ARTICLE IN PRESS

Annual Reviews in Control xxx (xxxx) xxx

Contents lists available at ScienceDirect

Annual Reviews in Control

journal homepage: www.elsevier.com/locate/arcontrol

Vision article

Learning quadrotor dynamics for precise, safe, and agile flight control

Alessandro Saviolo *, Giuseppe Loianno

New York University, Tandon School of Engineering, Brooklyn, 11201, NY, USA

ARTICLE INFO

Keywords:
Aerial vehicles
System identification
Modeling dynamics
Robot learning
Model learning
Neural networks

ABSTRACT

This article reviews the state-of-the-art modeling and control techniques for aerial robots such as quadrotor systems and presents several future research directions in this area. The review starts by introducing the benefits and drawbacks of classic physic-based dynamic modeling and control techniques. Subsequently, the manuscript presents the key challenges to augment or replace classic techniques with data-driven approaches that can offer several key benefits in terms of flight precision, safety, adaptation, and agility.

1. Introduction

In the past decade, aerial robots have become important platforms to help humans solve a wide range of time-sensitive problems including logistics, search and rescue for a post-disaster response, and more recently reconnaissance and monitoring during the COVID-19 pandemic (Yang et al., 2020). Among different types of aerial robots, quadrotors have gained interest for applications in uncertain and cluttered indoor environments due to their simplicity in design, low cost, small size, lightweight, and great maneuverability combined with the ability to hover in place (Emran & Najjaran, 2018). These time-sensitive tasks often require quadrotors to make fast decisions and agile maneuvers. Hence, to safely control these systems, it is critical to accurately model and estimate their dynamics and to capture the highly nonlinear effects generated by aerodynamic forces and torques, propeller interactions, vibrations, model approximations, and other phenomena. However, such effects cannot be easily measured or modeled and thus often remain hidden (Saviolo, Li, & Loianno, 2022). Moreover, in some aerial robot applications, the platform may be endowed with external appendages (e.g., payload, manipulators, cables) that would significantly change the dynamics by varying the system configurations (e.g., mass and moment of inertia). Overall, failing to model such system configuration changes would result in significant degradation of the flight performances and may cause catastrophic failures.

Classic modeling of the quadrotor's dynamics is performed using physics-based principles approaches which result in nonlinear Ordinary Differential Equations (ODE) (Loianno, Brunner, McGrath, & Kumar, 2017). However, these nominal models only approximate the actual system dynamics and do not take into account external effects caused by aggressive maneuvers or modifications of the system configuration. To circumvent this issue, recent works have investigated the use of data-driven approaches for modeling system dynamics. These methods have demonstrated inspiring results, enabling quadrotors to fly at extraordinary speeds up to 65 km/h and accelerations up to 46.8 $\mathrm{m/s^2}$ (Bauersfeld, Kaufmann, Foehn, Sun, & Scaramuzza, 2021), carry unknown payloads (Belkhale et al., 2021; Saviolo, Frey, Rathod, Diehl, & Loianno, 2022), operate under challenging wind conditions with wind speeds up to 43.6 km/h (O'Connell et al., 2022), and robustly adapt to unexpected actuation failures (Song, He, Zhang, Qian, & Fu, 2019).

The scope of this article is to review the classic physic-based dynamic modeling and control techniques for quadrotor systems, show the exciting successes of learning-based methods to complement or replace classic techniques, and present future research directions in this area. These include data denoising, transformer architectures, physics-inspired and meta-learned priors, active learning, and novel visual-based state representations that directly relate perception and action.

In summary, the highlights of this article are

- While model-free controllers directly optimize the input–output behavior of the system, model-based controllers exploit prior dynamics knowledge to actuate the robot, resulting in better adaptability, scalability, and sample efficiency. However, the performance of model-based controllers heavily relies on having access to an accurate dynamics model which is often unrealistic due to uncontrollable and often unobservable environment variations (Section 2);
- Classic modeling of the robot dynamics is performed using physics-based principles. While these approaches can precisely identify rigid-body systems, they fail to represent complex nonlinear disturbance phenomena, such as friction, deformation,

* Corresponding author.
E-mail address: alessandro.saviolo@nyu.edu (A. Saviolo).

https://doi.org/10.1016/j.arcontrol.2023.03.009
Received 9 January 2023; Received in revised form 17 March 2023; Accepted 17 March 2023

aerodynamic effects, and vibrations, that cannot be directly measured and therefore do not have explicit analytic equations (Section 3);

- Learning and adapting the dynamics purely from data enables accurate modeling and high-performance control even in challenging flight operating conditions (Section 4);
- Data preparation and denoising are key to ensuring that the learning model extracts the desired dynamical system behavior (Section 5.1);
- Learning quadrotor control and system identification using transformer-based algorithms is still in the twilight zone but represents an exciting opportunity for future research in light of the recent breakthroughs in the area of Natural Language Processing and Computer Vision (Section 5.2);
- Purely learning dynamics from data would require infinite samples and computing power, therefore either inductive priors must be instilled during the design process of the learning algorithm (Section 5.3) or active learning strategies must be employed to extract more information from the available data (Section 5.4);
- Coupling perception and action by expressing the system dynamics and perception variables in the same topological space reduces the inference latency while directly parsing the perception information from the sensor to the robot state space. This contributes to improved accuracy and efficiently maximizes the robot's future knowledge (Section 5.6).

2. Robot control

Control theory is a field of mathematics that deals with the control of dynamic systems. The objective of control theory algorithms is to govern the system inputs to make it reach a desired state. The earliest algorithms date back to the late 19th century with the introduction of the first theoretical basis for the operation of governors (Maxwell, 1868) and several control stability criteria (Hahn et al., 1967; Routh, 1877). Over the years, control theory has introduced a myriad of different control algorithms, ranging from robust (Zhou & Doyle, 1998) and adaptive (Cao, Ma, & Xu, 2012) to optimal (Garcia, Prett, & Morari, 1989) and stochastic (Hashemian & Armaou, 2017) controllers. While these control algorithms have been successfully implemented across multiple industries, such as manufacturing, traffic systems, and robotics (Glad & Ljung, 2018), modern control theory still holds many challenges from both theoretical and practical perspectives (Hou & Wang, 2013). A central dilemma is whether to use prior knowledge of the system or directly control it based on its input–output behavior. Based on this design decision, control strategies are categorized into model-based and model-free.

2.1. Model-free control

Model-free controllers optimize the input–output behavior of the system without requiring any explicit dynamical model. The absence of any prior knowledge of the system and the implementation simplicity have made these controllers successfully applied in the most diverse fields, ranging from intelligent transportation systems to energy management (Fliess & Join, 2013). The most notable example of a model-free controller is the Proportional–Integral–Derivative (PID) (Minorsky, 1922).

A PID controller continuously applies modulated, responsive corrections to a control function, where modulation is performed by multiplication of the control response with proportional, integral, and derivative terms to take into account both the current and future trajectory tracking error outlook. For example, consider the trajectory tracking problem where the system is required to follow a given desired trajectory of states $\mathbf{x}_{des,i}$ and $\tilde{\mathbf{x}}_i = \mathbf{x}_{des,i} - \mathbf{x}_i$ represents the error between the desired state and the actual state at time i. Then, the PID optimization problem is formulated as follows

$$K_p \tilde{\mathbf{x}}_i + K_i \int_0^i \tilde{\mathbf{x}}_\tau \mathrm{d}\tau + K_d \frac{\mathrm{d}\tilde{\mathbf{x}}_i}{\mathrm{d}i} \tag{1}$$

where K_p, K_i, and K_d denote the coefficients for the proportional, integral, and derivative terms, respectively.

2.2. Data-driven model-free control

PID controllers provide adequate performance for simple set-point stabilization problems, but their application to more complex tasks requires a specialized design and a careful selection of the tuning parameters. This limitation can be minimized by combining or replacing classical model-free control frameworks based on reinforcement learning techniques (Degris, Pilarski, & Sutton, 2012; Devo, Mao, Costante, & Loianno, 2022) which seek to learn the consequences of their actions purely through trial and error. Generally, model-free reinforcement learning is divided into policy-based, such as REINFORCE and PPO (Schulman, Wolski, Dhariwal, Radford, & Klimov, 2017), and value-based, such as SARSA and DQN (Mnih et al., 2013). Policy-based methods explicitly build a representation of the policy and store it in memory during learning. Contrarily, value-based methods store a value function and learn an implicit policy that can be directly derived from the value function (e.g., pick the action with maximum value). Model-free policy-based is often the reinforcement learning approach of choice in robotics since it is better at coping with the inherent challenges faced by real-world robots during operation (Atkeson & Santamaria, 1997). For example, a key advantage of policy-based strategies is that they allow stability and robustness guarantees by selecting suitable policy parametrizations (Bertsekas, 2005). Moreover, imitation learning from expert demonstrations may be used to obtain initial estimates of the policy parameters (Peters & Schaal, 2006). Over the last decade, model-free policies have demonstrated inspiring results in a wide range of challenging robotics applications, including control of multi-limb soft robots (Vikas, Grover, & Trimmer, 2015), flapping-wing micro-robots (Pérez-Arancibia, Duhamel, Ma, & Wood, 2015), visual servoing of elastic objects using robotic manipulators (Navarro-Alarcon, Liu, Romero, & Li, 2013), and robust active visual perching with quadrotors on inclined surfaces (Mao, Nogar, Kroninger, & Loianno, 2023). More recently, Kumar, Fu, Pathak, and Malik (2021) trained a robust model-free policy to enable legged-robot locomotion across many real-world terrains, such as sand, mud, hiking trails, tall grass and dirt pile. The policy is trained completely in simulation using roughly 4.5 months worth of cumulative experience. For quadrotor control, Hwangbo, Sa, Siegwart, and Hutter (2017) proposed a model-free deterministic on-policy method to dynamically stabilize the vehicle after an upside-down throw. The method is demonstrated over multiple throws in simulation. The key limitation of model-free reinforcement learning is that it is difficult to apply in the real world to unstable systems such as quadrotors due to the high sample complexity and the possibility of catastrophic failure during training. The high sample complexity requirement is due to the double objective that model-free approaches are required to solve: (i) modeling a succinct but informative state representation and (ii) predicting accurate control actions with such representations (Pari, Shafiullah, Arunachalam, & Pinto, 2021). Contrarily, model-based methods explicitly solve the two objectives individually, hence resulting in higher sample efficiency.

2.3. Model-based control

Model-based control leverages a prior dynamical model of the system to accurately control it. By introducing a strong prior over the system dynamics, model-based controllers achieve higher adaptivity, scalability, and sample efficiency than model-free approaches (Harrison, Sharma, Calandra, & Pavone, 2018). These properties make model-based controllers often preferred in challenging real-world applications where accurate modeling is available, such as aerospace (Marzat, Piet-Lahanier, Damongeot, & Walter, 2012).

Model Predictive Control (MPC) is the most notable example of a model-based controller. By executing real-time repeated optimal control, MPC is capable to solve multi-variable, constrained, and (possibly) nonlinear control problems. For example, consider the trajectory tracking problem where the system is required to follow a given desired trajectory of states $\mathbf{x}_{des,i}$ and $\tilde{\mathbf{x}}_i = \mathbf{x}_{des,i} - \mathbf{x}_i$ and $\tilde{\mathbf{u}}_i = \mathbf{u}_{des,i} - \mathbf{u}_i$ represent the error between the desired state and input and the actual state and input at time i, respectively. Then, the MPC optimization problem with N multiple shooting steps is formulated as follows:

$$\begin{aligned} \min_{\substack{\mathbf{x}_0,\dots,\mathbf{x}_N \\ \mathbf{u}_0,\dots,\mathbf{u}_{N-1}}} \quad & \frac{1}{2}\sum_{i=0}^{N} \tilde{\mathbf{x}}_i^\top \mathbf{Q}_x \tilde{\mathbf{x}}_i + \frac{1}{2}\sum_{i=0}^{N-1} \tilde{\mathbf{u}}_i^\top \mathbf{Q}_u \tilde{\mathbf{u}}_i \\ \text{s.t.} \quad & \mathbf{x}_{i+1} = h(\mathbf{x}_i, \mathbf{u}_i; \theta), \quad \text{for} \quad i = 0,\dots,N-1 \\ & \mathbf{x}_0 = \hat{\mathbf{x}}_0 \\ & g(\mathbf{x}_i, \mathbf{u}_i) \le 0 \end{aligned} \tag{2}$$

where $\mathbf{Q}_x$, $\mathbf{Q}_u$ are positive semi-definite diagonal weight matrices and the initial state $\mathbf{x}_0$ is constrained to the current state estimate $\hat{\mathbf{x}}_0$. The problem is further constrained by state and input constraints $g\left(\mathbf{x}_i, \mathbf{u}_i\right) \le 0$ (e.g., actuator constraints). For quadrotors, the desired control inputs can be obtained from the flat outputs of a differential-flatness planner (Sun, Romero, Foehn, Kaufmann, & Scaramuzza, 2022).

Despite the successes, model-based control still experiences significant challenges, such as (i) the requirement to have access to an accurate model of the system dynamics, (ii) the computational burden imposed by the optimization problem solved online with the limited computational power of small-scale platforms (e.g., the optimization problem in Eq. (2)), (iii) the need to use heuristic design choices to craft the controller which inevitably leads to sub-optimal control (e.g., design of the optimization problem and tuning its weight matrices) , and (iv) the requirement for explicit full state information (e.g., position, orientation, velocity) from an estimator that limits the capability to learn end-to-end state representations (Giurato & Lovera, 2016; Invernizzi, Lovera, & Zaccarian, 2022). Solving these challenges is still an open challenge for Robotics researchers. While this article focuses on (i) and discusses learning-based techniques to accurately model the system dynamics, the remainder of this section mentions a few related approaches for tackling (ii–iv).

2.4. Data-driven model-based control

Model-based reinforcement learning can be used to learn controllers from robot experience by trial and error, similar to model-free policies. The learned policy is sample efficient, computationally inexpensive, and requires minimal heuristic design choices, hence solving the aforementioned challenges. However, it still requires the knowledge of an accurate system dynamic model. Generally, model-based reinforcement learning algorithms are classified according to the policy update strategy. Notable examples are PILCO for greedy policy updates (Deisenroth & Rasmussen, 2011) and guided policy search for bounded policy updates (Levine & Koltun, 2013). While model-based policies have achieved impressive results in a wide range of robotics tasks, ranging from motion control of surgical manipulators to underwater robotic tasks (Deisenroth, Rasmussen, & Fox, 2011; El-Fakdi & Carreras, 2013; Englert, Paraschos, Peters, & Deisenroth, 2013), these controllers lack the interpretability of classical model-based approaches. Therefore, several lines of work have focused on combining model-based controllers with learning-based techniques. For example, Zhang, Kahn, Levine, and Abbeel (2016) combined MPC with guided policy search by using the MPC to generate the data at training time and a policy parametrized by a deep neural network. The framework is demonstrated in simulation on a quadrotor for obstacle avoidance. More recently, Song and Scaramuzza (2022) augmented an MPC for quadrotor control with several learned high-level policies that are devoted to automatically choosing hard-to-optimize decision variables.

2.5. Iterative learning control

Another popular approach is iterative learning control (Bristow, Tharayil, & Alleyne, 2006) which has been successfully combined with MPC in a Batch Model Predictive Control (BMPC) approach to control chemical processes (Lee & Lee, 2007) and refine their performances over multiple task iterations. For example, Rosolia and Borrelli (2017, 2018) proposed a Learning Model Predictive Control (LMPC) approach and applied it to ground vehicles. In this approach, the vehicle collects the states and their corresponding costs, across multiple successful iterations of the same task. The vehicle learns from the collected data to explore new ways to decrease cost in the same task as long as it maintains the ability to reach a state that has already been demonstrated to be safe during previous iterations and therefore belongs to the sampled safety set

$$SS^j = \left\{ \bigcup_{i \in M^j} \bigcup_{t=0}^{T^i} \mathbf{x}_t^i \right\} \tag{3}$$

where M^j is a set of indexes that represents the iterations that completed the task and T^i is the time stamp when the task is completed at the ith iteration. The approach does not require a reference trajectory as in previously mentioned works; thus, it is especially versatile and useful during tasks where the desired trajectory is not known or difficult to compute due to the system complexity or parameter uncertainty.

Common multi-rotor platforms including quadrotors evolve on the nonlinear manifold configuration space $SO(3) \times \mathbb{R}^3$ making the LMPC problem substantially different and more complex for these types of systems compared to ground vehicles. The challenges of building a safety set that includes members of the rotation group $SO(3)$ is addressed in Li, Tunchez, and Loianno (2022), where an appropriate numerical integration approach for the group elements is employed to ensure that the forward integration results adhere to the $SO(3)$ structure once employed in the discrete MPC formulation. This formulation for each task iteration is similar to Eq. (2), except that the reference trajectory is not specified and the final state $\mathbf{x}_{t+N} \in SS^{j-1}$ with SS^{j-1} denoting the safety set at the previous iteration step.

Iterative learning control has also been shown to be effective in enabling cooperative learning among multi-agent systems by integrating it into distributed control (Hock & Schoellig, 2016; Huang, Chen, Meng, & Sun, 2019; Meng & Zhang, 2021). This involves each agent's controller being derived iteratively and cooperatively. For instance, in the case of quadrotors, Hock and Schoellig (2019) proposed a decentralized control architecture using distributed iterative learning control for formation control applications. Simulation results showed that the proposed method effectively addressed communication and computational constraints, leading to precise formation flying.

2.6. Automatic tuning of controller gains

Carefully tuning the controller gains is critical to making the control loop stable and responsive and minimizing overshooting (Wang, Yuan, & Zhu, 2016). However, this translates in practice into a tedious trial-and-error process performed by a human expert, difficult even for the simplest tasks. Therefore, several works have investigated automatic approaches for tuning the gains of model-free and model-based controllers. The traditional line of work expresses the desired performance metric, such as the tracking error, as a quadratic function of the controller parameters, and then optimizes the controller with gradient-based optimization. A notable example is the

MIT rule (Åström, 1983; Grimble, 1984). Other approaches propose to iteratively estimate the optimization function and use the estimate to find optimal parameters (Berkenkamp, Schoellig, & Krause, 2016; Trimpe, Millane, Doessegger, & D'Andrea, 2014) or directly search for the optimal controller parameters by sampling (Davidor, 1991; Loquercio, Saviolo, & Scaramuzza, 2022).

The major disadvantage of these approaches is that either they require iterative experiments (Loquercio et al., 2022; Ostafew, Schoellig, & Barfoot, 2016; Romero, Govil, Yilmaz, Song, & Scaramuzza, 2022), which is often unfeasible for robotics applications, or limit the tuning to limited time horizons (Cheng et al., 2022), hence resulting in sub-optimal performance over the entire task.

2.7. Adaptive control

The history of adaptive control systems is almost as long as the entire field of control systems. Originating during the 1950s, Model Reference Adaptive Control (MRAC) (Joshi, Virdi, & Chowdhary, 2021) introduced the core concept of adjusting the system output to align with a desired reference dynamical model that distinguishes adaptive control approaches. Over the decades, MRAC has been further refined into Composite MRAC (Gregory, Gadient, & Lavretsky, 2011) and L1 adaptive control (Gahlawat, Zhao, Patterson, Hovakimyan, & Theodorou, 2020). For example, Hanover, Foehn, Sun, Kaufmann, and Scaramuzza (2021) proposed an approach to quadrotor control by augmenting an MPC with an L1 adaptive law. Their study demonstrated that L1 adaptive laws can significantly enhance the quadrotor's ability to accurately follow desired trajectories under wind disturbances and when carrying payloads of varying weights. The reader is referred to Annaswamy and Fradkov (2021) for an overview and historical perspective of adaptive control.

Generally, adaptive control methods trade-off robustness with performance, which leads to sub-optimal control (Ortega & Panteley, 2014). Moreover, in case an MPC is employed (Pravitra, Ackerman, Cao, Hovakimyan, & Theodorou, 2020), the optimization problem is solved by respecting defective dynamics constraints and thus potentially degrading the controller's predictive performance and accuracy as well as erroneously evaluating the feasibility of additional constraints. Conversely, by using the data-driven techniques presented in this article, the controller can fully exploit its predictive power to generate actions, resulting in superior control compared to adaptive control (Saviolo, Frey, et al., 2022).

3. Physics-based modeling of quadrotors

The field of quadrotor dynamics modeling relies on two dominant methods: Euler–Lagrange and Newton–Euler (Kim, Kang, & Park, 2010; Pounds, Mahony, & Corke, 2010; Zhang et al., 2014). The former offers a more concise and generalized formulation, while the latter is more intuitive and aligned with physical principles. Although they present several differences, both approaches provide consistent description of the quadrotor's dynamics.

This section focuses on the problem statement proposed under the system identification rubric using the Newton–Euler formalism. The literature on modeling system dynamics can be broadly categorized into two types: continuous-time and discrete-time. Additionally, this section provides a mathematical model of the quadrotor system based on physics-based principles and a quadratic approximation of the dynamics.

3.1. System identification task

Consider a dynamical system with state $\mathbf{x}$ and control action $\mathbf{u}$, then the discrete-time system identification task is formulated as follows: find a function f, parametrized by $\boldsymbol{\theta}$, that maps at every time step t from state-control space to state space

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t; \boldsymbol{\theta}) \,. \tag{4}$$

While Eq. (4) is defined as a discrete-time task, it can also be formulated in continuous-time and then integrated using an ODE solver. Formally, the continuous-time system identification task is formulated as follows: find a function f, parametrized by $\boldsymbol{\theta}$, that maps at every time step t from state-control space to state-derivative space

$$\begin{aligned} \mathbf{x}_{t+1} &= h(\dot{\mathbf{x}}) \,, \\ \dot{\mathbf{x}} &= f(\mathbf{x}_t, \mathbf{u}_t; \boldsymbol{\theta}) \,. \end{aligned} \tag{5}$$

where h is a function that integrates its input forward in time. The function h can be defined in various ways, with different trade-offs between accuracy and computational complexity. One widely used method for discretization of the time derivative is the Euler method (Hahn, 1991), which involves computing the difference quotient at the current time step. However, the Euler method can become numerically unstable for certain parameter values or time step sizes. Another popular method is the Runge–Kutta method (Butcher, 1996), which approximates the state change over the time step by evaluating the time derivative at multiple intermediate points and using a weighted average. While the Runge–Kutta method is more accurate than the Euler method, it can be computationally expensive, particularly for higher-order approximations. In addition to these methods, other techniques like the backward differentiation formula method (Cash, 1980), the trapezoidal method (Yeh & Kwan, 1978), and the linear multistep method (Gear & Wells, 1984) have also been used. These methods generally offer higher accuracy than the Euler method, but can be more complex to implement and may require more computational resources.

In general, the choice of h depends on various factors, such as the desired level of accuracy, the available computational resources, and the stability constraints of the system. Therefore, it is essential to carefully consider these factors and choose an appropriate method for the specific application requirements. For quadrotor control, h is often chosen from the Runge–Kutta family of functions.

3.2. Quadrotor dynamics

The quadrotor (Fig. 1) is an aerial vehicle modeled in a cross configuration with four symmetrical arms. Each arm is equipped with a brushless motor and a propeller with fixed-pitch blades that creates the desired airflow to lift the quadrotor. The quadrotor is controlled by using force (thrust) and torque (along the three body axis) that are directly mapped into propeller velocities. Most quadrotors are therefore underactuated, with only four control actions for their six degrees of freedom. This imposes several challenges when controlling the quadrotor's system, as the dynamics model is not fully linearizable. Additionally, the quadrotor's dynamics change dramatically during operation. For example, during near-hover flight, the forces and torques generated by the single propellers are typically accurately modeled as quadratic in speed. Contrarily, extreme maneuvering induces a large variety of highly nonlinear effects on the system, such as rotor-to-rotor and rotor-to-body aerodynamic effects, platform vibrations, and actuator disturbances. Additional stochastic effects are generated when the quadrotor is flying near the ground, in presence of measurement noise, or goes through significant system changes (payload attached) and failures (motor death). Furthermore, changes in battery voltage during flight can also have a significant impact on the system dynamics (Bauersfeld & Scaramuzza, 2022). As the quadrotor's battery drains, the available power to the motors decreases, resulting in lower thrust and increased instability. This effect can be particularly pronounced in quadrotors that are designed to carry heavy payloads, as the battery drains faster under increased load. Taking into account all these conditions is very cumbersome if not impossible. Therefore, when designing a model of the quadrotor's system, it is inevitable that some assumptions and simplifications must be taken. Based on the significance of these assumptions and simplifications, different models are designed.

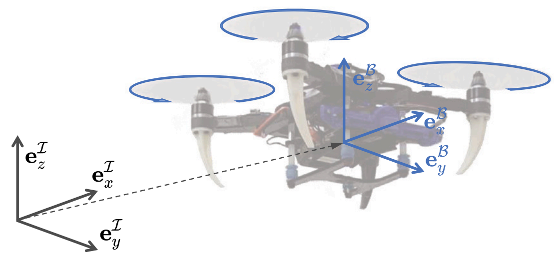

Fig. 1. System convention of the quadrotor model with $\mathcal{I}$ and $\mathcal{B}$ denoting inertial and body frames, respectively.

3.2.1. Quadratic dynamical model

The simplest physics-based dynamical model of the quadrotor is a quadratic fit which assumes that the forces and torques generated by the single propellers are proportional to the square of their speed. Consider an inertial frame $\mathcal{I} = \{\mathbf{e}_x^{\mathcal{I}}, \mathbf{e}_y^{\mathcal{I}}, \mathbf{e}_z^{\mathcal{I}}\}$ and a body frame $\mathcal{B} = \{\mathbf{e}_x^{\mathcal{B}}, \mathbf{e}_y^{\mathcal{B}}, \mathbf{e}_z^{\mathcal{B}}\}$ located at the center of mass of the quadrotor. The quadrotor's system can be described by the state

$$\mathbf{x} = \begin{bmatrix}\mathbf{p}^\top & \mathbf{v}^\top & \mathbf{q}^\top & \boldsymbol{\omega}^\top\end{bmatrix}^\top , \tag{6}$$

and the control action

$$\mathbf{u} = \begin{bmatrix}u_0 & u_1 & u_2 & u_3\end{bmatrix}^\top , \tag{7}$$

where $\mathbf{p} \in \mathbb{R}^3$ represents the position of the center of mass relative to $\mathcal{I}$, $\mathbf{v} \in \mathbb{R}^3$ represents the translational velocity relative to $\mathcal{I}$, $\mathbf{q} \in \mathbb{R}^4$ represents the rotation[2] from $\mathcal{B}$ to $\mathcal{I}$, $\boldsymbol{\omega} \in \mathbb{R}^3$ represents the rotational velocity relative to $\mathcal{B}$, and $u_i \in \mathbb{R}$ represents the ith motor command generated by the quadrotor's controller. Therefore, the quadrotor's continuous-time dynamical system evolves as follows:

$$\dot{\mathbf{x}} = \begin{bmatrix}\dot{\mathbf{p}} \\ \dot{\mathbf{v}} \\ \dot{\mathbf{q}} \\ \dot{\boldsymbol{\omega}}\end{bmatrix} = \begin{bmatrix}\mathbf{v} \\ \frac{1}{m}(\mathbf{q} \odot \boldsymbol{f}) + g\mathbf{e}_z \\ \frac{1}{2}(\mathbf{q} \odot \boldsymbol{\omega}) \\ \mathbf{J}^{-1}(\boldsymbol{\tau} - \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega})\end{bmatrix} = f(\mathbf{x}, \mathbf{u}) , \tag{8}$$

where $\odot$ is the quaternion-vector product as in $\mathbf{q} \odot \mathbf{v} = \mathbf{q}\mathbf{v}\bar{\mathbf{q}}$ and $\bar{\mathbf{q}}$ is the quaternion's conjugate, g is the gravity constant, m is the quadrotor's mass, $\mathbf{J}$ is the moment of inertia matrix which is generally assumed to be diagonal $\mathbf{J} = diag(J_{xx}, J_{yy}, J_{zz})$, f is the continuous-time system identification task in Eq. (5), and the collective thrust $\boldsymbol{f}$ and torque $\boldsymbol{\tau}$ of the quadrotor are defined as follows:

$$\boldsymbol{f} = k_f \sum_{i=0}^{3} u_i^2 , \quad \boldsymbol{\tau} = \begin{bmatrix} k_f l(u_0^2 + u_1^2 - u_2^2 - u_3^2) \\ k_f l(-u_0^2 + u_1^2 + u_2^2 - u_3^2) \\ k_\tau (u_0^2 - u_1^2 + u_2^2 - u_3^2)\end{bmatrix} , \tag{9}$$

where k_f and k_τ are the rotor thrust and torque constants, while l is the length of the quadrotor's arm.

The parameters $J_{xx}, J_{yy}, J_{zz}, m, k_f, k_\tau, l$ are closely related to the quadrotor's platform and strictly define the physics-based model's accuracy.

3.2.2. The challenges of physics-based models

Physics-based models, such as the quadratic model (8), are computationally efficient and describe well the quadrotor's system dynamics in low-speed and acceleration regimes and basic platforms, where external forces and torques are negligible. However, as speed or acceleration increase or additional payloads are applied to the platform, external complex effects increase as well, significantly degrading the flying performance (Alkayas, Chehadeh, Ayyad, & Zweiri, 2022). Therefore, physics-based models have been further refined to model these effects. Notable examples are models based on blade element momentum theory (Bouabdallah & Siegwart, 2007), kinematic constraints (Sanchez-Gonzalez et al., 2018), and Hamiltonian and Lagrangian mechanics (Das, Lewis, & Subbarao, 2009). However, even though these approaches better capture the system dynamics — including aerodynamic forces and torques acting on the single rotors, their accuracy still depends on the choice of the system parameters. Parameter identification approaches can be used to empirically identify their values (Svacha, Paulos, Loianno, & Kumar, 2020; Wüest, Kumar, & Loianno, 2019). However, uncertainties remain due to the nonlinearity of the external effects that make the parameters difficult to be accurately estimated.

4. Learning dynamics from robot experience

Learning robot dynamics from data requires solving the system identification tasks in Eqs. (4) and (5) by approximating f using a learning-based model. There exists a large variety of learning-based models that can be picked and each one of these models differs in the assumptions and biases it injects into the form of the underlying system dynamics. Learning-based models that simplify the dynamics function to a known form belong to the class of parametric algorithms (e.g., neural networks). Contrarily, algorithms that do not make strong assumptions about the form of the underlying dynamics function are called nonparametric algorithms (e.g., Gaussian processes). Both parametric and nonparametric models have been widely investigated for modeling quadrotor system dynamics. After formulating the previously defined system identification task as a supervised learning problem, this section focuses on the advantages and disadvantages of the two learning classes.

4.1. System identification as supervised learning

The system identification tasks in Eqs. (4) and (5) can be formulated as a supervised learning problem by modeling f as a learning-based model and finding the function's parameters $\boldsymbol{\theta}$ that minimize the prediction error of $f(\boldsymbol{\theta})$ over a set of demonstration data. For example, a typical choice is to minimize the Mean Squared Error (MSE) objective function by solving

$$\min_{\boldsymbol{\theta}} \mathcal{L}_{\text{MSE}} ,$$
$$\mathcal{L}_{\text{MSE}} = \frac{1}{|B_T|} \sum_{i=1}^{|B_T|} \|\mathbf{x}_i - f(\mathbf{x}_{i-1}, \mathbf{u}_{i-1}; \boldsymbol{\theta})\|^2 , \tag{10}$$

where B_T is a batch of demonstration data, $\mathbf{x}_i$ and $f(\mathbf{x}_{i-1}, \mathbf{u}_{i-1}; \boldsymbol{\theta})$ represent the measured and the predicted states, respectively.

Data for the system identification task consists of state-control trajectories with no additional labels. Hence, collecting data is rather inexpensive because of the absence of an external supervisory signal. For this reason, Eqs. (4) and (5) can also be formulated as a self-supervised learning problem (Saviolo, Frey, et al., 2022), hence highlighting the absence of human supervision.

As the dimensionality of the state-control pairs is relatively small, the input information to the learning-based models is typically extended with the history of state-control pairs. The history length is usually treated as a hyper-parameter that depends on the available computational power and size of the dynamical model. Section 5.1.2 addresses the use of history when learning dynamical models, its advantages, and disadvantages.

It is important to note that in this setting the data collection procedure is independent of the chosen controller used for operating the quadrotor. Data should simply include the information of what state is reached after applying a specific action from a given state. It does not matter how the control action is selected. Therefore, when collecting data, the practitioner can both manually or autonomously control the quadrotor and record the history sequence of state estimations and control actions. All these characteristics make data collection for the system identification task simple and accessible.

[2] We consider the quaternion representation $\mathbf{q} = \begin{bmatrix}q_w & q_x & q_y & q_z\end{bmatrix}$ to represent rotations on the group $SO(3)$ because it allows a singularity free mapping from the unit sphere S^3 to $SO(3)$.

4.2. Parametric algorithms

Parametric algorithms simplify the learning process by limiting the underlying dynamics function to a known form. This assumption allows these models to earn very quickly from data and generate interpretable results. However, it also limits their learning capabilities as it is unlikely that the assumed known form matches the underlying dynamics function.

The most well-known parameter learning-based models are neural networks. These models fit the collected data with a set of parameters of fixed size. Hence, any prior over the dynamical system directly translates into selecting the proper architecture and training optimization details. Feedforward Neural Networks (FNNs) were the first parametric models devised for the system identification task of a quadrotor (Bansal, Akametalu, Jiang, Laine, & Tomlin, 2016). These models are characterized by dense connections that move and transform information from the input nodes to the output nodes with no cycles. Thanks to their dense connections, FNNs are well suited for modeling highly complex phenomena (e.g., the dynamical system of the quadrotor). However, the simple FNN's architecture is not designed for learning time-correlated features, hence it can only deal with a single state-control pair as input.

When modeling time-series data, Recurrent Neural Networks (RNNs) (Rumelhart, Hinton, & Williams, 1986) provide better architectures than FNNs. RNNs are characterized by cycles between nodes that allow output from some nodes to affect subsequent input to the same nodes. Consequently, these recurrent connections allow the network to memorize an internal state and process variable-length sequences of inputs. However, RNNs have some important limitations, from vanishing and exploding gradient problems to the difficulty to process long sequences. Such limitations make these networks complex to properly train and poorly suited for online Robotics applications (Pascanu, Mikolov, & Bengio, 2013).

Recently, convolution-based approaches have emerged as a superior alternative to RNNs (Bai, Kolter, & Vladlen, 2018). In particular, Lea, Vidal, Reiter, and Hager (2016) introduced the Temporal Convolutional Network (TCN) to perform fine-grained action segmentation. Unlike RNNs, TCNs take advantage of asynchronous and parallel convolution operations, avoid gradient instability problems, and offer flexible receptive field size thus better control of the model's memory size while still inherently accounting for temporal data structures like RNNs. Thanks to their favorable characteristics, TCNs have then been successfully employed in multiple sequences and time-series modeling tasks (Borovykh, Bohte, & Oosterlee, 2017; Luo & Mesgarani, 2019). Related to quadrotor control, Kaufmann et al. (2020) trained multiple TCNs to learn directly from raw sensory data an end-to-end policy for performing acrobatic maneuvers. For quadrotor system identification, Bauersfeld et al. (2021) learned the rotor-to-rotor and rotor-to-body aerodynamic forces and torques using two separate TCNs and demonstrated the learned dynamical system in simulation experiments. Saviolo, Li, and Loianno (2022) extended prior work by incorporating the entire nonlinear dynamical system in a TCN and demonstrating the utility and applicability of these network's architectures for learning the full system dynamics in the real world.

4.3. Nonparametric algorithms

Designing the learning-based model based on assumptions on the underlying dynamics function can greatly simplify the learning process, but it also limits what can be learned. Nonparametric algorithms seek to relax any assumption over the dynamics function form and learn free from the training data. This key advantage often translates into high flexibility to fit a large number of functional forms and consequent higher-performance models for prediction. However, these advantages come at the cost of a significant increase in data to estimate the underlying functional form and minor output interpretability.

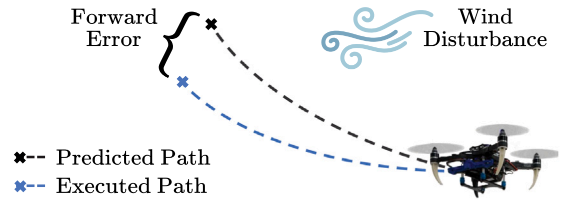

Fig. 2. Unmodeled disturbances, such as wind, deteriorate the predictive performance of the dynamical model and result in suboptimal control performance.

Gaussian Processes (GPs) are an attractive class of learning-based nonparametric models for modeling robot dynamical systems (Crocetti, Mao, Saviolo, Costante, & Loianno, 2023; Wang, Theodorou, & Egerstedt, 2018). The key advantages of GPs are their versatility — easily employing different kernels, even custom-made for the task, online adaptation to new data, and probabilistic prediction. However, a major drawback of GP regression is computational complexity. As for all nonparametric models, their complexity increases with the size of the training set. This implies the need to carefully choose a subset of the fitted data that best represents the true dynamics through data reduction techniques (Ambikasaran, Foreman-Mackey, Greengard, Hogg, & O'Neil, 2016; Das, Roy, & Sambasivan, 2018). However, since the dynamics are unknown, selecting these points might be challenging. Moreover, due to their poor scaling properties, GPs need to explicitly model individual dynamical effects, such as drag force (Torrente, Kaufmann, Fohn, & Scaramuzza, 2021). Generally, the computational complexity of these non-parametric models can be reduced by learning the dynamics component-wise (Sarkka, Solin, & Hartikainen, 2013). However, decoupling translational and rotational accelerations in single independent components results in the failure to capture the hidden dependencies that bound forces and torques for nonholonomic and underactuated systems like the quadrotors.

4.4. From static to adaptive dynamics

Learning robot dynamics purely from previously collected data ensures both learning without posing any physical risk to the robot and training over the entire data (i.e., capturing a global dynamical model of the system). However, this offline learning procedure assumes that the global model will remain accurate over time and therefore cannot deal with varying operating conditions that characterize most real-world environments. Online learning extends offline learning by continuously adapting the model during operation with newly collected data, therefore relaxing the need for a perfect system prior.

There exists extensive prior work on online learning applied to Robotics, from humanoid control (Gaskett & Cheng, 2003; Jamone, Natale, Nori, Metta, & Sandini, 2012) and obstacle avoidance (Losing, Hammer, & Wersing, 2015) to system identification and control of a manipulator and a quadruped (Bechtle, Hammoud, Rai, Meier, & Righetti, 2021; Fu, Levine, & Abbeel, 2016). Recently, several works have considered the system identification task for quadrotors. Belkhale et al. (2021) demonstrated quick adaptation against changing suspended payloads attached to the quadrotor's body. O'Connell et al. (2022) continuously refined the system dynamics to account for varying wind conditions. Saviolo, Frey, et al. (2022) effectively and efficiently adapted the dynamical model online to account for unseen wind disturbances, suspended payloads, and severe system configuration changes.

The key intuition behind the online learning strategies is to leverage the *forward* error between the predicted path by the dynamical model and the actual executed path by the quadrotor (Fig. 2). This error represents the mismatch between the dynamical model available to the controller and the true world dynamics and therefore its optimization directly translates into adapting the dynamics to the current operating regime.

Two optimization strategies can be employed to update the dynamical model, based on whether they update the model parameters entirely (Finn, Abbeel, & Levine, 2017a; Nagabandi et al., 2018) or only partially (McKinnon & Schoellig, 2021; Peng, Zhu, & Jiao, 2021). Generally, trading-off between these two strategies depends on the similarity of the online and offline data distributions as well as the available computational budget.

5. Future research directions

This section concludes the article by introducing the future directions for learning-based system identification for quadrotors and highlighting the key takeaway messages.

5.1. Towards a data-centric view

Data is the fuel that powers any learning-based model. Without careful collection and preprocessing of data, any model can learn very little and generate completely unexpected results. Despite this being a well-known fact, the importance of data has been rarely discussed in the research community, where most of the approaches have focused only on improving the predictive performance of the learning-based models. This section focuses on this important gap and details how data should be collected and preprocessed to maximize the learning outcome for the system identification task.

5.1.1. Data preparation

Designing the dataset for training the learning-based model requires first identifying the input information to the model and then labeling this data. These choices should reflect what features the model will focus on while making the predictions and what is the desired outcome. For example, for the quadrotor's system identification task, Torrente et al. (2021) employed the observed linear velocities for estimating the drag effects, while (Bauersfeld et al., 2021) fed the model with observed linear velocities, angular velocities, and control actions to estimate the rotor-to-rotor and rotor-to-body aerodynamic forces and torques. Although there are no rules on what information is best to extract from the observed measurements, a common practice is to assume that the dynamical evolution of the system is position-independent and omit this information. In fact, if the model would have access to this information, over-fitting to the environment where the data was collected would be inevitable.

Labeling data reflects the designated system identification task. When extracting data for modeling discrete-time dynamics, the label is the state of the system at the next time iteration. Therefore, data processing only requires synchronizing the measurements from the different sensors and obtaining the state and control estimates as noise-free as possible. On the contrary, the label for continuous-time dynamics is the state-derivative of the system — linear and angular accelerations, which cannot be directly observed. This adds one extra computation to the data collection which usually simply consists of computing the first-order derivative of velocities. As the data collected is inevitably corrupted by measurement noise, the computed accelerations are affected by noisy spikes that limit the learning capabilities of the neural model (Fig. 3). Various methods can be used to reduce the noise in the collected data, such as Butterworth (Butterworth et al., 1930; Saviolo, Li, & Loianno, 2022) and notch (Hirano, Nishimura, & Mitra, 1974; Xu, Gu, Qing, Lin, & Zhang, 2019) filters, total variation regularizers (Brunton, Proctor, & Kutz, 2016; Rudin, Osher, & Fatemi, 1992), as well as B-splines (Jayasree, Raj, Kumar, Siddavatam, & Ghrera, 2013; Knott, 1999). However, there is no guarantee that the filtered outputs are the true dynamics and that the discarded information is mere noise. Moreover, using filters introduces time delays that may render any online learning strategy not effective. Conversely,

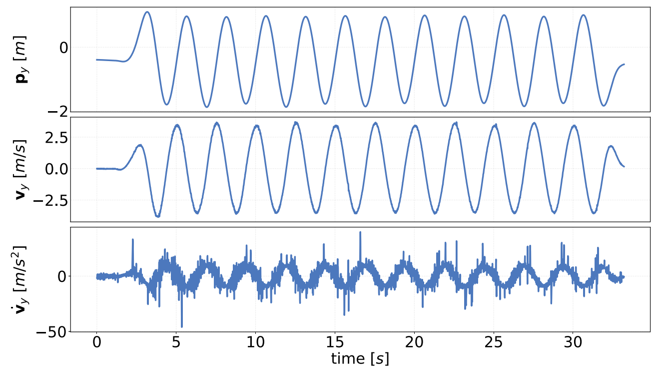

Fig. 3. State estimations collected while tracking a circular trajectory with an agile quadrotor. Position information is obtained from a Vicon[2] motion capture system. Velocities and linear accelerations are recovered by first and second-order differentiation of the measured positions, respectively. The sensor noise corrupts the quality of the observed measurements, resulting in degraded velocities and spiky accelerations.

learning discrete-time dynamics does not require any velocity differentiation and therefore the model is trained to regress less noisy labels, resulting in more accurate predictive models (Saviolo, Frey, et al., 2022).

5.1.2. Dealing with measurement noise

When collecting data from real-world sensors, the observed measurements are inevitably affected by noise. This corrupted data significantly affects the training process and increases the probability that the model will overfit the dataset and poorly generalize when deployed (Zhang, Bengio, Hardt, Recht, & Vinyals, 2021). Over the past few years, several regularization techniques have been proposed to minimize the effect of noisy labels during training, such as data augmentation (Shorten & Khoshgoftaar, 2019), weight decay (Krogh & Hertz, 1991), dropout (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014), and batch normalization (Ioffe & Szegedy, 2015). However, the generalization gap between models trained on clean and noisy data is significant even after applying all the aforementioned regularization techniques (Tanno, Saeedi, Sankaranarayanan, Alexander, & Silberman, 2019).

More recently, machine learning techniques have been investigated for overcoming noisy labels. The most popular techniques include redesigning the model architecture by adding noise adaptation layers (Xiao, Xia, Yang, Huang, & Wang, 2015), enforcing the model to overfit less noisy data (Jenni & Favaro, 2018), weighting the loss function (Wang, Liu, & Tao, 2017), or directly identifying noisy labels (Shen & Sanghavi, 2019). For an in-depth analysis of these techniques, Song, Kim, Park, Shin, and Lee (2022) and Han et al. (2020) provide comprehensive surveys on robust learning with noisy labels. However, these techniques are specifically designed for classification tasks, where labels are discrete. On the contrary, system identification is a regression task with continuous labels. Therefore, deploying these techniques in this setting is not straightforward and adds even more challenges.

One effective method for minimizing the impact of noisy observed measurements is to utilize data redundancy. This typically involves drawing on historical information, as the state-control pairs collected at high frequencies often contain substantial redundant information between consecutive time frames. Additionally, in multi-robot scenarios, collaborative information can also be utilized to further improve estimation accuracy and reduce the impact of noise on observations (Zhou, Xiao, Zhou, & Loianno, 2022). For quadrotor's system identification, Saviolo, Li, and Loianno (2022) demonstrated quantitatively that increasing the length of the history sequence in input to a

[2] https://www.vicon.com/

parametric model induces a tremendous improvement in the model's predictive accuracy. When the model is fed with only the current state-control pair, it is too sensitive to the measurement noise and struggles to capture the underlying dynamics of the quadrotor's system. In other words, the Markov assumption over the current state is weakened by the noise in the state estimations and control action readings. Therefore, as a general key takeaway, whenever it is possible, the history of state-control pairs should always be fed as input to the learning-based model.

5.2. Analogy to time-series forecasting

Innovation is rooted in taking ideas, approaches, and technologies from multiple disciplines and bringing them together. Therefore, understanding the relationship between different tasks that require solving the same problem but in different scientific fields is critical for the progress of research.

Time-series forecasting is a general problem of great practical interest in nearly all fields of science and engineering, such as Computer Vision, Natural Language Processing, and Robotics. A time series is defined as a sequence of values that are chronologically ordered, with some margin of error, and observed over time. Forecasting time series translates into discovering the future values of a series from its past values. Based on this definition, the system identification tasks Eqs. (4) and (5) belong to the time-series forecasting problem.

Advancements in time-series forecasting applied to one field have often been propagated to others. For example, RNNs were first introduced for speech recognition (Fernández, Graves, & Schmidhuber, 2007; Graves & Schmidhuber, 2008; Rumelhart et al., 1986; Wu, Kwasny, Kalman, & Engebretson, 1993) before being widely adopted in the related fields of Computer Vision (Saad, Caudell, & Wunsch, 1999; Vinyals, Toshev, Bengio, & Erhan, 2015) and Robotics (Funahashi & Nakamura, 1993). Similarly, Convolutional Neural Networks (CNNs) were first proposed for image and speech recognition (Lea et al., 2016; LeCun et al., 1995), and only then employed in Robotics for learning control policies (Kaufmann et al., 2020) and system identification (Bauersfeld et al., 2021; Saviolo, Li, & Loianno, 2022).

More recently, transformer neural networks (Vaswani et al., 2017) have been proposed for the machine translation task. A transformer is a learning-based parametric model that uses the mechanism of self-attention to understand time series, hence differentially weighting the significance of each part of the input data. These new architectures have represented a breakthrough in Natural Language Processing research and paved the way for the modern large language models, such as BERT (Devlin, Chang, Lee, & Toutanova, 2018) and GPT (Radford et al., 2018).

Inspired by the astounding results achieved by transformer models in Natural Language Processing, researchers have considered applying these architectures to Computer Vision. Despite being largely dominated by CNNs for almost a decade (Li, Liu, Yang, Peng, & Zhou, 2021), many vision tasks have completely shifted to transformer-based architectures. The most notable examples are ViT (Dosovitskiy et al., 2020), DeiT (Touvron et al., 2021) and SWIN-Transformers (Liu et al., 2021). By treating images as time series of patches, transformers demonstrate high learning capabilities in very large data sets, dominating CNNs due to their superior modeling capacity, global receptive field, and lower inductive bias (Dosovitskiy et al., 2020).

Robotics research has also been invaded by transformer-based architectures, even though at a much slower pace. The first transformers were used for trajectory forecasting (Giuliari, Hasan, Cristani, & Galasso, 2021), motion planning (Chaplot, Pathak, & Malik, 2021), control learning (Chen, Lu, et al., 2021), and multi-task representation learning (Bonatti et al., 2022). Research on transformer-based models for quadrotors and system identification is still in the twilight zone and represents an exciting opportunity in light of the recent breakthroughs in related tasks and fields.

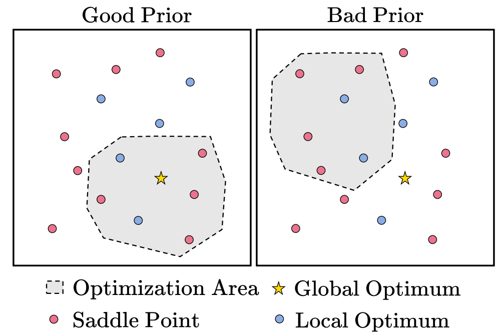

Fig. 4. Effects of different prior choices over the optimization landscape. While instilling priors is necessary for any task, choosing the proper priors is challenging. Some priors may be too restrictive and constrain the optimization search to sub-optimal solutions.

5.3. Instilling inductive priors

Nature versus nurture is a historical debate that takes roots millennia ago[2] and centers on the contributions of genetics and environmental factors to human development (Marcus, 2004). Naturalists as Plato Bluck (1961) advocate that human traits are inherited through generations, while nativists such as Locke (Locke, 1948) support the idea that the human mind begins as a blank slate and develops naturally through experience.

The contemporary view of this debate has abandoned the idea of categorically decoupling nature and nurture, leaving room for an inextricably intertwined belief where their influence overlap. Generally, it is recognized that both nature and nurture play a critical role in the human mind development (Levitt, 2013).

Over the centuries, the debate has spread beyond philosophy and psychology. In machine learning, nature corresponds to the inductive priors[3] injected into the model, while nurture represents how much a model can learn purely from data. The no-free-lunch theorems demonstrated that no learning algorithm is universally superior but different algorithms outperform each other on different datasets (Wolpert, 2002; Wolpert & Macready, 1997; Wolpert et al., 1995). As a result, purely nature approaches are not practical because finding the proper model for the task at hand without instilling any prior knowledge would require infinite data and computing power.

While instilling priors to reduce the possible algorithms is always necessary, introducing too many priors into the model would severely limit or bias what it can learn (Fig. 4). In fact, as the number of priors increases, the greater the chance that the model will not be able to learn the underlying functions (e.g., dynamical system evolution). Generally, the number of priors to inject into the model is tightly coupled to the available amount of informative data (Fig. 5). As the dataset size approaches infinite, the priors should be close to zero. On the opposite, when the data is scarce, priors are essential for effective learning.

Priors can be injected into the model in multiple ways, from the data collection and labeling to the architecture design and training procedure. The most common prior is injected when the problem is formulated as regression or classification and supervised or unsupervised. For example, a supervised classification problem assumes that the model will learn from finite discrete labels. Architectural priors

[2] During Chen Sheng and Wu Guang rebellion in 209 B.C., Sima (1993) chronicles that Chen Sheng asked the rhetorical question as a call to war: "Are kings, generals, and ministers merely born into their kind?".

[3] A prior is an innate behavior injected into the model before seeing any data. This innate behavior may be enforced throughout all the design choices taken during the model formulation, from the data collection and labeling to the architecture design and training procedure.

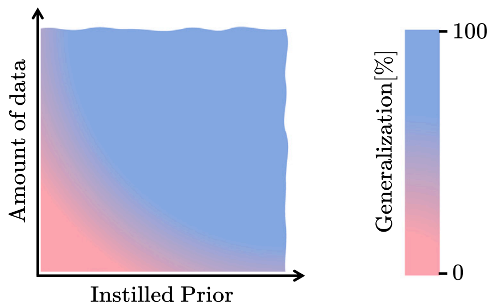

Fig. 5. Relationship between instilled prior knowledge and required amount of data for training learning-based models. Data alone is enough to learn robust representations as the dataset size approaches infinity. Instilling priors allows to relax this unpractical requirement and achieve the same level of generalization. Note that this illustration considers the instillation only of true priors.

are other notable examples of injected priors (Fig. 6). For example, RNNs are specifically designed to handle time series data by introducing recurrent connections. These connections are used to induce an innate recurrent behavior in the information flowing through the model to memorize time-dependent features. Another great example of architectural prior is CNN. In these models, weight sharing is enforced based on the innate assumption that image processing algorithms are translationally invariant. This results in learning features that are agnostic to what patch of the input image is being considered. Other priors may be injected through the training procedure, such as the loss function (Wu et al., 2021) or the optimization algorithm (Ilboudo, Kobayashi, & Sugimoto, 2022).

In Robotics, leveraging priors is crucial to minimizing the number of dangerous situations that the robot undergoes during the data collection. For aerial vehicles such as quadrotors, this is even more exaggerated. While the ideal solution is to allow a mixture of nature and nurture, priors and data, the challenge of selecting the proper priors remain open. The following sections present several approaches that have demonstrated extraordinary results for injecting priors over learning-based models trained for the quadrotor's system identification task.

5.3.1. Physics-inspired machine learning

For the majority of Robotics applications, physical laws governing the system dynamics are partially known — some parameter values or entire terms in the ODE of the system are unknown (Karniadakis et al., 2021). This prior physical understanding of the system may be injected into the learning-based model to foster its learning convergence and robustness to imperfect data while also providing physically consistent predictions. Therefore, physics-inspired machine learning provides a unified framework to integrate data and physical laws. Notable examples of physics-inspired learning-based models are the family of physics-inspired neural networks (Raissi, Perdikaris, & Karniadakis, 2019).

The simplest solution to steer the learning process towards physically consistent solutions is to augment and penalize the loss function (Jia et al., 2019). For quadrotor system identification, Saviolo, Li, and Loianno (2022) extended the MSE loss function in Eq. (10) with a Physics-Inspired (PI) loss between the physics laws' solution and the model's predictions, hence minimizing the following objective function

$$\begin{aligned}
&\min_{\theta} \mathcal{L}_{\mathrm{MSE}} + \lambda \mathcal{L}_{PI} \,, \\
&\mathcal{L}_{\mathrm{MSE}} = \frac{1}{|B_T|} \sum_{i=1}^{|B_T|} \|\mathbf{x}_i - f(\mathbf{x}_{i-1}, \mathbf{u}_{i-1}; \theta)\|^2 \,, \\
&\mathcal{L}_{\mathrm{PI}} = \frac{1}{|B_P|} \sum_{j=1}^{|B_P|} \|f_{\mathrm{Quad}}(\mathbf{x}_{j-1}, \mathbf{u}_{j-1}) - f(\mathbf{x}_{j-1}, \mathbf{u}_{j-1}; \theta)\|^2 \,,
\end{aligned} \tag{11}$$

where λ is a hyper-parameter, B_T is a batch of demonstration data points, B_P is a batch of points sampled from the entire input space, $\mathbf{x}_i$ is the measured state at time i, $f(\mathbf{x}_{i-1}, \mathbf{u}_{i-1}; \theta)$ and $f_{\mathrm{Quad}}(\mathbf{x}_{j-1}, \mathbf{u}_{j-1})$ give the predicted state at time i by the learning-based model and the quadratic model (8), respectively.

While the MSE loss ensures that the model learns the full dynamics purely from data, the PI loss constrains the predictions to match the underlying equations derived from physics-based principles. Therefore, the PI loss gives the model a physical interpretation of its internal states. From a machine learning perspective, the PI loss can be viewed as an unsupervised regularizer that fosters the model's generalization performance by stabilizing the training process.

When injecting physics laws in the loss function, the choice of the weighting of the different loss terms plays an important role in the convergence of the learning procedure (Wang, Yu, & Perdikaris, 2022) - λ in Eq. (11). The weights should reflect the confidence given to the prior physical understanding of the system. If the prior is uncertain, the weight assigned to the PI loss should be rather small. On the other hand, if the available prior is certain, then its corresponding weight should be large. The weights may also be dynamically changed during training following a curriculum learning strategy. For example, Moseley, Markham, and Nissen-Meyer (2020) warm-started the optimization by switching off the PI loss during half of the training iterations. This strategy resulted in a well-rounded exploration of the optimization space during the first optimization steps and a full exploitation of the physical laws during the remaining training iterations.

While instilling soft physics constraints provides a flexible and general solution to teach physical laws to learning-based models, it is not the only working strategy. Physics-inspired architectural priors may be instilled in the model to encode desired properties of the learned solutions. Notable examples are Hamiltonian constraints (Jin, Zhang, Zhu, Tang, & Karniadakis, 2020), Dirichlet boundary conditions (Sheng & Yang, 2021), even/odd symmetry and energy conservation (Mattheakis, Protopapas, Sondak, Di Giovanni, & Kaxiras, 2019).

5.3.2. Meta-learning: Learning to learn

Despite their remarkable effectiveness, online learning approaches (Section 4.4) are not capable yet to achieve fast online adaptation, as the time required to adapt to different operating regimes remains above 10 s. A key bottleneck is the global prior model learned offline. Generally, this model does not contain any information about the testing operating regime but has only knowledge of the platform in nominal conditions (Saviolo, Frey, et al., 2022). Therefore, during the online adaptation to an unseen operating regime, the model needs to change considerably its parameters to adapt. Trading off this unbounded generalizability by instilling more solid priors into the model would improve the adaptation time of the online learning procedure (Pautrat, Chatzilygeroudis, & Mouret, 2018). Recently, a popular learning scheme that has captured significant attention in the learning and robotic communities is meta-learning (Finn, Abbeel, & Levine, 2017b; Harrison, Sharma, & Pavone, 2018).

Meta-learning is a framework that seeks to inject prior task[4] distribution knowledge into the model to enable fast online adaptation. The learned model is then ideally capable of quickly adapting to new tasks given limited amounts of data. The core assumption of meta-learning algorithms is that learning to solve a single task has the potential to aid in solving another. This implicitly requires that the training and testing tasks are all drawn from the same task distribution. Hence, meta-learning algorithms can exploit the shared common representations between tasks during operation to quickly adapt. Based on how the task prior is instilled into the model, meta-learning algorithms are classified into optimization-based (Harrison, Sharma, & Pavone,

[4] For system identification, a "task" is the dynamical model associated with a specific operating regime.

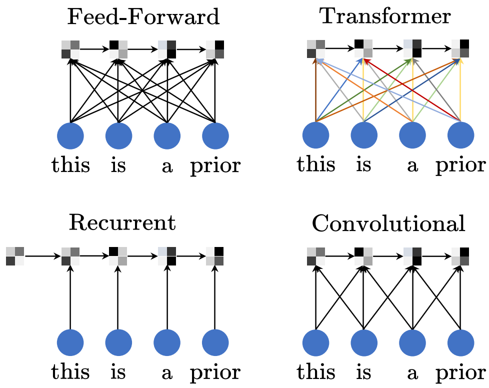

Fig. 6. Notable examples of architectural priors. Colored edges represent fast varying weights due to the Transformer's self-attention mechanism (Adaloglou & Karagiannakos, 2020).

2018; Rajeswaran, Finn, Kakade, & Levine, 2019) and architecture-based (Clavera et al., 2019; Perez, Such, & Karaletsos, 2018).

Optimization-based algorithms, such as MAML (Finn et al., 2017b), FAMLE (Kaushik, Anne, & Mouret, 2020), and DAIML (O'Connell et al., 2022), explicitly seek the initial model parameters that could ensure generalization to the testing task with only few gradient steps. Optimization-based algorithms have been further reformulated as a form of hierarchical Bayes to incorporate Laplace approximations into the weight update (Grant, Finn, Levine, Darrell, & Griffiths, 2018). This allows capturing a larger family of distributions, such as multimodal distributions, and boosting even further the data efficiency (Finn, Xu, & Levine, 2018; Kim et al., 2018).

Architecture-based algorithms, such as IIDA (Evans, Thankaraj, & Pinto, 2022) and ML-GP (Sæmundsson, Hofmann, & Deisenroth, 2018), condition the dynamical model with an auxiliary latent variable that represents the prior over the distribution over the tasks. The dynamical model is trained offline jointly with the latent variable, such that the variable encodes the information of the task and can infer it to the model during operation.

For robotic applications, meta-learning has shown exciting results for fast online adaptation in highly dynamic operating regimes, from legged (Kaushik et al., 2020; Nagabandi et al., 2018) and wheeled (Banerjee, Harrison, Furlong, & Pavone, 2020; McKinnon & Schoellig, 2021) robots to manipulators (Evans et al., 2022). For quadrotors, meta-learning approaches have demonstrated impressive online adaptation on several tasks, including payload transportation (Belkhale et al., 2021) and wind disturbance adaptation (O'Connell et al., 2022; Richards, Azizan, Slotine, & Pavone, 2022).

5.4. Active adaptation

While instilling prior knowledge in the learned model benefits the data efficiency, it concurrently limits its generalization capabilities. For example, meta-learning algorithms are based on the assumption that training and testing tasks share common representations. However, this severely limits the optimization space and assumes knowledge of the testing distribution that, by definition, must remain unknown. Is it possible to improve the data efficiency of the learned model without degrading its generalization capabilities?

Online learning approaches treat the learned model as a passive recipient of data to be processed, hence ignoring the ability of the system to act and influence the operating environment for effective data gathering. Contrarily, active (online) learning studies the model's ability to actively select actions to influence what data to learn from (Cohn, Ghahramani, & Jordan, 1996; Ren et al., 2021). The action selection is performed by the model itself or an auxiliary "oracle" with the common goal to maximize a performance metric, such as the maximum entropy (Sahli Costabal et al., 2020). When the actions are selected properly, the model achieves higher sample efficiency while reinforcing its generalization capabilities.

Recently, active learning has attracted significant attention for the system identification task of robots (Abraham & Murphey, 2019; Capone et al., 2020; Chakrabarty, Danielson, Cairano, & Raghunathan, 2022). For example, Lew, Sharma, Harrison, Bylard, and Pavone (2022) proposed an active learning strategy to reduce the dynamical model uncertainty during the exploration of the state-control space. Once the uncertainty level is below a user-defined threshold and a globally accurate model of the system is obtained, the robot is controlled to execute the desired task by exploiting the global model. Despite being coupled with probabilistic adaptation, safety, and feasibility guarantees and demonstrated on a free-flyer robot, this approach is not practical for systems with highly-nonlinear dynamics and fast varying operating conditions such as aerial vehicles. The continuously changing dynamics would never allow the exploration stage to converge. Moreover, the proposed exploration-exploitation strategy may not reasonably achievable by parametric models, such as FNNs, which can potentially require an infinite number of points to model the entire state-control space. In practice, parametric models that are continuously adapted online are not required to fully capture an accurate global model of the system to ensure controllability and stability, but should only be precise locally on a subset of the state-control space.

Tackling these problems, Saviolo, Frey, et al. (2022) proposed an uncertainty-aware model predictive controller as oracle to select actions that jointly optimize the control performance and online learning data efficiency. The oracle continuously seeks to exploit the learned model and choose "safe" actions, hence conditioning the exploration phase. The approach is demonstrated on an agile quadrotor platform with multiple challenging operating regimes, ranging from wind disturbances and suspended payloads to drastic changes to the system configuration. Notably, the learning-based dynamical model is a FNN trained on data collected from demonstrations with a nominal system configuration of the quadrotor. Therefore, when deployed in different operating regimes, the system is fully tested in a never-seen-before distribution, hence demonstrating the full generalization power of active learning.

5.5. Neural implicit representations

Neural Radiance Fields (NeRFs) have emerged as a powerful tool for representing complex 3D scenes by learning a continuous function that maps a 3D point to its corresponding color and opacity in an image (Mildenhall et al., 2021). This representation enables highly realistic rendering of scenes from any viewpoint or lighting condition. NeRF has found applications in various fields, including 3D reconstruction (Qiu, Sun, Marques, & Hauser, 2022), virtual and augmented reality (Deng et al., 2022), and robotics where it has shown great promise in tasks such as state estimation and mapping (Moreau, Piasco, Tsishkou, Stanciulescu, & de La Fortelle, 2022; Rosinol, Leonard, & Carlone, 2022; Zhu et al., 2022). For instance, Rosinol et al. (2022) proposed a real-time monocular localization and mapping method using NeRF that can handle both static and dynamic scenes, making it applicable to robotics and augmented reality settings.

NeRF applications in robotics are rapidly increasing and evolving, including motion planning, navigation, and collision avoidance tasks (Chen, Culbertson, & Schwager, 2023; Kurenkov et al., 2022; Long, Qian, Cortés, & Atanasov, 2021; Ni & Qureshi, 2022; Pantic, Cadena, Siegwart, & Ott, 2022; Sznaier Camps, Dyro, Pavone, & Schwager, 2022). For example, Ha, Driess, and Toussaint (2022) proposed a novel approach for manipulation planning using neural implicit models. The proposed method employed a CNN to learn an implicit representation of the scene and predict the depth and surface normals of the scene objects. This approach enabled robots to plan their actions based solely on

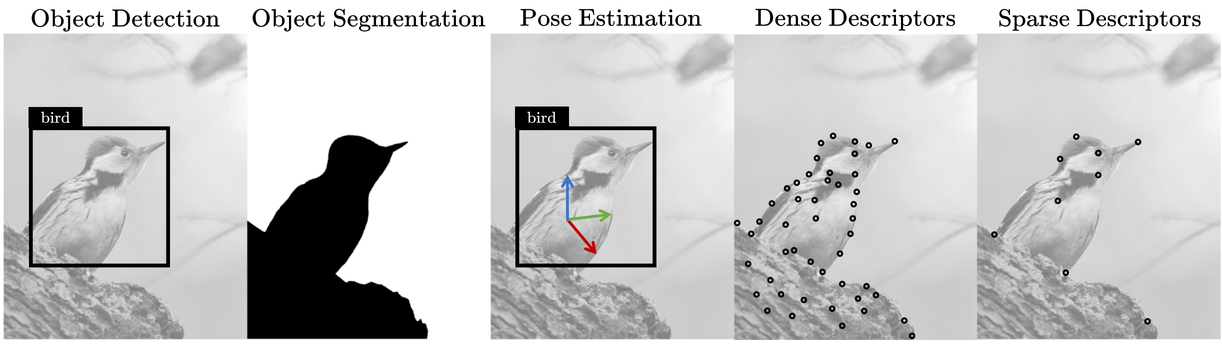

Fig. 7. Examples of state representations used when coupling perception and action. From left to right: object detection, object segmentation, object pose estimation, dense descriptor representation, sparse descriptor representation.

visual information, leading to more efficient and flexible manipulation strategies. Furthermore, for quadrotors, NeRF-based approaches have been proposed for planning in complex environments using only visual input Adamkiewicz et al. (2022). The proposed approach consisted of a FNN trained to estimate the robot's position and orientation from visual input generated through NeRF.

The advancements in NeRFs highlight their potential for a range of robotics applications and establish neural implicit representations as a pivotal method that will drive future research in robotics, including the area of modeling dynamics .

5.6. Coupling perception and action

The core components of classical robot control frameworks are globally-consistent state estimation, dynamically-feasible motion planning, robust perception, and handcrafted control design. Estimation algorithms (Cadena et al., 2016) extract characteristic descriptors in the environment and elaborate them to obtain partial or full information of the robot's state belief. Then, the planner generates a global trajectory and the controller operates the robot to follow it. This cascaded approach inevitably implies the need to highly engineer all these components (e.g., laborious manual tuning of parameters, handcrafting the mathematical relationships between perception and action spaces) and make simplifying assumptions (e.g., human heuristics), which inevitably leads to compounding delays and errors. Moreover, perception quantities have to be stored and processed at different levels of abstractions (e.g., features, point cloud map, voxels) requiring relevant computational and memory resources for processing redundantly the same information. This calls for novel representations that couple perception and action, hence reducing the inference latency to estimate the robot state and the overall memory requirements. Concurrently, these representations would produce more accurate motion estimates robust to camera calibration errors as well as the ability to directly parse the perception and uncertainty information on the robot's state.

Expressing the system dynamics and perception variables in the same topological space requires image-based state representations that are interpretable, robust to disturbances (e.g., rapid illumination changes, motion blur, objects deformation), invariant to translation and rotation, temporally consistent, capable to include geometric costs and constraints, and other key properties that should be tailored to the desired task. Prior work on image-based representations have considered different level of abstraction for sensory data, ranging from objects' bounding boxes, binary masks, and poses to sparse and dense descriptors (Fig. 7).

Object detection deals with detecting instances of objects of a certain class and marking them with bounding boxes (Zhu et al., 2014). Leveraging objects' bounding box representations allows to infer interpretability to the control framework because actions are predicted based on the bounding boxes dynamical evolution. However, fine-grained location and orientation of object parts is not achievable by simple bounding box predictions, hence preventing the controller to actuate precise actions. Moreover, object detectors can only regress the instances of object classes present in the training set. This severely limits the generalization capabilities of these methods for robotic applications. In fact, while bounding box representations may come useful for visual servoing tasks where the target is known in advance and it is assumed to remain in the camera's field of view (Ramon-Soria, Arrue, & Ollero, 2020), general robotic applications do not know a priori which objects will be seen by the camera.

Object segmentation extends bounding box detection to the pixel-level (He, Gkioxari, Dollár, & Girshick, 2017; Long, Shelhamer, & Darrell, 2015). The output of segmentation predictors is a binary mask, where pixels are 1 if belonging to the object and 0 otherwise. Binary mask representations enable both interpretable control and fine-grained location and orientation of object parts. However, the generalization problem for these representations is even more challenging than the one for bounding boxes. Building accurate segmentation data sets is extraordinarily difficult due to the tedious, time-consuming labeling process (Saviolo, Bonotto, et al., 2022). As a result, the size of the data set that can be collected is severely limited — at reasonable expense, and the scarcity of data induces the learning-based models to overfit the data's intrinsic noise.

Object pose estimation involves identifying the position and orientation of objects in the camera's field of view (Sahin & Kim, 2018; Wang et al., 2019). While leveraging these representations, the control framework is able to reason on the three-dimensional space that surrounds the robot and take proper actions. Moreover, the decision process performed by the control framework is interpretable because it is based on the dynamical evolution of the object poses in the camera field of view. However, pose representations provide little degrees of freedom to topology variations (e.g., deformations) affecting objects seen during training, leading to ambiguous predictions. Therefore, similarly to bounding boxes and masks, pose representations also offer inadequate generalization capabilities.

A descriptor is a feature vector that represents the essential characteristics of a keypoint in an image (e.g., corner, edge). The earliest approaches for extracting image descriptors have focused on handcrafted feature extractors (Lowe, 1999) due to their advantages in dealing with image transformations (e.g., rotation, scale, brightness). Recently, CNNs have attracted increasing interest in the community and demonstrated impressive performance, de facto representing the current state-of-the-art approaches for descriptor prediction (Güler, Neverova, & Kokkinos, 2018). However, learning-based approaches often rely on strong supervision in the form of hand-selected ground-truth labels, which are usually incredibly tedious and time-consuming to annotate. Moreover, manually extracting meaningful descriptors is challenging because the annotator has to identify salient points in the training images that would allow their re-identification in novel scenarios. As a result, recent works have focused on self-supervised approaches for extracting descriptors from images or videos (DeTone, Malisiewicz, & Rabinovich, 2018). For example, training data may be efficiently generated by using dense mapping and tracking methods (e.g., KinectFusion (Schmidt, Newcombe, & Fox, 2016)) or designing differentiable descriptor architectural bottlenecks that map a spatial probability map to image coordinates (Jakab, Gupta, Bilen, & Vedaldi, 2018).

Descriptor representations can be trained to incorporate multiple desirable properties and geometric costs and constraints. For example, (Tang et al., 2021) exploited the temporal context in videos to learn depth-aware descriptors without human supervision. Moreover, when multiple cameras facing the robot are available in the environment, descriptors can be trained to learn three-dimensional features (Chen, Abbeel, & Pathak, 2021). Consequently, the control framework makes predictions by reasoning on the three-dimensional structure of the world, which results in higher performance in partially observable environments.

Tuning the density of the extracted descriptors is key for achieving high-performing results in the desired task. For example, sparse descriptors are computationally efficient and well generalize to unseen object's topology variations and partially occluded objects. However, due to their sparsity, these descriptors significantly degrade in poorly textured scenes. In these settings, dense descriptors can leverage the stronger inductive priors over the geometry of the objects in the training data.

Leveraging descriptor representations offers an interpretable, unambiguous representation for control frameworks (Sundaresan et al., 2020). For example, Manuelli, Li, Florence, and Tedrake (2020) applied model predictive control using a descriptor representation for operating a manipulator robot. The descriptors are extracted by sampling dense object-centric representations collected without supervision (Florence, Manuelli, & Tedrake, 2018).

6. Summary and concluding remarks

Over the last decade, the system identification task in robotics has witnessed impressive advances subsequent to the extraordinary development and application of learning-based techniques. As traditional modeling of dynamics of aerial robots such as quadrotors is performed using physics-based principles, it fails to represent complex non-linear disturbance phenomena, such as friction, deformation, aerodynamic effects, and vibrations, that cannot be directly measured and therefore do not have explicit analytic equations. Contrarily, learning-based techniques that extract and adapt the dynamics model purely from data enable accurate modeling and high-performance control even in challenging flight operating conditions, capturing all the non-linear effects that act on the system and are hidden to traditional physics-based approaches. Furthermore, learning the system dynamics from data has the potential to terrifically impact the development of multiple robotic systems, enabling fast and accurate modeling. However, purely learning the dynamics model from data poses several challenges. First, data preparation and denoising are key to ensuring that the learning model extracts the desired dynamical system behavior. Second, purely learning from data would require infinite samples and computing power, therefore inductive priors must be instilled during the design process of the learning algorithm. At the same time, introducing too many priors into the algorithm would severely limit or bias what it can learn. Therefore, only "true" priors should be instilled. Third, active learning strategies jointly ensure sample efficient adaptation of the dynamics model and continuous reinforcement of the model generalization capabilities. Finally, coupling perception and action by expressing the system dynamics and perception variables in the same topological space reduces the inference latency while directly parsing the perception information from the sensor to the robot state space. This contributes to improved accuracy and efficiently maximizes the robot's future knowledge.

Declaration of competing 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.

Funding

We gratefully acknowledge the following organizations for supporting this work: the National Science Foundation (NSF) Career Award under grant number 2145277, the Defense Advanced Research Projects Agency (DARPA) Young Faculty Award Grant under grant number D22AP00156-00, Qualcomm Research, Nokia, and NYU Wireless.

Data availability

No data was used for the research described in the article.

References

Abraham, I., & Murphey, T. D. (2019). Active learning of dynamics for data-driven control using Koopman operators. *IEEE Transactions on Robotics*, *35*(5), 1071–1083.

Adaloglou, N., & Karagiannakos, S. (2020). How attention works in deep learning: understanding the attention mechanism in sequence models. https://Theaisummer.Com/. URL: https://theaisummer.com/attention/.

Adamkiewicz, M., Chen, T., Caccavale, A., Gardner, R., Culbertson, P., Bohg, J., et al. (2022). Vision-only robot navigation in a neural radiance world. *IEEE Robotics and Automation Letters*, *7*(2), 4606–4613.

Alkayas, A. Y., Chehadeh, M., Ayyad, A., & Zweiri, Y. (2022). Systematic online tuning of multirotor UAVs for accurate trajectory tracking under wind disturbances and in-flight dynamics changes. *IEEE Access*.

Ambikasaran, S., Foreman-Mackey, D., Greengard, L., Hogg, D. W., & O'Neil, M. (2016). Fast direct methods for Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(2), 252–265.

Annaswamy, A. M., & Fradkov, A. L. (2021). A historical perspective of adaptive control and learning. *Annual Reviews in Control*, *52*, 18–41.

Åström, K. J. (1983). Theory and applications of adaptive control—A survey. *Automatica*, *19*(5), 471–486.

Atkeson, C. G., & Santamaria, J. C. (1997). A comparison of direct and model-based reinforcement learning. In *Proceedings of international conference on robotics and automation. Vol. 4* (pp. 3557–3564). IEEE.

Bai, S., Kolter, J. Z., & Vladlen, K. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271.

Banerjee, S., Harrison, J., Furlong, P. M., & Pavone, M. (2020). Adaptive meta-learning for identification of rover-terrain dynamics. arXiv preprint arXiv:2009.10191.

Bansal, S., Akametalu, A. K., Jiang, F. J., Laine, F., & Tomlin, C. J. (2016). Learning quadrotor dynamics using neural network for flight control. In *IEEE conference on decision and control* (pp. 4653–4660).

Bauersfeld, L., Kaufmann, E., Foehn, P., Sun, S., & Scaramuzza, D. (2021). NeuroBEM: Hybrid aerodynamic quadrotor model. *Robotics: Science and Systems Foundation*.

Bauersfeld, L., & Scaramuzza, D. (2022). Range, endurance, and optimal speed estimates for multicopters. *IEEE Robotics and Automation Letters*, *7*(2), 2953–2960.

Bechtle, S., Hammoud, B., Rai, A., Meier, F., & Righetti, L. (2021). Leveraging forward model prediction error for learning control. In *2021 IEEE international conference on robotics and automation* (pp. 4445–4451). IEEE.

Belkhale, S., Li, R., Kahn, G., McAllister, R., Calandra, R., & Levine, S. (2021). Model-based meta-reinforcement learning for flight with suspended payloads. *IEEE Robotics and Automation Letters*, *6*(2), 1471–1478.

Berkenkamp, F., Schoellig, A. P., & Krause, A. (2016). Safe controller optimization for quadrotors with Gaussian processes. In *2016 IEEE international conference on robotics and automation* (pp. 491–496). IEEE.

Bertsekas, D. P. (2005). *Dynamic programming and optimal control, Volume 1 of optimization and computation series* (3rd ed.). (p. 2). Belmont, MA, USA: Athena Scientific, Cited on.

Bluck, R. S. (1961). Plato's meno. *Phronesis*, *6*(1–2), 94–101.

Bonatti, R., Vemprala, S., Ma, S., Frujeri, F., Chen, S., & Kapoor, A. (2022). Pact: Perception-action causal transformer for autoregressive robotics pre-training. arXiv preprint arXiv:2209.11133.

Borovykh, A., Bohte, S., & Oosterlee, C. W. (2017). Conditional time series forecasting with convolutional neural networks. arXiv preprint arXiv:1703.04691.

Bouabdallah, S., & Siegwart, R. (2007). Full control of a quadrotor. In *2007 IEEE/RSJ international conference on intelligent robots and systems* (pp. 153–158). IEEE.

Bristow, D., Tharayil, M., & Alleyne, A. (2006). A survey of iterative learning control. *IEEE Control Systems Magazine*, *26*(3), 96–114.

Brunton, S. L., Proctor, J. L., & Kutz, J. N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, *113*(15), 3932–3937.

Butcher, J. C. (1996). A history of Runge-Kutta methods. *Applied Numerical Mathematics*, *20*(3), 247–260.

Butterworth, S., et al. (1930). On the theory of filter amplifiers. *Wireless Engineer*, *7*(6), 536–541.

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., et al. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, *32*(6), 1309–1332.

Cao, C., Ma, L., & Xu, Y. (2012). Adaptive control theory and applications. *Journal of Control Science and Engineering*, *2012*.

Capone, A., Noske, G., Umlauft, J., Beckers, T., Lederer, A., & Hirche, S. (2020). Localized active learning of Gaussian process state space models. In *Proceedings of machine learning research*: *vol. 120*, *Proceedings of the 2nd conference on learning for dynamics and control* (pp. 490–499). PMLR.

Cash, J. (1980). On the integration of stiff systems of ODEs using extended backward differentiation formulae. *Numerische Mathematik*, *34*, 235–246.

Chakrabarty, A., Danielson, C., Cairano, S. D., & Raghunathan, A. (2022). Active learning for estimating reachable sets for systems with unknown dynamics. *IEEE Transactions on Cybernetics*, *52*(4), 2531–2542.

Chaplot, D. S., Pathak, D., & Malik, J. (2021). Differentiable spatial planning using transformers. In *International conference on machine learning* (pp. 1484–1495). PMLR.

Chen, B., Abbeel, P., & Pathak, D. (2021). Unsupervised learning of visual 3d keypoints for control. In *International conference on machine learning* (pp. 1539–1549). PMLR.

Chen, T., Culbertson, P., & Schwager, M. (2023). CATNIPS: Collision avoidance through neural implicit probabilistic scenes. arXiv preprint arXiv:2302.12931.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., et al. (2021). Decision transformer: Reinforcement learning via sequence modeling. *Advances in Neural Information Processing Systems*, *34*, 15084–15097.

Cheng, S., Kim, M., Song, L., Wu, Z., Wang, S., & Hovakimyan, N. (2022). DiffTune: Auto-tuning through auto-differentiation. arXiv preprint arXiv:2209.10021.

Clavera, I., Nagabandi, A., Liu, S., Fearing, R. S., Abbeel, P., Levine, S., et al. (2019). Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. In *International conference on learning representations*. URL: https://openreview.net/forum?id=HyztsoC5Y7.

Cohn, D., Ghahramani, Z., & Jordan, M. (1996). Active learning with statistical models. *Journal of Artificial Intelligence Research*, *4*, 129–145.

Crocetti, F., Mao, J., Saviolo, A., Costante, G., & Loianno, G. (2023). GaPT: Gaussian process toolkit for online regression with application to learning quadrotor dynamics. arXiv preprint arXiv:2303.08181.

Das, A., Lewis, F., & Subbarao, K. (2009). Backstepping approach for controlling a quadrotor using Lagrange form dynamics. *Journal of Intelligent and Robotic Systems*, *56*, 127–151.

Das, S., Roy, S., & Sambasivan, R. (2018). Fast Gaussian process regression for big data. *Big Data Research*, *14*, 12–26.

Davidor, Y. (1991). *Genetic algorithms and robotics: A heuristic strategy for optimization. Vol. 1*. World Scientific.

Degris, T., Pilarski, P. M., & Sutton, R. S. (2012). Model-free reinforcement learning with continuous action in practice. In *2012 American control conference* (pp. 2177–2182). IEEE.

Deisenroth, M., & Rasmussen, C. E. (2011). PILCO: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th international conference on machine learning* (pp. 465–472).

Deisenroth, M. P., Rasmussen, C. E., & Fox, D. (2011). Learning to control a low-cost manipulator using data-efficient reinforcement learning. *Robotics: Science and Systems VII*, *7*, 57–64.

Deng, N., He, Z., Ye, J., Duinkharjav, B., Chakravarthula, P., Yang, X., et al. (2022). Fov-nerf: Foveated neural radiance fields for virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, *28*(11), 3854–3864.

DeTone, D., Malisiewicz, T., & Rabinovich, A. (2018). Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 224–236).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Devo, A., Mao, J., Costante, G., & Loianno, G. (2022). Autonomous single-image drone exploration with deep reinforcement learning and mixed reality. *IEEE Robotics and Automation Letters*, *7*(2), 5031–5038.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

El-Fakdi, A., & Carreras, M. (2013). Two-step gradient-based reinforcement learning for underwater robotics behavior learning. *Robotics and Autonomous Systems*, *61*(3), 271–282.

Emran, B. J., & Najjaran, H. (2018). A review of quadrotor: An underactuated mechanical system. *Annual Reviews in Control*, *46*, 165–180.

Englert, P., Paraschos, A., Peters, J., & Deisenroth, M. P. (2013). Model-based imitation learning by probabilistic trajectory matching. In *2013 IEEE international conference on robotics and automation* (pp. 1922–1927). IEEE.

Evans, B., Thankaraj, A., & Pinto, L. (2022). Context is everything: Implicit identification for dynamics adaptation. In *International conference on robotics and automation* (pp. 2642–2648).

Fernández, S., Graves, A., & Schmidhuber, J. (2007). An application of recurrent neural networks to discriminative keyword spotting. In *International conference on artificial neural networks* (pp. 220–229). Springer.

Finn, C., Abbeel, P., & Levine, S. (2017a). Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning* (pp. 1126–1135). PMLR.

Finn, C., Abbeel, P., & Levine, S. (2017b). Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of machine learning research*: *vol. 70*, *International conference on machine learning* (pp. 1126–1135). PMLR.

Finn, C., Xu, K., & Levine, S. (2018). Probabilistic model-agnostic meta-learning. *Advances in Neural Information Processing Systems*, *31*.

Fliess, M., & Join, C. (2013). Model-free control. *International Journal of Control*, *86*(12), 2228–2252.

Florence, P. R., Manuelli, L., & Tedrake, R. (2018). Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. In *Proceedings of the 2nd conference on robot learning* (pp. 373—385).

Fu, J., Levine, S., & Abbeel, P. (2016). One-shot learning of manipulation skills with online dynamics adaptation and neural network priors. In *2016 IEEE/RSJ international conference on intelligent robots and systems* (pp. 4019–4026). IEEE.

Funahashi, K.-i., & Nakamura, Y. (1993). Approximation of dynamical systems by continuous time recurrent neural networks. *Neural Networks*, *6*(6), 801–806.

Gahlawat, A., Zhao, P., Patterson, A., Hovakimyan, N., & Theodorou, E. (2020). L1-GP: L1 adaptive control with Bayesian learning. In *Conference on learning for dynamics and control. Vol. 120* (pp. 826–837).

Garcia, C. E., Prett, D. M., & Morari, M. (1989). Model predictive control: Theory and practice—A survey. *Automatica*, *25*(3), 335–348.

Gaskett, C., & Cheng, G. (2003). Online learning of a motor map for humanoid robot reaching. In *Proceedings of the 2nd international conference on computational intelligence, Robotics and autonomous systems* (pp. 1–6).

Gear, C. W., & Wells, D. R. (1984). Multirate linear multistep methods. *BIT Numerical Mathematics*, *24*(4), 484–502.

Giuliari, F., Hasan, I., Cristani, M., & Galasso, F. (2021). Transformer networks for trajectory forecasting. In *2020 25th International conference on pattern recognition* (pp. 10335–10342). IEEE.

Giurato, M., & Lovera, M. (2016). Quadrotor attitude determination: a comparison study. In *2016 IEEE conference on control applications* (pp. 21–26).

Glad, T., & Ljung, L. (2018). *Control theory*. CRC Press.

Grant, E., Finn, C., Levine, S., Darrell, T., & Griffiths, T. (2018). Recasting gradient-based meta-learning as hierarchical bayes. arXiv preprint arXiv:1801.08930.

Graves, A., & Schmidhuber, J. (2008). Offline handwriting recognition with multidimensional recurrent neural networks. *Advances in Neural Information Processing Systems*, *21*.

Gregory, I., Gadient, R., & Lavretsky, E. (2011). Flight test of composite model reference adaptive control (CMRAC) augmentation using NASA AirSTAR infrastructure. In *AIAA guidance, navigation, and control conference* (p. 6452).

Grimble, M. J. (1984). Implicit and explicit LQG self-tuning controllers. *IFAC Proceedings Volumes*, *17*(2), 941–947.

Güler, R. A., Neverova, N., & Kokkinos, I. (2018). Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7297–7306).

Ha, J.-S., Driess, D., & Toussaint, M. (2022). Deep visual constraints: Neural implicit models for manipulation planning from visual input. *IEEE Robotics and Automation Letters*, *7*(4), 10857–10864.

Hahn, G. (1991). A modified Euler method for dynamic analyses. *International Journal for Numerical Methods in Engineering*, *32*(5), 943–955.

Hahn, W., et al. (1967). *Stability of motion. Vol. 138*. Springer.

Han, B., Yao, Q., Liu, T., Niu, G., Tsang, I. W., Kwok, J. T., et al. (2020). A survey of label-noise representation learning: Past, present and future. arXiv preprint arXiv:2011.04406.

Hanover, D., Foehn, P., Sun, S., Kaufmann, E., & Scaramuzza, D. (2021). Performance, precision, and payloads: Adaptive nonlinear mpc for quadrotors. *IEEE Robotics and Automation Letters*, *7*(2), 690–697.

Harrison, J., Sharma, A., Calandra, R., & Pavone, M. (2018). Control adaptation via meta-learning dynamics. In *NeurIPS workshop on meta-learning*.

Harrison, J., Sharma, A., & Pavone, M. (2018). Meta-learning priors for efficient online bayesian regression. In *International workshop on the algorithmic foundations of robotics* (pp. 318–337). Springer.

Hashemian, N., & Armaou, A. (2017). Stochastic MPC design for a two-component granulation process. In *2017 American control conference* (pp. 4386–4391). IEEE.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961–2969).

Hirano, K., Nishimura, S., & Mitra, S. (1974). Design of digital notch filters. *IEEE Transactions on Communications*, *22*(7), 964–970.

Hock, A., & Schoellig, A. P. (2016). Distributed iterative learning control for a team of quadrotors. In *2016 IEEE 55th conference on decision and control* (pp. 4640–4646). IEEE.

Hock, A., & Schoellig, A. P. (2019). Distributed iterative learning control for multi-agent systems: Theoretic developments and application to formation flying. *Autonomous Robots*, *43*, 1989–2010.

Hou, Z.-S., & Wang, Z. (2013). From model-based control to data-driven control: Survey, classification and perspective. *Information Sciences*, *235*, 3–35, Data-based Control, Decision, Scheduling and Fault Diagnostics.

Huang, D., Chen, Y., Meng, D., & Sun, P. (2019). Adaptive iterative learning control for high-speed train: A multi-agent approach. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *51*(7), 4067–4077.

Hwangbo, J., Sa, I., Siegwart, R., & Hutter, M. (2017). Control of a quadrotor with reinforcement learning. *IEEE Robotics and Automation Letters*, *2*(4), 2096–2103.

Ilboudo, W. E. L., Kobayashi, T., & Sugimoto, K. (2022). Robust stochastic gradient descent with student-t distribution based first-order momentum. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(3), 1324–1337.

Invernizzi, D., Lovera, M., & Zaccarian, L. (2022). Global robust attitude tracking with torque disturbance rejection via dynamic hybrid feedback. *Automatica*, *144*, Article 110462.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448–456). PMLR.

Jakab, T., Gupta, A., Bilen, H., & Vedaldi, A. (2018). Unsupervised learning of object landmarks through conditional image generation. *Advances in Neural Information Processing Systems*, *31*.

Jamone, L., Natale, L., Nori, F., Metta, G., & Sandini, G. (2012). Autonomous online learning of reaching behavior in a humanoid robot. *International Journal of Humanoid Robotics*, *9*(03), Article 1250017.

Jayasree, P. S., Raj, P., Kumar, P., Siddavatam, R., & Ghrera, S. (2013). A fast novel algorithm for salt and pepper image noise cancellation using cardinal B-splines. *Signal, Image and Video Processing*, *7*, 1145–1157.

Jenni, S., & Favaro, P. (2018). Deep bilevel learning. In *Proceedings of the European conference on computer vision* (pp. 618–633).

Jia, X., Willard, J., Karpatne, A., Read, J., Zwart, J., Steinbach, M., et al. (2019). Physics guided RNNs for modeling dynamical systems: A case study in simulating lake temperature profiles. In *Proceedings of the 2019 SIAM international conference on data mining* (pp. 558–566). SIAM.

Jin, P., Zhang, Z., Zhu, A., Tang, Y., & Karniadakis, G. E. (2020). SympNets: Intrinsic structure-preserving symplectic networks for identifying Hamiltonian systems. *Neural Networks*, *132*, 166–179.

Joshi, G., Virdi, J., & Chowdhary, G. (2021). Asynchronous deep model reference adaptive control. In *Conference on robot learning. Vol. 155* (pp. 984–1000).

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, *3*(6), 422–440.

Kaufmann, E., Loquercio, A., Ranftl, R., Müller, M., Koltun, V., & Scaramuzza, D. (2020). Deep drone acrobatics. In *Robotics: science and systems* (pp. 1—10).

Kaushik, R., Anne, T., & Mouret, J.-B. (2020). Fast online adaptation in robotics through meta-learning embeddings of simulated priors. In *2020 IEEE/RSJ international conference on intelligent robots and systems* (pp. 5269–5276). IEEE.

Kim, J., Kang, M.-S., & Park, S. (2010). Accurate modeling and robust hovering control for a quad-rotor VTOL aircraft. In *Selected papers from the 2nd international symposium on UAVs, Reno, Nevada, USA June 8–10, 2009* (pp. 9–26). Springer.

Kim, T., Yoon, J., Dia, O., Kim, S., Bengio, Y., & Ahn, S. (2018). Bayesian model-agnostic meta-learning. arXiv preprint arXiv:1806.03836.

Knott, G. D. (1999). *Interpolating cubic splines. Vol. vol. 18*. Springer Science & Business Media.

Krogh, A., & Hertz, J. (1991). A simple weight decay can improve generalization. *Advances in Neural Information Processing Systems*, *4*.

Kumar, A., Fu, Z., Pathak, D., & Malik, J. (2021). Rma: Rapid motor adaptation for legged robots. arXiv preprint arXiv:2107.04034.

Kurenkov, M., Potapov, A., Savinykh, A., Yudin, E., Kruzhkov, E., Karpyshev, P., et al. (2022). NFOMP: Neural field for optimal motion planner of differential drive robots with nonholonomic constraints. *IEEE Robotics and Automation Letters*, *7*(4), 10991–10998.

Lea, C., Vidal, R., Reiter, A., & Hager, G. D. (2016). Temporal convolutional networks: A unified approach to action segmentation. In *European conference on computer vision* (pp. 47–54). Springer.

LeCun, Y., Bengio, Y., et al. (1995). Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*, *3361*(10), 1995.

Lee, J. H., & Lee, K. S. (2007). Iterative learning control applied to batch processes: An overview. *Control Engineering Practice*, *15*(10), 1306–1318, Special Issue - International Symposium on Advanced Control of Chemical Processes (ADCHEM).

Levine, S., & Koltun, V. (2013). Guided policy search. In *International conference on machine learning* (pp. 1–9). PMLR.

Levitt, M. (2013). Perceptions of nature, nurture and behaviour. *Life Sciences, Society and Policy*, *9*(1), 1–11.

Lew, T., Sharma, A., Harrison, J., Bylard, A., & Pavone, M. (2022). Safe active dynamics learning and control: A sequential exploration–exploitation framework. *IEEE Transactions on Robotics*, 1–20.

Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*.

Li, G., Tunchez, A., & Loianno, G. (2022). Learning model predictive control for quadrotors. In *2022 International conference on robotics and automation* (pp. 5872–5878).

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., et al. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 10012–10022).

Locke, J. (1948). An essay concerning human understanding, 1690. *Essay*, *1*.

Loianno, G., Brunner, C., McGrath, G., & Kumar, V. (2017). Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and IMU. *IEEE Robotics and Automation Letters*, *2*(2), 404–411.

Long, K., Qian, C., Cortés, J., & Atanasov, N. (2021). Learning barrier functions with memory for robust safe navigation. *IEEE Robotics and Automation Letters*, *6*(3), 4931–4938.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431–3440).

Loquercio, A., Saviolo, A., & Scaramuzza, D. (2022). Autotune: Controller tuning for high-speed flight. *IEEE Robotics and Automation Letters*, *7*(2), 4432–4439.

Losing, V., Hammer, B., & Wersing, H. (2015). Interactive online learning for obstacle classification on a mobile robot. In *2015 International joint conference on neural networks* (pp. 1–8). IEEE.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision. Vol. 2* (pp. 1150–1157). IEEE.

Luo, Y., & Mesgarani, N. (2019). Conv-TasNet: Surpassing ideal time-frequency magnitude masking for speech separation. *IEEE Transactions on Audio, Speech, and Language Processing*.

Manuelli, L., Li, Y., Florence, P., & Tedrake, R. (2020). Keypoints into the future: Self-supervised correspondence in model-based reinforcement learning. arXiv preprint arXiv:2009.05085.

Mao, J., Nogar, S., Kroninger, C. M., & Loianno, G. (2023). Robust active visual perching with quadrotors on inclined surfaces. *IEEE Transactions on Robotics*.

Marcus, G. F. (2004). *The birth of the mind: How a tiny number of genes creates the complexities of human thought*. Basic Civitas Books.

Marzat, J., Piet-Lahanier, H., Damongeot, F., & Walter, E. (2012). Model-based fault diagnosis for aerospace systems: a survey. *Proceedings of the Institution of Mechanical Engineers, Part G (Journal of Aerospace Engineering)*, *226*(10), 1329–1360.

Mattheakis, M., Protopapas, P., Sondak, D., Di Giovanni, M., & Kaxiras, E. (2019). Physical symmetries embedded in neural networks. arXiv preprint arXiv:1904.08991.

Maxwell, J. C. (1868). I. On governors. *Proceedings of the Royal Society of London*, (16), 270–283.

McKinnon, C. D., & Schoellig, A. P. (2021). Meta learning with paired forward and inverse models for efficient receding horizon control. *IEEE Robotics and Automation Letters*, *6*(2), 3240–3247.

Meng, D., & Zhang, J. (2021). Cooperative learning for switching networks with nonidentical nonlinear agents. *IEEE Transactions on Automatic Control*, *66*(12), 6131–6138.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. (2021). Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, *65*(1), 99–106.

Minorsky, N. (1922). Directional stability of automatically steered bodies. *Journal of the American Society for Naval Engineers*, *34*(2), 280–309.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., et al. (2013). Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.

Moreau, A., Piasco, N., Tsishkou, D., Stanciulescu, B., & de La Fortelle, A. (2022). LENS: Localization enhanced by NeRF synthesis. In *Conference on robot learning* (pp. 1347–1356). PMLR.

Moseley, B., Markham, A., & Nissen-Meyer, T. (2020). Solving the wave equation with physics-informed deep learning. arXiv preprint arXiv:2006.11894.

Nagabandi, A., Clavera, I., Liu, S., Fearing, R. S., Abbeel, P., Levine, S., et al. (2018). Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. arXiv preprint arXiv:1803.11347.

Navarro-Alarcon, D., Liu, Y.-H., Romero, J. G., & Li, P. (2013). Model-free visually servoed deformation control of elastic objects by robot manipulators. *IEEE Transactions on Robotics*, *29*(6), 1457–1468.

Ni, R., & Qureshi, A. H. (2022). NTFields: Neural time fields for physics-informed robot motion planning. arXiv preprint arXiv:2210.00120.

O'Connell, M., Shi, G., Shi, X., Azizzadenesheli, K., Anandkumar, A., Yue, Y., et al. (2022). Neural-Fly enables rapid learning for agile flight in strong winds. *Science Robotics*, *7*(66).

Ortega, R., & Panteley, E. (2014). L1–"adaptive" control always converges to a linear PI control and does not perform better than the PI. *IFAC Proceedings Volumes*, *47*(3), 6926–6928.

Ostafew, C. J., Schoellig, A. P., & Barfoot, T. D. (2016). Robust constrained learning-based NMPC enabling reliable mobile robot path tracking. *International Journal of Robotics Research*, *35*(13), 1547–1563.

Pantic, M., Cadena, C., Siegwart, R., & Ott, L. (2022). Sampling-free obstacle gradients and reactive planning in Neural Radiance Fields. In *Workshop on" motion planning with implicit neural representations of geometry" at 2022 IEEE international conference on robotics and automation*.

Pari, J., Shafiullah, N. M., Arunachalam, S. P., & Pinto, L. (2021). The surprising effectiveness of representation learning for visual imitation. arXiv preprint arXiv:2112.01511.

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International conference on machine learning* (pp. 1310–1318). PMLR.

Pautrat, R., Chatzilygeroudis, K., & Mouret, J.-B. (2018). Bayesian optimization with automatic prior selection for data-efficient direct policy search. In *2018 IEEE international conference on robotics and automation* (pp. 7571–7578). IEEE.

Peng, M., Zhu, B., & Jiao, J. (2021). Linear representation meta-reinforcement learning for instant adaptation. arXiv preprint arXiv:2101.04750.

Perez, C. F., Such, F. P., & Karaletsos, T. (2018). Efficient transfer learning and online adaptation with latent variable models for continuous control. arXiv preprint arXiv:1812.03399.

Pérez-Arancibia, N. O., Duhamel, P.-E. J., Ma, K. Y., & Wood, R. J. (2015). Model-free control of a hovering flapping-wing microrobot. *Journal of Intelligent and Robotic Systems*, *77*(1), 95–111.

Peters, J., & Schaal, S. (2006). Policy gradient methods for robotics. In *2006 IEEE/RSJ international conference on intelligent robots and systems* (pp. 2219–2225). IEEE.

Pounds, P., Mahony, R., & Corke, P. (2010). Modelling and control of a large quadrotor robot. *Control Engineering Practice*, *18*(7), 691–699.

Pravitra, J., Ackerman, K. A., Cao, C., Hovakimyan, N., & Theodorou, E. A. (2020). L1-adaptive MPPI architecture for robust and agile control of multirotors. In *IEEE/RSJ international conference on intelligent robots and systems* (pp. 7661–7666).

Qiu, R.-Z., Sun, Y., Marques, J. M. C., & Hauser, K. (2022). Real-time semantic 3D reconstruction for high-touch surface recognition for robotic disinfection. In *2022 IEEE/RSJ international conference on intelligent robots and systems* (pp. 9919–9925). IEEE.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). *Improving language understanding by generative pre-training*: *Technical Report*, OpenAI.

Raissi, M., Perdikaris, P., & Karniadakis, G. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, *378*, 686–707.

Rajeswaran, A., Finn, C., Kakade, S. M., & Levine, S. (2019). Meta-learning with implicit gradients. *Advances in Neural Information Processing Systems*, *32*.

Ramon-Soria, P., Arrue, B. C., & Ollero, A. (2020). Grasp planning and visual servoing for an outdoors aerial dual manipulator. *Engineering*, *6*(1), 77–88.

Ren, P., Xiao, Y., Chang, X., Huang, P.-Y., Li, Z., Gupta, B. B., et al. (2021). A survey of deep active learning. *ACM Computing Surveys*, *54*(9), 1–40.

Richards, S. M., Azizan, N., Slotine, J.-J., & Pavone, M. (2022). Control-oriented meta-learning. arXiv preprint arXiv:2204.06716.

Romero, A., Govil, S., Yilmaz, G., Song, Y., & Scaramuzza, D. (2022). Weighted maximum likelihood for controller tuning. arXiv preprint arXiv:2210.11087.

Rosinol, A., Leonard, J. J., & Carlone, L. (2022). NeRF-SLAM: Real-time dense monocular SLAM with neural radiance fields. arXiv preprint arXiv:2210.13641.

Rosolia, U., & Borrelli, F. (2017). Learning model predictive control for iterative tasks: A computationally efficient approach for linear system. *IFAC-PapersOnLine*, *50*(1), 3142–3147, 20th IFAC World Congress.

Rosolia, U., & Borrelli, F. (2018). Learning model predictive control for iterative tasks. A data-driven control framework. *IEEE Transactions on Automatic Control*, *63*(7), 1883–1896.

Routh, E. J. (1877). *A treatise on the stability of a given state of motion: particularly steady motion. Being the essay to which the adams prize was adjudged in 1877, in the University of Cambridge*. Macmillan and Company.

Rudin, L. I., Osher, S., & Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, *60*(1–4), 259–268.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536.

Saad, E. W., Caudell, T. P., & Wunsch, D. C. (1999). Predictive head tracking for virtual reality. In *IJCNN'99. International joint conference on neural networks. Proceedings. Vol. 6* (pp. 3933–3936). IEEE.

Sæmundsson, S., Hofmann, K., & Deisenroth, M. P. (2018). Meta reinforcement learning with latent variable gaussian processes. arXiv preprint arXiv:1803.07551.

Sahin, C., & Kim, T.-K. (2018). Category-level 6d object pose recovery in depth images. In *Proceedings of the European conference on computer vision (ECCV) workshops*.

Sahli Costabal, F., Yang, Y., Perdikaris, P., Hurtado, D. E., & Kuhl, E. (2020). Physics-informed neural networks for cardiac activation mapping. *Frontiers in Physics*, *8*.

Sanchez-Gonzalez, A., Heess, N., Springenberg, J., Merel, J., Riedmiller, M., Hadsell, R., et al. (2018). Graph networks as learnable physics engines for inference and control. In *International conference on machine learning*.

Sarkka, S., Solin, A., & Hartikainen, J. (2013). Spatiotemporal learning via infinite-dimensional Bayesian filtering and smoothing: A look at Gaussian process regression through Kalman filtering. *IEEE Signal Processing Magazine*, *30*(4), 51–61.

Saviolo, A., Bonotto, M., Evangelista, D., Imperoli, M., Lazzaro, J., Menegatti, E., et al. (2022). Learning to segment human body parts with synthetically trained deep convolutional networks. In *Intelligent autonomous systems 16: Proceedings of the 16th international conference IAS-16* (pp. 696–712). Springer.

Saviolo, A., Frey, J., Rathod, A., Diehl, M., & Loianno, G. (2022). Active learning of discrete-time dynamics for uncertainty-aware model predictive contross2015primerrol. arXiv preprint arXiv:2210.12583.

Saviolo, A., Li, G., & Loianno, G. (2022). Physics-inspired temporal learning of quadrotor dynamics for accurate model predictive trajectory tracking. *IEEE Robotics and Automation Letters*, *7*(4), 10256–10263.

Schmidt, T., Newcombe, R., & Fox, D. (2016). Self-supervised visual descriptor learning for dense correspondence. *IEEE Robotics and Automation Letters*, *2*(2), 420–427.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

Shen, Y., & Sanghavi, S. (2019). Learning with bad training data via iterative trimmed loss minimization. In *International conference on machine learning* (pp. 5739–5748). PMLR.

Sheng, H., & Yang, C. (2021). PFNN: A penalty-free neural network method for solving a class of second-order boundary-value problems on complex geometries. *Journal of Computational Physics*, *428*, Article 110085.

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, *6*(1), 1–48.

Sima, Q. (1993). *Records of the grand historian: Han dynasty. Vol. 65*. Columbia University Press.

Song, Y., He, L., Zhang, D., Qian, J., & Fu, J. (2019). Neuroadaptive fault-tolerant control of quadrotor UAVs: A more affordable solution. *IEEE Transactions on Neural Networks and Learning Systems*, *30*(7), 1975–1983.

Song, H., Kim, M., Park, D., Shin, Y., & Lee, J.-G. (2022). Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*.

Song, Y., & Scaramuzza, D. (2022). Policy search for model predictive control with application to agile drone flight. *IEEE Transactions on Robotics*, *38*(4), 2114–2130.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929–1958.

Sun, S., Romero, A., Foehn, P., Kaufmann, E., & Scaramuzza, D. (2022). A comparative study of nonlinear MPC and differential-flatness-based control for quadrotor agile flight. *IEEE Transactions on Robotics*, 1–17.

Sundaresan, P., Grannen, J., Thananjeyan, B., Balakrishna, A., Laskey, M., Stone, K., et al. (2020). Learning rope manipulation policies using dense object descriptors trained on synthetic depth data. In *2020 IEEE international conference on robotics and automation* (pp. 9411–9418). IEEE.

Svacha, J., Paulos, J., Loianno, G., & Kumar, V. (2020). IMU-based inertia estimation for a quadrotor using Newton-Euler dynamics. *IEEE Robotics and Automation Letters*, *5*(3), 3861–3867.

Sznaier Camps, G., Dyro, R., Pavone, M., & Schwager, M. (2022). Learning deep SDF maps online for robot navigation and exploration. arXiv e-prints. arXiv–2207.

Tang, J., Ambrus, R., Guizilini, V., Pillai, S., Kim, H., Jensfelt, P., et al. (2021). Self-supervised 3d keypoint learning for ego-motion estimation. In *Conference on robot learning* (pp. 2085–2103). PMLR.

Tanno, R., Saeedi, A., Sankaranarayanan, S., Alexander, D. C., & Silberman, N. (2019). Learning from noisy labels by regularized estimation of annotator confusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11244–11253).

Torrente, G., Kaufmann, E., Fohn, P., & Scaramuzza, D. (2021). Data-driven MPC for quadrotors. *IEEE Robotics and Automation Letters*.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2021). Training data-efficient image transformers & distillation through attention. In *International conference on machine learning* (pp. 10347–10357). PMLR.

Trimpe, S., Millane, A., Doessegger, S., & D'Andrea, R. (2014). A self-tuning LQR approach demonstrated on an inverted pendulum. *IFAC Proceedings Volumes*, *47*(3), 11281–11287.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Vikas, V., Grover, P., & Trimmer, B. (2015). Model-free control framework for multi-limb soft robots. In *2015 IEEE/RSJ international conference on intelligent robots and systems* (pp. 1111–1116). IEEE.

Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3156–3164).

Wang, R., Liu, T., & Tao, D. (2017). Multiclass learning with partially corrupted labels. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(6), 2568–2580.

Wang, H., Sridhar, S., Huang, J., Valentin, J., Song, S., & Guibas, L. J. (2019). Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2642–2651).

Wang, L., Theodorou, E. A., & Egerstedt, M. (2018). Safe learning of quadrotor dynamics using barrier certificates. In *2018 IEEE international conference on robotics and automation* (pp. 2460–2465). IEEE.

Wang, S., Yu, X., & Perdikaris, P. (2022). When and why PINNs fail to train: A neural tangent kernel perspective. *Journal of Computational Physics*, *449*, Article 110768.

Wang, W., Yuan, X., & Zhu, J. (2016). Automatic PID tuning via differential evolution for quadrotor UAVs trajectory tracking. In *2016 IEEE symposium series on computational intelligence* (pp. 1–8). IEEE.

Wolpert, D. H. (2002). The supervised learning no-free-lunch theorems. *Soft Computing and Industry*, 25–42.

Wolpert, D., & Macready, W. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67–82.

Wolpert, D. H., Macready, W. G., et al. (1995). *No free lunch theorems for search*: *Technical Report*, Technical Report SFI-TR-95-02-010, Santa Fe Institute.

Wu, W., Kwasny, S. C., Kalman, B. L., & Engebretson, E. M. (1993). Identifying language from raw speech–an application of recurrent neural networks. In *Proceedings of the 5th artificial intelligence and cognitive science society conference* (pp. 53–57).

Wu, Q., Xu, K., Wang, J., Xu, M., Gong, X., & Manocha, D. (2021). Reinforcement learning-based visual navigation with information-theoretic regularization. *IEEE Robotics and Automation Letters*, *6*(2), 731–738.

Wüest, V., Kumar, V., & Loianno, G. (2019). Online estimation of geometric and inertia parameters for multirotor aerial vehicles. In *2019 International conference on robotics and automation* (pp. 1884–1890). IEEE.

Xiao, T., Xia, T., Yang, Y., Huang, C., & Wang, X. (2015). Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2691–2699).

Xu, W., Gu, H., Qing, Y., Lin, J., & Zhang, F. (2019). Full attitude control of an efficient quadrotor tail-sitter VTOL UAV with flexible modes. In *2019 International conference on unmanned aircraft systems* (pp. 542–550). IEEE.

Yang, G.-Z., J. Nelson, B., Murphy, R. R., Choset, H., Christensen, H., H. Collins, S., et al. (2020). Combating COVID-19—The role of robotics in managing public health and infectious diseases. *Science Robotics*, *5*(40), eabb5589.

Yeh, K., & Kwan, K. (1978). A comparison of numerical integrating algorithms by trapezoidal, Lagrange, and spline approximation. *Journal of Pharmacokinetics and Biopharmaceutics*, *6*, 79–98.

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2021). Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, *64*(3), 107–115.

Zhang, T., Kahn, G., Levine, S., & Abbeel, P. (2016). Learning deep control policies for autonomous aerial vehicles with MPC-guided policy search. In *2016 IEEE international conference on robotics and automation* (pp. 528–535). IEEE.

Zhang, X., Li, X., Wang, K., Lu, Y., et al. (2014). A survey of modelling and identification of quadrotor robot. In *Abstract and applied analysis. Vol. 2014*. Hindawi.

Zhou, K., & Doyle, J. C. (1998). *Essentials of robust control. Vol. 104*. Prentice hall Upper Saddle River, NJ.

Zhou, Y., Xiao, J., Zhou, Y., & Loianno, G. (2022). Multi-robot collaborative perception with graph neural networks. *IEEE Robotics and Automation Letters*, *7*(2), 2289–2296.

Zhu, M., Derpanis, K. G., Yang, Y., Brahmbhatt, S., Zhang, M., Phillips, C., et al. (2014). Single image 3D object detection and pose estimation for grasping. In *2014 IEEE international conference on robotics and automation* (pp. 3936–3943). IEEE.

Zhu, Z., Peng, S., Larsson, V., Xu, W., Bao, H., Cui, Z., et al. (2022). Nice-SLAM: Neural implicit scalable encoding for SLAM. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12786–12796).