

# Experimentation for Modular Robot Simulation by Python Coding to Establish Multiple Configurations

Muhammad Haziq Hasbulah<sup>1</sup>, Fairul Azni Jafar<sup>2</sup>, Mohd. Hisham Nordin<sup>3</sup>, Kazutaka Yokota<sup>4</sup>

Faculty of Manufacturing Engineering, Universiti Teknikal Malaysia Melaka<sup>1, 2, 3</sup>

Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia

Research Div. of Design and Eng. For Sustainability, Graduate School of Engineering<sup>4</sup>

Utsunomiya University 7-1-2 Yoto, Utsunomiya-shi, 321-8585, Japan

**Abstract**—Most of the Modular Self-reconfigurable (MSR) robots are being developed in order to have the capability of achieving different locomotion gaits. It is an approach of robotic system which involving a group of identical robotic modules that are connecting together and are able to perform specific tasks. In this study, a 3D-printed MSR robot named Dtto-Explorer Modular Robot was used to investigate the achievable propagation that can be made based on three of Dtto robot modules. As Dtto robot is developed based on Modular Transformer (M-TRAN), Dtto robot has the number of Degree of Freedom (DOF) same as M-TRAN which is 2 DOF. Hence, this study is done with the intentions to know the variation of configuration that can be made by multiple module robots which have only 2 DOF. The robot propagation was simulated in Virtual Robot Experimentation Platform (V-REP) software. The result of the simulation shows that the Dtto MSR robot can propagate multiple configurations after all the robot modules being connected to each other by different attachment orientation of the Dtto robot and it is suitable for the purpose of further research on MSR robot architecture.

**Keywords**—Dtto robot; simulation; configuration; locomotion; orientation

## I. INTRODUCTION

The Dtto modular robot is developed based on M-TRAN robot which was first developed by Murata et al. [1]. The module consists of two semi-cylindrical parts that can be rotated about its axis and with a link. It possess 2 DOF [2], which resemble M-TRAN robot module ability. However, the Dtto modular robot is being designed so that each module can be minimized as much as possible to have a large free space in half of the robot. It is purposely done so that it can be used by users to set up their preferred sensor such as Infrared (IR) sensor or install more actuators. It is 3D printable and at low cost to be builds. This study is done by using Dtto robot, design by Alberto [2] because the result from this study will be used as a references to our next research purpose as this robot can be redesigned as it is an open-source robot and it is fabricated by using only 3D printer.

As it is being designed based on M-TRAN modular robot, a review summation of Dtto and M-TRAN modular robot

comparison has been performed and can be seen as in Table I.

The comparison between the robots is based on several characteristics which are CPU, Communication, Battery, Sensor-applied and Robot Connection Mechanism. M-TRAN robot is a Hybrid type MSR robot which is in the form of Chain and Lattice, same goes with Dtto robot. Hence, we believe that Dtto robot is able to propagate multiple configurations. However, with only 2 DOF, it might not have many movements ability to propagate multiple configurations. It is believe that the collective behavior of modular robot could establish more movements or propagate multiple configurations of different locomotion gaits [5], even though the movement is limited for one robot module.

Basically, most of the MSR robots that have been developed are able to propagate snake-like motion, which is minimal possible configuration for modular robot to have locomotion ability. It is known as Serpentinoid curve or Sinusoidal curve which was discovered in 1976 by Hirose for application of biomechanics of snake to robot construction [6][7], so the robot can propelled forward by sending Sine and Cosine value. Basically, its implementation for the locomotion mechanism for snake-motion propagation can be referred from research study done by Gómez [8]. The mechanism theory will be discussed in Section 2 of this paper. Besides that, other possible propagation for the Dtto modular robot were also established as this study is done to ensure the Dtto robot with 2 DOF is able to propagate multiple configurations with multiple robot modules. Besides snake-like motion, the other configurations being propagate are Cube-shaped, U-shaped, Cuboid-shaped and Chain-shaped. This study has been done by establishing simulation in V-REP software, controlled by Python coding.

In the sections to follow, we discuss the theoretical concepts of Snake-like locomotion where it is being implemented for establish multiple configuration of the Dtto modular robot. Then we discussed the methodology approach for conducting this experimentation in Section III. The simulation results are illustrated and discussed in the Section IV and this paper is concluded in Section V.

TABLE I. SUMMARIZATION OF COMPARISONS FOR DTTO AND M-TRAN MODULAR ROBOT

Characteristics	Comparisons Summarization			
	M-TRAN	M-TRAN II	M-TRAN III	Dtto
CPU	BasicStamp II	Neuron chip, Three PICs	HD64F7047, HD64F3687, HD64F3694	Arduino Nano v3.0
Communication	Asynchronous serial	LonWorks & RS-485, Asynchronous serial	Bluetooth wireless modem	Bluetooth wireless, HC-05, RF24L01
Battery	DC 12V	Li-ion	Lithium-polymer	Li-Po
Sensor Applied		Acceleration sensor	IR proximit, IR diode, IR sensor	InfraRed LED Emitter- Receiver (Optiona)
Robot Connection Mechanism	SMA (shape memory alloy) coil and magnet	SMA (shape memory alloy) coil and magnet	Mechanism based on latch connector	Permanent magnet (Neodymium magnet)
References	[1]	[3]	[4]	[2]

## II. THEORETICAL FOR SNAKE-LIKE LOCOMOTION CONCEPTS

The theoretical concept for the robot to have Snake-like Locomotion needs to be unraveled so we can understand how the motion works. The angle of curvature that varies in sinusoidal with the distance along the curve [9] is based on the equation of curvature where it define the Serpentinoid curve or Sinusoidal curve. Equation (1) is the equation of curvature mentioned where  $k$  is the number of undulations,  $l$  is the length of the curve,  $s$  is the distance along the curve, and  $\alpha$  is the value of angle that forms the tangent to the curve when  $s = 0$  which determine the shape of the curve.

$$K(s) = -\frac{2\pi k}{l} \alpha \sin\left(\frac{2\pi k}{l} s\right) \quad (1)$$

The curvature definition can be defined by the rate of change of the vector unit tangent to the curve when it is provided with the Cartesian coordinates  $(x,y)$  of the points located in the Serpentinoid curve which depends on the temporal variable,  $t$  as in Equation (2) and (3).

$$x(s, t) = \int_0^s \cos\left(\alpha \cos\left(t + \frac{2\pi k}{l} s\right)\right) ds \quad (2)$$

$$y(s, t) = \int_0^s \sin\left(\alpha \cos\left(t + \frac{2\pi k}{l} s\right)\right) ds \quad (3)$$

However, according to Rodríguez [8] and Gómez [9], the mentioned equations, where it is proposed by Hirose, does not have analytical solution. Hence, it opened up various possibilities for establishing this configuration by coordinated the joints of the robot so that the serpentinoids wave can be reproduces. Hence, one of the alternative that we adopted in our study is by calculating the angles which will being applied to the robot's joints to reach the desired position. So, the reference of the robot movement will be based on joints target position according to a sinusoidal function.

Based on the simplified Serpenoid curve [10], the snake robot gaits generation way, arc length is defined as the curvature of the  $s$ , where  $b$  is constant:

$$p(s) = -ab \sin(bs) \quad (4)$$

Hence, for the generation of winding or oscillates movement gait, the relative angle of each joint has to be determined which will create a static configuration. However, for locomotion propagation for the robot to move, the  $s$  value has to be changed with respect to time sequentially. Hence, the corner of dynamic circumstances in the modular snake-like robot joint where  $\varphi$  is angle of rotation is as Equation (5) [7]:

$$\varphi_i(t) = A \sin(\omega t + (i - 1)\beta) \quad (5)$$

Hence, the Equation (5) is applied in our study for establishing this configuration where  $i = 0,1,2,3 \dots$ , is the number of modular robots series. The modular robot oscillates according to a sinusoidal function and its parameters such as amplitude, speed and phase shift is being determined at first.

## III. METHODOLOGY

The robot control is established by interfacing with Python as we want to provide external control to the Dtto robot in simulation environment for each configuration. This communication is maintained via a while loop and the configuration of the robot changed based on the input, where the robot configuration is predetermined which one of the configurations and is based on the snake-like motion equation [6][7][11], so the robot can propelled and create motion for the robot to move in space. The following flowchart (Fig. 1) is the establishment of Python to V-REP communication so that the robots propagate multiple configurations. V-REP is integrated simulation environment where each object can be controlled individually by remote Application Program Interface (API) client.

Three following files are needed for using remote API functionality in Python script which are vrep.py, vrepConst.py and remoteApi.dll. Those files have to be in the same directory with Python script project. It is because those elements directory is need to be known by Python. The three mentioned files can be located in V-REP installation directory.

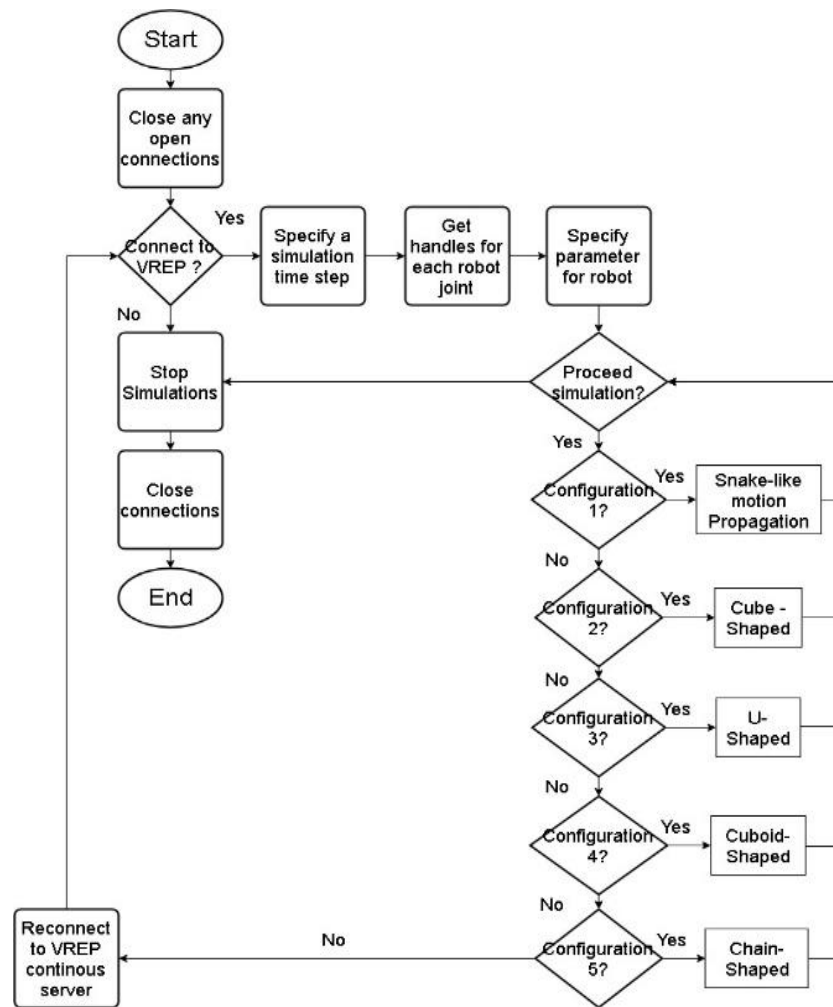


Fig. 1. All the Number of Joints for the Robot being Access in Python Coding.

For this research work, all the virtual Dtto modular robots will be controlled based on number of joints. Hence, as the robot has multiple of joints which is 2, by using the same name, the robot joint will be accessed based on the number. The Python list is created to access the elements of the list. As in this study, the number of module robot is 3 (which mean 3 elements in the list), the list will be accessed by indexing starting from 0 to 2 as in Fig. 2.

As the code is running, the while-loop code will start as the user enter the letter “Y” by the keyboard which indicate that the user want to proceed with the simulation. As the code running, user will need to enter the number ‘1’, ‘2’, ‘3’, ‘4’, and ‘5’ that indicate the configuration being simulated. The robot movement code in the while loop will keep running and the robot configuration can be changed from time to time based on the user input. In this methodology, Dtto MSR robot model is used and imported into V-REP. Besides that, time steps of V-REP simulation is specified for a small value which is 0.001. The increment of simulation time will be based on each of the simulation time step for each time the simulation is executed. Four pre-determined configurations were selected based on the three axis configurations that are achievable, which are Cube-shaped, U-shaped, Cuboid-shaped and Chain-shaped.

```

position(clientID, maleMotor[0],
((A_V[0] * (1 - s)) + (A_V[1] * s)) * (math.sin(t * (
((speed[0] * (1 - s)) + (speed[1] * s)) + (
0 * P_V[0] * (1 - s)) + (
P_V[1] * s))), vrep.simx_opmode_onesho
position(clientID, femaleMotor[0],
((A_H[1] * (1 - s)) + (A_H[0] * s)) *
(math.cos(t * (

```

Fig. 2. All the Number of Joints for the Robot being Access in Python Coding.

#### IV. RESULTS AND DISCUSSION

Discussion and interpretation based on the experimentation result are discussed in this sub-topic. The configurations propagate by the robot need to be discussed to have a better understanding on the capability of the Dtto modular robot to propagate multiple configurations.

It is believe that the main configuration for the Dtto robot propagation is the snake-like motion that has been discussed in Section 2. Dtto modular robot does not have specific item that give it mobile capabilities such as wheels. That is why snake-like motion is one of the configurations that is necessary to give Dtto modular robot an ability to move. This configuration

is chosen because it is suitable to be implemented to modular robot as it has similarity with snake type robot which is modularity that can be seen to snake robot for each of its segment. The other configuration possible with limited DOF is studied and all configurations are simulated by V-REP software.

To propagate multiple configurations for the robot, the approach as in Flowchart of Fig. 1 was followed to show that the robot is able to propagate multiple configurations with 3 modules of Dtto robot where one module only have 2 DOF.

Fig. 3(b) shows that to have 3 axis configuration (X, Y, and Z), one or more of the modular robot have to be in different position compare to the one in Fig. 3(a) whereas the modular robots is in same position attachment, it only have 2 axis configuration (X and Z). By having an ability to propagate configuration by 3 axis, it gives more possible configuration for the modular robots to propagate.

The first configuration being studied for Dtto modular robot is the ability to propagate a snake-like motion. At first, the robot will be studied as the robot has the same position attachment with 2 axis configuration as in Fig. 3(a). The robot orientation will be as Fig. 4 which at X and Z axis (Left) and X and Y axis (Right). Fig. 4 also shows the Axis-Time graph that shows the motion created for the robot. Axis-Time graph will be used to analyze the movement of the robot during its propagation in term of axis respected to simulation time. Based on the Equation (5), for the first configurations, the snake-like motion equation that being coded in Python language is as Equation (6) and (7) where *maleMotor* and *femaleMotor* is actually being specify for robot's joint.

$$maleMotor[i] = (A_{V[1]}(1 - s)) \sin(t(v_{[1]}(1 - s)) + i(P_{V[1]}(1 - s))) \quad (6)$$

$$femaleMotor[i] = (A_{V[2]}(1 - s)) \cos(t(v_{[1]}(1 - s)) + i(P_{V[2]}(1 - s))) \quad (7)$$

From Fig. 4 the modular robot at X-Z axis orientation create a caterpillar type motion and from the Axis-Time graph it shows that there is increasing value of the X-axis data. It means that, this configuration creates a propelled motion that move the 3 modular robot as one towards X-axis direction (propagate forward). For the modular robot at X-Y axis orientation, it creates a snake-like motion by sending a value of Sine and Cosine to the robot. We can see from the Axis-Time graph that the snake-like motion create a wave like graph that known as a serpenoid curve based on values of angles. However, for the modular robot at X-Y axis orientation, the graph shows unnoticeable changes for X-axis and Y-axis data. The simulation resulting as in Fig. 4(b), the motion of the robot for X axis is forward and reverse in serpenoid motion without propagate forward or backward. The motion for Y-axis is left and right in serpenoid motion, without propagating to left or right. Hence, this orientation is not suitable because we need

at least one configuration that gives movement to the robot. However, even though the configuration as Fig. 4(a) gives the movement ability to the robot, it only propagates by 2 axis of configuration which limits the number of configuration that is able to be made. That is why we choose to have the modular robots in attachment position as Fig. 3(b) so that the snake-like motion can be achieved with 3 axis configuration of the robot which give more possibilities for multiple configurations.

For simulating multiple configurations, as mentioned before, the first configuration simulated is the snake-like motion but with the orientation of modular robots as in Fig. 3(b). The simulation result can be seen in Fig. 5.

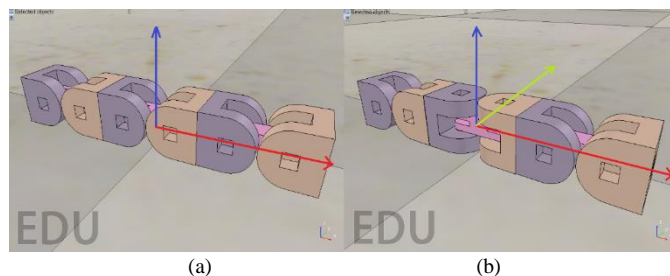


Fig. 3. Uniform Attachment Position (a) and Irregular Attachment Position of Robot (b).

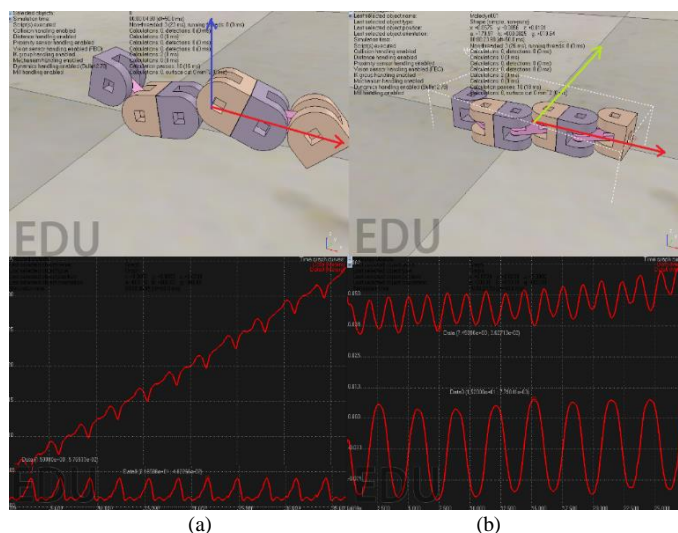


Fig. 4. Robot X-Z Axis Orientation (a) and X-Y axis orientation (b) with Axis-Time Graph.

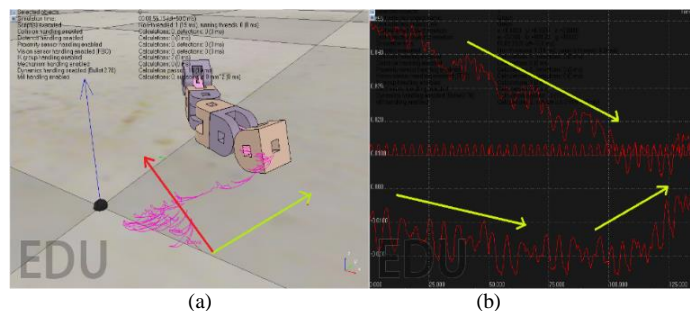


Fig. 5. Robot Snake-Like Motion Propagate (a) and Axis-Time Graph (b).

Based on Fig. 5, it can be seen that the robot is able to move towards right direction (Y-axis) and propagate backwards (X-axis) simultaneously which is proving that at this orientation, this configuration that provides locomotion to the Dtto modular robot can be achieved. Based on the Axis-time graph in Fig. 5(b), there are changes for X-axis data (downward direction) of the Sine and Cosine wave which means this configuration create a propelled motion that move the 3 modular robots as one towards X-axis direction but propagate backward. There are also changes for the Y-axis data and there is no change in Axis-time graph for Z-axis data which shows that the propagation of the robot for this orientation is at X-axis and Y-axis just as mentioned and shown in Fig. 5(a). Hence, it means that there is movement that the robot produces and it shows that this configuration is suitable for the purpose of robot locomotion. It is not necessary for the robot to propagate snake-like motion first, but we decided to have one propagation for robot mobility. The snake-like motion was studied first because it is complicated compared to other predetermined configurations that we decided. We decided for snake-like motion because it has similarity with normal snake robot (modular with redundant segment).

Then, the other propagation was simulated as we want to know whether the multiple configurations can be achieved with this configuration or not. As mentioned before, the other configurations being propagate are Cube-shaped, U-shaped, Cuboid-shaped and Chain-shaped. Fig. 6 shows that the other 4 predetermined configurations are able to be simulated which show that Dtto modular robot is able to propagate multiple configurations. The configuration as in Fig. 6 is being propagated in V-REP software by manipulating the equation coded as Equation (6) and (7). For configuration as Fig. 6(a), the value for *maleMotor* and *femaleMotor* is based on Sine value only. The propagation created is based on the Equation (8) and (9) that being coded in Python.

$$maleMotor[i] = (A_{V[1]}) \sin(t(v_{[1]}) + (P_{V[1]})) \quad (8)$$

$$femaleMotor[i] = (A_{V[2]}) \sin(t(v_{[1]}) + (P_{V[2]})) \quad (9)$$

Then the configuration as in Fig. 6(b) being developed for Dtto robot to propagate by having Sin and Cosine value for first *femaleMotor* and *maleMotor*, respectively for 1st robot module. The 2nd robot module have been specified with exactly 0 value for both *maleMotor* and *femaleMotor* and finally, the 3rd robot module being specify with Cosine and Sine value for *femaleMotor* and *maleMotor*, respectively.

$$maleMotor[0] = (A_{V[1]}) \sin(0(v_{[1]})) + (P_{V[1]}) \quad (10)$$

$$femaleMotor[0] = (A_{V[1]}) \cos(t(v_{[1]})) + (P_{V[1]}) \quad (11)$$

$$maleMotor[2] = (A_{V[1]}) \sin(t(v_{[1]})) + (P_{V[1]}) \quad (12)$$

$$femaleMotor[2] = (A_{V[1]}) \cos(0(v_{[1]})) + (P_{V[1]}) \quad (13)$$

The 4th configuration being developed by coded equation in which the value of Sine is being specified for

*femaleMotor* of the 1st robot module and for *maleMotor* of the 3rd robot module as in Equation (14) and (15).

$$maleMotor[0] = femaleMotor[1] = maleMotor[1] = femaleMotor[2] = (A_{V[1]}) \cos(0(v_{[1]})) + (P_{V[1]}) \quad (14)$$

$$femaleMotor[0] = maleMotor[2] = (A_{V[1]} + A_{V[2]}) \sin(t(v_{[1]} + v_{[2]} + (P_{V[2]}))) \quad (15)$$

Last but not least, the 5th configurations able to be propagating by specify the value 0 for *femaleMotor* of first module robot, *maleMotor* of third robot module. The Sine value being specify for *maleMotor* of first robot module and *femaleMotor* for third robot module. This propagation is being done as the Equation (16) being coded into Python language.

$$maleMotor[0] = femaleMotor[2] = (A_{V[1]}) \sin(0(v_{[1]})) + (P_{V[1]}) \quad (16)$$

All other configurations besides Serpentinoid configuration that being simulated by Python in V-REP simulation environment in this experiment can be seen as in Fig. 6.

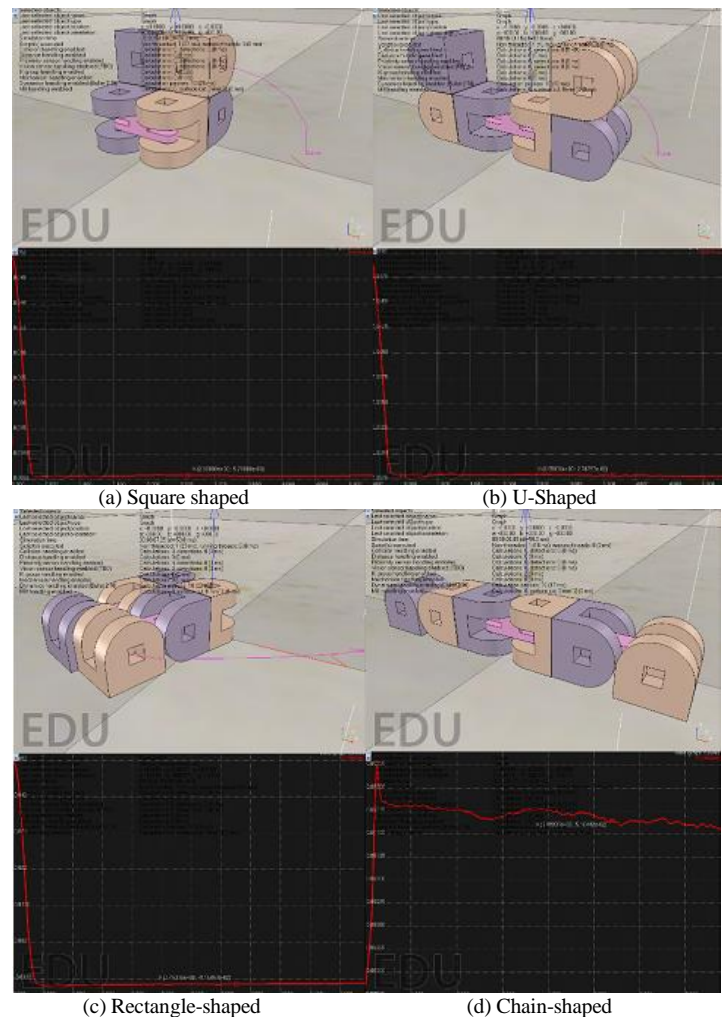


Fig. 6. Robot Snake-Like Motion Propagate (a) and Axis-Time Graph (b).

Based on Fig. 6, it shows that DTTTO modular robot able to propagate multiple configurations in state of orientation as in Fig. 3(b). As the axis-time graph in Fig. 6, it shows that there are movement for the robot to propagate its own configuration and the movement stop as shown in the graph where the axis data value is keep constant which shown the robot movement for propagation stop after it achieved the desired propagation shape.

## V. CONCLUSION

In conclusion, this study is conducted to ensure that the Dtto modular robot is able to propagate multiple configurations and will provide significant information for future research work. The simulation created is maintained by while-loop in Python coding so the robot is able to propagate multiple configurations without reprogram. Due to the robot limitation where Dtto only have 2 DOF, with multiple number of module robot, it is necessary to establish possible position attachment, and by establish orientation of 3 axis configuration for the robots, it will establish more propagation of different configurations and locomotion for the modular robot.

## REFERENCES

- [1] S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, and S. Kokaji, "M-Tran: Self-Reconfigurable Modular," *Ieee/Asme Trans. Mechatronics*, Vol. 7, No. 4, PP. 431–441, 2002.
- [2] Alberto, "Dtto - Explorer Modular Robot," Hackaday, 2016. [Online]. Available: <https://hackaday.io/project/9976-dtto-explorer-modular-robot>. [Accessed: 04-Jul-2017].
- [3] H. Kurokawa, A. Kamimura, E. Yoshida, K. Tomita, S. Kokaji, and S. Murata, "M-TRAN II: Metamorphosis from a four-legged walker to a caterpillar," in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*, vol. 3, pp. 2454–2459, 2003.
- [4] H. Kurokawa, K. Tomita, A. Kamimura, S. Kokaji, T. Hasuo, and S. Murata, "Distributed Self-Reconfiguration of M-TRAN III Modular Robotic System," *Int. J. Rob. Res.*, vol. 27, no. 3–4, pp. 373–386, 2008.
- [5] M. Yim, "New locomotion gaits," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pp. 2508–2514, 1994.
- [6] S. Hirose, *Biologically inspired robots: snake-like locomotors and manipulators*, Illustrate. Oxford University Press, 1993.
- [7] W. Nan, P. Bo, and Z. Sha-Sha, "Simulation study of snake-like robot's serpentine locomotion based on recurdyn," *Res. J. Appl. Sci. Eng. Technol.*, vol. 7, no. 1, pp. 37–41, 2014.
- [8] J. G. Gómez, "SERPENTINOID CURVE," Jupyter nbviewer, 2015.
- [9] A.S.M., Rodríguez, "Diseño , construcción y control de una serpiente robótica," in *Congreso Nacional de Investigación en Grado INVESGRADO*, pp. 1–15, 2012.
- [10] S. Ma and N. Tadokoro, "Analysis of Creeping Locomotion of a Snake-like Robot on a Slope," *Auton. Robots*, vol. 20, no. 1, pp. 15–23, 2006.
- [11] J. Gonzalez-gomez and E. Boemo, "Motion of Minimal Configurations of a Modular Robot: Sinusoidal , Lateral Rolling and Lateral Shift Motion," in *Climbing and Walking Robots: Proceedings of the 8th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR 2005)*, pp. 667–674, 2005.