PROJECT REPORT

CS684 Projects

Automatic Seed Sowing Robot

GROUP-3

Deepak Bhat, 113079007

Newton, 113079018

Ashwin Radhakrishnan, 113079011

1. Introduction

The aim of our project is to automate the seed sowing process in a Greenhouse. The user enters the trough number that has to be sown, through a GUI located remotely. This information is communicated to the bot using zigbee protocol. The bot moves to the location of the trough without manual intervention and sows the seeds at appropriate interseeding distances.

2. Problem Statement

The bot, prefilled with seeds, is placed in an arena consisting of troughs and aisles. It waits for the remote user to enter the trough number. As soon as the trough number is entered, the bot starts navigating towards it. The bot aligns itself with the troughs as it moves through the aisle using the sharp sensors mounted on it. On reaching the trough, the bot activates a servomotor, attached with a flap, to drop a single seed. Wheel encoder information is used to calculate the distance moved by the bot. This distance information is used to activate the flap mechanism so that the inter-seeding distance of the crop is maintained.

3. Requirements

3.1 Functional requirements

- The remote user enters the trough number through a GUI
- Zigbee protocol is used to communicate this information to the bot.
- Once this information is received, the bot starts navigating without user intervention, i.e. the robot is self-contained.
- The starting point of the BOT is always fixed
- Bot maintains proper distance with the troughs using Sharp sensor data.
- The bot must reach the appropriate trough by using the following features provided in the green-house
    1. Vacant space: - This is space between two troughs in a column or the space at the end of the column of troughs. This space is detected by the bot using the sharp sensors.
    2. The bot keeps track of the vacant spaces it has gone through to reach the appropriate trough
- The Bot must use a suitable seed dispensing mechanism to sow the seeds at appropriate inter seeding distance (between x1-x2 units)

3.2 Non-Functional requirements

- The time taken for the entire activity of seed sowing(i.e. the time instant from which it receives the start command to the time instant at which sowing is finished) must be predictable.
- The energy consumed for the activity of entire seeding process must be

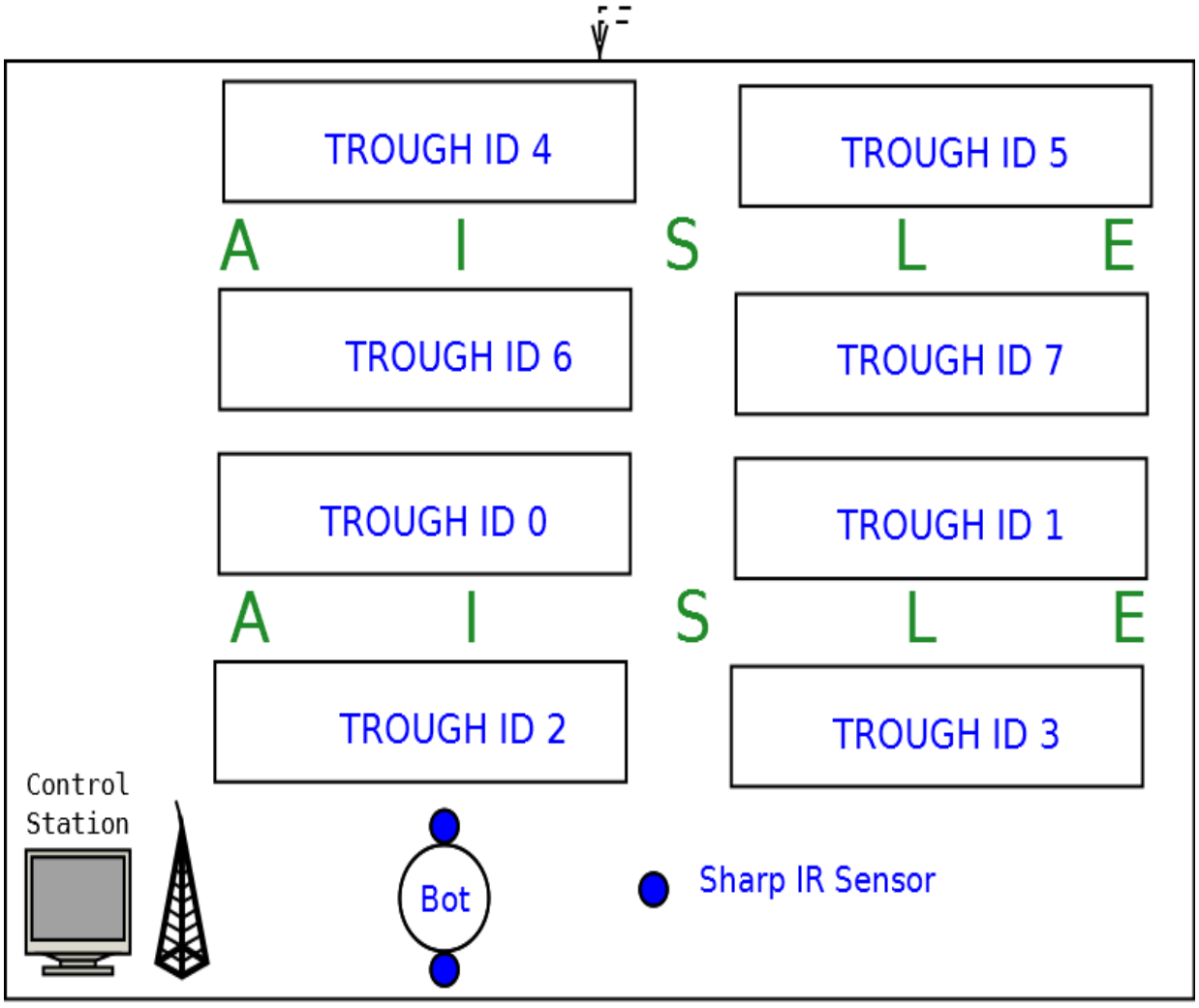
predictable. This includes energy attributed to tasks of navigating to the trough and seeding the trough.

- The bot should not drop more than one seed at the seed sowing location.
- The bot/arm MUST not hit the troughs/walls.

3.3 Hardware requirements

- Wheel encoder
- Zigbee USB adaptor
- Zigbee transmitter and receiver modules.
- Onboard battery support
- Sharp sensors
- Servomotor
- Fire-bird V

4. Arena Architecture



5. Implementation

5.1 C code description

One of the important features of the ATMEGA 2560 processor, present on the Firebird V, is the interrupt handler. Interrupt handlers have been utilized in our C code.

The Interrupt handler is a subroutine in microcontroller firmware, operating system or device driver that gets triggered on the reception of interrupt.

In our code, interrupt handler has been used to serve the real time trough number data sent by remote user through Zigbee interface. It has also been used to measure the shaft count. The shaft count data helps in measuring the distance travelled by the bot, required for maintaining interseeding distance.

Detailed explanation:

1. USART0 is initialized and then controller goes into an infinite loop waiting for an interrupt to arrive from the Zigbee module.
2. As soon as some data is received by the Zigbee module, the interrupt handler starts executing.
3. The data received would be of the form **BOTID$TASKID$TROUGHID#**. In our case, the BOTID has been set to 2, TASKID to 1 and the TROUGH ID can be in the range 0 to 7.
4. The ISR calls the function automaticSeedSowing(**TROUGHID**) using TROUGHID as the parameter.
5. Functions called within automaticSeedSowing()
   a) init_devices():- To initialise sharp sensors , servomotors etc
   b) initVacantSpaceStruct():- Initializes different variables of a structure pstVacantSpace based on the trough id received. These variables determine after which vacant space the bot has to start sowing (vacantSpace_SeedSow) , stop sowing (vacantSpace_SeedStop), when it has to rotate (vacantSpace_Rotate) and when it has to return (vacantSpace_Return). It also has a variable (vacantSpace_Count) which counts the number of vacant spaces the bot has gone through.
   c) alignment ():- This function deals with alignment of the bot with the troughs. For carrying out the alignment the bot continuously monitors the left and right sharp sensor values. We had arrived at the threshold values for the sensors through calibration. The threshold values and their definition in the code is as shown below.

```
#define LEFT_THRESHOLD_min      107
#define RIGHT_THRESHOLD_min     65
```

```
#define LEFT_THRESHOLD_max      117
#define RIGHT_THRESHOLD_max     75
```

The bot is aligned if the sharp sensor data is within these thresholds; else it follows the alignment strategy explained in **Section 6.2** to align itself correctly. In this function a check on the variables in the structure pstVacantSpace is done to see whether it has reached the destination trough. On reaching the destination trough a flag variable bCanSowSeed is set. The function activates the servomotor to drop a seed whenever this flag is set and the bot has moved a distance of 5 cm along the destination trough.

6. The code has interrupt handlers to increment right shaft position count and the left shaft position count. This information plays a very important role in the distance calculation and to rotate the robot through specific angle.

6. Testing Strategy and data

Testing strategy was devised to test the system against various functional as well as non-functional requirements.

The main requirements of interest were:

1) Interseeding distance.
2) Predictability of time in sowing the seeds and navigating to the trough
3) Predictability of energy consumed in the process of navigation as well as seed sowing apart from other safety requirements as in, avoiding collision with the wall/trough etc.

6.1 Interseeding distance

For the first requirement, i.e. interseeding distance, a metric namely variance of the interseeding distance was assigned to test the performance of the system. The bot was given a command to sow the seeds. The picture after sowing was taken offline and fed to MATLAB code. The MATLAB code was used to process the image to obtain the seed location. Pink colored beads were used to represent the seeds. The RGB value of the particular shade of pink was known. MATLAB code scanned the image, segment by segment and by applying thresholds for R, G and B values, the locations of the seeds could be identified. Once the coordinates of the seeds were known, distance between the seeds was calculated and mean and variance were determined. **Data**: The mean = 8.13cm

Variance = 0.8051 cm

6.2 Predictability of time in sowing the seeds

To measure the predictability of the system in terms of time taken to sow the seeds, a test code was written to make the bot move to a particular trough location sow the seeds and come back to its start location. The observation is as tabulated below.

Round No:	Time Seed Sowing (s)	Time in Return path (s)
1	129	81
2	135	70
3	120	75

We see that the mean of the total round trip time is 203.33s and the variance is 10.77s.

6.3 Predictability of energy consumed in navigation and seed sowing

The battery was fully charged and the voltage level was observed. The same test code as before was used. The battery voltage level was observed after the bot started to beep, indicating battery low. Average power consumed assuming both 2100mAh and the deteriorated capacity of 1000mAh is calculated. The following list summarizes the observations and calculations made.

- **START Battery Level : 9.706 V**
- **END(Discharge) Battery Level : 7.811 V**
- **Number of "round trips" made by bot :12**
- **Number of seeds sown in 1 trough : 8**
- **Average time taken for 1 round :196 sec**
- **Time taken for 12 rounds : 39.2 min**
- **Average current consumed (assuming 2100mAh) =3.2 Amps**
- **Average Power consumed :28.0272 Watts**
- **Average current consumed (assuming 1000mAh) =1.5 Amps**
- **Average Power consumed :13.125 Watts**

7. Discussion of the system:

Challenges that we had faced during our implementation were:

1) Dropping the seeds one by one (Requirement: Bot should not drop more than one seed at a time)
2) Alignment of the BOT

7.1 Dropping the seeds one by one

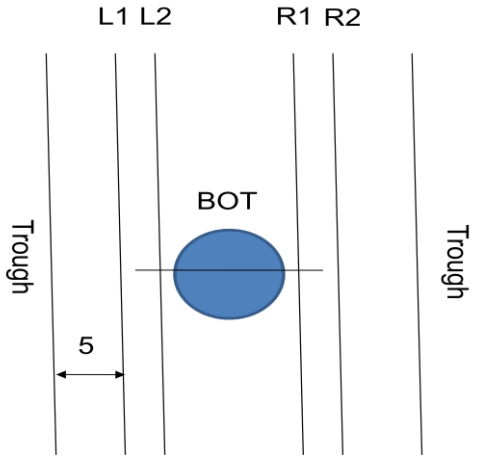
A mechanism had to be used, that could dispense the seeds one by one. The mechanism had to consume minimal energy and had to be simple enough to be implemented in a robust manner. Various mechanisms like vacuum based pick with robotic arm, robotic arm with gripper were considered, but later discarded because of feasibility in terms of power consumption and complexity. A simple and a robust mechanism involving tubular container and a servomotor attached with a flap mounted on with a thermocol groove was developed which could perform the job of dispensing robustly and efficiently.

7.2 Alignment of the BOT:

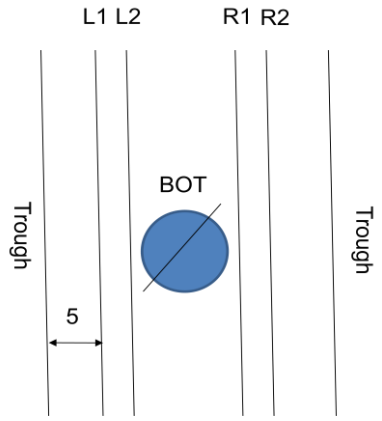
Alignment of the BOT was a challenge because of following reasons:

1) Sharp sensors are very sensitive
2) The wheels could not be rotated by precise amounts

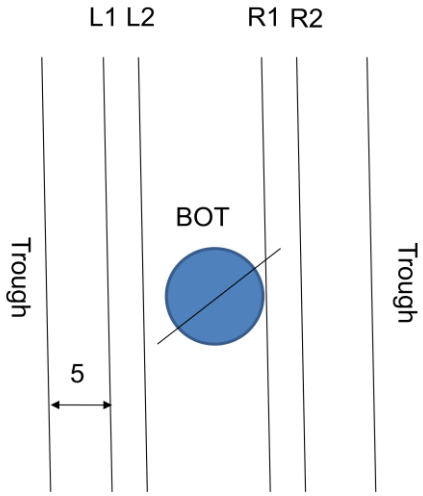
The alignment algorithm used is described as follows:
The ends of the line on the BOT indicates the location of sharp sensor on the bot. L1 L2 are the left min and left max thresholds, R1 and R2 are the right min and right max thresholds respectively. Based on the location of the trough sensor wrt thresholds certain actions will be taken as described below:



Action: Move forward

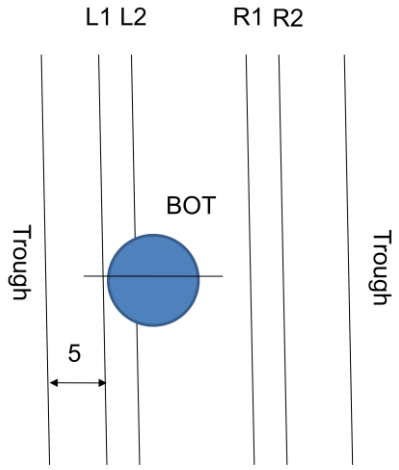


Action: Right Rotate by x degrees If Distance readings increased further, Left rotate by 2x degrees



Action: Soft Right Rotate by 4 degrees -> if Left sharp sensor distance decreased move back by 10mm

Else Soft reverse left rotate by 8 degrees -> move forward by 10mm



Action: -> Soft Right rotate by 4 degrees

The key challenges that were faced were to decide the thresholds and to identify the proper alignment algorithm. The alignment at 0 degrees is key to this project as the interseeding distance could vary vastly due to misalignment. The difficulty in assigning the threshold arises because, if the thresholds are relaxed then the BOT could misalign, where as if the thresholds are tighter BOT takes a lot of time to align at the “perfect zero” angle that results because of stricter thresholds.

8. Future Work

System can be fully automated, where the user has to just enter the trough ID, crop type and time to sow and the bot moves to particular locations based on schedules

Inter-seeding distance can be dynamically changed based on the crop chosen by the user.

Overhead camera to provide on the fly feedback to the bot, to maintain more precise inter-seeding distances.

Feedback mechanism to ensure that the bot moves only after dropping seeds at the sowing location can be used to make the system more robust.

Bot can suspend current work and can set back to the start position when the battery level reaches threshold.

The wheel cannot be rotated by the right amount because of various reasons like motor slip, miss of position encoder pulse, motor inertia etc. Some kind of check point based feed-back mechanism in the arena can be implemented so that the system knows “how much” to activate the motor.

9. Conclusion

The system that is developed is fully automatic except for the part where the user gives the command to start sowing the seeds. The fact that we used sharp sensors for navigating through the troughs encourages its applicability in the actual green house environment, along with line tracer. If the line is missed or stained, information from the sharp sensor can be used to navigate independently.

10. References

[1] Firebird-V ATMEGA2560 Hardware Manual
[2] Firebird-V ATMEGA2560 Software Manual
[3] Xbee/Xbee-PRO OEM RF Modules Manual