Disturbance Observer-Based Integral Backstepping Control for UAVs

Amir Moeini * **Zhijun Xue** ** **Muhammad Awais Rafique** *
Alan F. Lynch * **Qing Zhao** *

* *Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, T6G 2V4, Canada (e-mail: {moeini,mrafique,alan.lynch,qingz}@ualberta.ca).*
** *Huazhong University of Science and Technology, Wuhan, China (e-mail: xzj0319@hust.edu.cn)*

Abstract: In this paper, we propose a disturbance observer-based integral backstepping nonlinear control for motion control of a multirotor unmanned aerial vehicle (UAV). The closed-loop is proven globally asymptotically stable for the complete UAV dynamics and constant disturbances. We present a Software-in-the-loop simulation which validates the robust performance of the proposed control on the open-source PX4 autopilot. The simulation justifies the integral augmentation and disturbance observer included.

Keywords: Autonomous Mobile Robots, Nonlinear Control, Disturbance Rejection, Observers, Backstepping Control, Motion Control

1. INTRODUCTION

Multirotor Unmanned Aerial Vehicles (UAVs) are important in academic research and a number of industrial applications. Although the topic of UAV motion control has been well-studied (Hua et al., 2013), improving the robustness and performance of motion control remains a core research topic for emerging applications. For example, robust accurate motion control is important in enabling aerial manipulation where vehicles carry a manipulator arm and often directly interact with their environment (Khamseh et al., 2018). In such cases, motion control of the UAV determines capability of the manipulation task and the vehicle is subject to external forces from the environment and changing aerodynamic conditions when operating near and in contact with objects.

Backstepping motion control designs have received significant attention in the literature. Most of these designs and their variations compensate for some nonlinearity in the rigid body dynamics. Examples include an inner-outer loop approach (Hamel et al., 2002) and a centralized design based on the entire system dynamics (Castillo et al., 2005). A practical shortcoming of classical backstepping results is their sensitivity to model uncertainty and disturbances. These effects are often ignored during the design. One way to improve the robustness of backstepping is to combine it with a disturbance observer which provides an estimate of the disturbance using the system model and measurements. This estimate is used in the control law for disturbance rejection or attenuation. For example, in (He et al., 2014) a disturbance observer is combined with backstepping and implemented on a traditional helicopter UAV. The disturbance observer is designed independent of the feedback control and a low-pass filter is used to estimate the angular acceleration to avoid complex expressions in the backstepping controller. The design leads to ultimate boundedness of the tracking error. Work in (Shao et al., 2018) employs an inner-outer loop structure and combines backstepping with an extended state observer to achieve trajectory tracking. The observer is designed to estimate the external disturbance and unmeasured linear and angular velocities. Simplified rotational dynamics is assumed in the inner loop controller design. The trajectory tracking error is proven to be ultimately bounded in the presence of time-varying disturbances.

Another way to improve the robustness of backstepping is to add an integral term to the feedback control. As in PID controllers, integral terms can improve the robustness against model uncertainty and also reduce the steady state tracking error (Colmenares-Vazquez et al., 2015). Work in (Poultney et al., 2019) proposes an integral backstepping control for the system dynamics. The backstepping procedure starts with the integral of position error. Proof for constant disturbance rejection is presented, but the considered disturbance does not have a physical meaning since it appears in the tracking error dynamics rather than the UAV dynamics itself. Also, the method assumes the norm of the disturbance is bounded by a linear function of the tracking error coordinates. This is a restrictive assumption which is not satisfied for the case of constant disturbances. Work in (Jasim and Gu, 2015) considers an inner-outer loop control structure where the inner loop reference is generated by an integral backstepping controller for the translational subsystem. Disturbance or model uncertainties are not explicitly considered in the design. Simulations demonstrate robustness of the design due to the integral action added.

This paper proposes a nonlinear control which combines a disturbance observer with integral augmentation to improve the robust performance of a backstepping control. A main contribution of the work is to provide an analysis of

the closed-loop asymptotic stability for the full nonlinear rigid body UAV dynamics for the constant disturbance case. Another important contribution of the paper is the SITL (Software in the loop) simulation code and results it presents. SITL simulation is an important step in the development of flight testing a control law. However, most available simulation results are performed in Matlab which are not necessarily representative of an on-board implementation. The PX4 controller module developed should be useful to researchers in the field of applied nonlinear multirotor control. The simulation is developed using the open-source PX4 autopilot project Meier (2018) and compares a traditional backstepping design with integral backstepping, disturbance observer-based backstepping, and disturbance observer-based integral backstepping. We demonstrate the improved output tracking performance of the disturbance observer-based integral backstepping control in the presence of wind force disturbance.

2. QUADROTOR DYNAMICS

In this section we review a traditional rigid body quadrotor model and establish notation for control design. The dynamics of quadrotor has been presented in many works such as (Castillo et al., 2005) and (Bouabdallah, 2007). We consider a traditional quadrotor UAV as shown in Figure 1. Two reference frames are needed: a fixed inertial navigation frame $\mathcal{N}$ with orthonormal basis $\{n_1, n_2, n_3\}$ and a body frame $\mathcal{B}$ whose origin is at the vehicle's center of mass (CoM) and with orthonormal basis $\{b_1, b_2, b_3\}$. We define b_1 to point in the forward direction of vehicle, b_2 pointing right, and b_3, n_3 pointing down. The configuration of the quadrotor belongs to the special Euclidean group SE(3), and includes the position $p \in \mathbb{R}^3$ of the origin of $\mathcal{B}$ relative to $\mathcal{N}$, and the orientation $R \in \mathrm{SO}(3)$ of $\mathcal{B}$ with respect to $\mathcal{N}$. We assume each propeller generates thrust in the $-b_3$ direction and denote the total thrust due to all propellers by the scalar input $u > 0$, i.e., the thrust vector is $-ub_3$. Controlling individual propeller speeds creates an input torque denoted $\tau \in \mathbb{R}^3$ which is expressed in $\mathcal{B}$. To ease presentation of the control design we take torque τ and thrust u as system inputs. However, in practice the UAV is actuated by PWM signals to the Electronic Speed Controllers (ESCs). The dynamics are given by

$$\dot{p} = v \tag{1a}$$
$$m\dot{v} = mgn_3 - uRn_3 + d_f \tag{1b}$$
$$\dot{R} = RS(\omega) \tag{1c}$$
$$J\dot{\omega} = -\omega \times J\omega + \tau \tag{1d}$$

where $v \in \mathbb{R}^3$ is linear velocity expressed in $\mathcal{N}$, $\omega \in \mathbb{R}^3$ is angular velocity expressed in $\mathcal{B}$, m is mass, J is inertia, g is the gravity constant, $n_3 = [0, 0, 1]^\top$, and the skew operator $S(\cdot) : \mathbb{R}^3 \to so(3)$ is

$$S(x) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}, \quad \text{where } x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}.$$

The disturbance force $d_f \in \mathbb{R}^3$ is unknown and used to model external forces such as wind gusts or other model uncertainty. We choose not to include a torque disturbance in (1d) to simplify presentation, and the proposed approach can be extended to reject constant torque disturbances.

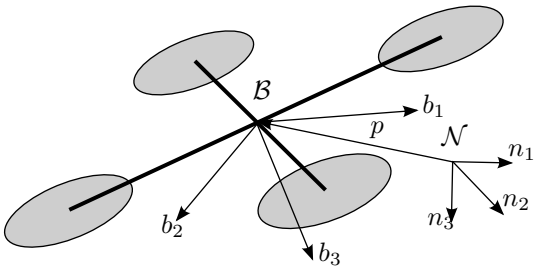

Fig. 1. Diagram of a quadrotor showing navigation frame $\mathcal{N}$ and body frame $\mathcal{B}$.

3. DISTURBANCE OBSERVER-BASED CONTROLLER DESIGN

In this section we solve an output tracking for UAV position. That is, given a smooth bounded desired trajectory for position, denoted by $p_d \in \mathbb{R}^3$, we derive a dynamic state feedback control for inputs u and τ to ensure asymptotic convergence of the tracking errors $p - p_d$ in the presence of constant bounded disturbances d_f. Our control regulates the yaw degree of freedom (DoF) to a constant trajectory. This is done for simplicity of presentation as including the additional problem of tracking general yaw trajectories would distract from the main contribution of the paper. We assume the full state is measured which is realistic when flying indoors with a motion capture system or outdoors with GPS. The design adopts a backstepping approach inspired by (Cabecinhas et al., 2014) and the references within. Unlike the approach in (Cabecinhas et al., 2014) which relies on high-dimensional parameter update laws to estimate d_f, the proposed method uses a 3-dimensional disturbance observer whose error dynamics is decoupled from the rest of the closed-loop.

To estimate d_f we consider the observer

$$\hat{d}_f = z_{d_f} + k_{d_f} mv \tag{2a}$$
$$\dot{z}_{d_f} = -k_{d_f}\hat{d}_f - k_{d_f}(mgn_3 - uRn_3). \tag{2b}$$

Assuming d_f is constant ($\dot{d}_f = 0$) then (2) implies

$$\begin{aligned}\dot{\hat{d}}_f &= -k_{d_f}\hat{d}_f - k_{d_f}(mgn_3 - uRn_3) \\ &\quad + k_{d_f}(mgn_3 - uRn_3 + d_f) = k_{d_f}\tilde{d}_f\end{aligned} \tag{3}$$

and therefore $\dot{\tilde{d}}_f = -k_{d_f}\tilde{d}_f$ which is globally exponentially stable for $k_{d_f} > 0$.

The proposed method uses backstepping with integral augmentation. We start by defining the integral of position tracking error $\delta_1 = \int_{t_0}^{t}(p(\xi) - p_d(\xi))d\xi$ and considering the first Lyapunov function candidate $V_1 = \frac{1}{2}\|\delta_1\|^2$. Taking its time derivative gives $\dot{V}_1 = \delta_1^\top \dot{\delta}_1 = \delta_1^\top (p - p_d)$. If we consider p as the first virtual control and define $\alpha_1 = -k_1\delta_1 + p_d$ as its desired value we get

$$\dot{V}_1 = \delta_1^\top(\delta_2 + \alpha_1 - p_d) = -k_1\|\delta_1\|^2 + \delta_1^\top\delta_2$$

where $\delta_2 = p - \alpha_1$. Now defining the second Lyapunov function as $V_2 = V_1 + \frac{1}{2}\|\delta_2\|^2$ we get $\dot{V}_2 = -k_1\|\delta_1\|^2 + \delta_1^\top\delta_2 + \delta_2^\top\dot{\delta}_2$, where $\dot{\delta}_2 = v + k_1(p - p_d) - \dot{p}_d$. Therefore,

$$\dot{V}_2 = -k_1\|\delta_1\|^2 + \delta_1^\top\delta_2 + \delta_2^\top(v + k_1(p - p_d) - \dot{p}_d)$$

Denoting α_2 as the desired value for the second virtual control v and taking

$$\alpha_2 = -\delta_1 - k_1(p - p_d) + \dot{p}_d - k_2\delta_2$$

yields $\dot{V}_2 = -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 + \delta_2^\top \delta_3/m$ where $\delta_3 = mv - m\alpha_2$. Now selecting $V_3 = V_2 + \frac{1}{2}\|\delta_3\|^2$ we get

$$\dot{V}_3 = -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 + \delta_2^\top \delta_3/m + \delta_3^\top \dot{\delta}_3$$

where

$$\dot{\delta}_3 = mgn_3 - uRn_3 + d_f + m\dot{\delta}_1 + mk_1(v - \dot{p}_d) - m\ddot{p}_d + mk_2\dot{\delta}_2$$

Next, we consider uRn_3, which is rotor thrust expressed in $\mathcal{B}$, as a virtual control and take its desired value as

$$\alpha_3 = \delta_2/m + k_3\delta_3 + mgn_3 + \hat{d}_f + m\dot{\delta}_1 + mk_1(v - \dot{p}_d) - m\ddot{p}_d + mk_2\dot{\delta}_2$$

Defining $\delta_4 = \alpha_3 - uRn_3$ we get

$$\dot{V}_3 = -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_3\|\delta_3\|^2 + \delta_3^\top \delta_4 + \delta_3^\top \tilde{d}_f$$

Note that we have replaced d_f which appeared in $\dot{\delta}_3$ by its estimate $\hat{d}_f$ in α_3. We define $V_4 = V_3 + \frac{1}{2}\|\delta_4\|^2$ so that

$$\dot{V}_4 = -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_3\|\delta_3\|^2 + \delta_3^\top \delta_4 + \delta_4^\top \dot{\delta}_4 + \delta_3^\top \tilde{d}_f \tag{4}$$

where $\dot{\delta}_4 = \dot{\alpha}_3 - \dot{u}Rn_3 - uRS(\omega)n_3$ and $\dot{\alpha}_3$ can be expressed as

$$\begin{aligned}\dot{\alpha}_3 = {} & \dot{\delta}_2/m + k_3(mgn_3 - uRn_3 + d_f + m\dot{\delta}_1 + mk_1(v - \dot{p}_d) \\ & - m\ddot{p}_d + mk_2\dot{\delta}_2) + k_{d_f}\tilde{d}_f + m(\dot{p} - \dot{p}_d) \\ & + k_1(mgn_3 - uRn_3 + d_f - m\ddot{p}_d) - m\dddot{p}_d \\ & + k_2(mgn_3 - uRn_3 + d_f + mk_1(v - \dot{p}_d) - m\dot{v}_d)\end{aligned} \tag{5}$$

We observe some of the terms in $\dot{\alpha}_3$ are unknown due to d_f. Therefore, we estimate $\dot{\alpha}_3$ by β which is derived by replacing d_f with $\hat{d}_f$

$$\dot{\alpha}_3 = \beta + (k_3 + k_1 + k_2 + k_{d_f})\tilde{d}_f$$

and therefore

$$\dot{\delta}_4 = \beta + (k_3 + k_1 + k_2 + k_{d_f})\tilde{d}_f - \dot{u}Rn_3 - uRS(\omega)n_3 \tag{6}$$

Now we try to assign expressions for $\dot{u}$ and ω to make $\dot{V}_4$ negative definite. Since u is the actual input, we can directly assign its value. However for ω we need to consider $uRS(\omega)n_3$ as the virtual control and define its desired value. From the expression for $\dot{V}_4$ in (4) and (6), negative definiteness of $\dot{V}_4$ follows by cancelling β, the indefinite term $\delta_3^\top \delta_4$, and adding a damping term $-k_4\|\delta_4\|^2$. Hence, taking

$$\dot{u} = n_3^\top R^\top(\beta + \delta_3 + k_4\delta_4) \tag{7}$$

defining $\alpha_4 = R[I - n_3n_3^\top]R^\top(\beta + \delta_3 + k_4\delta_4)$, and $\delta_5 = \alpha_4 - uRS(\omega)n_3$, we get

$$\begin{aligned}\dot{V}_4 = {} & -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_3\|\delta_3\|^2 - k_4\|\delta_4\|^2 + \delta_4^\top \delta_5 \\ & + \delta_3^\top \tilde{d}_f + \delta_4^\top(k_3 + k_1 + k_2 + k_{d_f})\tilde{d}_f\end{aligned}$$

Now by choosing $V_5 = V_4 + \frac{1}{2}\|\delta_5\|^2$,

$$\begin{aligned}\dot{V}_5 = {} & -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_3\|\delta_3\|^2 - k_4\|\delta_4\|^2 + \delta_4^\top \delta_5 \\ & + \delta_5^\top \dot{\delta}_5 + \delta_3^\top \tilde{d}_f + \delta_4^\top(k_3 + k_1 + k_2 + k_{d_f})\tilde{d}_f\end{aligned}$$

where $\dot{\delta}_5 = \dot{\alpha}_4 - \dot{u}RS(\omega)n_3 - uRS(\omega)^2 n_3 - uRS(\dot{\omega})n_3$ and $\dot{\alpha}_4$ given by

$$\begin{aligned}\dot{\alpha}_4 = {} & RS(\omega)[I - n_3n_3^\top]R^\top(\beta + \delta_3 + k_4\delta_4) \\ & + R[I - n_3n_3^\top]S(\omega)^\top R^\top(\beta + \delta_3 + k_4\delta_4) \\ & + R[I - n_3n_3^\top]R^\top(\dot{\beta} + \dot{\delta}_3 + k_4\dot{\delta}_4).\end{aligned} \tag{8}$$

the value of $\dot{\alpha}_4$ has to be estimated due to the presence of d_f and $\tilde{d}_f$ in $\dot{\beta}$, $\dot{\delta}_3$ and $\dot{\delta}_4$. However, these variables can be estimated by replacing the disturbance estimate in their expressions. We denote these "estimated" versions $\dot{\beta}'$, $\dot{\delta}_3'$, $\dot{\delta}_4'$. The relation between actual and estimated expressions are

$$\dot{\beta} = \left(\frac{1}{m^2} + k_3(k_{d_f} + mk_1 + k_2) + (k_1 + k_2)k_{d_f}\right)\tilde{d}_f + \dot{\beta}'$$

$$\dot{\delta}_3 = \dot{\delta}_3' + \tilde{d}_f$$

$$\dot{\delta}_4 = \dot{\delta}_4' + (k_3 + k_1 + k_2 + k_{d_f})\tilde{d}_f$$

Using the estimates expressions $\dot{\beta}'$, $\dot{\delta}_3'$, $\dot{\delta}_4'$ in (8) we can obtain an estimate of $\dot{\alpha}_4$

$$\dot{\alpha}_4 = \dot{\alpha}_4' + kR[I - n_3n_3^\top]R^\top\tilde{d}_f$$

where $k = (1/m^2 + k_3k_{d_f} + k_3k_2 + m + k_1k_{d_f} + k_2k_{d_f} + mk_2k_1) + 1 + k_4(k_3 + k_1 + k_2 + k_{d_f})$. At this stage from $\dot{\delta}_5$ the control variable is $\dot{\omega}$ which is algebraically related to torque via (1d) and can therefore be considered an actual control input. Hence, we assign its expression directly. If $\dot{\omega}$ satisfies

$$\dot{u}RS(\omega)n_3 + uRS(\omega)^2 n_3 - uRS(n_3)\dot{\omega} = \dot{\alpha}_4' + \delta_4 + k_5\delta_5 \tag{9}$$

then we can write

$$\begin{aligned}\dot{V}_5 = {} & -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_3\|\delta_3\|^2 - k_4\|\delta_4\|^2 - k_5\|\delta_5\|^2 \\ & + \tilde{d}_f^\top(\delta_3 + (k_3 + k_1 + k_2 + k_{d_f})\delta_4 \\ & + kR[I - n_3n_3^\top]R^\top\delta_5)\end{aligned}$$

Solving (9) for $\dot{\omega}_1$ and $\dot{\omega}_2$ gives

$$\dot{\omega}_1 = -\frac{n_2^\top R^\top}{u}(\dot{\alpha}_4' - \dot{u}RS(\omega)n_3 - uRS(\omega)^2 n_3 + \delta_4 + k_5\delta_5) \tag{10a}$$

$$\dot{\omega}_2 = \frac{n_1^\top R^\top}{u}(\dot{\alpha}_4' - \dot{u}RS(\omega)n_3 - uRS(\omega)^2 n_3 + \delta_4 + k_5\delta_5) \tag{10b}$$

In order to obtain the expressions for the torque input τ to achieve position tracking, we take $\dot{\omega}_3 = 0$ which implies the yaw DoF is uncontrolled. Using (1d) the control torque follows from solving for τ:

$$\tau = J\dot{\omega} + \omega \times J\omega \tag{11}$$

The closed-loop asymptotic stability of the proposed backstepping controller can be proven with the following theorem.

Theorem 1. Given system (1) with constant disturbance d_f, bounded smooth reference trajectory p_d and assuming $u > 0$, the equilibrium $[\delta_1^\top, \delta_2^\top, \delta_3^\top, \delta_4^\top, \delta_5^\top, \tilde{d}_f^\top] = 0$ of the closed-loop dynamics is globally uniformly asymptotically stable by the dynamic state feedback control (2), (7) (10) and (11).

Proof. If we consider $x_1 = [\delta_1^\top, \delta_2^\top, \delta_3^\top, \delta_4^\top, \delta_5^\top]^\top$ and $x_2 = \tilde{d}_f$, the closed-loop is as a cascade system

$$\dot{x}_1 = f_1(x_1, x_2) \tag{12a}$$

$$\dot{x}_2 = f_2(x_2) \tag{12b}$$

From (Khalil, 2002, Lemma. 4.7), the origin of the cascade system is globally uniformly asymptotically stable if

- The origin of (12b) is globally uniformly asymptotically stable.
- System (12a) is input to state stable (ISS) with x_2 taken as the input.

Since $\dot{\tilde{d}}_f = -k_{d_f}\dot{\tilde{d}}_f$ the first condition is satisfied and then we only need to check (12a) is ISS with x_2 taken as the input. To show this, we apply (Khalil, 2002, Lemma. 4.6) which states (12a) is ISS if

- The unforced system $\dot{x}_1 = f_1(x_1, 0)$ is globally exponentially stable at the origin.
- System (12a) is continuously differentiable and globally Lipschitz in x_1 and x_2.

From the expression for $\dot{V}_5$ we can observe that the first condition is satisfied since by selecting $x_2 = \tilde{d}_f = 0$ we get $\dot{V}_5 = -k_1\|\delta_1\|^2 - k_2\|\delta_2\|^2 - k_3\|\delta_3\|^2 - k_4\|\delta_4\|^2 - k_5\|\delta_5\|^2$ which is globally exponentially stable.

To show the second condition we consider the expression for f in (12a). We have $\dot{x}_1 = Ax_1 + Bx_2$ where

$$A = \begin{bmatrix} -k_1 & 1 & 0 & 0 & 0 \\ -1 & -k_2 & 1/m & 0 & 0 \\ 0 & -1/m & -k_3 & 1 & 0 \\ 0 & 0 & -1 & -k_4 & 1 \\ 0 & 0 & 0 & -1 & -k_5 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 1 \\ k_{d_f} + k_1 + k_2 \\ kR(I - n_3 n_3^\top)R^\top \end{bmatrix}$$

Since f is LTI, it is continuously differentiable and globally Lipschitz in x_1 and x_2. □

An important contribution of this work is the simple decoupled structure (12) of the disturbance observer-based backstepping control. This structure simplifies the implementation of the controller allowing it to be implemented on an actual platform. The controller expressions are relatively simple compared with those in (Cabecinhas et al., 2014) where a higher-dimensional parameter update law is used. As well, the design avoids any model simplifications or other approximations which are found in recent related work (Shao et al., 2018; Dong et al., 2014; Lee et al., 2016).

4. SITL JMAVSIM SIMULATION

When developing new motion control algorithms for actual flight there are clear benefits to simulation which recreates actual flight conditions and matches on-board implementation. One approach is to perform Software-In-The-Loop (SITL) simulation running on a desktop PC where actual autopilot code is interfaced to a simulator which models the vehicle dynamics and the environment (e.g., wind disturbance forces). Since the Applied Nonlinear Control Lab (ANCL) has already invested in the PX4 autopilot (Meier, 2018; Fink, 2018; Xie, 2016), it is natural to use this project's existing SITL framework and the jMAVSim simulator. Generally speaking, SITL simulation results are not usually presented in the literature. However, they are a key step towards a safer accelerated development including improved debugging and controller tuning. A contribution of this paper is to provide self-contained source for the simulation (Moeini et al., 2019).

PX4 Controller Module `mc_dobibs_control` We implement our proposed control on the PX4 autopilot which is an active and mature open-source project intended for research and industrial applications. PX4 runs on an increasing number of hardware platforms and various robot types. The PX4 code has a modular structure where individual autopilot functionality (e.g., motion control, state estimation) is separated into self-contained modules which each run as a task on NuttX RTOS (e.g., Pixhawk) platforms or as a thread within the main PX4 process on POSIX (e.g., SITL) platforms. This modular structure has a number of advantages. For example, we can switch between different modules (e.g., control methods) without restarting the autopilot. Also, adding a new control method to the system is relatively straightforward as only a small component of the code with a well-defined interface needs to be modified. The so-called PX4 middleware supports communication between the modules, sensor device drivers, and communication outside PX4. Module communication is implemented with the micro-object request broker (uORB) which provides a publish/subscribe bus. Using uORB, publishers send messages (e.g., a UAV control input) onto a bus instead of sending the messages directly to specific subscriber modules. Also, subscribers receive messages as soon as there are updates. This publish/subscribe model is commonly used in robotics (e.g., the Robot Operating System (ROS)) to avoid locking issues. PX4 collects measurements from the jMAVSim simulator which include the IMU (gyroscope, accelerometer, magnetometer), GPS, and barometer. jMAVSim accepts the PWM signals to the ESCs from PX4. The PX4/jMAVSim communication uses the Micro Aerial Vehicle Link (MAVLink)(Meier, 2015) to communicate the simulated sensor measurements to PX4 using localhost User Datagram Protocol (UDP). PX4 Debugging is facilitated with a PXH shell. For example, you can start, stop, and query the status of modules, listen to PX4 messages, and set and query parameters.

The proposed controller is in the module `mc_dobibs_control` located in `src/modules/mc_dobibs_control`. To simplify its implementation the module uses a so-called "Block" structure in the Controller Library contained in `UAV_IBS/src/lib/controllib`. This library simplifies the process of subscription and publication. The library also provides functions for numerical integration which are used in the disturbance observer (2), integration for thrust in (7), and δ_1. The controller module also uses the PX4 Matrix Library (`https://github.com/PX4/Matrix`) to improve the readability of the control law expressions given in Section 3. Control gains are implemented using PX4's parameter system. This simplifies controller tuning by allowing users to tune gains from the PXH command line or from a QGroundControl ground station.

jMAVSim is a simple multirotor-specific simulator included in the PX4 code. It is developed in Java and uses the `java3d` library for visualizing the UAV in a 3D virtual world. See Figure 2 for a screenshot of the visualization.

jMAVSim numerically integrates the rigid body model (1) and in its `Simulator` class the user can hardcode the UAV model parameters. We chose these parameters to match ANCLQ2 quadrotor described in (Fink, 2018). An "x" vehicle configuration is specified with armlength 16.5 cm, $m = 1.6\,\text{kg}$, $J = \text{diag}(0.03, 0.03, 0.05)\ \text{kg}\,\text{m}^2$, a max individual rotor thrust of 7.75 N, a default max rotor torque of 0.05 N m, and the default low-pass filter with

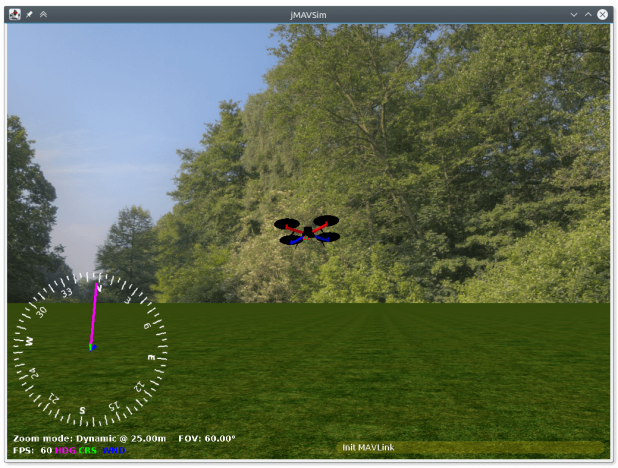

Fig. 2. Visualizing the flying quatrotor in jMAVSim.

time constant of 5 ms was taken to model rotor inertia. Note that the jMAVSim SITL naturally includes thrust saturation as it uses the normalized thrust and torque inputs from PX4. These normalized inputs are low pass filtered and multiplied by the user specified maximum thrust parameter to obtain individual rotor thrust used in simulation (see class `Rotor`). jMAVSim adds user-configurable time delay to GPS measurements and zero mean Gaussian white noise to the accelerometer, gyro, and magnetometer measurements. The defaults for these noise parameters were chosen. Since PX4 only accepts normalized thrust, we have scaled thrust such that it equals 0.5 in hover. This number was been calculated from the stock `mc_pos_control` controller when the UAV is in hover.

jMAVSim can conveniently add environmental disturbances with wind. Hence, we use this feature to model the force disturbance d_f. From the `AbstractMultiCopter` class the model for the force d_f acting on the UAV is proportional to the airspeed, i.e.,

$$d_f = c \cdot (w - v) \tag{13}$$

where $w \in \mathbb{R}^3$ is the wind velocity vector in $\mathcal{N}$ which is a random process whose variance and mean can be controlled by the user, the constant c is a drag coefficient whose default value 0.03 is used. Wind velocity is obtained from

$$\frac{dw}{dt} = -\frac{w}{\tau} + \left(\frac{W}{\tau} + \nu\right) \tag{14}$$

where $\nu = [\nu_1, \nu_2, \nu_3]^T \in \mathbb{R}^3$ with its components $\mathcal{N}(0, (\tau\sigma_{wk})^2)$, i.e., Gaussian white noise processes with zero mean and variance $(\tau\sigma_{wk})^2$, where $\tau = 2\,\text{s}$, $\sigma_{w1} = 6\,\text{m/s}, \sigma_{w2} = 8\,\text{m/s}, \sigma_{w3} = 0$, and $W = [0, -15, 0]^T$ m/s is the constant which determines the average steady state value of w. This value of W corresponds to fairly challenging wind conditions. The parameters σ_{wk} and W can be input by the user using the GUI or can be hardcoded in the `Simulator` class.

The reference trajectory is taken as a figure-8 given by $p_d(t) = [A\sin(\frac{2\pi t}{T}) + 1, B\sin(\frac{4\pi t}{T}), -0.85]$ m, with $A = 1.5$ and $B = 0.75$. The velocity of the trajectory is increased by decreasing T from 20 at $t = 0$ to 12 in 8 seconds. The setpoint for yaw is zero. In order to investigate the effect of the disturbance observer and integral augmentation on performance, four controller are tested: I-Backstepping (BS), II-Disturbance Observer-based Backstepping (DOB-BS), III-Integral Backstepping (IBS), IV-Disturbance Observer-based Integral Backstepping (DOB-IBS). For all cases we take $k_1 = k_2 = k_3 = k_4 = k_5 = 4$ and an observer gain $k_{d_f} = 0.5$. The vehicle is initialized in hover at $p(0) = [0.5, 0.5, -1]^\top$ m. Hence, a relatively large initial error exists when the controller is switched on, i.e., $\tilde{p}(0) = [0.5, 0.5, -0.15]^\top$ m.

Simulation results for the DOB-IBS controller are shown in Figures 5–8. As can be observed, the tracking error in Figure 5 converges to an acceptable (i.e., $\|\tilde{p}\| \approx 5\,\text{cm}$ in steady-state) neighborhood of the origin in approximately 10 s and remains there. The disturbance estimates are shown in Figure 6. We observe $\hat{d}_{f1}$ has approximately zero average in steady-state and oscillations at the frequency of the reference trajectory. The average value of $\hat{d}_{f2}$ in steady-state is about -0.5 N which is consistent with the applied wind disturbance in the $-n_2$ direction. The steady-state average of $\hat{d}_{f3}$ is -5 N which is due to error in thrust scaling. Plots for the Euler angles are shown in Figure 7, and normalized torques and thrust are in Figure 8. The inputs have reasonable magnitude and remain unsaturated in steady-state. As expected, the thrust has an average value close to 0.5 N.

The simulation results for the actual and desired trajectories are shown in 2D for all controllers in Figure 3. Table 1 provides the root mean square of the position error in steady-state. The trajectory for the norm of the tracking error is in Figure 4. We observe from Table 1 the DOB-IBS control achieves the lowest steady-state RMSE norm of tracking error. The 2D plots clearly show this reduced error. The BS control suffers from a lack of robustness to the wind forces and thrust scaling and has the largest steady state error. Adding the disturbance observer (DOB-BS) or the integral term (IBS) improves robustness, but not to the level of the combine disturbance observer and integral augmentation in DOB-IBS.

5. CONCLUSION

A disturbance observer-based integral backstepping controller has been proposed for trajectory tracking of multirotor UAVs. Global asymptotic stability of the closed loop is proven in the presence of constant force disturbance. SITL simulations demonstrate improved controller performance of the disturbance observer-based integral backstepping design.

REFERENCES

Bouabdallah, S. (2007). *Design and control of quadrotors with application to autonomous flying.* Ph.D. thesis, Ecole Polytechnique Federale de Luasanne, Lausanne, Switzerland.

Cabecinhas, D., Cunha, R., and Silverstre, C. (2014). A nonlinear quadrotor trajectory tracking controller with disturbance rejection. *Control Eng. Practice*, 26, 1–10. doi:10.1016/j.conengprac.2013.12.017.

Castillo, P., Lozano, R., and Dzul, A. (2005). *Modelling and control of mini flying machines.* Springer-Verlag, New York City, USA.

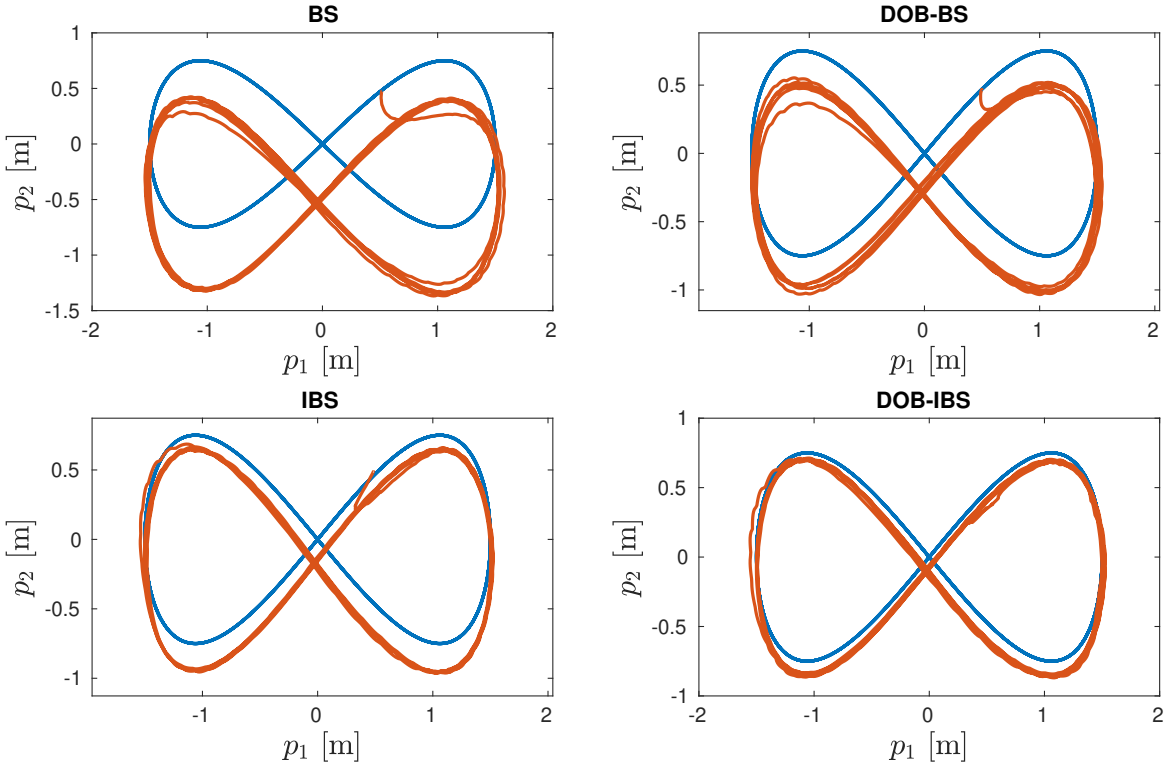

Fig. 3. Desired and actual trajectories for the control methods simulated in jMAVSim

Table 1. Root mean square error (RMSE) of the steady-state position tracking error.

Controller	BS	DOB-BS	IBS	DOB-IBS
RMSE($\|\tilde{p}\|$)m	0.26645	0.14387	0.088845	0.052095

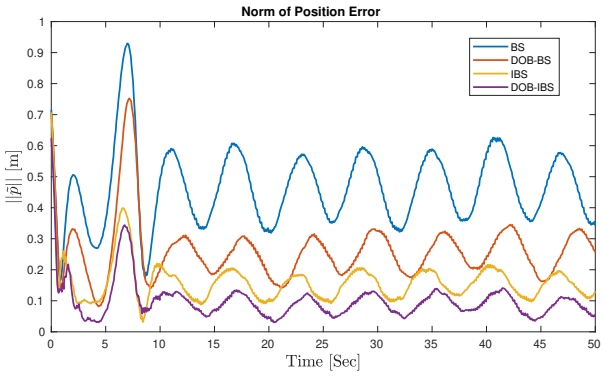

Fig. 4. Norm of trajectory tracking error of the simulated methods in jMAVSim

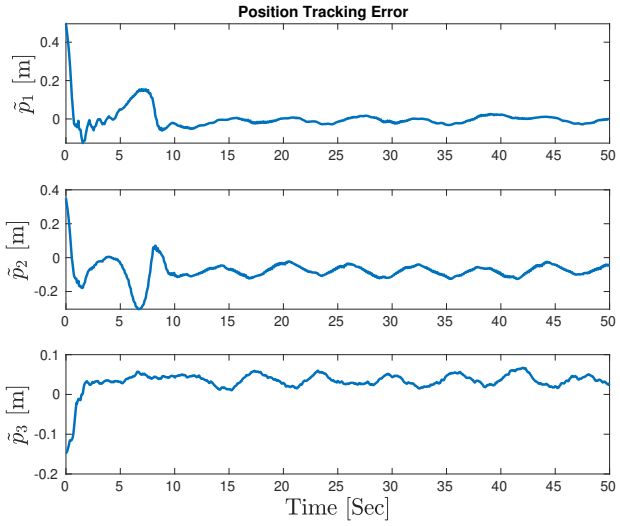

Fig. 5. Position tracking error for the DOB-IBS controller

Colmenares-Vazquez, J., Marchand, N., Castillo, P., Gomez-Balderas, J.E., Alvarez-Munoz, J.U., and Tellez-Guzman, J.J. (2015). Integral backstepping control for trajectory tracking of a hybrid vehicle. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 209–217. doi:10.1109/ICUAS.2015.7152293.

Dong, W., Gu, G.Y., Zhu, X., and Ding, H. (2014). High-performance trajectory tracking control of a quadrotor with disturbance observer. *Sensors and Actuators A: Physical*, 211, 67–77. doi:10.1016/j.sna.2014.03.011.

Fink, G. (2018). *Computer Vision-Based Motion Control and State Estimation for Unmanned Aerial Vehicles (UAVs)*. Ph.D. thesis, Dept. of Electrical and Computer Engineering, University of Alberta, Edmonton, AB.

Hamel, T., Mahony, R., Lozano, R., and Ostrowski, J. (2002). Dynamic modelling and configuration stabiliza-

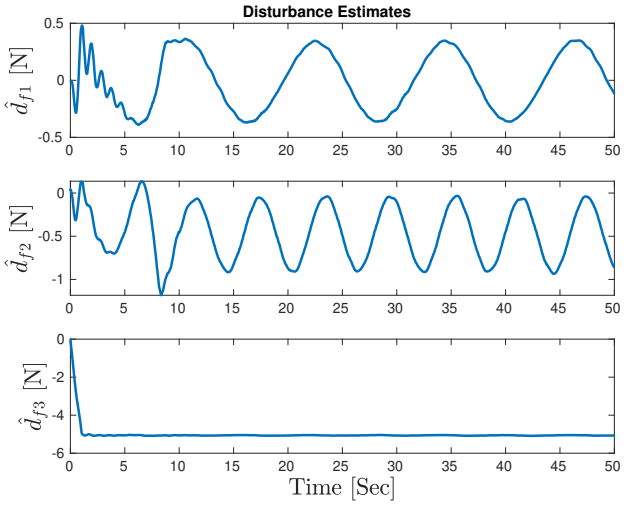

Fig. 6. Disturbance Estimates for the DOB-IBS controller

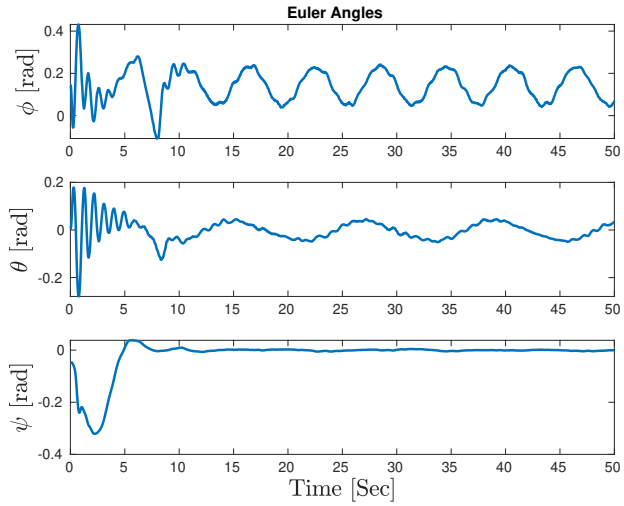

Fig. 7. Euler angles for the DOB-IBS controller

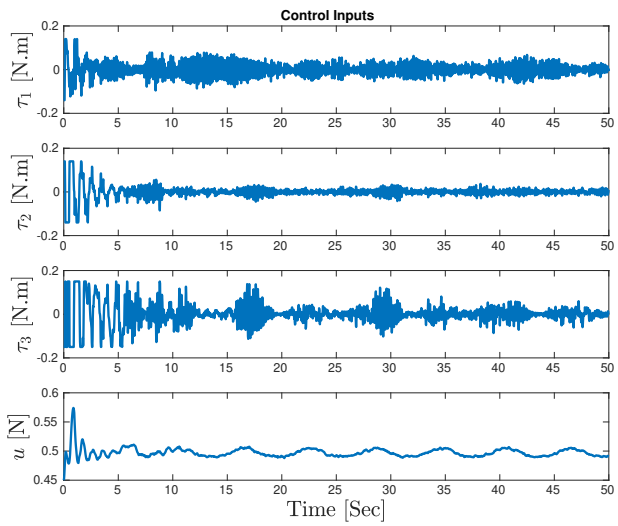

Fig. 8. Control inputs for the DOB-IBS controller

012. URL http://www.sciencedirect.com/science/article/pii/S0921889017305535.

Lee, S.J., Kim, S., Johansson, K.H., and Kim, H.J. (2016). Robust acceleration control of a hexarotor UAV with a disturbance observer. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, 4166–4171. doi:10.1109/CDC.2016.7798901.

Meier, L. (2015). Mavlink: Micro air vehicle communication protocol. http://qgroundcontrol.org/mavlink/start [accessed 01 May 2015]. URL http://qgroundcontrol.org/mavlink/start.

Meier, L. (2018). PX4 autopilot. http://pixhawk.org/ [accessed 01 Jan 2018]. URL http://pixhawk.org/.

Moeini, A., Xue, Z., and Lynch, A. (2019). Modified px4 autopilot firmware. https://github.com/ANCL/UAV_DOBIBS [accessed Nov 14, 2019]. URL https://github.com/ANCL/UAV_DOBIBS.

Poultney, A., Gong, P., and Ashrafiuon, H. (2019). Integral backstepping control for trajectory and yaw motion tracking of quadrotors. *Robotica*, 37(2), 300–320. doi:10.1017/S0263574718001029.

Shao, X., Liu, J., Cao, H., Shen, C., and Wang, H. (2018). Robust dynamic surface trajectory tracking control for a quadrotor UAV via extended state observer. *Int. J. Robust Nonlin.*, 28, 2700–2719. doi:10.1002/rnc.4044.

Xie, H. (2016). *Dynamic Visual Servoing of Rotary Wing Unmanned Aerial Vehicles*. Ph.D. thesis, Dept. of Electrical and Computer Engineering, University of Alberta, Edmonton, AB.

tion for an X4-Flyer. In *Proc. IFAC World Congress*, volume 35, 217–222. Barcelona, Spain. doi:10.3182/20020721-6-es-1901.00848.

He, Y., Pei, H., and Sun, T. (2014). Robust tracking control of helicopters using backstepping with disturbance observers. *Asian Journal of Control*, 16(5), 1387–1402. doi:10.1002/asjc.827.

Hua, M.D., Hamel, T., Morin, P., and Samson, C. (2013). Introduction to feedback control of underactuated VTOL vehicles: A review of basic control design ideas and principles. *IEEE Contr. Sys. Mag.*, 33(1), 61–75. doi:10.1109/MCS.2012.2225931.

Jasim, W. and Gu, D. (2015). Integral backstepping controller for quadrotor path tracking. In *2015 International Conference on Advanced Robotics (ICAR)*, 593–598. doi:10.1109/ICAR.2015.7251516.

Khalil, H.K. (2002). *Nonlinear Systems*. Prentice Hall, Upper Saddle River, NJ, 3 edition.

Khamseh, H.B., Janabi-Sharifi, F., and Abdessameud, A. (2018). Aerial manipulation: A literature survey. *Robotics and Autonomous Systems*, 107, 221 – 235. doi:https://doi.org/10.1016/j.robot.2018.06.