Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 50-1 (2017) 10488–10494

Autonomous Landing of a Multirotor Micro Air Vehicle on a High Velocity Ground Vehicle ⋆

Alexandre Borowczyk * Duc-Tien Nguyen *
André Phu-Van Nguyen * Dang Quang Nguyen *
David Saussié * Jerome Le Ny *

* *Mobile Robotics and Autonomous Systems Laboratory, Polytechnique Montreal and GERAD, Montreal, Canada (e-mail: {alexandre.borowczyk, duc-tien.nguyen, andre-phu-van.nguyen, dang-quang.nguyen, d.saussie, jerome.le-ny}@polymtl.ca).*

Abstract: While autonomous multirotor micro aerial vehicles (MAVs) are uniquely well suited for certain types of missions benefiting from stationary flight capabilities, their more widespread usage still faces many hurdles, due in particular to their limited range and the difficulty of fully automating the deployment and retrieval. In this paper we address these issues by solving the problem of the automated landing of a quadcopter on a ground vehicle moving at relatively high speed. We present our system architecture, including the structure of our Kalman filter for the estimation of the relative position and velocity between the quadcopter and the landing pad, as well as our controller design for the full rendezvous and landing maneuvers. The system is experimentally validated by successfully landing in multiple trials a commercial quadcopter on the roof of a car moving at speeds of up to 50 km/h.

Keywords: Kalman filters, Autonomous vehicles, Mobile robots, Guidance systems, Computer vision, Aerospace control

1. INTRODUCTION

The ability of multirotor micro aerial vehicles (MAVs) to perform stationary hover flight makes them particularly interesting for a wide variety of applications, e.g., site surveillance, parcel delivery, or search and rescue operations. At the same time however, they are challenging to use on their own because of their relatively short battery life and short range. Deploying and recovering MAVs from mobile Ground Vehicles (GVs) could alleviate this issue and allow more efficient deployment and recovery in the field. For example, delivery trucks, public buses or marine carriers could be used to transport MAVs between locations of interest and allow them to recharge periodically (Garone et al., 2014; Mathew et al., 2015). For search and rescue operations, the synergy between ground and air vehicles could help save precious mission time and would pave the way for the efficient deployment of large fleets of autonomous MAVs.

The idea of better integrating GVs and MAVs has indeed already attracted the attention of multiple car and MAV manufacturers (Kolodny, 2016; Lardinois, 2016). Research groups have previously considered the problem of landing a MAV on a mobile platform, but most of the existing work is concerned with landing on a marine platform or precision landing on a static or slowly moving ground target. Lange et al. (2009) provide an early example, where a custom visual marker made of concentric rings was created to allow for relative pose estimation, and control was performed using optical flow and velocity commands. More recently, Yang et al. (2015) used the ArUco library from Garrido-Jurado et al. (2014) as a visual fiducial and IMU measurements fused in a Square Root Unscented Kalman Filter for relative pose estimation. The system however still relies on optical flow for accurate velocity estimation. This becomes problematic as soon as the MAV aligns itself with a moving ground platform, at which point the optical flow camera suddenly measures the velocity of the platform relative to the MAV instead of the velocity of the MAV in the ground frame.

Muskardin et al. (2016) developed a system to land a fixed wing MAV on top of a moving GV. However, their approach requires that the GV cooperates with the MAV during the landing maneuver and makes use of expensive RTK-GPS units. Kim et al. (2014) show that it is possible to land on a moving target using simple color blob detection and a non-linear Kalman filter, but test their solution only for speeds of less than 1 m/s. Most notably, Ling (2014) shows that it is possible to use low cost sensors combined with an AprilTag fiducial marker (Olson, 2011) to land on a small ground robot. He further demonstrates different methods to help accelerating the AprilTag detection. He notes in particular that as a quadcopter pitches forward to follow the ground platform, the bottom facing camera frequently loses track of the visual target, which stresses the importance of a model-based estimator such as a Kalman filter to compensate.

⋆ This work was partially supported by CFI JELF Award 32848 and a hardware donation from DJI.

2405-8963 © 2017, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2017.08.1980

The references above address the terminal landing phase of the MAV on a moving platform, but a complete system must also include a strategy to guide the MAV towards the GV during its approach phase. Proportional Navigation (PN) (Kabamba and Girard, 2014) is most commonly known as a guidance law for ballistic missiles, but can also be used for UAV guidance. Holt and Beard (2010) develop a form of PN specifically for road following by a fixed-wing vehicle and show that it is suitable for use with visual feedback coming from a gimbaled camera. Gautam et al. (2015) compare pure pursuit, line-of-sight and PN guidance laws to show that out of the three, PN is the most efficient in terms of the total required acceleration and the time required to reach the target. On the other hand, within close range of the target PN becomes inefficient. To alleviate this problem, Tan and Kumar (2014) propose a switching strategy to move from PN to a PD controller. Finally, to maximize the likelihood of visual target acquisition for a smooth transition from PN to PD, it is possible to follow the strategy of Lin and Yang (2014) to point a gimbaled camera towards a target.

Contributions and organization of the paper. We describe a complete system allowing a multirotor MAV to land autonomously on a moving ground platform at relatively high speed, using only commercially available and relatively low-cost sensors. The system architecture is described in Section 2. Our algorithms combine a Kalman filter for relative position and velocity estimation, described in Section 3, with a PN-based guidance law for the approach phase and a PID controller for the terminal phase. Both controllers are implemented using only acceleration and attitude controls, as discussed in Section 4. Our design was tested both in simulations and through extensive experiments with a commercially available MAV, as Section 5 illustrates. To the best of our knowledge, we demonstrate experimentally automatic landing of a multirotor MAV on a moving GV traveling at the highest speed to date, with successful tests carried up to a speed of 50 km/h.

2. SYSTEM ARCHITECTURE

In this section we describe the basic elements of our system architecture, both for the GV and the MAV. Specific details for the hardware used in our experiments are given in Section 5.

The GV is equipped with a landing pad, on which we place a 30 × 30 cm visual fiducial named AprilTag designed by Olson (2011), see Fig. 4. This allows us to visually measure the 6 Degrees of Freedom (DOF) pose of the landing pad using onboard cameras. In addition, we use position and acceleration measurements for the GV. In practice, low quality sensors are enough for this purpose, and in our experiments we simply place a mobile phone on the landing pad, which can transmit its GPS data to the MAV at 1 Hz and its Inertial Measurement Unit (IMU) data at 25 Hz at most. We can also integrate the rough heading and velocity estimates typically returned by basic GPS units, based simply on successive position measurements.

The MAV is equipped with an Inertial Navigation System (INS), an orientable 3-axis gimbaled camera (with separate IMU) for target tracking purposes, as well as a camera with a wide angle lens pointing down, which allows us to keep track of the AprilTag even at close range during the very last instants of the landing maneuver. The approach phase can also benefit from having an additional velocity sensor on board. Many commercial MAVs are equipped with velocity sensors relying on optical flow methods, which visually estimate velocity by computing the movement of features in successive images, see, e.g., (Zhou et al., 2015).

Four main coordinate frames are defined and illustrated in Fig. 1. The global North-East-Down (NED) frame, denoted {N}, is located at the first point detected by the MAV. The MAV body frame {B} is chosen according to the cross "×" configuration, i.e., its forward x-axis points between two of the arms and its y-axis points to the right. The gimbaled camera frame {G} is attached to the lens center of the moving camera. Its forward z-axis is perpendicular to the image plane and its x-axis points to the right of the gimbal frame. Finally, the bottom facing rigid camera frame {C} is obtained from the MAV body frame by a 90° rotation around the z^B axis and its origin is located at the optical center of camera.

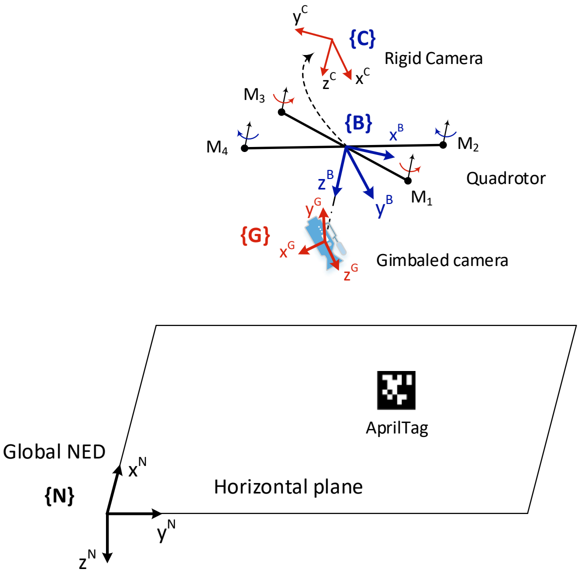

Fig. 1. Frames of reference used

3. KALMAN FILTER

Estimation of the relative position, velocity and acceleration between the MAV and the landing pad, as required by our guidance and control system, is performed by a Kalman filter running at 100 Hz. The architecture of this filter is shown in Fig. 2 and described in the following paragraphs.

3.1 Process model

In order to land on a moving target, the system estimates the three dimensional position $p(t)$, linear velocity $v(t)$ and acceleration $a(t)$ of the MAV and of the GV. Thus, the state variables can be written as

$$\mathbf{x} = [\mathbf{x}_m^\top \;\; \mathbf{x}_a^\top]^\top \in \mathbb{R}^{18} \tag{1}$$

where $\mathbf{x}_m = \left[p_m^\top \;\; v_m^\top \;\; a_m^\top\right]^\top$ and $\mathbf{x}_a = \left[p_a^\top \;\; v_a^\top \;\; a_a^\top\right]^\top$ are respectively the state vectors for the MAV and AprilTag,

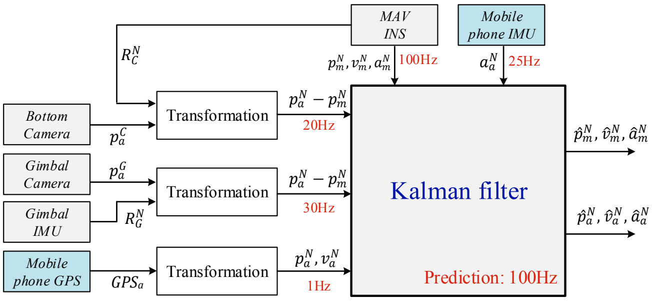

Fig. 2. Kalman filter architecture

expressed in the NED frame. The superscript $\top$ denotes the matrix transpose operation.

We use a simple second-order kinematic model for the MAV and the GV dynamics

$$a(t) = \ddot{p}(t) = w(t) \tag{2}$$

where $w(t)$ is a white noise process with power spectral density (PSD) q_w. The corresponding discrete time model using zero-order hold (ZOH) sampling is given by

$$\mathbf{x}_{k+1} = \mathrm{F}\mathbf{x}_k + \mathbf{w}_k \tag{3}$$

$$\mathrm{F} = \begin{bmatrix} \mathrm{F}_m & 0 \\ 0 & \mathrm{F}_a \end{bmatrix}, \ \mathrm{F}_m = \mathrm{F}_a = \begin{bmatrix} 1 & T_s & \frac{T_s^2}{2} \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix} \otimes \mathrm{I}_3$$

where T_s is the sampling period, $\otimes$ denotes the Kronecker product and I_3 is the 3×3 identity matrix. The process noise $\mathbf{w}_k$ is assumed to be a Gaussian white noise with a covariance matrix Q given by

$$\mathrm{Q} = \begin{bmatrix} q_{w_m} & 0 \\ 0 & q_{w_a} \end{bmatrix} \otimes \mathrm{Q}_0, \quad \mathrm{Q}_0 = \begin{bmatrix} \frac{T_s^5}{20} & \frac{T_s^4}{8} & \frac{T_s^3}{6} \\ \frac{T_s^4}{8} & \frac{T_s^3}{3} & \frac{T_s^2}{2} \\ \frac{T_s^3}{6} & \frac{T_s^2}{2} & T_s \end{bmatrix} \otimes \mathrm{I}_3$$

where q_{w_m} and q_{w_a} are the PSD of the MAV and GV acceleration, respectively. These parameters are generally chosen *a priori* as part of an empirical tuning process.

3.2 Measurement Model

In general, the measurement vector at time k is given by

$$\mathbf{z}_k = \mathrm{H}\mathbf{x}_k + \mathbf{v}_k, \tag{4}$$

where H is a known matrix, $\mathbf{v}_k$ is a Gaussian white noise with covariance matrices V_k and not correlated with the process noise $\mathbf{w}_k$. The following subsections describe the rows of the matrix H for the various kinds of sensor measurements, which we simply call H for simplicity of notation.

MAV position, velocity and acceleration from INS The INS of our MAV combines IMU, GPS and visual measurements to provide us with position, velocity and gravity compensated acceleration data directly expressed in the global NED frame

$$\mathbf{z}_k = \begin{bmatrix} p_m^\top & v_m^\top & a_m^\top \end{bmatrix}^\top, \quad \mathrm{H} = [\mathrm{I}_9 \ 0_{9\times 9}] \tag{5}$$

As mentioned previously, the velocity measurement relying on optical flow methods is not correct when the MAV flies above a moving platform. Therefore, we increase the standard deviation of the velocity noise from 0.1 m/s in the approach phase to 10 m/s in the landing phase.

Target's GPS measurements The GPS unit of the mobile phone on the GV provides position, speed and heading measurements. This information is sent to the MAV onboard computer (OBC) via a wireless link, which gives access to the landing pad's position in the global NED frame $\{\mathsf{N}\}$

$$x_a^{\mathsf{N}} \approx (la - la_0)R_E, \quad y_a^{\mathsf{N}} \approx (lo - lo_0)\cos(la)R_E$$
$$z_a^{\mathsf{N}} = al_0 - al,$$

where $R_E = 6378137\,\mathrm{m}$ is the Earth radius, la, lo, al are the latitude, longitude (both in radians) and altitude of the landing pad respectively. The subscripts $_0$ corresponds to the starting point. The above equations are valid when the current position is not too far from the starting point and under a spherical Earth assumption, but more precise transformations could be used (Groves, 2013).

The GPS heading ψ_a and speed U_a are also used to calculate the current velocity in the global NED frame

$$\dot{x}_a^{\mathsf{N}} = U_a\cos(\psi_a), \quad \dot{y}_a^{\mathsf{N}} = U_a\sin(\psi_a).$$

So the measurement model (4) is expressed as follows

$$\mathbf{z}_k = \begin{bmatrix} x_a^{\mathsf{N}} & y_a^{\mathsf{N}} & z_a^{\mathsf{N}} & \dot{x}_a^{\mathsf{N}} & \dot{y}_a^{\mathsf{N}} \end{bmatrix}^\top, \quad \mathrm{H} = [0_{5\times 9} \ \mathrm{I}_5 \ 0_{5\times 4}]$$

However, because the GPS heading measurements have poor accuracy at low speed, we discard them if $U_a < 2.5\,\mathrm{m/s}$, in which case

$$\mathbf{z}_k = \begin{bmatrix} x_a^{\mathsf{N}} & y_a^{\mathsf{N}} & z_a^{\mathsf{N}} \end{bmatrix}^\top, \quad \mathrm{H} = [0_{5\times 9} \ \mathrm{I}_3 \ 0_{5\times 6}]$$

The measurement noise covariance matrix V_k is provided by the GPS device itself.

Our GPS receiver is a low-cost device with output rate of about 1 Hz. This source of data is only used to approach the target but is insufficient for landing on the moving GV. For the landing phase, it is necessary to use the AprilTag detection with the gimbaled and bottom facing cameras.

Gimbaled Camera measurements This camera provides measurements of the relative position between the MAV and the landing pad with centimeter accuracy, at range up to 5 m. This information is converted into the global NED frame $\{\mathsf{N}\}$ by

$$p_m^{\mathsf{N}} - p_a^{\mathsf{N}} = \mathrm{R}_{\mathsf{G}}^{\mathsf{N}} p_{m/a}^{\mathsf{G}}$$

where $\mathrm{R}_{\mathsf{G}}^{\mathsf{N}}$ is the rotation matrix from $\{\mathsf{N}\}$ to $\{\mathsf{G}\}$ returned by the gimbal IMU. Therefore, the observation model (4) corresponds to

$$\mathbf{z}_k = \begin{bmatrix} x_m^{\mathsf{N}} - x_a^{\mathsf{N}} & y_m^{\mathsf{N}} - y_a^{\mathsf{N}} & z_m^{\mathsf{N}} - z_a^{\mathsf{N}} \end{bmatrix}^\top$$
$$\mathrm{H} = [\mathrm{I}_3 \ 0_{3\times 6} \ -\mathrm{I}_3 \ 0_{3\times 6}].$$

Here the standard deviation of the measurement noise is empirically set to 0.2 m. To reduce the chances of target loss, the gimbaled camera centers the image onto the AprilTag as soon as visual detection is achieved. When the AprilTag cannot be detected, we follow the control scheme proposed by Lin and Yang (2014) to point the camera towards the landing pad using the estimated line-of-sight (LOS) information obtained from the Kalman filter.

Bottom camera measurements The bottom facing camera is used to assist the last moments of the landing, when the MAV is close to the landing pad, yet too far to cut off the motors. At that moment, the gimbaled camera can not perceive the whole AprilTag but a wide angle camera like the mvBlueFOX can still provide measurements.

This camera measures the target position in the frame $\{\mathsf{C}\}$, as illustrated in Fig. 1. Hence, the observation model is the same as (3.2.3), except for the transformation to the global NED frame

$$p_m^{\mathsf{N}} - p_a^{\mathsf{N}} = \mathrm{R}_{\mathsf{C}}^{\mathsf{N}}\, p_{m/a}^{\mathsf{C}}$$

where $\mathrm{R}_{\mathsf{C}}^{\mathsf{N}}$ denotes the rotation matrix from $\{\mathsf{N}\}$ to $\{\mathsf{C}\}$.

Landing pad acceleration using the mobile phone's IMU Finally, since most mobile phones also contain an IMU, we leverage this sensor to estimate the GV's acceleration.

$$\mathbf{z}_k = \left[\ddot{x}_a^{\mathsf{N}}\ \ddot{y}_a^{\mathsf{N}}\ \ddot{z}_a^{\mathsf{N}}\right]^\top,\ \ \mathrm{H} = [0_{3\times 15}\ \mathrm{I}_3]$$
$$\mathrm{V}_k = \mathrm{diag}(0.6^2,\, 0.6^2,\, 0.6^2)\ \mathrm{m/s}^2.$$

The Kalman filter algorithm follows the standard two steps, with the prediction step running at 100 Hz and the measurement update step executed as soon as new measurements become available. The output of this filter is the input to the guidance and control system described in the next section, which is used by the MAV to approach and land safely on the moving platform.

4. GUIDANCE AND CONTROL SYSTEM

For GV tracking by the MAV, we use a guidance strategy switching between a Proportional Navigation (PN) law (Kabamba and Girard, 2014) for the approach phase and a PID for the landing phase, which is similar in spirit to the approach in (Tan and Kumar, 2014).

The approach phase is characterized by a large distance between the MAV and the GV and the absence of visual localization data. Hence, in this phase, the MAV has to rely on the data transmitted by the GV's GPS and IMU. The goal of the controller for this phase is to follow an efficient "pursuit" trajectory, which is achieved here by a PN controller augmented with a closing velocity controller. In contrast, the landing phase is characterized by a relatively close proximity between the MAV and the GV and the availability of visual feedback to determine the target's position. This phase requires a higher level of accuracy and faster response time from the controller, and a PID controller can be more easily tuned to meet these requirements than a PN controller. In addition, the system should transition from one controller to the other seamlessly, avoiding discontinuity in the commands sent to the MAV.

Proportional Navigation Guidance The well-known PN guidance law uses the fact that two vehicles are on a collision course if their LOS remains at a constant angle in order to steer the MAV toward the GV. It works by keeping rotation of the velocity vector proportional to the rotation of the LOS vector. Our PN controller provides an acceleration command that is normal to the instantaneous LOS

$$a_\perp = -\lambda|\dot{u}|\frac{u}{|u|} \times \Omega, \text{ with } \Omega = \frac{u \times \dot{u}}{u \cdot u}, \qquad (6)$$

where λ is a gain parameter, $u = p_a^{\mathsf{N}} - p_m^{\mathsf{N}}$ and $\dot{u} = v_a^{\mathsf{N}} - v_m^{\mathsf{N}}$ are obtained from the Kalman filter and represent the LOS vector and its derivative expressed in the NED frame, and Ω is the rotation vector of the LOS. We then supplement the PN guidance law (6) with an approach velocity controller determining the acceleration $a_\parallel$ along the LOS direction, which in particular allows us to specify a high enough velocity required to properly overtake the target. This acceleration component is computed using the following PD structure

$$a_\parallel = K_{p_\parallel} u + K_{d_\parallel} \dot{u}$$

where $K_{p_\parallel}$ and $K_{i_\parallel}$ are constant gains. The total acceleration command is obtained by combining both components

$$a = a_\perp + a_\parallel.$$

As only the horizontal control is of interest, the acceleration along z-axis is disregarded. The desired acceleration then needs to be converted to attitude control inputs that are more compatible with the MAV input format. In frame $\{\mathsf{N}\}$, the quadrotor dynamic equations of translation read as follows

$$m a_m = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + \mathrm{R}_{\mathsf{B}}^{\mathsf{N}} \begin{bmatrix} 0 \\ 0 \\ -T \end{bmatrix} + F_D$$

where $\mathrm{R}_{\mathsf{B}}^{\mathsf{N}}$ denotes the rotation matrix from $\{\mathsf{N}\}$ to $\{\mathsf{B}\}$,

$$\mathrm{R}_{\mathsf{B}}^{\mathsf{N}} = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix},$$

T the total thrust created by rotors, F_D the drag force, m the MAV mass and g the standard gravity. The force equations simplify as

$$m \begin{bmatrix} \ddot{x}_m \\ \ddot{y}_m \\ \ddot{z}_m \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} - \begin{bmatrix} c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\phi s_\theta s_\psi - s_\phi c_\psi \\ c_\phi c_\theta \end{bmatrix} T + \begin{bmatrix} -k_d \dot{x}_m|\dot{x}_m| \\ -k_d \dot{y}_m|\dot{y}_m| \\ -k_d \dot{z}_m|\dot{z}_m| \end{bmatrix}$$

where the drag is roughly modeled as a force proportional to the signed quadratic velocity in each direction and k_d is constant, which we estimated by recording the terminal velocity for a range of attitude controls at level flight and performing a least squares regression on the data. For constant flight altitude, $T = mg/c_\phi c_\theta$ and assuming $\psi = 0$, it yields

$$m \begin{bmatrix} \ddot{x}_m \\ \ddot{y}_m \end{bmatrix} = mg \begin{bmatrix} -\tan\theta \\ \tan\phi/\cos\theta \end{bmatrix} - \begin{bmatrix} k_d \dot{x}_m|\dot{x}_m| \\ k_d \dot{y}_m|\dot{y}_m| \end{bmatrix}$$

The following relations can then be obtained:

$$\theta = -\arctan\left((m\ddot{x}_m + k_d\dot{x}_m|\dot{x}_m|)/mg\right)$$
$$\phi = \arctan\left(\cos\theta\left(m\ddot{y}_m + k_d\dot{y}_m|\dot{y}_m|\right)/mg\right),$$

where θ and ϕ are the desired pitch and roll angles for specific acceleration demands.

PID controller The landing phase is handled by a PID controller, with the desired acceleration computed as

$$a = K_p u + K_i \int u + K_d \dot{u},$$

where K_p, K_i and K_d are constant gains. The tuning for the PID controller was selected to provide aggressive dynamic path following, promoting a quick disturbance rejection. The controller was first tuned in simulation and then the settings were manually adjusted in flight.

Controller switching The controller switching scheme chosen is a simple fixed distance switching condition with a slight hysteresis. The switching distance selected was 6 m to allow for any perturbation due to the switching to dissipate before reaching the landing platform.

Vertical control The entire approach phase is done at a constant altitude, which is handled by the internal vertical position controller of the MAV. The descent is initiated once the quadrotor has stabilized over the landing platform. A constant vertical velocity command is then issued to the MAV and maintained until it reaches a height of 0.2 m above the landing platform at which point the motors are disarmed.

5. EXPERIMENTAL VALIDATION

5.1 System Description

We implemented our system on a commercial off-the-shelf DJI Matrice 100 (M100) quadcopter shown in Fig. 3. All computations are performed on the standard OBC for this platform (DJI Manifold), which contains an Nvidia Tegra K1 SoC. The 3-axis gimbaled camera is a Zenmuse X3, from which we receive 720p YUV color images at 30 Hz. To reduce computations, we drop the U and V channels and downsample the images to obtain 640 × 360 monochrome images. We modified the M100 to rigidly attach a downward facing Matrix Vision mvBlueFOX camera, equipped with an ultra-wide angle Sunex DSL224D lens with a diagonal field of view of 176 degrees. The M100 is also equipped with the DJI Guidance module, an array of up to five stereo cameras, which are meant to help developers create mapping and obstacle avoidance solutions for robotics applications. This module seamlessly integrates with the INS to provide us with position, velocity and acceleration measurements for the M100 using a fusion of on-board sensors described in Zhou et al. (2015). This information is used in our Kalman filter in equation (5).

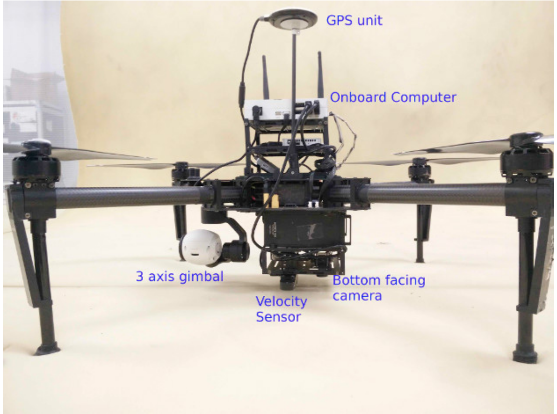

Fig. 3. The M100 quadcopter. Note that all side facing cameras of the Guidance module were removed and the down facing BlueFox camera sits behind the bottom guidance sensor.

Our algorithms were implemented in C++ using ROS (Quigley et al., 2009). Using an open source implementation of the AprilTag library based on OpenCV and with accelerations provided by OpenCV4Tegra, we can run the tag detection at a full 30 fps using the X3 camera and 20 fps on the BlueFOX camera.

As mentioned in Section 4, we implemented our control system using pure attitude control in the xy axes and velocity control in the z axis. The reason for this is that the internal velocity estimator of the M100 fuses optical flow measurements from the Guidance system. These measurements become extremely inaccurate once the quadcopter flies over the landing platform. Although optical flow could be used to measure the relative velocity of the car, it would be difficult to pinpoint the moment where flow measurements transition from being with respect to the ground to being with respect to the moving car.

5.2 Experimental Results

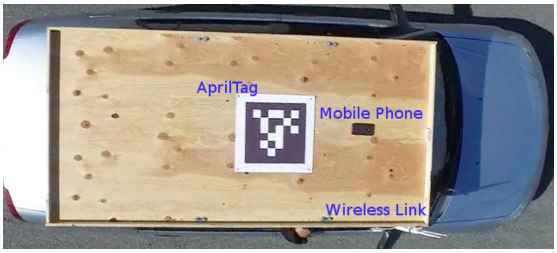

Fig. 4. Experimental setup showing the required equipment on the car. In practice the mobile phone could also be held inside the car as long as GPS reception is still good.

A first experiment was done using only data from the mobile phone (no visual feedback) to prove the validity of our Proportional Navigation controller for the long range approach. Figure 5 shows the output of our EKF estimating the MAV's and the target's positions. With the target and the MAV starting at about 30 meters from each other, we see the MAV fly a smooth rendezvous trajectory eventually ending on top of the target.

The close range system was experimentally validated with the GV moving at speeds as low as a human jogging and on a private race track at 30, 40 and 50 km/h with successful landings in each case. A further experiment was attempted at 55 km/h; this resulted in a missed landing as it was too close to the maximum speed of the MAV. However we can affirm that we can land easily at 40 km/h or less, and acceptably at 50 km/h. Videos of our experiments can be found at `https://youtu.be/ILQqD2xQ4tg`.

Figure 6 shows the landing sequence where the quadrotor takes off, tracks and lands on the landing pad. The curves gain in altitude as the trajectory progresses because of the elevation profile of the race track. The effect is seen more clearly in Fig. 7, where we can also see the filtered AprilTag altitude rise, thanks to visual data and the M100's internal altitude estimator, even before the phone's GPS data indicates a change in altitude.

Furthermore, we can see in Fig. 7 how the M100 closely matches the velocity of the AprilTag to perform the landing maneuver. The two peaks at the 24 and 27 second marks are strongly correlated to the visual loss of the tag by the BlueFOX camera which we can observe in Fig. 8. The descent starts at the 24 second mark slightly before the car hits its designated velocity of 14 m/s or 50.4 km/h. We can also see in Fig. 7 how the M100's velocity estimation from the INS becomes incorrect when it is on top of the car. Which explains why we decided to dynamically increase the standard deviation of the measurement as described in Section 3.2. Figure 9 shows the quadcopter's attitude during the flight. Notice how the roll is stable close to 0° while the pitch stays between −10° and −25° for consistent forward flight. Finally, the yaw changes drastically after 10 seconds at the moment the car starts moving.

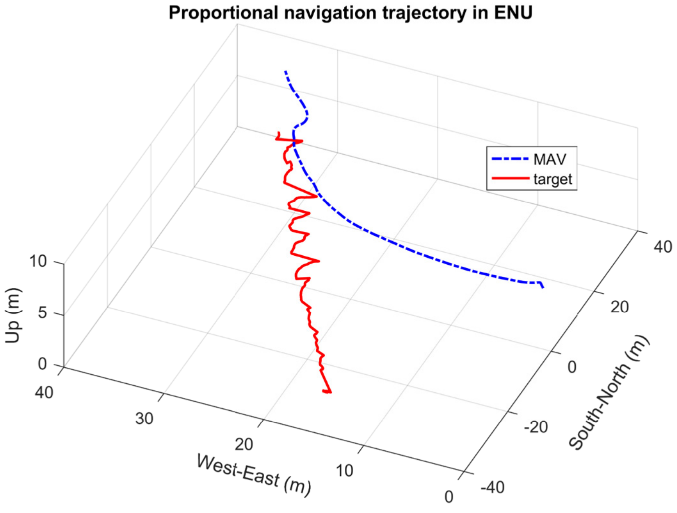

Fig. 5. The PN controller efficiently catches up with the target at long range even when the only source of information is the mobile phone's GPS and IMU.

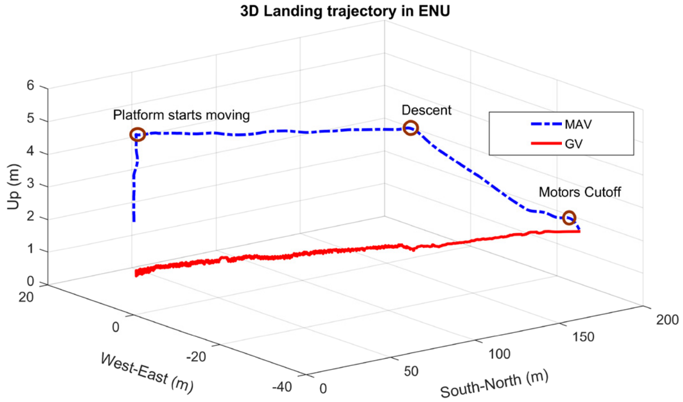

Fig. 6. Landing trajectory at 50km/h using the PID controller.

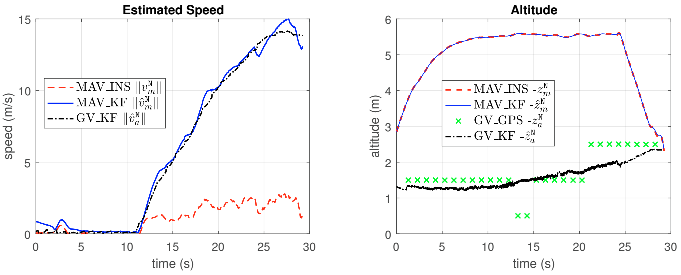

Fig. 7. Motions of M100 and AprilTag.

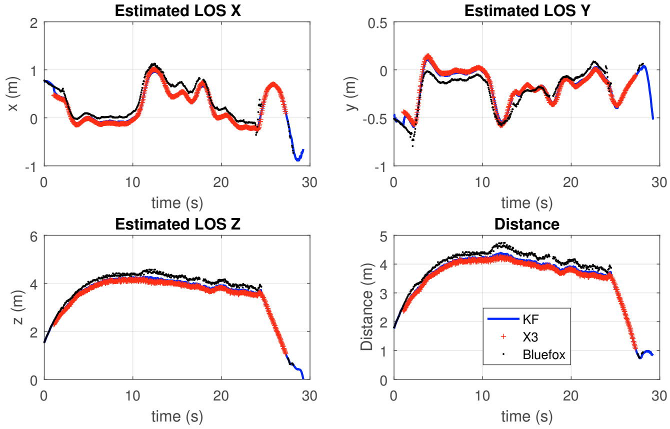

Fig. 8. LOS and distance (coordinates in NED frame).

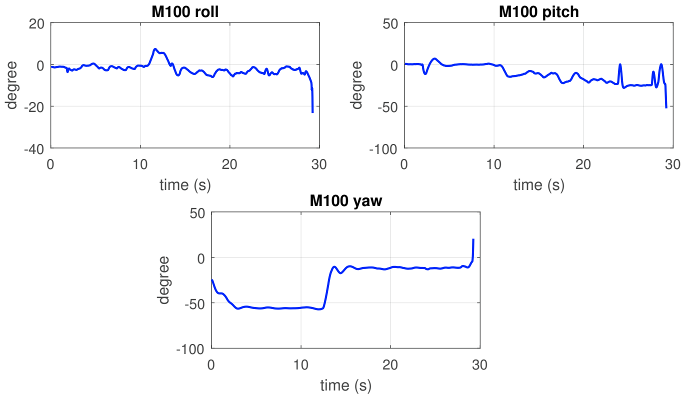

Fig. 9. M100 attitude.

6. CONCLUSION

The problem of the automatic landing of a MAV on a moving vehicle was solved with experimental tests going up to 50 km/h. A Proportional Navigation controller was used for the long range approach, which subsequently transitioned into a PID controller at close range. A Kalman filter was used to estimate the position of the MAV relative to the landing pad by fusing together measurements from the MAV's onboard INS, a visual fiducial marker and a mobile phone. Furthermore, it was shown that this system can be implemented using only commercial off-the-shelf components.

Future work may include using a better multiscale visual fiducial on the landing pad to allow visual target tracking at longer and closer ranges using a single camera or simplifying the system by removing the requirement for a mobile phone. Performance improvements could also be achieved by adding a model of the ground vehicle's turbulence or adding wind estimation in the control system.

REFERENCES

Garone, E., Determe, J.F., and Naldi, R. (2014). Generalized traveling salesman problem for carrier-vehicle systems. *AIAA Journal of Guidance, Control and Dynamics*, 37(3), 766–774.

Garrido-Jurado, S., noz Salinas, R.M., Madrid-Cuevas, F., and Marín-Jiménez, M. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6), 2280 – 2292.

Gautam, A., Sujit, P.B., and Saripalli, S. (2015). Application of guidance laws to quadrotor landing. In *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, 372–379.

Groves, P.D. (2013). *Principles of GNSS, inertial, and multisensor integrated navigation systems*. Artech House, 2nd edition.

Holt, R. and Beard, R. (2010). Vision-based road-following using proportional navigation. *Journal of Intelligent and Robotic Systems*, 57(1-4), 193 – 216.

Kabamba, P.T. and Girard, A.R. (2014). *Fundamentals of Aerospace Navigation and Guidance*. Cambridge University Press.

Kim, J., Jung, Y., Lee, D., and Shim, D.H. (2014). Outdoor autonomous landing on a moving platform for quadrotors using an omnidirectional camera. In *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, 1243–1252.

Kolodny, L. (2016). Mercedes-benz and Matternet unveil vans that launch delivery drones. http://tcrn.ch/2c48his. (Accessed on 09/09/2016).

Lange, S., Sunderhauf, N., and Protzel, P. (2009). A vision based onboard approach for landing and position control of an autonomous multirotor uav in gps-denied environments. In *Advanced Robotics, 2009. ICAR 2009. International Conference on*, 1–6.

Lardinois, F. (2016). Ford and DJI launch $100,000 developer challenge to improve drone-to-vehicle communications. tcrn.ch/1O7uOFF. (Accessed on 09/28/2016).

Lin, C.E. and Yang, S.K. (2014). Camera gimbal tracking from uav flight control. In *Automatic Control Conference (CACS), 2014 CACS International*, 319–322.

Ling, K. (2014). *Precision Landing of a Quadrotor UAV on a Moving Target Using Low-cost Sensors*. Master's thesis, University of Waterloo.

Mathew, N., Smith, S.L., and Waslander, S.L. (2015). Planning paths for package delivery in heterogeneous multirobot teams. *IEEE Transactions on Automation Science and Engineering*, 12(4), 1298–1308.

Muskardin, T., Balmer, G., Wlach, S., Kondak, K., Laiacker, M., and Ollero, A. (2016). Landing of a fixed-wing uav on a mobile ground vehicle. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 1237–1242.

Olson, E. (2011). Apriltag: A robust and flexible visual fiducial system. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 3400–3407.

Quigley, M., Conley, K., Gerkey, B.P., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A.Y. (2009). ROS: an open-source robot operating system. In *ICRA Workshop on Open Source Software*.

Tan, R. and Kumar, M. (2014). Tracking of ground mobile targets by quadrotor unmanned aerial vehicles. *Unmanned Systems*, 2(02), 157–173.

Yang, S., Ying, J., Lu, Y., and Li, Z. (2015). Precise quadrotor autonomous landing with SRUKF vision perception. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2196–2201.

Zhou, G., Fang, L., Tang, K., Zhang, H., Wang, K., and Yang, K. (2015). Guidance: A visual sensing platform for robotic applications. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 9–14.