Dynamic Landing of an Autonomous Quadrotor on a Moving Platform in Turbulent Wind Conditions

Aleix Paris, Brett T. Lopez, and Jonathan P. How

Abstract—Autonomous landing on a moving platform presents unique challenges for multirotor vehicles, including the need to accurately localize the platform, fast trajectory planning, and precise/robust control. Previous works studied this problem but most lack explicit consideration of the wind disturbance, which typically leads to slow descents onto the platform. This work presents a fully autonomous vision-based system that addresses these limitations by tightly coupling the localization, planning, and control, thereby enabling fast and accurate landing on a moving platform. The platform’s position, orientation, and velocity are estimated by an extended Kalman filter using simulated GPS measurements when the quadrotor-platform distance is large, and by a visual fiducial system when the platform is nearby. The landing trajectory is computed online using receding horizon control and is followed by a boundary layer sliding controller that provides tracking performance guarantees in the presence of unknown, but bounded, disturbances. To improve the performance, the characteristics of the turbulent conditions are accounted for in the controller. The landing trajectory is fast, direct, and does not require hovering over the platform, as is typical of most state-of-the-art approaches. Simulations and hardware experiments are presented to validate the robustness of the approach.

Index Terms—Unmanned aerial vehicles, autonomous vehicles, landing on a moving platform, disturbance compensation.

Supplementary Material

Video of the paper summary and experiments is available at https://youtu.be/xKolrY4riJQ.

I. INTRODUCTION

Autonomous unmanned aerial vehicles (UAVs) are becoming more and more popular in industry for their flexibility and fast deployment, and have demonstrated their usefulness in applications such as aerial photography for topology and agriculture [1]–[4], search and rescue operations [5], [6], and mapping [7]–[9]. The large recent growth in online shopping has also attracted interest in reducing package shipment time and costs, and UAVs provide an efficient alternative to delivery trucks [10], [11]. However, their payload and flight time is limited, so several researchers have investigated using a truck-drone delivery system [12], [13]. The UAVs in current truck-drone delivery methods can only take off and land when the truck is stopped visiting a customer node, which has substantial synchronization costs. Thus, autonomous landing on a moving truck is a promising research direction [14]. This work presents a system capable of landing a quadrotor on a moving platform, even in the presence of turbulent wind. We demonstrate a boundary layer sliding controller (BLSC) which takes into account these conditions, a planner with changing objectives that allows a fast maneuver, and a vision-based extended Kalman filter (EKF) to estimate the moving platform’s state. The UAV’s state estimator currently uses a motion capture system, but the important information in the maneuver we demonstrate (the relative position of the UAV and the moving platform) is estimated onboard by the quadrotor using a vision system.

A. Related Work

Autonomous UAV landing has been investigated by several researchers. Ref. [15] landed a drone on a static kayak in a reservoir with mild wind and water ripple conditions, but the image processing was done off-board and the landing time was close to 1 min. Ref. [16] developed a system capable of landing a UAV on a moving platform with PID control, an EKF estimator, and an AprilTag visual fiducial bundle. However, the computation was also done off-board, and there was no relative wind present: the platform’s speed was just 0.18m/s and the tests were done indoors. Ref. [17] demonstrated a quadrotor landing on a moving platform using only onboard sensing and computation, but the environment was not turbulent due to the tests being indoors and the platform was moving relatively slowly at 1.2m/s. Furthermore, besides two cameras, a distance sensor was required to estimate the UAV-platform relative pose. Ref. [18] demonstrated an autonomous landing of a quadrotor on a car moving at 14m/s by detecting an AprilTag on its roof. A proportional navigation-based guidance law was used for the approach phase, and a PID controller for the landing phase, with no disturbance rejection considerations. The landing maneuver consisted of acquiring the tag while hovering, and the descent was initiated once the quadrotor stabilized over it. Additionally, the quadrotor used was large and had a broad sensor suite, including a downward-facing camera, a three-axis gimbaled camera to track the target, and an inertial navigation system. Furthermore, besides the...
ground vehicle’s GPS coordinates, the quadrotor also used the ground vehicle’s IMU data to improve its estimation of the landing platform’s state. Ref. [19] also demonstrated an outdoors landing, but the landing platform’s speed was only 0.5m/s and the landing maneuver took 12–20 s to complete. Moreover, the ground vehicle’s wheel encoders data was used to estimate its state. Ref. [20] used model predictive control (MPC) to land a quadrotor on a moving ship. Both the UAV and the ship collaborated to reach their goal. The waves were modeled as sinusoidal, but the wind disturbances considered were not turbulent – this approach only compensated the effects caused by a steady-state wind. Furthermore, the vehicle needed to hover above the platform for at least 5 s, and the ship was simulated. A simulated boat landing was also carried out in [21], where the platform’s state was estimated fusing GPS and visual measurements. This work incorporated a velocity feed-forward term to the controller to catch the platform, but its only consideration of external wind was an offset of the hovering position to ensure the target was inside the field of view of the downwards-facing camera, and the landing maneuver lasted 24 s. Ref. [22] developed an adaptive controller to track a ground vehicle with only relative position data from ArUco tags, and tested it in outdoor experiments at 5.6m/s. But this approach only considered disturbances due to ground effect, and the landing maneuver needed 20 s for tracking and 10 s for descent.

In summary, most current approaches for quadrotor landing on moving platforms involve hovering above the platform for a period of time to visually acquire it, which is then followed by a relatively slow descent. In contrast, our approach investigates a direct trajectory to the landing platform, which has the possible advantages of providing faster landings and enabling several UAVs to approach the platform at the same time from different directions (improving the utilization of this limited resource). Additionally, most of current approaches do not make special considerations to reject the turbulent wind present near the target vehicle, and thus safety in challenging conditions cannot be guaranteed.

B. Contributions

This paper demonstrates a vision-based system capable of dynamic landing (i.e., the multirotor does not need to hover above the vehicle before descending) which also accounts for the turbulent conditions that would be present near a rapidly-moving ground vehicle. The resulting framework allows thus a maneuver that will be crucial for efficient truck-drone delivery systems. The contributions of this paper are as follows:

- Demonstration of a boundary layer sliding controller to incorporate and compensate for turbulence based on a model of the conditions near the landing platform.
- An algorithm for computing fast, vision-based dynamic landing maneuvers, is demonstrated both in simulation and hardware experiments that include challenging steady/turbulent wind conditions.

II. SYSTEM OVERVIEW

This work addresses the current limitations of landing on a moving platform by: 1) using optimization-based trajectory generation to enable dynamic landing; and 2) using robust control to explicitly compensate for turbulent wind conditions. The system that achieves this is comprised of several components, described in this section and shown in Fig. 2.

A. Finite State Machine

The quadrotor’s behavior is determined by a finite state machine (FSM) comprised of four states:

1) **Stand By**: This is the initial state of the quadrotor, which consists of taking off and hovering at a predefined altitude above the starting point. The FSM then transitions to **Search** mode.

2) **Search**: The quadrotor uses simulated GPS coordinates of the unmanned ground vehicle (UGV) —as explained in Section II-D— to predict a rendezvous location and flies there. When the front-facing camera detects the landing platform as described in Section II-E, the state automatically switches to **Landing**.

3) **Landing**: In this mode, the quadrotor approaches the target following a direct trajectory towards it. When the distance and relative velocity UAV-UGV are below threshold values, the quadrotor switches to the **End** mode. If the last detection happened more than 0.8s ago, the mode returns to **Search**.

4) **End**: motors are stopped, maneuver is finished.

B. Trajectory Planning: Model Predictive Control

The planner solves a convex optimization problem with changing objectives depending on the state of the FSM. In **Search** mode, the UAV-UGV rendezvous point is predicted by assuming a constant linear velocity and yaw rate for the UGV, and a constant velocity for the UAV. This position is then offset by a small distance backwards in the direction of the UGV to ensure target detection by the front-facing camera, and is used by the planner as the final position of the trajectory, while the final velocity is the UGV’s. The time taken to reach the UGV is minimized, to reduce delivery turnaround. In **Landing** mode, the planner initially minimizes the jerk to obtain a trajectory which ensures adequate tag acquisition. As the UAV approaches the target, the effect of disturbances increases so the planner minimizes the time spent in the turbulent area. The final position of the planned trajectory is the vertical tag’s position (offset...
a few cm backwards) and predicted ahead (by an amount that depends on the computation time) assuming constant linear/angular velocities. The final velocity of the trajectory is set to match the UGV’s.

The minimum jerk approach produces smooth trajectories and has a long heritage for planning quadrotor paths [17], [23], [24]. The trajectory is re-planned using an MPC approach every time a new estimate of the platform’s position is obtained. We used CVXGEN [25] to generate the code to solve the following convex optimization problem

\[
\min_{x, u} J = \sum_{t=0}^{N} \ell (\dot{u}, d) [t] \\
\text{subject to } x[t+1] = Ax[t] + B\dot{u}[t] \\
\|a[t]\|_{\infty} \leq a_{\max}, \text{ and } \|\dot{u}[t]\|_{\infty} \leq j_{\max} \\
x[0] = x_0, x[N] = x_f \\
\text{for } t = 0 \ldots N
\]

where \( N \) is the timestep when the quadrotor has to reach the final state \( x_f \). \( \ell \) is a quadratic cost function, \( \dot{u} \) is the open-loop control input (the jerk of the trajectory), \( d \) is the UAV-tag distance, \( x \) is the position, velocity, and acceleration of the UAV, \( A \) and \( B \) are, respectively, the state and input matrices for a triple integrator, and \( a \) is the subvector of \( x \) representing the acceleration. Note that we can plan using this linear model for a nonlinear system because the nonlinear dynamics of the quadrotor are canceled by the ancillary controller, as explained next.

C. Ancillary Controller: Boundary Layer Sliding Controller

While MPC has been used extensively in industry [26], the control of systems with nonlinear dynamics requires expensive optimization. Sliding control [27] has proven to be effective in quadrotors [28], [29]. This control strategy guarantees bounds on the tracking error and has been combined with MPC [30], [31]. In our approach, we derive a nonlinear ancillary controller using sliding control that models the disturbances found near the landing platform.

In quadrotors, the attitude dynamics are much faster than the position dynamics, and thus control of both can be decoupled [32]: the output of a controller is the setpoint for the other. The position and velocity controller is derived in this section to account for the turbulent wind present near the landing platform, and the attitude control is performed by a quaternion-based controller [33].

The following derives the BLSC. Define the manifold \( S(t) \) by \( s = \dot{x} + \lambda \ddot{x} = 0 \), where \( \dot{x} = x - x_d \) and \( \lambda > 0 \). The objective of sliding control is to maintain \( s = 0 \) at all times. If the control action’s frequency is high enough, zero tracking error is guaranteed [27]. This high control action is impractical in many applications because of actuator limits and the excitation of unmodelled dynamics. An approach taken in [28], [31] is BLSC, where the control discontinuity is smoothed in a thin boundary layer of thickness \( \Phi \):

\[
\mathcal{B} := \{ s : |s| \leq \Phi \} \tag{2}
\]

Consider a system whose dynamics can be expressed as

\[
\dot{x} = f(x) + b(x) u + d \tag{3}
\]

where \( d \) is the disturbance. Then, the BLSC strategy is

\[
u = \dot{b}^{-1} [\dot{x}_d - \lambda \ddot{x} - \dot{f}(x) - K \operatorname{sat}\left(\frac{s}{\Phi}\right)] \tag{4}
\]

where \( \operatorname{sat}(\cdot) \) is the saturation function, \( \dot{f} \) is the estimated acceleration caused by drag, and \( K \) is determined by the uncertainty in the dynamics and the disturbance of the system. As noted before, we can plan in (1) using linear MPC because of the cancellation of \( f \) in (3) and (4).

We generate turbulence using leaf blowers as shown in Figs. 1 and 4. The turbulent wind parameters are the mean \( \nu_w \) and standard deviation \( \sigma \) of the speed. Define \( V = \nu + \nu_w \) where \( \nu \) is the quadrotor’s speed. Then, \( V \) is the total wind speed relative to the UAV and \( \dot{f} = \hat{c} ||V|| V \), where \( \hat{c} \) is the estimated drag coefficient of the quadrotor.

The quadrotor plans a trajectory to approach the UGV in the direction it is facing to match its speed, and thus is never outside the wind field generated by the leaf blowers during the landing maneuver. Therefore, it is reasonable to assume that this wind field is constant in the directions perpendicular to where the leaf blowers point to. The true acceleration caused by drag is

\[
f = c ||V| + 2 \sigma u_w|| (V + 2 \sigma u_w) \tag{5}
\]

where \( u_w \) is a unit vector in the direction of the wind.

The variation of \( b \) is very small for a UAV with constant weight. Therefore, \( \beta = (b_{\max}/b_{\min}) \) is approximately 1, where \( b_{\max} \) and \( b_{\min} \) are the maximum and minimum control gains respectively (or throttle gains in the context of quadrotors). Thus, \( K \) is simplified as [27] \( K = F + \eta \), where \( \eta > 0 \) is a constant in the sliding condition

\[
\frac{1}{2} \frac{d}{dt} s^2 \leq -\eta |s|. \tag{6}
\]

The larger the \( \eta \), the faster the system will reach the sliding surface. Nevertheless, \( K \) should only be as large as the disturbance magnitude requires to avoid a high-frequency control signal. \( F \) is

\[
F = |f - \dot{f}| \leq \dot{F} \tag{7}
\]

\[
\dot{F} = (\hat{c} + \hat{c}) ||V|| (V + 2 \sigma u_w) - \hat{c} ||V|| V
\]

where \( \hat{c} > 0 \) is a bound on the absolute value of the drag coefficient error \( c - \hat{c} \). By taking the sign that makes this coefficient larger, we have defined \( K \). In our application, the quadrotor moves towards the generated wind and therefore this occurs when the \( 2 \sigma \) is increasing the magnitude of \( \nu_w \).

D. Landing Platform Estimation: Extended Kalman Filter

To estimate the state of the moving platform, an extended Kalman filter (EKF) is used. This filtering algorithm minimizes the mean of the squared error and has demonstrated its efficacy in robot localization [34]–[36]. The state vector of the platform is \( \mathbf{x}_p = [p_x, p_y, v_y, \theta, \dot{\theta}]^\top \), where \( p_x, p_y \) are the 2D coordinates, \( v_y \) is the magnitude of the velocity, \( \theta \) is
the orientation angle with respect to the $x$-axis, and $\dot{\theta}$ is the angular velocity. Since real-world roads are mostly horizontal and in particular our experiments were carried on completely flat surfaces, the velocity in the $z$-direction is not estimated. The moving platform is modeled as a unicycle with dynamics

$$\dot{x}_p(t) = f_p(x_p(t)) + w(t),$$

where $w(t)$ is the process noise, assumed to be a white Gaussian noise. We consider a constant linear and angular velocity, and the UGV dynamics are

$$\dot{p}_x = v_p \cos(\theta), \quad \dot{v}_p = 0,$$
$$p_y = v_p \sin(\theta), \quad \dot{\theta} = 0.$$

The measurement vector is $z_p = [p_x, p_y, \theta]^	op$, and when a measurement is received, the EKF approach is used to perform an update of the estimated state. The measurements are obtained in two ways. First, when the quadrotor is far from the platform (that is, in the Search state defined in Section II-A), these measurements are obtained by adding a white Gaussian noise to the ground truth pose of the vehicle (obtained from the motion capture data) to simulate inaccurate GPS measurements that a ground vehicle could provide to the UAV for the rendezvous. Note that receiving $\theta$ is not necessary to estimate the orientation of a moving platform because its motion could be used to infer that quantity. Nevertheless, $\theta$ measurements are used in this work, which enables testing for static platform experiments. The update frequency is 2Hz, which is realistic for UAV applications [37]. These first set of measurements are simply used to help the UAV locate the ground vehicle, but that information could be estimated without requiring a link between the two vehicles.

Second, and most importantly, when the quadrotor is near the ground vehicle and detects the onboard tag, we fuse both the simulated GPS and the visual detection measurements to estimate $x_p$. The vision-based detection is explained in the next subsection, and it provides a far more accurate estimate of the position and orientation of the tag/platform. The 2D positions $p_x$ and $p_y$ and the heading angle $\theta$ are then used to update the platform’s state, and the estimated velocity $v_p$ is incorporated into the vector $x_f$ in (1) to ensure the quadrotor matches the moving platform’s speed at the landing point.

### E. Visual detection: AprilTag visual fiducial system

When the UAV is relatively close to the platform, visual estimation provides more accurate UAV-UGV poses than GPS. We used the AprilTag visual fiducial system [38] to obtain them, and a ROS wrapper [39] based on AprilTag 2 [40] to interface with the core detection algorithm. A tag bundle is a set of several coplanar tags used simultaneously by the visual fiducial system: the algorithm extracts the information of all of them to estimate a single “bundle pose”. Thus, they are useful when accurate detection is required, which is the case in this paper. Additionally, by using tags of different sizes, detection at various distances is ensured. We used a tag bundle comprised of a $14 \times 14$cm tag on top of a $5 \times 5$cm tag. Despite the relatively small bundle size, it can be detected at a maximum distance of approximately 3.5m, and a minimum distance of about 5cm. The bundle can be seen in Fig. 4.

## III. EXPERIMENTAL RESULTS

### A. Simulations

Due to the difficulty of accurately modeling the complex turbulent wind effects on a quadrotor, our simulations consider steady wind but serve to test our finite state machine, planner, controller, and estimator. For space constraints, simulation experiments are not analyzed here, but the video accompanying the paper (linked in the Supplementary Material Section) shows a landing simulation.

### B. Hardware

Static and moving platform landing tests were done (see Figs. 1 and 4), both of which had turbulent wind at the landing site from the leaf blowers. The quadrotor used for the hardware experiments (Fig. 3) weighs 0.564kg including the 1500mAh 3S battery. It is $36 \times 29$cm (approximately half the platform size) and its thrust-to-weight ratio is 1.75. The onboard computer is a Qualcomm Snapdragon Flight APQ8074, whose front-facing camera provides $640 \times 480$ black-and-white images at a rate of 30fps to the AprilTag detection module. Hover tests were carried out to determine $b$ in (4). To measure the drag coefficient $c$ and the bound on its
error \( \tilde{c} \), we performed tests with the quadrotor flying in front of a strong wind and the accurate pitch angle was obtained by the motion capture system. By balancing forces, the drag could be determined, yielding \( \tilde{c} \). Additionally, we used the IMU utils package [41] to compute the Snapdragon’s IMU accelerometer and gyroscope noise density and bias random walk. The Kalibr visual-inertial toolbox then used this IMU intrinsic information to find the camera-IMU transform [42].

C. Static Platform Experiments

1) Experiment Setting: The first set of hardware experiments presents a static platform in front of an array of 5 leaf blowers, as shown in Fig. 4. The leaf blowers are set at two different heights and point to the negative \( x \)-direction. The platform is composed of a 60\( \times \)60cm horizontal plate and a 60\( \times \)30cm vertical plate, which has attached the tag bundle described in II-E.

The parameters \( v_w \) and \( \sigma \) of the turbulent wind were measured at distances \( l \) from the vertical platform every 0.5m until \( l = 3.5 \)m, which is approximately the tag detection range. For every \( l \), a measurement of the speed was taken every second for 60s using a high-precision hot-wire anemometer. Fig. 5 shows the data obtained. Interestingly, at \( l = 0.5 \)m, the mean speed decreases while the standard deviation increases to its maximum value. We believe this is due to the vortices caused when the flow traverses the vertical platform.

2) Results: First, we tested a standard BLSC that does not take into account the turbulent wind generated by the leaf blowers, that is, the factors \( f \) and \( K \) only consider the drag generated by the quadrotor’s speed relative to the ground. Fig. 6 shows the tracking and estimation performance obtained. The quadrotor starts at \( p_q = (-3.5, 2.0, 1.3) \)m, and the platform is located at \( p_p = (1.1, -1.1, 0.8) \)m. As expected, the tracking performance is poor, and the landing takes a long time: 27.2s since the first tag detection, which occurs at \( t_d = 3.3 \)s (indicated with a vertical dashed line). Also note that the platform’s estimation improves after \( t_d \). In the video, the covariance ellipses for the tag’s 2D position can be seen, and decrease abruptly in size at \( t_d \).

Next, we used our BLSC with the same initial conditions as in Fig. 6 to compare its improvement. The results are shown in Fig. 7. It can be seen that the tracking is much better, and it only worsens slightly when the quadrotor is inside the wind field, after the time of the first tag detection \( t = 2.5 \)s. The landing time is just 3.2s (measured since the first tag detection) even in challenging conditions, which is 8.5 times faster compared to the standard BLSC.

D. Moving Platform Experiments

1) Experiment Setting: To fully test our approach, we also performed landing experiments on a moving platform that is mounted on top of a dolly. The Clearpath Jackal was used as the ground vehicle that tows the landing platform (see Fig. 1) and carries two of the leaf blowers. This provided turbulent air at the landing pad even though the vehicles were not moving very quickly. The distance from the leaf blowers to the platform is such that this turbulent wind follows the same plot as Fig. 5.

2) Results: A standard BLSC was also tested first for this experiment setting. There was considerable tracking error and the quadrotor was not able to land on the platform before the vehicle arrived at the final point (see video linked in the Supplementary Material Section for details). The results obtained using our BLSC are shown in Fig. 8. The quadrotor starts at \( p_q = (-1.5, 2.7, 0.9) \)m, and the UGV starts at \( p_p = (-3.9, 0.1, 0.4) \)m. When the maneuver begins, the UGV is commanded to move at 1m/s and rotate to its right at a rate of 4°/s. The tag is detected at \( t_d = 4.7 \)s, and the landing time is 6.8s. Note that the quadrotor is at approximately 4m from the moving platform at \( t_d \), a value larger than for the static experiment (1.5m).
Fig. 7. Position and velocity tracking and estimation performance of a static platform experiment using our BLSC. The tracking error remains small during the flight.

Fig. 8. Position and velocity tracking and estimation performance of a moving platform experiment using our BLSC. The quadrotor quickly matches the UGV’s velocity and is able to track successfully the planned trajectory.

IV. CONCLUSION

This paper developed a boundary layer sliding controller to allow a quadrotor to fly in challenging conditions, and demonstrated its effectiveness by performing fast landing experiments. Future work includes incorporating adaptation to allow for more varied flight conditions, and landing on the back of a pickup truck driving outdoors, to test this work’s approach in a more realistic setting. This will additionally require visual-inertial odometry (VIO) for onboard-only state estimation.

ACKNOWLEDGMENT

Work supported by Ford Motor Company, and the authors would like to thank Michael Everett for his help configuring the UGV.

REFERENCES

[1] F. Mancini, M. Dubbini, M. Gattelli, F. Stecchi, S. Fabbri, and G. Gabbianelli, “Using unmanned aerial vehicles (UAV) for high-resolution reconstruction of topography: The structure from motion approach on coastal environments,” Remote Sensing, vol. 5, no. 12, pp. 6880–6898, 2013.
[2] K. Anderson and K. J. Gaston, “Lightweight unmanned aerial vehicles will revolutionize spatial ecology,” Frontiers in Ecology and the Environment, vol. 11, no. 3, pp. 138–146, 2013.
[3] E. Honkavaara, H. Saari, J. Kaivosoja, I. Polönen, T. Hakala, P. Likey, J. Mäikynen, and L. Pesonen, “Processing and assessment of spectrometric, stereoscopic imagery collected using a lightweight UAV spectral camera for precision agriculture,” Remote Sensing, vol. 5, no. 10, pp. 5006–5039, 2013.
[4] P. Tokekar, J. Vander Hook, D. Mulla, and V. Isler, “Sensor planning for a symbiotic UAV and UGV system for precision agriculture,” IEEE Transactions on Robotics, vol. 32, no. 6, pp. 1498–1511, 2016.
[5] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. L. Gria, F. Rue, M. Suppa, and D. Burschka, “Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue,” IEEE robotics & automation magazine, vol. 19, no. 3, pp. 46–56, 2012.
[6] J. Scherer, S. Yahyanejad, S. Hayat, E. Yannaz, T. Andre, A. Khan, V. Yukadinovic, C. Betsetter, H. Hellwagner, and B. Rinne, “An autonomous multi-UAV system for search and rescue,” in Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use. ACM, 2015, pp. 33–38.
[7] F. Caballero, L. Merino, J. Ferruz, and A. Oller, “Vision-based odometry and SLAM for medium and high altitude flying UAVs,” Journal of Intelligent and Robotic Systems, vol. 54, no. 1-3, pp. 137–161, 2009.

[8] P. Schmuck and M. Chli, “Multi-UV collaborative monocular SLAM,” in 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017, pp. 3863–3870.

[9] A. R. Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, “Ultimate SLAM? Combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios,” IEEE Robotics and Automation Letters, vol. 3, no. 2, pp. 994–1001, 2018.

[10] J. Scott and C. Scott, “Drone delivery models for healthcare,” in Proceedings of the 50th Hawaii international conference on system sciences, 2017.

[11] S. M. Shavarani, M. G. Nejad, F. Rismanchian, and G. Izbirak, “Application of hierarchical facility location problem for optimization of a drone delivery system: a case study of Amazon Prime Air in the city of San Francisco,” The International Journal of Advanced Manufacturing Technology, vol. 95, no. 9-12, pp. 3141–3153, 2018.

[12] A. M. Ham, “Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming,” Transportation Research Part C: Emerging Technologies, vol. 91, pp. 1–14, 2018.

[13] S. M. Fernandez, T. Harbison, T. Weber, R. Sturges, and R. Rich, “Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm,” Journal of Industrial Engineering and Management (JEM), vol. 9, no. 2, pp. 374–388, 2016.

[14] A. Otto, N. Agatz, J. Campbell, B. Golden, and E. Pesch, “Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey,” Networks, vol. 72, no. 4, pp. 411–458, 2018.

[15] T. Venugopalan, T. Taher, and G. Barbastathis, “Autonomous landing of an unmanned aerial vehicle on an autonomous marine vehicle,” in 2012 Oceans. IEEE, 2012, pp. 1–9.

[16] O. Araar, N. Aouf, and I. Vitanov, “Vision based autonomous landing of multirotor UAV on moving platform,” Journal of Intelligent & Robotic Systems, vol. 85, no. 2, pp. 369–384, 2017.

[17] D. Falanga, A. Zanchettin, A. Simovic, J. Delmerico, and D. Scaramuzza, “Vision-based autonomous quadrotor landing on a moving platform,” in 2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR). IEEE, 2017, pp. 200–207.

[18] A. Borowczyk, D.-T. Nguyen, A. Phu-Van Nguyen, D. Q. Nguyen, D. Sausišč, and J. Le Ny, “Autonomous landing of a multicopter micro air vehicle on a high velocity ground vehicle,” IFAC-PapersOnLine, vol. 50, no. 1, pp. 10488–10494, 2017.

[19] B.-Y. Xing, F. Pan, X.-X. Feng, W.-X. Li, and Q. Gao, “Autonomous landing of a micro aerial vehicle on a moving platform using a composite landmark,” International Journal of Aerospace Engineering, vol. 2019, 2019.

[20] L. Persson and B. Wahlberg, “Model predictive control for autonomous ship landing in a search and rescue scenario,” in AIAA Scitech 2019 Forum, 2019, p. 1169.

[21] J. S. Wynn and T. W. McLain, “Visual servoing with feed-forward for precision shipboard landing of an autonomous multicopter,” in 2019 American Control Conference (ACC). IEEE, 2019, pp. 3928–3935.

[22] M. Bhargavapur, A. K. Shastry, H. Sinha, S. R. Sahoo, and M. Kothari, “Vision-based autonomous tracking and landing of a fully-actuated rotorcraft,” Control Engineering Practice, vol. 89, pp. 113–129, 2019.

[23] J. Yu, Z. Cai, and Y. Wang, “Minimum jerk trajectory generation of a quadrotor based on the differential flatness,” in Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference. IEEE, 2014, pp. 832–837.

[24] S. K. Phang, S. Lai, F. Wang, M. Lan, and B. M. Chen, “Systems design and implementation with jerk-optimized trajectory generation for UAV calligraphy,” Mechatronics, vol. 30, pp. 65–75, 2015.

[25] J. Mattingley and S. Boyd, “CVXGEN: A code generator for embedded convex optimization,” Optimization and Engineering, vol. 12, no. 1, pp. 1–27, 2012.

[26] S. J. Qn and T. A. Badgwell, “A survey of industrial model predictive control technology,” Control engineering practice, vol. 11, no. 7, pp. 733–764, 2003.

[27] J.-J. E. Slotine, W. Li, et al., Applied nonlinear control. Prentice Hall Englewood Cliffs, NJ, 1991, vol. 199, no. 1.

[28] K. Runchaloon and V. Srichatrapimuk, “Sliding mode control of quadrotor,” in 2013 The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAECEE). IEEE, 2013, pp. 552–557.

[29] Y. Yang and Y. Yan, “Attitude regulation for unmanned quadrotors using adaptive fuzzy gain-scheduling sliding mode control,” Aerospace Science and Technology, vol. 54, pp. 208–217, 2016.

[30] M. Rubagotti, D. M. Raimondo, A. Ferrara, and L. Magni, “Robust model predictive control with integral sliding mode in continuous-time sampled-data nonlinear systems,” IEEE Transactions on Automatic Control, vol. 56, no. 3, pp. 556–570, 2010.

[31] B. T. Lopez, J.-J. Slotine, and J. P. How, “Robust collision avoidance via sliding control,” in 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018, pp. 2962–2969.

[32] J. P. How, E. Frazoli, and G. V. Chowdhary, “Linear flight control techniques for unmanned aerial vehicles,” Handbook of Unmanned Aerial Vehicles, pp. 529–576, 2015.

[33] B. T. Lopez, “Low-latency trajectory planning for high-speed navigation in unknown environments,” Ph.D. dissertation, Massachusetts Institute of Technology, 2016.

[34] E. Kiriy and M. Buehler, “Three-state extended Kalman filter for mobile robot localization,” McGill University., Montreal, Canada, Tech. Rep. TR-CIM, vol. 5, p. 23, 2002.

[35] A. Georgiev and P. K. Allen, “Localization methods for a mobile robot in urban environments,” COLUMBIA UNIV NEW YORK NEW YORK United States, Tech. Rep., 2004.

[36] L. Teslić, I. Skrjanc, and G. Klančar, “EKF-based localization of a wheeled mobile robot in structured environments,” Journal of Intelligent & Robotic Systems, vol. 62, no. 2, pp. 187–203, 2011.

[37] A. A. A. Salih, N. L. A. C. A. Zaini, and A. Zahir, “The suitability of GPS receivers update rates for navigation applications,” in Proceedings of World Academy of Science, Engineering and Technology, no. 78. World Academy of Science, Engineering and Technology (WASET), 2013, p. 192.

[38] E. Olson, “AprilTag: A robust and flexible visual fiducial system,” in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). IEEE, May 2011, pp. 3400–3407.

[39] D. Malypa, “Guidance, navigation, control and mission logic for quadrotor full-cycle autonomy,” Master’s thesis, ETH Zurich, 2018.

[40] J. Wang and E. Olson, “AprilTag 2: Efficient and robust fiducial detection,” in 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2016, pp. 4193–4198.

[41] W. Gao, “imu_utils: A ROS package tool to analyze the IMU performance,” https://github.com/gaowenli/imu_utils, 2018.

[42] P. Furgale, H. Sommer, J. Maye, J. Rehder, T. Schneider, and L. Oth, “Kalibr: A unified camera/IMU calibration toolbox,” https://github.com/ethz-asl/kalibr, 2014.