POLITECNICO DI TORINO
Repository ISTITUZIONALE

One-shot backpropagation for multi-step prediction in physics-based system identification

Original
One-shot backpropagation for multi-step prediction in physics-based system identification / Donati, C., Mammarella, M., Dabbene, F., Novara, C., Lagoa, C.. - ELETTRONICO. - 58:(2024), pp. 271-276. (SYSID 2024 - 20th IFAC Symposium on System Identification Boston (USA) July 17-18, 2024) [10.1016/j.ifacol.2024.08.540].

Availability:
This version is available at: 11583/2992672 since: 2024-09-23T07:54:41Z

Publisher:
Elsevier

Published
DOI:10.1016/j.ifacol.2024.08.540

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

17 November 2026

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 58-15 (2024) 271–276

One-shot backpropagation for multi-step prediction in physics-based system identification

Cesare Donati [*,⋆] **Martina Mammarella** [**] **Fabrizio Dabbene** [**]
Carlo Novara [*] **Constantino Lagoa** [***]

[*] *DET, Politecnico di Torino, Turin, Italy*
(e-mail: cesare.donati, carlo.novara@polito.it)
[**] *CNR-IEIIT, Turin, Italy*
(e-mail: martina.mammarella, fabrizio.dabbene@cnr.it)
[***] *EECS, The Pennsylvania State University, University Park, PA*
(e-mail: cml18@psu.edu)

Abstract: The aim of this paper is to present a novel physics-based framework for the identification of dynamical systems, in which the physical and structural insights are reflected directly into a backpropagation-like learning algorithm. The main result is a method to compute in *closed form* the gradient of a *multi-step* loss function, while enforcing physical properties and constraints. The derived algorithm has been exploited to identify the unknown inertia matrix of a space debris, and the results show the reliability of the method in capturing the physical adherence of the estimated parameters.

Copyright © 2024 The Authors. This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0/)

Keywords: Nonlinear system identification, Grey-box modeling, Parametric optimization, Time-invariant systems, Mechanical and aerospace estimation

1. INTRODUCTION

In real-world applications, systems of interest are often not precisely known, and physically-consistent approximating models are challenging to identify. This is especially true in modern problems, which often involve complex, nonlinear, and possibly interconnected systems (Ljung et al., 2011). Moreover, incorporating physical insights while preserving simulation accuracy is not trivial, demanding a fusion between theoretical understanding and computational accuracy.

Recently, a new model class has become the subject of relevant research activities, the so-called *physics-informed neural networks* (PINNs) (Karniadakis et al., 2021). These kinds of NNs are positioned between grey-box and black-box models and allow to incorporate the available physical information, either by introducing a physics-based loss function (Gokhale et al., 2022), or directly modifying the structure of the model ensuring a consistent physical correlation between input and output (Di Natale et al., 2022). PINN techniques have been gaining a large interest for their capability of handling the main challenges posed by modern system identification. However, in PINNs usually the NN weights lack of physical interpretability.

Motivated by the previous considerations, in this paper we propose a novel identification framework, which places itself at the intersection of classical grey-box identification, where often nonlinear phenomena are ignored or simplified, and modern PINN methods, where a black-box model is embedded with prior knowledge of the system's physics (Nghiem et al., 2023), aiming to exploit the best features of these approaches. The method is based on a (possibly partial) knowledge of the physical description of a nonlinear system, which is used for the definition of a NN-like structure as a substitute for the system dynamical multi-step model. However, unlike PINNs which generally consider partial differential equations (PDEs) and a cost function with one term for fitting the data and one for fulfilling the PDE, in this paper we consider ordinary differential equations (ODEs), combining the model fulfillment and data fitting into a single loss term.

Moreover, unlike the majority of nonlinear system identification methods, which are based on the minimization of the one-step prediction error, the proposed approach relies on a multi-step loss function. Indeed, the single-step techniques used, for example, to identify Nonlinear AutoRegressive with eXogenous inputs (NARX) models or, in some cases, used in the framework of PINNs (e.g., Daw et al. (2022)), may not be accurate in multi-step prediction or simulation, and they may also fail to capture the relevant dynamics of the real system. On the other hand, solutions based on the minimization of a multi-step loss function (see e.g., Mohajerin and Waslander (2019)) provide satisfactory performance in simulation and allow to increase the long-term prediction accuracy. This comes at the expense of a high computational effort and involves, in general, solution of non-convex problems, more difficult to solve than in the one-step case.

In this paper, an efficient method for solving a multi-step-based optimization problem is proposed. Relying on the aforementioned model structure, we develop a gradient-

⋆ corresponding author *cesare.donati@polito.it*.

2405-8963 Copyright © 2024 The Authors. This is an open access article under the CC BY-NC-ND license.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2024.08.540

based identification algorithm, exploiting the well-known backpropagation method, typically used for classical NN training. The philosophy is similar to classical backpropagation, where we leverage the specific characteristics of our problem. First, we enforce the weights to be the same at *each time step* (i.e., in each layer) along the prediction horizon, since they have the same physical interpretation and being the system *time-invariant*. Second, in our proposed architecture the "activation functions" are fixed using the physical dynamics f in *each layer*. Consequently, the *weights* have an *explainable* and *interpretable* meaning, representing the *physical* parameters of the system $\mathcal{S}$ to be identified. Similarly, in (Abbasi and Andersen, 2022) the authors introduce the concept of *physical activation functions* (PAFs), where the mathematical expression of the activation function is inherited from the physical laws of the investigated phenomena. However, these PAFs are applied only in one *hidden layer*, and combined with other general activation functions, e.g., sigmoids.

Our formulation allows the definition of an *analytical*, *one-shot* computation of the gradient, that exploits all the available physics-based constraints on the system states and parameters and, if any, the system structural information. The *generality* of the underlying structure allows us to deal with real-world situations where the system to identify may be partly inherited from the physics and partly unknown, and the values of some parameters may be available, while others need to be identified. Moreover, the proposed approach allows to reflect the physical characteristics of the system behavior through the introduction of specific penalty terms in the cost function (Zakwan et al., 2022; Medina and White, 2023), ensuring models adherence to fundamental physics principles.

The remainder of the paper is structured as follows. In Section 2, we define the considered framework, introducing the main features of the considered system dynamics and of the estimation model. The analytic computation of the gradient is detailed in Section 3, together with the approach used to enforce possible physics-based constraints based on prior knowledge of the system. Simulation results obtained with the proposed approach are discussed in Section 4. Main conclusions are drawn in Section 5.

Notation Given a vector v, we denote by $\mathbf{v}_{1:T} \doteq \{v_k\}_{k=1}^{T}$ the set of vectors $\{v_1, \dots, v_T\}$. Given integers $a \leq b$, we denote by $[a, b]$ the set of integers $\{a, \dots, b\}$. The Jacobian matrix of α_k with respect to β_k is denoted as $\mathcal{J}_k^{\alpha/\beta} \in \mathbb{R}^{n_\alpha \times n_\beta}$, i.e., $\frac{\partial \alpha_k}{\partial \beta_k}$. Similarly, $\mathcal{J}_k^{\alpha/\alpha} \in \mathbb{R}^{n_\alpha \times n_\alpha}$ is the Jacobian matrix of α_k with respect to α_{k-1}, i.e., $\frac{\partial \alpha_k}{\partial \alpha_{k-1}}$.

2. FRAMEWORK DEFINITION

2.1 Problem setup

We consider a nonlinear, time-invariant dynamical system $\mathcal{S}$, possibly composed by interconnected subsystems. We are given a physics-based description of the system, i.e., defined by means of discrete-time state equations capturing the physical interaction between variables:

$$\mathcal{S}: \quad \begin{aligned} x_{k+1} &= f(x_k, u_k, \theta), \\ z_k &= g(x_k) + d_k, \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^{n_x}$ is the state vector, $u \in \mathbb{R}^{n_u}$ is the (external) input vector to $\mathcal{S}$, $z \in \mathbb{R}^{n_z}$ is the observation vector, and d is the measurement noise. The functions $f(x, u, \theta)$ and $g(x)$ are known, and represent the dynamical laws and the observation function respectively. They are assumed to be nonlinear, time-invariant, and at least C^1 differentiable. The goal is to identify both the physical parameters $\theta \in \mathbb{R}^{n_\theta}$ and the initial condition $x_0 \in \mathbb{R}^{n_x}$, starting from measured input-output sequences, leading to an estimation model $\widehat{\mathcal{S}}$ of $\mathcal{S}$ of the form

$$\widehat{\mathcal{S}}: \quad \begin{aligned} \hat{x}_{k+1} &= f(\hat{x}_k, u_k, \hat{\theta}), \\ \hat{z}_k &= g(\hat{x}_k), \end{aligned} \tag{2}$$

where $\hat{x}_k$, and $\hat{z}_k$ are the estimated state and output at time k, respectively.

We assume we have available a T-step measured, input sequence $\widetilde{\mathbf{u}}_{0:T-1}$ and the corresponding T collected observations $\widetilde{\mathbf{z}}_{0:T-1}$. The objective is to estimate the optimal values of the parameters $\hat{\theta}^\star$ and initial condition $\hat{x}_0^\star$ over the horizon T such that $\widehat{\mathcal{S}}$ is the best approximation of $\mathcal{S}$, given its underlying physical structure and the measured data $\{\widetilde{\mathbf{u}}_{0:T-1}, \widetilde{\mathbf{z}}_{0:T-1}\}$ [1].

First, given the output predictions $\hat{z}$ and the true measurements $\widetilde{z}$, we define the prediction error at time k as

$$e_k \doteq \hat{z}_k - \widetilde{z}_k. \tag{3}$$

Note that this is a multi-step prediction error, since $\hat{z}_k$ is obtained through successive iterations of equations (2). The local loss at time k is defined by the weighted norm of the error,

$$\mathcal{L}(e_k, \theta) \doteq \frac{1}{T}\|e_k\|_{\mathcal{Q}}^2 \doteq \frac{1}{T} e_k^\top \mathcal{Q} e_k, \tag{4}$$

with $\mathcal{Q} \succeq 0$.

In this paper, we consider a *multi-step regression cost* $\mathcal{C}$ as a sum of local losses over the prediction horizon T as

$$\mathcal{C}(e_k, \theta) = \sum_{k=0}^{T-1} \mathcal{L}(e_k, \theta) \doteq \sum_{k=0}^{T-1} \mathcal{L}_k. \tag{5}$$

Then, we can define our nonlinear, parametric model identification problem as

$$(\hat{\theta}^\star, \hat{x}_0^\star) \doteq \arg\min_{\theta, x_0} \ \mathcal{C}(e_k, \theta), \tag{6}$$

in which we want to minimize the mean squared error over sampled measurements to obtain an estimate of θ and x_0.

2.2 Multi-step dynamics propagation

Given the dynamical model $\mathcal{S}$, it is possible to propagate each state variable x_i, $i \in [1, n_x]$ over a desired horizon T, simply applying the model $\mathcal{S}$ *recursively*, i.e.,

$$x_{i,k+1} = f_i(x_k, u_k, \theta), \quad k \in [0, T]. \tag{7}$$

The model can be depicted as in Fig. 1, where the recursion is captured by the delay block. Clearly, this can also be represented opening the output loop T steps ahead from the initial time $k = 0$.

We observe that what we obtain closely resembles the well-known structure of neural networks. Indeed, each

[1] The proposed algorithm can be adapted to the case of multiple trajectories with the same length T.

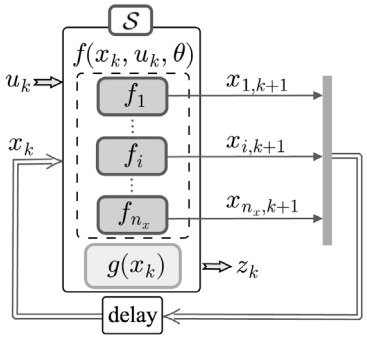


Fig. 1. Recursive representation of a dynamical system.

time step k can be seen as a "*layer*" composed by n_x "*neurons*", and the interconnection links between layers and neurons, are activated or deactivated according to the system dynamical structure defined in $\mathcal{S}$. In particular, if $x_{i,k+1}$ does not depend on $x_{j,k}$, the corresponding link is null. This allows to envision the model $\mathcal{S}$ as a neural network graph and, consequently, the "weights" of the network are the interpretable, physical parameters of the system.

Since the overall objective function in (6) is (in general) *non-convex*, due to the nonlinearity in θ and x_k of $f(x_k, u_k, \theta)$ and $g(x_k)$ (1), we rely on gradient-based algorithms (Sun et al., 2019) to address the optimization problem, aiming to reach some (local) minima and eventually compute a (sub)optimal estimation of θ and x_0.

We observe that, inspired by the approach typically adopted for neural network graphs, we can exploit a classical backpropagation scheme to analytically compute the gradient of the loss function, thanks to the structure of the physics-based model $\mathcal{S}$. However, as it will be clarified in Section 3, differently from neural network backpropagation, the proposed scheme presents the *same weigths* θ and the *same functions* in all layers. This crucial feature allows us to derive a useful closed form of the gradient of $\mathcal{C}(e_k, \theta)$ with respect to θ and x_0, i.e., $\nabla\mathcal{C} = [\nabla_\theta\mathcal{C},\ \nabla_{x_0}\mathcal{C}]$. Once these gradients are computed, it is possible to apply a gradient-based algorithm to solve the optimization problem (5), such that the estimate of θ and x_0 are updated at each epoch ℓ. For instance, if a classical gradient descent method is applied, we would have

$$\hat{\theta}^{(\ell+1)} = \hat{\theta}^{(\ell)} - \eta_\theta \nabla_\theta^\top \mathcal{C}^{(\ell)} \tag{8}$$

$$\hat{x}_0^{(\ell+1)} = \hat{x}_0^{(\ell)} - \eta_{x_0} \nabla_{x_0}^\top \mathcal{C}^{(\ell)} \tag{9}$$

with learning rates η_θ, η_{x_0}. In this paper, we select the ADAM first-order method (Kingma and Ba, 2015) with decay rates β_1, β_2.

The whole procedure is presented in Algorithm 1. At epoch ℓ, we first propagate the system with initial conditions $\hat{x}_0^{(\ell)}$ and parameters $\hat{\theta}^{(\ell)}$ through the network layer-by-layer (i.e., along the horizon T). Then, we evaluate the gradient based on the computed predictions, and accordingly, we update the weights, i.e., $\hat{\theta}^{(\ell)}$ and $\hat{x}_0^{(\ell)}$. This process repeats over ℓ until at least one of the following conditions is satisfied: (a) the maximum number of epochs, i.e., E_{max}, is reached; (b) the structure converges to a (possibly local) minimum of the loss function, or below a given threshold ε; (c) the magnitude of the gradient is lower than a given minimum step size δ.

Algorithm 1 One-shot backpropagation-based identification

1: Given T input-output observations $\{\widetilde{\mathbf{u}}_{0:T-1}, \widetilde{\mathbf{z}}_{0:T-1}\}$, choose η_θ, η_{x_0}, β_1, β_2, E_{max}, ε, and δ.
2: Initialize $\ell = 0$ and $\hat{x}_0^{(0)}$, $\hat{\theta}_0^{(0)}$.
3: **while** $\ell \leq E_{max}$ **and** $\mathcal{C}^{(\ell)} \geq \varepsilon$ **and** $\|\nabla\mathcal{C}^{(\ell)}\|_2 \geq \delta$ **do**
4: Simulate (2) for $k \in [0, T-1]$ using $\hat{\theta}^{(\ell)}$, $\hat{x}_0^{(\ell)}$ to obtain $\hat{\mathbf{x}}_{1:T}^{(\ell)}, \hat{\mathbf{z}}_{0:T-1}^{(\ell)}$.
5: Compute $\mathbf{e}_{0:T-1}^{(\ell)}$ (3) and $\mathcal{C}^{(\ell)}$ (5).
6: Compute $\nabla_\theta\mathcal{C}^{(\ell)}$ (17) and $\nabla_{x_0}\mathcal{C}^{(\ell)}$ (20).
7: Update the weights using ADAM, i.e.,
$\hat{\theta}^{(\ell+1)} = \mathrm{ADAM}(\hat{\theta}^{(\ell)}, \eta_\theta, \beta_1, \beta_2, \nabla_\theta\mathcal{C}^{(\ell)})$,
$\hat{x}_0^{(\ell+1)} = \mathrm{ADAM}(\hat{x}_0^{(\ell)}, \eta_{x_0}, \beta_1, \beta_2, \nabla_{x_0}\mathcal{C}^{(\ell)})$.
8: $\ell \leftarrow \ell + 1$.
9: **end while**
10: Return $\hat{\theta}^\star = \hat{\theta}^{(\ell)}$ and $\hat{x}_0^\star = \hat{x}_0^{(\ell)}$

3. CLOSED-FORM GRADIENT COMPUTATION

In this section, we describe the procedure to compute the gradient in closed form relying on the structure of $\mathcal{S}$ and the available measurements. This procedure is the core of Algorithm 1. We compute the gradient of the cost function $\mathcal{C}$ with respect to θ and x_0, i.e., $\nabla_\theta\mathcal{C} = \frac{\mathrm{d}\mathcal{C}}{\mathrm{d}\theta}$ and $\nabla_{x_0}\mathcal{C} = \frac{\mathrm{d}\mathcal{C}}{\mathrm{d}x_0}$ as the product of some intermediate partial derivatives that, unlike what happens in standard neural networks, share a common formulation and allow to compute the gradient in a fully analytic way. Hence, at epoch ℓ, the analytic form of the gradient can be simply *evaluated* at the current value of $\hat{\theta}^{(\ell)}, \hat{x}_0^{(\ell)}$ and the ensuing predictions, that is

$$\nabla_\theta\mathcal{C}^{(\ell)} = G_\theta\left(\hat{\theta}^{(\ell)}, \hat{x}_0^{(\ell)}, \hat{\mathbf{x}}_{1:T}^{(\ell)}, \hat{\mathbf{z}}_{0:T-1}^{(\ell)}\right)$$

$$\nabla_{x_0}\mathcal{C}^{(\ell)} = G_{x_0}\left(\hat{\theta}^{(\ell)}, \hat{x}_0^{(\ell)}, \hat{\mathbf{x}}_{1:T}^{(\ell)}, \hat{\mathbf{z}}_{0:T-1}^{(\ell)}\right).$$

The closed-form expressions of the two gradients are presented in the following sections. In the sequel, for readability, we omit the superscript (ℓ) denoting the epochs.

3.1 Gradient with respect to parameters

In the proposed framework, we can obtain the closed-form expression of $\nabla_\theta\mathcal{C}$ on the measured data $\{\widetilde{\mathbf{u}}_{0:T-1}, \widetilde{\mathbf{z}}_{0:T-1}\}$ by considering the effect of the (current, in terms of epochs) estimate $\hat{\theta}$ *for each time step* k on the cost $\mathcal{C}$. The desired gradient can be obtained as

$$\nabla_\theta\mathcal{C} = \sum_{k=1}^{T-1} \left.\frac{\mathrm{d}\mathcal{C}}{\mathrm{d}\theta}\right|_k, \tag{10}$$

where $\left.\frac{\mathrm{d}\mathcal{C}}{\mathrm{d}\theta}\right|_k$ is the effect of $\hat{\theta}$ on the cost $\mathcal{C}$ at an arbitrary time step k within the prediction horizon T, and for each k we have

$$\left.\frac{\mathrm{d}\mathcal{C}}{\mathrm{d}\theta}\right|_k = \left.\frac{\partial\mathcal{C}}{\partial\theta}\right|_{k|k} + \sum_{\tau=k+1}^{T-1} \left.\frac{\mathrm{d}\mathcal{C}}{\mathrm{d}\theta}\right|_{\tau|k}. \tag{11}$$

Indeed, this analysis takes into account both the "direct" effect of $\hat{\theta}$ at time k on $\mathcal{L}_k$, i.e., $\frac{\partial \mathcal{C}}{\partial \theta}\big|_{k|k}$, and the "collateral" effects, i.e., $\sum_{\tau=k+1}^{T-1} \frac{\mathrm{d}\mathcal{C}}{\mathrm{d}\theta}\big|_{\tau|k}$, on the subsequent local losses $\mathcal{L}_\tau$ for all $\tau \in [k+1, T]$, arising from the propagation of the error originated from $\hat{\theta}$ to the predicted state $\hat{x}_k$.

For the first term in (11), we can apply the chain rule of differentiation, as typically done in classical backpropagation, and we obtain

$$\begin{aligned} \frac{\partial \mathcal{C}}{\partial \theta}\bigg|_{k|k} &= \frac{\partial \mathcal{L}_k}{\partial \theta} + \frac{\partial \mathcal{L}_k}{\partial e_k}\frac{\partial e_k}{\partial z_k}\frac{\partial z_k}{\partial x_k}\frac{\partial x_k}{\partial \theta} \\ &= \nabla_\theta \mathcal{L}_k + \nabla_e \mathcal{L}_k \mathcal{J}_k^{e/z} \mathcal{J}_k^{z/x} \mathcal{J}_k^{x/\theta}. \end{aligned} \tag{12}$$

Then, for the general term $\frac{\mathrm{d}\mathcal{C}}{\mathrm{d}\theta}\big|_{\tau|k}$, we apply again the chain rule and we have

$$\begin{aligned} \frac{\mathrm{d}\mathcal{C}}{\mathrm{d}\theta}\bigg|_{\tau|k} &= \frac{\partial \mathcal{L}_\tau}{\partial e_\tau}\frac{\partial e_\tau}{\partial z_\tau}\frac{\partial z_\tau}{\partial x_\tau}\prod_{c=0}^{\tau-k-1}\frac{\partial x_{\tau-c}}{\partial x_{\tau-c-1}}\frac{\partial x_k}{\partial \theta} \\ &= \nabla_e \mathcal{L}_\tau \mathcal{J}_\tau^{e/z} \mathcal{J}_\tau^{z/x} \prod_{c=0}^{\tau-k-1} \mathcal{J}_{\tau-c}^{x/x} \mathcal{J}_k^{x/\theta}, \end{aligned} \tag{13}$$

where the chain multiplication of $\mathcal{J}^{x/x}$ evaluated at different time steps is exploited to back-propagate the error from τ to k and compute the exact desired contribution of $\hat{\theta}$ to $\mathcal{C}$ due to the propagation of $\hat{x}_k$ from time k to time τ.

Then, let us define the following two quantities, i.e.,

$$\gamma_k \doteq \nabla_\theta \mathcal{L}_k, \quad \Gamma_k \doteq \nabla_e \mathcal{L}_k \mathcal{J}_k^{e/z} \mathcal{J}_k^{z/x}, \tag{14}$$

such that

$$\frac{\partial \mathcal{C}}{\partial \theta}\bigg|_{k|k} = \gamma_k + \Gamma_k \mathcal{J}_k^{x/\theta}, \tag{15}$$

$$\frac{\mathrm{d}\mathcal{C}}{\mathrm{d}\theta}\bigg|_{\tau|k} = \Gamma_k \prod_{c=0}^{\tau-k-1} \mathcal{J}_{\tau-c}^{x/x} \mathcal{J}_k^{x/\theta}, \tag{16}$$

and substituting these terms in (11), we obtain the one-shot formulation for computing $\nabla_\theta \mathcal{C}$ as

$$\begin{aligned} \nabla_\theta \mathcal{C} = &\sum_{k=1}^{T-1} \gamma_k + \sum_{k=1}^{T-1} \Gamma_k \mathcal{J}_k^{x/\theta} \\ &+ \sum_{k=1}^{T-1}\sum_{\tau=k+1}^{T-1} \left(\Gamma_\tau \prod_{c=0}^{\tau-k-1} \mathcal{J}_{\tau-c}^{x/x}\right) \mathcal{J}_k^{x/\theta}. \end{aligned} \tag{17}$$

Remark 1. By incorporating the model structure $\mathcal{S}$ directly into the network structure, the backpropagation of errors can be efficiently computed using the chain multiplication of the same Jacobian matrix $\mathcal{J}_k^{x/x}$. The parametric computation of this Jacobian can be performed once for all, and later evaluated at different time steps. This will allow to reduce the number of partial derivatives to be computed and, consequently, the computational complexity of the proposed approach.

3.2 Gradient with respect to initial condition

Let us now consider the explicit formulation for the gradient with respect to the initial condition

$$\nabla_{x_0} \mathcal{C} = \sum_{k=1}^{T-1} \frac{\mathrm{d}\mathcal{C}}{\mathrm{d}x_0}\bigg|_{k|0}. \tag{18}$$

The one-shot, analytical expression can be derived by considering the effect of x_0 on each subsequent prediction $\hat{x}_k$ and, consequently, on the cost $\mathcal{C}$. In this case, there is no "direct" effect of $\hat{x}_0$ on the final cost, but we must account for the "collateral" effects of $\hat{x}_0$ on the subsequent local-losses $\mathcal{L}_\tau$ for all $\tau = [1, T]$. These effects arise from the error originating from $\hat{x}_0$ and propagated throughout the predictions along T. Consequently, we obtain

$$\begin{aligned} \frac{\mathrm{d}\mathcal{C}}{\mathrm{d}x_0}\bigg|_{k|0} &= \frac{\partial \mathcal{L}_k}{\partial e_k}\frac{\partial e_k}{\partial z_k}\frac{\partial z_k}{\partial x_k}\prod_{c=0}^{k-1}\frac{\partial x_{k-c}}{\partial x_{k-c-1}} \\ &= \nabla_e \mathcal{L}_k \mathcal{J}_k^{e/z} \mathcal{J}_k^{z/x} \prod_{c=0}^{k-1} \mathcal{J}_{k-c}^{x/x}, \end{aligned} \tag{19}$$

which in compact form can be rewritten as

$$\nabla_{x_0} \mathcal{C} = \sum_{k=1}^{T-1} \Gamma_k \prod_{c=0}^{k-1} \mathcal{J}_{k-c}^{x/x}. \tag{20}$$

3.3 Physics-based constraints

To guarantee the coherence among the physics of the phenomena and the estimated parameters, exploiting the physical laws as activation functions may be not sufficient. It may be needed to reflect the specificity of the system behavior, such as e.g. passivity, monotonicity, divergence, symmetry of variables, stability (Medina and White, 2023; Zakwan et al., 2022), thus ensuring that the identified models adhere to fundamental laws and are consistent with physical principles. This aspect can be formally embedded into the cost $\mathcal{C}$ by means of *penalty terms* that introduce physical constraints, of the form

$$h(\hat{x}_k, \theta) \leq 0, \ \ \forall k \in [0, T],$$

with $h : \mathbb{R}^{n_x} \times \mathbb{R}^{n_\theta} \to \mathbb{R}$ a time-invariant function, (at least) C^1 differentiable. Specifically, the general cost $\mathcal{C}$ is modified as follows

$$\mathcal{C} = \sum_{k=0}^{T-1} \mathcal{L}_k + \lambda h(\hat{x}_k, \theta), \tag{21}$$

where $\lambda \in \mathbb{R}$ is a Lagrange multiplier that controls the relevance of the physical constraint $h(\hat{x}_k, \theta)$ such that the higher is the violation of the physical properties in the predicted states and weights, the larger is the associated loss value.

Deterministic physical constraints exhibit themselves in a wide range of forms from simple algebraic equations to nonlinear integer-differential equations and inequalities. Thus, it is possible to enforce a large variety of physics-based constraints through a sharp customization of $h(\hat{x}_k, \theta)$. For instance, one possibility is to define the penalty term to minimize the deviation of the total energy of the system with respect to the reference level, thus enforcing the principle of energy conservation for mechanical systems. Another example may be the use of an *exponential barrier function* as constraint to guarantee some physical properties of the state variables. In case we need to constrain the parameters into a specified convex set, we can enhance the identification algorithm introducing a *projection step* immediately after the parameters update, such that in case of constraints violation, the parameters are projected onto the desired set. Similarly, equality constraints may be enforced by adding a quadratic penalty

term in the cost. More details and examples can be found in Donati et al. (2023).

4. NUMERICAL RESULTS

The attitude dynamics of the satellite is usually modeled using the standard Euler equations, i.e.,

$$I\dot{\omega} = M - \omega \times I\omega, \quad \widetilde{\omega} = \omega + e_\omega, \tag{22}$$

where $\omega = [\omega_x, \omega_y, \omega_z]^\top$ is the angular velocity and $\widetilde{\omega}$ the measured output, I is the satellite inertia tensor, M is the input torque, and e_ω is the measurement noise. In the follows, we assume $M \sim \mathcal{N}(10^{-5}, \sigma_{M_d})$ with $\sigma_{M_d} = 10^{-7}\frac{\text{rad}}{\text{s}}$, representing for instance solar radiation pressure, and $e_\omega \sim \mathcal{N}(0, \sigma_\omega)$ with $\sigma_\omega = 10^{-4}$rad/s. [2]

Here, the objective is to estimate the optimal value for the satellite diagonal inertia matrix (i.e., the physical parameters $\hat{\theta}$ are the diagonal elements of $\hat{I}$) and the initial angular velocity $\hat{\omega}_0$ (i.e., $\hat{x}_0$), starting from some tentative values (I, ω_0) and given collected output samples, applying the proposed approach. For the validation, we generated a sequence of $T = 50$ data, integrating (22) with a sampling time of 0.1 s. The true systems is initialized with $\omega_0 = [9.915 \cdot 10^{-6}, -1.102 \cdot 10^{-3}, 1.3179 \cdot 10^{-5}]^\top$ and $\theta = [0.0403, 0.0404, 0.0080]^\top$. Moreover, physical constraints are imposed on the diagonal elements of $\hat{I}$ by introducing a projection step immediately after the update phase, such that for all i we have $\theta_i > 0$.

Remark 2. While the emphasis in this section lies on θ due to its higher significance in the considered framework, it is important to note that the achieved results were obtained by estimating both θ and x_0.

In Fig. 2, we can observe the convergent behavior of the loss function over the algorithm iteration epochs ℓ, represented in the estimated parameter space. This behavior is confirmed in Fig. 3, where we depict the evolution of the estimated parameters with respect to ℓ for different initial conditions of $\hat{\theta}$ and $\hat{x}_0$. It is worth noting that the

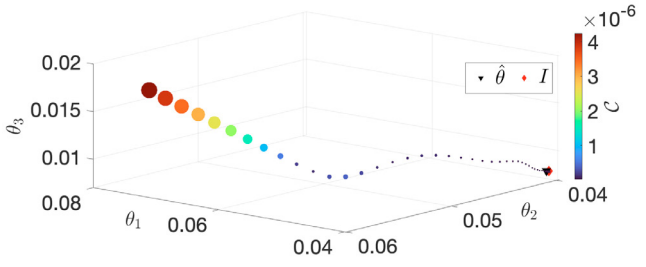


Fig. 2. Evolution of $\mathcal{C}$ over the estimation parameter space.

computed gradient might initially move some parameters away from their intended final values (e.g., the peak in the second plot). This temporary shift allows focusing on correcting more crucial parameters first, before eventually re-adjusting the divergent parameter towards convergence.

Then, in Fig. 4 we compare the performance of the proposed algorithm (red line) with respect to three different

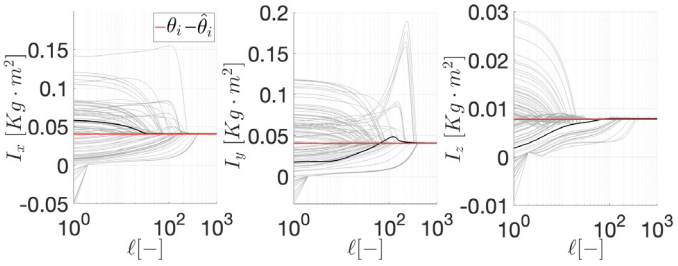


Fig. 3. Comparison between estimated parameters $\hat{\theta}_i$ and real ones θ_i.

approaches: (i) a gray-box (GB) model [3] (green line), which is fed with the dynamical model in (22) and minimizes a single-step prediction error; (ii) a multi-step (ms) model (orange line) and (iii) a single-step (ss) model (purple line), both implemented using the same cost function as our approach but different algorithms to compute the gradient, i.e., `fmincon` function with a `sqp` setting. Given

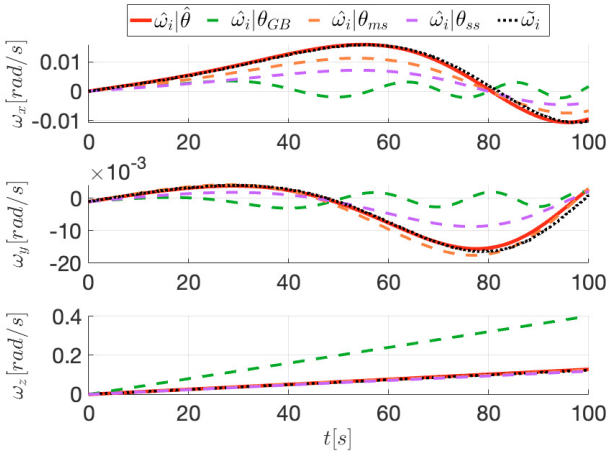


Fig. 4. Evolution of $\hat{\omega}_i(t)$ with different approaches.

the same training dataset, we use all the aforementioned approaches to estimate the physical parameters θ, and then to propagate the dynamics over a longer simulation horizon (i.e., $t \in [0, 100]$), overlapping the results with the true measurements (black line). We can observe that both multi-step approaches are able to properly capture the physics of the system better than the GB and ss. However, we need to emphasize that, due to the inherent instability of the trajectories generated by the nonlinear system (22), it is expected that also the trajectory estimated using our approach could eventually diverge from the actual one. Indeed, in this context, the goal of multi-step identification is to identify parameters that enable the longest horizon of accurate predictions given a training sequence of T data.

Between the two multi-step approaches the main difference resides in the gradient computation, i.e., *analytically computed* in our approach and *numerically approximated* for the standard multi-step approach, and how this affects the estimation algorithm. This is highlighted in Fig. 5 where we compare three multi-step approaches, sharing the same solver `fmincon` with $E_{max} = 100$, in terms of estimation error $\|\hat{\theta} - \theta\|_2$. We can observe that by providing

[2] The noise values, despite appearing rather small, are compatible with the case study selected (i.e., around 10% of the state values).

[3] We exploited the MATLAB *System identification Toolbox* to implement the GB method, using the `nlgreyest` function.

our one-shot, analytic gradient to the `fmincon` solver, we can achieve a significant improvement (one order of magnitude) in the estimation accuracy (green diamond) with respect to the results obtained by using numerically approximated gradients provided by the differentiation tool of `fmincon` with `ipopt` (red circles) or `sqp` (black squares) optimization methods. Moreover, we can also notice that by using the same one-shot, analytic gradient fed to a different solver, i.e., `Adam` (yellow triangles), we can further improve the identification performance.

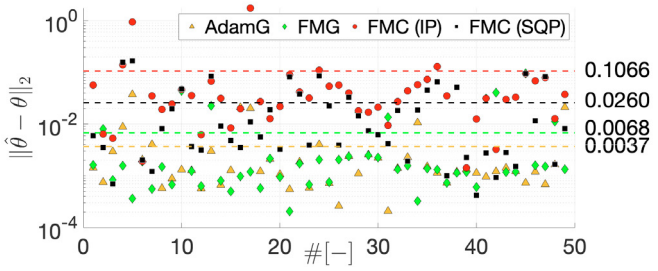


Fig. 5. Comparison among four multi-step approaches: 1) `Adam` with analytic gradient (triangle), 2) `fmincon` with analytic gradient (diamond), 3) `ipopt-fmincon` (circle), and 4) `sqp-fmincon` (square).

The last aspect analyzed is the correlation among the prediction horizon T, the quality of the estimated parameters $\hat{\theta}$ and the computation time for the proposed multi-step identification scheme. To compare the performance with respect to the required time we performed different simulations using different prediction horizons. As shown in Fig. 6, the larger is T (i.e., the larger is the number of data used to compute the gradient), the higher the computation time (blue line) required to complete the identification will be. Observing the estimation performance, we can select a trade-off horizon between performance improvement and required computation time ($T = 50$, $\hat{\theta} = [0.0398, 0.0389, 0.0076]^\top$).

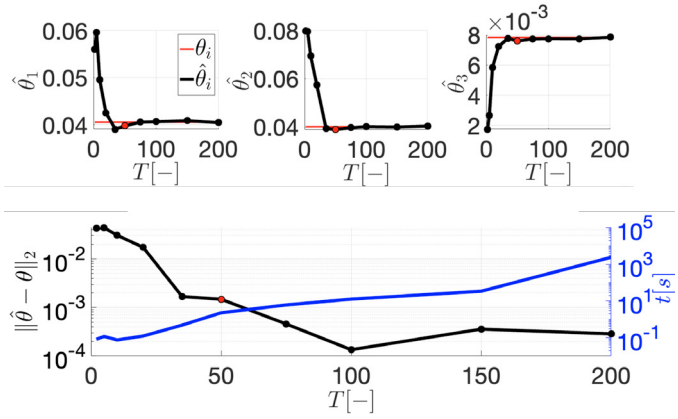


Fig. 6. Estimated $\hat{\theta}_i$ and estimation error for different prediction horizons T.

5. CONCLUSIONS AND FUTURE RESEARCH

In this work, we have proposed a general framework for the identification of complex dynamical systems focusing on multi-step prediction accuracy. We have presented here the main technical steps, concentrating on the case when a physical description of each subsystem is available. However, we want to remark that the approach is general, and it can be extended to situations where only partial information on the structure or on the state equations is available. This is the subject of current research. In particular, in the case of partially known equations, the idea is to assume that the model to estimate is given by the sum of two contributions: a term directly modeled according to the (underlying) physics of the system, and another one capturing the unmodeled dynamics.

REFERENCES

Abbasi, J. and Andersen, P.Ø. (2022). Physical Activation Functions (PAFs): An Approach for More Efficient Induction of Physics into Physics-Informed Neural Networks (PINNs). *arXiv preprint arXiv:2205.14630*.

Daw, A., Karpatne, A., Watkins, W.D., Read, J.S., and Kumar, V. (2022). Physics-guided neural networks (PGNN): An application in lake temperature modeling. In *Knowledge Guided Machine Learning*, 353–372.

Di Natale, L., Svetozarevic, B., Heer, P., and Jones, C.N. (2022). Physically consistent neural networks for building thermal modeling: Theory and analysis. *Applied Energy*, 325.

Donati, C., Mammarella, M., Dabbene, F., Novara, C., and Lagoa, C. (2023). One-shot backpropagation for multi-step prediction in physics-based system identification – EXTENDED VERSION. *arXiv preprint arXiv:2310.20567*.

Gokhale, G., Claessens, B., and Develder, C. (2022). Physics informed neural networks for control oriented thermal modeling of buildings. *Applied Energy*, 314.

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

Kingma, D. and Ba, L. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR 2015)*.

Ljung, L., Hjalmarsson, H., and Ohlsson, H. (2011). Four encounters with system identification. *European Journal of Control*, 17(5), 449–471.

Medina, J. and White, A.D. (2023). Active learning in symbolic regression performance with physical constraints. *arXiv preprint arXiv:2305.10379*.

Mohajerin, N. and Waslander, S.L. (2019). Multistep prediction of dynamic systems with recurrent neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11), 3370–3383.

Nghiem, T.X., Drgoňa, J., Jones, C., Nagy, Z., Schwan, R., Dey, B., Chakrabarty, A., Di Cairano, S., Paulson, J.A., Carron, A., Zeilinger, M.N., Shaw Cortez, W., and Vrabie, D.L. (2023). Physics-informed machine learning for modeling and control of dynamical systems. In *2023 American Control Conference (ACC)*.

Sun, S., Cao, Z., Zhu, H., and Zhao, J. (2019). A survey of optimization methods from a machine learning perspective. *IEEE Transactions on Cybernetics*, 50(8), 3668–3681.

Zakwan, M., Di Natale, L., Svetozarevic, B., Heer, P., Jones, C.N., and Trecate, G.F. (2022). Physically consistent neural ODEs for learning multi-physics systems. *arXiv preprint arXiv:2211.06130*.