

# Proportional navigation missile guidance using image recognition

*MIEDZIŃSKI Dariusz\**, *RODO Piotr\*\**, *KACZMAREK Kacper\*\**, *SOCHACKI Mateusz\*\*\** and *HAŁOŃ Michał\*\*\*\**

*\*Division of Mechanics, Faculty of Power and Aeronautical Engineering, Warsaw University of Technology, Nowowiejska 24, 00-665 Warsaw, Poland*

*\*\*Students' Space Association, Faculty of Power and Aeronautical Engineering, Warsaw University of Technology, Nowowiejska 21/25, 00-665 Warsaw, Poland*

*\*\*\*Division of Automation and Aeronautical Systems, Faculty of Power and Aeronautical Engineering, Warsaw University of Technology, Nowowiejska 24, 00-665 Warsaw, Poland*

*\*\*\*\*Institute of Control and Computation Engineering, Faculty of Electronics and Information Technology, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland*

## Abstract

Proportional Navigation Guidance (PNG) is the most frequently used guidance scheme in missiles' control systems. Since the time it was first described, many modifications have been developed to enhance its effectiveness. In its basic formulation, it requires only the information about the missile's closing velocity and a rate of change of the target's relative angular position. The location of the missile's target is usually not known a priori with high precision, which makes target detection an important aspect of any guided missile. Various target detection methodologies like radar, lidar or electro-optics are used.

This paper will study the impact point precision and control quality of a guided missile equipped with vision-based navigation. The control scheme utilizing various types of image recognition algorithms and methods of converting its output to the input of control algorithms will be investigated. A nonsymmetrical ArUco marker of 4 by 4 units size was chosen as a mission target. Target detection will be performed using algorithms based on basic image processing methods, such as thresholding, morphological operations and edge detection, as well as using deep neural networks. Guidance towards the target will utilize the three-loop acceleration autopilot using proportional navigation for generating commands. The control signals will be converted to four independent forward control surfaces' deflections.

In order to achieve satisfying control effectiveness, the commands from the guidance system to the control effectors should be generated with a high frequency. This requires a high computational power of the on-board computer to perform both the image recognition and generating control commands. Designing a small and cost-effective missile presents significant challenges. The bottleneck is the output frequency of the visual navigation system, as a powerful dedicated computer for image analysis requires a significant amount of space and power supply.

There exist several ways of overcoming this problem. One of them is the estimation of the target's absolute position, which allows to utilize this information in the control algorithm in longer timespans, as the target might be considered stationary. This can be achieved by combining the absolute position of the rocket - obtained from the inertial navigation and satellite systems - with the target's position relative to the rocket. Target's relative position is determined from a single camera frame using target's visual features, such as size, shape, or angular orientation.

The research described in this paper is a part of the FOK project. FOK is an aerodynamically controlled rocket being developed by the Rocketry Division of the Students' Space Association at the Warsaw University of Technology since 2017. The rocket, 75 mm in diameter, 1.4 m long, equipped with four canards, was designed as a cheap and fully reusable platform intended for studying guidance, navigation and control algorithms.

## 1. Introduction

Precision guidance is a very important aspect of missile control systems development. It is an interdisciplinary problem involving development of both hardware and software components. Various types of sensors are used to detect the target such as radars, infrared (IR) or imaging infrared (IIR) or television (TV). Depending on the choice of the sensor, different types of information about the target can be obtained. In all cases the angular displacement of the target is provided. Radars can also measure range to the target and range rate, however as they are an active type of a sensor, they give the target information that it is being detected. The IR, IIR, visual sensors are passive, they utilize only the energy that is being emitted from the target, but they are unable to measure the range (directly). The choice of the sensor, as a result, is not a straightforward task and requires it to be compatible with the guidance law used.

Many different guidance and control laws are used in the field of missile design. Sliding Mode Control is a robust control method dealing well with model uncertainties and external disturbances. In [1] a guidance law effective against uncertainties and disturbances using second order sliding surface is developed accounting for the target maneuver and impact angle constraint. Authors of [2] developed a guidance law based on the adaptive fast nonsingular terminal sliding mode control theory to intercept a maneuvering target. A robust observer-based guidance law using sliding mode control avoiding chattering is presented in [3]. Authors of [4] designed a nonsingular terminal sliding mode guidance accounting for impact angle constraints for varying speed interceptors.

Many variants of optimal control are utilized as it allows for control and state constraints to be specified. An optimal guidance law with impact time and impact angle constraints was developed in [5] for hitting a stationary target. Authors of [6] designed an adaptive three-dimensional guidance law based on proportional navigation using convex optimization for choosing the gains. An optimal guidance law that directly uses gravity for generating acceleration commands was developed in [7]. Very interesting variant of optimal control is Model Predictive Control (MPC). Instead of calculating control commands offline, as it is usually done in optimal control, in MPC the optimization is done online at every control step. A robust model predictive control using neural network optimization is designed in [8]. Authors of [9] designed a control strategy of the coordination of impact time of a group of interceptors using nonlinear model predictive control. A guidance law using stochastic model predictive control to achieve minimum miss distance satisfying impact constraints was developed in [10].

One of the most widely used precision guidance law is the Proportional Navigation Guidance (PNG) and its variants. It needs only information about the Line of Sight (LOS) angular rate and closing velocity, or missile velocity when the target is stationary. Authors in [11] developed a guidance law based on Pure Proportional Navigation to intercept a moving non-maneuvering target with desired impact angle constraints while avoiding obstacles. The coefficient diagram method with the proportional navigation guidance are used in the short range Surface-to-Air missile control system design in [12]. In [13] a study of the possibilities of the impact point dispersion reduction for a gasdynamically controlled projectile using the proportional navigation guidance was performed. Impact time control using data-driven method based on proportional navigation is presented in [14].

In this work a Proportional Navigation Guidance will be used along with the three-loop triple-channel acceleration autopilot to guide the rocket towards a ground-based target. The results of the guidance will be provided based on the characteristics of the FOK guided missile developed by the Rocketry Division of the Students' Space Association (SSA). The rest of the paper is organized as follows. In the Section 2 a short introduction to the work of the Rocketry Division will be provided along with the description of the FOK missile. Section 3 will present the mission concepts, description of the used target as well as utilized control scheme. The methods of detecting marker from the image frame will be discussed in Section 4. Section 5 will describe the methods used to convert the output of the image detection algorithm to the input needed for the control law. In Section 6 the results of the performed simulations will be presented and Section 7 will give a short summary of the paper.

## 2. FOK project

Students' Space Association was established in 1996 at the Faculty of Power and Aeronautical Engineering of Warsaw University of Technology. For more than 10 years, the Rocketry Division has been designing, manufacturing and launching sounding rockets. One of the projects developed by the members of the Association is the FOK guided rocket. The project was started at the end of 2017. For the first version of the project, control algorithms that use data provided by inertial sensors in order to perform the rocket's missions have been developed. For the second version of the project, the team has decided to incorporate means to actively search for a ground-based target, allowing the rocket to navigate towards it.

Three iterations of the rocket were built since the beginning of the project, named FOK-1A, FOK-1B, and FOK-1C. Each resulted in a successful launch in May 2018, September 2019 and May 2021, respectively. These launches served the purpose of testing all the subsystems of the rocket: propulsion, recovery, and onboard electronics.

They highlighted any necessary modifications and design changes that led to the fourth iteration, FOK-1D, with its flight planned in September 2022.

The FOK rocket is aerodynamically controlled, propelled by a solid rocket motor. The airframe of FOK is made mostly out of aluminium and consists of five separate modules: nose cone, control module, avionics module, recovery system, and rocket motor (see Figure 1). The project's goal was to build a simple and relatively inexpensive testbed for automatic flight control systems.

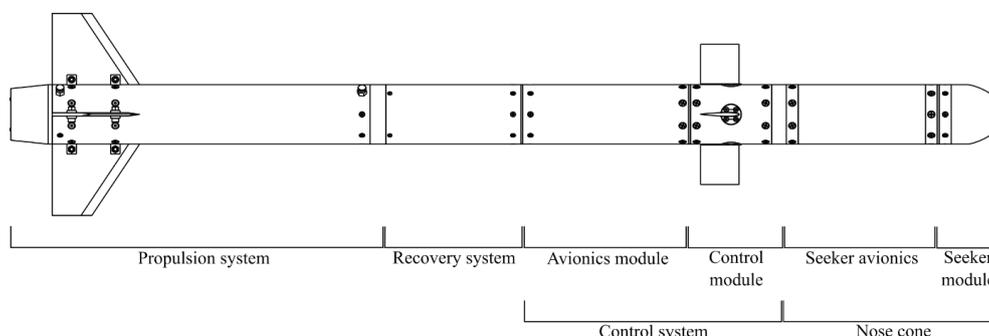


Figure 1: FOK-2 subsystems.

The nose cone houses the secondary, redundant on-board computer, as well as telemetry and GPS antennas. It is partially made of glass fibre composite. Four servomechanisms, responsible for deflecting the canards, are located in the control module. The avionics module stores an in-house developed primary onboard computer, called Ganymede. The computer is used for sending control signals to the servomechanisms and recording flight data provided by the onboard GNSS/INS unit. Safe recovery of the whole rocket is possible thanks to the recovery system. Triggered pyrotechnically it splits the rocket into two parts, both connected to the parachute. The propulsion system is a solid rocket motor. During slightly more than 2 seconds of work, it provides 590 N of mean thrust. The rocket is capable of reaching speed of about Mach 0.6 and an apogee of about 1700 m [15].

In order to increase the rocket's guidance capabilities, the FOK-2 project was started. The project aims to develop a guidance system based on live data, provided by an on-board camera placed in the nose cone, either a visible-light-spectrum or an infrared one. The data would then be processed using machine vision and image recognition techniques, enabling the rocket to perform a fully autonomous, guided flight to a ground-based target.

### 3. Mission plan

The mission of FOK-2 rocket is to perform a controlled descent towards a ground-based target utilizing image recognition algorithms. The controlled flight phase is relatively short – it is a period between the apogee and the moment at which a parachute is deployed, see Figure 3. The rocket lands on a parachute instead of impacting the target to limit the costs of the project by making the rocket reusable. For the simulation purposes and to have a means of evaluating the performance of the used algorithms, the parachute descend phase of the flight was skipped, so that the missile can impact the target. The target is a 5 by 5 meters ArUco marker (see Figure 2). The reasoning behind selecting this type of marker is presented in [16].

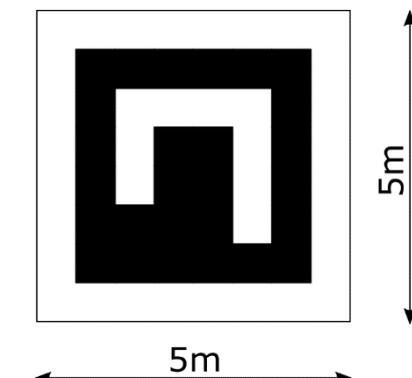


Figure 2: ArUco marker used as a target.

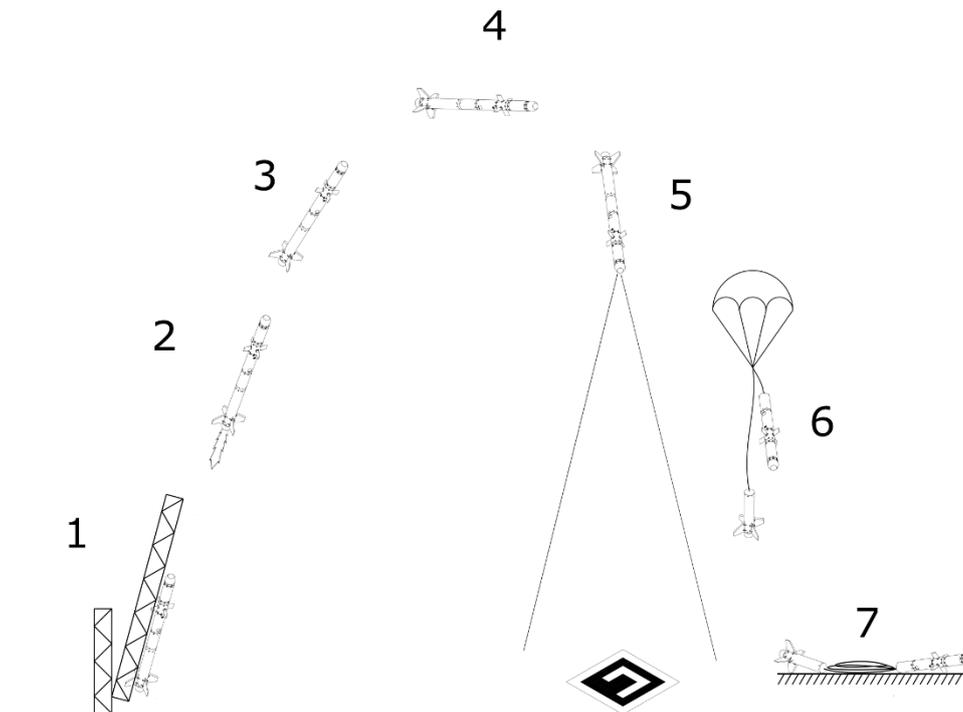


Figure 3: FOK2 mission diagram. 1- launch from launch rail, 2 – powered ascent, 3 – unpowered ascent, 4 – apogee, 5 – marker detection and controlled descent, 6 – parachute deployment, 7 – parachute landing.

The control is realized with a three-loop triple-channel acceleration autopilot using Proportional Navigation Guidance (PNG) law for generating commands as shown in Figure 4. The PNG algorithm was derived for the 3D case using the methodology presented in [17, 18]. All the PID controllers were tuned manually from inner loop to outer. The inner loop uses the feedback from angular velocities in three channels of yaw, pitch and roll and generates the control signal for canards' deflections. The middle loop uses feedback from the angle by which the missile rotated, calculated by integrating the angular velocities in pitch and yaw channels. In the outer loop the error is calculated as the difference between measured accelerations and the commanded accelerations provided by the PNG algorithm.

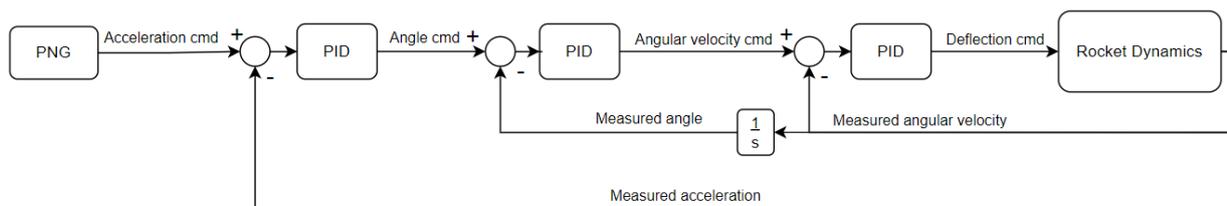


Figure 4: Block diagram of the used control law.

#### 4. Target recognition algorithms

One of the FOK rocket main navigation system components is the target detection algorithm, whose task is to detect a marker in an image obtained from the camera seeker. The algorithm should be able to detect the marker during the flight after burnout of solid rocket motor as early as possible to enable rocket navigation to the target as quickly as possible. It should also convey information about the detected marker in a strictly defined form (given coordinates of the object in the analysed photo) and be resistant to erroneous detection of objects that are not the flight target (so-called false positives). Moreover, the detection frequency should enable the proper functioning of the navigation algorithm by minimizing the delay in obtaining algorithm output in relation to the receipt of the input image, which implies a relatively low processing time on the on-board computer. The selected marker detection algorithm used in the navigation system should ultimately enable the rocket to hit target location.

This chapter describes marker detection algorithms used in the project, which can be divided into those using standard image processing methods and those using machine learning methods (neural networks). Both ready-made, available solutions and those developed for the project were used during testing phase.

## 4.1 Algorithms based on standard image processing methods

Among the standard image processing algorithms, two own methods (one based on edge detection, the other on the input image binarization) [16] and the function available within the OpenCV programming library (which is intended for ArUco markers detection) were used.

### 4.1.1 OpenCV ArUco

ArUco markers [19, 20] detection algorithm, available as a part of the OpenCV library, contains implemented function to detect several sets of markers, including the one containing a marker used in the mission as a navigation target. This algorithm has found many applications, including augmented reality and mobile robotics applications [22, 23]. It is divided into two main phases: marker detection (edge searching, combining them into quadrilaterals) and marker identification number decoding. The output of the algorithm is the detected marker corners' coordinates (in pixels) and its identification number (in this case, there is only one marker class).

The detector available as part of the OpenCV library has several parameters (such as thresholding, filtering or bits extraction fine tuning) enabling the algorithm adjustment to the current needs and requirements [24]. Due to this fact, it was decided to use two versions of this algorithm - with default parameters (with only the *minMarkerPerimeterRate* changed, which specifies the minimum size of the detected marker in the image) and second with parameters optimized by means of real photos.

### 4.1.2 Binarization-based

The first algorithm developed for the project purpose assumes that the black and white colours of the detected marker strongly differentiate from the background and all other objects on images acquired during the flight. This characteristic feature can be highlighted during the image binarization process with a properly selected threshold, after which often the largest object remaining in the resulting image is the target marker [16]. Thus, described algorithm identifies the largest object remaining after the binarization process as a detected marker. Additional modifications to this method are also considered, such as verification of the marker presence on the detected image (to reduce false-positive detections, especially when the marker is not in the camera's field of view at all).

### 4.1.3 Edges-detection-based

Another algorithm developed within the project is edges-detection-based one [16], which in its complete version, similarly to the algorithm available in the OpenCV library, is based on two stages: locating the marker on the image and then decoding the identification number. In the first stage, image processing methods such as binarization, gradient calculation and morphological operations are used [25, 26, 27]. The second stage is based on a comparative analysis of the corresponding areas brightness inside the marker. The output of the algorithm, as for the algorithm in OpenCV, are the detected marker corners coordinates.

Due to the problematic identification of the marker from long distances (the small size of the marker makes its code practically unreadable), it was decided to develop two versions of the described algorithm during the tests - the first, using only the marker detection stage, and the second, complete version. The first case allows the detection of the marker even from long distances, but at the same time exposes the system to erroneous detection of objects that are not the markers itself (due to the lack of its content verification), while the use of the complete version practically eliminates the risk of obtaining a false positive at the expense of the lack of distant markers detection. Within this study only the version with marker detection stage was used.

## 4.2 YOLO neural network

In recent years, there has been a noticeable increase in the use of machine learning techniques (including deep neural networks) in many areas. Often, artificial intelligence algorithms outperform the traditional methods used so far. Machine vision is an area where this trend is especially visible. This can be seen, for example, in the ILSVRC (ImageNet Large Scale Visual Recognition Challenge) competition results [28, 29]. Therefore, as part of the project, a decision was made to use the above-mentioned techniques in the task of marker recognition.

To achieve this goal, the YOLO (You Only Look Once) network [30], which is intended for the detection of many object types on images, was used as a first choice (more specifically, YOLOv5 network was utilized [31]). During the FOK rocket mission, there is only one class of object to detect, hence the performed task seems to be much simpler. The input of the YOLO network is an RGB image, while the output of it are the so-called bounding boxes - the coordinates of the rectangles inside which the detected objects are located, together with the detected object class prediction (in this case, there is only one class) – see Figure 5.

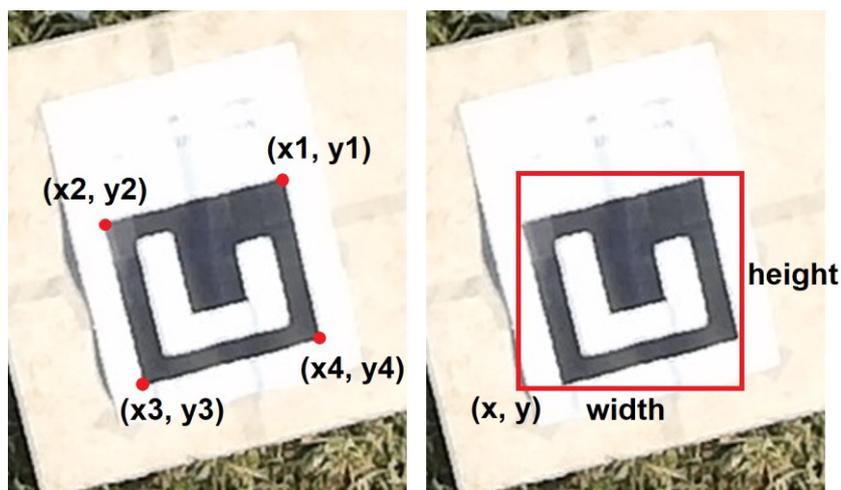


Figure 5 : Visualization of the marker localization in the image using coordinates of its corners (left) and the so-called bounding boxes (right).

To adapt the YOLO network to the mission's goal, it was retrained (transfer learning method) on the prepared training set of camera seeker images obtained during the mission simulation (see Figure 6).

Ideas for modifying the original YOLO network architecture to output marker corner coordinates instead of bounding boxes, or to output marker position vector and its orientation angles relative to the camera are also considered. In addition, other deep convolutional networks for object detection are also considered, such as various versions of the R-CNN network [32] and smaller neural network architectures dedicated to the presented issue.

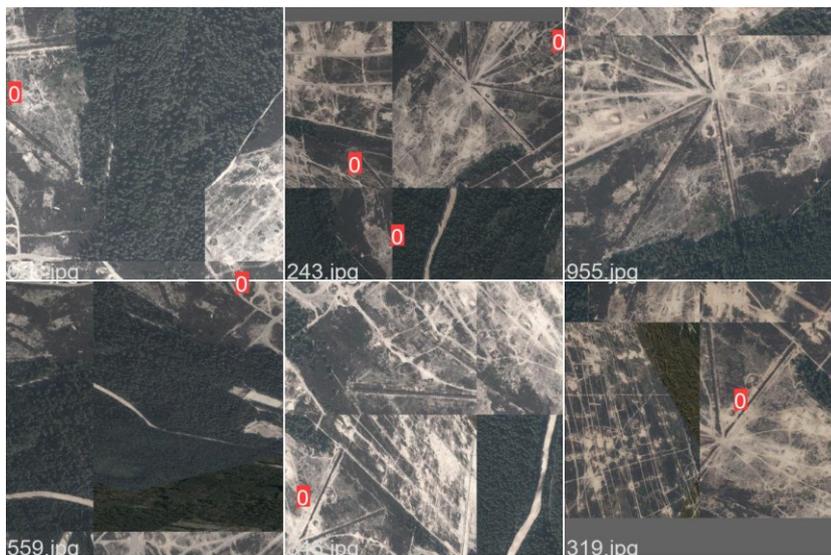


Figure 6: Sample simulation augmented images from the training set for the YOLO network (marker presence is indicated by a red rectangle).

## 5. Output conversion methods

To achieve a good quality of control the command signals should be generated with a high frequency. It was determined however that the image processing, as it requires high computational power, will not be able to satisfy this requirement. To not degrade the quality and results of the guidance and control processes a way of overcoming this issue was needed. Therefore instead of generating control commands based on the relative position of the target on the image frame, its absolute position on the Earth surface was determined. There are many ways of calculating the absolute target position. In this work three of them will be presented. All of them require the determination of direction vector towards the target. This direction vector can be determined based on image recognition using camera model. In the first approach marker position was determined calculating the intersection between the direction vector to the target center and the local tangent plane, using flat Earth assumption. In the second approach the local tangent plane was replaced with the

WGS-84 ellipsoid. Last approach, called camera pose estimation, utilized the direction vectors to the three corners of the target and the target's size to determine the position and orientation of the target relative to the camera and then transforming it into absolute position.

### 5.1 Flat Earth model

The general plane equation in the 3D case can be expressed by means of Eq. (4):

$$\mathbf{n}(\mathbf{r} - \mathbf{r}_0) = \mathbf{0} \quad (4)$$

where  $\mathbf{n}$  is a unit vector normal to the plane,  $\mathbf{r}_0$  describes some point lying on the plane and  $\mathbf{r}$  is a vector describing the position of any point of the plane. The plane normal unit vector can be taken as the vector opposite to the vertical unit vector of the NED coordinate system expressed in ECEF coordinate system. The line corresponding to the direction vector between the rocket and the target can be described by the Eq. (5):

$$\mathbf{r} = \mathbf{r}_c + \mathbf{v}t \wedge t \in \mathbb{R} \quad (5)$$

where  $t$  is a parameter, and  $\mathbf{r}_c$  is sensor position vector with respect to Earth expressed in ECEF coordinate system, and  $\mathbf{v}$  is the direction vector from the sensor frame to the target. The  $\mathbf{r}_0$  is the projection of missile's position onto the Earth's surface (horizontal plane). Using this nomenclature and substituting the terms in Eq. (4), the Eq. (6) can be obtained:

$$\mathbf{n}(\mathbf{r}_c + \mathbf{v}t - \mathbf{r}_0) = \mathbf{0} \quad (6)$$

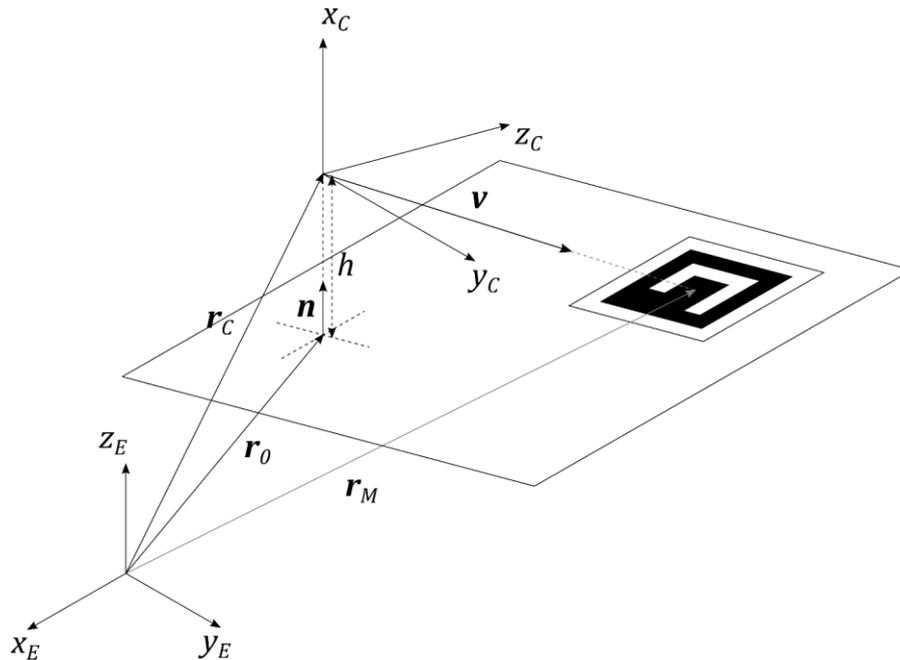


Figure 7: The relation between the marker's position and the camera's position above the surface of the Earth.

From Figure 7 the expression for  $\mathbf{r}_c$  can be derived as  $\mathbf{r}_c = \mathbf{r}_0 + \mathbf{n}h$ , where  $h$  is the missile height above the Earth's surface. Taking that into consideration the Eq. (6) can be rearrange and the expression for the parameter  $t$  can be obtained as Eq. (7):

$$t = \frac{\mathbf{n} \cdot \mathbf{n}h}{\mathbf{n} \cdot \mathbf{v}} \stackrel{|n|=1}{=} \frac{h}{\mathbf{n} \cdot \mathbf{v}} \quad (7)$$

The target absolute position given in ECEF coordinates can then be obtained from Eq. (5) substituting calculated from Eq. (7) parameter  $t$ .

## 5.2 Ellipsoidal Earth model

The Earth's ellipsoidal surface expressed in ECEF coordinate system is given by the Eq. (8):

$$\mathbf{r}^T \begin{bmatrix} a_{WGS-84}^{-2} & 0 & 0 \\ 0 & a_{WGS-84}^{-2} & 0 \\ 0 & 0 & b_{WGS-84}^{-2} \end{bmatrix} \mathbf{r} = 1 \quad (8)$$

where  $a_{WGS-84}$  and  $b_{WGS-84}$  are the semi-major and semi-minor axes of the WGS-84 ellipsoid and  $\mathbf{r}$  describes the position of any point on the surface of the ellipsoid. The line corresponding to the direction vector between the rocket and the target can again be described by the Eq. (5). Marker position is the point of intersection of the line defined by Eq. (5) and the ellipsoid from Eq. (8) and is given by the Eq. (9).

$$(\mathbf{r}_c + \mathbf{v}t)^T \begin{bmatrix} a_{WGS-84}^{-2} & 0 & 0 \\ 0 & a_{WGS-84}^{-2} & 0 \\ 0 & 0 & b_{WGS-84}^{-2} \end{bmatrix} (\mathbf{r}_c + \mathbf{v}t) = 1 \quad (9)$$

The above equation is a scalar quadratic equation which can be easily solved for  $t$  as presented in Eq. (10).

$$\begin{cases} At^2 + Bt + C = 0 \\ A = \mathbf{v}^T \begin{bmatrix} a_{WGS-84}^{-2} & 0 & 0 \\ 0 & a_{WGS-84}^{-2} & 0 \\ 0 & 0 & b_{WGS-84}^{-2} \end{bmatrix} \mathbf{v} \\ B = \mathbf{v}^T \begin{bmatrix} a_{WGS-84}^{-2} & 0 & 0 \\ 0 & a_{WGS-84}^{-2} & 0 \\ 0 & 0 & b_{WGS-84}^{-2} \end{bmatrix} \mathbf{r}_c + \mathbf{r}_c^T \begin{bmatrix} a_{WGS-84}^{-2} & 0 & 0 \\ 0 & a_{WGS-84}^{-2} & 0 \\ 0 & 0 & b_{WGS-84}^{-2} \end{bmatrix} \mathbf{v} \\ C = \mathbf{r}_c^T \begin{bmatrix} a_{WGS-84}^{-2} & 0 & 0 \\ 0 & a_{WGS-84}^{-2} & 0 \\ 0 & 0 & b_{WGS-84}^{-2} \end{bmatrix} \mathbf{r}_c - 1 \end{cases} \quad (10)$$

The location of the intersection point is given by smallest positive solution. Negative solution implies intersection point lying behind the sensor, and larger solution corresponds to intersection point lying on the opposite side of the Earth. The target absolute position given in ECEF coordinates can then be obtained from Eq. (5) substituting parameter  $t$  calculated from Eq. (10).

## 5.3 Camera pose estimation

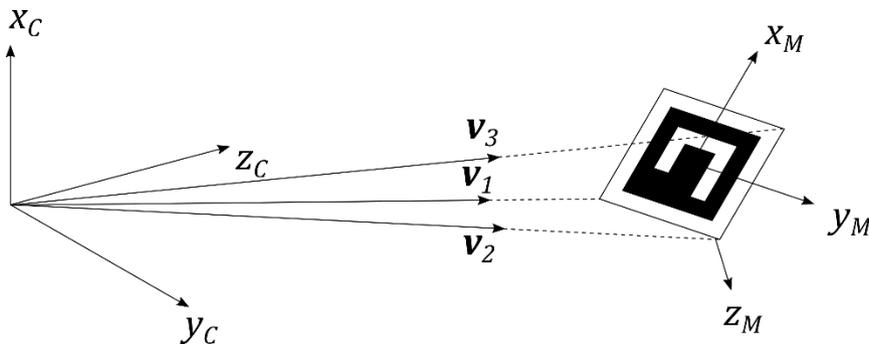


Figure 8: Geometry of camera pose estimation problem.

In the camera pose estimation problem it is assumed that the direction vectors describing the directions from the camera frame to three corners of the marker are known (see Figure 8). Knowing that the size of the marker is  $d$  the position vectors corresponding to the three corners with respect to the marker frame can be expressed as Eq. (11):

$$\mathbf{r}_{M1} = \begin{bmatrix} d \\ -\frac{d}{2} \\ \frac{d}{2} \\ 0 \end{bmatrix}, \quad \mathbf{r}_{M2} = \begin{bmatrix} -\frac{d}{2} \\ \frac{d}{2} \\ \frac{d}{2} \\ 0 \end{bmatrix}, \quad \mathbf{r}_{M3} = \begin{bmatrix} \frac{d}{2} \\ \frac{d}{2} \\ \frac{d}{2} \\ 0 \end{bmatrix} \quad (11)$$

Furthermore the vectors describing the displacement from corner 1 to corner 2 and from corner 2 to corner 3, expressed in marker frame are:

$$\mathbf{r}_{M12} = \begin{bmatrix} 0 \\ d \\ 0 \end{bmatrix}, \quad \mathbf{r}_{M23} = \begin{bmatrix} d \\ 0 \\ 0 \end{bmatrix} \quad (12)$$

Therefore the relation between successive corners expressed in camera frame can be written in the form of two vector equations:

$$\begin{aligned} \mathbf{C}_M^C \mathbf{r}_{M12} &= r_2 \mathbf{v}_2 - r_1 \mathbf{v}_1 \\ \mathbf{C}_M^C \mathbf{r}_{M23} &= r_3 \mathbf{v}_3 - r_2 \mathbf{v}_2 \end{aligned} \quad (13)$$

where the  $\mathbf{C}_M^C$  is the transformation matrix from the marker to the camera frame. This matrix can be expressed using three Euler angles  $\theta_1$ ,  $\theta_2$  and  $\theta_3$  ( $\mathbf{C}_M^C = \mathbf{C}_M^C(\theta_1, \theta_2, \theta_3)$ ). The above set of two vector equations has six unknowns – the distances from the camera to the three corners  $r_1, r_2, r_3$  and three angles  $\theta_1, \theta_2$  and  $\theta_3$  describing the relative orientation of marker and camera frames.

After solving the set of equations (see Eq. (13)), the marker position relative to the camera  $\mathbf{r}_{C2M}$  can be calculated using any of the three relations:

$$\begin{aligned} \mathbf{r}_{C2M} &= r_1 \mathbf{v}_1 - \mathbf{C}_M^C \mathbf{r}_{M1} \\ \mathbf{r}_{C2M} &= r_2 \mathbf{v}_2 - \mathbf{C}_M^C \mathbf{r}_{M2} \\ \mathbf{r}_{C2M} &= r_3 \mathbf{v}_3 - \mathbf{C}_M^C \mathbf{r}_{M3} \end{aligned} \quad (14)$$

and the absolute marker position can be calculated as:

$$\mathbf{r}_M = \mathbf{r}_C + \mathbf{r}_{C2M} \quad (15)$$

## 6. Simulation study

In order to test the prepared algorithms for target detection and output conversion methods a simulation study was conducted. First, the three output conversion methods were tested using the binarization-based detection algorithm, called Fast Finder. Figure 9 presents the estimated target's location error as function of distance to the target. The estimated target's location error can be calculated using Eq. (16), where  $\tilde{\mathbf{r}}_M$  is the estimated target location.

$$\delta_{loc} = \mathbf{r}_M - \tilde{\mathbf{r}}_M \quad (16)$$

The ellipsoidal Earth methods proved to be the most accurate, along with the flat Earth method resulting in errors with magnitude less than 1 meter. The biggest errors were encountered using the pose estimation method. Figure 10 presents the estimated target's direction error as function of distance to the target. The direction error is calculated by means of Eq. (17):

$$\delta_{dir} = \arccos \left( \frac{\mathbf{r}_{rel} \cdot \tilde{\mathbf{r}}_{rel}}{|\mathbf{r}_{rel}| |\tilde{\mathbf{r}}_{rel}|} \right) \quad (17)$$

where  $\mathbf{r}_{rel}$  and  $\tilde{\mathbf{r}}_{rel}$  are the relative target's positions calculated as  $\tilde{\mathbf{r}}_{rel} = \mathbf{r}_M - \mathbf{r}_C$  and  $\mathbf{r}_{rel} = \tilde{\mathbf{r}}_M - \mathbf{r}_C$ . The direction error in all cases were similar and very low which results in direct hits in all cases. Figure 11 presents the estimated target's distance error as function of distance to the target. Target's distance error is calculated using the Eq. (18).

$$\delta_{dist} = \left| |\tilde{\mathbf{r}}_{rel}| - |\mathbf{r}_{rel}| \right| \quad (18)$$

Again, the pose estimation method gave the biggest error and the ellipsoidal Earth gave the smallest error. It can be seen that for pose estimation method almost all of the location error from Figure 9 was due to the distance error.

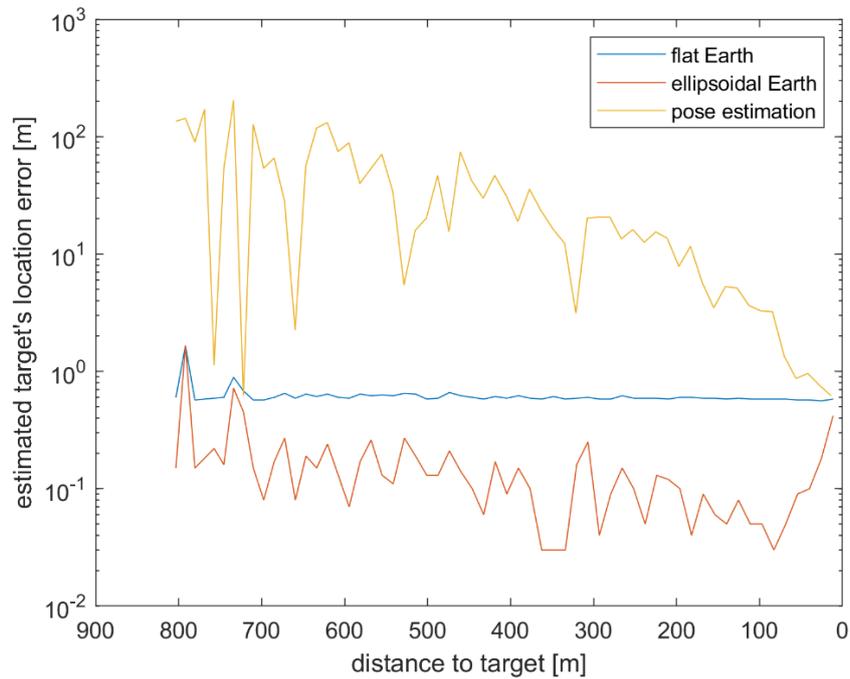


Figure 9: Estimated target's location error as function of distance to the target, for various output conversion methods.

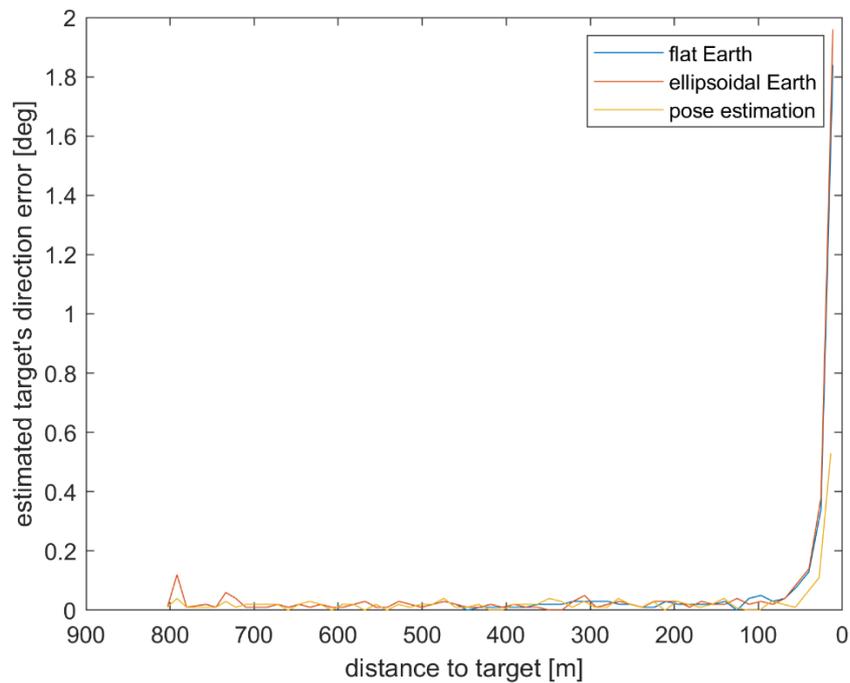


Figure 10: Estimated target's direction error as function of distance to the target, for various output conversion methods.

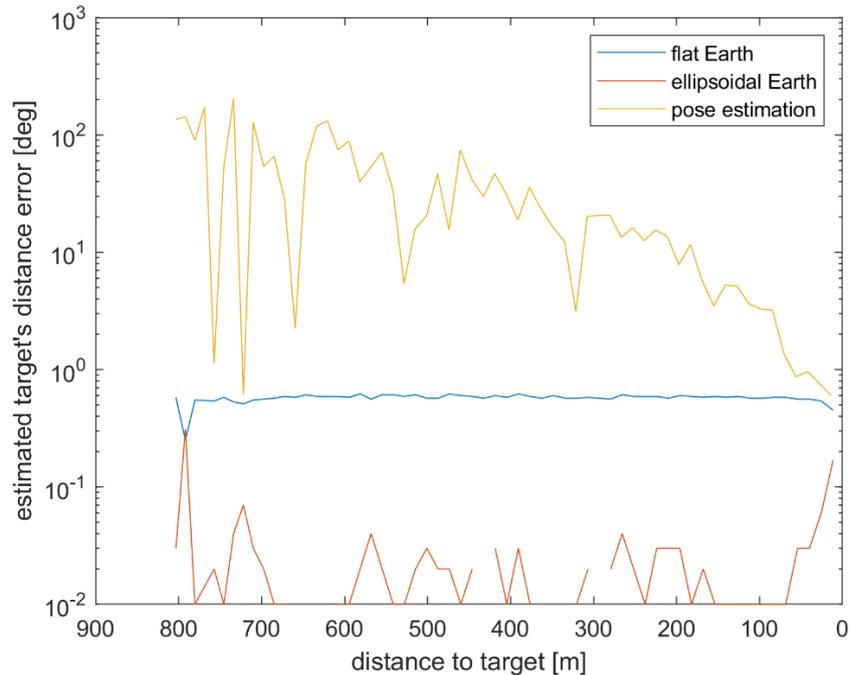


Figure 11: Estimated target's distance error as function of distance to the target, for various output conversion methods.

The same errors were calculated for all five target detection algorithms using the flat Earth conversion method. The binarization-based (Fast Finder), Edge detection, two ArUco algorithms and YOLO neural network were tested and the results are presented in Figure 12, Figure 13 and Figure 14. The smallest overall location error was obtained with the Fast Finder algorithm. The edge detection and YOLO algorithms also gave small error, which is a result of no false positives, which will be shown later.

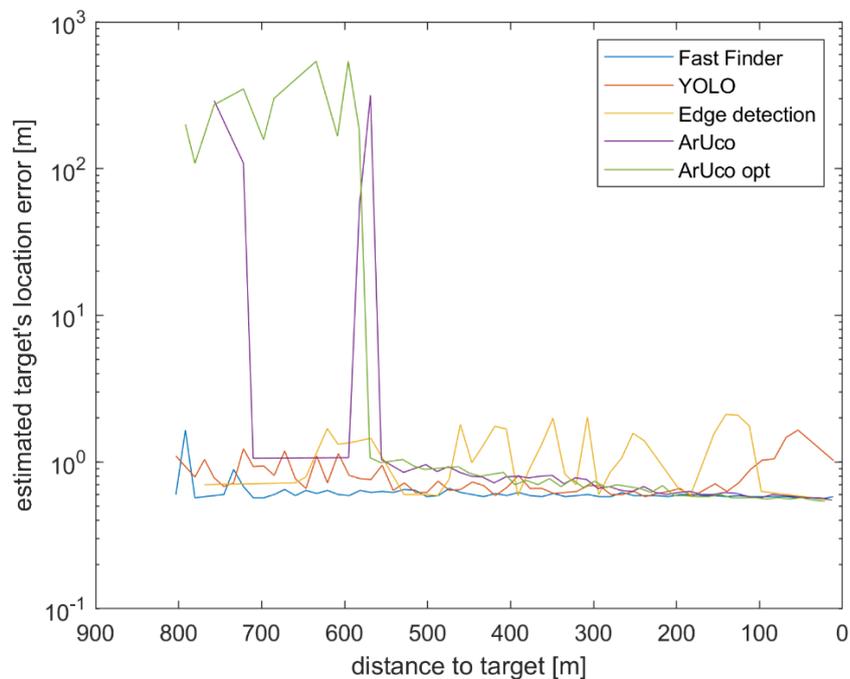


Figure 12: Estimated target's location error as function of distance to the target, for various marker detection algorithms.

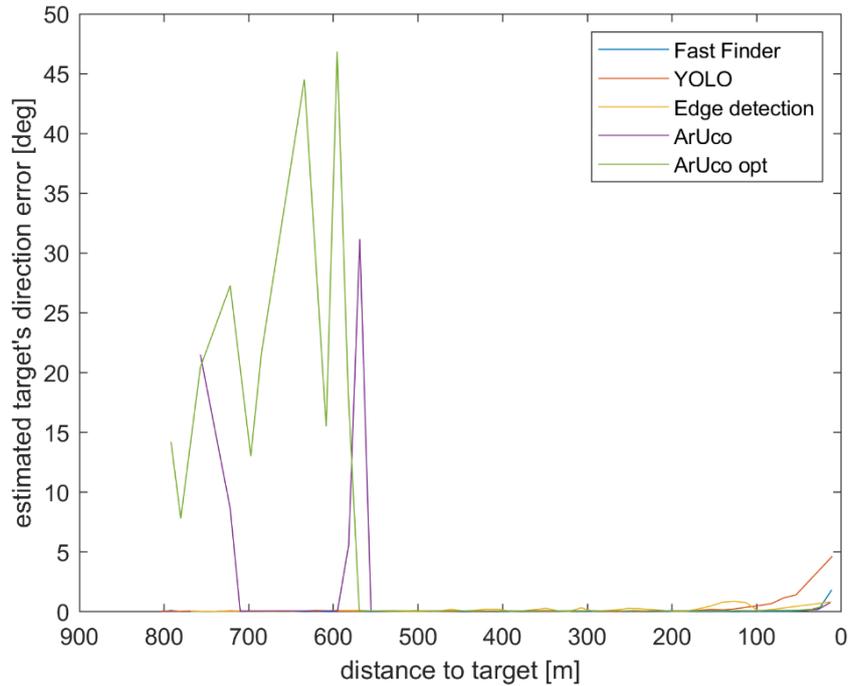


Figure 13: Estimated target's direction error as function of distance to the target, for various marker detection algorithms.

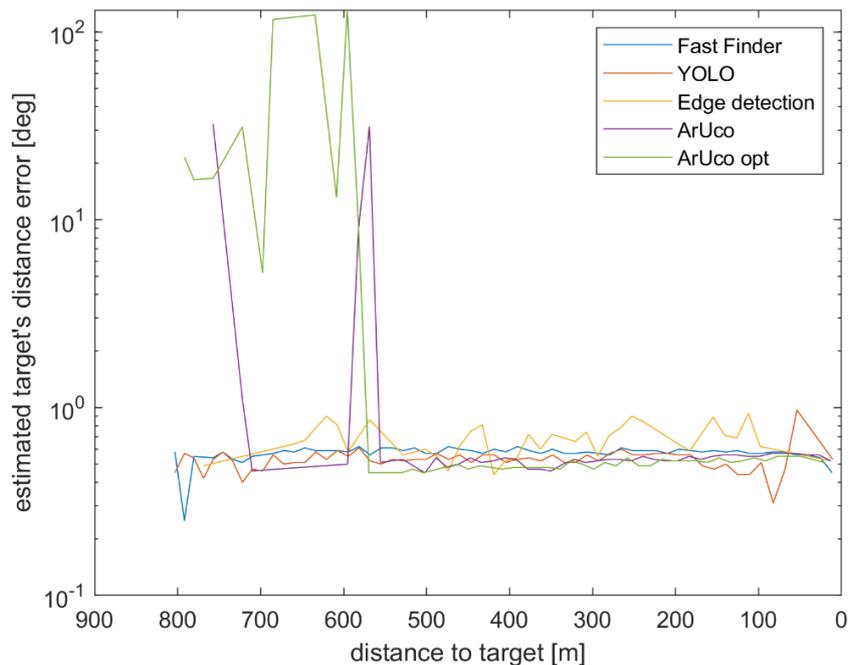


Figure 14: Estimated target's distance error as function of distance to the target, for various marker detection algorithms.

Figure 15 presents correct and incorrect detections as function of distance to the target and Table 1 shows the total correct and incorrect detections for all algorithms. The Fast Finder algorithm and YOLO neural network had the most detections and all of them were correct. Fast Finder detected the marker from all distances simulated, where the YOLO had problems at the final stage of the flight, which could be caused by the training pictures not containing images with such small distances to the marker. The edge detection algorithm also had no false positives and had poor performance in very long and very short distances. The ArUco algorithms detected very well from small distances. The optimized version had a lot of false positive detections from long distances and the standard version had almost no detections there.

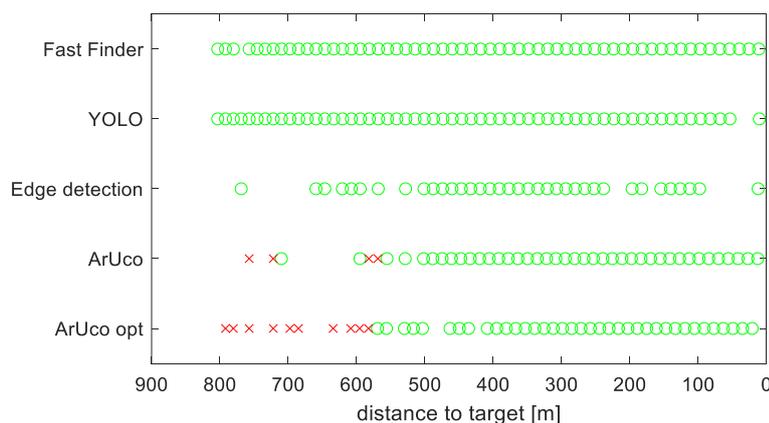


Figure 15: Correct (green circles) and false positive (red crosses) detections as function of distance to the target, for various marker detection algorithms.

Table 1: Total correct and incorrect detections for all algorithms.

	Correct	Incorrect
Fast Finder	59	0
YOLO	58	0
Edge detection	36	0
ArUco	40	4
ArUco opt	37	10

## 7. Conclusions

In this article the proportional navigation guidance was used to guide the missile towards the ground-based target, detected using various image recognition methods. Five methods were prepared: the binarization-based, edge detection based, two version the ArUco algorithm and YOLO neural network. The output of all of these methods was converted to absolute target position using three methods: flat Earth model, ellipsoidal Earth model and camera pose estimation method. The simulation study showed the applicability of the methods to the studied problem. The results were compared using the estimated and true target position errors in terms of direction, distance and overall location errors and the number of positive and false detections. Further research will focus on training YOLO network with images from lower altitudes, and comparison of each method performance in realistic hardware implementation (hardware-in-the-loop tests).

## References

- [1] T. Yamasakia, S. N. Balakrishnanb, H. Takanoa and I. Yamaguchia, "Sliding mode-based intercept guidance with uncertainty and disturbance compensation," *Journal of the Franklin Institute*, vol. 352, pp. 5145 - 5172, 2015.
- [2] Y. Si and S. Song, "Three-dimensional adaptive finite-time guidance law for intercepting maneuvering targets," *Chinese Journal of Aeronautics*, pp. 1-19, June 2017.
- [3] S. He and D. Lin, "Observer-based guidance law against maneuvering targets without line-of-sight angular rate information," *Journal of Aerospace Engineering*, pp. 1-13, 2015.
- [4] S. R. Kumar, S. Rao and D. Ghose, "Nonsingular Terminal Sliding Mode Guidance with Impact Angle Constraints," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 4, pp. 1-17, August 2014.

- [5] X. Chen and J. Wang, "Optimal control based guidance law to control both impact time and impact angle," *Aerospace Science and Technology*, pp. 1-10, 2018.
- [6] F. Shengnan, L. Xiaodong, Z. Wenjie and X. Qunli, "Multiconstraint adaptive three-dimensional guidance law using convex optimization," *Journal of Systems Engineering and Electronics*, vol. 31, no. 4, pp. 791-803, August 2020.
- [7] S. He and C. H. Lee, "Gravity-Turn-Assisted Optimal Guidance Law," *Journal of Guidance, Control, and Dynamics*, pp. 1-13, July 2017.
- [8] L. Zhijun, X. Yuanqing, S. Chun-Yi, D. Jun, F. Jun and H. Wei, "Missile Guidance Law Based on Robust Model Predictive Control Using Neural-Network Optimization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 8, pp. 1803-1809, August 2015.
- [9] J. Zhao, S. Zhou and R. Zhou, "Distributed time-constrained guidance using nonlinear model predictive control," *Nonlinear Dynamics*, pp. 1399-1416, 2016.
- [10] J. Roger, "Stochastic Model Predictive Control for Guided Projectiles Under Impact Area Constraints," *Journal of Dynamic Systems, Measurement and Control*, vol. 137, pp. 1-8, March 2014.
- [11] P. D. Dharmendrabhai, A. Gholap, N. K. Singh and S. Hota, "Impact Angle Constrained Guidance Law for Intercepting Non-maneuvering Targets Avoiding Obstacles," in *AIAA SciTech Forum*, San Diego, 2022.
- [12] A. Budiyo and H. Rachman, "Proportional Guidance and CDM Control Synthesis for a Short-Range Homing Surface-to-Air Missile," *Journal of Aerospace Engineering*, vol. 25, pp. 168 - 177, April 2012.
- [13] M. Jacewicz, P. Lichota, D. Miedziński and R. Głębocki, "Study of Model Uncertainties Influence on the Impact Point Dispersion for a Gasodynamically Controlled Projectile," *Sensors*, vol. 22, pp. 1-20, April 2022.
- [14] Y. Guo, X. Li, H. Zhang, M. Cai and F. He, "Data-Driven Method for Impact Time Control Based on Proportional Navigation Guidance," *Journal of Guidance, Control, and Dynamics*, pp. 1-12, February 2020.
- [15] D. Miedziński, M. Sochacki, K. Bresler, S. Małecki, A. Mochol and K. Wojciechowski, "Low cost rocket guidance and control development platform," in *71st International Astronautical Congress 2020*, Paryż, 2020.
- [16] D. Miedziński, K. Bresler, S. Małecki, P. Umiński, K. Wojciechowski, A. Bakhmat, M. Matak, M. Hałoń and M. Sochacki, "Preliminary design of a Homing Rocket using Image Recognition," in *72nd International Astronautical Congress*, Dubai, 2021.
- [17] P. H. Zipfel, *Modeling and simulation of aerospace vehicle dynamics*, Reston, Va: American Institute of Aeronautics and Astronautics, 2007.
- [18] P. Zarchan, *Tactical and Strategic Missile Guidance*, Sixth Edition, Reston, Va: American Institute of Aeronautics and Astronautics, 2012.
- [19] F. Romero-Ramirez, R. Muñoz-Salinas and R. Medina-Carnicer, "Speeded up detection of squared fiducial markers," *Image and Vision Computing*, vol. 76, pp. 38-47, 2018.
- [20] Garrido-Jurado, R. Muñoz Salinas, F. Madrid-Cuevas and R. Medina-Carnicer, "Generation of fiducial marker dictionaries using mixed integer linear programming," *Pattern Recognition*:51, pp. 481-491, 2016.
- [21] "OpenCV: Detection of ArUco Markers," [Online]. Available: [https://docs.opencv.org/4.x/d5/dac/tutorial\\_aruco\\_detection.html](https://docs.opencv.org/4.x/d5/dac/tutorial_aruco_detection.html).
- [22] M. F. Sani and G. Karimian, "Automatic navigation and landing of an indoor AR. drone quadrotor using ArUco marker and inertial sensors," *2017 International Conference on Computer and Drone Applications (IConDA)*, pp. 102-107, 2017.
- [23] A. Marut, K. Wojtowicz and K. Falkowski, "ArUco markers pose estimation in UAV landing aid system," *2019 IEEE 5th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, pp. 261-266, 2019.
- [24] "OpenCV: ArUco Marker Detection," [Online]. Available: [https://docs.opencv.org/3.4/d9/d6a/group\\_\\_aruco.html](https://docs.opencv.org/3.4/d9/d6a/group__aruco.html).
- [25] "Morphological operations on binary images - MATLAB bwmorph," [Online]. Available: <https://www.mathworks.com/help/images/ref/bwmorph.html>.
- [26] "Binarize 2-D grayscale image by thresholding - MATLAB," [Online]. Available: <https://www.mathworks.com/help/images/ref/imbinarize.html>.
- [27] "Find gradient magnitude and direction of 2-D image - MATLAB," [Online]. Available: <https://www.mathworks.com/help/images/ref/imgradient.html>.
- [28] O. Russakovsky, J. Deng and H. Su, "ImageNet Large Scale Visual Recognition Challenge," *Int J Comput Vis* 115, p. 211-252, 2015.

- [29] “ImageNet Large Scale Visual Recognition Challenge (ILSVRC),” [Online]. Available: <https://www.image-net.org/challenges/LSVRC/>.
- [30] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779-788, 2016.
- [31] G. Jocher, A. Chaurasia, A. B. J. Stoken, Y. Kwon, T. Jiacong Fang, imyhxy, K. Michael, Lorna, A. V, D. Montes, J. L. t. y. S. P. W. Z. Nadar and M. Minh, “ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference,” *Zenodo*, 2022.
- [32] R. Shaoqing, H. Kaiming, G. Ross and S. Jian, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 2015.