Project – Part 3

Meenakshisundram Ganapathi Subramanian
[1]Arizona State University, Tempe, Arizona

Article Info

Article history:

May 01, 2025

ABSTRACT

The objective of this project is to enable the Parrot Mambo drone to execute a controlled landing on a moving platform mounted on a line-following robot. This task involves synchronizing the drone's vertical descent with the real-time motion of the ground vehicle. The drone is programmed to continuously track the RGB-colored landing platform and initiate a controlled descent only when the platform is directly beneath it. To ensure successful landing either before or exactly at the end of the path, the system integrates visual detection with trajectory prediction and timing control.

1. INTRODUCTION

This lab focuses on developing a Simulink-based control system for the Parrot Mambo Mini drone to perform a dynamic landing on a moving platform. The platform is mounted on a line-following ground robot and features an RGB-colored landing pad that serves as a visual cue for the drone. The primary goal is to enable the drone to track the motion of the line follower, hover above it in real time, and execute a controlled descent for landing while the platform is in motion. This task simulates a complex multi-agent coordination scenario and highlights the integration of computer vision, real-time control, and autonomous landing strategies using Simulink and onboard drone sensors.

2. METHOD

The Parrot Mambo drone is interfaced with the host computer via Bluetooth. A successful connection is verified by pinging the default IP address 192.168.3.1. Upon establishing the connection, MATLAB is launched, and the command parrotMinidroneKeyboardControl is used to initiate communication and bring up the Simulink development environment. Within Simulink, the preconfigured Flight Control System model is accessed. The focus of this project is on modifying the Path Planning subsystem, which already includes modules for vision-based detection and a Landing Enable block. These subsystems serve as the foundation for integrating our custom logic.

To implement the landing behavior, a new Stateflow chart is introduced into the model. This chart is responsible for managing the drone's operational states, including tracking, hovering, and initiating descent. The Stateflow logic uses inputs from the vision-based module to detect the presence of the RGB-colored landing pad and triggers a controlled descent only when the platform is correctly aligned beneath the drone. Additionally, the logic ensures that the landing sequence aligns with the motion of the line-following robot, enabling the drone to land while the platform is still moving.

2.1 Position Control Logic Override

To enable controlled landing on a moving platform, the default position control logic is overridden using a custom Stateflow chart. The inputs to this chart include xestimate, yestimate, yawestimate, x_command, y_command, and targetdetectflag. These signals represent the drone's estimated position and heading, its target commands, and the visual detection status of the landing platform.
The Stateflow chart processes these inputs to determine the appropriate drone motion. Its outputs include xout, yout, zout, yawout, visionland, and controlflag.

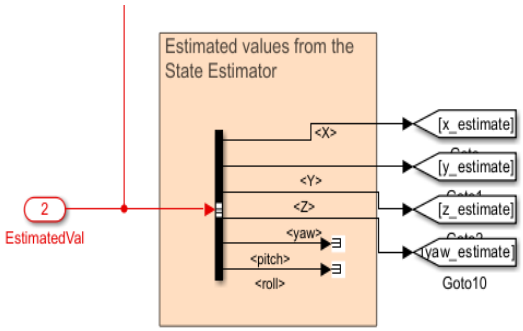

Fig 1 State estimator

2.2 Image Processing System

The image processing system is responsible for detecting the RGB-colored landing pad using the drone's onboard camera. This subsystem integrates image acquisition, color-based masking, and coordinate estimation to guide the drone's positioning. The process begins with capturing live images from the drone's camera. These images are processed using the custom MATLAB function createMask, which isolates the green-colored region corresponding to the landing pad. The function converts the detected green pixels into a binary (black and white) mask, where the target area appears white. This binary image enables easier calculation of the pad's location.

To validate the masking process, a *Video Viewer* block is introduced, allowing real-time display of both the original RGB image and the resulting binary (BW) mask for visual cross-verification. Once the green region is extracted, the system computes the number and distribution of white pixels to determine the centroid of the landing pad. These coordinates are then forwarded to the *Path Planning* subsystem as target inputs, enabling the drone to align itself with the moving platform.

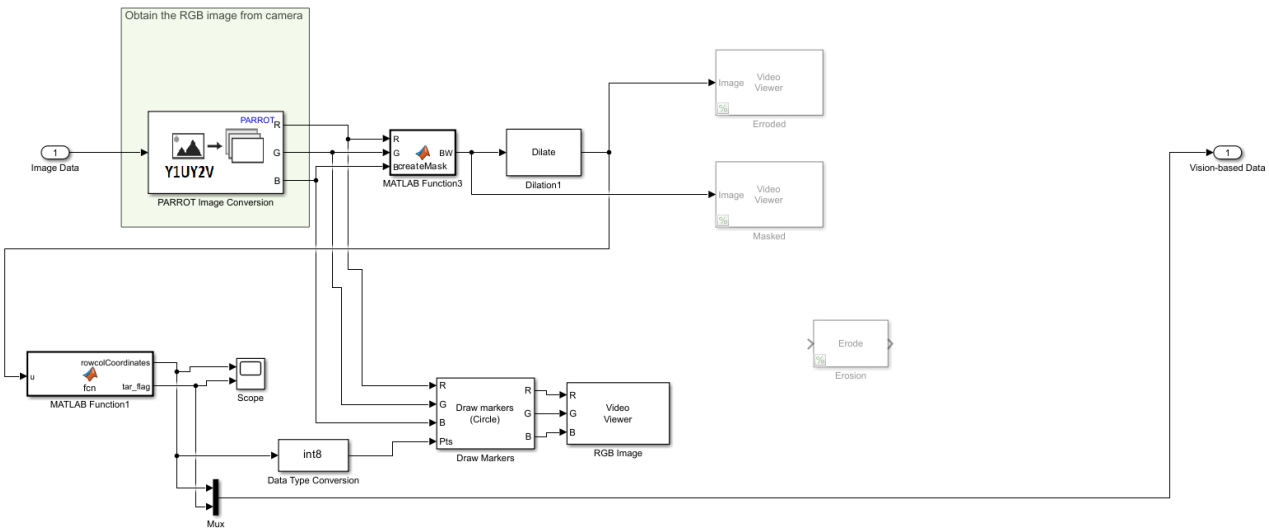

Fig 2 Image Processing System

2.3 Landing Enable

The landing enable logic uses vision-based input to trigger the Parrot Mambo drone's landing sequence. A Boolean signal from the image processing system, indicating detection of the colored landing pad, is added as an input to the OR block in the landing submodel. This modification allows landing to be triggered either manually or through visual detection.

During the flight, the drone is controlled via keyboard commands. Once the landing pad is detected, the system sets the landing flag, prompting the drone to begin a controlled descent onto the moving platform.

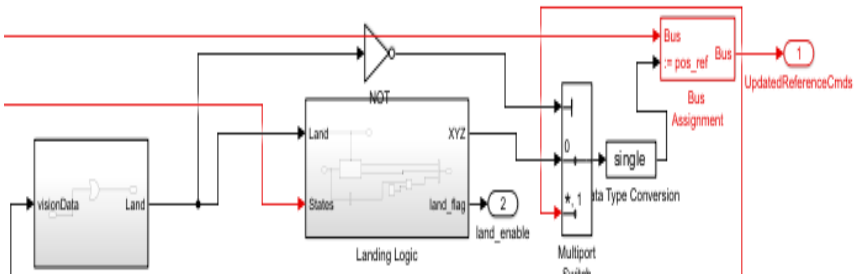

Fig 3 Landing Enable Logic

2.4 State Flowchart for Drone movement

The Stateflow chart defines the sequential control logic that guides the Parrot Mambo drone's autonomous behavior. It transitions through distinct states to achieve takeoff, forward movement, and landing upon detecting a visual marker. The *Takeoff* state commands the drone to ascend and hover at a fixed height of 1.1 meters. Once stable, the system transitions to the *MoveForward* state, where the drone moves ahead while continuously monitoring for the landing pad. Upon detecting the circular marker, the system shifts to the *Land* state, triggering the landing flag to initiate the drone's descent.

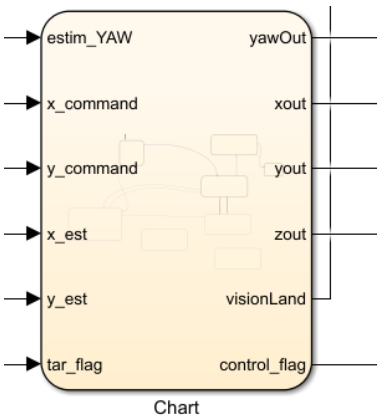

Fig 4 State Flow chart overlay

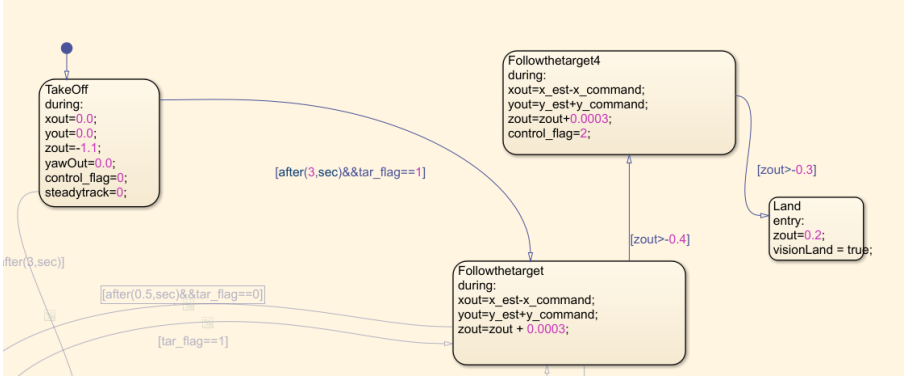

Fig 5 State Flow chart Logic

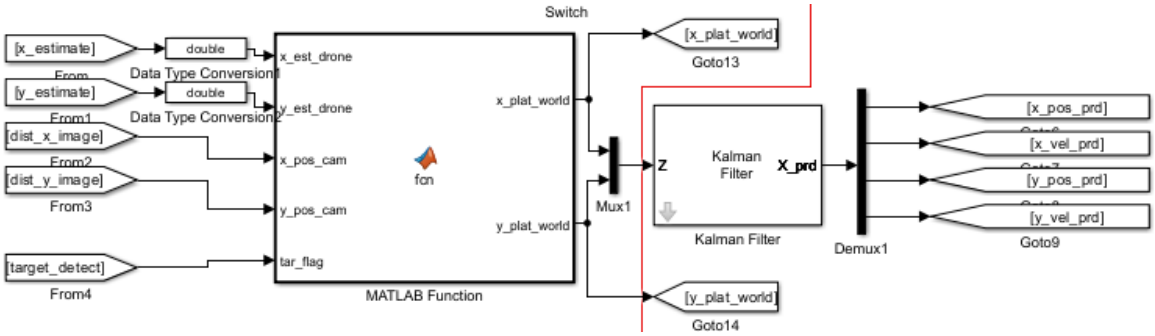

Fig 6 X,Y Position prediction to follow the line following robot

3. RESULTS

Upon loading the program onto the Parrot Mambo drone and initiating the flight control system, the drone successfully tracks and follows the moving landing pad mounted on the line-following robot. When the RGB-colored marker is detected, the drone initiates a slow and controlled descent. Once it reaches a predefined altitude, the landing flag is triggered, allowing the drone to complete its landing sequence accurately on the moving platform.