Hexapod Walking Gait Evolution

Authors: Bastian Busch & Fraser Borrett

Introduction

This project investigated evolving robotic controllers for a Hexapod robot with 3 DOF legs using two controllers: an artificial neural network (ANN) and lookup table (LUT). Both controllers were evolved using the same genetic algorithm to perform the same task. The task these controllers are being evolved for is getting a hexapod to walk forward one meter in a straight line.

Goals

- Evolve a gait for the LUT and ANN via simulation so robot walks one meter in a straight line.
- Compare the performance of the LUT and ANN for gait evolution.
- Design schematics and PCB for the hexapod's controller interfacing to the DE0 Nano FPGA.
- Create a GUI running on a mobile phone to control the robot.
- Create a NIOS II softcore CPU and write Verilog and C++ software for the controller.
- Implement evolved gaits into the hardware robot.

Outcomes

- Physical hardware created for the hexapod with a custom-built controller programmed with standard gait methods, controllable via a mobile phone.
- Both the LUT and ANN controllers were successfully evolved in simulation.
- Both the evolved LUT and ANN controllers were able to control the physical robot.

Evolution

Genetic algorithm:

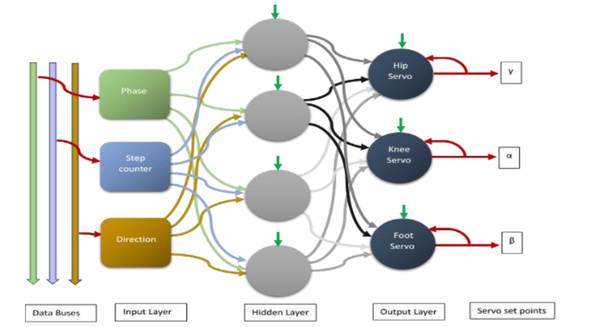
- A genetic algorithm was created and used to evolve two walking gait controllers for a hexapod with 3DOF legs. The fitness is based on the desired leg motion for an optimal gait.
- The first controller is a nx3 lookup table containing the servo angles for a leg, for each step in the gait. The LUT of joint set positions is the chromosome being evolved by the GA to achieve an optimal walking gait.
- The second controller is an ANN, a table of the neural networks weights and bias is the chromosome being evolved.

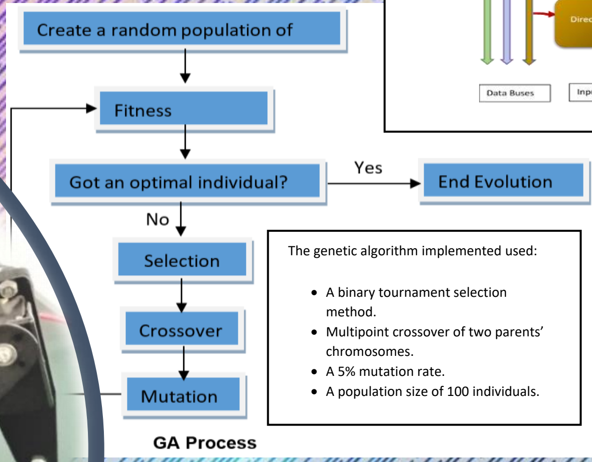
ANN Description:

The neural network developed has 7 neurons and uses a feedforward architecture with an input layer, a single hidden layer and an output layer.

- The inputs are a phase signal (- ground/+ lift), heading and a step counter.
- Outputs are three set positions, one for each joint in the leg of a hexapod.

This architecture is not completely feedforward as the joint set positions from t_{step-1} are fed back into their respective output layer neurons.



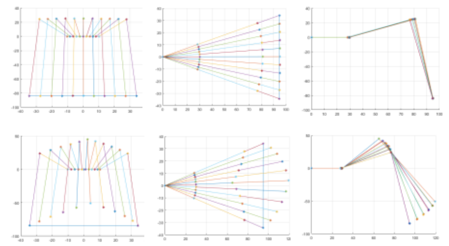


GA Process

The genetic algorithm implemented used:

- A binary tournament selection method.
- Multipoint crossover of two parents' chromosomes.
- A 5% mutation rate.
- A population size of 100 individuals.

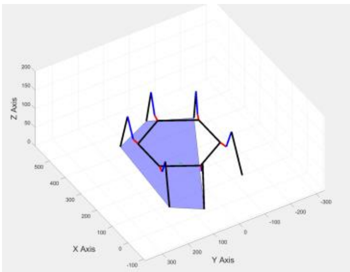
Evolved Motion

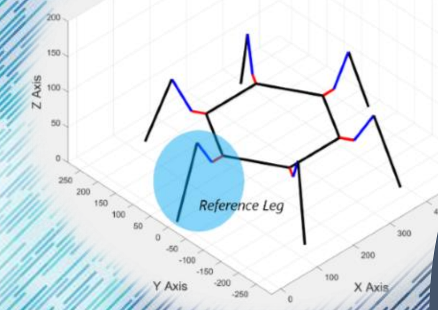


Mathematical Model

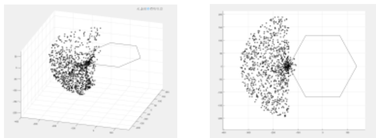
A Mathematical Model of the physical hexapod was designed in MATLAB to simulate the movement of the physical hexapod. Trigonometric path analysis was used to derive angular directional motion.

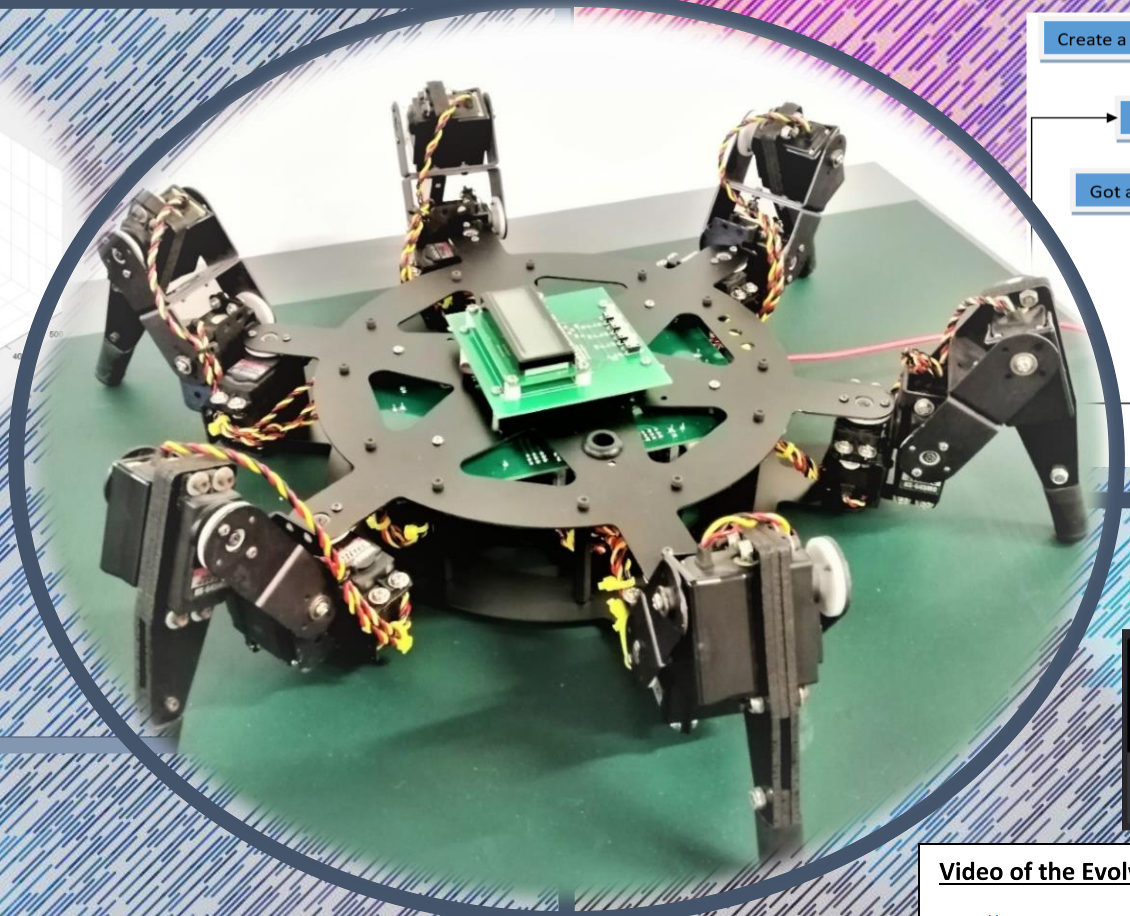
The MATLAB simulation animated the moving hexapod on the ground plane with contact points, stability analysis as well as simulation complex coordinated leg sequences.





The full leg motion range of each leg was derived numerically in MATLAB.





Physical Hardware

Hardware:

The hexapod chassis used is a Lynx-motion BH3-R, this robot came with 18 HS-485 servo motors. A PCB needed to be designed with allowing the hardware to interface to a DE0-Nano board

- PWM drivers and a power supply for the 18 servos.
- A communication device for external control.
- Onboard feedback for users.
- A processor to control all the peripherals.

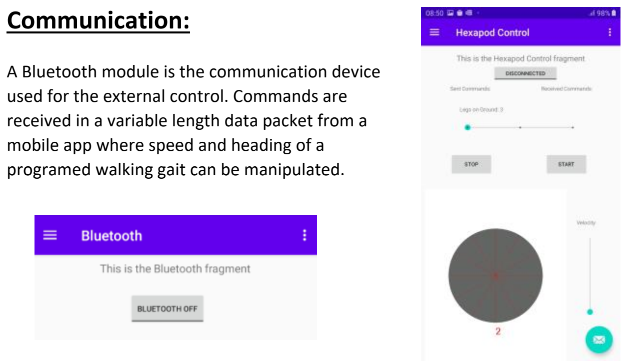
Two PCBs were designed, the controller PCB for the processor and its associated peripherals, and a display board consisting of an 2x16 LCD and momentary push buttons.

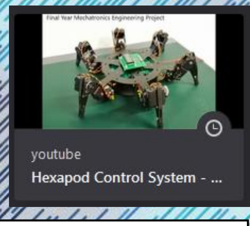
Controller PCB:

For the processor a DE0 nano development board is used, this board uses an Altera Cyclone® IV FPGA. Using this board allowed for programming in hardware languages to create the servo PWM drivers. A NIOS ii based system was put onto the FPGA to program the hexapod in C. The NIOS ii is a 32-bit processor and is what controls all the peripherals on the controller PCB.

Communication:

A Bluetooth module is the communication device used for the external control. Commands are received in a variable length data packet from a mobile app where speed and heading of a programed walking gait can be manipulated.





Video of the Evolved Walking Gaits:

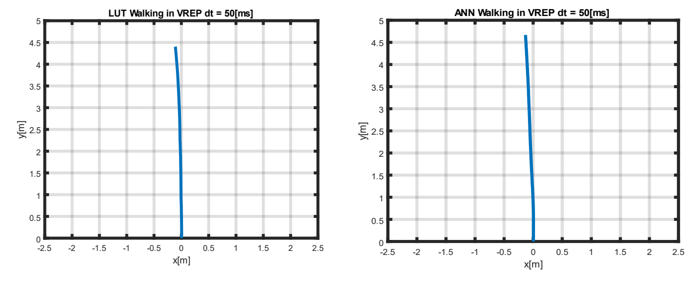
https://www.youtube.com/watch?v=fJ4JiNA1nbg

Results

Evolved walking gaits:

The trajectory data for the evolved gaits comes from the controller's implementation in VREP, VREP is a robotics simulator by COPPELIA Robotics. Both results come from evolutions where the achieved fitness was less than 0.4(a fitness value of zero is best in this evolution) and were stepped through 30 times to get the trajectories shown.

- The hexapod is started at the position y=0 [m], x= 0 [m].
- The evolved ANN walked the hexapod to a final position of y=4.67 [m] and x=-0.13[m].
- The evolved LUT walked the hexapod to a final position of y=4.40 [m] and x=-0.11[m].



Evolution Results:

Multiple evolution tests were conducted to obtain data to compare the evolution of the two controllers.

- In each test 100 evolutions were attempted, using the GA as described above.
- Tests were conducted using a binary tournament selection as well as an Elitist approach.
- The diagrams below show how the number of generations required to successfully evolve a walking gait varied for each of the tests.

The averages of the Box and whisker diagrams show how well the ANN evolution went compared to the lookup table evolution. The evolution of the LUT took on average 10.46 times the number of generations the ANN required to achieve the desired fitness level.