# Disturbance Observer-Based Integral Backstepping Control for UAVs

**Amir Moeini** * **Zhijun Xue** ** **Alan F. Lynch** * **Qing Zhao** *

* *Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, T6G 2V4, Canada (e-mail: {moeini,alan.lynch,qingz}@ualberta.ca).*
** *Huazhong University of Science Technology, (e-mail: zxue2@ualberta.ca)*

**Abstract:** In this paper, we propose an integral backstepping nonlinear controller coupled with a disturbance observer for motion control of a multirotor unmanned vehicle. The controller is developed considering the complete dynamics of the multirotor and without any approximation. The proposed controller is easy to implement asymptotic stability of the tracking error is proved in presence of constant disturbance. The performance of the proposed disturbance observer-based controller is evaluated by numerical simulations on Matlab/Simulink and jMAVsim simulator in the presence of external disturbances and modeling errors. The result suggests a significant improvement in the tracking error when both the integral action and the disturbance observer are employed in the closed-loop system.

*Keywords:* Multirotor Unmanned Aerial Vehicles (UAVs), backstepping control, disturbance observer, 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. Although many variations exist, most of these designs 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 are 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-Vázquez 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 actual dynamics. 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 desired 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.

The other approach is to include both the disturbance observer and the integral term into the controller since they have a complementary effect (the disturbance observer can reject the external disturbances and the integral action can counteract with the model uncertainties). Paper (Fang and Gao, 2011) assumes constant force disturbance and retain a hierarchical structure to design the controller. Three different controllers, adaptive integral backstepping, adaptive backstepping and integral backstepping are designed respectively for horizontal, altitude and attitude subsystems. To overcome the large overshoots which are consequences of integral actions, the integral terms come into action when the error is less than a specified value. Simulation results in Matlab are presented for the mass variation and external constant disturbance cases and the results are compared to the integral backstepping and PID controllers. In (Younes et al., 2014) a cascaded strategy is designed by two adaptive integral backstepping controllers to ensure the stability of the vehicle. The integral action is used to reduce the steady state error caused by the modeling error and the adaptive part is added to estimate the disturbance. First the controller is designed by the assumption that the disturbance is known, then the estimate of the disturbance is replaced in the control expressions. Small angle perturbation is assumed for the rotational subsystem. experimental results for a trajectory tracking task are presented, however, the effect of external disturbance on the closed-loop system is not investigated.

Mention importance of SITL simulation as an intermediate step to flight testing.

The contribution of this paper is ... In this paper, we have designed a disturbance observer-based integral backstepping controller considering the full dynamics of the multirotor and without any simplifying assumptions. The control is developed by starting from the integral of position tracking error and using the disturbance estimate in the design. Simulation results for validation of the proposed controller are presented in Matlab/Simulink and jMAVsim simulator. The results are compared to backstepping, integral backstepping and disturbance observer-based backstepping in presence of modeling errors and external time-varying disturbances showing a significant improvement in the performance for the proposed method.

## 2. QUADROTOR DYNAMICS

In this section we review a traditional rigid body quadrotor model to 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 $\tau$ and thrust $u$ as system inputs. However, the physical input to the UAV are PWM signals to the Electronic Speed Controller (ESC). 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, and $n_3 = [0, 0, 1]^\top$. 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. The skew operator $S(\cdot) : \mathbb{R}^3 \to so(3)$ is defined as

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

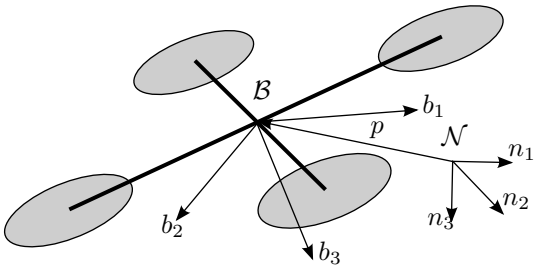


Fig. 1. Diagram of a quadrotor showing navigation frame $\mathcal{N}$ and body frame $\mathcal{B}$.

## 3. OBSERVER-CONTROLLER DESIGN

This section solves an output tracking problem for (1) where the output includes $p$ and $\psi$. Given smooth bounded desired trajectories for position and yaw, denoted by $p_d \in \mathbb{R}^3$ and $\psi_d \in \mathbb{R}$, respectively, we derive a dynamic state feedback control for inputs $u$ and $\tau$ to ensure exponential convergence of the tracking errors $p - p_d$ and $\psi - \psi_d$ in the presence of constant bounded disturbances $d_f$ and $d_\tau$. 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 two 3-dimensional disturbance observers to reject disturbances $d_f$ and $d_\tau$. The use of disturbance observers is important for ensuring closed-loop asymptotic stability.

Using tuning functions ensures a minimal order of 3 for the disturbance observer of $d_f$. Since $p$ and $\psi$ can be independently controlled, the design is divided into two sections: position and yaw control.

*Disturbance Observer* We consider the following observer for estimating the force disturbance

$$\hat{d}_f = z_{d_f} + k_{d_f} mv \tag{2a}$$

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

Assuming $d_f$ is slowly time-varying ($\dot{d}_f \approx 0$) 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.

*Position Tracking Control* In this subsection we develop a motion controller for position by employing integral backstepping. We start by defining the integral position tracking error $\delta_1 = \int_{t_0}^{t}(p(\xi) - p_d(\xi))d\xi$ and considering the first Lyapunov function candidate as $V_1 = \frac{1}{2}\|\delta_1\|^2$. Taking the time derivative of $V_1$ 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

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

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 $\dot{\delta}_2 = v + k_1(p - p_d) - \dot{p}_d$ and therefore

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

By considering $\alpha_2$ as the desired value for the second virtual control $v$ and considering

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

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

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

Now by considering $uRn_3$ as the virtual control and define its desired value by

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

and consider $\delta_4 = \alpha_3 - uRn_3$, it results in

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

Note that we have estimated the force disturbance appeared in $\dot{\delta}_3$ by its estimate in $\alpha_3$. Next we add the $\delta_4$ to the Lyapunov function and define $V_4 = V_3 + \frac{1}{2}\|\delta_4\|^2$,

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

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

$$\begin{aligned}\dot{\alpha}_3 &= \dot{\delta}_2/m + k_3(mgn_3 - uRn_3 + d_f + m\dot{\delta}_1 + mk_1(v - \dot{p}_d) \\ &\quad - m\ddot{p}_d + mk_2\dot{\delta}_2) + k_{d_f}\tilde{d}_f + m(\dot{p} - \dot{p}_d) \\ &\quad + k_1(mgn_3 - uRn_3 + d_f - m\ddot{p}_d) - m\dddot{p}_d \\ &\quad + k_2(mgn_3 - uRn_3 + d_f + mk_1(v - \dot{p}_d) - m\dot{v}_d)\end{aligned} \tag{4}$$

As it can be observed some of the terms in $\dot{\alpha}_3$ are unknown due to the unknown force disturbance, therefore we estimate $\dot{\alpha}_3$ by $\beta$ 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

$$\dot{\delta}_4 = \beta + (k_3 + k_1 + k_2 + k_{d_f})\tilde{d}_f - \dot{u}Rn_3 - uRS(\omega)n_3 \tag{5}$$

Now we try to assign values of $\dot{u}$ and $\omega$ to make $\dot{V}_4$ negative definite. Since $u$ is the actual input, we can directly assign its value. However for $\omega$ we need to consider $uRS(\omega)n_3$ as the virtual control and define a desired value for it. From $\dot{V}_4$ and (5) we can see that for negative definiteness of $\dot{V}_4$ we need to cancel out $\beta$, the indefinite term $\delta_3^\top\delta_4$ and also add a damping term $-k_4\|\delta_4\|^2$. So by considering

$$\dot{u} = n_3^\top R^\top(\beta + \delta_3 + k_4\delta_4) \tag{6}$$

and 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

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

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

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

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

$$\begin{aligned}\dot{\alpha}_4 &= RS(\omega)[I - n_3n_3^\top]R^\top(\beta + \delta_3 + k_4\delta_4) \\ &\quad + R[I - n_3n_3^\top]S(\omega)^\top R^\top(\beta + \delta_3 + k_4\delta_4) \\ &\quad + R[I - n_3n_3^\top]R^\top(\dot{\beta} + \dot{\delta}_3 + k_4\dot{\delta}_4).\end{aligned} \tag{7}$$

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 equations ($\dot{\beta}'$, $\dot{\delta}_3'$, $\dot{\delta}_4'$). The relation between the actual values and the estimates are given below

$$\dot{\beta} = \dot{\beta}' + (1/m^2 + k_3k_{d_f} + k_3mk_1 + k_3k_2 + k_1k_{d_f} + k_2k_{d_f})\tilde{d}_f$$

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

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

By using the estimates in (7) we can derive an estimate of $\dot{\alpha}_4$

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

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

$$\dot{u}RS(\omega)n_3 + uRS(\omega)^2n_3 - uRS(n_3)\dot{\omega} = \dot{\alpha}_4' + \delta_4 + k_5\delta_5 \tag{8}$$

then we can write

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

Denoting $\dot{\omega}_d = [\dot{\omega}_{d1}, \dot{\omega}_{d2}, \dot{\omega}_{d3}]^\top$ and solving (8) for $\dot{\omega}_{d1}$ and $\dot{\omega}_{d2}$ gives

$$\dot{\omega}_{d1} = -\frac{n_2^\top R^\top}{u}(\dot{\alpha}_4' - \dot{u}RS(\omega)n_3 - uRS(\omega)^2n_3 + \delta_4 + k_5\delta_5) \tag{9a}$$

$$\dot{\omega}_{d2} = \frac{n_1^\top R^\top}{u}(\dot{\alpha}_4' - \dot{u}RS(\omega)n_3 - uRS(\omega)^2n_3 + \delta_4 + k_5\delta_5). \tag{9b}$$

In order to obtain the expressions for the torque input $\tau$ to achieve position tracking, we assume $\dot{\omega}_{d3} = 0$ which means the yaw rotational degree of freedom is uncontrolled. The torque can be directly derived from

$$\tau = J\dot{\omega} + \omega \times J\omega \tag{10}$$

The closed loop stability of the proposed backstepping controller is proved by 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), (6) (9) and (10).

*Proof* : If we consider $x_1 = [\delta_1^\top, \delta_2^\top, \delta_3^\top, \delta_4^\top, \delta_5^\top]$ and $x_2 = \tilde{d}_f$, the disturbance observer and the backstepping controller can be considered as a cascade system

$$\dot{x}_1 = f_1(x_1, x_2) \tag{11a}$$
$$\dot{x}_2 = f_2(x_2) \tag{11b}$$

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

- The origin of (11b) is globally uniformly asymptotically stable.
- System (11a) is input to state stable with respect to $x_2$ as 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 the input to state stability of (11a) with respect to $x_2$ as the input. For this purpose we use (Khalil, 2002, Lemma. 4.6) which states (11a) is input to state stable if

- The unforced system $\dot{x}_1 = f_1(x_1, 0)$ is globally exponentially stable at the origin.
- System (11a) is continuously differentiable and globally Lipschitz in $x_1$ and $x_2$.

From equation given 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.

For the second condition we write the explicit equation for $\dot{x}_1 = f_1(x_1, x_2)$ which gives us $\dot{x}_1 = Ax_1 + Bx_2$ where

$$A = \begin{bmatrix} -k_1 & 1 & 0 & 0 & 0 \\ -1 & -k_2 & 1/m & 0 & 0 \\ 0 & -1/m & -k_3 & 1 & 0 \\ 0 & 0 & -1 & -k_4 & 1 \\ 0 & 0 & 0 & -1 & -k_5 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \\ k_{d_f} + k_1 + k_2 \\ kR(I - n_3 n_3^\top)R^\top \end{bmatrix}$$

which is a linear time invariant system and therefore continuously differentiable and globally Lipschitz in $x_1$ and $x_2$.

An important contribution of this work is the decoupled structure of the observer from the backstepping controller. This aspect makes the controller design simple and easily implementable on the real platform. Our approach should be compared to adaptive backstepping control (Cabecinhas et al., 2014) where parameter estimators only yield asymptotic stability. In fact, all the abovementioned existing work only provide local and almost global asymptotic stability results, sometimes only for an outer loop subsystem (Antonelli et al., 2018) or for overly simplified models (Shao et al., 2018; Dong et al., 2014; Lee et al., 2016). Another novel aspect of the proposed method is avoiding the overparameterization as in (Cabecinhas et al., 2014). This is important to ensure the controller's simple implementation without having to employ approximations such as dynamic surface control.

## 4. NUMERICAL VALIDATION

In this section we present simulations of the proposed control using Matlab/Simulink and jMAVsim.

### *4.1 Matlab/Simulink*

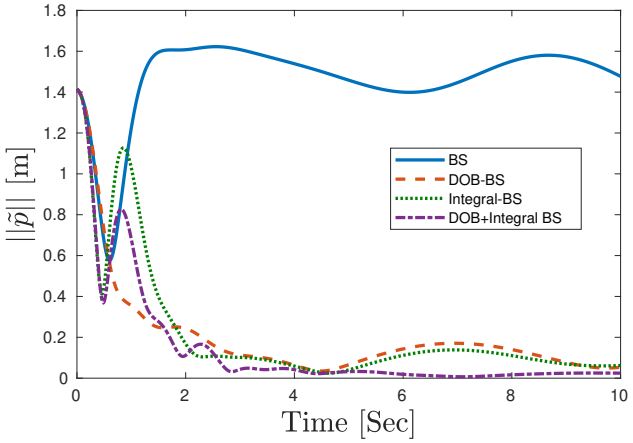


Fig. 2. Norm of trajectory tracking error of the simulated methods in Matlab/Simulink

In this subsection we simulate the proposed method in MATLAB/Simulink and present the results. The desired trajectory for position is considered a figure-8 shape trajectory given by the equation $p_d(t) = [A\sin(\frac{2\pi t}{T}), B\sin(\frac{4\pi t}{T}), -1]$ m, where $A = 1.5$ is the amplitude of the trajectory in the $n_1$-direction, $B = 1$ is the amplitude in the $n_2$-direction, and $T = 12$ is the period of the trajectory. This trajectory is sufficiently aggressive to evaluate the performance of the proposed controller and at the same time smooth and bounded as required. The initial conditions of the quadrotor are $p(0) = [-1, 1, -1]^\top$ m, $R(0) = I_3$. The initial conditions for the linear and angular velocities are $v(0) = 0$ and $\omega(0) = 0$, respectively. The parameters of the simulation are chosen to match the quadrotor platform in ANCL lab (Fink et al., 2015) $m = 1.6$ kg $J = \text{diag}(0.03, 0.03, 0.05)$ kg m$^2$. For accurate comparison of the effects of disturbance observer and the integral action on the robustness of the motion controller, we have performed the simulation for the following cases: I-Backstepping, II-Disturbance Observer-based Backstepping, III-Integral Backstepping, IV-Disturbance Observer-based Integral Backstepping. For having a fair evaluation of the effect of the integral action and the disturbance observer all the controller and observer gains are set to 5. For implementation of the backstepping controller we just need to cancel the integral action by setting $k_1$ and $\delta_1$ to zero. Also, the observer can be turned off by setting $k_{d_f}$ to zero. For performance evaluation, we have considered an uncertainty in mass and considered the mass as $m = 1.2$ kg which has resulted in a mismatch equal to 0.4 kg. Also the inertia matrix is considered $J = \text{diag}(0.025, 0.025, 0.04)$ kg m$^2$. Also the external disturbance is considered equal to $d_f(t) = [6 + 4\sin(0.5\pi t), -6 - 3\cos(0.8\pi t), -6 +$

$3\sin(0.4\pi t)]^\top$ N. Figure 2 shows the the norm of tracking error for each of the implemented methods. As it can be seen the best performance is achieved when a combination of disturbance observer and integral action is added to the backstepping controller, while the backstepping itself is not converging. Also the performance of the DOB-based backstepping and integral backstepping are similar except for the transient response which the DOB-BS has a smoother response. This is actually expected since the integral terms is known to add overshoot. Therefore, we can conclude that the controller robustness is increased when both the disturbance observer and the intergal action are added to the backstepping controller and therefore the simple backstepping itself is not a robust controller (as opposed to what is known in the literature).

### *4.2 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 on a regular PC where actual autopilot code is interfaced to a simulator which models the vehicle dynamics and the environment (e.g., wind disturbance forces).

Since our lab has already invested in the PX4 autopilot platform Fink (2018); **?**, it is natural to use this project's existing SITL simulation tools. We choose the simple jMAVSim simulator which accepts ESC PWM inputs from the PX4 code and supplies the UAV's sensor GPS, IMU, and barometer measurements which correspond to the simulated rigid body motion of the UAV. Generally speaking, SITL simulation results are not usually presented in the literature. However, they are key to a safer accelerated development including improved debugging and controller tuning.

*PX4 Controller Implementation* `mc_dobibs_control` We implement our proposed control on the PX4 autopilot which is active open-source project intended for research and industrial applications and which runs on multiple hardware platforms and various robot types. The PX4 autopilot 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 platforms or as a thread within the main PX4 process on POSIX 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 and well-defined portion of the code needs to be modified. The so-called PX4 middleware supports communication between the modules, sensor device drives, and communication outside PX4. Module communication is implemented with the micro-object request broker (uORB) which provides a publish/subscribe bus. Using uORB (module source is in `UAV_IBS/src/modules/uORB`) publishers send messages (e.g., a UAV control input) onto a bus instead of sending the messages directly to specific subscribers. Also, subscribers receive messages whenever there are updates. This publish/subscribe model is commonly used in robotics (e.g., the Robot Operating System (ROS)) to avoid locking issues.

The PX4 software collects measurements from a range of sensors such as gyroscope, accelerometer, magnetometer, GPS, barometer. Using this data it outputs the PWM control signals to the ESCs. It provides driver modules for the hardware components (e.g. IMU, GPS and PWM outputs) and libraries for programming.

The proposed controller is implemented as a new module `mc_dobibs_control` located in `UAV_IBS/src/modules/mc_dobibs_` The implementation uses a

We have employed various libraries to facilitate the implementation of the controller. We defined a class based on block structure in control library (*px4/src/lib/controllib*) to simplify subscriptions and publications tasks. This structure enables us to create a module by defining a class which includes all the necessary subscribed and published topics. By employing this structure, all the necessary steps for subscription/publication of each topic will be performed automatically by the functions defined in control library. We have also used the PX4 matrix library (*https://github.com/PX4/Matrix*) in writing the code of the controller. This has helped improve the readability of the code for the proposed controller as it closely matches the mathematical expressions derived. Since the PX4 autopilot only accepts normalized thrust, we have normalized the thrust such that it is equal to 0.61 in hover. This number has been calculated from the PID controller when the quadrotor is at hover.

Table 1. Classes description in jMAVsim

| Class | Description |
|---|---|
| SimpleEnviroment | Simulates the magnetic field, earth gravity, air pressure. |
| DynamicObject | Calculates the systems states (position, attitude, linear and angular velocities) from the force and torque acting on the multirotor. |
| AbstractMultiCopter | Generic quadcopter constructor. Receives the rotors speeds and the airspeed and outputs the force and torque that are applied to the multirotor. |
| Environment | Calculates the gravity, current wind and magnetic field and send them to visualizer |
| Rotor | Calculates torque and thrust generated by each rotor by low-pass filtering the control signals. |
| Visualizer3D | Graphical visualization of multirotor |
| Sensors | Collects the data as measurement and pass them to MAVlink and AbstractVehicle classes. |
| Simulator | Simulator configuration. Includes quadrotor parameters (mass, inertia thrust constant, drag constant) and other simulation environment information. |
| MAVLinkHILSystem | Connects the jMAVsim to PX4 autopilot. |

For the simulator environment we have chosen jMAVSim. It is a simple multirotor simulator with MAVLink protocol support developed by PX4 team. It is developed based on JAVA and uses java3d library for graphical visualization. It

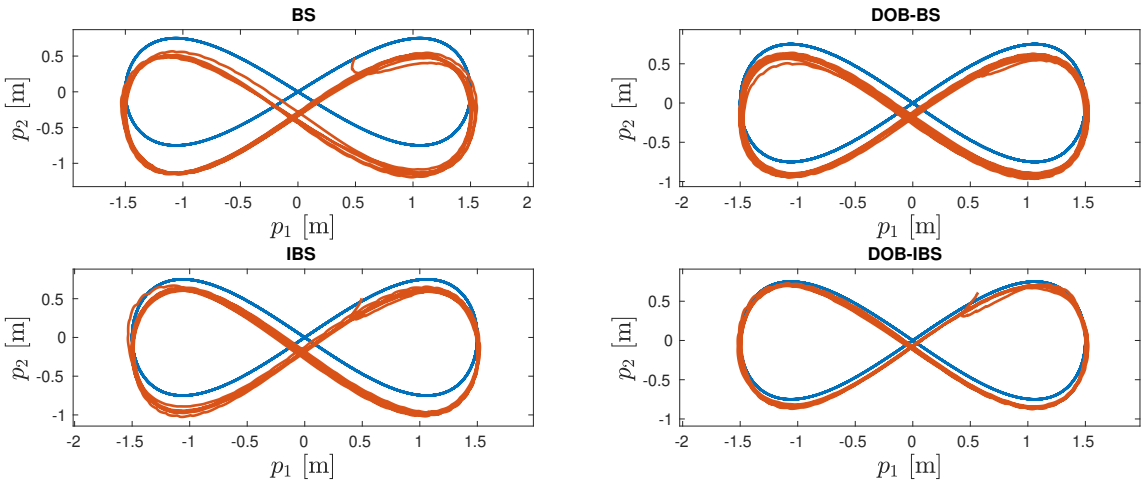


Fig. 3. Desired and actual trajectories for the control methods simulated in jMAVsim

has the ability to simulate the flight characteristics of multirotors with accurate dynamical and sensor noise modeling. It uses Micro Aerial Vehicle Link (MAVLink)(Meier, 2015) for communication with px4 firmware through a User Datagram Protocol (UDP) communication and has the ability to add extra sensor noise and environmental disturbances (wind-gust). It has a built-in multicopter models that produce the attitude and position of the multicopter. Sensor models are employed to simulate the real world characteristics and their output will be the input for the flight controller to calculate the control commands. The parameters of the quadcopter model in jMAVSim is chosen exactly equal to the quadrotor parameters that we have in ANCL lab (Fink et al., 2015). To have a better understanding of jMAVsim simulator, its main classes are briefly explained in Table tab:jMAVSim.

The code for this implementation along with the instructions are uploaded on GitHub (**?**) which can be downloaded and tested.

The external disturbance affecting the multirotor is originated from the air speed that the quadrotor is sensing. Air speed is calculated by adding the speed of the multirotor and the wind speed. Then the force is calculated by multiplying the airspeed with a drag coefficient which is considered to be 0.01 in this simulation.

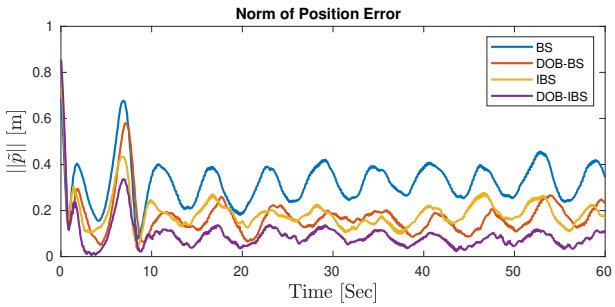


Fig. 4. Norm of trajectory tracking error of the simulated methods in jMAVsim

For the experiment the commanded desired trajectory is considered a figure-8 shape trajectory given by $p_d(t) = [A\sin(\frac{2\pi t}{T}) + 1, B\sin(\frac{4\pi t}{T}), -0.8]$ m, with $A = 1.5$ and $B = 0.75$. The velocity of the trajectory is increased by time when $T$ starts from 20 and goes to 12 in 8 seconds. The setpoint for yaw is considered to be zero. Similar to the Matlab simulation, four controller were tested: I-Backstepping, II-Disturbance Observer-based Backstepping, III-Integral Backstepping, IV-Disturbance Observer-based Integral Backstepping. All the controller gains are considered equal 4 and the observer gain is $k_{d_f} = 0.5$. To test the robustness of the controllers we disturbed the multirotor by a $-15$ m/s wind in the direction of $n_2$. This will affect the quadrotor as an external force disturbance. The simulation results for the actual and desired trajectories and also the norm of tracking errors are shown in Figures 3 and 4, respectively. As it can be seen the best performance in terms of tracking error is achieved for the Disturbance Observer-based Integral Backstepping controller. There is an offset about 21 cm in $-n_2$ direction for the backstepping controller. The reason can be considered the lack of any robustness enhancing action (integral term or disturbance observer) to the controller. As can be observed when one of the disturbance observer or the integral term is added to the controller, the tracking error will be significantly reduced, however, the best performance is achieved when both the disturbance observer and the integral term are in action.

## 5. CONCLUSION

A disturbance observer-based integral backstepping controller has been proposed for trajectory tracking of multirotors. Asymptotic stability of the closed loop is proven in the presence of constant force disturbance. Disturbance observer and integral action, which are known to add robustness to the closed loop system, are included in the controller development. Numerical simulation shows a significant improvement in the controller performance when both the disturbance observer and the integral action are in the loop in comparison to the other scenarios.

## REFERENCES

Antonelli, G., Cataldi, E., Arrichiello, F., Giordano, P.R., Chiaverini, S., and Franchi, A. (2018). Adaptive trajectory tracking for quadrotor MAVs in presence of pa-

rameter uncertainties and external disturbances. *IEEE Trans. Contr. Syst. Technol.*, 26(1), 248–254. doi:10.1109/TCST.2017.2650679.

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.

Colmenares-Vázquez, J., Marchand, N., Castillo, P., Gómez-Balderas, J.E., Alvarez-Muñoz, J.U., and Téllez-Guzmán, 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.

Fang, Z. and Gao, W. (2011). Adaptive integral backstepping control of a micro-quadrotor. In *2011 2nd International Conference on Intelligent Control and Information Processing*, volume 2, 910–915. doi:10.1109/ICICIP.2011.6008382.

Fink, G., Xie, H., Lynch, A.F., and Jagersand, M. (2015). Nonlinear dynamic visual servoing of a quadrotor. *J. Unmanned Vehicle Systems*, 3(1), 1–21. doi:10.1139/juvs-2014-0011.

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 stabilization 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.2018.06.012. 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`.

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.

Younes, Y.A., Drak, A., Noura, H., Rabhi, A., and Hajjaji, A.E. (2014). Quadrotor position control using cascaded adaptive integral backstepping controllers. In *Aerospace and Mechanical Engineering*, volume 565 of *Applied Mechanics and Materials*, 98–106. Trans Tech Publications Ltd. doi:10.4028/www.scientific.net/AMM.565.98.