**Project Title:** Improving Robotic Arm performance by using an FPGA For Control

**Proposer:** *Self*

**Supervisor:** *Nicholas Outram*

**Objective**

* The Objective of this project is to be able to control a robotic arm using a Field Programmable Gate Array as the main controller.
* The ultimate goal of this project is to use the FPGA's potential for parallel processing to quickly process information and provide instructions to an arm while reducing programming related delays.
* This Project has academic merit in helping me develop my FPGA skills as it is a field I hope to work in. It will also hopefully serve as the initial proof of concept for a system that can autonomously detect and correct varied errors on production lines with no human input.

**Staged Deliverables *(at least 4)***

* *What constitutes a PASS (40%+)*
* The arm can be controlled by the FPGA, inverse kinematics might be estimated using machine learning or Q-learning might be completed but neither are fully implemented in the project.
* *What constitutes a 2:2 Mark (50%+)*
* The arm can be controlled by the FPGA, either inverse kinematics or Q learning have been implemented and the arm can carry out simple actions with limited input.
* *What constitutes a 2:1 Mark (60%+)*
* Q learning and inverse kinematics have both been implemented and the arm can autonomously complete a preset maze using a preprogrammed transition matrix, ideally multiple mazes can be switched between using switches on the FPGA.
* *What constitutes a FIRST CLASS Mark (70%+)*
* A camera has been added and the controller is capable of receiving a picture of the maze and using it to create a map of the maze. The arm should now be able to move through any maze put in front of it with no preprogrammed transition matrix.

**Description:**

***Describe the project in non-technical terms. This should contain detailed actions of how your aims and objectives will be achieved. Text should be augmented with Block Diagrams*.**

* To start with inverse kinematics will be calculated using machine learning, this will be done by first using forward kinematics to calculate the endpoint location from randomly generated angle values. The random angles and calculated endpoint locations will then respectively be used as the target and input for a multi-layer perceptron which will generate weights that could be used to find the angles which will move the end point to any specified position. This would most likely be completed using MATLAB.
* A camera placed above the maze would take a phot and send the image’s colour data to the FPGA or a microcontroller if the FPGA can’t process the information or has insufficient spare registers, this information will then be used to map out where the walls on the maze are.
* The microcontroller or if possible the FPGA will then create a transition matrix for the maze and perform Q-learning to find the optimal path through the maze, creating a Q-table.
* The FPGA will then use the Q-table to find the next point on the path through the maze and the inverse kinematics weights to calculate the servo angles needed to move the arm there.
* The FPGA will then send a PWM signal to the arm’s servos, moving the arm.
* This project would then be proof of concept for being able to use visual data and machine learning to identify a target and quickly navigate an arm through new environments with minimal delays and no human input.

A diagram of a computer system

Description automatically generated

**Resources**

***Equipment***

* *DEO-CV- Development board using Altera Cyclone V FPGA Device – Booked out of Smeaton Store*
* *Desktop Robotic Arm - Booked out of Smeaton Store*
* *An as of yet unchosen microcontroller may be required*
* *An as of yet unchosen camera will be required*

***Locations***

* Desk space in the lab will be required for the final demonstration however most of the work only needs to be done in the lab when technical support is required.

***Support & outside services***

* *Minimal 3D printing may be required to build the mazes, however the mazes will work almost as well as just a drawing or printout on a sheet of paper.*

**Schedule**

I expect that each staged deliverable should take roughly 2-4 weeks depending on how much unrelated work I need to do and how many unexpected issues I have to deal with.  
I also expect that the first staged deliverable will take much longer than the others as I spend time solving any problems as they crop up.

Stage 1 – Complete 19th January

Stage 2 – Complete 11th march

Stage 3 – Complete 1st April

Stage 4 – Complete 22nd march

This schedule gives a generous amount of time for finishing each stage while leaving a reasonable amount of time at the end to prepare for the Vivas/project showcase.

**Related Documents**

***Data Sheets***

* *Found on the webpages.*

***Web Pages***

* <http://wiki.sainsmart.com/index.php/6-Axis_Desktop_Robotic_Arm,_Assembled>
* <https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=163&No=921&PartNo=4#contents>

***Books***

* *None so far*