A control architecture for fast and precise autonomous landing of a VTOL UAV onto an oscillating platform

Botao Hu
Graduate Research Assistant

Lu Lu
Postdoctoral Research Associate

Sandipan Mishra
Assistant Professor

Department of Mechanical, Aerospace and Nuclear Engineering
Rensselaer Polytechnic Institute
Troy, New York

ABSTRACT

In this paper, the problem of landing an unmanned aerial vehicle (UAV) on a platform with unknown oscillating motion is investigated. A control architecture that enables fast, safe and precise landing process is proposed. This control architecture consists of three modules: a tracking control module, a trajectory generation module and a vision-based motion estimation module. For the tracking control module, an Adaptive Robust Controller (ARC) is used to robustly adapt to the changes in thrust due to ground effect. In the trajectory generation module, a time optimal reference trajectory is generated to follow the platform motion. In the motion estimation module, the motion of the moving platform and the UAV is estimated based on only relative distance measurement and acceleration measurement. Comparative simulation and experimental results are presented to validate the effectiveness of the proposed control architecture.

INTRODUCTION

Vertical take-Off and landing (VTOL) unmanned aerial vehicles (UAVs) are beginning to play an important role in a variety of applications, including aerial imaging, surveillance, package delivery and law enforcement. Such VTOL UAVs should be able to perform an autonomous landing maneuver onto a platform, which may be either stationary or moving. The ability to land on a moving platform is particularly critical to many maritime applications, where landing onto a moving shipboard is necessary especially in high sea states. Further, it is desirable that the shipboard landing be fast (i.e., time optimal), safe (i.e., adhering to hard physical constraints to avoid impact and collision) and precise (i.e., smooth and accurate landing maneuvers). However, shipboard landing is typically challenging because of the nonlinear dynamics of the UAV, uncertainties in the system and the time varying nature of the shipboard motion.

A variety of feedback controllers have been developed to address the VTOL UAV shipboard landing problem. Herisse *et al*. (Ref. 1) developed a feedback control algorithm based on optical flow inspired by the approach used by honeybees to land on flowers. Oh *et al*. (Ref. 2) developed a controller that used a tether to guide the shipboard landing operation of an autonomous helicopter. Lee *et al*. (Ref. 3) and Ling *et al*. (Ref. 4) used vision as feedback and developed a controller that enabled a quadrotor landing on a moving platform. Although these approaches achieve closed-loop stability during the landing process, there are typically no guarantees for fast landing under physical constraints such as the input saturation, position and velocity limitations.

Instead of designing a purely feedback-driven controller with no knowledge of the platform motion, an alternate approach towards autonomous landing focuses on identification of the pattern of platform motion and feedforward compensation based on this motion profile to achieve good landing performance. When the landing maneuver is performed on a moving deck on open ocean, the vertical movement of the wave (which is approximated as the same as the platform motion itself) can be modeled as a superposition of several sine functions with different amplitudes, frequencies and phases (Refs. 5, 6). If the wave motion can be estimated, the shipboard landing problem can then be reduced to a UAV motion control problem in which a UAV is controlled to follow the wave motion. The motion control problem can be further divided into two subproblems: the trajectory generation problem that generates a reference trajectory and the trajectory tracking problem that regulates the UAV to track the trajectory. The following is a brief review on the mentioned three subproblems.

For wave motion estimation, methods such as recursive least square estimation (Ref. 7), Kalman filter (Ref. 8), Extended Kalman Filter (Refs. 9, 10), Unscented Kalman Filter (Ref. 5) and adaptive identifier (Ref. 11) can be used. For the VTOL UAV trajectory generation problem, optimal con-

Presented at the AHS 71st Annual Forum, Virginia Beach, Virginia, May 5–7, 2015.

trol based trajectory planning algorithms have been developed such as time optimal trajectory planning (Refs. 12, 13), minimal jerk trajectory planning (Ref. 14), minimal snap trajectory planning (Ref. 15) and minimal thrust trajectory planning (Ref. 16). In addition to optimal trajectory planning methods, polynomial trajectories such as cubic spline trajectory (Ref. 12) and quartic polynomial trajectory (Ref. 17) have also been used to guide the autonomous landing of VTOL UAVs. For the trajectory tracking problem, linear control laws such as PID (Refs. 18, 19) and LQR (Ref. 20) were first developed. Subsequently, nonlinear control design techniques such as backstepping (Refs. 21,22), feedback linearization (Ref. 23), adaptive control (Ref. 24) and iterative learning control (Refs. 25, 26) have been applied to perform accurate tracking of a given trajectory.

In our previous work (Ref. 27), we proposed a control architecture that combines the advantage of feedback control and time optimal control together to achieve fast, safe and precise landing of a VTOL UAV on a pure sinusoidally oscillating platform. The control architecture includes three modules: an motion estimation module which estimates the wave motion based on only relative distance and accelerometer measurement; a trajectory generation module that generates a time optimal reference trajectory to follow the estimated wave motion; a tracking control module which controls the UAV to track the reference trajectory. In this paper, we extend our previous work in three respects. First, we introduce a vision based approach for relative distance measurement. Second, we extend the motion estimation to multiple sine functions. Finally, we conduct a series of experiments and simulations to validate the effectiveness of the control architecture.

PROBLEM FORMULATION AND CONTROL OBJECTIVE

In this paper, we focus on controlling a VTOL UAV for vertical landing on an oscillating platform as shown in Figure 1. We restrict ourselves first to movement only in the z-direction. A quadrotor is used as the testbed for control algorithm development, implementation, and validation. A detailed analysis of quadrotor dynamics can be found in (Ref. 28). Here, a brief description for quadrotor dynamics is introduced first, followed by a description for the platform motion. The control objective is then presented formally at the end of this section.

For the description of the dynamics of the quadrotor, let $\{\mathbb{E}_I, O_I\}$ denote the inertial frame and $\{\mathbb{E}_B, O_B\}$ denote the body frame. $R_{OB}^{3\times3} \in SO(3)$ denotes the relative rotation matrix of the body frame with respect to the inertia frame. $[x\ y\ z]^T, [\dot{x}\ \dot{y}\ \dot{z}]^T, [\ddot{x}\ \ddot{y}\ \ddot{z}]^T$ denote the absolute position, velocity and acceleration vectors of the quadrotor in the inertia frame. For the translational motion, let m denote the mass of the quadrotor, and u denote the total thrust command sent into the quadorotor. The dynamic equation can be written as equation (1).

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = R_{OB} \begin{bmatrix} 0 \\ 0 \\ S(u) \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}, \qquad (1)$$

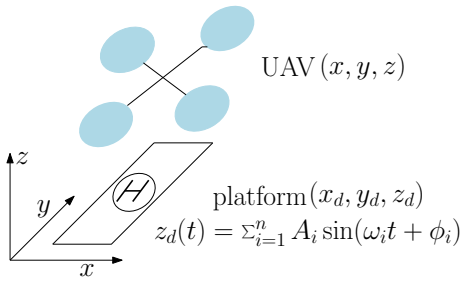

Fig. 1. This figure shows the UAV shipboard landing problem. The (x_d, y_d, z_d) denote the platform position in inertial frame and the motion in z direction (heave motion) is modeled as a combination of several sinusoidal functions.

$S(u)$ is thrust that the quardrotor generates and it is a function of thrust command input. When the quadrotor is approaching to the ground, the rotor will generate more thrust for a given power. This phenomenon is caused by the ground effect (Ref. 29). The ground effect is function of the height of the quadrotor above the ground and can be modeled as equation (2) where R is a constant (Ref. 29).

$$S(u) = k_G u = \frac{1}{1 - (\frac{R}{4(z-z_d)})^2} u \qquad (2)$$

For the platform motion, according to (Ref. 5), the wave motion can be modeled as a superposition of several sinusoidal functions. Thus, the platform height z_d and velocity in vertical directions $\dot{z}_d$ are modeled as equation (3).

$$\begin{aligned} z_d(t) &= \textstyle\sum_{i=1}^{n} A_i \sin(\omega_i t + \phi_i) \\ \dot{z}_d(t) &= \textstyle\sum_{i=1}^{n} A_i \omega_i \cos(\omega_i t + \phi_i) \end{aligned} \qquad (3)$$

where A_i, ω_i and ϕ_i represents the amplitude, frequency and phase of one sinusoidal component in the motion.

Finally, we present the control objective. The problem of a quadrotor landing on an unknown motion can be divided into a motion estimation problem and a precision motion control problem. For the motion control part, the three requirements namely, speed, safety and preciseness of the landing, can be formulated analogous to requirements for motion control design. 'Precise' means that states of the quadrotor $(z(t), \dot{z}(t))$ converges to those of the platform $(z_d(t), \dot{z}_d(t))$ with sufficient accuracy for some t. In other words, the final value of $(z(t), \dot{z}(t))$ should be within a small neighborhood of $(z_d(t), \dot{z}_d(t))$. 'Fast' means that the time, t_f needed for $(z(t), \dot{z}(t))$ to reach the small neighborhood of $(z_d(t), \dot{z}_d(t))$ is as short as possible. 'Safe' means that the physical safety requirements such as the position constraint (the quadrotor does not hit the platform), velocity limitation (such as maximum descending velocity) and and the acceleration limitation are satisfied. For the motion estimation part, the goal is to use only the relative distance $z(t) - z_d(t)$ and inertia measurement of the UAV $\ddot{z}(t)$ to get all the information needed in the motion control problem. It is because the real time height measurement and velocity measurement is not always available in outdoor environment, especially in GPS denied situations.

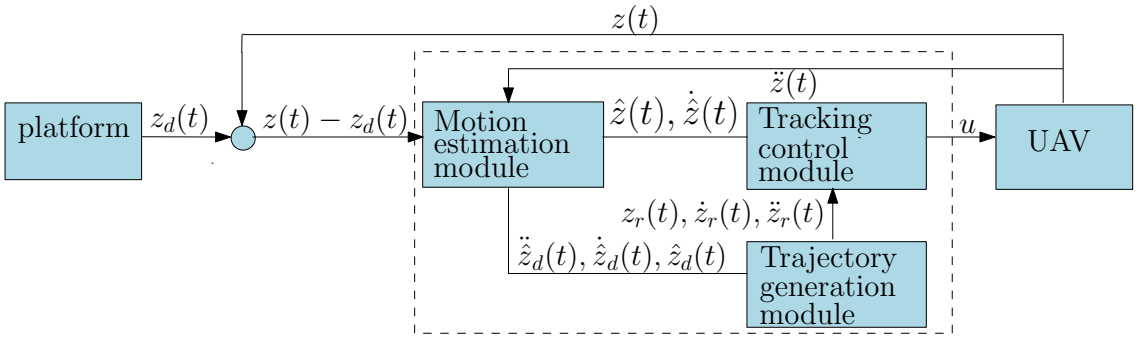

Fig. 2. The control architecture consists of three modules in the dashed box.The motion estimation module generates the estimation of the platform motion. The trajectory generation module outputs the reference trajectory and the tracking control module generates the thrust command into the quadrotor

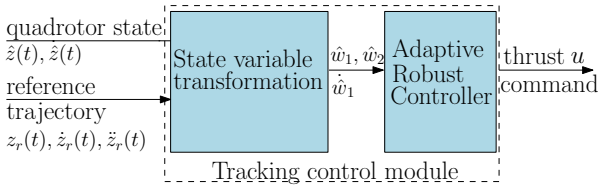

Fig. 3. This figure explains the structure of tracking control module. There are two parts in the module. A state transformation part that changes the state variables and an Adaptive Robust Controller that generates the thrust command. The input for this module includes the quadrotor state and reference trajectory. The output for this module is the thrust command

In brief, the control objective is to design a control input u such that based on $z(t) - z_d(t)$and $\ddot{z}(t)$ as measurements, $z \to z_d$ and $\dot{z} \to \dot{z}_d$ as fast and accurately as possible, while the motion constraints (such as the position constraint and velocity/acceleration limitations) are satisfied.

TRACKING CONTROL MODULE DESIGN

The objective of the tracking control module is to control the quadrotor track a given reference trajectory which composed of position, velocity and acceleration $[z_r(t)\ \dot{z}_r(t)\ \ddot{z}_r(t)]$. The reference trajectory is generated by the trajectory generation module, which will be introduced in the following section. Figure.3 illustrates the two parts structure of this module. This section will first analyze the influence of the ground effect on the quadrotor dynamics and then introduce the state transformation. Then the design of the Adaptive Robust Controller will be introduced.

Analysis of the ground effect influence

When designing a controller, it is crucial to take the ground effect into consideration since the ground effect term in equation (2) will significantly change the system dynamics. Figure.4 shows an offline identification of the ground effect. From the figure, the control effort needed for hover decreases as the hovering height decreases, thus introduces nonlinearity on the system dynamics.

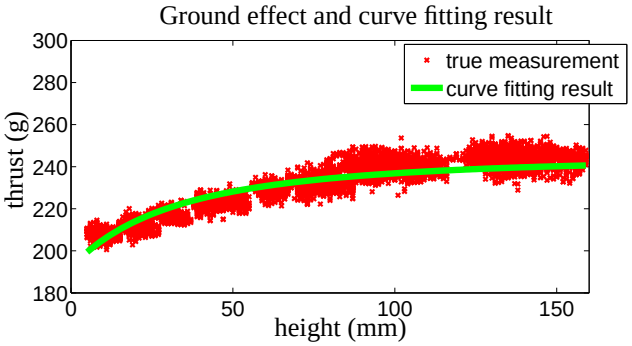

Fig. 4. This figure shows the control effort when hovering at different height. The red dots are the data points and the green line is a least square fit. From the figure, it is obvious the less control effort is needed when hovering near the ground than hovering far away from the ground

Transformation of the state variables

In our previous paper, we compensated the ground effect online by applying an ARC (adaptive robust controller). This paper will give a brief introduction to the controller design. The detail design can be found in (Ref. 27).

Since only vertical landing is considered in this paper, the dynamics equation (1) can be simplified as equation (4) where the ground effect term k_G is divided on both sides.

$$\begin{aligned} \dot{z} &= v \\ \theta\dot{v} &= u - \theta g \\ \theta &= \frac{m}{k_G} \end{aligned} \tag{4}$$

θ captures the actual ground effect. The input for the controller includes the height and velocity estimations of the quadrotor $[\hat{z}(t)\ \dot{\hat{z}}(t)]^T$ and reference trajectory $[z_r(t)\ \dot{z}_r(t)\ \ddot{z}_r(t)]^T$. The estimations come from the motion estimation module and the reference trajectory comes from the trajectory generation module and will be introduced later.

Let $\hat{w}_1$,$\hat{w}_2$,$\hat{\varphi}$ be defined as new state variables and perform a transformation from the input state variables as equation (5). The controller design will be based on these new state vari-

ables.

$$\begin{aligned} \hat{w}_1 &= \hat{z} - z_r \\ \dot{\hat{w}}_1 &= \dot{\hat{z}} - \dot{z}_r \\ \hat{w}_2 &= \dot{\hat{w}}_1 + k_1 \hat{w}_1 \\ \hat{\varphi} &= -(g + \ddot{z}_r - k_1 \dot{\hat{w}}_1) \end{aligned} \tag{5}$$

Design of the Adaptive Robust Controller

Based on the modified states $[w_1, \dot{w}_1, w_2]$, the control effort u is generated as equation (6).

$$\begin{aligned} u &= u_s + u_a \\ u_a &= -\hat{\theta}\hat{\varphi} \\ u_s &= -k_2 \hat{w}_2 + u_{s2} \end{aligned} \tag{6}$$

u_{s2} is a robust performance feedback term and satisfies the equation (7).

$$\begin{aligned} \hat{w}_2\{u_{s2} - \hat{\varphi}(\hat{\theta} - \theta)\} \leq \varepsilon \\ \hat{w}_2 u_{s2} \leq 0 \end{aligned} \tag{7}$$

where ε is a design variable that can be arbitrarily small. k_1 and k_2 are positive parameters to be tuned. $\hat{\theta}$ is an estimation of the θ term and it is updated as equation (8).

$$\dot{\hat{\theta}} = \mathrm{Proj}_{\hat{\theta}}(\gamma \hat{\varphi} \hat{w}_2) = \begin{cases} 0, \text{ if } \hat{\theta} = m \text{ and } \gamma \hat{\varphi} \hat{w}_2 > 0 \\ 0, \text{ if } \hat{\theta} = \frac{m}{k_{Gmax}} \text{ and } \gamma \hat{\varphi} \hat{w}_2 < 0 \\ \gamma \hat{\varphi} \hat{w}_2, \text{ else} \end{cases}, \tag{8}$$

γ is a positive gain. This controller guaranteed the tracking error to be always bounded in a small region. If the estimation is accurate, this controller enables zero tracking error of the reference trajectory under ground effect. The proof can be seen in (Ref. 30).

TRAJECTORY GENERATION MODULE DESIGN

The objective of this module is to generate a time optimal reference trajectory $z_r(t)$ to follow the platform motion $z_d(t)$ and guarantee the motion constraints are met. This module is triggered only after the landing process begins. Figure. 5 shows the structure of this module. There are two parts in the module: a time optimal problem solver and an integrator. This section will briefly introduce these two parts and the detail design of this module can be found in (Ref. 27).

Time optimal problem solver

Let $z_a = z_r - \hat{z}_d, \dot{z}_a = \dot{z}_r - \dot{\hat{z}}_d$ denote the relative distance and velocity. The relative acceleration $\ddot{z}_a$ can be calculated by solving the time optimal problem of (9).

$$\begin{aligned} &\min_{\dddot{z}'_a(\tau), \tau \in (t, t_f)} (t_f) \\ &\text{subject to } z'_a(\tau) = z_a(t), \dot{z}'_a(\tau) = \dot{z}_a(t) \\ &z'_a(t_f) = 0, \dot{z}'_a(t_f) = 0, \\ &z'_a(\tau) \geq 0, \forall \tau \in [t, t_f], \\ &-\dot{z}_{rmax} - \dot{\hat{z}}_d(\tau) \leq \dot{z}'_a(\tau) \leq \dot{z}_{rmax} - \dot{\hat{z}}_d(\tau), \forall \tau \in [t, t_f], \\ &\ddot{z}_{rmin} - \ddot{\hat{z}}_d(\tau) \leq \ddot{z}'_a(\tau) \leq \ddot{z}_{rmax} - \ddot{\hat{z}}_d(\tau), \forall \tau \in [t, t_f], \end{aligned} \tag{9}$$

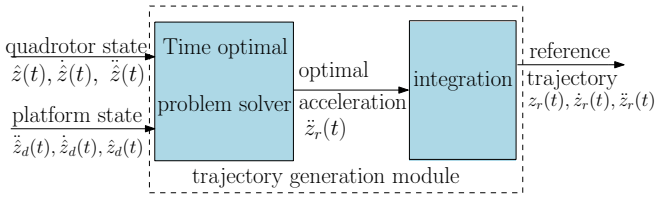

Fig. 5. This figure shows the structure of the trajectory generation module. This module takes in the quadrotor states and platform states and outputs a reference trajectory. This module first solves a time optimal problem and generates the optimal reference acceration. Then the reference velocity and position is updated by integration.

When $t = t_0$, the initial values $z_a(t_0)$ and $\dot{z}_a(t_0)$ are set as $\hat{z}(t_0) - \hat{z}_d(t_0)$ and $\dot{\hat{z}}(t_0) - \dot{\hat{z}}_d(t_0)$. The final value of $z_a(t_f)$ and $\dot{z}_a(t_f)$ are both zero, means zero relative velocity when the quadrotor lands on the platform. After solving the problem, the optimal solution is $\dddot{z}_a$.

According to the Pontryagin's theorem, the optimal solution $\dddot{z}_a(t)$ is a bang-bang type control law (Ref. 31).

$$\dddot{z}_a(t) = \begin{cases} \dddot{z}_{rmin} - \dddot{\hat{z}}_d(t) & \forall (z_a(t), \dot{z}_a(t)) \in \Omega_1(t) \\ -\dddot{\hat{z}}_d(t) & \forall (z_a(t), \dot{z}_a(t)) \in \Omega_2(t) \\ \dddot{z}_{rmax} - \dddot{\hat{z}}_d(t) & \forall (z_a(t), \dot{z}_a(t)) \in \Omega_3(t) \end{cases}, \tag{10}$$

where $\Omega_1(t)$ is the region of maximum downward acceleration, $\Omega_2(t)$ is the region of zero acceleration to maintain maximum downward velocity, $\Omega_3(t)$ is the region of maximum downward deceleration in order to make a stop. The procedure to determine which region the current state $(z_a(t), \dot{z}_a(t))$ belongs to can be found in (Refs. 27, 32).

Trajectory generation

Given the acceleration of the reference trajectory $\ddot{z}_a(t)$, the reference trajectory $[z_r(t)\ \dot{z}_r(t)\ \ddot{z}_r(t)]^T$ can be generated by integration as equation (11).

$$\begin{aligned} \ddot{z}_r(t) &= \ddot{z}_a(t) + \ddot{\hat{z}}_d(t) \\ \dot{z}_r(t) &= \dot{z}_r(t_0) + \int_{t_0}^{t} \ddot{z}_r(\tau) d\tau \\ z_r(t) &= z_r(t_0) + \int_{t_0}^{t} z_r(\tau) d\tau \end{aligned} \tag{11}$$

MOTION ESTIMATION MODULE DESIGN

The previous two modules have shown that given a sufficiently accurate estimation of the absolute height and velocity of the quadrotor as well as the platform motion, a fast and accurate landing maneuver can be achieved. In a typical outdoor VTOL UAV landing operation, the absolute motion of the UAV $[z(t),\ \dot{z}(t),\ \ddot{z}(t)]^T$ and the oscillating platform motion $[z_d(t),\ \dot{z}_d(t),\ \ddot{z}_d(t)]^T$ are usually not directly measurable in real time, especially in GPS denied environment. However, the relative distance between the UAV and the platform can be easily measured by computer vision-based approaches such as those in (Refs. 5,9). Furthermore, the acceleration of the UAV can also be measured by an IMU (inertial measurement unit).

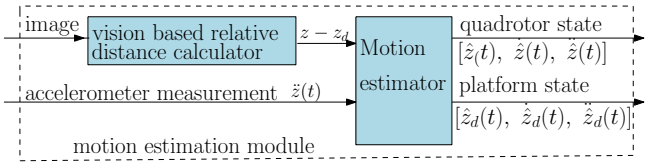

Fig. 6. This figure shows the structure of the motion estimation module. There are two parts in this module, the first one is a relative distance calculator that adopts a vision based approach to get the relative distance from a single image. The second part is the unscented Kalman Filter based motion estimator that estimates the quadrotor and platform motion with relative distance and accelerometer measurement.

Therefore, in this module, the relative distance $z - z_d$ and the acceleration of the quadrotor $\ddot{z}$ are used to obtain the estimations of $[\hat{z},\ \dot{\hat{z}},\ \hat{z}_d,\ \dot{\hat{z}}_d,\ \ddot{\hat{z}}_d]^T$ based on the Unscented Kalman Filter similar to that proposed in (Ref. 10). Figure .6 shows the structure of this module. This section will first introduce the construction of motion estimator and then introduce the relative distance calculator.

Motion estimator design

In this subsection, the system dynamics will be introduced first and observability analysis will be explained to show the local observability. Then the implementation will be described.

System dynamics For the motion estimation module design, the following assumptions are made:

- The platform motion is modeled as a combination of 3 sinusoidal motions, with different nonzero frequencies. In (Ref. 33), three components were shown to be able to approximate the actual wave motion. So the platform motion is modeled as (12)

$$z_d = \sum_{i=1}^{3} A_i \sin(\omega_i t + \phi_i) \tag{12}$$

- The amplitude A_i, frequency ω_i and phase ϕ_i in equation (3) are sufficiently slowly time varying (or time invariant).

Define the state vector to be estimated as $x = [x_1, x_2, ..x_{11}]^T$, each state is defined as following equations.

$$\text{Quadrotor state:} \begin{cases} x_1 : \text{height} \\ x_2 : \text{velocity} \end{cases} \tag{13}$$

$$\text{Wave height:} \begin{cases} x_3 & : & A_1 \sin(\omega_1 t + \phi_1) \\ x_6 & : & A_2 \sin(\omega_2 t + \phi_2) \\ x_9 & : & A_3 \sin(\omega_3 t + \phi_3) \end{cases} \tag{14}$$

$$\begin{cases} x_4 & : & A_1 \cos(\omega_1 t + \phi_1) \\ x_7 & : & A_2 \cos(\omega_2 t + \phi_2) \\ x_{10} & : & A_3 \cos(\omega_3 t + \phi_3) \end{cases} \tag{15}$$

$$\text{Frequencies:} \begin{cases} x_5 & : & \omega_1 \\ x_8 & : & \omega_2, \\ x_{11} & : & \omega_3 \end{cases} \tag{16}$$

$$\text{System measurement } y = x_1 - x_3 - x_6 - x_9 \tag{17}$$

v is the quadrotor acceleration measurement from the IMU. The system dynamics $\dot{x} = f(x, v)$ and output equation $y = h(x)$ can be written as (18).

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= v \\ \dot{x}_3 &= x_4 x_5 \\ \dot{x}_4 &= -x_3 x_5 \\ \dot{x}_5 &= 0 \\ \dot{x}_6 &= x_7 x_8 \\ \dot{x}_7 &= -x_6 x_8 \\ \dot{x}_8 &= 0 \\ \dot{x}_9 &= x_{10} x_{11} \\ \dot{x}_{10} &= -x_9 x_{11} \\ \dot{x}_{11} &= 0 \\ y &= x_1 - x_3 - x_6 - x_9. \end{aligned} \tag{18}$$

Observability analysis In order to estimate the states, the system should be at least locally observable. The Lie derivative of the system is denoted by G as (19).

$$G = \begin{bmatrix} L_f^0(h) \\ L_f^1(h) \\ L_f^2(h) \\ L_f^3(h) \\ L_f^4(h) \\ L_f^5(h) \\ L_f^6(h) \\ L_f^7(h) \\ L_f^8(h) \\ L_f^9(h) \\ L_f^{10}(h) \end{bmatrix}_{11\times 1} = \begin{bmatrix} x_1 - x_3 - x_6 - x_9 \\ x_2 - x_4 x_5 - x_7 x_8 - x_1 0 x_1 1 \\ v + x_3 x_5^2 + x_6 x_8^2 + x_9 x_{11}^2 \\ x_4 x_5^3 + x_7 x_8^3 + x_{10} x_{11}^3 \\ -x_3 x_5^4 - x_6 x_8^4 - x_9 x_{11}^4 \\ -x_4 x_5^5 - x_7 x_8^5 - x_{10} x_{11}^5 \\ x_3 x_5^6 + x_6 x_8^6 + x_9 x_{11}^6 \\ x_4 x_5^7 + x_7 x_8^7 + x_{10} x_{11}^7 \\ -x_3 x_5^8 - x_6 x_8^8 - x_9 x_{11}^8 \\ -x_4 x_5^9 - x_7 x_8^9 - x_{10} x_{11}^9 \\ x_3 x_5^{10} + x_6 x_8^{10} + x_9 x_{11}^{10} \end{bmatrix} \tag{19}$$

The gradient of the Lie derivative O, also known as the observability matrix, can be calculated by taking partial derivative of the Lie matrix G as (20).

$$O = \begin{bmatrix} \frac{\partial L_f^0(h)}{\partial x_1}\dot{x}_1 & .. & \frac{\partial L_f^0(h)}{\partial x_j}\dot{x}_j & .. & \frac{\partial L_f^0(h)}{\partial x_{11}}\dot{x}_{11} \\ \vdots & .. & \frac{\partial L_f^i(h)}{\partial x_j}\dot{x}_j & .. & \vdots \\ \frac{\partial L_f^{10}(h)}{\partial x_1}\dot{x}_1 & .. & \frac{\partial L_f^{10}(h)}{\partial x_j}\dot{x}_j & .. & \frac{\partial L_f^{10}(h)}{\partial x_{11}}\dot{x}_{11} \end{bmatrix}_{11\times 11} \tag{20}$$

Given the assumption that the frequencies are nonzero, through a nonlinear observability analysis by checking the rank of the derivative of the Lie matrix O, this system is locally observable. Thus given a close-enough guess for the initial state, all the states can be estimated based on a standard Unscented Kalman Filter algorithm.

Digital Implementation of UKF The equation (18) is defined in continuous time form. In order to implement the unscented Kalman Filter to estimate the states on the microcontroller, a discrete representation is used as in equation (21), where T_s is the sampling time.

$$\begin{aligned}
x_1(k+1) &= x_1(k)+0.5x_2(k)Ts+0.25vTs^2 \\
x_2(k+1) &= x_2(k)+0.5vTs^2 \\
x_3(k+1) &= x_3(k)\cos(x_5(k)Ts)+x_4(k)\sin(x_5(k)Ts) \\
x_4(k+1) &= -\sin(x_5(k)Ts)x_4(k)+\cos(x_5(k)Ts)x_4(k) \\
x_5(k+1) &= x_5(k) \\
x_6(k+1) &= x_6(k)\cos(x_8(k)Ts)+x_7(k)\sin(x_8(k)Ts) \\
x_7(k+1) &= -\sin(x_8(k)Ts)x_6(k)+\cos(x_8(k)Ts)x_7(k) \\
x_8(k+1) &= x_8(k) \\
x_9(k+1) &= x_9(k)\cos(x_{11}(k)Ts)+x_{10}(k)\sin(x_{11}(k)Ts) \\
x_{10}(k+1) &= -\sin(x_{11}(k)Ts)x_9(k)+\cos(x_{11}(k)Ts)x_{10}(k) \\
x_{11}(k+1) &= x_{11}(k)
\end{aligned} \tag{21}$$

After obtaining the state estimations $\hat{x}(k) = [\hat{x}_1(k) \; .. \; \hat{x}_{11}(k)]^t$, the absolute position of the quadrotor $\hat{z}(k)$, the velocity of the quadrotor $\hat{\dot{z}}_d(k)$ and the platform motion $\hat{z}_d(k)$, velocity $\hat{\dot{z}}_d(k)$ and the acceleration $\hat{\ddot{z}}_d(k)$ can be estimated as equation (22).

$$\begin{aligned}
\hat{z}(k) &= \hat{x}_1(k) \\
\hat{\dot{z}}(k) &= \hat{x}_2(k) \\
\hat{z}_d(k) &= \hat{x}_3(k)+\hat{x}_6(k)+\hat{x}_9(k) \\
\hat{\dot{z}}_d(k) &= \hat{x}_4(k)\hat{x}_5(k)+\hat{x}_7(k)\hat{x}_8(k)+\hat{x}_{10}(k)\hat{x}_{11}(k) \\
\hat{\ddot{z}}_d(k) &= -\hat{x}_3(k)\hat{x}_5^2(k)-\hat{x}_6(k)\hat{x}_8^2(k)-\hat{x}_9(k)\hat{x}_{11}^2(k)
\end{aligned} \tag{22}$$

Vision based relative distance calculator

For the above UKF, measurement of the acceleration from the IMU and the relative distance measurement are necessary.We now present a vision based relative distance calculator. The relative distance is obtained from a monocular camera by observing an AprilTag pattern as Figure. 7. The AprilTag is introduced by Olson (Ref. 34). It is a novel visual fiducial system that enables a single camera to measure the relative rotation and translation. The AprilTags have been used by (Refs. 3, 4) as visual feedback to land a quadrotor. The benefits of using the AprilTag include the following:

1. It is robust to occlusions, warping and rotation. The tag can be recognized even if the lens is not perfectly focused.
2. The software is open source so modifications can be made to implement the system into this case.
3. The tag detection method is simple enough to be run at 20 fps to 25 fps (Frame Per Second). This update rate is fast enough for the motion estimation module to work properly.

In order to derive the relative distance measurement, the camera must be calibrated first. Here, the camera calibration functions of OpenCV (Ref. 35) are used. The intrinsic parameters from the camera calibration are passed into the AprilTag code to get an accurate distance measurement.

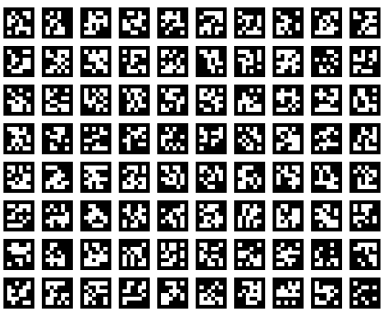

Fig. 7. This is an AprilTag used in our experiment. The pattern is printed and each tag has a perimeter of 36 millimeters

EXPERIMENTAL AND SIMULATION RESULTS

Experimental setup

In our experiment, the Nano Plus quadrotor from KMel Robotics is used. The vertical multi-sinusoidal motion of the platform is generated using an XSlide linear stage from Velmex Inc. The absolute heights of the quadrotor and the stage are measured by an external motion capture system Optitrack. The Optitrack system sends the position information to a computer through the TCP connection. The computer runs the controller through Matlab and sends the control command wirelessly to the quadrotor through a ZigBee module. The camera used in the experiment is a Logitech C270 webcam, with a resolution of 640×480. The experiment setup is illustrated in Figure. 8.

We now present six experiments and simulations to validate the three modules. The first demonstrates relative distance measurement algorithm (from camera images) in experiment, while the second demonstrates the ability to estimate multiple sinusoidal motion in simulation. These two experiments are used to validate the effectiveness of the motion estimation module using a monocular camera. The third is a simulated comparison between the proposed control architecture with a benchmarked landing algorithm proposed in (Ref. 1). This simulation is to demonstrate the time optimal characteristic of the trajectory generation module. The following ground effect compensation experiment and the mass adaptation experiment are then carried out to illustrate the adaptation capability tracking control module. The final landing experiment is to show the overall fast, precise and safe landing capability of the control architecture, without a camera but with relative position measurements from the Optitrack System.

Experiment evaluation of relative distance measurement accuracy

The objective of this experiment is to demonstrate the relative distance measurement accuracy and get a benchmark for simulation purpose. In this experiment, the camera is fixed. The AprilTag pattern (Figure. 7) is pasted on a platform. To

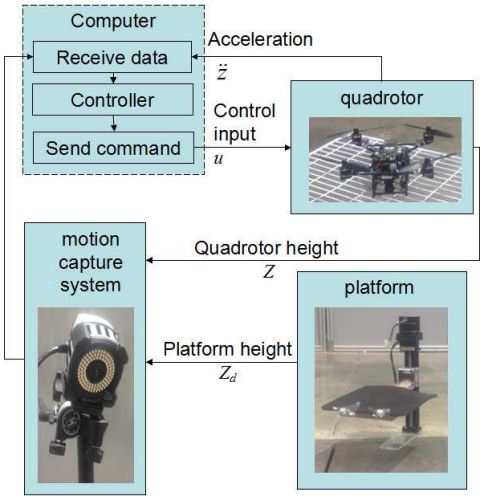

Fig. 8. This figure shows the experiment setup. The wave motion is generated by a linear stage, there are markers on the platform as well as the quadrotor. The motion capture system captures the markers feed the relative distance to the computer. The computer runs the control architecture and sent the thrust command to the quadrotor wirelessly at 50 Hertz

improve the processing speed, the resolution of the camera is reduced to 320×240. The square length of each AprilTag is 9mm and the intrinsic parameter for the camera is obtained after calibration as in equation (23).

$$M = \begin{bmatrix} 410 & 0 & 157 \\ 0 & 410 & 118 \\ 0 & 0 & 1 \end{bmatrix} \tag{23}$$

The platform is controlled to move sinusoidally in vertical direction. The ground truth is obtained by tracking the platform motion using the motion capture system. The distance measurement from the camera and the ground truth are compared. Figure.9 shows the comparison result. The standard deviation of the measurement is 1.7 mm. Figure.10 shows the measurement comparison on a multiple sine function motion. The standard deviation of the measurement error is found to be 3.1 mm. Based on this, we note that the relative distance measurement based on vision is accurate for landing.

Simulation evaluation of motion estimation

We now present a simulation to evaluate the effectiveness of estimation the motion of a multiple sinusoidal motion based on acceleration and relative distance measurement. In this simulation, the platform motion is modeled as equation (24).

$$z_d(t) = 40\sin(2t) + 20\sin(1t) + 30\sin(4t) \tag{24}$$

The quadrotor is assumed to be hovering at 500 mm above the platform with a slow sinusoidal motion. The height of the quadrotor $z_(t)$ is modeled as equation (25).

$$z(t) = 500 + 10\sin(0.1t) \tag{25}$$

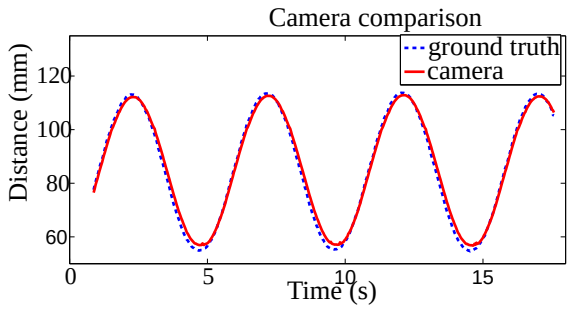

Fig. 9. This is an experimental comparison of distance measurement between a monocular camera (the solid red line) and the motion capture system (the blue dashed line). From the plot, we note that the camera can capture the relative distance quite well.

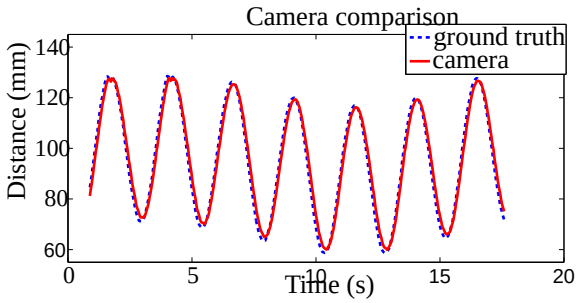

Fig. 10. This is the an experimental comparison of distance measurement for a multiple sine motions. The error between the camera measurement (red line) and the ground truth (blue dashed line) is small.

The motion estimation starts with an initial frequency guess $[2.5\ 1.5\ 4] rad/s$ and an initial height estimation of $500mm$. The sampling time T_s is set to be 0.02 second. The relative distance is given to the motion estimation module by subtracting the platform height $z_d(t)$ from the quadrotor height $z(t)$. Figure.11 shows the comparison between the estimated quadrotor height and the actual quadrotor height. Figure.12 shows the frequency tracking result. From the result, it is able to track the frequencies well after 10 seconds. Figure.13 shows the comparison between the estimated platform motion and the actual motion. From the results of given figures, it is clear that the motion estimation module is able to accurately estimate both the platform motion and quadrotor motion based only on acceleration measurement and relative distance measurement.

Simulation evaluation of landing performance comparison

This simulation demonstrates the trajectory comparison between proposed method with the method used in (Ref. 1) for bench-marking purposes. It is key to note though that in (Ref. 1), the only information available as feedback measurement is optical flow from the camera. The simulation environment is set to be the same as (Ref. 1). The platform motion is modeled as a sinusoidal signal as in (26), and the

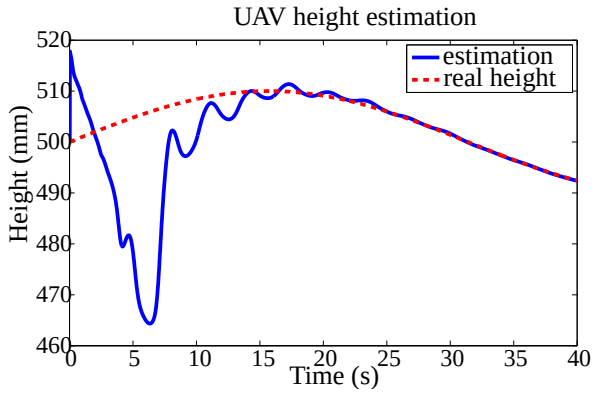

Fig. 11. Comparison of the estimated quadrotor height (solid blue line) with the actual quadrotor height (dotted red line). The actual height is slowly time varying and the estimation is able to converge to the actual height accurately after 20 seconds of estimation

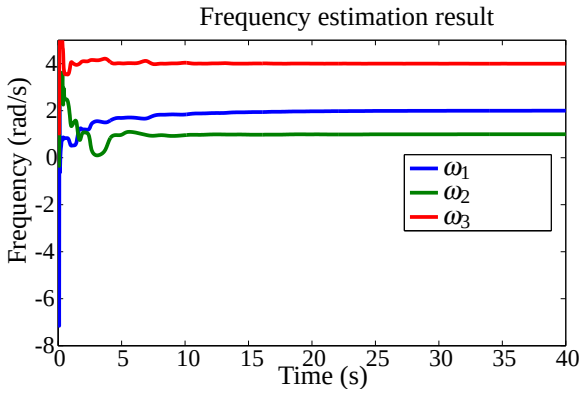

Fig. 12. Plot of frequency estimates of the three sinusoids. The solid red, blue and green lines are the frequency estimation results, namely, $[\hat{x}_3(t),\ \hat{x}_6(t),\ \hat{x}_9(t)]^T$. The frequency tracker converges to the true value of $[2\ 1\ 4]^T rad/s$ after about 10 seconds.

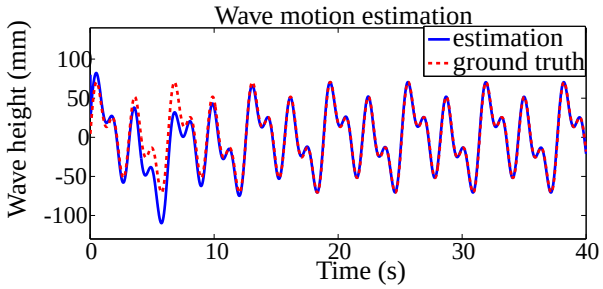

Fig. 13. Plot of wave height estimation. The dotted red line is the actual wave height while the solid blue line is the estimated wave height. There are noticeable tracking errors in the first 10 seconds. After 10 seconds, the estimation of wave height converges to the actual wave height.

quadrotor starts landing at 3 meters above the platform.

$$z_d(t) = a\sin(2\pi ft) \text{ with } a = 0.1\text{m}, f = 0.3\text{s}^{-1} \qquad (26)$$

Fig. 14 shows the comparison between the trajectory z_r of proposed method and the method in (Ref. 1). Fig. 15 compares the relative distance $z_r - z_d$ for the two methods. From Fig.15, we note that the rate of convergence is faster for the proposed method because of the time optimal characteristic of the algorithm.

Experimental evaluation of ground effect compensation

This experiment aims to demonstrate the adaptation ability of the controller for the ground effect. In this experiment, a quadrotor lands onto a stationary ground using the proposed control architecture. The motion estimation module is not activated. Two cases are compared:1) the landing without adaptation ability ($\dot{\hat{\theta}}$ in (8) is set to be zero and $\hat{\theta}$ is set to be a constant. 2) the landing with adaptation. Fig.16 shows the reference trajectory generated and the actual height for case 1). Fig.17 shows the reference trajectory generated and the actual height for case 2). The deviation from the reference trajectory near the ground in Fig.16 is caused by ground effect. From the figures, the controller in case 2) successfully adapted to the ground effect thus have a better tracking performance. So the proposed controller has a good ground effect adaptation ability.

Experimental demonstration of mass adaptation

This experiment aims to demonstrate the adaptation ability of the controller during landing under inaccurate initial guess of ground effect $\hat{\theta}$. Figure.(18) shows the calculated thrust command sent to the quadrotor under five different cases during the landing process. In the first 0.5 second, the thrust commands are different for different cases, the quadrotor also descends at different acceleration. After few seconds, the thrust commands becomes similar so the descending acceleration are similar in different cases. Figure.(19) shows the different trajectories in the landing. The case with overestimation descends slower than underestimation case in the beginning as larger thrust is generated . Then all the trajectories converge after few second. At this time, the quadrotor has a similar descending acceleration and velocity. The experimental result shows the control architecture is able to adapt to system uncertainties during the landing process.

Experimental demonstration of landing on a partially estimated oscillating platform

This experiment demonstrates the landing result of the quadrotor onto an oscillating platform while the motion is only partially estimated. The motion of the platform consists of two sinusoidal components as (27).

$$y = 10\sin(0.5t) + 25\sin(3t)(\text{mm}) \qquad (27)$$

In the motion estimation part, only the higher frequency term $25\sin(3t)$ is estimated. Figure.(20) shows the comparison between the platform estimation and actual platform motion.

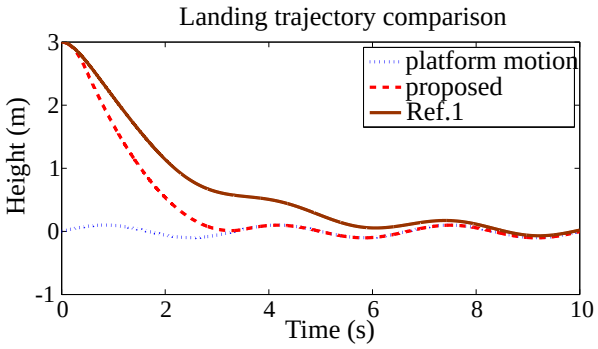

Fig. 14. This figure compares the height in the landing process. The proposed method (red line) decreases faster than (Ref. 1) (brown line) method

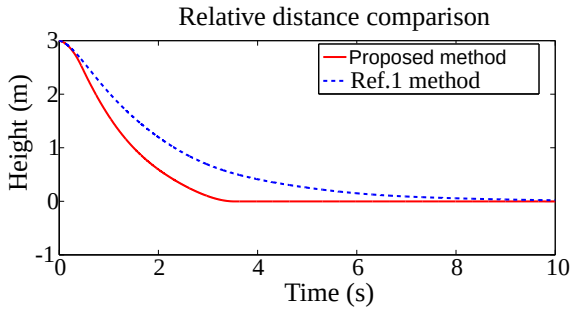

Fig. 15. This figure shows the relative distance between the proposed method (red line) and (Ref. 1) (brown line) method. The relative distance of (Ref. 1) method converges to zero exponentially, and the proposed method has a faster convergence rate

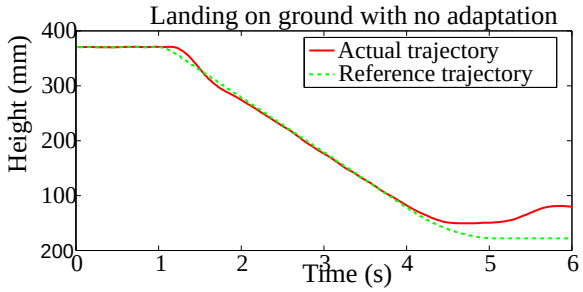

Fig. 16. This figure shows the landing trajectory onto the ground with no adaptation. When the quadrotor (red line) approaches to the ground, there is huge deviation from the reference trajectory (brown line), which is caused by the ground effect.

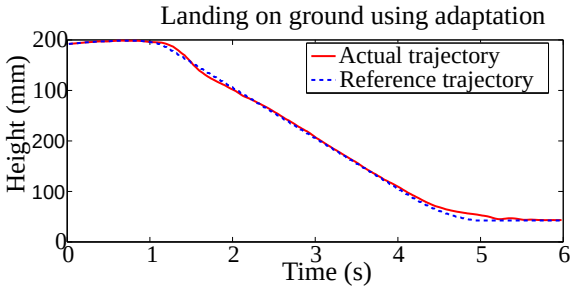

Fig. 17. This figure shows the landing trajectory onto the ground with adaptation. The height (red line) tracks the reference trajectory (brown line) well even near the ground.

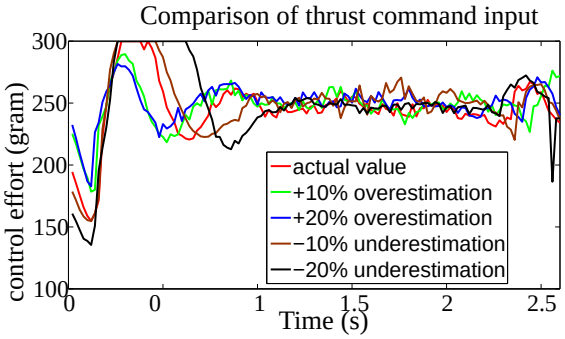

Fig. 18. This figure shows the thrust command input to the quadrotor under different initial mass guesses during the landing operation. All the thrust values converge within a few seconds, demonstrating the speed of adaptation.

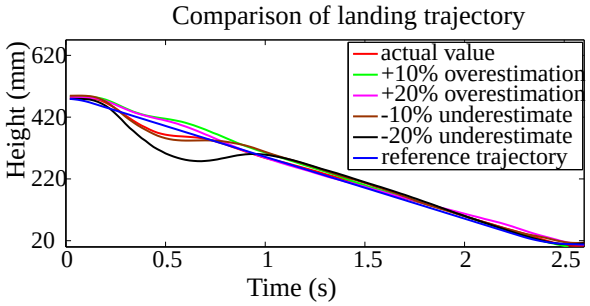

Fig. 19. This figure shows the landing trajectory under different initial mass guesses. For all cases, the trajectories converge to the reference trajectory after 1 second.

The higher frequency component is well estimated while the lower frequency term is not reflected in the estimation. Figure.(21) shows the comparison between actual landing trajectory of the quadrotor and the reference trajectory. The quadrotor is able to follow the reference trajectory well, demonstrating that the control architecture is robust in motion uncertainties.

CONCLUSIONS

In this paper, we have presented a novel control architecture that achieves the fast and accurate landing of VTOL UAV onto

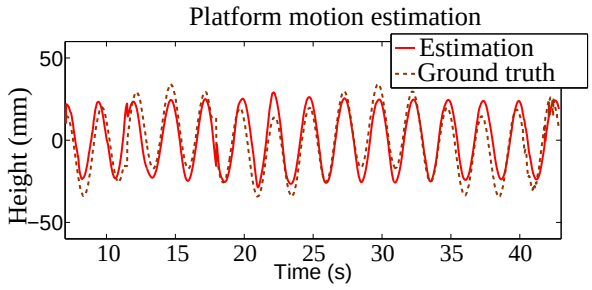

Fig. 20. This figure shows the platform motion estimation result. The estimation (solid blue line) successfully captures the significant component of the platform motion(dashed brown line). The error between these two lines is the low frequency component of the motion

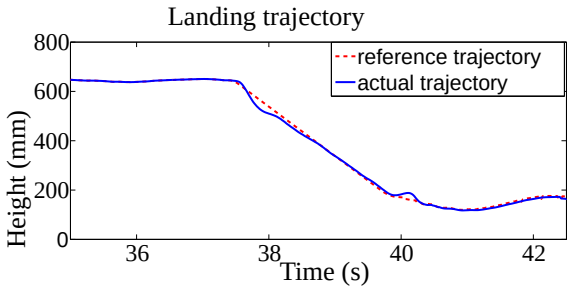

Fig. 21. This figure shows the landing trajectory (solid brown line) of the quadrotor on to the oscillating platform. The trajectory follows the reference trajectory (dashed blue line) well. The quadrotor completes the landing maneuver at time $t = 41$ sec.

a vertically moving platform. The control architecture consists of three modules: a motion estimation module; a trajectory generation module and a tracking control module. Experimental and simulation results show that the proposed control architecture is able to perform fast, safe and precise landing on an oscillating platform with robustness against ground effect, nonlinearity in the system dynamics, and system uncertainties.

In our future work, we will first focus on the autonomous landing experiment with vision and then focus on the autonomous landing on a rolling and pitching platform. Finally, the algorithm will be extended to 3D landing case where the initial position of the quadrotor differs from that of the platform in all the x, y and z directions.

Author contact: Botao Hu, hub2@rpi.edu; Lu Lu, lulu.lvlv@gmail.com, Sandipan Mishra, mishrs2@rpi.edu.

ACKNOWLEDGMENTS

The authors would like to thank S.Saha and Dr. A.Julius from the Electrical and Computer Science Department at RPI. This work was also funded in part by Sikorsky Inc.

REFERENCES

[1]Herisse, B., Hamel, T., Mahony, R., and Russotto, F.-X., "Landing a VTOL Unmanned Aerial Vehicle on a Moving Platform Using Optical Flow," *Robotics, IEEE Transactions on*, Vol. 28, (1), Feb 2012, pp. 77–89.

[2]Oh, S.-R., Pathak, K., Agrawal, S., Pota, H., and Garratt, M., "Approaches for a tether-guided landing of an autonomous helicopter," *Robotics, IEEE Transactions on*, Vol. 22, (3), June 2006, pp. 536–544.

[3]Lee, D., Ryan, T., and Kim, H., "Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing," Robotics and Automation (ICRA), 2012 IEEE International Conference on, May 2012.

[4]Ling, K., Chow, D., Das, A., and Waslander, S., "Autonomous Maritime Landings for Low-Cost VTOL Aerial Vehicles," Computer and Robot Vision (CRV), 2014 Canadian Conference on, May 2014.

[5]Benjamin Truskin, J. W., "Vision-based deck state estimation for autonomous ship-board landing," AHS 69th Auunal Forum, May 2013.

[6]Marconi, L., Isidori, A., and Serrani, A., "Autonomous vertical landing on an oscillating platform: an internal-model based approach," *Automatica*, Vol. 38, (1), 2002, pp. 21 – 32.

[7]Yang, X., Pota, H., Garratt, M., and Ugrinovskii, V., "Prediction of vertical motions for landing operations of UAVs," Decision and Control, 2008. CDC 2008. 47th IEEE Conference on, Dec 2008.

[8]Triantafyllou, M., Bodson, M., and Athans, M., "Real time estimation of ship motions using Kalman filtering techniques," *Oceanic Engineering, IEEE Journal of*, Vol. 8, (1), Jan 1983, pp. 9–20.

[9]Sanchez-Lopez, J., Saripalli, S., Campoy, P., Pestana, J., and Fu, C., "Toward visual autonomous ship board landing of a VTOL UAV," Unmanned Aircraft Systems (ICUAS), 2013 International Conference on, May 2013.

[10]Bittanti, S. and Savaresi, S., "On the parametrization and design of an extended Kalman filter frequency tracker," *Automatic Control, IEEE Transactions on*, Vol. 45, (9), Sep 2000, pp. 1718–1724.

[11]Hou, M., "Estimation of Sinusoidal Frequencies and Amplitudes Using Adaptive Identifier and Observer," *Automatic Control, IEEE Transactions on*, Vol. 52, (3), March 2007, pp. 493–499.

[12]Saripalli, S. and Sukhatme, G., "Landing a Helicopter on a Moving Target," Robotics and Automation, 2007 IEEE International Conference on, April 2007.

[13]Hehn, M. and D'Andrea, R., "Quadrocopter trajectory generation and control," Proceedings of the IFAC world congress, 2011.

[14]Mueller, M., Hehn, M., and D'Andrea, R., "A computationally efficient algorithm for state-to-state quadrocopter trajectory generation and feasibility verification," Intelligent

Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, Nov 2013.

[15]Mellinger, D. and Kumar, V., "Minimum snap trajectory generation and control for quadrotors," Robotics and Automation (ICRA), 2011 IEEE International Conference on, May 2011.

[16]Augugliaro, F., Schoellig, A., and D'Andrea, R., "Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach," Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, Oct 2012.

[17]Dougherty, J., Lee, D., and Lee, T., "Laser-based guidance of a quadrotor uav for precise landing on an inclined surface," American Control Conference (ACC), 2014, June 2014.

[18]Mellinger, D., Michael, N., and Kumar, V., "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *The International Journal of Robotics Research*, Vol. 31, (5), 2012, pp. 664–674.

[19]Hoffmann, G. M., Wasl, S. L., and Tomlin, C. J., "Quadrotor helicopter trajectory tracking control," In Proc. AIAA Guidance, Navigation, and Control Conf, 2008.

[20]How, J., Bethke, B., Frank, A., Dale, D., and Vian, J., "Real-time indoor autonomous vehicle test environment," *Control Systems, IEEE*, Vol. 28, (2), April 2008, pp. 51–64.

[21]Bouabdallah, S. and Siegwart, R., "Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor," Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on, April 2005.

[22]Guenard, N., Hamel, T., and Mahony, R., "A Practical Visual Servo Control for an Unmanned Aerial Vehicle," *Robotics, IEEE Transactions on*, Vol. 24, (2), April 2008, pp. 331–340.

[23]Hauser, J., Sastry, S., and Meyer, G., "Nonlinear control design for slightly non-minimum phase systems: Application to V/STOL aircraft," *Automatica*, Vol. 28, (4), 1992, pp. 665 – 679.

[24]Mahony, R. and Hamel, T., "Adaptive compensation of aerodynamic effects during takeoff and landing manoeuvers for a scale model autonomous helicopter," *European Journal of Control*, Vol. 0, (0), 2001, pp. 1–15.

[25]Mueller, F., Schoellig, A., and D'Andrea, R., "Iterative learning of feed-forward corrections for high-performance tracking," Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, Oct 2012.

[26]Hamer, M., Waibel, M., and D'Andrea, R., "Knowledge transfer for high-performance quadrocopter maneuvers," Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, Nov 2013.

[27]Hu, L. L., B. and Mishra, S., "Fast, safe and precise landing of a quadrotor on an oscillating platform," American Control Conference, 2015., June 2015.

[28]Mahony, R., Kumar, V., and Corke, P., "Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor," *Robotics Automation Magazine, IEEE*, Vol. 19, (3), Sept 2012, pp. 20–32.

[29]Leishman, J. G., *Principles of helicopter aerodynamics*, Cambridge aerospace series, Cambridge University Press, Cambridge, New York, 2000.

[30]Yao, B. and Xu, L., "Adaptive robust motion control of linear motors for precision manufacturing," *Mechatronics*, Vol. 12, (4), 2002, pp. 595 – 616.

[31]Lu, L. and Yao, B., "A performance oriented multi-loop constrained adaptive robust tracking control of one-degree-of-freedom mechanical systems: Theory and experiments," *Automatica*, Vol. 50, (4), 2014, pp. 1143 – 1150.

[32]Lu, L., Yao, B., and Lin, W., "A two-loop contour tracking control for biaxial servo systems with constraints and uncertainties," American Control Conference (ACC), 2013, June 2013.

[33]Chung, J. C., Bien, Z., and Kim, Y. S., "A note on ship-motion prediction based on wave-excitation input estimation," *Oceanic Engineering, IEEE Journal of*, Vol. 15, (3), 1990, pp. 244–250.

[34]Olson, E., "AprilTag: A robust and flexible visual fiducial system," Robotics and Automation (ICRA), 2011 IEEE International Conference on, May 2011.

[35]Bradski, G. and Kaehler, A., *Learning OpenCV: Computer vision with the OpenCV library*, " O'Reilly Media, Inc.", 2008.