**MATRIX BARCODE BASED DETECTION AND TRACKING FOR AUTONOMOUS UAV LANDING**

**A SEMINAR REPORT**

***submitted b*y**

**VELAYUDHAN A - AM.EN.U4EEE17101**

**NANDAKISHORE R NAIR - AM.EN.U4EEE17123**

**NEERAJ S - AM.EN.U4EEE17124**

**VISHNU S - AM.EN.U4EEE17147**

***in partial fulfilment for the award of the degree***

***of***

**BACHELOR OF TECHNOLOGY**

**IN**

**ELECTRICAL AND ELECTRONICS ENGINEERING**

****

**AMRITA SCHOOL OF ENGINEERING, AMRITAPURI**

**AMRITA VISHWA VIDYAPEETHAM**

**AMRITAPURI 690 525**

**MAY 2020**

i

**AMRITA VISHWA VIDYAPEETHAM**

**AMRITA SCHOOL OF ENGINEERING, AMRITAPURI, 690525**

****

**BONAFIDE CERTIFICATE**

This is to certify that the seminar report entitled “**MATRIX BARCODE BASED DETECTION AND TRACKING FOR AUTONOMOUS UAV LANDING**” submitted by

“ **VELAYUDHAN A - AM.EN.U4EEE17101**

**NANDAKISHORE R NAIR - AM.EN.U4EEE17123**

**NEERAJ S - AM.EN.U4EEE17124**

**VISHNU S - AM.EN.U4EEE17147**

in partial fulfilment of the requirements for the award of the Degree, **Bachelo**r **of Technology** in “ **ELECTRICAL AND ELECTRONICS ENGINEERING** ” is a bonafide record of the work carried out under my guidance and supervision at Amrita School of Engineering, Amritapuri.

Vivek A

Assistant professor

Dept. Electrical and Electronics Engineering

Dr Manjula G Nair

Professor & Chairperson

Dept. of Electrical and Electronics Engineering

ii

**ACKNOWLEDGEMENT**

Presentation inspiration and motivation have always played a key role in the success of any venture.

We pay our deep sense of gratitude to Dr Manjula G Nair (HOD) Electrical and electronics department Amrita School of Engineering Amritapuri to encourage us to the highest peak and to provide us with the opportunity to prepare the project. We are immensely obliged to our friends for their elevating inspiration, encouraging guidance and kind supervision in the completion of our report.

We feel to acknowledge our indebtedness and a deep sense of gratitude to our guide Mr Vivek Asst Prof Department of Electrical and Electronics Engineering, valuable guidance and kind supervision is given to us throughout the course which shaped the present work as it shows.

Last but not least our parents are also an important inspiration for us. So with due regards, we express our gratitude to them.

iii

**Abstract**

The autonomous landing of drones is always a field of interest for attaining tasks such as rapid deployment or recovery of UAVs, continuous flight tasks, mobile recharge stations, extended operational ranges. This system would be helpful in the warehouse management, delivery system, for security purposes. Creating and deploying a drone having a fully automated operational feature along with the focus of reduction in economic expenditure, attaining more accuracy over the landing zone with the effect of external disturbances like wind, lighting etc is the focus of the research.

**Table of contents**

| **SL NO** | **TOPIC** | **PageNo** |
| --- | --- | --- |
| **1** | **Introduction** | **5** |
| **2** | **Literature review** | **6** |
| **3** | **Navigation** | **11** |
| **4** | **Guidance(Trajectory generation)** | **20** |
| **5** | **Control** | **21** |
| **6** | **Conclusion** | **27** |
| **7** | **Reference** | **28** |

iv

**Introduction**

Dynamically Remotely Operated Navigation Equipment (DRONE) is a field of mass development having almost infinite scope in the current era of technological advancement. They are true mechatronic systems having elements of Mechanical, Electrical, Electronics, Software and Control Engineering. The amount of time required in constructing an entire vehicle to undertake research for our project is time-consuming and hence the use of an already constructed platform to perform the desired experiments would help in speed up the research.

There are various types of UAV based on the number of rotors, the design of the chassis, the operational feature, remotely operated or autonomous.

Here we have constrained ourselves in Quad-copter as they can be fully controlled solely by varying the speed of the four rotors and no mechanical linkages are required to vary the rotor blade pitch angles as with a conventional helicopter.

Matrix Barcode such as QR Code, April tag is a visual entity having the capability of storing significant data in a two-dimensional pixelated format which can be read using imaging devices such as cameras and processed by techniques such as Reed Solomon error correction techniques, and various other detection algorithms. They have the capability of omnidirectional encoding which permits high-speed reading by use of position identification designs.

There have been various upgrades in the field of autonomous drone propagation and control with or without using geometrical figures such as letters for landing zone, special patterns, circles polygons, two-dimensional barcodes which include apriltags and QR codes. Currently one of the most popular visual fiducial detectors and patterns is apriltag algorithm as they are robust to lighting conditions and view angle, with low false positive detection rates.

The Matrix code having data on a two-dimensional space and hence extracting the data from the location requires an imaging device and processors for data extraction. The above stage of data extraction from above said Code describes the possibility of it being Static. Our project focuses on a situation where the Code is in motion as well as making the UAV autonomous. The above task in motion Matrix Code recognition along with landing is planned to be performed in the research.

For proper indentation of the report, it has been divided into various parts.

**LITERATURE REVIEW**

The matrix barcode implementation has been in projects such as for a long time as they provide a basis of recognition of an object [1] whether it is moving or stationary. UAV are been used for a wide range of applications for their very versatile nature of mobility. As the applications increase, for a long operational implementation over a large area, the drone must be able to land at various spots either for recharge either by battery swapping or induction based drone recharging [2]. The method of autonomous landing consists of various steps such as control, the guidance of the UAV navigation etc But for easy recognition of the landing platform either for the above-said tasks needed for achieving long operational range task efficiency the landing platform must be recognised and localised. This task can be done in two ways either by continuous mapping for the environment with identification of the platform [3], using Visual SLAM and Extended Kalman along with feature point estimate using RANSAC in order to find an elevated platform for UAV Landing or with image processing of a platform based on symbols, letters, special characters or encrypted code. QR code platform detection can be achieved by the said paper below.

**Localization and navigation using QR code for a mobile robot in an indoor environment**

The navigation of autonomous robots in environments without GPS support is an area least explored. An approach for localization and navigation for a mobile robot in an indoor environment using QR code recognition is proposed in this paper [4].In this paper, the possibility of QR codes as a landmark for autonomous robots is explored and explained.

In this paper QR codes are used as a landmark which provides relative location and directions to the autonomous robots or UAVs.Exposure adjustable cameras are used to provide the input feed to the robot.

Unlike the usual QR code detection scenario a different configuration of the Z-bar library is used in this process. The video feed from the camera is acquired by the opencv for image processing and QR code detection when a QR code is detected an image is captured by the camera and this image is sent to the Z-bar library after preprocessing. This method improves the accuracy by 20% because Z-bar is more precise while processing jpg images. The papers also state a possibility of using both the techniques simultaneously for higher accuracy and processing speed.

**Warehouse Management Using Real-Time QR-Code and Text Detection**

[5] In this paper the process of QR code recognition is split into 3 stages paper is about developing the computer vision tools for efficient inventory management of packages of a warehouse using QR code.An industrial camera with 120 fps is used to detect the QR code .The Z-bar library and google text recognition tesseract are used for QR code recognition and text detection respectively.

In the first stage, the acquired image is rescaled cropped and the distortions are removed by blurring.The image is cropped to convert the QR code part of the video feed into a jpg file.In the next stage, the image obtained is passed to the z bar library and the QR code is detected and decoded. In the next stage, the image is preprocessed and passed to tesseract for text recognition.In this project this decoded information is used to arrange the warehouse goods in their respective places.This paper provides an overview of the two main methods used for specific object detection that is QR code detection and text detection and explains the necessary algorithms required for carrying out this process.

The paper provides an overview of the necessary tools for image recognition, QR code decoding and text recognition.The paper also briefs about the algorithm of preprocessing the acquired image which include rescaling, distortion removal,cropping .The tools and information required for image recognition and decoding like Z-bar and tesseract which is necessary for our project was obtained from the paper.

**Development of a Human-Tracking Robot Using QR Code Recognition**

[6] In this paper they conducted research on human-tracking robots. They implemented a tracking system which differentiates individuals using QR codes and tracks subjects using shape-based pattern matching. They also proposed a rediscovery process using an IR camera to handle cases when the QR code strays completely out of the camera field of vision.

There are three main requirements for realizing

a human-following robot:

(a) Continuous identification of the individual person being followed.

(b) Detection of the location of the person being followed.

(c) Control of the robot based on the detected location data.

Since case (a) is difficult with image processing alone, QR-tag

is used for tracking.

**Eye in the Sky: Drone-Based Object Tracking and 3D Localization**

[7] In this paper, a multi-object tracking and 3D localization scheme is implemented based on deep learning based on object detection. A multi-object tracking method called TrackletNet Tracker (TNT) which utilizes temporal and appearance information to track detected objects located on the ground for UAV are applied can be used for the project . The following novelties such as Accurate object detection , Multi-object tracking, Visual odometry and ground plane estimation , 3D object localization can be implemented in our project .

**Structure From Motion Technique for Scene Detection Using Autonomous Drone Navigation**

[8]From this paper information about scene detection and estimation using high-resolution imagery acquired through autonomous drone navigation aided with landmark detection and recognition.

The drone can capture images and provide real-time video streaming of the ground cover using a camera.

* *Autonomous Navigation*  - For safe and efficient navigation, we are able to localize them autonomously using their onboard sensors and interpretation of the unknown environment features. It is a vision-based target detection and localization system.
* The regions of interest (ROIs) in each video frame during the flight and trigger autonomous responses based on the detected landmark and the current drone status.

Considering the two-dimensional matrix an Apriltag is an extended form is QR code but have relative ease in finding attributes related to it their robust feature and flexibility [9]

Based on the different layout their detection can be increased and the efficiency of the detection can be further improved as stated by Olson [10[11]].

**Autonomous Landing of a UAV on a Moving Platform Using Model Predictive Control** This paper is about the autonomous landing of an unmanned aerial vehicle (UAV) on a mobile platform using new autonomous landing methods which can be implemented on micro UAVs that require high-bandwidth feedback control loops for safe landing under various uncertainties and wind disturbances. These are effective for rapid deployment and recovery of a fleet of UAVs, mobile recharging stations etc.

The drone has an autonomous landing with an error of less than 37 cm from the centre of a mobile platform travelling at a speed of up to 12 m/s under the condition of noisy measurements and wind disturbances.

* Vision-based target position measurement.
* Kalman filter for optimal target localization.
* Model predictive control for guidance of the UAV.
* Integral control for robustness.

**Autonomous Landing of a Multirotor Micro Air Vehicle on a High Velocity Ground Vehicle**

Drones with application in almost all the fields have been developed in the recent past ranging from transport and surveillance to firefighting and military application.This paper [12] is about adding an ability of landing on a vehicle moving with high velocity.

In the paper an april tag on top of a car is defined as the landing spot and the drone has to land on top of this car moving with relatively high velocity.

The drone uses a Proportional Navigation law driven algorithm for the process of following the car and landing on it.This process is broken down into two main phases,the first phase includes the instance when the car and the drone is at a certain distance. At this point the car has to be tracked and followed using the gps signal from the car.In this phase a pursuit trajectory is used which follows the cars movement according to the cars gps signal .The second phase is initiated the the car and the drone are at relatively close proximity and visual feedback of the car is available (LOS).At this instance the PN controller produces an acceleration signal which takes parameters from kalman filter and produces a gain constant .this gain constant is used to calculate a velocity high enough to overtake the car. During this process the acceleration in the z direction is not considered resulting in a one directional acceleration.The landing is controlled by the PID controller and is initiated once the drone is stabilised over the landing platform.The phases of landing on a moving platform mentioned in this paper can be adapted and modified for smooth landing process.

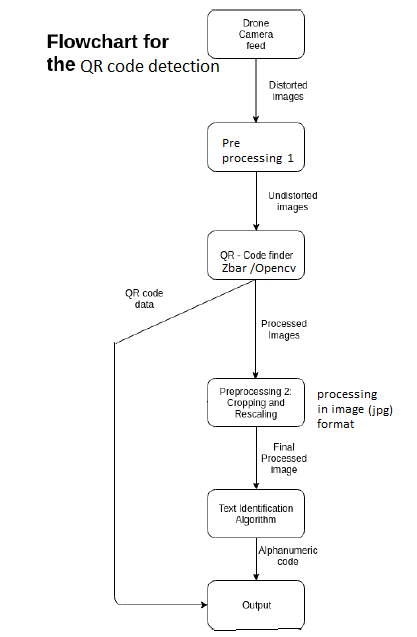
**Content**

**Navigation**

The intention of an autonomous drone is to estimate its current location along with the location of the next position it needs to traverse, calculate the optimum trajectory for path and travel through it. The first stage of the autonomic nature of quadcopter of localisation. This process is intended to be performed in the Navigation part. There are various methods of localisation of the positions of the future state of the drone, based on Simultaneous Localisation and mapping, GPS coordinate Estimation etc. Roland Brockers in his paper suggested a single camera method with Visual SLAM and Extended Kalman along with feature point estimate using RANSAC in order to find an elevated platform for UAV Landing. However, this method requires high-end processors and the method consumes large power for algorithm processing. The process would be halved if we can place a pattern, an object, a shape for the landing zone such that the rather than computing the whole environment for the platform estimation. Shapes such as Ⓗ are used by manned helicopters for the precise landing zone identification. QR codes have been used in moving vehicles as Hogparcha [13] but the fact of placing QR code on a moving platform and imaging device being stationary is not always the case. Also, the QR code has a significant amount of information. Extraction of the QR code with a stationary camera is not always practical. The QR code Detection is explained below **Z-bar and python attributes for QR Code detection**

Drones have a comprehensive application in almost all the sectors like surveillance, transportation and military services. Scrutinising over this aspect the idea of a drone with the ability to detect and read moving QR code from an altitude was broached. A System which detects QR code consists of three stages which include QR code detection, QR code recognition and tracking the QR code.

For the ease of implementation, the QR code detection is done using Opencv and Z-bar Python library files. A camera module in the drone inputs the feed to the Z-bar library. Inside the Z-bar library, the necessary preprocessing required for the detection like resizing distortion correction blurring etc is done. Then the QR code is detected with OpenCV and the image is cropped and resized. Further, this image is used for QR code recognition and then for tracking. In this algorithm process, there are two ways of processing which include QR detection using the direct video feed from the camera as the input and by using the frames from the video feed as image input. Both these methods can be carried out at the same time for increased efficiency.



**Code**

*from \_\_future\_\_ import print\_function*

*import pyZ-bar.pyZ-bar as pyZ-bar*

*import numpy as np*

*import cv2*

*def decode(im) :*

*# Find QR codes*

*decodedObjects = pyZ-bar.decode(im)*

*# Print results*

*for obj in decodedObjects:*

*print('Type : ', obj.type)*

*print('Data : ', obj.data,'\n')*

*return decodedObjects*

*# Display QR code location*

*def display(im, decodedObjects):*

*# Loop over all decoded objects*

*for decodedObject in decodedObjects:*

*points = decodedObject.polygon*

*# If the points do not form a quad, find convex hull*

*if len(points) > 4 :*

*hull = cv2.convexHull(np.array([point for point in points], dtype=np.float32))*

*hull = list(map(tuple, np.squeeze(hull)))*

*else :*

*hull = points;*

*# Number of points in the convex hull*

*n = len(hull)*

*# Draw the convext hull*

*for j in range(0,n):*

*cv2.line(im, hull[j], hull[ (j+1) % n], (255,0,0), 3)*

*# Display results*

*cv2.imshow("Results", im);*

*cv2.waitKey(0);*

*# Main*

*if \_\_name\_\_ == '\_\_main\_\_':*

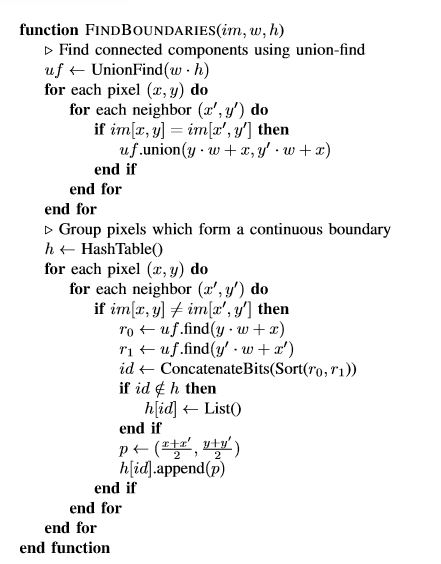
*# Read image*

*im = cv2.imread('Z-bar-test.jpg')*

*decodedObjects = decode(im)*

*display(im, decodedObjects)*

However, the QR code detection can be incorporated but is subject to constraints such as the resolution of the onboard camera. Moreover, the autonomous feature can be implemented only if the localisation of the object is incorporated. Hence switching from Qr code to the new modern matrix barcode would intent reduce our errors of detection. As a result, the April tag invented by April Robotics Laboratory is incorporated in our project. April tag is a visual fiducial system used for a wide variety of tasks such as augmented reality, camera calibration, robotics and much more. Here we are going to use the April Tag as the marker to estimate the location of the landing zone. The AprilTag 2 algorithm in Robot Operating Software(ROS) enables us to detect relative pose between detected tag and camera, highlighting detected tags of the input image frame. Before the above process, we intend to speed up the detection by using an adaptive thresholding technique of CLAHE (Contrast Limited Adaptive Histogram Equalisation) on the Grayscale image inputted from the camera feed.



The above algorithm helps us to achieve boundary segmentation based on the union-find algorithm [14].

**Object Recognition Using MATLAB**

**Code**

%% Load reference image, and compute surf features

ref\_img = imread('reference.jpg');

ref\_img\_gray = rgb2gray(ref\_img);

ref\_pts = detectSURFFeatures(ref\_img\_gray);

[ref\_features, ref\_validPts] = extractFeatures(ref\_img\_gray, ref\_pts);

figure; imshow(ref\_img);

hold on; plot(ref\_pts.selectStrongest(50));

%% Visual 25 SURF features

figure;

subplot(5,5,3); title('First 25 Features');

for i=1:25

scale = ref\_pts(i).Scale;

image = imcrop(ref\_img,[ref\_pts(i).Location-10\*scale 20\*scale 20\*scale]);

subplot(5,5,i);

imshow(image);

hold on;

rectangle('Position',[5\*scale 5\*scale 10\*scale 10\*scale],'Curvature',1,'EdgeColor','g');

end

%% Compare to video frame

image = imread('mixed.jpg');

I = rgb2gray(image);

% Detect features

I\_pts = detectSURFFeatures(I);

[I\_features, I\_validPts] = extractFeatures(I, I\_pts);

figure;imshow(image);

hold on; plot(I\_pts.selectStrongest(50));

%% Compare card image to video frame

index\_pairs = matchFeatures(ref\_features, I\_features);

ref\_matched\_pts = ref\_validPts(index\_pairs(:,1)).Location;

I\_matched\_pts = I\_validPts(index\_pairs(:,2)).Location;

figure, showMatchedFeatures(image, ref\_img, I\_matched\_pts, ref\_matched\_pts, 'montage');

title('Showing all matches');

%% cross compare on top

[tform,inlierIdx] = estimateGeometricTransform2D(I\_matched\_pts,ref\_matched\_pts,'similarity');

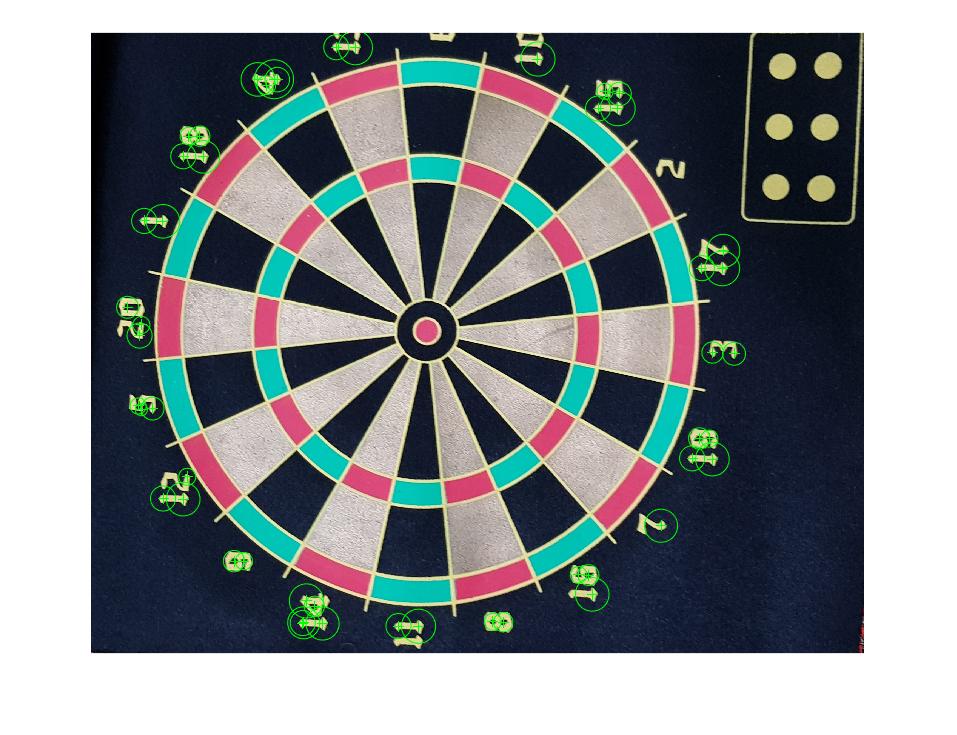
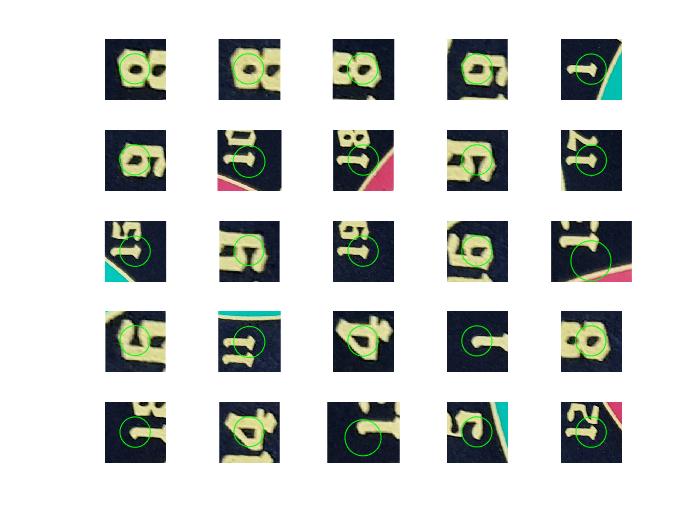
inlierPtsDistorted = I\_matched\_pts(inlierIdx,:);

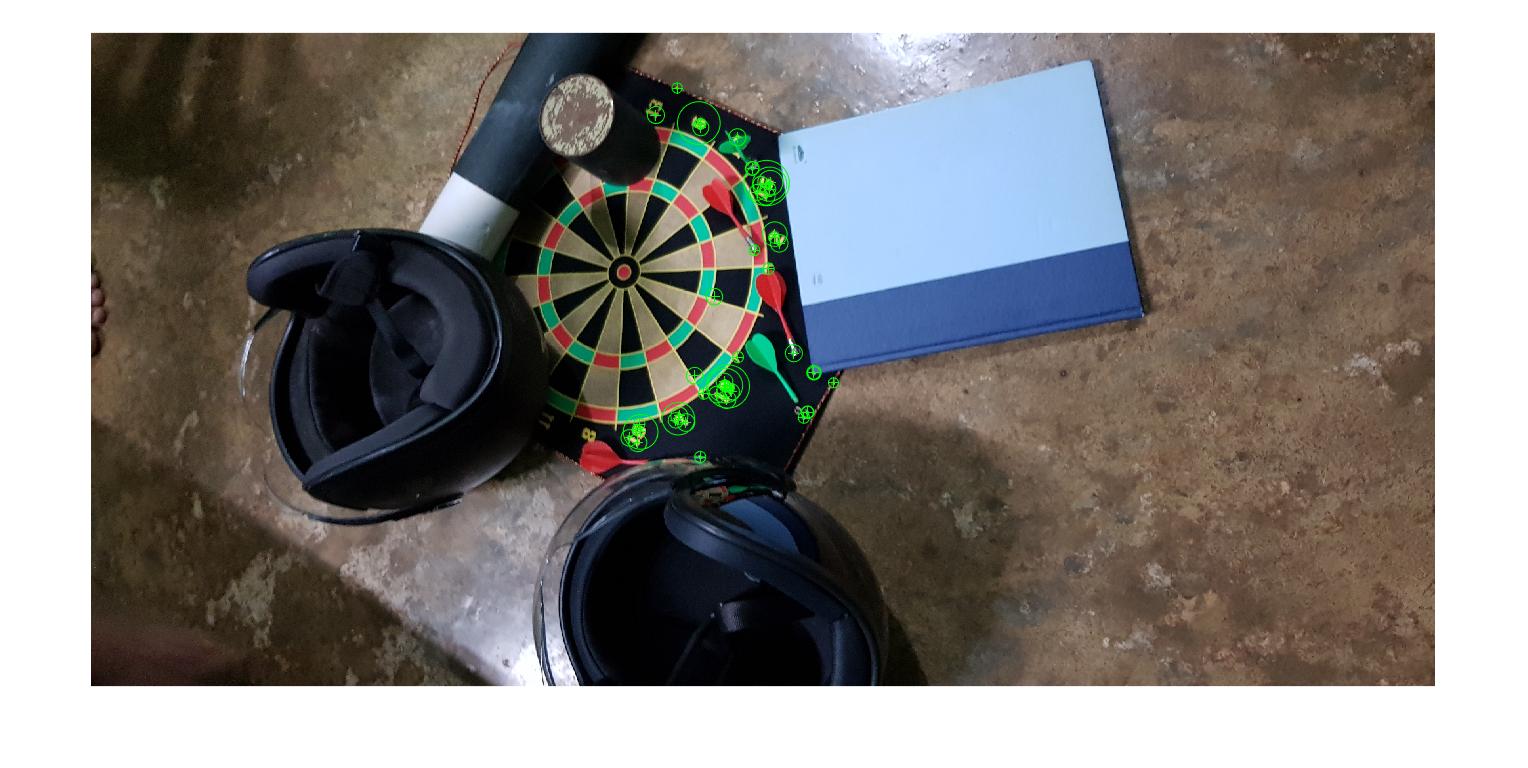
inlierPtsOriginal = ref\_matched\_pts(inlierIdx,:);

figure

showMatchedFeatures(ref\_img,I,inlierPtsOriginal,inlierPtsDistorted)

title('Matched Inlier Points')

Reference Image Visual 25 SURF features 

Mixed Image Matched Inlier Points

Showing all matches

**MATLAB code for Apriltag Detection and pose estimation**

**Code**

%% Detecting Apriltag in Image

I1 = imread("aprilTagsMulti.png");

tagFamily = ["tag36h11","tagCircle21h7","tagCircle49h12","tagCustom48h12","tagStandard41h12"];

[id,loc,detectedFamily] = readAprilTag(I1,tagFamily);

for idx = 1:length(id)

% Display the ID and tag family

disp("Detected Tag ID, Family: " + id(idx) + ", " ...

+ detectedFamily{idx});

% Insert markers to indicate the locations

markerRadius = 8;

numCorners = size(loc,1);

markerPosition = [loc(:,:,idx),repmat(markerRadius,numCorners,1)];

I1 = insertShape(I1,"FilledCircle",markerPosition,"Color","red","Opacity",1);

end

figure;

imshow(I1);

%% Estimating Apriltag Pose

I2 = imread("aprilTag36h11.jpg");

figure;

imshow(I2)

data = load("camIntrinsicsAprilTag.mat");

intrinsics = data.intrinsics;

tagSize = 0.04;

I2 = undistortImage(I2,intrinsics,"OutputView","same");

[id,loc,pose] = readAprilTag(I2,"tag36h11",intrinsics,tagSize);

worldPoints = [0 0 0; tagSize/2 0 0; 0 tagSize/2 0; 0 0 tagSize/2];

for i = 1:length(pose)

% Get image coordinates for axes.

imagePoints = worldToImage(intrinsics,pose(i).Rotation, ...

pose(i).Translation,worldPoints);

% Draw colored axes.

I2 = insertShape(I2,"Line",[imagePoints(1,:) imagePoints(2,:); ...

imagePoints(1,:) imagePoints(3,:); imagePoints(1,:) imagePoints(4,:)], ...

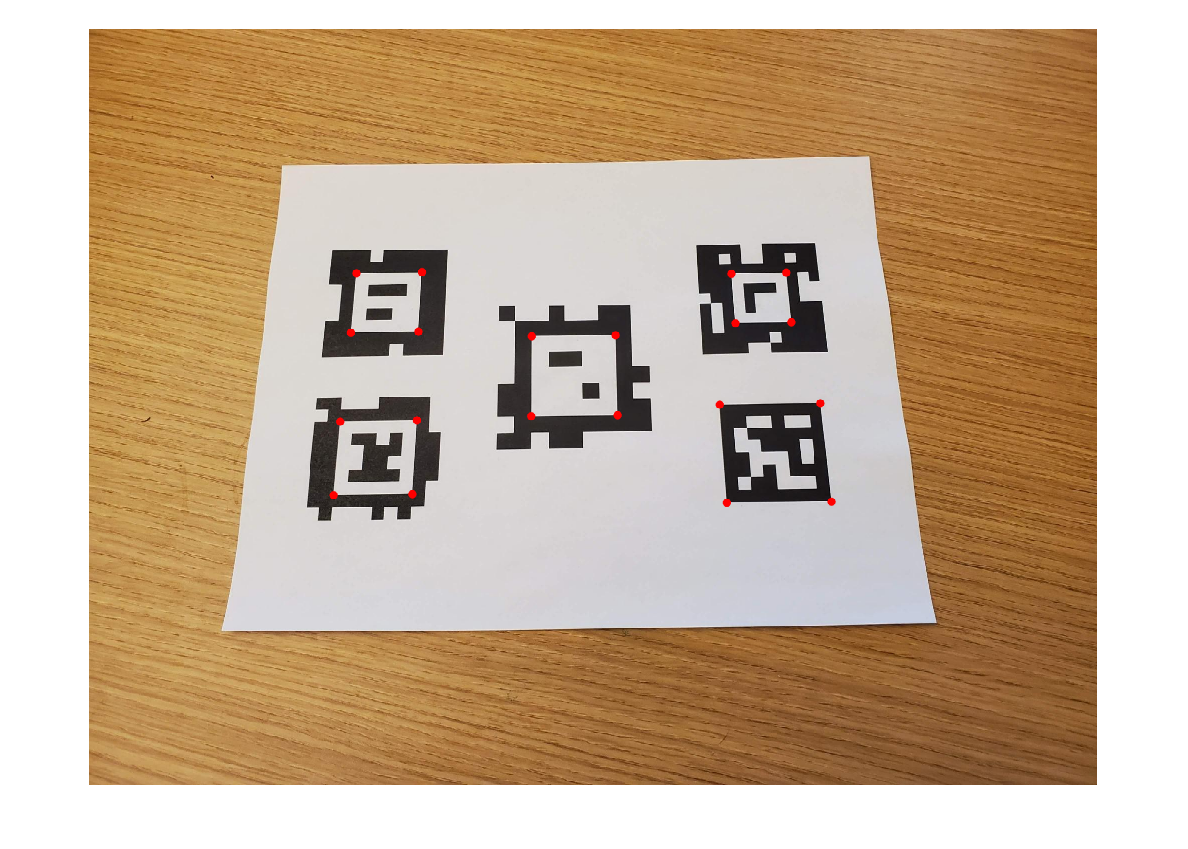
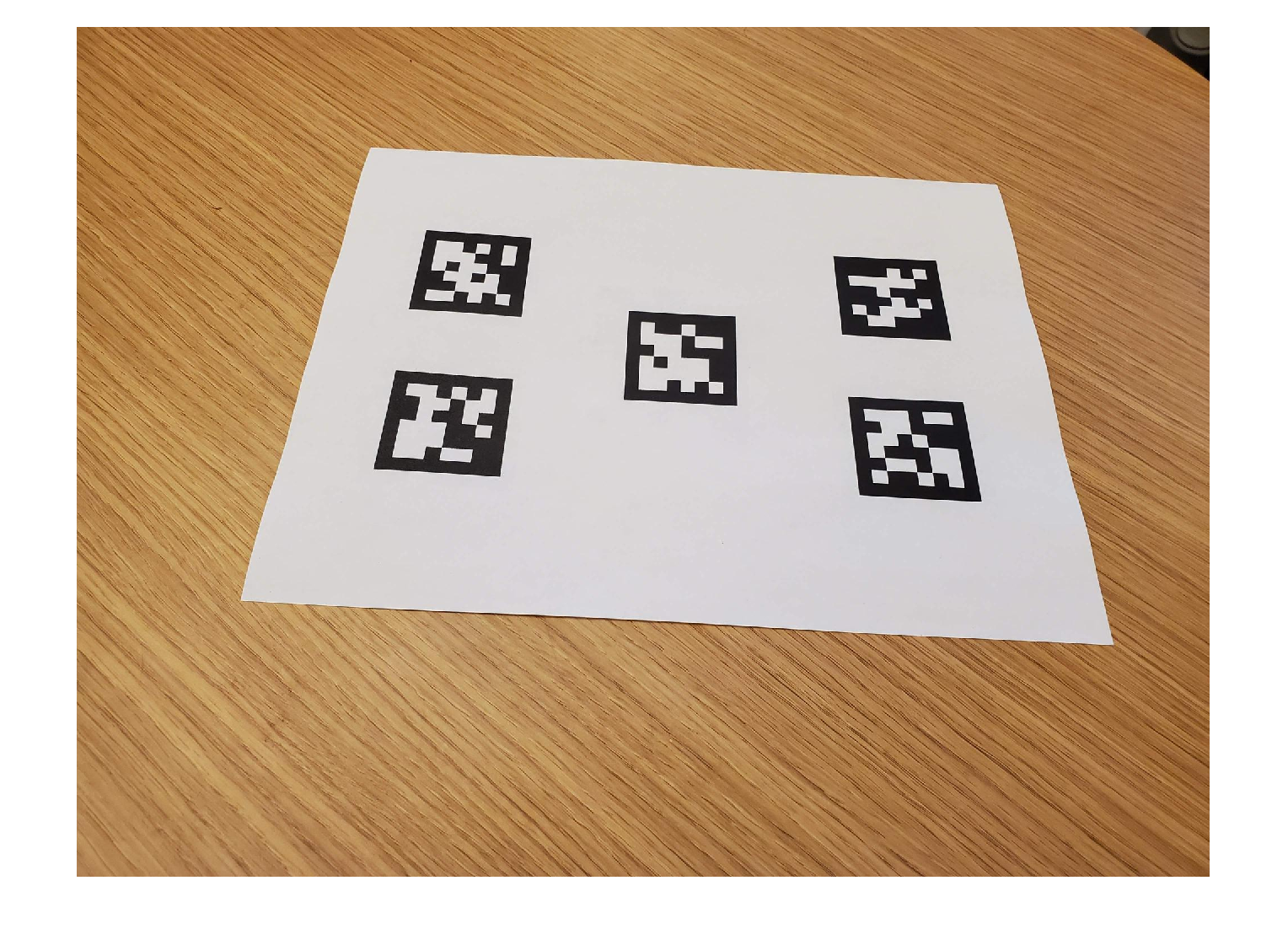
"Color",["red","green","blue"],"LineWidth",7);

I2 = insertText(I2,loc(1,:,i),id(i),"BoxOpacity",1,"FontSize",25);

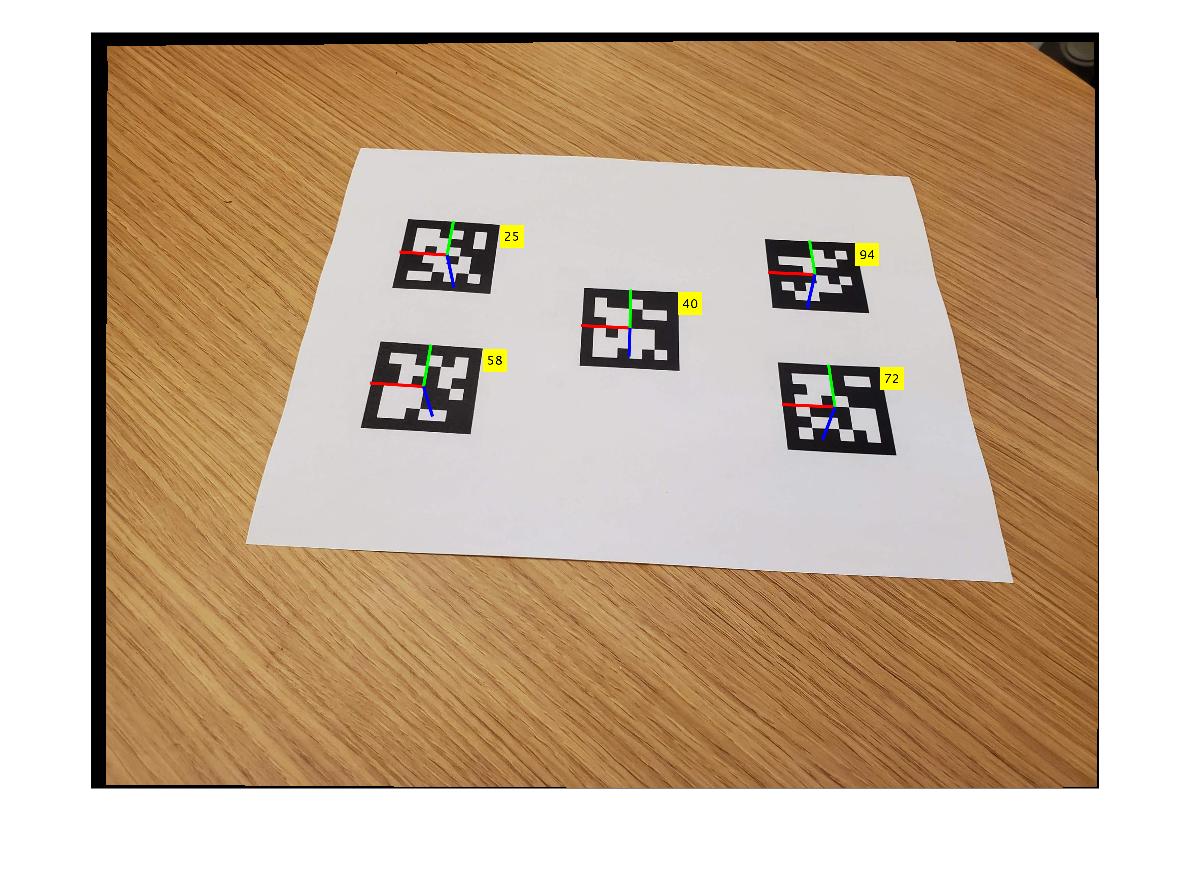
end

figure;

imshow(I2);

** **

April Tag Detection Image for pose estimation



**Guidance(Trajectory generation)**

[15] Quadcopters showcase high agility and large potential in angular movements due to their outward mounted propellers.It is also highlighted for the safety offered by small rotors storing relatively little energy and the ability to hover through cluttered environment .A key feature of the Quadcopter that is required for the use in complex conditions and cluttered environment is the trajectory generation.The trajectory generation is related to the flight path that archives the task objective constrained by the dynamics of the quadcopter.

The Trajectory generation can be classified broadly into two , the first one generates trajectory by decoupling geometric and temporal planning while the second uses the differential flatness of the dynamics of the drone to derive the constraints of the trajectory. The first method can be further broken down to two steps, In the first step, a geometric trajectory without the time information is constructed for example using line and polygons. In the second step, the geometric trajectory is parameterised in time in order to derive feasibility and optimisation with respect to the dynamics of the quadcopter. The second group of algorithms uses the differential flatness of drone dynamics to form the constraints on trajectory and then solve the optimisation problem over the class of the derived trajectories [16].The algorithms for the trajectory generation are commonly driven by factors like end point, time constrain and the quadcopter dynamics parameters.In the above groups of algorithms the space is searched for the possible trajectory for the give parameters without the feasibility condition and in the next stage the received trajectories are searched for the feasible ones.This approach helps in speeding up the computational speed of the trajectories and has an ability to output large number of trajectory in given time constraints ,thus this algorithm is best suited for application where the quadcopter is interacting with a dynamic environment where time is the most inevitable constraint.

In cases of a generation of trajectories for landing on a moving landing pad the use of a **Proportional Navigation law-driven** approach is proposed to be efficient [12].In a phase of the large distance between the quadcopter and the landing pad (where the input to track the landing pad is only GPS signal )following a pursuit trajectory achieved by a PN controller augmented with a closing velocity, controller is preferred. The landing phase is characterised by close proximity between the quadcopter and the landing pad and when visual feedback for the tracking of the landing pad is available.PN guidance law uses the fact that two vehicles are on a straight line if their LOS remains at a constant angle in order to steer the quadcopter towards the landing pad. At this point the PN controller provides an acceleration command which accelerates the quadcopter normally forward from the LOS.The PN controller provides an acceleration gain which helps in specifying a velocity which is high enough to overtake the moving landing pad.At this point the acceleration in one direction is only of concern hence the acceleration along the z axis is disregarded.The desired acceleration specified from the acceleration gain then needs to be converted to attitude control inputs that are more compatible with the quadcopter input format.The phase of landing the drone is triggered once the drone is vertically above the landing pad and when the drone and the landing pad is moving at same speed. When this state is reached a vertical downward velocity signal is sent which descends the drone.When the drone reaches a certain threshold

Height very close to the landing pad the rotors are disarmed. This method of guidance is used for landing on moving platforms like ship deck etc.

**Control**

The control subsystem is responsible for manipulating the motor thrusts in order to execute the guidance commands while maintaining the airborne quadrotor’s stability. The control subsystem is subject to the design specifications such as the robustness of the system, the performance requirement for precision landing, making the design future-proof and aim to minimize the learning curve for new users such that deployment and improvements to the current implementation can be done more easily.

The flight control system consists of two hierarchical layers: the trajectory generator and the tracking controller. The trajectory generator is based on Proportional Navigation law-driven approach [12]. The tracking controller guides the vehicle to follow the given trajectory with minimal error using the output from the trajectory generation layer.

[17] For automatic surveying, the control system should be able to guide the vehicle through the requested waypoints with minimal deviation while keeping the vehicle within its dynamic performance boundary. The flight control should not induce any excessive vibration or rapid turns that may cause motion blur in the camera image.

Since the vision system requires a smooth flight for high-quality image acquisition, the vehicle needs to fly at a constant velocity and a reasonable yaw rate to avoid motion blur of the image. In particular, in order to achieve a smooth

transition around a waypoint with a constant cruise speed, we have to know the next waypoint a priori while the vehicle is approaching the current waypoint so that the flight

control system can prepare for the bank-to-turn manoeuvre without any abrupt changes in heading or cruise velocity. Therefore, the trajectory planner needs to know the next

waypoint as well as the current waypoint so that it can plan ahead around the waypoint. For additional flexibility, the high-level planner may also specify the previous waypoint,

which is not required to have been the actual previous waypoint; it is merely a point that defines a vector of approach to the current waypoint. When a new waypoint request is received from the high-level planner, regardless of the flight mode it requests, the given waypoints are initially connected with linear segments.

The control of the drone using guidance command was simulated using \_\_\_\_\_\_\_\_\_\_\_\_ platform.The below code was executed and the corresponding movement of the drone was simulated.

import quadrotor.command as cmd

from math import sqrt

def plan\_mission(mission):

commands = [

cmd.up(1),

cmd.forward(1),

cmd.turn\_right(90),

cmd.forward(2),

cmd.turn\_left(90),

cmd.forward(4),

cmd.turn\_left(90),

cmd.forward(4),

cmd.turn\_left(90),

cmd.forward(4),

cmd.turn\_left(135),

cmd.forward(sqrt(8))

]

mission.add\_commands(commands)

The command generated by the above python code was converted to a pwm that can supplied as input to the electronic speed control module using the below driver code.

import matplotlib.pyplot as plt

###freq=50Hz time period of 1 pwm=20ms max rpm at ontime=2ms min at ontime=1ms

### stable idle pwm ontime=1.25ms

##motor 1,4 in clockwise and 2,3 in anticlockwise

command=input("input the command :")

command.lower()

motor1=[]

motor2=[]

motor3=[]

motor4=[]

time=[]

for i in range(0,100):

motor1.append(0)

motor2.append(0)

motor3.append(0)

motor4.append(0)

time.append(i)

if command=="up":

print("up")

for i in range(0,100):

if i<10:

motor1[i]=1

motor2[i]=1

motor3[i]=1

motor4[i]=1

elif command=="down":

print("down")

for i in range(0,100):

if i<3:

motor1[i]=1

motor2[i]=1

motor3[i]=1

motor4[i]=1

else:

motor1[i]=0

motor2[i]=0

motor3[i]=0

motor4[i]=0

elif command=="left":

print("left")

for i in range(0,100):

if i<3:

motor1[i]=1

motor2[i]=1

motor3[i]=1

motor4[i]=1

elif i<10:

motor1[i]=1

motor2[i]=1

elif command=="right":

print("right")

for i in range(0,100):

if i<3:

motor1[i]=1

motor2[i]=1

motor3[i]=1

motor4[i]=1

elif i<10:

motor3[i]=1

motor4[i]=1

elif command=="clockwise":

print("yaw clockwise")

for i in range(0,100):

if i<3:

motor1[i]=1

motor2[i]=1

motor3[i]=1

motor4[i]=1

elif i<10:

motor2[i]=1

motor3[i]=1

elif command=="anticlockwise":

print("yaw anti-clockwise")

for i in range(0,100):

if i<3:

motor1[i]=1

motor2[i]=1

motor3[i]=1

motor4[i]=1

elif i<10:

motor1[i]=1

motor4[i]=1

print("motor1:{}".format(motor1))

print("motor2:{}".format(motor2))

print("motor3:{}".format(motor3))

print("motor4:{}".format(motor4))

plt.plot(time, motor1);

plt.axis([0,20, 0,2])

plt.plot(time, motor2);

plt.axis([0,20, 0,2])

plt.plot(time, motor3);

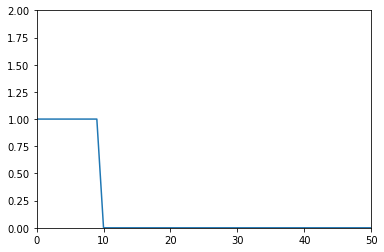
plt.axis([0,20, 0,2])

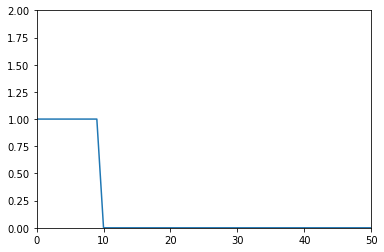
plt.plot(time, motor4);

plt.axis([0,20, 0,2])

The above code was simulated with anticlockwise yaw as input and the below given pwm was obtained.

Motor 1,2,4,3 respectively





Direction of rotation:

Motor 1 & Motor 4 in clockwise direction

Motor 2 & Motor 3 in anti-clockwise direction

In graph:

X-axis=time(T)

Y-axis=Voltage(V)

Thus with the two python scripts simulations a much precise insight was gained regarding the control of the drone using the generated commands and pwm controlled ESC module.

**Conclusion**

In this work, we propose a new control method that allows a micro UAV to land autonomously on a moving platform in the presence of uncertainties and disturbances. Our main focus with this control method lies in the implementation of such an algorithm in a low-cost, lightweight embedded system that can be integrated into micro UAVs.

Implementation of 3 segment study.Literature survey was carried out on the different aspects and segments of implementing the system. The process of landing autonomously on a moving platform was broken down into the three segments namely navigation guidance and control. Each segment was scrutinized separately and the necessary algorithms and codes were corroborated.

**Reference**

[1] Recognition system for QR Code on Moving Car ,The 10th international conference on computer science and education, (ICCSE 2015) cambridge university, UK describes.

[2] Angrisani, L., d’ Alessandro, G., D’Arco, M., Paciello, V., & Pietrosanto, A. (2015). Autonomous recharge of drones through an induction based power transfer system. 2015 IEEE International Workshop on Measurements & Networking (M&N). doi:10.1109/iwmn.2015.7322968 .

[3] Fully Self-Contained Vision-Aided Navigation and Landing of a Micro Air Vehicle Independent from External Sensor Inputs Roland Brockers, Sara Susca, David Zhu, Larry Matthies

Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA

[4] Localization and navigation using QR code for mobile robot in indoor environment Huijuan Zhang, Chengning Zhang, Wei Yang, Chin-Yin Chen, Member, IEEE

IEEE Conference on Robotics and Biomimetics Zhuhai, China, December 6-9, 2015

[5] Warehouse Management Using Real-Time QR-Code and Text Detection

Debjoy Saha Indian Institute of Technology Kharagpur

11th INTERNATIONAL MICRO AIR VEHICLE COMPETITION AND CONFERENCE

[6] Development of a Human-Tracking Robot Using QR Code Recognition Takashi Anezaki\* ,Koki Eimon\* ,Suriyon Tansuriyavong\* ,Yasushi Yagi\*\*

[7] Eye in the Sky: Drone-Based Object Tracking and 3D Localization Haotian Zhang∗ haotiz@uw.edu University of Washington Seattle, Washington Gaoang Wang gaoang@uw.edu University of Washington Seattle, Washington Zhichao Lei zl68@uw.edu University of Washington Seattle, Washington Jenq-Neng Hwang hwang@uw.edu University of Washington Seattle, Washington

[8] Structure From Motion Technique for Scene Detection Using Autonomous Drone Navigation yo-Ping Huang.http/IEEE.com, Senior Member, IEEE, Lucky Sithole, and Tsu-Tian Lee, Fellow, IEEE

[9] AprilTag: A robust and ﬂexible visual ﬁducial system Edwin Olson University of Michigan 2011 IEEE International Conference on Robotics and Automation Shanghai International Conference Center May 9-13, 2011, Shanghai, China.

[10] Flexible Layouts for Fiducial Tags Maximilian Krogius Acshi Haggenmiller Edwin Olson,

[11] AprilTag 2: Efﬁcient and robust ﬁducial detection John Wang and Edwin Olson2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) Daejeon Convention Center October 9-14, 2016, Daejeon, Korea.

[12] Autonomous Landing of a Multirotor Micro Air Vehicle on a High Velocity Ground Vehicle,Mobile Robotics and Autonomous Systems Laboratory, Polytechnique Montreal and GERAD, Montreal, Canada.

[13] UNIVERSITÉ DE MONTRÉAL ,TRAJECTORY GENERATION FOR A QUADROTOR UNMANNED AERIAL VEHICLE,DOUGLAS CONOVER,DÉPARTEMENT DE GÉNIE ÉLECTRIQUEÉCOLE POLYTECHNIQUE DE MONTRÉAL

[14] The 10th International Conference on Computer Science & Education (ICCSE 2015) July 22-24, 2015. Fitzwilliam College, Cambridge University, UK recognition system for QR code on a moving car

[15] 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) Daejeon Convention Center October 9-14, 2016, Daejeon, Korea AprilTag 2: Efﬁcient and robust ﬁducial detection.

[16] A computationally efficient motion primitive for quadrocopter trajectory generation,Mark W. Mueller, Markus Hehn, and Raffaello D’Andrea,SWISS NATIONAL SCIENCE FOUNDATION(SNSF)

[17] Autonomous Vision-based Landing and Terrain Mapping Using an MPC-controlled Unmanned Rotorcraft

Todd Templeton, David Hyunchul Shim, Christopher Geyer, and S. Shankar Sastry