Missile Detection and Interception systems

Prasoon Dhaneshwar

MT2015515

IIIT Bangalore

Guided by: Dr. Sachit Rao

IIIT Bangalore

Contents

I. Introduction

In this project, 2-D model representation of a missile interception systems is portrayed. Two bots are used to act as missile and interceptor, and a camera acting as a satellite to detect them.

With the help of Computer Vision and Image processing, camera detects both missile and interceptor. The system then computes the next move of interceptor when missile approaches the target.

Here, Arduino and MATLAB are used. Arduino to program the interceptor bot for directions, and MATLAB to detect and compute the next move for interception.

Look for folder “Images and Videos” to see the program working in various environment and different conditions.

Look through “fig (1)” to “fig (7)”, to see the working of detection and movement frame by frame.

Image named “figure”, explains the working of next move for interception.

Videos from 1 to 9 are different test cases in different environment, and positions of missile(stationary, moving from various directions).

II. Components used in project

i. Arduino Uno
ii. L293D Motor Shield
iii. Two 60 RPM motors
iv. Two 100 RPM motors
v. 12V 3000mah battery
vi. Adaptor for constant DC supply
vii. Two chassis and two sets of wheels
viii. Logitech HD Webcam C270
ix. USB male to female cable
x. 1m fibre rod for camera
xi. White board markers for colouring
xii. Connecting wires

III. Putting up the parts

i. Missile

Here, a DC supply of approx. 17 volts is given to two motors of 60RPM as shown in fig.1. It is fixed on a chassis of length 19.5cm x10.5 cm.

A red marker is also attached so that it becomes easier to detect the bot.

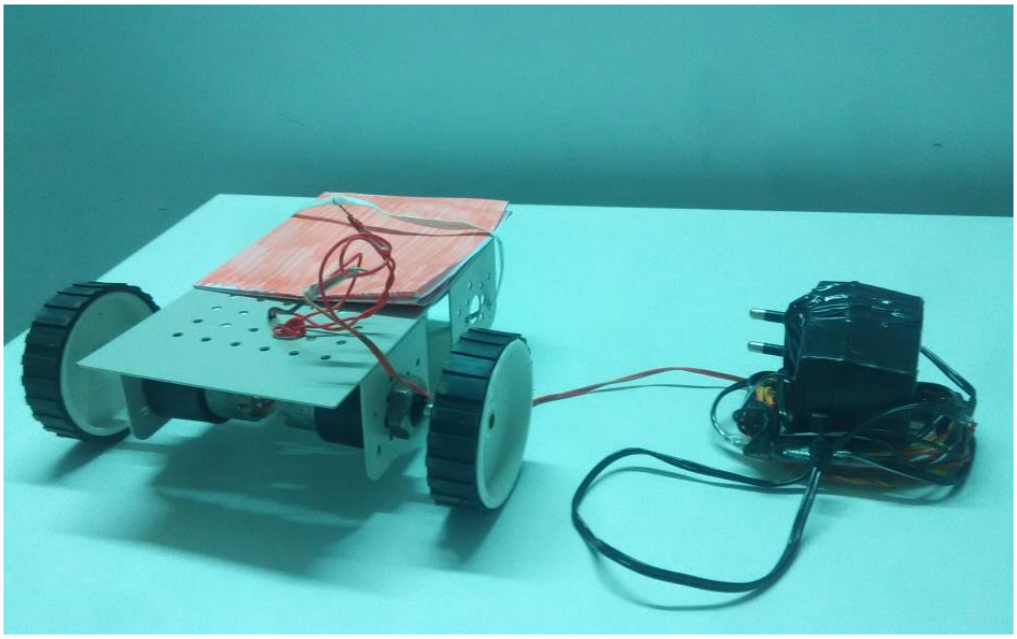


Fig.1. bot as missile with red marker and constant DC supply

ii. Interceptor

Arduino Uno is programmed to control the two motors of 100RPM, connected by L293D motor shield controller, and driven by a 12V battery, as shown in fig.2.
Bot is covered by blue and green markers for detection and control.

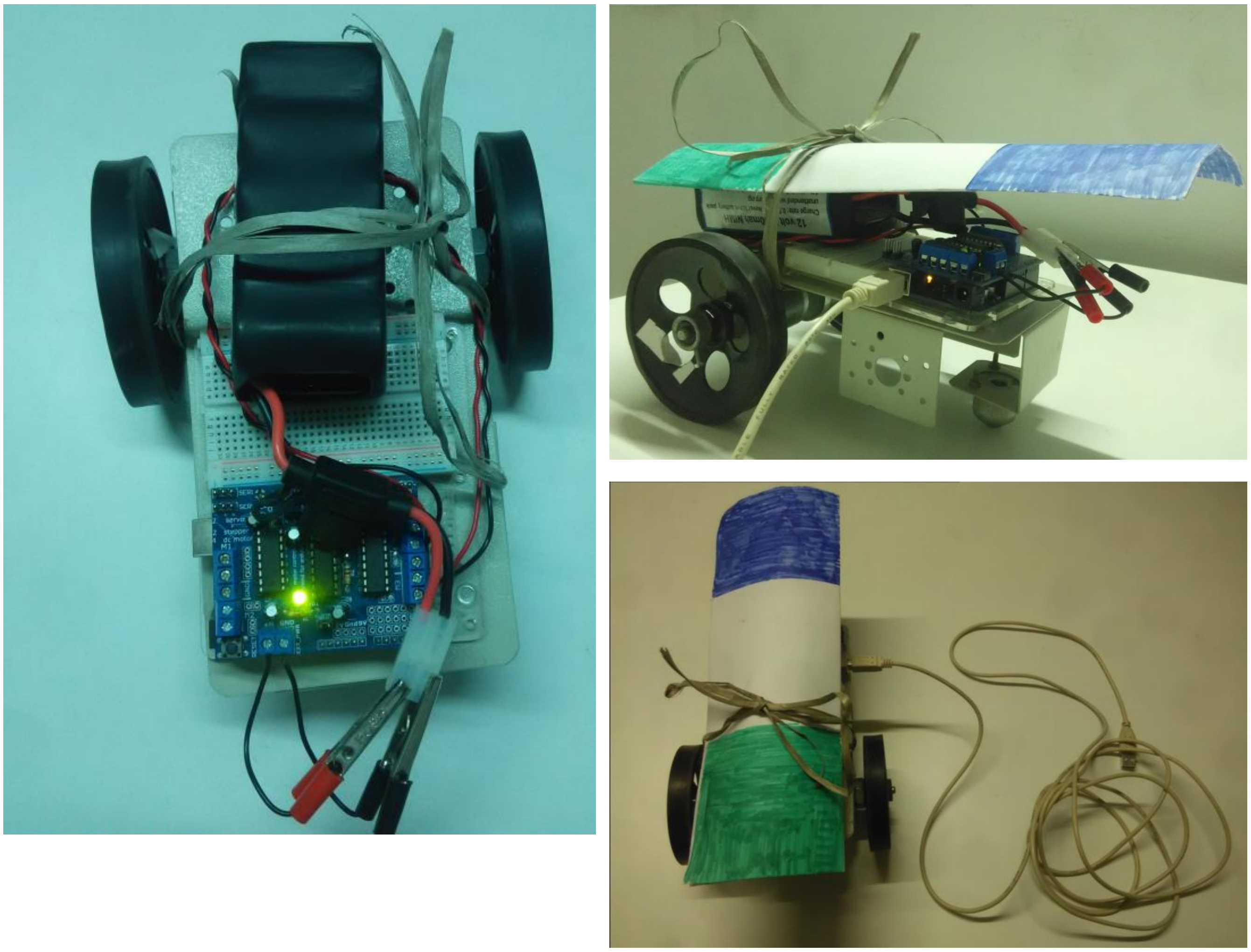
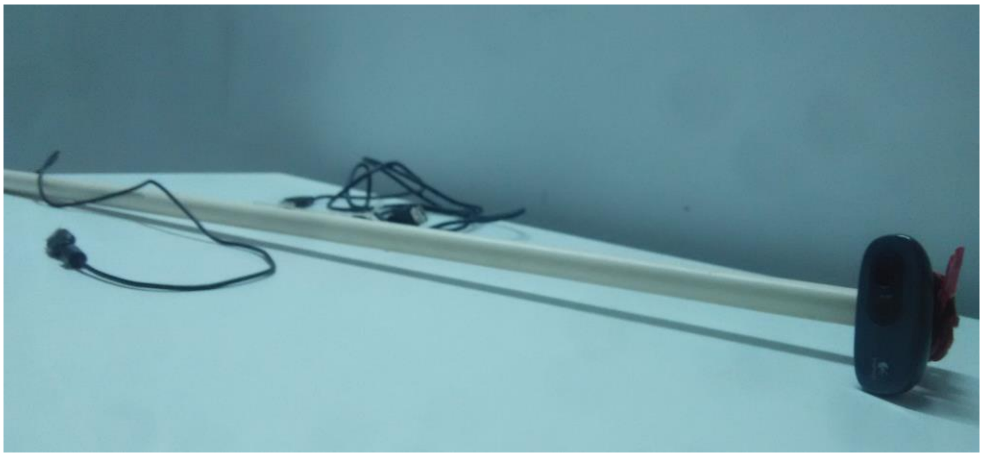


Fig.2. bot as Interceptor with green and blue markers.

iii. Camera

Logitech HD Webcam C270 is attached to a pole for getting a certain amount of height such that it grabs maximum area for movement of bots(fig.3).



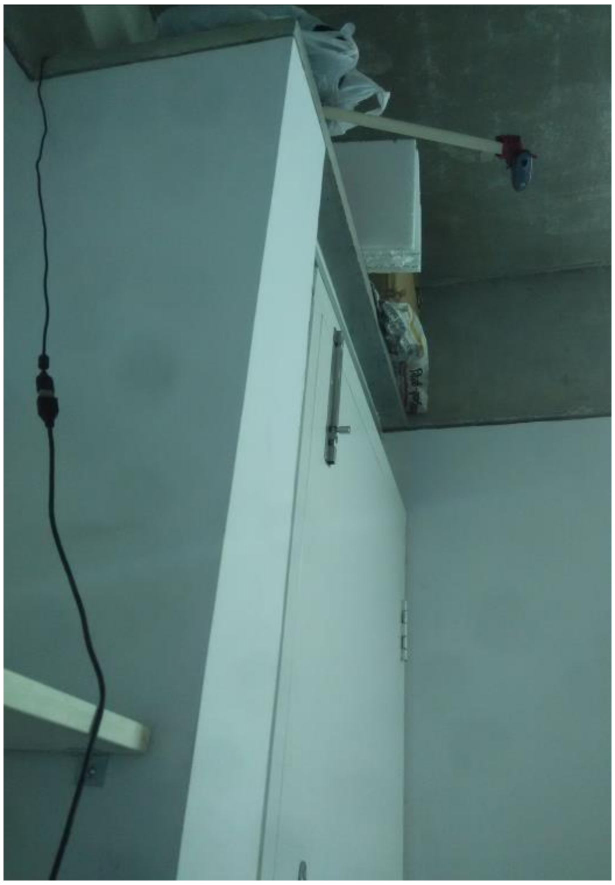


Fig.3. fixing camera to a certain height with the help of a pole

IV. Building guidance systems

Refer to the source code mentioned in subsequent sub-sections to understand the code along with the concepts used.

i. Programming the Interceptor bot

- Adafruit Motor shield library is used to program the motors. M1 and M2 are left and right motors controlled by PWM of Arduino. The maximum setSpeed is 255. To calibrate the motors correctly to move straight, left, right and stop, the values are modified accordingly[1].
- Refer _Arduino__Motor_shield_test_interceptor.ino for details.
- Ex: motor1.run(FORWARD) makes the motor M1 run, motor1.run(RELEASE) makes M1 stop.
- By communicating serially with a fixed baud rate (9600 bits per second here), characters can be sent via serial monitor to move straight, left and right.
- Here, ‘0’ = Stop, ‘1’ = Straight, ‘2’ = Left, ‘3’ = Right ‘4’ = Hard right.

ii. Detection of missile and Interceptor

- Refer MissileDetectionAndInterception.m for details.
- redThresh, greenThresh and blueThresh determines the threshold values to separate red, green and blue colors uniquely from image[2].
- Video objects are created to capture from webcam. Here, image acquisition toolbox is used[3].
- For blob analysis[4] (surrounding a bounding box), determining centroid coordinates and text showing the detection of missile above is done using hblob htextinsCent, hshapeinsBox and htextinsRed, htextinsBlue and htextinsGreen.
- hVideoIn is the video player for all the analysis.

- In the processing loop, image processing is done to create blob for red, green and blue. First, image is subtracted, Ex: Only red space with gray color space. Therefore, only red component of image is obtained.
- Finally the image is converted to binary with the red objects as white im2bw.
- Same process is done for getting green and blue components.
- The frame is then merged together with step function(see Line 111 to 114) to get all the components together.
- The analysis of centroids and updating them is done in the loops above. They are stored in centXRed, centYRed, centXGreen, centYGreen, centXBlue, centYBlue.

iii. Calculating the movement of Interceptor

- Function computemove.m takes care of the movement of interceptor.
- Based on blue and green markers, a vector **a** is defined, with red and green markers **b** is defined.

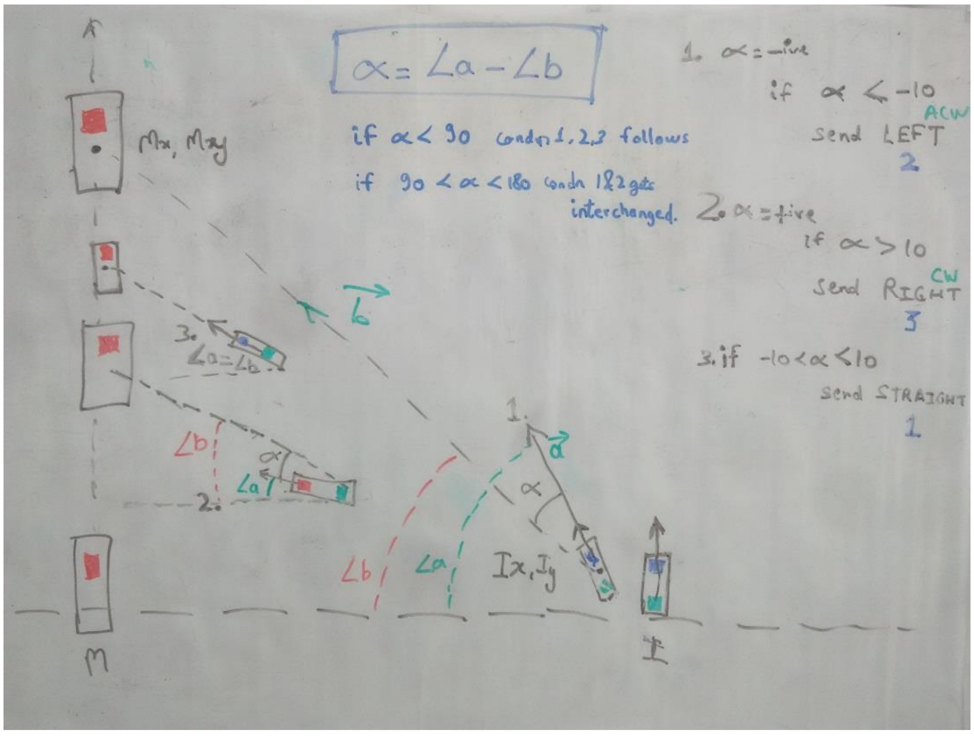


Fig.4. calculating the next move of interceptor.

- Angle α is the difference between angle a and b, which are defined with respect to a horizontal axis (ref = [800, 0]).
  From fig.4:

  Case 1: here α > 0 (in code, a tolerance angle is taken to avoid wobbling of bot every computation). Thus, bot is moved right to make α small by communicating serially[5] and sending ‘3’.

  Case 2: α < 0
  The bot now overshoots because of case 1. So, bot is moved left to make α small by communicating serially and sending ‘2’.

  Case 3: -10 < α < 10
  Now, α has come in a tolerance range to go straight. Hence, ‘1’ is sent serially to Arduino.

- In the else case of if, when 90 < α < 180, the cases gets reversed with respect to the horizontal(ref), since MATLAB calculates the shortest angle between two vectors. So, instead of sending right in else loop, left is sent and vice versa.

- After interceptor catches up with missile, the magnitude of **b** reduces. So, when it comes to a certain range(here, 80) and hits the missile, the interceptor stops and loop comes into break(calculated by distance.m). Thus, the path of missile gets intercepted.

After the loop, serial communication is stopped and video writing[6] function is also closed. All memory for camera object is released, and the program stops.

V. Overcoming limitations

- The threshold to detect red, green and blue becomes tricky when lighting environment is changed. So, it becomes hit and trial for all colours to detect simultaneously.
  This can be resolved if respective coloured LEDs are placed on top of them. Due to high intensity of colours, they wont get subtracted in the computation to detect, and will always show.

- The camera frame can be increased for more area coverage by mounting on a greater height.

- Noise can come when same color range gets in the frame, the interceptor then confuses and vector calculation will start from that noise. Therefore, a suitable environment is required for the computation to work.

VI. References

[1] Adafruit Motor Shield Tutorial

http://www.instructables.com/id/Adafruit-Motor-Shield-Tutorial/

[2] Red, Green and Blue Object Detection and Tracking

https://www.youtube.com/watch?v=QPzynkdsuW8

[3] How to acquire image through webcam using MATLAB 2014?

https://www.youtube.com/watch?v=cCYYugyI3Fg

[4] Labelling of objects in an image using segmentation in MATLAB

https://www.youtube.com/watch?v=XrC1r80-p-k

[5] Reading serial data in MATLAB

https://www.youtube.com/watch?v=divB4t1pJgo

[6] VideoWriter

https://in.mathworks.com/help/matlab/ref/videowriter.html

https://in.mathworks.com/help/matlab/import_export/export-to-video.html?s_tid=gn_loc_drop