BIRMINGHAM CITY University

Quick Sensor Pattern Recognition for Drone Control using a Radial Basis Function Neural Network

BSc Computer Science
Ahmad Najiy Wahab
S14120038
Supervisor Abdel-Rahman Tawil

1 Abstract

As drones are gradually becoming more autonomous with more inherent intelligent capabilities it is inevitable that we find an effective way of interacting with these imminent intelligent machines.

2 Motivation

Traditional thumb-stick radio controllers are very limited in what they can transmit – limited by the number of channels for each input variable mechanism. Traditional thumb-stick control is ideal for the case that the user needs full control over drones but not for such cases where drones are made to be autonomous.

The use of image recognition is theoretically inferior against the proposed neural network if such network can recognise accelerometer and gyrometer data as gestures more accurately. There would be less latency between consecutive iterative process because there is no pre-processing involved like in that of image recognition.

Furthermore image sensors must be placed locally near the user therefore does not allow the user much freedom of movement together with interaction. With the gyro-accelerometer recognition network the user will have much more freedom to move to places as the sensor is bound to the user.

It would benefit to replace traditional control mechanisms with one that provides a more pragmatic interaction with more intelligent drones.

3 Aim & Objectives

Aim:

The aim of this project is to develop a variant of a neural network known as the radial basis function network (RBFN), and to optimise such network specifically for sensor pattern recognition to control drones.

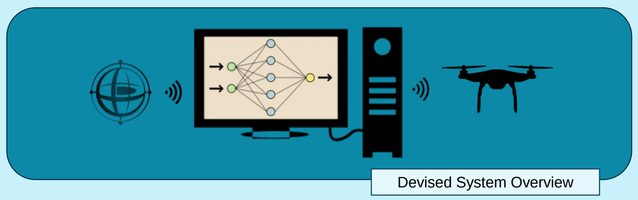
Objectives:

Data Capture Hardware Implementations:
1. Connect sensor components together (MPU6050s) for I2C bus configuration.
2. Connect sensors to an Arduino (Pro Mini) programmable development board.

Sensor Data Capture Arduino Software Implementations:
3. Core procedure to fetch sensor data into a byte array.
4. Normalisation procedure to normalise the array prior transmission.
5. Capability to transmit the array.
6. Make whole procedure iterative.

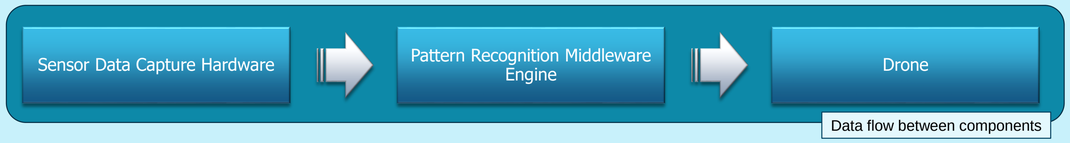
Pattern Recognition Middleware Engine Implementations:
7. Radial basis function neural network for the middleware engine.
8. Byte-array queueing system for neural feed.
9. Pulse mechanism that update feeds and fires network.
10. Event trigger bindings for output nodes.
11. Few common operating system resource bindings for testing.
12. COM port trigger binding for communications via COM ports.
13. Socket trigger bindings for communication via socket.

System Solution Approach



Devised System Overview

4 System Data Flow



Data flow between components

The Sensor Hardware captures the accelerometer and gyrometer data and normalises into byte-array.

The Sensor Hardware then transmit the byte array to the Middleware Engine for processing.

The Middleware Engine queues array received and dequeues the oldest array.

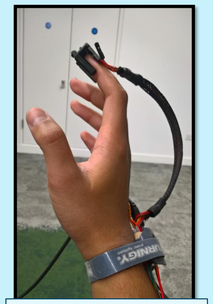
A FIFO queue system provides a stacked stream that captures samples over time.

When the output values passes a given threshold the Middleware Engine sends control data to Drone.

The Drone receives control data transmitted from the Middleware Engine and queue data as instructions.

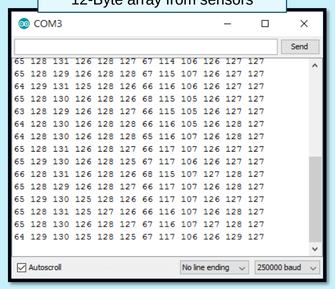
The drone acts accordingly to queued instructions.

5 Capturing MPU6050 Data



Sensors at the tip of finger and on the wrist

12-Byte array from sensors



The Arduino extracts the accelerometer and gyrometer data from the MPU6050s then normalizes the data of the raw values into an array of bytes prior transmission.

The value represents the orientation and the magnitude of referential acceleration taken from the accelerometer and gyrometer respectively.

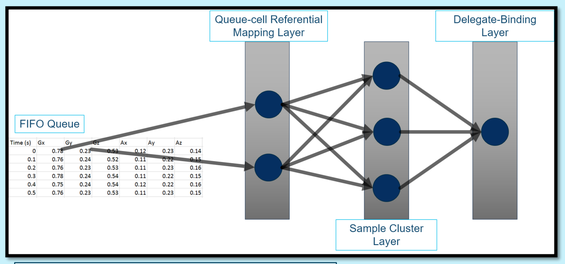
The sensor component then transmits the array of bytes to the Middleware Engine for pattern recognition processing with the RBF network.

6 The RBF Network Infrastructure

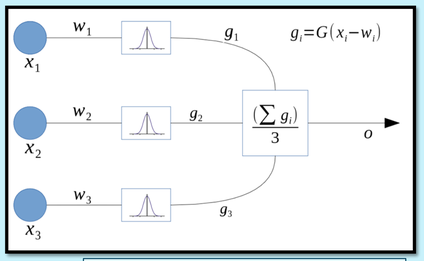
The Middleware receives and appends the array to the end of the feed queue and dequeues the top array.

Input layer nodes are referentially bound to each queue-array cell. Each row of the queue serves as a portion of the sample – the whole queue is one sample.

The delegate layer consists of nodes that are mutually exclusive - each represents one distinct gesture. The nodes propagates the highest value from layer beneath.



Referential value mapping from FIFO queue



Gaussian function is applied to every link

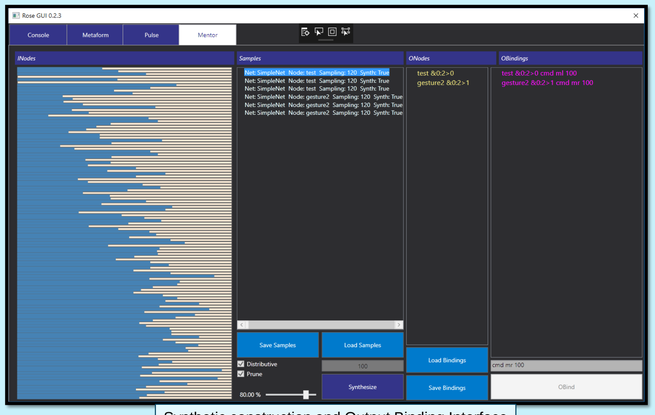
The sample cluster layer consists of nodes that make up one sample of a gesture.

The values of the samples are stored as link weights to form Gaussian centres of reference. The link weights deducted by the input values will yield deviation values.

The lower the deviation value the input has from the centre the more impact the input effectively have.

The application of the Gaussian function on individual input links eliminates major occlusions – a defect exhibited when using linear functions in radial applications that utilises orientations.

7 Middleware Engine



Synthetic construction and Output Binding Interface

Synthetic Construction:

The engine constructs the RBF network with for every sample marked for synthesis. A cluster node is one sample. Multiple samples with the same name have its cluster node directed to the same delegate node.

Output Bindings:

The Middleware fires at regular occurring intervals evaluating the output nodes value. Exceeding a given threshold the bounded triggers assigned to the nodes would be made effective.

Ideally future drones may be developed with new dedicated firmware for intelligent capabilities where in such circumstance communication protocols may change so the Middleware should not commit to a single protocol but to versatility.

Due to the diverse radio communication protocols the Middleware Engine will implement a broad range of standard communication protocols including COM ports, sockets and HTTP.

8 Evaluation

Though the system developed for this project did not implement any wireless communication the aim of this project is to develop a neural network capable of recognising patterns.

The wired implementation of the system proves the system developed to be reliable for the purpose and perhaps could be even more so with further improvements.

9 Potential Improvements

- Scaling the output nodes values appropriately to even out sensitivity relative to other gestures.
- Reforming network on the go to eliminate complete reconstruction of the network every time when using synthetic construction.
- Enforce synthetic construction to move Gaussian regression centres of existing sample for two samples that are in comparison considered negligibly similar rather than constructing new cluster nodes and making respective connections.
- Middleware Engine can be made to take samples of unfixed length and adapt by adding additional nodes as required.