VISUAL INVERSE KINEMATICS FOR COTTON PICKING ROBOT

K. G. Fue
College of Engineering and Entomology, University of Georgia
Tifton, Georgia
W. M. Porter
Crop and Soil Sciences, University of Georgia
Tifton, Georgia
E. M. Barnes
Cotton Incorporated
Cary, North Carolina
G. C. Rains
Entomology, University of Georgia
Tifton, Georgia

Abstract

Fast cotton picking requires a fast-moving arm. The Cartesian arm remains the most simple and quick moving arm compared to other configurations. In this study, an investigation of the 2D Cartesian arm controlled with a stepper-drive is investigated. The arm is designed and mounted to a research rover. Two stereo cameras are installed and used to take the images of the cotton plants in two different angles. One camera is directly pointing downward while the other camera is pointing perpendicular to the row. This configuration allows the robot to view the cotton plants and bolls. The robot arm can move upward and downward or left and right. The rover uses two linear servos connected to a variable displacement pump swashplate for powering four hydraulic wheel motors and the engine accelerator linkage to move forward. The forward and backward movement of the rover makes the cotton-picking robot arm movement 3-dimensional. The downward camera gives feedback to the robotic system on the position of the arm. The rover moves forward along the row and stops whenever the cotton boll is perpendicular to the cartesian arm. The sideways camera gives an alternative view of the cotton boll that allows the robot servos to stop accurately. The arm uses vacuum suction to pick the cotton bolls. The vacuum suction end effector is mounted on the arm and pointing perpendicular to the row. In this paper, the kinematics and movement of the cotton arm and boll picking are demonstrated.

Introduction

The cotton production industry has been integrating new technologies since Eli Whitney introduced the cotton ginning machine in the late 18th century (Iles, 2012). One technology that improved dramatically from the 1940s was the introduction of mechanical cotton harvesting (Holley, 2000). However, cotton production economics has led to the development of larger and larger cotton pickers that are quite difficult to maintain and very expensive (Hayes (2017)). Most of these machines require cotton plants to be defoliated first and then harvested. Farmers who have small acreages cannot rent or own these large machines and earn a profit. Therefore, there is an opportunity to develop new and alternative harvesters that are small and cheap (Fue et al., 2018). The emergence of artificial intelligence and robotic systems provide an opportunity to develop such machines that would be different from the previous machines. These small machines would not require cotton defoliation, could be energy efficient and preserve the quality of the fiber. Robotic cotton-picking is generally slow compared to hand-harvesting due to the complexity of the robot arm. Deployment in a heterogeneous environment with so many uncertainties and variations are challenging. However, the complexity of the robot arm or degrees of freedom has been reported to increase the speed of controlling a lower number of degrees of freedom.

Hence, in this study, a 2DOF cartesian arm that moves vertically and horizontally is developed, and kinematics of the arm is analyzed. Kinematic control of the robot arm is developed by using an artificial neural network so that it becomes easier for farmers to automatically recalibrate the machine every time before using it.

Materials and Methods

The robot has three main parts; robotic arm, imaging system, and the red research rover.

Imaging System

An embedded kit (NVIDIA Jetson TX2 development kit, Nvidia Corp., Santa Clara, CA, USA) was installed on the research rover and, together with machine vision software, used to extract features of cotton boll images and determine the 3D position of the boll relative to camera and ground (Fue et al., 2018). NVIDIA Jetson TX2 (NVIDIA Pascal 256 CUDA cores, Quad ARM and HMP Dual Denver CPU, 8GB 128-bit LPDDR4 RAM, 32GB eMMC SATA drive) with ZED SDK installed was used to provide high graphics computing resources for fast image analysis. An RGB stereo camera (ZED camera, Stereo labs Inc, San Francisco, CA, USA) was installed and used to acquire images. ZED is 175 x 30 x 33 mm and weighs 159g. ZED has a 4M pixel sensor per lens with large 2-micron pixels. The left and right sensors are 120 cm apart. ZED was chosen due to the nature of the tasks, such as needing to work outdoor and provide depth data in real-time. ZED camera provides a 3D rendering of the scene using the ZED software development kit (SDK) which is compatible with other platforms like Robot operating system (ROS), OpenCV library, MATLAB, and Unity. ROS was chosen since it provides all the services required for robot development like device drivers, visualizers, message-passing, package design, and management and hardware abstraction (ROS, 2017 and Fue et al., 2018). ROS was initiated remotely by using a client machine and images were acquired using the ROS topics feature provided by the ZED wrapper. Images were parsed to the processing unit and analyzed using OpenCV (version 3.3.0) machine vision algorithms.

The ZED camera system was mounted on a research rover (Rains et al., 2015) at 90° below the horizontal (means it was directly pointing downward) and took images at the rate of 60 frames per second at WVGA quality while the rover was stationary. The research rover is a custom-built articulated vehicle (West Texas Lee Corp.,) with modifications to meet the field conditions, navigation, and obstacle avoidance requirements of an unstructured (such as open field, end of row) and structured row crop field (Rains et al., 2015). The distance of the camera to the ground was 220 cm.

Robotic Arm

In this study, a robotic arm was designed to work as 2D cartesian system (Figure 1) and developed using two 2-phase stepper motors (MS048HT2 and MS200HT2, isel Germany AG, Eichenzell,Germany). MS048HT2 model was installed to run the horizontal linear axis (60 cm long), and MS200HT2 was installed to run the vertical linear axis (190 cm long). The connecting plates and mounting brackets use a toothed belt that is driven by the stepper motor to move back and forth (Figure 2). Two stepper drives (Surestep STP-DRV-6575 micro-stepping drive, AutomationDirect, Cumming, Georgia) were installed so as to provide accurate position and speed control with a smooth motion. The DIP switch of the drive was set to 400 steps per revolution. The vacuum pumping machine was installed on the red rover. The machine connects to the vacuum picking end-effector. Cotton bolls are vacuumed into the end-effector which is placed close to the cotton bolls (Figure 3). Then, the cotton bolls move to the storage bag that is connected to the mechanical vacuum pump.

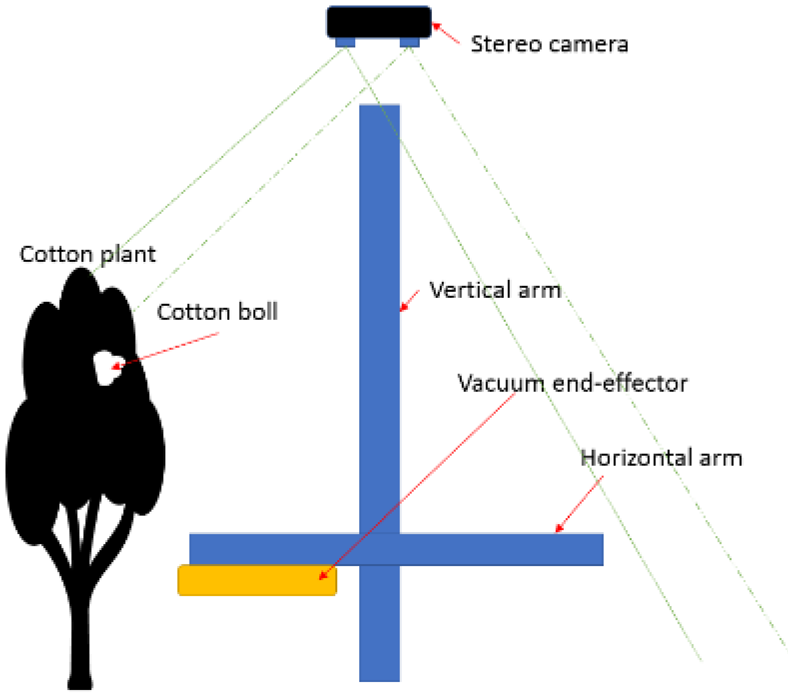


Figure 1. Robotic cartesian arm contextual diagram

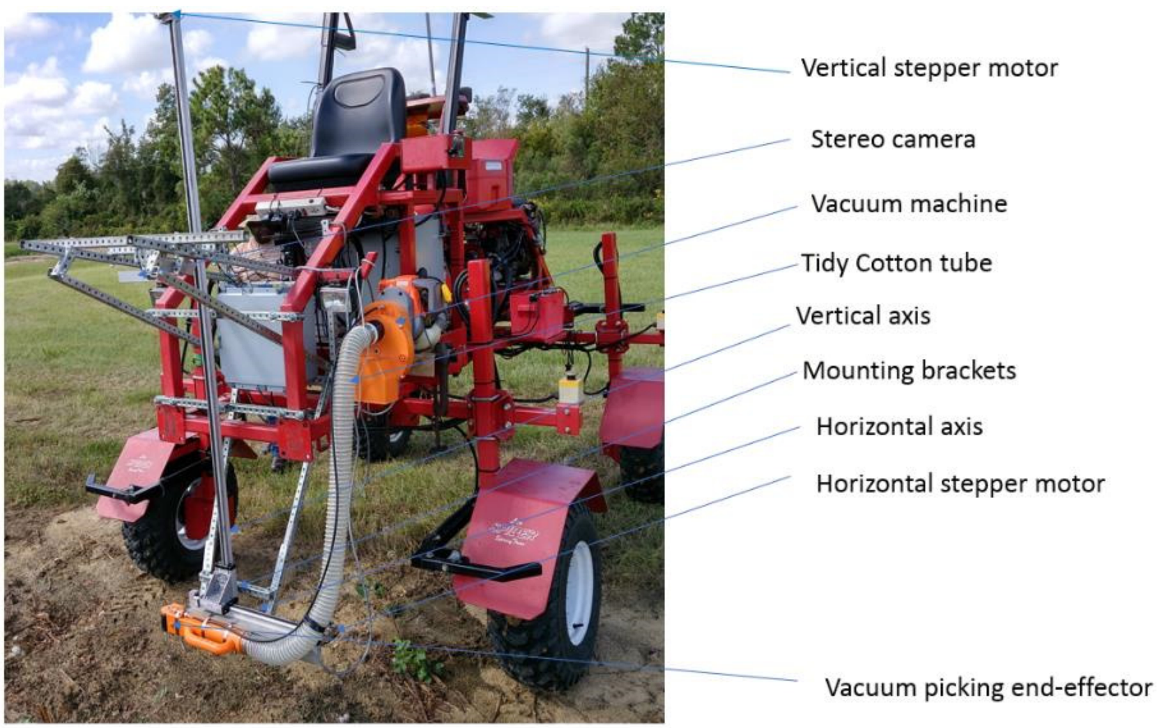


Figure 2. The robotic arm mounted on a red research rover

Figure 3. the end-effector comes close to the cotton bolls and sucks them using vacuum suction

Data acquisition, Boll Image, and end-effector features Extraction

Each image frame was acquired using ZED camera and analyzed by the algorithm developed using a 4-step machine vision algorithm (1. depth processing, 2. color segmentation, 3. feature extraction and 4. depth matching with the features). These steps are handled by the graphics optimized rugged development kit (NVIDIA Jetson TX2) to achieve improved performance as image calculations required massive graphics computing resources like NVIDIA CUDA cores. The ZED SDK acquired the images and processed them to get depth disparity and rectified images for both lenses. In this case, the ZED SDK was able to provide 60 fps for WVGA quality images.

The images acquired (Figure 4a) were first analyzed for arm movements. Since the arm is orange in color, the threshold color was determined to segment the image to obtain only the arm (Figure 4b). The cotton boll and end-effector segmentation task involved four steps (Gong and Sakauchi,1995):

1. Grab an image
2. Using RGB color threshold, separate each RGB component of the image. For cotton bolls, the white components of the image can be masked (all pixels with the value of Red, Blue, and Green greater than 220). And for the end-effector, the orange can be masked (Red from 200 to 255, Green from 0 to 255 and Blue from 0 to 50).
3. Subtract the image background from the original image.
4. Remove all the region where the contours are less than value M. Value M can be determined by estimating the number of pixels defining the smallest boll.

Then, feature extraction is done by finding contours of the continuous points which have the same intensity and are clustered. Color masking of the grey image was performed, then boundary curves were applied to detect and distinguish all white pixels of the image (Fue et al., 2018). The cotton boll then can be obtained after segmenting the contour of the arm (Figure 4c and 4d).

After, obtaining the contours for each of the objects (arm and bolls), the segmented image is matched with depth disparity. All the depths are calculated for bolls and arm.

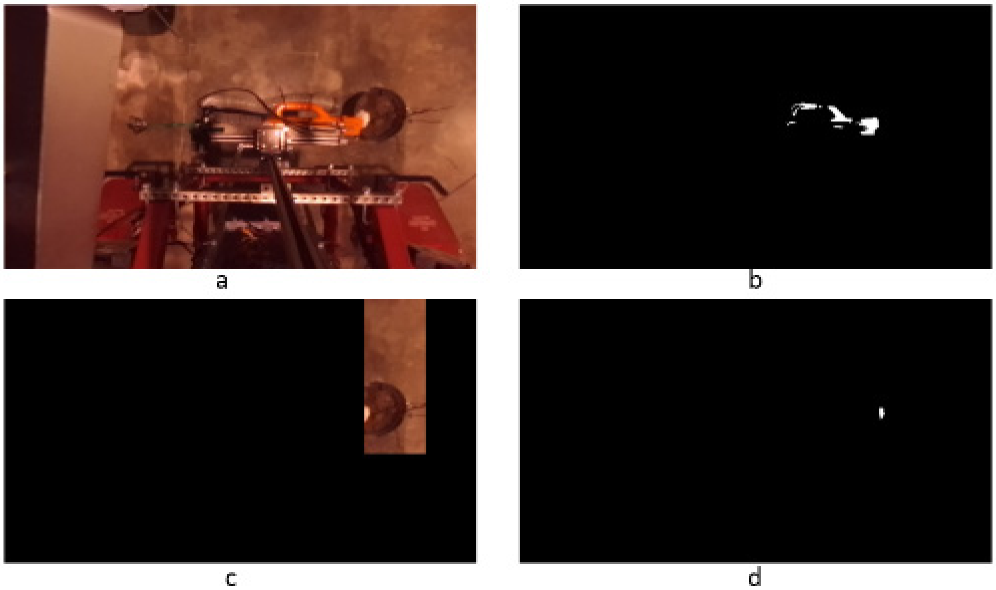


Figure 4. Color segmentation to get the arm and cotton boll

Depth and coordinates of cotton bolls and arm determination

After matching the depths and contours of the arm and bolls. Each reading of the position is logged. Then, by using the tip of the arm (Figure 4c), the system gets the image coordinates of the front part of the arm. Then, using centroids of each boll, the system takes a reading of the most variable depths for each contour which may represent separate bolls (Figure 4d). The system calculates the real world coordinates (W) from the image coordinates obtained (I) by using image geometry. Since the camera was calibrated using the ZED Calibration tool of the ZED SDK, the camera matrix (equation 1) values were obtained. The procedures to calibrate ZED camera can be obtained from their website. The camera matrix consists of **fx**, and **fy** (the focal length in pixels), C**x** and **Cy** (the optical center coordinates in pixels), and **k1** and **k2** (distortion parameters). This means the real world coordinates of a cotton boll, Wx and Wy (Equation 3 and 4) can be obtained if we know the value of Ix and Iy which is the coordinate of the centroid of the front part of the arm. Alternatively, by finding the inverse of the camera matrix and multiply with Vector Image (I), the world coordinates can be obtained. Cx, fx, Cy and fy are found by the calibrated camera matrix while Wz can be found from the depth disparity map provided by the ZED SDK.

$$\mathrm{C} = \begin{Bmatrix} f_x & 0 & C_x & 0 \\ 0 & f_y & C_y & 0 \\ 0 & 0 & 1 & 0 \end{Bmatrix} \qquad (1)$$

$$\begin{Bmatrix} I_x \\ I_y \\ 1 \end{Bmatrix} = \begin{Bmatrix} f_x & 0 & C_x & 0 \\ 0 & f_y & C_y & 0 \\ 0 & 0 & 1 & 0 \end{Bmatrix} * \begin{Bmatrix} W_x \\ W_y \\ W_z \\ 1 \end{Bmatrix} \qquad (2)$$

$$W_x = (I_x - Cx) * \left\{\frac{W_z}{f_x}\right\} \qquad (3)$$

$$W_y = (I_y - Cy) * \left\{\frac{W_z}{f_y}\right\} \qquad (4)$$

After obtaining such measurements, the system can execute other tasks like controlling the arm. For the machine to be able to execute each task separately but in connection to the other tasks, the finite state machine was developed.

Finite State Machine (FSM) using SMACH

FSM moves from one state to another after it is triggered by certain input. The states in the systems are detecting the boll, moving the vehicle forward or back after obtaining the position of the vehicle relative to the cotton boll, moving the arm up or down or back and forth after obtaining the boll position and arm position. The FSM guides the system from harvesting the first boll to the last one. The robot will start by detecting the boll and then decides to move the end-effector to the boll to harvest it. First, the arm will move up (move_up) or down (move_down) and match the vertical distance of the boll and the arm. If they are vertically matched, then, the arm will move horizontally to harvest (harvest_boll) the boll (Figure 5). The arm will always start with the boll which is at the highest distance from the ground and harvest other lower located bolls after picking the upper bolls.

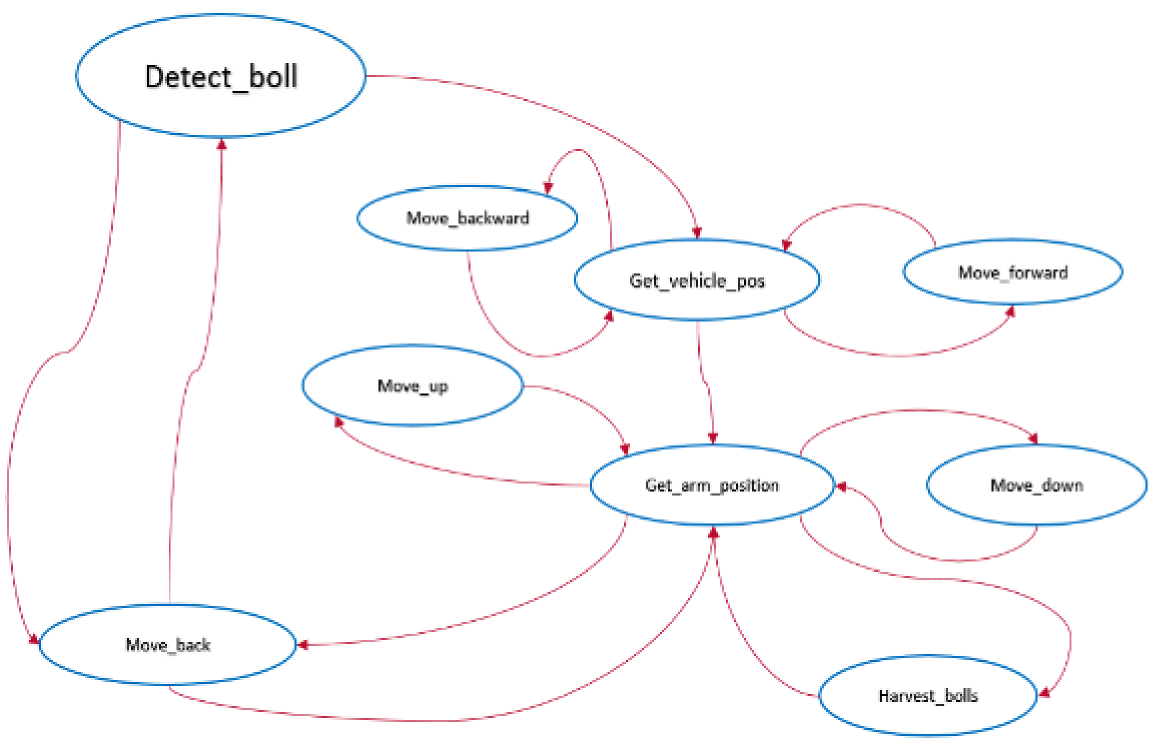


Figure 5. Finite State Machine of the system

In order to achieve a real state-of-the-art system, the robot operating system was used to deploy the FSM. SMACH which is a ROS task-level architecture was deployed. SMACH provides a very good ROS independent architecture that can be deployed with simple programming rules. Each of the states and transitions can be published by the SMACH. Each mode of the machine can modularly be developed and incorporated to achieve state-of-the-art and robust behavior of the robot. Each state was programmed using python and then deployed to Jetson TX 2. If the decision is made, the robot arm will move according to the signal sent from Jetson to the Arduino controller which sends signals to micro-stepping drive and instructs the motors to move to the target.

Robot controller

Robot arm controller (Arduino Mega 2560, Arduino LLC) receive a 4-bytes digital signal from the Jetson TX 2. The signal provides the number of steps and direction of the arm (Up, Down, back and forth). Then, the controller sends the signal to micro-stepping drive which in turn sends to the appropriate stepper for action. Arduino is connected to the Jetson using a USB 3.0 hub shared by the ZED camera. The micro-stepping drive that controls the motors were set to run a step pulse at 2MHz and 400 steps per revolution. This setting provides smooth motion for the arm.

Recalibration of the system using Artificial Neural Networks (ANN)

The ANN was developed so that the machine can automatically be recalibrated before use. ANN consisted of the input layer, 2 hidden layers, and the output layer. The input layer had 10 neurons, first hidden layer had 8 neurons, the second hidden layer had 5 neurons, and the output layer has only one (Figure 7). The ANN training involved two stages of training. When the results were estimated by equation (5) and when the results were projected using the ANN. The arm was moved randomly, and the new position was recorded together with the number of steps it has made. The data were shuffled to get testing and training data. The training data was separated by making 10% of data as testing data and 90% of data as the training data.

The system was calibrated on its' movements horizontally and vertically by doing the act repetitively for more than 1 hour. The linear inputs were the distance measured from the camera to the arm (Da), a distance of the camera to the target cotton boll (Db) and the differences in distance between the arm and target cotton boll (Dd). The output is a number of steps the machine should execute. The system steps were calculated as the modulus value of the equation of the steps against distance. The equation was obtained after taking a measurement of the distance of the arm from the camera when the arm is the furthest and change by each step until it is closest to the camera (Figure 6). The equation obtained by fitting the points in Figure 6.

$$\text{motor steps} = -410.7 * (\text{distance}) + 483.29 \qquad (5)$$

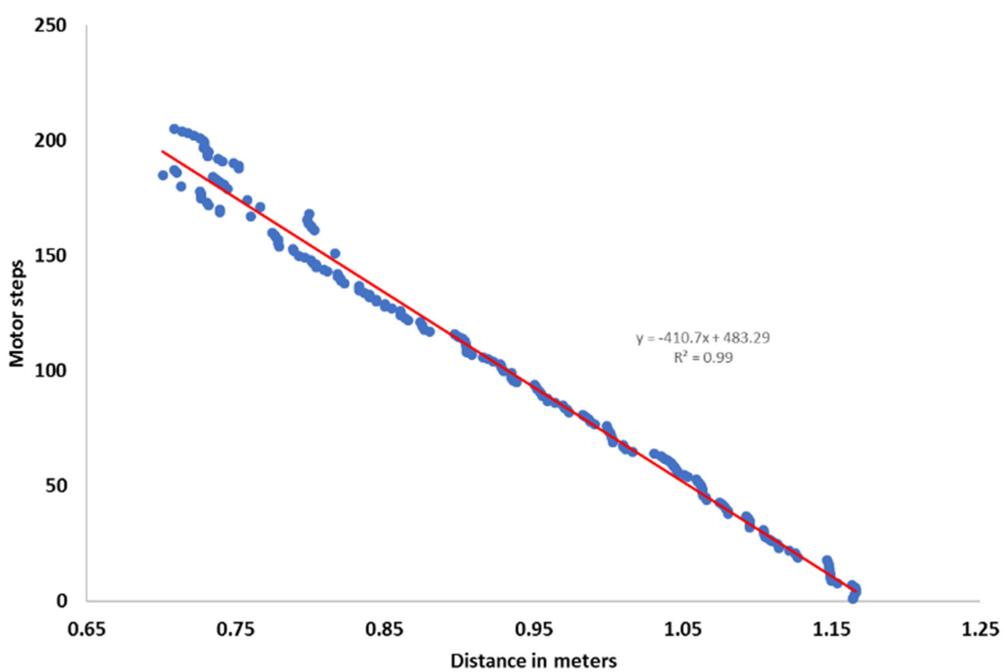
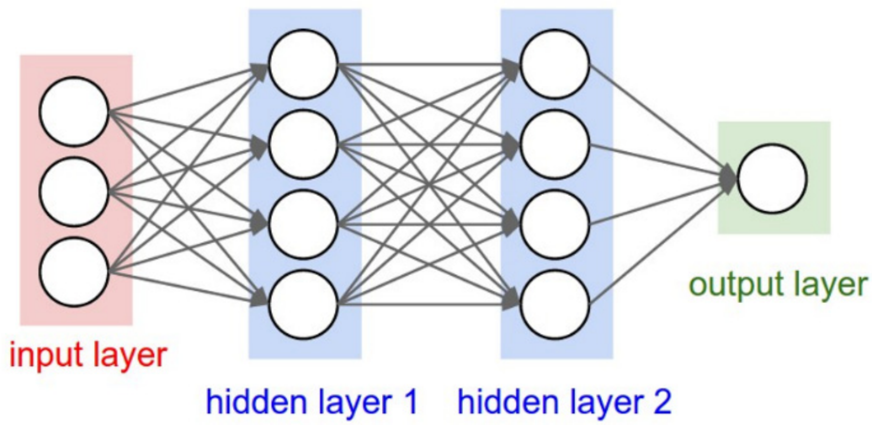


Figure 6. Calibration of the distance against steps of the stepper motor

Since the behavior of the motor changes, recalibrating the motor is necessary to achieve the best performance of the Cartesian arm. Hence, using these points, the ANN compared the true value obtained using equation 5 to start with and rerun the model so that it can adjust and predict arm position accurately.

The second stage, the values are adjusted accordingly after the true arm position (Db) is determined. Then, the system is retrained using the corrected Db for the second time with Dd being the difference of the corrected arm position and boll position. Boll positions are randomly assigned to give the system zigzag training data on all important movements of the arm. It should be noted that distance of the arm to move vertically was determined to be between 600mm to 1200mm from the camera while horizontally was determined to be from the center of the horizontal axis 125 mm to 570 mm. This is important so that the arm cannot attempt to move beyond the arms limits and destroy the toothed belt.



Neurons	10	8	5	1

Figure 7. Artificial Neural Network (ANN) for Arm Recalibration

The system first moved horizontally for 30 minutes and then vertically, 30 minutes. Then, it moved again but with corrected values. The results of the experiment are presented in the results and discussion section.

Inverse Kinematics of the Arm (Manipulator)

In order to simplify the system for the first test, vehicle movement states were removed and the system state transitions directly from boll detection to arm movement. Hence, the system used the model developed to estimate the travel distance of the arm to the boll.

Assume (α_x, α_y) is the number of steps predicted by the ANN to move the arm from (x_0, y_0) to (u, v). α_x is the number of steps horizontally while α_y is the number of steps vertically. Arm location (x_0, y_0) is any point of the arm that is closest to the boll while (u, v) is the closest point of the cotton boll which is the closest to the arm.

Assume D is the ANN model matching the values by taking 3 inputs (3 x 2 matrix) and output one vector which is the recommended number of steps the arm should move to the target boll.

$$\begin{pmatrix} x_0 & u & x-u \\ y_0 & v & y-u \end{pmatrix} * D = \begin{pmatrix} \alpha_x \\ \alpha_y \end{pmatrix}$$

$$\begin{pmatrix} \alpha_x \\ \alpha_y \end{pmatrix} * S = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$$

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} - \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \xrightarrow{yields} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

The system tries to minimize the value of $\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$

This means, when the front part of the arm is touching the boll, the difference in x and y will be 5 and 15 mm, respectively. The kinematics of the arm is explained in Table 1 using pseudocode.

Table 1. Pseudocode of the manipulator kinematics

	Pseudocode of the manipulator kinematics	
1	Vertical distance = V_d = v – y	
2	Horizontal distance = H_d = u – x	
3	If (Vd =<-15) move the arm up	Δy > 15
4	If (Vd >= 15) move the arm down	Δy > 15
5	If (-15 < Vd < 15) and (Hd > 5) move the arm to harvest	Δy < 15
6	If (-15 < Vd < 15) and (Hd < 5) move the arm back to position for next boll (1)	Δy < 15 and Δx < 5

Results and Discussions

The arm moved up and down for a few minutes, and the random movement was recorded (Figure 8). For each sample, the absolute error was recorded. The mean absolute error (MAE) for the samples was 10.9 steps, which indicates the system was making an error of 10.9 steps for each arm movement. After obtaining such results, the system used the steps predicted by ANN.

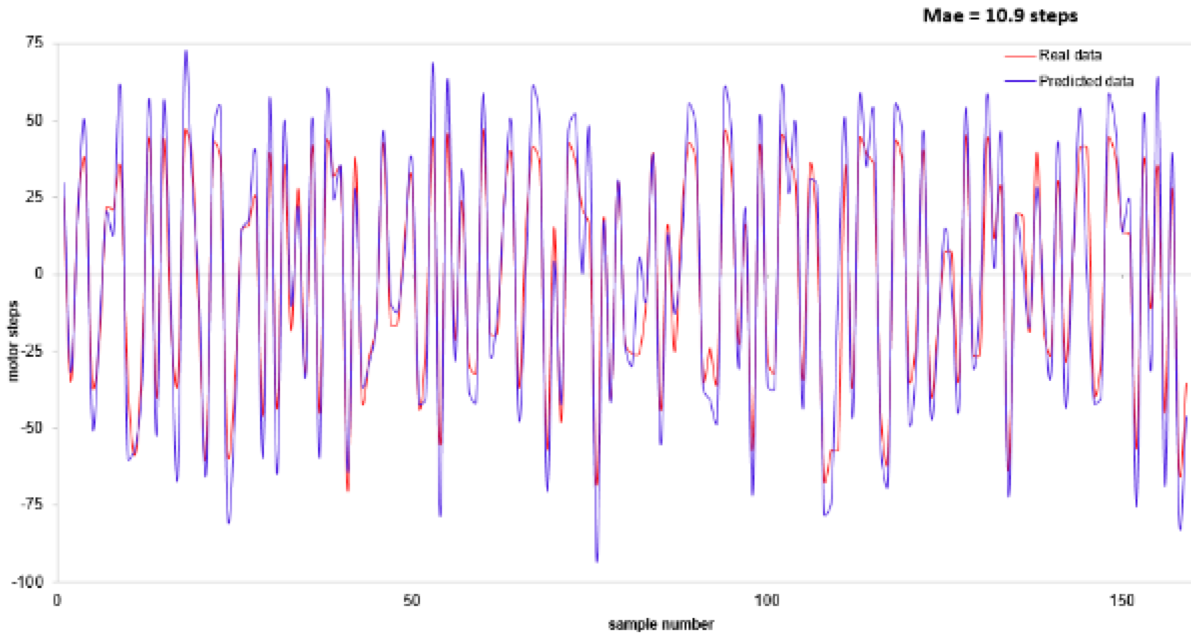


Figure 8. Movement of an arm using equation 5 got MAE = 10.9 steps

The system trained on the given data of equation 5 and true data obtained. The system used the data to predict the movement of the arm to reduce errors that are due to the camera and motor. The system used the prediction to learn the behavior of the true movement of the motor and recording camera. The data collected after training the ANN was tested and real movement compared to predicted movement (Figure 9). The absolute error was recorded for each sample. The mean absolute error (MAE) was 7.0 steps which are lower compared to 10.9 steps recorded in the training data obtained using equation 5. Looking at Figure 8 and 9, it can be noted that the real data is presented by red lines while predicted data is presented by blue lines. By comparing the red and blue lines in Figure 8, the blue lines peaks are either high or low all the times compared to red lines while in Figure 9, the red lines peaks are closely alternating with blues lines and the prediction becomes good. This shows the ANN learns the behavior of the system and predicts well by compensating for the linearity in the equation which only prediction or real value to be large or small all the time and instead it is easily equal, or any can be large or small. However, this is not the most important aspect of the use of ANN. The ANN achievement of lower MAE is considered more desirable for arm movements.

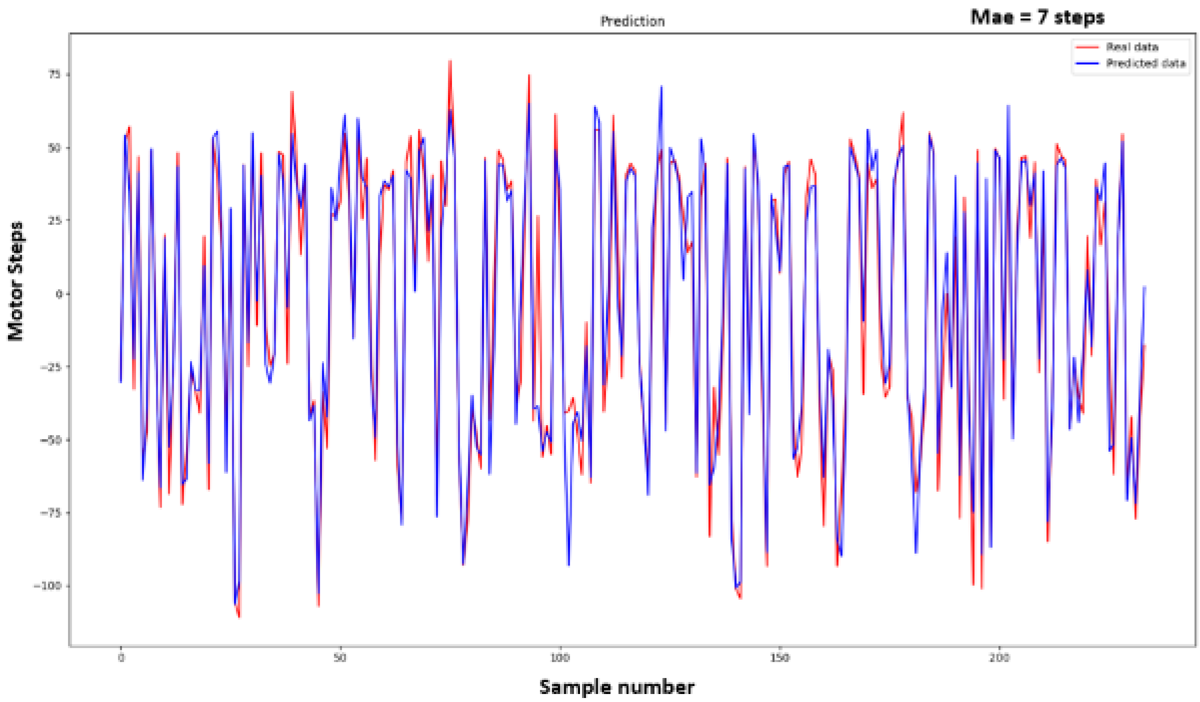


Figure 9. The real motor steps compared to predicted steps given by ANN. The MAE obtained is 7 steps

This is an important component of the robotic arm control to obtain higher precision (step pulse at 2MHz and 400 steps per revolution) as the smooth movement of the robot arm mainly will depend on the accuracy obtained. The motor steps are very smooth, and camera depth maps introduce the error. However, the ZED camera manufacturer claims that the accuracy of the camera is 1 mm. Nevertheless, this claim might be true for the movement of the arm horizontally. The arm movement horizontal using the ANN prediction was 0.37 steps (Figure 10). This might be due to short movements of the horizontal axis arm which is 60 cm long compared to vertical which is 190 cm long.

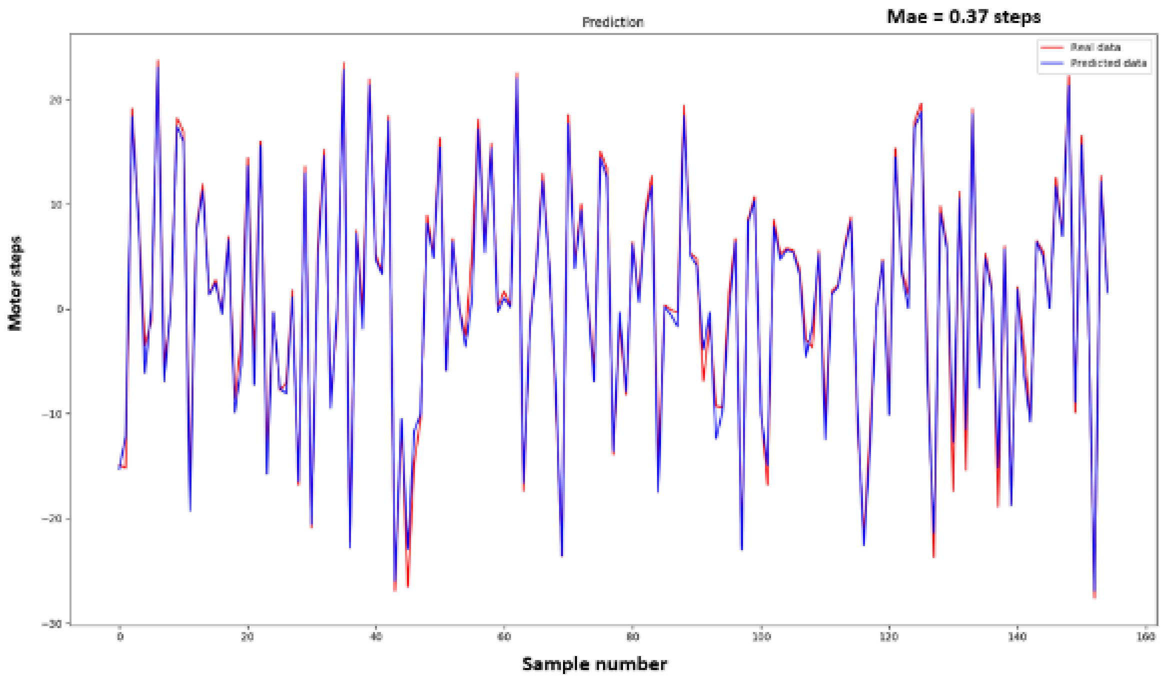


Figure 10. Horizontal axis arm steps movement as predicted by the ANN. The MAE obtained was 0.37 steps

Conclusions

The cartesian arm was developed and tested. The ANN algorithm to predict the arm movements were also developed. The ANN algorithm is intended to be used on the system for auto-calibration. Also, the ANN can be used to improve the inverse kinematics of the arm when camera settings or arm movement change. The equation (5) can be used well in inverse kinematics for short distances movement, but the equation usually changes if the camera is not fixed exactly as it is designed with the arm. This means it can be a challenge for normal uses of the system by the farmer if the system changes or camera position is slightly altered. Unfortunately, Changes in outdoor systems are inevitable as the systems are exposed to harsh conditions while working in the field. The accuracy changes of the system from 10.9 steps to 7.0 steps are very important in improving the system performance.

Acknowledgements

The authors are very thankful for project support from Cotton Incorporated, Agency Assignment #17-038. We also thank Mr. Ricky Fletcher and Mr. Gary Burnham for their helpfulness and technical support in building camera platform and collection of data.

References

Fue, K., Porter, W., and Rains, G. (2018). Real-Time 3D Measurement of Cotton Boll Positions Using Machine Vision Under Field Conditions. BWCC Paper No. 18385. Cordova, TN: NCC

Fue, K. G., Porter, W. M., & Rains, G. C. (2018). Deep Learning based Real-time GPU-accelerated Tracking and Counting of Cotton Bolls under Field Conditions using a Moving Camera. In *2018 ASABE Annual International Meeting*. ASABE Paper No 1800831. St. Joseph, MI: ASABE. doi:10.13031/aim.201800831

Gong, Y., and M. Sakauchi. 1995. Detection of regions matching specified chromatic features. Computer vision and image understanding, 61(2), pp. 263-269.

Hayes, L. (2017). Those Cotton Picking Robots. Available at http://georgia.growingamerica.com/features/2017/08/those-cotton-picking-robots/

Holley, D. (2000). The second great emancipation : the mechanical cotton picker, Black migration, and how they shaped the modern South. Fayetteville, Ark.: University of Arkansas. ISBN 155728606X.

Iles, G. (1912). Leading American Inventors. New York: Henry Holt and Company. pp. 75–103. Available at https://archive.org/details/leadingamericani00ilesrich/page/n113

Rains, G., B. Bazemore, K. Ahlin, A. Hu, N. Sadegh, & McMurray, G. (2015). Steps towards an Autonomous Field Scout and Sampling System. ASABE Paper No. 15219077. St. Joseph, MI: ASABE.