2014 American Control Conference (ACC)
June 4-6, 2014. Portland, Oregon, USA

Trajectory-Tracking Control Law Design for Unmanned Aerial Vehicles with an Autopilot in the Loop

Liang Sun[1], Randal W. Beard[2], and Daniel Pack[3]

Abstract— **This paper presents a strategy of designing a trajectory-tracking control law for unmanned aerial vehicles where an unmodifiable autopilot is incorporated in the closed-loop system. The main contributions of this paper include a novel control structure that enables an explicit design of tracking methods for a system with an unmodifiable autopilot in the loop, and a novel controller that augments the existing control law. To begin with, a controller is developed based on a generalized design model using a Lyapunov-based backstepping technique. To incorporate the constraints due to the unmodifiable autopilot, the design model and the developed controller are augmented by introducing additional terms. Lyapunov stability theory is used to prove that, without any modification of the autopilot, the augmented controller exponentially stabilizes the vehicle to a desired trajectory while the other higher order and attitude states remain bounded. A software-in-the-loop simulation result is presented to validate the strategy. The proposed strategy can be employed to ensure the vehicle remains safe while experimenting with new tracking and path following control options. This type of control design are most useful for systems where the autopilot is unmodifiable, and possibly unknown, as would occur with commercial autopilots being augmented with high level tracking control.**

I. Introduction

Controlling unmanned aerial vehicles (UAVs) poses research challenges in both theory and application. A typical onboard architecture for closed-loop control of a UAV is shown in the gray box in Fig. 1, where the autopilot stabilizes the vehicle and follows the input commands, such as the airspeed, altitude and attitude angles. To design motion control algorithms for UAVs, a specific design model (kinematics or dynamics) is typically employed to represent the UAV motion. The controller can then be developed based on that design model to achieve desired goals (e.g. trajectory tracking and path following). The dashed box in Fig. 1 illustrates the structure of the controller design using a specific design model.

In particular, given an autopilot, the gray box shown in Fig. 1 is fixed and cannot be changed. Most commercial autopilots such as Kestrel, Piccolo, etc., are black-box systems, in the sense that they implement specific autopilot functionality which cannot be altered. However, the input/output model of the autopilot/UAV system, which is usually overly simplistic, may not be sufficient to capture the design requirements for a control scheme (e.g., the car-trailer system [1] and the towed-cable system [2]). Therefore, it becomes necessary to design a control strategy based on a medium fidelity model, and then to implement that strategy on the autopilot. The current paper addresses this problem.

Recently, a variety of motion control algorithms for autonomous vehicles have been proposed. A trajectory-tracking control law was developed for UAVs in [3], where the gain-scheduled approach was employed based on a generalized error-dynamics model. In [4], a motion control strategy for marine craft was presented. The algorithm combines both trajectory-tracking and path-following objectives using the Lagrange multiplier method. The controller was developed using a Lyapunov-based backstepping technique, which is also employed in [2], [5]–[7]. In [5], the experimental results show that the actual vehicle trajectory followed the reference with relatively large tracking errors. The authors concluded that the modeling inaccuracy was the main reason. In their later work [7], a motion control strategy accounting for parametric modeling uncertainty was proposed. An adaptive switching scheme was used to pick optimal parameters from a predefined parameter set. Experimental results show that the performance was significantly improved compared to the results in [5]. In [8] and [9], a vector field based path following guidance law was developed. Straight lines and circles were employed to validate the algorithm and Lyapunov stability arguments were also presented. In [2], a trajectory-tracking control law was developed for a towing vehicle which tows a long cable connected to a drogue. A dynamic model reflecting cable tension forces was used to derive the controller that can handle wind disturbances.

In previous studies of motion planning and control for autonomous vehicles, most work focused on the derivation of control laws, while few discussed the implementation of the developed control strategy. A survey of autopilots used on small fixed-wing UAVs can be found in [10]. path following strategy for UAVs based on adaptive control techniques is found in [11] and [12], where the authors employed the waypoint following capability of an autopilot in the closed-loop dynamics. Experimental results were presented in [12] to validate the adaptive controller. As far as we know, a trajectory-tracking control law that incorporates an existing autopilot has not been presented in the literature.

In this paper, we are interested in the situation where the controllers discussed in the previous paragraph cannot be implemented on an autopilot directly, and the outputs of the controller are not the same variables as the input commands for the autopilot. In order to use an existing controller

[1] Liang Sun, Postdoctoral Fellow, Academy Center for UAS Research, US Air Force Academy, CO 80840 solbuaa@gmail.com

[2] Randal W. Beard, Professor in Department of Electrical and Computer Engineering, Brigham Young University, Provo, UT 84602 beard@byu.edu

[3] Daniel Pack, Professor in Department of Electrical and Computer Engineering, University of Texas at San Antonio, TX, 78249 daniel.pack@utsa.edu

978-1-4799-3274-0/$31.00 ©2014 AACC

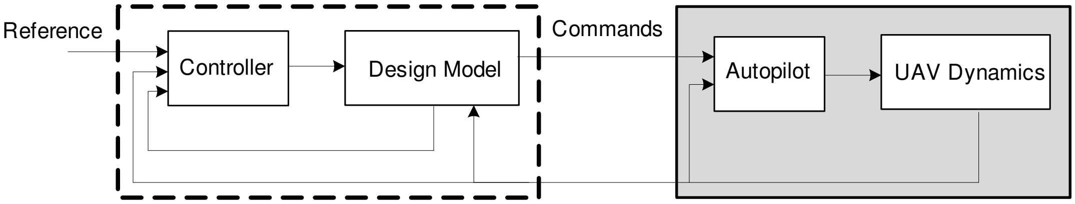


Fig. 1. Closed-loop control system for trajectory tracking with an autopilot in the loop.

without modifying the autopilot, a combined closed-loop system is proposed, as shown in Fig. 1. The objective of this paper is to develop a strategy for augmenting the overall control approach to work with an existing autopilot system to achieve asymptotic trajectory tracking.

The remainder of the paper is organized as follows. Section II presents the derivation of a trajectory-tracking control law for a dynamic system with an autopilot in the loop. Section III shows the result of a software-in-the-loop simulation using a dynamics model of a UAV, followed by some concluding remarks in Section IV.

II. Trajectory-Tracking Control Law Design with an Autopilot in the Loop

In this section, a trajectory-tracking control law is developed based on a generalized design model using a Lyapunov-based backstepping technique. The controller and the design model are then augmented to incorporate the autopilot in the closed-loop system.

A. Preliminary Trajectory-Tracking Control Law Design

A design model is typically established based on the known information of a physical system. Disturbances such as wind gusts are typically estimated using, for example, a Kalman filter, and are treated as known variables in the design model. In the following lemma, we treat disturbances as bounded known variables while the unknown component is handled in the subsequent theorem. In Lemma 1, to derive the trajectory tracking control law, we applied generalized equations of motion that can be used to represent dynamics for a fixed wing UAV.

Lemma 1: *Consider a nonlinear system with states $\hat{\eta} \in D_\eta \subset \mathbb{R}^3$, $\hat{\xi} \in D_\xi \subset \mathbb{R}^3$, $\hat{\varphi} \in D_\varphi \subset \mathbb{R}$, system inputs $\hat{\mathbf{u}} \triangleq (\hat{u}_1, \hat{u}_2, \hat{u}_3)^T \in D_u \subset \mathbb{R}^3$, and equations of motion*

$$\dot{\hat{\eta}} = f_1\left(\hat{\xi}\right) + \mathbf{w}_c, \tag{1}$$

$$\dot{\hat{\xi}} = f_2\left(\hat{\xi}, \hat{\varphi}\right) + g\left(\hat{\xi}, \hat{\varphi}\right)\hat{\mathbf{u}}_c, \tag{2}$$

$$\dot{\hat{\varphi}} = \hat{u}_3, \tag{3}$$

where $\hat{\mathbf{u}}_c$ is an intermediate variable and defined by $\hat{\mathbf{u}}_c \triangleq (\hat{u}_1, \hat{u}_2, \sigma(\hat{\varphi})) \in D_{u_c} \subset \mathbb{R}^3$, $\sigma : D_\varphi \mapsto \mathbb{R}$ is a smooth vector field and locally Lipschitz in $\hat{\varphi}$, $f_1 : D_\xi \mapsto \mathbb{R}^3$, $f_2 : D_\xi \times D_\varphi \mapsto \mathbb{R}^3$, and $g : D_\xi \times D_\varphi \mapsto \mathbb{R}^{3\times 3}$ are smooth vector fields and locally Lipschitz in $\left(\hat{\xi}^T, \hat{\varphi}\right)^T$, and $\mathbf{w}_c$ denotes bounded estimated disturbances of the system. Assume that $L_g f_1 \triangleq \frac{\partial f_1}{\partial \mathbf{x}} g$ [13] and $\nabla_\sigma \triangleq \frac{\partial \sigma}{\partial \mathbf{x}}$ are invertible for $\hat{\xi}$ on D_ξ and $\hat{\varphi}$ on D_φ, $\eta^d(t)$ is C^2 with bounded time-derivatives. Define the tracking error as $\mathbf{e}_{\hat{\eta}} \triangleq \hat{\eta} - \eta^d$, and following variables

$$\begin{aligned}\hat{\alpha} \triangleq\ & (L_g f_1)^{-1}\left(\ddot{\eta}^d - L_{f_2} f_1 - (k_1 + k_2)\,\dot{\mathbf{e}}_{\hat{\eta}}\right.\\ & \left. - (1 + k_1 k_2)\,\mathbf{e}_{\hat{\eta}}\right),\end{aligned} \tag{4}$$

$$z_{\hat{\varphi}} \triangleq \sigma(\hat{\varphi}) - \mathbf{I}_3^T \hat{\alpha}, \tag{5}$$

where $\mathbf{I}_3 \triangleq (0, 0, 1)^T$ and k_1, k_2 and k_3 are positive control gains. If the system inputs are selected as

$$\begin{pmatrix} \hat{u}_1 \\ \hat{u}_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}\hat{\alpha}, \tag{6}$$

$$\hat{u}_3 = \nabla_\sigma^{-1}\left(\mathbf{I}_3^T \dot{\hat{\alpha}} - \left(\dot{\mathbf{e}}_{\hat{\eta}} + k_1 \mathbf{e}_{\hat{\eta}}\right)^T L_g f_1 \mathbf{I}_3 - k_3 z_{\hat{\varphi}}\right), \tag{7}$$

the tracking error $\mathbf{e}_{\hat{\eta}}$ exponentially converges to the origin with bounded $\hat{\xi}$ and $\hat{\varphi}$.

Proof: *Error dynamics and convergence:*

The error dynamics can be calculated as

$$\dot{\mathbf{e}}_{\hat{\eta}} = \dot{\hat{\eta}} - \dot{\eta}^d = f_1 + \mathbf{w}_c - \dot{\eta}^d.$$

Define a Lyapunov function candidate $V_1 \triangleq \frac{1}{2}\left\|\mathbf{e}_{\hat{\eta}}\right\|^2$, which has the time derivative

$$\dot{V}_1 = \mathbf{e}_{\hat{\eta}}^T \dot{\mathbf{e}}_{\hat{\eta}} = \mathbf{e}_{\hat{\eta}}^T\left(f_1 - \dot{\eta}^d + \mathbf{w}_c\right). \tag{8}$$

At this stage of the development, we consider f_1 as a virtual control, where $\dot{V}_1$ can be made negative definite by setting f_1 equals to $\dot{\eta}^d - k_1 \mathbf{e}_{\hat{\eta}} - \mathbf{w}_c$. Thus, we can introduce the error variable

$$\mathbf{z} \triangleq \dot{\eta}^d - k_1 \mathbf{e}_{\hat{\eta}} - \mathbf{w}_c - f_1. \tag{9}$$

Backstepping for $\mathbf{z}$:

Adding and subtracting $\mathbf{z}$ in Eq. (8) gives

$$\dot{V}_1 = -k_1\left\|\mathbf{e}_{\hat{\eta}}\right\|^2 - \mathbf{e}_{\hat{\eta}}^T \mathbf{z}.$$

Consider an augmented Lyapunov function candidate

$$V_2 \triangleq V_1 + \frac{1}{2}\left\|\mathbf{z}\right\|^2,$$

which has the time derivative

$$\begin{aligned}\dot{V}_2 =\ & -k_1\left\|\mathbf{e}_{\hat{\eta}}\right\|^2\\ & + \mathbf{z}^T\left(-\mathbf{e}_{\hat{\eta}} + \ddot{\eta}^d - k_1 \dot{\mathbf{e}}_{\hat{\eta}} - L_{f_2} f_1 - L_g f_1 \hat{\mathbf{u}}_c\right).\end{aligned}$$

If $\hat{\mathbf{u}}_c$ is selected as

$$\hat{\mathbf{u}}_c = \hat{\alpha} + \mathbf{I}_3 z_{\hat{\varphi}}, \tag{10}$$

the time derivative of V_2 becomes

$$\dot{V}_2 = -k_1 \left\|\mathbf{e}_{\hat{\eta}}\right\|^2 - k_2 \left\|\mathbf{z}\right\|^2 - \mathbf{z}^T L_g f_1 \mathbf{I}_3 z_{\hat{\varphi}}.$$

Backstepping for $z_{\hat{\varphi}}$:

Consider an augmented Lyapunov function candidate

$$V_3 \triangleq \frac{1}{2}\left\|\mathbf{e}_{\hat{\eta}}\right\|^2 + \frac{1}{2}\left\|\mathbf{z}\right\|^2 + \frac{1}{2}z_{\hat{\varphi}}^2, \tag{11}$$

which has the time derivative

$$\begin{aligned}\dot{V}_3 \;&=\; -k_1 \left\|\mathbf{e}_{\hat{\eta}}\right\|^2 - k_2 \left\|\mathbf{z}\right\|^2 \\ &\quad + z_{\hat{\varphi}}\left(-\mathbf{z}^T L_g f_1 \mathbf{I}_3 + \nabla_\sigma \hat{u}_3 - \mathbf{I}_3^T \dot{\alpha}\right).\end{aligned}$$

If $\hat{u}_3$ is given by Eq. (7), we can obtain

$$\dot{V}_3 = -k_1 \left\|\mathbf{e}_{\hat{\eta}}\right\|^2 - k_2 \left\|\mathbf{z}\right\|^2 - k_3 z_{\hat{\varphi}}^2. \tag{12}$$

Letting $0 < \mu \leq \min\{k_1,\ k_2,\ k_3\}$, we have

$$\dot{V}_3 \leq -2\mu V_3.$$

According to the Lyapunov stability theorem [13] it can be concluded that $\mathbf{e}_{\hat{\eta}}$, $\mathbf{z}$, and z_φ exponentially converge to the origin.

Boundedness of $\hat{\xi}$ and $\hat{\varphi}$:

Because $\mathbf{e}_{\hat{\eta}}$ and $\mathbf{z}$ exponentially converge to the origin, and $\dot{\eta}^d$ and $\mathbf{w}_c$ are bounded, we can obtain that $f_1\left(\hat{\xi}\right)$ is bounded from Eq. (9). Since f_1 is locally Lipschitz in $\hat{\xi}$, we can conclude that $\hat{\xi}$ is bounded [13], which implies that $\dot{\hat{\xi}}$ is bounded. From Eq. (2) and the assumptions that f_2 and σ are locally Lipschitz in $\left(\hat{\xi}^T, \hat{\varphi}\right)^T$, we have $\hat{\mathbf{u}}_c$ is bounded which implies that $\hat{\varphi}$ is bounded.

□

B. Augmented Trajectory Tracking Control Law Design with an Autopilot in the Loop

The control strategy, shown in Fig. 1, is developed in this section by augmenting the design model and the controller developed in Lemma 1. A combined system using specific equations of motion is shown in Fig. 2. The UAV dynamics are selected as the same as the design model except that the unknown disturbance is modeled as a bounded noise. The commands sent to the autopilot are selected as $\hat{y} = H\left(\hat{\xi}, \hat{\varphi}\right) = \hat{\xi}$ in this section.

Assumption 1: The autopilot is capable of guaranteeing that ξ exponentially converges to $\hat{\xi}$.

Theorem 1: *Defining states $\hat{\eta}, \eta \in D_\eta \subset \mathbb{R}^3$, $\hat{\xi}, \xi \in D_\xi \subset \mathbb{R}^3$, consider the nonlinear systems described in Fig. 2 with definitions and assumptions of the states and functions as described in Lemma 1, where the inputs of the modified design model are $\hat{\mathbf{u}}' \triangleq \left(\hat{u}_1', \hat{u}_2', \hat{u}_3'\right)^T \in D_u \subset \mathbb{R}^3$, the inputs of the system dynamics are $\mathbf{u} \triangleq \left(u_1, u_2, u_3\right)^T \in D_u \subset \mathbb{R}^3$, and the intermediate inputs are defined as $\hat{\mathbf{u}}_c' \triangleq \left(\hat{u}_1', \hat{u}_2', \sigma\left(\hat{\varphi}\right)\right) \in D_{u_c} \subset \mathbb{R}^3$ and $\mathbf{u}_c \triangleq \left(u_1, u_2, \sigma\left(\varphi\right)\right) \in D_{u_c} \subset \mathbb{R}^3$, $H : D_\eta \times D_\xi \times D_\varphi \mapsto \mathbb{R}^3$ is a smooth vector field and locally Lipschitz in $\left(\hat{\eta}^T, \hat{\xi}^T, \hat{\varphi}\right)^T$, $\mathbf{N}\left(t\right) \in \mathbb{R}^3$ is a time-varying unknown noise with known upper limit $\bar{N} \geq 0$ such that $\left\|\mathbf{N}\right\| \leq \bar{N}$. Suppose that $\eta^d(t)$ is C^2 with bounded time-derivatives, ξ and $\hat{\xi}$ satisfy Assumption 1. If ℓ is a positive constant, δ is selected as*

$$\delta\left(t\right) \triangleq \ell\left(\eta - \hat{\eta}\right),$$

$\hat{\mathbf{u}}$ is selected as in Eqs. (6) and (7), and ν is selected as

$$\nu\left(t\right) \triangleq -\left(L_g f_1\left(\hat{\xi}\right)\right)^{-1}\dot{\delta}, \tag{13}$$

then the tracking error $\mathbf{e}_{\hat{\eta}} = \hat{\eta} - \eta^d$ exponentially converges to the origin, and the tracking error $\tilde{e}_\eta \triangleq \eta - \hat{\eta}$ exponentially converge to an ultimate bound $\bar{N}/\ell$ with bounded $\hat{\xi}$ and ξ.

Proof:

Exponential convergence of $\tilde{\mathbf{e}}_\eta$:

The time derivative of $\tilde{e}_\eta$ can be calculated by

$$\dot{\tilde{\mathbf{e}}}_\eta = \dot{\eta} - \dot{\hat{\eta}} = f_1\left(\xi\right) - f_1\left(\hat{\xi}\right) - \ell\tilde{\mathbf{e}}_\eta + \mathbf{N}.$$

Define a Lyapunov function candidate

$$V_4 \triangleq \frac{1}{2}\left\|\tilde{\mathbf{e}}_\eta\right\|^2, \tag{14}$$

which has the time derivative

$$\begin{aligned}\dot{V}_4 \;&=\; \tilde{\mathbf{e}}_\eta^T\left(f_1\left(\xi\right) - f_1\left(\hat{\xi}\right) - \ell\tilde{\mathbf{e}}_\eta\right) + \tilde{\mathbf{e}}_\eta^T \mathbf{N} \\ &\leq\; -\ell\left\|\tilde{\mathbf{e}}_\eta\right\|^2 + \left\|\tilde{\mathbf{e}}_\eta\right\|\left(\left\|\mathbf{N}\right\| + \left\|f_1\left(\xi\right) - f_1\left(\hat{\xi}\right)\right\|\right).\end{aligned}$$

By using the Young's inequality, it can be concluded that for any $\hat{\beta} > 0$

$$\begin{aligned}\dot{V}_4 \;&\leq\; -\left(\ell - \frac{\hat{\beta}}{2}\right)\left\|\tilde{\mathbf{e}}_\eta\right\|^2 \\ &\quad + \frac{1}{2\hat{\beta}}\left(\left\|\mathbf{N}\right\| + \left\|f_1\left(\xi\right) - f_1\left(\hat{\xi}\right)\right\|\right)^2.\end{aligned}$$

Suppose we choose $0 < \hat{\beta} < 2\ell$ so that the term $\ell - \frac{\hat{\beta}}{2}$ is positive. Then, if Assumption 1 holds, we can conclude that there is a sufficiently small positive constant $\hat{\lambda}$ satisfying

$$0 < \frac{\hat{\lambda}}{2} \leq \ell - \frac{\hat{\beta}}{2}, \tag{15}$$

such that

$$\dot{V}_4 \leq -\hat{\lambda}V_4 + \frac{1}{2\hat{\beta}}\bar{N}^2.$$

Therefore, it can be obtained that

$$V_4\left(t\right) \leq e^{-\hat{\lambda}t}V_4\left(0\right) + \frac{1}{2\hat{\lambda}\hat{\beta}}\bar{N}^2,\ t \geq 0.$$

Then V_4 converges to a ball of radius $\bar{N}^2/\left(2\hat{\lambda}\hat{\beta}\right)$ and $\tilde{\mathbf{e}}_\eta$ exponentially converges to a ball of radius $\bar{N}/\sqrt{\hat{\lambda}\hat{\beta}}$ because of Eq. (14).

Letting $\hat{\beta} = b\ell$, where $0 < b < 2$, Eq. (15) becomes

$$0 < \hat{\lambda} \leq \left(2 - b\right)\ell.$$

Then we have

$$0 < \hat{\lambda}\hat{\beta} \leq b\left(2 - b\right)\ell^2.$$

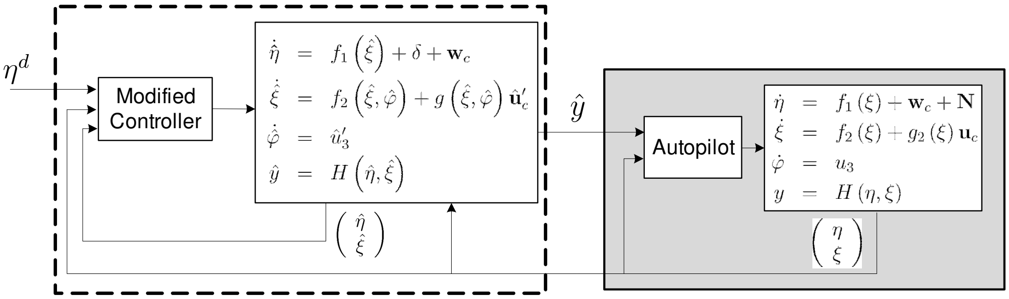

η^d
Modified Controller
$\dot{\hat{\eta}} = f_1(\hat{\xi}) + \delta + \mathbf{w}_c$
$\dot{\hat{\xi}} = f_2(\hat{\xi},\hat{\varphi}) + g(\hat{\xi},\hat{\varphi})\hat{\mathbf{u}}'_c$
$\dot{\hat{\varphi}} = \hat{u}'_3$
$\hat{y} = H(\hat{\eta},\hat{\xi})$
$\begin{pmatrix}\hat{\eta}\\ \hat{\xi}\end{pmatrix}$
$\hat{y}$
Autopilot
$\dot{\eta} = f_1(\xi) + \mathbf{w}_c + \mathbf{N}$
$\dot{\xi} = f_2(\xi) + g_2(\xi)\mathbf{u}_c$
$\dot{\varphi} = u_3$
$y = H(\eta,\xi)$
$\begin{pmatrix}\eta\\ \xi\end{pmatrix}$

Fig. 2. The structure of a specific closed-loop control for trajectory tracking with the autopilot in the loop. The commands sent to the autopilot are selected as $\hat{y}$. $\hat{\mathbf{u}}'_c \triangleq \left(\hat{u}'_1, \hat{u}'_2, \sigma(\hat{\varphi})\right)$ and $\mathbf{u}_c \triangleq (u_1, u_2, \sigma(\varphi))$.

For any $0 < b < 2$, we have

$$b(2-b)\,\ell^2 \le \ell^2,$$

where the equal sign occurs when $b = 1$. Therefore, we can obtain

$$\frac{\bar{N}}{\sqrt{\hat{\lambda}\hat{\beta}}} \ge \frac{\bar{N}}{\ell}.$$

Convergence of $\hat{\eta} \to \eta^d$:

The proof of this part is similar to Lemma 1.

Boundedness of ξ and $\hat{\xi}$.

From Lemma 1, it can be concluded that $\hat{\xi}$ is bounded. Because ξ exponentially converges to $\hat{\xi}$, so ξ is also bounded. □

III. Software-In-the-Loop Simulation

To validate the control strategy developed in Section II-B, we first perform a MATLAB/Simulink simulation and the result shows that the tracking error quickly converges to and remains in its ultimate bound. In addition, we also conduct a software-in-the-loop (SIL) simulation in which the autopilot and UAV dynamics are emulated by a Virtual Cockpit used as the ground station in a flight test. Due to the limitation of the space of this paper, we only present the SIL simulation result in this section.

A. SIL Simulation Environment

The desired/reference trajectory, η^d, is selected as an inclined circular orbit. Let R be the desired orbit radius in the horizontal plane, V_g^d be the desired constant ground speed, h_0 be the desired average altitude and h_c be the desired magnitude of the altitude oscillation. The reference trajectory in a North-East-Down (NED) frame is given by

$$\eta^d(t) = \begin{pmatrix} R\sin(\omega_0 t) \\ R\cos(\omega_0 t) \\ -h_0 + h_c \sin(\omega_0 t) \end{pmatrix}, \qquad (16)$$

where $\omega_0 = V_g^d / R$. We select R as 200 m, V_g^d as 15 m/s, h_0 as 100 m and h_c as 5 m.

The design model used to develop a trajectory-tracking controller is selected as a dynamic model of the vehicle. Letting $(p_n, p_e, p_d)^T$ be the position of a UAV in the inertial NED frame, V_a be the airspeed, ψ be the heading angle, γ_a be the air mass referenced flight path angle, which is defined as the angle from the inertial North-East plane to the velocity vector of the aircraft relative to the air mass, m be the mass of the UAV, g_c be the gravitational constant at Earth sea level, L and D be the aerodynamic lift and drag forces, respectively, ϕ be the roll angle, $(w_n, w_e, w_d)^T$ be the wind estimation in north, east and down directions, respectively, the control inputs of the system be the thrust u_τ, the load factor $u_n \triangleq \frac{L}{mg_c}$ and the roll angular rate u_ϕ, the equations of motion of the UAV in a coordinated flight can be written as [9]

$$\begin{aligned}
\dot{p}_n &= V_a \cos\psi \cos\gamma_a + w_n, && (17)\\
\dot{p}_e &= V_a \sin\psi \cos\gamma_a + w_e, && (18)\\
\dot{p}_d &= -V_a \sin\gamma_a + w_d, && (19)\\
\dot{V}_a &= \frac{u_\tau - D}{m} - g_c \sin\gamma_a, && (20)\\
\dot{\psi} &= \frac{L\sin\phi}{mV_a\cos\gamma_a}, && (21)\\
\dot{\gamma}_a &= \frac{g_c}{V_a}\left(u_n\cos\phi - \cos\gamma_a\right), && (22)\\
\dot{\phi} &= u_\phi. && (23)
\end{aligned}$$

To match the UAV dynamics to Eqs. (1) through (3), Equations (20) through (22) are rearranged as

$$\begin{pmatrix} \dot{V}_a \\ \dot{\gamma}_a \\ \dot{\psi} \end{pmatrix} = f_2 + g\hat{\mathbf{u}}_c,$$

where

$$f_2(\cdot) = \begin{pmatrix} -\frac{D}{m} - g_c\sin\gamma_a \\ -\frac{g_c}{V_a}\cos\gamma_a \\ 0 \end{pmatrix},$$

$$g(\cdot) = \begin{pmatrix} \frac{1}{m} & 0 & 0 \\ 0 & \frac{g_c}{V_a}\cos\phi & 0 \\ 0 & 0 & \frac{L}{mV_a\cos\gamma_a} \end{pmatrix},$$

$$\hat{\mathbf{u}}_c = \begin{pmatrix} u_\tau \\ u_n \\ \sigma(\phi) \end{pmatrix},\ \sigma(\cdot){=}\sin\phi,\ \text{and}\ \hat{\mathbf{u}} = \begin{pmatrix} u_\tau \\ u_n \\ u_\phi \end{pmatrix}.$$

The experimental platform consists of a MATLAB/Simulink program which implements the design model and developed controller, a control station, Virtual Cockpit (VC), which emulates the hardware code in the simulation, and a MEX-function which is an I/O program transferring data between the MATLAB and VC. The commands sent to the applied autopilot (emulated in VC) are the airspeed V_a, the course angle χ and the altitude $-p_d$. It should be noted that the commands $\hat{y} = \left(\hat{V}_a, \hat{\chi}, -\hat{p}_d\right)$ sent to the aircraft are actually not $\hat{\xi} = \left(\hat{V}_a, \hat{\psi}, -\hat{p}_d\right)$. However, the course angle χ can be calculated by

$$\chi = \tan^{-1} \frac{V_a \sin\psi \cos\gamma_a + w_e}{V_a \cos\psi \cos\gamma_a + w_n}.$$

It is assumed that the wind estimation is accurate enough to guarantee that $\chi \to \hat{\chi}$ implies $\hat{\psi} \to \psi$, which satisfies Assumption 1.

The MEX-function used in the simulation offers a communication rate between MATLAB and VC approximate 5 Hz, so the step size in MATLAB/Simulink is selected as 0.2 s. Because of the communication lag between the MATLAB and VC, the control gain ℓ is tuned and selected as 0.1 to compromise between the tracking errors of the actual and simulated UAVs, $\tilde{\mathbf{e}}_\eta$, and the tracking error of the simulated UAV and the desired trajectory, $\mathbf{e}_{\hat{\eta}}$. A constant wind speed of 3 m/s from the west is added in VC. Control gains are tuned and selected as $k_1 = 0.2$, $k_2 = 1$, and $k_3 = 10$. Assuming that the design model is able to represent the system dynamics with high fidelity, and selecting $\bar{N}$ as 1 m/s, the ultimate bound can be calculated as 10 m.

B. Simulation Results

Figure 3 shows the SIL simulation results of tracking an inclined circular orbit. Figure 3 (a) presents the top-down view of the system trajectories. It can be seen that the actual trajectory (line with “Asterisk”) started at approximately $(-60, -40)$ m and quickly converged to the desired orbit (solid line). The UAV trajectory obtained from the design model (“diamond” dots) also quickly converges to the desired orbit. Figure 3 (b) shows the side view of system trajectories. It can be seen that the actual UAV altitude is able to essentially track the desired orbit. Figure 3 (c) shows the tracking errors in north, east and altitude directions, respectively. It can be seen that tracking errors converge to the neighborhood of the origin and oscillate within ±10 m in three directions, respectively. Figure 3 (d) illustrates the distance between the actual UAV position and the desired trajectory. It can be seen that the distance error converges to and remains within approximately 15 m, which is slightly greater than the calculated ultimate bound (10 m). Figure 3 (e) shows the commanded values of the autopilot inputs and their actual values of the UAV. It can be seen that the airspeed and course angle trackers work precisely while the altitude tracker has an approximate 10 s lag, which also contributes to the tracking error. Figure 3 (f) shows the estimated wind speed obtained from VC.

It should be noted that the relatively large tracking error, shown in Fig. 3 (d), is attributed to the inherent communication delay between the MATLAB and VC. An airplane with an average airspeed of 15 m/s and a relatively low communication frequency (5 Hz) cannot efficiently correct its deviation to the desired trajectory. In other words, Assumption 1 does not hold well in the SIL simulation. Therefore, a larger communication band width between the MATLAB and VC or an onboard implementation is demanded to obtain a smaller tracking error.

IV. Conclusion

This paper presented a trajectory tracking strategy for UAVs with an autopilot in the closed-loop system. A Lyapunov-based backstepping technique was used to derive a trajectory-tracking controller based on a design model with generalized equations of motion. To incorporate the autopilot in the closed-loop system, the developed trajectory-tracking controller and the design model were augmented. The updated controller was shown to be capable of driving the vehicle to exponentially converge to and follow reference trajectories by keeping the other higher order and attitude states bounded. The result of a software-in-the-loop simulation showed the effectiveness of the developed trajectory tracking control strategy.

References

[1] A. W. Divelbiss and J. T. Wen, “Trajectory tracking control of a car-trailer system,” *IEEE Transactions on Control Systems Technology*, vol. 5, no. 3, pp. 269–278, May 1997.

[2] L. Sun and R. W. Beard, “Towed-body trajectory tracking in aerial recovery of micro air vehicle in the presence of wind,” in *American Control Conference*, San Francisco, CA, USA, 2011, pp. 3209–3214.

[3] I. Kaminer, A. Pascoal, E. Hallberg, and C. Silvestre, “Trajectory tracking for autonomous vehicles: An integrated approach to guidance and control,” *AIAA Journal of Guidance, Control, and Dynamics*, vol. 21, no. 1, pp. 29–38, 1998.

[4] P. Encarnaçào and A. Pascoal, “Combined trajectory tracking and path following for marine craft.” Orlando, FL: the IEEE Conference on Decision and Control, 2001, pp. 964–969.

[5] A. P. Aguiar, L. Cremean, and J. P. Hespanha, “Position tracking for a nonlinear underactuated hovercraft: controller design and experimental results,” in *Proc. 42nd IEEE Conference on Decision and Control*, vol. 4, 9–12 Dec. 2003, pp. 3858–3863.

[6] A. P. Aguiar and J. P. Hespanha, “Position tracking of underactuated vehicles,” in *Proc. American Control Conference the 2003*, vol. 3, 4–6 June 2003, pp. 1988–1993.

[7] ——, “Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty,” *IEEE Transactions on Automatic Control*, vol. 52, no. 8, pp. 1362–1379, August 2007.

[8] D. R. Nelson, D. B. Barber, T. W. McLain, and R. W. Beard, “Vector field path following for miniature air vehicles,” *IEEE Transactions on Robotics*, vol. 37, pp. 519–529, June 2007.

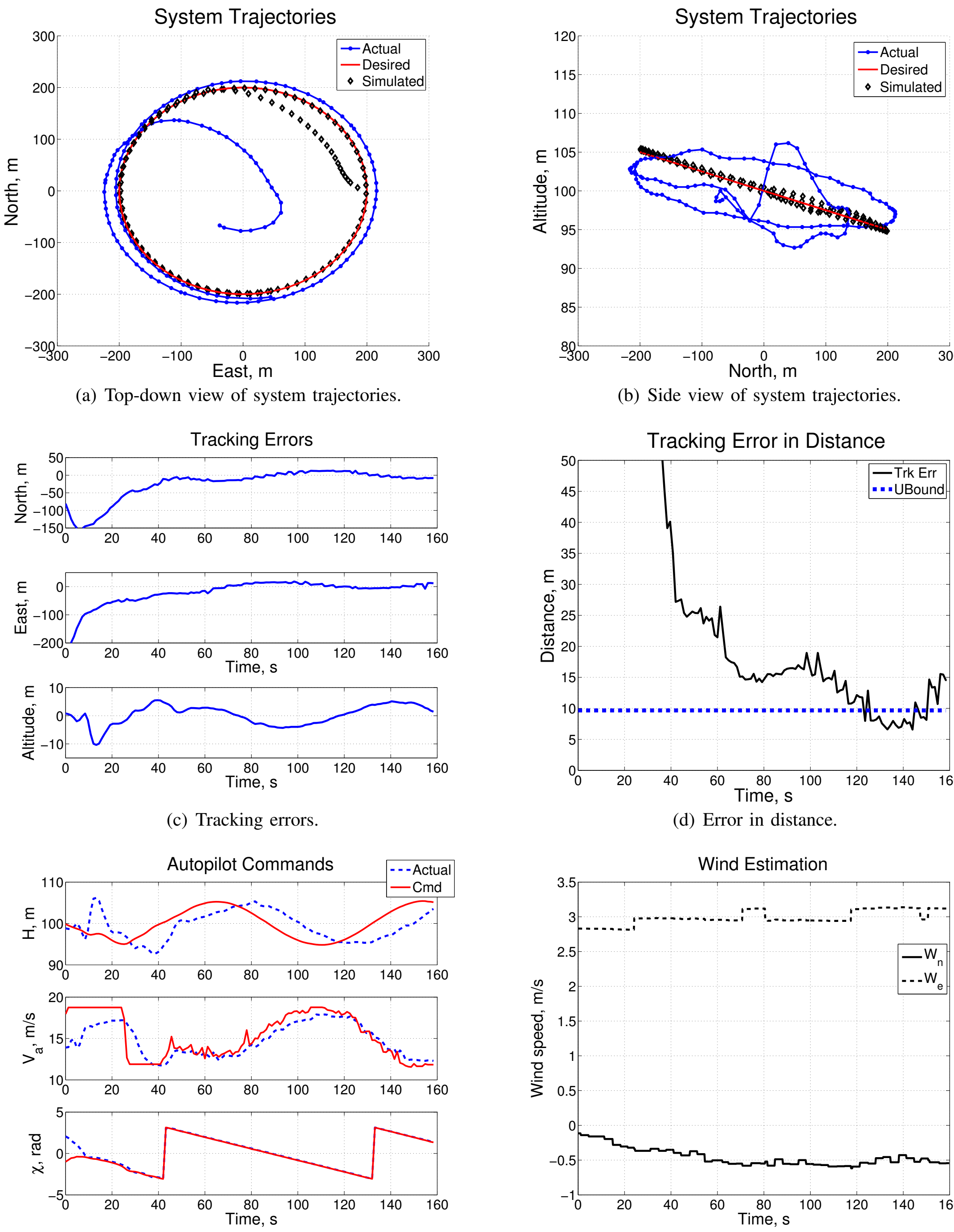
[9] R. W. Beard and T. W. McLain, *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, 2012.

[10] H. Chao, Y. Cao, and Y. Chen, “Autopilots for small unmanned aerial vehicles: A survey,” *International Journal of Control, Automation, and Systems*, vol. 8, no. 1, pp. 36–44, 2010.

[11] I. Kaminer, A. Pascoal, E. Xargay, N. Hovakimyan, C. Cao, and V. Dobrokhodov, “Path following for unmanned aerial vehicles using l1adaptive augmentation of commercial autopilots,” *AIAA Journal of Guidance, Control, and Dynamics*, vol. 33, no. 2, pp. 550–564, 2010.

[12] V. Dobrokhodov, I. Kaminer, I. Kitsios, E. Xargay, N. Hovakimyan, C. Cao, I. M. Gregory, and L. Valavani, “Experimental validation of L1 adaptive control: The rohrs counterexample in flight,” *AIAA Journal of Guidance, Control, and Dynamics*, vol. 34, no. 5, pp. 1311–1328, 2011.

[13] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2002.



(a) Top-down view of system trajectories.

(b) Side view of system trajectories.

(c) Tracking errors.

(d) Error in distance.

(e) Commands and actual autopilot inputs.

(f) Wind estimation.

Fig. 3. Software in the loop simulation results of tracking a circular orbit.