Autonomous Drone Landing on a Mobile Platform Using Parrot Mambo and Line Follower Robot

Keywords:

Parrot Minidrone

Autonomous Landing

MATLAB Simulink

Image Thresholding

Moving Platform Tracking

Vision-Based Control

Real-Time Navigation

ABSTRACT

This project demonstrates an autonomous system developed to land a Parrot Minidrone on a moving platform using MATLAB and Simulink. The platform, an 11” x 11” red square mounted on a line-following ground robot, moves along a linear path. The drone is programmed to detect and track this platform in real-time through image processing, follow its motion, and execute a landing sequence even while the platform is in motion. The system utilizes image thresholding, spatial segmentation, and real-time path control logic to achieve reliable landing behavior. The results validate the system’s ability to detect, follow, and land on the platform with minimal positional error.

1. INTRODUCTION

In modern robotics and drone applications, landing on moving platforms is an essential capability, especially in scenarios such as package delivery, search and rescue, and autonomous docking. This project focuses on enabling a Parrot Minidrone to autonomously land on a moving platform using only onboard vision processing and control logic built in MATLAB Simulink.

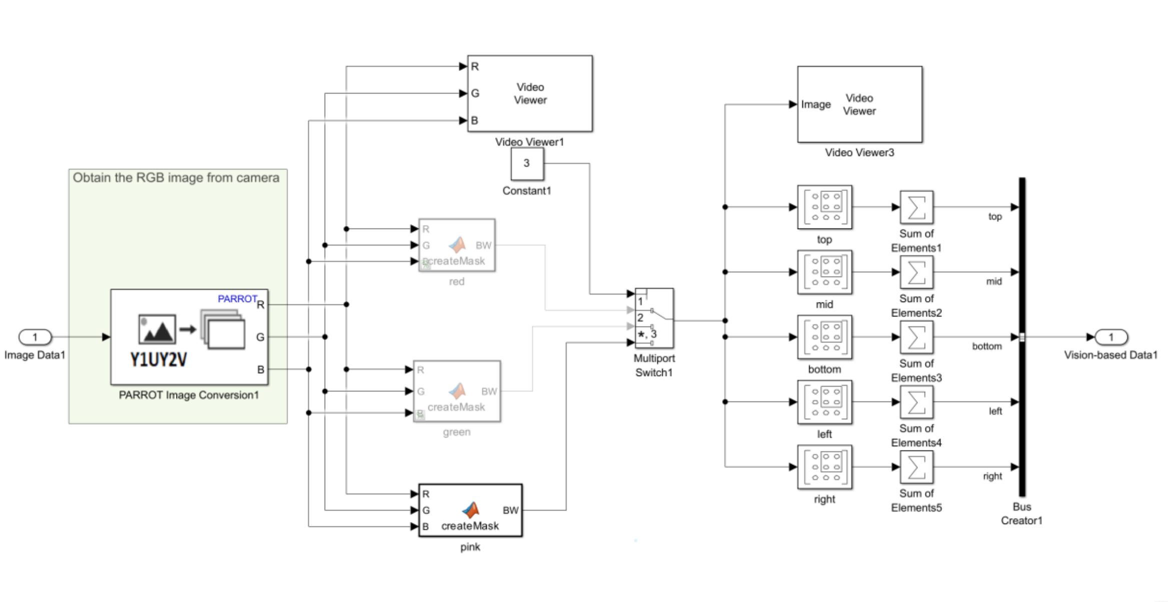
The objective is to demonstrate the real-time integration of computer vision, trajectory control, and state-based decision logic to ensure smooth tracking and successful landing on a red-colored platform that is continuously moving on a predefined path.

2. Methodology

1) System Overview

a) **Drone**: Parrot Minidrone (Simulink supported)
b) **Platform**: 11” x 11” red square on a line-following mobile robot moving in a straight path
c) **Software**: MATLAB Simulink with Simulink Support Package for Parrot Minidrone

2) Image Processing Pipeline



a) **Thresholding**:

i) The downward-facing camera captures RGB images.

ii) A MATLAB Function Block processes the frame using color thresholding to isolate the red platform.

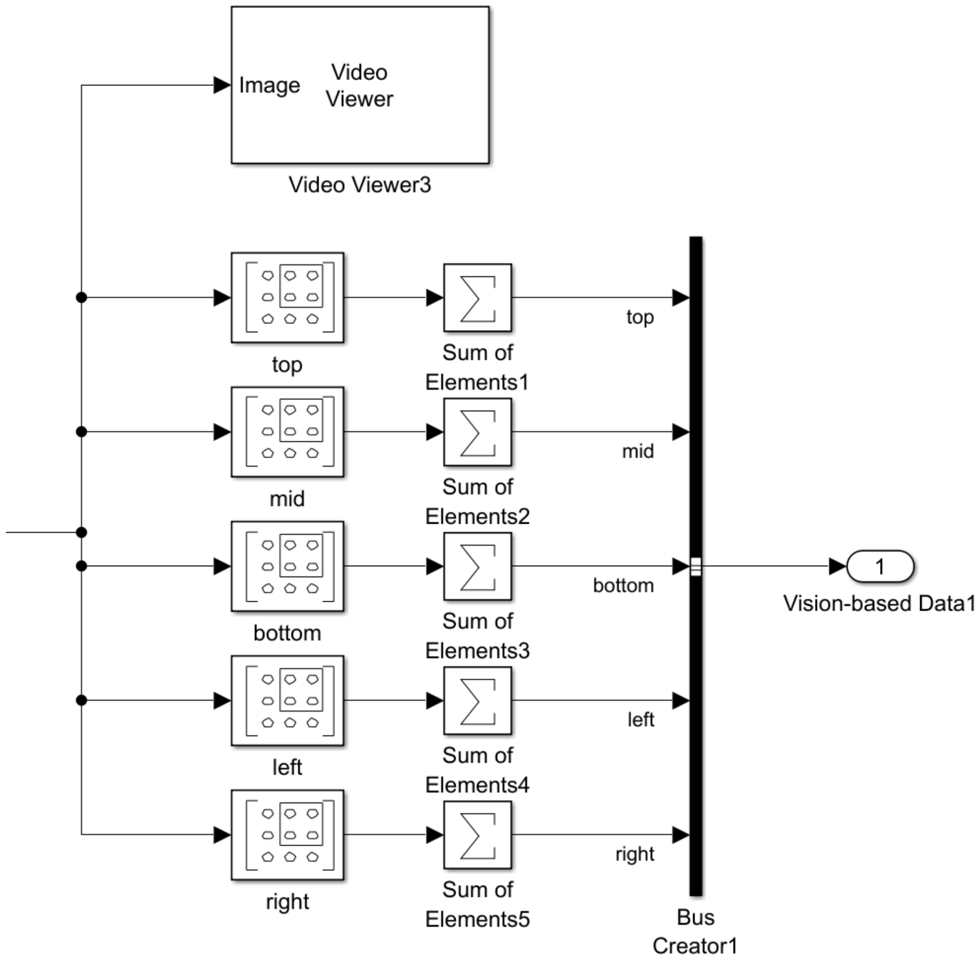
iii) The result is a binary image highlighting only the red platform.

```
function BW = createMask(R,G,B)
I=cat(3,R,G,B);

% Define thresholds for channel 1 based on histogram settings
channel1Min = 225.000;
channel1Max = 255.000;

% Define thresholds for channel 2 based on histogram settings
channel2Min = 90.000;
channel2Max = 190.000;

% Define thresholds for channel 3 based on histogram settings
channel3Min = 125.000;
channel3Max = 247.000;

% Create mask based on chosen histogram thresholds
sliderBW = (I(:,:,1) >= channel1Min ) & (I(:,:,1) <= channel1Max) & ...
    (I(:,:,2) >= channel2Min ) & (I(:,:,2) <= channel2Max) & ...
    (I(:,:,3) >= channel3Min ) & (I(:,:,3) <= channel3Max);
BW = sliderBW;
end
```

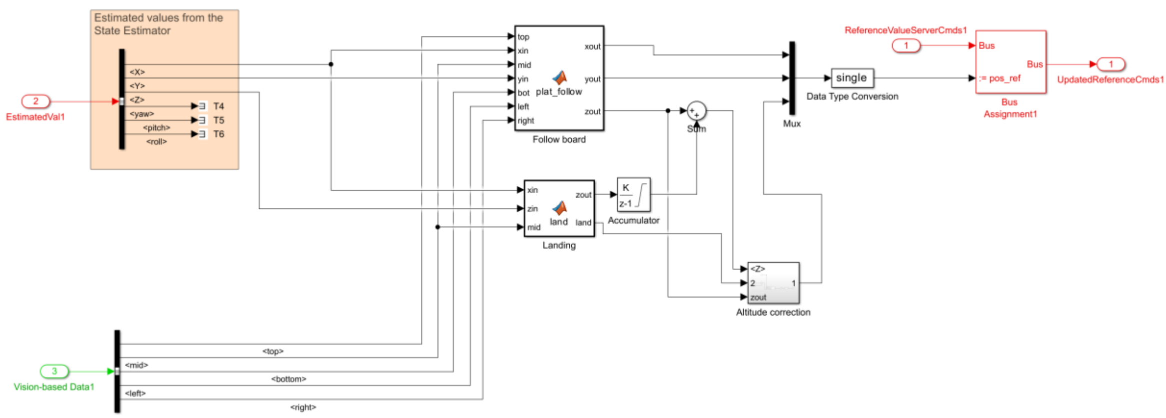
b) **Segmentation**:

i) The binary image is divided into five regions (Top, Bottom, Left, Right, Mid) using five Submatrix blocks.

ii) The sum of white pixels in each region is calculated, providing a signal for platform position in the field of view.



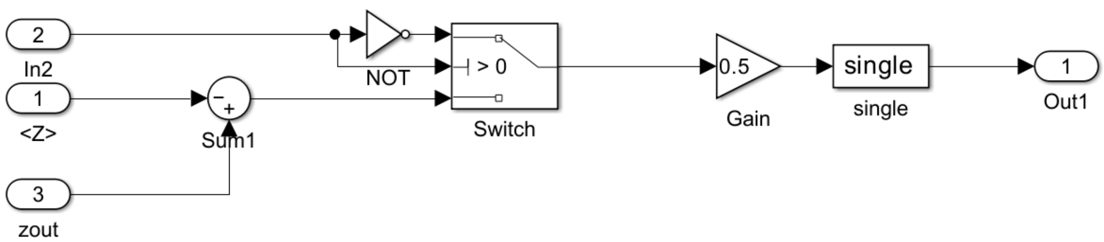
3) Path Control Logic



a) The pixel sums from subregions provide cues for position estimation:
   i) X-direction control = Left – Right/10000
   ii) Y-direction control = Top – Bottom/5000
b) These values are treated as gain signals to adjust the drone's position in both axes.
c) The adjustment is made to the estimated X, Y values and sent to the pos_ref bus for X, Y control.

```
function [xout, yout, zout] = plat_follow(top,xin,mid, yin,bot,left,right)
ygain = 0;

if mid<100
    xgain = 0.1;
else
    xgain = (top-bot)/5000;
    ygain = (right-left)/10000;
end


xout = single(xin + xgain);
yout = single(yin + ygain);
zout = single(-0.9);
```

d) An Altitude control subfunction is placed which is feedback block, it controls the altitude for overshoots.



4) Landing Decision Logic

a) The drone initiates landing if:
   i) The X-position of the platform (estimated from visual input or telemetry) is **> 1 meter**
   ii) The sum of the **Mid** region is **> 500** (i.e., the platform is directly under the drone)
b) This landing block does not interfere with the plaltform tracking block so the tracking continuous while the drone vertically descents until it reaches a defined altitude threshold or touch-down detection.

```
function [zout,land] = land(xin,zin,mid)
    if zin<=0.2 && mid>500 && xin>0.7
        zout = single(10);
        land = single(1);
    else
        if xin>0.7 && mid>500
            zout = single(0.1);
            land = single(1);
        else
            zout = single(0);
            land = single(0);
        end
    end
end
```

3. Results

1) **Tracking Accuracy**: The drone successfully followed the platform under consistent lighting conditions.
2) **Landing Reliability**: The drone was able to land on the moving platform in over the moving platform when the landing criteria were met.
3) **Response Time**: The control system exhibited low latency in following the platform, ensuring smooth trajectory adjustments.
4) **Limitations**:
    a) Slight tracking instability under varying light conditions
    b) Thresholding performance depends heavily on uniform lighting
    c) Platform must remain within camera's FOV