POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Physics-informed learning and Data-driven control of an Airborne Wind Energy System

Laurea Magistrale in Automation And Control Engineering - Ingegneria dell'Automazione

Author: Marco Rositani

Advisor: Prof. Lorenzo Mario Fagiano

Academic year: 2023-2024

1. Introduction

The growing problem of climate change claims for technological solutions, capable of satisfying the rising of the global energy demand, avoiding the emissions of greenhouse gases. Among the renewable energy plants, wind turbines are the most diffused. However, a new technology is emerging named Airborne Wind Energy Systems (AWES) [3]. These systems utilize a tethered autonomous aircraft, to convert wind energy into electricity. The fundamental advantage over wind turbines is that AWES do not require the tower and foundations, which are substituted by the tether, resulting into a more economical solution. In particular this thesis deals with the modeling and control of a rigid wing AWE system, relying on data-driven techniques. The prototype KM1 is taken into consideration, produced by Kitemill, for which test flight data and aerodynamic data related to a previous version, named KM0, are available. Thus, the thesis starts by presenting the dynamical model of the KM1 prototype. Then it introduces a model identification technique for generic nonlinear dynamical systems, based on the simulation error method (SEM). The latter is solved by exploiting a Sequential Quadratic Programming (SQP) approach. Then, it provides the applied techniques for identifying the flight controller, based on Gaussian basis functions. The derived controller is used to set up a closed-loop identification algorithm for the identification of the KM1 prototype parameters. Then, motivated by the limitations of models derived from physical laws, a hybrid modeling technique is introduced, that combines data-driven approaches and physics-based models. In particular, the main challenge of the thesis consists in developing a Physics-Informed Neural Ordinary Differential Equation (ODE) and defining a suitable optimization program and training algorithm for this architecture.

2. System Modeling

To derive the kite's equation of motions define the inertial coordinate system and the body frame. The unit vector i^i is directed north, j^i is directed east, and k^i is directed toward the center of the earth. The body frame is the frame fixed with the kite, defined with the x-axis pointing forward along the fuselage, y-axis going through the right wing, and the z-axis pointing down, while the origin is located at the center of mass of the kite. The dynamical equation of the kite are represented by the following equations:

$$\begin{pmatrix}\dot{p}_n\\ \dot{p}_e\\ \dot{p}_d\end{pmatrix} = \begin{pmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{pmatrix} \begin{pmatrix} u\\ v\\ w\end{pmatrix},$$

$$\begin{pmatrix}\dot{u}\\ \dot{v}\\ \dot{w}\end{pmatrix} = \begin{pmatrix} rv - qw\\ pw - ru\\ qu - pv\end{pmatrix} + \frac{1}{m}\begin{pmatrix} f_x\\ f_y\\ f_z\end{pmatrix},$$

$$\begin{pmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{pmatrix} = \begin{pmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{pmatrix} \begin{pmatrix} p\\ q\\ r\end{pmatrix},$$

$$\begin{pmatrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{pmatrix} = \begin{pmatrix} \Gamma_1 pq - \Gamma_2 qr \\ \Gamma_5 pr - \Gamma_6 (p^2 - r^2) \\ \Gamma_7 pq - \Gamma_1 qr \end{pmatrix} + \begin{pmatrix} \Gamma_3 m_x + \Gamma_4 m_z \\ \frac{1}{J_y} m_y \\ \Gamma_4 m_x + \Gamma_8 m_z \end{pmatrix},$$

In the previous equations (p_n, p_e, p_d) are the inertial positions of the kite, defined along i^i, j^i and k^i respectively. (u, v, w) are the body frame velocities measured along i^b, j^b and k^b respectively. (ϕ, θ, ψ) are the Euler angles. (p, q, r) are the angular rates, defined along i^b, j^b and k^b respectively. Therefore the states of the system can be defined as:

$$z = [p_n, p_e, p_d, u, v, w, \phi, \theta, \psi, p, q, r]^\top$$

The vectors (f_x, f_y, f_z) and (m_x, m_y, m_z) are the body frame forces and moments acting on the kite, respectively. Moreover m is the mass of the kite and the Γ coefficients are combinations of moments of inertia, completely defined in [1].

2.1. Forces and moments

The forces and moments acting on the kite are mainly due to, gravity, aerodynamics and the tether. Then, the total force acting on the airframe and the total moment, respectively are:

$$\begin{aligned} \mathbf{f}^b &= \mathbf{f}_g^b + \mathbf{f}_a^b + \mathbf{f}_t^b, \\ \mathbf{m}^b &= \mathbf{m}_a + \mathbf{m}_t \end{aligned} \tag{1}$$

The gravitational force in the body frame can be expressed as:

$$\mathbf{f}_g^b = \begin{pmatrix} -mg\sin\theta \\ mg\cos\theta\sin\phi \\ mg\cos\theta\cos\phi \end{pmatrix}$$

The aerodynamic forces and moments depends on the control surfaces of the aircraft (aileron δ_a, elevator δ_e, rudder δ_r and flaps δ_f), the angle of attack α, sideslip angle β and the airspeed V_a. These quantities are fully described in [1]. Accordingly, $\mathbf{f}_a^b$ is defined as:

$$\mathbf{f}_a^b = \begin{pmatrix} \cos\beta\cos\alpha & -\sin\beta\cos\alpha & -\sin\alpha \\ \sin\beta & \cos\beta & 0 \\ \cos\beta\sin\alpha & -\sin\beta\sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} -F_{\text{drag}} \\ F_y \\ -F_{\text{lift}} \end{pmatrix},$$

, where F_{drag}, F_y and F_{lift} are determined by the aerodynamic coefficients C_D, C_Y and C_L, respectively. These coefficients, taken from [4], are polynomial functions of α, β, Va and the control surface deflections. Similarly, the moment components of $m_a = (L, M, N)^\top$, depend on the coefficients C_l, C_m and C_n also given in [4] and expressed as polynomials of α, β, Va and the control surface deflections.

In order to compute the tether force, a kinematic constraint between the norm of the kite's position $\mathbf{p_k}$ and the tether length l_t, plus an offset Δ_l, is enforced. The net force along the tether direction is captured by:

$$\bar{F}_1 = \frac{\mathbf{p_k}^\top}{\|\mathbf{p_k}\|} \mathbf{R_v^b}^\top \left(\mathbf{f}_a^b + \mathbf{f}_g^b - m \begin{pmatrix} rv - qw \\ pw - ru \\ qu - pv \end{pmatrix}\right) \tag{2}$$

where $\mathbf{R}_v^b$ is a rotation matrix applied to go from the body frame to the inertial frame. Then, define a virtual force

$$\bar{F}_2 = k_v \big(l_t + \Delta_l - \|\mathbf{p_k}\|\big) \tag{3}$$

Their sum, projected along $\mathbf{p_k}$ and rotated into the body frame, gives the tether force

$$\bar{F}_t^b = \mathbf{R}_v^b \big(\bar{F}_1 + \bar{F}_2\big) \frac{\mathbf{p_k}}{\|\mathbf{p_k}\|}. \tag{4}$$

To account for aerodynamic drag on the tether, one approximates

$$F_{tD} = \frac{1}{8} \rho \, d_t \, C_{tD} \, l_t \, \mathbf{V}_a^2, \tag{5}$$

causing the tether to form a catenary rather than a straight line. The resulting angle is

$$\theta_t = \arcsin\!\Big(\tfrac{-F_{tD}}{\bar{F}_1 + \bar{F}_2}\Big), \tag{6}$$

and the tether force direction in the body frame is adjusted by

$$\mathbf{f}_t^b = \begin{pmatrix} \cos\theta_t & 0 & \sin\theta_t \\ 0 & 1 & 0 \\ -\sin\theta_t & 0 & \cos\theta_t \end{pmatrix} \bar{F}_t^b \tag{7}$$

Because the tether attaches at $\mathbf{r}_{t/CG}$ rather than the kite's center of gravity, the tether force also induces a moment, expressed as:

$$\mathbf{m}_t = \mathbf{r}_{t/CG} \times \mathbf{f}_t^b \tag{8}$$

2.2. Control System

The flight controller C_{δ_r} employed by the system uses only the rudder deflections to actively control the kite, while the others control surfaces are set to constant values.

The ground station oversees the reeling in and reeling out of the tether during flight maneuvers and it is idealized as a system capable of perfectly tracking a given tether length reference l_{tref}.

3. Dataset

The dataset consists of time series measurements collected during flight test operations. It includes state variable measurements, control surface deflections, wind speed values at different heights at the test site, tether load, and tether length. By inspecting the dataset, it is possible to notice that the values for the elevator, aileron, and flaps remain constant, while only the rudder is actively utilized to control the kite. Consequently, the fixed control surface deflections are directly input into the model based on the available measurements, while the tether length serves as a reference for the ground-station subsystem.

4. Physics-based model identification

4.1. Problem abstraction

The identification problem can be addressed in an optimization setting, by means of the following program:

$$\min_{\gamma} \quad (\hat{Y}-\tilde{Y})^\top Q(\hat{Y}-\tilde{Y}) \tag{9a}$$

subject to:

$$\hat{z}(i+1|i) = f_z(\hat{z}(i), \tilde{u}(i), \hat{u}(i), \gamma) \tag{9b}$$

$$\hat{y}(i) = g_z(\hat{z}(i), \tilde{u}(i), \hat{u}(i), \gamma) \tag{9c}$$

$$\hat{z}(0) = \hat{z}_0 \tag{9d}$$

$$\hat{u}(i) = K(\hat{z}(i)) \tag{9e}$$

$$h_\gamma(\gamma) \geq 0 \tag{9f}$$

The optimization variable γ consists of the model's parameters to be estimated. The vectors $\tilde{Y}$ and $\hat{Y}$ contain the measured and simulated outputs, respectively, while $\tilde{u}(i)$ represents the measured inputs used to obtain $\tilde{Y}$, and $\hat{u}(i)$ are the control inputs computed by the same controller K employed during experimental data collection. The cost function f quantifies the discrepancy between $\tilde{Y}$ and $\hat{Y}$, using a weight matrix Q to normalize output magnitudes. The first four constraints (9b)-(9e), which define the system and controller dynamics from a known initial state, are inherently embedded in the simulation, thus there's no need to add them explicitly in the optimization. Additionally, h_γ represents a set of linear constraints imposing prior bounds on the parameters.This optimization problem is efficiently solved using an SQP approach with a constrained Gauss-Newton algorithm. For further information on numerical optimization, refer to [2].

4.2. Controller identification

The flight controller C_{δ_r} must be identified directly from experimental data, as its structure is unknown. This ensures that the controller used during the identification process closely replicates the behavior of the one employed during flight tests. The resulting controller takes the states z as input and computes the desired rudder deflection δ_r. To address this problem it is proposed to identify the control law with Gaussian basis functions. This approach is entirely based on convex optimization. In particular the controller is approximated as:

$$C_{\delta_r}(z) = \sum_{i=1}^{M} \hat{a}_i \varphi_i(z) \tag{10}$$

where z is the vector of feedback variables, φ_i are the Gaussian basis functions. The coefficients $\hat{a}_i$ are identified by means of the algorithms described in [5].

4.3. KM1 kite identification

The proposed optimization program for the identification of the KM1 kite is:

$$\min_{\gamma} \quad (\hat{Y}-\tilde{Y})^\top \tilde{Q}(\hat{Y}-\tilde{Y}) \tag{11a}$$

subject to:

$$\hat{z}(i+1|i) = f_z\big(\hat{z}(i), \tilde{u}(i), \hat{\delta_r}(i), \gamma\big) \tag{11b}$$

$$\hat{z}(i_k) = \hat{z}_0^{(k)} \tag{11c}$$

$$\hat{\delta_r}(i) = C_{\delta_r}\big(\hat{z}(i)\big) \tag{11d}$$

$$h_\gamma(\gamma) \geq 0 \tag{11e}$$

This optimization program employs a Simulation Error Method (SEM), where equations

(11a)-(11e) are numerically integrated over the time interval $[0, N_s T_s]$, during which N_r reinitializations of the state occur. The cost function (11a) quantifies the mismatch between measured and predicted states by focusing on positional, orientational, rudder deflection and tether force errors. Specifically, linear and angular velocities were not included, to prevent the cost function from becoming overly complex, which could degrade the performance of the optimization procedure. The vector γ contains the parameters of the kite model, including the inertia matrix, aerodynamic coefficients, tether drag coefficient, and the tether attachment point. The vector $\tilde{u}$ contains exogenous inputs such as the constant control surface deflection, tether length reference and wind speed v_w. The constraints $h_\gamma(\gamma)$ are derived from the parameter physical meaning and the KM0 prototype simulator.

5. Hybrid modeling

Motivated by the limitations of models derived from physical laws, a hybrid modeling technique is introduced, that combines data-driven approaches and physics-based models. Specifically, the proposed architecture is based on the novel concept of Neural-ODEs [6], which combines the power of machine learning with the formalism of differential equations. Specifically, the hybrid model is composed of several parts. The function f_k represents the physics-based kinematics derivatives of (p_n, p_e, p_d) and (ϕ, θ, ψ). Similarly, f_{dy} describes the physics-based kinetics derivatives of (u, v, w) and (p, q, r). In addition, f_{nn} represents the Neural ODE, which is parametrized as a feed-forward neural network and is responsible for computing the compensation for the kinetics derivatives. Finally, g_f takes as input the kinetics derivatives computed by f_d and f_{nn} and produces the final derivatives of linear and angular velocities, where f_{nn} applies an additive correction to the outputs of f_d. Essentially, f_k and f_d constitute the physics-based model identified in section 4.3, therefore those equations are completely defined. The function g_f is fully known, since just an additive correction is considered. The only unknown component is f_{nn}, characterized by the parameter set θ_{nn}, which is identified by means of the following optimization program:

$$\min_{\theta_{nn}} \quad (\hat{Z} - \tilde{Z})^\top \tilde{Q} (\hat{Z} - \tilde{Z})$$

subject to:

$$\begin{aligned}
&\hat{z}_k(i+1|i) = f_k(\hat{z}(i)) \\
&\hat{z}_{dy}(i+1|i) = f_{dy}(\hat{z}(i), \tilde{u}(i), \hat{\delta}_r(i)) \\
&\hat{z}_{nn}(i+1|i) = f_{nn}(\hat{z}(i), u_{nn}(i), \hat{\delta}_r(i), \theta_{nn}) \\
&\hat{z}_f(i+1|i) = g_z(f_d, f_{nn}) \\
&\hat{z}(i_k) = \hat{z}_0^{(k)} \\
&\hat{\delta}_r(i) = C_{\delta_r}(\hat{z}(i))
\end{aligned}$$

Where z_k contains the kinematic variables, z_f contains the linear and angular velocities, z_{dy} and z_{nn} contain their predictions from the physics-based model and the Neural ODE, respectively, while u_{nn} contains $\tilde{u}$, α, β and V_a. This optimization program, similar to the one discussed in section 4.3, employs a SEM, however its cost function includes the entire system state. This approach enhances the model's predictive accuracy by using a neural network to refine velocity-dependent dynamics that the physics-based model under represents. The neural network exclusively adjusts the time derivatives of linear and angular velocities, as position and orientation derivatives are purely kinematic. A schematic overview of the physics-informed architecture is shown in figure 1.

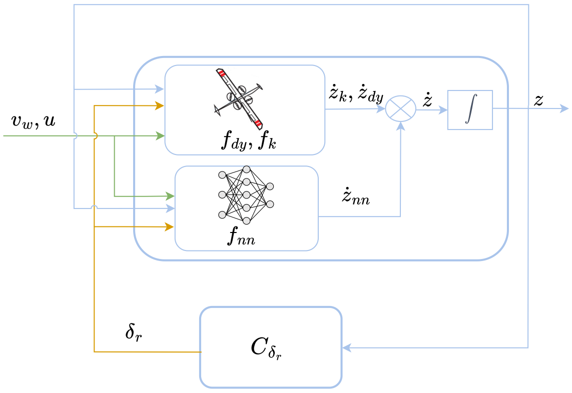

Figure 1: Hybrid model scheme: f_k and f_{dy} represent the physics-based model, f_{nn} is the Neural ODE. The variable u represents a simplified notation for inputs and it contains δ_e, δ_a, δ_f, l_t.

5.1. Training Neural ODEs

Training the Neural ODE, requires to backpropagate the gradient of the loss function , with respect the model parameters θ_{nn}, through the differential equation to be solved. In this study, the the discretize-then-optimize method is employed, which backpropagates through the internal operations of the ODE solver. This approach computes gradients directly on the discretized version of the differential equation, making them highly accurate and time-efficient. However, it requires storing each internal operation of the ODE solver.

Then, a combination of first-order and second-order optimizers is employed for training. Typically, first-order optimizers are favored for training deep neural networks due to the large parameter space. However, in this case, a physics-informed Neural ODE is used, reducing the need for a very large network. This allows to exploit even second-order optimization algorithms, such as Low memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS).

Specifically the training process is structured as follows: the dataset is divided into three subsets: training, validation, and testing. During the early stages of training, the first-order optimizer Adaptive Moment Estimation (ADAM) is utilized to ensure rapid progress in reducing the training loss. However, as the training progresses and the validation loss reaches a plateau, the optimizer switches to L-BFGS. This strategy helps to fine-tune the model effectively.

6. Results

In order to quantitatively evaluate the identification results it is useful to introduce a performance index, such as the NRMSE, defined as

$$NRMSE = \frac{\sqrt{\frac{\sum_{i=1}^{N_s}(\tilde{y}(i)-\hat{y}(i))^2}{N_s}}}{\tilde{y}_{\max} - \tilde{y}_{\min}}$$

6.1. Controller identification results

The derived Gaussian controller was able to stabilize the kite's trajectory simulated with the initial parameters, achieving an NRMSE value of 0.0181, computed by considering as inputs the measured states $\tilde{z}$.

6.2. Physics-based model results

This section demonstrates that the physical model identification algorithm effectively enhances the predictive capabilities of the kite model across all states. Figure 2 highlights the differences in performance for attitude predictions before and after the identification process.

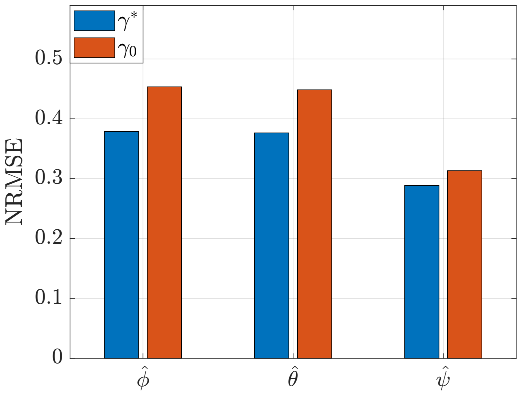

Figure 2: NRMSE of the Euler angles: physical model with identified parameters (blue), and the physical model with initial parameters (orange).

6.3. Hybrid modeling results

The effectiveness of the training technique outlined in Section 5.1 is demonstrated by comparing two different approaches: training with the ADAM optimizer alone and training with a combination of ADAM and L-BFGS. As shown in Figure 3, the model trained exclusively with ADAM exhibits lower performance, for example in the inertial position predictions. In conclusion, the hybrid model significantly improves prediction accuracy for linear and angular velocities compared to the physical model. Figure 4 shows improvements in linear velocity predictions, while figure 5 presents the simulated kite trajectory with the hybrid model.

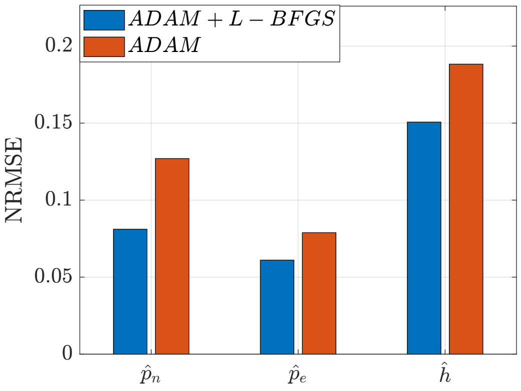

Figure 3: NRMSE of the inertial positions: hybrid model identified with ADAM (orange) and with ADAM and L-BFGS (blue).

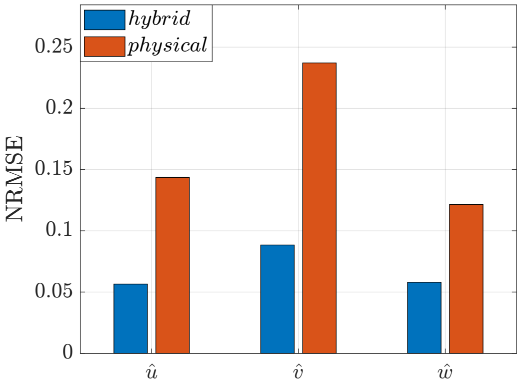

Figure 4: NRMSE of the linear velocities: hybrid model learned with ADAM and L-BFGS (blue), and the physical model with the identified parameters (orange).

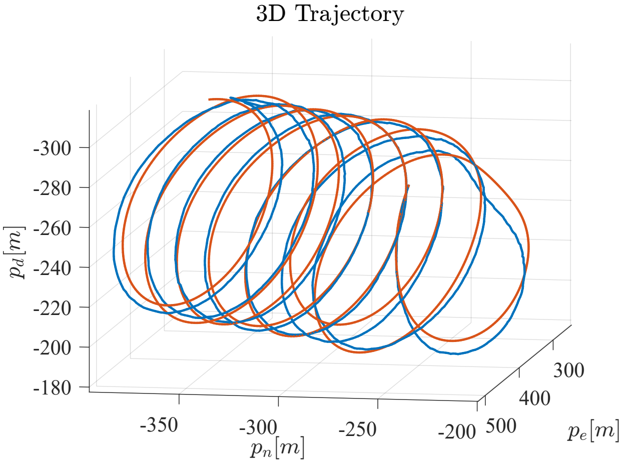

Figure 5: 3D kite trajectory obtained with the hybrid model from a multi-step prediction with the Gaussian controller: measured data (blue) vs. simulation results (orange).

7. Conclusions

The aim of this thesis was the modeling and control of an AWE system, exploiting available data. The identification of the flight controller exploiting two different classes of universal approximators was successfully implemented. Subsequently, the identified controller were employed to set up a closed loop optimization program for identifying the physical parameters of the KM1 prototype. The identification process proved satisfying results, improving the prediction accuracy compared to the initial parameter estimates. A hybrid modeling technique was introduced, based on the concept of Neural-ODEs, along with its training procedure and optimization strategies. A specific optimization program was defined for training this architecture, so as to overcome the limitations of the physics-based model. The hybrid model demonstrated enhanced predictive performance across all states, particularly in velocity estimations. Moreover, the training process of such architecture was described in depth. Specifically the combination of first order and second order optimizers was explored, for the training process, and showed excellent results in overcoming the limits of the ADAM optimizer. In conclusion, the obtained hybrid model was extensively validated using experimental flight data. The most straightforward extension of this work could be the study of the retraction phase of the AWE system. This could include the application of hybrid techniques to enhance accuracy and performance. Then a switched system can be obtained that models separately the traction and retraction phases. Furthermore it could be interesting to integrate the hybrid model into a model predictive control framework, where for example the rudder predictions could be computed by solving an optimization program which maximizes the power generation while guarantying the safety of the system.

References

[1] R. W. Beard and T. W. McLain. *Small Unmanned Aircraft: Theory and Practice*. Princeton Univ. Press, 2012.

[2] M. Diehl. Lecture notes on numerical optimization (preliminary draft), 2017.

[3] L. Fagiano et al. Autonomous airborne wind energy systems: Accomplishments and challenges. *Annu. Rev. Control, Robot. Auton. Syst.*, 5:603–631, 2022.

[4] T. Mohammed et al. Large-scale reverse pumping for rigid-wing airborne wind energy systems. *J. Guid. Control Dyn.*, 47(8), 2024.

[5] L. Fagiano and C. Novara. Learning a nonlinear controller from data: Theory, computation, and experimental results. *IEEE Trans. Autom. Control*, 61(7):1854–1868, 2016.

[6] Patrick Kidger. On neural differential equations. *arXiv*, 2022.