

A Data-driven Hierarchical Control Structure for Systems with Uncertainty

Lu Shi, Hanzhe Teng, Xinyue Kan, and Konstantinos Karydis

Abstract—The paper introduces a Data-driven Hierarchical Control (DHC) structure to improve performance of systems operating under the effect of system and/or environment uncertainty. The proposed hierarchical approach consists of two parts: 1) A data-driven model identification component to learn a linear approximation between reference signals given to an existing lower-level controller and uncertain time-varying plant outputs. 2) A higher-level controller component that utilizes the identified approximation and wraps around the existing controller for the system to handle modeling errors and environment uncertainties during system deployment. We derive loose and tight bounds for the identified approximation’s sensitivity to noisy data. Further, we show that adding the higher-level controller maintains the original system’s stability. A benefit of the proposed approach is that it requires only a small amount of observations on states and inputs, and it thus works online; that feature makes our approach appealing to robotics applications where real-time operation is critical. The efficacy of the DHC structure is demonstrated in simulation and is validated experimentally using aerial robots with approximately-known mass and moment of inertia parameters and that operate under the influence of ground effect.

I. INTRODUCTION

As robots increasingly venture outside of the lab, the effect of uncertainty within the system and/or at the robot-environment interactions becomes more pronounced. Motivating examples include the influence of surface effects for Unmanned Aerial Vehicles (UAVs), unknown flow dynamics for marine robots, and inherently uncertain leg-ground contacts for legged robots.

Central to exploiting system and robot-environment uncertainty is the ability to quickly identify deviations from nominal behaviors based on data collected during robot deployment. Pre-deployment model identification and calibration tools (e.g., [1]–[3]) can help improve model-based control. However, even though such models can be obtained with high precision, the presence of uncertain, time-varying disturbances during deployment may turn the utilized model invalid to the extent that an otherwise well-tuned controller will fail [4]. A way to tackle this is to incorporate Uncertainty Quantification (UQ) into control design.

Methods using UQ, like a-priori estimation, rely heavily on the employed model structure. Generated models are usually complex and hard to incorporate into controller design in practice [5], [6]. When prior information about

The authors are with the Dept. of Electrical and Computer Engineering, University of California, Riverside. Email: {lshi024, hteng007, xkan001, karydis}@ucr.edu.

We gratefully acknowledge the support of NSF under grants #IIS-1910087 and #IIS-1724341, and ONR under grant #N00014-19-1-2264. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

the environment is limited, methods based on principles of adaptive control [7] can apply. An appealing feature of these methods is that they can provide some form of performance and safety guarantees (e.g., regarding system stability). Despite their overall effectiveness, such methods can still be challenged in two key ways. First, when the intensity of the uncertainty affecting system behavior violates the underlying controller assumptions. Second, when modeling errors [8], such as unmodeled dynamics, render the model invalid [4]. To address the problem of unmodeled dynamics, data-driven control techniques have been proposed.

Machine learning methods, such as Gaussian Process [9], [10], or Deep Neural Networks [11], can be used to either model the system [12], [13] or determine control inputs [14], [15] following a black-box input-output training procedure. However, as state dimensionality and system structure complexity increase (e.g., by using ‘deeper’ neural networks), the aforementioned methods may be challenged when it comes to be implemented in real-time for robotics research (e.g., [16]–[18]). One way to address this issue is by adopting a hierarchical structure to involve deep neural networks in control design. Instead of designing new controllers directly, reference trajectories are generated to deal with uncertainty through learning [19], [20]. Unfortunately, certain limitations still exist even when adopting a hierarchical structure. Neural-network-based approaches require a large body of data to train well [21]. At the same time, they remain limited in terms of offering some form of performance and safety guarantees, although this is currently an active research topic (e.g., [20], [22], [23]).

Besides neural-network-based approaches, dimensionality reduction and dynamics decomposition approaches can play an important role in data-driven algorithms. Methods like DMD (Dynamic Mode Decomposition [24]), EDMD (Extended DMD [25]), POD (Proper Orthogonal Decomposition [26], [27]) and their various kernel [28] and tensor [29] extensions have been successfully applied to across areas. A benefit of decomposition algorithms is that they can significantly reduce the amount of data required for approximating a system’s model through data (e.g., [30]–[32]).

Fueled by the potential of hierarchical methods and dimensionality-reduction approaches, this paper presents a new Data-driven Hierarchical Control (DHC) structure to handle uncertainty. The approach hinges on DMD with Control (DMDc [33]) to learn a higher-level controller and then generate a refined reference to improve an existing lower-level controller’s performance (Fig. 1). Our approach can be particularly appropriate in practice when a low-level, high-rate, pre-tuned controller is already in place, and a

second higher-level controller is wrapped around the lower-level one to allow for the system to operate under uncertainty. The quadrotor UAV is one illustrative case, where it is now common practice to employ a hierarchical control structure [34], [35]. The low-level controller is responsible for controlling the attitude of the robot, while the high-level controller determines its position.

Succinctly, the paper's contribution is twofold.

- We propose a hierarchical control structure that refines a reference signal sent to a lower-level controller to deal with uncertainty. The structure builds on top of a model-identification block based on DMDc, and is shown that it retains stability of the underlying controller.
- We analyze the sensitivity of DMDc to noisy data, and provide loose and tight bounds for the model identification component of our method.

Furthermore, we evaluate and validate the methodology using aerial robots both in simulation and experimentally. In turn, this effort can help design controllers able to learn how to harness uncertain aerodynamics, such as ground effect. A supplementary video can be found at <https://youtu.be/OznDCskVnJU>.

II. TECHNICAL BACKGROUND

Dynamic Mode Decomposition (DMD) can characterize nonlinear dynamics through analysis of an approximated linear system [36]. Most relevant to our work here, DMD with control (DMDc)—an extension of the original DMD [37]—aims to incorporate the effect of external inputs [33].

The goal of DMDc is to analyze the relation between a future system measurement $\xi_{k+1} \in \mathbb{R}^n$, a current measurement $\xi_k \in \mathbb{R}^n$, and the current input $u_k \in \mathbb{R}^m$. For each triplet of measurement data (ξ_{k+1}, ξ_k, u_k) , a pair of linear operators $A \in \mathbb{R}^{n \times n}$, and $B \in \mathbb{R}^{n \times m}$ is determined to approximate

$$\xi_{k+1} \approx A\xi_k + Bu_k. \quad (1)$$

Operators A and B are selected as the best-fit solution for all triplets of available data. Given observations and inputs up to time instant M arranged in vectors $\Xi = [\xi_1, \xi_2 \dots, \xi_{M-1}]$, $\Xi_P = [\xi_2, \xi_3 \dots, \xi_M]$, and $U = [u_1, u_2 \dots, u_{M-1}]$, approximation (1) can be rewritten compactly as

$$\begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} \Xi \\ U \end{bmatrix} = A_P \Omega^T \approx \Xi_P. \quad (2)$$

Then we could seek the best-fit solution as:

$$A_P = \Xi_P(\Omega^T)^\dagger,$$

where \dagger denotes the pseudo-inverse. The problem can be solved immediately by Singular Value Decomposition (SVD), or QR decomposition, among others [38], [39].

III. DATA-DRIVEN HIERARCHICAL CONTROL (DHC)

A. Controller Structure

Given an existing low-level controller for a plant of interest, the proposed DHC structure adds a model identification block and a higher-level (“outer”) controller to extract

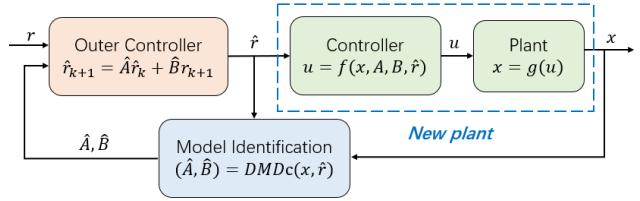


Fig. 1. Overview of the Data-driven Hierarchical Control (DHC) structure proposed in this work. Model identification (via spectral methods) is combined with a linear controller to adjust the reference input given to an existing (pre-tuned) lower-level controller.

uncertainties and remedy for them in real-time, respectively (Fig. 1). The low-level controller and the original plant are combined into a new plant. The model identification block is built from DMDc and estimates linear operators \hat{A} and \hat{B} from new plant output and reference signals. The identified operators are then used by the higher-level controller to refine the reference signal sent to the new plant.

B. Control Procedure

At timestep $k+1$, the outer controller refines the original reference r_{k+1} given to the lower-level controller as per $\hat{r}_{k+1} = \hat{A}_k \hat{r}_k + \hat{B}_k r_{k+1}$. To estimate \hat{A}_k and \hat{B}_k using DMDc, we treat the measured state x_k as the “input” and the refined references \hat{r}_k and \hat{r}_{k+1} as the evolving “output.” The reason for doing so is that the controller operates on the reference signal, which in turn needs to be “learned” based on observed system state. Then, DMDc operates on the linear approximation $\hat{r}_{k+1} = \hat{A}_k \hat{r}_k + \hat{B}_k x_k$. Note that to avoid singularities, we check the rank of the measurement matrices used to approximate models. After an initialization for M steps,¹ the identification process repeats to refine the estimated model until the control task is finished (Algorithm 1).

Algorithm 1: DHC Procedure

```

1 initialize: Set  $\hat{r}_i = r_i$  and evolve the system for the
   first  $M$  time steps. Then compute
    $(\hat{A}_0, \hat{B}_0) = DMDc(\{\hat{r}_i\}_{i=1}^M, \{x_i\}_{i=1}^M)$ 
2 for  $k \geq M$  do
3   repeat
4     Outer Controller: Refine reference based on
       approximated model:  $\hat{r}_{k+1} = \hat{A}_k \hat{r}_k + \hat{B}_k r_{k+1}$ .
5     Plant: Propagate the system with  $\hat{r}_{k+1}$ :
        $x_{k+1} = g(\hat{r}_{k+1}, w_k)$ , where  $g$  is the evolving
       law of real plant and  $w_k$  the uncertainty.
6     Model: Check rank of measurement matrices.
7     if full column rank then
8        $(\hat{A}_k, \hat{B}_k) = DMDc(\{\hat{r}_i\}^{k+1}, \{x_i\}^{k+1})$ 
9     else
10       $(\hat{A}_k, \hat{B}_k) = (\hat{A}_{k-1}, \hat{B}_{k-1})$ 
11       $k \leftarrow k + 1$ 
12 until control task is finished;

```

¹From a general standpoint, (2) can be solved when $m + n + 1$ linearly independent measurements are available [33]. Thus, a lower bound for M is $m + n + 1$. In this work we set $M = m + n + 1$.

C. Stability Analysis

Let the underlying low-level controller be stable, and consider the origin as the reference, i.e. $r_k = 0$. Given the Lyapunov function $V(x_k) = x_k^T x_k$, we calculate $\Delta V(x_k) = V(x_{k+1}) - V(x_k) = g(\hat{r}_{k+1}, w_{k+1})^T g(\hat{r}_{k+1}, w_{k+1}) - x_k^T x_k$. When $\hat{r}_{k+1} = \hat{A}_k \hat{r}_k + \hat{B}_k r_{k+1}$, we have $g(\hat{r}_{k+1}, w_{k+1}) = r_{k+1} = 0$. Then, $\Delta V(x_k) = r_{k+1}^T r_{k+1} - x_k^T x_k = 0 - x_k^T x_k \rightarrow \Delta V(x_k) = -x_k^T x_k$.

Thus, V decays over time and satisfies $V(0) = 0$. From the standard Lyapunov argument, the equilibrium $r_k = 0$ is then stable. Hence, wrapping the outer controller around the underlying plant respects the stability properties of the original lower-level controller.

IV. SENSITIVITY TO NOISY DATA

The model identification block extracts dynamics when measurement data are noisy. Below we analyze the sensitivity to noisy data and derive two estimation error bounds.

Lemma 2.1. Consider the system

$$\begin{aligned} H\mathbf{z} &= \mathbf{s}, & H \in \mathbb{R}^{p \times q}, \mathbf{s} \in \mathbb{R}^{p \times t} \\ (H + \delta H)\hat{\mathbf{z}} &= \mathbf{s} + \delta\mathbf{s}, & \delta H \in \mathbb{R}^{p \times q}, \delta\mathbf{s} \in \mathbb{R}^{p \times t} \end{aligned} \quad (3)$$

The sensitivity of system (3) to perturbations in \mathbf{s} and H is

$$\frac{\|\hat{\mathbf{z}} - \mathbf{z}\|}{\|\mathbf{z}\|} \leq (\kappa(H)^2 \tan(\theta) + \kappa(H)) \frac{\|\delta H\|}{\|H\|} + \kappa(H) \sec(\theta) \frac{\|\delta\mathbf{s}\|}{\|\mathbf{s}\|}. \quad (4)$$

where $\kappa(H)$ is the condition number of linear operator H , and $\theta = \arccos \frac{\|H\|}{\|\mathbf{s}\|}$.

Proof of Lemma 2.1: Start with the normal equations

$$H^T H \mathbf{z} = H^T \mathbf{s}.$$

The first-order perturbation relation is

$$\delta H^T H \mathbf{z} + H^T \delta H \mathbf{z} + H^T H \delta\mathbf{z} = \delta H^T \mathbf{s} + H^T \delta\mathbf{s},$$

which we re-arrange to get

$$\delta\mathbf{z} = (H^T H)^{-1} \delta H^T (\mathbf{s} - H\mathbf{z}) + (H^T H)^{-1} H^T (\delta\mathbf{s} - \delta H\mathbf{z}).$$

Define $r = \mathbf{s} - H\mathbf{z}$, and let $\sigma_1, \dots, \sigma_p$ be singular values of H . Then,

$$\begin{aligned} \|\delta\mathbf{z}\| &\leq \|(H^T H)^{-1}\| \|\delta H\| \|r\| \\ &\quad + \|(H^T H)^{-1} H^T\| (\|\delta\mathbf{s}\| + \|\delta H\| \|\mathbf{z}\|). \end{aligned}$$

Setting $\|H\| = \sigma_1$, $\|(H^T H)^{-1}\| = 1/\sigma_q^2$, we can derive [40]

$$\|\delta\mathbf{z}\| \leq \frac{\|\delta H\|}{\sigma_q^2} \|r\| + \frac{1}{\sigma_q} (\|\delta\mathbf{s}\| + \|\delta H\| \|\mathbf{z}\|).$$

Dividing both sides by $\|\mathbf{z}\|$ and after some algebraic manipulation so that $\|\delta H\|$ and $\|\delta\mathbf{s}\|$ only appear in ratios of $\frac{\|\delta H\|}{\|H\|}$ and $\frac{\|\delta\mathbf{s}\|}{\|\mathbf{s}\|}$, as $\kappa(H) = \sigma_1/\sigma_q$, we get

$$\frac{\|\delta\mathbf{z}\|}{\|\mathbf{z}\|} \leq \kappa(H)^2 \frac{\|r\|}{\|H\| \|\mathbf{z}\|} \frac{\|\delta H\|}{\|H\|} + \kappa(H) \left(\frac{\|\mathbf{s}\|}{\|H\| \|\mathbf{z}\|} \frac{\|\delta\mathbf{s}\|}{\|\mathbf{s}\|} + \frac{\|\delta H\|}{\|H\|} \right).$$

Then with

$$\begin{aligned} \frac{\|r\|}{\|H\| \|\mathbf{z}\|} &\leq \frac{\|r\|}{\|H\| \|\mathbf{z}\|} = \tan(\theta), \\ \frac{\|\mathbf{s}\|}{\|H\| \|\mathbf{z}\|} &\leq \frac{\|\mathbf{s}\|}{\|H\| \|\mathbf{z}\|} = \sec(\theta), \end{aligned}$$

we arrive at (4). \blacksquare

Corollary 2.2. The estimation error of problem (2) is bounded by the measurements' noise intensity, i.e.

$$\frac{\|\delta A_P\|}{\|A_P\|} \leq (\kappa(\Omega)^2 \tan(\theta) + \kappa(\Omega)) \frac{\|\delta\Omega\|}{\|\Omega\|} + \kappa(\Omega) \sec(\theta) \frac{\|\delta\Xi_P\|}{\|\Xi_P\|}, \quad (5)$$

where $\kappa(\Omega)$ is the condition number of observation matrix Ω , and $\theta = \arccos \frac{\|A_P \Omega\|}{\|\Xi_P\|}$.

Proof of Corollary 2.2: The proof follows directly from Lemma 2.1 by setting $H = \Omega$, $\mathbf{z} = A_P^T$, and $\mathbf{s} = \Xi_P^T$. \blacksquare

Lemma 2.3 [A tighter bound]. Let $\epsilon = \frac{\|\delta\mathbf{s}\|}{\|\mathbf{s}\|}$. In the case that $\|\delta\mathbf{s}\| = \|\delta H\|$, the sensitivity of (3) is reduced to

$$\frac{\|\hat{\mathbf{z}} - \mathbf{z}\|}{\|\mathbf{z}\|} \leq \epsilon \kappa(H) (1 + \|\hat{\mathbf{z}}\|). \quad (6)$$

Proof of Lemma 2.3: From system (3), we have

$$H(\hat{\mathbf{z}} - \mathbf{z}) = \delta\mathbf{s} - \delta H \hat{\mathbf{z}},$$

and it follows that

$$\hat{\mathbf{z}} - \mathbf{z} = (H^T H)^{-1} H^T \delta\mathbf{s} - (H^T H)^{-1} H^T \delta H \hat{\mathbf{z}}.$$

Then,

$$\|\hat{\mathbf{z}} - \mathbf{z}\| = \|(H^T H)^{-1} H^T\| (\|\delta\mathbf{s}\| + \|\delta H\| \|\hat{\mathbf{z}}\|).$$

Setting $\|\delta\mathbf{s}\| = \|\delta H\|$ yields

$$\begin{aligned} \|\hat{\mathbf{z}} - \mathbf{z}\| &= \|(H^T H)^{-1} H^T\| \|\delta\mathbf{s}\| (1 + \|\hat{\mathbf{z}}\|) \\ &= \epsilon \|(H^T H)^{-1} H^T\| \|\mathbf{s}\| (1 + \|\hat{\mathbf{z}}\|) \\ &\leq \epsilon \|(H^T H)^{-1} H^T\| \|H\| \|\mathbf{z}\| (1 + \|\hat{\mathbf{z}}\|) \\ &= \epsilon \kappa(H) \|\mathbf{z}\| (1 + \|\hat{\mathbf{z}}\|), \end{aligned}$$

which concludes the proof. \blacksquare

Corollary 2.4. When the disturbance occurs in observed states, i.e. $\|\delta\Xi_P\| = \|\delta\Xi\| = \|\delta\Omega\|$ a tighter error bound is

$$\frac{\|\delta A_P\|}{\|A_P\|} \leq \kappa(\Omega) \frac{\|\delta\Xi_P\|}{\|\Xi_P\|} (1 + \|\hat{A}_P\|). \quad (7)$$

Proof of Corollary 2.4: The proof follows directly from Lemma 2.3 by setting $H = \Omega$, $\mathbf{z} = A_P^T$, and $\mathbf{s} = \Xi_P^T$. \blacksquare

Toy Example: Consider the simple linear system

$$\begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}_k + \begin{bmatrix} 1 \\ 0.7 \end{bmatrix} u_k. \quad (8)$$

We add zero-mean white noise to corrupt state observations, i.e. $N(0, \sigma^2)$, with $\sigma \in [0.05, 1]$ in 0.05 increments. The initial condition is set at $[0.01, 0.01]$. We perform a Monte Carlo simulation for 10,000 trials to produce the average error. With reference to Fig. 2, the effect of the disturbance increases as variance increases (as expected). The general bound given in (5) expands at higher rates than the tighter one we obtain via (7) as the noise variance increases.

V. CONTROL STRUCTURE EXPERIMENTS

We demonstrate the utility and evaluate the performance of DHC in simulating a planar quadrotor with different noise intensity, and then test experimentally with a quadrotor flying under the influence of ground effect.

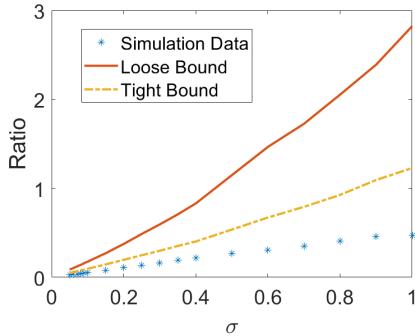


Fig. 2. Estimation sensitivity to perturbation $\frac{\|\delta A_P\|}{\|A_P\|}$ and predicted bounds as noise magnitude increases.

A. Simulation in a Planar Quadrotor

While quadrotor dynamics is well understood, there is a certain limit on the degree of variations in parameters like mass and moment of inertia that model-based controllers can handle. Such variations make the utilized model inaccurate, which in turn may render the controller unstable. Online adjustment can help quantify model errors and inform the controller so as to maintain stable operation.

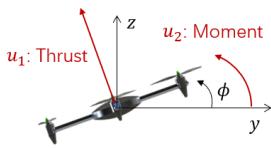


Fig. 3. Planar quadrotor.

1) *System Description:* The planar quadrotor depicted in Fig. 3 can be modeled as [41]

$$\begin{cases} m\ddot{y} = -u_1 \sin \phi \\ m\ddot{z} = u_1 \cos \phi - mg \\ I_{xx}\ddot{\phi} = u_2 \end{cases} .$$

Linearization around hover configurations yields

$$\dot{\xi} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \\ 0 & 0 & -g \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \xi + \begin{bmatrix} \mathbf{0}_{4 \times 2} \\ 1/m & 0 \\ 0 & 1/I_{xx} \end{bmatrix} u, \quad (9)$$

where $\xi = [y, z, \phi, \dot{y}, \dot{z}, \dot{\phi}]$ and $u = [u_1 - mg, u_2]$. In all simulations, nominal parameter values match those of the physical Crazyflie robot used later for experiments, that is, $m = 0.03kg$, $I_{xx} = 1.43 \times 10^{-5}kgm^2$, and $g = 9.8m/s^2$.

2) *Controller Design:* Besides our proposed DHC structure, we design and compare against a State Feedback Controller (SFC), a linear Model Predictive Controller (MPC), and a Model Reference Adaptive Controller (MRAC).

- *State Feedback Controller (SFC):* With the closed-loop system poles set at $[0.9; 0.8; -0.9; -0.8; 0.95; -0.95]$, we get

$$K_p = \begin{bmatrix} -0.0001 & -0.0243 & -0.0008 & 0.0012 & 0 & 0.0149 \\ -0.0000 & -0.0000 & -0.0000 & -0.0000 & 0 & -0.0000 \end{bmatrix} .$$

- *Model Predictive Control (MPC):* A linear MPC is designed directly with model (9) at hand. The prediction and control horizon is set to be 10 and 2, and the pitch angle ϕ is constrained in the interval $[-\pi/2, \pi/2]$.
- *Model Reference Adaptive Control (MRAC):* A parameter adaptation rule is used together with the previously described MPC to adjust the controller parameter so that the output of the plant tracks the output of the reference model having the same reference input. The adaptation rule is designed based on [42] as

$$\begin{aligned} \frac{dm}{dt} &= a_1(y_d - y) + b_1(z_d - z) \\ \frac{dI_{xx}}{dt} &= a_2(y_d - y) + b_2(z_d - z) \end{aligned}$$

where the weights were selected empirically to $a_1 = b_1 = 0.1$ and $a_2 = b_2 = 10^{-4}$. To tune the weights, we first chose their order to be consistent with the physical quantity they relate to. Then, we solved a sequential optimization problem to identify those parameter values that maximize MRAC's performance.

- *DHC:* Our method also uses the linear MPC as the underlying controller. Then, we follow the steps in Algorithm 1 to simulate the system.

3) *Simulation:* The quadrotor's task is to reach position $[3; 5] m$ from initial condition $[0; 0] m$. We analyze the performance of each controller in two cases: when parameters are uncertain with low noise intensity (c1) where $\sigma_m = 0.2m$ and $\sigma_{I_{xx}} = 0.2I_{xx}$, and with high noise intensity (c2) where $\sigma_m = 0.6m$ and $\sigma_{I_{xx}} = 0.6I_{xx}$. Perturbations in model parameters are generated at initialization of each simulated trial by sampling from normal distributions. Distribution means are the nominal values for m and I_{xx} . Distribution variances are selected as above. Once the perturbation is sampled, it remains constant for the duration of the trial.² For every different condition we perform 10 trials.

4) *Results and Discussion:* Simulation results suggest that all controllers can stabilize the system in low noise intensity (see Table I). However, as noise intensity increases SFC fails. Linear MPC converges faster than all other methods, but produces increasingly large steady-state error. MRAC can reduce such steady-state errors, at the expense of longer settling time (within 2%) and larger control effort, and could possibly cause a large overshoot as in SFC. Despite its efficiency shown here, tuning MRAC can be tedious, and its performance relies a lot on the tuned weights. On the other hand, our proposed DHC structure keeps updating the model that implicitly includes noise. As such, it can quickly extract refined dynamics from uncertain data to maintain system stability faster, with less control energy and smaller steady state error. Further, as the disturbance gets embedded in the learned model, the performance of controller does not deteriorate significantly as noise intensity increases.

²The specific selection of constant perturbations in model parameters emulates the case of inaccurate measurements of mass and inertia of the vehicle; these values may be approximately known (or estimated), but do not change during operation. The robustness properties of the proposed control structure to general perturbations is analyzed in Section IV.

TABLE I
PERFORMANCE OF DIFFERENT CONTROLLERS IN LINEAR PLANAR QUADROTOR MODEL

Case	Controller	Settling time T_s (s)	Total control effort to stabilize $\Sigma u $	Steady state error $ x_{desired} - x_{steady} $	Overshoot $ x_{max} - x_{steady} / x_{steady} $
c1	SFC	1.95	1.2453	[0:0]	2.5828
	MPC	0.04	0.6020	[0.6846;0.7740]	0
	MRAC	0.72	11.1852	[0.2986;0.7038]	1.4575
	DHC	0.38	4.0847	[0;0.0011]	0
c2	SFC	Unstable	\	\	\
	MPC	0.04	0.6032	[1.6951;1.1906]	0
	MRAC	0.92	11.0816	[1.0351;1.0357]	0.9718
	DHC	0.38	3.6052	[0;0.0085]	0

B. Controlling a Crazyflie Quadrotor Under Ground Effect

We evaluate experimentally our DHC structure on a Crazyflie quadrotor operating under the influence of ground effect over sustained periods of time. Ground effect [43] manifests itself as an increase in the generated rotor thrust given constant input rotor power [44] when operating close to the ground or over other surfaces. When operating in ground effect, the dynamics of the robot-environment interaction change to a degree that an underlying model tuned for mid-air operation may no longer be valid [4], [27]. In turn, model mismatches caused by varying robot-environment interaction (i.e. a form of uncertainty) degrade the robot's performance, both in terms of stability [45] and control. As we show below, controllers unaware of ground effect will tend to raise the height of the robot when the latter flies sufficiently close over an obstacle. This behavior may lead to crashes when operating in confined and cluttered environments. Wrapping DHC around such a controller can help the robot adjust the impact of ground effect without any specific models for it.

1) *Controller Design*: We follow the controller structure shown in Fig. 1 (as in simulation). We test using two distinct lower-level controllers: a nonlinear geometric controller [46], and a PID controller. Controller gains are those pre-tuned by the manufacturer. Hence, note that the controllers have been tuned for operation in mid-air, not in near-ground operation. The controllers are different from those in simulation since 1) they are often used in practice, and 2) their underlying performance in simulation would depend on quality of tuning (especially for PID) and would thus not add significant value beyond the controllers already tested in simulation.

We evaluate the performance of both low-level controllers with and without DHC wrapped around them. The task is to fly at a constant height between an initial and a goal position (Fig. 4). During part of the trajectory, the robot flies over an obstacle which creates sustained ground effect and influences robot behavior. Performance is quantified in terms of deviating from the desired height due to the ground effect.



Fig. 4. (Left) Experimental setup, and (Right) the Crazyflie. Sample experiments can be viewed at <https://youtu.be/OznDCskVnJU>.

2) *Experiment*: The robot is commanded to follow 2.4 m straight-line constant-height trajectories. The height of the obstacle is $h_o = 0.290$ m. The robot propeller radius is

$r = 0.023$ m. We collect data at three distinct heights $h \in \{h_o + r, h_o + 2r, h_o + 3r\}$, and while flying at six distinct forward speeds $v \in \{0.4, 0.6, 0.8, 1.0, 1.2, 1.4\}$ m/s. Experiments are run with the four aforementioned control structures. This leads to a total of 72 distinct case studies. For each case study, we collect data from 10 repeated trials. To minimize depleting battery effects, a fully charged battery is used at the beginning of each 10-trial experimental session. Position data are collected via motion capture (VICON).

3) *Results and Discussion*: Experimental results (Figs. 5 and 6) reveal that wrapping the proposed hierarchical structure around a low-level controller (red curves in figures) can help keep the quadrotor closer to the desired height when compared to the standalone original controller (blue curves), on average. This indicates that incorporating model identification and controller adaptation modules through DHC helps the overall control structure adjust better and faster to uncertain perturbations as, in this case, ground effects.

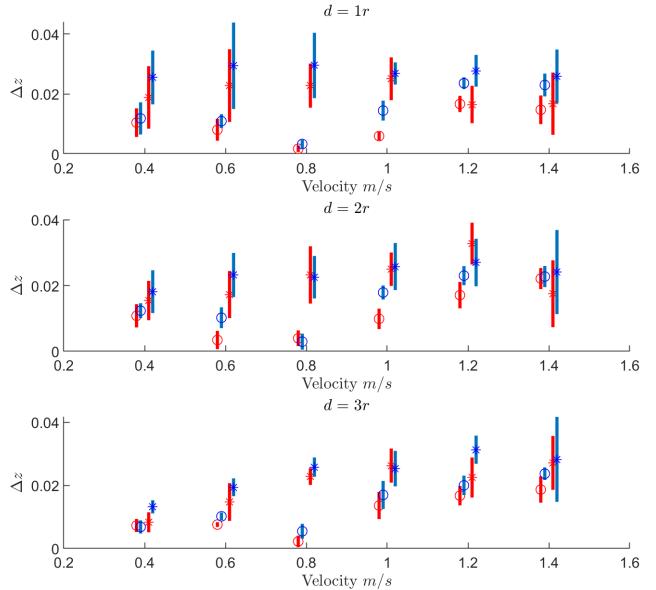


Fig. 5. 1σ error plots for the standalone geometric controller (blue circle) and PID controller (blue star), and with wrapped DHC (red circle and star).

Furthermore, model/controller adaptation appears to perform better at low distances from the ground and as the forward velocity increases. This observation is in line with aerodynamics suggesting that 1) ground effect starts diminishing when the flying height above the obstacle increases to $d > 3r$, and 2) as forward speed increases both ground effect and drag affect rotorcraft flight in still not well-

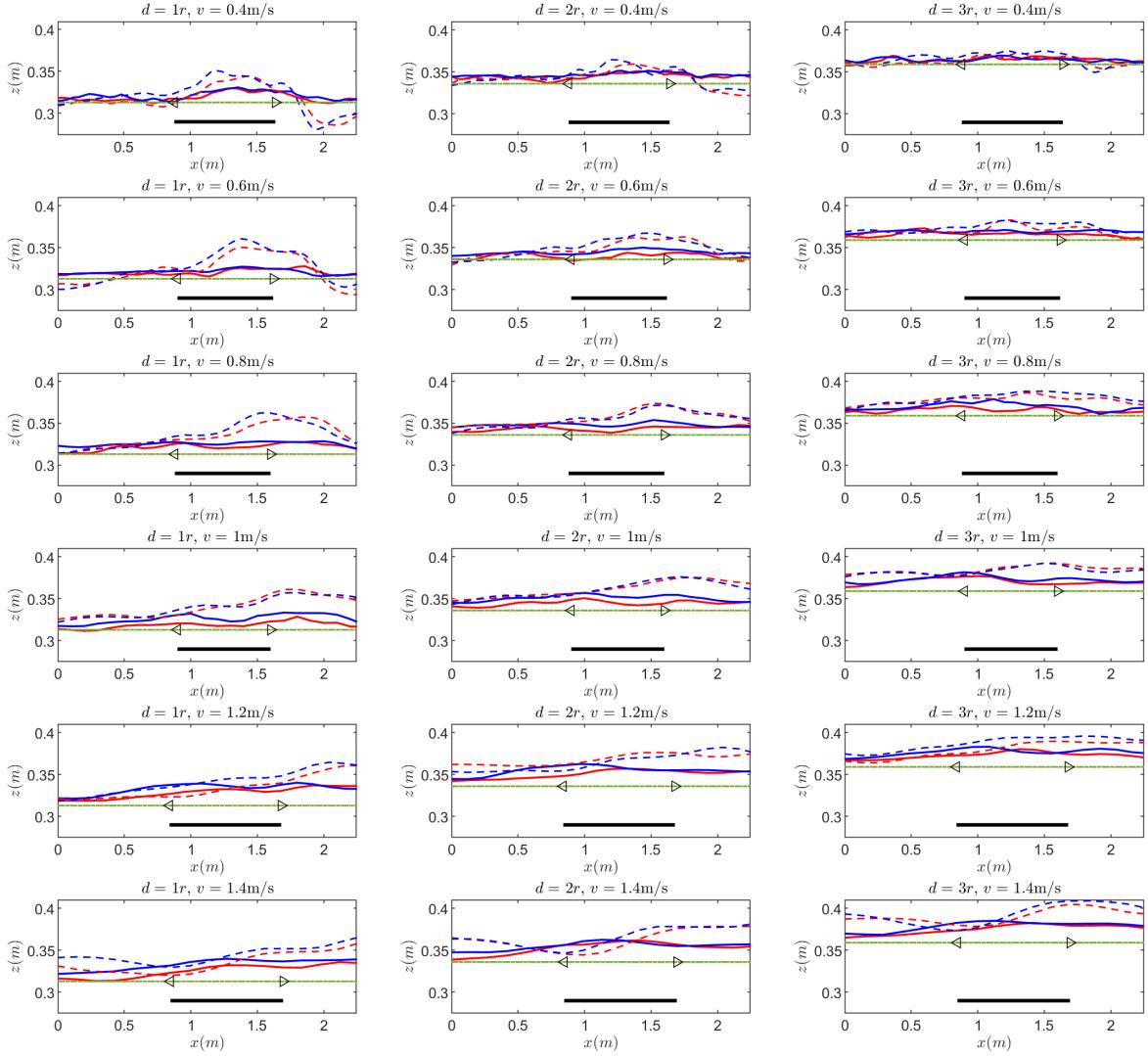


Fig. 6. Trajectories of the Crazyflie tasked to fly at desired heights, d , over the obstacle (green dashed-dotted) and various forward speeds, v , under ground effect with standalone geometric controller (blue solid) and PID controller (blue dashed), and with wrapped DHC (red dashed and solid). Obstacle locations (black) are shown for clarity.

understood ways [44]. Hence, through this experiment we show how DHC can be used to render a controller adaptive to environment changes that excite unmodeled dynamics.

The hierarchical structure may still remain bound to the low-level controller's possible limitations, as shown in simulation. In the experiments, this is observed as forward speed increases. After clearing out the obstacle, the controller's apparent steady-state error seems to increase (c.f. bottom two rows of Fig. 6) at higher speeds. While this may be an artefact of the limited experimental volume, the settling time of the controller with and without DHC still increases. Future work will investigate the interplay between the lower-level controller and DHC, and consider different hierarchical adaptive control structures to change an underlying controller's behavior more drastically when needed.

VI. CONCLUSIONS

The paper introduced a Data-driven Hierarchical Control (DHC) structure with two key functions. 1) DHC utilizes

an online data-driven approach to learn a model of input reference and true outputs. 2) DHC utilizes the derived model to keep refining a reference signal given to an underlying low-level controller to ensure system performance in the presence of system and/or system-environment uncertainty. We showed that DHC retains the stability properties of the underlying lower-level controller, and further determined lower bounds for our method's sensitivity to noisy data.

The utility and performance of the proposed approach are tested in simulation and experimentally with a quadrotor. The test cases were designed to excite specific types of uncertainty that are common in practice: 1) constant model parameter errors in simulation; and 2) variable uncertain robot-environment interaction experimentally through operation under ground effect. Results suggest that DHC can successfully be wrapped around existing controllers in practice, to improve their performance by discovering and harnessing unmodeled dynamics during deployment.

REFERENCES

[1] S. Zarovy, M. Costello, A. Mehta, G. Gremillion, D. Miller, B. Ranganathan, J. S. Humbert, and P. Samuel, "Experimental study of gust effects on micro air vehicles," *AIAA Atmospheric Flight Mechanics Conference*, p. 7818, 2010.

[2] G. S. Aoude, B. D. Luders, J. M. Joseph, N. Roy, and J. P. How, "Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns," *Autonomous Robots*, pp. 51–76, 2013.

[3] C. Powers, D. Mellinger, A. Kushleyev, B. Kothmann, and V. S. A. Kumar, "Influence of aerodynamics and proximity effects in quadrotor flight," *Int. Symp. on Experimental Robotics (ISER)*, 2012.

[4] K. Karydis, I. Poulakakis, J. Sun, and H. G. Tanner, "Probabilistically valid stochastic extensions of deterministic models for systems with uncertainty," *The International Journal of Robotics Research*, vol. 34, no. 10, pp. 1278–1295, 2015.

[5] H. Yoon, D. B. Hart, and S. A. McKenna, "Parameter estimation and predictive uncertainty in stochastic inverse modeling of groundwater flow: Comparing null-space monte carlo and multiple starting point methods," *Water Resources Research*, pp. 536–553, 2013.

[6] M. L. Rodriguez-Arvalo, J. Neira, and J. A. Castellanos, "On the importance of uncertainty representation in active slam," *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 829–834, 2018.

[7] I. D. Landau, R. Lozano, and M. M'Saad, *Adaptive Control*, J. W. Modestino, A. Fettweis, J. L. Massey, M. Thoma, E. D. Sontag, and B. W. Dickinson, Eds. Berlin, Heidelberg: Springer-Verlag, 1998.

[8] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[9] D. Piga, M. Forgiore, S. Formentin, and A. Bemporad, "Performance-oriented model learning for data-driven mpc design," *IEEE Control Systems Letters*, vol. 3, no. 3, pp. 577–582, 2019.

[10] K. Pereida and A. P. Schoellig, "Adaptive model predictive control for high-accuracy trajectory tracking in changing conditions," *CoRR*, 2018.

[11] G. Shi, X. Shi, M. O'Connell, R. Yu, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S. Chung, "Neural lander: Stable drone landing control using learned dynamics," *CoRR*, 2018.

[12] C. D. McKinnon and A. P. Schoellig, "Experience-based model selection to enable long-term, safe control for repetitive tasks under changing conditions," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 2977–2984, 2018.

[13] T. Wang, H. Gao, and J. Qiu, "A combined adaptive neural network and nonlinear model predictive control for multirate networked industrial process control," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 2, pp. 416–425, 2015.

[14] M. T. Frye and R. S. Provence, "Direct inverse control using an artificial neural network for the autonomous hover of a helicopter," *IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*, pp. 4121–4122, 2014.

[15] H. Suprijono and B. Kusumoputro, "Direct inverse control based on neural network for unmanned small helicopter attitude and altitude control," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, pp. 99–102, 2017.

[16] S. A. Nivison and P. P. Khargonekar, "Development of a robust deep recurrent neural network controller for flight applications," in *American Control Conf. (ACC)*. IEEE, 2017, pp. 5336–5342.

[17] B. J. Emran and H. Najjaran, "Adaptive neural network control of quadrotor system under the presence of actuator constraints," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2017, pp. 2619–2624.

[18] S. Bansal, A. K. Akametalu, F. J. Jiang, F. Laine, and C. J. Tomlin, "Learning quadrotor dynamics using neural network for flight control," in *IEEE Conf. on Decision and Control (CDC)*, 2016, pp. 4653–4660.

[19] Q. Li, J. Qian, Z. Zhu, X. Bao, M. K. Helwa, and A. P. Schoellig, "Deep neural networks for improved, impromptu trajectory tracking of quadrotors," *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 5183–5189, 2017.

[20] S. Zhou, M. K. Helwa, and A. P. Schoellig, "Design of deep neural networks as add-on blocks for improving impromptu trajectory tracking," *IEEE Conf. on Decision and Control (CDC)*, pp. 5201–5207, 2017.

[21] M. K. Helwa, A. Heins, and A. P. Schoellig, "Provably robust learning-based approach for high-accuracy tracking control of lagrangian systems," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1587–1594, 2019.

[22] T. Westenbroek, D. Fridovich-Keil, E. Mazumdar, S. Arora, V. Prabhu, S. S. Sastry, and C. J. Tomlin, "Feedback linearization for unknown systems via reinforcement learning," *arXiv preprint:1910.13272*, 2019.

[23] M.-B. Radac, R.-E. Precup, E.-L. Hedrea, and I.-C. Mituleti, "Data-driven model-free model-reference nonlinear virtual state feedback control from input-output data," in *26th Mediterranean Conf. on Control and Automation (MED)*. IEEE, 2018, pp. 1–7.

[24] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, "On dynamic mode decomposition: theory and applications," *Journal of Computational Dynamics*, 2014.

[25] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data-driven approximation of the koopman operator: Extending dynamic mode decomposition," *Journal of Nonlinear Science*, vol. 25, no. 6, pp. 1307–1346, 2015.

[26] A. Chatterjee, "An introduction to the proper orthogonal decomposition," *Current science*, pp. 808–817, 2000.

[27] K. Karydis and M. A. Hsieh, "Uncertainty quantification for small robots using principal orthogonal decomposition," in *Int. Symp. on Experimental Robotics*. Springer, 2016, pp. 33–42.

[28] I. Kevrekidis, C. Rowley, and M. Williams, "A kernel-based method for data-driven koopman spectral analysis," *Journal of Computational Dynamics*, vol. 2, no. 2, pp. 247–265, 2015.

[29] P. Gelß, S. Klus, J. Eisert, and C. Schütte, "Multidimensional approximation of nonlinear dynamical systems," *Journal of Computational and Nonlinear Dynamics*, vol. 14, no. 6, p. 061006, 2019.

[30] I. Abraham and T. D. Murphy, "Active learning of dynamics for data-driven control using koopman operators," *arXiv preprint:1906.05194*, 2019.

[31] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz, "Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control," *PloS one*, vol. 11, no. 2, p. e0150171, 2016.

[32] M. Haseli and J. Cortés, "Approximating the koopman operator using noisy data: noise-resilient extended dynamic mode decomposition," in *American Control Conf. (ACC)*. IEEE, 2019, pp. 5499–5504.

[33] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Dynamic mode decomposition with control," *SIAM Journal on Applied Dynamical Systems*, vol. 15, no. 1, pp. 142–161, 2016.

[34] V. Kumar and N. Michael, "Opportunities and challenges with autonomous micro aerial vehicles," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1279–1291, 2012.

[35] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The grasp multiple micro-uav testbed," *IEEE Robotics & Automation Magazine*, vol. 17, no. 3, pp. 56–65, 2010.

[36] I. Mezić, "Analysis of fluid flows via spectral properties of the koopman operator," *Annual Review of Fluid Mechanics*, vol. 45, pp. 357–378, 2013.

[37] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of fluid mechanics*, vol. 656, pp. 5–28, 2010.

[38] G. Strang, *Introduction to Linear Algebra*. Wellesley, MA: Wellesley-Cambridge Press, 2009.

[39] C. L. Lawson and R. J. Hanson, *Solving least squares problems*, ser. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM), 1995.

[40] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. The Johns Hopkins University Press, 1996.

[41] F. Sabatino, "Quadrotor control: modeling, nonlinear control design, and simulation," 2015.

[42] R. Ibarra, S. Florida, W. Rodríguez, G. Romero, D. Lara, and I. Pérez, "Attitude control of a quadrocopter using adaptive control technique," in *Applied Mechanics and Materials*, vol. 598. Trans Tech Publ, 2014, pp. 551–556.

[43] W. Johnson, *Rotorcraft Aeromechanics*, ser. Cambridge Aerospace Series. Cambridge University Press, 2013.

[44] X. Kan, J. Thomas, H. Teng, H. G. Tanner, V. Kumar, and K. Karydis, "Analysis of ground effect for small-scale uavs in forward flight," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3860–3867, 2019.

[45] S. Aich, C. Ahuja, T. Gupta, and P. Arulmozhivarman, "Analysis of ground effect on multi-rotors," *Int. Conf. on Electronics, Communication and Computational Engineering (ICECCE)*, pp. 236–241, 2014.

[46] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor uav on $se(3)$," *IEEE Conf. on Decision and Control (CDC)*, pp. 5420–5425, 2010.