International Journal of Dynamics and Control (2022) 10:1829–1846
https://doi.org/10.1007/s40435-022-00922-8

A Practical Guideline for Designing and Tuning Adaptive Backstepping Controllers for a Class of Second-Order Systems based on PID Similarity

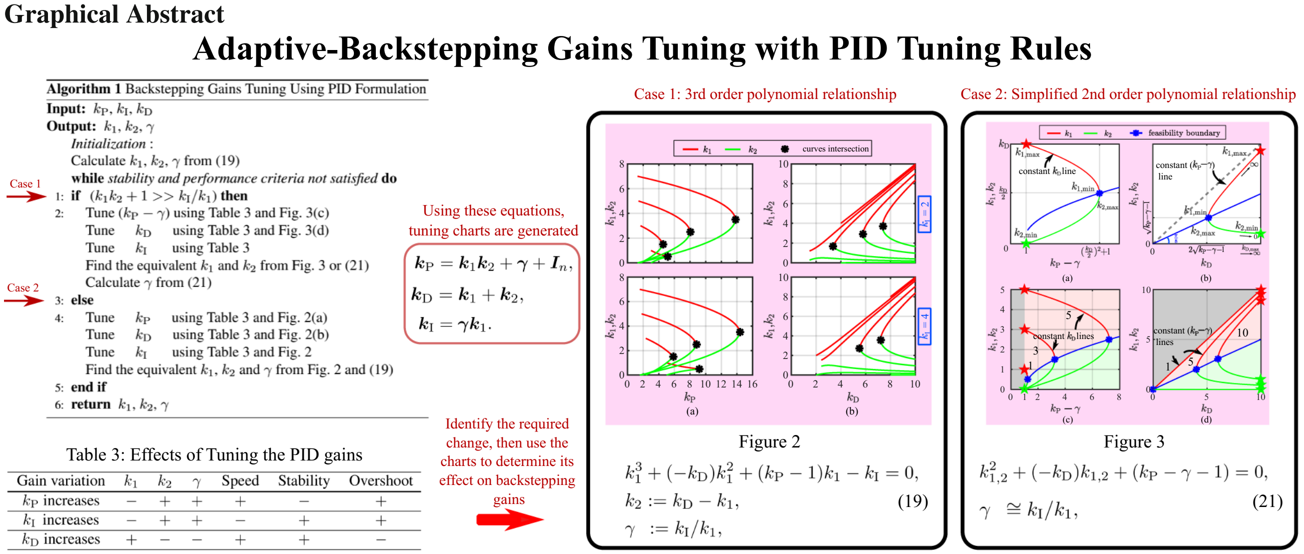
Ahmad Kourani[1] · **Naseem Daher**[2]

Received: 27 August 2021 / Revised: 19 November 2021 / Accepted: 27 January 2022 / Published online: 9 March 2022

Abstract

This work proposes a practical guideline for designing and tuning adaptive backstepping control systems by leveraging the similarity with PID control laws for a class of second-order nonlinear systems. A complete set of mathematical formulations, visual aids, and a well-structured algorithm are provided to exploit the benefits of the established link. This aims at facilitating the adoption of advanced nonlinear control laws in more real-life and industrial applications while benefiting from the legacy of PID tuning rules. Furthermore, the proposed guideline allows for upgrading primitive PID controllers to more advanced nonlinear control system. The adaptive backstepping control law is formulated as a two degrees-of-freedom control law that combines the sum of a feedback PID control component and a feedforward model compensation component. The relationship between backstepping and PID gains is provided in the form of a third-order polynomial, and a simplified second-order one, with practical design algorithm and tuning guidelines. The proposed control law and tuning methodology are validated on a quadrotor unmanned aerial vehicle (UAV) system in both simulation and experimentally.

Graphical Abstract



Keywords Adaptive backstepping control · PID Tuning · Lyapunov stability · Quadrotor UAV

✉ Naseem Daher
nd38@aub.edu.lb

Ahmad Kourani
ahk42@mail.aub.edu

[1] Department of Mechanical Engineering, American University of Beirut, Riad El-Solh, Beirut 1107 2020, Beirut, Lebanon

[2] Department of Electrical and Computer Engineering, American University of Beirut, Riad El-Solh, Beirut 1107 2020, Beirut, Lebanon

1 Introduction

The effectiveness and simple architecture of proportional-integral-derivative (PID) control systems has lead to their wide adoption in real-world industrial applications [1]. Furthermore, PID controllers can be tuned intuitively based on basic concepts that relate system performance characteristics to each one of its three gains [2], or systematically following certain equivalence-based gain selection and tuning techniques [3,4], amongst other methods. However, PID controllers suffer from their inability to address the needs of increased precision and complexity of advanced nonlinear systems due to their oversimplification amongst other issues [5], which lead to the continuous evolution of other control system designs.

The stability and tuning of PID controllers are critical aspects of their deployment. To automate the tuning process of control systems, several self-tuning methods have been proposed by relying on Lyapunov's stability theory, such as designing self-tuning PID controllers for a class of linear and nonlinear PID control systems [1,6].

Still, PID controllers cannot meet the requirements of complex real-life systems and applications, thus more advanced control system designs continued to evolve [7]. However, the complexity and required deep theoretical knowledge of advanced control systems, among-which is backstepping control that is a common method when dealing with nonlinear systems [8], presented a challenge towards their wide adoption. Complex backstepping-based controllers suffer from an intricate tuning process that is required to achieve the desirable system performance.

The complexity of the backstepping design process becomes more challenging when dealing with systems of high-order dynamics [9], namely in systems with order higher than three [10]. Smoothing functions and filters [11] that are introduced at the virtual commands can reduce the effect of excess differentiation [10], but at the expense of introducing time delays in the control signals [10]. It was also demonstrated in [12] that the increased number of design parameters in a backstepping-based adaptive tuning function deteriorated the performance of the controller.

A similarity in the formulation of backstepping-based control laws with the PID control law has been established. For instance, backstepping control with integral action was used to design a robust adaptive PID controller for linear systems in [6]. The previous formulation was later modified in [13] by incorporating the integral action within the first backstepping error, and it was shown that the regulation poles can be placed at any stable location. Later, various forms of the backstepping method with integral action were applied in [14] to a class of nonlinear second-order systems, where the resulting control law was formulated as a feedback PID plus feedforward terms, and the characteristics of each form were compared, albeit without making the connection of the designs to the resultant closed-loop pole locations and overall system performance. Later works on this subject were mostly application-based. For instance, a backstepping-based PID controller was designed in [15] for a quadrotor UAV system, where the form proposed by [13] was generalized to include nonlinear terms. In addition, the developed backstepping analysis tools were employed to design a robust PID controller with guaranteed performance in [16], yet it was only applied to the linearized version of a quadrotor's inner-loop controller. Last but not least, an explicit PID-based tuning guideline of backstepping controllers was proposed for a class of second order nonlinear systems in [17].

The motivation of this work lies in the need for control systems that are easy to design and deploy based on PID control laws, yet advanced enough to deal with uncertainties and disturbances. Even though some prior works have established a link between the backstepping design method and PID control, the similarity remains underappreciated and has not been fully exploited, as evidenced by the fact that the general implicit form of backstepping control design still dominates newly published works in the related literature. On the other hand, a select few robotic applications benefited from the PID representation of the backstepping method to achieve easier controller tuning with guaranteed closed-loop system stability [4,15–19].

This work formulates the similarity between the gains of PID and adaptive backstepping controllers in a form that may enable easier and wider adoption of this nonlinear control method. Here, we emphasize that the scope of this work focuses on establishing an original formulation of the link between existing control techniques, not on providing a new or superior control system design. We present a compact formulation of the adaptive backstepping control law, which consists of two components: PID feedback and feedforward model compensation, for a class of second-order nonlinear systems, which cover a wide spectrum of real-life practical systems [20]. The practicality of this work lies in the unique link that it establishes between backstepping controller gains and the PID gains, whereby tuning the adaptive backstepping gains can be achieved by standard PID tuning rules (e.g., Ziegler–Nichols). We also make available on GitHub the relevant scripts that are needed to compute the gains and plot the various relationships between the backstepping and PID gains. Furthermore, the hereby presented formulation provides insight into the contents of the virtual backstepping commands, which allows for a more prudent employment of filters, thus reducing excessive time-delays. Moreover, we note that a manual tuning process may prove useful and superior since it is based on intuitive understanding of the control law, whereas optimization-based techniques typically require complex or non-intuitive efforts, as well as knowledge of exact system parameters, which is not always feasible [21].

This work offers a unique representation of the similarity between the PID gains structure and the adaptive backstepping gains of a class of nonlinear second-order systems subject to uncertainties, and features multiple advantages. First, we represent the similarity to the PID gains with the minimal number of parameters (three) as compared to the representations in [13,15] that use four parameters. Second, we use a minimum number of dependencies between the adaptive backstepping gains and the PID gains (one multiplication and one summation) as compared to [6,13], which use up to two multiplications and two summations. Third, this work offers a unique and explicit analysis of the implications on this similarity in terms of gains selection and stability, powered by clear and easy-to-follow guidelines in the form of tables, charts, and algorithm. Fourth, we provide a survey on the formulations of the backstepping-PID structure similarity, which makes it easier to choose the suitable method and directly apply it to the systems at hand. These differences are substantiated in a comparative table that is later provided to compare this work to similar works in the literature (Table 1).

The paper starts with defining the problem under consideration in Sect. 2. A baseline controller is presented in Sect. 3, followed by its reformulation into the proposed form in Sect. 4, which also provides the tuning guidelines after establishing the similarity link to PID control. The proposed methodology is validated in Sect. 5 on a quadrotor UAV system in numerical simulations as well as in experimentation. Finally, the work is summarized in Sect. 6 and suggestions on future work are provided.

2 Problem formulation

In this section, we introduce the general form of second-order systems, which are amongst the most popular and practical system models, and that may benefit from the nonlinear control law in the PID-feedback plus feedforward-model-compensation form presented in this work.

2.1 Preliminaries

Notations that facilitate the formulations developed in this paper are first provided here. The set of positive-real numbers $\{x \in \mathbb{R} \mid x > 0\}$, and the set of non-negative real numbers $\{x \in \mathbb{R} \mid x \geq 0\}$, are respectively denoted by $\mathbb{R}_{>0}$ and $\mathbb{R}_{\geq 0}$. The mean absolute value of a signal $(\cdot)$ is denoted by $MAV(\cdot)$, and its L_2 norm is denoted by $\|\cdot\|$. Also, for an angle $(\bullet)$, the cosine, sine, and tangent functions are denoted by $c_\bullet$, $s_\bullet$, and $t_\bullet$, respectively.

In addition, for an estimate $\hat{x}$ of $x \in [x_{\min}, x_{\max}]$ and a signal $(\cdot)$, let the projection function, $\mathrm{Proj}_x(\cdot)$, be defined as [22]:

$$\mathrm{Proj}_{\hat{x}}(\cdot) = \begin{cases} 0 & \text{if } \hat{x} = x_{\min} \text{ and } \cdot < 0 \\ 0 & \text{if } \hat{x} = x_{\max} \text{ and } \cdot > 0 \\ \cdot & \text{otherwise.} \end{cases} \tag{1}$$

2.2 Problem formulation

Let $\boldsymbol{X}_1 = [x_1, \ldots, x_n]^\intercal$ and $\boldsymbol{X}_2 = [\dot{x}_1, ..., \dot{x}_n]^\intercal$ represent the state vectors of a second-order nonlinear time-varying system, and let $\boldsymbol{U} = [u_1, ..., u_n]^\intercal \in \mathbb{R}^n$ represent the control input vector, where $n \in \mathbb{N}^+$. The system is represented as:

$$\begin{aligned} \dot{\boldsymbol{X}}_1 &= \boldsymbol{X}_2, \\ \dot{\boldsymbol{X}}_2 &= \boldsymbol{g}(\bar{\boldsymbol{X}}, t)\boldsymbol{U} + \boldsymbol{\phi}(\bar{\boldsymbol{X}})\boldsymbol{\theta}(t) + \boldsymbol{f}(\bar{\boldsymbol{X}}) + \boldsymbol{\Delta}(t), \end{aligned} \tag{2}$$

where $\bar{\boldsymbol{X}} = \{\boldsymbol{X}_1, \boldsymbol{X}_2\}$; $\boldsymbol{g} \in \mathbb{R}^{n\times n}$ is the input-multiplied unknown nonlinear functions; $\boldsymbol{\theta} \in \mathbb{R}^l$ represents the vector of unknown linear system parameters with $l \in \mathbb{N}^+$; $\boldsymbol{\phi} \in \mathbb{R}^{n\times l}$ is the known nonlinear regressors matrix; $\boldsymbol{f} \in \mathbb{R}^n$ is the vector of known nonlinear functions; and vector $\boldsymbol{\Delta} = [\Delta_1, \ldots, \Delta_n]^\intercal \in \mathbb{R}^n$ represents the external disturbances and unmodeled system nonlinearities. The controller objective is described by following the reference command $\boldsymbol{X}_{1\mathrm{d}}(t) = [x_{1\mathrm{d}}, \ldots, x_{n\mathrm{d}}]^\intercal$.

Assumption 1 It is assumed that $\boldsymbol{X}_2$ is bounded, which implies that the functions $\boldsymbol{g}$, $\boldsymbol{\phi}$, and $\boldsymbol{f}$ are bounded along with their first- and second-order derivatives with respect to $\boldsymbol{X}_2$.

Assumption 2 $\boldsymbol{X}_{1\mathrm{d}}(t)$ is smooth and bounded up to the second order.

Assumption 3 The unmodeled nonlinearities and external disturbances term in $\boldsymbol{\Delta}$, and the parametric uncertainties in $\boldsymbol{\theta}$ and $\boldsymbol{g}$ are restricted to bounded sets such that:

$$\begin{aligned} &\|\Delta\| \leq \bar{\Delta}, \\ &g_{ij,\min} \leq g_{ij} \leq g_{ij,\max},\ i, j = 1, \ldots, n, \\ &\theta_{i,\min} \leq \theta_i \leq \theta_{i,\max},\ i = 1, \ldots, l, \end{aligned} \tag{3}$$

where $\bar{\Delta}$ is a positive constant, and θ_i and g_{ij} represent the elements of $\boldsymbol{\theta}$ and $\boldsymbol{g}$, respectively.

With Assumptions 1, 2 and 3, we ensure that the states and signals in the considered system are bounded.

Remark 1 As a case-study for validating the hereby proposed algorithm, we consider multirotor UAVs, which can be modelled (after proper reduction) as second-order time-varying systems that fit the model presented in (2). This work offers a contribution for this type of systems, without claiming generalization to other system types.

3 Adaptive backstepping control system design

In this section, we design an adaptive backstepping control law for the second-order nonlinear time-varying system defined in Sect. 2 in two steps with virtual inputs. The control law incorporates a model-based parameter estimation scheme and a robust feedback component to improve its transient response, steady-state accuracy, and robustness. This section serves as an overview of the adaptive backstepping control system design process leading to the similarity with PID control laws.

3.1 Parameter estimation law

The achievement of desirable tracking performance depends on the knowledge of the system parameters, which cannot be guaranteed using the nominal values of $\boldsymbol{\theta}(t)$ and $\boldsymbol{g}(\bar{\boldsymbol{X}}, t)$. Hence, the control law should employ their estimates, $\hat{\boldsymbol{\theta}}(t)$ and $\hat{\boldsymbol{g}}(t)$, instead of their unknown (true) values. The parameter estimation algorithm relies on a recursive least squares (RLS) adaptation law with projection mapping to provide a model-based estimation of the system's unknown parameters, which tend to be faster and more accurate than error-based estimates. Since this is outside the scope of this paper, readers are referred to [23,24] for details on the RLS estimator design. The parameter estimation vector includes $\boldsymbol{\theta}$ and the linear combination of unknown parameters in the term $\boldsymbol{g}(\bar{\boldsymbol{X}}, t)\boldsymbol{U}$. Note that $\hat{\boldsymbol{g}}$ is conditioned to remain invertible.

3.2 Controller design

In this section, and for the sake of notation simplicity and compactness, we do not explicitly express the in-brackets vector dependencies, that we use $\boldsymbol{g}$ instead of $\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{t})$. Hence, the error dynamics vector is given by:

$$\boldsymbol{e}_1 = \boldsymbol{X}_1 - \boldsymbol{X}_{1\mathrm{d}}. \tag{4}$$

Two steps are required in the backstepping control design process for second-order systems. In the first step, we propose a candidate Lyapunov function, $\mathcal{V}_1$, based on the error dynamics, and its derivative $\dot{\mathcal{V}}_1$, to be:

$$\begin{aligned} \mathcal{V}_1 &= \frac{1}{2}\boldsymbol{e}_1^\intercal \boldsymbol{e}_1, \\ \dot{\mathcal{V}}_1 &= \boldsymbol{e}_1^\intercal \dot{\boldsymbol{e}}_1 = \boldsymbol{e}_1^\intercal(\dot{\boldsymbol{X}}_1 - \dot{\boldsymbol{X}}_{1\mathrm{d}}), \end{aligned} \tag{5}$$

where $\boldsymbol{e_1}$ can be stabilized by employing a virtual control input, $\boldsymbol{\alpha} \in \mathbb{R}^n$, that substitutes $\dot{\boldsymbol{X}}_1$, as follows:

$$\boldsymbol{\alpha} = \dot{\boldsymbol{X}}_{1\mathrm{d}} - \boldsymbol{k}_1\boldsymbol{e}_1, \tag{6}$$

where $\boldsymbol{k}_1 \in \mathbb{R}^{n\times n}_{>0}$ is a diagonal matrix. In step two of the backstepping controller design, we define the second error term relating the state vector derivative to the virtual input as follows:

$$\boldsymbol{e_2} = \boldsymbol{X}_2 - \boldsymbol{\alpha}. \tag{7}$$

The error dynamics are obtained by differentiating (7) and substituting $\dot{\boldsymbol{X}}_2$ from (2):

$$\dot{\boldsymbol{e}}_2 = \boldsymbol{f} + \boldsymbol{\phi}\boldsymbol{\theta} + \boldsymbol{g}\boldsymbol{U} + \boldsymbol{\Delta} - \dot{\boldsymbol{\alpha}}, \tag{8}$$

which explicitly includes the control input vector $\boldsymbol{U}$. However, due to the variable system parameters and the presence of unknown external disturbances, (8) is expressed in the following form:

$$\dot{\boldsymbol{e}}_2 = \boldsymbol{f} + \boldsymbol{\phi}\hat{\boldsymbol{\theta}} + \hat{\boldsymbol{g}}\boldsymbol{U} + \boldsymbol{d} - \dot{\boldsymbol{\alpha}}, \tag{9}$$

where $\boldsymbol{d} = [d_1, \ldots, d_n]^\intercal$ lumps the errors in modeling and estimation with the external disturbances. The term $\boldsymbol{d}$ is split into two components: one for low-frequency signals, $\boldsymbol{d}_\mathrm{c}$, and the other for high-frequency signals, $\tilde{\boldsymbol{d}}^*$ [23]:

$$\boldsymbol{d} := \boldsymbol{d}_\mathrm{c} + \tilde{\boldsymbol{d}}^* = -\tilde{\boldsymbol{g}}\boldsymbol{U} - \boldsymbol{\phi}\tilde{\boldsymbol{\theta}} + \boldsymbol{\Delta}, \tag{10}$$

where $\tilde{\boldsymbol{g}} = \hat{\boldsymbol{g}} - \boldsymbol{g}$ and $\tilde{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}$.

Remark 2 The system disturbances and uncertainties are bounded as per Assumption 3, such that $\|\boldsymbol{d}\| \leq \bar{d}$, with $\bar{d} \in \mathbb{R}_{\geq 0}$. Additionally, vector $\boldsymbol{\Delta}$ includes both disturbances and unmodeled nonlinearities, thus it can be split as per (10) into low-frequency and high-frequency terms: the low-frequency terms can be compensated for via an integrator, or they can be included in a parameter estimation scheme if they were structured; and high-frequency terms are handled by a robust control component [22] as shown next.

Finally, the proposed nominal control law possesses the form:

$$\begin{aligned} \boldsymbol{U} &= \hat{\boldsymbol{g}}^{-1}(\dot{\boldsymbol{\alpha}} - \boldsymbol{\phi}\hat{\boldsymbol{\theta}} - \boldsymbol{f} - \boldsymbol{e}_1 - \boldsymbol{k}_2\boldsymbol{e}_2 + \boldsymbol{u}_\mathrm{r} - \hat{\boldsymbol{d}}_\mathrm{c}), \\ \boldsymbol{u}_\mathrm{r} &= -\frac{1}{4\epsilon}h^2\boldsymbol{e}_2, \qquad \dot{\hat{\boldsymbol{d}}}_\mathrm{c} = \mathrm{Proj}_{\hat{\boldsymbol{d}}_\mathrm{c}}(\boldsymbol{\gamma}\boldsymbol{e}_2), \end{aligned} \tag{11}$$

where $\epsilon \in \mathbb{R}_{>0}$ is a design parameter that influences the error attenuation level, steady-state accuracy, and transient response of the tracking error; and h is any smooth function that guaranties the following [25]:

$$h \geq \|\boldsymbol{\phi}\| \, \|\boldsymbol{\theta}_{\max} - \boldsymbol{\theta}_{\min}\| + \|\boldsymbol{U}\| \, \left\|\boldsymbol{g}_{\max} - \boldsymbol{g}_{\min}\right\| + \left\|\bar{\boldsymbol{\Delta}}\right\|, \tag{12}$$

$\mathrm{Proj}_{\hat{\boldsymbol{d}}_\mathrm{c}}(\cdot) = [\mathrm{Proj}_{\hat{d}_{\mathrm{c}1}}(\cdot_1), \ldots, \mathrm{Proj}_{\hat{d}_{\mathrm{c}n}}(\cdot_n)]^\intercal$, and $\boldsymbol{\gamma} \in \mathbb{R}^{n\times n}_{\geq 0}$ is a diagonal matrix that adjusts the adaptation rate. Note that $\mathrm{Proj}_{\hat{\boldsymbol{d}}}(\cdot)$ limits $\hat{\boldsymbol{d}}_\mathrm{c}$ to the set $\Omega_{\boldsymbol{d}_\mathrm{c}} \triangleq \{\hat{\boldsymbol{d}}_\mathrm{c} : -\bar{d} \leq \hat{\boldsymbol{d}}_\mathrm{c} \leq \bar{d}\}$.

Assumption 4 It is safe to assume that $\dot{\boldsymbol{d}}_{\mathrm{c}} = 0$ given that $\boldsymbol{d}_{\mathrm{c}}$ includes low-frequency signals only.

Theorem 1 *Consider the system model described in (2) that is subject to unknown disturbances, and whose fidelity is affected by unmodeled nonlinearities and parametric uncertainties. Also, let Assumptions 1, 2, 3, and 4 hold true. The stability of this system and the boundedness of its signals are guaranteed by the control law (11), with $\boldsymbol{\alpha}$ defined in (6) and $\boldsymbol{e}_2$ defined in (7), for some gains $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$ [17].*

Proof We start by defining the augmented candidate Lyapunov function, $\mathcal{V}_2$, differentiating it once with respect to time, then substituting $\boldsymbol{d}$ from (10) and $\boldsymbol{U}$ from (11) into (9) to obtain:

$$
\begin{aligned}
\mathcal{V}_2 &= \frac{1}{2}\boldsymbol{e}_1^{\mathsf{T}}\boldsymbol{e}_1 + \frac{1}{2}\boldsymbol{e}_2^{\mathsf{T}}\boldsymbol{e}_2 + \frac{1}{2}\tilde{\boldsymbol{d}}_{\mathrm{c}}^{\mathsf{T}}\boldsymbol{\gamma}^{-1}\tilde{\boldsymbol{d}}_{\mathrm{c}},\\
\dot{\mathcal{V}}_2 &= -\boldsymbol{e}_1^{\mathsf{T}}\boldsymbol{k}_1\boldsymbol{e}_1 - \boldsymbol{e}_2^{\mathsf{T}}\boldsymbol{k}_2\boldsymbol{e}_2 + \tilde{\boldsymbol{d}}_{\mathrm{c}}^{\mathsf{T}}\boldsymbol{\gamma}^{-1}\dot{\hat{\boldsymbol{d}}}_{\mathrm{c}}\\
&\quad + \boldsymbol{e}_2^{\mathsf{T}}(\boldsymbol{d}_{\mathrm{c}} - \hat{\boldsymbol{d}}_{\mathrm{c}} + \tilde{\boldsymbol{d}}^{*} + \boldsymbol{u}_{\mathrm{r}}),
\end{aligned}
\tag{13}
$$

where $\tilde{\boldsymbol{d}}_{\mathrm{c}} = \hat{\boldsymbol{d}}_{\mathrm{c}} - \boldsymbol{d}_{\mathrm{c}}$, and $\dot{\boldsymbol{d}}_{\mathrm{c}} = 0$ per Assumption 4.

By substituting $\dot{\hat{\boldsymbol{d}}}_{\mathrm{c}}$ from (11) in (13), we get: $\dot{\mathcal{V}}_2 = -\boldsymbol{e}_1^{\mathsf{T}}\boldsymbol{k}_1\boldsymbol{e}_1 - \boldsymbol{e}_2^{\mathsf{T}}\boldsymbol{k}_2\boldsymbol{e}_2 + \boldsymbol{e}_2^{\mathsf{T}}(\tilde{\boldsymbol{d}}^{*} + \boldsymbol{u}_{\mathrm{r}})$.

In addition, two robust performance conditions are to be satisfied by the robust feedback term, $\boldsymbol{u}_{\mathrm{r}}$, such that [26]:

$$
\begin{aligned}
&\boldsymbol{e}_2^{\mathsf{T}}\boldsymbol{u}_{\mathrm{r}} \le 0,\\
&\boldsymbol{e}_2^{\mathsf{T}}(\boldsymbol{u}_{\mathrm{r}} - \tilde{\boldsymbol{d}}_{\mathrm{c}} + \tilde{\boldsymbol{d}}^{*}) \le \boldsymbol{\epsilon},
\end{aligned}
\tag{14}
$$

which is achieved by choosing $\boldsymbol{u}_{\mathrm{r}}$ as described in (11), and robust performance under the prescribed conditions can be guaranteed [26]. This results in $\dot{\mathcal{V}}_2$ being negative semi-definite, thus stability in Lyapunov's sense is concluded. It can also be shown that $\mathcal{V}_2$ is lower bounded and that $\dot{\mathcal{V}}_2$ is uniformly continuous by Assumptions 1 and 3; therefore, by Barbalat's lemma, the asymptotic convergence of $\mathcal{V}_2$ to zero is obtained [27]. □

4 From backstepping to PID form

In this section, we formulate the backstepping control law in a PID feedback plus feedforward form that has two degrees-of-freedom (2-DOF), then we present a visualization of the obtained relations with ample discussion and practical tips to design and tune the controller gains, which justifies the adoption of the proposed form of the control law. Finally, we relate the backstepping gains to the location of the closed-loop system poles.

4.1 Expressing the backstepping control law in PID form

First, we seek to reformulate and simplify the control law while maintaining its practical results. In practice, a large-enough positive constant, $\boldsymbol{k}_{2,0}$, can produce robust performance that is similar to that of the nonlinear robust gain, $\frac{1}{4\epsilon}h^2$, in (11), such that:

$$
\boldsymbol{k}_{2,0,ii} > n_{\mathrm{r}}\frac{1}{4\epsilon}h^2, \quad i = 1, \dots, n,
\tag{15}
$$

where $n_{\mathrm{r}} > 1$ is a constant multiplier. This causes the effect of $\boldsymbol{u}_{\mathrm{r}}$ to be similar to that of the linear term $-\boldsymbol{k}_2\boldsymbol{e}_2$. Thus, by choosing the elements of matrix $\boldsymbol{k}_2$ to be greater than some positive constants ($\boldsymbol{k}_2 \succ_{\mathbb{R}^{n\times n}_{\ge 0}} \boldsymbol{k}_{2,0}$), robust performance can be guaranteed within a large-enough and practical operating range [22,28]. With this modification, the control law (11) is approximated as $\boldsymbol{U} \approx \boldsymbol{U}'$, where $\boldsymbol{U}'$ approximates the effect of the nonlinear robust term, $\boldsymbol{u}_{\mathrm{r}}$, by the high gain feedback term, $-\boldsymbol{k}_2\boldsymbol{e}_2$, and can be reformulated by substituting (4), (6), and (7) into (11):

$$
\begin{aligned}
\boldsymbol{U}' &= \hat{\boldsymbol{g}}^{-1}\big[(\ddot{\boldsymbol{X}}_{1\mathrm{d}} - \boldsymbol{k}_1\dot{\boldsymbol{e}}_1) - \boldsymbol{\phi}\hat{\boldsymbol{\theta}} - \boldsymbol{f} - \boldsymbol{e}_1\\
&\qquad - \boldsymbol{k}_2(\dot{\boldsymbol{X}}_1 - (\dot{\boldsymbol{X}}_{1\mathrm{d}} - \boldsymbol{k}_1\boldsymbol{e}_1)) - \hat{\boldsymbol{d}}_{\mathrm{c}}\big]\\
&= \hat{\boldsymbol{g}}^{-1}\big[-(\boldsymbol{k}_1\boldsymbol{k}_2 + \boldsymbol{I}_n)\boldsymbol{e}_1 - (\boldsymbol{k}_1 + \boldsymbol{k}_2)\dot{\boldsymbol{e}}_1 - \hat{\boldsymbol{d}}_{\mathrm{c}}\\
&\qquad - \boldsymbol{\phi}\hat{\boldsymbol{\theta}} - \boldsymbol{f} + \ddot{\boldsymbol{X}}_{1\mathrm{d}}\big],\\
\dot{\hat{\boldsymbol{d}}}_{\mathrm{c}} &= \mathrm{Proj}_{\hat{\boldsymbol{d}}_{\mathrm{c}}}\big(\boldsymbol{\gamma}\big[\dot{\boldsymbol{X}}_1 - (\dot{\boldsymbol{X}}_{1\mathrm{d}} - \boldsymbol{k}_1\boldsymbol{e}_1)\big]\big)\\
&= \mathrm{Proj}_{\hat{\boldsymbol{d}}_{\mathrm{c}}}\big(\boldsymbol{\gamma}(\dot{\boldsymbol{e}}_1 + \boldsymbol{k}_1\boldsymbol{e}_1)\big),
\end{aligned}
\tag{16}
$$

where $\dot{\boldsymbol{e}}_1 = \dot{\boldsymbol{X}}_1 - \dot{\boldsymbol{X}}_{1\mathrm{d}}$ and $\boldsymbol{I}_n \in \mathbb{R}^{n\times n}$ is the identity matrix. Thus, (16) is now expressed as a 2-DOF controller with a feedforward model compensation component and a feedback PID component, designated as the Integrated Control System (ICS):

$$
\begin{aligned}
\boldsymbol{U}' &= \hat{\boldsymbol{g}}^{-1}\big[-\boldsymbol{k}_{\mathrm{P}}\boldsymbol{e}_1 - \boldsymbol{k}_{\mathrm{I}}\boldsymbol{e}_1^{\mathrm{I}} - \boldsymbol{k}_{\mathrm{D}}\dot{\boldsymbol{e}}_1\\
&\qquad - \boldsymbol{\phi}\hat{\boldsymbol{\theta}} - \boldsymbol{f} + \ddot{\boldsymbol{X}}_{1\mathrm{d}}\big],\\
\dot{\boldsymbol{e}}_1^{\mathrm{I}} &= \mathrm{Proj}_{\boldsymbol{e}_1^{\mathrm{I}}}(\boldsymbol{e}_1 + \boldsymbol{k}_1^{-1}\dot{\boldsymbol{e}}_1),
\end{aligned}
\tag{17}
$$

where $\boldsymbol{e}_1^{\mathrm{I}} = \int \dot{\boldsymbol{e}}_1^{\mathrm{I}}dt$ and the bounds of the projection function vary accordingly as per (1) by a factor $(\boldsymbol{\gamma}\boldsymbol{k}_1)^{-1}$, and

$$
\begin{aligned}
\boldsymbol{k}_{\mathrm{P}} &= \boldsymbol{k}_1\boldsymbol{k}_2 + \boldsymbol{I}_n,\\
\boldsymbol{k}_{\mathrm{I}} &= \boldsymbol{\gamma}\boldsymbol{k}_1,\\
\boldsymbol{k}_{\mathrm{D}} &= \boldsymbol{k}_1 + \boldsymbol{k}_2.
\end{aligned}
\tag{18}
$$

The control law (17), with its six terms, can now be seen as a combination of an error-based PID feedback component with gains $\boldsymbol{k}_{\mathrm{P}}$, $\boldsymbol{k}_{\mathrm{I}}$, and $\boldsymbol{k}_{\mathrm{D}}$ that are included in the first three terms, and a feedforward component based on model compensation desired accelerations that are included in the last three terms.

When operating in the nominal region, it is safe to assume that saturation of the integral term does not occur, which permits bypassing the projection function representation, so that after integrating the term $\boldsymbol{k}_1^{-1}\dot{\boldsymbol{e}}_1$, the control law in (17) and the control gains in (18) undergo the following modifications:

$$\begin{aligned} \dot{\boldsymbol{e}}_1^{\mathrm{I}} &= \boldsymbol{e}_1, \\ \boldsymbol{k}_{\mathrm{P}} &= \boldsymbol{k}_1\boldsymbol{k}_2 + \boldsymbol{\gamma} + \boldsymbol{I}_n. \end{aligned} \tag{19}$$

Let the diagonal elements of an arbitrary single row of matrices $\boldsymbol{k}_{\mathrm{P}}$, $\boldsymbol{k}_{\mathrm{I}}$, $\boldsymbol{k}_{\mathrm{D}}$, $\boldsymbol{k}_1$, $\boldsymbol{k}_2$, and $\boldsymbol{\gamma}$ be represented by k_{P}, k_{I}, k_{D}, k_1, k_2, and γ, respectively. By solving the following third-order equation, which is derived from the definition of $\boldsymbol{k}_{\mathrm{P}}$ in (19) and that of $\boldsymbol{k}_{\mathrm{I}}$ and $\boldsymbol{k}_{\mathrm{D}}$ in (18), for some desired values of k_{P}, k_{I}, and k_{D}, we can determine the set of equivalent k_1, k_2, and γ gains as follows:

$$\begin{aligned} &k_1^3 + (-k_{\mathrm{D}})k_1^2 + (k_{\mathrm{P}} - 1)k_1 - k_{\mathrm{I}} = 0, \\ &k_2 := k_{\mathrm{D}} - k_1, \\ &\gamma \ \ := k_{\mathrm{I}}/k_1, \end{aligned} \tag{20}$$

where the solution is valid only when k_1 and k_2 are positive and real. To examine the behavior of (20), we present a sample solution with respect to k_{P} for $k_{\mathrm{D}} = 5$ and $k_{\mathrm{I}} = 1$ in Fig. 1a, and with respect to k_{D} for $k_{\mathrm{P}} = 5$ and $k_{\mathrm{I}} = 1$ in Fig. 1b. Note that the solutions for k_1 (red), k_2 (green), and γ (blue) in each subplot is divided into three subsets, based on their change of direction at the inflection points, as indicated by their line style (Subset #1-solid, Subset #2-dashed, and Subset #3-dotted) to simplify the analysis and arrive at practical recommendations:

1. Subset #1 (solid lines) has a parabola-like shape for k_1 and k_2 in Fig. 1a, and a hyperbola-like shape in Fig. 1b, which result in a wide range of variation for these two gains and thus offers more leverage in tuning them to meet the system's stability and performance requirements. On the other hand, the value and variation of γ are minimal (near zero) as compared to Subsets #2 and #3.
2. Subset #2 (dashed lines) corresponds to the smallest domain of the solution space and features large variation of all three parameters (k_1, k_2, and γ). The strong dependencies amongst the parameters, combined with their large variations in this relatively small subset, make it difficult and impractical to interpret the relationship between the backstepping and PID gains within Subset #2. In fact, any formulated relationship is of a limited merit since it encompasses a very small domain of coupled k_{P} and k_{D} values.

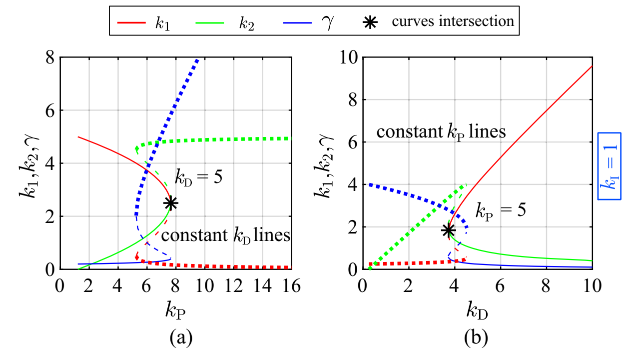


Fig. 1 Coupled backstepping gains, k_1, k_2, and γ, for specific k_{P}, k_{D}, and k_{I} values: (a) at constant k_{D} and (b) at constant k_{P}. Three solution subsets exist: Subset #1 - solid lines, Subset #2 - dashed lines, and Subset #3 - dotted lines

3. Subset #3 (dotted lines) is characterized by large γ values and a narrow tuning range (nearly constant) for both k_1 (near zero) and k_2 as compared to Subsets #1 and #2, which limits the ability to meet the stability and performance requirements of the system.

Based on the above analysis of Fig. 1, it is prudent to only consider the first subset of the solution of (20) given its low dependency on γ, high dependency on k_1 and k_2, and the relatively larger tuning range to achieve the desired system stability and performance requirements. The recommendation to adopt Subset #1 is further justified in Sect. 4.4.

Remark 3 The differentiated term $\dot{\boldsymbol{e}}_1$ typically requires filtration in order to reduce the signal's noise level. However, $\dot{\boldsymbol{e}}_1$ is implicitly embedded within $\dot{\boldsymbol{\alpha}}$ and $\boldsymbol{e}_2$ of the control law (11), whereas the PID-like form of the control law in (17) explicitly shows it, which makes it easier to design an appropriate filter with minimal time delay and information loss that may arise form excessive filtration of $\dot{\boldsymbol{\alpha}}$ and $\boldsymbol{e}_2$ [10].

Remark 4 By inspecting (18), (19), and (20), we see that k_1 and k_2 can be used interchangeably. This is achieved by solving for k_2 first instead of k_1 as was done in (20):

$$\begin{aligned} &k_2^3 + (-k_{\mathrm{D}})k_2^2 + (k_{\mathrm{P}} - 1)k_2 - k_{\mathrm{I}} = 0, \\ &k_1 := k_{\mathrm{D}} - k_2, \\ &\gamma \ \ := k_{\mathrm{I}}/(k_{\mathrm{D}} - k_1). \end{aligned} \tag{21}$$

Note that the results in (21) are identical to those in (20), with the only exception being that k_1 and k_2 values have been switched. This can be proven by substituting the interchanged values of k_1 and k_2 into the definition k_{D} in (18) and the definition k_{P} in (19), where the same exact control law in (17) can be obtained by setting:

$$k_{\mathrm{I}} := \gamma(k_{\mathrm{D}} - k_1). \tag{22}$$

Remark 5 As seen in Fig. 1, k_1 is larger than k_2 in the gains solution subset of interest (#1), in which k_2 can have very small values. However, based on Remark 4, k_2 can be the larger gain in the $\{k_1, k_2\}$ solution set, thus the condition for large enough k_2 gain to overtake the effect of the robust gain $(\frac{1}{4\epsilon}h^2)$ remains valid.

4.2 Comparison to other formulations

To help differentiate this work and to validate the proposed PID-like adaptive backstepping control law, Table 1 is provided to summarize previous works that are found in the literature dealing with the backstepping-PID gains similarity. We note that in order to get a clear comparison and arrive at a better interpretation of the results, the notations of the original formulations were mathematically manipulated to show that the various efforts lead to similar relationships, albeit with minor differences. Note that in the PID gains column, the gains $k_i, i = \{1, 2, 3\}$ refer to the Lyapunov function gains, the gains c_e and c_i relate to the integral action within the control law, and the other parameters relate to the system model.

In this work, the similarity formulation to the PID gains is simpler in the sense that it is established based on three parameters only as shown in Table 1, whereas the integral backstepping representations in [15] use four parameters (k_1,k_2, c_e, and c_i) and two additional model-dependent constants (a_0 and a_1) in [13], respectively. Furthermore, the number of dependencies between the adaptive backstepping gains and the PID gains in this work is also minimal, where we use only one multiplication and one summation as compared to [6,13], which use up to two multiplications and two summations. Therefore, an advantage of the formulation presented in this work over other representations lies in the fact that it provides a simpler method to design and tune backstepping controllers based on PID laws and tuning rules, which makes it more practical and easier to adopt in real-life applications.

4.3 Implementation of the backstepping-PID gains relationship

An implementation of (20) for four preset values of k_I (k_I=0,1,2,4) is shown in Fig. 2, where Fig. 2a on the left shows the backstepping gains, k_1 and k_2, as a function of k_P with a constant k_D condition, and Fig. 2b on the right shows the gains as a function of k_D with a constant k_P condition. For every streamline marking a constant k_D or k_P, the black asterisks mark the intersections of the k_1 and k_2 curves as applicable, which correspond to the maximum k_P value in Fig. 2a and the minimum k_D value in Fig. 2b. Note that γ is not plotted on the same figure in order to maintain a clear and legible representation of the analysis results, noting that

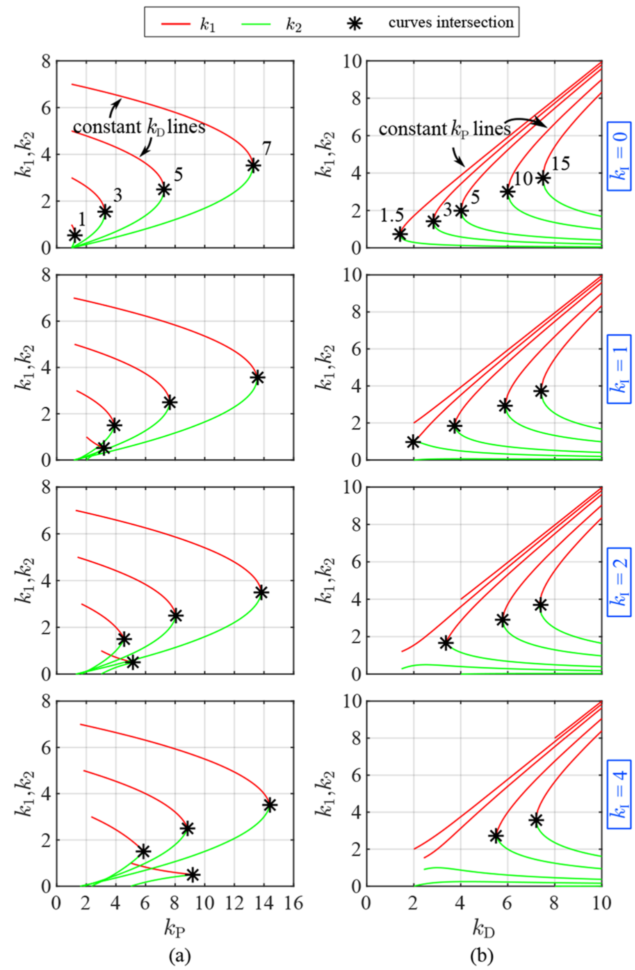


Fig. 2 Coupled backstepping gains, k_1 and k_2, as a function of k_P and k_D at constant k_I values ($k_I = 0, 1, 2, 4$ corresponding to four rows). The left column in (a) shows the backstepping gains at constant k_D and k_I, and the right column in (b) shows the backstepping gains at constant k_P and k_I

it can be directly calculated from (20) and is perceived as a scaled version of the inverse of k_1 for each corresponding plot.

The significance of the above results lies in making the design and tuning process of the backstepping-based ICS gains easier by allowing the application of well-established PID tuning rules. Additionally, by excluding the unfeasible PID gains that result in non-positive or non-real k_1 and k_2 roots, the tuning interval is tightened. Furthermore, the formulation provides insight into both the traditional stability perspective as well as the performance perspective of the adaptive backstepping control law.

Tuning each one of the gains k_1 and k_2 has a dual effect on both k_P and k_D, as illustrated in Fig. 2, yet it does not influence the system performance directly as typically expected from tuning PID gains, which can make the tuning process cumbersome and not straightforward. For instance, increas-

Table 1 Backstepping-PID relationships in the literature

Reference	Control method	System type	Control law	PID gains
[6]	Backstepping with integral action	Minimal phase linear system: $G(s)=\frac{Y(s)}{U(s)}=\frac{B(s)}{A(s)}=\frac{s+b_0}{s^2+a_1s+a_0}$	$U(s)=\frac{k_\mathrm{D}s+k_\mathrm{I}/s+k_\mathrm{P}}{B(s)}E(s)+\frac{A(s)}{B(s)}Y_d(s)$	$k_\mathrm{P}=k_1k_2+k_3(k_1+k_2)+2-a_0,$ $k_\mathrm{I}=k_1k_2k_3+k_1+k_3,$ $k_\mathrm{D}=k_1+k_2+k_3-a_1$.
[13]	Backstepping	Second-order system: $\ddot{x}=b_0u(t)-a_1\dot{x}-a_0x$	$u=\frac{1}{b_0}\big(-k_\mathrm{P}e-k_\mathrm{I}\int edt-k_\mathrm{D}\dot{e}+a_0x_\mathrm{d}+a_1\dot{x}_\mathrm{d}+\ddot{x}_\mathrm{d}\big)$	$k_\mathrm{P}=k_1k_2+\frac{c_i}{c_e}(k_1+k_2)+c_e^2-a_0,$ $k_\mathrm{I}=k_1k_2\frac{c_i}{c_e}+c_ec_i,$ $k_\mathrm{D}=k_1+k_2+\frac{c_i}{c_e}-a_1$.
[14]	Integral Backstepping	Nonlinear mechanical systems of degree 2: $\dot{x}_1=f_1(x_1)+g_1(x_1)x_2$ $\dot{x}_2=f_2(\bar{x})+g_2(\bar{x})u+b$	$u=g_2^{-1}\big(-k_\mathrm{P}e-k_\mathrm{I}\int edt-k_\mathrm{D}[x_2-g_1^{-1}(\dot{x}_{1\mathrm{d}}-f_1)]+\alpha_1^{x_1}[f_1+g_1x_2]+\alpha_1^t-f_2\big),$ $\alpha_1=g_1^{-1}(-c_i\int edt-k_1e-f_1+\dot{x}_{1\mathrm{d}})$	$k_\mathrm{P}=k_2g_1^{-1}k_1+g_1^{-1}c_i+g_1^\mathsf{T},$ $k_\mathrm{I}=k_2g_1^{-1}c_i,$ $k_\mathrm{D}=k_2$.
[15]	Integral Backstepping	Quadrotor rotational subsystem: $\dot{x}_1=x_2$ $\dot{x}_2=f+gu$	$u=g^{-1}(-k_\mathrm{P}e-k_\mathrm{I}\int edt-k_\mathrm{D}\dot{e}+\ddot{y}_d-f)$	$k_\mathrm{P}=k_1k_2+\frac{c_i}{c_e}(k_1+k_2)+c_e^2,$ $k_\mathrm{I}=k_1k_2\frac{c_i}{c_e}+c_ec_i,$ $k_\mathrm{D}=k_1+k_2+\frac{c_i}{c_e}$.
[16]	Nonlinear Backstepping	Quadrotor rotational subsystem: $\dot{x}_1=x_2$ $\dot{x}_2=f+gu$	$u=g^{-1}(-k_\mathrm{P}e-k_\mathrm{I}\int edt-k_\mathrm{D}\dot{e}+\ddot{x}_{1\mathrm{d}})$	$k_\mathrm{P}=(k_1k_2+k_3)/mg,$ $k_\mathrm{I}=(k_1k_3)/mg,$ $k_\mathrm{D}=(mk_1+k_2)/mg$.
This work	Integral Backstepping	A class of nonlinear systems of degree 2: $\dot{x}_1=x_2$ $\dot{x}_2=f(\bar{x})+\phi(\bar{x})\theta(t)+g(\bar{x},t)u+\Delta(t)$	$u=g^{-1}(-k_\mathrm{P}e-k_\mathrm{I}\int edt-k_\mathrm{D}\dot{e}+\ddot{x}_{1\mathrm{d}}-f-\phi\hat{\theta})$	$k_\mathrm{P}=k_1k_2+\gamma+1,$ $k_\mathrm{I}=\gamma k_1,$ $k_\mathrm{D}=k_1+k_2$.

ing any of the backstepping gains k_1 and k_2 will also result in an increase in both k_P and k_D simultaneously, and similarly, tuning γ affects both k_P and k_I.

Remark 6 To provide a more detailed analysis, we introduce the following notes to guide the tuning process based on the results obtained in Fig. 2:

- For a given value of k_D, the k_P value that attains the desired performance and stability margin can be obtained from Fig. 2a.
- For a given value of k_P, the k_D value that attains the desired performance and stability margin can be obtained from Fig. 2b.
- As the value of k_I increases, the shape of the solutions starts to deviate from the uniform parabola and hyperbola contours.
- Since k_1 and k_2 must be positive per Lyapunov stability requirements, a streamline stops when one of its two branches reaches the abscissa.
- The dual point (asterisk) can be the optimal solution in terms of stability, for some preset values of k_P or k_D.
- k_D linearly relates to the sum of k_1 and k_2, which manifests in Fig. 2b as all solution sets (k_1 and k_2) are below the identity line (unity slope), and their sum is equal to it ($k_1 + k_2 = 1$). As such, one can promptly compute the value of k_2 in Fig. 2b for any given k_1 based on (20).
- While other solutions besides the ones presented in Fig. 2 may exist (as per Fig. 1), they are disregarded due to their irrelevance to this work.

4.4 Simplified backstepping-PID relationships

If (20) has a weak dependence on γ, then by referring to (19) and (20), the following inequality holds:

$$k_1 k_2 + 1 >> k_\mathrm{I}/k_1, \tag{23}$$

and (20) can be simplified into the following second-order form:

$$\begin{aligned}
&k_{1,2}^2 + (-k_\mathrm{D})k_{1,2} + (k_\mathrm{P} - \gamma - 1) = 0,\\
&k_1 \cong \max\{k_{1,2}\},\\
&k_2 \cong \min\{k_{1,2}\},\\
&\gamma \ \cong k_\mathrm{I}/k_1,\\
&\text{such that:}\quad k_{1,2} > 0.
\end{aligned} \tag{24}$$

As a rule of thumb, the condition in (23) holds if:

$$k_\mathrm{P} > 10\, k_\mathrm{I}/k_\mathrm{D}.$$

Hence, the general solution of the second order polynomial, $k_{1,2}$, is given by:

$$k_{1,2} = \frac{k_\mathrm{D}}{2} \pm \sqrt{(\frac{k_\mathrm{D}}{2})^2 - (k_\mathrm{P} - \gamma - 1)}. \tag{25}$$

Note that k_1 and k_2 are directly obtained from solving the polynomial in (24), unlike in (20) where only k_1 is obtained. To further analyze (24) and determine the behavior of $k_{1,2}$ when k_P and k_D vary, the following analysis based on conic representation is introduced.

The general second-order equation of conics is of the form:

$$\kappa_1 \mathrm{x}_1^2 + \kappa_2 \mathrm{x}_1 \mathrm{x}_2 + \kappa_3 \mathrm{x}_2^2 + \kappa_4 \mathrm{x}_1 + \kappa_5 \mathrm{x}_2 + \kappa_6 = 0. \tag{26}$$

where x_1 and $\mathrm{x}_2 \in \mathbb{R}$ are the variables of the equation, and $\kappa_i \in \mathbb{R}, i = \{1, \ldots, 6\}$ are constants [29]. If $\kappa_2 \neq 0$, then the principle axes of the conic incur a counterclockwise rotation ϑ with respect to x_1, such that:

$$\cot 2\vartheta = \frac{\kappa_1 - \kappa_3}{\kappa_2}. \tag{27}$$

4.4.1 Conic representation with constant k_D

If (24) is solved for specific k_D values, as in Fig. 2a, we obtain:

$$\begin{aligned}
&\mathrm{x}_1 := k_{1,2}, \quad \mathrm{x}_2 := k_\mathrm{P} - \gamma, \quad \kappa_1 := 1, \quad \kappa_2 := 0,\\
&\kappa_3 := 0, \qquad \kappa_4 := -k_\mathrm{D}, \qquad \kappa_5 := 1, \quad \kappa_6 := -1.
\end{aligned}$$

Noticing that $\kappa_1 \neq 0$ and $\kappa_1 \kappa_3 = 0$, we conclude that (24) becomes a horizontal parabola that relates $k_{1,2}$ to k_P in the form:

$$\begin{aligned}
&\mathrm{x}_1 - \Big((\frac{k_\mathrm{D}}{2})^2 + 1 \Big) = -\Big(\mathrm{x}_2 - \frac{k_\mathrm{D}}{2} \Big)^2,\\
&\mathrm{V}_\mathrm{P} = \Big\{ (\frac{k_\mathrm{D}}{2})^2 + 1, \frac{k_\mathrm{D}}{2} \Big\},
\end{aligned} \tag{28}$$

where V_p is the parabola's vertex.

4.4.2 Conic representation with constant k_P

Similarly, (24) can be solved for preset k_P values, as in Fig. 2b, to relate $k_{1,2}$ to k_D, that is:

$$\begin{aligned}
&\mathrm{x}_1 := k_{1,2}, \quad \mathrm{x}_2 := k_\mathrm{D}, \quad \kappa_1 := 1, \quad \kappa_2 := -1,\\
&\kappa_3 := 0, \qquad \kappa_4 := 0, \qquad \kappa_5 := 0, \quad \kappa_6 := k_\mathrm{P} - \gamma - 1.
\end{aligned}$$

Noticing that $\kappa_2 \neq 0$, the rotation angle of the conic axes is calculated from (27) to be $\vartheta_\mathrm{h} = \frac{\pi}{8}$. We can then substitute

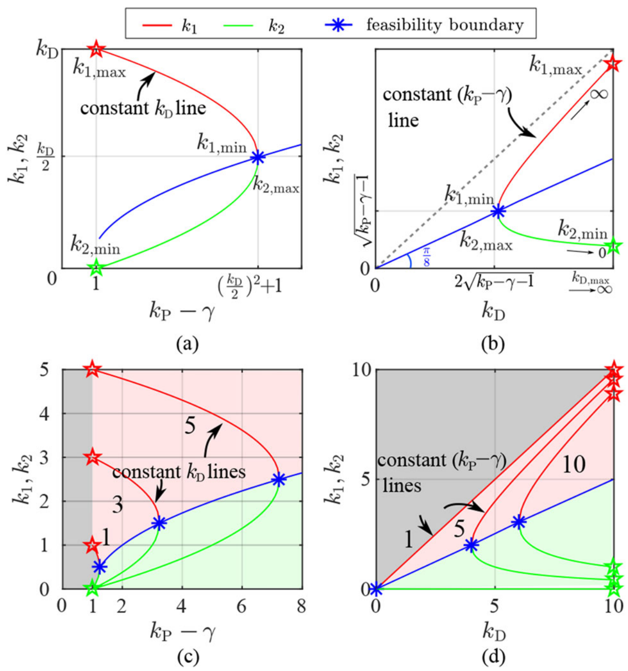


Fig. 3 Coupled backstepping gains, k_1 and k_2, in the simplified form, for specific k_P, k_I, and k_D values at constant k_D in (a) on the left, and at constant k_P in (b) on the right. The colored regions in red and green represent the sets of all feasible k_1 and k_2 combinations, respectively. The blue line marks the boundary between the two regions, and the grey region represents the set of unfeasible solutions. The colored stars mark the boundaries of the solution set

$x_1 = x'_1 c_{\vartheta_h} - x'_2 s_{\vartheta_h}$ and $x_2 = x'_1 s_{\vartheta_h} + x'_2 c_{\vartheta_h}$ in (24), where x'_1 and x'_2 are the principle axes of the rotated conic, to obtain:

$$\frac{x_1'^2}{a^2} - \frac{x_2'^2}{b^2} = 1, \qquad V_D = \left\{ a c_{\vartheta_h}, a s_{\vartheta_h} \right\}, \tag{29}$$

which represent a rotated horizontal hyperbola, with V_D being the hyperbola's vertex, $a = \sqrt{\frac{2(k_P-\gamma-1)}{\sqrt{2}-1}}$, and $b = \sqrt{\frac{2(k_P-\gamma-1)}{\sqrt{2}+1}}$.

However, the function of $k_{1,2}$ with respect to k_D is interpreted easier when it has the same form as (24), since we seek solutions for a single value of k_D one at a time. The double roots occur at V'_D such that:

$$V'_D = \left\{ 2\sqrt{k_P-\gamma-1}, \sqrt{k_P-\gamma-1} \right\}. \tag{30}$$

4.4.3 Interpretation

The resultant plots of the parabola and hyperbola formulations in (28) and (29) are shown in Fig. 3, which can be used to promptly and clearly interpret the relationship between the backstepping and the PID gains, while mainly benefiting from the vertices' locations. For instance, starting from an initial combination of k_1 and k_2, one can easily find the corresponding k_P, k_D, and k_I gains using Fig. 3; at the same time, one can navigate along the curves to arrive at the desired PID gains while keeping an eye on the resulting k_1 and k_2 gains. Note that the feasible k_1 and k_2 solutions are marked in red and green colors, respectively, while the unfeasible region is marked in grey.

To maximize the benefits of Fig. 3, we provide an explicit interpretation of its results:

- For some preset k_D, the maximum $(k_P - \gamma)$ value can be deduced from (28) as:

$$(k_P - \gamma)_{max} = \left(\frac{k_D}{2}\right)^2 + 1, \tag{31}$$

which corresponds to the last real solution (two identical roots) of (28):

$$k_{1,2} = \frac{k_D}{2}.$$

- For some preset $(k_P - \gamma)$, the minimum k_D value can be approximated from (30) as:

$$k_{D,min} = 2\sqrt{k_P - \gamma - 1}, \tag{32}$$

which corresponds to the first real solution (two identical roots) of (24):

$$k_{1,2} = \sqrt{k_P - \gamma - 1}.$$

- If it is desired to increase k_P while maintaining a constant k_D, k_1 should be decreased while k_2 should be increased.
- If it is desired to increase k_D while maintaining a constant k_P, k_1 should be increased while k_2 should be decreased.

A summary of stable tuning intervals of the PID gains as well as the backstepping gains intervals is presented in Table 2, which serves as a practical guide for the tuning process concerning the gains and stability bounds. In addition, a summary of the influence of tuning the PID gains on the backstepping gains, as well as their effect on the system performance is presented in Table 3.

Remark 7 Unlike the PID gains, a single backstepping gain variation, i.e. k_1, k_2, or γ, does not result in a unified effect on the system performance as per Table 3, which can be verified by referring to (18) and (19).

By combining the results of Fig. 3 and Table 3, and starting from an initial system performance, we can specify the desired performance modification, i.e. speed, stability, or overshoot, then decide on the PID gains tuning direction

Table 2 PD Gains Tuning Intervals and Effect on Backstepping Gains

	$(k_P-\gamma)_{\min}$	$(k_P-\gamma)_{\max}$	$k_{D,\min}$	$k_{D,\max}$	$k_{1,\min}$	$k_{1,\max}$	$k_{2,\min}$	$k_{2,\max}$
Constant k_D	1	$(\frac{k_D}{2})^2+1$	–	–	$\frac{k_D}{2}$	k_D	0	$\frac{k_D}{2}$
Constant k_P	–	–	$2\sqrt{k_P-\gamma-1}$	∞	$\sqrt{k_P-\gamma-1}$	∞	0	$\sqrt{k_P-\gamma-1}$

Table 3 Effects of Tuning PID Gains [30] and their Influence on the Backstepping Gains

Gain variation	k_1	k_2	γ	Speed	Stability	Overshoot
k_P increases	−	+	+	+	−	+
k_I increases	−	+	+	−	+	+
k_D increases	+	−	−	+	+	−

based on Table 3, then by referring to Fig. 3, the effect on the backstepping gains can be directly assessed.

Given the above discussion, and by perceiving the seemingly complex control law in (11) as a PID feedback plus feedforward terms as in (17), the ambiguity can be clarified by a direct visual interpretation that can be obtained from Fig. 3, thus reducing any complexity concerns that are typically associated with adopting nonlinear control laws.

4.5 From backstepping to pole placement

After substituting (17) and (19) in (8), the resulting error dynamics of the system become:

$$E_1\big(s^3+s^2(k_1+k_2)+s(k_1k_2+\gamma+1)+(\gamma k_1)\big)=D, \tag{33}$$

where s is the Laplace operator, and E_1 and D are the Laplace transforms of e_1 and Δ, respectively. In the PID representation, (33) becomes:

$$E_1\big(s^3+s^2k_D+sk_P+k_I\big)=D. \tag{34}$$

The error dynamics are expressed in the poles form to show the relationship between the PID gains and the locations of the closed-loop poles in the s-plane:

$$\begin{aligned}&E_1\big((s+p_1)(s+p_2)(s+p_3)\big)=D,\\&E_1\Big(s^3+s^2(p_1+p_2+p_3)+s\big(p_1p_2+p_3(p_1+p_2)\big)\\&\quad+(p_1p_2p_3)\Big)=D,\end{aligned} \tag{35}$$

where p_1 and $p_2\in\mathbb{C}$, and $p_3\in\mathbb{R}_{>0}$ are the roots of (35), such that they always have positive real parts $\in\mathbb{R}_{>0}$ for a stable system performance. With a basic analysis that compares (34) to (35), the required PID gains for a specific roots location in the s-plane are obtained as follows:

$$\begin{aligned}k_P&=p_1p_2+p_3(p_1+p_2),\\k_I&=p_1p_2p_3,\\k_D&=p_1+p_2+p_3.\end{aligned} \tag{36}$$

The resulting PID gains can then be substituted in (18) and (20) to obtain the equivalent backstepping gains, if feasible. Furthermore, a visual relationship can be established between the system error dynamics roots to the backstepping gains as follows:

$$\begin{aligned}k_1k_2+\gamma+1&\equiv p_1p_2+p_3(p_1+p_2),\\\gamma k_1&\equiv p_1p_2p_3,\\k_1+k_2&\equiv p_1+p_2+p_3.\end{aligned} \tag{37}$$

Note that although a simple analytical solution cannot be attained for this problem, some similarity still exists between the backstepping gains and the error dynamics roots. For instance, k_P and k_I include multiplication of the backstepping gains, and k_D includes summation of the gains. In addition, one can notice the stark similarity captured in Table 1 between the formulations in [6,13] and that in [15], relative to the s-plane poles in (36). This observation stipulates that the backstepping gains have a close manifestation as the system closed-loop poles, when specifically formulated to lend themselves for such analysis.

To provide a clear visual aid that illustrates the above compelling finding, Fig. 4 shows an implementation of the system of equations in (36) for the case when $k_I=0$, such that the root locus of the error dynamics is drawn in Fig. 4a for some constant k_D values, and in Fig. 4b for some constant k_P values. Note that the colored (red, green, and blue) asterisks and stars in Fig. 4 correspond to the exact same gains of their counterparts in Fig. 3. For instance, the solution sets at the stars that mark distant values of k_1 and k_2 in Fig. 3 correspond to more-damped closed-loop system poles, but with one pole being closer to the imaginary axis in most cases as per Fig. 4. Also, the solution sets at the asterisks that mark dual (equal) values of k_1 and k_2 in Fig. 3 correspond to less-damped closed-loop system poles, with their real-parts lying in between the real-part of the sets marked by the stars as per Fig. 4. Hence, it can be stated that moving towards the dual solution of k_1 and k_2 in Fig.3 (at the asterisks) during

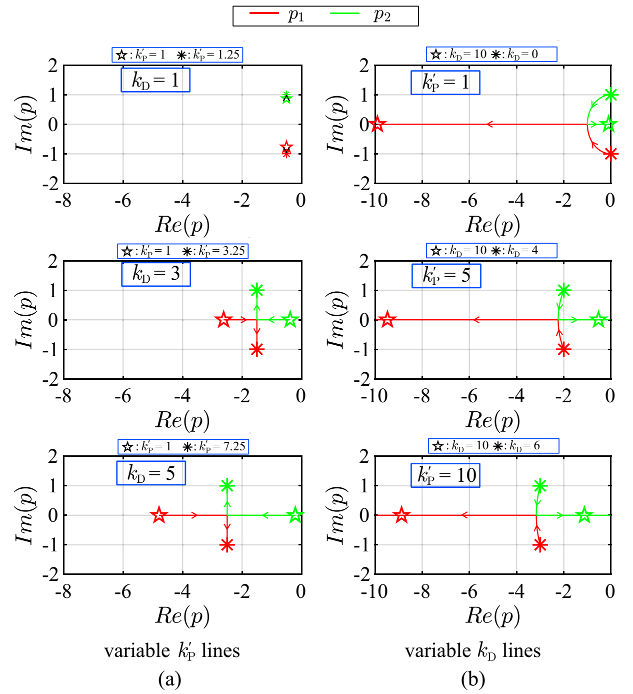


Fig. 4 Coupled pole locations, p_1 and p_2, for specific k_P and k_D values at constant k_D in **a** on the left, and at constant $k'_P = k_P - \gamma$ in **b** on the right. The stars and asterisks mark the same set of the corresponding gains in Fig. 3

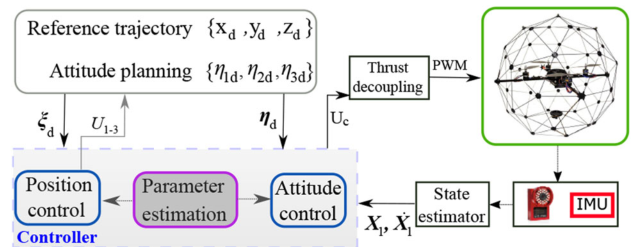


Fig. 5 Control system architecture for a quadrotor UAV

Algorithm 1 Backstepping Gains Tuning Using PID Formulation

Input: k_P, k_I, k_D
Output: k_1, k_2, γ
Initialization :
Calculate k_1, k_2, γ from (20)
while *stability and performance criteria not satisfied* **do**
if $(k_1 k_2 + 1 >> k_I/k_1)$ **then**
  Tune $(k_P - \gamma)$ using Table 3 and Fig. 3c
  Tune k_D using Table 3 and Fig. 3d
  Tune k_I using Table 3
  Find the equivalent k_1 and k_2 from Fig. 3 or (24)
  Calculate γ from (24)
else
  Tune k_P using Table 3 and Fig. 2a
  Tune k_D using Table 3 and Fig. 2b
  Tune k_I using Table 3 and Fig. 2
  Find the equivalent k_1, k_2 and γ from Fig. 2 and (20)
end if
return k_1, k_2, γ

the gains tuning process tends to reduce the damping ratio of the resultant s-plane poles.

The unique combination of Figs. 3 and 4 allows for understanding and visually expressing the relationship between backstepping gains, the PID gains, and the poles of the closed-loop system.

4.6 Tuning procedure

After deriving the necessary equations, expressing them in the proper form, and providing visual aids and practical implementation tips, we consolidate the work in an algorithm that can be used to tune the gains of a backstepping-based controller based on the findings of this paper, which is presented in Algorithm 1.

Note that if a primitive PID controller is designed as a baseline to get a first assessment of the system performance, the obtained PID gains can be plugged into Algorithm 1 for initialization. It is also possible to calculate the set of PID gains that satisfy the design specifications given by the pole locations in (36). Finally, the fine-tuning of the gains can be performed to strike a balance between the performance or stability of the system by referring to Figs. 2 and 3.

5 Simulation and experimental validation

Among various possible applications, a quadrotor UAV platform with variable payload is adopted to validate the hereby proposed ICS, where a motion control system is required to achieve good tracking performance of desired trajectories in the presence of disturbances and changing operating conditions. Collectively achieving the aforementioned objectives is faced with several challenges including design complexity, internal and external disturbances, high frequency noise, and applicability to various quadrotor platforms. To deal with the challenges while achieving the performance objectives, the controller structure shown in Fig. 5 is adopted, which features an inner-loop for attitude control and an outer-loop for position control, thus the proposed ICS has the advantage of reducing the complexity of the design and tuning process.

Nonetheless, the main purpose of this validation section is to show the tuning process for a complex robotic system using the methodology presented in this work, and not necessarily to show the superiority of the control law. The proposed algorithm is validated via numerical simulations on a quadrotor UAV's dynamic model of high-fidelity, and experimentally on the Quanser QBall-2 quadrotor platform [31].

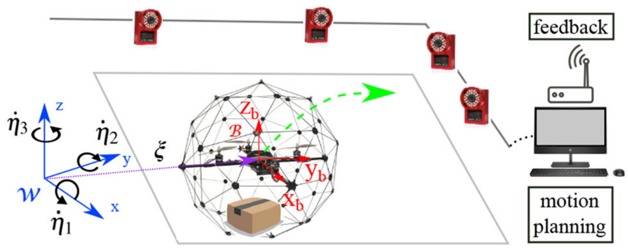


Fig. 6 Quadrotor system setup with an indoor motion capture system

5.1 Dynamic model

Consider the system illustrated in Fig. 6, where the inertial and body frames of reference are $\mathcal{W}$ and $\mathcal{B}$, respectively. The latter is fixed to the quadrotor's centroid, $\mathcal{O}_B$. The quadrotor's mass and inertia tensor relative to $\mathcal{O}_B$ are respectively $m \in \mathbb{R}_{>0}$ and $\boldsymbol{J} \in \mathbb{R}^{3\times3}$. The quadrotor's center of mass (CoM) coordinates in $\mathcal{B}$ are denoted by $\boldsymbol{r} = [r_x, r_y, r_z]^\intercal \in \mathbb{R}^3$, its translational position in $\mathcal{W}$ is $\boldsymbol{\xi} = [\mathrm{x}, \mathrm{y}, \mathrm{z}]^\intercal$, and its Euler angles vector for roll, pitch, and yaw motions is $\boldsymbol{\eta} = [\eta_1, \eta_2, \eta_3]^\intercal$.

The motor and propeller dynamics are superficially modelled as a reduced first-order transfer function, but they are not included in the controller design process due to their relatively fast dynamics as compared to those of the quadrotor's chassis. The quadrotor's dynamics are represented by the following model:

$$\begin{aligned} m\ddot{\boldsymbol{\xi}} &= -m\boldsymbol{G} + \mathbf{R}_\mathrm{t}\boldsymbol{F}_\mathcal{B} + \Delta_\xi, \\ \boldsymbol{J}\mathbf{R}_\mathrm{r}\ddot{\boldsymbol{\eta}} &= \boldsymbol{\tau}_\mathcal{B} + \boldsymbol{r} \times \boldsymbol{F}_\mathcal{B} + \Delta_\eta, \end{aligned} \tag{38}$$

where $\mathbf{R}_\mathrm{r}$ and $\mathbf{R}_\mathrm{t}$ are the rotational and translational transformation matrices between $\mathcal{W}$ and $\mathcal{B}$ [32]; $\boldsymbol{G} = [0, 0, g]^\intercal$ is the gravitational acceleration vector and g is the gravity constant; $\boldsymbol{F}_\mathcal{B} = [0, 0, F_\mathrm{t}]^\intercal$ and $\boldsymbol{\tau}_\mathcal{B} = [T_1, T_2, T_3]^\intercal \in \mathbb{R}^3$ are the total force and torque vectors of the rotors in $\mathcal{B}$; the relationship between the motor speeds and the net thrust and torques is defined as $\mathrm{U} = \boldsymbol{\Gamma}\boldsymbol{\Omega}^2$, and explicitly expressed as:

$$\begin{bmatrix} F_\mathrm{t} \\ T_1 \\ T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} K_\mathrm{t} & K_\mathrm{t} & K_\mathrm{t} & K_\mathrm{t} \\ 0 & 0 & dK_\mathrm{t} & -K_\mathrm{t}L \\ K_\mathrm{t}L & -K_\mathrm{t}L & 0 & 0 \\ K_\mathrm{Q} & K_\mathrm{Q} & -K_\mathrm{Q} & -K_\mathrm{Q} \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}, \tag{39}$$

where K_t and K_Q are respectively the rotor constants of thrust and torque, L is the quadrotor's arm length, ω_i is the angular speed of the i^{th} rotor with $i \in [1, 2, 3, 4]$, and $\boldsymbol{\Omega}$ is the vector of the four rotors' angular speeds. The control inputs, ω_i, are calculated from the above equations using matrix inversion and square root operations.

The terms, Δ_ξ and $\Delta_\eta \in \mathbb{R}^3$, respectively denote the external disturbances and unmodeled nonlinearities (not explicitly shown for brevity purposes), and they are assumed to be uniformly bounded. For more details on the quadrotor modeling, readers are referred to [32,33].

5.2 Problem formulation

The dynamics of the quadrotor are expressed in the form of (2) by choosing:

$$\begin{aligned} &\boldsymbol{X}_1 = [\mathrm{x}, \mathrm{y}, \mathrm{z}, \eta_1, \eta_2, \eta_3]^\intercal, \ \boldsymbol{X}_2 = [\dot{\mathrm{x}}, \dot{\mathrm{y}}, \dot{\mathrm{z}}, \dot{\eta}_1, \dot{\eta}_2, \dot{\eta}_3]^\intercal, \\ &\boldsymbol{g} = [m^{-1}\boldsymbol{I}_3, \boldsymbol{O}_3; \boldsymbol{O}_3, (\boldsymbol{J}\mathbf{R}_\mathrm{r})^{-1}], \\ &\boldsymbol{f} = [0, 0, -g, 0, 0, 0]^\intercal, \quad \boldsymbol{\phi} = \left[\boldsymbol{O}_{3\times2}; \boldsymbol{\phi}_2\right], \\ &\boldsymbol{\Delta} = [\Delta_\xi; \Delta_\eta], \ \boldsymbol{U} = [\boldsymbol{F}_\mathcal{W}; \boldsymbol{\tau}_\mathcal{B}], \ \boldsymbol{\theta} = [r_x, r_y]^\intercal, \end{aligned} \tag{40}$$

where the vectors $\boldsymbol{\theta}$, $\boldsymbol{g}$, and $\boldsymbol{\Delta}$ are the unknown entities in the system, and $\boldsymbol{O}_3 \in \mathbb{R}^{3\times3}$ is the null matrix. Acrobatic maneuvers are not considered in this work, and the singular point of $\eta_3 = \frac{\pm\pi}{2}$ is not to be reached, which implies that $\boldsymbol{g}$ remains Lipschitz continuous. The matrix $\boldsymbol{\phi}_2 \in \mathbb{R}^{3\times2}$ is defined as:

$$\boldsymbol{\phi}_2 = \begin{bmatrix} 0 & -F_\mathrm{t}c_{\eta_2}/J_{xx,0} \\ F_\mathrm{t}/J_{yy,0} & -F_\mathrm{t}t_{\eta_1}s_{\eta_2}/J_{xx,0} \\ 0 & F_\mathrm{t}s_{\eta_2}/(c_{\eta_1}J_{xx,0}) \end{bmatrix},$$

where $J_{xx,0}$ and $J_{yy,0}$ are the nominal values of J_{xx} and J_{yy}, respectively. Choosing these nominal values is realistic and practical since the quadrotor's inertia variation is typically a small fraction of its nominal value, which simplifies the formulation of the adaptation law. The control input vector for the translational subsystem is calculated as follows:

$$\boldsymbol{F}_\mathcal{W} = \begin{bmatrix} s_{\eta_2}c_{\eta_3} + s_{\eta_1}c_{\eta_2}s_{\eta_3} \\ s_{\eta_2}s_{\eta_3} - s_{\eta_1}c_{\eta_2}c_{\eta_3} \\ c_{\eta_1}c_{\eta_2} \end{bmatrix} F_\mathrm{t}. \tag{41}$$

5.3 Control inputs and reference signals

The aim of the controller is to achieve trajectory tracking by bringing the state vector, $\boldsymbol{X}_1$, to a desired state vector, $\boldsymbol{X}_{1\mathrm{d}} = [\boldsymbol{\xi}_\mathrm{d}; \boldsymbol{\eta}_\mathrm{d}] = [\mathrm{x_d}, \mathrm{y_d}, \mathrm{z_d}, \eta_{1\mathrm{d}}, \eta_{2\mathrm{d}}, \eta_{3\mathrm{d}}]^\intercal$, by generating an appropriate control input vector, $\mathrm{U_c} = [F_\mathrm{tc}, T_{1\mathrm{c}}, T_{2\mathrm{c}}, T_{3\mathrm{c}}]^\intercal$, denoted by the subscript $(\bullet)_\mathrm{c}$. The motor's rotational speed can be modelled as:

$$\omega_i = H_\mathrm{m}(s)\omega_{i,\mathrm{c}}, \quad i \in [1, 2, 3, 4], \tag{42}$$

where $H_\mathrm{m}(s)$ is the motor's transfer function, and $\omega_{i,\mathrm{c}}$ is the command rotational speed that is proportional to the input pulse width modulated (PWM) signal of each motor. $\omega_{i,\mathrm{c}}$ can be obtained form the inverse of the following equation: $\mathrm{U} = \Gamma\boldsymbol{\Omega}_\mathrm{c}^2$. In practice, the motor dynamics are much faster than those of the quadrotor, thus $H_\mathrm{m}(s)$ can be set to one.

By dividing the quadrotor's motion control system into two subsystems, namely an inner-loop for attitude control (Euler angles) and an outer-loop for position control, the complexity of the problem is considerably reduced. The position controller generates the control signal, $[u_1, u_2, u_3]$, from which the total thrust command is calculated as:

$$F_{\text{tc}} = \sqrt{u_1^2 + u_2^2 + u_3^2}, \tag{43}$$

and the desired attitude angles, $\eta'_{1\text{d}}$ and $\eta'_{2\text{d}}$, are calculated from (41) as follows:

$$\begin{aligned} \eta'_{1\text{d}} &= \arctan\big((u_1 s_{\eta_3} - u_2 c_{\eta_3})/u_3\big), \\ \eta'_{2\text{d}} &= \arcsin\big((u_1 c_{\eta_3} + u_2 s_{\eta_3})/F_{\text{tc}}\big). \end{aligned} \tag{44}$$

The inner-loop commands in (44) are smoothed and bounded by the function:

$$\eta_{i\text{d}} = \eta_{\max} \tanh\big(\eta'_{i\text{d}}/\eta_{\max}\big), \quad i = 1, 2, \tag{45}$$

where $0 < \eta_{\max} < \pi/2$ is the attitude angle's upper limit. As a result, $\boldsymbol{X}_{1\text{d}}$ is smooth and bounded. This completes the definition of the desired state vector, $\boldsymbol{X}_{1\text{d}}$, to be tracked by $\boldsymbol{X}_1$.

Note that the control law in (17) and its modification in (19) were implemented with the projection function being realised as an integrator, with upper and lower saturation bounds calculated as $\bar{d}/(\gamma k_1)$ based on Remark 2.

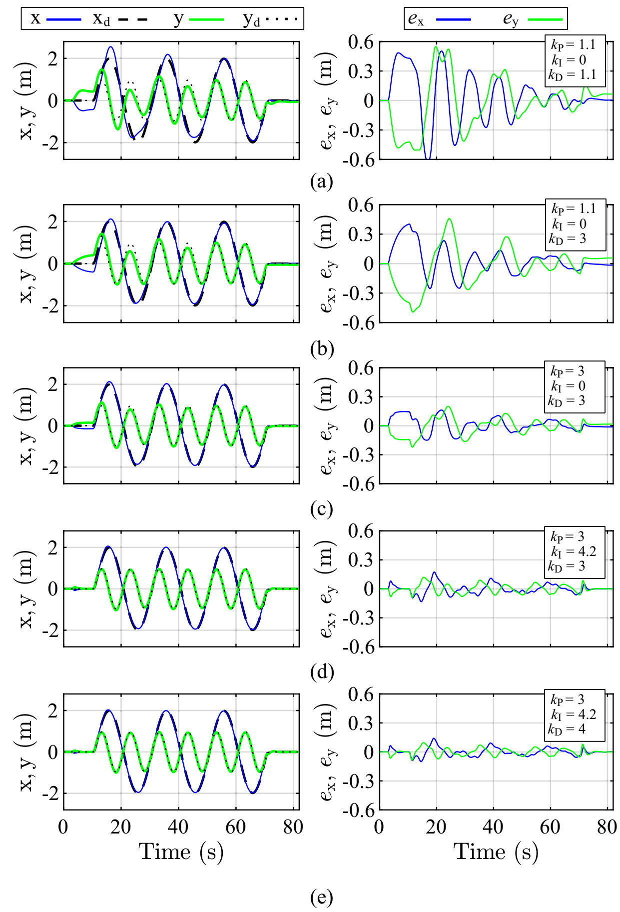


Fig. 7 Desired trajectory tracking of the x and y position states (left) and their corresponding tracking errors (right) of the ICS-controlled quadrotor in each step of the tuning process executed via numerical simulations

5.4 Simulation model elements

5.4.1 Simulation setup

The MATLAB/Simulink® environment is used for validating the proposed control design and tuning scheme, where a high-fidelity nonlinear dynamic model powers the flight mission. The model is augmented to include feedback delays, force and torque disturbances, and induced sensor noise. Estimates of the actual measurement signals are used in the feedback process, where a Kalman filter generates smooth signals from raw sensory data, and RLS algorithm provides estimates of the parameters m, J_{xx}, J_{yy}, J_{zz}, r_x, and r_y. An external wind gust is generated via the method described in [34] to disturb the quadrotor with a magnitude of 5 m/s, which results in a drag force that acts on the quadrotor's centroid.

5.4.2 Quadrotor specifications

The Quanser QBall-2 platform [31] is used as a baseline for the quadrotor's specifications. We set: $m = 1.76\,\text{kg}$, $J_{xx} = J_{yy} = 0.03\,\text{kg}\,\text{m}^2$, $J_{zz} = 0.04\,\text{kg}\,\text{m}^2$, $K_\text{t} = 13\,\text{N}$, $K_\text{Q} = 0.4\,\text{N}\,\text{m}$, and $L = 0.2\,\text{m}$. Upon adding a payload, the quadrotor's CoM and moment of inertia (MoI) vary as per [35], with their main effect manifesting in the moment arm shift due to pure thrust as expressed in (40) by the term $(\boldsymbol{r} \times \boldsymbol{F}_{\mathcal{B}})$. The thrust and torques feedback signals are estimated via a battery drain model due to the unavailability of a motor speed sensor [36].

5.5 Testing scenario

The controller testing scenario consists of the following actions: the quadrotor first takes-off from the ground while picking up a payload, it executes an ∞-shaped maneuver three times, and then performs landing on top of its starting position. The payload has a mass $m_\text{p} = 0.2\,\text{kg}$ and is placed beneath the original CoM of the quadrotor at coordinates $\boldsymbol{r}_\text{p} = [r_{\text{p}1}, r_{\text{p}2}, r_{\text{p}3}] = [0.05, 0.05, -0.1]\,\text{m}$ in $\mathcal{B}$. We set $J_{xx,0} = J_{xx}$, and $J_{yy,0} = J_{yy}$, and a wind gust of magnitude 5 m/s is activated at $t = 11\,\text{s}$.

5.6 Tuning process

The tuning process described in Sect. 4.6 is performed for the outer loop's lateral motion subsystems (x and y components) to serve as a guideline for utilizing the proposed tuning method. Tuning is based on the PID gains, since it is intended in this work to move away from the abstract backstepping gains, and the process is phased as follows:

1. The x and y subsystems' gains are initially set to $k_{\mathrm{P,x,y}} = 1.1$, $k_{\mathrm{D,x,y}} = 1.1$, and $k_{\mathrm{I,x,y}} = 0$. The results are shown in Fig. 7a, where a sluggish response with undesirable overshoot and steady-state error are obtained.
2. We attempt to improve the overshoot performance and increase the tuning range of $k_{\mathrm{P,x,y}}$, as we notice from Fig. 3c that for $k_{\mathrm{D}} \approx 1$, k_{P} is limited to a small set near unity. The specified objectives are attained by increasing $k_{\mathrm{D,x,y}}$ as per Table 3. We choose $k_{\mathrm{D,x,y}} = 3$. The results are shown in Fig. 7b, where we notice a significant reduction in the tracking error and overshoot. But the system's transient response is still slow in the initial time interval $t = [10, 19]$ s.
3. Next, we attempt to increase the speed of the system, which by referring to Table 3 can be achieved by increasing $k_{\mathrm{P,x,y}}$. By referring to Fig. 3c, we choose $k_{\mathrm{P,x,y}} = 3$. The results are shown in Fig. 7c, where we indeed observe an improved speed of response. But the system still has a steady-state error that is noticed at the time intervals $[0, 10]$ s and $[70, 80]$ s.
4. Next, we increase the integral gain to eliminate the steady-state error, which is due to the initial drift of the quadrotor that is induced by the presence of the eccentric payload. We gradually increase $k_{\mathrm{I,x,y}}$ until we find an optimal value, while maintaining $k'_{\mathrm{P,x,y}} := k_{\mathrm{P,x,y}} - \gamma_{\mathrm{x,y}}$ constant to reduce unwanted effects on the previously attained performance. We set $k_{\mathrm{I,x,y}} = 4.2$. The results are shown in Fig. 7d, where we notice that the initial drift in position has been promptly compensated for. Now, we can proceed to the final stage of fine-tuning the gains.
5. By inspecting Fig. 3b, we see that the overshoot can be improved at the expense of reducing $k_{2,\mathrm{x,y}}$. Thus, we proceed by slightly increasing $k_{\mathrm{D,x,y}}$ and setting it to $k_{\mathrm{D,x,y}} = 4$. The results are shown in Fig. 7e, where we notice that the overshoot has been reduced (e.g. $t = \{19, 39, 59\}$ s). At this stage, the performance is deemed satisfactory and the tuning process is terminated.

The tuned PID gains and their corresponding backstepping counterparts, which are obtained in each step of the tuning process, are shown in Table 4 along with the respective mean absolute error of the tracking signals (MAE). From the table, it is clearly noticed that predicting the combinations of k_1, k_2, and γ that yield the desired performance improvement is not a trivial task and may prove to be very challenging to be determined by control system engineers. This last result shows the power of exploiting the backstepping and PID gains relationship to speed-up the manual tuning process. On the other hand, when a PID controller is employed for controlling a second-order system of the form shown in (2), the relationships shown in Fig. 3 can serve as an indicator of the stability margins of the system.

Table 4 Tuning steps for the x and y subsystems of the quadrotor's adaptive backstepping controller starting from the selected PID gains

	k_{P}	k_{I}	k_{D}	k_1	k_2	γ	$MAV\ \{\lvert e_{\mathrm{x}}\rvert + \lvert e_{\mathrm{y}}\rvert\}$ (cm)
1.	1.1	0	1.1	1	0.1	0	37
2.	1	0	3	2.97	0.03	0	25
3.	3	0	3	2	1	0	12
4.	3	4.2	3	2.8	0.2	1.5	7
5.	3	4.2	4	3.8	0.2	1.1	6

5.7 Simulation results

The final controller gains are set to $\boldsymbol{k}_1 = diag(3.8, 3.8, 2.6, 30, 30, 4)$, $\boldsymbol{k}_2 = diag(0.2, 0.2, 0.4, 0.3, 0.3, 1)$, and $\boldsymbol{\gamma} = diag(1.1, 1.1, 1, 0.4, 0.4, 1)$ by following the proposed tuning process in conjunction with the performance results obtained in Fig. 7. The quadrotor planar trajectory is shown in the 'xy'-subplot of Fig. 8 (top), where the UAV makes three laps in an ∞-shape trajectory in the presence of wind. The quadrotor keeps oriented towards the direction of the trajectory while maintaining a level flight as seen in the 'heading' and 'z' subplots of Fig. 8, respectively. Finally, the sinusoidal x- and y-signals that produce the ∞-shape trajectory are shown in the 'x' and 'y' subplots of Fig. 8 (bottom right). Based on a mean absolute error $MAV\{[e_{\mathrm{x}}, e_{\mathrm{y}}, e_{\mathrm{z}}]\}$ of $[3.0, 2.7, 1.2]$ cm in position and $MAV\{e_{\eta_3}\} = 1.5°$ in heading angle, the ICS' overall performance is deemed acceptable.

5.8 Experimental validation

We first note that since tuning of controllers typically produces gains that yield sub-optimal performance, the design process is started in the simulation environment to prevent physical damage to the quadrotor system. After arriving at satisfactory simulation results, the experimental validation phase kicks off by following the same tuning procedure, which further validates the proposed algorithm. In the experimental validation, the ICS is tested on the Quanser QBall-2 platform [31], with specifications that are similar to those presented in Sect. 5.4.2. The flight experiment is conducted inside an indoor testing facility, where a motion capture system, depicted in Fig. 6, provides feedback for the closed-loop control system.

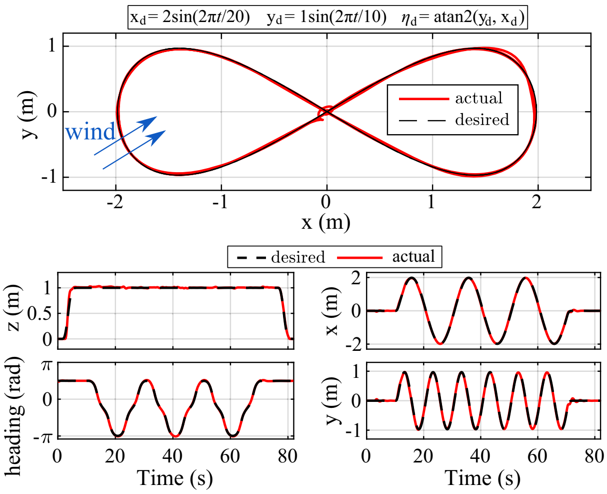


Fig. 8 Trajectory (top) and tracking states (bottom) of the quadrotor in numerical simulations after fine-tuning the ICS per the hereby proposed Algorithm 1

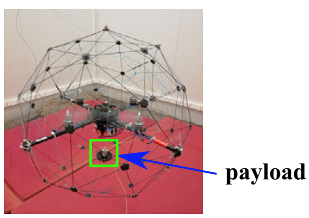


Fig. 9 Demonstration of the quadrotor platform (Quanser QBall-2) lifting a payload inside the UAV testing laboratory

5.8.1 Experimental scenario

The quadrotor is loaded with a payload of $m_{\mathrm{p}} = 0.2\,\mathrm{kg}$, placed at coordinates $\boldsymbol{r}_{\mathrm{p}} = [0, 0, -0.1]$ m in $\mathcal{B}$, which is beneath its original CoM as shown in Fig. 9. The flight maneuver starts by a vertical take-off, followed by a sideways motion in the x-y plane during which the carried payload is released midway, and ends by landing on the ground, as depicted at the top of Fig. 10.

5.8.2 Controller tuning

Given the expected inaccuracies between the simulation model and the physical quadrotor platform, fine-tuning of the obtained gains from the numerical simulations is performed through multiple flight tests using the same proposed tuning guidelines of Algorithm 1. We refrain from showing the detailed tuning step to avoid redundancy. The resulting controller gains that were deemed adequate in terms of performance and robustness are $\boldsymbol{k}_{\mathrm{P}} = diag(2.7, 2.7, 7.2, 32.2, 32.2, 6)$, $\boldsymbol{k}_{\mathrm{I}} = diag(1.3, 1.3, 4.1, 5.6, 5.6, 5.2)$, and $\boldsymbol{k}_{\mathrm{D}} = diag(3.7, 3.7, 4, 16.2, 16.2, 4)$, which give the following backstepping gains: $\boldsymbol{k}_1 = diag(3.3, 3.3, 1.96, 14, 14, 2)$, $\boldsymbol{k}_2 = diag(0.4, 0.4, 2.1, 2.2, 2.2, 2)$, and $\boldsymbol{\gamma} = diag(0.4, 0.4, 2.1, 0.4, 0.4, 1)$. Note that the ratio of change of the PID gains, as compared to those in simulation, is smaller than that of the backstepping gains. Since the PID gains better represent the controller's performance, they should be considered as the starting point for evaluating the transition between simulation and experimentation.

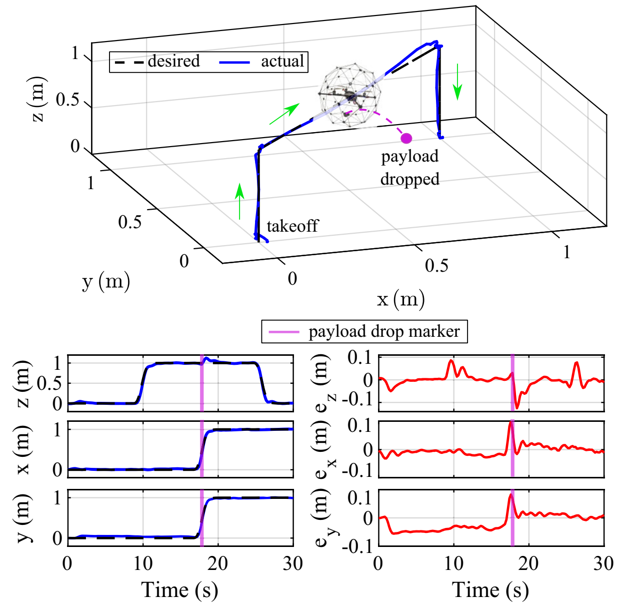


Fig. 10 3D trajectory tracking of the ICS-controlled quadrotor in the payload-drop experiment. The experiment is depicted at the top, and the x, y, and z positions and tracking errors are shown at the bottom

5.8.3 Experimental results

The final tracking performance of the quadrotor is shown at the bottom of Fig. 10, which captures its position in the z-, x-, and y-directions. After adequate fine-tuning, the ICS achieves stable and and accurate tracking performance, with a mean absolute error $MAV\{[e_{\mathrm{x}}, e_{\mathrm{y}}, e_{\mathrm{z}}]\}$ of $[1.3, 2.7, 1.8]$ cm in position. The recorded error in position-hold is minimal in all three directions, even during lifting the payload off the ground and while dropping it. The quadrotor recovery after the sudden drop of the payload (at $t = 18\,\mathrm{s}$, marked by the purple strips at the bottom of Fig. 10), is fairly quick, and the quadrotor successfully and accurately completes its maneuver. The experimental results demonstrate the efficacy of the adaptive backstepping-based ICS and the practicality of the proposed tuning algorithm based on the established PID similarity.

6 Conclusion

In this paper, we leveraged the underutilized similarity between backstepping and PID control laws for a class of second-order systems, and further developed it to formulate a special relationship between the backstepping and PID gains in the form of a third-order polynomial, which can be simplified to second-order under certain conditions. The obtained similarity provides deeper insight into the tuning of backstepping gains and allows for a better understanding of the backstepping control law, which can be represented as the sum of a feedback PID component and a feedforward model compensation component. We provided visual and mathematical tools that facilitate the design and tuning of the backstepping gains in a similar fashion as tuning PID controllers, which benefits from the legacy of well-established PID tuning rules, while maintaining the original stability perspective of the backstepping gains. This work contributes to the field of control systems by promoting the adoption of nonlinear Lyapunov-based integral backstepping controllers, which is achieved by facilitating certain aspects of the control law design and tuning and bridging the gap between complexity and practicality. The proposed control scheme and tuning algorithm are validated numerically via a quadrotor UAV dynamic model of high-fidelity, and experimentally on a quadrotor platform, to showcase the ease and utility of the proposed tuning process.

Future work entails the extension of the proposed method to other nonlinear control designs, in addition to backstepping, and to higher-order classes of nonlinear systems.

Author contribution: All authors contributed to the study conception and design. Material preparation, data collection, and analysis were performed by Ahmad Kourani and Naseem Daher. The first draft of the manuscript was written by Ahmad Kourani and all authors commented on subsequent versions of the manuscript. All authors read and approved the final manuscript.

Availability of data and material: No dataset of additional material is provided.

Declarations

Code availability A related code on the implementation of the methodology presented in this work is publicly available at github.com/AUBVRL/Tune-Backstepping-Like-PID. The code is available as a MATLAB/Simulink project.

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethics approval: Not applicable.

Consent to participate: Not applicable.

Consent for publication: Not applicable.

References

1. Chang WD, Hwang RC, Hsieh JG (2002) A self-tuning PID control for a class of nonlinear systems based on the Lyapunov approach. J Process Control 12(2):233–242. https://doi.org/10.1016/S0959-1524(01)00041-5
2. Astrom (1995) Adaptive control. Addison-Wesley, New York
3. Chang PH, Jung JH (2009) A systematic method for gain selection of robust PID control for nonlinear plants of second-order controller canonical form. IEEE Trans Control Syst Technol 17(2):473–483. https://doi.org/10.1109/TCST.2008.2000989
4. Lee JY, Jin M, Chang PH (2014) Variable PID gain tuning method using backstepping control with time-delay estimation and nonlinear damping. IEEE Trans Industr Electron 61(12):6975–6985. https://doi.org/10.1109/TIE.2014.2321353
5. Han J (2009) From PID to active disturbance rejection control. IEEE Trans Industr Electron 56(3):900–906. https://doi.org/10.1109/TIE.2008.2011621
6. Benaskeur A, Desbiens A (2002) Backstepping-based adaptive PID control. IEE Proc Control Theory Appl 149(5):54–59. https://doi.org/10.1049/ip-cta:20020100
7. Chen M, Ge SS, Ren B (2011) Adaptive tracking control of uncertain MIMO nonlinear systems with input constraints. Automatica 47(3):452–465. https://doi.org/10.1016/j.automatica.2011.01.025
8. Yu J, Shi P, Zhao L (2018) Finite-time command filtered backstepping control for a class of nonlinear systems. Automatica 92:173–180. https://doi.org/10.1016/j.automatica.2018.03.033
9. Swaroop D, Hedrick JK, Yip PP et al (2000) Dynamic surface control for a class of nonlinear systems. IEEE Trans Autom Control 45(10):1893–1899. https://doi.org/10.1109/TAC.2000.880994
10. Dong W, Farrell JA, Polycarpou MM et al (2012) Command filtered adaptive backstepping. IEEE Trans Control Syst Technol 20(3):566–580. https://doi.org/10.1109/TCST.2011.2121907
11. Wen C, Zhou J, Liu Z et al (2011) Robust adaptive control of uncertain nonlinear systems in the presence of input saturation and external disturbance. IEEE Trans Autom Control 56(7):1672–1678. https://doi.org/10.1109/TAC.2011.2122730
12. Pozo F, Ikhouane F, Rodellar J (2008) Numerical issues in backstepping control: sensitivity and parameter tuning. J Franklin Inst 345(8):891–905. https://doi.org/10.1016/j.jfranklin.2008.05.005
13. Ranger P, Desbiens A (2003) Improved backstepping-based adaptive PID control. In: 2003 4th international conference on control and automation proceedings, pp 123–127, https://doi.org/10.1109/ICCA.2003.1594997
14. Skjetne R, Fossen TI (2004) On integral control in backstepping: analysis of different techniques
15. Mian AA, Ahmad MI, Wang D (2008) Backstepping based PID control strategy for an underactuated aerial robot. IFAC Proc Vol 41(2):15,636-15,641. https://doi.org/10.3182/20080706-5-KR-1001.02644
16. Kartal Y, Kolaric P, Lopez V et al (2019) Backstepping approach for design of PID controller with guaranteed performance for micro-air UAV. Control Theory Technol 18(1):19–33. https://doi.org/10.1007/s11768-020-9145-y
17. Kourani A, Daher N (2021a) Leveraging PID gain selection towards adaptive backstepping control for a class of second-order systems. In: 2021 American control conference (ACC), pp 1174–1179, https://doi.org/10.23919/ACC50511.2021.9483159
18. Kourani A, Daher N (2021c) A tethered quadrotor UAV-buoy system for marine locomotion. In: 2021 IEEE international conference on robotics and automation (ICRA), pp 59–65, https://doi.org/10.1109/ICRA48506.2021.9560878
19. Kourani A, Daher N (2021b) Marine locomotion: a tethered UAV-buoy system with surge velocity control. Robot Auton Syst 145(103):858. https://doi.org/10.1016/j.robot.2021.103858

20. Pan Y, Liu G, Kumar KD (2010) Robust stability analysis of asymptotic second-order sliding mode control system using Lyapunov function. In: The 2010 IEEE international conference on information and automation, pp 313–318, https://doi.org/10.1109/ICINFA.2010.5512081
21. Mohd Basri M, Husain A, Danapalasingam K (2015) Enhanced backstepping controller design with application to autonomous quadrotor unmanned aerial vehicle. J Intell Robot Syst Theory Appl 79(2):295–321. https://doi.org/10.1007/s10846-014-0072-3
22. Hu C, Yao B, Wang Q (2010) Integrated direct/indirect adaptive robust contouring control of a biaxial gantry with accurate parameter estimations. Automatica 46(4):701–707. https://doi.org/10.1016/j.automatica.2010.01.022
23. Yao B, Dontha R (2002) Integrated direct/indirect adaptive robust precision control of linear motor drive system with accurate parameter estimations. IFAC Mechtron Syst 35(2):587–592. https://doi.org/10.1016/S1474-6670(17)34003-X
24. Krstic M, Kanellakopoulos I, Kokotovic P (1995) Nonlinear and adaptive control design. Wiley
25. Yao B, Tomizuka M (1997) Adaptive robust control of SISO nonlinear systems in a semi-strict feedback form. Automatica 33(5):893–900. https://doi.org/10.1016/S0005-1098(96)00222-1
26. Yao B, Tomizuka M (2001) Adaptive robust control of MIMO nonlinear systems in semi-strict feedback forms. Automatica 37:1305–1321. https://doi.org/10.1016/S0005-1098(01)00082-6
27. Zou Y (2017) Nonlinear robust adaptive hierarchical sliding mode control approach for quadrotors. Int J Robust Nonlinear Control 27:925–941. https://doi.org/10.1002/rnc.3607
28. Yao B (2000) Adaptive robust motion control of single-rod hydraulic actuators: theory and experiments. IEEE/ASME Trans Mechatron 5(1):79–91. https://doi.org/10.1109/ACC.1999.783142
29. Simmons GF (1996) Calculus with analytic geometry, 2nd edn. McGraw-Hill
30. Astrom K, Hagglund T (1995) PID controller: theory, design and tuning, 2nd edn. In: The instrumentation, systems, and automation society, ISA
31. Quanser (2018) Quanser innovate-educate, www.quanser.com
32. Kourani A, Kassem K, Daher N (2018) Coping with quadcopter payload variation via adaptive robust control. In: 2018 IEEE International multidisciplinary conference on engineering technology, pp 1–6, https://doi.org/10.1109/IMCET.2018.8603047
33. Zhang J, Gu D, Ren Z et al (2019) Robust trajectory tracking controller for quadrotor helicopter based on a novel composite control scheme. Aerosp Sci Technol 85:199–215
34. Raza SA, Etele J (2016) Autonomous position control analysis of quadrotor flight in urban wind gust conditions. In: AIAA guidance, navigation, and control conference. https://doi.org/10.2514/6.2016-1385
35. Lotfi B, Goharimanesh M (2015) Modelling and control of quadrotor maneuvers with variations of center of gravity (COG). pp 1570–1574. https://doi.org/10.1109/AIM.2015.7222766
36. Dhaybi M, Daher N (2020) Accurate real-time estimation of the inertia tensor of package delivery quadrotors. In: 2020 American control conference (ACC), pp 1520–1525, https://doi.org/10.23919/ACC45564.2020.9147948