Czech Technical University in Prague

Faculty of Electrical Engineering
Department of Cybernetics
Multi-robot Systems

Position estimation of a flying target from a camera onboard a UAV using visual tracking

Master's Thesis

Morhunenko Mykola

Prague, May 2024

Study programme: Cybernetics and Robotics
Supervisor: Ing. Matouš Vrba

Acknowledgments

I would like to thank my supervisor, Ing. Matouš Vrba, and all CTU MRS Group members for their help and advice. They all were kind, helped me a lot with advice and shared their experience.

I especially want to thank all the defenders of my Motherland, The Armed Forces of Ukraine, who are protecting the whole of Europe every day from Russian invaders at the cost of their own lives to make it possible for all of us to live in peace and for me to write this thesis.

Finally, I would like to thank my family who supported me during my whole life: my mother, Svitlana and my father, Roman.

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Morhunenko Mykola**	Personal ID number:	**504482**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Cybernetics**		
Study program:	**Cybernetics and Robotics**		

II. Master’s thesis details

Master’s thesis title in English:

Position Estimation of a Flying Target from a Camera Onboard a UAV Using Visual Tracking

Master’s thesis title in Czech:

Odhadování polohy létajícího cíle z palubní kamery UAV pomocí vizuálního sledování

Guidelines:

1. Implement a system for visual tracking of a target given its initial position in an onboard camera image.
2. Use the tracking results to estimate the target's 3D position in the environment.
3. Test and compare different methods of 3D position estimation based on the geometrical intersection of rays, Kalman Filter, and optimization.
4. Evaluate the localization accuracy in relation to the tracking precision and the uncertainty of the pose of the UAV carrying the camera.

Bibliography / sources:

[1] Matouš Vrba, Viktor Walter, Václav Pritzl, Michal Pliska, Tomáš Báča, Vojtěch Spurný, Daniel Heřt and Martin Saska, „On Onboard LiDAR-based Flying Object Detection,“ arXiv preprint cs.RO 2303.05404, 2023.
[2] J. Li, Z. Ning, S. He, C. -H. Lee and S. Zhao, "Three-Dimensional Bearing-Only Target Following via Observability-Enhanced Helical Guidance," Transactions on Robotics, vol. 39, no. 2, pp. 1509-1526, April 2023.
[3] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 34(7):1409–1422, 2012.
[4] Tobias Nägeli, Samuel Oberholzer, Silvan Plüss, Javier Alonso-Mora, and Otmar Hilliges. 2018. Flycon: real-time environment-independent multi-view human pose estimation with aerial vehicles. ACM Trans. Graph. 37, 6,2018.

Name and workplace of master’s thesis supervisor:

Ing. Matouš Vrba Multi-robot Systems FEE

Name and workplace of second master’s thesis supervisor or consultant:

Date of master’s thesis assignment: **12.01.2024** Deadline for master's thesis submission: **24.05.2024**

Assignment valid until: **21.09.2025**

Ing. Matouš Vrba Supervisor’s signature	prof. Dr. Ing. Jan Kybic Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the master’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master’s thesis, the author must state the names of consultants and include a list of references.

.

Date of assignment receipt Student’s signature

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Grammarly[1] was used to correct grammar and punctuation.

Date ....

[1] https://www.grammarly.com/

Abstract

The problem of position estimation of a moving target from a flying Micro Aerial Vehicle (MAV) with an onboard monocular camera is tackled in this thesis. Multiple approaches based on the Kalman Filter (KF) were implemented and compared to a baseline method formulated as an optimization problem based on a geometric intersection of lines. The bearing-based uniform Pseudo Linear Kalman Filter (uPLKF) and the Degenerate Kalman Filter (DKF) methods were implemented and compared along with their extensions utilizing information about the size of the object in the image to improve range estimation. The considered methods rely on the results of visual tracking of the target and the observer's self-localization as input and provide the position estimation of the target observed by the camera. The implemented methods were tested and compared both in simulations and on real-world data. The influence of uncertainty of the observer's self-localization and the visual tracking on the resulting 3D position estimation is analyzed and discussed. The comparison showed better performance of KF-based methods with the size estimation over bearing-only approaches and the baseline geometric intersection of lines approach. Furthermore, the influence of the observer's self-localization accuracy and visual tracking accuracy on the estimation were evaluated.

Keywords Unmanned Aerial Vehicles, Kalman Filter, 3D Object Localization

Abstrakt

V této práci je řešen problém odhadu polohy pohybujícího se cíle z dat palubní monokulární kamery umístěné na bezpilotním letounu. Několik přístupů, založených na Kálmánově Filtru (KF), je implementováno a porovnáno s referenční metodou, která je formulována jako optimalizační úloha založená na geometrickém průsečíku přímek. Konkrétně jsou testovány metody uniform Pseudo Linear Kalman Filter (uPLKF) a Degenerate Kalman Filter (DKF), jejichž vstupem jsou směrové vektory cíle měřené palubní kamerou, a také rozšíření těchto metod využívající rozměr cíle v obraze ke zpřesnění odhadu vzdálenosti cíle. Srovnávané metody využívají výstup vizuálního sledování cíle a sebelokalizaci pozorovatele a poskytují odhad polohy sledovaného cíle. Implementované metody jsou testovány v simulacích a na reálných datech se zaměřením na vyhodnocení vlivu nepřesností vizuálního sledování a sebelokalizace pozorovatele na výsledný odhad polohy. Metody založené na KF a využívající rozměr cíle v obraze vykazují přesnější výsledky a vyšší odolnost vůči nepřesnostem vstupních dat než přístupy založené pouze na směrových vektorech a výrazné zlepšení oproti referenční geometrické metodě.

Klíčová slova Bezpilotní letounu, Kalmanův Filtr, Lokalizace objektů ve 3D

Abbreviations

FOV Field of View

KF Kalman Filter

RGB Red Green Blue

PLKF Pseudo Linear Kalman Filter

LKF Linear Kalman Filter

UKF Unscented Kalman Filter

EKF Extended Kalman Filter

DKF Degenerate Kalman Filter

uPLKF uniform Pseudo Linear Kalman Filter

LiDAR Light Detection and Ranging

LTI Linear Time-Invariant

UAV Unmanned Aerial Vehicle

MAV Micro Aerial Vehicle

MRS Multi Robot Systems

ROS Robot Operating System

SVD Singular Value Decomposition

RTK Real-Time Kinematic

RMSE Root Mean Square Error

Contents

Chapter 1

Introduction

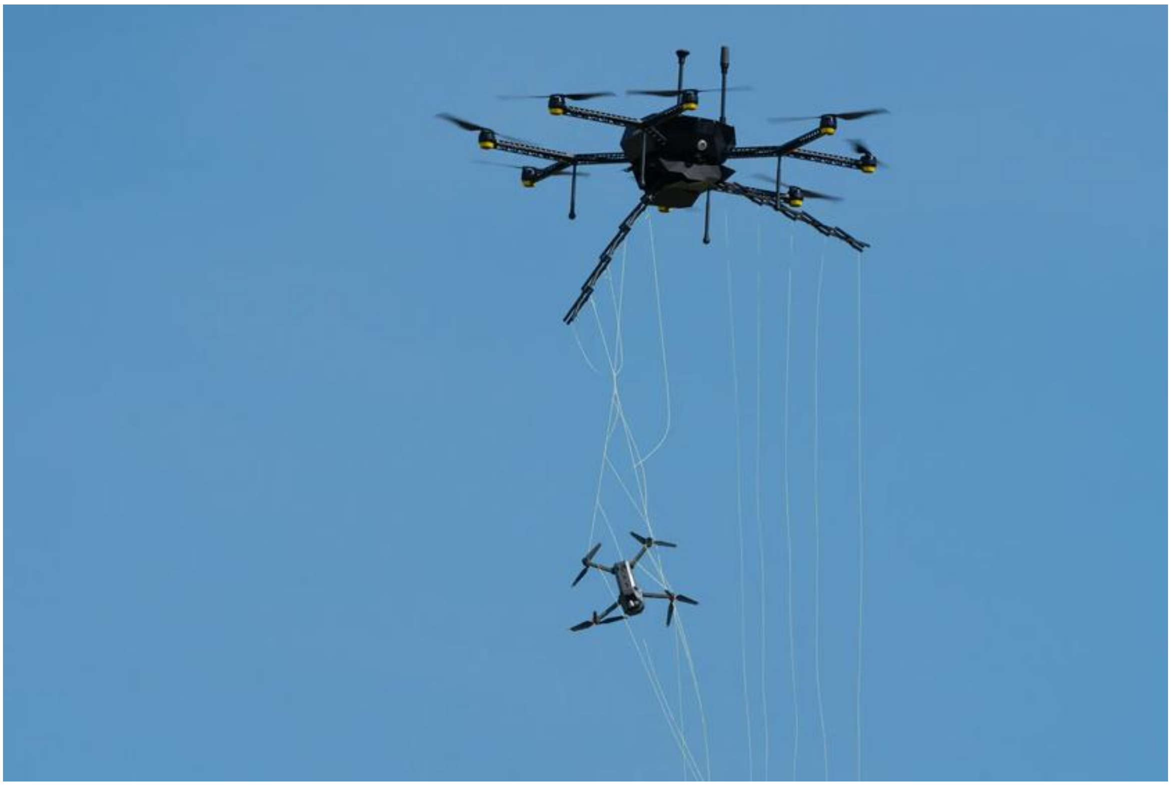


Figure 1.1: Eagle.One MAV, image taken from [6].

MAVs, also referred to as drones, have emerged as unique and indispensable tools across the world in various fields, including disaster management, water surface monitoring, [4], firefighting [14], medicines and blood delivery [8], search and rescue [5], entertainment [13], agriculture [10], scientific research [11] and many more. The simplicity and availability of MAVs make them a suitable platform for research and testing different hypotheses. The size varies from a few centimeters and a few grams (Intel Tello) to 40 m and 6700 kg (Global

Hawk).

While some MAVs are used to improve and even save lives, their widespread adoption has also raised concerns regarding safety, security, and privacy. Unauthorized and malicious MAVs, commonly referred to as rogue drones, have been implicated in incidents ranging from airspace violations to espionage and terrorism.

For example, in 2018 in New Zealand, a MAV almost caused the crash of a Boeing 777 by flying within 5 m of it[1]. In the same year, "two GPS-guided drones laden with explosives were used in a failed attempt to assassinate Venezuelan President Maduro"[2], which can be classified as a terrorist attack. The MAVs were carrying explosives, and they could cause people harm or even death. United Nations published an 80-page "practice guide"[3] about "Protecting vulnerable targets from terrorist attacks involving unmanned aircraft systems".

To protect private property, critical government infrastructure, etc., various intercepting devices were developed. There are anti-drone guns that launch strong, narrow, directed electromagnetic pulses towards the MAV line, jamming and spoofing devices, and different military electronic warfare systems. There are also big drones-interceptors, for example, the USA Air Force Research Laboratory (AFRL) interceptor drone[4] and the Czech startup Eagle.One[5]. Using the set of onboard sensors, the Eagle.One detects the target, finds the target's position, flies towards it, and catches it with a net. The Eagle.One interceptor has an onboard Light Detection and Ranging (LiDAR) sensor from which it estimates the position and velocity of a desired target, but when the target is out of the Field of View (FOV) of LiDAR, it loses the track. The problem can be solved by adding more LiDARs (which are costly), stereo cameras, which often have a limited range of view, or monocular cameras, which are relatively cheap but require intelligent algorithms to estimate the depth from the monocular camera and find the distance to the target.

Formations and swarms of drones are other fields where the localization of a moving object from a flying MAV is important. In [16], the visual relative localization for swarms is proposed. Authors use the Convolutional Neural Network (CNN) to estimate the distance between the observer and multiple targets in an image taken by the onboard camera. Usually CNNs are more computationally demanding than the standard approaches, for example, KF-based, so the KF-based approach probably can be used for relative localization in swarms.

The aim of this thesis is to implement and compare the position estimation algorithms. The input provided to those methods is the output of the target's visual tracking from the camera onboard the MAV called observer and the output of the observer's self-localization. The geometric intersection of lines in 3D was chosen as a baseline solution. Thanks to the recent advancements in the field of bearing-based localization, multiple KF-based methods were implemented and tested in order to improve the results of the baseline method. All approaches are compared in different simulation scenarios and tested on real-world data. The influence of the deviations in visual tracking and the observer's self-localization imprecision

[1] BBC: https://www.bbc.com/news/world-asia-43551373.

[2] Crisis-response: https://www.crisis-response.com/Articles/682409/Use_of_drones.aspx.

[3] UN counter-terrorism: https://www.un.org/counterterrorism/sites/www.un.org.counterterrorism/files/2118451e-vt-mod5-unmanned_aircraft_systems_final-web.pdf.

[4] San.com: https://san.com/cc/afrl-drone-interceptor-is-both-offensive-and-defensive-weapon-of-the-week/

[5] Eagle.One: https://eagle.one/cs/

is tested and evaluated. Because visual object tracking on a sequence of images is a widely researched and mostly solved problem, the thesis mainly focused on the bearing-based target's state estimation, as this was identified as an area with a high potential for innovation.

1.1 Related works

Visual tracking

Visual tracking refers to the process of following an object on video frames or a set of consecutive images. Visual tracking is a complicated problem due to object deformation, illumination changes, camera motion, and occlusion. Some trackers, like Kanade-Lucas-Tomasi (KLT) [27], are easy to understand and implement for simple tasks. The KLT is a feature tracker, and it assumes the displacement of two consecutive frames is small and approximately constant within the neighborhood of the feature points. Then, the sum of the squared differences of image intensities is minimized iteratively. The feature point is removed from the further tracking if the algorithm does not converge.

Others, like FEAR Tracker [12] or STARK tracker [15], rely on machine learning approaches. They are more power- and computation-demanding but more efficient and robust. Visual trackers usually provide only information in the image plane, without depth information. That is where the 3D position estimation (or depth estimation) as a separately formulated and solved problem is useful.

Some open-source libraries, like OpenCV[6] provide the implementation of robust state-of-the-art trackers that can be used out-of-the-box, for example, MOSSE Tracker [21], The Median Flow tracker [22], TLD Tracker [20] and many more.

Depth estimate

An image taken by a monocular camera is usually a two-dimensional projection of a three-dimensional scene onto the image plane. As a result of such projection, the information about the depth of the observed objects is lost. However, this information can be useful, and there are methods to get it. The task of retrieving depth information is often called "depth estimation", and it is a wide-researched topic. Most of the proposed approaches require some prior knowledge about some elements of the scene. The vast majority of those approaches using the monocular camera are based on CNNs using a single image, with no sequence required. In [9], multiple methods of monocular depth estimation are compared, and all of them are based on some Neural Networks. However, those approaches are very well generalized to estimate the distance to any object in the image scene with some precision. For the specific task when the prior knowledge is provided, like the region of interest in an image, amount of important objects in an image scene and the dynamics of an object, better approaches can be applied, for example, the KF-based, utilizing the prior knowledge about the moving target and its dynamics and the dynamics.

Another similar problem would be the "structure from motion". This is the task of estimating a 3D scene from a set of monocular images. In [19], an overview of the problem is presented. The main idea is to search for common features in the sequence of images, match

[6] OpenCV: https://opencv.org/

them, estimate the camera shift, and reconstruct the 3D scene. However, prior knowledge is still required to compute the scale of the scene. If the camera is mounted on a drone, such information could be the odometry from an onboard Inertial Measurement Unit (IMU) or Global Navigation Satellite System (GNSS) so the scene can be reconstructed. Another disadvantage of this method is that it only applies to static objects and is not suitable for moving objects, which is necessary for the task tackled in this thesis.

Bearing-based 3D position estimation

The state estimation of a flying target from a MAV is an important task for tracking, following, avoiding, or intercepting the target. The MAV equipped with a LiDAR can estimate the distance and bearing to the target, which determines the relative position. Using additional intelligent algorithms, a target's position, velocity, and acceleration can be estimated for further interception, as described in [2].

If the LiDAR or other sources of distance measurement are not available, the position of a target can still be estimated using bearings obtained e.g. from a monocular camera. It involves determining the bearing from the camera to the target, usually represented as angles to the reference direction in spherical coordinates or simply a vector. To track the object using bearings, the dynamics of a target must be of a lesser order than observer [28]. In other words, if the acceleration of a target is to be estimated, the observer must have a known non-linear jerk. For position estimation, non-zero velocity is enough for the system to be observable.

The general idea and principles of bearing angles and their usage are described in [24], where the bearing-angles Simultaneous Localization and Mapping for a ground vehicle is tackled. By collecting and combining the bearing angles with the key objects from the observer's different poses, it is possible to retrieve the position of the observer using triangulation, optimization, or other approaches. Typically, the KF is used to fuse the data from different sensors or multiple observing points and improve the estimation.

The Pseudo Linear Kalman Filter (PLKF) is a non-linear KF which, instead of direct linearisation using Jacobian for Taylor approximation as for Extended Kalman Filter (EKF), reformulates the nonlinearities or employs the transformations to yield a linear-like structure. A two-dimensional case of bearing-only target tracking is proposed in [13]. The authors propose the “Unbiased Pseudo Linear Kalman Filter” and compare it to other PLKFs. Since bearing angle measurements are not linear, the simple Linear Kalman Filter (LKF) can not be used here. The EKF is also unsuitable, as shown in [25] because it uses the Taylor approximation for linearisation, which is not stable for functions that are hard to derivate. In contrast to LKF and EKF, the PLKF shows good results, so it is widely used in this task.

Multiple approaches to bearing-only 3D tracking are compared in [13]. All methods described in the article are the modifications of the PLKF that use angles, not vectors, to define the bearing to the target. In the end, all methods are compared in numerical simulation. All methods showed similar performance and proved to be suitable for the bearing-only position estimation.

In this thesis, the approach from [3] is adopted. In the original paper, not bearing angles but bearing vectors are used to simplify computations. Also, the optimal trajectory for the pursuing MAV to better estimate the position of the target is provided and proved.

1.2 Problem statement

A MAV called observer with an onboard PC running ROS, utilizing Real-Time Kinematic (RTK) sensor for self-localization is tasked with the position estimation of a target in 3D space observed using an equipped monocular camera. Visual tracking of a target is provided. The MAV relies on initial position and velocity data given, for instance, by a LiDAR detection and estimation algorithm from [6]. The target is moving in a 3D space with linear velocity and, when viewed from different perspectives, is assumed to be invariant in width. The challenge lies in developing a robust system that can accurately estimate the target's movement in real time despite the limitations of monocular vision and the dynamic environment in which the UAV operates. This involves addressing issues such as ensuring continuous and accurate target localization and maintaining stable tracking performance under varying conditions and viewpoints from a moving camera.

1.3 Mathematical notation

The mathematical notation used in this thesis is defined in Table 1.1.

$\boldsymbol{x}, \boldsymbol{\eta}$	vector or pseudo-vector
P	a point
$\mathbf{X}, \mathbf{\Omega}$	matrix
$\mathbf{I}_{n\times m}$	identity matrix of size n by m
$\mathbf{0}_{n\times m}$	matrix of zeros of size n by m
$x_n = \mathbf{x}^\intercal \hat{\mathbf{e}}_n$	nth vector element (row), $\mathbf{x}, \mathbf{e} \in \mathbb{R}^3$
$\mathbf{X}_{(a,b)}$	matrix element, (row, column)
$\dot{x}, \ddot{x}$	1st and 2nd time derivative of x
$x_{[k]}$	x at time instant k
$\mathbf{A}, \mathbf{B}, \mathbf{x}$	LTI system matrix, input matrix and input vector
$SO(2)$	2D special orthogonal group of rotations
$SO(3)$	3D special orthogonal group of rotations

Table 1.1: Mathematical notation, nomenclature, and notable symbols.

Chapter 2

Theoretical background

2.1 Kalman Filter

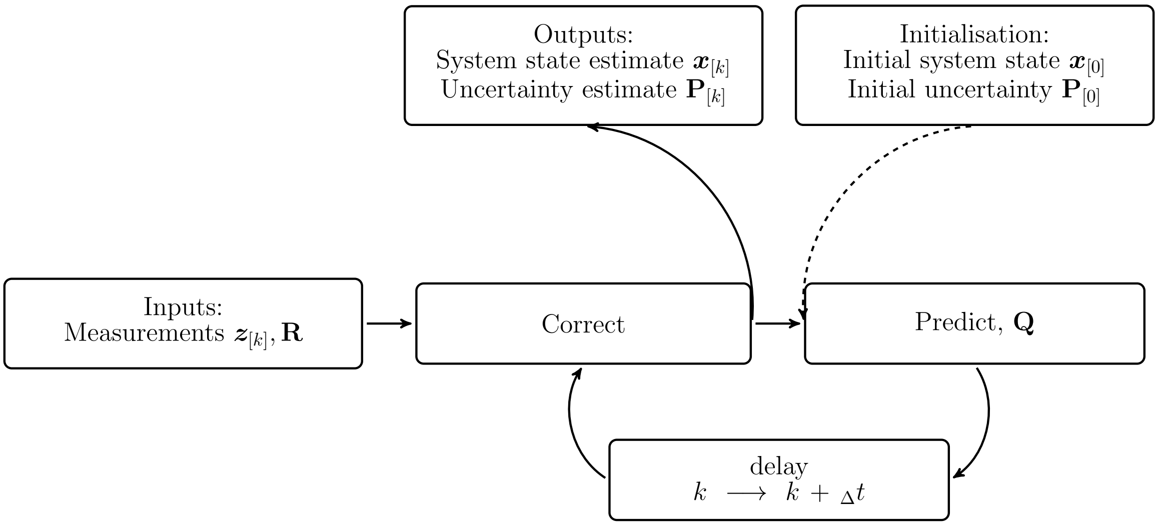

Outputs:
System state estimate $\boldsymbol{x}_{[k]}$
Uncertainty estimate $\mathbf{P}_{[k]}$
Initialisation:
Initial system state $\boldsymbol{x}_{[0]}$
Initial uncertainty $\mathbf{P}_{[0]}$
Inputs:
Measurements $\boldsymbol{z}_{[k]}, \mathbf{R}$
Correct
Predict, $\mathbf{Q}$
delay
$k \longrightarrow k + {}_{\Delta}t$

Figure 2.1: The scheme of a KF iteration. The initialization is done only once, at the very beginning.

The general stochastic state-space Linear Time-Invariant (LTI) system model used for the KF is described with two equations. The first one is a state transition equation, and the second one is the measurement model:

$$\begin{aligned} \boldsymbol{x}_{[k+1]} &= \mathbf{A}\boldsymbol{x}_{[k]} + \mathbf{B}\boldsymbol{u}_{[k]} + \boldsymbol{w}_{[k]}, \\ \boldsymbol{z}_{[k]} &= \mathbf{H}\boldsymbol{x}_{[k]} + \boldsymbol{e}_{[k]}, \end{aligned} \tag{2.1}$$

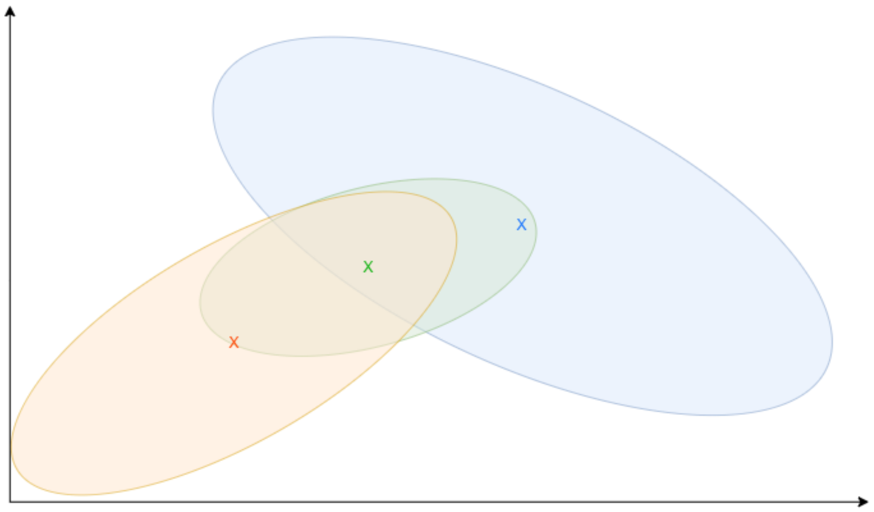


Figure 2.2: An example of a single KF iteration. The blue ellipse illustrates the initial estimate $\boldsymbol{x}_{[k|k-1]}$ with covariance $\mathbf{P}_{[k|k-1]}$, the green ellipse the estimate after the correction $\boldsymbol{x}_{[k|k]}$ with covariance $\mathbf{P}_{[k|k]}$ utilizing the current measurement, and the orange ellipse the predicted estimate $\boldsymbol{x}_{[k+1|k]}$ with covariance $\mathbf{P}_{[k+1|k]}$.

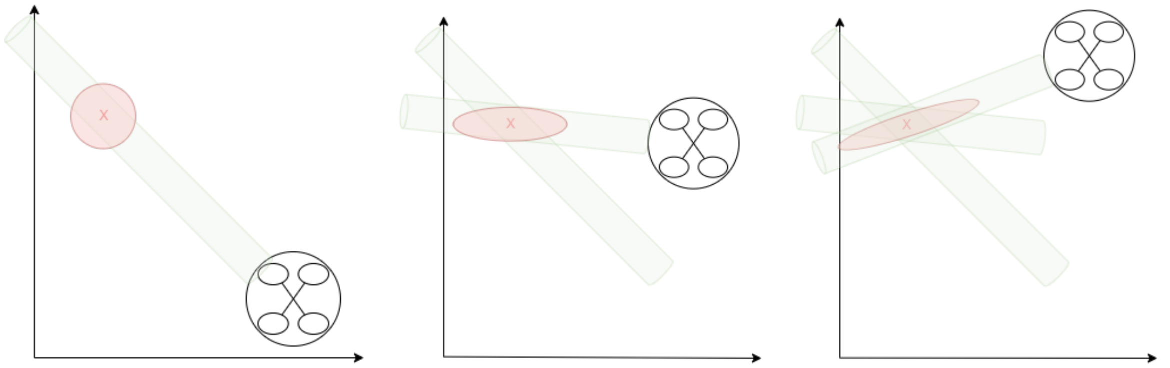
where the deterministic part is

$$\begin{aligned} \boldsymbol{x}_{[k]} &\in \mathbb{R}^{n} \text{ is the state of the system at time instant } k, \\ \boldsymbol{z}_{[k]} &\in \mathbb{R}^{p} \text{ is the observed output of the system at time instant } k, \\ \boldsymbol{u}_{[k]} &\in \mathbb{R}^{m} \text{ is the input at time instant } k, \\ \mathbf{A} &\in \mathbb{R}^{m \times n} \text{ is the state transition matrix,} \\ \mathbf{B} &\in \mathbb{R}^{n \times m} \text{ is the control matrix,} \\ \mathbf{H} &\in \mathbb{R}^{p \times n} \text{ is the measurement-state mapping,} \end{aligned}$$

and the stochastic part is

$$\begin{aligned} \boldsymbol{w}_{[k]} &\sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \text{ is the process noise,} \\ \boldsymbol{e}_{[k]} &\sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \text{ is the measurement noise,} \end{aligned}$$

which is assumed to follow a Gaussian random distribution with zero mean and covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ for any $\{k, n, m, p\}, \in \mathbb{N}$.



(a) $k = 0$, initialisation.

(b) $k = 1$, prediction and correction.

(c) $k = 2$, prediction and correction.

Figure 2.3: An illustration of bearing-based position estimation using KF-based approaches, top view. The red cross is the estimated position of the target, the red ellipse is the covariance, and the green cylinder is a bearing vector with radial uncertainty. The observer moves forward with a constant velocity while the target is static. The initialization is shown in subfigure 2.3a. The result of the next prediction and correction steps after $k = 1$ producing a slightly skewed covariance matrix is in subfigure 2.3b. The result after $k = 2$ timestamp is in subfigure 2.3c.

The general idea behind the KF is to estimate and predict the state $\boldsymbol{x}_{[k+1|k]}$ of a noisy system, and its uncertainty $\mathbf{P}_{[k+1|k]}$, which is also called the state covariance, which can be done even if the system itself is unknown. This filter is a powerful tool used in navigation [23], tracking [18], [26], and control [7], [17], but overall, it can be applied to dynamic systems with noise. The LKF is optimal and stable, but it can be used only with the LTI systems under the assumption of the system and measurement noise being Gaussian. Unfortunately, in the real world, systems are often non-linear, so other modifications of the KF like EKF or the Unscented Kalman Filter (UKF) are used. They are not guaranteed to be optimal and stable, but they are still chosen by many engineers in robotics (and other fields). One of the biggest advantages of KF is its ability to process the data in real-time, while for filters like median filter, mean filter, or moving average, a window of some size is needed, so the result is always delayed. Another advantage is that despite the fact that KF results and estimates for non-LTI systems are non-optimal, under good initialization and after tuning, they can offer very good performance.

The KF is a two-step recursive filter that both estimates and, given the measurement, predicts the future development of the system using the model. It operates with the above-defined state-space model of a dynamic system. In the prediction step, the state of the model with its uncertainty is propagated through the KF to the next time step. In the correction step, the state with its uncertainty is corrected with respect to the incoming measurement data. Figure 2.1 shows the general KF.

For the state-space model defined in (2.1), the correction of the KF is done in two steps. First, calculate the Kalman gain $\mathbf{K}_{[k]}$ is calculated given $\mathbf{P}_{[k|k-1]}$ and the system model

$$\mathbf{K}_{[k]} = \mathbf{P}_{[k|k-1]}\mathbf{H}^{\intercal}(\mathbf{H}\mathbf{P}_{[k|k-1]}\mathbf{H}^{\intercal} + \mathbf{R})^{-1}. \tag{2.2}$$

Then, the state and covariance are updated given $\boldsymbol{x}_{[k|k-1]}$, $\mathbf{P}_{[k|k-1]}$, $\mathbf{K}_{[k]}$, $\boldsymbol{z}_{[k]}$ as

$$\begin{aligned} \boldsymbol{x}_{[k|k]} &= \boldsymbol{x}_{[k|k-1]} + \mathbf{K}_{[k]}(\boldsymbol{z}_{[k]} - \mathbf{H}\mathbf{x}_{[k|k-1]}), \\ \mathbf{P}_{[k|k]} &= \mathbf{P}_{[k|k-1]} - \mathbf{K}_{[k]}\mathbf{H}\mathbf{P}_{[k|k-1]}. \end{aligned} \tag{2.3}$$

The prediction is done as

$$\begin{aligned} \boldsymbol{x}_{[k+1|k]} &= \mathbf{A}\boldsymbol{x}_{[k|k]} + \mathbf{B}\boldsymbol{u}_{[k]}, \\ \mathbf{P}_{[k+1|k]} &= \mathbf{A}\mathbf{P}_{[k|k]}\mathbf{A}^{\intercal} + \mathbf{Q}. \end{aligned} \tag{2.4}$$

In Figure 2.2, an example of a single iteration of a KF is displayed. The lower index in $\boldsymbol{x}_{[f|j]}$ means the best value for $\boldsymbol{x}$ at a time instant f using all data available up and including time instant j, so the $\boldsymbol{x}_{[k|k]}$ is an update of $\boldsymbol{x}$ at time instant k considering the additional input received at time k, and $\boldsymbol{x}_{[k+1|k]}$ is the update of the vector $\boldsymbol{x}$ for time instant $k+1$ considering only the information available at time instant k.

Figure 2.3 illustrates the expected behavior of the KF for the bearing-based position estimation, the initialization, and two update steps. A detailed explanation of how it can be achieved in practice is in chapter 3.

Chapter 3

Methods

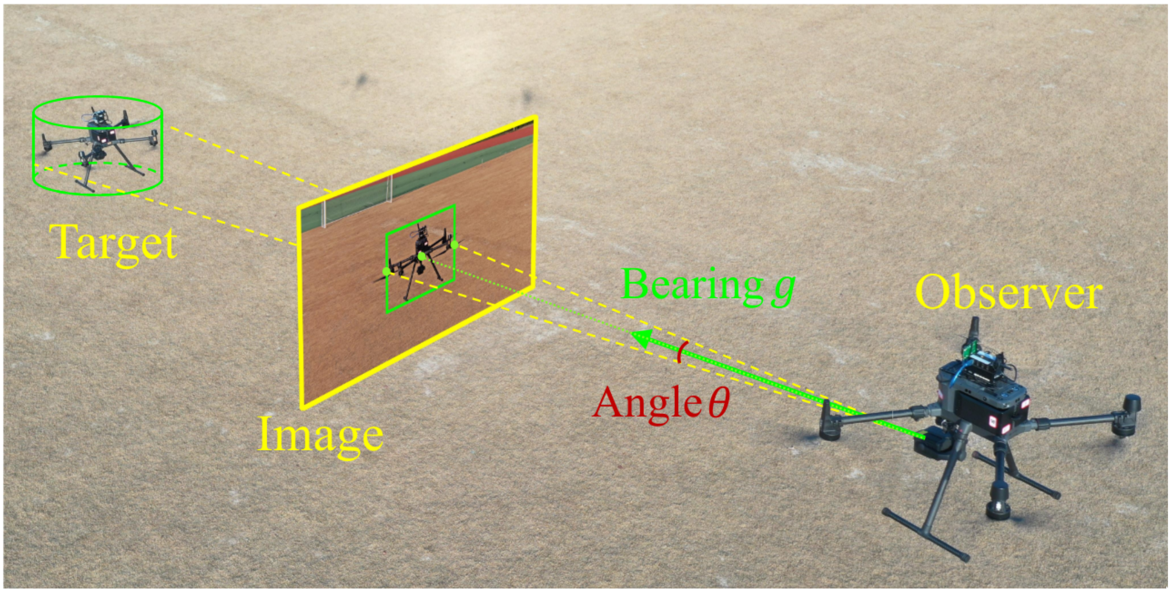


Figure 3.1: An example relative position estimation in side view. The observer (interceptor) measures the target bearing vector $\boldsymbol{g}$ and the subtended angle θ. Image taken from [1].

In this chapter, the MAV model is defined, and the approaches used in the thesis are described. In section 3.2, the baseline approach formulated as the geometric intersection of rays in 3D is introduced. One of the biggest disadvantages of this approach is that it is very sensitive to noise and model mismatch. Another disadvantage is the lack of tunable parameters to adjust the performance.

In section 3.3, approaches based on bearing-angle KF are described. Their advantages include the possibility of tuning them using the measurement and state covariance matrices. However, KF-based methods require prior knowledge about the dynamics of the tracked object and a good initialization. The improved version of the KF-based methods is presented in section 3.4, with the size of the tracked bounding box incorporated in the state to estimate the real size of the object and improve the target localization quality.

3.1 MAV model

The state vector $\boldsymbol{x}$ describing the position $\boldsymbol{p} = \begin{bmatrix} x & y & z \end{bmatrix}^\mathsf{T}$ and velocity $\boldsymbol{p} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^\mathsf{T}$ of a moving object in 3D is defined as

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{v} \end{bmatrix}, \ \boldsymbol{x} \in \mathbb{R}^6, \ \boldsymbol{p} \in \mathbb{R}^3, \ \boldsymbol{v} \in \mathbb{R}^3. \tag{3.1}$$

For this thesis, dynamics of the MAV is modeled as a mass-point using the following mathematical formulation:

$$\begin{aligned} \boldsymbol{p}_{[k+1]} &= \boldsymbol{p}_{[k]} + \vartriangle t \boldsymbol{v}_{[k]} + \frac{\vartriangle t^2}{2} \boldsymbol{a}_{[k]}, \\ \boldsymbol{v}_{[k+1]} &= \boldsymbol{v}_{[k]} + \vartriangle t \boldsymbol{a}_{[k]}, \\ \boldsymbol{a}_{[k]} &\sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_a), \end{aligned} \tag{3.2}$$

where the unknown acceleration is modeled as a Gaussian random variable with zero mean and covariance $\mathbf{Q}_a$. The state transition equation (2.1) is then simplified to

$$\boldsymbol{x}_{[k+1]} = \mathbf{A}\boldsymbol{x}_{[k]} + \boldsymbol{w}_{[k]}, \tag{3.3}$$

where $\boldsymbol{w}_{[k]} = \mathbf{B}\boldsymbol{a}_{[k]}$, so the process noise is modelled as $\boldsymbol{w}_k \sim \mathcal{N}(0, \mathbf{B}\mathbf{Q}_a\mathbf{B}^\mathsf{T})$. Matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{Q}$ for this particular case are

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_{3\times3} & \vartriangle t\mathbf{I}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \end{bmatrix}, \tag{3.4}$$

$$\mathbf{B} = \begin{bmatrix} \frac{\vartriangle t^2}{2}\mathbf{I}_{3\times3} \\ \vartriangle t\mathbf{I}_{3\times3} \end{bmatrix}, \tag{3.5}$$

$$\mathbf{Q} = \alpha\mathbf{I}_{3\times3}, \tag{3.6}$$

where $\mathbf{A} \in \mathbb{R}^{6\times6}$ and $\mathbf{B} \in \mathbb{R}^{6\times3}$, and α is a nonzero tunable variable.

3.2 Geometric lines intersection in 3D

Static target

Any two lines in 3D space can either be parallel, coincident, skew, or intersecting. More lines make the problem of finding their intersection more complicated, but the optimal point minimizing the distance to each line can be found in terms of least squares if lines are not parallel or coincident. Having a set of $n \in \mathbb{N}$ lines defined by points $O_i = \left[O_{i,x}, O_{i,y}, O_{i,z}\right]^\mathsf{T}$

and $P_i = \left[P_{i,x}, P_{i,y}, P_{i,z}\right]^\intercal, \in \mathbb{R}^3$, and any point $L_i = \left[L_{i,x}, L_{i,y}, L_{i,z}\right]$ on a line in 3D is defined as

$$L_i = O_i + (P_i - O_i)\lambda_i,\ i \in [1;\ n] \tag{3.7}$$

which is the same as

$$L_i = O_i + \boldsymbol{d}_i\lambda_i, \tag{3.8}$$

where the $\boldsymbol{d}_i = \left[d_{i,x}, d_{i,y}, d_{i,z}\right]^\intercal = (P_i - O_i)$ is a direction vector, and $\lambda_i \in \mathbb{R}$ is a free parameter. The equation (3.7) is in a vector form, so one line is a set of 3 equations with 4 unknowns, which can be written as

$$\begin{aligned} L_x - O_x - d_x\lambda &= 0 \\ L_y - O_y - d_y\lambda &= 0 \\ L_z - O_z - d_z\lambda &= 0 \end{aligned} \tag{3.9}$$

where L_x, L_y, L_z, λ are unknown. Starting from two lines (six equations and five unknowns), the system of equations is already determined so the solution can be found. For n lines, there are $n+3$ variables and $3n$ equations, and the optimisation problem is formulated as

$$L^* = \arg\min \left\|\mathbf{F}\boldsymbol{x} - \boldsymbol{b}\right\|^2, \tag{3.10}$$

with elements

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 & -d_{1,x} & 0 & \cdots & 0 \\ 0 & 1 & 0 & -d_{1,y} & 0 & \cdots & 0 \\ 0 & 0 & 1 & -d_{1,z} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & 0 & \cdots & 0 & -d_{n,x} \\ 0 & 1 & 0 & 0 & \cdots & 0 & -d_{n,y} \\ 0 & 0 & 1 & 0 & \cdots & 0 & -d_{n,z} \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} O_{1,x} \\ O_{1,y} \\ O_{1,z} \\ \vdots \\ O_{n,x} \\ O_{n,y} \\ O_{n,z} \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} L_x \\ L_y \\ L_z \\ \lambda_1 \\ \vdots \\ \lambda_n \end{bmatrix} \tag{3.11}$$

where $\mathbf{F} \in \mathbb{R}^{3n\times(n+3)}, \boldsymbol{x} \in \mathbb{R}^{n+3}, \boldsymbol{b} \in \mathbb{R}^{3n}, n \in [2, \infty)$. The point $L^* = [L^*_x, L^*_y, L^*_z]^\intercal$ is the closest point to all n lines.

Target moving with a constant velocity

A moving target's velocity can be added to the model to improve estimation. The equation (3.7) is modified as

$$L_i + \boldsymbol{v}{}_\triangle t_i = O_i + (P_i - O_i)\lambda_i, \tag{3.12}$$

where $\boldsymbol{v} = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^\intercal$ is the target's velocity, and ${}_\triangle t_i$ is the time from the first sample, ${}_\triangle t_1 = 0$. The objective is the same - to solve the optimization problem defined in the equation

(3.10), but with elements

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 & \Delta t_1 & 0 & 0 & -d_{1,x} & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \Delta t_1 & 0 & -d_{1,y} & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t_1 & -d_{1,z} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \Delta t_n & 0 & 0 & 0 & \cdots & 0 & -d_{n,x} \\ 0 & 1 & 0 & 0 & \Delta t_n & 0 & 0 & \cdots & 0 & -d_{n,y} \\ 0 & 0 & 1 & 0 & 0 & \Delta t_n & 0 & \cdots & 0 & -d_{n,z} \end{bmatrix}, \ \mathbf{b} = \begin{bmatrix} O_{1,x} \\ O_{1,y} \\ O_{1,z} \\ \vdots \\ O_{n,x} \\ O_{n,y} \\ O_{n,z} \end{bmatrix} \ \boldsymbol{x} = \begin{bmatrix} L_x \\ L_y \\ L_z \\ v_x \\ v_y \\ v_z \\ \lambda_1 \\ \vdots \\ \lambda_n \end{bmatrix}, \tag{3.13}$$

where $\mathbf{F} \in \mathbb{R}^{3n \times (n+6)}, \boldsymbol{x} \in \mathbb{R}^{n+6}, \boldsymbol{b} \in \mathbb{R}^{3n}$, and $n \in [3, \infty)$. Starting from three lines (nine equations and nine unknowns), the system of equations is already determined so the solution can be found. For n lines, $n+6$ variables and $3n$ equations are needed to solve the system.

Solution

Both problems defined in 3.2.1 and 3.2.2 are formulated as least-squares optimization problems. The optimal solution can be computed as

$$\boldsymbol{x} = \mathbf{F}^{-1}\boldsymbol{b}, \tag{3.14}$$

which is not possible if the matrix $\mathbf{F}$ is not invertible. This can be resolved by taking the left pseudo-inverse of matrix $\mathbf{F}$ as

$$\mathbf{F}^{\dagger} = (\mathbf{F}^{\intercal}\mathbf{F})^{-1}\mathbf{F}^{\intercal}; \tag{3.15}$$

where † stands for the pseudo-inverse. A more numerically stable way to compute the pseudo-inverse is using the Singular Value Decomposition (SVD), which can also handle numerically unstable and nearly rank-deficient matrices. It is formulated as

$$\mathbf{F} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\intercal}; \ \mathbf{F}^{\dagger} = \mathbf{U}\boldsymbol{\Sigma}^{-1}\mathbf{V}^{\intercal}, \tag{3.16}$$

where $\mathbf{U}$ and $\mathbf{W}$ are orthogonal matrices and $\boldsymbol{\Sigma}$ is a diagonal matrix with singular values of $\mathbf{F}$ on the diagonal. Regardless of the way in which the pseudo-inverse was found, the optimal solution is computed as

$$\boldsymbol{x} = \mathbf{F}^{\dagger}\boldsymbol{b}. \tag{3.17}$$

3.3 Bearing-only position estimation using Kalman Filters

The bearing vector points from the observer to the target, as shown in Figure 3.1. For the KF-based approaches considering the state vector $\boldsymbol{x}$ as described in section 3.1, the bearing is not a linear measurement but can be utilized to estimate the state as described further in section 3.3.1. Bearing measurement can be linearized and incorporated in EKF, or as was described in the introduction chapter 1, transformed to a pseudo-linear form and used in PLKF.

Uniform Pseudo Linear Kalman Filter

One of the uPLKF approaches used in this thesis is proposed in [3], which is a bearing-only 3D position estimation algorithm based on a KF utilizing the bearing vector instead of bearing angles to simplify the computations and make the algorithm more numerically stable by avoiding trigonometrical functions. The estimated measured state $\boldsymbol{x}_{[k]}$ of the system is not the absolute position of the target but rather the relative position of a target with respect to the observer. The matrices $\mathbf{A}$ and $\mathbf{B}$ from section 3.1 remain the same, but the state vector is changed to

$$\boldsymbol{x}_{[k]} = \begin{bmatrix} \boldsymbol{p}_{T[k]} - \boldsymbol{p}_{I[k]} \\ \boldsymbol{v}_{T[k]} - \boldsymbol{v}_{I[k]} \end{bmatrix} \in \mathbb{R}^6, \tag{3.18}$$

where $\{\boldsymbol{p}_{T[k]}, \boldsymbol{v}_{T[k]}\} \in \mathbb{R}^3$ are the position and velocity of the target, and $\{\boldsymbol{p}_{I[k]}, \boldsymbol{v}_{I[k]}\} \in \mathbb{R}^3$ are position and velocity of the observer at time instant k. The measurement from equation (2.1) is expressed with a direction vector $\boldsymbol{d}_{[k]}$, which can be defined as a non-linear measurement equation

$$\boldsymbol{d}_{[k]} = \boldsymbol{d}^*_{[k]} + \boldsymbol{\eta}_{[k]}, \tag{3.19}$$

where $\boldsymbol{d}^*_{[k]}$ is the unit direction vector pointing from the observer to the target, and $\boldsymbol{\eta}_{[k]} \sim \mathcal{N}(0, \sigma^2_\eta \mathbf{I}_{3\times3})$ is bearing vector measurement noise. The direction vector $\boldsymbol{d}^*$ is defined as

$$\boldsymbol{d}^*_{[k]} = \frac{\boldsymbol{p}_{T[k]} - \boldsymbol{p}_{I[k]}}{m_{[k]}}, \tag{3.20}$$

where $m_{[k]} = \|\boldsymbol{p}_{T[k]} - \boldsymbol{p}_{I[k]}\|$ is the distance between the observer and the target. To linearize the measurements, the authors of [3] proposed to use the matrix $\mathbf{P}_{\boldsymbol{d}_{[k]}}$ which represents an orthogonal projection to a plane perpendicular to the measurement bearing vector and defined as

$$\mathbf{P}_{\boldsymbol{d}_{[k]}} = \mathbf{I}_{3\times3} - \frac{\boldsymbol{d}_{[k]}\boldsymbol{d}^\intercal_{[k]}}{\|\boldsymbol{d}_{[k]}\|^2} \in \mathbb{R}^{3\times3}, \tag{3.21}$$

where $\mathbf{I}_{3\times3} \in \mathbb{R}^{3\times3}$ is the identity matrix. For any vector $\boldsymbol{g} \in \mathbb{R}^3$, the orthogonal projection of $\boldsymbol{g}$ onto a plane perpendicular to $\boldsymbol{d}$ is $\mathbf{P}_{\boldsymbol{d}}\boldsymbol{g}$. It also has properties $\mathbf{P}^\intercal_{\boldsymbol{d}} = \mathbf{P}_{\boldsymbol{d}}$, $\mathbf{P}^2_{\boldsymbol{d}} = \mathbf{P}_{\boldsymbol{d}}$ and $\text{null}(\mathbf{P}_{\boldsymbol{d}}) = \boldsymbol{d}$. By multiplying equation (3.19) by $\mathbf{P}_{\boldsymbol{d}_{[k]}}$ from the left, and substituting the $\boldsymbol{d}^*$ from (3.20), the following form is obtained:

$$\mathbf{0}_{3\times1} = \mathbf{P}_{\boldsymbol{d}}(\boldsymbol{p}_{T[k]} - \boldsymbol{p}_{I[k]}) + m_{[k]}\mathbf{P}_{\boldsymbol{d}_{[k]}}\boldsymbol{\eta}_{[k]}. \tag{3.22}$$

This is a pseudo-linear equation that can be used in the KF framework after minor modification. The final measurement equation is then

$$\boldsymbol{z}_{[k]} = \mathbf{0}_{3\times1} = \begin{bmatrix} \mathbf{P}_{\boldsymbol{d}_{[k]}} & \mathbf{0}_{3\times3} \end{bmatrix} \boldsymbol{x} + m_{[k]}\mathbf{P}_{\boldsymbol{d}}\boldsymbol{\eta}_{[k]}, \tag{3.23}$$

the measurement-state mapping matrix from equation (2.1) is

$$\mathbf{H}_{[k]} = \begin{bmatrix} \mathbf{P}_{\boldsymbol{d}_{[k]}} & \mathbf{0}_{3\times3} \end{bmatrix}, \tag{3.24}$$

and the measurement noise covariance matrix is obtained as

$$\mathbf{R}_{[k]} = \mathbf{V}_{[k]}\mathbf{R}\mathbf{V}^\intercal_{[k]}, \tag{3.25}$$

where the matrix $\mathbf{V}_k$ is

$$\mathbf{V}_{[k]} = m_{[k]}\mathbf{P}_{\boldsymbol{d}}, \tag{3.26}$$

following from equation (3.23). The measurement equation (3.23) depends on the real distance $m_{[k]}$ between the observer and the target, but it is not known, and it is exactly the value to be estimated. The true $m_{[k]}$ can be replaced by a predicted one, as it was done in [3] and proved to be a good estimate.

To express the approach described in this section consistent with the model defined in section 3.1, the state vector should be $\boldsymbol{x} = \begin{bmatrix}\boldsymbol{p}_T & \boldsymbol{v}_T\end{bmatrix}^\intercal$ and not the relative position in equation (3.18), as defined in original article [3]. To do so, the $\mathbf{P}_{\boldsymbol{d}_{[k]}}\boldsymbol{p}_{I[k]}$ is moved on another side of the equation (3.22), transforming the measurement $\boldsymbol{z}_{[k]}$ from (3.23) to

$$\boldsymbol{z}_{[k]} = \mathbf{P}_{\boldsymbol{d}_{[k]}}\boldsymbol{p}_{I[k]} = \begin{bmatrix}\mathbf{P}_{\boldsymbol{d}_{[k]}} & \mathbf{0}_{3\times 3}\end{bmatrix}\boldsymbol{x} + m_{[k]}\mathbf{P}_{\boldsymbol{d}}\boldsymbol{\eta}_{[k]}. \tag{3.27}$$

As a result, the measurement-state matrix $\mathbf{H}_{[k]}$ from (3.24), the measurement $\boldsymbol{z}_{[k]}$ from (3.27) and the measurement noise covariance matrix $\mathbf{R}_{[k]}$ from (3.25) can directly be used in the KF framework defined in section 2.1 by using the model from section 3.1.

Degenerate Kalman Filter

The DKF is a framework for integrating general linear subspace measurements of the state-space using the LKF and can be utilized for the bearing-based estimation in a similar manner as the uPLKF. It was proposed and derived by Matouš Vrba and Viktor Walter from the Multi-Robot Systems group but has not yet been published. The DKF implementation is available within the Multi Robot Systems (MRS) Unmanned Aerial Vehicle (UAV) system [11], which was modified for the purposes of this thesis. In this section, the DKF is firstly introduced and derived in a simplified 2D example and then extended to specific bearing-only 3D position estimation.

2D DKF example

The step-by-step derivation of DKF formulation for the 2D case follows. If the observer is located at the origin of the coordinate frame and the target is in front of it in x direction as shown in subfigure 3.2a, the measurement is expressed with offset and uncertainty only as the

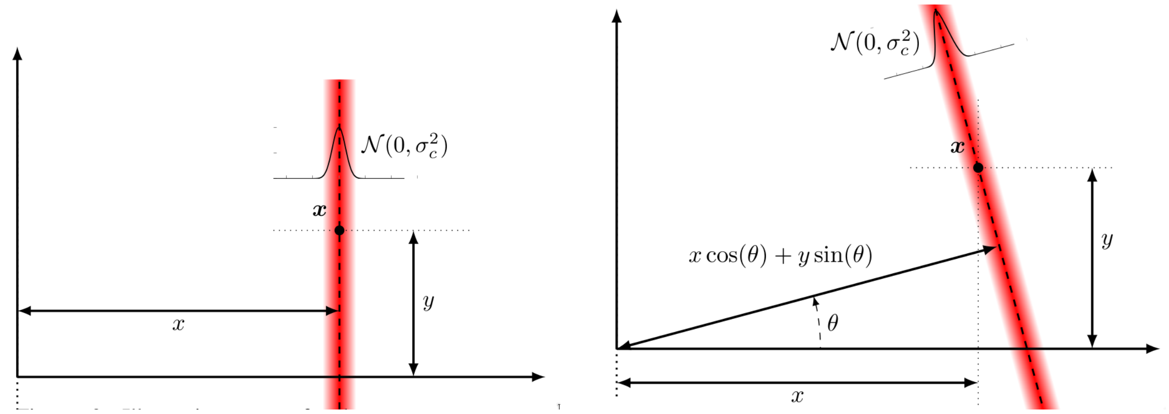
$$c = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x_T \\ y_T\end{bmatrix} + e, \tag{3.28}$$

where c is a measured distance to the target, the vector $\begin{bmatrix}x_T & y_T\end{bmatrix}^\intercal$ is the real position of the target, and $e \sim \mathcal{N}(0, \sigma_c^2)$ is Gaussian measurement noise. For a generally positioned target shown in subfigure 3.2a, the measurement can be expressed as

$$c = \begin{bmatrix}1 & 0\end{bmatrix}\mathbf{R}'\begin{bmatrix}x_T \\ y_T\end{bmatrix} + e, \tag{3.29}$$

where $\mathbf{R}' \in \mathit{SO(2)}$ is a rotation matrix aligning the target's direction with the x axis. Finally, if the observer is not at the origin, an offset $\begin{bmatrix}x_0 & y_0\end{bmatrix}^\intercal$ as

$$c = \begin{bmatrix}1 & 0\end{bmatrix}\mathbf{R}'\begin{bmatrix}x_T \\ y_T\end{bmatrix} + \begin{bmatrix}x_0 \\ y_0\end{bmatrix} + e. \tag{3.30}$$


$\mathcal{N}(0, \sigma_c^2)$
$\boldsymbol{x}$
x
y
$\mathcal{N}(0, \sigma_c^2)$
$\boldsymbol{x}$
$x\cos(\theta) + y\sin(\theta)$
θ
y
x

(a) illustration of the subspace measurement of a single coordinate in 2D case. The red line shows the measurement with missing knowledge in the y axis, and uncertainty expressed as a Gaussian probability distribution with x mean and σ_c deviation.

(b) Illustration of the subspace measurement of generally positioned target, the line with known inclination. The red line shows the same measurement type as in subfigure 3.2b.

Figure 3.2: The illustration of a simplified 2D case for deriving the DKF. Images taken from unpublished paper (written by Matouš Vrba and Viktor Walter) mentioned in section 3.3.2.

Bearing-only estimation using DKF

For the bearing-only position estimation, the measurement can be interpreted as a line defined as a position of the observer $\boldsymbol{p}_{I[k]}$ with the direction vector $\boldsymbol{d}_{[k]}$ obtained from the visual tracking. The specific measurement equation for the bearing-only position, extending the 2D example from equation (3.30) is defined as

$$\begin{bmatrix}\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3}\end{bmatrix}\begin{bmatrix}\boldsymbol{p}_{T[k]}\\ \boldsymbol{v}_{T[k]}\end{bmatrix} = \boldsymbol{d}_{[k]}\lambda + \boldsymbol{p}_{I[k]} + \boldsymbol{\epsilon}_{[k]}, \tag{3.31}$$

where λ is a nonzero scalar and $\boldsymbol{\epsilon}_{[k]} \sim \mathcal{N}(0, \sigma_\epsilon^2 \mathbf{I}_{3\times3})$ is a Gaussian noise. Let us define the matrix $\mathbf{M}$ to be the mapping from a state vector $\boldsymbol{x}_{[k]}$ to the measurement subspace. The general form of the measurement equation for 3D case used in the DKF framework is

$$\mathbf{M}\boldsymbol{x}_{[k]} = \mathbf{W}_{[k]}\lambda + \mathbf{O} + \boldsymbol{\epsilon}_{[k]}, \tag{3.32}$$

where matrix $\mathbf{W}_{[k]}$ is a measured subspace and $\boldsymbol{o}_{I[k]}$ is the offset. From equations (3.31) and (3.32), the matrix $\mathbf{W}_{[k]}$ for the state $\boldsymbol{x}$ defined above is a bearing vector $\boldsymbol{d}_{[k]}$. The mapping matrix $\mathbf{M}$ between the state vector subspace $\boldsymbol{x} = \begin{bmatrix}\boldsymbol{p}_T & \boldsymbol{v}_T\end{bmatrix}^\intercal$ to the 3D subspace of the position only, dropping the velocity, is specifically set to

$$\mathbf{M} = \begin{bmatrix}\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3}\end{bmatrix} \in \mathbb{R}^{3\times6}. \tag{3.33}$$

However, such measurement from equation (3.32) can not be directly used in KF. Let the matrix $\mathbf{N}_{[k]}$ in general case be a null space of $\mathbf{W}$, such that $\mathbf{N}_{[k]}^\intercal \mathbf{W}_{[k]} = \mathbf{0}$. For the bearing-only position estimation, matrix $\mathbf{N}_{[k]} \in \mathbb{R}^{3\times2}$ is the null space of the bearing vector $\boldsymbol{d}_{[k]}$. By left multiplication of $\mathbf{N}^\intercal$ the bearing vector is eliminated from the measurement formula, and the resulting measurement equation is transformed to

$$\mathbf{N}_{[k]}^\intercal \mathbf{M}\boldsymbol{x}_{[k]} = \mathbf{N}_{[k]}^\intercal \boldsymbol{p}_{I[k]} + \mathbf{N}_{[k]}^\intercal \boldsymbol{\epsilon}_{[k]}, \tag{3.34}$$

which is in a canonical form used in the KF framework. The measurement noise $\boldsymbol{\epsilon}_{[k]}$ is a Gaussian noise with zero mean, so a change of the sign will keep the correctness of the equation. Finally, the measurement-state mapping matrix $\mathbf{H}_{[k]}$ for the KF from equation (2.1) is

$$\mathbf{H}_{[k]} = \mathbf{N}_{[k]}^\intercal \mathbf{M} \in \mathbb{R}^{2\times 6}, \tag{3.35}$$

the measurement vector is

$$\boldsymbol{z}_{[k]} = \mathbf{N}_{[k]}^\intercal \boldsymbol{p}_{I[k]} \in \mathbb{R}^{2\times 1}, \tag{3.36}$$

and the measurement noise covariance matrix is computed as

$$\mathbf{R}_{[k]} = \sigma_\epsilon^2 \mathbf{N}_{[k]}^\intercal \mathbf{N}_{[k]} \in \mathbb{R}^{3\times 3}. \tag{3.37}$$

The bearing vector measurement noise $\boldsymbol{\epsilon}$ for DKF is not the same as $\boldsymbol{\eta}$ used for bearing for uPLKF and defined in equation (3.19). The standard deviation ϵ_η for DKF can be visualized as the radius of an uncertainty cylinder around the bearing vector.

3.4 Bearing-based estimation with subtended angle using KF

One of the main disadvantages of all approaches described above is the lack of information in the dimension orthogonal to the bearing vector. To gain this information, for example, the observer can follow some special trajectory as described in [3]. By following the circular trajectory in a plane orthogonal to the initial bearing vector, more information about the size l shown in Figure 3.3 can be gained. Another way to receive that information is to utilize knowledge about the subtended angle θ shown in Figure 3.3, which is already available from most visual trackers, providing not a point but a bounding box.

Uniform Pseudo Linear Kalman Filter with subtended angle

A uPLKF variant using the subtended vector of the target is proposed in [1]. The work is based on research from [16] and the uPLKF [3] explained in section 3.3.1. The idea is to incorporate the subtended angle measurement in a KF framework to estimate the size of the object under the assumption that the bounding box of the object's 2D tracking is similar from most viewpoints. In Figure 3.1, the subtended angle is shown as θ, and the bearing vector as $\boldsymbol{g}$. In Figure 3.3, the overall scene is shown with the same notation from chapter 3. By adding the new value to the state, the state vector $\boldsymbol{x}$ defined in equation (3.1) is transformed to

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{p}_T \\ \boldsymbol{v}_T \\ l \end{bmatrix} \in \mathbb{R}^7, \tag{3.38}$$

where l is the object's physical size in the real world, in the dimension orthogonal to the bearing vector $\boldsymbol{d}$. If the object's size is only slightly different from different perspectives, it can still be estimated, as proved in [1].

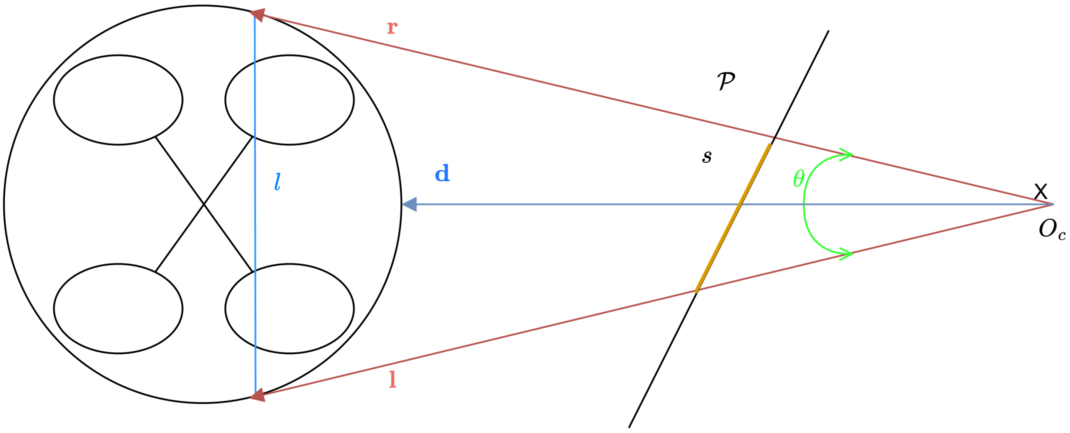


Figure 3.3: An example schematic of relative position estimation with the subtended angle in top view on the scene defined in Figure 3.1. The observer located at O_c measures the bearing vector $\boldsymbol{d}$ with a subtended angle θ. Vectors $\boldsymbol{l}$ and $\boldsymbol{r}$ point to the left and right sides of the target, respectively. Plane $\mathcal{P}$: An example relative position estimation in side view. The observer measures the target bearing vector in the camera projection plane, s is the projected target size, and l shows the target's physical size in the dimension orthogonal to the bearing vector.

Usage of additional information in the dimension orthogonal to the bearing vector can improve the distance estimation, and, as follows, the 3D tracking results in situations where the system state described in previous sections causes the singularities and makes it impossible to locate the target, for example in the case of a static target, the observer moves at a constant speed in a straight line towards the target.

The subtended angle $\theta^*_{[k]}$ is computed as

$$\theta^* = \frac{\boldsymbol{l}^\intercal \boldsymbol{r}}{||\boldsymbol{l}|| ||\boldsymbol{r}||} \in \mathbb{R} \tag{3.39}$$

where $\boldsymbol{l}$ is the vector pointing to the middle of the left bounding box edge, and $\boldsymbol{r}$ is pointing to the middle of the right side. The measured $\theta_{[k]}$ computed as

$$\theta_{[k]} = \theta^*_{[k]} + \omega,\ \omega \sim \mathcal{N}(0, \sigma^2_\omega). \tag{3.40}$$

The angle θ is considered to be relatively small because the object is observed from a big distance. For small angles, the tangent is assumed to be equal to the angle itself,

$$\tan\theta \approx \theta, \tag{3.41}$$

and the error from this estimation is shown in Figure 3.4. This error is computed as

$$\epsilon = \frac{|l_{ideal} - l_{approx}| \cdot 100\,\%}{l_{ideal}}, \tag{3.42}$$

with $l_{ideal} = 2m\tan(\frac{\theta^*}{2})$ and $l_{approx} = 2m\tan(\frac{\theta}{2})$. Each color represents the actual target size in the dimension orthogonal to the bearing angle $\boldsymbol{d}$. It is clear that starting from 2.5 m

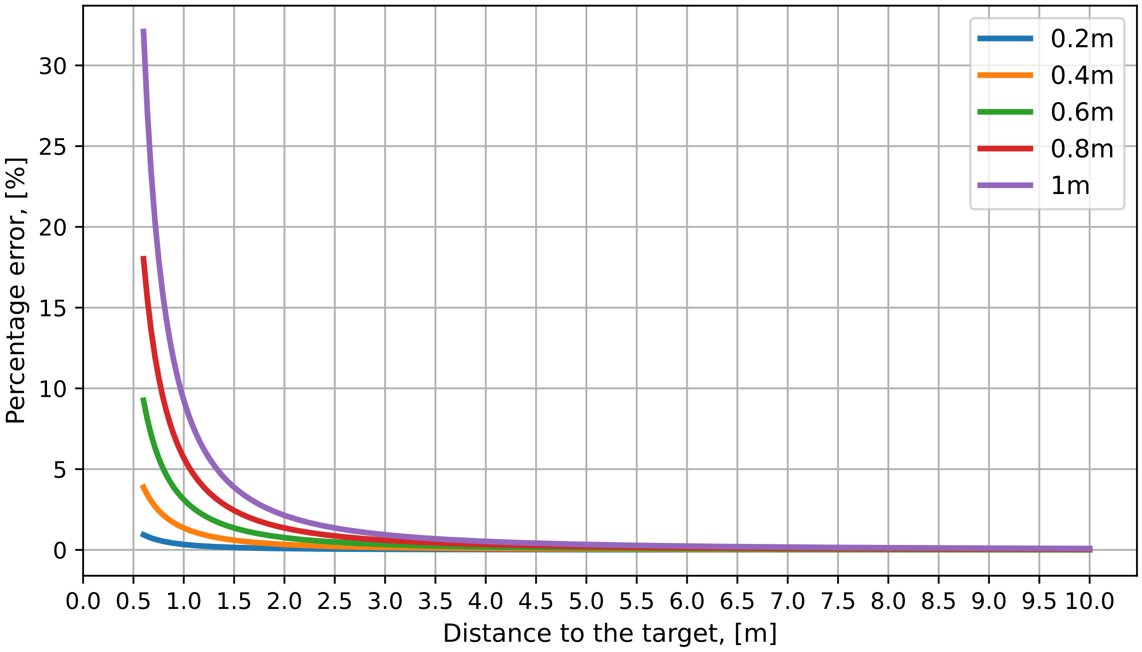


Figure 3.4: Error of the target's estimated size using the tangent approximation. The relation of the tangent of an angle approximation with just the angle with respect to the distance to the object of interest. Different colors show a different target size in the dimension orthogonal to the bearing vector $\boldsymbol{d}$.

distance, the estimation error is less than $3\,\%$ for any object size, so the approximation can be used without a significant impact on the measurement precision.

From Figure 3.3, the ideal subtended angle is computed as

$$\theta^*_{[k]} = 2 \arctan \frac{\frac{1}{2} l}{m_{[k]}}, \tag{3.43}$$

which using the approximation, can be simplified to

$$\theta^*_{[k]} \approx \frac{l}{m_{[k]}}, \tag{3.44}$$

where $m_{[k]} = \|\boldsymbol{p}_{T[k]} - \boldsymbol{p}_{I[k]}\|$ is the absolute distance from the observer to the target. Combining equations (3.39), (3.40) and (3.44), the nonlinear measurement equation of the subtended angle is

$$\theta_{[k]} = \frac{l}{m_{[k]}} + \omega. \tag{3.45}$$

To linearize, this equation can be combined with the bearing vector measurement. From equations (3.19) and (3.20) follows that

$$m_{[k]} \boldsymbol{d}_{[k]} = \boldsymbol{p}_{T[k]} - \boldsymbol{p}_{I[k]} + m_{[k]} \boldsymbol{\eta}, \tag{3.46}$$

and after multiplying equations (3.45) and (3.46) the result is

$$\theta_{[k]}(\boldsymbol{p}_{T[k]} - \boldsymbol{p}_{I[k]} + m_{[k]}\boldsymbol{\eta}) = l\boldsymbol{d}_{[k]} + \omega m_{[k]}\boldsymbol{d}_{[k]}, \tag{3.47}$$

and the final pseudo-linear measurement equation for the subtended angle is

$$\theta_{[k]}\boldsymbol{p}_{I[k]} = \begin{bmatrix}\theta_{[k]}\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & -\boldsymbol{d}_{[k]}\end{bmatrix}\begin{bmatrix}\boldsymbol{p}_{T[k]}\\ \boldsymbol{v}_{T[k]}\\ l\end{bmatrix} + m_{[k]}(\theta_{[k]}\boldsymbol{\eta} - \omega\boldsymbol{d}_{[k]}). \tag{3.48}$$

The resulting uPLKF matrices from 3.3.1 are modified to satisfy the extension of the measurement $\boldsymbol{x}$. The state transition matrix is

$$\mathbf{A} = \begin{bmatrix}\mathbf{I}_{3\times3} & \triangle t\mathbf{I}_{3\times3} & \mathbf{0}_{3\times1}\\ \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times1}\\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 1\end{bmatrix} \in \mathbb{R}^{7\times7}, \tag{3.49}$$

the matrix $\mathbf{B}$ remains the same as in 3.1, but the noise vector is changed as

$$\boldsymbol{w}_{[k]} \sim \mathcal{N}(0, \begin{bmatrix}\mathbf{BQB}^\intercal & \mathbf{0}_{6\times1}\\ \mathbf{0}_{1\times6} & \sigma_l^2\end{bmatrix}). \tag{3.50}$$

with σ_l variance of the target size, letting the l slowly vary and not remain static. If the target is either a cylinder, sphere, or any other figure with a constant size in the horizontal or vertical dimension, σ_l should be zero. The measurement-state mapping is transformed to

$$\mathbf{H}_{[k]} = \begin{bmatrix}\mathbf{P}_{\boldsymbol{d}} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1}\\ \theta_{[k]}\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & -\boldsymbol{d}_{[k]}\end{bmatrix} \in \mathbb{R}^{6\times7}, \tag{3.51}$$

and the measurement vector

$$\boldsymbol{z}_{[k]} = \begin{bmatrix}\mathbf{P}_{\boldsymbol{d}_{[k]}}\boldsymbol{p}_{I[k]}\\ \theta_{[k]}\boldsymbol{p}_{I[k]}\end{bmatrix} \in \mathbb{R}^6. \tag{3.52}$$

The measurement noise vector can be written as

$$\boldsymbol{e}_{[k]} = \begin{bmatrix}m_{[k]}\mathbf{P}_{\boldsymbol{d}_{[k]}}\boldsymbol{\eta}\\ m_{[k]}(\theta_{[k]}\boldsymbol{\eta} - \omega\boldsymbol{d}_{[k]})\end{bmatrix} \in \mathbb{R}^6. \tag{3.53}$$

To find the covariance matrix $\mathbf{R}$ of the vector $\boldsymbol{e}_{[k]}$, let us introduce the matrix $\mathbf{E}$ equals

$$\mathbf{E}_{[k]} = m_{[k]}\begin{bmatrix}\mathbf{P}_{\boldsymbol{d}_{[k]}} & \mathbf{0}_{3\times1}\\ \theta_{[k]}\mathbf{I}_{3\times3} & -\boldsymbol{d}_{[k]}\end{bmatrix} \in \mathbb{R}^{6\times4} \tag{3.54}$$

making available to write the measurement noise as a linear transformation of Gaussian noises

$$\boldsymbol{e}_{[k]} = \mathbf{E}_{[k]}\begin{bmatrix}\boldsymbol{\eta}\\ \omega\end{bmatrix} \in \mathbb{R}^6 \tag{3.55}$$

with the covariance matrix

$$\mathbf{R}_{[k]} = \mathbf{E}_{[k]}\begin{bmatrix}\sigma_\eta^2\mathbf{I}_{3\times3} & \mathbf{0}_{3\times1}\\ \mathbf{0}_{1\times3} & \sigma_w^2\end{bmatrix}\mathbf{E}_{[k]}^\intercal \in \mathbb{R}^{6\times6}. \tag{3.56}$$

The value of m - the absolute distance from the observer to the target - can not be found; instead, the estimated value from the KF loop is used, as was described previously.

Degenerate Kalman Filter with subtended angle

For now, the DKF approach implemented in MRS library utilized only the knowledge about the bearing vector with some uncertainty. After seeing that the subtended angle fusion with the uPLKF improved the estimation results, the fusion of subtended angle measurements with the DKF was developed and implemented. By combining the algorithms from sections 3.3.2 and 3.4.1, the subtended angle measurements can be incorporated into the DKF approach. However, the model described did not use the bearing vector directly before, which is needed for the angle measurements. Adding this to the model will increase the amount of noise sources and make the tuning process harder.

Combining the derived measurement equation for the subtended angle (3.48), the measurement-state mapping $\mathbf{H}$ from the section 3.3.2 is transformed to

$$\mathbf{H}_{[k]} = \begin{bmatrix} \mathbf{N}^{\intercal}_{[k]}\mathbf{M} & \mathbf{0}_{2\times 1} \\ \theta_{[k]}\mathbf{M} & -\boldsymbol{d}_{[k]} \end{bmatrix} \in \mathbb{R}^{5\times 7}, \tag{3.57}$$

where the matrix $\mathbf{M} = \begin{bmatrix}\mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 3}\end{bmatrix}$ is a mapping from a position and velocity to position only and the matrix $\mathbf{N}$ is a null space of $\boldsymbol{d}$ such that $\mathbf{N}^{\intercal}\boldsymbol{d} = \mathbf{0}$. The measurement vector is changed to

$$\boldsymbol{z}_{[k]} = \begin{bmatrix} \mathbf{N}_{[k]}\boldsymbol{p}_{I[k]} \\ \theta_{[k]}\boldsymbol{p}_{I[k]} \end{bmatrix} \in \mathbb{R}^{5}. \tag{3.58}$$

The measurement noise vector is changed to

$$\boldsymbol{e}_{[k]} = \begin{bmatrix} \mathbf{N}^{\intercal}_{[k]}\boldsymbol{\epsilon} \\ m_{[k]}(\theta_{[k]}\boldsymbol{\eta} - \omega\boldsymbol{d}_{[k]}) \end{bmatrix} \in \mathbb{R}^{5}. \tag{3.59}$$

Same as for the uPLKF with subtended angle, it can be written as

$$\boldsymbol{e}_{[k]} = \mathbf{E}_{[k]} \begin{bmatrix} \boldsymbol{\epsilon} \\ \boldsymbol{\eta} \\ \omega \end{bmatrix} \in \mathbb{R}^{5}, \tag{3.60}$$

where

$$\mathbf{E}_{[k]} = \begin{bmatrix} \mathbf{N}^{\intercal}_{[k]} & \mathbf{0}_{2\times 3} & \mathbf{0}_{2\times 1} \\ \mathbf{0}_{3\times 3} & m_{[k]}\theta_{[k]}\mathbf{I}_{3\times 3} & -m_{[k]}\boldsymbol{d}_{[k]} \end{bmatrix} \in \mathbb{R}^{5\times 7}, \tag{3.61}$$

so the measurement covariance matrix is

$$\mathbf{R} = \mathbf{E}_{[k]} \begin{bmatrix} \sigma_e^2\mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{3\times 3} & \sigma_\eta^2\mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \sigma_w^2 \end{bmatrix} \mathbf{E}^{\intercal}_{[k]} \in \mathbb{R}^{5\times 5}, \tag{3.62}$$

where σ_η is the variance of bearing vector $\boldsymbol{d}$ measurement, the σ_ϵ is the variance of the DKF measurement (radius of an uncertainty cylinder in this case) and σ_ω is the variance of subtended angle.

Chapter 4

Evaluation

In this chapter, the implementation and evaluation of concidered methods in both simulation and the real world are described. All approaches were implemented as described in section 4.1. The pipeline for testing the advantages and disadvantages of the implemented methods is described in section 4.2. The 3D position localization precision with respect to the visual tracking deviation and the observer's self-localization deviation was evaluated on different scenarios in simulation and described in section 4.5. The evaluation of the real-world scenarios is presented in section 4.6.

4.1 Implementation

All methods described in section 3 were implemented in the C++ programming language and the postprocessing scripts for data analysis and visualization in Python3. The MRS UAV system [11] was used as a middleware for control and state estimation of the drone. It is a set of libraries and programs to accelerate and simplify algorithms development and hypothesis testing on drones in both simulations and in the real world, for indoor and outdoor environments. It provides basic implementations of controllers, state estimators, and trajectory generators, supporting different kinds of sensors, including monocular cameras, stereo cameras, LiDARs, sonars, etc. The MRS system uses the Robot Operating System (ROS)[1] for communication and connecting all parts. ROS is an open-source ecosystem containing dozens of already implemented drivers for the hardware, abstract algorithms, and libraries for communicating onboard a robot and managing coordinate transformations. The Eigen[2] open-source library was used for all linear algebra manipulations with vectors and matrices, the implementation of SVD and pseudo-inversions.

The baseline methods from section 3.2 were implemented from scratch, and the **uPLKF-based!** (**uPLKF-based!**) methods from sections 3.3.1, 3.4.1 were implemented according to [3] and [1], respectively. The initial implementation of the method described in section 3.3.2 was taken from the MRS library. The method from section 3.4.2 was implemented by improving the DKF and combining it with the implementation of uPLKF with the subtended angle. All implemented approaches were tested in a realistic simulation using

[1] ROS: https://www.ros.org/

[2] Eigen library: http://eigen.tuxfamily.org

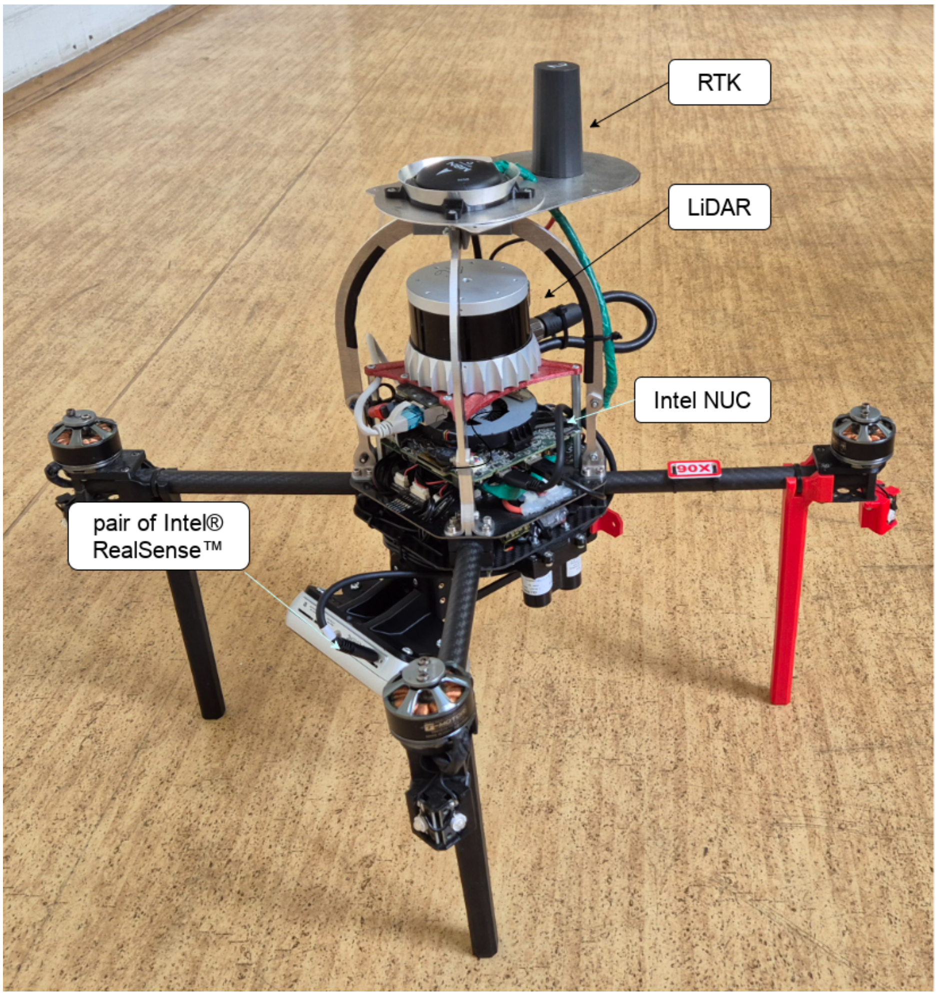


Figure 4.1: The picture of a drone used for real-world data collection. It carries an onboard LiDAR for detection and tracking, two tilted Intel Realsence D455 sensors pointing down for tracking outside the FOV of a LiDAR, RTK sensor for precise localization, and Intel NUC as an onboard computer.

the Gazebo simulator[3]. The best-performing approaches were also evaluated on the data recorded from real-world flights. Figure 4.2 shows the observer and target models in the environment of the Gazebo simulator.

In the real world for data recording, the drone shown in Figure 4.1 was used, carrying all necessary sensors. The onboard computer was Intel NUC 10i7FNK with Intel Core i7 processor inside. Red Green Blue (RGB) images were taken by one of the Intel RealSense D455 mounted on the drone. Two such sensors were mounted, tilted up and down to cover a bigger region under the drone. The Ouster OS1-16 LiDAR was used for the initial detection and tracking of the target that was used for the KF state initialization. The ground-truth states of the observer and the target were recorded using the RTK localization system to

[3] Gazebo simulator: https://gazebosim.org/home

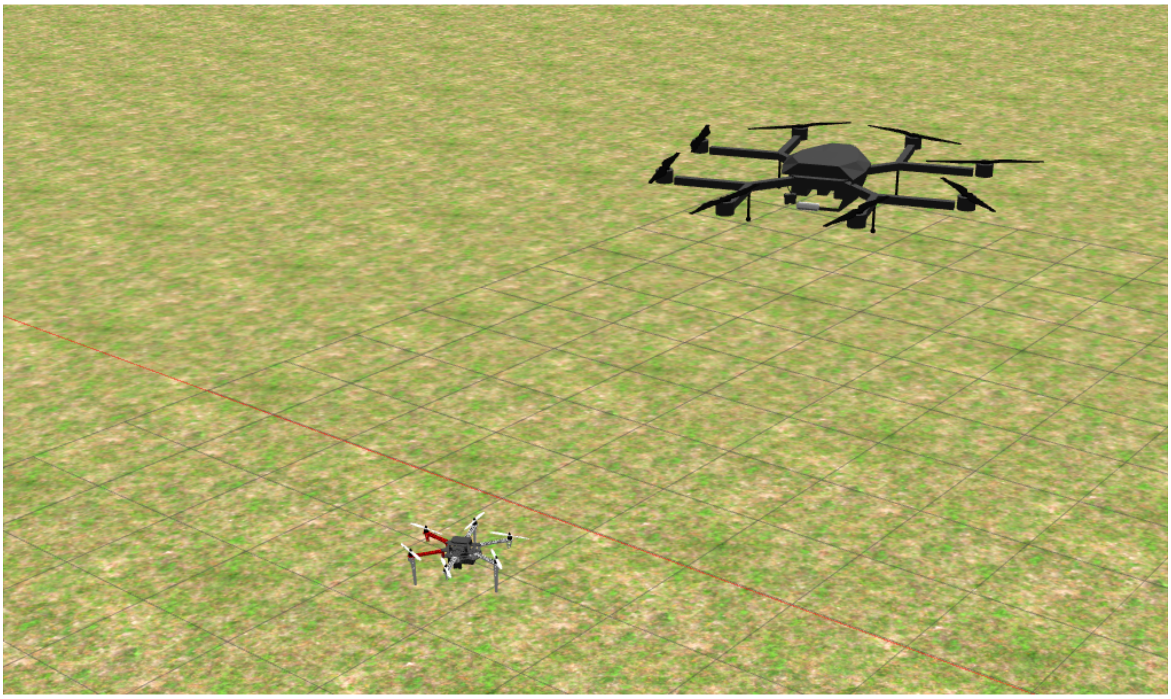


Figure 4.2: Observer and target models in the environment of the Gazebo simulator. The MAV at the top is the observer, and at the bottom one is the target.

receive precise positions, and the Pixhawk PX4 was used as a flight controller.

4.2 Architecture of the pipeline

Figure 4.3 shows the general testing pipeline. The observer's state estimation was performed by the MRS UAV system. Ground-truth odometry with the position of the observer in the world frame was taken as ground-truth $\boldsymbol{p}_I^*$. To evaluate the localization accuracy in relation to the uncertainty of the pose of the MAV carrying the camera, Gaussian noise with zero mean and parameterized variance was added to the ground truth measurement $\boldsymbol{p}_I$ as

$$\boldsymbol{p}_I = \boldsymbol{p}_I^* + \boldsymbol{e}_{pI}, \tag{4.1}$$

where

$$\boldsymbol{e}_{pI} \sim \mathcal{N}(\mathbf{0}_{3\times1}, \sigma_{pos}^2 \mathbf{I}_{3\times3}). \tag{4.2}$$

The standard deviation of the observer's position σ_{pos} is a parameter that is changed to evaluate the quality of the state estimation with respect to the observer's self-localization precision.

To make the environment more controllable and evaluate the localization accuracy in relation to the uncertainty of the visual tracking, Gaussian noise with zero mean and parameterized variance was added to the measurement. The ground truth position of the target $\boldsymbol{p}_T^*$ was projected onto the image plane providing the ground-truth image coordinates

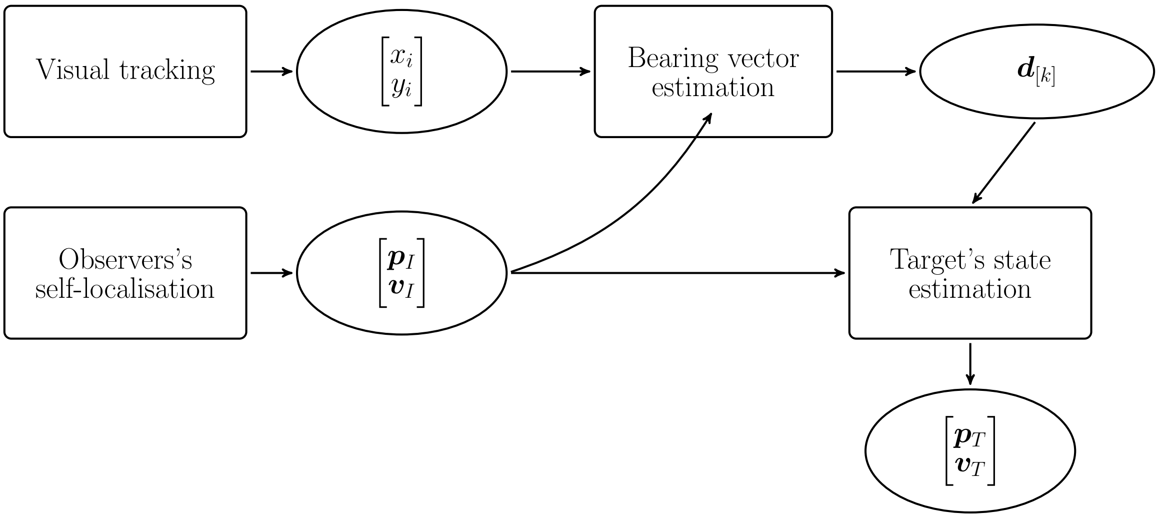

Visual tracking
$\begin{bmatrix} x_i \\ y_i \end{bmatrix}$
Bearing vector estimation
$\boldsymbol{d}_{[k]}$
Observers's self-localisation
$\begin{bmatrix} \boldsymbol{p}_I \\ \boldsymbol{v}_I \end{bmatrix}$
Target's state estimation
$\begin{bmatrix} \boldsymbol{p}_T \\ \boldsymbol{v}_T \end{bmatrix}$

Figure 4.3: The pipeline architecture used in testing. The rectangles show the processing nodes, and the corresponding results are in ellipses.

$\boldsymbol{j}_i^*$ of the center of the target

$$\boldsymbol{l}_i^* = \begin{bmatrix} x_i^* \\ y_i^* \end{bmatrix}. \tag{4.3}$$

The measurement with the added noise is defined as

$$\boldsymbol{j}_i = \begin{bmatrix} x_i^* \\ y_i^* \end{bmatrix} + \boldsymbol{e}_i, \tag{4.4}$$

$$\boldsymbol{e}_i \sim \mathcal{N}(\mathbf{0}_{2\times 1}, \sigma_{tr}^2 \mathbf{I}_{2\times 2}), \tag{4.5}$$

where σ_{tr} is the standard deviation of the visual tracking in x and y coordinates in pixels. This is a parameter used to adjust the visual tracking precision and evaluate the localization accuracy.

The bearing vector $\boldsymbol{d}$ is computed using both the state of the observer and the point in the image at time instance k. The state of the observer and the bearing angle are then by the implemented methods to estimate the position of the target in 3D. The expected result of the pipeline is the estimated state of the target $\boldsymbol{x} = \begin{bmatrix} \boldsymbol{p}_{T[k]}^\intercal & \boldsymbol{v}_{T[k]}^\intercal \end{bmatrix}^\intercal$.

Except for parameters σ_{pos} and σ_{tr} for testing the limits of proposed solutions, there are also the process noise and the measurement noise covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ that have to be tuned for the KF to produce reasonable results. , The covariance matrices were tuned in order to be the same for the four scenarios in the simulation described further and produce reasonable results, ignoring any prior knowledge about each scenario. The same covariance matrices were used to test the KF-based methods with increased observer's self-localization and visual tracking uncertainties. The bearing vector uncertainties for $\boldsymbol{\eta}$, $\boldsymbol{\epsilon}$ and ω from equations (3.19), (3.32) and (3.40) were adjusted with respect to the visual tracking noise increase.

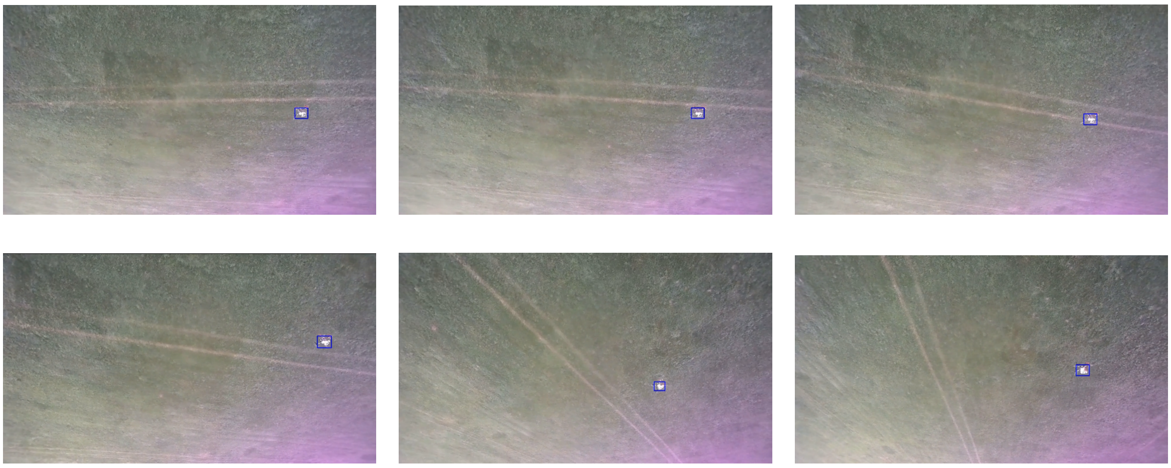


Figure 4.4: Visual tracking with bounding box visualization from the real-world experiment.

The Euclidean distance was used to compare the precision along the path. It is defined as

$$err_{[k]} = \|\boldsymbol{p}_{T[k]} - \boldsymbol{p}^*_{T[k]}\|, \tag{4.6}$$

where $\boldsymbol{p}_{T[k]}$ is the estimated value and $\boldsymbol{p}^*_{T[k]}$ is the ground-truth value at time instant k. The Root Mean Square Error (RMSE) metric was used to evaluate the overall localization precision along the path and compare methods with each other. It is defined as

$$\text{RMSE} = \sqrt{\frac{\sum_{k=1}^{N} \left(\boldsymbol{p}_{T[k]} - \boldsymbol{p}^*_{T[k]}\right)^2}{N}}, \tag{4.7}$$

where N is the number of non-missing data points.

4.3 Visual tracking

The visual tracking system was implemented using the OpenCV[4] implementation of the Median Flow tracker [22] and the ROS. The tracker was initialized by projecting the initialization for the KF, which is either a result of the LiDAR tracker or the ground-truth position of the target in the world coordinate frame, onto the image plane of the observer's onboard camera. The actual size of the target in 3D is assumed to be known. The Median Flow tracker is suitable for smooth and predictable movements. It is a feature tracker based on the Lucas-Kanade method but robust to partial conclusions that estimates the translation and scale. It tracks features within a given the bounding box using the Lucas-Kanade tracker. The features are filtered based on the forward-backward error proposed by the authors. From the remaining features, the displacement of the bounding box is estimated.

The sequence of images illustrating the process of visual tracking is shown in Figure 4.4. Images correspond to every tenth frame in a video recorded by a camera mounted onboard the observer in a real-world experiment. Visual tracking was not used to compare the 3D

[4] OpenCV: https://opencv.org/

tracking algorithms because it is's tracking error varies, which makes it hard to estimate the influence of the tracking error on the state estimation precision.

4.4 Simulated scenarios

In every scenario tested in a simulation, the initial velocity estimate for the state $\boldsymbol{x}$ was zero, to test the 3D tracking precision of every approach when there is no prior knowledge about the initial target's velocity. Otherwise, every KF-based method is showed good results due to the constant velocity and heading of the target, making the comparison too optimistic. In plots for comparing scenarios, only the KF-based approaches are shown to clarify the comparison.

For the comparison in this section, the initial pose of the observer is assumed to be known, $\boldsymbol{p}_I = \boldsymbol{p}_I^*$, and the visual tracking is assumed to be exact, $\boldsymbol{j}_i = \boldsymbol{j}_i^*$. The same $\mathbf{Q}$ and $\mathbf{R}$ covariance matrices were used to test the KF-based methods. To analyze the influence of the noise in section 4.5, the bearing vector uncertainties for $\boldsymbol{\eta}$, $\boldsymbol{\epsilon}$ and ω from equations (3.19), (3.32) and (3.40) were adjusted with respect to the visual tracking noise increase when testing it's influence on the state estimation.

The method described in section 3.2.1 is called "*svd-static*", and the method from section 3.2.2 is "*svd-synamic*". The methods from sections 3.3.1 and 3.3.2 are "*plkf*" and "*dkf*" respectively, and with the subtended angle named as "*plkft*" and "*dkft*" respectively.

Scenario 1

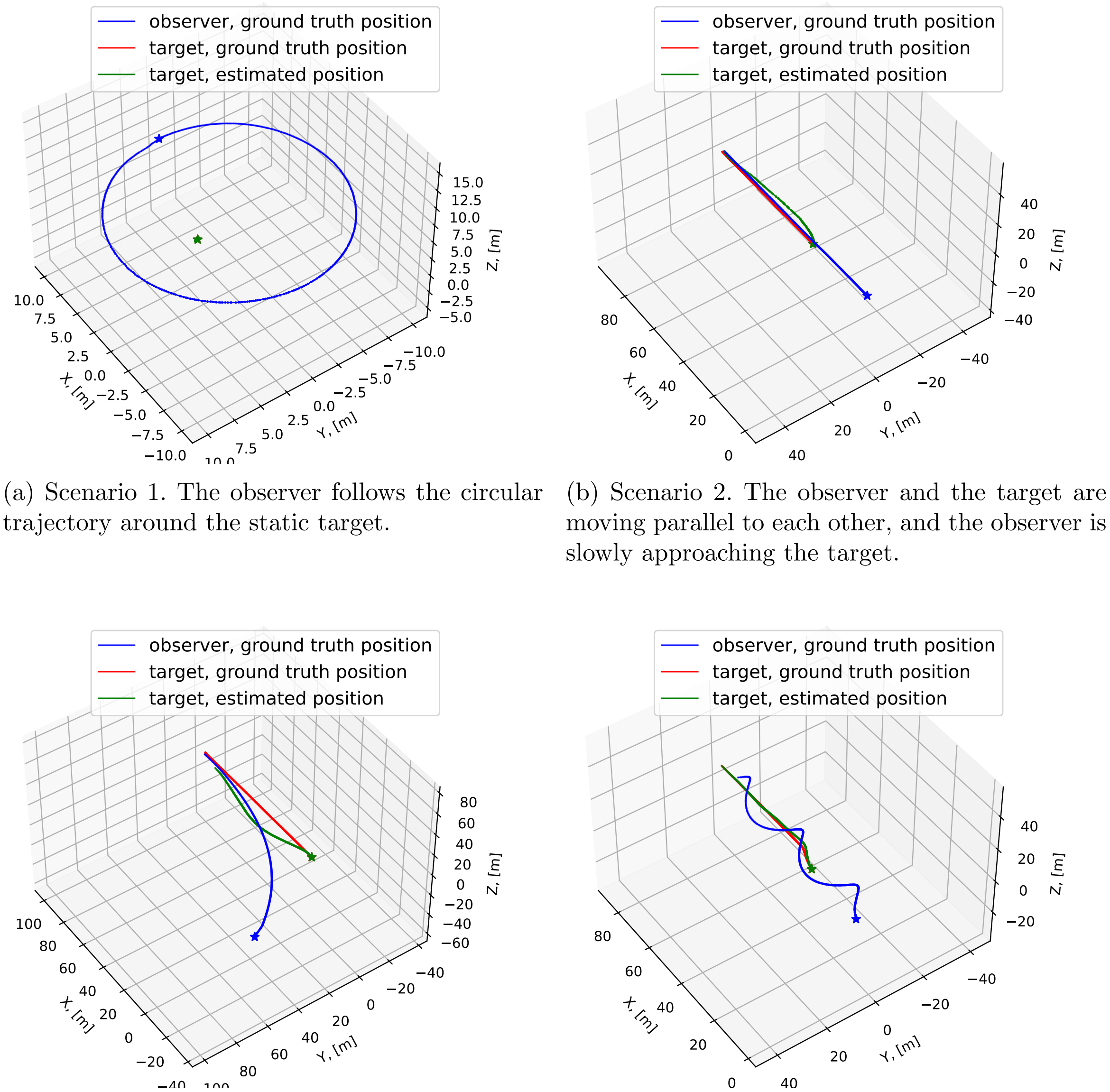
In the first scenario, the target was located on the position $\boldsymbol{p}_T^* = \begin{bmatrix} 3 & 0 & 4 \end{bmatrix}^\intercal$ and the observer was moving around the target on a circular trajectory in the xy plane with center $\boldsymbol{c} = \begin{bmatrix} 0 & 0 & 6 \end{bmatrix}^\intercal$ and radius 10 m. The target was not in the center of the circle to avoid a constant distance to the target and make the experiment more challenging. This trajectory was chosen to show the consistency and stability of the proposed approaches and compare the localization precision. The scheme of the experiment is in subfigure 4.5a.

Results

The results are shown in Figure 4.6. The "*svd-dynamic*" method always shows significantly worse results than the others methods. This can also be observed in Figure 4.10. It can be caused by the imperfection of the model that does not consider noise and relies on accurate time synchronization, as was described in section 3.2.2, or due to the sensitivity of the SVD solution and small values in the optimization matrix, which is the case because $_{\triangle}t$ change is very small. The KF-based approaches show much better results. In Figure 4.6, the comparison of KF-based methods for this scenario is presented. All KF-based methods showed localization precision of up to 50 cm.

Scenario 2

In the second scenario, the target and the observer were moving with a constant velocity along the x axis, but the observer was slowly approaching the target. It is one of the possible



(a) Scenario 1. The observer follows the circular trajectory around the static target.

(b) Scenario 2. The observer and the target are moving parallel to each other, and the observer is slowly approaching the target.

(c) Scenario 3. The target follows the line trajectory, and the observer flies using the pure pursuit guidance strategy.

(d) Scenario 4. The target follows the line trajectory, and the observer follows the helix trajectory.

Figure 4.5: Scenarios used in the Gazebo simulation. The best-performed estimation was visualized as a target's estimated position.

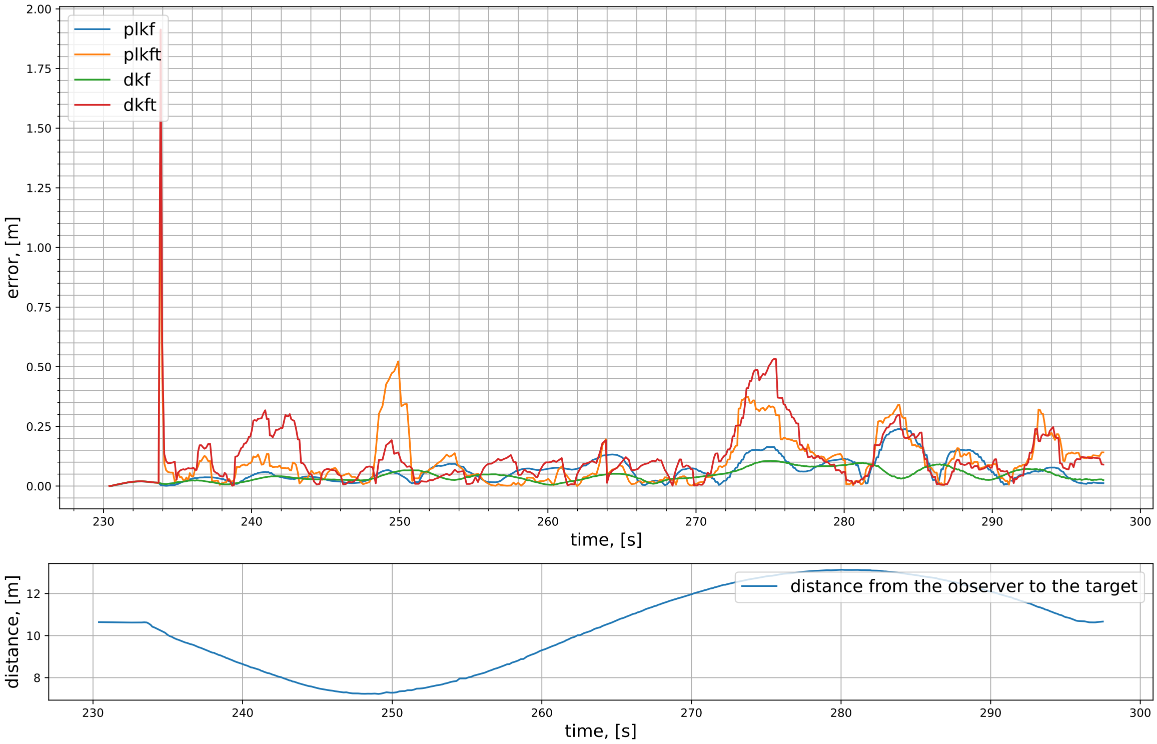


Figure 4.6: Results from scenario 1. The top plot shows the Euclidean distance between the estimated position and ground truth in meters. The bottom plot shows the distance between the observer and the target.

real-world situations for interception purposes. The observer started at a position $\boldsymbol{p}_I^* = \begin{bmatrix} 0 & 0 & 6 \end{bmatrix}^\intercal$ and was moving with velocity $1.5\,\mathrm{m\,s^{-1}}$ towards the point $\begin{bmatrix} 100 & 0 & 6 \end{bmatrix}^\intercal$. The target started at point $\boldsymbol{p}_T^* = \begin{bmatrix} 33 & 0 & 4 \end{bmatrix}^\intercal$ and was moving with velocity $1\,\mathrm{m\,s^{-1}}$ towards the point $\begin{bmatrix} 100 & 0 & 4 \end{bmatrix}^\intercal$. This trajectory was chosen to show the possible limitations of the presented approaches. The scheme of the experiment is in subfigure 4.5b.

Results

The results are shown in Figure 4.7. This scenario was much more challenging for the bearing-only approaches because they had no information in the dimension orthogonal to the bearing vector, and due to that, they struggled to estimate the distance correctly while moving on a straight line behind the target. However, at the end of the recorded scenario, the observer flew above the target, gaining more information on the dimensions not available before, quickly improving the estimate. As expected, both methods with subtended angles showed much better results from the beginning to the end, improving the estimate while approaching the target.

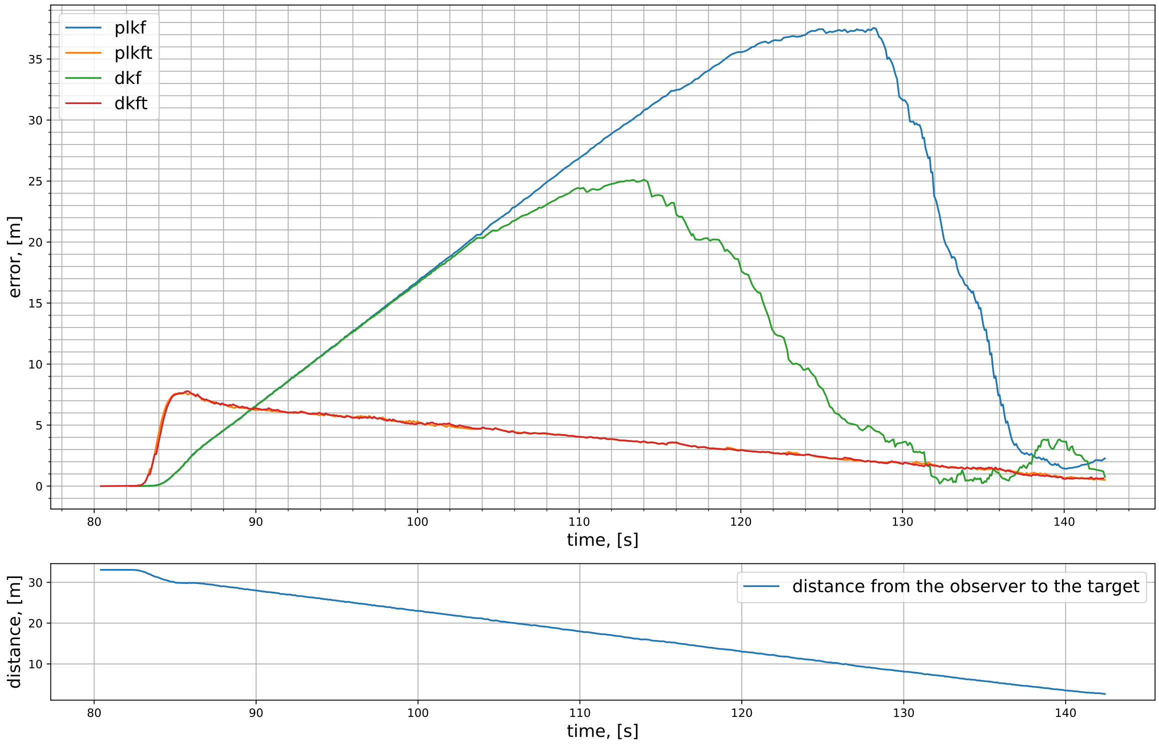


Figure 4.7: Results from scenario 2. The top plot shows the Euclidean distance between the estimated position and ground truth in meters. The bottom plot shows the distance between the observer and the target.

Scenario 3

The third scenario is an illustration of pure pursuit guidance. The observer's velocity vector $\boldsymbol{v}^*_{I[k]}$ was pointing towards the current position of the target at every time instance k and was computed as

$$\boldsymbol{v}^*_{I[k]} = g\frac{\boldsymbol{p}^*_{T[k]} - \boldsymbol{p}^*_{I[k]}}{\|\boldsymbol{p}^*_{T[k]} - \boldsymbol{p}^*_{I[k]}\|}, \tag{4.8}$$

where g is a gain. It is a trivial missile guidance law, but it illustrates one of the possible real applications for the observer. The target started at point $\boldsymbol{p}^*_T = \begin{bmatrix}0 & 0 & 4\end{bmatrix}^\intercal$ and was moving with velocity $3\,\mathrm{m\,s^{-1}}$ towards the point $\begin{bmatrix}100 & 0 & 4\end{bmatrix}^\intercal$. The observer started at point $\boldsymbol{p}^*_I = \begin{bmatrix}-40 & 70 & 16\end{bmatrix}^\intercal$ and was moving with velocity $4.5\,\mathrm{m\,s^{-1}}$ towards the target following the pure pursuit guidance law. The scheme of the experiment is in subfigure 4.5c.

Results

The results are shown in Figure 4.8. As expected, the methods fusing the subtended angle outperformed the bearing-only approaches significantly, as in the previous scenario.

In a real-world application, if the initial estimate of the target is known e.g. from a radar, the observer can rely on the estimates provided by the "*dkft*" and "*plkft*" methods to

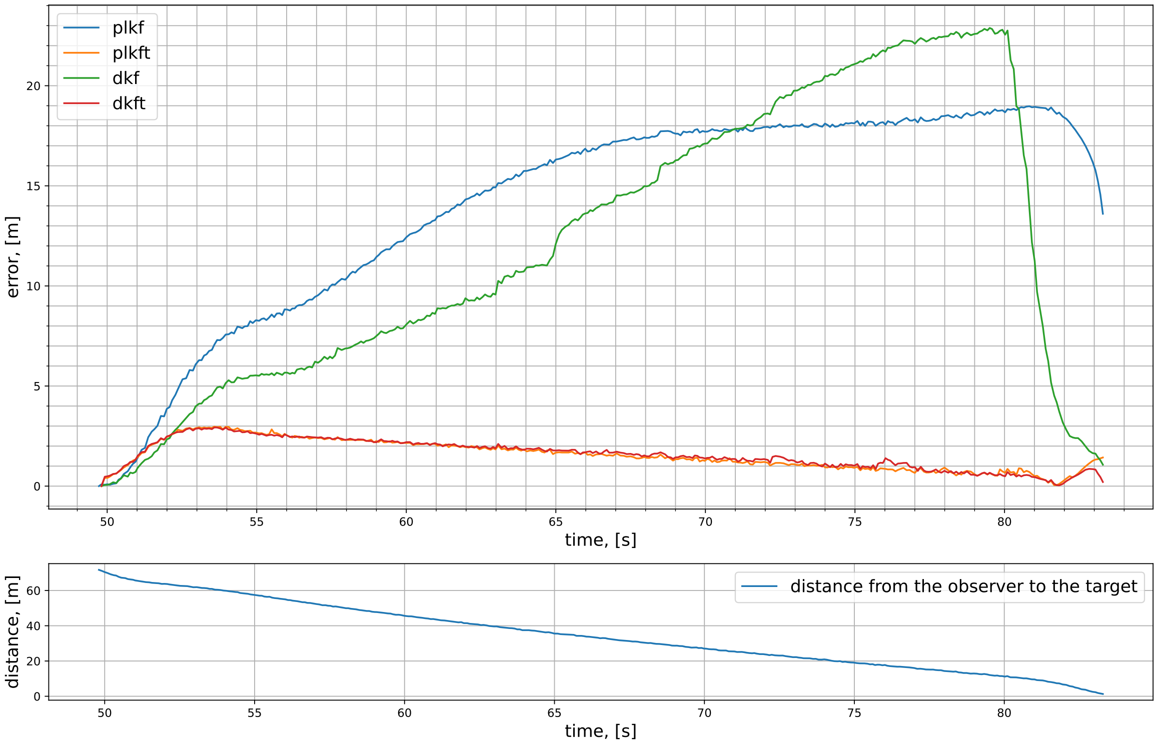


Figure 4.8: Results from scenario 3. The top plot shows the Euclidean distance between the estimated position and ground truth in meters. The bottom plot shows the distance between the observer and the target.

follow and catch the target by following the pure-pursuit guidance.

Even though the starting distance to the target in the second scenario is smaller than in this one, the maximum and average errors along the whole path for this experiment are smaller. The reason could be that the observer saw the target from more angles and gained more information about it in different linear subspaces, which is essential for both bearing-only approaches and those fused with the subtended angle.

Scenario 4

A helix shape of the observer's trajectory was chosen for the following scenario, shown in subfigure 4.5d as the observer's ground truth trajectory. The helix is a 3D figure generally defined with a set of equations

$$\begin{aligned} x(t) &= t, \\ y(t) &= \cos(t), \\ z(t) &= \sin(t). \end{aligned} \tag{4.9}$$

The helix trajectory of the observer is optimal for the bearing angle approaches to localize the target, as shown in article [3].

The target started at point $\boldsymbol{p}_T^* = \begin{bmatrix}33 & 0 & 4\end{bmatrix}^\intercal$, went up to reach height $z = 12$ and proceeded with the motion along the x axis with a constant velocity of $1\,\mathrm{m\,s^{-1}}$ towards the point $\begin{bmatrix}100 & 0 & 12\end{bmatrix}^\intercal$. The observer followed the helix trajectory with a constant velocity in the target's direction with velocity $1.5\,\mathrm{m\,s^{-1}}$. The radius of a helix was $5\,\mathrm{m}$, and the center of the circular part of the helix was at $\boldsymbol{c}_h = \begin{bmatrix}0 & 0 & 10\end{bmatrix}$. The scheme of the experiment is in subfigure 4.5d.

Results

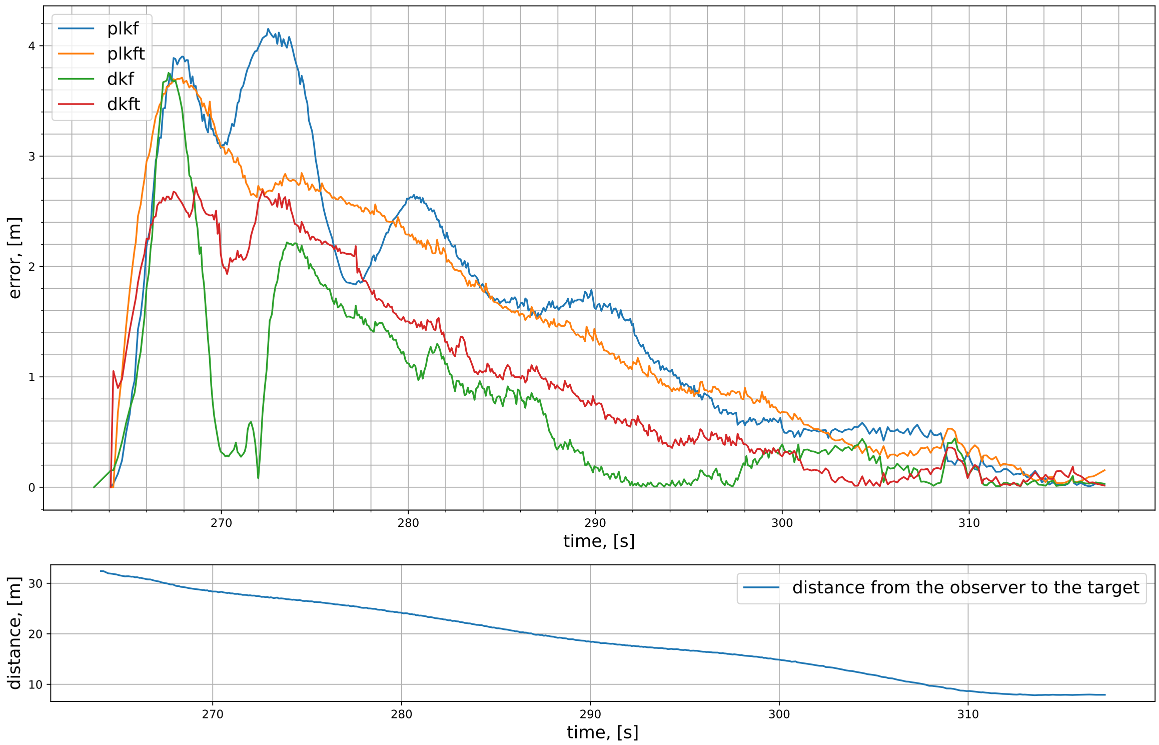


Figure 4.9: Results from scenario 4. The top plot shows the Euclidean distance between the estimated position and ground truth in meters. The bottom plot shows the distance between the observer and the target.

The results are shown in Figure 4.9. Using the helix trajectory, the observer saw the target from different angles and gained more information about it in a dimension orthogonal to the initial bearing vector, even from longer distances. The DKF-based approaches show better results here, probably since the visual tracking is almost perfect and the DKF relies more on it than uPLKF due to the parameters tuned for all scenarios. However, there was no significant domination of the approaches with subtended angle over bearing-only approaches. It can be concluded that this trajectory is optimal for the KF-based approaches to estimate the target's position, even without the fusion of the subtended angle, which improved the estimate in previous scenarios.

4.5 Evaluation of estimation precision with respect to noise

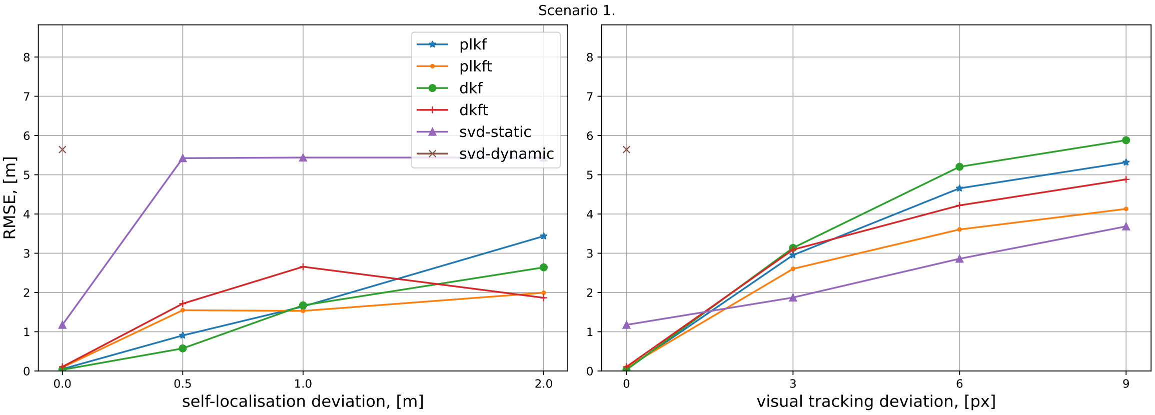


Figure 4.10: Scenario 1, the influence of self-localisation and tracking uncertainties on a 3D tracking.

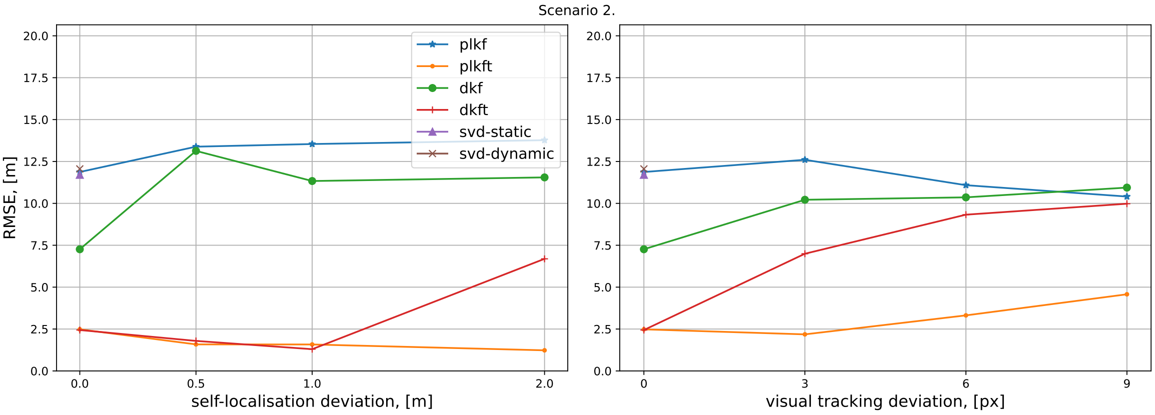


Figure 4.11: Scenario 2, the influence of self-localisation and tracking uncertainties on a 3D tracking.

The influence of visual tracking precision and the observer's self-localization quality is analyzed in this section. Gaussian noise with parameterized variance was added to the measurements as described at the beginning of the section to evaluate every approach with respect to noisy measurements and test the limits of the approaches. Even when visual tracking and the observer's state are given to the bearing vector estimator with a very small uncertainty caused by simulation noise and rounding the floating points to receive measurements in pixels for tracking, the results are imperfect, as shown in the previous section.

To analyze the localization precision regarding the observer's self-localization noise, σ_{pos} was chosen from a set $\sigma_{pos} \in \{0.5\,\text{m}, 1\,\text{m}, 2\,\text{m}\}$. To analyze the localization precision with respect to the visual tracking deviation, σ_{tr} was chosen from a set $\sigma_{tr} \in \{3\,\text{px}, 6\,\text{px}, 9\,\text{px}\}$. Each of the four scenarios described in the previous section was repeated with new param-

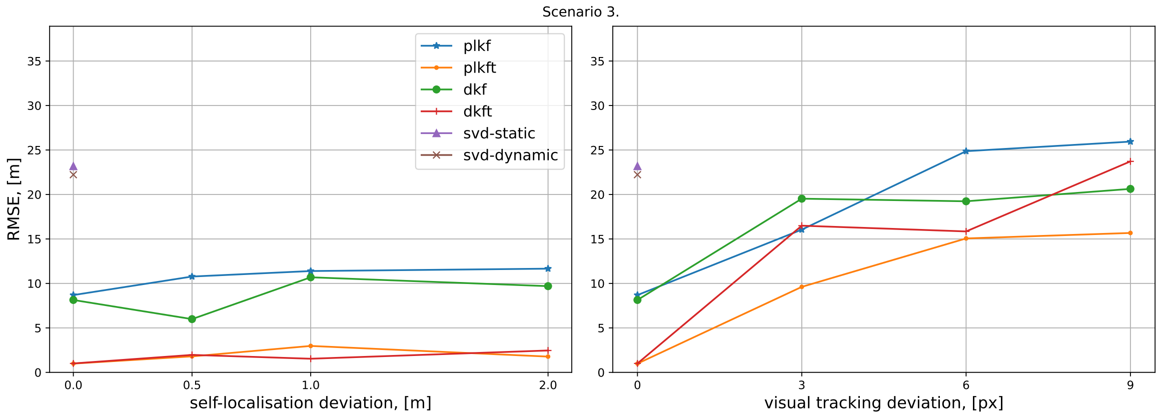


Figure 4.12: Scenario 3, the influence of self-localisation and tracking uncertainties on a 3D tracking.

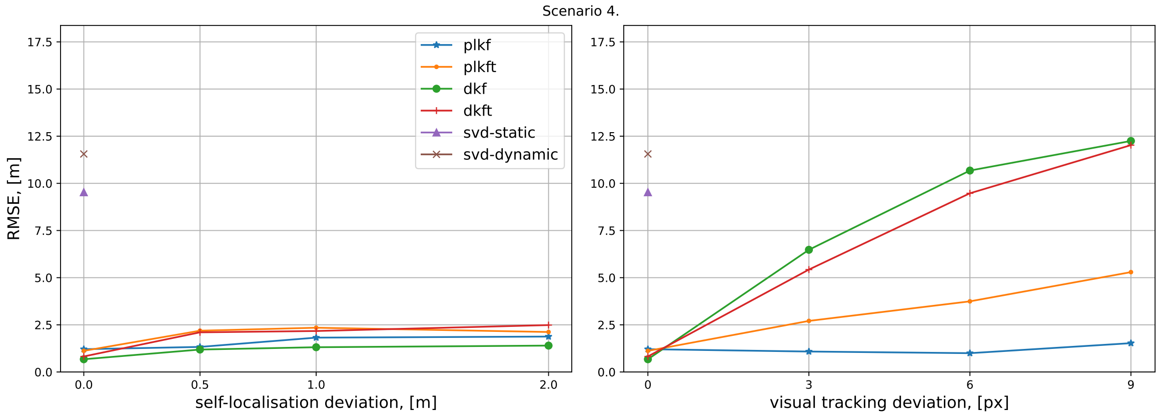


Figure 4.13: Scenario 4, the influence of self-localisation and tracking uncertainties on a 3D tracking.

eters. Results for scenarios 1, 2, 3, and 4 are shown in Figures 4.10, 4.11, 4.12, and 4.13, correspondingly.

	Scenario 1		**Scenario 2**		**Scenario 3**		**Scenario 4**	
	plkft	**dkft**	**plkft**	**dkft**	**plkft**	**dkft**	**plkft**	**dkft**
σ_{pos}, [m]	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
σ_{tr}, [px]	6	6	6	6	6	6	6	6
RMSE, [m]	3.5	7.2	3.7	7.3	10.8	14.8	5.4	7.6

Table 4.1: Results of the best-performed methods with noise added to visual tracking and observer's self-localisation.

In every scenario, the baseline methods "*svd-static*" and "*svd-dynamic*" showed the worst results even without noise added, so they were not profoundly analyzed and evaluated only for zero additional noise. Except for the first scenario, where "*svd-static*" overperformed the KF-based methods with the visual tracking deviation increased. The KF-based approaches were tuned to estimate the dynamics of the target, and the oscillations in detection were propagated through the model and led to more significant errors than the "*svd-static*" method formulated to estimate only the position of the static object.

In the third and fourth scenarios, the estimated state was barely affected by the noise caused by the observer's self-localization and showed stability and good performance. In contradiction, even a minor deviation of $\sigma_{tr} = 3\,\text{px}$ for the visual tracking caused a significant degradation of the estimate in every tested scenario for every approach. As expected, methods utilizing the subtended angle outperformed the bearing-only approaches in the first three scenarios.

Combined noise was also tested for the best-performing methods. In Table 4.1, the comparison of methods with the subtended angles is presented. The observer's self-localisation was set to 2 m, and the tracking variance to 6 px. The obtained RMSE values are not much bigger than the errors received with no deviation in localization precision. It supports the claim that the estimation precision is more influenced by deviation in the visual tracking than the observer's self-localization.

4.6 Real-world evaluation

Method	**Scenario**		
	1	**2**	**3**
plkf	5.12	6.90	2.57
plkft	2.67	1.72	0.85
dkf	4.09	3.16	1.55
dkft	0.75	0.90	0.66

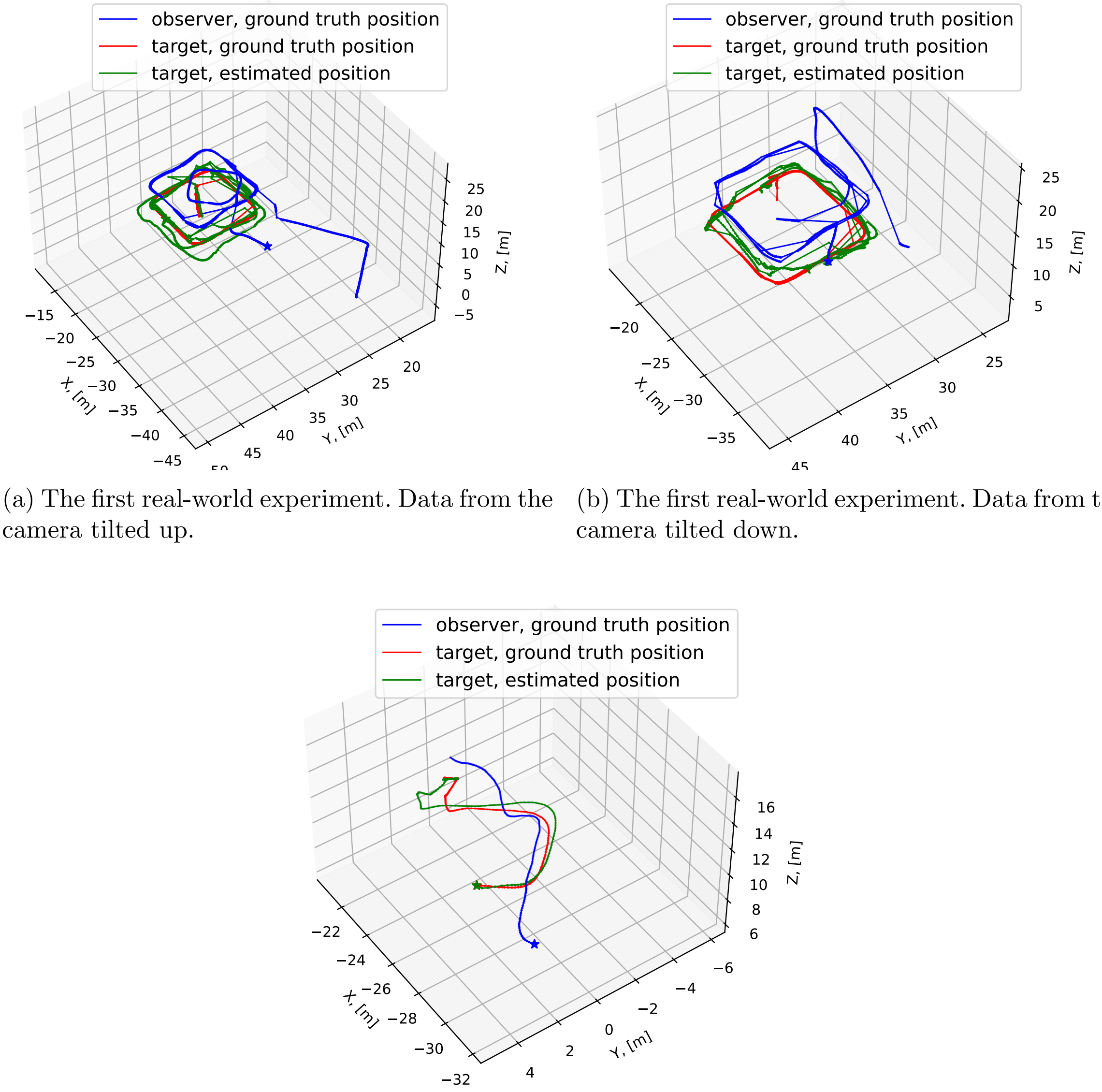
Table 4.2: RMSE comparison of the real world experiments. Values are in meters.

Setup for the real-world data collection is described in section 4.1. Data was recorded during the MRS experimental campaign, and every KF approach was tested on data recorded from two flights.

In the first flight, the target followed a periodic square-shape trajectory with a side length of 10 m, and the observer flew above the target for 140 s, and then both landed. In the second experiment, the target followed the same trajectory, but the observer performed an intercepting maneuver, after which the experiment ended. Only the up-tilted camera recorded the target.

The KFs were initialized with the ground-truth odometry of the target only when the target first appeared in the image. For the bearing-with-subtended-angle methods, the size of the object was also initialized with the ground truth size of the target. To evaluate the state estimation precision without the influence of visual tracking deviation, the projection of the target's real position onto the image plane of the camera with zero noise was used.

The trajectories of the observer and target, together with the estimated state, are presented in Figure 4.14. The comparison table of all KF-based methods evaluated on real-world data is in Table 4.2.



(a) The first real-world experiment. Data from the camera tilted up.

(b) The first real-world experiment. Data from the camera tilted down.

(c) The second real-world experiment. Data from the camera tilted up.

Figure 4.14: Trajectories during the real-world experiments. The estimated target position is visualized for the best-performing method for each experiment.

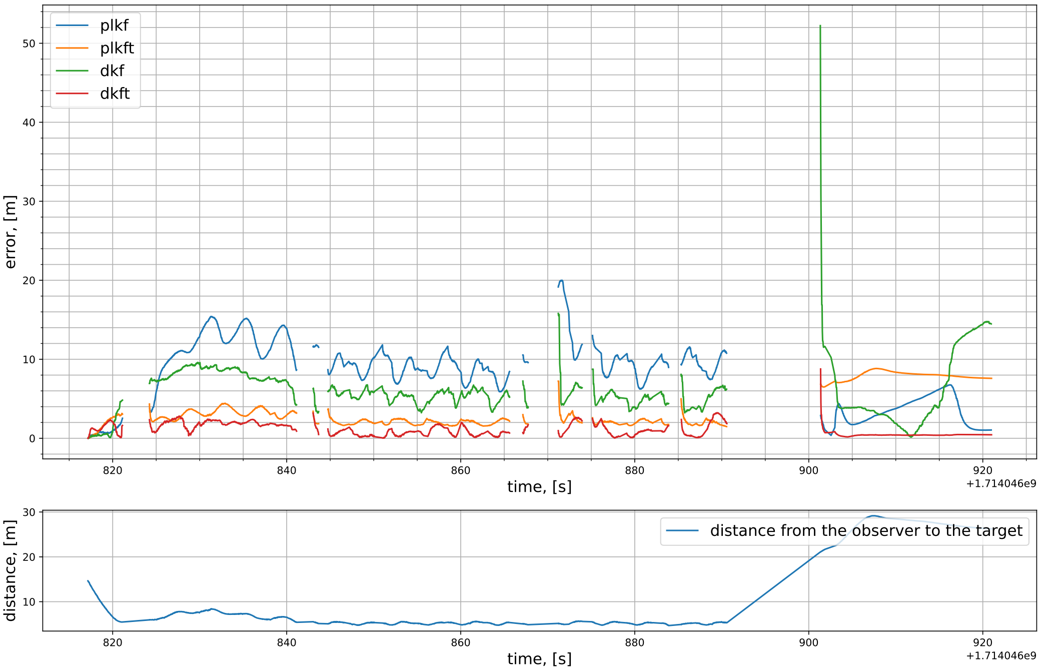


Figure 4.15: The results of the real-world experiment's first flight, the camera tilted up. The distance between the estimated and ground-truth position of the target is in the top plot. The bottom plot shows the distance between the observer and the target.

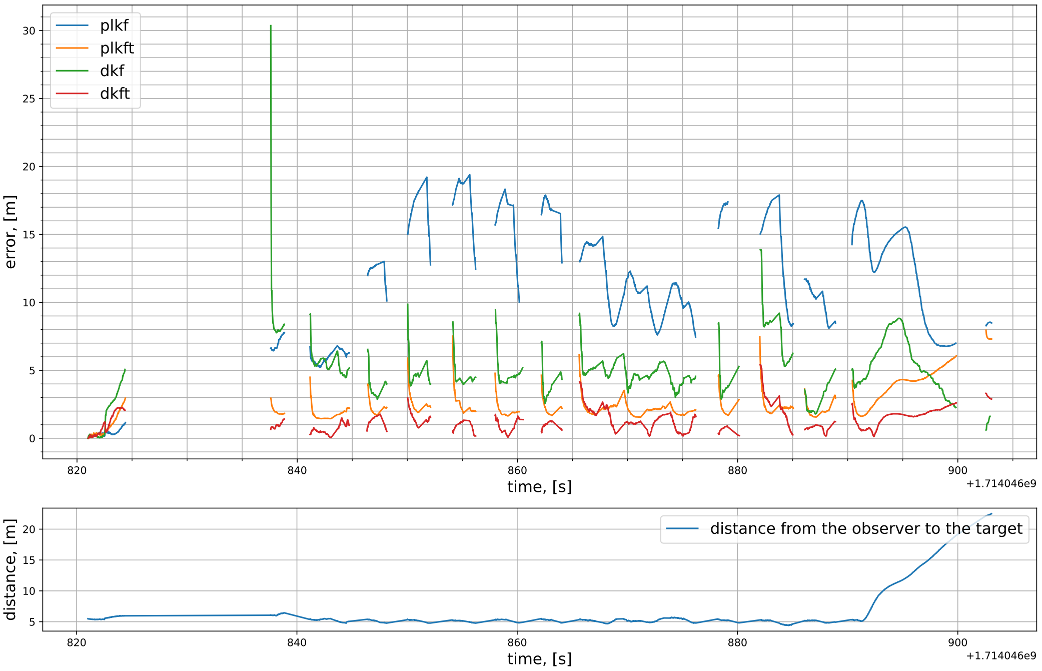


Figure 4.16: The results of the real-world experiment's first flight, the camera tilted down. The distance between the estimated and ground-truth position of the target is in the top plot. The bottom plot shows the distance between the observer and the target.

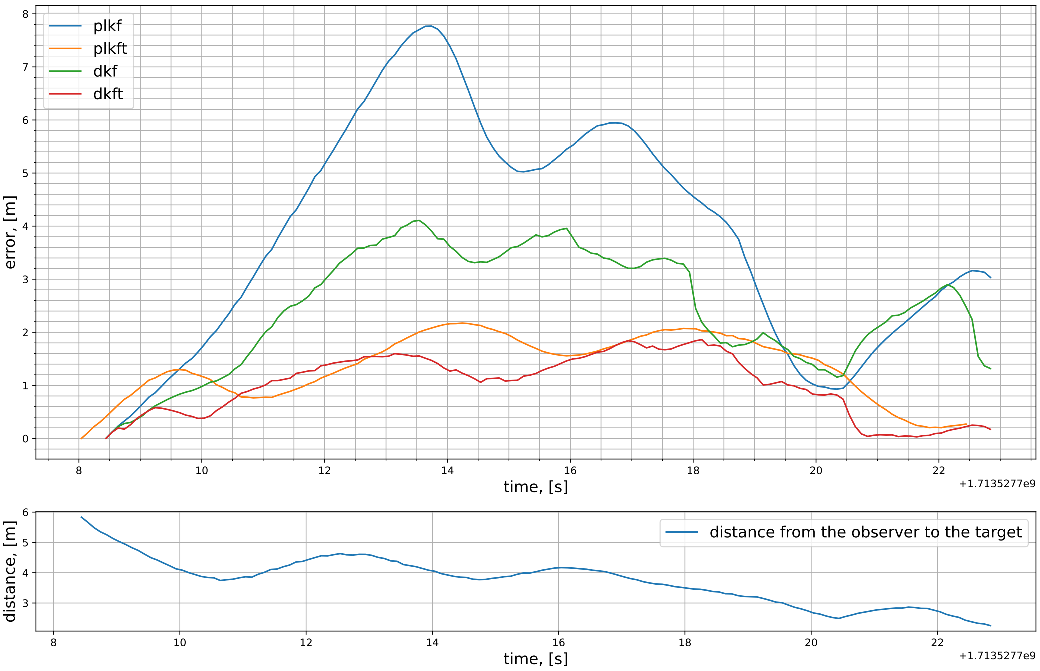


Figure 4.17: The results of the real-world experiment's second flight, the camera tilted up. The distance between the estimated and ground-truth position of the target is in the top plot. The bottom plot shows the distance between the observer and the target.

One of the limits of the implemented methods was discovered after analyzing the flight data. When the target and the observer follow the same trajectories with the same velocities, the position estimation is significantly degraded when the target rapidly changes the trajectory. Nevertheless, the subtended angle methods handled such situations and performed much better than the bearing-only methods. The comparison of Euclidean distances along the whole path are in Figures 4.15, 4.16 and 4.17. The plots are interrupted in places when the target was outside the camera's FOV.

The real-world experiments showed that the methods described in 3.4 can also be used for rapidly changing velocities and demonstrated the considered methods in a practical, real-world deployment.

Chapter 5

Conclusion and future work

Bearing-based position estimation of a moving target from a flying MAV equipped with a monocular camera, was investigated in this thesis. Multiple approaches based on the geometric intersection of lines and the Kalman Filters bearing-based algorithms were implemented and tested. The Pseudo Linear Kalman Filter and the Degenerate Kalman Filter were improved by adding a subtended angle measurement and compared to the bearing-only approaches. Multiple scenarios were analyzed in simulations, including pure pursuit guidance, simulating the process of intercepting the target by a drone interceptor, which is one of the intended applications. The subtended angle measurement fused with the bearing provides the KF with more information about the target in the dimension orthogonal to the bearing vector, improving the estimate and decreasing the covariance in the correspondent subspace. Visual tracking was also tackled, and the Median Flow tracker was tested in a real-world experiment, showing that it can be used as a source of measurement for the position estimation methods described.

Localization accuracy in relation to the visual tracking precision and the observer's self-localization precision was tested by running the same approach on the prerecorded data multiple times and adding the Gaussian noise with zero means and chosen deviation severally to the tracking precision and observer's self-localization. The experiments demonstrated the severe influence of the visual tracking precision on the target localization in 3D in all tested scenarios. Additional noise in the observer's self-localization also causes the estimate to deteriorate, but it is not even comparable to the one caused by tracking. To support and verify this conclusion, an additional test with mixed noise was performed. As a result, the estimated state along the path was not much worth it than when it was tested with the same deviation only for the tracking, proving that the tracking noise is much more destructive for the 3D localization using described KF methods.

Additionally, the trajectory proposed in [3] as optimal for the bearing-only position estimation was tested. There was no visible impact from adding the subtended angle in this scenario, and the tracking precision was barely influenced by the observer's self-localization. This supports the conclusion from [3] that the suggested trajectory is optimal for bearing-only position estimation. However, it is still affected by the tracking precision noise. It again highlights the importance of the visual tracking precision for the localization from bearing measurements.

The KF-based approaches were also evaluated on real-world data. Despite the target moving with a rapidly changing velocity vector which is not modeled in the KF state-space model, the KF-based methods with subtended angle estimated the position accurately. Although the uPLKF with subtended angle showed better results in simulation, the DKF with subtended angle outperformed it on real-world data, although not by a large margin.

In the future, the proposed pipeline can be extended to use multiple cameras covering a larger area around the MAV so that when one camera loses the target from its FOV, another camera will continue the tracking. Furthermore, the theoretical analysis of the target's trajectory on the estimation accuracy could be performed.

Chapter 6

References

[1] Z. Ning, Y. Zhang, J. Li, Z. Chen, and S. Zhao, "A bearing-angle approach for unknown target motion analysis based on visual measurements," *The International Journal of Robotics Research*, Feb. 2024, ISSN: 1741-3176. DOI: 10.1177/02783649241229172. [Online]. Available: http://dx.doi.org/10.1177/02783649241229172.

[2] M. Pliska, M. Vrba, T. Báča, and M. Saska, *Towards safe mid-air drone interception: Strategies for tracking and capture*, 2024. arXiv: 2405.13542 [cs.RO].

[3] J. Li, Z. Ning, S. He, C.-H. Lee, and S. Zhao, "Three-dimensional bearing-only target following via observability-enhanced helical guidance," *IEEE Transactions on Robotics*, vol. 39, no. 2, 1509–1526, Apr. 2023, ISSN: 1941-0468. DOI: 10.1109/tro.2022.3218268. [Online]. Available: http://dx.doi.org/10.1109/TRO.2022.3218268.

[4] F. Nekovář, J. Faigl, and M. Saska, "Multi-vehicle dynamic water surface monitoring," *IEEE Robotics and Automation Letters*, vol. 8, no. 10, 6323–6330, Oct. 2023, ISSN: 2377-3774. DOI: 10.1109/lra.2023.3304533. [Online]. Available: http://dx.doi.org/10.1109/LRA.2023.3304533.

[5] M. Petrlik, P. Petracek, V. Kratky, *et al.*, "UAVs Beneath the Surface: Cooperative Autonomy for Subterranean Search and Rescue in DARPA SubT," *Field Robotics*, vol. 3, pp. 1–68, Jan. 2023. DOI: https://doi.org/10.55417/fr.2023001.

[6] M. Vrba, V. Walter, V. Pritzl, *et al.*, "On onboard LiDAR-based flying object detection," preprint, arXiv 2303.05404, 2023. arXiv: 2303.05404 [cs.RO].

[7] M. Ali and S. Mandal, "Preparation of papers for ifac conferences; symposia: Kalman filter based control of inverted pendulum system," *IFAC-PapersOnLine*, vol. 55, no. 1, 58–63, 2022, ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2022.04.010. [Online]. Available: http://dx.doi.org/10.1016/j.ifacol.2022.04.010.

[8] T. Amukele, "Using drones to deliver blood products in rwanda," *The Lancet Global Health*, vol. 10, no. 4, e463–e464, Apr. 2022, ISSN: 2214-109X. DOI: 10.1016/s2214-109x(22)00095-x. [Online]. Available: http://dx.doi.org/10.1016/S2214-109X(22)00095-X.

[9] A. Diab, M. Sabry, and A. El Mougy, "Comparing monocular camera depth estimation models for real-time applications," in *Proceedings of the 14th International Conference on Agents and Artificial Intelligence*, SCITEPRESS - Science and Technology Publications, 2022. DOI: 10.5220/0010883700003116. [Online]. Available: http://dx.doi.org/10.5220/0010883700003116.

[10] A. Rejeb, A. Abdollahi, K. Rejeb, and H. Treiblmaier, "Drones in agriculture: A review and bibliometric analysis," *Computers and Electronics in Agriculture*, vol. 198, p. 107 017, Jul. 2022, ISSN: 0168-1699. DOI: 10.1016/j.compag.2022.107017. [Online]. Available: http://dx.doi.org/10.1016/j.compag.2022.107017.

[11] T. Baca, M. Petrlik, M. Vrba, *et al.*, “The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles,” *Journal of Intelligent & Robotic Systems*, vol. 102, no. 26, pp. 1–28, 1 May 2021. DOI: 10.1007/s10846-021-01383-5. [Online]. Available: https://link.springer.com/article/10.1007/s10846-021-01383-5.

[12] V. Borsuk, R. Vei, O. Kupyn, T. Martyniuk, I. Krashenyi, and J. Matas, *Fear: Fast, efficient, accurate and robust visual tracker*, 2021. DOI: 10.48550/ARXIV.2112.07957. [Online]. Available: https://arxiv.org/abs/2112.07957.

[13] Z. Huang, S. Chen, C. Hao, and D. Orlando, “Bearings-only target tracking with an unbiased pseudo-linear kalman filter,” *Remote. Sens.*, vol. 13, p. 2915, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:237734340.

[14] V. Pritzl, P. Stepan, and M. Saska, “Autonomous flying into buildings in a firefighting scenario,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2021. DOI: 10.1109/icra48506.2021.9560789. [Online]. Available: http://dx.doi.org/10.1109/ICRA48506.2021.9560789.

[15] B. Yan, H. Peng, J. Fu, D. Wang, and H. Lu, *Learning spatio-temporal transformer for visual tracking*, 2021. DOI: 10.48550/ARXIV.2103.17154. [Online]. Available: https://arxiv.org/abs/2103.17154.

[16] V. Walter, M. Vrba, and M. Saska, “On training datasets for machine learning-based visual relative localization of micro-scale UAVs,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 10 674–10 680.

[17] G. Morani, F. Nebula, F. Corraro, and M. Ariola, “Dynamic control allocation through kalman filtering,” *American Journal of Engineering and Applied Sciences*, vol. 12, no. 1, 46–56, Jan. 2019, ISSN: 1941-7020. DOI: 10.3844/ajeassp.2019.46.56. [Online]. Available: http://dx.doi.org/10.3844/ajeassp.2019.46.56.

[18] K. Saho, “Kalman filter for moving object tracking: Performance analysis and filter design,” in *Kalman Filters - Theory for Advanced Applications*. InTech, Feb. 2018. DOI: 10.5772/intechopen.71731. [Online]. Available: http://dx.doi.org/10.5772/intechopen.71731.

[19] O. Özyeşil, V. Voroninski, R. Basri, and A. Singer, “A survey of structure from motion.,” *Acta Numerica*, vol. 26, 305–364, May 2017, ISSN: 1474-0508. DOI: 10.1017/s096249291700006x. [Online]. Available: http://dx.doi.org/10.1017/S096249291700006X.

[20] Z. Kalal, K. Mikolajczyk, and J. Matas, “Tracking-learning-detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, 1409–1422, Jul. 2012, ISSN: 2160-9292. DOI: 10.1109/tpami.2011.239. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2011.239.

[21] D. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, “Visual object tracking using adaptive correlation filters,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2010. DOI: 10.1109/cvpr.2010.5539960. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2010.5539960.

[22] Z. Kalal, K. Mikolajczyk, and J. Matas, “Forward-backward error: Automatic detection of tracking failures,” *2010 20th International Conference on Pattern Recognition*, pp. 2756–2759, 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:13175840.

[23] J. Ali and M. Ushaq, “A consistent and robust kalman filter design for in-motion alignment of inertial navigation system,” *Measurement*, vol. 42, no. 4, 577–582, May 2009, ISSN: 0263-2241. DOI: 10.1016/j.measurement.2008.10.002. [Online]. Available: http://dx.doi.org/10.1016/j.measurement.2008.10.002.

[24] M. Deans, “Bearings-only localization and mapping,” Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, Sep. 2005.

[25] X. Lin, T. Kirubarajan, Y. Bar-Shalom, and S. Maskell, “Comparison of ekf, pseudomeasurement, and particle filters for a bearing-only target tracking problem,” in *Signal and Data Processing of Small Targets 2002*, O. E. Drummond, Ed., SPIE, Aug. 2002. DOI: 10.1117/12.478508. [Online]. Available: http://dx.doi.org/10.1117/12.478508.

[26] R. Marslin, G. D. Sullivan, and K. D. Baker, “Kalman filters in constrained model based tracking,” in *BMVC91*. Springer London, 1991, 371–374, ISBN: 9781447119210. DOI: 10.1007/978-1-4471-1921-0_53. [Online]. Available: http://dx.doi.org/10.1007/978-1-4471-1921-0_53.

[27] C. Tomasi and T. Kanade, “Shape and motion from image streams: A factorization method - part 3 detection and tracking of point features,” Pittsburgh, PA, Tech. Rep. CMU-CS-91-132, 1991.

[28] E. Fogel and M. Gavish, “Nth-order dynamics target observability from angle measurements,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 24, no. 3, 305–308, May 1988, ISSN: 0018-9251. DOI: 10.1109/7.192098. [Online]. Available: http://dx.doi.org/10.1109/7.192098.