DRONE LANDING SYSTEM USING COMPUTER VISION

Gromyko Andrew, Golbraikh Mark, Shakhardin Kirill (Supervisor), Zolkin Denis (Supervisor)

Introduction

Our research focuses on developing a high-precision drone landing system that achieves an accuracy of 10cm without relying on GPS. Our algorithm addresses the challenges of GPS-denied environments and significantly enhances the reliability of autonomous drone operations.

Key Features:

·Utilizes computer vision and ArUco markers for positioning

·Implements an Alpha-Beta-Gamma filter for stable flight control

·Achieves 10cm landing accuracy in GPS-denied environments

Applications:

·Autonomous package delivery in urban environments

·Search and rescue operations in remote or disaster-stricken areas

·Precision agriculture and crop monitoring

System Architecture

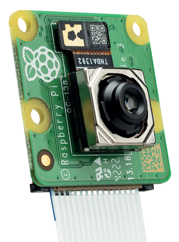
Hardware:

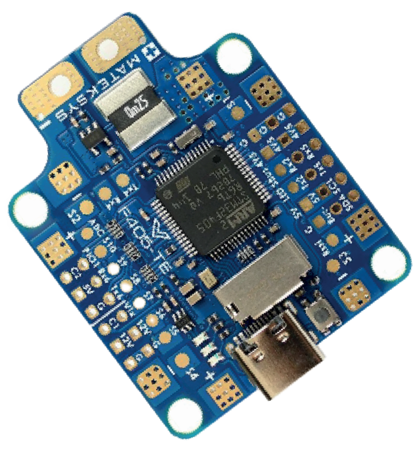
·Drone Platform: Custom-built quadcopter

·Camera: Raspberry Pi Module v3 (120° field of view)

·Flight Controller: Matek F405-TE (MAVLink compatible)

·Onboard Computer: Raspberry Pi 4 (4GB RAM)





Software:

·OpenCV for image processing and ArUco marker detection

·Custom Python scripts for drone control and navigation

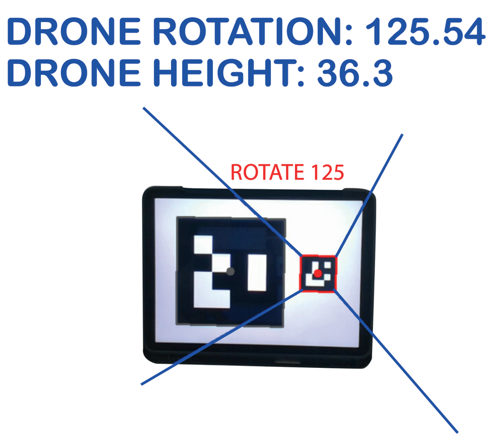
·Alpha-Beta-Gamma filter implementation for trajectory smoothing

ArUco Marker System:

·Large Marker (2m x 2m): Used for initial positioning from heights up to 30m

·Small Marker (0.5m x 0.5m): Employed for final precision landing from 5m and below

Key Observations

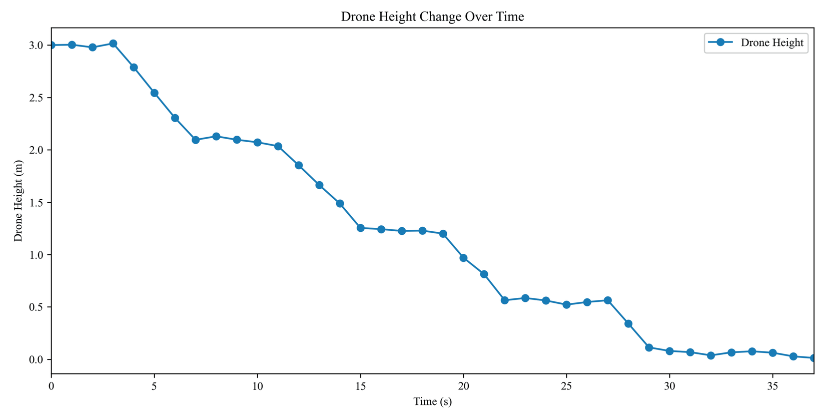


1. Drone Height vs. Time:

·Steady decrease during active descent phase

·Plateaus during horizontal centering for fine adjustments

·Final rapid descent during landing phase

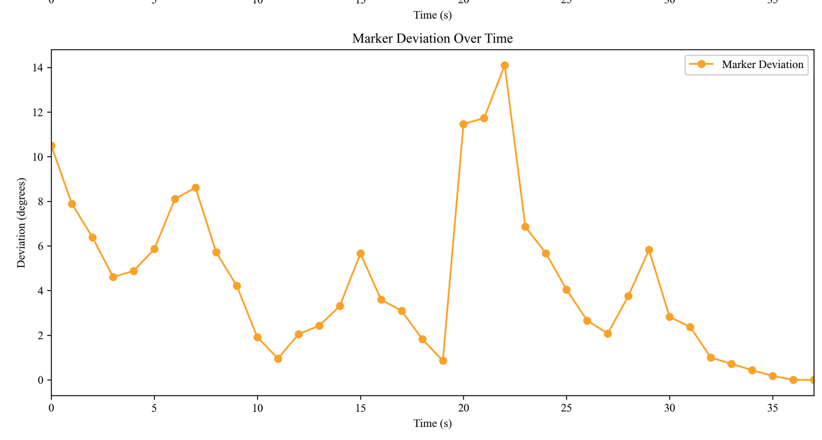


2. Relative Deviation vs. Time:

·Fluctuations increase during initial descent as the drone adjusts its position

·Sharp increase observed when switching from l arge to small ArUco marker

·Gradual decrease during final approach, indicating successful centering

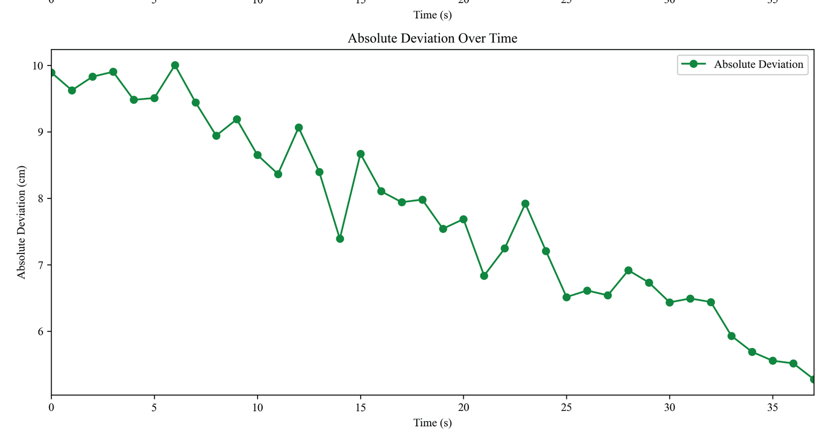


3. Absolute Deviation vs. Time:

·Overall decreasing trend throughout the landing process

·Temporary increases correspond to phase transitions (e.g., marker switch)

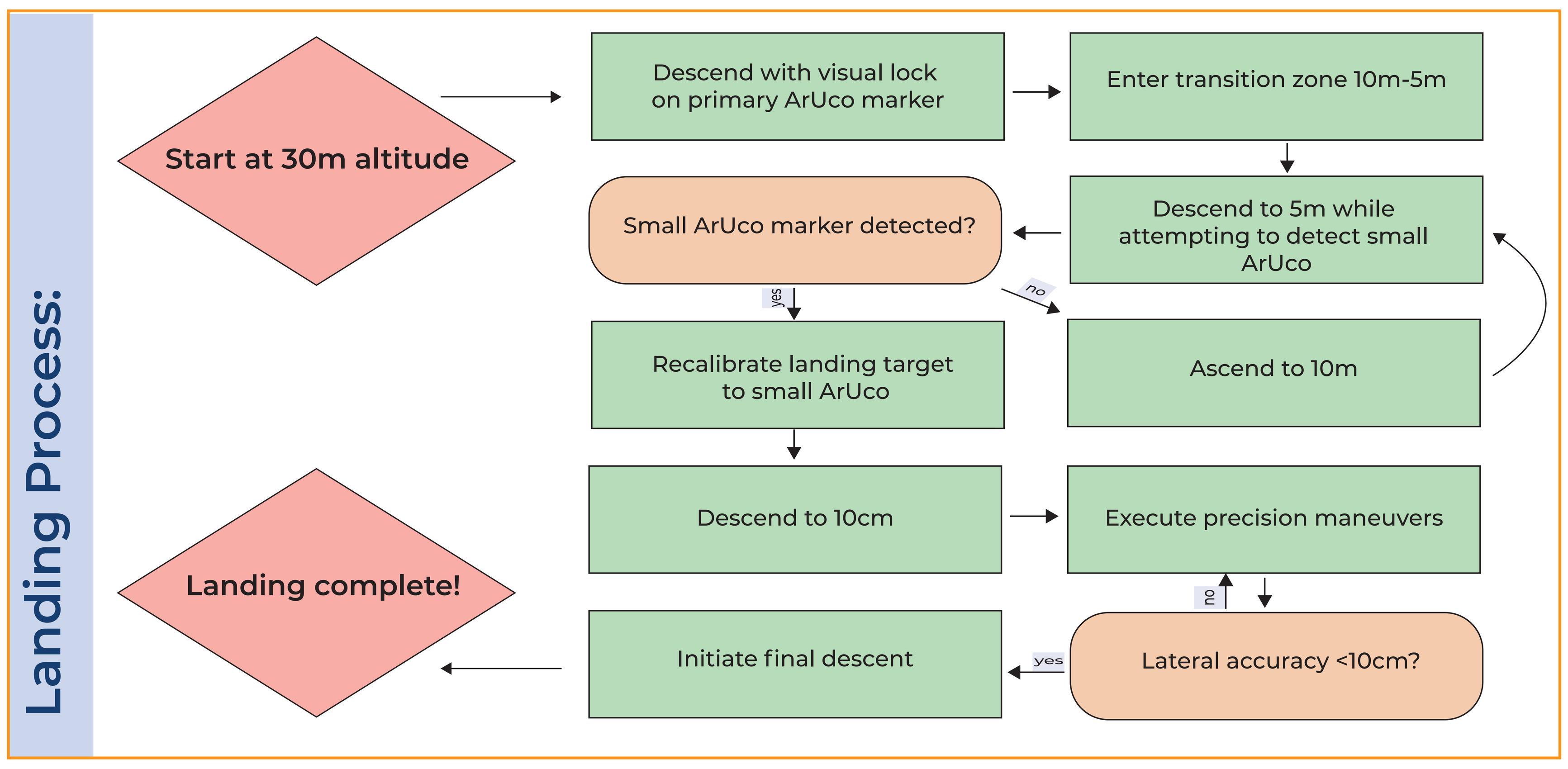
·Final values consistently below 10cm, meeting our accuracy target



ArUco Marker Detection Process

1. Image Acquisition: Capture frame from Raspberry Pi camera
2. Preprocessing: Apply image enhancements (e.g., contrast adjustment)
3. Marker Detection: Utilize OpenCV's ArUco library to identify markers
4. Pose Estimation: Calculate marker position and orientation relative to camera

Considering the real physical size of ArUco markers, we can use camera image and trigonometric formulas to translate pixel measurements to real-world distances and calculate:



1. Real-world distance to the marker
2. Drone's height above the ground
3. Drone's location relative to the marker
4. Drone's velocity
5. Drone's rotation

Distance	Drone Rotation	Height
$d = \frac{p_d}{p_w} \times w_a$	$\alpha = \arctan 2(y_c - y_d, x_c - x_d)$	$h = \frac{w_f}{2\tan(\theta/2)}$

α-β-γ filter

We implement an Alpha-Beta-Gamma (α-β-γ) filter to predict the drone's ArUco position reducing the impact of measurement noise and sudden movements.

The α-β-γ filter operates in three main steps:

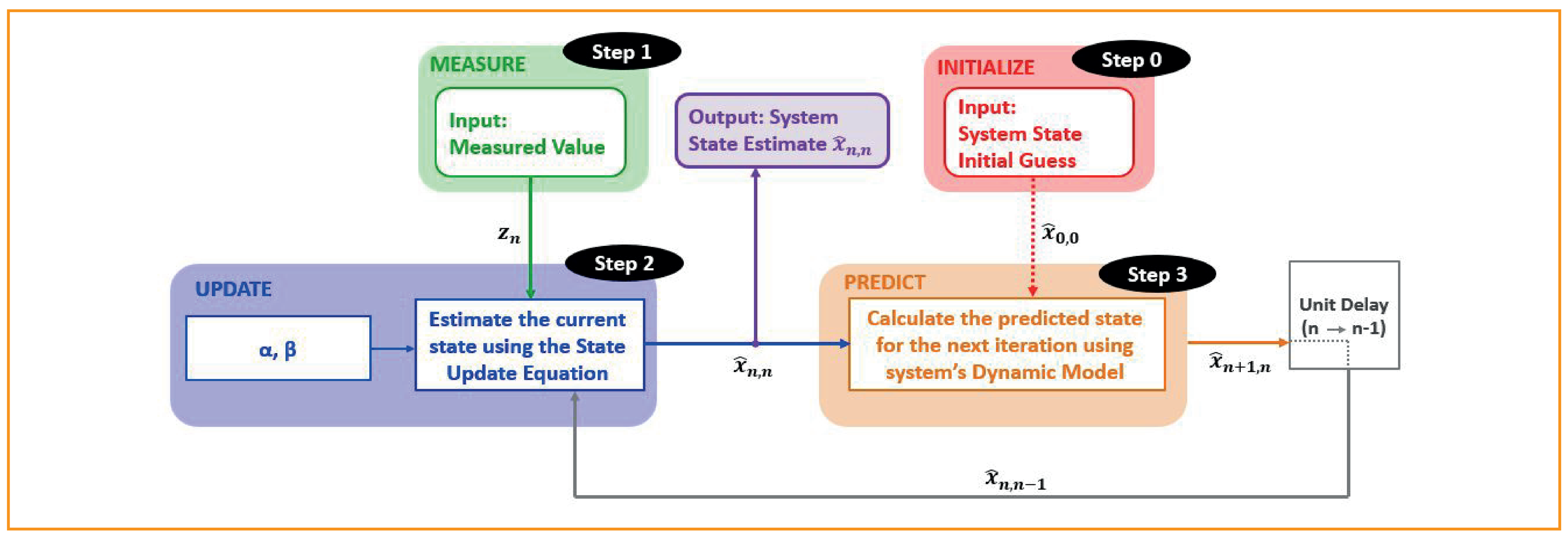
1. Prediction: Estimate the next state based on previous measurements
2. Measurement: Obtain new data from sensors
3. Update: Combine prediction and measurement to get the final estimate

Filter parameters:

·α (alpha): Adjusts the influence of the most recent measurement

·β (beta): Accounts for the linear trend in the data

·γ (gamma): Incorporates acceleration for more accurate predictions



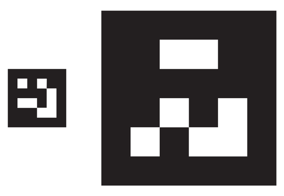
Statistical Analysis

·Mean landing accuracy: 7.2cm (based on 10 test flights)

·Standard deviation: 1.8cm

·Success rate (landing within 10cm): 90%

·These results validate the effectiveness of our dual-marker system and Alpha-Beta-Gamma filtering approach in achieving precise, repeatable landings.



Challenges and Future Improvements

·Lens Distortion Correction

·Advanced Filtering Techniques

·Environmental Adaptability

·Obstacle Avoidance

References:
Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F. J., & Marín-Jiménez, M. J. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recognition, 47(6), 2280-2292. Romero-Ramirez, F. J., Muñoz-Salinas, R., & Medina-Carnicer, R. (2018). Speeded up detection of squared fiducial markers. Image and Vision Computing, 76, 38-47.Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering, 82(1), 35-45. Meier, L., Tanskanen, P., Heng, L., Lee, G. H., Fraundorfer, F., & Pollefeys, M. (2012). PIXHAWK: A micro aerial vehicle design for autonomous flight using onboard computer vision. Autonomous Robots, 33(1-2), 21-39.

Blösch, M., Weiss, S., Scaramuzza, D., & Siegwart, R. (2010). Vision based MAV navigation in unknown and unstructured environments. 2010 IEEE International Conference on Robotics and Automation, 21-28. OpenCV Documentation. ArUco Marker Detection. https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html
Faragher, R. (2012). Understanding the Basis of the Kalman Filter Via a Simple and Intuitive Derivation. IEEE Signal Processing Magazine, 29(5), 128-132.
Entire code for the project: https://github.com/or7nge/Delivery-Drone