# Real-Time Autonomous Take-off, Tracking and Landing of UAV on a Moving UGV Platform

**Conference Paper** · June 2016

DOI: 10.1109/MED.2016.7535886



**4 authors**, including:

Khaled Ali Ghamry
Concordia University
**19** PUBLICATIONS **609** CITATIONS

SEE PROFILE

Mohamed A. Kamel
Military Technical College
**38** PUBLICATIONS **1,006** CITATIONS

SEE PROFILE

Youmin Zhang
Concordia University
**782** PUBLICATIONS **26,637** CITATIONS

SEE PROFILE

24th Mediterranean Conference on Control and Automation (MED)
June 21-24, 2016, Athens, Greece

# Real-Time Autonomous Take-off, Tracking and Landing of UAV on a Moving UGV Platform

Khaled A. Ghamry[1], Yiqun Dong[1], Mohamed A. Kamel[1], and Youmin Zhang[1,2]

*Abstract*— **This paper presents a control strategy for take-off, tracking, and landing of a quadrotor unmanned aerial vehicle (UAV) on an unmanned ground vehicle (UGV) to be applied to missions of forest fires monitoring, detection, and fighting and other applications. A combination of sliding mode control (SMC) and linear quadratic regulator (LQR) is presented as the UAV local controller, while pure-pursuit strategy is applied as the UGV controller. Leader-follower formation controller approach is used during take-off, tracking and landing phases based on SMC. Experimental results are presented in order to demonstrate the performance of the team in different scenarios.**

*Index Terms*— **Autonomous landing, unmanned aerial vehicles, unmanned ground vehicles.**

## I. INTRODUCTION

Recently, cooperative control of multiple unmanned vehicles have attracted a great deal of attention from scientific, industrial, and military aspects due to their advantages such as flexibility, task allocation, fault tolerance, cost effectiveness, and safety to achieve the desired outcome. These applications include [1]: surveillance, search and exploration, formation control, cooperative reconnaissance, environmental monitoring, and cooperative manipulation.

Unmanned aerial vehicles (UAVs) usually have limited payload since the main features of UAV design is to be smaller with better maneuverability for easy use and operation. As a result, they have a limited battery life, and consequently limited running time. On the other hand, unmanned ground vehicles (UGVs) offer high payload and longer running time. The pairing of UAVs with UGVs can solve the problem of limitation of UAV in long endurance necessary to a practical mission. UGVs can be used to transport UAVs from a ground station to a place close enough for UAVs to take-off for mission execution. Therefore, the coordination between UAVs and UGVs receives a great interest in reconnaissance, surveillance, and exploration applications [2]. Another advantage of pairing the two different types of autonomous vehicles is the ability to gain multiple perspectives during a mission [3]. In order to have a fully autonomous system that pairs a UAV with a UGV, it is of great importance that the vehicles have the ability to rendezvous autonomously and perform coordinated landing [3].

Autonomous landing and tracking of aerial vehicle on a ground one has not fully investigated in the literature. In [4], an autonomous landing algorithm for quadrotor helicopter on a moving platform is presented. A camera is mounted on the helicopter as a target observer, and the Kalman filter is used to track the target using the data of the camera. Behavior-based approach is applied to track the reference trajectory during flight. In [5], an algorithm for landing a quadrotor on a moving platform is presented, while the platform is moving with constant velocity. In [2], a monocular vision based approach for cooperation between UAV and UGV is presented. The UAV can autonomously track and land on the moving UGV by tracking a target marker on the UGV. In [3], a controller for autonomous coordinate landing between a quadrotor UAV and a skid-steered UGV is presented. A coordinated controller that incorporates the state of both vehicles is designed to drive them together, and the effects of time delays in the communications are addressed.

The problem of autonomous take-off, tracking, and landing of a UAV on a moving UGV platform is addressed in this work for mimicking forest fires monitoring, detection, and fighting and other relevant applications. Local controllers for each vehicle is designed to track their reference trajectories, while a formation controller is developed in a leader-follower manner using a sliding mode control (SMC) [6]. The UGV is assumed as the leader, and UAV follows it during tracking and landing stages. Two cases are addressed in this work. The first one is the regular tracking and landing. While in the second one, the UAV is assumed to accomplish a specific mission. It takes-off from the UGV, tracks it, and then it is assumed that it received a command to do a specific mission. The UAV will accomplish the mission and then track and land on the UGV finally to be prepared for recharging/refueling to next mission. The main contributions of this work are: 1) developing novel algorithm for tracking and landing a UAV on a moving UGV platform in different scenarios; and 2) real-time implementation and validation with a team of UAV-UGV testbeds available at the Networked Autonomous Vehicles Laboratory of Concordia University (NAV-Lab) for the proposed control algorithms.

The paper is organized as follows. In Section II the description of the UAV and UGV, their control architecture and mathematical model are presented. Formation controller for tracking and landing is developed in Section III. Section IV presents the experimental results for different scenarios, and in Section V conclusions of this work are presented.

[1]Khaled A. Ghamry, Yiqun Dong, Mohamed A. Kamel and Youmin Zhang are with the Department of Mechanical and Industrial Engineering, Concordia University, Montréal, Quebéc H3G 1M8, Canada. k_ghamry@encs.concordia.ca, yiqundong90@gmail.com,m_atef@encs.concordia.ca, youmin.zhang@concordia.ca

[2]Youmin Zhang is with the Department of Information and Control Engineering, Xi'an University of Technology, Xi'an, Shaanxi 710048, China youmin.zhang@xaut.edu.cn (On sabbatical leave from Concordia University; Financially supported by NSERC, NSFC #61573282 and SPNSF #2015JZ020; Corresponding Author)

## II. PRELIMINARIES

To develop the tracking and landing controller, the local controllers for both the UAV and UGV should be designed first. This section presents dynamic and kinematic modeling of the UAV and UGV respectively, with their local control architectures.

### A. Quadrotor UAV Dynamics and Modeling

The UAV considered here is a quadrotor helicopter, where four rotors laid up symmetrically around its center as illustrated in Fig. 1a. The quadrotor dynamic model is obtained using Lagrange method. The simplified quadrotor dynamics are:

$$\begin{aligned} \ddot{x} &= (\sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi)\frac{U_z}{M} & \ddot{\phi} &= \frac{U_\phi}{J_{xx}} \\ \ddot{y} &= (\sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi)\frac{U_z}{M} & \ddot{\theta} &= \frac{U_\theta}{J_{yy}} \\ \ddot{z} &= -g + (\cos\theta\cos\phi)\frac{U_z}{M} & \ddot{\psi} &= \frac{U_\psi}{J_{zz}} \end{aligned} \tag{1}$$

where $\phi$, $\theta$, and $\psi$ are the Euler angles which represent roll, pitch and yaw respectively. $M$ is the quadrotor mass. $J_{xx}$, $J_{yy}$, and $J_{zz}$ are the quadrotor moment of inertia according to $x$, $y$, and $z$ axes respectively. Forces and moments along quadrotor axes in Eq. (1) can be defined as:

$$\begin{aligned} U_z &= F_1 + F_2 + F_3 + F_4 \\ U_\phi &= L(F_3 - F_4) \\ U_\theta &= L(F_1 - F_2) \\ U_\psi &= K_{yaw}(F_1 + F_2 - F_3 - F_4) \end{aligned} \tag{2}$$

where $F_i, i = 1, \ldots, 4$ are the thrust generated by the four propeller's, $L$ is the distance between the motor and the quadrotor center. $K_{yaw}$ is a constant relating the propellers thrust with the yawing moment. The quadrotor forces and moments in Eq. (2) can be obtained by:

$$F_i = K\frac{\omega_m}{s + \omega_m}u_i \tag{3}$$

where $K$ is a positive gain, $\omega_m$ is the motor bandwidth, and $u_i$ is the control input.

### B. UGV Kinematics and Modeling

The robots considered in this work are differentially driven wheeled mobile robots (WMRs). The rear wheels are active and independent in performing driving and steering of the robot, As shown in Fig. 1b, point $q$ is the current posture with coordinates $(x,y)$ and orientation angle $\phi$. Assuming non-deforming wheels and robot moves without slipping, then defining the vector $q(t) = [x(t),y(t),\phi(t)]^T$. Differentially driven WMR is nonholonomic resulting from the assumption that the robots cannot slip in lateral direction. The nonlinear kinematic equations of the robot are represented as follows:

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\phi & 0 \\ \sin\phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{4}$$

where $v$ and $\omega$ are the linear and angular velocities of the robot. The right and left velocities of the robot wheels are:

$$v_R = v + h\omega, \qquad v_L = v - h\omega \tag{5}$$

where $h$ is the distance between the vehicle longitudinal axis and each wheel.

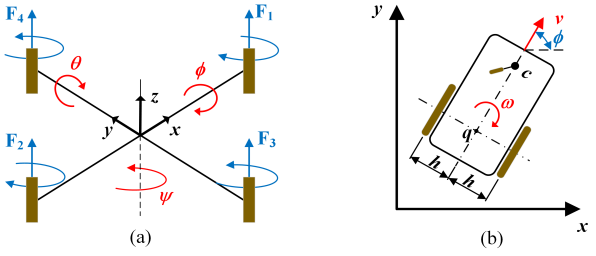



Fig. 1. (a) Coordination system of the quadrotor UAV. (b) Schematic diagram of the differentially-driven wheeled mobile robot

### C. UAV Control

UAV control algorithm is based on a combination of sliding mode control (SMC) and linear quadratic regulator (LQR). Details of this work can be found in authors' previous work [6]. A brief explanation is presented here for convenience, and Fig. 2 shows the control system block diagram.

Due to nonholonomic features of a quadrotor, the dynamic model is divided into two subgroups: a fully actuated system that represented by the following equation:

$$\begin{bmatrix} \ddot{z} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{U_1}{M}\cos\theta\cos\phi - g \\ \frac{U_4}{J_{zz}} \end{bmatrix} \tag{6}$$

and an under-actuated subsystem defined as:

$$\begin{aligned} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} &= \frac{U_1}{M}\begin{bmatrix} \cos\psi & \sin\psi \\ \sin\psi & -\cos\psi \end{bmatrix}\begin{bmatrix} \sin\theta\cos\phi \\ \sin\phi \end{bmatrix} \\ \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \end{bmatrix} &= \begin{bmatrix} \frac{U_2}{J_{xx}} \\ \frac{U_3}{J_{yy}} \end{bmatrix} \end{aligned} \tag{7}$$

The objective of this fully actuated subsystem controller is to minimize the error in the altitude and yaw angle $e_z$ and $e_\psi$ to satisfy the following conditions:

$$\lim_{t\to\infty} \|e_z\| = \|z_r - z\| = 0 \tag{8a}$$

$$\lim_{t\to\infty} \|e_\psi\| = \|\psi_r - \psi\| = 0 \tag{8b}$$

where $z_r$ and $\psi_r$ are the desired altitude and yaw angle respectively, this can be achieved by designing a SMC. The control laws for the altitude and yaw angle can be derived using classical SMC theory as [7]:

$$\hat{U}_1 = \left(\frac{M}{\cos\theta\cos\phi}\right)(g + \ddot{z}_r - \lambda_z\dot{e}_z) \tag{9}$$

$$\hat{U}_4 = J_{zz}(\ddot{\psi} - \lambda_\psi\dot{e}_\psi) \tag{10}$$

where $\lambda_z$ and $\lambda_\psi$ are control gains with $\lambda_z > 0$ and $\lambda_\psi > 0$. In order to satisfy sliding condition, a discontinuous term is added across the surface $s = 0$ such that:

$$U = \hat{U} - k\,\mathrm{sgn}(s) \tag{11}$$

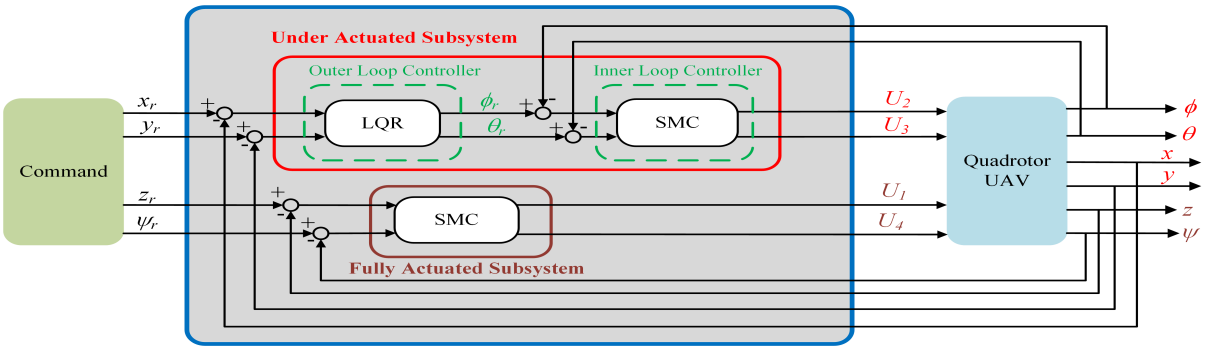



Fig. 2. UAV control system block diagram

where

$$\mathrm{sgn}(s) = \begin{cases} +1 & \text{if } s > 0 \\ -1 & \text{if } s < 0 \end{cases}$$

Two loops are designed to control the under-actuated subsystem, an outer loop and an inner loop. The objective of the outer loop controller is to obtain the desired position in $x$ and $y$ axes. This is achieved by applying LQR to the quadrotor linear dynamic model defined as:

$$\begin{aligned} \ddot{y} &= -\phi g \\ \ddot{x} &= \theta g \end{aligned} \tag{12}$$

The controller aims to find the matrix $K$ of the optimal control vector $u$ such that $u(t) = -Kx(t)$ to minimize the quadratic cost function

$$J = \int_0^\infty (x^T Q x + u^T R u) dt \tag{13}$$

where $Q$ and $R$ are the weighting matrices with $Q > 0$ and $R > 0$. The gain matrix $K$ can be obtained by solving the Ricatti equation.

The objective of the inner loop controller is to converge the actual values of the Euler angles $\phi$ and $\theta$ to their desired values $\phi_r$ and $\theta_r$ obtained from the outer loop controller. SMC is applied to generate the control inputs $U_2$ and $U_3$ to satisfy accurate quadrotor attitude stabilization. The control laws can be derived as:

$$\hat{U}_2 = J_{xx}(\ddot{\phi} - \lambda_\phi \dot{e}_\phi) \tag{14}$$

$$\hat{U}_3 = J_{yy}(\ddot{\theta} - \lambda_\theta \dot{e}_\theta) \tag{15}$$

where $\lambda_\theta$ and $\lambda_\phi$ are control gains with $\lambda_\theta > 0$ and $\lambda_\phi > 0$. $e_\phi$ and $e_\theta$ are the errors in roll and pitch angles, $e_\phi = \phi_r - \phi$ and $e_\theta = \theta_r - \theta$. $\phi_r$ and $\theta_r$ are the desired roll and pitch angles respectively. To satisfy the sliding conditions, Eq. (11) should be applied.

### D. UGV Control

UGV control algorithm is based on a pure-pursuit control. Pure-pursuit controller has been widely used for ground and aerial vehicle path (waypoints) tracking [8]. It is adopted here due to its simplicity, especially as it is an intuitive control laws which has relatively clear geometric meanings. A conceptual plot of the controller is shown in Fig. 3; generally the vehicle is turned at a constant angular velocity (constant turning radius and fixed turning origin), while the arc of the vehicle path is expected to go to the goal point. In Fig. 3, the vehicle starts turning at $O$, $P$ is the origin of turning arc, and $r$ is the vehicle turning radius. Goal point of the vehicle is $G$. Based on the geometric relation in Fig. 3:

$$\begin{aligned} r^2 &= d^2 + y^2 \\ r &= d + |x| = d - x \end{aligned} \tag{16}$$

one could have

$$r = -\frac{x^2 + y^2}{2x} - \frac{L^2}{2x} \tag{17}$$

In Eq. (17), $L$ is defined as the look-forward distance of pure-pursuit controller. Also in (17), as depicted on Fig. 3, vehicle turning to counter-clockwise corresponds to an algebraic positive turning radius.

From UGV kinematics:

$$\dot{\phi} = (V_R - V_L)/2h \tag{18}$$

$$V_c = (V_R + V_L)/2 \tag{19}$$

$$V_c = r\dot{\phi} \tag{20}$$

one could have:

$$\begin{aligned} V_R - V_L &= 2h\frac{V_c}{r} \\ V_R + V_L &= 2V_c \end{aligned} \tag{21}$$

Substitute $r$ into Eq. (21):

$$\begin{aligned} V_R &= V_c\left(1 - \frac{2h}{L^2}x\right) \\ V_L &= V_c\left(1 + \frac{2h}{L^2}x\right) \end{aligned} \tag{22}$$

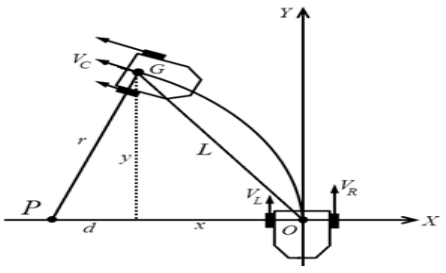



Fig. 3. Pure-pursuit motion controller

where $V_R$ and $V_L$ are the desired speed for right and left wheel respectively. In (22), note that $x$ is the lateral displacement of goal point defined in vehicle body axis. Speed of the vehicle is denoted as $V_c$ which could be controlled by a separate speed controller, or as in the work discussed herein a constant value of 0.10 *m/s* is used. To specify the value of the look-forward distance $L$, nevertheless, to authors best knowledge there is not a universal standard of deciding this value.

## III. TRACKING AND LANDING CONTROLLER

In this section, the controller is designed to achieve the desired formation using a decentralized leader-follower method. The UGV is assigned as the leader, and the UAV is the follower as shown in Fig. 4. The objective of the formation controller is to generate the commands for the UAV controller based on the UGV position and the desired height.

The proposed control algorithm is shown in Fig. 5 by applying SMC in its design to keep the formation even in perturbed and uncertain environment. During take-off, the formation controller sends the commands to the UAV local controller to stabilize the UAV position at the desired height $Z_d$. During tracking, the UAV controller should maintain the UAV at the desired height $Z_d$ as follows:

$$\lim_{t\to\infty} \|e_z\| = \|z_f - z_l\| = Z_d \tag{23}$$

Finally, during landing the commands sent to the UAV should satisfy the following conditions:

$$\lim_{t\to\infty} \|e_x\| = \|x_l - x_f\| = 0 \tag{24a}$$

$$\lim_{t\to\infty} \|e_y\| = \|y_l - y_f\| = 0 \tag{24b}$$

$$\lim_{t\to\infty} \|e_z\| = \|z_l - z_f\| = 0 \tag{24c}$$

Assuming zero yaw angle for simplicity, then the formation problem is solved as:

$$\begin{aligned} \ddot{x}_f &= \ddot{x}_l + \lambda_x(\dot{x}_l - \dot{x}_f) \\ \ddot{y}_f &= \ddot{y}_l + \lambda_y(\dot{y}_l - \dot{y}_f) \\ \ddot{z}_f &= \ddot{z}_l + \lambda_z(\dot{z}_l - \dot{z}_f) \end{aligned} \tag{25}$$

where Eq. (25) denotes the desired accelerations which are then enhanced using Eq. (11) to be used as the input for the UAV to achieve the required formation.

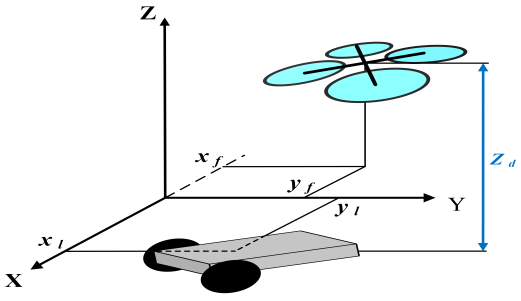



Fig. 4. Formation geometry of the UGV and the UAV

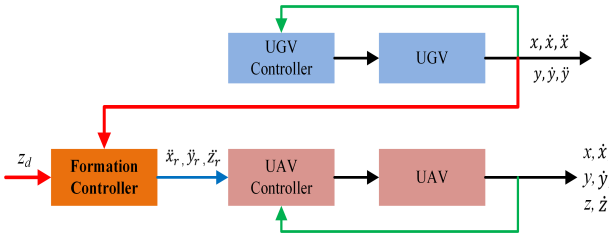



Fig. 5. Formation control scheme of the UGV and the UAV

## IV. EXPERIMENTAL RESULTS ANALYSIS

The control strategies discussed in Sections II and III are successfully implemented on a team of UAVs-UGVs. This work is performed in the Networked Autonomous Vehicles Laboratory (NAV-Lab) in the Department of Mechanical and Industrial Engineering, Concordia University. All the videos of the experiments performed can be found in the NAV-Lab YouTube Channel [9].

The experimental testbed includes the Qball-X4 as the UAV, Quanser QGV as the UGV, the ground station PC, and 24 OptiTrack cameras system as shown in Fig. 6. The Qball-X4 is a quadrotor helicopter design, propelled by four motors fitted with 10-inch propellers. The entire quadrotor is enclosed within a protective carbon fiber cage that ensures safe operation of the UAV. Quanser QGV is a two wheeled differentially driven robot with a 4 degrees of freedom robotic manipulator. Both Qball-X4 and QGV control module is comprised of a data acquisition board (HiQ DAQ) and an embedded Gumstix computer where QuaRC is the Quanser's real-time control software. Together with the Gumstix embedded computer, the HiQ controls the vehicle by reading on-board sensors and sending motor commands. The motor speed controller is connected to two PWM servo outputs on the HiQ. The on-board Gumstix computer runs QuaRC, allowing rapidly develop and deploy controllers for Qball-X4 and QGV real-time control. Runtime sensors measurement, data logging, and parameter tuning are supported between the ground host computer and both the Qball-X4 and the QGV.

Since the experiments taking place indoor in the absence of GPS, then the high level controller implemented on a PC receives the states of the Qball-X4 and the QGV from the

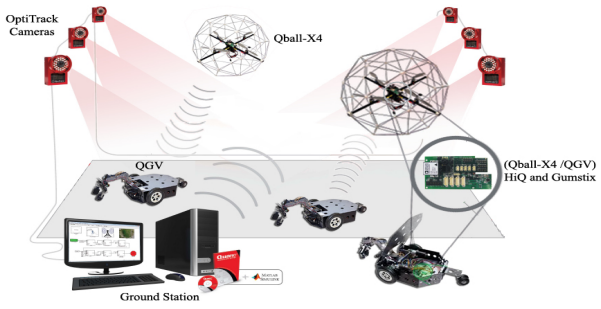



Fig. 6. Experimental setup

TABLE I

UAV CONTROLLER PARAMETERS

| | | |
|---|---|---|
| $J_{xx}$ = 0.03 kg.m$^2$ | $J_{yy}$ = 0.03 kg.m$^2$ | $M$ = 1.4 kg |
| $J_{zz}$ = 0.04 kg.m$^2$ | $K_{yaw}$ =4 N.m | $L$ = 0.2 m |
| $K$ = 120 N | $\omega_m$ =15 rad/sec | |

vision system consisting of 24 OptiTrack cameras system. The high-level controller uses this information to calculate the desired pulse width modulation (PWM) to be sent to the driving motors. The ground station PC used has a processor Intel(R) Core(TM) i7-3770 CPU @ 3.40 GHz, and 4 GB RAM. All the values of the UAV controller parameters are shown in Table I.

Two cases are presented in this work:

- Case 1: Regular tracking and landing.
- Case 2: UAV is assigned for mission execution during tracking.

### A. Case 1: Regular Tracking and Landing

In this case, the UAV and the UGV perform the regular take-off, tracking, and landing as shown in Fig. 7. The vehicles start the mission together. The UAV starts the take-off at $t$ = 13 $s$, and completes the take-off at $t$ = 20 $s$. The UGV continues following its planned path, while the UAV follows it until $t$ = 40 $s$. The UAV starts the landing stage, and finishes the landing at $t$ = 47 $s$. Then, the vehicles continue the planned path again. Fig. 8 shows the position error between the UGV and the UAV. As can be seen, the error converges to zero achieving the conditions presented in Eq. (24). Fig. 9 shows that the UAV is stabilized to the desired height during tracking for achieving the condition presented in Eq. (23). This figure also illustrates the stages of take-off, tracking, and landing, respectively.

### B. Case 2: The UAV Assigned for a Mission During Tracking

In this case, the UAV is assigned to perform a specific mission during the tracking stage. The mission located at the point (0,-0.5,0.7). As can be seen in Fig. 10, the UAV starts the take-off at $t$ = 13 $s$, and completes the take-off 7 seconds later. The UGV continues following its planned path, while

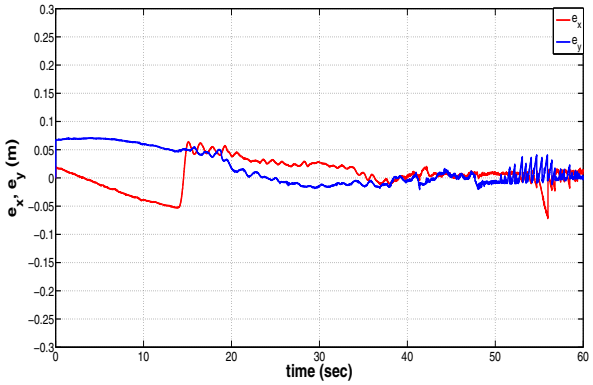



Fig. 8. Position error between UAV and UGV in Case 1

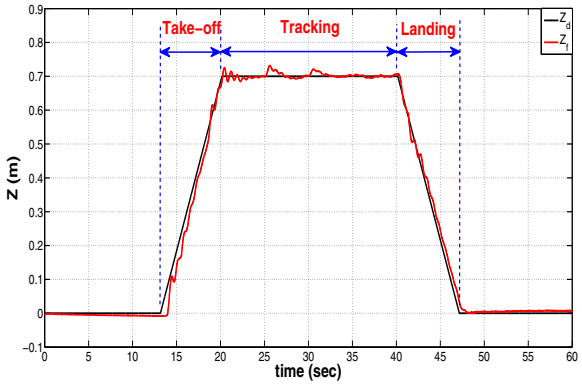



Fig. 9. The desired and actual UAV height in Case 1

the UAV follows it until $t$ = 20 $s$ when the mission is assigned for the UAV. It goes to the point (0,-0.5,0.7) in which the mission will be executed, while the UGV still following its planned path. The UAV executes the mission until $t$ = 40 $s$, then it returns to track the UGV again until landing at $t$ = 60 $s$. The UAV completes the landing on the moving UGV platform in 7 seconds, and then continues the motion on the platform. Fig. 11 shows the formation error in $x$ and $y$ axes.

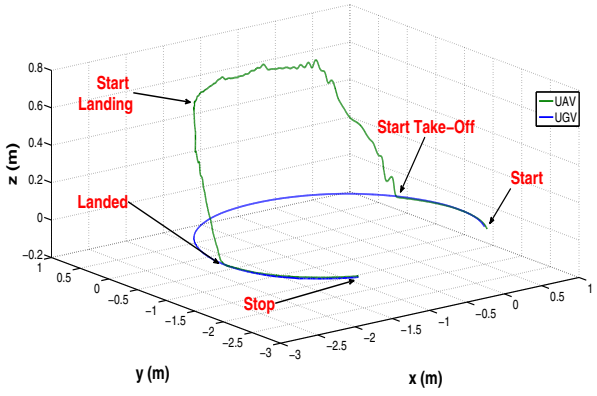



Fig. 7. UAV and UGV trajectories during Case 1

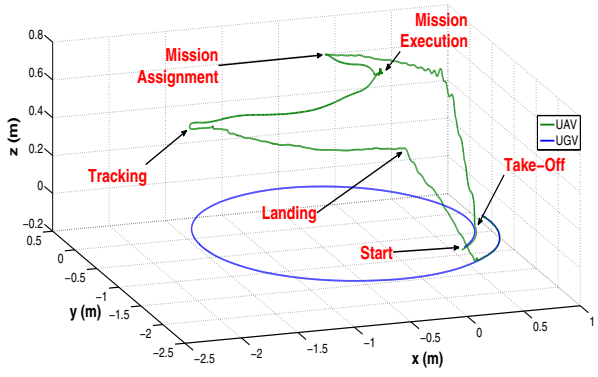



Fig. 10. UAV and UGV trajectories during Case 2

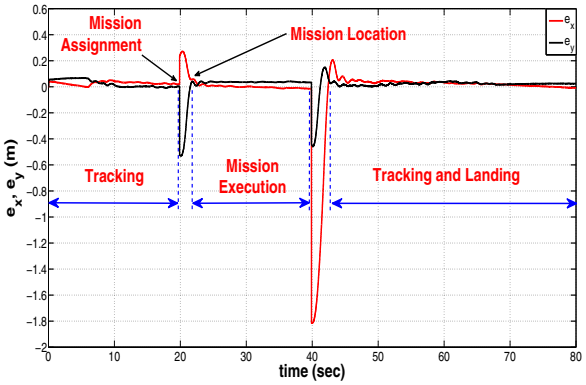



Fig. 11. Position error between UAV and UGV in Case 2

The formation error converges to zero during whole mission except the period of going to the mission way-point, and returning to track the platform.

Fig. 12 presents snapshots of the second case real experiment. When $t = 2$ $s$, the UAV moves on the UGV platform. At $t = 7$ $s$ the UAV is in the take-off stage. When $t = 30$ $s$, the UAV reaches the mission location and executes the mission now. When $t = 50$ $s$, the UAV accomplished the assigned mission and now tracks the UGV. At $t = 65$ $s$, the UAV is in the landing stage. When at $t = 75$ $s$, the UAV completes the landing on the moving UGV, and moves with it.

## V. CONCLUSION

Autonomous take-off, tracking and landing of a UAV on a UGV is discussed in this paper. The UAV controller is based on a combination of SMC and LQR, while the UGV used the pure-pursuit controller to plan its path. Formation controller based on SMC is used to allow the UAV to track the UGV in the leader-follower manner. The proposed algorithms are capable of allowing the UAV to take-off, track and land on the moving UGV. The main advantage of the proposed controller is its ability to apply in real-time as shown in the experimental results. The controller succeeds to perform the mission requirements in two different scenarios. Furthermore, it is illustrated that the control system is stabilized with good performance for accomplishing the specified missions.

## REFERENCES

[1] R. M. Murray, "Recent research in cooperative control of multivehicle systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 129, no. 5, pp. 571–583, 2007.

[2] H. Cheng, Y. Chen, X. Li, and W. S. Wong, "Autonomous takeoff, tracking and landing of a UAV on a moving UGV using onboard monocular vision," in *Chinese Control Conference (CCC)*, 2013, pp. 5895–5901.

[3] J. M. Daly, Y. Ma, and S. L. Waslander, "Coordinated landing of a quadrotor on a skid-steered ground vehicle in the presence of time delays," *Autonomous Robots*, vol. 38, no. 2, pp. 179–191, 2014.

[4] S. Saripalli and G. Sukhatme, "Landing on a moving target using an autonomous helicopter," in *Field and Service Robotics*, S. Yuta, H. Asama, E. Prassler, T. Tsubouchi, and S. Thrun, Eds. Springer Berlin Heidelberg, 2006, pp. 277–286.

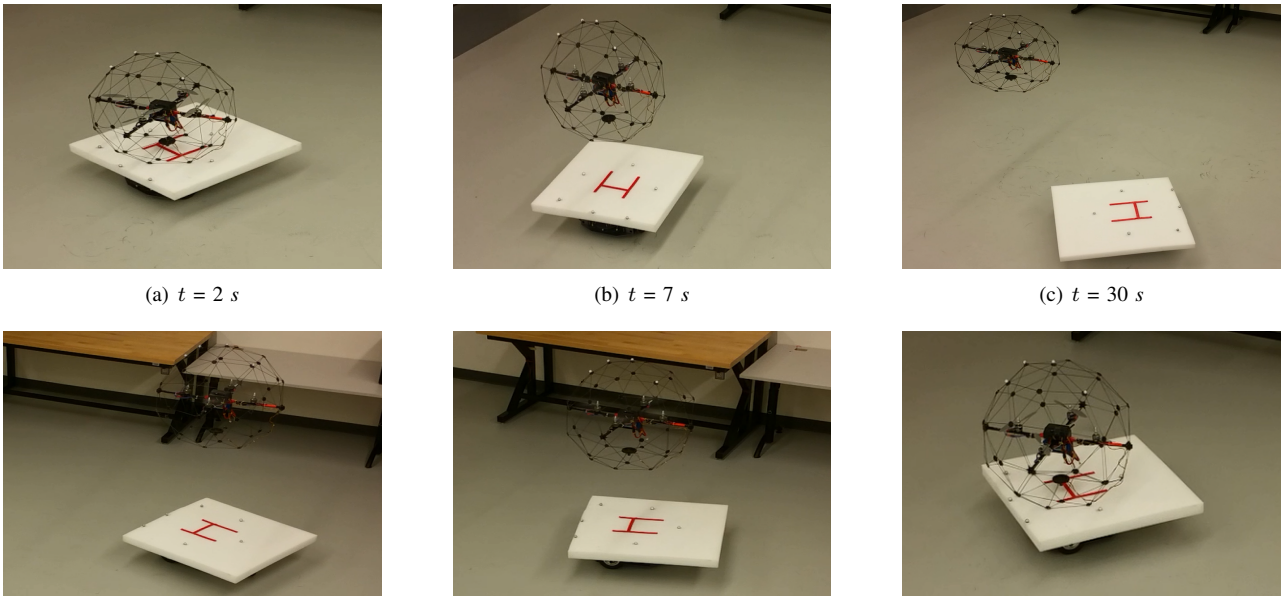
[5] H. Voos and H. Bou-Ammar, "Nonlinear tracking and landing controller for quadrotor aerial robots," in *IEEE International Conference on Control Applications (CCA)*, 2010, pp. 2136–2141.

[6] K. A. Ghamry and Y. M. Zhang, "Formation control of multiple quadrotors based on leader-follower method," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015, pp. 1037–1042.

[7] J.-J. E. Slotine and W. Li, *Applied nonlinear control*. Prentice-Hall, 1991.

[8] Y. Kuwata, S. Karaman, J. Teo, E. Frazzoli, J. P. How, and G. Fiore, "Real-time motion planning with applications to autonomous urban driving," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1105–1118, 2009.

[9] "NAV Laboratroy YouTube Channel," https://www.youtube.com/user/NAVConcordia/videos, 2015.



(a) $t = 2$ $s$ (b) $t = 7$ $s$ (c) $t = 30$ $s$

(d) $t = 50$ $s$ (e) $t = 65$ $s$ (f) $t = 75$ $s$

Fig. 12. Snapshots of Case 2 experiment