Tracking System Using Artificial Neural Network for FPGA Cart Follower

M F Ahmad¹, S S N Alhady¹, C C Moi¹, A A A Wahab¹, W A F W Othman¹, A A M Zahir¹ and E A Bakar²

¹School of Electrical and Electronic Engineering, Engineering Campus, Universiti Sains Malaysia, 14300 Nibong Tebal, Penang, Malaysia.
²School of Aerospace Engineering, Engineering Campus, Universiti Sains Malaysia, 14300 Nibong Tebal, Penang, Malaysia.
sahul@usm.my

Abstract. Wheelchair users may have difficulty carrying their luggage while travelling. A proposed solution is introduced based on the problem stated. A tracking system for cart followers introduced to improve wheelchair mobility while transporting their luggage. The cart will track and follow the wheelchair at an appropriate distance. A Camera acts as input to enable the cart to track and follow the wheelchair. Pixy CMUCam5 is used to detect the predefined colour code (CC) in this project. The visual-based sensor gathered the information of the CC, which situated behind the wheelchair and translated the collected data into relative position information, such as distance and skew angle, which helps the cart in following the wheelchair. This translation is done in the Artificial Neural Network (ANN). The type of ANN used is a Multilayer perceptron (MLP) with the Levenberg-Marquardt training algorithm. Log sigmoid (Logsig) and Pure linear (Purelin) activation functions used in the hidden and output layer, respectively. The errors for distances and angles after computed with ANN are presented accordingly in Simulink. The MSE value obtained is 0.14007. The ANN implemented in the Field Programmable Gate Array (FPGA). The implementation of the ANN on the FPGA done through software and hardware configuration. The time taken for hardware implementation is faster than software due to parallel computations. The error value for output distance is less than 0.8000, and 0.3000 for the output angle during the simulation.

1. Introduction
Autonomous is one of the branches of the field of artificial intelligence. An autonomous robot is a robot which able to perform tasks in an environment without external control or guidance [1-2]. Therefore, autonomous robot vehicles can be understood as vehicles that able to move independently and intelligently. The autonomous vehicle is currently being a concerning topic. It is helpful for use as reconnaissance or aid vehicles in the field of planetary exploration, land and marine environments, remote repair and maintenance, and even intelligent wheelchairs for the disabled [3-4].

Visual tracking is recently studied as it is a crucial component in the computer vision area. It is a fundamental technology which supports in developing computer vision applications such as intelligent motion robot vehicle, object recognition, and tracking in an environment [5]. Visual tracking vehicle is one of the critical technologies in transportation. However, visual tracking for an outdoor application
is still a challenging issue in accurately tracking the target [6]. The visual tracking system can improve by gathering sensory information.

The sensory information may come from the vision or ultrasonic range sensor. A vision sensor may provide sufficient information than other sensors [7]. There are several projects based on colour tracking technique with Pixy CMUcam5 were proposed [8-11].

An artificial neural network (ANN) is a mimic of a human biological neural nerve system [12]. It lets the computer behave like a human in solving a problem and able to act some of its intelligence. It gathered all the data, detecting the pattern, and produced an output. ANN is related to the learning and decision making of a machine. The ANN production gathered various resources to build a correct sequence of the rule to produce a model that satisfies the reasonable condition.

Field Programmable Gate Array (FPGA) was firstly introduced by Xilinx in 1985 [13]. It contains the arrays of logic blocks that are programmable. FPGAs are pre-fabricated silicon devices programmed in the field of digital circuit or system [14]. FPGAs have greatly enhanced efficiency, performance, and power consumption. An FPGA made up of a matrix of configurable logic blocks (CLB), and it gives physical support for the downloaded program [15]. It contains look-up tables, multiplexers, and flip-flops. Hence it supports the performance of complex combinatorial and sequential functions [16]. FPGA also includes a hierarchy of reconfigurable interconnects that allow the blocks to be connected.

The parallel implementation of ANN in FPGA is applicable since it can maintain the parallel neuron layer architecture and has a flexible performance. It can support high gate density, which is the main requirement to perform the parallel computation of ANN [15]. Most of the ANN work with FPGA was implemented in software due to ease of work. An application of the ANN tracking system with FPGA was proposed in 2017 [17] and 2019 [18], respectively.

2. Methodology

2.1. Research framework
Vision sensor (Pixy CMUcam5) used to detect the predefined colour code (CC). The visual-based sensor gathered the information of the CC situated behind the wheelchair. The information such as centroid x, centroid y, weight, and height used to train the ANN. The number of hidden layers, neurons, the use of activation function, and selecting the suitable training algorithm was done. Then, it follows the process of training the network. After that, the performance of the system is tested and implemented in FPGA. Figure 1 showed the overall project flow. The blocks that circled in a red colour rectangle are the parts where the focusing part of this research.

![Figure 1. Block diagram of Overall Project Flow](image_url)
2.2. **Design of experiment**

The experiment is necessary to designed to obtain data illustrated in Figure 2. The CC is going to move along the track. CC and Pixy CMUcam5 assumed were at the same level, which is 30 cm from the ground. The experiment was conducted by moving the perpendicular distance between the cart (denoted by Pixy CMUcam5) and the wheelchair (designated by CC) from 20 cm to 69 cm. The sensitivity of the size is 1 cm. Another distinct variable is the angle in both the left and the right view. The angle sensitivity is 5°. The range of angles based on the distance shown in Table 1.

![Figure 2. Experimental setup](image)

**Table 1. Range of angle based on the distance**

| Distance (cm) | Angle (°)   |
|---------------|------------|
| 20-29         | -15 to 15  |
| 30-39         | -20 to 20  |
| 40-49         | -25 to 25  |
| 50-59         | -30 to 30  |
| 60-69         | -30 to 30  |

2.3. **ANN design and training process**

A primary neuron made up of two inputs, one hidden and single output layer. Each of the layers has its activation functions. The input multiplied by its respective weight, which later added with bias. Then it generates a desired output through the activation function. In this research, the hidden layer was set to 10 neurons, while the output layer formed to two neurons. Networks simulated are affected according to the number of neurons in their hidden layer. Figure 3 showed the ANN terminology.

The input data and target data are fed into the network involving the hidden layer number, the number of neurons in each layer, the use of activation function, and selecting the suitable training algorithm. Then, the training parameter and stopping criteria were set. Before the training starts, the initial weight was set. The Training network will generate weight and bias. The weights and biases will be updated to an optimum value. The training is considered done if the ANN reaches the specified
level of accuracy. During the training phase, the network will be tested. Table 2 showed the initial parameters of the proposed ANN.

Levenberg-Marquardt Algorithm (trainlm) was used to train the ANN. Trainlm is one of the fastest algorithms in the toolbox and highly recommended as a first-choice supervised algorithm. Trainlm able to get a lower mean square error.

The activation function is a function used to map the input data into an output signal in a neuron. Logsig activation function maps the output of the neurons to a value between 0 and 1. The values between are mapped non-linearly. Purelin activation functions direct maps the output of the neurons as the same input value. Figure 4 showed the activation function graphs for logsig and purelin.

Mean squared normalized error performance function, MSE is one of the most usable performance functions. It calculates network performance according to the mean of squared errors. It shows the distance between the model estimate of the test values and the actual test value. The value of MSE calculated as in Equation (1):

\[
MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - y_f)^2
\]  

(1)

Where \( n \) is data points, \( y_i \) is the observed value, and \( y_f \) is the predicted value. The simulation error is calculated based on the difference value between the actual and simulated value as in Equation (2):

\[
Error = |Actual \ value - Simulated \ value|
\]  

(2)

| Table 2. Initial parameters of proposed ANN |
|--------------------------------------------|
| Descriptions                             | Parameters                           |
| Type                                      | Multilayer perceptron (MLP)           |
| Training algorithm                       | Levenberg-Marquardt                  |
| No of layers                             | 3                                     |
| Input layer neurons                      | 4                                     |
| Hidden layer neurons                     | 10                                    |
| Output layer neurons                     | 2                                     |
| Activation function in the hidden layer  | Log sigmoid (Logsig)                  |
| Activation function in the output layer  | Pure linear (Purelin)                 |

Figure 3. ANN Terminology
2.4. ANN in FPGA

Altera DE0-Nano FPGA is the microcontroller used in the project. There are two ways to implement the network into the FPGA. One of the methods is through the hardware configuration with Verilog HDL. ANNs are multiplication rich. It involves the choice of number representation, selection of sufficient arithmetic precision, and nonlinear functions implementation. All these limitations have to be carefully considered to achieve the desired performance of the ANN when implemented on FPGA. Data representation is vital as it can have a significant impact on FPGA resource usage. A number of bits, FLP, and fixed-point (FXP) need to be considered for the input, weights, and activation functions. It comes up from 16-bit, 32-bit or 64-bit. FLP arithmetic has the advantage that it can represent a broader range of numbers than FXP, for the same number of bits used. In this case, double precision is used, which is a 64-bit FLP. Hence, Altera FLP core used for writing Verilog HDL. The double-precision value of inputs, weights, biases, and other constant values converted to 64-bit hexadecimal. Hexadecimal FLP constants make it easy for the compiler to reproduce the exact value. Another way is through software programming with C code. The ANN model with the C code supported in FPGA. The ANN model in software configuration based on the circuit is own configured by the software itself. The software is running based on the processor of the board—the ANN program C code compiled in the Quartus Prime SOC builder tool that used to handle FPGA.

3. Results and Discussions

3.1. ANN in Matlab training

Four data inputs, which are x, y, w, and h, will be mapped to ten hidden layer neurons. One input mapped to a neuron on the next layer. Each input will generate each weight, and a single bias value will be created for each hidden layer neuron. For each mapping, 40 weights generated from input to hidden neurons, and ten biases developed for each hidden neuron in the hidden layer of the network. Table 3 showed the weight and bias generated for ten hidden neurons. Figure 5 showed the converging MSE graph where the weights and biases converge and minimize the MSE value of the validation. This graph showed the converged weights and biases in the early stages of the epochs. The minimum validation of the MSE value updated to 0.14007 at epoch 507.
Table 3. Weight and bias generated for ten neurons

| Hidden layer neurons | Weight 1     | Weight 2     | Weight 3     | Weight 4     | Bias         |
|----------------------|--------------|--------------|--------------|--------------|--------------|
| 1                    | 1.1236669    | 1.1776073    | -1.4264712   | 6.0413344    | 5.3148182    |
| 2                    | -3.6099409   | -7.1933588   | -1.0874162   | -8.6502485   | 1.4938078    |
| 3                    | -0.2405375   | -0.2808738   | 0.0460280    | -1.1592502   | 0.3718246    |
| 4                    | -0.2672651   | -0.0871380   | 0.0357359    | -0.4126431   | 0.5920290    |
| 5                    | 23.106525    | 2.8867585    | -3.4786270   | -3.3120306   | -0.751990    |
| 6                    | 0.1157172    | -0.4019770   | -0.1826802   | -1.5601343   | -0.350527    |
| 7                    | 2.8119271    | -0.2557995   | -0.8502259   | -0.6961556   | 2.1056551    |
| 8                    | 1.2963302    | 1.0977423    | -1.6552084   | 5.3682223    | 4.8701245    |
| 9                    | -0.0912603   | -0.7048600   | 0.2552271    | -9.0108453   | -4.976495    |
| 10                   | 0.2817997    | -0.8924819   | 1.2520365    | -3.2181090   | -5.593778    |

3.2. ANN in Simulink test
The simulation of Simulink tested by applying the four constant values, which are centroid x, centroid y, weight, and height, to the input block parameter. The first four constant values, 116, 93, 193, and 111, are applied to the input block parameter, respectively. Then it will pass through the ANN block and then generate the outputs, which are distance and angle, as shown in Figure 6. Table 4 showed the tabulated result for various angles at distances of 30cm and 60cm meanwhile, Table 5 showed various distances at angles of -15°, 0° and 15°. For distance, the average error value is 0.16 for 30cm and 0.08 for 60cm. For angle, the average error value is 0.27, 0.15, and 0.35 for -15°, 0° and 15°, respectively.
Figure 6. First simulation error test

Table 4. Error values for various angle at two different distances

| Actual angle | 30cm | 60cm |
|--------------|------|------|
|              | Simulated | Error | Simulated | Error |
| -30          | -      | -    | -30.09    | 0.09  |
| -25          | -      | -    | -24.88    | 0.12  |
| -20          | -19.80 | 0.20 | -19.77    | 0.23  |
| -15          | -14.88 | 0.12 | -14.85    | 0.15  |
| -10          | -9.72  | 0.28 | -10.09    | 0.09  |
| -5           | -4.77  | 0.23 | -5.02     | 0.02  |
| 0            | 0      | 0.07 | 0.04      | 0.04  |
| 5            | 4.98   | 0.02 | 4.95      | 0.05  |
| 10           | 10.21  | 0.21 | 10.18     | 0.18  |
| 15           | 15.12  | 0.12 | 15.03     | 0.03  |
| 20           | 20.19  | 0.19 | 20.04     | 0.04  |
| 25           | -      | -    | 24.99     | 0.01  |
| 30           | -      | -    | 30.00     | 0.00  |
3.3. ANN in FPGA
The ANN model in Verilog HDL simulated by using ModelSim, as shown in Figure 7. The execution time of the ANN in hardware configuration is faster than software configuration due to parallel computation. Table 6 showed the execution time for hardware and software configuration. Table 7 and Table 8 showed the error values for hardware and software configuration during the simulation for 15° angle and 30 cm distance, respectively. The difference between the simulation result of the ANN in hardware and software configuration is slightly different only. The error values for output distance is less than 0.8000, and 0.3000 for the output angle during the simulation.

| Actual distance (cm) | -15° Simulated (cm) | Error | 0° Simulated (cm) | Error | 15° Simulated (cm) | Error |
|----------------------|---------------------|-------|------------------|-------|-------------------|-------|
| 20                   | 20.21               | 0.21  | 19.92            | 0.08  | 19.90             | 0.10  |
| 30                   | 29.77               | 0.23  | 29.61            | 0.39  | 29.57             | 0.43  |
| 40                   | 40.05               | 0.05  | 39.91            | 0.09  | 39.83             | 0.17  |
| 50                   | 49.54               | 0.46  | 49.96            | 0.04  | 49.74             | 0.26  |
| 60                   | 60.40               | 0.40  | 59.85            | 0.15  | 59.20             | 0.80  |

Table 6. Execution Time for Software and Hardware Configuration

| Type of implementation | Time taken (µs) |
|------------------------|-----------------|
| Software               | 0.990           |
| Hardware               | 0.555           |
Table 7. Error Value for Hardware and Software Configuration for 15° angle

| Actual distance | Simulated distance   | Error          |
|-----------------|----------------------|----------------|
|                 | Hardware  | Software  | Hardware | Software |
| 20              | 19.901394 | 19.901306 | 0.0986   | 0.0987   |
| 30              | 29.570096 | 29.570103 | 0.4299   | 0.4299   |
| 40              | 39.825553 | 39.825523 | 0.1744   | 0.1745   |
| 50              | 49.738572 | 49.738625 | 0.2614   | 0.2614   |
| 60              | 59.201791 | 59.201736 | 0.7982   | 0.7983   |

Table 8. Error Value for Hardware and Software Configuration for 30cm distance

| Actual angle | Simulated distance   | Error          |
|-------------|----------------------|----------------|
|             | Hardware  | Software  | Hardware | Software |
| -20         | -19.801033 | -19.801025 | 0.1990   | 0.1990   |
| -15         | -14.880223 | -14.880180 | 0.1198   | 0.1198   |
| -10         | -9.717897  | -9.717865  | 0.2821   | 0.2821   |
| -5          | -4.7709154 | -4.7708990 | 0.2291   | 0.2291   |
| 0           | -0.0707291 | -0.0706670 | 0.0707   | 0.0707   |
| 5           | 4.9819675  | 4.9820400  | 0.0180   | 0.0180   |
| 10          | 10.207223  | 10.207214  | 0.2072   | 0.2072   |
| 15          | 15.121025  | 15.121078  | 0.1210   | 0.1211   |
| 20          | 20.194267  | 20.194328  | 0.1943   | 0.1943   |

4. Conclusions
A tracking system developed using ANN. Pixy CMUcam5 tracking camera used to detect the predefined CC in this project. It gathered the information of the CC and translated it into relative position information, such as distance and skew angle, which helps the cart in following the wheelchair—the collected data used for training the ANN. MATLAB tool with a consequence of steps used to train the ANN training. The training algorithm for MLP used is Levenberg-Marquardt. The activation functions used in the hidden and output layer are Logsig and Purelin, respectively. Float numbers used for the weights and biases. The MSE value obtained is 0.14007. For distance, the average error value is 0.16 for 30cm and 0.08 for 60cm. For angle, the average error value is 0.27, 0.15, and 0.35 for -15°, 0°, and 15° respectively. The Verilog HDL used to write the ANN model for hardware configuration and C language for software configuration. The neurons computed using the Altera FLP core. The ModelSim test bench tested the ANN in hardware configuration. Simulation of both ANN in hardware and software configuration get almost the same results. The time taken for hardware implementation is faster than software due to parallel computations. The error value for output distance is less than 0.8000, and 0.3000 for the output angle.

Acknowledgement
The authors would like to thank Universiti Sains Malaysia (USM) for the support of the research under grant no 1001/PELECT/8014160.
References

[1] Beer, J.M., Fisk, A.D. and Rogers, W.A.: Toward a framework for levels of robot autonomy in human-robot interaction. Journal of human-robot interaction, 3(2), pp.74-99 (2014).

[2] Oyejide, O.J., Okwu, M.O., Tartibu, L.K. and Olayode, O.I.: Development of Sensor Controlled Convertible Cart-Trolley. Procedia CIRP, 91, pp. 71-79 (2020).

[3] Cox, I.J.: Blanche: Position estimation for an autonomous robot vehicle. In Autonomous robot vehicles (pp. 221-228), Springer, New York (1990).

[4] Zahir, A.A.M., Othman, W.A.F.W., Alhady, S.S.N., Wahab, A.A.A. and Ahmad, M.F.: Conceptual Design of Autonomous Cart Follower for Wheelchair User. Platform: A Journal of Engineering, 4(1), pp. 2-11 (2020).

[5] Ribaric, S., Adrinek, G. and Segvic, S.: Real-time active visual tracking system. In Proceedings of the 12th IEEE Mediterranean Electrotechnical Conference (IEEE Cat. No. 04CH37521), Vol. 1, pp. 231-234 (2004).

[6] Xin, J.N., Du, X. and Zhang, J.: Deep learning for robust outdoor vehicle visual tracking. In 2017 IEEE International Conference on Multimedia and Expo (ICME), pp. 613-618 (2017).

[7] Smith, C.E., Richards, C.A., Brandt, S.A. and Papanikolopoulos, N.P.: Visual tracking for intelligent vehicle-highway systems. IEEE Transactions on Vehicular Technology, 45(4), pp.744-759 (1996).

[8] Ahmad, M.F., Alhady, S.S.N., Kaharuddin, S. and Othman, W.A.F.W.: Visual based sensor cart follower for wheelchair by using microcontroller. In 5th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), pp. 123-128 (2015).

[9] Ahmad, M.F., Rong, H.J., Alhady, S.S.N., Rahman, W. and Othman, W.A.F.W.: Colour tracking technique by using Pixy CMUcam5 for wheelchair luggage follower. In 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), pp. 186-191 (2017).

[10] Ahmad, M.F., Alhady, S.S.N., Rahman, W., Othman, W.A.F.W. and Zahir, A.A.M.: Visual Based Distance Recognition Technique by Using Pixy CMUcam5. In Intelligent Manufacturing & Mechatronics (pp. 479-485), Springer, Singapore (2018).

[11] Ahmad, M.F., Alhady, S.S.N., Rahman, W., Othman, W.A.F.W. and Zahir, A.A.M.: RGB Classification Determination with Different Light Intensity Using Pixy CMUcam5. In Intelligent Manufacturing & Mechatronics (pp. 517-525), Springer, Singapore, (2018).

[12] Zurada, J.M.: Introduction to artificial neural systems (Vol. 8), St. Paul: West (1992).

[13] Trimberger, S.M.S.: Three Ages of FPGAs: A Retrospective on the First Thirty Years of FPGA Technology: This Paper Reflects on How Moore's Law Has Driven the Design of FPGAs Through Three Epochs: The Age of Invention, the Age of Expansion, and the Age of Accumulation. IEEE Solid-State Circuits Magazine, 10(2), pp.16-29 (2018).

[14] Farooq, U., Marrakchi, Z. and Mehrez, H.: FPGA architectures: An overview. In Tree-based heterogeneous FPGA architectures (pp. 7-48), Springer, New York (2012).

[15] Abdu-Aljabar, R.D.: Design and implementation of neural network in FPGA. Journal of Engineering and Sustainable Development, 16(3), pp.73-90 (2012).

[16] Škoda, P., Lipić, T., Srp, A., Rogina, B.M., Skala, K. and Vajda, F.: Implementation framework for artificial neural networks on fpga. In Proceedings of the 34th International Convention MIPRO, pp. 274-278 (2011).

[17] Tat, L.Y., Alhady, S.S.N., Othman, W.A.F.W. and Rahiman, W.: Investigation on MLP Artificial Neural Network Using FPGA for Autonomous Cart Follower System. In 9th International Conference on Robotic, Vision, Signal Processing and Power Applications (pp. 125-131), Springer, Singapore (2017).

[18] Ahmad, M.F., Alhady, S.S.N., Oon, O.Z., Othman, W.A.F.W., Wahab, A.A.A. and Zahir, A.A.M.: Embedded Artificial Neural Network FPGA Controlled Cart. In Advances in Science, Technology and Engineering Systems Journal, vol. 4, no. 4, pp. 509-516 (2019).