CS-684 PROJECT
On
Aerial House Keeping Bot

Submitted by

Group No 13
Raj Agrawal 11305R004
Prashanth Kazipeta 09007033
Pranay Kumar Myana 09007034
Hricha Kabir 113050079

Under Guidance of
Prof. Kavi Arya & Prof. R. Kriti

Indian Institute of Technology, Bombay

November 2012

Table of Contents

1 Introduction 3
2 Problem Statement 3
3 Requirements 4
4 Interfaces 4
5 Techniques 5
6 Implementation Details 6
7 Test Strategy and Data 7
8 Final System 8
9 Challenges 9
10 Future Work 10
11 References 10

1 Introduction

In this document we present a detailed description on Aerial House Keeping Bot which we propose to design that includes purpose, features and also the constraints under which it must operate.

The project involves programming the robot to control the motion of a black box in a green house. The black box can be used for aerial surveillance, aerial spraying of nutrients, water sprinkling etc. The user provides the trough-coordinates of the destination as an input to the bot, which then moves the Black Box (BB) to the destination by means of pulleys and chords. The black box is suspended in the air using chords which is fastened to the wheels of firebird through pulleys fixed on the walls of the arena. BB is then moved from one location to another by winding or unwinding the chords. The pulling/releasing action is accomplished by rotating the wheels of the bot in an appropriate direction. Pulleys are used to change the direction of the chord. There will be a camera mounted at a fixed position above the black box, which will continuously monitor the position of black box and will send the feedback to the firebird about the accuracy of the position. The input coordinates and the feedback values are transmitted to firebird using wireless Xbee module.

Definitions, Acronyms and Abbreviations

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

ATMEGA2560: The microcontroller used by Firebird V robot

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

WinAVR: Open-source software which uses AVR GCC compiler

AVR Studio: Open-source software which uses AVR GCC compiler

Black Box (BB): Any device depending on the functionality required

2 Problem Statement

The aim of this project is to design and develop a system which can move a Black Box (BB) aerially and make sure that the BB is able to visit all the troughs present in the green house arena. Include a feedback system to ensure that the error in position is well below an acceptable value. The movement of the BB should not be too slow to be practical.

3 Requirements

Hardware

6 Pulleys

Chords

3 DC motors and extra wheels

USB Camera

1 Firebird V robot

2 Zigbee modules

1 Black Box

4 Interfaces

User Interfaces

User will input the Trough-coordinates of any trough in the green house where the black box is intended to move.

Hardware Interfaces

ATMega2560: Master Microcontroller in Firebird V

ATMega8: Slave Microcontroller in Firebird V

RAM and Flash Memory: Included in the microcontroller

Power Supply: External 7.4V, 1800mAh Lithium Polymer Battery

Black Box: Any device depending on the user requirement

Pulleys & Chords

Software Interfaces

Compiler: AVR GCC (Specific to Atmel AVR processors)

Development Tools and IDEs: WinAVR, AVR Studio 4.0

Communications Interfaces

Zigbee for communication between bot and the camera.

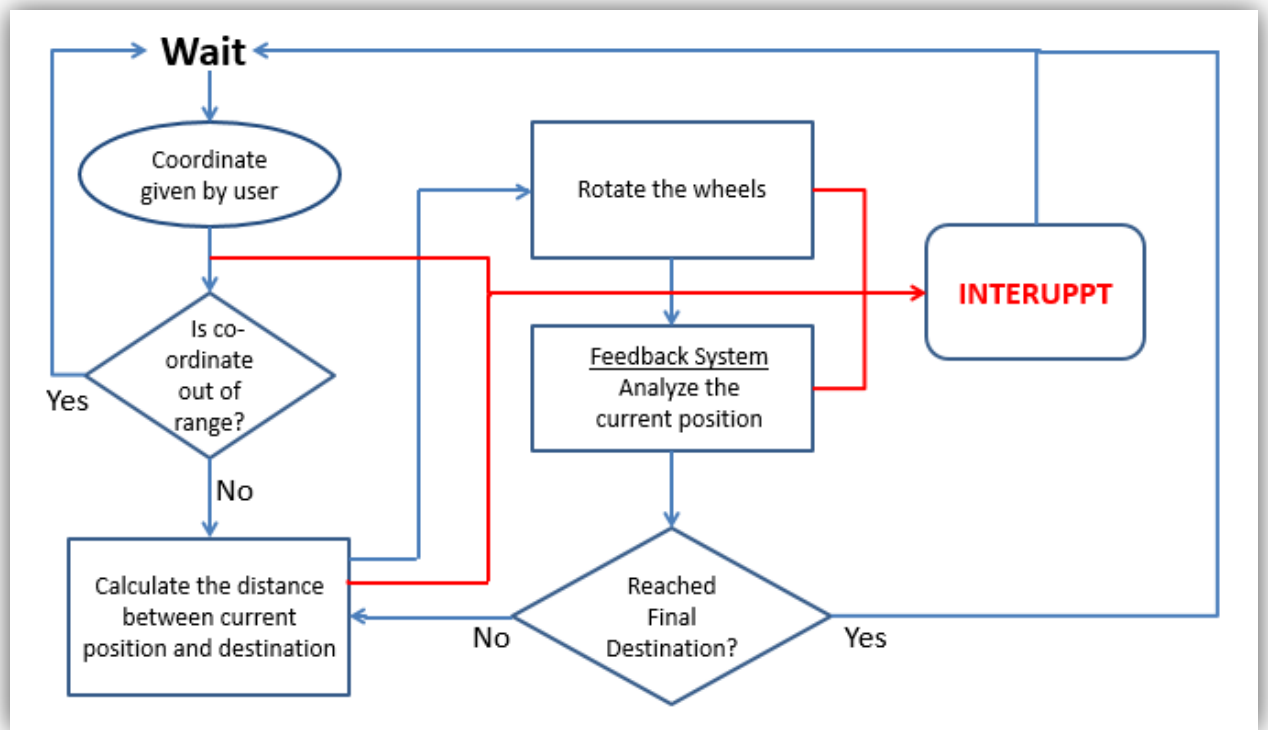
*Trough-Coordinates: The troughs in the green house arena are given particular coordinates to facilitate the user providing the inputs

5 Techniques

Description: We are implementing the movement of Black Box within a triangular arena (due to hardware constraints described later). The black box is suspended in the air using chords which is fastened to the wheels of firebird through pulleys fixed on the walls of the arena. BB is then moved from one location to another by winding/unwinding the chords. The required action is accomplished by rotating the wheels of the bot in an appropriate direction. The BB is moved from one location to another location directly updating the reference position without moving to the starting point. Working of housekeeping bot is described below

- Receive the input coordinates from the user and validate the coordinates.
- Calculate the required amount of rotation of each wheel
- Rotate the wheels of the bot.
- Find the actual position of the black box using the camera provided (feedback system).
- If according to the feedback system, bot is not in destined location, then move the BB to the actual destination by rotating the wheels accordingly
- Wait for another user input

The flowchart below depicts the overall architecture of housekeeping boot and latter section provides detailed description of three steps of geocoding.



6 Implementation Details

Input Passing: Green house is composed of troughs and each trough is identified by trough number. We are considering a triangular shaped greenhouse arena. One point is considered as origin and all troughs are numbered with reference to origin.
Input is passed to bot in form of (x, y) co-ordinate using zigbee module.

Validation: Compare the given input co-ordinates with greenhouse co-ordinates. If input is out of range, then send an error message to the user.

Rotation/Distance Calculation: Analyse the current bot position and calculate the distance from current bot position to destination position for each wheel then rotate each wheel by its respective distance.

Feedback System: Bot sends a request to feedback system for current position of BB; feedback system sends the co-ordinates of current position of BB to the bot. If the error in the difference between current position and the destination is above a predefined value, then calculate the distance accordingly and rotate wheels.
This process continues until the error is well within an accepted range.

7 Test Strategy and Data

Test Strategy:

- The initial position of the black box is fixed as coordinates (0, 0) in the arena.
- We have then tested the bot by giving the input as trough no. where the black box is to be moved.
- The bot will map the trough nos. To the actual coordinates in the arena, and will move to the required destination coordinates.

Expected Output:

- The black box should be moved to trough no. given as input.

Actual Output:

- The black box has moved to the destined trough no. with a small error.
- The error is then corrected using the feedback system.

Test Case 1:

Destination: (1, 1) Source: Origin

Test Case 2:

Destination 1: (1, 3) , Destination 2: (2,1) Source: Origin

Test Case 3: Repeatability

Sequence: Origin ⟶ (1, 1) ⟶ Origin ⟶ (2, 3) ⟶ Origin ⟶ (1, 1) ⟶ Origin

8 Final System

- Instead of moving BB in an rectangular greenhouse arena, the arena is constrained to be triangular
- The chords connecting the BB and the wheels of the bot run along the walls through the pulleys fixed at the top and bottom of the lateral walls.
- The BB moves to the new location from the previous location directly without moving to the origin by updating its reference to the previous position.
- Due to issues with collaboration of image processing with actual system, Instead of sending a request from bot to feedback system and awaiting its reply, we provide the coordinates of actual position manually.

9 Challenges

Interrupt couldn’t be generated for C1 motor

- We rely on position control using interrupts to control the amount of chords.
- In order to release/pull a particular length of a chord, the rotation of the wheels is monitored by Shaft encoders
- Shaft encoders generate interrupts to provide the amount of thread released/pulled
- But, by design, interrupt generation pin of C1 motor, is allotted for other functioning of bot so interrupt for C1 motor has been disabled which restricts us to three motors to work. So we had to fall back and restrict our motion to a triangular greenhouse arena.

The orientation of the wheels causes a major problem during winding/unwinding the thread

Dynamic nature of the circumference of the cylindrical surface on which the thread is wound

- As the thread is wound/unwound, the change in the number of layers of the thread leads to an error in the circumference of the surface
- Hence, the actual amount of thread released/pulled will be different from the intended

Motors couldn’t handle a little higher torque demand and stopped working. So we had to restrict the weight of the Black Box.

10 Future Work

1. Extrapolate the design to move the BB to any point in the green house (rectangular area)
    - Use another bot to control the fourth distance parameter
    - As we can control only 3 motors, control the 3rd coordinate using a movable pulley to cover a rectangular area
2. Introduce vertical motion by
    - Modifying the mathematical model (OR)
    - Using another motor to move the entire framework of the pulleys vertically
3. Place the wheels right below the pulleys to avoid
4. Image processing can be used to provide the feedback
    - Image processing can be used to provide the actual location of the BB, which is being done manually
5. Processing a string of inputs without interruption
6. Develop a mechanical system which can be used to wind and unwind the thread without any hassles

11 References

FIRE BIRD V ATMEGA2560 Robotic Research Platform Hardware Manual. IIT Bombay & NEX Robotics Pvt. Ltd.

FIRE BIRD V ATMEGA2560 Robotic Research Platform Software Manual. IIT Bombay & NEX Robotics Pvt. Ltd.

ATmega640/1280/1281/2560/2561 Datasheet provided by ATMEL