CS302 Report



Group 16

Project: Gesture Controlled Robot

Deepankar 09005060

Haseeb 09005070

Gaurav 09005073

Ravi 09005074

**PROJECT OBJECTIVE**

Now a days it becomes very important to control the robots just with gestures. This can be very helpful in many of the real physical situations where there can be danger to a human being. It is also very important to make the gestures very intuitive so that human being can easily adapt to them.

**Problem Statement:**

Control the firebird and gripper using gestures. Detect user gestures using kinect, Convey these gestures as signals to firebird, Make firebird/gripper take appropriate actions.

**REQUIREMENT SPECIFICATION**

Interpret a subset of gestures as commands. For intuitive gestures there is need to send the data continuously and then adapt the bot according to the data or the human psoture

**HARDWARE PLATFORM**

1. Firebird V ATMEGA2560 (Cost=Rs.16,875.00)
2. Zig-bee (XBee 802.15.4 OEM RF module 2.4GHZ) is used for communicating between the firebird and PC (Cost = Rs.1665).
3. Microsoft Kinect for recognizing the gestures

**SOFTWARE**

1. AVR Studio 4
2. Microsoft Kinect SDK
3. Visual Studio (for coding the program related to kinect)

**EXTENSIONS USED**

1. A gripper is attached to the firebird for collecting the ball. Three servo motors are used for this. Two servo motors are used for upward and downward movement of the gripper and one is used for opening and closing the arms of the gripper.

1. Thermocol attachment is placed in front of the robot to place camera and the two sharp IR sensors used.

**CODE DESCRIPTION**

**CODE FILES**

|  |  |  |
| --- | --- | --- |
| Filename | Purpose | Executes on |
| Zigbeegp17.c | Main Program | Robot |
| LCD.c | Handles the display functions of LCD screen. | Robot |
|  |  |  |

**PROJECT FILES**

|  |  |  |
| --- | --- | --- |
| Filename | Contains |  |
| C-code.rar | SourceCode of programs to be burnt on Robot.  Contains documentation of the code as well. |  |
| PC-interface.rar | Contains Matlab files. |  |
| Documents.rar | Contains Project related doc files. |  |

**SYSTEM DESIGN**

STATE CHART DIAGRAM

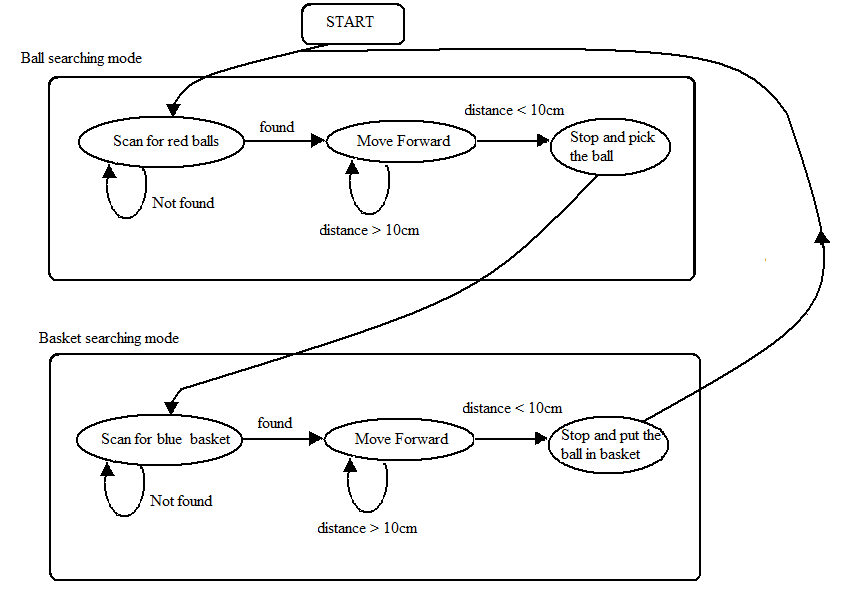


Fig. State Chart of the tennis ball collector robot

The robot works in two main states: the ball searching mode and the basket searching mode. The state chart given above describes the working of the tennis ball collector robot. In the ball searching mode, the robot searches for red coloured balls. If not found, it keeps on searching for balls. If ball is found, it goes towards it and picks it up. Then the robot switches to basket search mode where it searches for blue coloured basket. If not found, it keeps on searching for it. If a basket of blue colour is found then it goes towards it and drops it. After that it goes back to the ball searching state.

**ASSUMPTIONS AND LIMITATIONS**

1. Very fast gestures can’t be detected correctly. This is becaue the range of X-bee 0-128

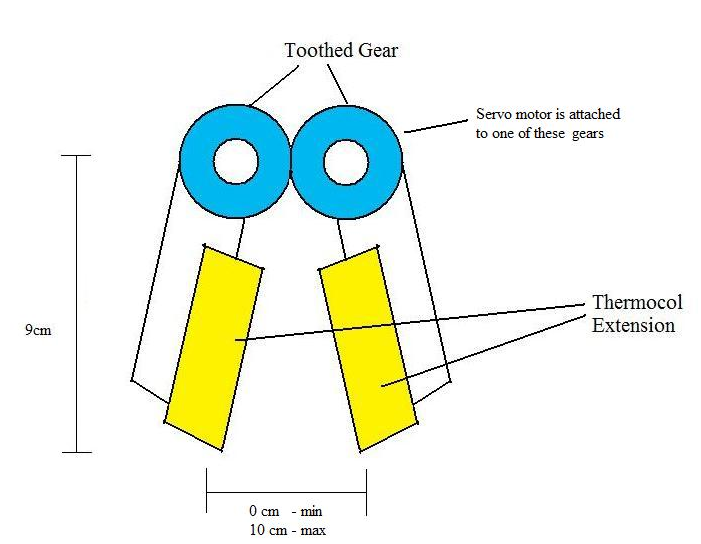
.

**SETUP AND EXTENSIONS ON THE ROBOT**

The gripper is a small attachment made of plastic, thermocol and servo motors for gripping the ball. The image of the gripper is shown in the figure below.

Materials required for making gripper are:

* Two toothed gear wheels.
* Plastic strips.
* Thermocol.
* Metal strips for holding the two wheels together.
* Three servo motors.

****

Plastic Strips

Fig. Diagram of gripper.

Two arms of the gripper are attached together using a gear arrangement. A servo motor is connected to one of the gear wheel. When this servo motor rotates, both the arms move due to the presence of gear wheels. A thermocol attachment is provided on the gripper for better catch. Actual photo of the gripper is shown below.



Plastic Strips

Thermocol

Metal strips

Fig. Photograph of gripper.

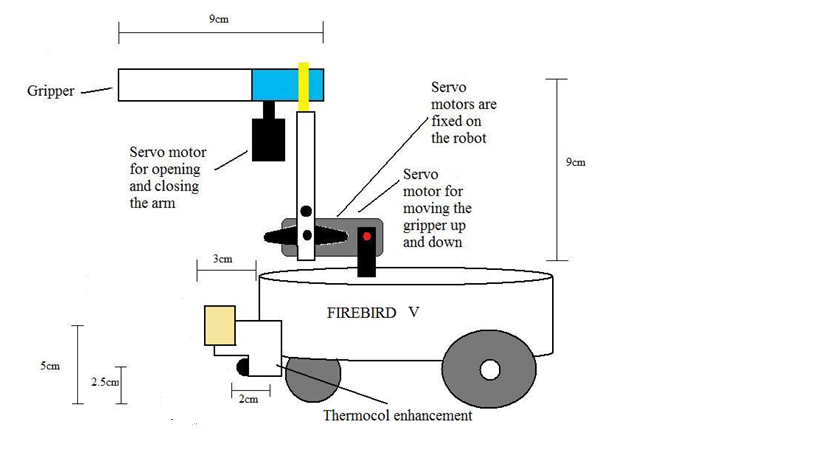
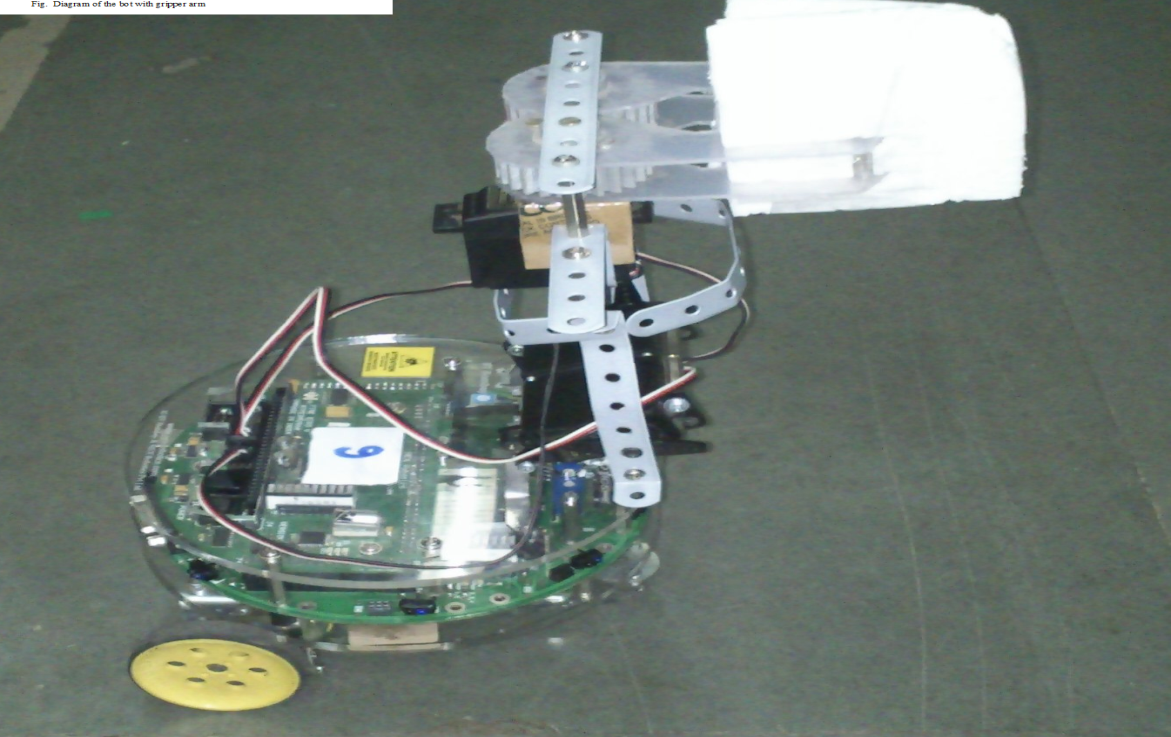


Fig. Diagram of the bot with gripper arm

Fig. 

Actual photograph of the bot

**IMPLEMENTATION**

The whole project has been divded to **three modes.**

**Mode 1:**

This mode is for Long range motion of the bot. So For this mode we have implanted three static gestures.

* + - Forward motion
    - Backward motion
    - Stop

To detect the above gesture we are measuring the consistency of few angles and then sending the data to kinect whether to execute which instruction

And for rotation purposes we have implemented a continuous gesture with hand such that kinect rotates exactly the same amount as when the hand rotates. For this to work we are finding the angle rotated and then pass it to firebird . firebird converts it to shaft count and then executes the command.

**Mode 2:**

This mode is for fine controlling of the robot. This is one of the most intuitive modes. For this mode you need to keep both arms in a certain posture and then move them as you wish. The kinect code we have written takes the position of arms samples them over a series of frames and then finds the change in the position. The data is send back to firebird first for left and then for right hand.

The firebird receives them and executes action appropriately

This mapping correctly correlates what is done

|  |  |  |
| --- | --- | --- |
| Left change | Right change | Motion |
| +ve | +ve | Fwd |
| +ve | -ve | Hard left |
| -ve | +ve | Hard right |
| -ve | -ve | Backwd |

Not only this if there is unequal change it first executes the equal part and then keeps one tire ideal and the takes a soft right or soft left according to situation

**Mode 3:**

This mode is entirely for gripper here also same situation as the mode2 but instead of sending the changes the mapping is direct the angle between shoulder is directly mapped to the of the holder of gripper arm, where as the elevation of our arms are mapped to the elevation of our arms

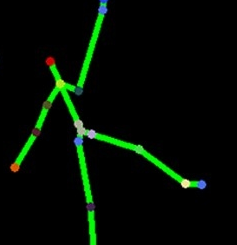
For this mode also like the previous mode we have found the two angles required and then send them one by one to the firebird to process

**EXECUTION INSTRUCTIONS**

* Open the project zigbeegp17 in AVR studio 4
* Compile the code.
* Connect the firebird V programmer on to the PC in which this code exists.
* Burn this code into Firebird V.
* Connect the wireless receiver of camera to the PC via a TV tuner card.
* Run the Matlab code on a PC connected with a Zigbee module (make change to the code according to the port number to which Zigbee is connected).
* Now switch on the Firebird.
* Make sure to connect the camera on the Firebird to the 9V power supply.
* Now the Firebird will start working and will respond to the code sent by the Matlab code.

**TEST CASES**

Mainly testing is done for verifying the right motion of bot. First the tests some of them failed due to the confusion that sometimes occurs as shown in the image below



This thing mainly occurs because due to the lot of disturbance in the background of where the person is standing

So if the person is standing before a white background then the Tests are nearly perfect

No of tests conducted : 10

No of tests without a flaw: 6

Main causes of errors:

1. disturbance in the background

2. second person enterd into the picture

**DISCUSSION OF OUR WORK `**

***What worked as we planned?***

Zig-bee communication: our aim of making the bot wireless have worked and thus we are able to move the bot wirelessly this is crucial as in the real life we cannot have long wires

Accuracy of movement over gestures: The moving of bot according to the gesutures is pretty accurate and thus we achieved our goal

Intutive gestures: the gestures are pretty intuitive in mode 2 and angle rotation of mode 1

***What we have done more than SRS?***

Gripper Arm Control: we have not include about this is in the srs but we have done that to demonstrate how easily new modules can be added to our code

***What went wrong?***

RTOS: Because we have to use interrupts with in interrupts (first interrupt is from signal and the second is form the shaft wheel encoder) apparently we were not able to find a function to handle such things in the case of RTOS.

Another problem is that if a nested interrupt occurs the Rtos bluntly drops the interrupt which stops us from doing the shaft wheel encoding

**PROBLEMS**  `

Problem 1:

Main problems are occurred due to the errors that kinect gives while detect in some frames. For intuitive Gestures because we have implemented continuous signal sending if the variation due to imperfection of kinect is large then the bot moves arbitly and randomly

This problem has been detected by the group of the last time project.

As a solution to this problem we have introduced the sampling of the data points over a series of gestures this reduces the above mentioned problem of large variation

Problem 2:

Because we have to send two values in the mode 2 and mode 3 and also we are sending them continuously. So if the firebird process the previous two signals slowly then it ignores the next signal thus changing the order of the signals This has been avery serious problems.

Problem 3:

One of the biggest challenges in the project was the communication between the robot and the Kinect code using zigbee. The communication is slow compared to the wired communication.

**INNOVATION**

**INNOVATIONS:**

The main innovation we have done is the sampling part of the kinect.This is very keen because it solves the first problem mentioned above by default .The good part of it is that it also solves the second part mentioned above because it take some time to sample the code there will be delay from the kinect side and hence there by giving ample time for firebird to process

**REUSABILITY FEATURE**

Modularity of Gestures,Signal processing:

* Divided the entire class of gestures into different modes each one dealing with a different aspect of control.
* Controlling an entirely different thing can be done by specifying a gesture for transition into the new mode and writing code specific to the mode.
* Same applies to code on firebird.
* So a new feature requires writing of only two if else statements by specifying a new gesture

**FUTURE ENHANCEMENT**

* Speech feature of kinect can be explored to make new gesutres
* By changing the design of a gripper arm we can really make a bomb diffusing robot
* Use of bluetooth instead of zigbee to solve the synchronization problem without delays.
* NUI of kinect can be used to control lots of other bots besides firebird thus getting greater functionality.
* Building a gesture library for kinect.

.