

Parameter Estimation and Dynamic Extension for a Quadrotor

Jesús Tordesillas Torres

Abstract—In this work I compare four different adaptive controllers to estimate the mass and inertia matrix of a quadrotor, which is an underactuated system. First, the dynamic model of a quadrotor is derived and decoupled. Then, the tracking error based (TEB), constant gain (CG), bounded-gain-forgetting (BGF) and cushioned-floor (CF) adaptive controllers are implemented and compared under a persistently exciting signal both with constant and time-varying parameters. Finally, once the parameters have been estimated, a feedback linearization and input-output decoupling of the quadrotor using dynamic extension is proposed to allow the tracking of a trajectory specified in the space of the flat outputs $[x \ y \ z \ \psi]^T$ (position and yaw angle).

I. RELATED WORK

Adaptive controllers can be classified according to the mechanisms they use [1]: The direct adaptive controllers (that only use the tracking errors), the indirect adaptive controllers (which are also called *self-tuning* controllers, and that only use the prediction error on the filtered joint torque), and the composite adaptive controllers (that use both the tracking and the torque prediction error) [1], [2]. Most of the adaptive controllers presented in the literature are usually implemented to satisfy the control task ([3], [4], [5]), and therefore the parameters are estimated in a need-to-know basis. In the first part of this paper, the focus is more on the accurate parameter estimation itself, with the idea of avoiding the use of expensive moment of inertia measurement instruments or relying on time-consuming and not completely accurate 3D CAD designs. In this paper I focus on the comparison between the direct and composite adaptive controllers, and analyze the improvement on the parameter convergence and tracking accuracy when the prediction error on the joint torque is also used in the adaptation law.

When the parameters of the quadrotor are known (or accurately estimated), different approaches have been proposed in the literature in order to track a trajectory specified in the flat output space $[x \ y \ z \ \psi]^T$. A common one is to assume a *n-th* order integrator model to generate a trajectory, and then stabilize the drone along this trajectory using an outer loop that controls position and an inner one that controls attitude [6], both with PIDs. Another way is to use dynamic inversion with zero-dynamics stabilization [7]. In this work

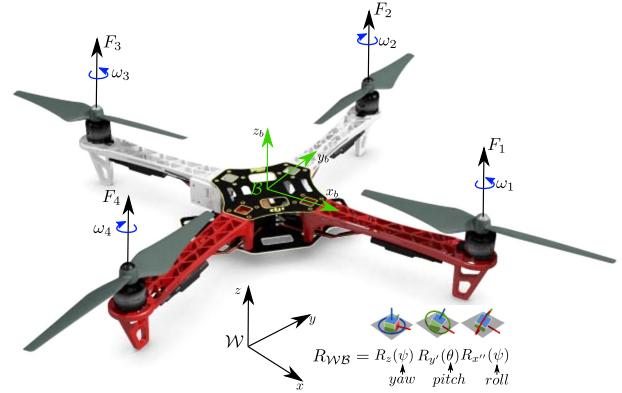


Figure 1: World and body frames considered. The figure of the rotations is taken from [10] .

I focus on the input-output decoupling and feedback linearization approach using dynamic extension [8], [9].

II. DYNAMIC MODEL OF A QUADROTOR

For the derivations of the equations, we will use the ZYX convention for the Euler angles (also called 321 convention). Let $\mathbf{1}$ be the identity matrix and \mathbf{I} be the inertia matrix. Using the Newton-Euler equations, and denoting $[\tau_{roll} \ \tau_{pitch} \ \tau_{yaw}]^T$ as the inputs of the system (thrust and torques in each axis), we have that ([11], [7], [12]):

$$\begin{bmatrix} m\mathbf{1}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{a} \\ \boldsymbol{\alpha} \end{bmatrix} + \begin{bmatrix} 0 \\ \boldsymbol{\omega} \times \mathbf{I} \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{eb} \mathbf{F}_b - mg \\ \boldsymbol{\tau}_{ang} \end{bmatrix}$$

where $\mathbf{R}_{eb} = \mathbf{R}_z(\psi) \mathbf{R}_y(\theta) \mathbf{R}_x(\phi)$ is the rotation matrix that transforms a vector expressed in the body frame to the world frame, $\mathbf{F}_b = [0 \ 0 \ T]^T$, and $\boldsymbol{\tau}_{ang} = [\tau_{roll} \ \tau_{pitch} \ \tau_{yaw}]^T$. We can assume that the quadrotor is symmetric with respect to the axes of the body frame, and therefore $\mathbf{I} = \text{diag}(I_{xx}, I_{yy}, I_{zz})$. Moreover we can also use the common approximation of $[\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T \approx [p \ q \ r]^T$ (where $[\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$ is the angular velocity in the world frame, and $[p \ q \ r]^T$ is the angular velocity in the body frame). This holds true for small angles of movement [11]. Hence, we have that:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \frac{(c_\phi c_\psi s_\theta + s_\phi s_\psi)T}{m} \\ \frac{(-c_\psi s_\phi + c_\phi s_\theta s_\psi)T}{m} \\ -g + \frac{c_\theta c_\phi T}{m} \\ \frac{I_{yy}\dot{\theta}\dot{\psi} - I_{zz}\dot{\theta}\dot{\psi} + \tau_{roll}}{I_{yy}} \\ \frac{-I_{xx}\dot{\theta}\dot{\psi} + I_{zz}\dot{\phi}\dot{\psi} + \tau_{pitch}}{I_{yy}} \\ \frac{I_{xx}\dot{\theta}\dot{\phi} - I_{yy}\dot{\theta}\dot{\phi} + \tau_{yaw}}{I_{zz}} \end{bmatrix} \quad (1)$$

Defining now $\mathbf{q}_{aug}^T := [x \ y \ z \ \phi \ \theta \ \psi]^T$, we can express the above system in the standard manipulator equation:

$$\mathbf{H}(\mathbf{q}_{aug})\ddot{\mathbf{q}}_{aug} + \mathbf{C}_{aug}(\mathbf{q}_{aug}, \dot{\mathbf{q}}_{aug})\dot{\mathbf{q}}_{aug} + \mathbf{G}_{aug}(\mathbf{q}_{aug}) = \mathbf{B}_{aug}\boldsymbol{\tau}$$

$$\mathbf{H}_{aug} = \begin{bmatrix} m\mathbf{1}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I} \end{bmatrix} \quad \mathbf{B}_{aug} = \begin{bmatrix} \mathbf{1}_{3 \times 1} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 1} & \mathbf{I} \end{bmatrix}$$

$$\mathbf{C}_{aug} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 0 & I_{zz}\dot{\psi} & -I_{yy}\dot{\theta} \\ \mathbf{0}_{1 \times 3} & -I_{zz}\dot{\psi} & 0 & I_{xx}\dot{\phi} \\ \mathbf{0}_{1 \times 3} & I_{yy}\dot{\theta} & -I_{xx}\dot{\phi} & 0 \end{bmatrix} \quad \mathbf{G}_{aug} = \begin{bmatrix} 0 \\ 0 \\ mg \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

In general, a system expressed as $\ddot{\mathbf{q}} = f_1(\mathbf{q}, \dot{\mathbf{q}}, t) + f_2(\mathbf{q}, \dot{\mathbf{q}}, t)\mathbf{u}$ is underactuated if $rank[f_2(\mathbf{q}, \dot{\mathbf{q}}, t)] < dim[\mathbf{q}]$ ([13]). Hence, we have to check $rank[\mathbf{H}_{aug}^{-1}\mathbf{B}_{aug}]$. As $\mathbf{H}(\mathbf{q}_{aug}) \succ 0$, $rank[\mathbf{H}_{aug}^{-1}\mathbf{B}_{aug}] = rank[\mathbf{B}_{aug}] = 4 < 6$, and therefore the system is underactuated (which intuitively makes sense, since the number of actuators is less than the number of degrees of freedom).

III. PARAMETER ESTIMATION

A. Algorithms

To do parameter estimation, we will exploit the hierarchy that exists in the above dynamic equations, and that allows us to focus only on $\mathbf{q}^T = [z \ \phi \ \theta \ \psi]^T$, and now the subsystem with only these states is clearly fully-actuated. The matrices \mathbf{H} , \mathbf{C} , \mathbf{G} , \mathbf{B} can be defined accordingly as subblocks of the matrices above.

A summary of all the adaptive controllers implemented in this project, and their convergence properties, are shown in Table I. Detailed proofs of the convergence properties can be found in [1], [14], [15].

All the adaptive controllers use the same control law:

$$\boldsymbol{\tau} = \mathbf{Y}\hat{\mathbf{a}} - \mathbf{K}_D s$$

For the adaptation law, all of them share the following expression:

$$\dot{\mathbf{a}} = -\mathbf{P}(t) [\mathbf{Y}^T s + \mathbf{W}^T \mathbf{R}(t) e]$$

but the values of \mathbf{R} and \mathbf{P} are different for each controller. In this adaptation law, the first term takes into account the tracking error and the second term takes into account the prediction error on the filtered torque. TEB is a direct adaptive controller and only uses the tracking error ($\mathbf{R} = \mathbf{0}$). The composite adaptive controllers maintain both terms, and $\mathbf{R}(t)$ is a therefore weighting matrix that specifies the attention that the adaptation law should pay to the prediction error e . CG has $\mathbf{P}(t) = \mathbf{P}_0 \succ 0$ constant.

An alternative to TEB and CG is to allow the gain matrix \mathbf{P} to be time-varying. One option would be to use simply $\frac{d}{dt}\mathbf{P}^{-1}(t) = \mathbf{W}^T \mathbf{W}$. However, under the presence of a persistently exciting signal, this may lead to $\mathbf{P}^{-1}(t)$ vanishing (or $\mathbf{P}(t)$ exploding) [1]. While this may not be a problem to estimate parameters that are constant, it is usually undesired when the parameters are changing over time. Both BFG and CF address this issue: On one hand, they use a time-varying adaptation gain $\mathbf{P}(t)$, which allows them to give more importance to current information (generated by current parameters) than to past data. But on the other hand, they use an update law that prevents \mathbf{P} from exploding:

- BGF uses the following gain update:

$$\frac{d}{dt}\mathbf{P}^{-1}(t) = -\lambda_{BGF}(t)\mathbf{P}^{-1} + \mathbf{W}^T \mathbf{W}$$

with a forgetting factor $\lambda_{BGF}(t) = \lambda_0 \left[1 - \frac{\|\mathbf{P}\|}{k_0}\right]$.

In this equation, k_0 behaves as an upper bound on $\|\mathbf{P}\|$. In this way, a strong persistent excitation will provoke $\|\mathbf{P}\|$ small, and therefore the forgetting factor will be high ($\lambda_{BGF}(t) \approx \lambda_0$). However, if the persistent excitation is small, then $\|\mathbf{P}\| \lesssim k_0$, and therefore the forgetting factor will be $\lambda_{BGF}(t) \approx 0$ [1], [14]. This forgetting factor used guarantees that $\mathbf{P}(t) \preceq k_0 \mathbf{I}$ [14].

- CF uses

$$\frac{d}{dt}\mathbf{P}^{-1}(t) = -\lambda_{CF}(t) [\mathbf{P}^{-1} - \mathbf{K}_0^{-1}] + \mathbf{W}^T \mathbf{W}$$

as update, with $\mathbf{K}_0 \succ 0$ and $\lambda_{CF}(t) > 0$. This law guarantees that $\mathbf{P}^{-1} \succeq \mathbf{K}_0^{-1}$ and therefore $\mathbf{P} \preceq \mathbf{K}_0$ [1].

Both CF and BGF guarantee exponential convergence of $\tilde{\mathbf{q}}$, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{a}}$ under a persistently exciting signal (this does not happen for TEB or CG).

In the equations of table I, the variable s is $s = \dot{\mathbf{q}} - \dot{\mathbf{q}}_r = \dot{\mathbf{q}} + \Lambda \tilde{\mathbf{q}}$ (measure of the tracking error), and the matrix $\mathbf{Y} = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)$ is defined such that:

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}}_r + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_r + \mathbf{G}(\mathbf{q}) = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)\mathbf{a}$$

For our case, we have that:

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}}_r + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_r + \mathbf{G}(\mathbf{q}) = \begin{bmatrix} m(\ddot{z}_r + g) \\ I_{xx}\ddot{\phi}_r + I_{zz}\dot{\theta}_r\dot{\psi} - I_{yy}\dot{\theta}_r\dot{\psi}_r \\ I_{yy}\ddot{\theta}_r - I_{zz}\dot{\phi}_r\dot{\psi} + I_{xx}\dot{\phi}\dot{\psi}_r \\ -I_{xx}\dot{\theta}_r\dot{\phi} + I_{yy}\dot{\theta}_r\dot{\phi}_r + I_{zz}\dot{\psi}_r \end{bmatrix}$$

	TEB	CG	BGF	CF
Cont. Law	$\tau = \mathbf{Y}\hat{\mathbf{a}} - \mathbf{K}_D s$			
Adap. Law	$\dot{\mathbf{a}} = -\mathbf{P}(t) [\mathbf{Y}^T s + \mathbf{W}^T \mathbf{R}(t)e]$			
Requir.	$\mathbf{P}(t) = \mathbf{P}_0 \succ 0$ $\mathbf{R} = \mathbf{0}$	$\mathbf{P}(t) = \mathbf{P}_0 \succ 0$	$\frac{d}{dt} \mathbf{P}^{-1} = -\lambda_{BGF}(t) \mathbf{P}^{-1} + \mathbf{W}^T \mathbf{W}$ $\lambda_{BGF}(t) = \lambda_0 \left[1 - \frac{\ \mathbf{P}\ }{k_0} \right]$	$\frac{d}{dt} \mathbf{P}^{-1} = -\lambda_{CF}(t) [\mathbf{P}^{-1} - \mathbf{K}_0^{-1}] + \dots$ $\dots + \mathbf{W}^T \mathbf{W}$ $\mathbf{K}_0 \succ 0$ $\lambda_{CF}(t) > 0$
				$\mathbf{W} = \frac{\lambda_f}{p+\lambda_f} [\mathbf{Y}_1(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})]$
If $\mathbf{q}_d, \dot{\mathbf{q}}_d$ bounded	$\ddot{\mathbf{q}}, \dot{\mathbf{q}} \rightarrow 0$ $\tilde{\mathbf{a}}$ bounded			$\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{e} \rightarrow 0$ $\tilde{\mathbf{a}}$ bounded
If traj. are p.e. and u.c.	$\tilde{\mathbf{a}} \rightarrow \mathbf{0}$	$\tilde{\mathbf{a}} \rightarrow \mathbf{0}$	$\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \tilde{\mathbf{a}} \rightarrow \mathbf{0}$ exponentially	$\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \tilde{\mathbf{a}} \rightarrow \mathbf{0}$ exponentially
V	$\frac{1}{2} (s^T H s + \tilde{\mathbf{a}}^T P^{-1} \tilde{\mathbf{a}})$			

Table I: Adaptive controllers studied in this work

As $\mathbf{a}^T = [m \quad I_{xx} \quad I_{yy} \quad I_z]^T$ we obtain:

$$\mathbf{Y} = \begin{bmatrix} \ddot{z}_r + g & 0 & 0 & 0 \\ 0 & \ddot{\phi}_r & -\dot{\theta}\dot{\psi}_r & \dot{\theta}_r\dot{\psi} \\ 0 & 0 & \ddot{\theta}_r & -\dot{\phi}_r\dot{\psi} \\ 0 & -\dot{\theta}_r\dot{\phi} & \dot{\theta}\dot{\phi}_r & \ddot{\psi}_r \end{bmatrix}$$

and $\mathbf{Y}_1(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ and $\mathbf{Y}_{1d}(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d)$ are obtained analogously. \mathbf{W} is a filtered version of \mathbf{Y}_1 , and therefore the explicit measurements of $\ddot{\mathbf{q}}$ are not needed in these algorithms. Finally, the prediction error is $\mathbf{e} = \mathbf{W}\tilde{\mathbf{a}} = \mathbf{W}\hat{\mathbf{a}} - \boldsymbol{\tau}_{\text{filtered}}$, with $\boldsymbol{\tau}_{\text{filtered}}$ being a filtered version of the input $\boldsymbol{\tau}$.

The parameters chosen for the simulations done in this paper have been:

Name	Value
\mathbf{K}_0	$50 \cdot \mathbf{1}_{4 \times 4}$
λ_0	4800
k_0	200
λ_f	10
$\lambda_{CF}(t)$	4800

Name	Value
\mathbf{P}_0	$5 \cdot \mathbf{1}_{4 \times 4}$
$\mathbf{R}(t)$	$\mathbf{1}_{4 \times 4}$
λ_c	5
\mathbf{K}_D	$\lambda_c \hat{\mathbf{H}}$
$\mathbf{\Lambda}$	$\lambda_c \mathbf{1}_{4 \times 4}$

The parameters to be estimated (mass and inertia matrix) have been chosen as

$$m = 0.5 \text{ kg}$$

$$\mathbf{I} = \text{diag}(9.6, 9.6, 17.6) \cdot 10^{-2} \text{ kg m}^2$$

We also compare the previous adaptive controllers with a simple PD controller using $\boldsymbol{\tau} = -\mathbf{K}'_D s$, with $\mathbf{K}'_D = 20 \cdot \mathbf{1}_{4 \times 4}$. The initial values of the states and estimated parameters have been chosen as 0, and the matrices \mathbf{P} and \mathbf{W} have been initialized as $10^3 \cdot \mathbf{1}_{4 \times 4}$ and $\mathbf{1}_{4 \times 4}$ respectively.

B. Persistent excitation

To estimate accurately the parameters, let us first find a persistently exciting signal (and therefore we

guarantee the exponential convergence for BGF and CF). Note that we have the following relation [16], [17]:

$$\left. \begin{array}{l} \ddot{\mathbf{q}}, \dot{\mathbf{q}} \rightarrow 0 \\ \mathbf{Y}_{1d} \text{ is p.e.} \\ \mathbf{Y}_{1d} \text{ is unif. cont} \end{array} \right\} \Rightarrow \mathbf{W}_d \text{ is p.e.} \quad \left. \begin{array}{l} \mathbf{W} \text{ is p.e.} \end{array} \right\}$$

So we should check that \mathbf{Y}_{1d} is p.e. and uniformly continuous. Let us choose the desired trajectories as:

$$z_d = 2(1 - \cos(\pi t))$$

$$\phi_d = \psi_d = \theta_d = \frac{\pi}{4}(1 - \cos(\pi t))$$

From the definition of \mathbf{Y}_{1d} above, it is clear that it is uniformly continuous for these trajectories. To check the persistent excitation, we will check if the condition

$$\exists \alpha_1 > 0 \text{ s.t. } \forall T > 0, \int_t^{t+T} \mathbf{Y}_{1d}^T \mathbf{Y}_{1d} d\tau \geq \alpha_1 \mathbf{1} \quad (2)$$

is satisfied for a given desired trajectory. For these trajectories, we obtain

$$\mathbf{Y}_{1d}^T \mathbf{Y}_{1d} = \begin{bmatrix} A_{11} & 0 & 0 & 0 \\ 0 & A_{22} & A_{23} & A_{24} \\ 0 & A_{23} & A_{33} & A_{34} \\ 0 & A_{24} & A_{34} & A_{44} \end{bmatrix}$$

where

$$A_{11} = (6\pi^2 \cos(\pi\tau) + 9.81)^2$$

$$A_{22} = A_{33} = A_{44} = \frac{1}{128} (\pi^8 \sin^4(\pi\tau) + 8\pi^6 \cos^2(\pi\tau))$$

$$A_{23} = A_{24} = A_{34} = -\frac{1}{256} \pi^8 \sin^4(\pi\tau)$$

We can now take the integral symbolically and compute the eigenvalues of the resulting matrix (the expressions obtained are omitted here due to their length). The plot of the eigenvalues as a function of t and T is shown in Fig. 2. Note that all of them are greater than zero, and therefore, the condition of persistent excitation (2) is satisfied.

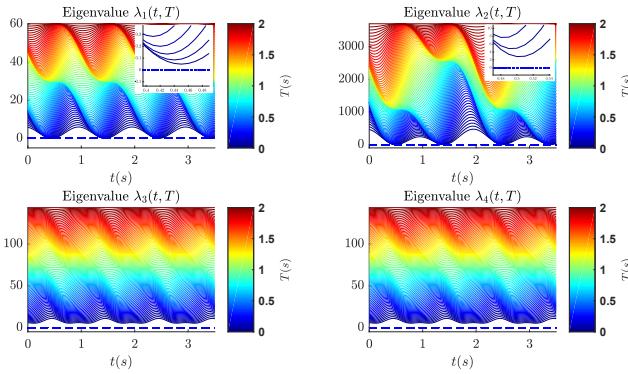


Figure 2: Eigenvalues of the matrix $\int_t^{t+T} \mathbf{Y}_{1d}^T \mathbf{Y}_{1d} dt$

C. Simulation with constant parameters

We first simulate using MATLAB the previous adaptive controllers for the case where the parameters are constant. The results obtained can be seen in Fig. 3, 4, and 5.

Note that TEB presents a much higher oscillatory behavior compared to CG, BGF and CF, specially in the estimation of the parameters. TEB in general also converges slower. For instance, the estimation of the inertia parameters of CG, BGF and CF converges in ≈ 0.1 seconds, while TEB converges in ≈ 1.4 s.

TEB also has high oscillations in the control inputs, reaching 30 N in the thrust at the beginning of the simulation. Both CG, BGF, CF and PD have similar thrust and torques profiles.

With respect to the position errors, BGF and CF have a really small error in all the Euler angles (< 0.05 deg), while both TEB and CG have an maximum error of around 0.4 deg. This can be justified from the exponential convergence property that BGF and CF exhibit when \mathbf{Y}_{1d} is persistently exciting, property that TEB or CG do not have. The PD controller performs much worse than the previous adaptive controllers, and the error is not reduced as time goes to infinity, due to the fact that there are no parameters being estimated.

D. Simulation with time-varying parameters

Even with drones, there are occasions where the parameters may be time-varying. Take for example the case shown in Fig. 6, where the drone is doing a grab maneuver¹. In this case, both the inertia matrix and the mass change over time. Assuming that the inertia matrix remains diagonal, we can model the increase of the values of the inertia and mass using a sigmoid:

$$\mathbf{a}(t) = \left(1 + 0.5 \frac{1}{1 - e^{-80(t - \frac{t_{total}}{2})}} \right) \mathbf{a}_{original}$$

¹Images taken from a project a I did with Gabriel Bousquet

where t_{total} is the total time of the simulation and $\mathbf{a}_{original}$ are the constant parameters used in the previous simulation. Using the same desired trajectories as before, the plots of the estimated parameters, the position errors and the control inputs are shown in Fig. 7, 8 and 9 respectively.

Both the parameter estimation (Fig. 7) and the control inputs (Fig. 8) obtained by CG, BGF and CF are again much smoother than the one of TEB. BGF and CF are faster than CG (and than TEB) when the parameter drastically changes in $t = 1.5$ s. This is due to the fact that BGF and CF are changing the gain in the adaptation law, which allows them to easily forget past data and focus on the new data produced by the new parameters.

In the error on the orientation, BGF and CF usually converge faster than CG, and all these three have have a similar behaviour in \tilde{z} (the error in the altitude). It is also important to note that both in this time-varying case, and also in the constant-parameters case, \tilde{z} is better for TEB than for the other adaptive controllers. This may be due to the gains chosen, that lead to TEB having less error in this particular state even when the estimation of the parameters is slightly worse.

IV. FEEDBACK LINEARIZATION

Once the unkown parameters of the quadrotor have been estimated accurately, one natural question that arises is how we could track an arbitrary trajectory for this underactuated system. If we are interested in tracking only $[z_d \ \phi_d \ \theta_d \ \psi_d]^T$, and we assume that the parameters were estimated exactly, we could simply use computed torque (feedback linearization) for the subsystem formed by these states to get an exponentially stable closed loop dynamics. However, it is usually of much more interest to desire to track a trajectory specified by $[x_d \ y_d \ z_d \ \psi_d]^T$. As it was proven in [18], quadrotors are differentiably flat, and therefore there is a one-to-one mapping between the flat outputs $[x(t) \ y(t) \ z(t) \ \psi(t)]^T$ and the state vector $[x(t) \ y(t) \ \dot{\mathbf{q}}^T(t) \ \dot{x}(t) \ \dot{y}(t) \ \dot{\mathbf{q}}^T(t)]^T$. Hence, one could try to obtain the whole desired state vector from the desired trajectory specified in the output space, and then use the computed torque mentioned above to track $\mathbf{q}(t)$ and $\dot{\mathbf{q}}(t)$. However, this clearly will not guarantee the convergence of x and y : as a simple counterexample, two quadrotors can be flying with zero error in $\mathbf{q}(t)$ and $\dot{\mathbf{q}}(t)$ (in the Euler angles, in z and in their derivatives), but with an offset in x and/or y .

Another way to try to achieve this is to use feedback linearization with input-output decoupling via dynamic extension [14]. Choosing $\mathbf{y}^T = \mathbf{h}^T(\mathbf{x}) = [x \ y \ z \ \psi]^T$ as the outputs of interest, it can be shown ([8], [19]) that input-output decoupling is possible via a static state feedback control law if the

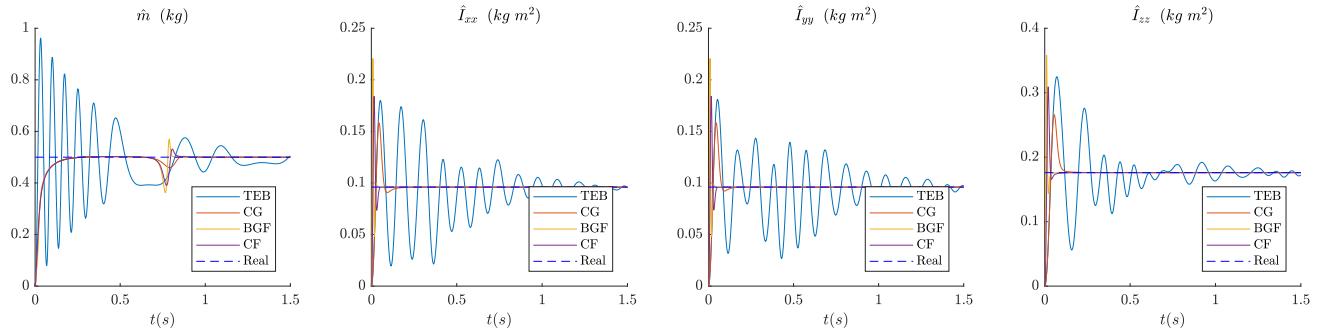


Figure 3: Estimated parameters for the simulation with constant parameters

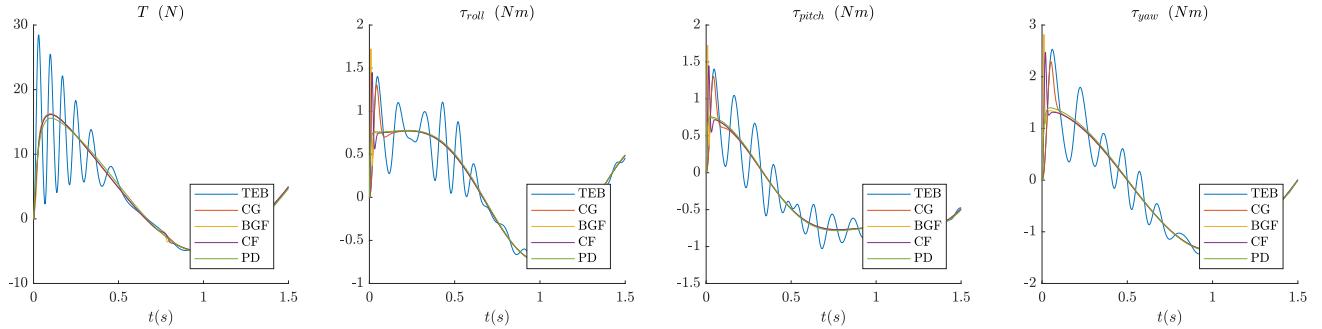


Figure 4: Control inputs for the simulation with constant parameters

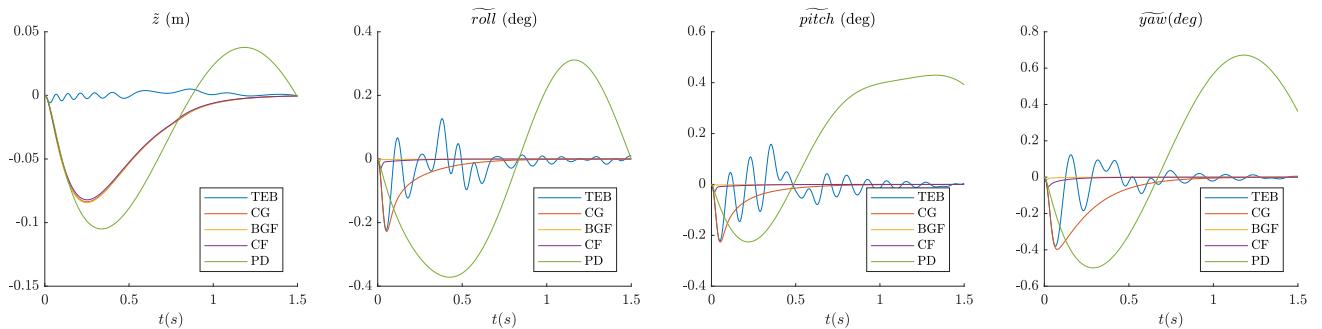


Figure 5: Position errors for the simulation with constant parameters



Figure 6: Sequence of movements in a grabbing maneuver of a hexacopter in our lab

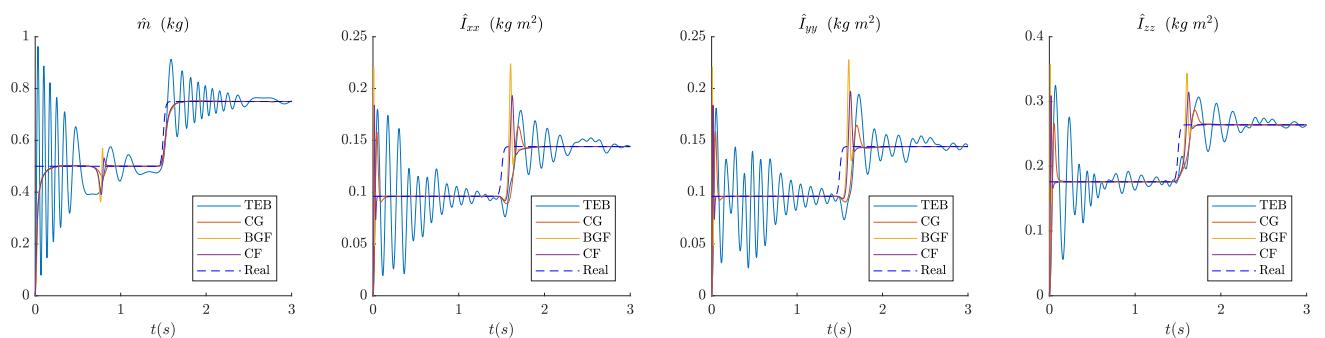


Figure 7: Estimated Parameters for the simulation with time-varying parameters

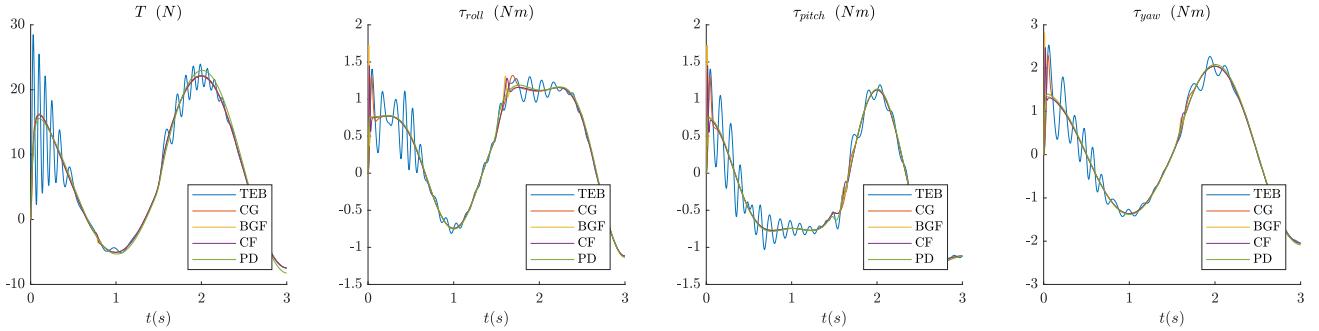


Figure 8: Control Inputs for the simulation with time-varying parameters

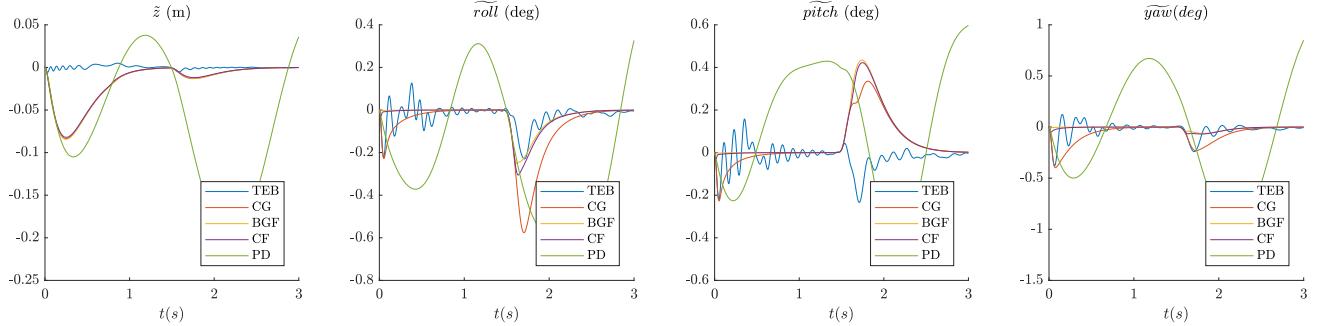


Figure 9: Position Errors for the simulation with time-varying parameters

matrix $\Delta(\mathbf{x})$ defined as $[\Delta(\mathbf{x})]_{ij} := \mathcal{L}_{\mathbf{g}_j} \mathcal{L}_{\mathbf{f}}^{r_i-1} \mathbf{h}_i$ is nonsingular. Here the operator $\mathcal{L}_{\mathbf{f}}^i h$ denotes the i -th Lie derivative of the scalar h with respect to \mathbf{f} , and r_i is the relative degree of the i -th output (number of times output y_i must be differentiated before at least one of the inputs appear). Let us rewrite the system 1 as

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}) + \sum_{i=1}^4 \mathbf{g}_i(\mathbf{x}) \mathbf{u}_i \\ \mathbf{x}^T &= [\ x \ y \ z \ \phi \ \theta \ \psi \ \dot{x} \ \dot{y} \ \dot{z} \ \dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T \\ \mathbf{u}^T &= \boldsymbol{\tau}^T = [\ T \ \tau_{roll} \ \tau_{pitch} \ \tau_{yaw}]^T\end{aligned}$$

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \frac{(c_\phi c_\psi s_\theta + s_\phi s_\psi)T}{(-c_\psi s_\phi + m c_\phi s_\theta s_\psi)T} \\ \frac{(-c_\psi s_\phi + m c_\phi s_\theta s_\psi)T}{(c_\phi c_\psi s_\theta + s_\phi s_\psi)T} \\ -g + \frac{m c_\theta c_\phi T}{I_{xx}} \\ \frac{I_{yy} \dot{\theta} \dot{\psi} - I_{zz} \dot{\theta} \dot{\psi} + \tau_{roll}}{I_{yy}} \\ \frac{-I_{xx} \dot{\theta} \dot{\psi} + I_{zz} \dot{\phi} \dot{\psi} + \tau_{pitch}}{I_{zz}} \\ \frac{I_{yy} \dot{\theta} \dot{\phi} - I_{yy} \dot{\theta} \dot{\phi} + \tau_{yaw}}{I_{zz}} \end{bmatrix}$$

It is clear that for this system we have that $r_i = 2 \forall i$. We could proceed and compute the whole matrix $\Delta(\mathbf{x})$. But note that if we focus on the second column of $\Delta(\mathbf{x})$, and denoting $\dot{\mathbf{h}}^T(\mathbf{x}) = [\ \dot{x} \ \dot{y} \ \dot{z} \ \dot{\psi}]^T$ we have that

$$[\Delta(\mathbf{x})]_{i2} = \mathcal{L}_{\mathbf{g}_2} \mathcal{L}_{\mathbf{f}}^{r_i-1} \mathbf{h}_i = \mathcal{L}_{\mathbf{g}_2} \dot{\mathbf{h}}_i = \mathbf{0}_{4 \times 1}$$

And therefore the matrix is singular. One option we can try to solve this is to use the so-called dynamic extension. [14], [8]. The idea is to redefine the inputs (and therefore increase the dimension of the state space) so that $\Delta(\mathbf{x})$ for the augmented system is nonsingular. This basically provides a dynamic feedback controller for the original system. For quadrotors, we can try to use the *acceleration* in the thrust \ddot{T} as new input, and include T and \dot{T} as states. Hence, we have now this augmented system [8]:

$$\bar{\mathbf{x}}^T = [\ \mathbf{x}^T \ T \ \dot{T}]^T \quad \bar{\mathbf{u}}^T = [\ \ddot{T} \ \tau_{roll} \ \tau_{pitch} \ \tau_{yaw}]^T$$

$$\bar{\mathbf{f}}(\bar{\mathbf{x}}) = \begin{bmatrix} \mathbf{f}(\mathbf{x}) \\ \dot{T} \\ 0 \end{bmatrix} \quad \bar{\mathbf{g}}_1(\bar{\mathbf{x}}) = \begin{bmatrix} \mathbf{0}_{6 \times 1} \\ \frac{(c_\phi c_\psi s_\theta + s_\phi s_\psi)T}{(-c_\psi s_\phi + m c_\phi s_\theta s_\psi)T} \\ \frac{(-c_\psi s_\phi + m c_\phi s_\theta s_\psi)T}{(c_\phi c_\psi s_\theta + s_\phi s_\psi)T} \\ -g + \frac{m c_\theta c_\phi T}{m} \\ \mathbf{0}_{3 \times 1} \end{bmatrix}$$

$$\bar{\mathbf{g}}_i^T(\bar{\mathbf{x}}) = [\ \mathbf{g}_i^T(\mathbf{x}) \ \ \mathbf{0}_{1 \times 2}]^T, i = \{2, 3, 4\}$$

For this augmented system, $r_i = 4$, $i = \{1, 2, 3\}$ and $r_4 = 2$ (and therefore $\sum r_i = 14 = n$). We can compute now the matrix $\Delta(\bar{\mathbf{x}})$ and show that it is nonsingular. The matrix I have obtained (using *Mathematica*) is available in [this link](#), but omitted in this report for the sake of brevity. With this matrix, we can finally use the following control law:

$$\bar{\mathbf{u}} = -\Delta(\bar{\mathbf{x}})^{-1} \mathbf{b}(\bar{\mathbf{x}}) + \Delta(\bar{\mathbf{x}})^{-1} \mathbf{v} \implies \dots$$

$$\dots \implies \mathbf{y}^{(r_i)} = \mathbf{b}(\bar{\mathbf{x}}) + \Delta(\bar{\mathbf{x}}) \bar{\mathbf{u}} = \mathbf{v}$$

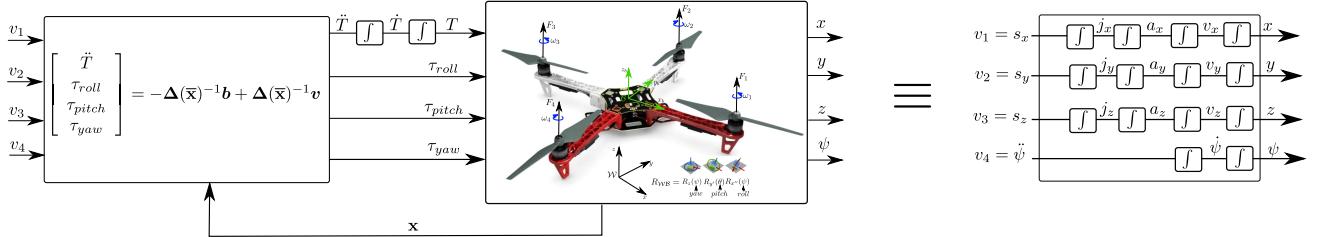


Figure 10: Feedback linearization and input-output decoupling of a quadrotor using dynamic extension. s , j , a and v denote snap, jerk, acceleration and velocity respectively. Based on [8].

where $[b(\bar{x})]_i = \mathcal{L}_f^{r_i} h_i$. This means that x , y and z behave like three independent 4th order integrator models driven by input v_i ($i = \{1, 2, 3\}$), and that the yaw angle ψ behaves like a 2nd order integrator model driven by v_4 (see Fig. 10). This result is in fact closely linked to the differential flatness of the quadrotors. And these two results make path planning for quadrotors relatively simple: We can assume linear integrator models for $[x \ y \ z \ \psi]^T$ and solve a simple convex optimization problem ([20], [21], [22], [23], [24]).

V. CONCLUSIONS AND FUTURE WORK

In this paper I studied how the TEB, CG, BFG and CF adaptive controllers can be applied to a quadrotor to do parameter estimation. I did different simulations with both constant and time-varying parameters, showing that BFG and CF can achieve better performance, and have much less oscillatory behavior than TEB or CG. Once the parameters were estimated, I presented a way to feedback linearize and input-outut decouple the dynamics of the quadrotor using dynamic extension.

As future work, I would like to continue exploring the feedback linearization. Particularly, I would like to use robust input-output feedback linearization [25] to handle the case where some parameters were estimated with some error, or there are some disturbances, and therefore the feedback linearization is not exact. Moreover, in this project I have decoupled the parameter estimation task from the tracking of the flat variables $[x_d \ y_d \ z_d \ \psi_d]^T$ task. I would like to investigate if there is a way to do both things at the same time (i.e. achieve a good tracking of $[x_d \ y_d \ z_d \ \psi_d]^T$ when the inertia matrix and the mass of the drone are completely unknown).

REFERENCES

- [1] Weiping Li and Jean-Jacques E Slotine. An indirect adaptive robot controller. *Systems & control letters*, 12(3):259–266, 1989.
- [2] Jean-Jacques E Slotine and Weiping Li. On the adaptive control of robot manipulators. *The international journal of robotics research*, 6(3):49–59, 1987.
- [3] Zachary T Dydek, Anuradha M Annaswamy, and Eugene Lavretsky. Adaptive control of quadrotor uavs: A design trade study with flight evaluations. *IEEE Transactions on control systems technology*, 21(4):1400–1406, 2013.
- [4] C Nicol, CJB Macnab, and A Ramirez-Serrano. Robust adaptive control of a quadrotor helicopter. *Mechatronics*, 21(6):927–938, 2011.
- [5] Brett T Lopez, Jean-Jacques Slotine, and Jonathan P How. Robust collision avoidance via sliding control. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2962–2969. IEEE, 2018.
- [6] Brett Thomas Lopez. *Low-latency trajectory planning for high-speed navigation in unknown environments*. PhD thesis, Massachusetts Institute of Technology, 2016.
- [7] Abhijit Das, Kamesh Subbarao, and Frank Lewis. Dynamic inversion with zero-dynamics stabilisation for quadrotor control. *IET control theory & applications*, 3(3):303–314, 2009.
- [8] V Mistler, A Benallegue, and NK M’sirdi. Exact linearization and noninteracting control of a 4 rotors helicopter via dynamic feedback. In *Proceedings 10th IEEE International Workshop on Robot and Human Interactive Communication. ROMAN 2001 (Cat. No. 01TH8591)*, pages 586–593. IEEE, 2001.
- [9] Dong Eui Chang and Yongsun Eun. Global chartwise feedback linearization of the quadcopter with a thrust positivity preserving dynamic extension. *IEEE Transactions on Automatic Control*, 62(9):4747–4752, 2017.
- [10] Euler matrix, wolfram language documentation. <https://reference.wolfram.com/language/ref/EulerMatrix.html>. (Accessed on 05/06/2019).
- [11] Francesco Sabatino. Quadrotor control: modeling, nonlinear control design, and simulation, 2015.
- [12] Nathan Michael. Quadrotor modeling and control. <http://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/16311/www/s15/syllabus/ppp/Lec08-Control3.pdf>. (Accessed on 05/06/2019).
- [13] Russ Tedrake. Underactuated robotics: Algorithms for walking, running, swimming, flying, and manipulation (course notes for mit 6.832). <http://underactuated.csail.mit.edu/tocite.html>. (Accessed on 05/07/2019).
- [14] Jean-Jacques E Slotine, Weiping Li, et al. *Applied nonlinear control*, volume 199. Prentice hall Englewood Cliffs, NJ, 1991.
- [15] JJ E Slotine. Theoretical issues in adaptive manipulator control. In *Proc. Fifth Yale Workshop on the Applications of Adaptive Systems Theory*, pages 252–258, 1987.
- [16] Jean-Jacques E Slotine and Weiping Li. Composite adaptive control of robot manipulators. *Automatica*, 25(4):509–519, 1989.
- [17] Jean-Jacques E Slotine and Weiping Li. Estimation of time-varying parameters. *MIT NSL Report*, 1988.
- [18] Daniel Mellinger and Vijay Kumar. Minimum snap trajectory generation and control for quadrotors. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2520–2525. IEEE, 2011.
- [19] Henk Nijmeijer and Arjan Van der Schaft. *Nonlinear dynamical control systems*, volume 175. Springer, 1990.
- [20] Brett T Lopez and Jonathan P How. Aggressive 3-d collision avoidance for high-speed navigation. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5759–5765. IEEE, 2017.

- [21] Helen Oleynikova, Michael Burri, Zachary Taylor, Juan Nieto, Roland Siegwart, and Enric Galceran. Continuous-time trajectory optimization for online uav replanning. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5332–5339. IEEE, 2016.
- [22] Jesus Tordesillas, Brett T Lopez, and Jonathan P How. Fas-trap: Fast and safe trajectory planner for flights in unknown environments. *arXiv preprint arXiv:1903.03558*, 2019.
- [23] Jesus Tordesillas, Brett T Lopez, John Carter, John Ware, and Jonathan P How. Real-time planning with multi-fidelity models for agile flights in unknown environments. *arXiv preprint arXiv:1810.01035*, 2018.
- [24] Shupeng Lai, Kangli Wang, Kun Li, and Ben M Chen. Path planning of rotorcrafts in unknown environment. In *Control Conference (CCC), 2016 35th Chinese*, pages 10900–10905. IEEE, 2016.
- [25] Jean-Jacques E Slotine and J KARL HEDRICK. Robust input-output feedback linearization. *International Journal of control*, 57(5):1133–1139, 1993.