Data-Driven System Identification of an Aeropendulum

Arthur Dimitri Brito Oliveira, Antonio Marcus Nogueira Lima
and Rafael Bezerra Correia Lima

October 23, 2023

Data-Driven System Identification of an Aeropendulum [★]

Arthur D.B. Oliveira [*] **Rafael B.C. Lima** [**] **Antonio M.N. Lima** [**]

[*] *Graduate Program in Electrical Engineering - PPgEE*
Center for Electrical Engineering and Informatics, Federal University
of Campina Grande, PB (e-mail: arthur.dimitri@ee.ufcg.edu.br).
[**] *Department of Electrical Engineering, Center for Electrical*
Engineering and Informatics, Federal University of Campina Grande,
PB (e-mail: amnlima,rafael@dee.ufcg.edu.br)

Abstract: In this work, we propose a refinement of the aeropendulum platform model, specifically focusing on the modeling of thrust force. We have derived a mathematical model, proposed acquisition schemes for data collection, and conducted experiments to describe the previously overlooked dynamics of the actuator. We have developed a simulation model that comprehensively captures the system's electrical, mechanical, and aerodynamic behaviors by utilizing electrical and mechanical information. The results obtained from the simulation data demonstrate the effectiveness of the WSINDy algorithm in accurately determining the thrust coefficient and predicting the dynamics of other variables during the testing phase.

Keywords: Aeropendulum, Identification of Nonlinear Dynamics, System Identification, SINDy, WSINDy

1. INTRODUCTION

System identification aims at determining the mathematical model of a system based on observed input-output data. What is known as the classic system identification is a mature domain with more than half a century of existence Papoulis (1968); Ljing (1999). The so-called data-driven approach, in a nutshell, aims at reducing the amount of a priori knowledge regarding the underlying system model and maximizing the use of statistics at the core of system identification workflow. With minimal a priori, one has to infer from data the system dynamics, and quite important progress has been made in this direction recently Brunton and Kutz (2022); Brunton et al. (2016).

The SINDy algorithm has been used to infer dynamic models from data. Bastos (2021) used the sparse identification algorithm to derive a model for a robotic application. Fangzheng Sun (2021) used the method to predict the dynamics of a drive system for prismatic joints of robots. What is common to these applications is that the use of the SINDy algorithm leads to unmodeled dynamics and a lack of physical explainability, mainly due to the necessity of computing approximate derivatives.

Here, we apply data-driven system identification to a relatively simple nonlinear aeropendulum system. An aeropendulum is a nonlinear dynamical system, quite simple to design and build, that is commonly used in graduate and undergraduate engineering courses for teaching feedback control systems Enikov and Campa (2012); Silva et al. (2020). Although the propulsion force that drives the aeropendulum is provided by an electric motor, few works include models of the electrical part in the control problem formulation. An attempt to apply the so-called classical system identification and include the electrical parts' model into the control problem formulation has been presented in Lucena et al. (2021). However, in that attempt, the experimental test results obtained with a controller synthesized based on the identified model revealed that the closed loop was stable, but the transient behavior did not fully match the prescribed performance template OLIVEIRA (2023).

In this paper, we revisit Lucena et al. (2021) to diagnose and fix the mismatch between the experimental transient behavior and the prescribed performance template used in the controller design phase. However, in the present case, we are reconsidering some of the hypotheses adopted for the electrical subsystem and its coupling to the mechanical subsystem, whereas exploiting the data-driven system identification proposed in Messenger and Bortz (2021) to determine the thrust coefficient of the propeller.

2. THE AEROPENDULUM PLATFORM

This section focuses on describing the laboratory prototype and its nonlinear dynamic model.

2.1 Physical representation

Figure 3 shows the free-body diagram of the aeropendulum. The rotating rod with length L connects the bearing axis to the DC motor. The motor is connected to the drive gear, whilst the driven gear is attached to a propeller. A thrust force T opposes the force weight component $mg\sin(y)$. This thrust is proportional to the speed of

[★] This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

the driven gear. It opposes the force weight component $mg\sin(y)$. The torque TL leads to the angular displacement concerning the vertical axis.

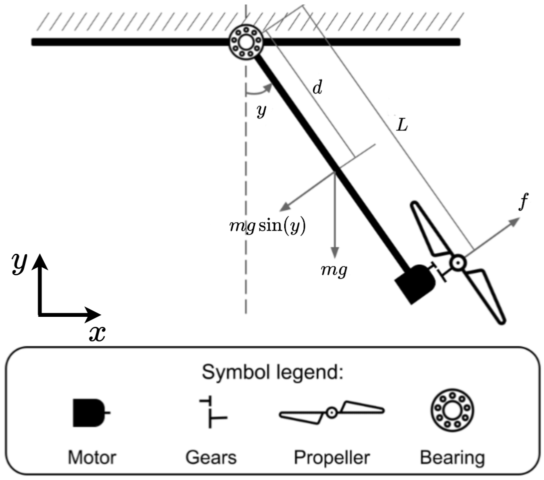


Figure 1. Free-body diagram of the aeropendulum. The basic set of components is the motor, the gearbox, the propeller, and the rotating bearing. Source: adapted from Lucena et al. (2021)

2.2 Electrical and aerodynamic parts

The aeropendulum comprises four parts, as depicted in Figure 2. A PWM signal controls the motor shield's H-bridge and delivers voltage to the motor armature.

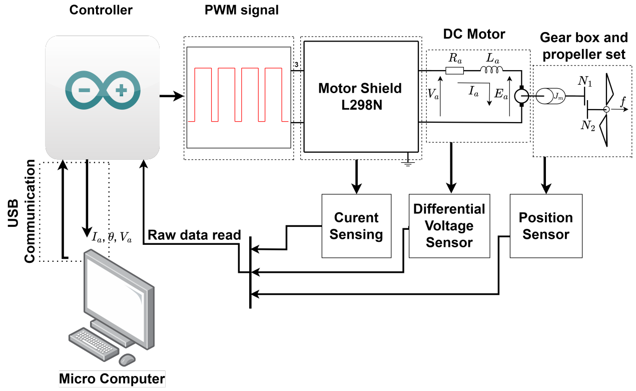


Figure 2. Aeropendulum system diagram: electrical, mechanical, and aerodynamic components.

Figure 3 shows photographs of the actual prototype. Supported by a base made of polystyrene, under the bearing, there is a rotary magnetic encoder AS5040 that is used to measure the angular position. The microcontroller is used to collect, process, and generate the necessary signals, whilst the motor driver is used to directly control the DC motor. The motor-propeller set is seen on the left side of the figure. A USB connection interface transfers commands and data to the microcomputer.

2.3 Mathematical description

From the free-body diagram and the equivalent circuit depicted in Figure 3 and 2, respectively, one can derive

$$\frac{di_a}{dt} = -\frac{R_a}{L_a}i_a - \frac{K_\omega}{L_a}\omega_1 + \frac{1}{L_a}v_a, \tag{1}$$

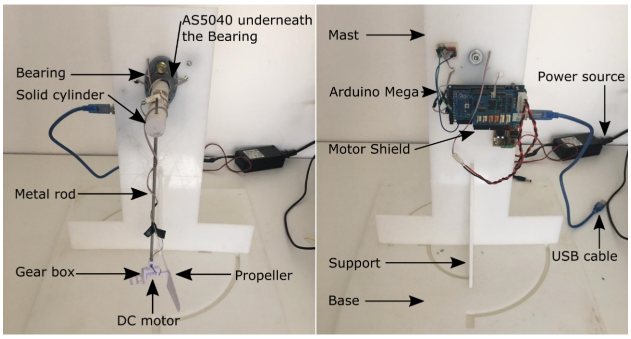


Figure 3. Front and backside photographs of the real aeropendulum plant. Source: Lucena et al. (2021)

where R_a represents the armature resistance, L_a the armature inductance, K_ω the electromotive constant, and v_a the voltage source. We can also describe

$$\frac{d\omega_1}{dt} = \frac{K_i}{J_m}i_a - \frac{F}{J_m}\omega_1 - \frac{1}{J_m}T_1, \tag{2}$$

where K_i represents the torque constant, ω_1 the rotational speed of the motor shaft, T_1 is the mechanical torque due to the gear-propeller set, J_m the moment of inertia of the rotor, and F the viscous friction constant. Since the propeller thrust is given by

$$f = K_q\omega_2^2, \tag{3}$$

where ω_2 is the speed of the propeller, and K_q is the thrust coefficient (Corke, 2017). Since the relationship of ω_1 and ω_2 is expressed by

$$\omega_2 = \frac{N_2}{N_1}\omega_1, \tag{4}$$

where $N_2 : N_1$ is the gearbox ratio.

Considering that the armature current can be measured and that the motor dynamics are fast enough to be neglected regarding the aeropendulum dynamics, the angular speed ω_1 can be rewritten in terms of the input voltage and armature current $\omega_1 = \frac{1}{K_\omega}v_a - \frac{R_a}{K_\omega}i_a$. Taking into consideration that the propeller's transient response is assumed to be faster than the speed of the pendulum rod and neglecting the moment of inertia of the propellers, (3) can be rewritten as

$$\begin{aligned} f = K_q\left(\frac{N_1}{N_2K_\omega}\right)^2 v_a^2 - 2R_aK_q\left(\frac{N_1}{N_2K_\omega}\right)^2 v_ai_a + \\ K_q\left(\frac{N_1}{N_2}\frac{R_a}{K_\omega}\right)^2 i_a^2. \end{aligned} \tag{5}$$

One can describe the pendulum's rotational dynamics from the free-body diagram as

$$\frac{d\Omega}{dt} = -\frac{mgd}{J}\sin(y) - \frac{C}{J}\Omega + \frac{L}{J}f \tag{6}$$

$$\frac{dy}{dt} = \Omega, \tag{7}$$

where y is the angular position, Ω is the angular speed of the aeropendulum.

3. EXPERIMENTAL RESULTS

The previous study Lucena et al. (2021) focused on acquiring position information only and used both the 10-bit Synchronous Serial Interface and a simulation model to communicate with the Arduino microcontroller. It hindered the possibility of increasing the sampling rate and constrained the low-level signal configuration.

Developing an acquisition scheme running solely on the Arduino board would yield two significant benefits. First, it would grant greater control over the sampling rate, enabling the utilization of different acquisition modes, such as the PWM built-in interface in the position sensor. Second, it would ease low-level hardware configuration, empowering the triggering of events based on Arduino Timers, thereby ensuring regular sampling intervals. Moreover, interrupt-based events would allow for sampling diverse signals due to hardware phase correction.

With these considerations in mind, this section proposes using a timing diagram to collect the effective armature voltage, armature current, and angular position using solely the Arduino board.

3.1 Timing diagram

By configuring the Wave Generation Mode bits of the Timer 1 Counter Control Register, the Arduino board can be set to operate in the phase-corrected PWM mode. This configuration enables an oscillator-based counter to function in a dual-slope manner. Leveraging this capability, the Timer1 Overflow interruption takes place whenever the PWM signal is operating at high level. The timing diagram illustrating the proposed acquisition scheme is presented in Figure 4.

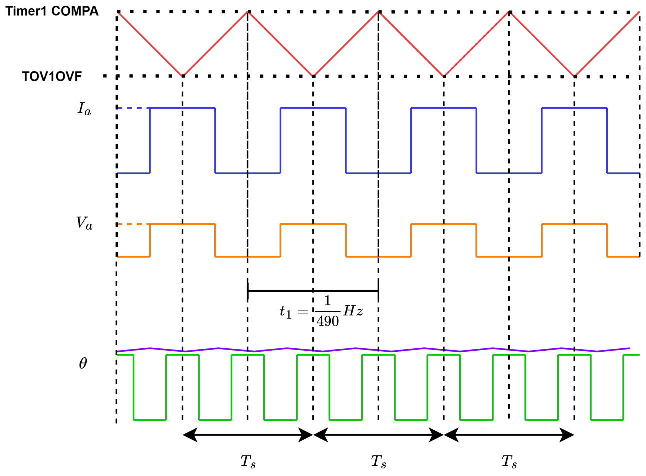


Figure 4. Data acquisition timing diagram: the signals of interest are measured in their high level, whose sampling depends on the triggering of the overflow flag from Timer 1. The sampling rate is directly related to the PWM frequency of the Arduino Board (originally 490 R1.1Hz).

Whenever the counter of Arduino's timer one reaches zero, an overflow interruption called TOV1OVF is triggered. During this interruption, the armature current, voltage data, and filtered position information are gathered. The sampling rate is directly influenced by the PWM frequency of the Arduino board, resulting in a sampling time of $T_s = \frac{1}{490} = 2,04$ ms. A set of voltage values is applied to carry out the data acquisition process.

3.2 Position data acquisition

The AS5040 sensor has a PWM interface that generates a PWM signal. The duty cycle of this signal is proportional to the angular position detected, while its frequency is 975,6 Hz. To process the position data, a second-order low-pass filter is employed. This filter effectively smooths the signal, and the Arduino's resulting analog output voltage, which is proportional to the angle, is read by the ADC (Analog-to-Digital Converter). Figure 5 illustrates the PWM signal generation, filtering, and conversion process. To achieve the desired results, $R_1 \geq 4K7\Omega$, $C_1 \geq 1\mu F$.

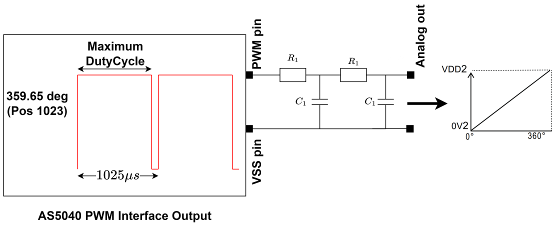


Figure 5. The PWM output provided by the AS5040 and the filtered to yield the angular displacement. $R_1 = R_2$, $C_1 = C_2$. Source: adapted from the AS5050 sensor's datasheet.

3.3 The relationship between the duty cycle and the armature voltage

Accurate measurement of the armature voltage is crucial as there are voltage drops across the H-bridge utilized for DC motor control, and these drops vary non-linearly with the armature current. However, during the experiments, it was observed that the differential measurement feature supported by the Mega microcontroller yielded imprecise results. Consequently, a curve-fitting relationship between the duty cycle to the H-bridge and the corresponding armature voltage value was established. This relationship is illustrated in Figure 6.

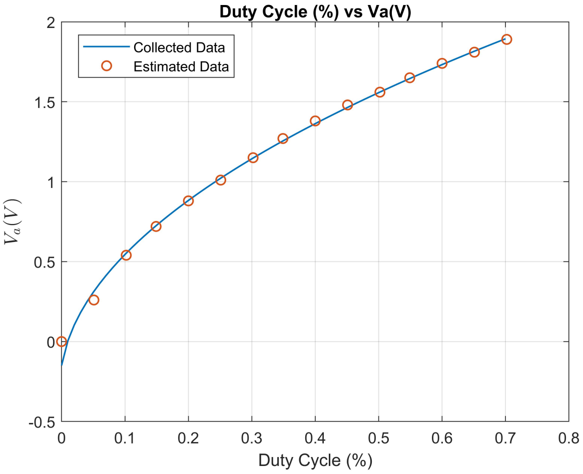


Figure 6. The armature voltage V_a as a function of the duty cycle τ; acquired and estimated data through (8).

The duty cycle τ was incrementally increased at a rate of 5% and capped at 70%. This limit ensures the aeropendulum rod does not exceed its maximum allowed displacement of 90 degrees. Through the utilization of a curve-fitting algorithm, the relationship between τ and the resulting V_a is expressed by:

$$V_a = k_0 + k_1\sqrt{k_2\tau + k_3} \tag{8}$$

where $k_0 = -0,2914$, $k_1 = 6,8\text{e-}4$, $k_2 = 1,469\text{e}7, k_3 = 4,3629\text{e}4$. This equation makes it feasible to determine the voltage applied to the DC motor based on the τ applied to the H-bridge.

3.4 Armature current data collection

The Arduino Rev3 motor shield incorporates a built-in feature for reading the armature current value. This functionality uses the *SENA* pin on the chip, which is connected to a $0,15\ \Omega$ resistor (R_1). The voltage across R_1 is amplified through an operational amplifier (op-amp) and can be directly measured using the analog pin *A0*. The analog pin provides a voltage proportional to the measured current, and the calibration value is 3,3 V when it delivers its maximum current of 2 A.

3.5 Acquiring the data

After uploading the programming routine to the board, we can write a set of PWM values to the DC motor controller and gather the open-loop information following the proposed acquisition scheme. The actual data collected during this process is illustrated in Figure 7.

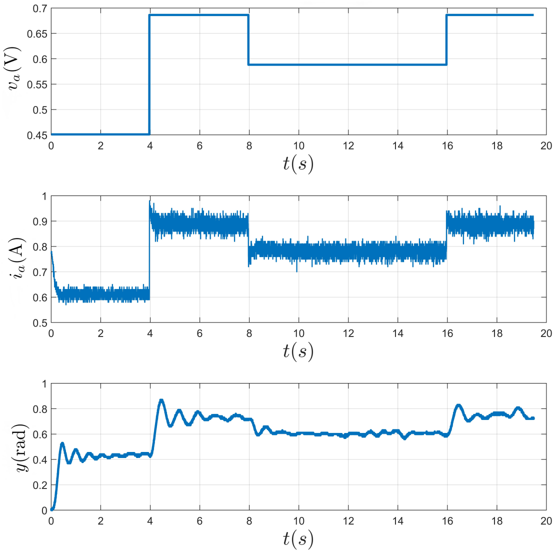


Figure 7. Open loop acquired signals.

3.6 Motor parameters

To determine R_a, we performed the so-called locked-rotor test. We calculated the armature resistance by using $R_a = \frac{V_a}{I_a}$ (steady state). By varying V_a, and measuring I_a, we found out that $\hat{R}_a = 1,08\ \Omega$ (mean value) and $\sigma_{\hat{R}_a} = 0,04\ \Omega$ (standard deviation). To determine K_ω, we applied different V_a values and measured the steady state values of ω_2 by using a stroboscope and I_a by using an ampere meter. We estimated K_ω by using

$$\hat{K}_\omega = \frac{V_a - \hat{R}_a I_a}{\frac{N_1}{N_2}\omega_2}, \tag{9}$$

where the mean value was $\hat{K}_\omega = 6,606\text{e-}4$ Vs/rad with a standard deviation of $\sigma_{\hat{K}_\omega} = 1,38\text{e-}5$ Vs/rad.

For the estimation of L_a, we applied a voltage step (V_a) and recorded I_a (see Figure 8). From the time it took for I_a to reach 63% of its steady state value, the electrical time $\tau_e = \frac{L}{R}$ was found $\hat{\tau}_e = 0,002$ s. Given the value of $\hat{R}_a$, we estimated $\hat{L}_a = 2,2$ mH.

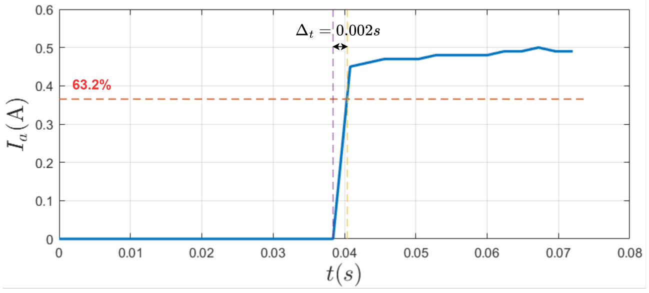


Figure 8. Armature current (I_a) time response for a duty cycle $\tau = 40\%$ in the armature voltage (V_a).

4. SIMULATION MODEL

While the retrieval of the propeller speed is still ongoing, we use the previously identified electrical and mechanical parameters to construct a model that accurately represents the real system. The remaining parameters, such as the aerodynamic ones, can be determined in an arbitrary manner. By implementing this representation of the aeropendulum platform in a simulated environment, we can gather the required data and subsequently apply the SINDy algorithm to model the unknown components.

The parameters of the model are:

Table 1. The parameters used in the simulation model.

	Model Parameters
Electrical	$R_a = 1,08\Omega,\ \ L_a = 2,2mH$
	$K_\omega = 6,6\text{e-}4Vs/rad,\ \ K_t = 0,235Nm/A$
Mechanical and Aerodynamic	$J_m = 5,2\text{e-}5Kgm^2,\ \ B_m = 0,01Ns/m^2$
	$\frac{c}{J} = 13,82,\ \ \frac{mgd}{J} = 131,75,\ \ \frac{L}{J} = 462,83$
	$N_1/N_2 = 55/9,\ \ K_q = 1\text{e-}4$

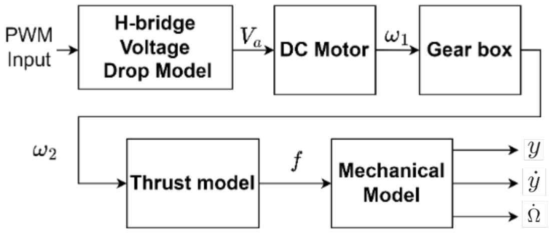


Figure 9. Simulation model.

5. DATA-DRIVEN SYSTEM IDENTIFICATION

One of the most promising methods for extracting dynamical equations from data with a minimal amount of a priori information is the so-called SINDy (Brunton et al., 2016). In this paper, we used the weak formulation of SINDy, denoted as WSINDy (Messenger and Bortz, 2021). The basic concepts of WSINDy are briefly described in the section.

5.1 *Weak SINDy*

The WSINDy is based on the so-called weak formulation in which the underlying differential equation is transformed into an integral equation. The weak formulation leads to robustness and less susceptibility to noise and errors since one needs to compute time derivatives. Consider a system described by

$$\frac{d\mathbf{x}}{dt} = \mathbf{F}(\mathbf{x}), \;\; \mathbf{x}(0) = x_0 \in \mathbb{R}^{\mathbb{D}}. \tag{10}$$

One assumes that the state variables measurements are contaminated with an identically distributed additive noise source. Thus, the noisy measurement is given by

$$\mathbf{y}_{md}(\mathbf{t}) = \mathbf{x}(\mathbf{t}) + \epsilon_{md}(\mathbf{t}), \tag{11}$$

where $\mathbf{y}_{md}(\mathbf{t})$, $\mathbf{x}(\mathbf{t}) \in \mathbb{R}^{\mathbb{M}\times\mathbb{D}}$, $\mathbf{t} = [t_1, \;\; ..., \;\; t_M]^T$, and $\epsilon_{md}(\mathbf{t}) \in \mathbb{R}^{\mathbb{M}\times\mathbb{D}}$.

For any smooth test function $\phi : \mathbb{R} \to \mathbb{R}$ supported on the interval $(a, b) \subset [0, T]$, (10) admits the weak formulation, when $0 \leq a \leq b \leq T$, given by

$$\phi(b)\mathbf{x}(b) - \phi(a)\mathbf{x}(a) - \int_a^b \phi'(u)\mathbf{x}(u)du = \int_a^b \mathbf{F}(\mathbf{x}(u))du, \tag{12}$$

which consists of the data-driven version of the Galerkin method for solving $\mathbf{F}$.

The function mapping $\mathbf{F}_d : \mathbb{R}^{\mathbb{D}} \to \mathbb{R}^D$ extracted from the noisy measurements is given by

$$\hat{\mathbf{F}}(\mathbf{y}_{md}(\mathbf{t})) = \sum_{j=1}^{J} \xi_J \theta_J(\mathbf{y}_{md}(\mathbf{t})), \tag{13}$$

where the library function $\{\theta_l(\cdot), l = 1, \cdots, J\}$ is a set of candidate basis functions used to represent $\hat{\mathbf{F}}$, an estimate for $\mathbf{F}$, and it is expressed by

$$\Theta(\mathbf{y}_{md}(\mathbf{t})) = [\theta_1(\mathbf{y}_{md}(t)) \; \theta_2(\mathbf{y}_{md}(t)) \; \ldots \; \theta_J(\mathbf{y}_{md}(t)] \, , \tag{14}$$

where each of the library function terms represents a trigonometric, polynomial function, or a product between each of the terms.

One can include the excitation input when composing the library of candidate functions Θ. In (12), if ϕ is non-constant and supported in the interval (a, b), the residual $R(\Xi, \phi)$ is defined with respect to a specific test function by substituting (13) in (12):

$$R(\Xi, \phi) = \int_a^b \phi'(u)\mathbf{y}_{md}(u)du + \int_a^b \phi(u) \sum_{j=1}^{J} \xi_j \theta_J(\mathbf{y}_{md}(u))du \tag{15}$$

The discrete-time version of (15) can be stated as:

$$R(\Xi, \phi_k) := (\mathbf{G}\Xi - \mathbf{b})_k \in \mathbb{R}^{1xD} \tag{16}$$

The Gram matrix $\mathbf{G}$ and the approximate dynamics $\mathbf{b}$ are determined through the use of the integration matrices $\mathbf{V}$ and $\mathbf{V}$'. They can be determined by

$$\mathbf{V}_k m = \Delta t \phi_k(t_m), \tag{17}$$

$$\mathbf{V}'_k m = \Delta t \phi'_k(t_m) \tag{18}$$

Hence, Eq. 16 can be rewritten as:

$$R(\Xi, \phi_k) := \mathbf{V}\Theta(\mathbf{y}_{md})\Xi - \mathbf{V}'\mathbf{y}_{md} \tag{19}$$

Defining the covariance matrix as $\Sigma = V'(V')^T$ and using it as a weighting factor, the solution to the generalized least-squares problem can be given by:

$$\hat{\Xi} = \arg\min_{\Xi} \left((\mathbf{G}\Xi - \mathbf{b}))^{\mathrm{T}} \Sigma^{-1} \left(\mathbf{G}\Xi - \mathbf{b}\right) + \gamma^2 ||\Xi||_2^2 \right) \tag{20}$$

5.2 *Input Signal Design*

One crucial aspect of a parameter estimation procedure is the design of the experiment (DoE). In the context of nonlinear systems identification, creating an informative excitation signal that captures the desired dynamics is essential, as described in Deflorian and Zaglauer (2011). While this topic is frequently discussed in system identification theory, it has received limited attention in relation to the SINDy algorithm. Previous studies (Fasel et al. (2021)) suggest the need for further investigation.

Two approaches can be used to gather valuable information through DoE: model-based or model-free. In this case, a suitable model-free DoE method involves utilizing the Latin Hypercube distribution to generate the excitation signal. The input space is limited by $\tau \in [\tau_{max}, \;\; \tau_{min}]$, $T_s \in [T_{s_{min}}, \;\; T_{s_{max}}]$, and divided into 10 intervals, which correspond to the prior information related to the actuation limits and operating characteristics. Figure 10 depicts the design points of the input space on the left side and the associated excitation signal on the right side.

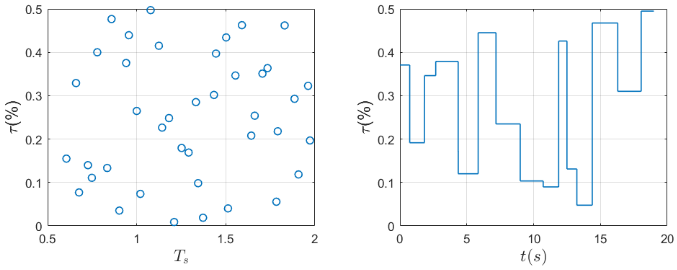


Figure 10. The excitation signal, where d_n represent the amplitude of each voltage step, and T_h represents the dwell time. Each point represents a step signal with amplitude τ and duration time T_s.

We demonstrate the application of the proposed algorithm using the aeropendulum simulation model. We retrieve data *in silico*, and the idea is to verify the performance of the WSINDy algorithm in obtaining the ODEs that characterize the system.

6. PARAMETER ESTIMATION USING THE WSINDY ALGORITHM

In this section, we propose using WSINDy to demonstrate its capabilities for determining the aerodynamic behavior of the aeropendulum using *in silico* data. By utilizing computer-generated data using the complete model, we can create a comprehensive dataset for training and testing purposes that closely mimics the dynamics of the actual system. This simulated dataset exhibits dynamics compatible with those observed in the real-world scenario. With prior knowledge of the actuator dynamics and the expected behavior within the simulation, our objective is to investigate the efficacy of the WSINDy algorithm in accurately identifying the governing equations that describe the system.

6.1 Design of excitation signal and data generation

Once we have determined the feasible electrical and mechanical parameters, we can use the simulation model to generate data for testing the capabilities of the WSINDy algorithm in recovering the system's ODEs. Particularly, we are interested in evaluating the algorithm's accuracy in capturing the aerodynamic behavior. The adapted WSINDy algorithm was implemented and is publicly available on GitHub. [1]

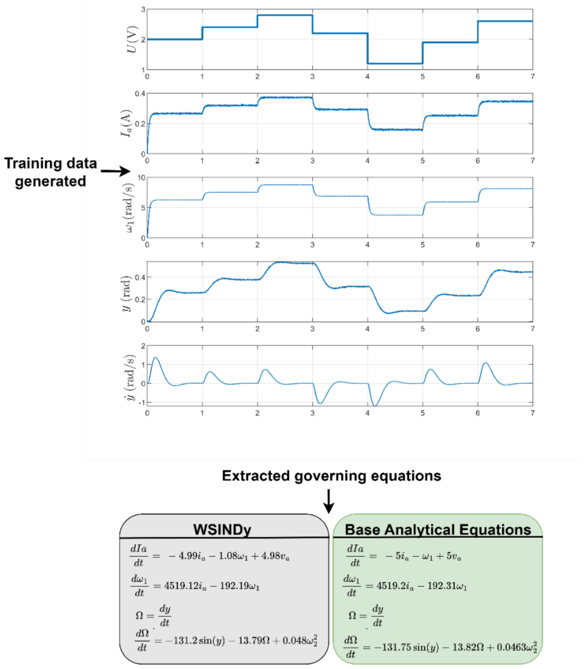
The simulation data is generated using the model, where different operating points are determined using the Latin Hypercube (LHC) input technique. The voltage operating points range from 0 V to $2,5$ V, and the step duration time ranges from $0,6$ s to 2 s. The main objective is to stimulate the model with inputs that provide valuable information and effectively showcase the system's nonlinear behavior. By doing so, the algorithm can identify a sparse representation within the dataset, capturing the essential dynamics of the system.

6.2 Noise contamination

Noise is carefully added to achieve a resulting signal with a PSNR (Peak Signal-to-Noise Ratio) of 50dB, mimicking real-world conditions with the lowest possible PSNR value before the algorithm fails. The measured state variables used in the analysis consist of the armature voltage (V_a), the armature current (I_a), the angular speed of the motor (ω_1), as well as the angular speed ($\dot{y}$) and angular position (y) of the system, as illustrated in Figure 11. These variables form the input data for the WSINDy algorithm. Consequently, the variables provided to the WSINDy algorithm are denoted as $X = [V_a \quad I_a \quad \omega_1 \quad y \quad \dot{y}]$.

6.3 Composition of the function library

A library function $\Theta(X)$ is created, which incorporates products and second power elements of each state variable and compositions involving sine and cosine functions. This selection of terms is based on prior knowledge of the system's dynamics. In this case, trigonometric functions are useful due to the force decomposition, and quadratic terms are also relevant. The training and testing data were split, with 80% of the data used for training and the remaining 20% for testing.



$$\frac{dIa}{dt} = -4.99i_a - 1.08\omega_1 + 4.98v_a$$
$$\frac{d\omega_1}{dt} = 4519.12i_a - 192.19\omega_1$$
$$\Omega = \frac{dy}{dt}$$
$$\frac{d\Omega}{dt} = -131.2\sin(y) - 13.79\Omega + 0.048\omega_2^2$$

$$\frac{dIa}{dt} = -5i_a - \omega_1 + 5v_a$$
$$\frac{d\omega_1}{dt} = 4519.2i_a - 192.31\omega_1$$
$$\Omega = \frac{dy}{dt}$$
$$\frac{d\Omega}{dt} = -131.75\sin(y) - 13.82\Omega + 0.0463\omega_2^2$$

Figure 11. Signals used in the training procedure and the extracted governing equations using the analytical model and the WSINDy algorithm.

6.4 Evaluating the results

Figure 11 depicts the training data, the expected ODEs, and the ODEs identified by the WSINDy algorithm. The algorithm successfully recovered the correct terms and values to represent the data without explicitly computing derivatives.

Specifically regarding the thrust force term, the identified coefficient was $\frac{L}{J}K_q = 0,048$. Since the mechanical parameter $\frac{L}{J}$ is known, we can determine that $\hat{K}_q = 1,037\text{e-}4$, which almost precisely matches the constant value used to describe the aerodynamic behavior of the propeller.

With the identified coefficients, we can utilize the test set to predict the dynamics' behavior, particularly $\mathbf{F}_d(\mathbf{y}_{md}(t))$, for these new state variables. Equation 13 demonstrates the application of the identified coefficients to predict the dynamics. The predicted trajectories for $\dot{I}_a$, $\dot{\omega}_2$, $\dot{y}$, and $\dot{\Omega}$ are depicted in Figure 12. The Mean Squared Error (MSE) between the ground truth obtained from the simulation and the predicted dynamics is on the order of $10e-4$.

These results ensure that the WSINDy algorithm, in tandem with the LHC-based signal design, was useful in the case of the aeropendulum platform to determine the thrust coefficient and characterize the aerodynamic behavior. Also, if there are any electrical and mechanical changes in the parameters, it helps to determine the best fit for the related ODE. In case the speed of the propeller can be directly measured during the data collection procedure, this

[1] https://github.com/dimitriarthur/Data-driven-identification-of-an-aeropendulum.

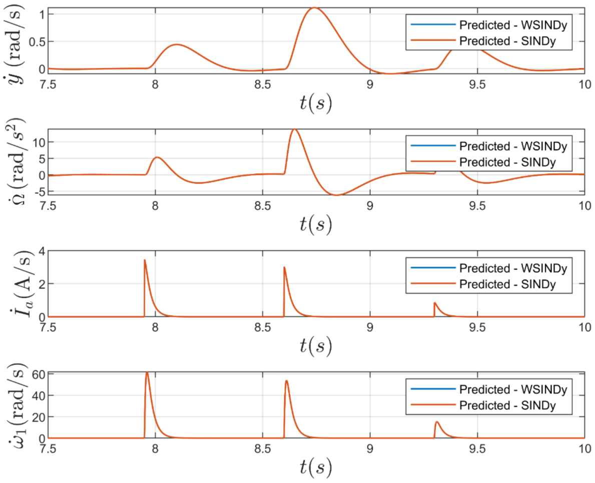


Figure 12. Dynamics predicted on the test set.

methodology should also be applied to the real scenario and able to describe the propeller's thrust.

7. CONCLUSION

In this study, we have taken a comprehensive approach to address the aeropendulum problem. Previous research efforts have encountered discrepancies between the simulated and experimental systems. This disparity arose from overlooking the actuator's electrical modeling and dynamics. We have introduced a non-linear relationship between the PWM input and the armature voltage while incorporating the electric model to address this issue. By incorporating these new component models, we have improved the system's overall modeling and conducted simulations under different operating conditions.

Furthermore, we have devised an acquisition scheme that captures the angular position and relevant variables associated with the armature equivalent circuit. This acquisition scheme plays a crucial role in the proposed estimation algorithm, as it enables the incorporation of state variables and enhances parameter estimation. To leverage our understanding of the plant's actuation limits, we have introduced a design of the input space using the LHC technique. Notably, the design of the input space using LHC has received limited attention within the WSINDy framework, making our contribution significant in this regard.

By adopting the WSINDy algorithm, results have demonstrated the effectiveness of utilizing it for describing the aerodynamic behavior of the propeller. The predicted dynamics using the WSINDy exhibited an MSE on the order of 10e-6 compared to the test set. Additionally, the algorithm has shown its capability to adapt to any parametric changes in the electrical or mechanical components, allowing it to find the correct representation of the system data.

For future research, it would be interesting to collect real-world data on the propeller's speed in the actual aeropendulum platform and assess the WSINDy algorithm's ability to accurately describe the thrust force behavior. Given the enhanced model description achieved in this study compared to previous works, evaluating different control approaches in a closed-loop setup would also be valuable. With respect to this matter, ongoing efforts are focused on refining the plant controller.

REFERENCES

Bastos, V.B. (2021). *Virtual Environments Assisted by Machine Learning for Modeling and Test of Robotic Platforms.* PhD dissertation, UNIVERSIDADE ESTADUAL DE CAMPINAS.

Brunton, S.L. and Kutz, J.N. (2022). *Data-driven science and engineering: Machine learning, dynamical systems, and control.* Cambridge University Press.

Brunton, S.L., Proctor, J.L., and 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.

Corke, P. (2017). *Robotics, Vision and Control.* Springer International Publishing.

Deflorian, M. and Zaglauer, S. (2011). Design of experiments for nonlinear dynamic system identification. *IFAC Proceedings Volumes*, 44(1), 13179–13184.

Enikov, E.T. and Campa, G. (2012). Mechatronic aeropendulum: Demonstration of linear and nonlinear feedback control principles with MATLAB/simulink real-time windows target. *IEEE Transactions on Education*, 55(4), 538–545.

Fangzheng Sun, Yang Liu, H.S. (2021). Physics-informed spline learning for nonlinear dynamics discovery. *The 30th International Joint Conference on Artificial Intelligence*, 3, 100014.

Fasel, U., Kaiser, E., Kutz, J.N., Brunton, B.W., and Brunton, S.L. (2021). SINDy with control: A tutorial. In *60th IEEE Conference on Decision and Control (CDC).* IEEE.

Ljing, L. (1999). *System Identification: Theory for the User 2nd Edition.* Prentice-Hall, Hoboken, New Jersey, U.S.

Lucena, E.R., Luiz, S.O.D., and Lima, A.M.N. (2021). Modeling, parameter estimation, and control of an aeropendulum. In *Anais do Simpósio Brasileiro de Automação Inteligente.* SBA Sociedade Brasileira de Automática.

Messenger, D.A. and Bortz, D.M. (2021). Weak sindy: Galerkin-based data-driven model selection. *Multiscale Modeling & Simulation*, 19(3), 1474–1497.

OLIVEIRA, A.D.B. (2023). *Feedback control of an a pendulum based on a data-driven identified dynamic model.* Master's thesis, Universidade Federal de Campina Grande.

Papoulis, A. (1968). *Systems and transforms with applications in optics.* McGraw-Hill Series in System Science, New York,New York, U.S.

Silva, H.R.M., Ramos, I.T.M., Cardim, R., ao, E.A., and Teixeira, M.C.M. (2020). Identification and switched control of an aeropendulum system. In *Anais do Congresso Brasileiro de Automática.* SBA.