EuroGNC Navigation Guidance Control CEAS Bristol, UK June 11th-13th 2024
University of BRISTOL ROYAL AERONAUTICAL SOCIETY AIAA

Robot Assisted Landing Process of small UAVs using Decentralized Kalman Filter

Vincent Konnow Research Assistant, FH Aachen, Faculty of Aerospace Engineering, 52064 Aachen, Germany. konnow@fh-aachen.de

Jannes Terlau Student Assistant, FH Aachen, Faculty of Aerospace Engineering, 52064 Aachen, Germany. jannes.terlau@alumni.fh-aachen.de

Lukas Hildebrand Research Assistant, FH Aachen, Faculty of Aerospace Engineering, 52064 Aachen, Germany. hildebrand@fh-aachen.de

Philipp Hartmann Professor, FH Aachen, Faculty of Aerospace Engineering, 52064 Aachen, Germany. p.hartmann@fh-aachen.de

ABSTRACT

This paper presents a new approach on robot-assisted landing of a small unmanned tilt-wing aircraft using a serial kinematic industrial robot arm to directly catch the aircraft in hover flight. The goal is to automate the landing and turnaround of the aircraft without the need for a large landing area and additional personnel. For this purpose, a base station is presented that integrates the robot arm along with other required hardware. An optical tracking system is used to acquire the aircraft pose. The catching process itself is formalized in three successive phases, each with its own set of constraints and tolerances. A direct tracking method for executing the catching process is proposed and evaluated through a real-time pose tracking setup with the robotic arm. The results indicate the necessity of a filtered setpoint pose as well as a short term prediction to overcome delays in the robot controller. Therefore, a decentralized Kalman filter is developed to obtain a robust estimation of the aircraft pose by incorporating measurements from a ground-based optical tracking system as well as measurements from the inertial measurement unit streamed directly from the aircraft. In addition, an outline on further work for robot trajectory generation is given.

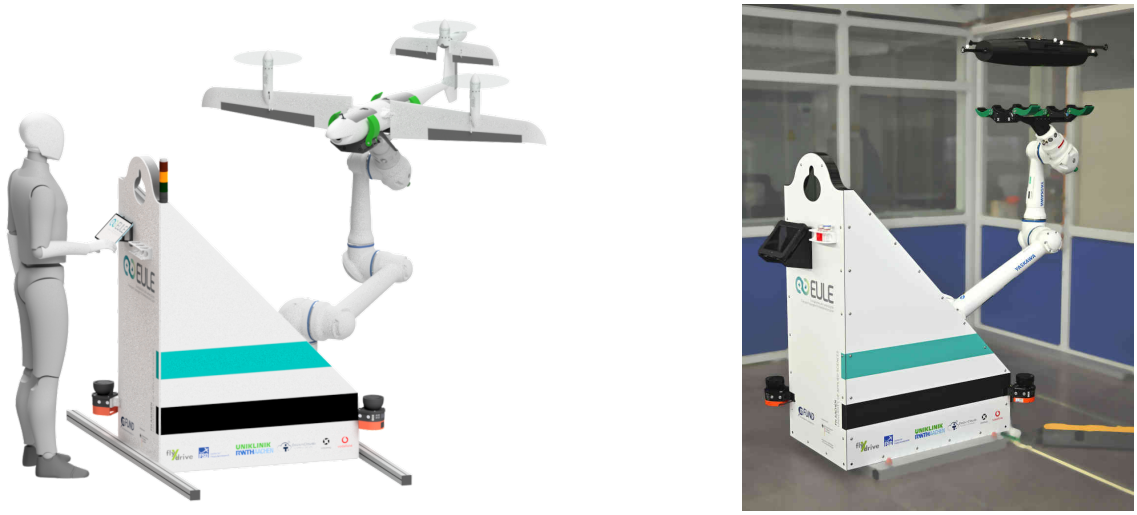
Keywords: Automated UAV Turnaround; Robot based Landing, Trajectory Optimization

Nomenclature

ϕ, θ, ψ	=	Euler angles (roll, pitch, yaw)
p, q, r	=	Body-fixed angular velocities in x-, y-, z-direction.
u, v, w	=	Body-fixed linear velocities in x-, y-, z-direction.
a_x, a_y, a_z	=	Body-fixed linear acceleration in x-, y-, z-direction.
d_{ac}	=	Distance to the aircraft's setpoint
T	=	Robot task space (Cartesian space in which the aircraft is caught)

1 Introduction

The EULE research project focuses on improving medical care for patients by enhancing the exchange of supplies between medical facilities using mid-range tilt-wing UAVs. The goal is to connect hospitals, laboratories and pharmaceutical wholesalers to transport medical supplies such as tissue samples or medications by air. This is faster and more efficient than current road transportation. The aircraft in use is a tilt-wing VTOL flight system with a wingspan of 1.8 m and a takeoff weight of 7.5 kg. A robotic base station, hereafter referred to as *medPort* acts as a link between the medical personnel and the flight system. To enable an automated turnaround process, the aircraft is caught and handled by a robotic arm and subsequently presented to the personnel for loading or unloading the payload.
The *medPort* base station is designed to catch the unmanned tilt-wing aircraft, provide end users with access to the payload, and move the aircraft in a suitable takeoff pose. For the commission assignment, it is integrated into a novel medical distribution network developed by a research partner. Figure 1 shows a rendering of the station side by side with the experimental setup. It has a footprint of 0.6 m × 1 m.



(a) Rendering **(b) Experimental setup**

Fig. 1 *medPort* station

It incorporates a 6 degree of freedom (DOF) industrial robot with serial kinematics for catching and handling the aircraft. To ensure safe operation of the robot, two safety laser scanners are placed on opposite edges of the *medPort*, providing complete coverage of the surrounding area. A custom gripper is attached to the robot's end effector for the purpose of catching and handling the aircraft. To capture the pose of the aircraft in real time, an optical tracking system is employed. The acquired pose signal is combined with live data from the aircraft's inertial measurement unit (IMU) in a decentralized Kalman filter to provide a robust pose estimation. Finally, different trajectory generation methods are discussed.

1.1 Robot based landing

Landing is one of the most critical phases in aircraft operation. This applies to both manned and unmanned aviation. In conventional operation, the given tilt-wing aircraft requires an extended landing platform to compensate for the deviation between commanded and real landing position. The deviation is mainly caused by GNSS errors and turbulent wind components (gustiness). GNSS errors are mitigated by using a real-time kinematic (RTK) base station. However, it still requires a 3 m × 3 m platform to land safely [1]. In addition, trained personnel are needed to perform the aircraft turnaround. A minimum turnaround consists of charging the flight batteries and exchanging the payload. Automating both the landing and turnaround process with a robot can result in safer aircraft operations, faster turnarounds, and high availability.

1.2 Related Work

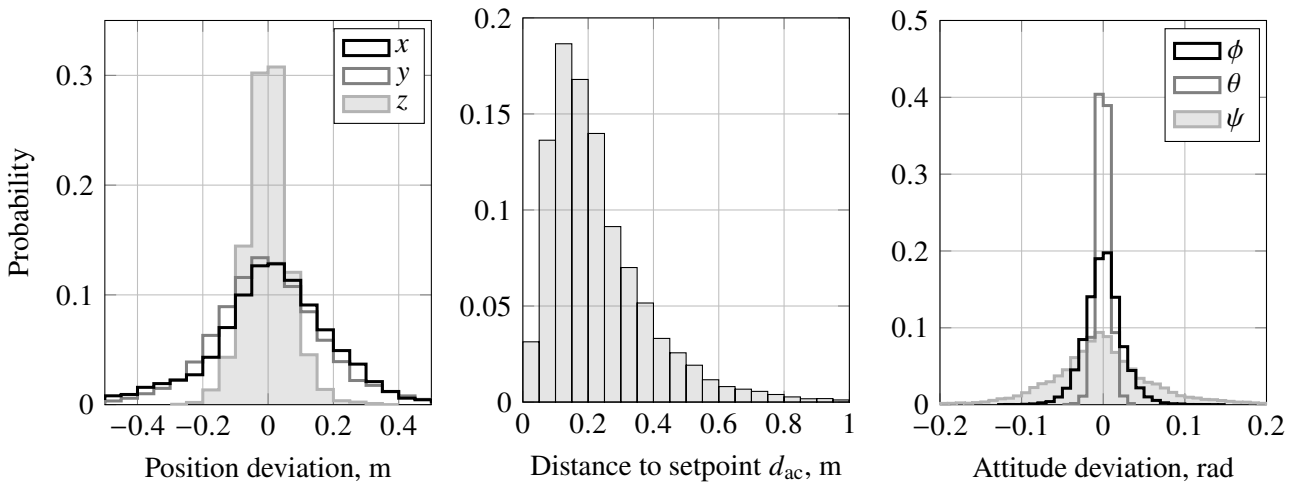
The idea of (semi) automated UAV operation has been explored in previous research and is part of existing business models. In a prior research project, a base station was built for a predecessor of the given UAV [1]. The handling and communication systems used were installed in a $4.5\,\mathrm{m} \times 4.5\,\mathrm{m}$ DHL parcel station. The UAV was positioned for battery and payload changes using a sliding mechanism on the platform. Other conceptually comparable base station are presented for geographic monitoring [2] and for medical delivery services by the company Matternet [3]. These stations are designed for long-term use and, once set up, their size impedes further portability.
There is a wide range of research being done on methods to assist UAVs in landing, especially for marine applications. In order to reduce the size of the required landing platform, a tether can be used as a connection with adjustable tension [4]. A recent publication demonstrates the use of a manipulator-based assistance system to mitigate the disturbances introduced by unknown sea conditions and to enable automated UAV operation [5]. However, this would require the addition of a winch system to the given aircraft and customized flight controllers, as a tether would affect the dynamics of the aircraft.
Maier et al. published four contributions on robot-assisted UAV landing, focusing on cooperative landing maneuvers [6–9]. The catch and touchdown phases are not considered.
In 2022 the company Quantum-Systems developed a comparable robot-based UAV landing system as part of the research project FreeRail [10]. The project focuses on autonomous monitoring rather than payload delivery, along with a different type of VTOL and catching technique.

2 Fundamentals

Of all the intended functions performed by the *medPort*, the landing process is the most challenging. It requires an immediate and continuous stream of input data from the installed sensors to reliably capture and predict the position of the aircraft. In addition, a real-time system architecture is required to acquire the input data, execute the control algorithms, and output the robot joint commands. The control process should plan ahead to catch the aircraft while taking into account the robot's constraints. The following section is therefore divided into 1. the requirements based on the analysis of the aircraft's motion in hover flight, 2. the description of the catching process, and 3. the key components of the system for carrying out the process. In addition to the catching process, the *medPort* performs aircraft takeoff and preparation for payload retrieval. However, this paper will focus on the catching process as it is the most challenging part.

2.1 Aircraft motion analysis

While approaching its destination, the given tilt-wing aircraft transitions from cruise to hover flight by tilting its wing and tailplane. Evaluations of test flights show a varying position deviation of the hovering aircraft from the target position. To assess the impact, an analysis of the aircraft's hover flight behaviour was performed. The analysis is based on a dataset provided by a project partner of nine flights in hover state with a total flight time of 49 minutes and a mean wind speed of up to $5.8\,\mathrm{m/s}$.
Figure 2a shows a histogram of the aircraft's control deviation. Based on the spread of the different distributions, it can be concluded that the volume of equal probabilities is of oblate ellipsoidal shape. The X and Y axes are the major axes and form approximately a circular cross section. Z is the minor axis with a ratio of 0.5. This means that the aircraft's deviation from the commanded setpoint in the horizontal direction will be twice as large as the deviation in the vertical direction.
The distance distribution to the aircraft's setpoint is displayed in Fig. 2b. In hover flight with fixed setpoint, the cumulative probability of the aircraft being in the 0.5 m range equals to $P(d_{\mathrm{ac}} \leq 0.5\,\mathrm{m}) = 0.93$. As a first approximation, the Cartesian space in which the aircraft is to be captured can be modeled as an oblate ellipsoid with a major cross section of 1 m.



(a) Position deviation **(b) Setpoint distance** **(c) Attitude deviation**

Fig. 2 Aircraft control deviation

A histogram of the deviation from the commanded attitude is shown in Fig. 2c. A clear gradation in the standard deviation can be seen. For the three angles, the standard deviation is given with $\pm 3\sigma_\phi = 0.07\,\text{rad}$ ($\approx 4.0\,\text{deg}$), $\pm 3\sigma_\theta = 0.03\,\text{rad}$ ($\approx 1.7\,\text{deg}$) and $\pm 3\sigma_\psi = 0.31\,\text{rad}$ ($\approx 17.8\,\text{deg}$). As a result, the deviation of attitude is most significant for the yaw angle. Figure 3 plots aircraft presence events against the distance to its setpoint. This should provide an indication of the required robot task space T. The time axis is plotted logarithmically. A first observation shows that as the range increases, the presence event time lengthens as well. For example, the aircraft stayed within the range of 0.8 m to its setpoint for a maximum of 1101.8 s, but only for a most of 14.4 s within the range of 0.1 m. A notable finding: even if the cumulative probability of presence in the 0.5 m range is $P(d_{ac} \leq 0.5\,\text{m}) = 0.93$, the event time of presence is spread between 1 s to 300 s. For a time span of more than 5 s, the cumulative probability is $P(t_{set} > 5\,\text{s}) = 0.61$. Therefore, to estimate a sufficient presence event time, the aircraft's velocity vector should be considered.

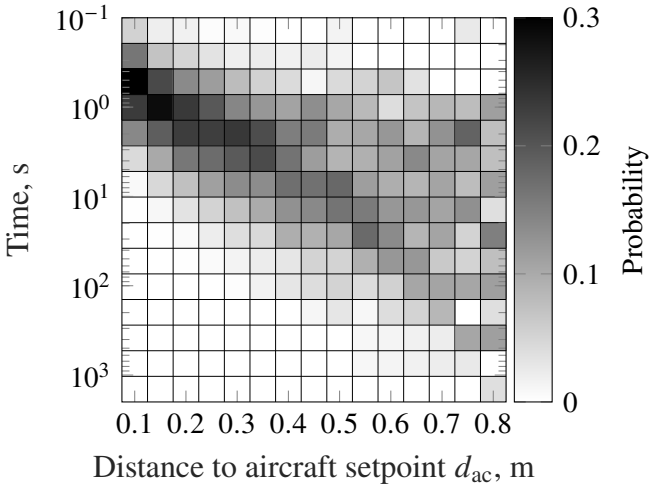


Fig. 3 Aircraft presence events

An analysis of the velocity and acceleration profile suggests a maximum velocity magnitude of roughly 1 m/s and a maximum acceleration magnitude of about $3.5\,\text{m/s}^2$. According to the manufacturer, the aircraft moves at translational speeds of up to 1.5 m/s during hovering phases with a constant setpoint and stronger wind conditions. At its rated limit, the aircraft is operated at mean wind speeds of up to 10 m/s. An additional data set of flights in these wind conditions would be needed to get a more accurate estimate of the maximum velocity and acceleration values.

The parameters outlined above provide the requirements for the capture process described in the following section.

2.2 Catching process

The catching process takes place inside the robot's task space. The task space T is a subset of the robot's workspace, in which the aircraft can be captured and decelerated without colliding with the station and without exceeding the kinematic limits of the robot. The catching process can generally be divided into three phases, which are listed below.

Phase 1: Synchronisation with the aircraft. As the aircraft approaches, the robot manipulator positions its tool center point (TCP) at the intersection of the task space boundary and the estimated trajectory of the aircraft. When the aircraft is entering the task space, the robot accelerates and synchronizes its TCP position with the catch position p_c. The position p_c is defined as the current aircraft position p_ac with a vertical offset, so the gripper remains below the aircraft.

Phase 2: Capture motion. Based on a target function F_t, a trajectory is computed and executed that captures the aircraft. The input parameters of F_t are primarily the aircraft's estimated pose and the remaining space in T for decelerating the aircraft. F_t returns a success probability for capturing the aircraft at the current time instance.

Phase 3: Deceleration. The deceleration has to be performed within T and without exceeding the maximum permissible forces of the aircraft and the robot. Different techniques can be used to generate the successive trajectories. This paper will focus on two different approaches described in Section 4 and Section 5.

The catching process described above is subject to a number of limitations imposed by the robot and the aircraft.

Restrictions imposed by the robot

- All trajectories executed must be within T to avoid damage to the station and to avoid reaching the end of the robot's range of motion.
- The task space T should preferably be structured in such a way that the potential for robot singularities is reduced or eliminated.
- The robot's maximum end-effector velocity and acceleration must be taken into account in order to obtain undelayed trajectory execution.

Restrictions imposed by the aircraft

- For all capture phases, sufficiently accurate pose detection is essential to catch in the right place and avoid damage to the aircraft. The required accuracy depends on the non-uniform gripping tolerances and the gripper design.
- As a result of the aircraft motion analysis, the pose of the aircraft needs to be estimated for a short period of time to avoid presence events that are unsuitable for catching.
- In **Phase 3**, the maximum permitted force acting on the fuselage must not be exceeded. The maximum values are axis-dependent.

In the following, the *medPort* components will be explained in relation to the restrictions and limitations mentioned above.

2.3 Key Components

From sensors to processing to actuators, the individual components of the *medPort* are explained. In Fig. 4 the system architecture is shown. All component connections are labeled with their communication protocol. Gray shaded parts are not part of the *medPort*. The gripper is physically connected to the robot, but has its own communication link. A real-time data interface is indicated by a bold line width.

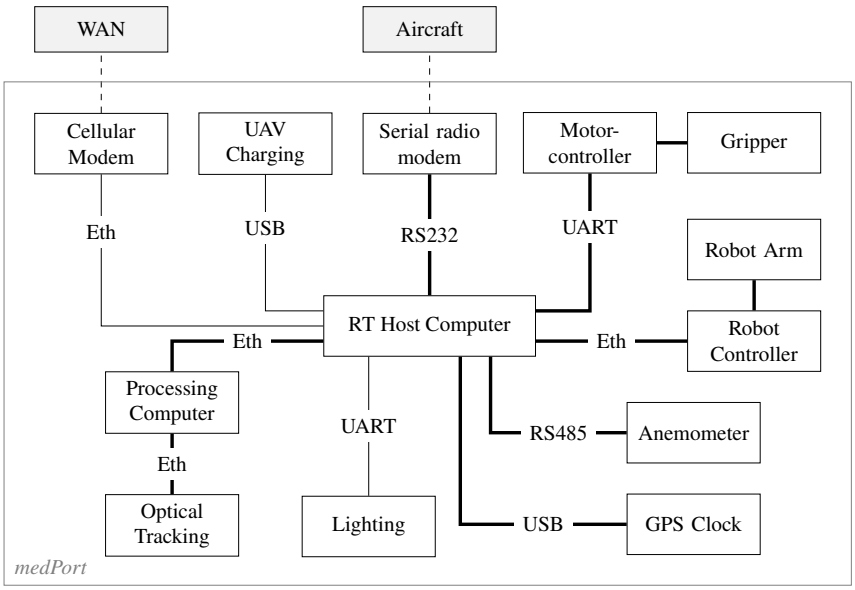


Fig. 4 ***medPort*** **system architecture**

2.3.1 Pose tracking

Several methods can be employed to obtain real-time tracking of the pose of the aircraft at hand. Previous research has utilized vision-based tracking using fiducial markers affixed to the wing [10]. However, this approach is not feasible for the tilt-wing aircraft given, since the wing is tilted vertically during hover flight. Additionally, the effectiveness of this tracking approach depends on suitable lighting conditions, which impedes continuous operation at night times. Other possible methods for tracking include the use of lidar, stereo cameras, structured light projection, and more.

As a robust and readily available solution, an optical tracking system using passive reflective infrared markers was chosen, often simply referred to as a motion capture system. In particular, four Qualisys Arqus A5 cameras are positioned on a regular square around the *medPort*, with each edge measuring 4 m. The processing computer shown in Fig. 4 is a barebone system required to execute the evaluation software Qualisys Track Manager (QTM). The manufacturer claims an accuracy of 0.06 mm for a distance up to 10 m, a low-latency streaming interface of detected object poses, as well as sun filters and active filtering for outdoor measurements. Accuracy and ability to function outdoors were assessed. Reflective foil stickers were tested first because they do not interfere with the aerodynamics of the aircraft. The stickers applied to the fuselage of the aircraft have demonstrated to be insufficient due to the reflective gloss varnish on the aircraft's surface. As a result, a significant number of erroneous marker detections occurred. Furthermore, as the fuselage rotates, the visible surface area of the planar foil stickers decreases, which hinders detection by the evaluation software. Small spherical markers with a diameter of 8 mm provide satisfactory results. Ten markers are applied in a randomized pattern on the fuselage to ensure safe detection. The randomized pattern eliminates ambiguities in the object detection.

Due to potential false detections or markers being invisible during live tracking, a backup solution has been developed to address detection gaps in the measurement method. The details are presented in Section 3.

2.3.2 Real-time operating system

To read sensor data, execute the control loops and write commands to the robot deterministically, a real-time operating system is needed. The main target is the robot, demading a jitter-free stream of joint commands at a rate of 250 Hz. The industrial embedded PC runs on the Ubuntu 22.04 Linux distribution, which is using a kernel patched with the PREEMPT_RT patchset for real-time support. As an alternative, a proprietary real-time operating system such as Wind River's VxWorks can be used.

The Robot Operating System (ROS) second generation is used as the main framework for implementing algorithms and device drivers and visualizing the results [11]. In ROS2, the system architecture is built around nodes, which are processes that perform computations. Nodes communicate with each other by publishing messages to topics, sending service requests and responses, or using actions for long-running tasks. ROS2 provides abstraction over DDS implementations, allowing users to choose from different DDS vendors to best suit the needs of their application without changing code. Although ROS2 has been designed to enable real-time behavior, special configuration and testing is required to achieve bounded latencies for node execution and inter-node communication. On a modern computer, multiple software and hardware components can introduce unexpected latency. Prominent examples include system management interrupts, dynamic frequency scaling based on power and temperature, memory latency, and more. In addition to the patchset, there are several measures that can be used to achieve bounded latency. The topic is part of ongoing research and continuous improvement. The following steps are suggested in [12, 13]:

- Prevent dynamic memory allocation to avoid page faults. This can be done for the executing thread of a node as well as for the entire DDS communication stack.
- Prioritizing the executing thread with $P > 49$. To get a reproducable latency, every involved thread needs to get an elevated priority.
- Optional: Shield CPU cores. Unlike isolating cores, shielding keeps essential core threads running and load balancing still works.

As an alternative to the distributed node-based approach, ROS2-Control can be used to implement control loops with chained controllers in a single thread, eliminating the need for communication middleware. The implementation described in Section 4 follows this approach.

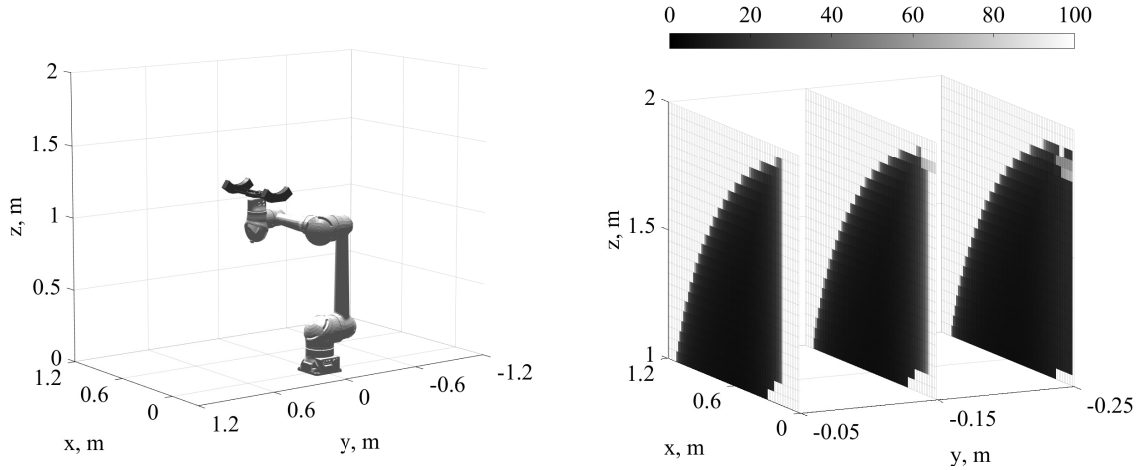
2.3.3 Robot

The *medPort* uses a 6-axis serial kinematic robotic arm, often referred to as an industrial robot (IR). Specifically, a Yaskawa HC10 DTP combined with a Yaskawa YRC1000 micro robot controller. The robot has a spherical workspace with a radius of 1.37 m (without a gripper attached) and a payload of 10 kg. It consists of a serial chain of links connected by six actuated joints. Unlike other types of robots, a 6-DOF serial kinematic robot has a larger workspace and high manipulability. This enables not only a large task space for catching, but also a consecutive handling process to execute the turnaround.
Compared to landing systems based on actuated platforms (such as hexapods), a manipulator arm is capable of catching the aircraft in flight rather than requiring a touchdown on the landing gear. This eliminates the impact of the aircraft on the landing surface and allows direct fixation for further handling compared to additional platform-based displacement systems such as those used in [1–3]. Compared to the previous manipulator-based catching, where the robot is placed under the aircraft while the aircraft is lowered on the gripper [10], the proposed method allows a continuous hovering within the catching task space. As a result, the method is more robust to misalignment as the aircraft is only a passive part of the process.
The given robot is primarily used for automation tasks with predefined trajectories. At the time of writing, there is no publicly available software interface to stream a series of joint setpoints (trajectory) to the controller in real-time. Therefore, an interface has been created that allows direct setpoint input to each axis velocity controller as well as fault tolerant behavior. The interface includes the following features and limitations:

- Setpoints must be sent at 250 Hz, the interpolation rate of the controller.
- Each setpoint is executed with a dead time of 40 ms and additionally delayed by each axis maximum acceleration. The maximum acceleration is dependent on the robots payload and parameterization. With the given setup, the robot's acceleration limit introduces an additional delay of up to 50 ms.
- Custom limits can be set for the maximum velocity and acceleration of each axis.

- Fault tolerant behavior is achieved by an algorithm that synchronizes each axis live to the given trajectory. This allows each axis to have its own recovery behavior, for all types of errors ranging from a missing setpoint to a complete connection loss.

The code runs as a task within the robot controller's real-time operating system and is publicly available as open source [14]. Further improvements are planned to limit the jerk of each axis as well.
The robot must meet the necessary speed and acceleration requirements. It is specified by the manufacturer for linear speeds up to 2 m/s. As the robot consists of a chain of links connected by actuated joints, each joint contributes differently to the path of the robot as it moves in a straight line in Cartesian space. If all joints contribute, the velocity in Cartesian space is maximized according to the manufacturer's values. A motion that is primarily performed by one joint is limited in speed. If these limitations in joint space are not considered in Cartesian space, the outcome is a divergence between the commanded and executed trajectory. The issue can be addressed by establishing cautious boundaries or by examining the task space T for potential speed limitations. The task space T is defined in Cartesian space with an initial pose P_{T_0} and a corresponding robot configuration type in which the joints of the robot remain angled to avoid collinearities (see Fig. 5a). This measure is taken to avoid robot singularities. The robot configuration type is defined by the arrangement of the joint axes to each other. In order to assure singularity avoidance during operation, the configuration type can be maintained by restricting joint space movement.



(a) Initial pose P_{T_0} **(b) Condition number in task space T**

Fig. 5 Task space visualization

To investigate speed limitations and possible singularities, the workspace T can be analyzed by evaluating the condition number of the robot's Jacobian matrix for the gripper orientation of P_{T_0}. Fig. 5b shows vertical slices through T. Here, only three planes in the y-direction are shown for clarity. Note that T is axisymmetric around z and x. The simulation displayed in Fig. 5b was performed for $x \in [0\,\text{m}, 1.2\,\text{m}]$, $y \in [-1.2\,\text{m}, 1.2\,\text{m}]$ and $z \in [1\,\text{m}, 2\,\text{m}]$. This volume is defined by the *medPort* geometry which restricts the collision-free task space (see Fig. 1a). An increased condition number indicates high accelerations in joint space with respect to small accelerations in Cartesian space and hence the onset of singularities. A condition number close to zero represents non-singular positions. By choosing a configuration type (see Fig. 5a), the task space T is visibly reduced compared to the aforementioned spherical work space. Especially the innermost plane ($y = -0.05\,\text{m}$) shows unattainable positions in the vicinity of $x \approx 0\,\text{m}$ which resemble a cylinder geometry. Furthermore, the onset of a singularity in the range of $z \approx 1.7\,\text{m}$ can be recognized which is caused by the alignment of two joint axes. Note that if the configuration type was not maintained, the condition number would show increased values within the task space, indicating a configuration type change and hence dividing the task space in multiple non-singular regions. Since the analyzed configuration type has sufficient non-singular regions in T it is advisable to keep this configuration type throughout operation.

2.3.4 Gripper

The objective is to catch the aircraft directly without relying on additional guidance aids. To meet the requirements of the catching process, a specialized robot gripper has been developed. The gripper is shown in Fig. 6. It consists of four individually motorized arms. The arms, which are modeled in proportion to the fuselage, are equipped with foam for a soft and secure grasp.

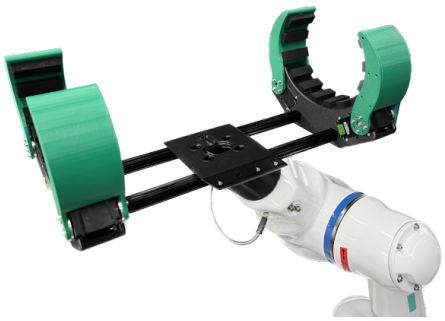


Fig. 6 Robot gripper

The servos utilized enable fast opening and closing as well as torque limiting to prevent damage to the aircraft. In the absence of maximum ratings, it is necessary to make several assumptions to determine the maximum sustainable force of the fuselage. First, the fuselage is orders of magnitude stiffer in the longitudinal and tangential directions than it is in the radial direction. Therefore, the planned deceleration trajectory of the gripper should prevent force input in the radial direction. Second, the applied force must not exceed the stiction between the fuselage and the gripper. The maximum stiction depends on the weight of the aircraft, the contact surface, and the gripping force. It will be determined by practical experiments.

Upcoming iterations of the gripper will feature passive tracking markers. As a result, a successful catch can be confirmed either by the force exerted on the aircraft's fuselage or the distance of the aircraft to the gripper measured by the tracking system.

2.3.5 Constraints and Tolerances

While requiring a precise tracking of the aircraft's pose, the catching process allows for some slack with respect to certain degrees of freedom. As an initial condition, an ideal catching pose is considered. The fuselage's shape in the grasping region can be approximated by using an elliptic cylinder. The gripper's jaws are designed to conform to the fuselage's shape while allowing a certain degree of misalignment.

- The pitch and yaw rotation tolerances are $\Delta\theta \pm 5\,\text{deg}$ and $\Delta\psi \pm 10\,\text{deg}$. The jaws will force the aircraft in the gripper's orientation. The force acting on the fuselage is proportional to the displacement. Excessive displacement may result in dropping out of the grip.
- Analogous considerations apply to the tolerance of linear displacement along the Y and Z axes. The tolerances are given with $\Delta y \pm 20\,\text{mm}$ and $\Delta z^{+0}_{-45}\text{mm}$ respectively. In addition to the force on the fuselage, linear displacement can cause roll rotation of the aircraft as it touches the gripper while being forced in a particular direction.
- Roll rotation tolerance is $\Delta\phi \pm 20\,\text{deg}$. Displacement at this angle will not damage the fuselage nor prevent a safe grip. However, the landing gear could make contact with the gripper at excessive roll angles.
- Linear displacement along the X axis is allowed within a tolerance of $\Delta x \pm 60\,\text{mm}$. Further displacement is prohibited by the aircraft's landing gear and the curvature of the fuselage.

The various tolerances mentioned above allow for optimization during the catching process. As an example, pitch and roll movements are small compared to the aircraft's yaw movement. Therefore, the yaw movement should be tracked continuously by the gripper whereas the pitch and roll movements may be neglected during **Phase 1** (see Section 2.2).

3 Data Preprocessing

The optical tracking system provides absolute pose data of a predefined rigid body with respect to its reference frame within a calibrated volume. The reference frame is the result of the system

calibration process. A rigid body is defined by a set of markers that remain constant in relation to each other. Under ideal conditions for a non moving target, the measurement for position has a standard deviation of $\sigma_{xyz} = 5 \times 10^{-3}$ m, and the measurement for rotation has a standard deviation of $\sigma_{\phi\theta\psi} = 5 \times 10^{-5}$ rad ($\approx 3 \times 10^{-3}$ deg). However, this pose signal is subject to random noise, depending on factors such as marker size and interference from extraneous light. Furthermore, as a rigid body is rotated within the calibrated volume, its geometry can mask the applied tracking markers to the cameras. Errors range from sporadic high-frequency interference to complete detection gaps in live tracking. The resulting pose signal is therefore unsuitable as a setpoint for the catching process.

Considering the aircraft as a rigid body, its motion has a lower frequency compared to the error in the measured pose signal. Therefore, a state estimation algorithm can be utilized to smooth the pose data and provide a linear extrapolation in the case of detection gaps. For this purpose, a Kalman filter is implemented [15]. As it is a recursive filter, it can process data in real-time, updating its current estimate as new data arrives. It is operated with an update frequency of 250 Hz to meet the requirements set by the robot controller.

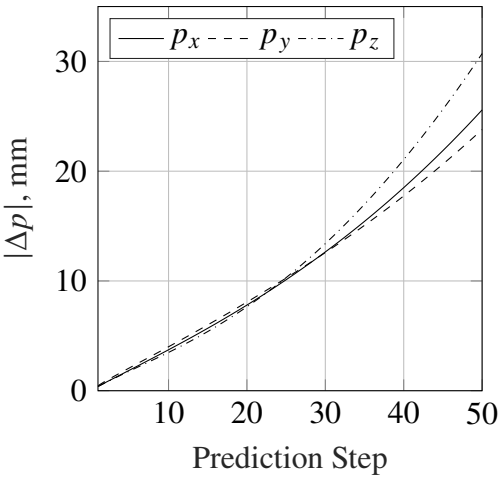


Fig. 7 Prediction deviation

To improve the filtering performance in the event of corrupted or missing optical tracking data, a decentralized approach has been developed. Aircraft telemetry is used as a secondary source of information. The data is transmitted using a low-latency data modem when the aircraft is in close proximity to the ground station.This approach allows different kinematic information to be included in the Kalman filter. For this application, accelerations and angular rates are chosen because these quantities can be measured directly and do not suffer from the random walk added by the integration of the aircraft's inertial navigation system (INS). The transmission rate was set at 250 Hz to provide data availability at each time step of the Kalman filter. In order to achieve such high transmission rates of less than 4 ms, a dedicated serial radio modem has been developed based on the ESP-NOW protocol [16]. The protocol selection is guided by previous research on the transmission latency of wireless protocols [17]. The IMU measurement of angular velocity has a standard deviation of $\sigma_{pqr} = 3 \times 10^{-3}$ rad/s (≈ 0.17 deg/s), and the measurement of acceleration has a standard deviation of $\sigma_{abc} = 3 \times 10^{-2}$ m/s^2. The filter is designed to run in two stages:

Stage 1: First, the aircraft's attitude is determined. The respective state vector $\vec{x}_{\text{att}}$ and measurement vector $\vec{z}_{\text{att}}$ is defined by Equation 1.

$$\vec{x}_{\text{att}} = \vec{z}_{\text{att}} = (p\ q\ r\ \phi\ \theta\ \psi)^{\text{T}} \tag{1}$$

The Euler angles are measured directly by the optical tracking system. The angular velocities with respect to the aircraft's body are transformed into the inertial reference frame by using the attitude propagation method of Euler angle derivatives.

Stage 2: Second, the aircraft's position is determined. The state vector is represented by Equation 2, while Equation 3 denotes the measurement vector.

$$\vec{x}_{\text{pos}} = (a_x\ a_y\ a_z\ u\ v\ w\ x\ y\ z\ \ddot{x}_{\text{off}}\ \ddot{y}_{\text{off}}\ \ddot{z}_{\text{off}})^{\text{T}} \tag{2}$$

$$\vec{z}_{\text{pos}} = (a_x\ a_y\ a_z\ x\ y\ z\ )^{\text{T}} \tag{3}$$

The position is measured directly by the optical system. The acceleration with respect to the aircraft's body has to be rotated in the inertial reference frame. This is done by using the aircraft's attitude determined in **Stage 1**. As the inertial reference frame's Z axis is not aligned with the earth's gravitation

vector, one must estimate the acceleration's offset. This is considered by $\ddot{x}_{\text{off}}, \ddot{z}_{\text{off}}$ and $\ddot{z}_{\text{off}}$ in the state vector. The values remain constant over time since the inertial reference frame is fixed.
Both stages are executed consecutively in each time step. The two data sources are integrated into the Kalman filter through two correction steps at each stage. After that, a single prediction step is used at each stage to advance the current state to the next time step.
The advantages of this filter design with respect to the capture process are as follows:

- Each data source can drop out independently without affecting the update cycle.
- In case of a detection gap in the optical tracking, the IMU enhances the filter prediction values.
- If both data sources are unavailable, the filter provides a short-term prediction of the pose signal.

The short-term prediction was evaluated on experimental flight data (see Fig. 7). Due to the experimental setup, one prediction step corresponds to 7 ms. It can be recognized that after ~308 ms of data unavailability the position deviation increased in average to an absolute of ~20 mm for the y-direction, violating the corresponding constraint (see Section 2.3.5). Thus, if the data unavailability lasts longer than ~308 ms, it is not possible to make a statement about the success of the capture.
Figure 8 shows an extract of a position filter measurement taken using the setup shown in Figure 1b. When optical tracking data is available, the signals from both the filter and the tracking system are generally consistent. The filter also performs signal smoothing without adding time delay. In the event of a detection gap, the filter is able to reproduce the measurement from the optical tracking system using the aircraft's telemetry data. Without the second data source, the filter would only do a quadratic prediction of the pose on the basis of the last measurement.

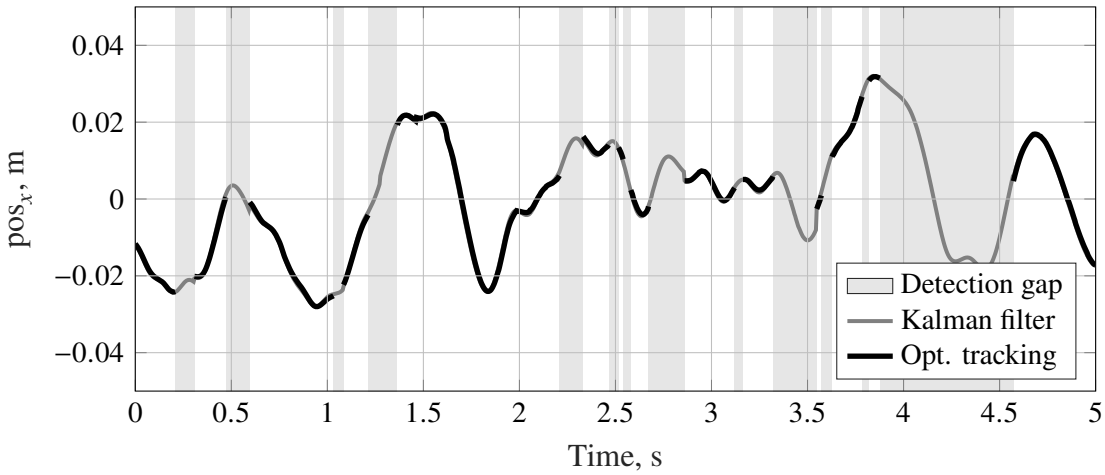


Fig. 8 Kalman filter for the x-direction

4 Direct Trajectory Tracking Method

In the subsequent section, a direct tracking method for the catching process without adaptive trajectory optimisation will be described. The method essentially involves setting successive command poses for the robot's inverse kinematic (IK) solver. The interpolation is accomplished via an inverse kinematic solver, which enables smooth transition between discretely sampled target poses by employing a modified Jacobian transpose method [18]. The joint angles are determined iteratively within a closed loop through the use of the forward kinematics of a dynamic model of the robot.
An experiment has been conducted to evaluate the most time-critical phase of the capture process, which is the real-time tracking of a target by the robot. Its purpose is to measure the delay between the input of the pose and the reaching of the pose by the robot's TCP. The expected delay can mostly be attributed to the IK algorithm and the robot hardware. The corresponding block diagram is presented in Fig. 9

together with a picture of the test setup in Fig. 10.

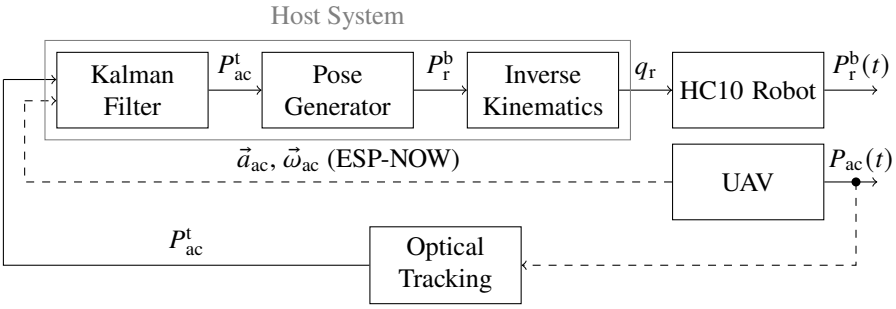


Fig. 9 Block diagram of the test setup

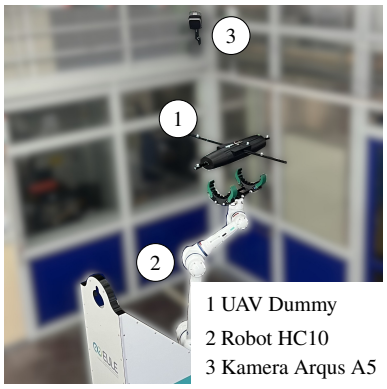


Fig. 10 Test setup

A dummy suspended from an aerial rope slide is used to simulate the movement of the aircraft. The dummy is equipped with tracking markers, an IMU and an ESP-NOW modem. The experiment procedure is as follows:

- The pose of the aircraft dummy wrt. the optical tracking reference frame, denoted by P^{t}_{ac}, is captured and provided as an input to the Kalman filter. Additionally, its telemetry data, comprising $\vec{a}_{ac}$ and $\vec{\omega}_{ac}$, is transmitted using the radio modem.
- The filter provides an estimate of the dummy pose to the pose generator.
- The pose generator transforms the pose into the robot's base reference frame. Moreover, it adds a vertical offset to the pose and neglects the pitch and roll movements (see Section 2.3.5), resulting in P^{b}_{r}.
- The IK solver maps the Cartesian pose into the joint space q_r, which are then passed to the robot via the real-time interface [14].
- Finally, the joint angles are executed by the robot controller, which also provides an accurate encoder feedback of the joint angles q_{fdb}. Using these with forward kinematics, an feedback pose P_{fdb} is calculated, which represents the real TCP pose.

The IK solver can be configured to provide fault tolerant tracking of a sparsely sampled target trajectory as well as direct tracking of a target with minimal delay. Based on Scherzinger et al. the IK solver was configured with error scaling $K_P = 5$ and iteration count $N = 20$ for a trade-off between trajectory smoothing and tracking error [18]. The results are shown in Fig. 11 and Fig. 12 respectively for the y-direction since it is the most constrained axis (see Section 2.3.5). A noticeable delay of ~116 ms can be recognized in Fig. 11 between the aircraft's position p_{ac} and the robot feedback position p_{fdb}. Additionally, a delay between the aircraft's position p_{ac} and the robot's commanded position p_r of ~16 ms can be attributed to the IK solver. Moreover, Fig. 12 portrays the absolute position error in y-direction between p_{ac} and p_{fdb}. The Δy-constraint is drawn in a dashed line for reference. Thus, it is visible that the constraint was not met during the test.
As an intermediate conclusion, the simple method presented is not sufficient for following and catching the aircraft. By increasing the error scaling of the IK solver, the tracking time error of 16 ms can be slightly reduced at the expense of the smoothing capabilities of the IK solver. In view of these aspects, the benefits of the intermediate waypoint interpolating IK solver in use cannot be substantiated. The problems could be mitigated by adaptive error scaling based on the current capture state. However, the inherent delay of the robot controller will always remain, resulting in a position error that violates the capture constraints. Instead, a solver can be used which directly yields the joint angles Δq_r to get from the current pose to the target pose together with explicit trajectory generation.

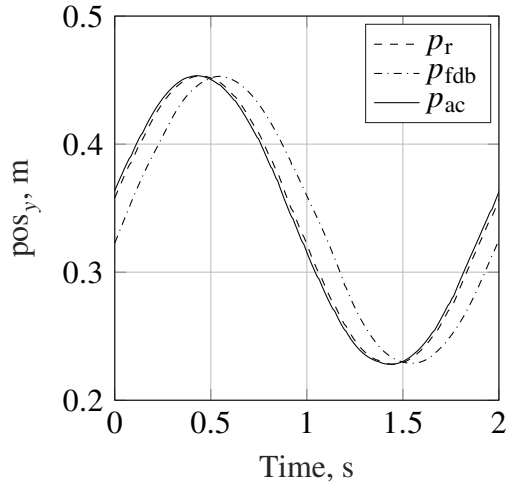


Fig. 11 Direct tracking method comparison

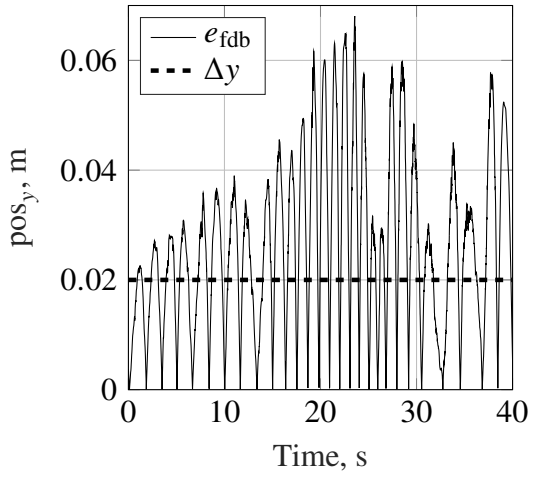


Fig. 12 Position error of direct tracking method

5 Trajectory Generation

The catching process, with its constraints and tolerances described in the previous sections suggests a trajectory generation based on optimization methods that exploit the given slackness. A model predictive approach for generating the successive trajectories described in Section 2.2 is currently under development. By maintaining a robot configuration type as described in Section 2.3.3 and defining a corresponding task space T in Cartesian space, fast linear motions can be performed without grazing singularities. However, with this approach, additional constraints in the joint space must be considered when planning to maintain the configuration type.

6 Conclusion

An approach for the automated turnaround of an unmanned tilt-wing aircraft has been presented.
The base station developed is designed to catch the aircraft in hover flight, for successive payload unloading and automated takeoff.
To first assess the aircraft's motion, an analysis of recorded flight data was conducted, which led to statements about the probability of the aircraft's presence during hover flight.
The key components of the station have been presented, along with their specific restrictions related to the process of catching the aircraft.
To ensure a robust estimation of the aircraft pose despite detection errors of the optical tracking system, a decentralized Kalman filter has been developed. The filter combines the pose signal from the optical tracking system with the aircraft acceleration and angular velocity transmitted using a low-latency radio modem. It has been demonstrated that the filter can compensate for short-term detection gaps in the optical tracking system. Depending on its error and frequency resolution, the onboard calculated pose can be used directly instead, eliminating the need for double integration.
A direct pose tracking method has been proposed for executing the catching process. To validate the method, a test was set up to evaluate the real-time pose tracking using a robotic arm. The setpoint pose was provided directly by the optical tracking system. Due to the measured time delay of the robot, the method proved insufficient since it does not allow to catch the aircraft in flight. Therefore, the model predictive control approach was proposed and is currently under development.

Acknowledgments

Our work has been conducted within the EULE project. This project is funded by the Federal Ministry for Digital and Transport (BMDV) within the mFund program. We thank all project partners for their work and contributions to the EULE project.

References

[1] Y. Dobrev and M. Schütt. Entwurf und Validierung eines Präzisionslandesystems für unbemannte Tiltwing-Fluggeräte. 2016.

[2] Fabian Walter, Elias Hodel, Erik S Mannerfelt, Kristen Cook, Michael Dietze, Livia Estermann, Michaela Wenner, Daniel Farinotti, Martin Fengler, Lukas Hammerschmidt, Flavia Hänsli, Jacob Hirschberg, Brian Mcardell, and Peter Molnar. Brief communication: An autonomous UAV for catchment-wide monitoring of a debris flow torrent. 22:4011–4018, 2022. DOI: 10.5194/nhess-22-4011-2022.

[3] Jenny Russo. Matternet Announces Commercial Deployment of the Matternet Station. 2021.

[4] So-Ryeok Oh, K. Pathak, S.K. Agrawal, H.R. Pota, and M. Garratt. Approaches for a tether-guided landing of an autonomous helicopter. 22(3):536–544, 2006. ISSN: 1552-3098. DOI: 10.1109/TRO.2006.870657.

[5] Carlo Giorgio Grlj, Nino Krznar, and Marko Pranjić. A Decade of UAV Docking Stations: A Brief Overview of Mobile and Fixed Landing Platforms. 6(1):17, 2022. ISSN: 2504-446X. DOI: 10.3390/drones6010017.

[6] Moritz Maier and Konstantin Kondak. Landing of VTOL UAVs using a stationary robot manipulator: A new approach for coordinated control. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 1497–1502. IEEE, 2015. ISBN: 978-1-4799-7886-1. DOI: 10.1109/CDC.2015.7402422, `http://ieeexplore.ieee.org/document/7402422/`.

[7] Moritz Maier, Andre Oeschger, and Konstantin Kondak. Robot-Assisted Landing of VTOL UAVs: Design and Comparison of Coupled and Decoupling Linear State-Space Control Approaches. 1(1):114–121, 2016. ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2015.2502920.

[8] Moritz Maier and Konstantin Kondak. Robot assisted landing of VTOL UAVs on ships: A simulation case study of the touch-down phase. In *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pages 2094–2101. IEEE, 2017. ISBN: 978-1-5090-2182-6. DOI: 10.1109/CCTA.2017.8062762, `http://ieeexplore.ieee.org/document/8062762/`.

[9] Moritz Maier, Konstantin Kondak, and Christian Ott. Enabling robot assisted landing of heavy UAV rotorcraft via combined control and workload sharing. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5128–5134. IEEE, 2017. ISBN: 978-1-5386-2682-5. DOI: 10.1109/IROS.2017.8206399, `http://ieeexplore.ieee.org/document/8206399/`.

[10] FreeRail mFUND Projekt- Gleisinspektion mittels UAS, 2022. `https://www.youtube.com/watch?v=iMkI_P5Za_A`.

[11] Steven Macenski, Tully Foote, Brian Gerkey, Chris Lalancette, and William Woodall. Robot Operating System 2: Design, architecture, and uses in the wild. 7(66), 2022. DOI: 10.1126/scirobotics.abm6074.

[12] L. Puck, P. Keller, T. Schnell, C. Plasberg, A. Tanev, G. Heppner, A. Roennau, and R. Dillmann. Distributed and Synchronized Setup towards Real-Time Robotic Control using ROS2 on Linux. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2020. DOI: 10.1109/case48305.2020.9217010.

[13] L. Puck, P. Keller, T. Schnell, C. Plasberg, A. Tanev, G. Heppner, A. Roennau, and R. Dillmann. Performance Evaluation of Real-Time ROS2 Robotic Control in a Time-Synchronized Distributed Network. In *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2021. DOI: 10.1109/case49439.2021.9551447.

[14] V. Konnow. Motoman ROS2, 2022. `https://github.com/adv4ncr/motoman_ROS2`.

[15] Jan Wendel. Integrierte navigationssysteme: Sensordatenfusion, gps und inertiale navigation. In *Integrierte Navigationssysteme*. Oldenbourg Wissenschaftsverlag. ISBN: 978-3-486-70572-0. DOI: 10.1524/9783486705720.

[16] Espressif Systems. ESP-NOW Wireless Communication Protocol, 2016. `https://www.espressif.com/en/solutions/low-power-solutions/esp-now`.

[17] Dania Eridani, Adian Fatchur Rochim, and Faiz Noerdiyan Cesara. Comparative Performance Study of ESP-NOW, Wi-Fi, Bluetooth Protocols based on Range, Transmission Speed, Latency, Energy Usage and Barrier Resistance. In *2021 International Seminar on Application for Technology of Information and Communication (iSemantic)*, pages 322–328, 2021. DOI: 10.1109/iSemantic52711.2021.9573246, `https://ieeexplore.ieee.org/document/9573246`.

[18] Stefan Scherzinger, Arne Roennau, and Rudiger Dillmann. Inverse Kinematics with Forward Dynamics Solvers for Sampled Motion Tracking. In *2019 19th International Conference on Advanced Robotics (ICAR)*. IEEE, 2019. DOI: 10.1109/icar46387.2019.8981554.