NEUROMORPHIC CONTROL OF QUADRUPEDAL ROBOTS

P. Raj, F. Forni, T. O'Leary
Control Group, Department of Engineering, University of Cambridge

UNIVERSITY OF CAMBRIDGE

Traditional Rhythmic Gaiting

- Slippage and deviation
  - Recovery can be difficult
- Terrain adaption
  - Swimming is especially difficult
  - Different control schemes are required
- Rigid definitions
  - Adapting a control scheme to different speeds, frequencies, etc. can be challenging

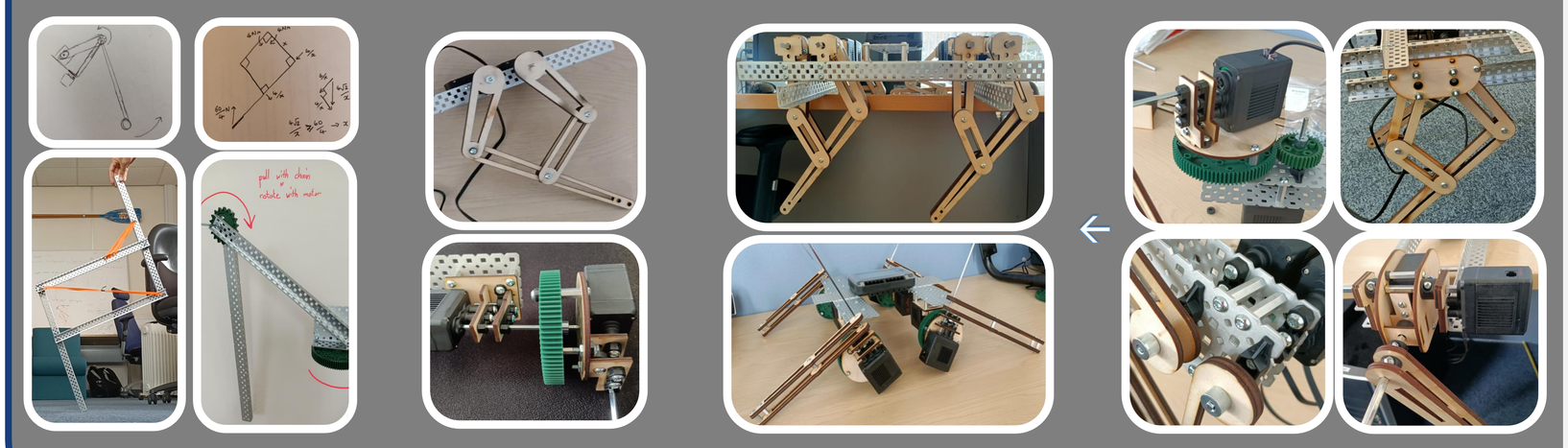
Bio-inspiration

- Central Pattern Generator
  - CPGs regulate repetitive locomotion in animals
- Robust recovery
  - Can swiftly reestablish patterns in the event of a disturbance/deviation
- Flexibility
  - Can adapt to different speeds and terrain with relative ease

Our Outputs

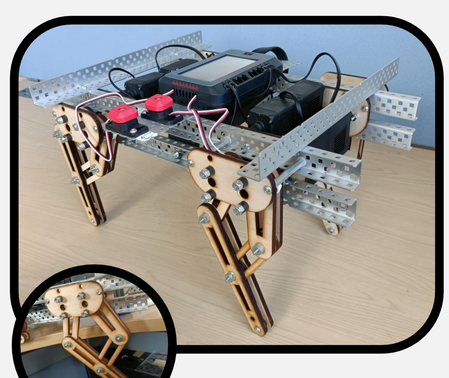
- Two Quadrupedal Robots
  - Synapider & NeuroPup were created
- Simulink Neurons
  - Computational neurons, CPGs, and oscillators were delivered for neuromorphic control of locomotion
- Parametrisation
  - Scripts for characterising neural output based on governing parameters were written

Two Robots to serve as a Platform for Research and Education



Design → *Realise* → *Prototype* → *Refine*

NeuroPup



Synapider

The MQIF Neuron as the Foundation for CPGs

Multi-Quadratic Integrate-and-Fire

- Sub-threshold ODEs

$$C\dot{V} = \bar{g}_f(V - V^0)^2 - \bar{g}_s(V_s - V_s^0)^2 - \bar{g}_{us}(V_{us} - V_{us}^0)^2 + I$$

$$\tau_s \dot{V}_s = V - V_s$$

$$\tau_{us} \dot{V}_{us} = V - V_{us}$$

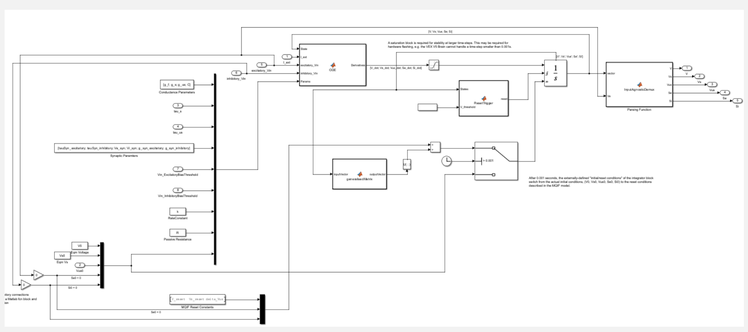
- Reset → 'Action Potential'

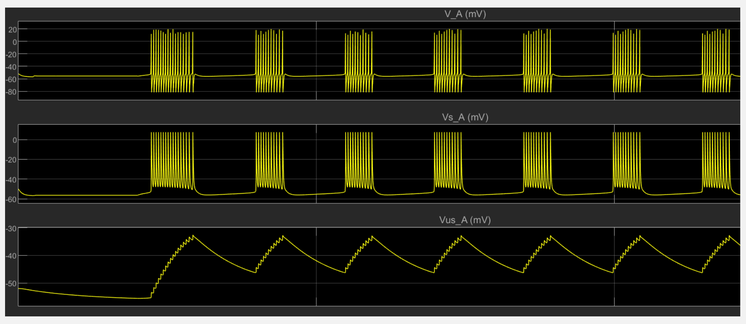
$$\text{if } V \geq V_{max}:$$

$$V \leftarrow V_r$$

$$V_s \leftarrow V_{s,r}$$

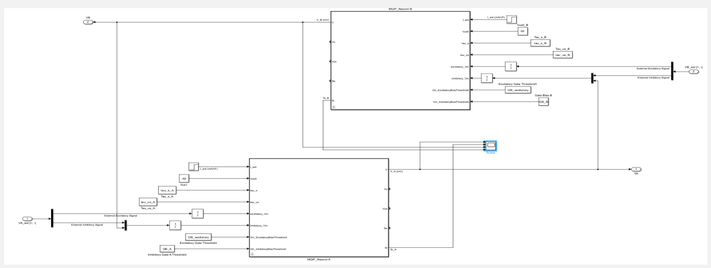
$$V_{us} \leftarrow V_{us} + \Delta V_{us}.$$

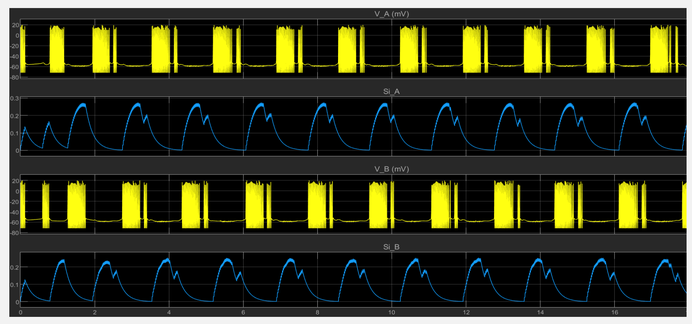




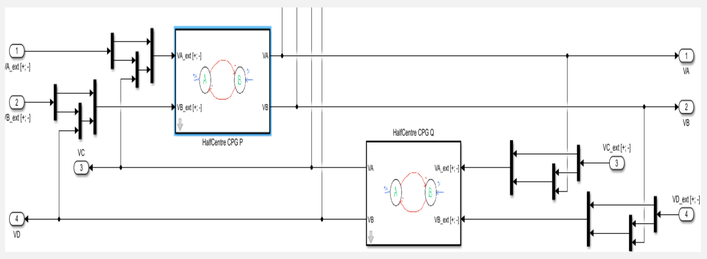
Generating Robotic Gaits with Derived Networks

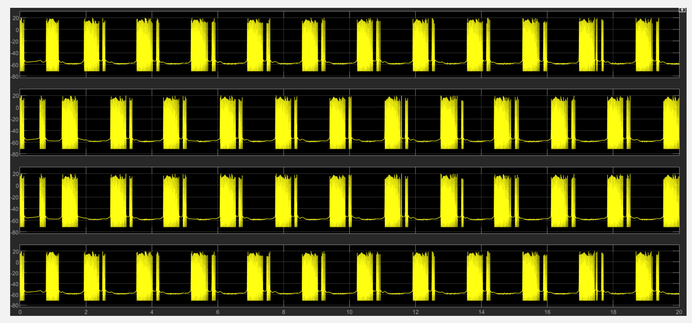
Half-Centre CPG



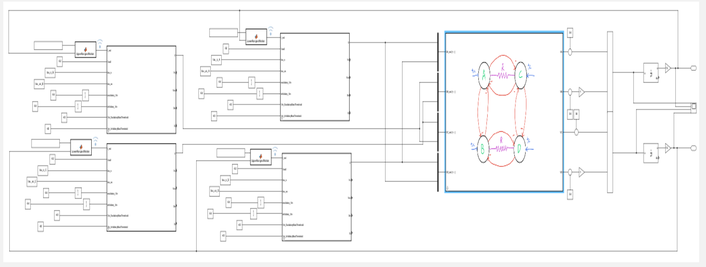


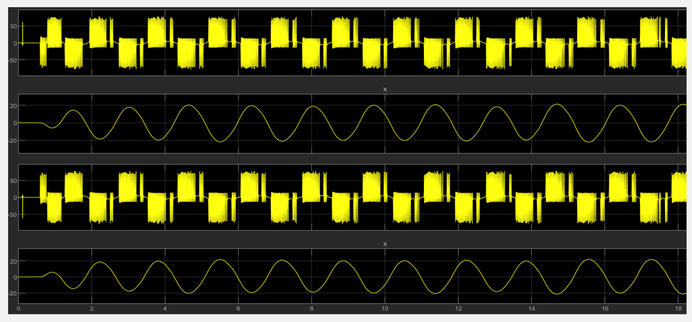
'Quad-Centre' CPG





'Quad-Centre' Oscillator





In collaboration with MathWorks®

MQIF Neuron Ref. Paper

T. Van Pottelbergh, G. Drion, R. Sepulchre, "Robust Modulation of Integrate-and-Fire Models," Neural Computation, 30(4), 987-1011, MIT Press, 2018. DOI: 10.1162/neco_a_01065