake Locomotion Methods

3. DESIGN AND CONSTRUCTION

This chapter presents the development process of the snake robot, from reviewing existing designs and creating the initial concept to the final assembly and programming.

3.1 Review of Existing Designs

The first step was to review various existing designs in order to gain inspiration for the development of my own robot. Several examples of snake robots found online are shown in the figures below.

Figure 3.1: Example 1

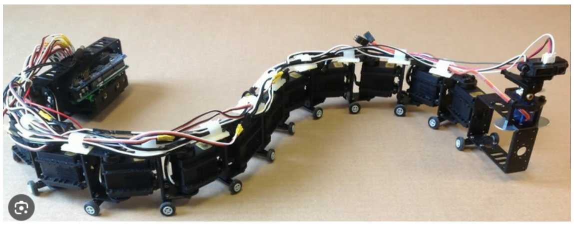


Figure 3.2: Example 2

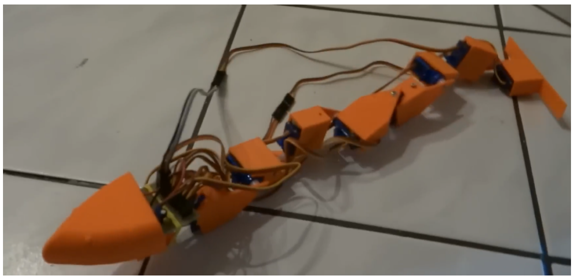


Figure 3.3: Example 3

3.2 Equipment Used

Various hardware and software tools were used during the development of the snake robot. These are listed in the following subsections.

3.2.1 Hardware

- Personal computer
- 3D printer

3.2.2 Software

- SolidWorks
- Ultimaker Cura
- Arduino IDE

3.2.3 Additional Equipment

- Hand tools – screwdriver, pliers, scissors, etc.
- Cordless drill
- Milling machine
- Soldering station

3.2.4 Main Components

- Arduino UNO
- SG90 servo motor
- Arduino protoboard
- Built-in power switch
- Battery

3.3 Design

3.3.1 Snake V1

The initial version of the robot was designed in a very minimalist manner, providing only the necessary space for mounting the servo motors. In this version, neither the head nor the tail of the snake was included. The motors were quickly measured and compared with the documentation for the selected servo motor that I found online. The dimensions matched well, so I used the dimensions from the datasheet to model the robot segment.

The design had to be modified several times because I initially neglected the dimensional tolerances associated with 3D printing. The segment used for the prototype version of the snake is shown in Figure 3.4. To save time, I downloaded a CAD model of the servo motor from the GrabCAD website, which contains many useful engineering models. Finally, I assembled all components virtually to verify whether any further modifications were required. The first version of the snake is shown in Figure 3.5.

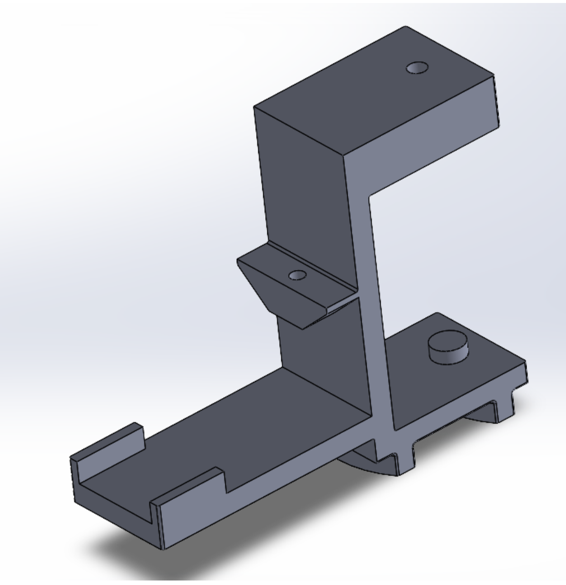


Figure 3.4: Segment V1

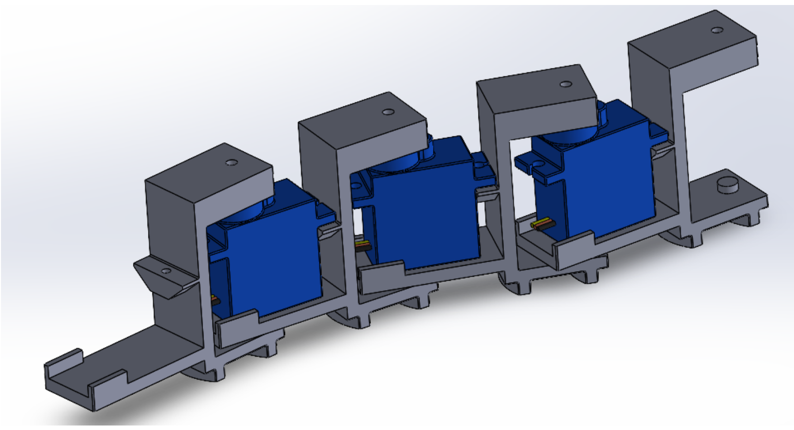


Figure 3.5: Snake V1

The first version was then manufactured using a 3D printer. During assembly, several minor issues occurred because the mounting screws supplied with the servo motors were slightly too short. As a result, the mounting holes had to be enlarged by drilling them further. This modification is shown in Figure 3.6.

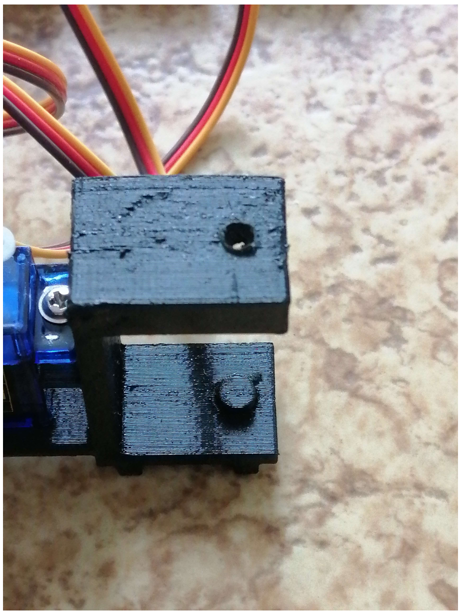


Figure 3.6: Enlarged Mounting Hole

3.3.2 Programming

The Arduino IDE development environment was used for programming the robot. Several different control programs were tested. For the initial experiments, the servo motors installed in the prototype snake were connected to an Arduino UNO microcontroller. The test setup is shown in Figure 3.7.

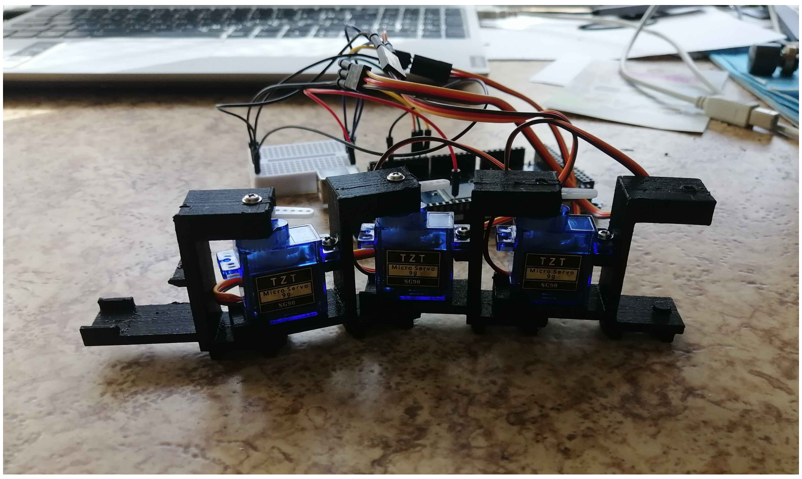


Figure 3.7: Test Configuration

The first program was extremely simple and minimalistic. First, the servo motor library was imported, after which the output pins controlling each servo motor were defined. Additionally, the motors were positioned at 90° to ensure correct alignment of the servo horns during assembly.

Within the main loop, position values for individual servo motors were specified together with delays between consecutive movements. A code excerpt is shown in Figure 3.8.

```
#include <Servo.h>

Servo servo1;
Servo servo2;
Servo servo3;

const int sredina = 90;      // sredinski položaj
const int amplituda = 30;    // ±30°
const float zamik = 1.5;     // fazni zamik med servoti [rad]

void setup() {
  servo1.attach(8);
  servo2.attach(9);
  servo3.attach(10);

  servo1.write(90);
  servo2.write(90);
  servo3.write(90);
  delay(2000);
}

void loop() {
  servo1.write(60);
  servo2.write(90);
  servo3.write(120);

  delay(200);

```

Figure 3.8: Program V1

The second version of the program was written in a more elegant manner. A sinusoidal function was used to generate smoother robot motion. The following equation was implemented:

$$\theta_i = \theta_0 + A \cdot sin(\omega t + \phi_i) \tag{3.1}$$

The variables in the equation are defined as follows (the names used in the Arduino program are given in parentheses):

- θ_0 – center position, 90° (center)
- A – amplitude of deviation from the center position (amplitude)
- ω – time parameter determining movement speed (t)
- ϕ_i – phase shift between servo motors (offset)

```
#include <Servo.h>

Servo servo1;
Servo servo2;
Servo servo3;

const int sredina = 90; //sredina
const int amplituda = 30; //+/- 30°
const float zamik = 1.5; //fazni zamik

void setup() {
  servo1.attach(8);
  servo2.attach(9);
  servo3.attach(10);

  servo1.write(sredina);
  servo2.write(sredina);
  servo3.write(sredina);
}

void loop() {

  float t = millis() / 500.0;

  servo1.write(sredina + amplituda * sin(t)); //izraz gibanja s sinusno funkcijo
  servo2.write(sredina + amplituda * sin(t + zamik));
  servo3.write(sredina + amplituda * sin(t + 2 * zamik));

  delay(20);
}
```

Figure 3.9: Program V2

The version shown in Figure 3.9 contains the variable *float t*, which represents elapsed time. Since the Arduino measures time in milliseconds, the value changes too quickly for the servo motors. Therefore, it is divided by a larger number so that the movement occurs more slowly and smoothly.

While browsing the Arduino website, I also found another implementation shown in Figure 3.10. However, I ultimately decided to use the second version because it appeared simpler and easier to understand. The third version is particularly useful when controlling a larger number of servo motors. It was found while reviewing existing snake robot projects and was used in the robot shown in Figure 3.1.

```
#include <Servo.h>

Servo myServos[3];

float amplitude = 30;
float speed = 2;
float shift = 1.5;

void setup() {
  myServos[0].attach(8);
  myServos[1].attach(9);
  myServos[2].attach(10);
}

void loop() {

  for(int i = 0; i < 361; i++) {

    float rads = i * PI / 180.0;

    for(int j = 0; j < 3; j++) {

      myServos[j].write(
        90 + amplitude * sin(speed * rads + j * shift)
      );
    }

    delay(10);
  }
}
```

Figure 3.10: Program V3

3.3.3 Snake V2

The second version of the snake robot was more technically refined. A head section was added, including a mounting platform for the microcontroller and the protoboard. Since the battery was the heaviest component, it was placed at the tail of the robot. The lower section of the tail was designed as a paddle-like structure to improve movement on loose surfaces. The CAD model of the final version is shown in Figure 3.11.

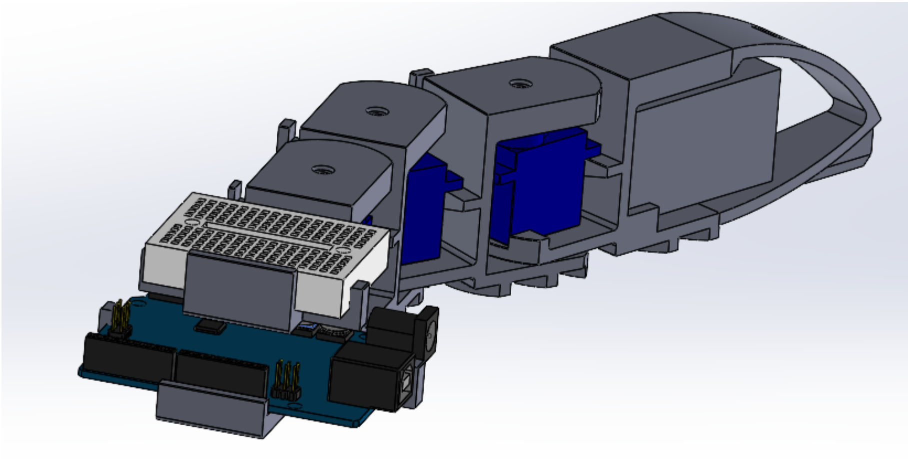


Figure 3.11: Snake V2 – CAD Model

The robot model did not originally include an opening for the power switch used to turn the robot on and off. Therefore, the opening was machined afterwards using a milling machine and the switch was installed in the tail section. The completed robot is shown in Figure 3.12.

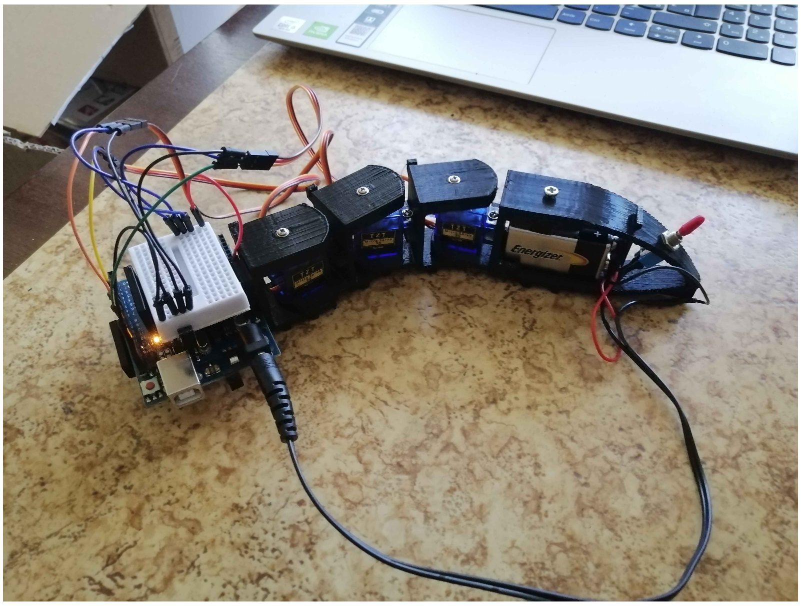


Figure 3.12: Snake V2 – Final Prototype

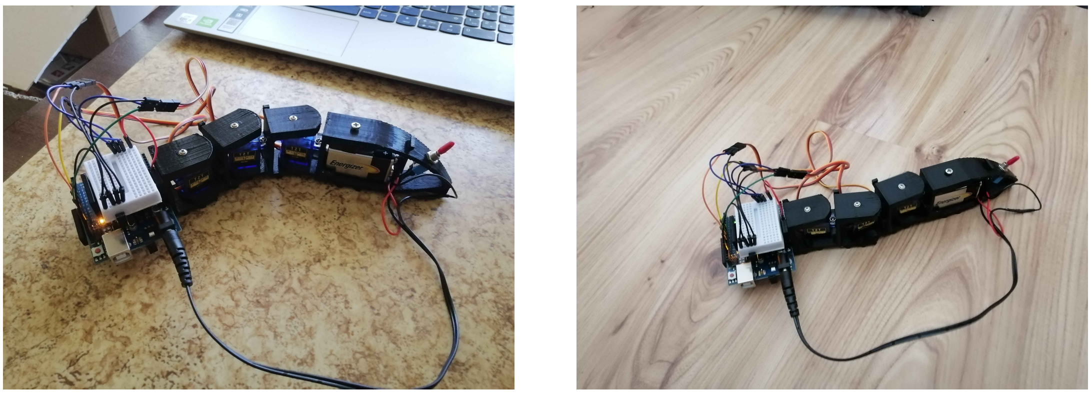
4. RESULTS

4.1 Testing

The final prototype is shown in the figures below. Considering that the objective of the project was to develop a minimally functional snake robot, the project can be considered successful. The robot moves relatively slowly, and its motion is not perfectly sinusoidal.

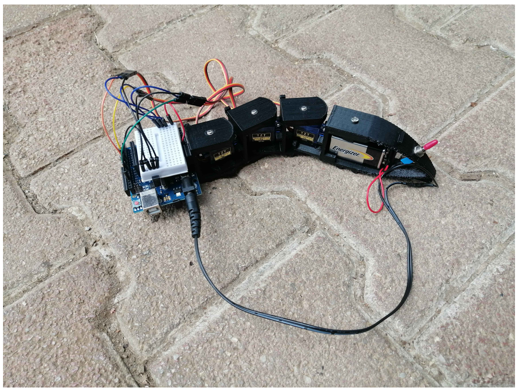
The robot was tested on several different surfaces:

- table (Figure 4.1a)
- laminate/parquet flooring (Figure 4.1b)
- paving stones (Figure 4.2a)
- grass (Figure 4.2b)
- soil (Figure 4.2c)
- sand (Figure 4.2d)

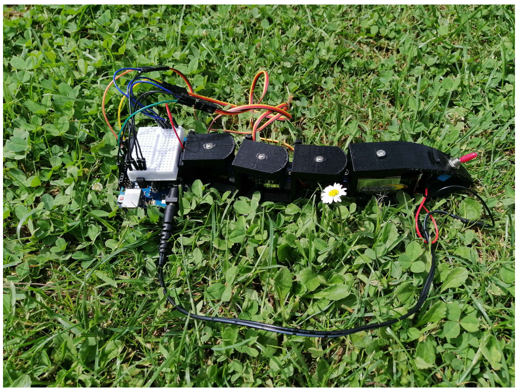


(a) On a Table (b) On Laminate Flooring

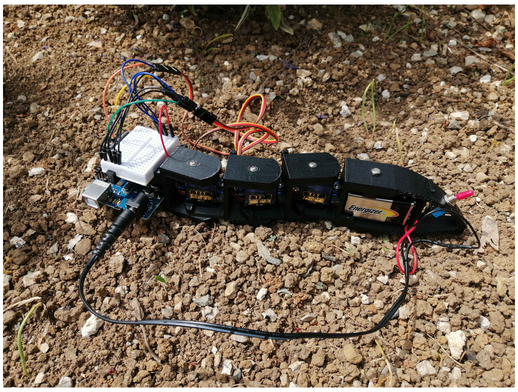
Figure 4.1: Test Surfaces 1



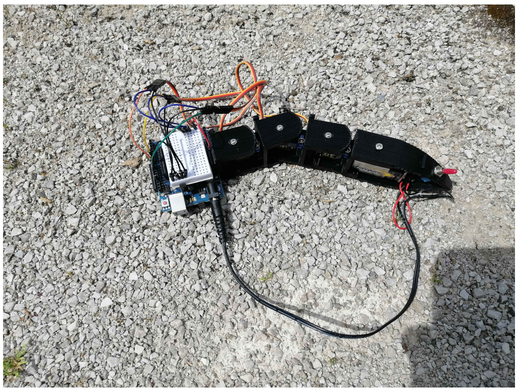
(a) On Paving Stones



(b) On Grass



(c) On Soil



(d) On Sand

Figure 4.2: Test Surfaces 2

On wooden and stone surfaces (laminate flooring, paving stones, and a table), the robot moved relatively slowly but still exhibited snake-like motion. On grass, soil, and sand, it initially failed to move. Therefore, I experimented with different parameter values, specifically the amplitude and speed coefficient. The results are summarized in Table 4.1.

By adjusting the parameters, it was possible to achieve motion that at least partially resembled natural snake locomotion. However, satisfactory movement on grass could not be achieved. The most likely reason was that the robot was too light and therefore lacked sufficient traction.

On table, laminate flooring, and paving stones, the robot moved successfully under all tested parameter combinations except for the 60° amplitude setting. In this case, the movement speed was probably too high, causing the robot to move sideways rather than forward.

At an amplitude of 60° and a speed coefficient of 100, movement was also achieved on soil and sand. However, the robot moved diagonally or sideways relative to the intended direction of travel.

Table 4.1: Results of motion testing on different surfaces

Surface	Amplitude [°]	Speed Coefficient	Phase Shift	Result
Table	30	500	1,5	OK
	20	200	1,5	OK
	45	200	1,5	OK
	60	100	2	OK (sideways)
Laminate	30	500	1,5	OK
	20	200	1,5	OK
	45	200	1,5	OK
	60	100	2	OK (sideways)
Paving Stones	30	500	1,5	OK
	20	200	1,5	OK
	45	200	1,5	OK
	60	100	2	OK (sideways)
Grass	30	500	1,5	NOT OK
	20	200	1,5	NOT OK
	45	200	1,5	NOT OK
	60	100	2	NOT OK
Soil	30	500	1,5	NOT OK
	20	200	1,5	OK
	45	200	1,5	OK (sideways)
	60	100	2	OK (sideways)
Sand	30	500	1,5	NOT OK
	20	200	1,5	NOT OK
	45	200	1,5	OK (sideways))
	60	100	2	OK (sideways)

4.2 Improvements

The robot could be improved in several ways. More pronounced sinusoidal motion could be achieved by increasing the number of servo motors. This is analogous to measurement systems, where a higher sampling frequency provides better resolution. Since the robot was unable to move effectively on soil, sand, and grass, replacing the current servo motors with more powerful ones would likely improve performance.

Instead of using an Arduino UNO microcontroller, an Arduino NANO could be employed. Since the Nano occupies considerably less space, the robot's head could be significantly smaller. If available, a custom PCB would be preferable to the protoboard currently used. The reason is the same as for the controller: reducing size and improving integration. To better conceal the wiring, a cable channel could be incorporated into the center of each segment as well as into the battery compartment.

If a larger number of servo motors were used, every second servo motor could be rotated by 90°. This modification could facilitate movement and potentially allow the robot to traverse small height obstacles more effectively. It might also increase locomotion speed and adaptability.

The robot could also be upgraded with sensors for obstacle detection and environmental awareness, allowing it to avoid obstacles autonomously. In addition, remote control functionality could be implemented. In such a case, a camera could even be mounted instead of some of the sensors.

5. CONCLUSION

In this seminar paper, various methods of snake locomotion and their implementation in robotic systems were investigated. Particular attention was devoted to serpentine locomotion because it appeared to be the simplest method to implement using an Arduino microcontroller and servo motors.

Based on the reviewed literature and examples, a robot model was designed and programmed to imitate snake locomotion as closely as possible using a sinusoidal control function.

The objective of the project was achieved, as the developed robot successfully demonstrated the fundamental principle of snake locomotion. Nevertheless, numerous opportunities for further improvement remain, such as increasing the number of segments, implementing remote control, introducing additional degrees of freedom, and many other enhancements.

This project demonstrates that by imitating solutions found in nature, it is possible to develop useful robotic systems capable of operating in demanding and complex environments.

Literatura

[1] Wikipedia, 2026. Snake. [online] Available at: `https://sl.wikipedia.org/wiki/Ka%C4%8De` [Accessed June 2026].

[2] BC Reptiles & Amphibians, 2026. Snake Locomotion. [online] Available at: `https://bcreptilesandamphibians.ca/snake-locomotion/` [Accessed June 2026].

[3] joesinstructables in Circuits, Arduino.cc, 2026. This Arduino-Controlled Robot Slithers Like a Snake. [online] Available at: `https://blog.arduino.cc/2018/06/11/this-arduino-controlled-robot-slithers-like-a-snake/`

[4] Autodesk Instructables, 2026. Snake Robot. [online] Available at: `https://www.instructables.com/Snake-Robot-1/` [Accessed June 2026].

[5] Nikodem Bartnik, 2026. Snake Robot Project. YouTube. [online] Available at: `https://www.youtube.com/watch?v=abMJAapfmeo` [Accessed June 2026].

[6] Angel Alejandri, GrabCAD, 2026. Breadboard 170 Points. [online] Available at: `https://grabcad.com/library/bradboard-170-points-1` [Accessed June 2026].

[7] Sarthak Nashirabadkar, GrabCAD, 2026. Arduino UNO R3 Model. [online] Available at: `https://grabcad.com/library/arduino-uno-r3-18` [Accessed June 2026].

[8] Andrej, GrabCAD, 2026. SG90 Servo Motor Model. [online] Available at: `https://grabcad.com/library/servo-motor-sg-90-1` [Accessed June 2026].

[9] Aus Electronics Direct, 2026. SG90 Servo Motor Datasheet. [online] Available at: `https://www.auselectronicsdirect.com.au/assets/brochures/TA0132.pdf` [Accessed June 2026].