POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Adaptive Vision-Based Control for Autonomous Air-to-Air Landing of Multi-Rotor UAVs

Laurea Magistrale in Aeronautical Engineering - Ingegneria Aeronautica

Author: Álvaro Perrinó Gamino and Alex Revoltos Salmador

Advisor: Prof. Davide Invernizzi

Co-advisor: Giovanni Gozzini and Alessandro Nazzari

Academic year: 2024-2025

1. Introduction

This thesis investigates the problem of Autonomous Air-to-Air Landing (AAAL), in which a small Unmanned Aerial Vehicle (UAV) autonomously lands on a larger carrier drone in motion. This complex task integrates vision-based navigation, state estimation, and control under dynamic conditions. In the proposed framework, the follower UAV tracks the carrier by detecting an ArUco marker mounted on the landing platform, which provides relative position measurements for the navigation system. The aim of this work is to build upon the approaches presented in [1] and [2] to enhance disturbance rejection and improve the estimation of the acceleration of the target, both of which are essential for achieving accurate tracking, as well as to facilitate the transition from preliminary simulations to in-flight experiments. In [2], the proposed design is able to ensure autonomous landing only for slowly varying velocity of the carrier trajectory with almost null acceleration. To address this limitation, three complementary approaches are investigated. The first involves replacing the standard state estimator with an alternative scheme capable of reconstructing the acceleration of the carrier, which represents the main exogenous influence on the system. The other two approaches are based on the development of an adaptive law which rely on internal-model–based compensators. With these methods, the controller is able to accurately track not only constant carrier acceleration but also harmonic ones. These methods are first evaluated in a dedicated simulation environment, and the most promising ones are subsequently tested in the Flying Arena for Rotorcraft Technologies of Politecnico di Milano.

2. Problem Description

This section describes the system dynamics and the control law developed in [2] to ensure safe and fast Autonomous Air-to-Air Landing.

2.1. Dynamics of Underactuated VTOL UAV

According to [2], the simplified dynamics used for control design of the small follower UAV is given by:

$$\dot{R}_f = R_f S(\omega_c), \tag{1}$$

$$\dot{x}_f = v_f, \tag{2}$$

$$m\dot{v}_f = -mge_3 + t_c R_f e_3 + f_e, \tag{3}$$

where $x_f \in \mathbb{R}^3$ is the position of the follower in the inertial frame, $t_c \in \mathbb{R}_{>0}$ is the control thrust, $\omega_c \in \mathbb{R}^3$ is the control angular velocity, $R_f \in SO_3$ is the rotation matrix that allows the change from the body frame to the inertial frame, $S(\omega)$ is the skew-symmetric operator, f_e is the exogenous force term, m is the drone mass and $g = 9.81\,\mathrm{m/s^2}$ is the gravity acceleration. Note that the subscript f denotes the follower while t refers to the target.
Once the dynamics are defined, the tracking error can be expressed as:

$$e_p := x_f - x_t, \tag{4}$$

where $x_t \in \mathbb{R}^3$ is the target position. From this, the error dynamics can be expressed as:

$$\dot{e}_p = \dot{x}_f - \dot{x}_t = v_f - v_t. \tag{5}$$

Defining the control input u_v such that the velocity error is $e_v = v_f - u_v$, the error dynamics can be re-written as:

$$\dot{e}_p = v_f - v_t \pm u_v = u_v - v_t + e_v. \tag{6}$$

From (3) and (6), the time derivative of the velocity error becomes:

$$\dot{e}_v = \dot{v}_f - \dot{u}_v = -ge_3 + a_c R_f e_3 + a_e - \dot{u}_v, \tag{7}$$

where $a_c := \frac{t_c}{m}$ is the control acceleration, and $a_e := \frac{f_e}{m}$ is the acceleration related to exogenous disturbances. Since $a_c \in \mathbb{R}_{>0}$ does not span $\mathbb{R}^3$, it cannot be used to control the velocity error dynamics. Therefore, a virtual input u_a must be included in (7):

$$\dot{e}_v = -(ge_3 + \dot{u}_v) + a_c R_f e_3 + a_e \pm u_a. \tag{8}$$

Re-adjusting equation (8) for underactuated UAVs results in:

$$\dot{e}_v = -\left(ge_3 + \dot{u}_v\right) + u_a + a_e + \Delta a(R_p, R_e, u_a), \tag{9}$$

where $\Delta a := \left(R_p R_e R_p^\top - I_3\right) u_a$ is the mismatch between virtual acceleration u_a and the acceleration actually delivered by the follower drone, R_p is the planned attitude [3] and R_e is the attitude error. Given that Δa depends on the attitude error R_e, it can be driven to zero using a suitable attitude controller [2]. Therefore, the proposed AAAL control law focuses on designing u_v and u_a to ensure that the tracking error converges to zero in a safe manner.

2.2. Control Law and Hybrid Logic

In this section, the fundamentals of the landing logic, specifically the design of u_v as introduced in [2], are briefly reviewed for completeness. The emphasis, however, is placed on the design of u_a, which represents the core contribution of this work.
The control law follows a three-layer hierarchical architecture, composed of a position layer, a velocity layer, and an attitude layer. The position controller is based upon the error dynamics equation. The error is split into two different components, the planar and vertical components, which can be defined as $e_p^\perp := \begin{bmatrix} e_{p_1} & e_{p_2} \end{bmatrix}^\top$ and e_{p_3}. In addition, the hybrid logic is used in order to limit the control input to perform a safe landing maneuver. The control input saturation levels are dependent on the logic state provided by the hybrid logic which in turn is dependent on the tracking error. The different logic states $q \in Q := \{0, 1, 2\}$ are defined as follows:

$$C_0 := \{e_p \in \mathbb{R}^3 : e_p^\perp \geq r_m, \quad e_{p_3} \geq h_a\} \tag{10}$$

$$C_1 := \{e_p \in \mathbb{R}^3 : e_p^\perp \leq r_m, \quad e_{p_3} \geq h_a\} \tag{11}$$

$$C_2 := \{e_p \in \mathbb{R}^3 : e_p^\perp \leq r_m, \quad e_{p_3} = 0\}, \tag{12}$$

where h_a is the altitude at which mode 1 (approach mode) ends, and r_m is the horizontal distance that determines where mode 0 (synchronization mode) ends. Since there are transitions between states, a term Δ is introduced as a hysteresis mechanism to avoid chattering. Figure 1 depicts how the logic works, the different modes and the conditions for each mode in terms of planar and vertical error.
Knowing this, the control input u_v is defined in terms of the hybrid logic, as well as saturation functions and levels.
After defining the position controller, a velocity controller is introduced to track u_v, the virtual input velocity. This is done with the use of the virtual acceleration input u_a. For the derivation of the velocity control law, the Δa mismatch is assumed to be zero under the assumption of fast attitude control. Therefore, the derivative of the error dynamics is given by:

$$\dot{e}_v = -ge_3 + u_a + a_e + D_{ep}\gamma_q(e_p)\dot{e}_p - \dot{v}_t \tag{13}$$

$$= -ge_3 + u_a + a_e + D_{ep}\gamma_q(e_p)(e_v - \gamma_q(e_p)) - \dot{v}_t, \tag{14}$$

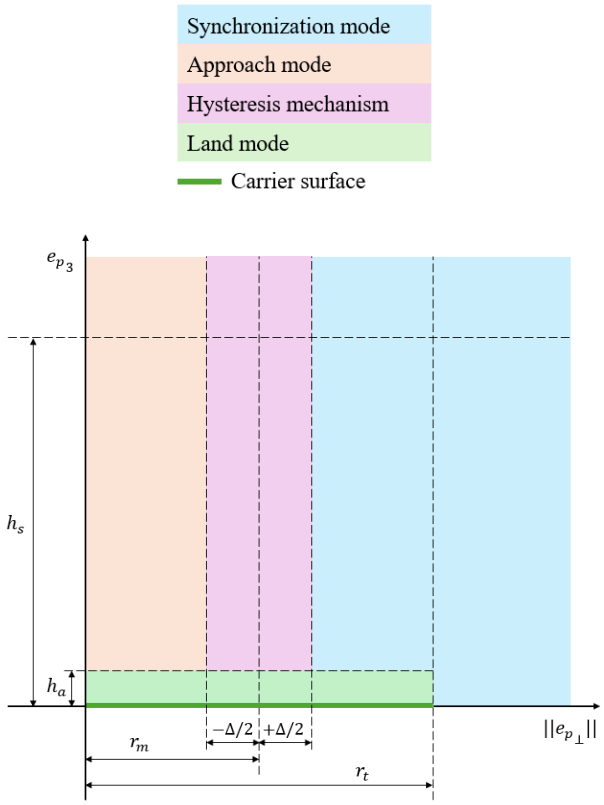

Figure 1: Landing Logic.

where $u_a := \frac{u}{m}$ is the control acceleration input. Since the acceleration of the carrier $\dot{v}_t$ is unknown, it was added to the exogenous term. Thus, the exogenous term, acting as disturbance acceleration, is given by:

$$a_d = a_e - \dot{v}_t, \tag{15}$$

with $\|a_d\| \leq a_M$, where $a_M \in \mathbb{R}_{\geq 0}$ is the assumed upper bound of the total disturbance acceleration. Since the disturbance cannot be measured, an observer-based adaptive control law was implemented to compensate for it:

$$\dot{\hat{e}}_v = -\gamma_v(e_v) + L(e_v - \hat{e}_v) \tag{16}$$

$$\dot{\hat{a}}_d = \text{proj}(\hat{a}_d, -\Gamma_d(\hat{e}_v - 2e_v)) \tag{17}$$

$$u_a = ge_3 - D_{e_p}\gamma_q(e_p)(e_v - \gamma_q(e_p)) + \gamma_v(e_v) - \hat{a}_d, \tag{18}$$

where $(\hat{e}_v, \hat{a}_d) \in \mathbb{R}^3 \times \mathbb{R}^3$ is the state of the controller, $\gamma_v(e_v) := \sigma_M(K_v e_v)$ is the saturated velocity stabilizer, $L \in \mathbb{R}^{3\times 3}$, $\Gamma_d \in \mathbb{R}^{3\times 3}$, $K_v \in \mathbb{R}^{3\times 3}$ are diagonal positive-definite matrices, and $M \in \mathbb{R}_{>0}$ are the saturation levels used for each axis. Finally, given $y \in \mathbb{R}$, the projection operator is defined as

$$\text{proj}(\theta, y) := \begin{cases} y - f(\theta)\dfrac{\nabla f(\theta)(\nabla f(\theta))^\top}{\|\nabla f(\theta)\|^2} y, \\ \qquad \text{if } \left[f(\theta) > 0 \wedge y^\top \nabla f(\theta) > 0\right], \\ y, \qquad \text{otherwise.} \end{cases} \tag{19}$$

where $\theta \in \mathbb{R}$ is the parameter estimate and $f(\theta) := \frac{\|\theta\|^2 - \theta_M^2}{2\epsilon\theta_M^2 + \epsilon^2}$, with $\epsilon > 0$ a tunable parameter and θ_M the maximum absolute value of θ. As stated in [2], when $\dot{a}_d = 0$, the equilibrium point is Globally Asymptotically Stable, and for any bounded $\dot{a}_d$, the closed-loop solutions of the system are Uniformly Ultimately Bounded. Thus, the controller reported in (16)-(18) achieves satisfactory performance for dynamic trajectories (such as circular motion), provided that the acceleration remains small and the gains are properly tuned.

3. Methodology

Despite providing acceptable results, the observer presents limitations in terms of performance and stability with $\dot{a}_d \neq 0$. To improve on this, different alternative solutions are investigated.

3.1. Adaptive Input and State Estimation (AISE)

The Adaptive Input and State Estimation (AISE) [4] is similar to a regular Kalman Filter (KF). The main difference between the AISE and the traditional KF is that this innovative method is capable of estimating higher-order derivatives of the position by solving an optimization problem. The underlying idea is that having an estimate of the acceleration, this term can be removed from the exogenous component of the dynamics expressed in (16).
The state equations for both filters are:

$$\text{AISE: } x_{\text{fc},k+1} = A_{\text{AISE}} x_{\text{da},k} + B\hat{d}_k \tag{20}$$

$$\text{Kalman Filter: } x_{\text{fc},k+1} = A_{\text{KF}} x_{\text{da},k}, \tag{21}$$

where x is the state vector, which belongs to $\mathbb{R}^2$ and $\mathbb{R}^3$ for KF and AISE, respectively; $B \in \mathbb{R}^3$ is the input matrix, $\hat{d} \in \mathbb{R}$ is the input estimate, and$A$ is the state matrix. The above matrices

are defined as:

$$A_{\text{AISE}} = \begin{bmatrix} 1 & T_\text{s} & \frac{1}{2}T_\text{s}^2 \\ 0 & 1 & T_\text{s} \\ 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} \frac{1}{6}T_\text{s}^3 \\ \frac{1}{2}T_\text{s}^2 \\ T_\text{s} \end{bmatrix},$$

$$A_{\text{KF}} = \begin{bmatrix} 1 & T_\text{s} \\ 0 & 1 \end{bmatrix}. \tag{22}$$

In this case, the input estimate $\hat{d}$ corresponds to the jerk of the target drone. As reported in [4], the estimate $\hat{d}_k$ is given by:

$$\hat{d}_k = \sum_{i=1}^{n_e} P_{i,k}\hat{d}_{k-i} + \sum_{i=0}^{n_e} Q_{i,k}z_{k-i}, \tag{23}$$

where $P_{i,k} \in \mathbb{R}$ and $Q_{i,k} \in \mathbb{R}$ are gains, and z is the innovation. To compute these gains, the following Recursive Least-Squares problem is solved:

$$P_{k+1}^{-1} = \lambda_k P_k^{-1} + (1-\lambda_k)R_\infty + \widetilde{\Phi}_k^\text{T}\widetilde{R}\widetilde{\Phi}_k \tag{24}$$

$$\theta_{k+1} = \theta_k + P_{k+1}\widetilde{\Phi}_k^\text{T}\widetilde{R}(\widetilde{z}_k + \widetilde{\Phi}_k\theta_k), \tag{25}$$

where $P_k \in \mathbb{R}^{l_\theta \times l_\theta}$ is the positive-definite covariance matrix, the positive-definite matrix $R_\infty \in \mathbb{R}^{l_\theta \times l_\theta}$ is the user-selected resetting matrix, $\widetilde{\Phi}_k$ is the filter regressor matrix, $\widetilde{R}$ is a weighting matrix, λ_k is the forgetting factor, and where, for all $k \geq 0$,

$$\hat{z}_k \triangleq \begin{bmatrix} z_k - \hat{d}_{\text{f},k} \\ 0 \end{bmatrix}. \tag{26}$$

Solving this problem which depends on the innovation z, provides the necessary gains which are given by:

$$\theta_k \triangleq \begin{bmatrix} P_{1,k} & \cdots & P_{n_\text{e},k} & Q_{0,k} & \cdots & Q_{n_\text{e},k} \end{bmatrix}^\text{T} \tag{27}$$

3.2. Harmonic Compensators

The harmonic compensators implemented in [5] are based on the internal model principle, which uses the difference between the process and the internal model outputs to obtain the estimate of the disturbance. In this context, the disturbance corresponds to the disturbance acceleration $\hat{a}_d$. The idea with the compensators is to replace the observer-based adaptive control law defined in (16)-(18) by the internal models. In the equations later explained, γ_c and γ_ω are related to the disturbance as follows:

$$\begin{cases} \dot{x}_c = \gamma_c(x_c, e_v) \\ \hat{a}_d = \gamma_\omega(x_c, e_v) \end{cases}, \tag{28}$$

where $x_c \in \mathbb{R}^3$ is the state of the internal model, and $\hat{a}_d \in \mathbb{R}^3$ is the disturbance acceleration estimation.

3.3. Standard Harmonic Compensator (SHC)

The standard harmonic compensator (SHC) is defined by:

$$\gamma_c(x_c, e_v) := A_w x_c + K_c C_w^\top e_v \tag{29}$$

$$\gamma_\omega(x_c, e_v) := C_w x_c + K_\omega e_v, \tag{30}$$

where $A_w = \text{blkdiag}\left(\mathbf{0}_3, \begin{bmatrix} 0 & -\Omega_1 \\ \Omega_1 & 0 \end{bmatrix}, \ldots, \begin{bmatrix} 0 & -\Omega_{n_d} \\ \Omega_{n_d} & 0 \end{bmatrix}\right) \in \mathbb{R}^{(3+2n_d)\times(3+2n_d)}$, and $C_w \in \mathbb{R}^{3\times(3+2n_d)}$ are assumed to be an observable pair, being n_d the number of disturbances. With tunable matrices $K_c = \text{blkdiag}(K_I, k_{c_1}I_2, \ldots, k_{c_d}I_2) \in \mathbb{R}_{>0}^{3+2n_d\times 3+2n_d}$, $K_I = \text{diag}(k_{I_1}, k_{I_2}, k_{I_3}) \in \mathbb{R}_{>0}^{3\times 3}$, where $k_{c_i} > 0$ $\forall i \in \{1, 2 \ldots n_d\}$. Note that for the SHC, the frequencies of the harmonics must be known and included in matrix A_w. For example, for a circular trajectory, the frequency of the harmonic corresponds to the angular frequency of the carrier trajectory ω_t.

3.4. Adaptive Harmonic Compensator (AHC)

The upside of this method is that no previous knowledge of the harmonic frequency is needed. The Adaptive Harmonic Compensator (AHC) is given by:

$$\gamma_{c_i}(x_c, e_v) := \\ := \begin{bmatrix} A_{o_i}\hat{\zeta}_i - b_{o_i}\left(k_{\omega_i}e_{v_i} - \hat{\theta}_i^\top\hat{\zeta}_i + e_{v_i}\right) \\ -k_{c_i}\hat{\zeta}_i e_{v_i} \end{bmatrix} \tag{31}$$

$$\gamma_{\omega_i}(x_c, e_v) := k_{\omega_i}e_{v_i} - \hat{\theta}_i^\top\hat{\zeta}_i + e_{v_i} \tag{32}$$

$\forall i \in \{1, 2, 3\}$, where $x_{c_i} = \begin{bmatrix} \eta_i^\top & \hat{\theta}_i^\top \end{bmatrix}^\top \in \mathbb{R}^{2+2}$, and $\hat{\zeta}_i := \eta_i - b_{o_i}me_{v_i}$. The pair $(A_{o_i}, b_{o_i}) \in \mathbb{R}_{<0}^{2\times 2} \times \mathbb{R}^2$ satisfies $M_iA_{w_i} - A_{o_i}M_i = b_{o_i}c_{w_i}^\top$ for

some non-singular matrix $M_i \in \mathbb{R}^{2\times2}$, $k_{c_i} > 0$ is the adaptation gain, and m is the mass of the follower.

3.5. Adaptive controllers comparison

Under ideal conditions, that is, with perfect position measurements and without delays, the observer-based adaptive controller with projection operator (PO), the SHC and AHC are tested. For the test the carrier is set to perform a circular trajectory, with the goal of finding the maximum angular speed of the carrier at which landing could be successful for each of the controllers. The results are summarized in Table 1. It is evident that both compensators greatly outperformed the adaptive controller used in [2].

Table 1: Landing results for different adaptive controllers.

	ω [rad/s]	Landing Time [s]
PO	0.8	12.2
SHC	3	11.13
AHC	3	17.09

4. Simulation Environment

A simulation environment is designed to ease the transition between preliminary simulations and in-flight tests by creating realistic simulations of the landing scenario. In experiments where information between the drones is not shared (non-cooperative scenario), the follower drone relies on a mounted camera which is used to detect an ArUco marker attached to the target to estimate its position and velocity.

4.1. Environment Design

To replicate this setup in Simulink, the Unreal Engine package is used to introduce vision-in-the-loop. The environment includes:

- Drones: Both the carrier and the follower;
- ArUco Marker: Attached to the top of the carrier;
- Vision Detection System: Camera attached to the follower and ArUco detection system implemented;
- Scene: A 3D simulation scene is implemented.

The interaction of the simulation environment with the rest of the system already developed in [2] can be seen in Figure 2.

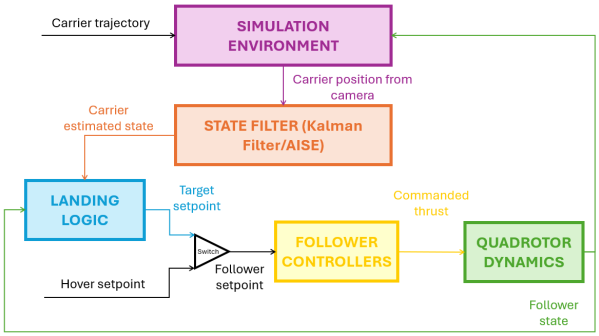

Figure 2: Complete AAAL System.

4.2. Simulation Results

Different combinations of filters and disturbance compensators are evaluated to determine the best performing control system. In all simulations, the carrier is set to perform a circular trajectory with radius 1 m and varying angular speed. The aim is to find the maximum angular speed at which landing is successful. The results can be observed in Table 2.

Table 2: Summary of circular trajectory.

Filter	Compensator	Max. ω [rad/s]
AISE	PO	0.5
KF	PO	0.55
KF	SHC	0.75
KF	AHC	0.65

Table 2 shows the maximum angular speed at which landing is successfully performed for different combinations of filter and compensator. The angular velocities reached using the simulation environment are significantly slower compared to the ones obtained using the simplified scenario (see Table 1), because of the delay introduced by the Kalman Filter in the estimation of the carrier velocity, which is approximately 0.3 seconds. As can be observed, the AISE filter is outperformed by the Kalman filter. This is due to the offset in phase and amplitude observed for both the velocity and acceleration estimates, which can be observed in Figure 3. Since this method does not improve the performance of the existing system, it is discarded. In addition, the

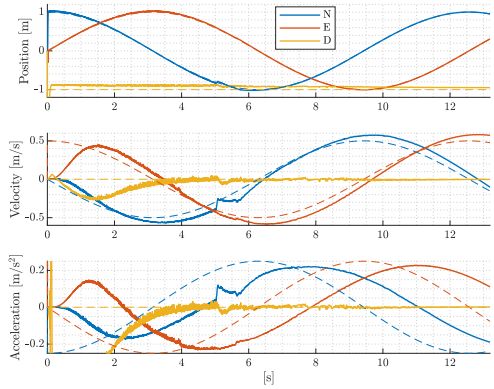

Figure 3: AISE Filter Estimation.

two compensators provide an improvement in terms of maximum angular speed of the carrier when performing a circular trajectory. In terms of landing speed, the best performance is obtained with the SHC. This can be expected since knowledge about the carrier's angular speed ω_t is provided to the compensator.

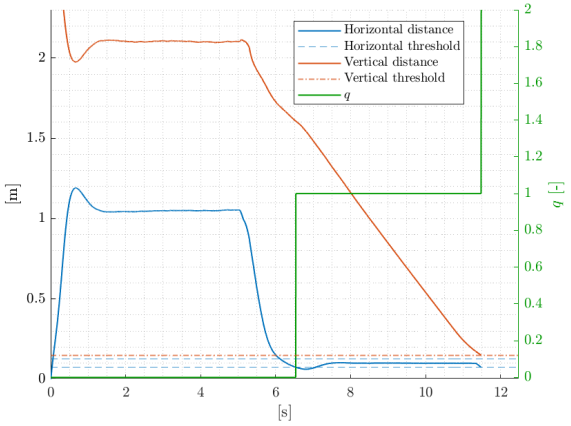

Figure 4: Landing logic state.

Figure 4 and Figure 5 illustrate the behavior of the hybrid logic during the landing procedure. Note that in Figure 5, $r_{in} = r_m - \Delta/2$ and $r_{out} = r_m + \Delta/2$. The logic state q switches from synchronization mode (q = 0) to approach mode (q = 1) as soon as the horizontal tracking error enters the predefined threshold. Once the vertical distance between the carrier and the follower becomes smaller than 0.15 m, q transitions from $q = 1$ to $q = 2$, marking the end of the simulation. The example shown in these figures corresponds to the SHC configuration with an angular frequency of $\omega = 0.75$ rad/s, which represents the most aggressive trajectory considered in this study. Nevertheless, the versatility that the AHC provides is worth considering since in non-cooperative scenarios, the carrier frequency is usually unknown.

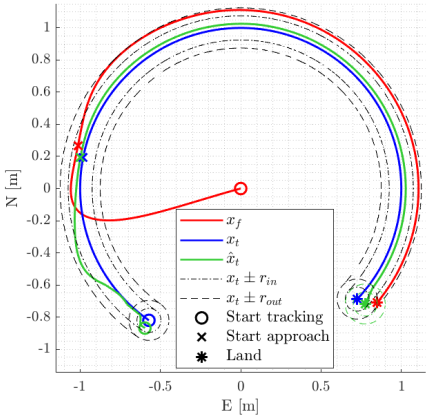

Figure 5: Top view.

5. In-Flight Experiments

This section describes the experimental setup and presents the results obtained.

5.1. Experimental Setup

Flight tests were performed in the FlyART facility at Politecnico di Milano, an indoor arena with a 12-camera motion capture system (Mo-Cap) tracking markers on the drones. By using the Mo-Cap system instead of the camera detection system of the follower, the idea is to study the compensators in a cooperative scenario with high precision detection to focus solely on the comparison between the different adaptive controllers. For the tests, the AISE filter is discarded since it previously showed poor performance. The three adaptive controllers tested are the observed-based adaptive controller with the projection operator, the SHC and the AHC. For the test, the carrier is set to perform a circular trajectory of radius 1 m and angular speed of either 0.6 rad/s or 1 rad/s.

5.2. Experiment Results

With $\omega_t = 0.6$ rad/s, all adaptive controllers, PO, SHC and AHC, successfully completed the landing procedure, providing a similar tracking error. When the carrier's angular velocity is increased to $\omega_t = 1$ rad/s, the landing maneuver is still completed with all controllers, but clear differences emerge in their behavior. Fig-

ure 6 shows the evolution of the tracking error for the three approaches. As expected, the Projection Operator (PO) exhibits the largest error, reflecting its limited capability to compensate for harmonic trajectories. For the AHC, the initial set of parameters resulted in excessively aggressive control action, causing the UAV to generate high attitude angles and thrust peaks that ultimately led to a crash. After retuning the parameters, AHC exhibits a larger tracking error at the beginning because it does not know the carrier's angular velocity, but it quickly converges to lower values, presenting a very similar behavior to the SHC.

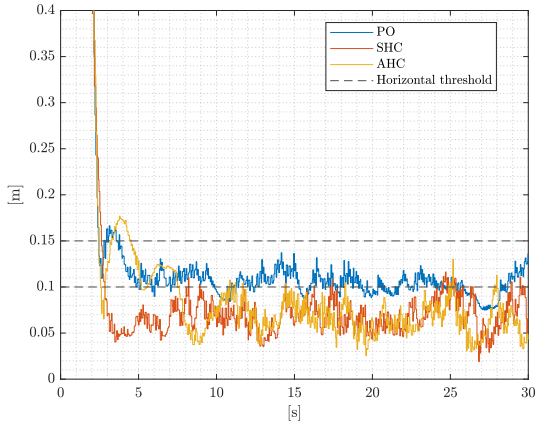

Figure 6: Circular $\omega = 1$ rad/s. Tracking error comparison.

6. Conclusions

The aims of the thesis were to develop a simulation environment as well as investigating new methods to improve the adaptive controller of UAVs to perform air-to-air landing. Both were successfully achieved.

Regarding the controllers, both harmonic compensators showed an improvement in performance with respect to the existing adaptive controller. On the one side, the SHC displayed the best performance, since knowledge about the harmonic frequency is provided to the system. On the other side, the AHC showed a decent performance despite being more aggressive than the SHC, with its main advantage being that no knowledge about the frequency is needed.

The AISE is ultimately discarded as a viable estimation method for the final implementation. Despite extensive efforts devoted to parameter tuning, it is not possible to reproduce the behavior reported in [4]. Significant discrepancies are observed between the estimated and the true velocity and acceleration, especially under dynamic conditions. These inconsistencies lead to degraded tracking performance and prevent the reliable use of the AISE within the landing architecture. For these reasons, only the Kalman Filter is retained for the experimental validation. The simulation environment proved to be a great transition between preliminary simulations and in-flight experiments. It showed important features regarding the behavior of the systems that were later confirmed with the experimental tests.

References

[1] G. Gozzini, D. Invernizzi, S. Panza, M. Giurato, and M. Lovera, "Air-to-Air Automatic Landing of Unmanned Aerial Vehicles: a quasi time-optimal hybrid strategy," *IEEE*, 2020.

[2] G. Roggi, G. Gozzini, D. Invernizzi, and M. Lovera, "Vision-Based Air-to-Air Autonomous Landing of Underactuated VTOL UAVs," *IEEE*, vol. 29, pp. 2338–2349, 2024.

[3] M. Manzoni, R. Rubinacci, G. Gozzini, and D. Invernizzi, "Control-oriented modeling and hierarchical control of multirotor uavs for research and teaching purposes," in *2nd IFAC Workshop on Aerospace Control Education - WACE 2024*, vol. 58, pp. 147–152, 2024.

[4] S. Verma and D. S. Bernstein, "Frenet-Serret-Based Trajectory Prediction," Tech. Rep. 1, Cornell University, https://doi.org/10.48550/arXiv.2501.04273, 01 2025.

[5] L. Zuccarian, D. Invernizzi, and M. Lovera, "Global robust attitude tracking with torque disturbance rejection via dynamic hybrid feedback," *Elsevier Ltd*, vol. Automatica 144, 2022.