# Autonomous precision landing of unmanned aerial vehicles using fiducial optical markers

1st Jakub Prvý
*Dept. of Control and Instrumentation*
*Brno University of Technology*
Brno, Czech Republic
247060@vutbr.cz

2nd Jan Klouda
*Dept. of Theoretical and Experimental Electrical Engineering*
*Brno University of Technology*
Brno, Czech Republic
xkloud04@vutbr.cz

***Abstract*— This paper deals with the design and implementation of a system for autonomous landing of an unmanned aircraft using image recognition technology. The main objective is to use ArUco markers as visual reference points that allow accurate localization and navigation during the landing process. A key part of the work is the integration of the OpenCV library for marker detection and image analysis with the ArduPilot platform, which provides flight control.**

***Keywords—landing, ArUco, Python, MAVLink, unmanned aircraft, autonomous***

## I. Introduction

Drones are increasingly becoming a normal part of our lives, finding applications in cinematography, transport, logistics, reconnaissance and rescue operations. Unmanned aerial vehicles, also known as drones, play a key role in these fields due to their ability to perform complex tasks with minimal human intervention. Recently, there has been a great emphasis on fully autonomous operation of drones due to the use of large numbers that would be impossible to ensure safe and efficient flight using human pilots alone. For landing, one of the most critical features is landing accuracy and reliability.

The objective of this paper is to design, implement and debug a system for autonomous landing of an unmanned aircraft using image recognition technology. For this purpose, ArUco markers will be used as visual reference points for easy implementation of the recognition software. These markers will be recognized and processed using the OpenCV library for Python. The system will be implemented using the MAVLink communication protocol and the ArduPilot autopilot software suite, allowing fast and efficient communication between the recognition software and the controller.

## II. Types of landing

Safe landing is important for the operation of unmanned aircraft. It is divided into manual and autonomous landing. For manual landing, the knowledge and experience of the pilot is important and determines how fast or accurately the aircraft will land. Manual landing cannot be replicated due to the effect of human error in control. Autonomous landing, on the other hand, is controlled purely by the autopilot control unit. In this method, parameters such as landing speed, accuracy, etc. can be precisely defined.

### A. GNSS based landing

Global Navigation Satellite Systems (GNSS) are systems that can provide coverage of the Earth's surface with navigation signals with a limited number of radio beacons, allowing positioning anywhere on Earth. The main advantage of these systems is that they can determine position in a single coordinate system, anywhere on Earth, regardless of the time of day or weather [1].

*1) Standard GNSS:* Ground-based receivers, such as the sensors in your phone, pick up signals from satellites orbiting the Earth. The signal carries information about the satellite's position and the time it was sent. Based on the time delay between when a signal is sent from a satellite and when it is received, the receiver can determine its distance from each satellite. It uses a method called trilateration, where the position is calculated as the intersection of spheres around the satellites, each representing the receiver's distance from that satellite. A minimum of three satellites are needed to determine a two-dimensional position (latitude and longitude), four for a three-dimensional position (including altitude) [2]. This method achieves accuracy of a few meters, which is not suitable for precision landing.

*2) Differential GNSS (DGNSS):* Differential GNSS (DGPS) is a method of improving the accuracy of standard GNSS by using correction signals from a reference station. In conventional GNSS, the receiver calculates its position based on time-delay signals from satellites, but these signals can be affected by various errors such as atmospheric interference, reflections or clock inaccuracies. DGNSS minimizes these errors by using a reference station with an accurately known position [3]. This method is suitable for use in autonomous landing in smaller areas.

*3) Real-Time Kinematic (RTK) GNSS:* RTK (Real-Time Kinematic) is a high-precision positioning method that uses the differences between signals from GNSS satellites (e.g. GPS) and the phase shift of their carrier wave. Unlike conventional GNSS, which is capable of determining position within meters, RTK can determine position to within centimeters, making it a key technology for applications requiring extreme accuracy, such as autonomous driving, surveying, or agriculture. A reference station, located at a precisely known position, plays a vital role in the RTK system, receiving signals from satellites, analyzing them, and calculating errors based on known positions and phase shift of the carrier wave. It then transmits this correction information in real time to mobile GNSS receivers in range, which can then determine their position more accurately, even while moving [4].

### B. *Inertial navigation landing*

Inertial navigation landing uses sensors that measure changes in the motion and orientation of the aircraft (or UAV) without the need for external reference points such as GPS or visual markers. The inertial navigation system consists of gyroscopes and accelerometers that continuously sense rotations and accelerations in all axes. Based on this data, the inertial navigation system calculates changes in the aircraft's speed, position and orientation from its starting point.

### C. *Camera-assisted landing*

Camera-assisted landing allows unmanned aerial vehicles (UAVs) and drones to accurately navigate to a landing site using image analysis. The camera scans the environment in front of or below the aircraft, with the software targeting specific visual features or markers (such as ArUco tags or other optical markers) to estimate position and movement relative to the landing pad.

*1) Optical marker landing:* Optical marker landing is a technique used to land unmanned aerial vehicles (UAVs) accurately and reliably. The principle of this method is the use of special visual markers that are placed on the landing area and serve as a visual reference [5]. The tags, such as ArUco tags, have specific patterns that are easily detectable by camera systems and allow for accurate positioning of the UAV relative to the landing pad.

*2) Computer vision and machine learning:* Computer vision and machine learning allow the UAV to autonomously identify the landing area and obstacles in real time. Using a camera, the UAV captures an image of the environment and computer vision algorithms analyze the image to identify patterns, shapes or colors that correspond to the landing zone. This is often done by trained machine learning models that learn to recognize the distinctive features of the landing area from the vast amount of image data [6]. As a result, the UAV can accurately identify the location and dimensions of the landing area even in environments with varying background or lighting conditions.

## III. MARKER DETECTION

Position estimation is very important in many computer vision applications: robot navigation, augmented reality, etc. This process is based on finding connections between the real environment and the two-dimensional image projection. This step is usually challenging, and therefore the use of artificial optical reference markers is common.

One of the most used techniques is the use of binary square reference markers. Their main advantage is that one marker contains enough points to learn the camera position (4 corners). Another advantage is the binary encoding of the information inside the marker, this allows error detection and correction.

### A. *OpenCV*

OpenCV is a popular open-source library for image processing and computer vision. It was developed to provide powerful and efficient tools for image analysis, object detection, pattern recognition and machine vision [7].

### B. *ArUco tags*

ArUco tags are special black and white square markers used in computer vision for identification and localization of objects in space.

One of the main advantages is the quick recognition of the marker in the image thanks to the external frame. Inside this frame, the unique identifier and error correction data are encoded in a binary matrix. Due to the square shape, we can determine exactly where the marker is in space from the known marker size and the 4 corners.

The detection itself takes place in the following steps. *Initialization:* The OpenCV library uses dictionaries to distinguish marker types for ArUco marker detection. They can range in size from 4x4 to 7x7 and can contain different sets of unique identifiers. If we choose a dictionary with a larger number of identifiers. *Detection:* ArUco tags can be detected using the detector function. The detector then returns information about the identifier, the corners of the marker, and the incorrectly detected part of the marker [8]. Figure 1 shows detected marker inside green square labeled with ID number of the tag. The red square in the corner of marker is first corner of the tag. With this information we can determine the rotation of the marker.

OpenCV offers a function to calculate a perspective transformation that can determine the exact position of the marker in the image, i.e. relative to the drone. The function needs to know the camera matrix, which stores the outer and inner parameters of the matrix, as well as the distortion coefficients that allow the correction of image distortions caused by light passing through the optical system. The return values of the function are the X,Y and Z coordinates of the marker relative to the camera [9].

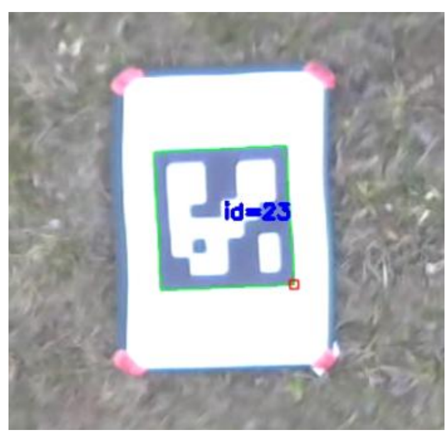


Fig. 1. Detected marker from drone camera

## IV. PRECISION LANDING

### A. MAVLink

MAVLink is a communication protocol used in robotic systems and unmanned aerial vehicles. It is used in these areas because of its low latency, cost-effectiveness and reliability. Compared to standard types of transmissions such as AM/FM signals or PPM, MAVLink provides bi-directional and efficient data exchange.

MAVLink transmits data in the form of short messages that contain information about the drone's status, its telemetry or navigation commands. Each message contains a voice message identifying the message type, a destination ID, and a checksum to ensure data integrity [10].

This protocol includes a message that conveys information about the location of the landing target. The message type is called LANDING_TARGET and carries all the necessary information for accurate landing. The message contains information about the time stamp, target ID, vehicle reference frame type, distance from the ground, landing target type and X,Y and Z coordinates.

### B. ArduPilot

ArduPilot is open-source autopilot software that has become one of the most popular systems for controlling drones, robots and other autonomous vehicles. ArduPilot uses the MAVLink protocol for communication, which enables data transfer between the autopilot and the ground station, providing reliable control and monitoring of autonomous operations [11].

For precise landing, it is necessary to set the parameters in the PLND_ group in ArduPilot. After activating the PLND_ENABLE parameter, other parameters for landing are displayed. The most important thing is to set PLND_EST_TYPE to 0, otherwise the landing process is very unstable and unsuccessful. Next, set the PLND_TYPE parameter to 1, i.e. landing using the MAVLink protocol information [12].

Detection and calculation of the landing target position are performed on a computer mounted on the drone body. All processing of the target position information is done in a Python script. The position of the target is transmitted via a MAVLink message to the autopilot in the drone's control unit. The actual guidance and landing process is handled in the control unit. The ArduPilot implementation supports precision landing only when assisted with GPS, for safety reasons. If the target were to be visually lost and GPS was not present the drone could behave erratically.

## V. TESTING

Testing was carried out on a Holybro X500 V2 drone with Pixhawk flight controller. The computer used was an Intel NUC with a Logitech C290 webcam.

For testing purposes, manually flying over the target is sufficient. Once the algorithm detects it, it starts sending messages about its location. In Mission Planner, a message will appear indicating that the target has been successfully detected. At this point, just switch the drone to LAND mode. Autonomous landing will begin and the drone will begin to descend. If the target is lost from the camera's view, it will fly higher and try to reacquire the target. If it fails to re-find it, it will land vertically at the current location.

Depending on GPS accuracy and properly set parameters, the drone can achieve landing accuracy from 5 to 15 centimeters from the target.

Fig. 2. Landing attempt accuracy

## VI. CONCLUSION

Based on testing, we can say that the system is accurate enough for landing in tight spaces or small landing areas. One potential refinement is to use multiple ArUco tags of different sizes, where the program can switch between them dynamically based on altitude.

The disadvantage of this system is that it only works in good light conditions due to visibility. Usability may be limited even in fog. Future improvements could focus on enhancing robustness by integrating additional sensor inputs or developing more advanced image-processing techniques to mitigate these challenges.

## REFERENCES


[1] P. Rapant, "Družicové polohové systémy", Publikace, Institut ekonomiky a systémů řízení, 2002.

[2] P. J. G. Teunissen and O. Montenbruck, *Springer Handbook of Global Navigation Satellite Systems*. Springer, 2017.

[3] A. Delépaut, A. Minetto, and F. Dovis, "Code-Based Differential GNSS Ranging for Lunar Orbiters: Theoretical Review and Application to the NaviMoon Observables", *Remote Sensing*, vol. 16, no. 15, p. 28, 2024.

[4] "Performance Evaluation of the RTK-GNSS Navigating under Different Landscape", p. 5.

[5] Q. Huang, J. DeGol, V. Fragoso, S. Sinha, and J. Leonard, *Optimizing Fiducial Marker Placement for Improved Visual Localization*. 2022/11/02.

[6] B. Ali and R. Sadekov, "A Review of Navigation Algorithms for Unmanned Aerial Vehicles Based on Computer Vision Systems", *Gyroscopy and Navigation*, vol. 30, pp. 87–105, 2023/01/30.

[7] "About", *OpenCV*, 02025.

[8] "Detection of ArUco Markers", *OpenCV*, 02024. .

[9] "Perspective-n-Point (PnP) pose computation", *OpenCV*, 02024..

[10] "Mavlink Basics", *ArduPilot*, 02024..

[11] "ArduPilot Documentation", *ArduPilot*, 02024..

[12] "Precision Landing and Loiter", *ArduPilot*, 02024. .