CS684 Project
Group No: 8
Automated Pesticide Dispenser

Akanksha Patidar 123050094
Dixita Bothra 123059005
Swadesh Jain 123050085

18 Nov, 2012

Contents

1 Introduction

The project is about automate the pesticide dispensing in greenhouse using robot which is sent into the greenhouse arena. The robot make motion over black lines and localize itself using checkpoints. The robot take the trough number as input using zig-bee protocol and go to that trough and detect the empty spaces between plant rows in trough and dispense the pesticide in it.

1.1 Definitions

Firebird V: A robot indigenously designed at ERTS laboratory, IIT Bombay

ATMEGA2560: The microcontroller used by Firebird V bot

AVR GCC: The platform specific compiler which compiles C code to run various AVR microcontrollers developed by Atmel

WinAVR: Open-source software which uses AVR GCC Complier

ZigBee: is a specification for a suite of high level communication protocols using small, low- power digital radios based on an IEEE 802 standard for personal area networks.

Green House: is a building in which plants are grown. These structures range in size from small sheds to very large buildings. In general, big green house has many troughs containing multiple rows of plants and some space between these plant rows is left free for dispense pesticides.

Pesticides: are substances or mixture of substances intended for preventing, destroying, repelling or mitigating any pest.

2 Problem Statement

This project aims to use the Firebird V as an automated Pesticide dispenser for a green house. With given a greenhouse map and trough number, it will find empty spaces in specified trough of a Green House and dispense pesticide in them.

3 Requirements

3.1 Functional Requirements

- Takes input from user using zig-bee protocol.
- Moving the bot to the particular trough.
- Moving the bot parallel to the trough.
- Detect empty spaces in between plant rows and align itself.
- Dispense pesticide in the empty space.
- Return to its initial position, when task completed.

3.2 Non-Functional Requirements

- The robot must move such that all the empty spaces are detected.
- The robot must not go inside the trough
- The bot must not harm the plants
- The power consumption should be minimum
- **Performance** Pesticide dispensing should be uniform over the partitions.
- **Accuracy** Dispensing should be according to user instructions.
- **Safety** Pesticides should not dispense outside the trough.

3.3 Hardware Requirements

- Sharp IR Sensors

- Servo Motors

- Fire-bird

- Zig-bee Module

- Auxiliary Power Supply
- Sifter and Container with knob open/close mechanism using servo motor

- Lifter arm servo motor attachment

4 Assumptions

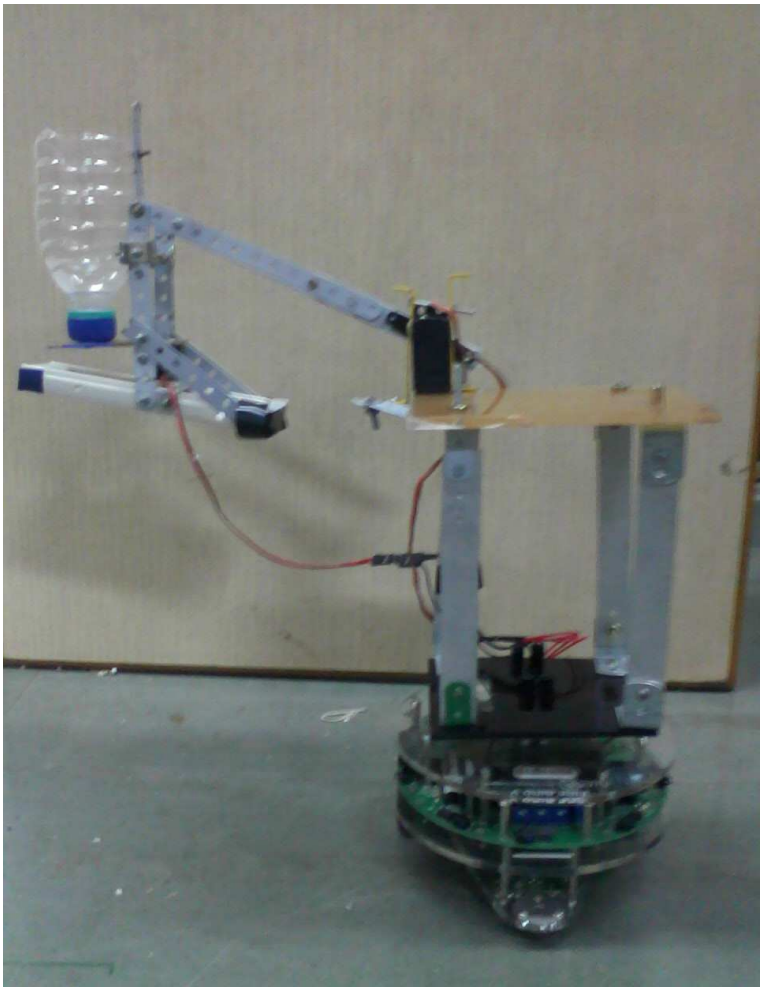
- Troughs containing multiple rows of plants and some space between these plant rows is left free for dispense pesticides and nutrients.
- Plants has height between 10 to 25 cm.

- Space between plant rows are equal to the width of sifter.
- The aisles in the greenhouse have black lines to help the bot navigate through it.
- Checkpoints are provided between the trough in the form of an black horizontal line.

5 Overall Description

5.1 Hardware Features

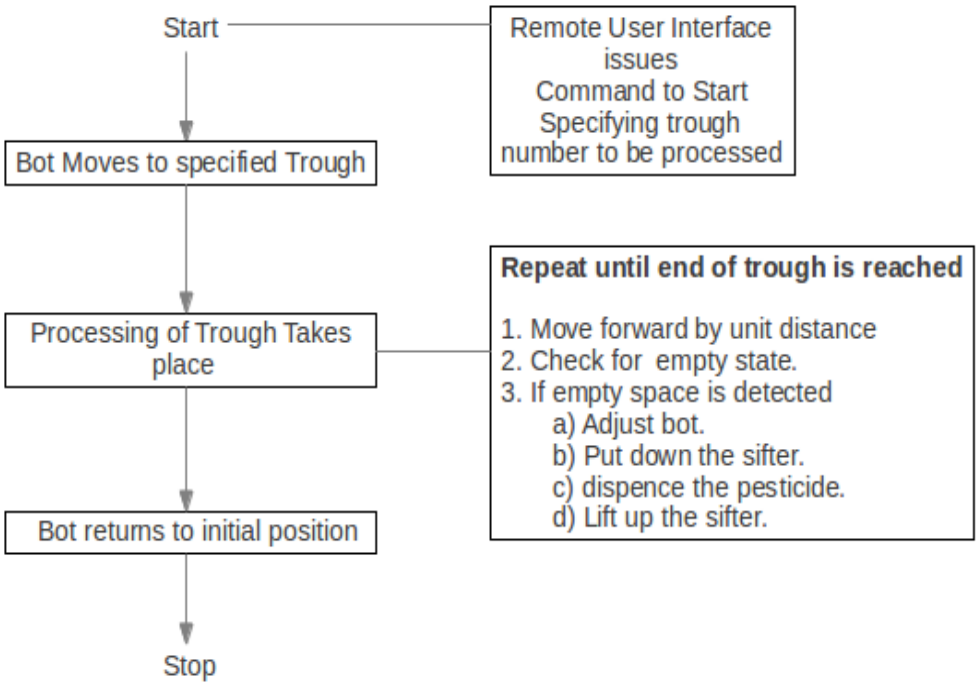
Our Pesticide dispenser bot was built with the following features:



- A sifter for provide uniform dispensing over trough.
- An lifter arm mechanism to hold the sifter.

  - It grips the sifter from the middle of it, so that it's surface always parallel to the land.
  - It also holds a container for pesticide.
  - It has 1 servo motor to provide up and down motion of the sifter so that it avoids any contact of sifter with plants.
  - It also provides vibration to the sifter using same servo motor.
  - It is placed at the back right side of the bot.
- The bot has a knob opening/closing mechanism for container, controlled a servo motor.
- To identify the empty spaces in the trough, it has 1 sharp IR sensors in the side of the bot (at height 10 cm).
- The bot makes motion over black line using 3 white line sensors and localize itself using checkpoints. Checkpoint is detected when all 3 white line sensors come over black.
- It takes input from user using zigbee module.

5.2 State Chart



- When bot gets a request from user using zig-bee protocol then it go to that trough and start searching for empty space.
- The bot moves parallel to the trough sensing for the empty space. It traverse 10 mm each time and check for the empty space.
- As soon as it identifies the empty space between plant rows, it moves 100mm (distance between IR sensor and the sifter) forward so that the sifter come over that space and then put down the sifter in that empty space and dispense pesticide in it for 5 sec.(time for controlling quantity).
- For dispensing pesticide, container's knob is opened by servo motor and vibrations provided by lifter arm using up/down motion by 10 degree. When dispensing is done then knob would closed and lifter arm lift up the sifter.
- It searching for empty space until it comes to the end of the trough.
- When it arrived at the end of trough, it returns to the initial position.

6 Implementation

6.1 Hardware Implementation

To start building required hardware, we must have its proper design. Otherwise, it will be much difficult for us and will also take much time to build it. So, the first requirement is to think of functionalities that the hardware must have. Now, we should think of the design of hardware. But, its not easy to think of an efficient working design of hardware. Lots of expected or unexpected problems arise when we deal with hardware. So, we should identify risks (as much as possible) associated with our designs in advance.

According to our project needs, we have find out following requirements:

- Sensors for detecting empty space and positioning of these sensors.
- A lifter arm to hold sifter.
- A mechanism which will move the lifter arm up and down.
- Servo motors for knob open/close and lifter arm mechanism.
- Some structure on which arm mechanism will rest.

To meet all these requirements, we started working on the design of our bot. We have seen videos of several projects on

`http://www.e-yantra.org/home/intership/student-projects`

. We got many ideas after looking those videos.
As arm mechanism of previous projects was not available, we decided to build it. So, our first goal was to build lifter arm mechanism with sifter. And to build required arm mechanism, we first build the sifter and container holding attachment.
The sifter and container holding attachment was made using :

- Sifter attached with container using strips
- Servo motor reside on sifter for knob open/close mechanism

After that we designed an arm which can hold the sifter and container holding attachment. We can think of many mechanism as we just need an up/down movement of attachment. We tried many mechanisms and face many problems also. Tried mechanisms are as follows.

- Slider mechanism for up/down movement, but it failed because of container's weight.
- Slider mechanism for forward/back movement, but it failed because of sifter bending.

At last we come up with an idea in which this attachment loosely hanged on one side of a ladder type arm and this arm moves up and down using servo motor attached on the other end. But, there was one problem. The attachment that we have built was heavy and the weight is on one side of the bot, that is risky. So we put some weight on the other side on the bot for balance.

Now, our basic structure was ready. Next we built the platform for placing lifter arm assembly. After building all necessary hardware components, we integrated all these components and other components like sensors.

During hardware design, we faced many problems. Several times we modified our hardware structure because of some problems. All these problems, their details and their solutions are mentioned in later sections.

6.2 Software Implementation(To Run Hardware)

Initially the bot is at the starting point. To run the bot, we need to connect the bot to the common interface through zigbee. Then trough number and start command is given. The bot goes to that particular trough and executes the task of dispensing for that particular trough and comes back to the starting point.

7 Testing Strategy and Data

We tested the bot for the following test cases

- **Empty spaces are at same distance and of sifter width size along the trough**
  For this test data, our BOT dispense perfectly.
- **Space size is more than sifter width**
  Run fine for sufficiently spaced plant rows.
- **No plants in trough**
  Run fine but dispensing is not uniform.
- **No spaces**
  BOT scanned the whole trough and came back to home position without doing anything.

8 System description

The bot consist of following major components.

- **Sifter and container holding attachment:**
  To hold and dispense pesticide.
- **Lifter Arm attached with sifter and container holding attachment on one side and servo motor on other side:**
  Used to hold sifter and container and move up and down.
- **Side IR sensor:**
  To detect the space and align sifter exactly in front of space.
- **Servo motor with a square plate attached, placed on sifter**
  This is used for knob open/close mechanism.
- **A heightened platform**
  This is used for placing lifter arm assembly and provide hight so that sifter would not collide with plants.

8.1 Important Problems Faced

- **Building lifter mechanism**
  For lifting Sifter and container upward and downward we first use a up/down sliding mechanism, but when we built it we found that the weight of container, sifter and servo motor placed at sifter causes more friction. Because of that slider moves downward perfectly but have problem for moving upward. We tried different things to solve it, like put two servo motor instead of one, stop downward motion on a significant height etc. But all ideas failed for it.

  Now we tried a forward backward slider mechanism which also failed because in this case sifter will collide with plants which is not affordable. We also modified it so that it never collide with plants but in this case we hold the sifter from one side not from the middle which causes bending of sifter. We tried so many ideas to overcome it, like provide additional support for other side of sifter, but it not worked.
- **Sifter bending**
  First we tried a sifter built by cardboard but it bends because of weight. So we tried a aluminium sifter, but it was weighty and bending on one side. Now we tried plastic sifter. It is light but also bend from one side. At last we modify our mechanism and hold the sifter from the middle instead from one side.
- **Balancing problem and non straight movement of bot**
  At last we found a working lifter arm assembly idea which uses an arm holing sifter from its middle. This idea have a problem of unbalanced weight on bot because the weight of container and sifter is on one side. So bot's moving wheel rotates with different speed because one side

have more weight than other. We solved it by putting some additional weight on other side of bot.

- **Wrong white line sensor readings**
  Sometimes white line sensors give wrong readings. If they are not showing stable readings even if the bot is stable then their are problem with the sensor sensitivity. We felt this problem when our bot was not following black line but at the same time we think that this happening may be because of weight problem. So we checked the reading for stable bot and found the problem. Solution for it simple. We just have to set the sensitivities of sensors using screwdriver.

- **Checkpoint detection**
  As we were fighting with Balancing problem and non straight movement of bot with Wrong white line sensor readings. Our bot sometimes missed to detect the checkpoint which causes many problems because bot needs checkpoint detection for localization. We are confused with all these 3 problems. Which one actually causes current problem. Finally we got the solution by checking for black line reading with less interval.

- **Knob open/close mechanism**
  We got a problem with knob because the weight of container vary with time. When container was full, knob was not able to hold the weight and pesticide spills out from side. Solution for this is a good mechanical knob but we are unable to find because we need a proper size and weight and in general they are big and weighty.

9 Future Work

- More precise and sophisticated methods using image processing can be used to identify the spaces based on size of space.

- Proper mechanism for knob can be implemented.

- More than one type of pesticide and more than one trough, combination for dispensing can be implemented.

10 Conclusions

Manual pesticide dispensing is a monotonous and frequent activity involving large amount of resources(labour, time and money). However, using an automated bot can do the same activity given energy requirements are taken care of. This way, not only manual labour can be put to more sophisticated

work, but also using a bot for pesticide dispensing will improve the yield of the greenhouse.

The things that seem to be achievable using software simulation is not always accomplished as expected. So the feasibility of the project (especially an embedded project) should be analysed from a hardware perspective ,Real time analysis. Simulation is essential. Consulting an experienced person in the same area shows the best path.

11 References

[1] WinAVR User Manual 20100110 [url : http://dybkowski.net/download/winavr-user- manual.html]
http://e-yantra.org
http://www.wikipedia.org
Spark 5 documentation
Spark 5 hardware and software manuals