BTP II Presentation

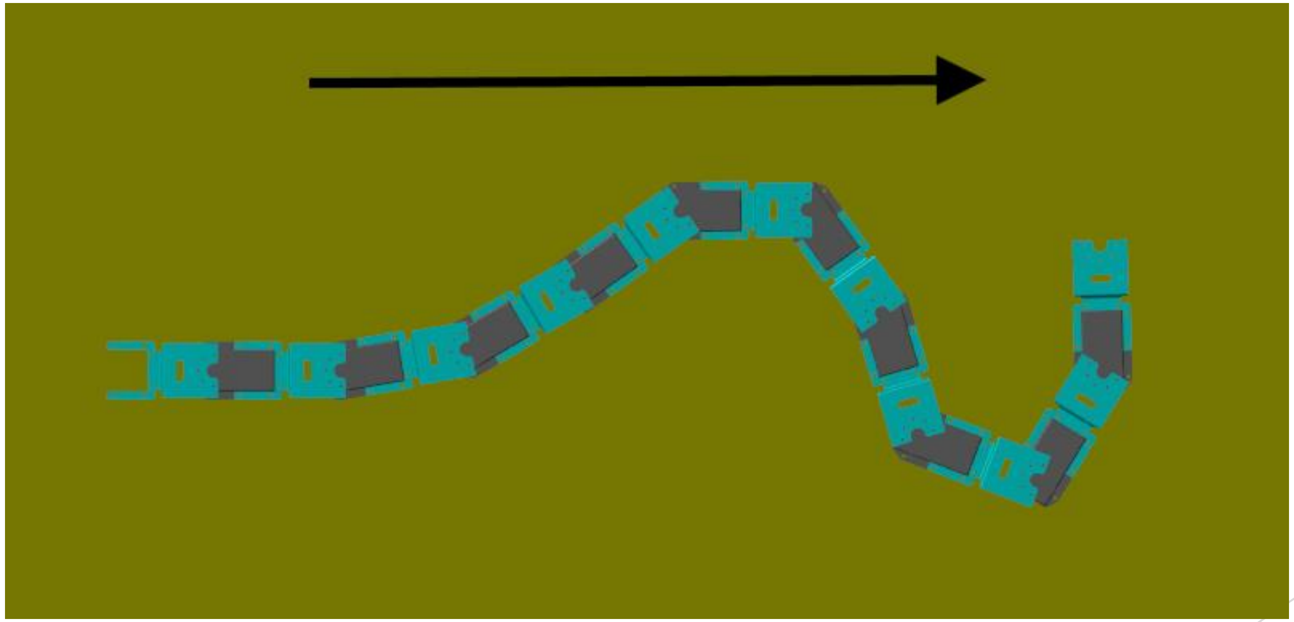
- Saayuj Deshpande, 200100139

Previous Work

- Mathematical modeling of the snake robot for our case
- Preliminary simulation of planar snake robot

Testing – Amplitude & Phase

- Tested snake robot for various scenarios in software & verified with team in hardware
- Frequency and bias of sine wave kept at 1 rad/s and 0 rad respectively for first four cases
- Figure below shows robot and directions used for further analysis:



Testing – Amplitude & Phase - 2

- Case 1:
  - Increasing amplitude – from 0 to 1.4 rad, in steps of 0.2 rad, with last 3 links at 1.4 rad
  - Negative phase difference – successive links have a phase difference of -1 radian
  - Bias - 0

Result: Snake moves leftward (opposite to black arrow)

- Case 2:
  - Increasing amplitude – from 0 to 1.4 rad, in steps of 0.2 rad, with last 3 links at 1.4 rad
  - Positive phase difference – successive links have a phase difference of +1 radian
  - Bias – 0

Result: Snake moves rightward (in direction of black arrow)

Testing – Amplitude & Phase - 3

- Case 3:
  - Decreasing amplitude – from 1.4 to 0 rad, in steps of -0.2 rad, with last 3 links at 0 rad
  - Negative phase difference – successive links have a phase difference of -1 radian

Result: Snake moves leftward (opposite to black arrow)

- Case 4:
  - Decreasing amplitude – from 1.4 to 0 rad, in steps of -0.2 rad, with last 3 links at 0 rad
  - Positive phase difference – successive links have a phase difference of +1 radian

Result: Snake moves rightward (in direction of black arrow)

- Conclusion: These 4 cases show us that only phase difference matters – snake moves in direction of increasing/positive phase difference

Testing – Frequency

- Now, let us check effect of frequency of sine wave on speed of snake
- Fixed conditions:
  - Snake length – approx. 800 mm
  - Increasing amplitude – from 0 to 1.4 rad, in steps of 0.2 rad, with last 3 links at 1.4 rad
  - Negative phase difference – successive links have a phase difference of -1 radian
  - Bias – 0 rad
- Results: (simulated for 150 s)

Frequency (rad/s)	Speed (mm/s)
1	17.5
1.5	26.7
2	36
2.5	22.9
3	7.2

- Conclusion: Speed increases with frequency, until it starts slipping

Testing – Variable & Constant Amplitude

- Fixed conditions:
  - Negative phase difference – successive links have a phase difference of -1 radian
  - Bias – 0 rad
  - Frequency – 1 rad/s
- <u>Case 1: Variable Amplitude</u>
  - Increasing amplitude from 0 to 1.4 rad, in steps of 0.2 rad, with last 3 links at 1.4 rad

<u>Result</u>: Speed – 17.5 mm/s

- <u>Case 2: Constant Amplitude</u>
  - All links have an amplitude of 1.4 rad

<u>Result</u>: Speed – 3.7 mm/s (exhibits circular motion/wriggling of sorts)

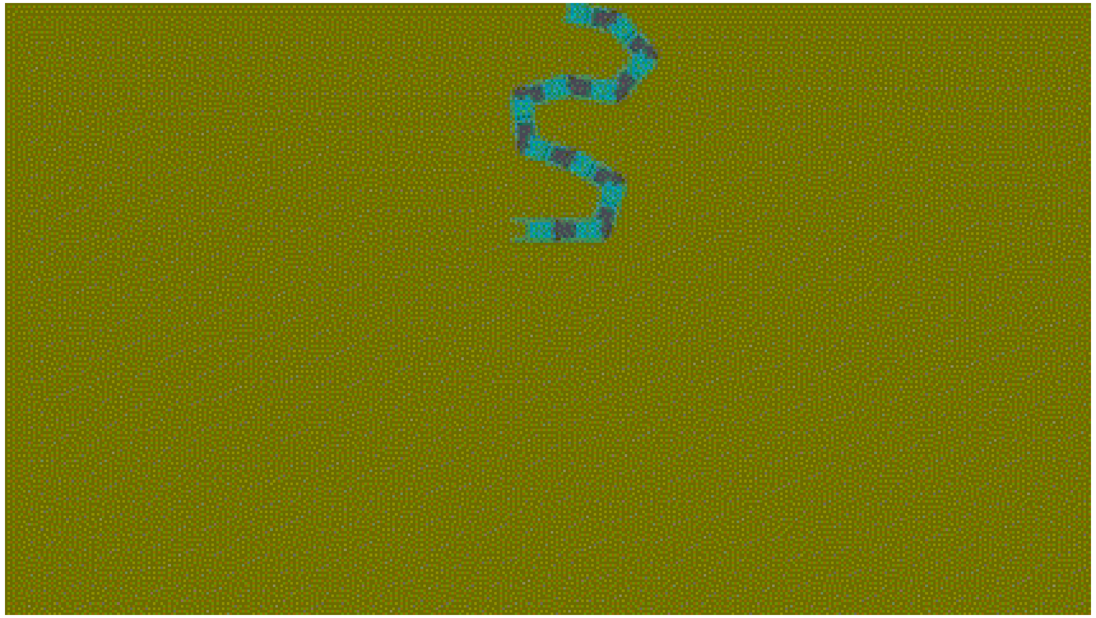
- <u>Case 3: Constant Amplitude</u>
  - All links have an average amplitude of 0.82 rad (pertaining to above variable amplitude case)

<u>Result</u>: Speed – 1.1 mm/s (hardly moves)

- <u>Conclusion</u>: Variable amplitude is much better, especially for feedforward

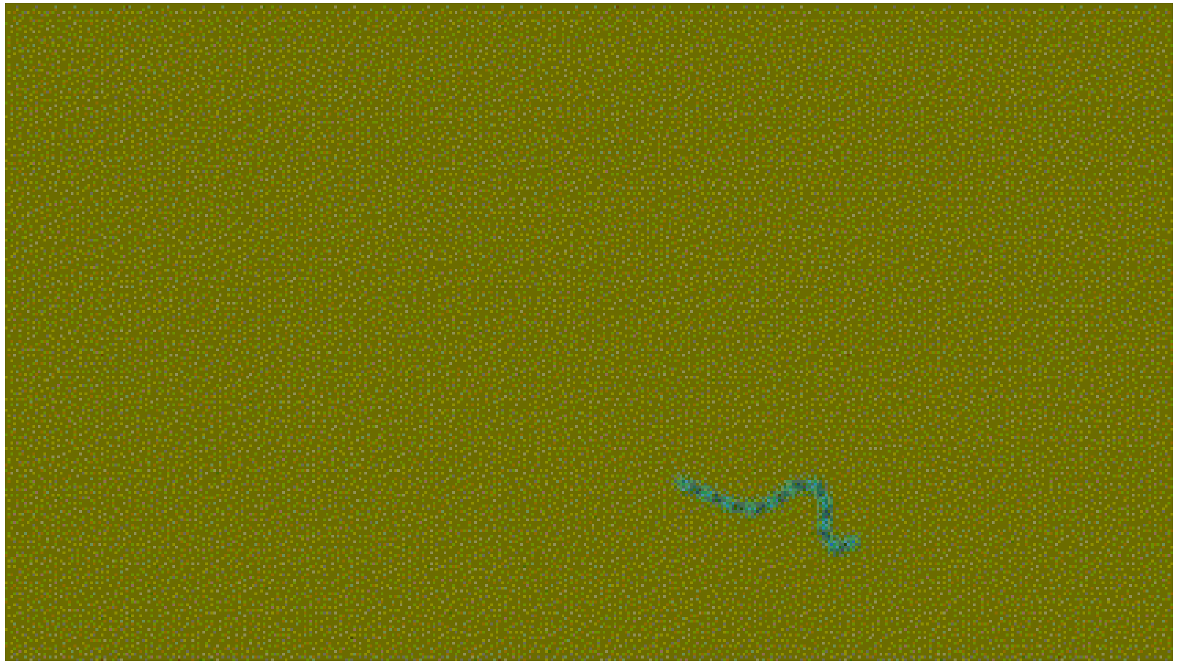
Testing – Bias - 1

- Fixed conditions:
  - Negative phase difference – successive links have a phase difference of -1 radian
  - Frequency – 1 rad/s
- <u>Case 1</u>: Constant amplitude
  - Snake always moves in a periodic circular fashion (wriggling), regardless of bias



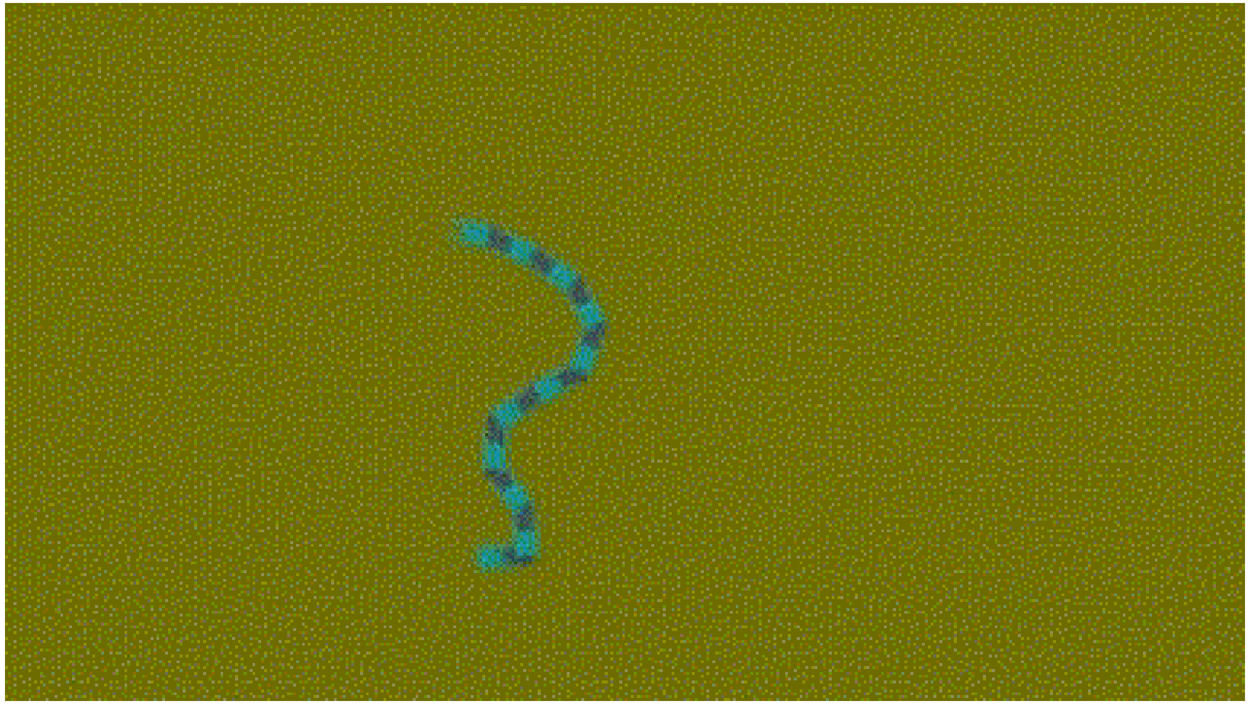
Testing – Bias - 2

- <u>Case 2</u>: Variable amplitude
  - Bias = 0 rad: Snake moves in a somewhat straight line



Testing – Bias - 3

- Bias $\neq$ 0 rad: Snake exhibits circular motion (not complete wriggling)



Testing – Conclusions

- Variable, increasing amplitude and a frequency of around 2 rad/s, with negative phase difference between successive links, works best
- Bias can be used as a control input for steering the robot

3D Snake Robot Model

- To explore another model for the snake robot, studied the 3D model
- Snake has 10 revolute joints – 5 allow motion in dorsal plane, 5 in lateral

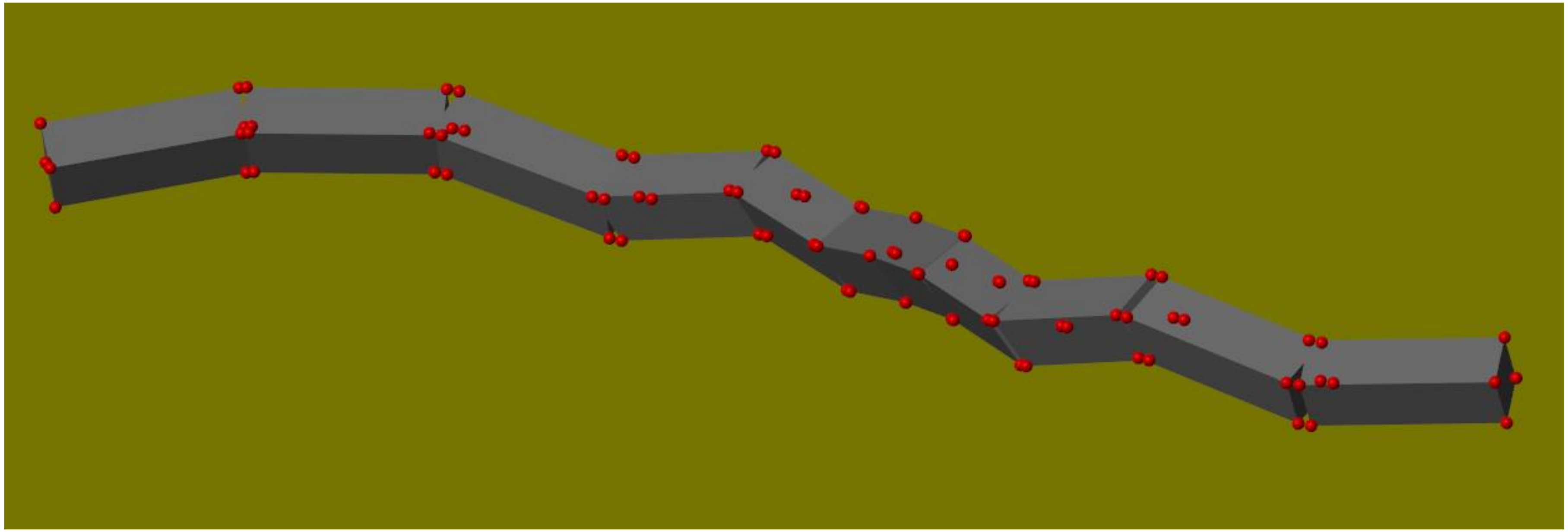


Fig: 3D model showing lateral and dorsal joints

- Links have been modeled as rectangular blocks for simplicity, and contact proxies have been used for accurate surface contacts

3D Snake Robot Model - 2

- We require two separate serpenoid motions to be passed through the lateral and dorsal planes to move the snake robot
- The compound serpenoid curve is described in *Control and Design of Snake Robots* by **David Rollinson** as follows:

$$\theta(n,t) = \begin{cases} \beta_{\text{lat}} + A_{\text{lat}} \sin(\xi_{\text{lat}}) & \text{lateral} \\ \beta_{\text{dor}} + A_{\text{dor}} \sin(\xi_{\text{dor}} + \delta) & \text{dorsal} \end{cases}$$

$$\xi_{\text{lat}} = \omega_{\text{lat}} t + \nu_{\text{lat}} n$$

$$\xi_{\text{dor}} = \omega_{\text{dor}} t + \nu_{\text{dor}} n.$$

- The parameters are chosen to make the simulation efficient in the following slides

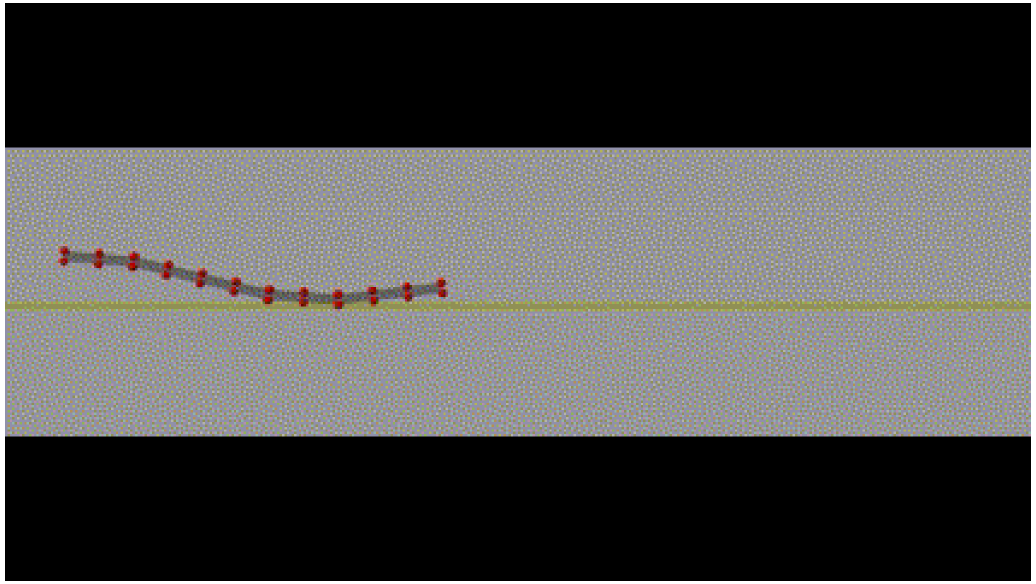
Gaits – Linear Progression

- For linear progression/caterpillar motion, the joint angles were programmed as follows:

$$\theta(n,t) = \begin{cases} 0.2 \; * \sin\left(1 \; * t - \dfrac{n}{2} - 1\right) & , if \; n \; is \; even/lateral \\ 0 & , if \; n \; is \; odd/dorsal \end{cases}$$

Here, n is the link number and t is the time. The frequency is 1 rad/s, amplitude is 0.2 rad and successive phase difference is 1 rad (only for lateral joints).

- This leads to a caterpillar-like motion as shown below:



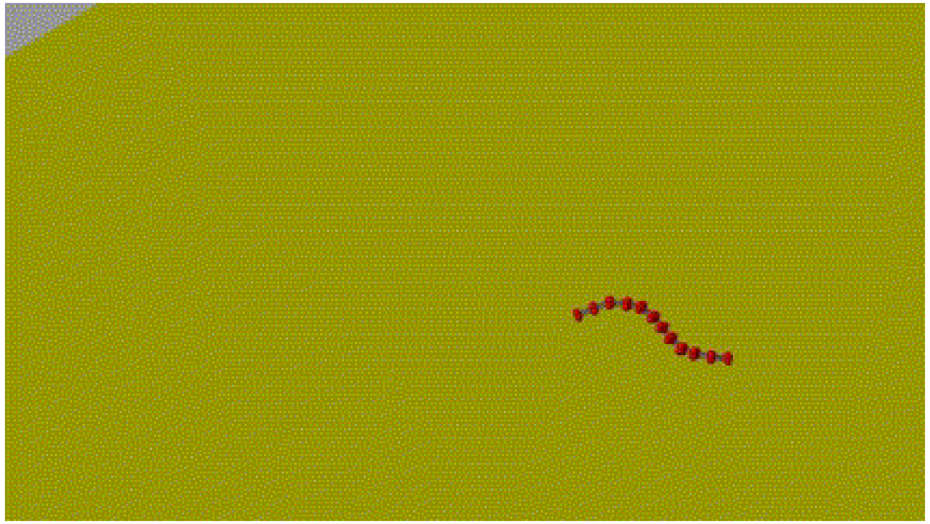
Gaits – Rolling & Sidewinding

- For this case, the joint angles were programmed as follows:

$$\theta(n,t) = \begin{cases} 0.5 \, * \sin\left(1 \, * t - \dfrac{n \, - 1}{2} + \dfrac{\pi}{6}\right) & , if\ n\ is\ odd/dorsal \\ 0.5 \, * \sin\left(1 \, * t - \dfrac{n}{2} - 1\right) & , if\ n\ is\ even/lateral \end{cases}$$

Here, the amplitude is 0.5 rad, frequency is 1 rad/s, and successive phase difference is 1 rad. Further, the phase difference between the dorsal and lateral joints is pi/6.

- This leads to the following motion: (similar to planar circular rolling)



3D Snake – Conclusions

- This model offers flexibility:
  - Can be used for caterpillar-like motion: by switching off dorsal joints
  - Can be used for planar snake motion: by switching off lateral joints
  - Can be used for sidewinding and rolling by using all joints
- Challenges:
  - Amplitude cannot be made large in simulation due to tilting of robot
  - Jerks at the start of the simulation – used a smoothening function for the same (provided a linear function for angles for the first 10 seconds)

Motor Control

- Till now, in all models, angles/motions were being fed directly to revolute joints
- Developed the motor control loop which takes in desired angular position as input and outputs torque which is fed to the revolute joints, which in turn rotates the links
- The setup consists of a:
  - DC motor
  - Rotational multibody interface
  - Controlled voltage source
  - PI control loop

Motor Control - Model

To link assembly

Desired angular position

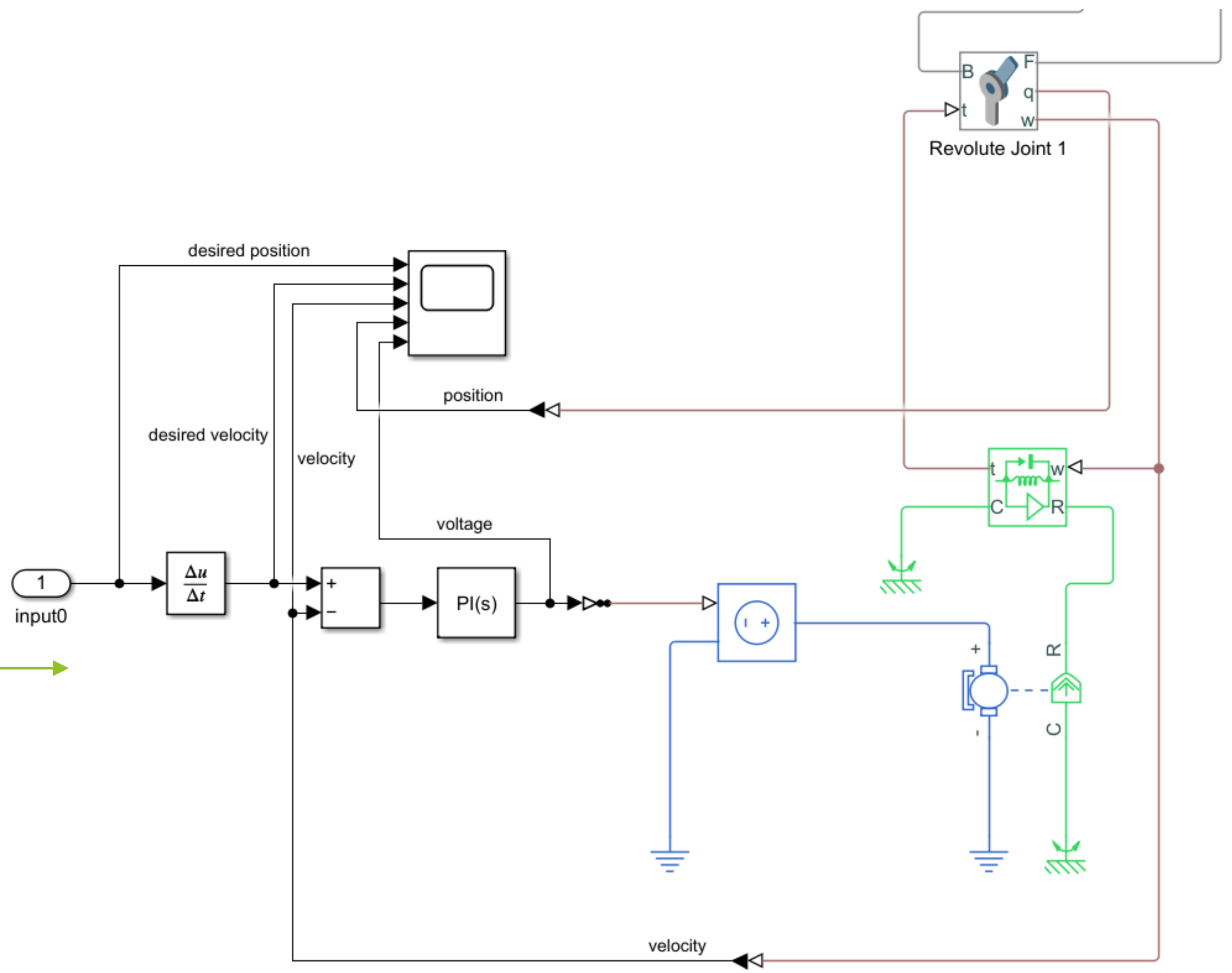


Fig: Motor control loop showing the various components

Motor Control - Results

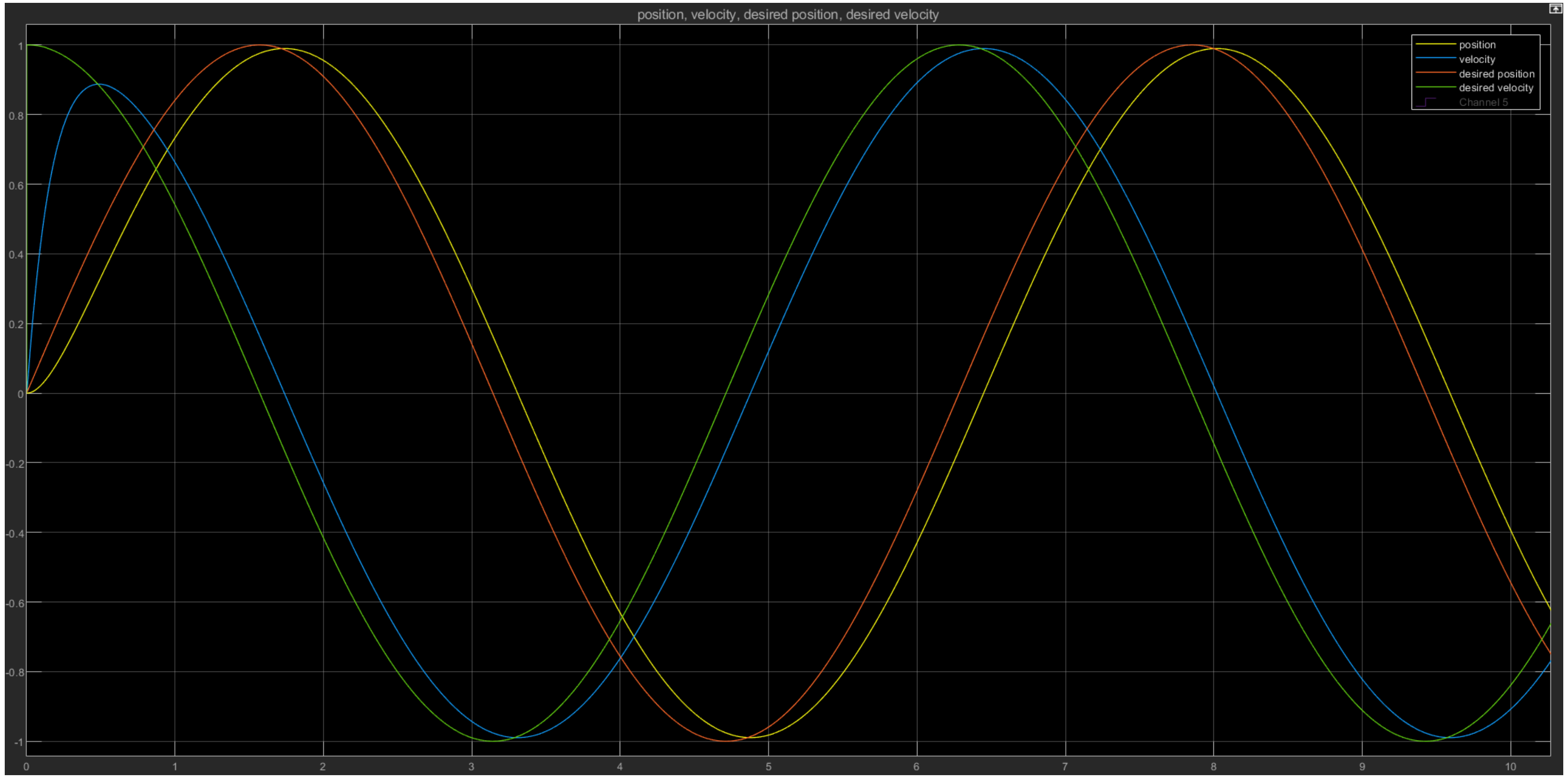


Fig: Position and velocity characteristics

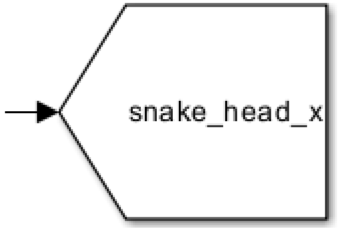
Legend: red – desired position; green – desired velocity; yellow – position; blue - velocity

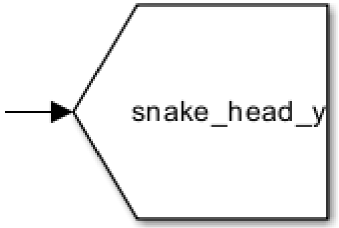
Motor Control - Drawbacks

- Pursued motor control to bring more realism into simulation, but voltage values obtained for trajectory tracking are too high
- The same motor control loop is not working for all links in the original planar snake simulation – clash between trajectory tracking and contact physics

Reference Tracking

- Essential for snake to be able to reach a specified point
- Need some sort of feedback for that
- Reading real-time positions of snake head:





- Defined a global variable *bias* as:

$$bias = \frac{arctan\left(\frac{y_{target} - y_{snake_{head}}}{x_{target} - x_{snake_{head}}}\right)}{4}$$

Reference Tracking - 2

- The variable *bias* cannot be directly fed to the links, so we store it first:

- To feed it to the links, we access it from a register:

- The bias is fed only to the first 4 links to avoid complete circular motion.

Reference Tracking - 3

- To avoid complete circular motion, we feed the bias to the links only for about $1/4^{th}$ of the simulation time.
- For the rest $3/4^{th}$, we give 0 bias so that snake moves in a straight line.

Future Work

- Study and implement state estimation for planar robot
- Combine motor velocity control and setpoint tracking for planar snake model
- Develop a feedback control law for planar snake model
- Enable contact physics between the links themselves

References

- **David Rollinson**. Control and Design of Snake Robots. *Carnegie Mellon University, June 2014.*
- **Tarun S**. Motion patterns and path convergence of snake robots on the lunar surface. *Indian Institute of Technology Bombay, June 2021.*

THANK YOU!