Quadruped Locomotion

Development of a neuro-inspired control system for quadrupeds to emulate sensorimotor processes in animals

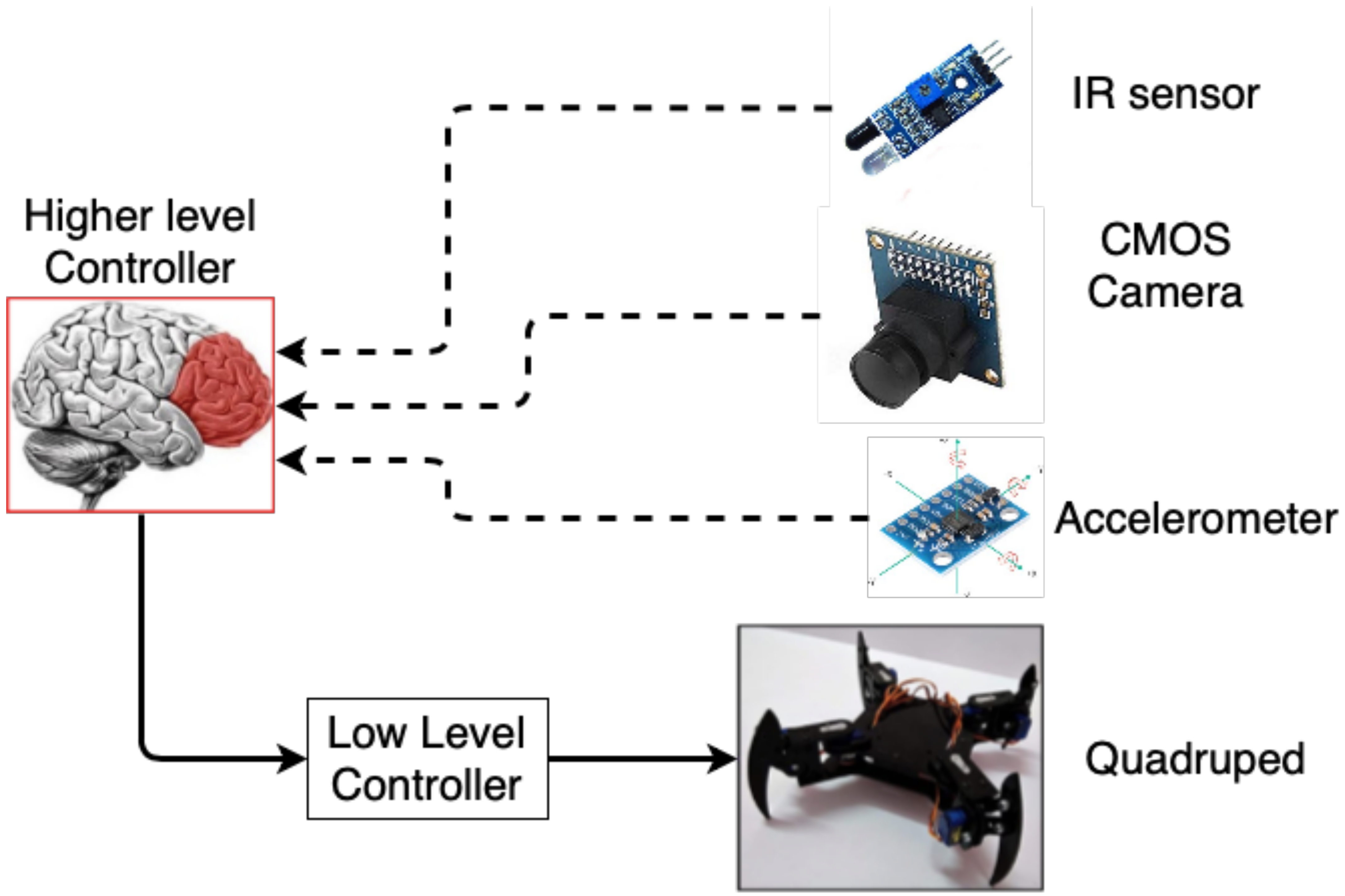
Shreyas Shandilya, Dipayan Biswas, V Srinivasa Chakravarthy

Introduction

Background

Animals are highly adept at locomotion and navigation under challenging terrains, capable of responding to a sudden stimulus with extraordinary agility and dexterity. They exhibit behaviour like gait switching, rapid acceleration and deceleration, evasive manoeuvres, climbing, jumping and search. The seamless transition between different behaviours and agile response to an incoming stimulus resulted from the evolution of neural pathways for adaptive locomotion, perception and embodied decision making. A model of such integrated sensorimotor processing can provide greater autonomy and deftness to robotic locomotion and navigation.

System Description



Methods and Materials

A Comparative Study

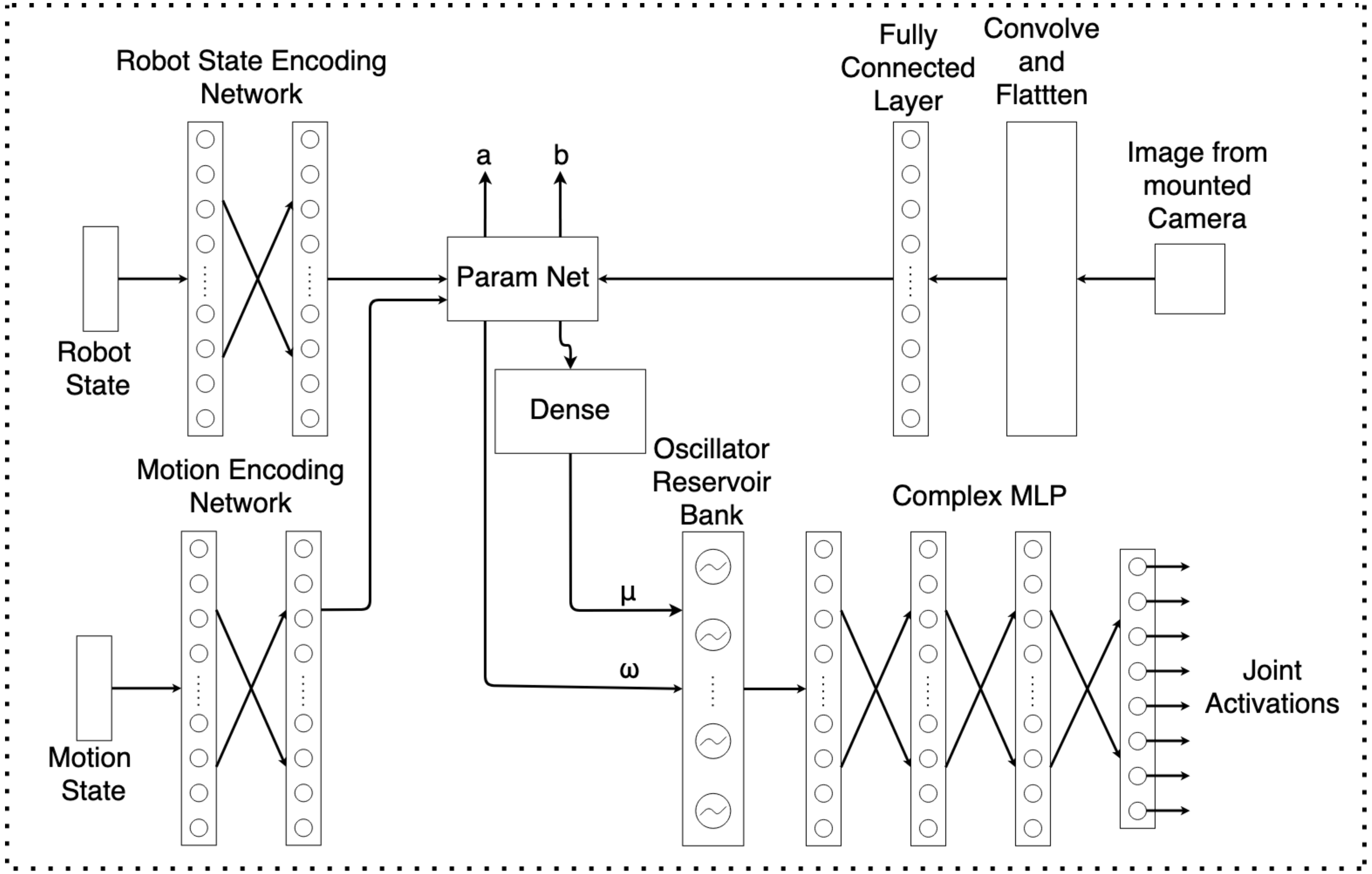
A Comparative Study between three approaches to low level control using neural networks was performed. The three studied architectures were as follows:

1. A feedforward DNN-CPG architecture

2. A fully connected feedforward architecture

3. A CPG architecture using Modified Hopf Oscillator

DNN-CPG Architecture



Methods and Materials

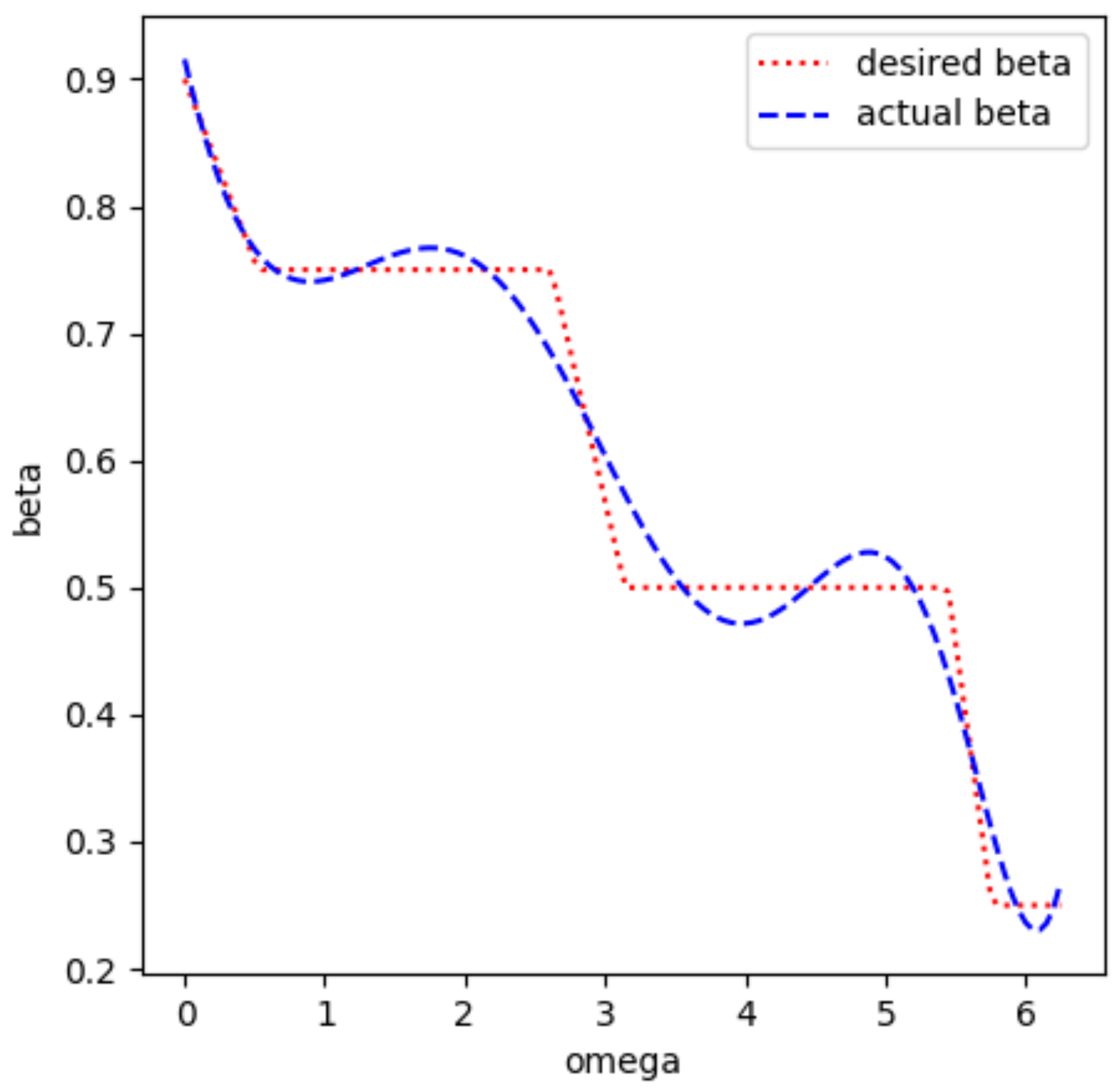
Modified Hopf Oscillator

A Hopf Oscillator with a switching frequency was formulated to directly produce joint activations.

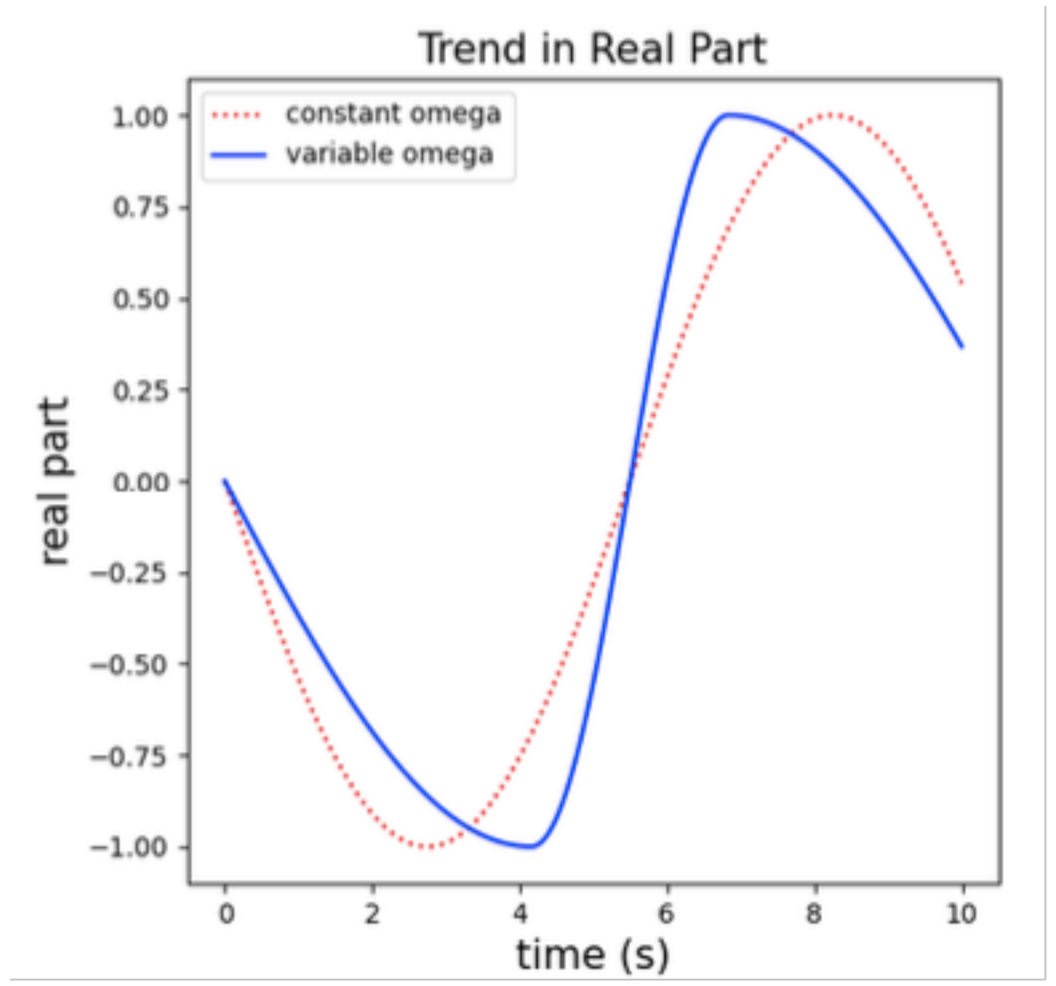
$$\dot{r} = (\mu_{\circ} - r^2)r$$

$$\dot{\phi} = \omega$$

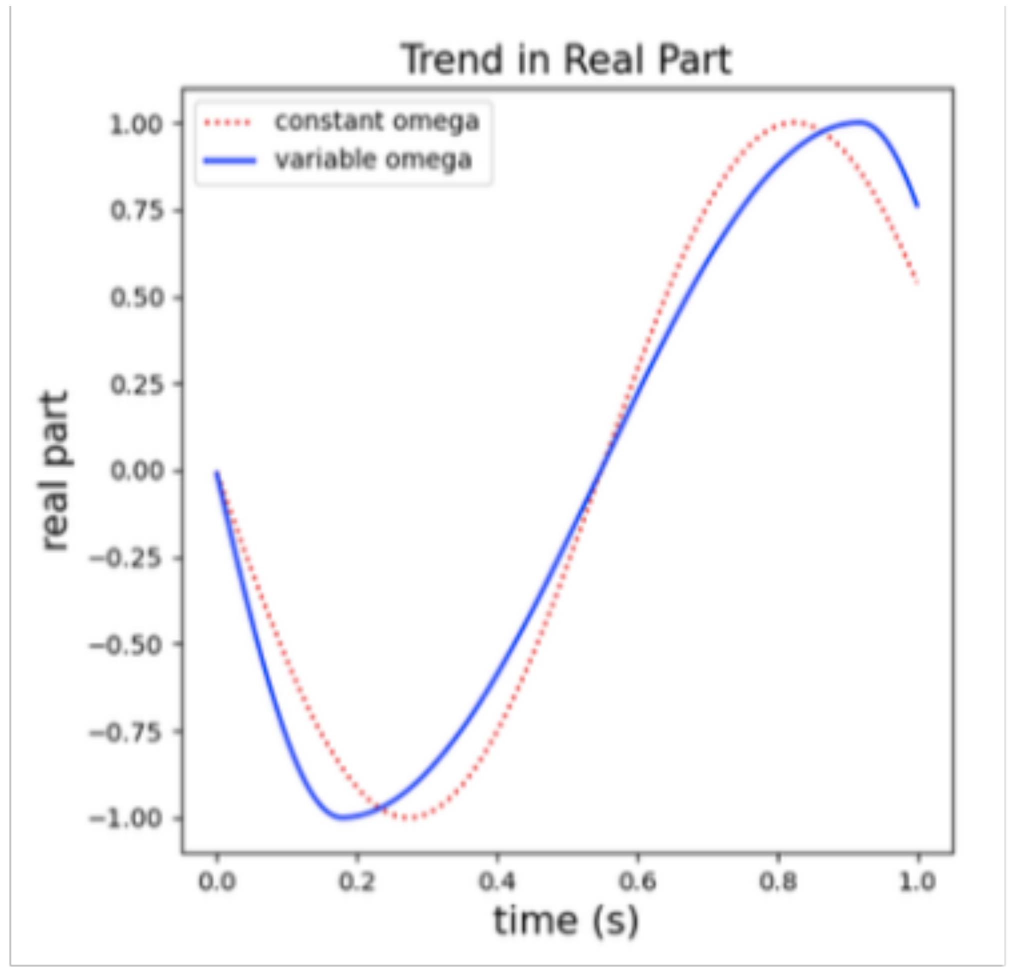
$$\beta = \sum_{i=0}^{i=N} a_i \left|\omega_{\circ}\right|^{i}$$



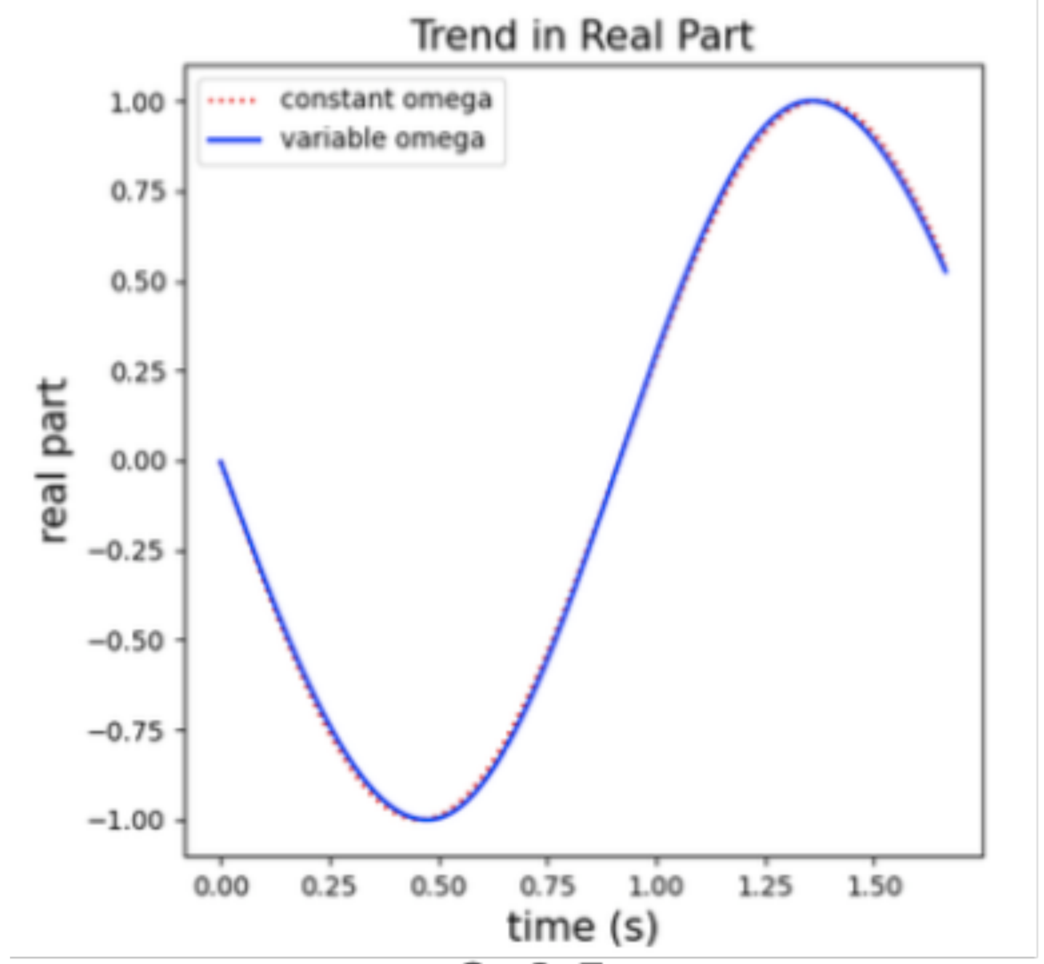
$$\omega = \left|\omega_{\circ}\right| \times \left(\left| \frac{1}{2\beta(1-\beta)} \right| + \frac{(1-2\beta)}{2\beta(1-\beta)} \times tanh\left(10^3\phi\right)\right)$$



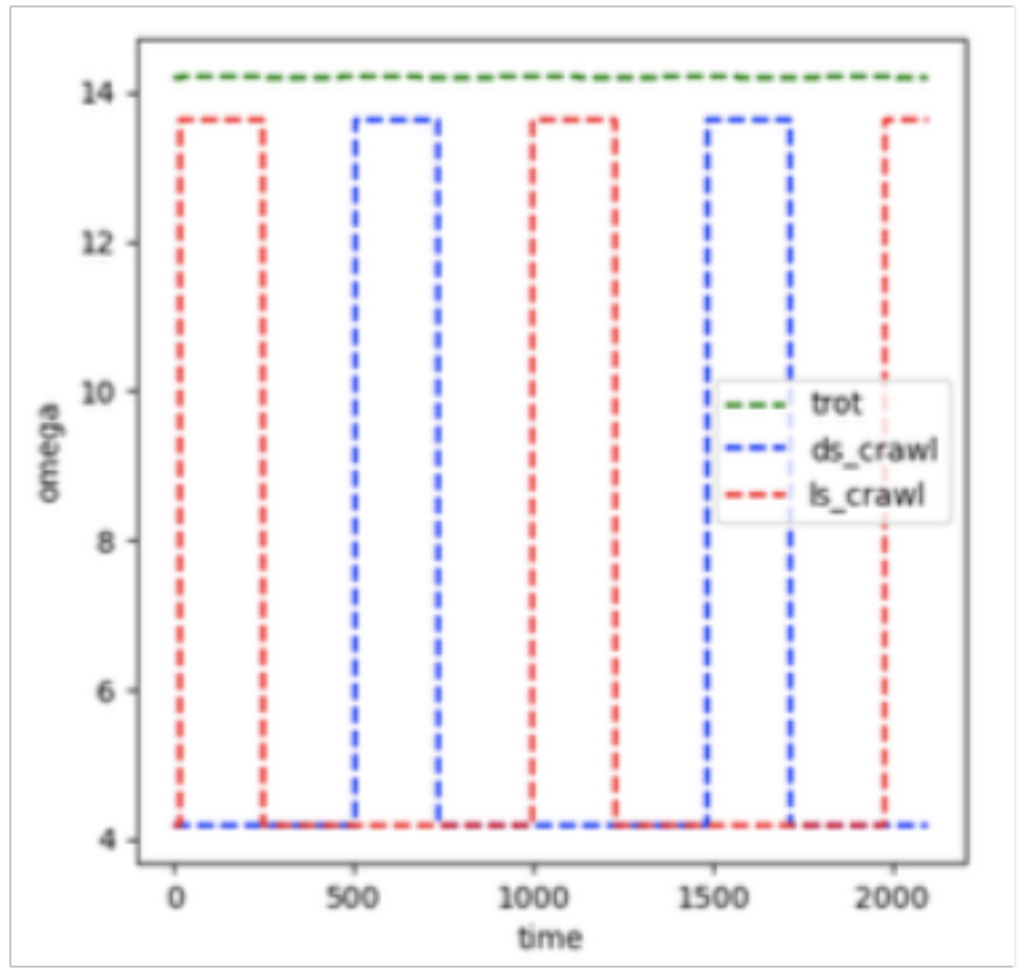
β=0.75



β=0.25



β=0.5



Frequency switching seen in different gaits

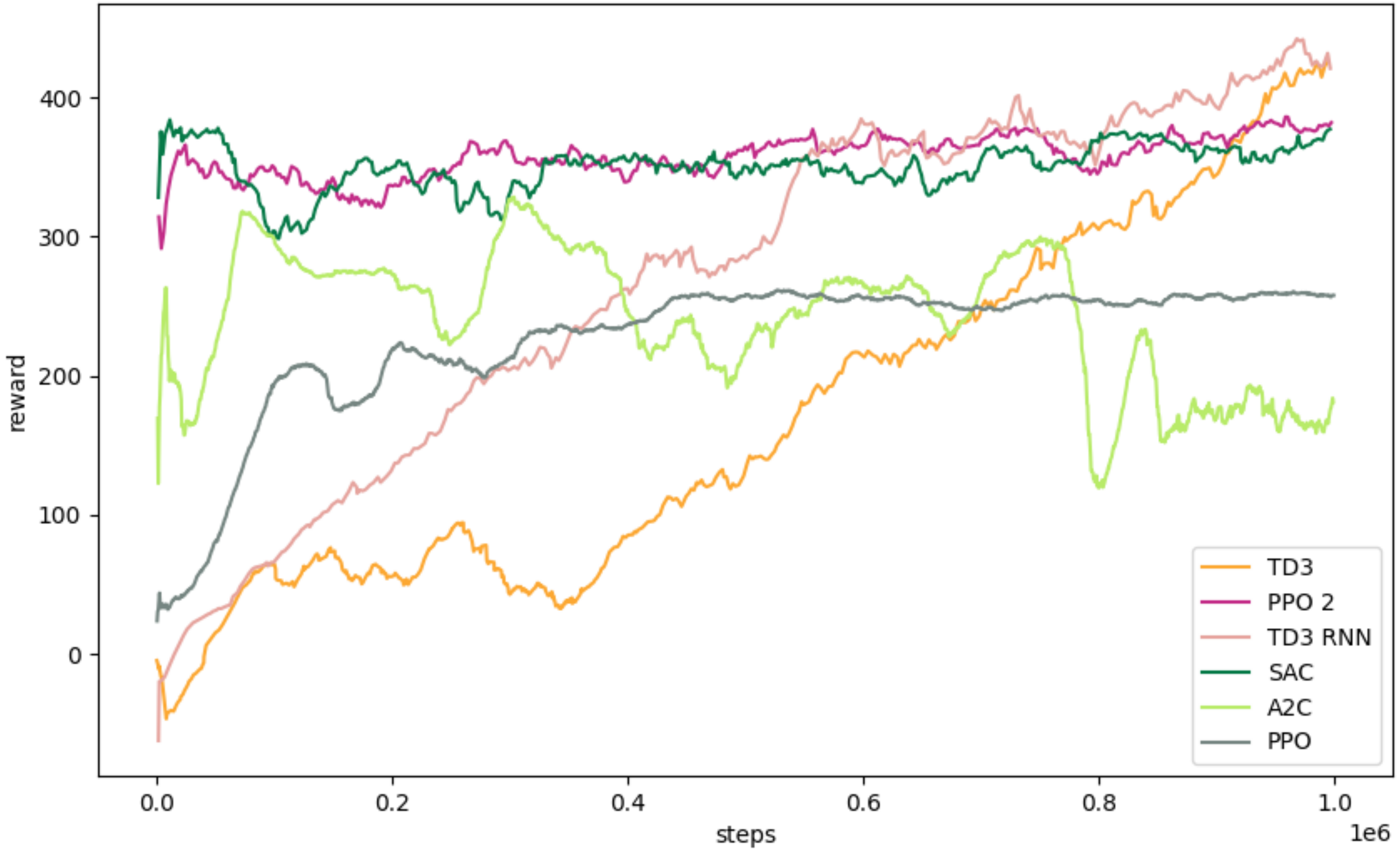
Results

Reinforcement Learning

- Reinforcement Learning Optimisation of Neural Network parameters was performed for the DNN-CPG architecture and the fully connected feedforward architecture

- Models failed to learn a meaningful sequence of actions to make the quadruped move in the desired directions

- Stochastic Algorithms learnt random movement, whereas Deterministic Algorithms found no movement to be the optimum solution.



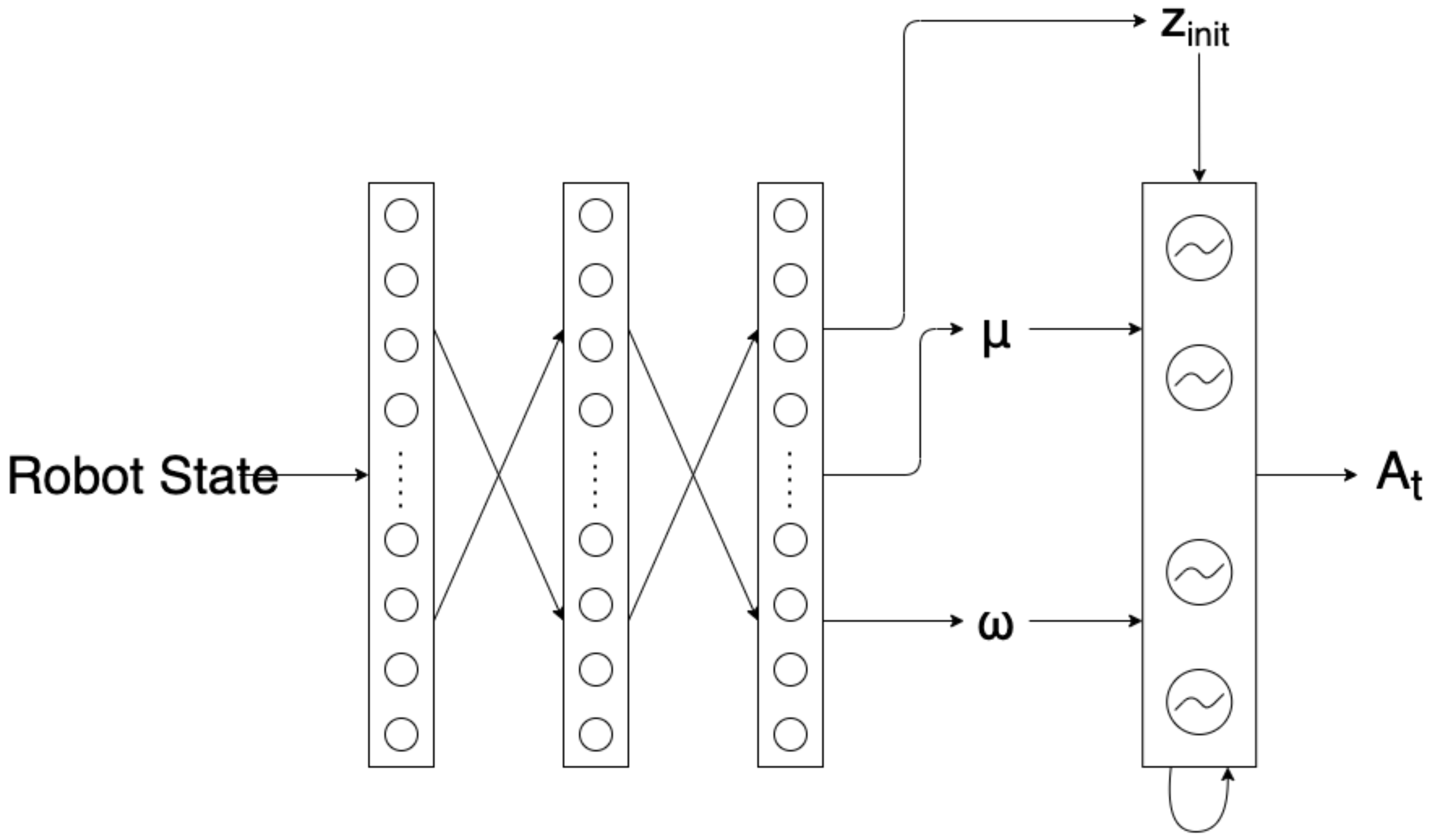
Methods and Materials

- Failure of the DNN-CPG and the fully connected feedforward architectures using Reinforcement Learning points to the need for restructuring the action space

- The Engineered solution using the CPG architecture exhibits complex behaviours such as variable speeds and gait transition by variation of the CPG parameters

- One to One mapping between the CPG parameters and the motion state of the quadruped can be observed and such a mapping can be easily learnt using Deep Learning/Reinforcement Learning using the shown architecture

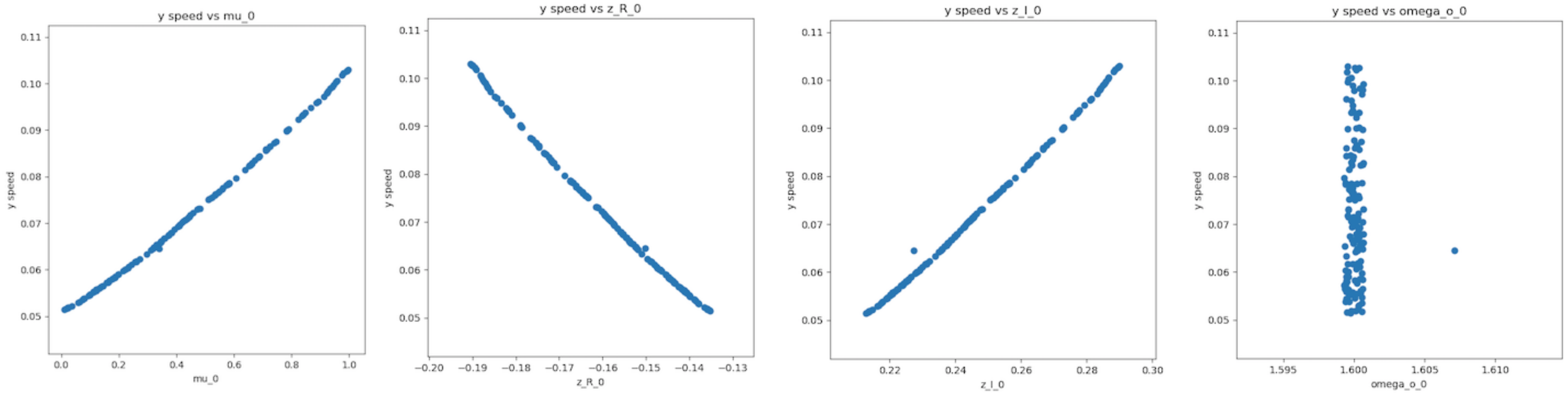
DNN-CPG with Modified Hopf Oscillators to be used as Low Level Controller



Methods and Materials

Modified Hopf Oscillator Parameter Space vs Robot Motion State

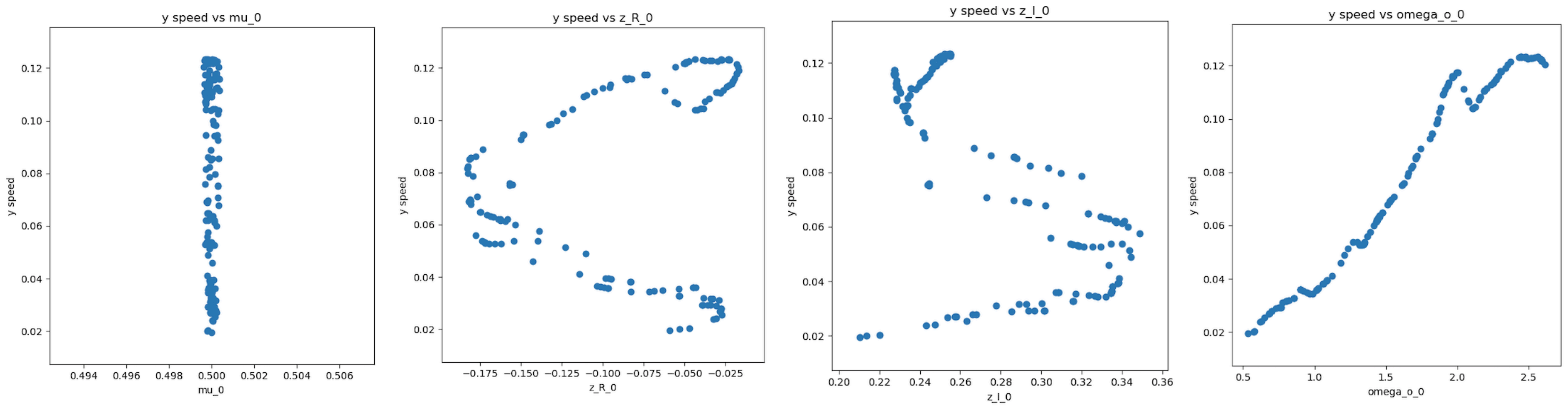
LS Crawl Gait Forward Direction Motion for constant ω_o



Methods and Materials

Modified Hopf Oscillator Parameter Space vs Robot Motion State

LS Crawl Gait Forward Direction Motion for constant μ_o Similar trends can be seen for other gaits and types of motion.



Results

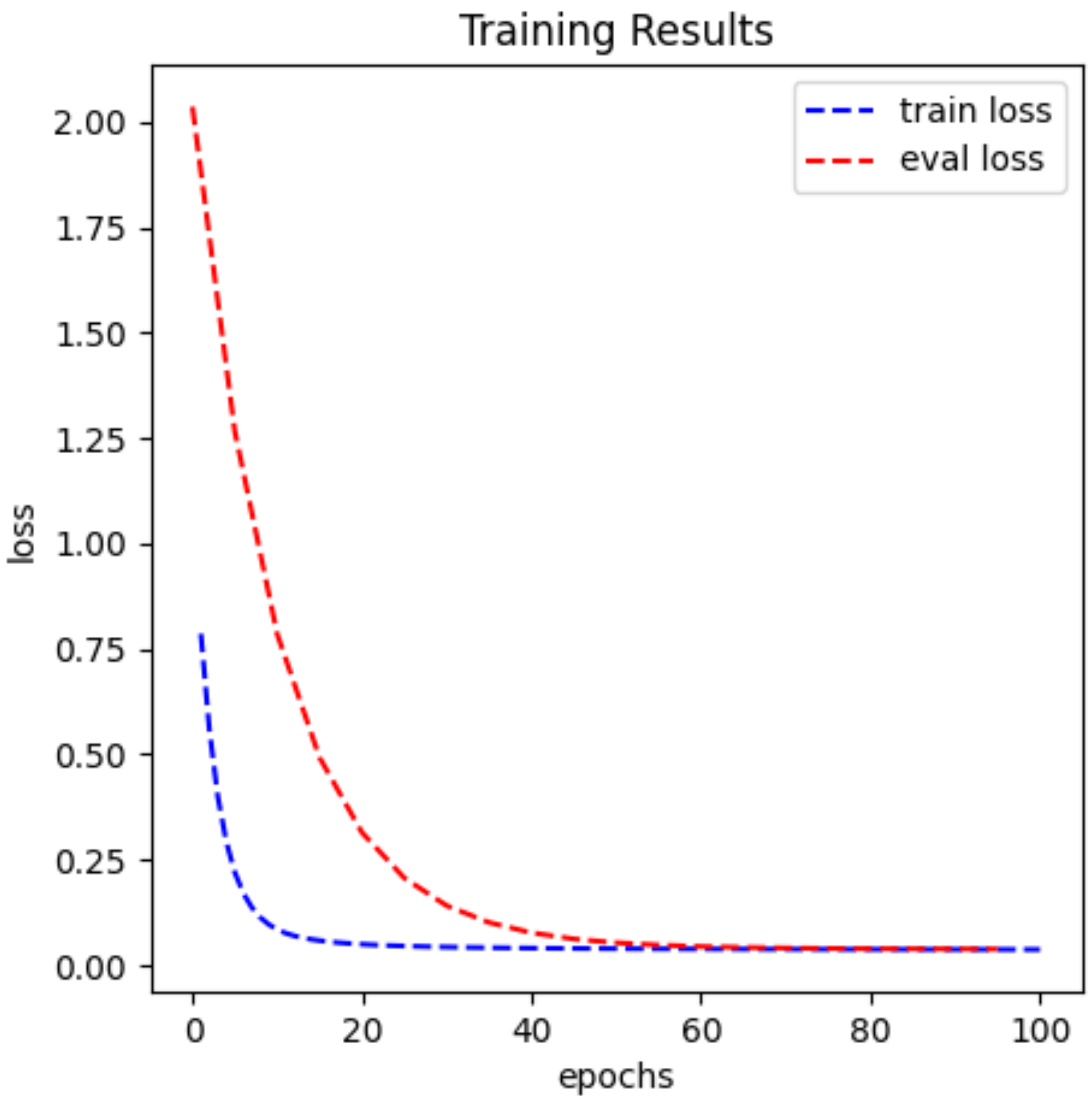
Deep Learning

- 11,000 episodes of locomotion at different speeds, different directions and types of motions were simulated to generate locomotion dataset

- Input data consists of 2,30,000 132 dimensional vectors obtained by sampling from the 11,000 episodes of locomotion simulation

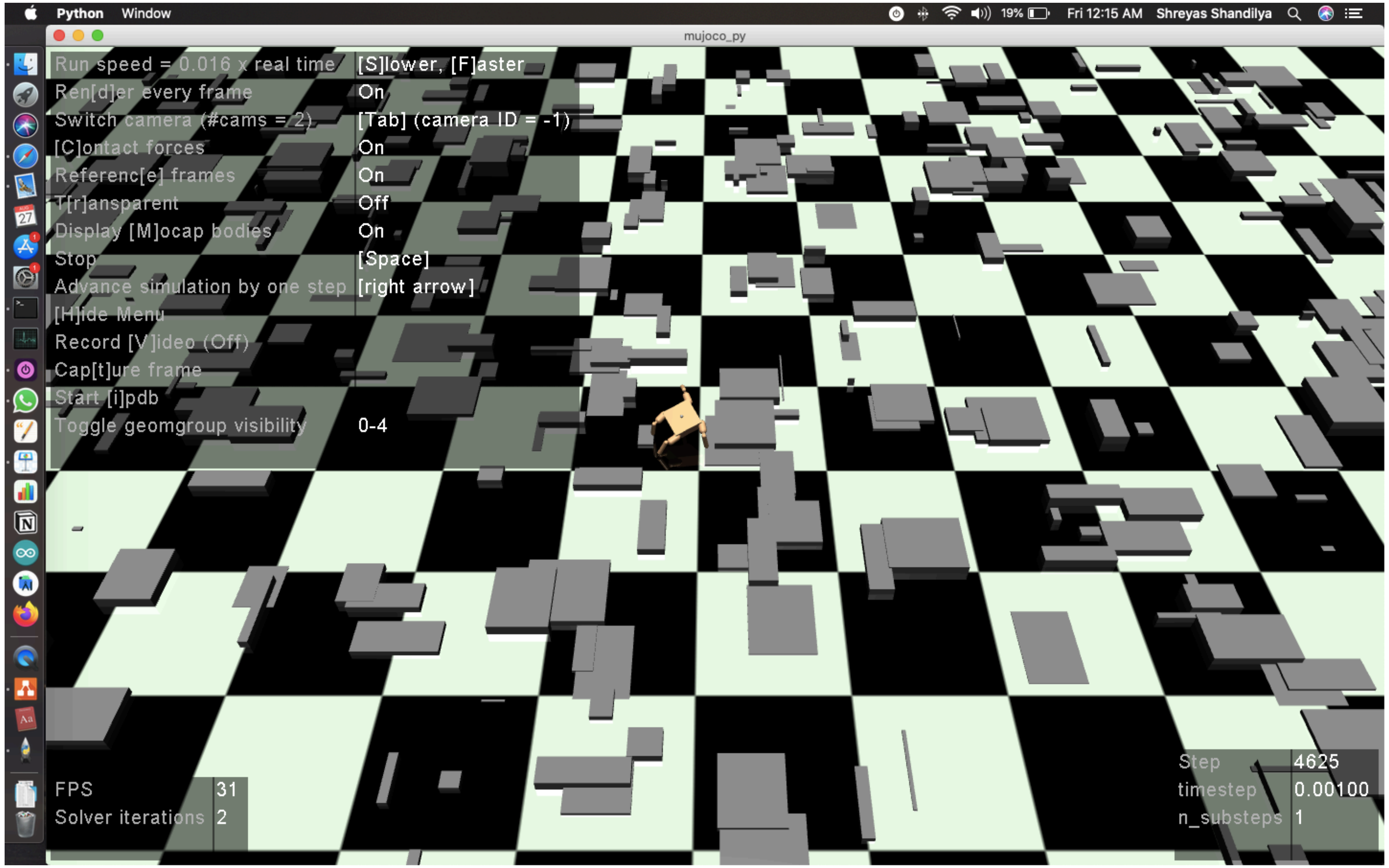
- The 132 dimensional vector consists of the desired motion state, the achieved motion state and 15 previous joint angle values

- The output labels consist of a 16 dimensional vector which include the $\omega_\circ$, $\mu_\circ$ and $z_\circ$ of the Modified Hopf Oscillator



Methods and Materials

Difficult Quadruped Locomotion Environment

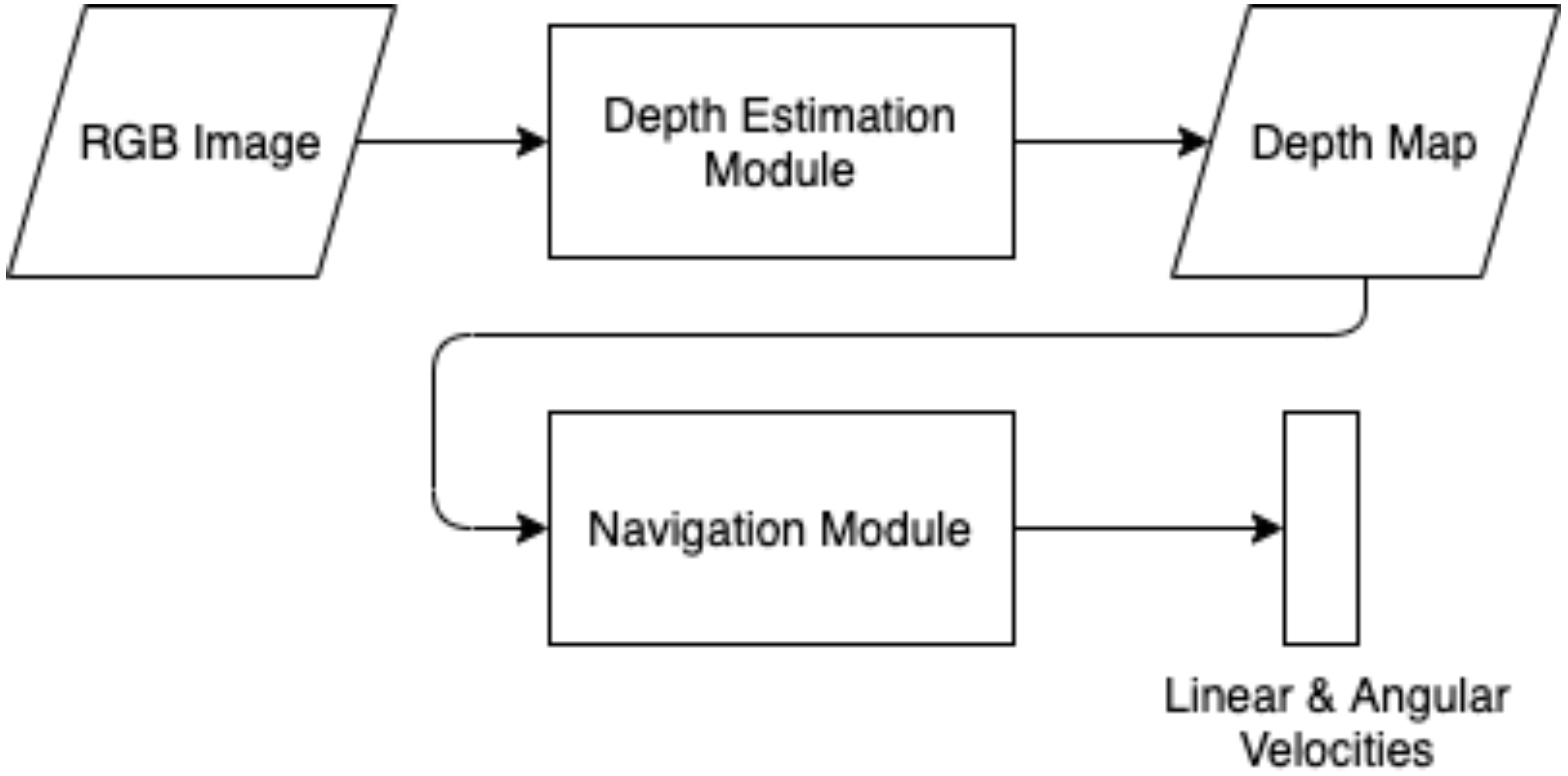


Methods and Materials

Obstacle Avoidance Module

- Obstacle Detection using Deep Reinforcement Learning makes use of Depth Maps from specialised RGBD cameras or a specialised module for RGB to RGBD transformation[1][2].

- The camera module available with the MePed Quadruped is only a RGB camera.



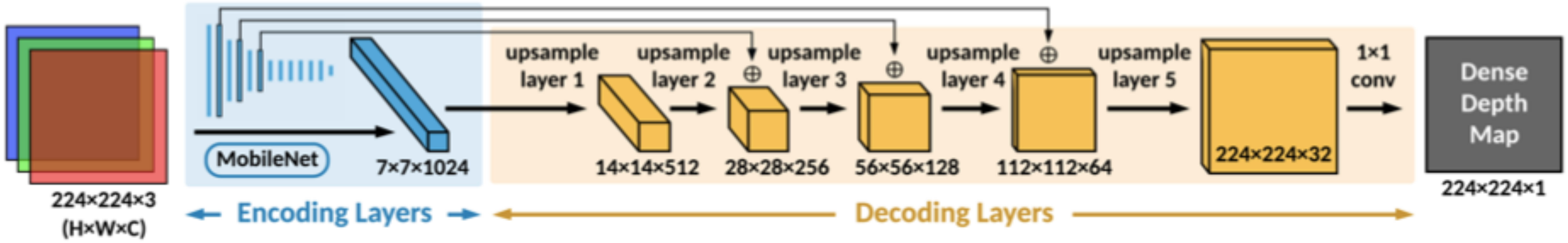
Methods and Materials

FastDepth: Monocular RGB Image to Depth Maps[3]

- The Auto-encoder architecture of FastDepth makes it suitable for usage in the subsumption architecture for autonomous control of the Quadruped

- The intermediate learnt representation from the encoder can be used for other visual tasks as well

- FastDepth is also suitable for embedded system architecture



References

1. Xie, L., Wang, S., Markham, A. and Trigoni, N., 2017. Towards monocular vision based obstacle avoidance through deep reinforcement learning. *arXiv preprint arXiv:1706.09829*.

2. Ramezani Dooraki, A. and Lee, D.J., 2018. An end-to-end deep reinforcement learning-based intelligent agent capable of autonomous exploration in unknown environments. *Sensors*, *18*(10), p.3575.

3. Wofk, D., Ma, F., Yang, T.J., Karaman, S. and Sze, V., 2019, May. Fastdepth: Fast monocular depth estimation on embedded systems. In *2019 International Conference on Robotics and Automation (ICRA)* (pp. 6101-6108). IEEE.