CS 684 Project Report

Group #5

Harvesting and sorting Bot

Himanshu Ojha (133050041)

Yogender Solanki (133050079)

Deepak Ranjan Swain (133050049)

Abhay Gajbhiye (133050083)

under the guidance of

Prof. Kavi Arya

Prof. Krithi Ramamritham

November 14, 2013

Table of Contents

List of Figures

1 Introduction

The propose of the project is, to design a bot which pluck the fruits and sort them according to their size. Harvesting and Sorting bot is designed in a way it will take trough number from the user and it will pluck all the fruits from that trough. Then it will go to that trough and continuously scan the plants of fruits from the mounted camera. When it will scan a ripen fruit then it will hold the fruit and cut it.. Then it will put them into the containers attached to it, according to the size .

1.1 1.1 Definitions, Acronyms and Abbreviations

Zigbee - A wireless communication protocol, which is operatre over 2.4GHz frequency.

ADC- Analog to Digital Converter, which will use to decode the data we get from various sensors. **FireBird** A robot indigenously designed at ERTS laboratory, IIT Bombay. **OpenCV** -Open Source Computer Vision Library use for image detection

2 Problem Statement

The main purpose of our project is to make the harvesting and sorting task automated for a smart greenhouse with low cost high performance robot. This product will ensure that all ripen fruits from specified trough will be pluck without damaging them.User will guide the bot by the interface. Then bot will go to the particular trough and pluck the fruits and sort them in the bot and deliver it to the user.

3 Requirements

3.1 Hardware Requirements

- FireBird V Robot
- IP Camera
- Zigbee module
- IR Sensor
- External Power circuit
- 2 Servo motors and 1 DC motor (Excluding 2 inbuilt DC motor)
- Robotic arm with Gripper-cum-Cutter
- Laptop with Windows OS

3.2 Functional Requirements

1. User interface to specify trough number and fruit type.
2. Bot will go to that particular trough by identifying checkpoints.
3. Scanning the fruit plants for ripen or non ripen fruits .
4. Detecting the fruit and align the gripper cum cutter to cut the fruit and hold it.
5. It will estimate the size of fruit and put them into the container according to the size.
6. After cutting one side of fruit it will start it for next side.

3.3 Non-Functional Requirements

1. User friendly GUI for an end-user.
2. Cut and collect the fruits without damaging it.
3. Open CV software library.

3.4 Design Constraints

1. Height and depth of the plant from the white line should be limited because the robotic arm and slider size is limited(In specific fruit should be at 7-11 inch height and in 8-14 inch depth).
2. The vision system will not able to detect fruit behind the leaves and same as leaves due to similarity of color with leaves.

On-line User Documentation and Help System Requirements There is need of the server where all documentation and codes can store. Interfaces User Interfaces The User interface will contains the communication controls and the guidance control to the robot for the user. Hardware Interfaces Zigbee Module that provide serial communication with peer to peer connections. Software Interfaces The system will be powered by the .NET framework and the OpenCV image Processing Library. Communications Interfaces We will use Universal serial Bus and Serial Ports to communicate with camera and robot.

4 Implementation

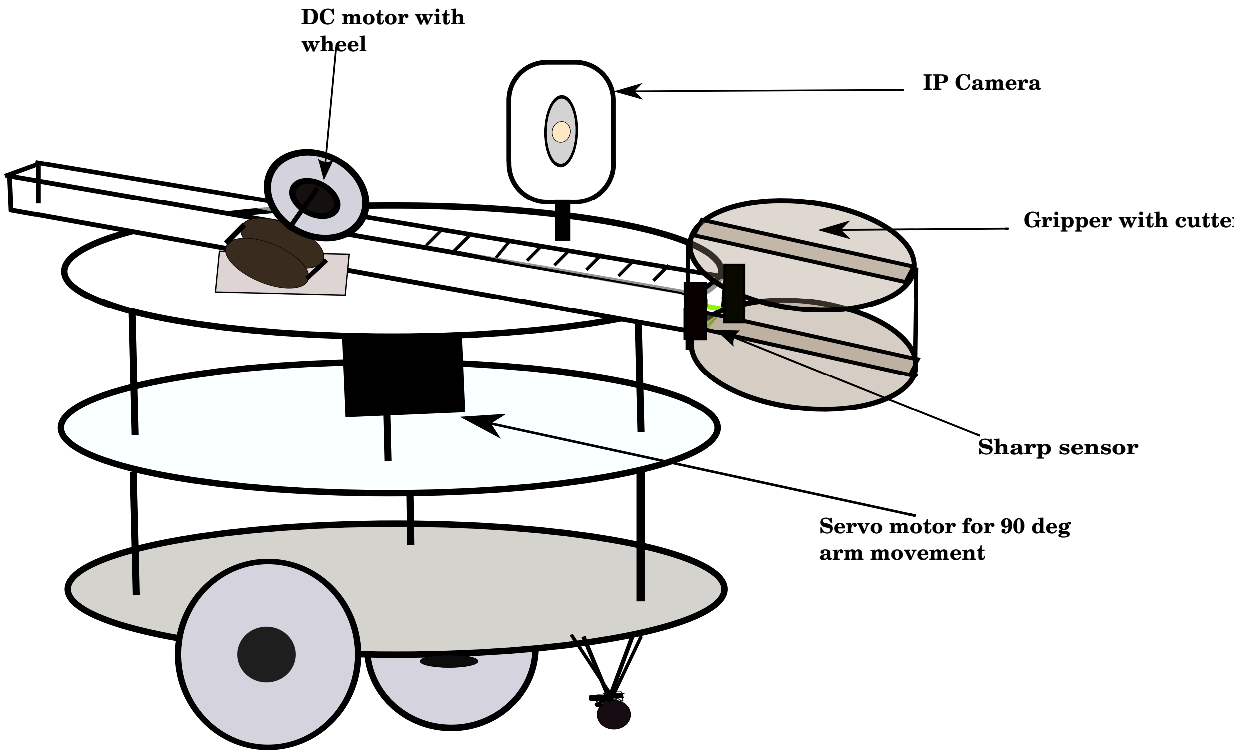

Figure 1: Harvesting and Sorting Bot: 3D view

4.1 FireBird Bot System Architecture

FireBird V has different layers of modules. Role of each module is summarized below:

- Firebird: This refers to hardware layer of FireBird Bot. Hardware is controlled by changing values in command and status registers. Refer to hardware [2] and software [3] manuals of FireBird V for further details.
- HAL Layer: Hardware abstraction layer (HAL) module hides hardware registers. It offers typical driver like primitives. init¡Device¿() API initializes the hardware of that device. Table 1 lists all HAL primitives and their use.
- Assertions: Assertion module mimics C assertions. It supports macro ASSERT(condition). If assertion fails (i.e. condition is false), Bot halts, shows line number, source file name and failed statement on LCD screen

and starts beeping until reset. Assertion is quite useful for debugging. Define macro NDEBUG at the start of the file to turn assertions off. This replaces ASSERT() statements by empty statements.

- ROM Filesystem: This module mimics ROM based file system in limited manner.It redirects standard C file streams stdin and stdout to ZigBee. Standard I/O functions like printf(), putchar(), etc. send data over ZigBee to remote console.

4.2 Bot Guidance System

It is the basic need of the project. For this purpose we use whiteline follower. There will white line for moving forward and backward. For checkpoint purpose we use broad white mark because there are three whiteline sensor in the bot when all detect white then it will read as checkpoint. User give a particular checkpoint to bot and will go there by counting them. This count will also help it to backtrack at the starting point.

Primitives	Use
print_sensor()	print the ADC channel value on the desired location on the LCD
update_sensor_value()	Update the values of the while line sensors global variable
toint()	convert the byte value to integer
servo1_rotate()	Directly rotate the servo to given angle
MVservo_1()	rotate the servo1 (5 dergree in each step)
sort()	After harvesting put the object into specific bin
lcd_print(,,,)	print the value on the LCD
distance_cm_4to30sensor()	get the value from the sharp sensor attached at the specific channel
motor_c2_forward()	slide the arm forward
motor_c2_backward()	slide the arm backward
servo_2()	rotate servo 2 to the angle
line_follower()	Will follow all the line and detect the check point
init_devices()	will initialize all the devices

4.3 Object Detection Using IP camera

Our primary interest lies in detecting multiple ripened fruits.We have to calculate the size of fruit at the time of detection. At the very begining user can specify what should be the range for different size. User have to specify what is color range for ripe or unripe fruit. Such requirement are necessary because it will generalize bot for all different type of fruits and user have freedom to specify particular size.

4.4 Fruit Cutting

Gripper cum Cutter fitted on bot can move along horizontal planes to cover certain depth. There is slider mechanism whic manage its movement. After detection of a fruit slider will move forward for the fruit. There are two sharp sensor to limit the slider motion. When slider stop moving the gripper will close and at the top of it there are two blades which will cut the fruit. And gripper will prevent it from dropping.

4.5 Fruit Sorting

During detection of the fruit by IP camera, the size is calculated at the time of detection. Accordingly signal(t-for small fruit AND T- for large fruit) is sent to the bot through wireless XBee communication. For sorting purpose we are using a servo motor which rotates the upper part of the bot containng gripper-cum-holder and one cart with two bins(One for Small and another for Large) attached behind the bot.After cuttinng the fruit,the servo motor retates 90 degree for small one and 140 degree for large one and then drops in the right bin.

5 Testing Strategy and Data

5.1 Bot Guidance System

For testing Bot guidance system, we used a reasonable sized white line flex and a GUI through which user will give the trough number. For checking each check point the bot covers we are using **checkpoint count** number of beeps.After reaching the final check point it will beep **once** and stop moving forward and rotate by 90 degree to go in the trough. Accordingly if there is any distrubance in number of beeps, we will rectify the code.

5.2 Testing Fruit Detection

For this purpose we use red color or green color balls of different sizes. We once scan them and specify ranges of acceptance color and size description. Then we hang all the balls of green and red color and different sizes. We check the image detectected by camera and after processing what it will generate for output.

5.3 Testing Fruit Cutting

Fruit size is categorized into two calsses: Small and Large. We used one tennis ball for small fruit and one regular ball for large fruit. There was a designed plant in the trough with one small and one large ripen fruit and one raw fruit. Then we checked our bot was able to cut the ripen small and large fruit and ignored the raw one.

5.4 Testing Fruit Sorting

We used one cart as fruit collector behind the bot. In the designed cart there are two bins one for small fruit and another one for large fruit. After cutting ripen fruits bot will sort the fruits and droped in the appropriate bin. If any error appeared then we looked towards the servo motor codingand differnt possible angles and its range because servo motor is not always moving equal degrees .

5.5 Energy Consumption

Due to large number of motor and IP camera we dont rely on the battery itself. For camera we design an extra power circuit. But camera consme much power than any other. At last we test it with external power supply.

6 Design Challenges and Open Issues

To cover a major area related to our project we first design quite heavy system. But our first design fails due to various issues. Yhen we come up with a reliable design with some constraints. There are some important issues that we experienced during our course project :

6.1 Flow and Error Control for ZigBee

ZigBee, being wireless communication medium, is lossy. We designed mechanism to detect errors and recover from them. We need flow control mechanism so that Bot receives commands at the rate which it can process. If there are many Zigbee sender and reciever are there then it recieves extra data. So we have to uniquely recognize a Zigbee by some address. We cannot send raw binary data over ZigBee. You must first encode it into printable ASCII characters. Non-printable ASCII characters are reserved for serial link control commands. Although this approach serves our purpose, it has lower efficiency. Future projects should consider porting TCP/IP protocol stack on FireBird platform.

6.2 Error in servo motor angle

The servo motor that we use are rotate only 0 to 180 degree. But this angle is not fix. If you set the angle for a particular task then it is the possibility that it will not go in the same position. There is error range of 10 - 5 degree. So that's why we use angle ranges for our task. There is also ambiguous behaviour occur when we reach the limit of 0 or 180 degree. So to avoids above issue we move servo in small angles with small delays.

6.3 Bot guidence System

We use white line following code for this purpose. But bots movements are not so perfect. we caliberate our white line sensors to get good values and use an cover to avoid extra light that affect sensor reading. Bot wheels are slipping on clear surfeces that affect the movement, if it is limited.

6.4 Weight Carrying Capacity

FireBird is able to handle much weight on it. Its fuctionality is fine untill carrying weight is less than 500-600 gms. But when weight will increase beyond this the motors for bot movement not work well and stuck even by any small object. We first design moch heavy system and the whole system rotated by the servo motor. At centre servo motor torque is high but when distance increase it is stuck.

6.5 Power Consumption

In firebird there is quite power issue. When we use 3-4 motors it will not able to provide them proper power supply. If any servo motor stuck the it will start draining power and due to this whole microcontoller will reset and running code is crashed. Servo motor is designed for 1amp current but in the above specified scenario it consume till 1.7amp current. So we use a technique to free servo motors after completition of its task, but it affected our design.

6.6 IP camera delay

We are using continuous image processing in our project. Due to this need we get a large ammount of image data. But IP camera when interface through wifi causes delay in sending images. The delay is upto 2 seconds. So when obeject detected camera delay in sending image , by this time bot moves forward and when server got late images it produces late reponse and bot is in the inconsistent sate of cutting fruit from where there is no fruit. So we use ethernet cable to interface with camera to avoid delay.

7 Future Work

To make this bot more efficient following work Needed : Future works:

7.1 Vertical arm Movement

In our design the main issue arises is overweight of our arm structure. So we eliminate one part of it which is vertical movement of arm. So our design is not working for fruit at variable height. So this enhancement will make bot quite efficient.

7.2 Image processing algorithm

Our image processing algorithm only work for color. So thats why it is unable to differntiating between fruit that are same colors like laeves. So better image processing algorithm with shape detection also works well.

7.3 Better Cutter Mechanism

The cutting mechanism we use in our bot is not much reliable because there is jerk in the servo motor motion the bades we used are not align properly. So better cutting idea cut the fruit without damaging.

7.4 Gripper Mechanism

We can use pressure sensors to avoid extra force on gripper. this will prevent fruit from damaging and better handling is possible with delicate fruits like strawberry.

7.5 power Management

This is the are where lots of work needed. Due to quite number of motors we are not handle power consumption in the bot.

8 Conclusion

Our system that is developed for fully automated harvesting and soting purpose except for the part where the user gives the command to start the task.Work efficiency of this system is high as it performs two tasks: Harvesting and sorting.Thus this project helps in reducing the manpower as well as ensuring minimal energy consumption.

References

[1] E-yantra website. http://www.e-yantra.org.

[2] Firebird v atmega2560 robotic research platform hardware manual. IIT Bombay and NEX Robotics Pvt. Ltd.

[3] Firebird v atmega2560 robotic research platform software manual. IIT Bombay and NEX Robotics Pvt. Ltd.

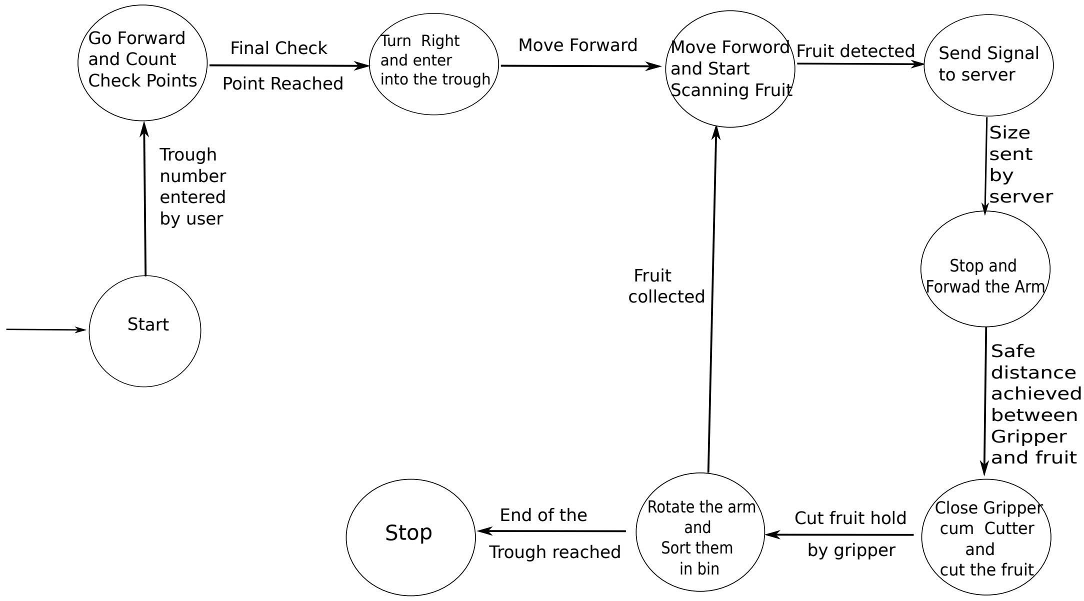

Figure 2: Finite State Machine

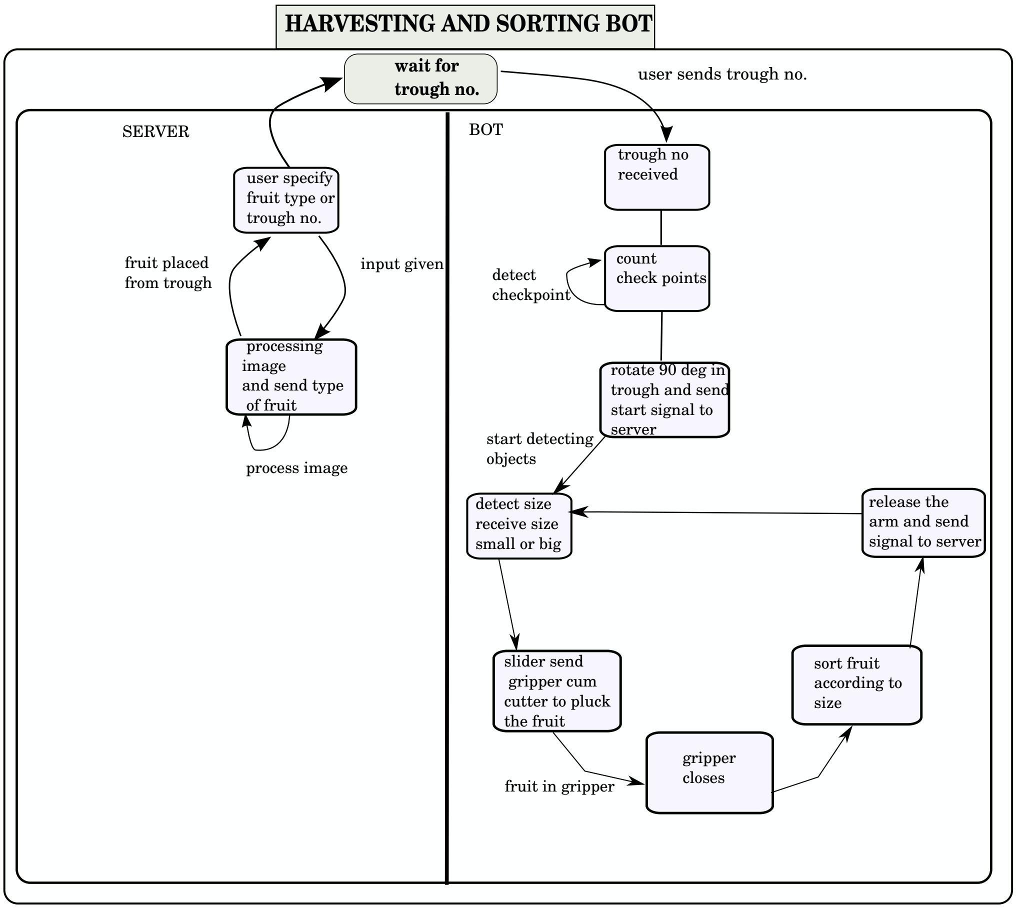

Figure 3: State Chart