MATRIX BARCODE BASED DETECTION AND TRACKING FOR AUTONOMOUS UAV LANDING

https://docs.google.com/presentation/d/1RnZVztZLN8G16zf2r0c12j8SaH65aVJu2ztDEzGIjlE/edit?usp=sharing

Mentor : Mr Vivek

Asst. Professor,
Electrical and Electronics,
School of Engineering,
Amritapuri

Group:

Velayudhan A	:	AM.EN.U4EEE17101
Nandakishore R Nair	:	AM.EN.U4EEE17123
Neeraj S	:	AM.EN.U4EEE17124
Vishnu S	:	AM.EN.U4EEE17147

Introduction

Dynamic remotely operated navigation equipment (DRONE) is the state of art technology that is most often associated with the military ,where they were used initially for anti-aircraft target practice intelligence gathering and then,more controversially,as weapons platform.Drones are now also used in a wide range of civilian roles ranging from search and and rescue, surveillance , traffic monitoring, weather monitoring and fire fighting , to personal drones for photography , agriculture and even delivery services.

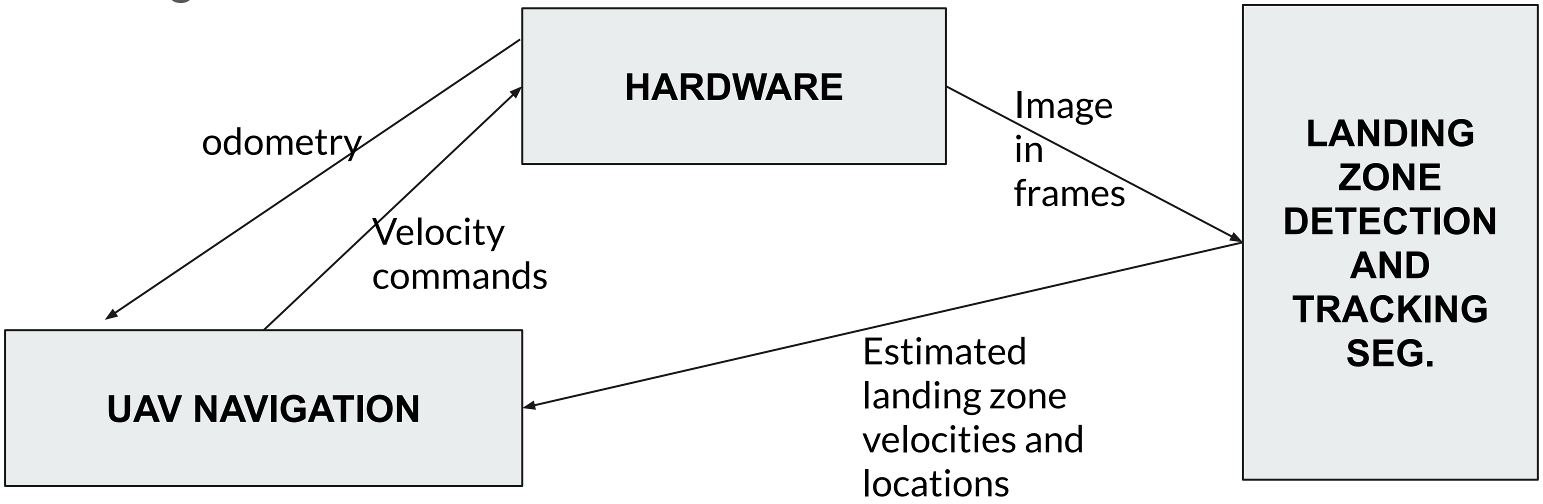
The proposed system is composed of a Matrix barcode Code analyser mounted on board an unmanned aerial vehicle (UAV) along with on board processor, which allows the drone to be versatile enough in order to specifically to be used in object recognition,Human tracking, navigation, moving body identification, godown inspection, etc.

Objectives

1. Apriltag recognition with enhanced precision.
2. Localisation with respect to global frame.
3. Code the drone for tracking of the localised region in frame of interest.
4. Attain all of the above mentioned steps without third party intervention(autonomous nature).

Description of the system

Block diagram



Work done

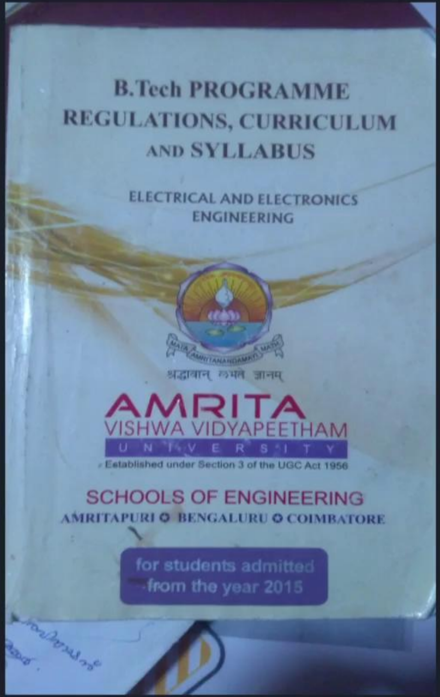
We have been constantly updating the project prerequisites and learning the ways by which this task of autonomous uav landing using fiducial markers can be implemented. As a result we have enrolled ourselves in various courses offered by various online platforms such as edX so that the basics is strong.

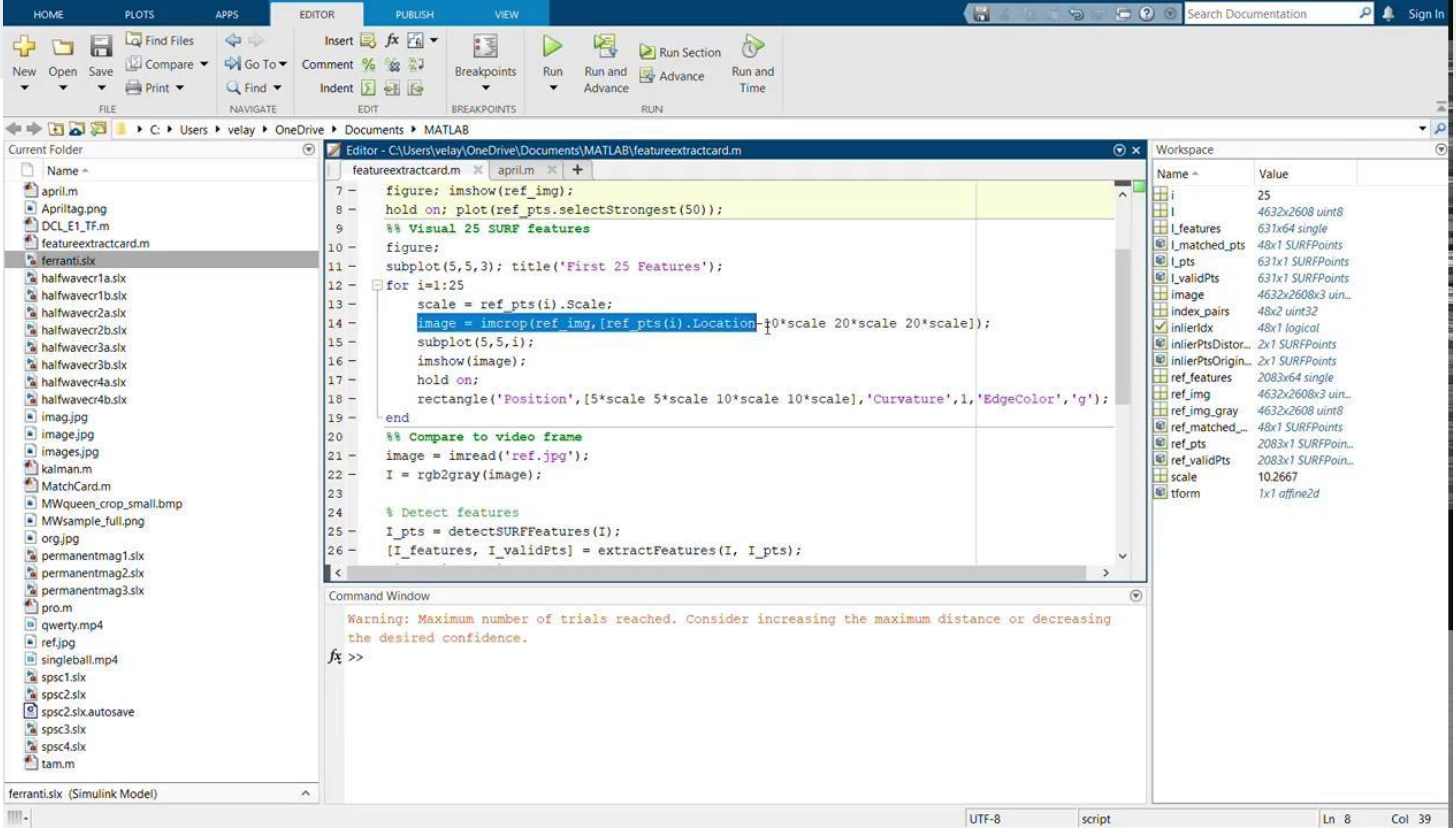
Also as the trend of robotic systems is prevalent in the vast sector, proper simulation and documentation needs to be facilitated as a result the totally new concept of ROS is been focal point. We have made ways to educate ourselves with this meta operating software.

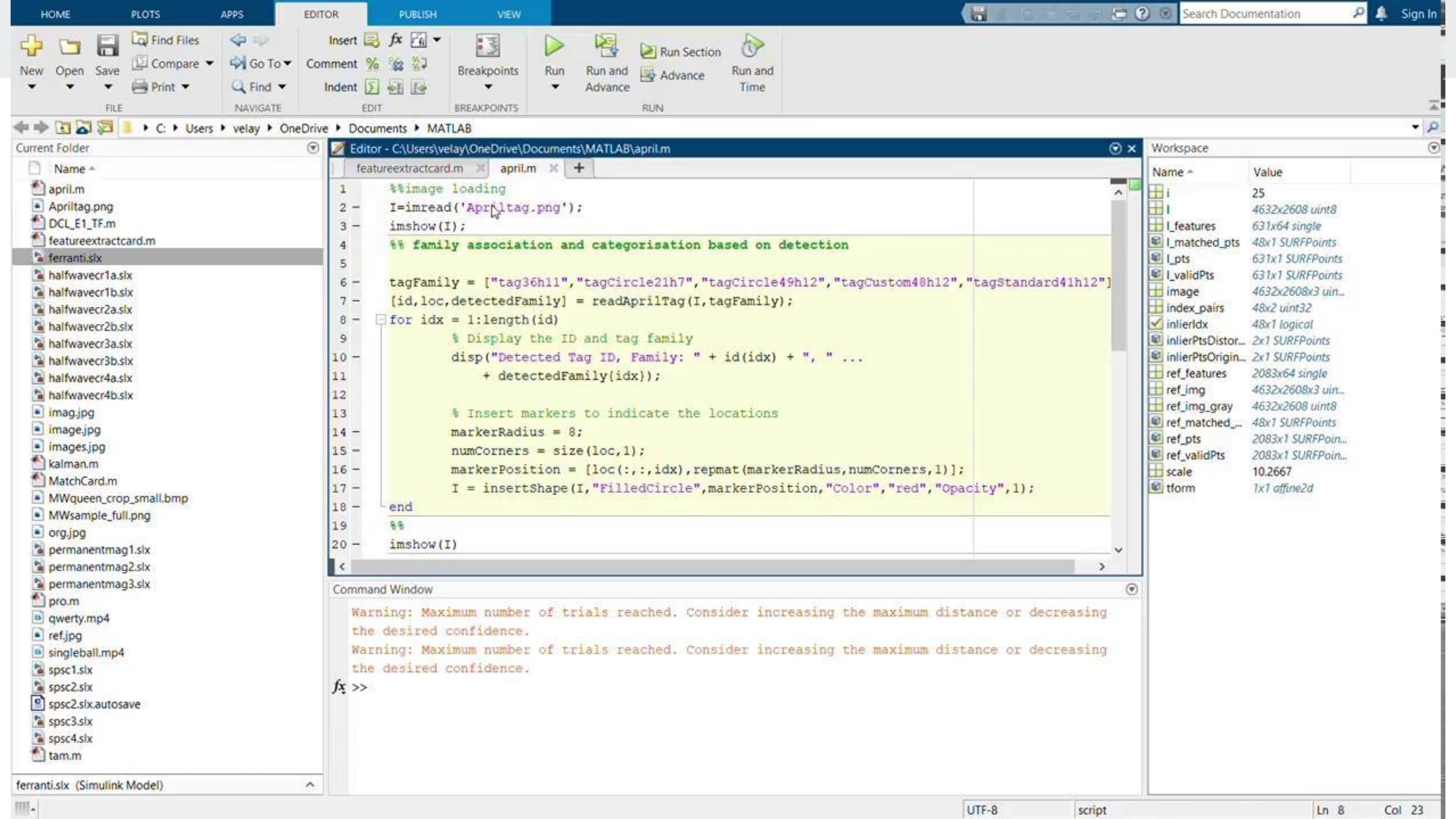
MATlab implementation of the feature extraction is been understood as it is been used in the existing packages prevalent in the UAV techs. Also the codes for the drone fight and general navigation is been exported.

Designing

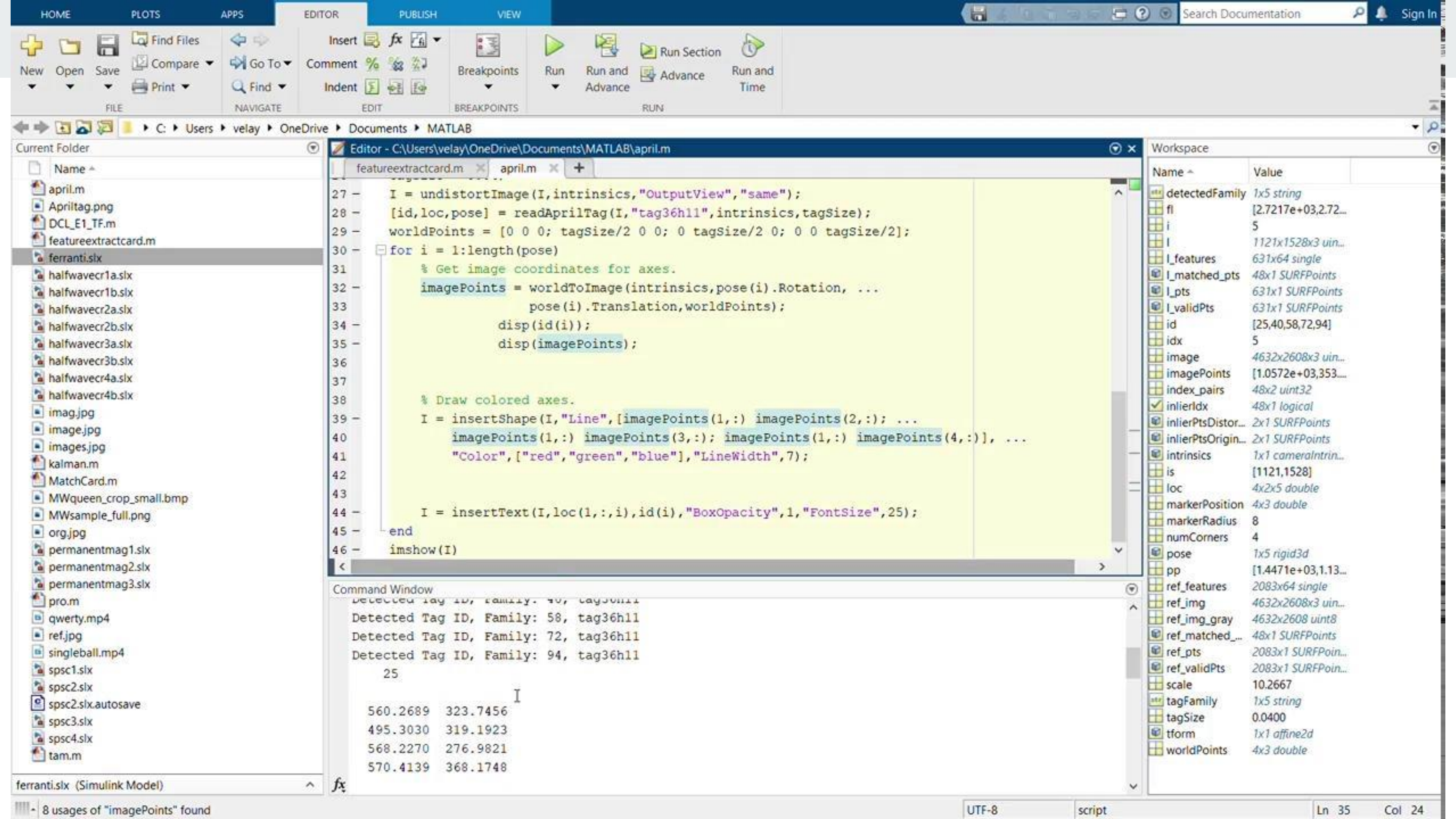
Feature extraction




HOME PLOTS APPS EDITOR PUBLISH VIEW
Search Documentation
Sign In
New Open Save Find Files Compare Print
FILE
Go To Find
NAVIGATE
Insert Comment Indent
EDIT
Breakpoints
BREAKPOINTS
Run Run and Advance Run Section Advance Run and Time
RUN
C: ▸ Users ▸ velay ▸ OneDrive ▸ Documents ▸ MATLAB
Current Folder
Name
april.m
Apriltag.png
DCL_E1_TF.m
featureextractcard.m
ferranti.slx
halfwavecr1a.slx
halfwavecr1b.slx
halfwavecr2a.slx
halfwavecr2b.slx
halfwavecr3a.slx
halfwavecr3b.slx
halfwavecr4a.slx
halfwavecr4b.slx
imag.jpg
image.jpg
images.jpg
kalman.m
MatchCard.m
MWqueen_crop_small.bmp
MWsample_full.png
org.jpg
permanentmag1.slx
permanentmag2.slx
permanentmag3.slx
pro.m
qwerty.mp4
ref.jpg
singleball.mp4
spsc1.slx
spsc2.slx
spsc2.slx.autosave
spsc3.slx
spsc4.slx
tam.m
ferranti.slx (Simulink Model)
Editor - C:\Users\velay\OneDrive\Documents\MATLAB\featureextractcard.m
featureextractcard.m
april.m
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('ref.jpg');
I = rgb2gray(image);
% Detect features
I_pts = detectSURFFeatures(I);
[I_features, I_validPts] = extractFeatures(I, I_pts);
Command Window
Warning: Maximum number of trials reached. Consider increasing the maximum distance or decreasing the desired confidence.
>>
Workspace
Name Value
i 25
I 4632x2608 uint8
I_features 631x64 single
I_matched_pts 48x1 SURFPoints
I_pts 631x1 SURFPoints
I_validPts 631x1 SURFPoints
image 4632x2608x3 uin...
index_pairs 48x2 uint32
inlierIdx 48x1 logical
inlierPtsDistor... 2x1 SURFPoints
inlierPtsOrigin... 2x1 SURFPoints
ref_features 2083x64 single
ref_img 4632x2608x3 uin...
ref_img_gray 4632x2608 uint8
ref_matched_... 48x1 SURFPoints
ref_pts 2083x1 SURFPoin...
ref_validPts 2083x1 SURFPoin...
scale 10.2667
tform 1x1 affine2d
UTF-8 script Ln 8 Col 39



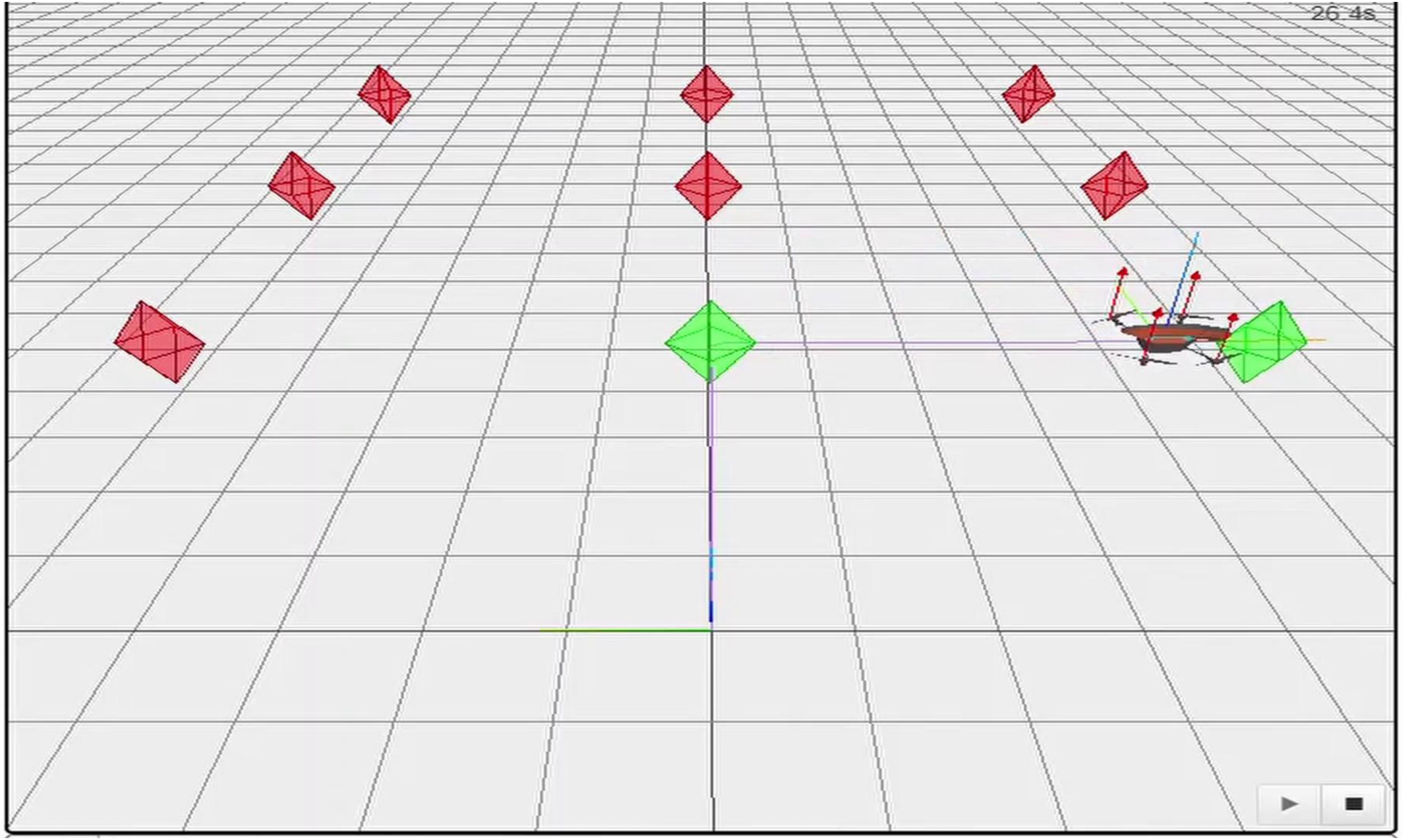
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
Sign In
New
Open
Save
Find Files
Compare
Print
FILE
Go To
Find
NAVIGATE
Insert
Comment
Indent
EDIT
Breakpoints
BREAKPOINTS
Run
Run and Advance
Run Section
Advance
Run and Time
RUN
C: ▸ Users ▸ velay ▸ OneDrive ▸ Documents ▸ MATLAB
Current Folder
Name
april.m
Apriltag.png
DCL_E1_TF.m
featureextractcard.m
ferranti.slx
halfwavecr1a.slx
halfwavecr1b.slx
halfwavecr2a.slx
halfwavecr2b.slx
halfwavecr3a.slx
halfwavecr3b.slx
halfwavecr4a.slx
halfwavecr4b.slx
imag.jpg
image.jpg
images.jpg
kalman.m
MatchCard.m
MWqueen_crop_small.bmp
MWsample_full.png
org.jpg
permanentmag1.slx
permanentmag2.slx
permanentmag3.slx
pro.m
qwerty.mp4
ref.jpg
singleball.mp4
spsc1.slx
spsc2.slx
spsc2.slx.autosave
spsc3.slx
spsc4.slx
tam.m
ferranti.slx (Simulink Model)
Editor - C:\Users\velay\OneDrive\Documents\MATLAB\april.m
featureextractcard.m
april.m
%%image loading
I=imread('Apriltag.png');
imshow(I);
%% family association and categorisation based on detection
tagFamily = ["tag36h11","tagCircle21h7","tagCircle49h12","tagCustom48h12","tagStandard41h12"]
[id,loc,detectedFamily] = readAprilTag(I,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)];
I = insertShape(I,"FilledCircle",markerPosition,"Color","red","Opacity",1);
end
%%
imshow(I)
Command Window
Warning: Maximum number of trials reached. Consider increasing the maximum distance or decreasing the desired confidence.
Warning: Maximum number of trials reached. Consider increasing the maximum distance or decreasing the desired confidence.
fx >>
Workspace
Name
Value
i
25
I
4632x2608 uint8
I_features
631x64 single
I_matched_pts
48x1 SURFPoints
I_pts
631x1 SURFPoints
I_validPts
631x1 SURFPoints
image
4632x2608x3 uin...
index_pairs
48x2 uint32
inlierIdx
48x1 logical
inlierPtsDistor...
2x1 SURFPoints
inlierPtsOrigin...
2x1 SURFPoints
ref_features
2083x64 single
ref_img
4632x2608x3 uin...
ref_img_gray
4632x2608 uint8
ref_matched_...
48x1 SURFPoints
ref_pts
2083x1 SURFPoin...
ref_validPts
2083x1 SURFPoin...
scale
10.2667
tform
1x1 affine2d
UTF-8
script
Ln 8
Col 23


Editor - C:\Users\velay\OneDrive\Documents\MATLAB\april.m
featureextractcard.m
april.m
I = undistortImage(I,intrinsics,"OutputView","same");
[id,loc,pose] = readAprilTag(I,"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);
disp(id(i));
disp(imagePoints);
% Draw colored axes.
I = insertShape(I,"Line",[imagePoints(1,:) imagePoints(2,:); ...
imagePoints(1,:) imagePoints(3,:); imagePoints(1,:) imagePoints(4,:)], ...
"Color",["red","green","blue"],"LineWidth",7);
I = insertText(I,loc(1,:,i),id(i),"BoxOpacity",1,"FontSize",25);
end
imshow(I)
Command Window
Detected Tag ID, Family: 58, tag36h11
Detected Tag ID, Family: 72, tag36h11
Detected Tag ID, Family: 94, tag36h11
25
560.2689 323.7456
495.3030 319.1923
568.2270 276.9821
570.4139 368.1748
8 usages of "imagePoints" found
UTF-8
script
Ln 35
Col 24

Diamond collection code

```
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)
```

Simulation of the previous commands



PWM generation code

```
import matplotlib.pyplot as plt

###freq=50Hz timeperiod 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):
```

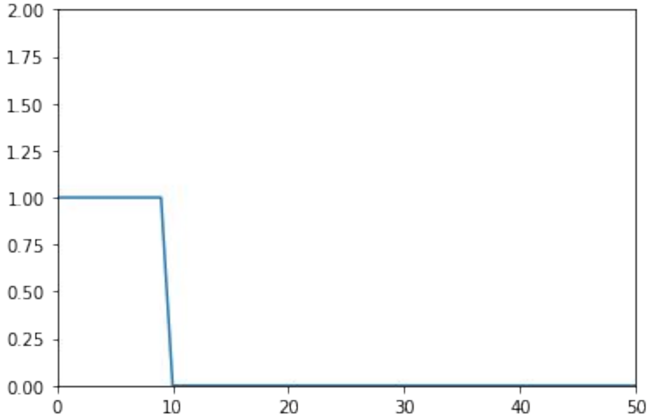
```
        if i<3:
            motor1[i]=1
            motor2[i]=1
            motor3[i]=1
            motor4[i]=1
        elif i<10:
            motor1[i]=1
            motor2[i]=1


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")
```

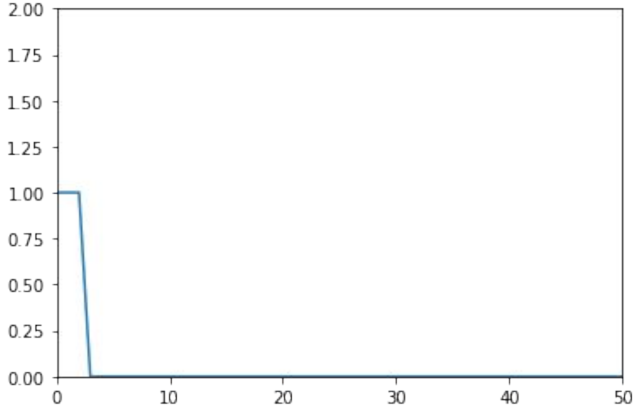
```
for i in range(0,100):
        if i<3:
            motor1[i]=1
            motor2[i]=1
            motor3[i]=1
            motor4[i]=1
        elif i<10:
            motor1[i]=1
            motor4[i]=1
print("motor1:{}".format(motor1))
print("motor2:{}".format(motor2))
print("motor3:{}".format(motor3))
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])
```

PWM Generated with code for anticlockwise yaw

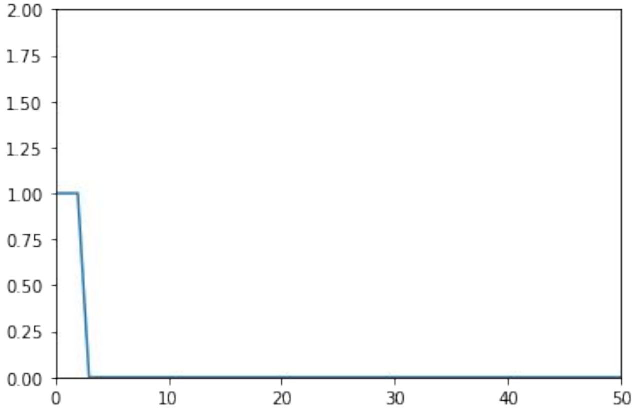
Motor 1



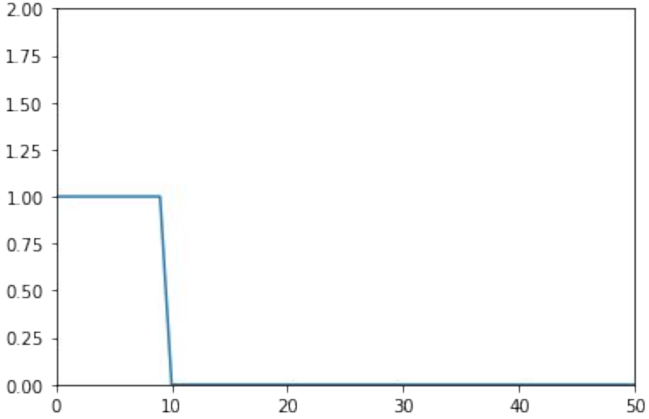
Motor 2



Motor 3



Motor 4



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)

Time plan

December 2020	Controlling the motor output with respect to the given movement commands.
January 2021	Localisation with respect to global frame.
February 2021	Code the drone for tracking of the localised region in frame of interest.
March 2021	Finding the required landing pad by scanning the area.
April 2021	Attain all of the above mentioned steps without third party intervention(autonomous nature).
May 2021	Completing the simulation clearing all the faults in the above stages.

Conclusion

We intend to create a robust modular approach for tackling the autonomous UAV landing procedure by which it would be great advantages in various fields such as UAV landing in ships, UAV landing in remote charging pods for long distance coverage course and much more.

Reference

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

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

[3]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

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

[5]Warehouse Management Using Real-Time QR-Code and Text Detection,Debjoy Saha,IIT Kharagpur
https://www.researchgate.net/profile/Debjoy_Saha/publication/336218818_Warehouse_Management_Using_Real-Time_QR-Code_and_Text_Detection/links/5d9795ac299bf1c363f8d4d6/Warehouse-Management-Using-Real-Time-QR-Code-and-Text-Detection.pdf