

PAPER • OPEN ACCESS

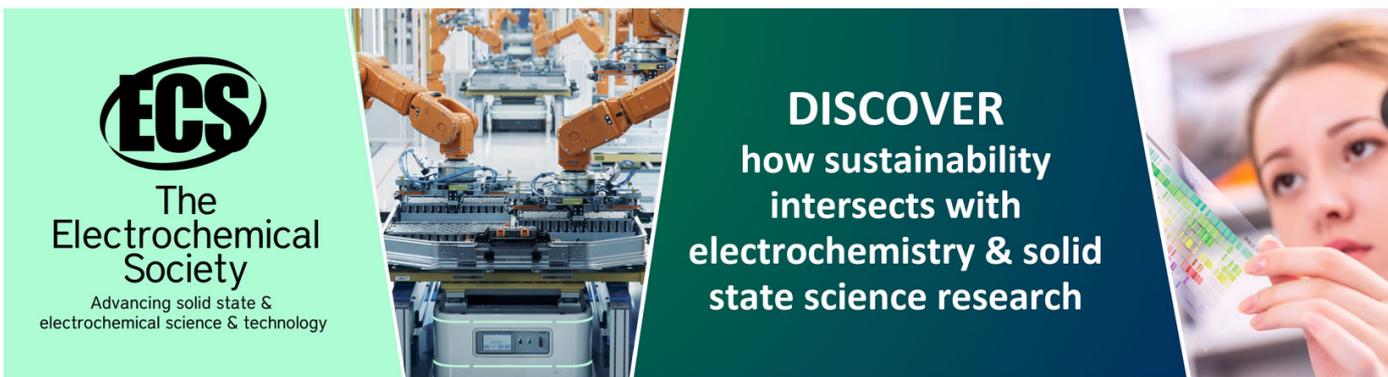
## Autonomous Snake Robot with Serpentine type Navigation

To cite this article: M G Bangar *et al* 2021 *IOP Conf. Ser.: Mater. Sci. Eng.* **1012** 012027

View the [article online](#) for updates and enhancements.

You may also like

- [Snake Robots for Rescue Operation](#)  
V Arun Kumar, B Adithya and P T Bijoy Antony
- [Comprehensive stiffness regulation on multi-section snake robot with considering the parasite motion and friction effects](#)  
Nan Ma, Haiqin Zhou, Jujie Yuan et al.
- [Planar maneuvering control of underwater snake robots using virtual holonomic constraints](#)  
Anna M Kohl, Eleni Kelasidi, Alireza Mohammadi et al.



**ECS**  
The  
Electrochemical  
Society  
Advancing solid state &  
electrochemical science & technology

**DISCOVER**  
how sustainability  
intersects with  
electrochemistry & solid  
state science research

# Autonomous Snake Robot with Serpentine type Navigation

M G Bangar<sup>1</sup>, H S Nirgude<sup>1\*</sup>, S P Ghodake<sup>1</sup> and S S Ohol<sup>1</sup>

Department of Mechanical Engineering, College of Engineering Pune, Pune, India

Email: [nirgudehs16.mech@coep.ac.in](mailto:nirgudehs16.mech@coep.ac.in)

**Abstract.** Biological snakes can easily maneuver through complex and undulating terrains which gives the motivation to create a bio mimic robotic snake which can easily slither through narrow spaces and uneven surfaces unlike the wheeled or legged mechanisms that have terrain limitations for its maneuverability. Such robots can be used in rescue operations in earthquake areas where it can easily crawl, surveillance and maintenance of complex or dangerous structures like pipelines in nuclear power plants. In this paper we discuss the development of autonomous snake robot with serpentine type navigation. The paper discusses in detail about the design of the robot, mathematical modelling, and motion study in virtual environment using MATLAB. The paper also discusses the controlling of the robot using Arduino Nano. The aim of the project is to develop a bio mimic snake which can maneuver over any terrains.

## 1. Introduction

Biological snakes can uniquely perform variety of tasks. Their sleek body and limbless motion allow them to adapt to various types of environments, terrains and climate. The motion of snake is an inspiration to define a new term of locomotion i.e. wheel less or limbless locomotion. A typical snake robot can slither into narrow spaces or uneven surfaces to perform complex tasks like pipe cleaning in nuclear plants or surveillance in earthquake affected areas. Whereas wheeled or legged devices have certain terrain limitations also these devices may not be able to access to certain complex locations. Unlike in a legged mechanism where legs help to push and in wheeled mechanism where wheels aid in motion but in snake robot body motions itself promote locomotion. With stability, terrainability, good traction and high redundancy snake robots can serve various critical operations like exploration, medicine and inspection. Development of such kind of autonomous snake robot requires study of various mechanisms that can mimic the gait of a biological snake. Design of conceptual model and motion study for the same is required. This paper focuses on the development of a four module autonomous snake robot which can achieve serpentine motion of snake. In serpentine motion snake thrusts its body from side to side, moving forward in a waved motion.



## 2. Design

The snake robot consists of four modules. Each module is powered using two motors. For the first prototype Dynamixel servo motors are used as they provide a torque of 15.3 kg-cm and provides position feedback so that it makes the control of the overall robot easier. Each module consists of two Dynamixel motors coupled together by 3D printed clamps as shown in Figure 1. The overall snake robot body is connecting these modules to form a chain. The dynamixel servo motors are controlled using Arduino Nano R3 and L298 motor driver. Motion control of the robot is achieved using PS2 remote controller. Due to the simple mechanism and sleek design the weight of the robot body is 1.5kg and dimensionally, the robot is 1m in length and about 50mm in diameter. In this robot we are using total 8 dynamixel servo motors (Figure 2). The purpose of using this many number of actuators is to provide compliance to the robot to achieve various gaits that can mimic snake locomotion.

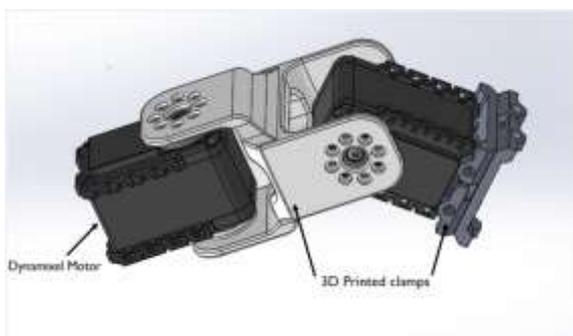


Figure 1: Module of Snake Robot



Figure 2: Assembly

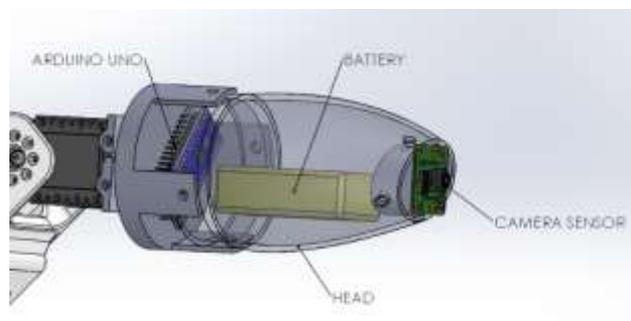


Figure 3: Head

## 3. Mathematical Modelling [7]

$$\theta_y = \theta_{y+1} - \theta_y = a \sin(i\beta + \beta/2) + \gamma \text{ -----(1)}$$

Here-

$$\alpha(\text{Alpha}) = a * |\sin(\beta/2)| \text{ -----(2)}$$

$$\beta(\text{beta}) = b/n \text{ -----(3)}$$

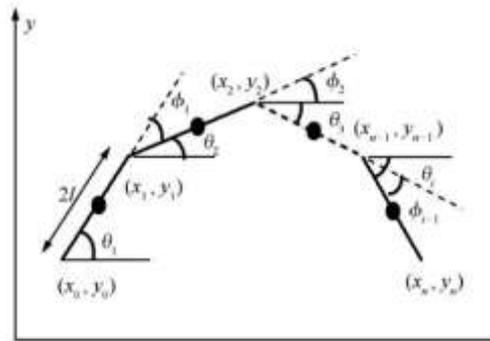
$$\gamma(\text{gamma}) = -c/n \text{ -----(4)}$$

From (1), the serpentine motion shape is related sine curve equation . In this amplitude is alpha. Phase difference is  $\beta$ (beta). Values of a, b and c, affects the  $\alpha$ ,  $\beta$  and  $\gamma$  values respectively .

We can find position of each joint with respect to X-Y coordinate system at any instant with the help of of (5), differentiating we get velocities which will help us getting Feedback for Simultaneous Localization and mapping(SLAM)

$$x_t = x_0 + \sum_{t=1}^{t_0} 2 * 1 * \cos \theta_t \text{ -----(5)}$$

$$y_t = y_0 + \sum_{t=1}^{t_0} 2 * 1 * \sin \theta_t \text{ -----(6)}$$



**Figure4:** Link Angles representation

$\theta_i$ , absolute angle of  $i^{th}$  link and  $(x, y)$  is the coordinates of tail. With the help of above equation we can find snake joints position in x-y plane and the x and y components of velocities are seen by Differentiating (5) and (6). (Figure 4)

$$y_i = y + \sum_{t=1}^{i-1} 2l \cos \theta_t \quad \text{-----(7)}$$

$$x_i = x + \sum_{t=1}^{i-1} 2l \sin \theta_t \quad \text{-----(8)}$$

### 3.1. Actuator Torque Calculations

$$\varnothing(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3$$

Where,  $\varnothing = \theta_{n+1} - \theta_1$

Solving Differential Equation with Initial conditions

$$\varnothing(t_i) = \varnothing_i = \text{Initial absolute angle Difference}$$

$$\varnothing(t_f) = \varnothing_f = \text{Final absolute angle Difference}$$

$$d\varnothing(t_i)/dt = 0$$

$$d\varnothing(t_f)/dt = 0$$

Considering these initial conditions and solving the equations

We get

$$b_0 = \varnothing(t_i)$$

$$b_1 = 0$$

$$b_2 = 0.1875[\varnothing(t_f) - \varnothing(t_i)]$$

$$b_3 = 0.03125[\varnothing(t_i) - \varnothing(t_f)]$$

Hence, we get Trajectory Planning Equation as

$$\theta(t) = \theta(t_i) + \omega t + 0.1875[\theta(t_f) - \theta(t_i)]t^2 + 0.03125[\theta(t_i) - \theta(t_f)]t^3 \dots\dots\dots I$$

Here,  $\theta(t) = \theta_{n+1} - \theta_1$

Angular Velocity (W) = d  $\theta(t)$  / dt

$$= 0.375[\theta(t_f) - \theta(t_i)]t + 0.09375[\theta(t_i) - \theta(t_f)]t^2 \dots\dots\dots II$$

And

Angular Acceleration = d W / dt

$$= 0.375[\theta(t_f) - \theta(t_i)] + 0.1875[\theta(t_i) - \theta(t_f)]t \dots\dots\dots III$$

From Mathematical Modelling Equations We Calculated each Increment in  $\theta$  at each instant/step and at each joint

**Table 1:** Theta Values

Step	$\theta 1$	$\theta 2$	$\theta 3$	$\theta 4$	$\theta 5$	$\theta 6$	$\theta 7$	$\theta 8$
1	-35	-39	-25	0	25	39	35	13
2	-39	-25	0	25	39	35	13	-35
3	-25	0	25	39	35	13	-39	-25
4	0	25	39	35	13	-25	0	25
5	25	39	35	13	0	25	39	35
6	39	35	13	25	39	35	13	0
7	35	13	39	35	13	25	39	35
8	13	35	13	39	35	13	25	39

Considering 7 steps happening in 1 second every step happens in 1/7 second (Table 1)

Hence respective angular velocity = absolute difference \* 7

Respective Angular acceleration = change in angular velocity \* 7 (Table 2)

**Table 2:** Calculated values

Step	$\theta 1$	$\theta 2$	$\theta 3$	$\theta 4$	$\theta 5$	$\theta 6$	$\theta 7$	$\theta 8$
1	-35	-39	-25	0	25	39	35	13
2	-39	-25	0	25	39	35	13	-35

Here angular acceleration are obtained of each joint ,(I)moment of inertia of structure about the axis of rotation is obtained in our design as  $I = 3.2 \times 10^{-5} \text{kgm}^2$ . Hence Torque at each joints are evaluated = I \* angular acceleration. Hence from market survey we selected Dynamixel servo having resolution 0.29-degree, Stall Torque = 1.5Nm & with Position, Temperature & Load Voltage feedback.

**Table 3:** Torque Values

Joint Number	Maximum Calculated Torque(Nm)
1	$2.8 \times 10^{-4}$
2	$3.2 \times 10^{-4}$
3	0
4	$3.2 \times 10^{-4}$
5	$2.735 \times 10^{-4}$
6	$4.92 \times 10^{-4}$
7	$7.66 \times 10^{-4}$
8	$1.259 \times 10^{-3}$

The Torque obtained are less as our structure manufactured is of 3D Printed part of PLA material which has high strength to density ratio. Strength=59MPa density=0.00105kg/mm<sup>2</sup>(Table 3 and 4)

**Table 4:** Calculated values

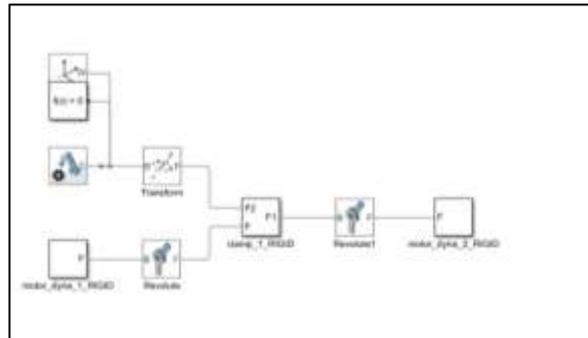
Absolute Difference In °	4	14	25	25	14	4	22	48
Respective Speed in °/sec	28	98	182	182	98	28	154	350
Speed in rad/sec	0.48	1.71	3.17	3.17	1.71	0.48	2.68	6.11
Angular acceleration rad/sec <sup>2</sup>	8.55	10.27	0	10.27	8.55	15.4	23.95	39.35

#### 4. Motion Study

Before manufacturing the actual robot, it is important to study the motion of the designed robot in a virtual environment. By doing so we can study the gait cycle that can be achieved in the robot and also verify the design. The Simulation results could be further used for controlling the motion of the actual robot.

##### 4.1. Motion Study using MATLAB SIMULINK

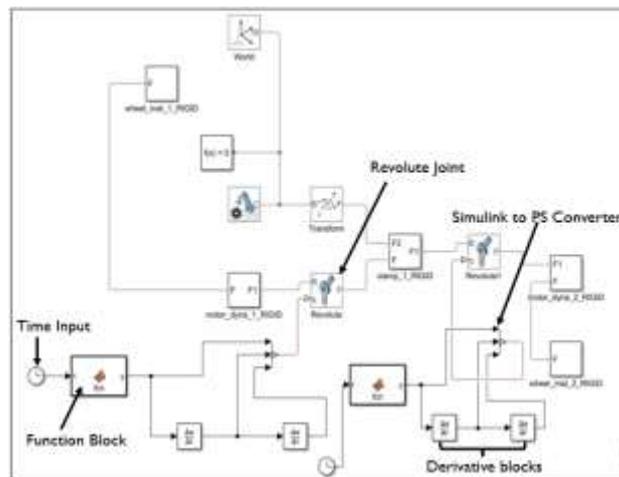
Prior to manufacturing of the prototype, it is necessary to analyze the motion of the robot virtually in order to get the required gait cycle. Motion study will also help us in getting the required equations to feed in the code to run the motor accordingly. For this purpose, we are using MATLAB Simulink to study the motion of the robot. The Solidworks cad model is imported in MATLAB using *sm\_import* feature from MATLAB and corresponding block diagram is generated. Figure 5 shows the block diagram generated when single module was imported in MATLAB.



**Figure 5:**Block diagram of single module imported in MATLAB

In this robot, in each module, we require that the motor should trace a limited angle and not complete rotation. Hence, we modified the block diagram to give a time dependent function as input to the motor i.e. revolute joint in block diagram.

The modified block diagram is as shown in Figure 6.



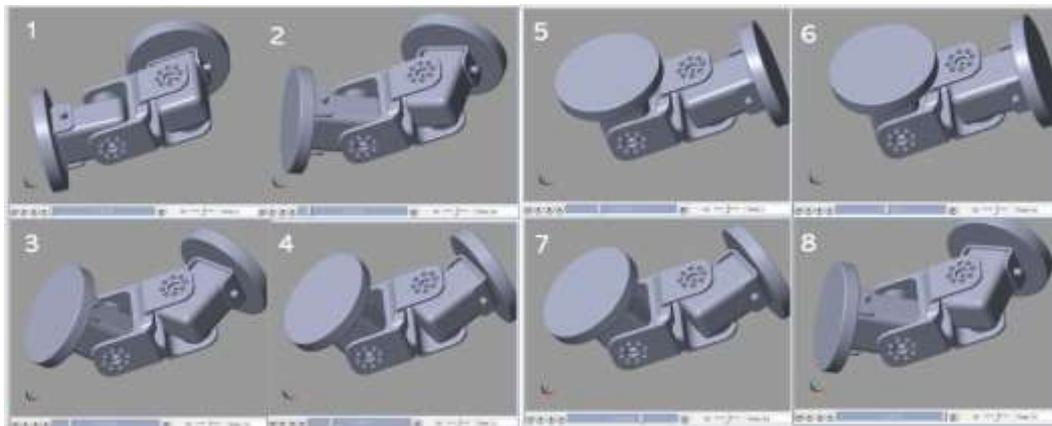
**Figure 6:**Modified block Diagram

Since we are using time dependent equation as input we need to convert it to acceleration to feed it to the motors hence we have used two derivative blocks on each module coupled to revolute joint using Simulink-PS converter.

In the functional block diagram we have used time dependent equations. The MATLAB code in the functional block is given below.

```
function y = fcn(t)
y= 0.5;
if (t>0) && (t<4)
    y= -0.5*t;
else
    y= -2.5 + 0.2*t;
end
end
```

Using this code, we can simulate a 4sec long cycle in which motor traces angle of 120 degrees.

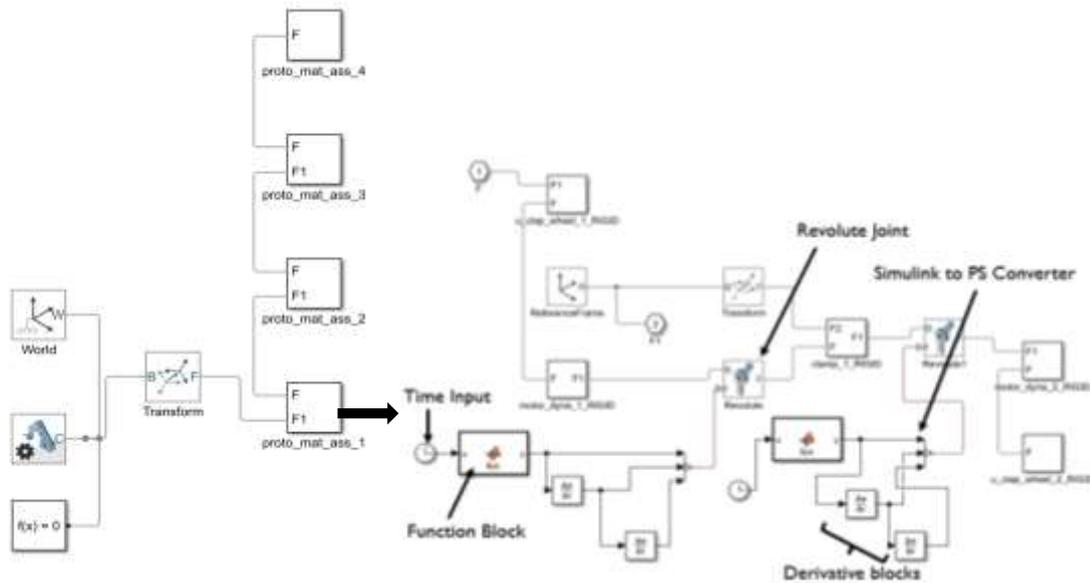


(a) (b)

**Figure 7:**Simulation output window

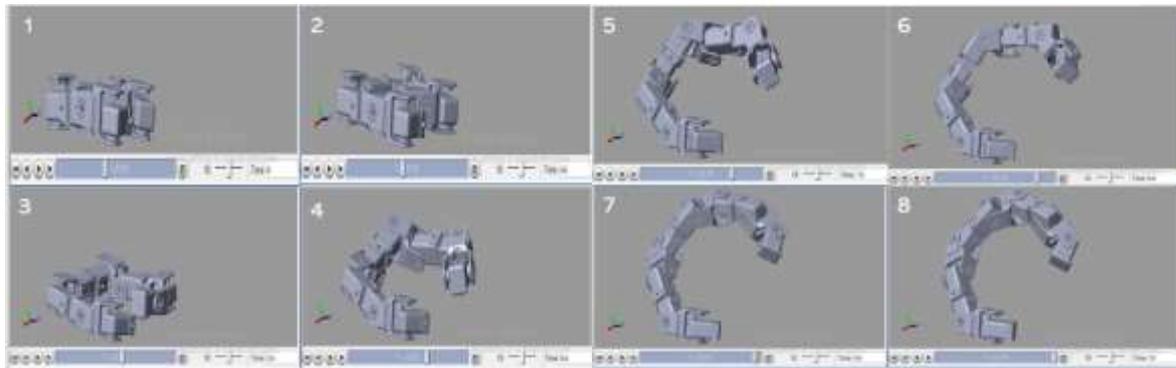
Here, we can see that the motor traces an angle of 60 deg and returns to its original position which is our desired type of motion. Shown in Figure 7.

Similarly, all the modules altogether were simulated using the same method to study serpentine type navigation.



**Figure 8:**Block diagram for four module design of snake robot

Each port block represents 1 module and internally consists of block diagram as shown in Figure 8. The output for the block diagram can be seen in MALAB output window as shown in Figure 8.



(a) (b)

**Figure 9:** Simulation output window

Here we can see that the snake robot opens up and closes and returns back to original position by changing the sign if the equation fed to the function block we can achieve reverse orientation of the above Figure 9.

*4.1.1. Motion Study using Denavit-Hartenberg Parameters* Denavit – Hartenberg parameters are used to determine position of end effector of the robot when joint variables are numerically specified. For any complex link mechanism we need to define only four parameters i.e.  $\alpha$ ,  $a$ ,  $d$ ,  $\theta$ . Our snake robot mechanism is way much similar to industrial robot mechanism. But, with 8 actuators calculating position using D-H matrix becomes complex. MATLAB Robotics system toolbox helps to study the various orientation of robot for various gaits. MATLAB code for the same is given below

```
>> r= SerialLin
k(dh)
r =
noname:: 8 axis, RRRRRRRR, stdDH, slowRNE
+---+-----+-----+-----+-----+
|j|  theta |    d |    a |  alpha |  offset |
+---+-----+-----+-----+-----+
| 1|   q1 |    0 |    1 |  1.57 |    0 |
| 2|   q2 |    0 |    1 |  1.57 |    0 |
| 3|   q3 |    0 |    1 |  1.57 |    0 |
| 4|   q4 |    0 |    1 |  1.57 |    0 |
| 5|   q5 |    0 |    1 |  1.57 |    0 |
| 6|   q6 |    0 |    1 |  1.57 |    0 |
| 7|   q7 |    0 |    1 |  1.57 |    0 |
| 8|   q8 |    0 |    1 |  1.57 |    0 |
+---+-----+-----+-----+-----+
>> r.plot([0 0 0 0 0 0 0 0])
>> r.teach
```

The command `r.teach` gives us a teach pendant to control the joint variables. Thus, by changing joint variables we can study various types of orientations of the robot.

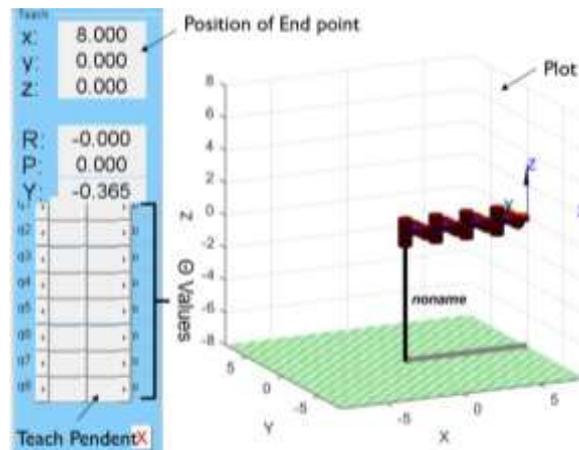


Figure 10: Plot for D-H matrix table

On the Teach pendant we can control the values of  $\theta$  and thus for various angles traced by motor we can get different orientations of the robot (Figure 10). Thus, we can study virtually the motion of the robot. We tried various orientations of the robot keeping Z axis co – ordinates constant and Y axis co – ordinates constant, respectively. The result is as shown in above figure. But we cannot precisely determine the exact values of  $\theta$  that are required for serpentine motion just by this trial and error method. The  $\theta$  values required for desired serpentine motion can be calculated using method mentioned in pervious sections. Assuming link length of 1 unit we found values of  $\theta$  as given in table.

Table 5:  $\theta$  Values

Step	$\theta 1$	$\theta 2$	$\theta 3$	$\theta 4$	$\theta 5$	$\theta 6$	$\theta 7$	$\theta 8$
1	-35	-39	-25	0	25	39	35	13
2	-39	-25	0	25	39	35	13	-35
3	-25	0	25	39	35	13	-39	-25
4	0	25	39	35	13	-25	0	25
5	25	39	35	13	0	25	39	35
6	39	35	13	25	39	35	13	0
7	35	13	39	35	13	25	39	35
8	13	35	13	39	35	13	25	39

These steps are repeated to continue motion of the robot. The plot corresponding to these values of  $\theta$  is as shown in Figure 11 and 12.

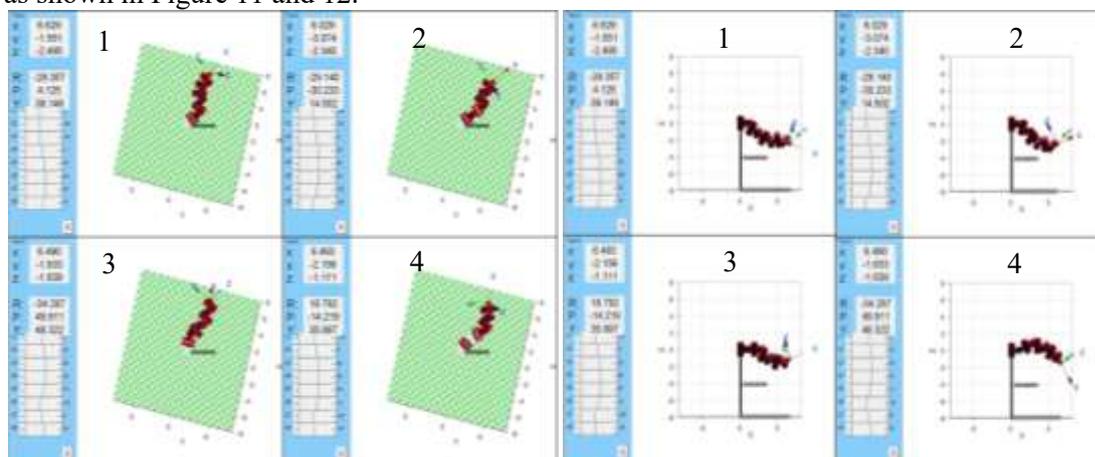


Figure 11: Top view plot of step 1 to 4

Figure 12: Side View plot of step 1 to 4

The plot of steps 5 to 8 are mirror images of steps 1 to 4 in reverse order which is similar to snake motion. Above results clearly shows that the robot performs sidwinding gait when we are using the calculated values of angular positions of the motor shaft. Thus, by varying  $\theta$  values on the teach pendant we can study the motion of the snake robot in virtual environment. According to the above study we can infer that the snake robot can perform serpentine gait from the calculated values. Hence, the calculated values are correct and can be used to feed to the motors through code for actual run of the robot.

## 5. Control System

The Dynamixel servo motors are controlled through Arduino nano and is powered by 12 Volt Lithium Polymer (LiPo) battery. The Arduino nano is small, complete and breadboard friendly board based on ATmega328 (Arduino Nano 3.x.), it has the same functionality as that of Arduino UNO but in a small size. This is helpful in building a compact circuit for our snake robot considering its number of actuators and compact structure. Arduino nano has an operating voltage of 5V however, the input voltage may vary from 7V to 12V.

Arduino nano can be easily programmed using Arduino IDE which is Integrated Development Environment. Arduino software IDE has library functions through which we can control the positions of Dynamixel servo motors according to the type of motion required for the snake robot.

## 6. Conclusions

The Dynamixel Servo AX-12A used gives a torque of 15.3 kg-cm and repeatability up to 90% and 100% accurate positional feedback for tracing required angular position of motor shaft. The number of modules used are 4 and hence 8 actuators are used providing a compliant motion to the robot as we get eight Degrees of freedom. High Degree of freedom ensures that the serpentine gait is achieved, and the robot can maneuver like a snake. Glossy PLA material is used for 3D printed parts it has a coefficient of friction of 0.42. PLA is less prone to warping during print and is much more 'stickier' than ABS. As it holds the layers firmly hence there is very less probability of breakage thus it gives required strength and material properties as compared to ABS and normal PLA material. The behaviour of the snake robot in a virtual environment was studied using MATLAB Simulation and Robotics Toolkit which generated results corresponding Denavit Hartenberg Parameters of the robot these results were used to feed angular positions to the motors and soft code was generated using the same angular positions. Arduino Nano is used as controller and code is generated in Arduino software which has functions to control motion of Dynamixel Servo motors.

## References

- [1] C. Ye, S. Ma, B. Li, Y. Wang and J. Tao, "Dynamics analysis on serpentine locomotion of a new snake-like robot", Robot, vol. 27, no. 6, (2005).
- [2] C. Holden, O. Stavadahl, and J. T. Gravadahl. "Optimal dynamic force mapping for obstacle-aided locomotion in 2D snake robots". In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Chicago, Illinois, United States. 2013, pp. 321–328.
- [3] F. Sanfilippo et al. "A review on perception-driven obstacle-aided locomotion for snake robots". In: Proc. of the 13th International Conference on Control, Automation, Robotics and Vision (ICARCV), Phuket, Thailand. 2016, pp. 1–7
- [4] J. Burdick, J. Radford, and G. Chirikjian, "A 'sidwinding' locomotion gait for hyper-redundant

- robots” in Proc. IEEE Int. Conf. Robot. Autom., May, 1993, pp. 101–106.
- [5] A.A. Transteth and K. Y. Pettersen, “*Developments in snake robot modeling and locomotion,*” in Proc. IEEE Int. Conf. Control, Autom., Robot. Vis., Singapore, Dec. 2006, pp. 1393–1390.
- [6] O. Egeland and J.T. Gravdahl, *Modeling and Simulation for Automatic Control*. Trondheim, Norway: Marine Cybernetics, 2002, pp.306- 311.
- [7] M.H.A. Majid, M.R. Khan, S.N. Sidek “*Development of Wheel- Less Snake Robot with Two Distinct Gaits and Gait Transition Capability*”. In International Journal of Automation and Computing. December 2013, pp. 255-259.
- [8] [X. Zhao](#), [L. Dou](#), [Zhong Su](#) and [N. Liu](#),” *Study of the Navigation Method for a Snake Robot Based on the Kinematics Model*“, 2018, pp.1255-1252.
- [9] L. Pfozter, S. Klemm, A. Roennau, J. M. Zöllner, and R. Dillmann, “*Autonomous navigation for reconfigurable snake-like robots in challenging, unknown environments,*” Robotics and Autonomous Systems, 2017,pp. 610-613.
- [10] Manzoor S, Cho YG, Choi Y “*Neural oscillator based CPG for various rhythmic motions of modular snake robot with active joints*”, 2019, pp. 1124-1132.

### Acknowledgements

We express our sincere gratitude towards Dr S.S. Ohol for guiding us throughout the project. We also thank Dr M.R. Nandgaonkar and Department of Mechanical Engineering for providing the facilities. We are immensely grateful towards Hon. Director Dr B.B. Ahuja for being a constant source of inspiration. We thank the Robot Study Circle of the Production Engineering for providing state of the art facilities which have proved to be invaluable for the completion of this report.