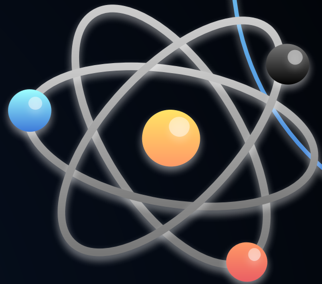
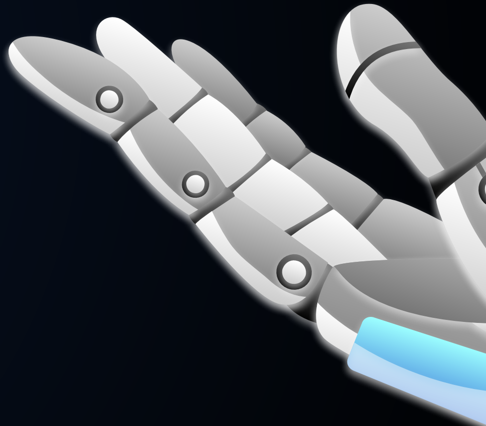
Laukik B Nakhwa

BITS Goa





About me

Education

- Bachelor of Engineering (Hons.) in Electronics and Instrumentation Engineering

- Master of Science (Hons.) in Biological Sciences

About me

Relevant Courses

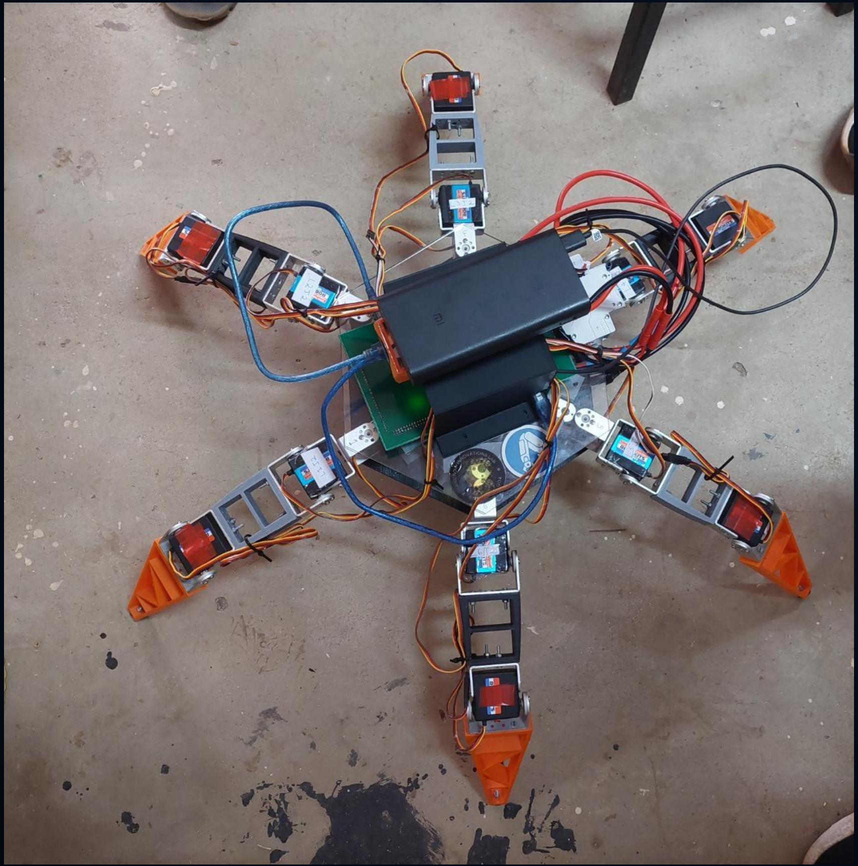
Data Structures and Algorithms

- Electrical Sciences
- Control Systems Engineering
- Modern Control Systems
- Microelectronic Circuits
- Analog and Digital VLSI Design
- Microprocessor and interfacing
- Signals and Systems

- Transducers and Measuring Systems
- Electronic instruments and instrumentation technology
- Mathematics I (Multivariable and Vector Calculus)
- Mathematics II (Linear Algebra and Complex Analysis)

- Mathematics III (Differential Equations)
- Probability and Statistics
- Machine Learning
- Biochemistry
- Molecular Biology
- Genetics
- Bioinformatics

PROJECT : SPIDERBOT



PROJECT : SPIDERBOT

Project Lead : Spiderbot, as a part of Electronics and Robotics Club, BITS Goa

- SpiderBot is a ROS based custom designed and manufactured symmetrical hexapod robot with each leg having 3DoF.
- Created and tested various gait patterns, including 3-3 tripod and 4-2 gait.
- Implemented a bio-inspired Central Pattern Generator (CPG) controller for smooth motion.
- Conducted precise tests for trotting, on-spot twisting, and more, both in the air and on the ground.
- My team and I secured 2nd position at the Open Design Contest conducted by The Department of Electrical and Electronics Engineering.

PROJECT : SPIDERBOT

Electronics :

List of Components used
Single pole MCB, 10A buck converter, 10amg wires, 35kg-cm servo (RKI 1202), 16kg-cm servo (RKI 1206), Lippo battery 4200mah, Raspberry pi 4,2 Arduino Mega

PCB design
Trace widths used: 188.89mils, 13.79mils, 6mils Components: Arduino, Male-Male Headers, XT-60 connector Current rating: 20A Copper weight: 2oz

PROJECT : SPIDERBOT

Software :

Have used ROS (Robot Operating system) framework. The programming language used is Python

Rosserial is used for communication between the raspi and arduino

Rosserial
Rosserial is a protocol to send data through a serial interface. In a client-server rosserial implementation, a rosserial-server is a computer running ROS and a rosserial-client is the microprocessor that receives sensors' data and transports it to the server in the form of ROS messages. rosserial-server in this implementation is a publishing node while rosserial-client is a subscriber node, although this can sometimes be the other way round.

PROJECT : SPIDERBOT

CPG on SpiderBot

- Bio inspired
- 6 Hopf oscillators which generate rhythmic signal gait
- Mapping function transforms signal to foot coordinates
- IK module computes angle of motion of joints

Why Hopf

- Matsuoka neuron oscillator model, the Van der Pol (VDP) relaxation oscillator, the Kuramoto phase oscillator, etc. The Hopf harmonic oscillator model has the advantages of high stability, few parameters, clear physical meaning, and easy tuning. (Li, Wei, & Qiu, 2023)

PROJECT : SPIDERBOT

- $\dot{x} = \alpha(\mu - x2 - y2)x - \omega y$
- $\dot{y} = \beta(\mu - x2 - y2)y + \omega x$, (Li, Wei, & Qiu, 2023)

where x, y are the output of the oscillator; α , β are convergence rate coefficients; μ is the square of the output signal amplitude; ω is the frequency of the oscillator

$$\begin{cases} \omega = \dfrac{\omega_{st}}{e^{-ay}+1} + \dfrac{\omega_{sw}}{e^{-ay}+1} \\ \omega_{st} = \dfrac{1-\varepsilon}{\varepsilon}\omega_{sw} \end{cases},$$

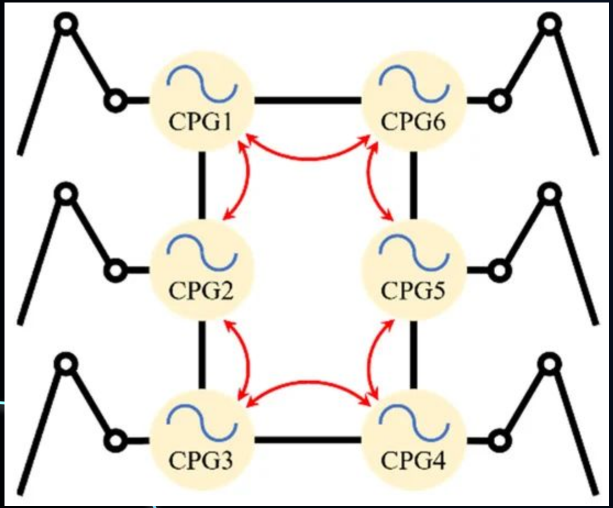
ε is the leg coverage coeff, wst stance phase freq, wsw swing phase freq

ε varies as per gait, Tripod ½ Tetrapod ⅔, Wave ⅚

PROJECT : SPIDERBOT

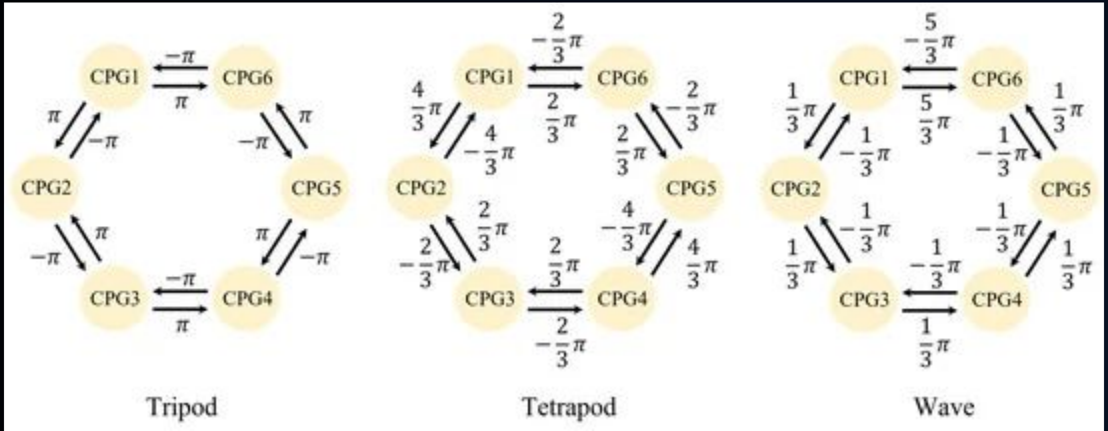
Each leg is controlled by one oscillator unit, and bidirectional coupling of neighbouring oscillator units.

Lambda : coupling strength, coupling term Δij
Theta ij : phase difference between oscillators



$$\begin{cases} \dot{x}_i = \alpha\left(\mu - x_i^2 - y_i^2\right)x_i - \omega_i y_i \\ \dot{y}_i = \beta\left(\mu - x_i^2 - y_i^2\right)y_i + \omega_i x_i + \lambda \sum_j \Delta_{ji} \\ x_i = \left(x - \mathrm{offset}_{x_i}\right) \\ y_i = \left(y - \mathrm{offset}_{y_i}\right) \end{cases},$$

PROJECT : SPIDERBOT



$$\Delta_{ji} = \begin{cases} (y_j \cos\theta_{ji} - x_j \sin\theta_{ji}) & |j-i| \le 1; i,j = 1,2,\ldots,6 \\ 0 & |j-i| > 1; i,j = 1,2,\ldots,6 \end{cases},$$

Mapping Function : maps output signal of oscillator to control signal of joints

fix, fiy,fiz : initial positions of the foot
k0, k1,k2 : scaling coefficients
Phi : direction of stride

PROJECT : SPIDERBOT

Code :

Keyframe-based CPG for Stable Gait Design and Online Transitions in Legged Robots

- Keyframe based approach : each gait descritized to check points must be passed to complete a gait cycle.
- Kuramoto model based : leg phase difference coupling
- Reactive Controller : Steady forward progression while transition between gaits(ensure stability by updating body's position and limiting leg usage)
- Better than global planning (expensive)
-

Fundamental, what signals, what angles, why cpg, what it gives, what converge etc, check paper references, the process, and kuramoto fnd out