JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 45, No. 2, February 2022

Filtered Integral Formulation of the Sparse Model Identification Problem

Damien Guého* and Puneet Singla†
The Pennsylvania State University, University Park, Pennsylvania 16802
Manoranjan Majji‡
Texas A&M University, College Station, Texas 77840
and
Robert G. Melton§
The Pennsylvania State University, University Park, Pennsylvania 16802

https://doi.org/10.2514/1.G005952

This paper presents a generalized approach to identify the structure of governing nonlinear equations of motion from the time history of state variables and control functions. An integral form involving a low-pass filter in conjunction with sparse approximation tools is used to find a parsimonious model for underlying true dynamics from noisy measurement data. Two chaotic oscillatory systems as well as the well-known problem of identifying the central force field from position-only observation data are considered to validate the developed approach. The simulation results considered in the paper demonstrate the performance of the developed approach in learning unknown nonlinear system dynamics accurately with fewer basis functions as compared with classical least-squares regression techniques and emerging deep learning approaches. A comparison of the sparse identification techniques with classical least-squares regression techniques and emerging deep learning approaches reveals the utility of the methodology developed in the paper.

I. Introduction

THE field of system identification has been an important discipline within the automatic control area, structural engineering, reduced order modeling and model testing [1,2]. The special process of dynamic system identification corresponds to identifying a mathematical model describing a relationship between input and output of a real system. Over the past six decades, the field of system identification has provided multiple tools for design, analysis, and control of engineering systems. Since the initial efforts, the system identification community has sought a reliable methodology to derive a mathematical model capturing the main characteristics of dynamic systems. However, the most mature part of the theory deals with linear systems using well-established techniques of linear algebra and the theory of ordinary differential or difference equations. In contrast, nonlinear system identification problems are still treated mostly on a system-by-system basis. Earlier efforts in the field of linear system identification concentrated on construction of state-space representations of linear systems. Building upon initial work by Gilbert [3] and Ho and Kálmán [4], several methods have been developed to identify most observable and controllable subspaces of the system from given input–output (I/O) data [5–15]. Subspace methods for system identification such as the Eigensystem Realization Algorithm (ERA) [12] and the Observer/Kalman Identification Algorithm (OKID) [14] are used to recover minimal observable and controllable realizations of system models of large distributed systems from I/O data. In the last decade, the focus has been generalizing these methods to identify time-varying linear systems [16–18] with a step toward nonlinear system identification.

In contrast to linear system identification, nonlinear system identification problems are still treated mostly on a system-by-system basis, with popular methods being Volterra series models [2,19,20], global–local learning [21,22], and neural network (NN) models [23]. The main essence of nonlinear system identification methods has been to expand the nonlinear unknown function as a linear combination of basis functions or kernels and their amplitude. Many of these methods differ in their choice of basis functions and their learning methodology. Methods like Volterra series approximation use Volterra kernels to provide a global approximation of the underlying dynamics, whereas global–local approximation methods merge various local approximations valid in a local region to find a global approximation of the underlying dynamics [21]. More prevalent machine learning methods such as multilayered NNs (also known as deep NNs) use a composition of nonlinear transformations to approximate the unknown I/O mapping. Each layer of the NN corresponds to one nonlinear transformation that is represented by a linear combination of fixed bases such as sigmoid functions known as neurons or perceptrons. According to Cover [24] and Kolmogorov's theorems, multilayer NNs can serve as universal approximators, but in actuality, they offer no guarantee on accuracy for a reasonable dimensionality (global and distributed approximation can be at the expense of high parametric dimensionality). Furthermore, the learning of parameters for multilayer NN often involves nonlinear optimization due to composition of multiple nonlinear transformations. All of these methods focus on improving the approximation accuracy by increasing the number of parameters of the models in a brute force manner by increasing the number of basis functions, local models, and/or layers of the network. A key issue arises because if one fixes the architecture and basis functions, the adequacy of the method can be deduced only after the learning process is completed. Adaptation of the approximation architecture, not simply adjusting weights in a fixed architecture, has emerged as the key to convergence reliability and accuracy. Therefore, approximation capabilities of state-of-the-art machine learning approaches (particularly deep learning) in capturing the underlying physical characteristics of a dynamic system remain poorly understood because these algorithms are unable to learn the underlying physical features (or characteristics) of the system.

Presented as Paper 20-585 at the AAS/AIAA Astrodynamics Specialist Conference, South Lake Tahoe, CA, August 9–13, 2020; received 12 February 2021; revision received 10 August 2021; accepted for publication 10 August 2021; published online 19 October 2021.

*Graduate Student, Department of Aerospace Engineering; djg76@psu.edu. Student Member AIAA.

†Professor, Department of Aerospace Engineering; psingla@psu.edu. Associate Fellow AIAA.

‡Assistant Professor, Department of Aerospace Engineering; mmajji@tamu.edu. Senior Member AIAA.

§Professor, Department of Aerospace Engineering; rgmelton@psu.edu. Associate Fellow AIAA.

There is no doubt that the choice of basis functions significantly influences the approximation accuracy and complexity of the model. For many known physical systems, the nonlinearities can be represented by only a few terms with a judicious choice of basis functions. In this respect, many efforts have focused on adapting the architecture of the network by selecting appropriate models from a predefined dictionary of models [25–29]. However, this leads to an exhaustive search algorithm to learn the appropriate basis functions to represent the network dynamics.

More recently, advances in compressed sensing and sparse regression have been exploited to learn appropriate basis functions from an overcomplete dictionary of basis functions without performing an exhaustive search [30–32]. To determine the form of the dynamics from data, these papers collect a time history of the state and its derivative sampled at a number of instances in time. In the case where the derivative is not part of the measurement model, they construct state derivative information by finite difference methods, which make derivative calculations susceptible to noise in measurements. After carefully arranging relevant basis functions in a dictionary, a linear least-squares problem is posed to find unknown coefficients of the basis functions. To enforce sparsity, an iterative least-squares problem is solved where the dictionary size is reduced by removing basis functions whose amplitude is lower than a prescribed threshold. Although this sparse representation through the iterative least-squares problem guarantees the balance between model complexity and accuracy, the resulting algorithm is susceptible to noise in state measurement. In [33,34], an approach named "subsampling-based threshold sparse Bayesian regression" (Subtsbr) is presented to accommodate high noise in the measurements for states and state derivatives. A Galerkin formulation that involves projecting the errors on a set of basis functions known as test functions is considered in [35] to avoid estimating time derivatives of the state variables. Although this formulation provides better results in the presence of noise, the choice of test function severely affects the performance of the algorithm. The formulation in [36] considers a direct integral form of the dynamics for first-order systems in conjunction with a regularized ℓ_1 optimization problem to find the appropriate basis functions to approximate the unknown system dynamics.

As an extension of these recent formulations, the main objective of this paper is to consider an integral form of the differential equation to estimate unknown amplitudes of basis functions with only state and input measurements for a first-order system [37]. Rather than a pure integral form considered in [36], a low-pass filter is designed to avoid infinite response at low frequencies or large time intervals. The secondary objective of this work is to generalize this approach for identification of second- and higher-order systems with only position-level measurement data and systems with a control input. Furthermore, the iterative least-squares problem is replaced with an iterative regularized ℓ_1 optimization problem as used in our earlier work on sparse collocation methods for optimal feedback control laws [38]. This guarantees that the sparse solution is found with high probability using convex optimization methods. The methodology is validated by considering two nonlinear oscillators with or without noisy measurements and on a second-order system involving a central force field. Comparison between a deep-learning approach and a sparse solution is presented at the end of the paper.

The structure of the paper is as follows: Section II provides a mathematical treatment of the system identification problem, and Sec. III provides the derivation of the developed methodology for first- and second-order systems. Section IV shows the efficacy of the developed approach by considering two nonlinear oscillator problems and identification of Newton's law of gravitation through satellite motion data. The paper concludes with a summary of results in Sec. V and generalization of the developed approach for the generic order systems in the Appendix.

II. Problem Statement

This work aims to provide an extended, unified, and automatic framework to discover governing equations underlying a dynamic system simply from data measurements, based on the assumption that the structure of the dynamic model is governed by only a few important terms. Considering a general dynamic model with affine control input

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t)) + G\boldsymbol{u}(t) \tag{1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$ represents the state of the system and $\boldsymbol{u}(t) \in \mathbb{R}^r$ the control action at time t and $G \in \mathbb{R}^{n\times r}$ is the constant input-influence matrix. The unknown nonlinear function $\boldsymbol{f}:\mathbb{R}^n \to \mathbb{R}^n$ represents the dynamics constraints for the system. The goal here is to find the structure of the unknown function $\boldsymbol{f}$ given the time history of $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$ and constant control influence matrix G. Considering a set of basis functions $\{\phi_i\}_{i=1\ldots\infty}$, $\phi_i:\mathbb{R}^n \to \mathbb{R}$, $\boldsymbol{f}$ can be approximated as a linear combination of these basis functions [2,21]:

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{i=1}^{\infty} \boldsymbol{\alpha}_i \phi_i(\boldsymbol{x}) \tag{2}$$

where $\{\boldsymbol{\alpha}_i\}_{i=1\ldots\infty}$, $\boldsymbol{\alpha}_i \in \mathbb{R}^n$, is a set of unknown coefficients. There are infinitely many choices for basis functions, such as polynomials, trigonometric functions, and radial basis functions. A central difficulty in learning $\boldsymbol{f}$ lies in choosing appropriate basis functions, and the choice of basis functions unfortunately depends on the characteristics of an unknown I/O map. In an appropriate basis, the equations are often sparse in nature and the resulting model is parsimonious; i.e., a very few of $\boldsymbol{\alpha}_i$ are nonzero. It is desired to choose the basis functions that allow $\boldsymbol{f}$ to be represented with as few terms in Eq. (2) as possible [30,31]. In this respect, the summation in Eq. (2) is taken over a finite number of N basis functions:

$$\boldsymbol{f}(\boldsymbol{x}) \approx \sum_{i=1}^{N} \boldsymbol{\alpha}_i \phi_i(\boldsymbol{x}) \tag{3}$$

or equivalently

$$\boldsymbol{f}(\boldsymbol{x}) \approx \boldsymbol{\alpha}^T \boldsymbol{\phi}(\boldsymbol{x}) \tag{4}$$

where $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1 \;\; \boldsymbol{\alpha}_2 \;\cdots\; \boldsymbol{\alpha}_N]^T \in \mathbb{R}^{N\times n}$ and $\boldsymbol{\phi}(\boldsymbol{x}) = [\phi_1(\boldsymbol{x}) \;\; \phi_2(\boldsymbol{x}) \cdots \phi_N(\boldsymbol{x})]^T \in \mathbb{R}^N$. Here, $\boldsymbol{\alpha}_i$ is the ith row of the matrix $\boldsymbol{\alpha}$. The objective is to search a given handbook of known functions for a set that best represents the given data. Recent advances in compressed sensing and sparse regression [30,31,36,39] can be exploited to learn these few nonzero terms from an overcomplete dictionary of basis functions without performing a combinatorially intractable brute-force search. The next section provides the mathematical details corresponding to finding a sparse solution for $\boldsymbol{\alpha}$.

III. General Methodology

As stated in the previous section, the objective is to find a sparse solution for the $\boldsymbol{\alpha}_i's$ given the time histories of $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$. If the time history of $\dot{\boldsymbol{x}}(t)$ is known, then one can solve for the unknown coefficients $\boldsymbol{\alpha}_i's$ through a least-squares solution. In Refs. [30–34], different iterative algorithms are proposed to find the best set of basis functions to represent $\boldsymbol{f}$ accurately. If the time derivative of $\boldsymbol{x}(t)$ is not available, then one needs to reconstruct this information through time history knowledge of $\boldsymbol{x}(t)$ via finite difference. Such an approach is sensitive to noise in measurements of $\boldsymbol{x}(t)$ [30]. An integral formulation is considered in [36] to avoid the finite difference to reconstruct the state time derivative information. Though an integral formulation attenuates the high-frequency content, it provides limited attenuation at low frequencies and can lead to infinite response for a long time integration. In this section, an alternate formulation is presented to find unknown $\boldsymbol{\alpha}_i's$ without any knowledge of $\dot{\boldsymbol{x}}(t)$. A low-pass filter design is considered to provide short memory and better control over attenuation at different frequencies. First, this formulation is presented for the first-order systems and then generalized for second-order systems.

A. First-Order Systems

Considering a first-order system, Eq. (1) can be rewritten as

$$\dot{\boldsymbol{x}}(t) = \sum_{i=1}^{N} \boldsymbol{\alpha}_i \phi_i(\boldsymbol{x}) + G\boldsymbol{u}(t) \tag{5}$$

and its componentwise unilateral Laplace transformation is

$$X_j^{0\mathrm{f}}(s) = sX_j(s) - x_j(0) = \sum_{i=1}^{N} \alpha_{j,i}\Phi_i(s) + \sum_{k=1}^{r} G_{j,k}U_k(s), \quad j = 1 \ldots n \tag{6}$$

where $\mathcal{L}\{x_j(t)\} = X_j(s)$, $\mathcal{L}\{\phi_i(\boldsymbol{x})\} = \Phi_i(s)$, $\mathcal{L}\{u_k(t)\} = U_k(s)$, and s is the Laplace variable. Note that $\phi_i(\boldsymbol{x})$ is an implicit function of time, and hence the Laplace transform of this time-varying signal can be considered. From now on, capital letters are used for functions in the Laplace domain. $X_j^{0\mathrm{f}}(s)$ is the original filtered signal (filtered 0th time). For $\lambda_1 \in \mathbb{R}_+^*$, let us consider the Laplace filtering operator

$$\mathcal{I}_{\lambda_1}\colon \mathbb{R} \to \mathbb{R}, \quad \bullet \mapsto \frac{\bullet}{s + \lambda_1} \tag{7}$$

Applying the operator to the signal $X_j^{0\mathrm{f}}$ yields

$$X_j^{1\mathrm{f}}(s) = \mathcal{I}_{\lambda_1}(X_j^{0\mathrm{f}}(s)) = \frac{sX_j(s) - x_j(0)}{s + \lambda_1} = \sum_{i=1}^{N} \alpha_{i,j}\Phi_i^{1\mathrm{f}}(s) + \sum_{k=1}^{r} G_{j,k}U_k^{1\mathrm{f}}(s) \tag{8}$$

where

$$\Phi_i^{1\mathrm{f}}(s) = \frac{\Phi_i(s)}{s + \lambda_1} \quad \text{and} \quad U_k^{1\mathrm{f}}(s) = \frac{U_k(s)}{s + \lambda_1} \tag{9}$$

The superscript "1f" corresponds to the filtered signal. Note that Eq. (8) corresponds to the integral form of Eq. (1) of $\lambda_1 = 0$. To this end, the filtered integral formulation is a generalization of the pure integral approach that can have an infinite response either at very low frequency or for integration over a long time. The use of the filter allows the analyst to introduce fading memory in the approximation process, which allows specific control of the signal-to-noise ratio of the signals used in the identification process. Note that this section describes a method that uses a filter to implement a system that approximates the derivatives at low frequencies. This filter realization, however, could be generalized with larger degrees of design freedom. For instance, writing $\phi^{1\mathrm{f}} = F(s)\phi(\boldsymbol{x}(t))$ with $F(s)$ a general strictly passive real filter of arbitrary order would allow one to extract the desirable signal properties of importance to the physics of the problem. Though the development presented in this paper assumes λ_1 to be a scalar quantity, one can ideally use different filters for different components of the state vector, with λ_1 being a vector quantity. Now, adding and subtracting $\lambda_1 X_j(s)$ to Eq. (8) leads to

$$X_j^{1\mathrm{f}}(s) = \frac{(s + \lambda_1)X_j(s) - \lambda_1 X_j(s) - x_j(0)}{s + \lambda_1} = X_j(s) + \frac{-\lambda_1 X_j(s) - x_j(0)}{s + \lambda_1} = X_j(s) + Y_{j,1}(s) \tag{10}$$

where

$$Y_{j,1}(s) = \frac{-\lambda_1 X_j(s) - x_j(0)}{(s + \lambda_1)} \tag{11}$$

Equation (11) can be rewritten as

$$sY_{j,1}(s) + x_j(0) = -\lambda_1 Y_{j,1}(s) - \lambda_1 X_j(s) \tag{12}$$

and its inverse Laplace transform yields

$$\dot{y}_{j,1}(t) = -\lambda_1 y_{j,1}(t) - \lambda_1 x_j(t), \qquad y_{j,1}(0) = -x(0) \tag{13}$$

Similarly, the inverse Laplace transforms for Φ_i^{1f} and U_k^{1f} yield the corresponding first-order differential equations (ODEs):

$$\dot{\phi}_i^{1f}(t) = -\lambda_1 \phi_i^{1f}(t) + \phi_i(\boldsymbol{x}), \quad \phi_i^{1f}(0) = 0, \quad i = 1, 2, \ldots, N \tag{14}$$

$$\dot{u}_k^{1f}(t) = -\lambda_1 u_k^{1f}(t) + u_k(t), \quad u_k^{1f}(0) = 0, \quad k = 1, 2, \ldots, r \tag{15}$$

For $\lambda_1 = 0$, then ϕ_i^{1f} and u_k^{1f} result in the time integration of $\phi_i(x(t))$ and $u_k(t)$. For $\lambda_1 > 0$, these equations correspond to a stable linear system of equations. By appropriately choosing the λ_1, one can control how quickly the initial condition response of these equations will go to zero. Finally, the final equation in the time domain can be written as

$$x_j^{1f}(t) = x_j(t) + y_{j,1}(t) = \sum_{i=1}^{N} \alpha_{j,i}\phi_i^{1f}(t) + \sum_{k=1}^{r} G_{j,k}u_k^{1f}(t) \tag{16}$$

Note that the aforementioned equation provides a linear relationship between filtered signals $x_j^{1f}(t)$, $u_k^{1f}(t)$, and $\phi_i^{1f}(t)$. Furthermore, these filtered signals can be constructed directly from the given time histories of system state and control input by integrating $N + r + 1$ equations given by Eqs. (13–15). Stacking time histories for $x_j^{1f}(t)$, $\phi_i^{1f}(t)$, and $u_k^{1f}(t)$ leads to

$$\boldsymbol{x}_j^{1f} = \boldsymbol{\phi}^{1fT}\boldsymbol{\alpha}_j + (G_j\boldsymbol{u}^{1f})^T \tag{17}$$

where $\boldsymbol{x}_j^{1f} \in \mathbb{R}^{l\times 1}$, $\boldsymbol{\phi}^{1f} \in \mathbb{R}^{N\times l}$, $\boldsymbol{\alpha}_j \in \mathbb{R}^{N\times 1}$, $G_j \in \mathbb{R}^{1\times r}$, and $\boldsymbol{u}^{1f} \in \mathbb{R}^{r\times l}$, with l being the number of data points. In this equation, $\boldsymbol{\alpha}_j$ is the jth column of the coefficient matrix $\boldsymbol{\alpha}$ introduced in Eq. (4), and G_j is the jth row of the coefficient matrix G introduced in Eq. (1). Now, one can find an optimal value of coefficient vector $\boldsymbol{\alpha}_j$ by solving the weighted two-norm minimization:

$$\boldsymbol{\alpha}_j^{\star} = \min_{\boldsymbol{\alpha}_j} \frac{1}{2}\boldsymbol{e}^T R\boldsymbol{e}, \qquad \boldsymbol{e} = \boldsymbol{x}_j^{1f} - \boldsymbol{\phi}^{1fT}\boldsymbol{\alpha}_j - (G_j\boldsymbol{u}^{1f})^T \tag{18}$$

The weight matrix R can be chosen appropriately depending upon the noise in the measurement data. Depending upon the size of l and N, the aforementioned optimization problem can be overdetermined or underdetermined. In both the cases, one can find the solution with an appropriate pseudo-inverse of $\boldsymbol{\phi}^{1f}$, i.e.,

$$\boldsymbol{\alpha}_j^{\star T} = (\boldsymbol{x}_j^{1fT} - G_j\boldsymbol{u}^{1f})\boldsymbol{\phi}^{1f\dagger} = \tilde{\boldsymbol{x}}^{1f}\boldsymbol{\phi}^{1f\dagger} \tag{19}$$

where $\tilde{\boldsymbol{x}}_j^{1f} = \boldsymbol{x}_j^{1fT} - G_j\boldsymbol{u}^{1f}$ and "†" stands for the pseudo-inverse; $\boldsymbol{\phi}^{1f\dagger}$ corresponds to the least-squares solution for the overdetermined problem, i.e., $\boldsymbol{\phi}^{1f\dagger} = (\boldsymbol{\phi}^{1fT}R\boldsymbol{\phi}^{1f})^{-1}\boldsymbol{\phi}^{1fT}R$, whereas $\boldsymbol{\phi}^{1f\dagger}$ corresponds to the minimum-norm solution for the underdetermined case, i.e., $\boldsymbol{\phi}^{1f\dagger} = \boldsymbol{\phi}^{1fT}R(\boldsymbol{\phi}^{1f}R\boldsymbol{\phi}^{1fT})^{-1}$. This procedure is repeated n times (for $j = 1, 2, \ldots, n$) to compute the full coefficient matrix $\boldsymbol{\alpha}$. Note that one can also compute an estimate for the control influence matrix G through this procedure given that it also appears linearly in Eq. (17).

Equation (19) is a minimization problem obtained by choosing certain collocation points. The choice of the collocation points typically interferes with the filter parameter choices, and one should pick the collocation points judiciously such that their spectral characteristics are not coincident. This is because if they are, the filtered states result in a null solution for the system of equations. In this paper, a time-uniform distribution is chosen so that the spectral

characteristics of the physics of interest are captured for each problem.

The $\boldsymbol{\alpha}_j^\star$ corresponds to the optimal solution in terms of minimizing the two-norm of state output error response. However, the two-norm solution is not guaranteed to be sparse in nature and is known to pick all the basis functions in our dictionary especially in the case of noise-corrupted measurements. To enforce sparsity, ideally the ℓ_0 norm of the coefficient vector $\boldsymbol{\alpha}_j$ needs to be minimized subject to constraints of Eq. (17). The ℓ_0 norm corresponds to the cardinality of the coefficient vector, and its minimization leads to a nonconvex problem. However, the ℓ_0 norm minimization problem can be approximated by an iterative ℓ_1-norm minimization problem, which is convex in nature with a guaranteed solution [39]:

$$\min_{\boldsymbol{\theta}_j^p} \| W^p \boldsymbol{\theta}_j^p \|_1 \tag{20}$$

$$\text{s.t. } \|\tilde{\boldsymbol{x}}^{1f} - \boldsymbol{\theta}_j^{pT} \boldsymbol{\phi}^{1f}\|_2 \leq \varepsilon \|\tilde{\boldsymbol{x}}^{1f} - \boldsymbol{\alpha}_j^{\star T} \boldsymbol{\phi}^{1f}\|_2, \qquad \varepsilon \geq 1 \tag{21}$$

where p is the iteration, $\boldsymbol{\theta}_j^p$ is the optimization variable, $\tilde{\boldsymbol{x}}^{1f}$ is the pseudosignal, $\boldsymbol{\phi}^{1f}$ is the dictionary of basis functions, and $\boldsymbol{\alpha}_j^{\star T}$ is the optimal two-norm solution derived in the previous section. Notice that the two-norm constraint of Eq. (21) corresponds to the satisfaction of Eq. (17). Rather than using the equality constraint of Eq. (17), a two-norm error is bounded by the optimal pseudonorm solution, with ε being the user-defined relaxation on two-norm error. This allows one to tradeoff sparsity with approximation error. Furthermore, W^p is a diagonal matrix containing a known weight w_i for the ith optimization variable. Initially, w_i can be chosen based upon any a priori knowledge about the structure of $\boldsymbol{f}$, the form of the least-squares solution, or can simply be chosen to be one. In the subsequent iterations, the value of w_i is adapted according to the following formula to penalize the coefficients that are smaller than a predefined threshold δ:

$$w_i^p = \frac{1}{|\theta_{i,j}^{p-1}| + \eta} \tag{22}$$

where η is a small number to avoid division by zero. This iterative procedure is repeated unless the computed coefficients converge within a prescribed tolerance. The solution of this iterative ℓ_1 minimization problem provides us a subset of basis functions from an overcomplete dictionary, which plays a dominant role in underlying unknown dynamics. An optimal pseudo-inverse solution for coefficients is obtained for only this subset of basis functions. Figure 1 illustrates these steps to obtain the sparse solution. The algorithm is given in Algorithm 1. Note that the parameter δ acts as a threshold to separate the active basis function from the nonactive ones. In practice, with normalized trajectories and independently of the dynamic system considered, it is a relatively safe assumption to neglect the least dominant basis function, i.e., the basis functions with associated coefficient two to three orders of magnitude smaller than other basis function coefficients. In this respect, δ is usually chosen to be at least one order of magnitude smaller than the least dominant basis function coefficient. For noisy I/O data, the value of δ is chosen based upon the signal-to-noise ratio to avoid overfitting the data. A good discussion on the choice of these hyperparameters on approximation accuracy has been provided in [39].

Algorithm 1: Iterative sparse algorithm for model selection

function SparseID (δ, η, ε, $\text{nb}_{\text{iterations}}$, $\boldsymbol{\phi}^{1f}$, $\tilde{\boldsymbol{x}}^{1f}$, $\boldsymbol{\alpha}_j^\star$)
  Initialize weighting matrix $W = I_N$
  Initialize $p = 0$
  while $p < \text{nb}_{\text{iterations}}$ **do**
    Solve for $\boldsymbol{\theta}_j^p$: $\min_{\boldsymbol{\theta}_j^p} \| W^p \boldsymbol{\theta}_j^p \|_1$ such that $\|\tilde{\boldsymbol{x}}^{1f} - \boldsymbol{\theta}_j^{pT} \boldsymbol{\phi}^{1f}\|_2 \leq \varepsilon \|\tilde{\boldsymbol{x}}^{1f} - \boldsymbol{\alpha}_j^{\star T} \boldsymbol{\phi}^{1f}\|_2$
    Update weighting matrix: $W[i,i] = \frac{1}{|\theta_{i,j}^{p-1}| + \eta}$
  end while
  $\boldsymbol{\theta}_j = \boldsymbol{\theta}_j^p$
  for $i = 1$ **to** N **do**
    if $\boldsymbol{\theta}_j[i] < \delta$ **then**
      $\boldsymbol{\theta}_j[i] = 0$
      Remove column $\boldsymbol{\phi}_i^{1f}$ from dictionary $\boldsymbol{\phi}^{1f}$
    end if
  end for
  Compute final least-squares solution with updated dictionary: $\boldsymbol{\alpha}_j^{*T} = \tilde{\boldsymbol{x}}^{1f} \boldsymbol{\phi}^{1f\dagger}$
end function

B. Second-Order Systems

Derived from the rates of generalized momenta at the acceleration level, most dynamic systems in engineering mechanics are characterized by second-order differential equations. If both position and velocity measurements are available, the second-order differential equation can be reshaped as a first-order differential system and the methodology presented in the last section can be applied. In this section, the methodology presented in the last section is generalized for the identification of the second-order system with time histories of position-level measurements and control input vectors being available. Consider the following special class of second-order nonlinear dynamic system with nonlinearities being a function of only position-level state variables:

$$\ddot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t)) + G\boldsymbol{u}(t) \tag{23}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$ represents the state of the system and $\boldsymbol{u}(t) \in \mathbb{R}^r$ the control action at time t and $G \in \mathbb{R}^{n \times r}$ is the constant-time input influence matrix. Once again, the function $\boldsymbol{f}$:$\mathbb{R}^n \to \mathbb{R}^n$ represents the dynamics constraints that define the equations of motion of the system, and our objective is to identify this nonlinear function from time histories of $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$. Following the same development as before, the unknown nonlinear function $\boldsymbol{f}$ can be expanded in terms of a dictionary of basis functions, and Eq. (23) can be rewritten as

$$\ddot{\boldsymbol{x}}(t) = \sum_{i=1}^{N} \boldsymbol{\alpha}_i \phi_i(\boldsymbol{x}) + G\boldsymbol{u}(t) \tag{24}$$

In a quest to determine the analog of Eq. (17), the componentwise Laplace transform of the aforementioned vector equation leads to

$$X_j^{0f}(s) = s^2 X_j(s) - s x_j(0) - \dot{x}_j(0) = \sum_{i=1}^{N} \alpha_{j,i} \Phi_i(s) + \sum_{k=1}^{r} G_{j,k} U_k(s), \qquad j = 1 \ldots n \tag{25}$$

Now, applying the integral operator $\mathcal{I}_{\lambda_1}$ to the original signal X_j^{0f} yields

$$X_j^{1f}(s) = \mathcal{I}_{\lambda_1}(X_j^{0f}(s)) = \frac{s^2 X_j(s) - s x_j(0) - \dot{x}_j(0)}{s + \lambda_1} = \sum_{i=1}^{N} \alpha_{j,i} \frac{\Phi_i(s)}{s + \lambda_1} + \sum_{k=1}^{r} G_{j,k} \frac{U_k(s)}{s + \lambda_1} \tag{26}$$

For $\lambda_2 \in \mathbb{R}_+^*$, applying the integral operator $\mathcal{I}_{\lambda_2}$ to X_j^{1f} of Eq. (26) results in

$$X_j^{2f}(s) = \mathcal{I}_{\lambda_2}(X_j^{1f}(s)) = \frac{s^2 X_j(s) - s x_j(0) - \dot{x}_j(0)}{(s + \lambda_1)(s + \lambda_2)} \tag{27}$$

$$= \sum_{i=1}^{N} \alpha_{j,i} \frac{\Phi_i(s)}{(s + \lambda_1)(s + \lambda_2)} + \sum_{k=1}^{r} G_{j,k} \frac{U_k(s)}{(s + \lambda_1)(s + \lambda_2)} \tag{28}$$

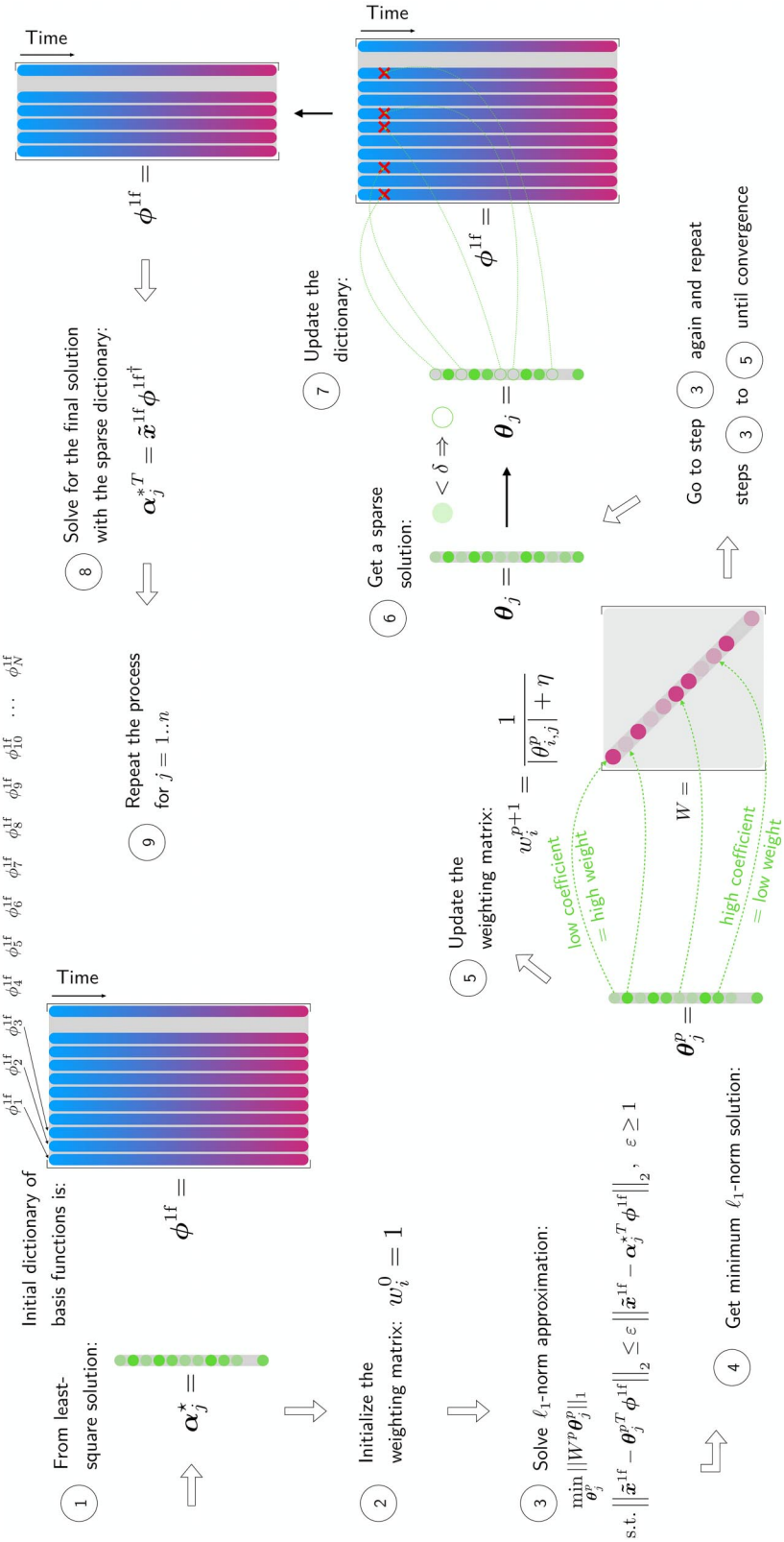


Fig. 1 Illustration of the iterative procedure to derive a sparse solution for a first-order system.

Notice that $\lambda_1 = \lambda_2 = 0$ corresponds to a double-integration of the state vector, and nonzero values for λ_1 and λ_2 help to accommodate for initial condition errors. Finally, with

$$\Phi_i^{2f}(s) = \frac{\Phi_i(s)}{(s+\lambda_1)(s+\lambda_2)} \quad \text{and} \quad U_k^{2f}(s) = \frac{U_k(s)}{(s+\lambda_1)(s+\lambda_2)} \tag{29}$$

the Laplace filtered (twice) version of Eq. (24) is

$$X^{2f}(s) = \sum_{i=1}^{N} \alpha_{i,j} \Phi_i^{2f}(s) + \sum_{k=1}^{r} G_{j,k} U_k^{2f}(s) \tag{30}$$

Here the superscript "2*f*" corresponds to a signal that has been filtered twice. Adding and subtracting $\lambda_1 X_j(s)$ and $\lambda_2 X_j(s)$ leads to

$$X_j^{2f}(s) = X_j(s) - \frac{(\lambda_1+\lambda_2)X_j(s)+x_j(0)}{(s+\lambda_2)} + \frac{\lambda_1^2 X_j(s)+\lambda_1 x_j(0)-\dot{x}_j(0)}{(s+\lambda_1)(s+\lambda_2)} = X_j(s)+Y_{j,1}(s)+Y_{j,2}(s) \quad (31)$$

where

$$Y_{j,1}(s) = \frac{-(\lambda_1+\lambda_2)X_j(s)-x(0)}{(s+\lambda_2)},$$
$$Y_{j,2}(s) = \frac{\lambda_1^2 X_j(s)+\lambda_1 x(0)-\dot{x}(0)}{(s+\lambda_2)(s+\lambda_1)} \quad (32)$$

The inverse Laplace transform of Eq. (32) yields the linear ODEs:

$$\dot{y}_{j,1}(t) = -\lambda_2 y_{j,1}(t) - (\lambda_1+\lambda_2)x_j(t), \qquad y_{j,1}(0) = -x(0) \quad (33)$$

$$\ddot{y}_{j,2}(t) = -(\lambda_1+\lambda_2)\dot{y}_{j,2}(t) - \lambda_1\lambda_2 y_{j,2}(t) + \lambda_1^2 x_j(t),$$
$$y_{j,2}(0) = 0, \qquad \dot{y}_{j,2}(0) = \lambda_1 x(0) - \dot{x}(0) \quad (34)$$

Similarly, the inverse Laplace transforms for Φ_i^{2f} and U_k^{2f} yield the corresponding second-order ODEs:

$$\ddot{\phi}_i^{2f}(t) = -(\lambda_1+\lambda_2)\dot{\phi}_i^{2f}(t) - \lambda_1\lambda_2\phi_i^{2f}(t) + \phi_i(\boldsymbol{x}),$$
$$\phi_i^{2f}(0) = 0 \quad \dot{\phi}_i^{2f}(0) = 0, \quad i = 1, 2, \ldots, N \quad (35)$$

$$\ddot{u}_k^{2f}(t) = -(\lambda_1+\lambda_2)\dot{u}_k^{2f}(t) - \lambda_1\lambda_2 u_k^{2f}(t) + u_k(t),$$
$$u_k^{2f}(0) = 0, \quad \dot{u}_k^{2f}(0) = 0 \quad (36)$$

Finally, the inverse Laplace transformation of Eq. (31) results in the following linear equation in unknown coefficient vector $\boldsymbol{\alpha}_j$:

$$x_j^{2f}(t) = x_j(t) + y_{j,1}(t) + y_{j,2}(t) = \sum_{i=1}^{N}\alpha_{j,i}\phi_i^{2f}(t) + \sum_{k=1}^{r} G_{j,k}u_k^{2f}(t) \quad (37)$$

This is analogous to Eq. (17) for the first-order systems and requires only knowledge of $x_j(t)$ and $u_k(t)$ to find unknown coefficient vector. One can now employ the iterative ℓ_1 solution in conjunction with two-norm minimization to find the appropriate basis functions and their corresponding contributions from a large dictionary of basis functions as discussed in Sec. III.A. For completeness, this procedure is generalized to a generic dth-order system in the Appendix.

IV. Numerical Results

The method described in the previous section to identify governing equations from measurement data is now validated on three examples of different complexity. The first example corresponds to identification of nonlinear dynamics for the Duffing oscillator, whereas the second example corresponds to the identification of the chaotic Lorenz oscillator. The third example corresponds to identification of Newton's law of gravitation by considering the motion of a satellite in an orbit around the Earth. The first two examples correspond to chaotic dynamic systems that show some interesting dynamic behavior, whereas the third problem corresponds to the classical central-force field dynamic model. The first-order system formulation presented in Sec. III.A is used to identify the unknown dynamics for the first two examples, and the second-order formulation presented in Sec. III.B is used for the identification of central force field in the third example with position-only measurements. For the central force field identification, the results are compared with a multilayer NN as used in our prior work [40,41].

A. Duffing Oscillator

The first example aims to legitimize the new approach of this paper with a low-pass-filter-based integral formulation by comparing it with a pure integral formulation for parameter estimation of the nonlinear Duffing oscillator. The Duffing oscillator represents a nonlinear spring–mass–damper system and shows dynamic behaviors of interest for many real engineering applications. The governing equations of motion for the Duffing oscillator are given as

$$\ddot{x} + \gamma\dot{x} + \alpha x + \beta x^3 = u \quad (38)$$

where $\gamma = 0.2$, $\alpha = 1$, $\beta = -1$, and u is a random excitation following a Gaussian distribution with mean 0 and standard deviation 1. With $x_1 = x$ and $x_2 = \dot{x}$ (x_1 is the analogous of a position, x_2 of a velocity), Eq. (38) can be rewritten as the first-order dynamic system:

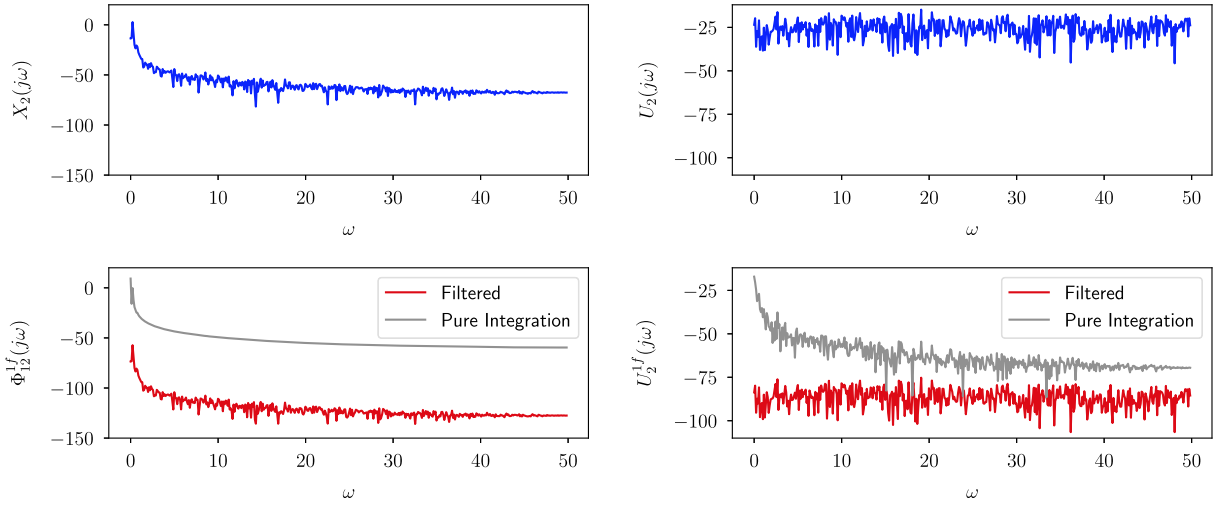
$$\dot{x}_1 = x_2 \quad (39a)$$

$$\dot{x}_2 = -\gamma x_2 - \alpha x_1 + \beta x_1^3 + u \quad (39b)$$

The training dataset for the identification purpose corresponds to response of the system for the initial condition $\boldsymbol{x}(0) = \{1.4, \quad 0\}^T$. The state and control input data are simulated at a frequency of 200 Hz for 10 s. Two test cases are considered: the first test case corresponds to perfect measurements, whereas the second test case corresponds to state measurements being corrupted with zero mean Gaussian white noise with variance of 10^{-4}. The initial dictionary of basis function consists of all monomials up to 10th order in x_1–x_2 space, resulting in a total of 66 basis functions to approximate the unknown system dynamics:

$$\mathbb{R}_{10}[x_1,x_2] = \{1 \quad x_1 \quad x_1^2 \quad x_1^3 \quad x_1^4 \quad x_1^5 \quad x_1^6 \quad x_1^7 \quad x_1^8 \quad x_1^9 \quad x_1^{10} \quad x_2 \quad x_1x_2 \quad x_1^2x_2 \quad x_1^3x_2 \quad x_1^4x_2 \quad x_1^5x_2 \quad x_1^6x_2 \quad x_1^7x_2 \quad x_1^8x_2 \quad x_1^9x_2$$
$$x_2^2 \quad x_1x_2^2 \quad x_1^2x_2^2 \quad x_1^3x_2^2 \quad x_1^4x_2^2 \quad x_1^5x_2^2 \quad x_1^6x_2^2 \quad x_1^7x_2^2 \quad x_1^8x_2^2 \quad x_2^3 \quad x_1x_2^3 \quad x_1^2x_2^3 \quad x_1^3x_2^3 \quad x_1^4x_2^3 \quad x_1^5x_2^3$$
$$x_1^6x_2^3 \quad x_1^7x_2^3 \quad x_2^4 \quad x_1x_2^4 \quad x_1^2x_2^4 \quad x_1^3x_2^4 \quad x_1^4x_2^4 \quad x_1^5x_2^4 \quad x_1^6x_2^4 \quad x_2^5 \quad x_1x_2^5 \quad x_1^2x_2^5 \quad x_1^3x_2^5 \quad x_1^4x_2^5 \quad x_1^5x_2^5$$
$$x_2^6 \quad x_1x_2^6 \quad x_1^2x_2^6 \quad x_1^3x_2^6 \quad x_1^4x_2^6 \quad x_2^7 \quad x_1x_2^7 \quad x_1^2x_2^7 \quad x_1^3x_2^7 \quad x_2^8 \quad x_1x_2^8 \quad x_1^2x_2^8 \quad x_1x_2^9 \quad x_2^{10}\} \quad (40)$$

Two values of λ are investigated in order to compare the low-pass-filter-based integral formulation regression model ($\lambda = 100$) with a pure integration based model ($\lambda = 0$). First, a signal analysis is performed on the fundamental signals used in both regression models, specifically, their spectral signatures. Figure 2a compares the spectral content of the signal that is a direct integration of the state ($\lambda = 0$) with the filtered state ($\lambda = 100$). The signal of interest is Φ_{12}^{1f} corresponding to the filtered signal x_2 at position 12 in the dictionary. While the direct integration loses some of the spectral content as denoted by the smooth Fourier transform, Φ_{12}^{1f} seems to qualitatively share the spectral signature that looks very similar in shape to $X_2(j\omega)$ of Fig. 2a. Similarly, the spectral content of the filtered input matches



a) Input Signal u **b) State x_2**

Fig. 2 Duffing oscillator: spectral signatures of the state x_2 and the input signal and their filtered version.

that of the original broad-band white noise excitation, shown in Fig. 2b. Because the amplitude of the spectral content in the integrated signals gets eroded uniformly across all frequencies because of the smoothness of the integration operation, the pure integration based model fails to capture the I/O relationships at higher frequencies. Although this model can still be used in parameter estimation, the magnitude and frequency content of the inputs should be adjusted to improve the learning process. For the low-pass-filter-based integral formulation regression model, experience of the analyst in choosing the filter time constants enables a better control and conditioning of the approximation problem. Therefore in many signal analysis and adaptive control problems of online learning, the low-pass filter-based model is preferred to a pure-integration-based model. Furthermore, the low-pass filter automatically rejects high-frequency noise in practical applications of interest in systems and control.

The procedure listed in Sec. III.A is adopted to find the unknown coefficient vector. While the least-squares solution is a combination of all the 66 basis functions, the sparse solution accurately identifies the true basis functions participating in the actual dynamics. Tables 1 and 2 present the values of the identified coefficients for both the test cases considered. These identified values appear to be very close to the true ones, which shows the efficacy of the methodology in identifying the true dynamics of the system. Relative error percentages are calculated as

$$\text{Relative Error}\% = \frac{\text{true} - \text{identified}}{\text{true}} \times 100 \tag{41}$$

Figures 3a and 3b display the value of the resulting least-squares and sparse coefficients for both formulations and cases. Although the least-squares solution is the best fit possible in the sense that the square of the error between the true signal and the identified signal is minimized, this results in an overfitting with more basis functions than necessary, particularly true in the presence of noise. Both the pure integral and filtered formulations perform well when the data are not corrupted with any noise. As seen in Fig. 3b, the integral formulation starts picking up basis functions that do not appear in the dynamics, resulting in an overfitting as well. This is because, in addition to the spectral content of the noise perturbation, the true spectral content of the signal has been smoothed out, making it difficult for the algorithm to distinguish the true signal from the noise. Figures 4a and 4c show the error resulting in the propagation for both test cases while using the least-squares solution for the identified dynamics. Though the least-squares solution provides the optimal value of the coefficients while minimizing the two-norm error of the measurement data at discrete time instants, the presence of basis functions that are not participating in the true dynamics leads to overfitting and hence large propagation errors. Notice that the pure-integration-based model leads to an error in propagation of the same order of magnitude as the low-pass-filter-based formulation. On the other hand, the resulting sparse-identified model inherently and automatically balances model complexity with accuracy and results in small absolute error for both the test cases as seen in Figs. 4b and 4d. Here, the low-pass-filter-based formulation is able to preserve the spectral content of the original signals used in the regression process and differentiate that content from the white noise spectral signature, leading to the right selection of the basis functions. Finally, Table 3 presents the root-mean-square (RMS) errors for 10 random initial conditions generated from Gaussian distribution with mean $\boldsymbol{x}(0) = \{1.4, 0\}^T$ and covariance $P_0 = I_2$. Once again, as expected, the sparse-approximation-based identified system leads to minimal RMS error even in the presence of noise. This better performance of the sparse approximation method can be attributed to its ability to identify the inherent true dynamics of the system.

Setting $\lambda = 0$, the direct integration regression model is obtained in our approach. The integral equation transformation is a special case of what is implemented in this paper. The first takeaway is that the integration operation uniformly suppresses the frequency content in the signals involved in the approximation process, meaning that some key spectral content at moderate frequencies will also be smoothed out by the direct integration. This mandates the direct integration process to have input signals (for training) with low frequency and larger amplitudes. Of course when the right-hand side becomes a nonlinear function of the states, this problem becomes compounded. Another byproduct of this method is that the learning process continues even when the system is turned off ($u = 0$) and the filtered integration process preserves the spectral content of the

Table 1 Value of the coefficients for the sparse solution vs the true coefficients: $\dot{x}_1$ equation

Value of λ	Basis function	No. in the dictionary	True value	Sparse value (no noise)	Sparse value (with noise)	Rel. error % (no noise)	Rel. error % (with noise)
0	$x_1, x_2 \mapsto x_2$	12	1	0.99999964	1.000014	$3.6 \cdot 10^{-5}$	$-1.4 \cdot 10^{-3}$
100	$x_1, x_2 \mapsto x_2$	12	1	0.999999984	0.99999971	$1.6 \cdot 10^{-6}$	$2.9 \cdot 10^{-5}$

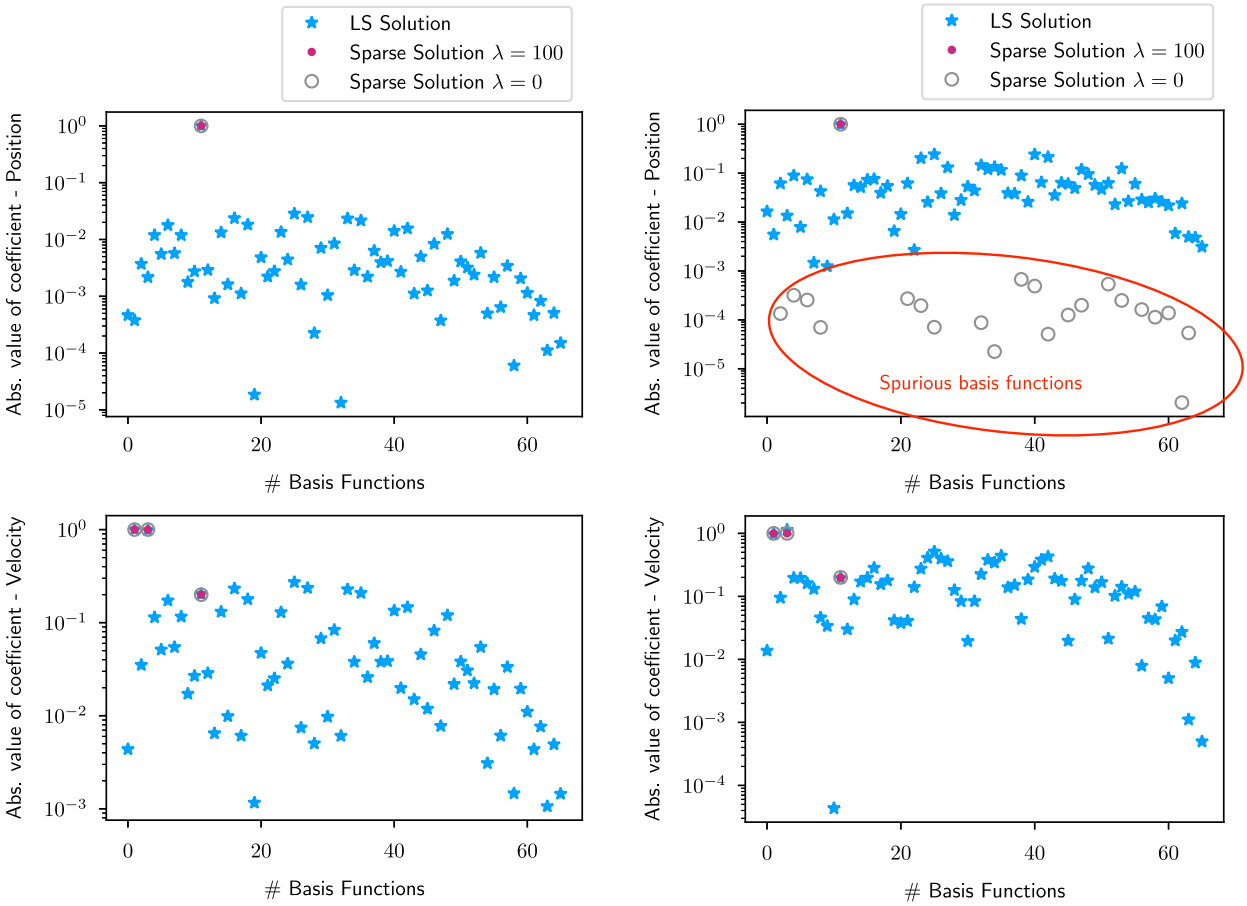


Fig. 3 Duffing Oscillator: value of the coefficients for the basis functions.

original signals used in the regression process because a fading memory is implemented in the filtered integration process. The data are thus exponentially deweighted and the learning stops as soon as excitation stops.

B. Lorenz System

The second example corresponds to the chaotic Lorenz oscillator with governing dynamic equations given as

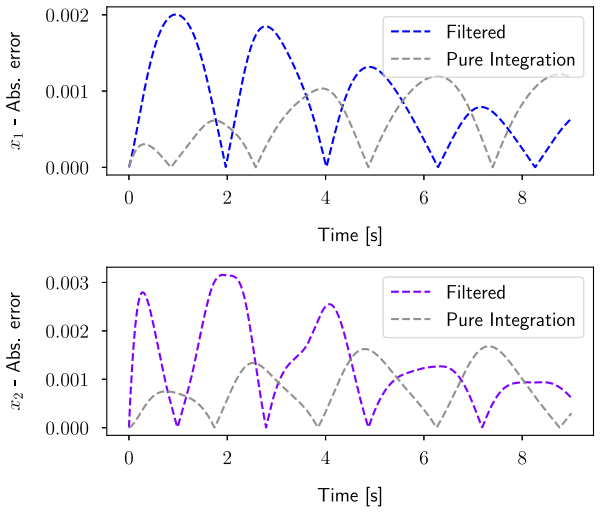
$$\dot{x} = \sigma(y - x) \quad \dot{y} = x(\rho - z) - y \quad \dot{z} = xy - \beta z \tag{42}$$

The training data set corresponds to $\sigma = 10$, $\rho = 28$, $\beta = 8/3$, and an initial condition of $\boldsymbol{x}(0) = \{-8, 7, 27\}^T$. The data are recorded at a frequency of 500 Hz for 4 s. A value of $\lambda_1 = 10$ is chosen for the low-pass filter. Once again two test cases (with random initial condition, different from the ones used to generate the training data set) are considered, with the first test case corresponding to perfect measurements and the second test case corresponding to measurements being corrupted by zero mean Gaussian white noise of variance 10^{-4}. The initial dictionary of basis functions consists of a total of 56 polynomial basis functions up to degree 5 in state variables.

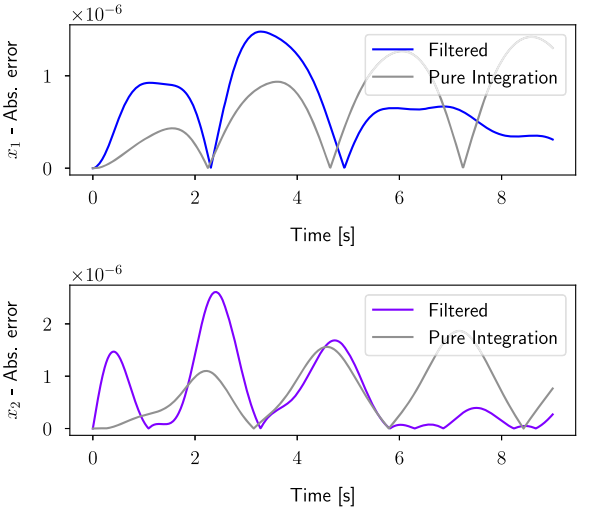
Once again, the procedure listed in Sec. III.A is used to find the unknown coefficient vector. Figures 5a and 5b show the least-squares as well as sparse solution for coefficients for both test cases. While the sparse solution correctly identifies the correct basis functions, the least-squares fits nonzero amplitude for most of the basis functions. The iterative procedure converges within three iterations to accurately identify the participating basis functions and drives coefficients of nonparticipating basis functions to zero. Tables 4–6 present the values of the identified coefficients for both test cases. These identified values appear to be very close to the true ones, which shows the efficacy of the developed methodology in identifying the true dynamics of the system.

Figures 6a–6d show the error resulting in the propagation for both test cases while using the least-squares as well as sparse identification of inherent dynamics. Observe that the amplitude of the error is correlated to the dynamics. Regions of the phase space associated with high velocity (occurs at a lobe switching) relate with larger errors, especially in the presence of noise (see Figs. 6c and 6d, where there is a lobe switching at $t \simeq 2$ s). Finally, Table 7 presents the RMS errors for 10 random initial conditions generated from Gaussian distribution with mean $\boldsymbol{x}(0) = \{-8, 7, 27\}^T$ and covariance $P_0 = 0.5I_3$. From these results, it is clear that the proposed sparse approximation solution leads to an order of magnitude improvement

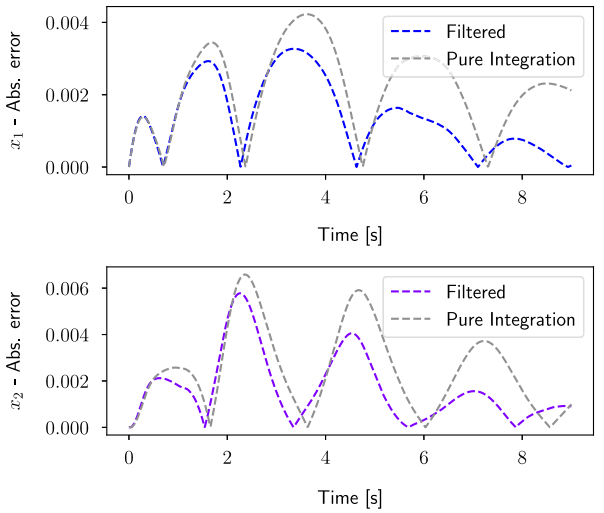
$$\begin{aligned}\mathbb{R}_5[x,y,z] = \Big\{ & 1 \quad x \quad x^2 \quad x^3 \quad x^4 \quad x^5 \quad y \quad xy \quad x^2y \quad x^3y \quad x^4y \quad y^2 \quad xy^2 \quad x^2y^2 \quad x^3y^2 \quad y^3 \quad xy^3 \quad x^2y^3 \quad y^4 \quad xy^4 \quad y^5 \\ & z \quad xz \quad x^2z \quad x^3z \quad x^4z \quad yz \quad xyz \quad x^2yz \quad x^3yz \quad y^2z \quad xy^2z \quad x^2y^2z \quad y^3z \quad xy^3z \quad y^4z \quad z^2 \quad xz^2 \quad x^2z^2 \\ & x^3z^2 \quad yz^2 \quad xyz^2 \quad x^2yz^2 \quad y^2z^2 \quad xy^2z^2 \quad y^3z^2 \quad z^3 \quad xz^3 \quad x^2z^3 \quad yz^3 \quad xyz^3 \quad y^2z^3 \quad z^4 \quad xz^4 \quad yz^4 \quad z^5 \Big\}\end{aligned} \tag{43}$$



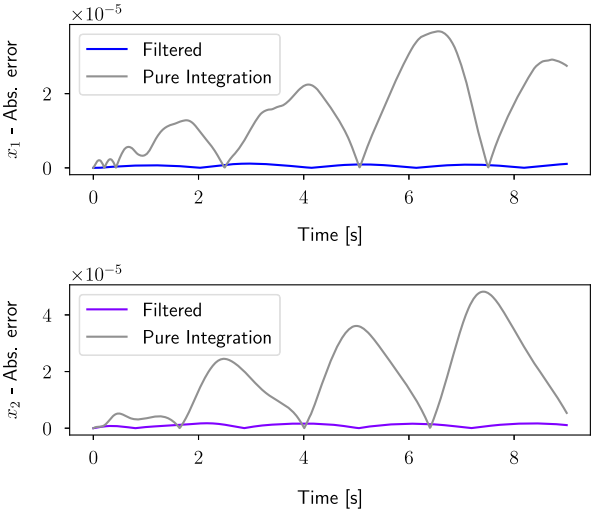
a) No Noise case: LS



b) No Noise case: Sparse



c) Noise case: LS



d) Noise case: Sparse

Fig. 4 Duffing oscillator: absolute relative error.

Table 2 Value of the coefficients for the sparse solution vs the true coefficients: $\dot{x}_2$ equation

Value of λ	Basis function	No in the dictionary	True value	Sparse value (no noise)	Sparse value (with noise)	Rel. error % (no noise)	Rel. error % (with noise)
0	$x_1, x_2 \mapsto x_1$	2	−1	−1.00000003	−1.000011	$-3.0 \cdot 10^{-6}$	$-1.1 \cdot 10^{-3}$
0	$x_1, x_2 \mapsto x_1^3$	4	−1	−0.99999996	−0.999988	$4.0 \cdot 10^{-6}$	$1.2 \cdot 10^{-3}$
0	$x_1, x_2 \mapsto x_2$	12	−0.2	−0.20000003	−0.2000060	$-1.5 \cdot 10^{-5}$	$-3.0 \cdot 10^{-3}$
100	$x_1, x_2 \mapsto x_1$	2	−1	−1.0000004	−1.0000016	$-4.0 \cdot 10^{-5}$	$-1.6 \cdot 10^{-4}$
100	$x_1, x_2 \mapsto x_1^3$	4	−1	−0.9999995	−0.9999974	$5.0 \cdot 10^{-5}$	$2.6 \cdot 10^{-4}$
100	$x_1, x_2 \mapsto x_2$	12	−0.2	−0.200000008	−0.1999995	$4.0 \cdot 10^{-6}$	$2.5 \cdot 10^{-4}$

in state propagation errors. It should be noted that this approach provides 10^{-4}–10^{-7}% error as compared with an accuracy of 0.03% as reported in [30,31] for the same level of noise but at a higher sampling frequency of 1000 Hz. This better performance of the sparse approximation method can be attributed to its ability to identify the inherent true dynamics of the system.

Table 3 Duffing oscillator: RMS error on 10 random initial conditions

	$\lambda = 0$		$\lambda = 100$	
Solution	No noise case	Noise case	No noise case	Noise case
Least-squares	$7.41 \cdot 10^{-4}$	$4.02 \cdot 10^{-3}$	$8.56 \cdot 10^{-4}$	$3.41 \cdot 10^{-3}$
Sparse	$9.77 \cdot 10^{-7}$	$3.45 \cdot 10^{-5}$	$9.02 \cdot 10^{-7}$	$1.45 \cdot 10^{-6}$

C. Two-Body Keplerian Dynamics

The third example corresponds to the identification of the inverse square law of gravitation from the observation data corresponding to a satellite orbiting the Earth. Kepler's laws, Newton's laws of motion, and Newton's gravitational law were developed with critical reliance on observational data. Based on Giuseppe Piazzi's observations of Ceres in 1801, Gauss calculated the orbit of Ceres from the observation data for Ceres using the theory of least-squares and initiated the

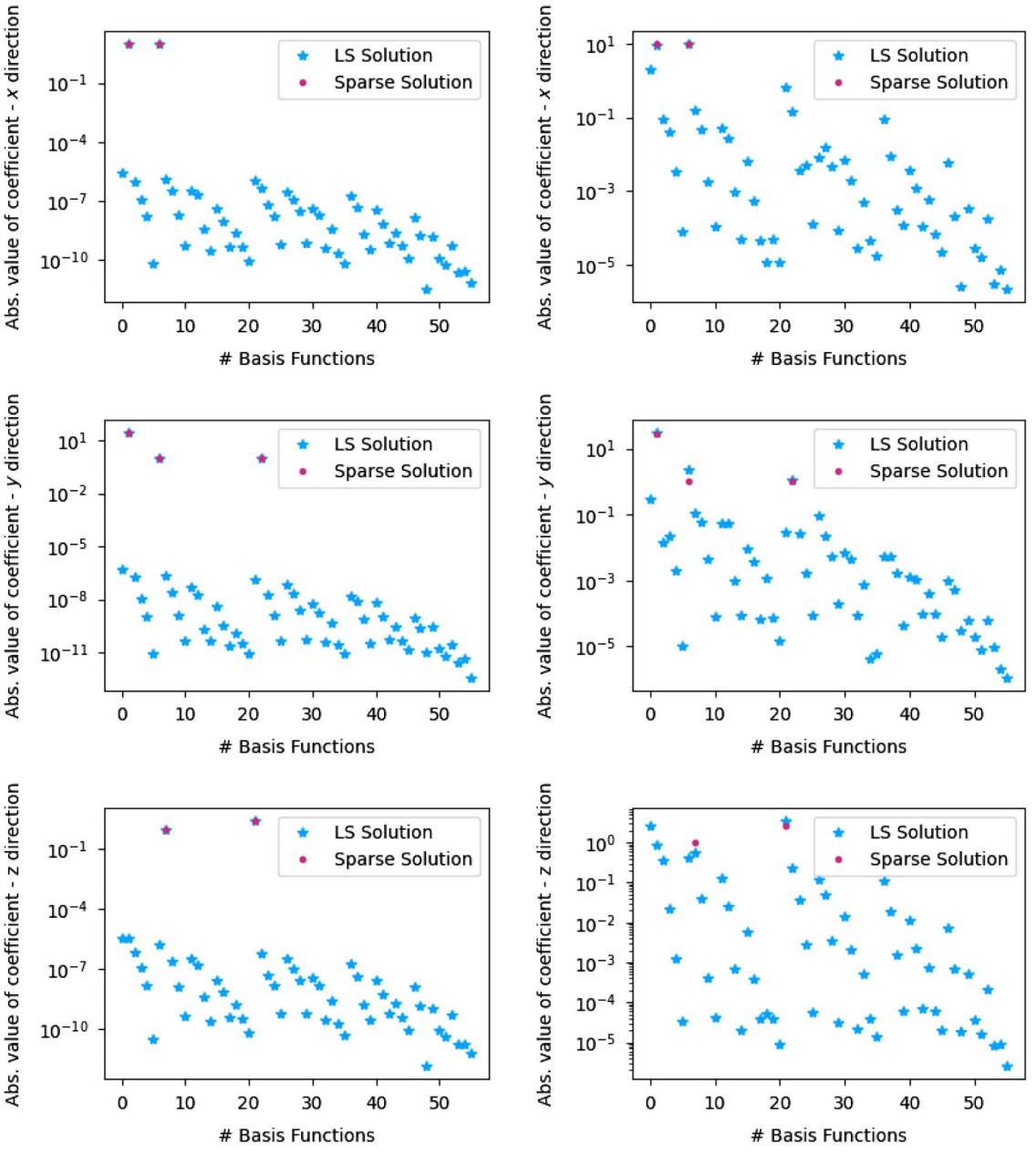


Fig. 5 Lorenz problem: value of the coefficients for the basis functions.

Table 4 Value of the coefficients for the sparse solution vs the true coefficients: *x* direction

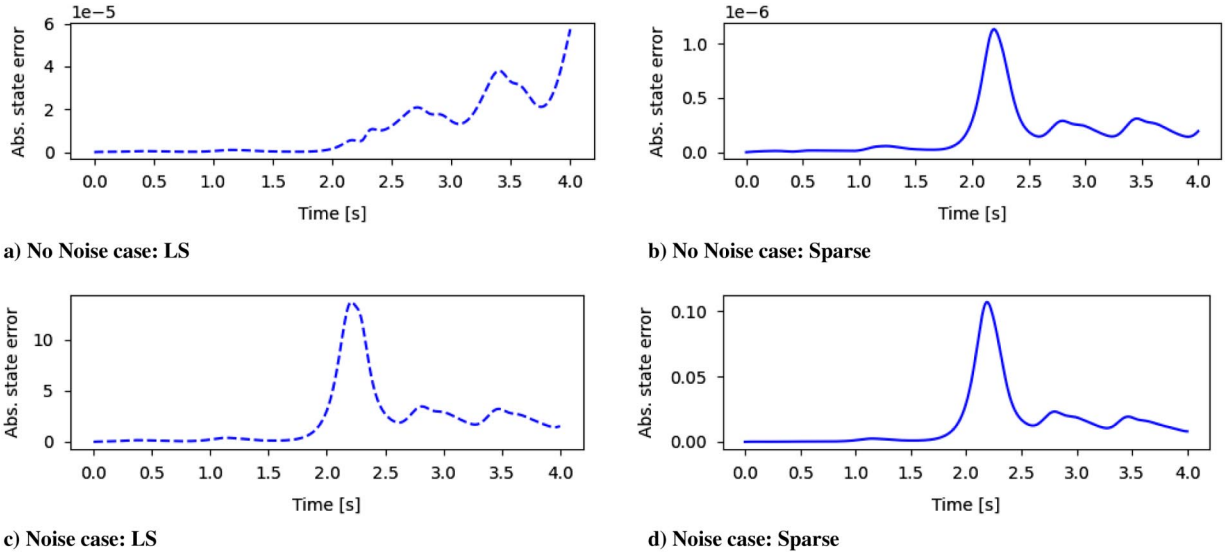
Basis function	No. in the dictionary	True value	Sparse solution (no noise)	Sparse solution (with noise)	Relative error % (no noise)	Relative error % (with noise)
$x, y, z \mapsto x$	2	−10	−10.00000000035	−9.9999949	$-3.54 \cdot 10^{-9}$	$5.10 \cdot 10^{-7}$
$x, y, z \mapsto y$	7	10	10.00000000030	10.0000333	$-3.02 \cdot 10^{-9}$	$-3.33 \cdot 10^{-6}$

Table 5 Value of the coefficients for the sparse solution vs the true coefficients: *y* direction

Basis function	No. in the dictionary	True value	Sparse solution (no noise)	Sparse solution (with noise)	Relative error % (no noise)	Relative error % (with noise)
$x, y, z \mapsto x$	2	28	28.000000038	28.0008596	$-1.34 \cdot 10^{-7}$	$-3.07 \cdot 10^{-5}$
$x, y, z \mapsto y$	7	−1	1.0000000053	−1.0002722	$-5.33 \cdot 10^{-7}$	$-2.72 \cdot 10^{-4}$
$x, y, z \mapsto xz$	23	−1	−1.0000000012	−1.0000214	$-1.19 \cdot 10^{-7}$	$-2.14 \cdot 10^{-5}$

Table 6 Value of the coefficients for the sparse solution vs the true coefficients: z direction

Basis function	No. in the dictionary	True value	Sparse solution (no noise)	Sparse solution (with noise)	Relative error % (no noise)	Relative error % (with noise)
$x, y, z \mapsto xy$	8	1	1.00000000058	0.9999914	$-5.81 \cdot 10^{-8}$	$8.60 \cdot 10^{-6}$
$x, y, z \mapsto z$	22	$-8/3$	2.66666666776	−2.6666329	$-4.11 \cdot 10^{-8}$	$1.27 \cdot 10^{-5}$



a) No Noise case: LS b) No Noise case: Sparse

c) Noise case: LS d) Noise case: Sparse

Fig. 6 Lorenz problem: absolute relative error.

Table 7 Lorenz problem: RMS error on 10 random initial conditions

Solution	No noise case	Noise case
Least-squares	$9.369 \cdot 10^{-6}$	1.418
Sparse	$1.704 \cdot 10^{-7}$	$3.117 \cdot 10^{-3}$

Table 8 Orbital elements for the four considered orbits of the training data set

Orbital elements	ISS	Molniya	Polar	GEO
Semimajor axis a, m	6,789,500	26,600,000	9,240,000	42,164,000
Eccentricity e	0.0001912	0.74	0.00025	0
Inclination i, deg	51.6414	63.4	89.8	0.01
RAAN Ω, deg	259.0449	128	120	120
Argument of perigee ω, deg	182.9557	270	360	360
True anomaly ν, deg	0	0	0	0

RAAN = Right Ascension of the Ascending Node.

premises of data driven models. In this section, the classical problem of identifying the central force field from position-only observation data is considered to validate the developed approach. In addition, the results from the methods developed in this paper are compared with the same analysis performed with a multilayered NN-based approach. Previous work conducted in this respect [40,41] will allow us to precisely compare the machine-learning-based approach with the sparse approximation method.

1. Two-Body Problem in Cartesian Coordinates

Let $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ be the position vector of two bodies, and m_1 and m_2 be their respective mass. If $\boldsymbol{r} = \boldsymbol{r}_2 - \boldsymbol{r}_1$ is the relative position vector between the two bodies, the dynamics of the two-body problem are given by

$$\ddot{\boldsymbol{r}} = -\frac{\mu}{r^3}\boldsymbol{r} \tag{44}$$

with $\mu = G(m_1 + m_2)$ and G is the universal gravitational constant. In an inertial reference frame and using Cartesian coordinates, with $\boldsymbol{r} = \{x, \quad y, \quad z\}^T$ and $r = \sqrt{x^2 + y^2 + z^2}$, Eq. (44) can be written as

$$\boldsymbol{f}: \boldsymbol{r} \mapsto \ddot{\boldsymbol{r}} \Leftrightarrow \boldsymbol{f}: \begin{bmatrix} x \\ y \\ z \end{bmatrix} \mapsto \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = -\frac{\mu}{r^3}\begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{45}$$

The idea is to identify the governing equations of the function $\boldsymbol{f}$ introduced in Eq. (45) without any a priori knowledge about its structure and therefore to determinate if the resulting identified model has the ability to identify the underlying dynamics embedded in some data set.

2. Training Set and Dictionary of Basis Functions

Four different types of orbits are selected to build the training set: a low Earth orbit (LEO), a Molniya orbit, a polar orbit, and a geo-synchronous orbit. Table 8 summarizes their orbital elements. The training data set corresponds to a fraction of a revolution on these orbits: range data are recorded every 1 s for 1.2 h for the International Space Station (ISS), and 2 h for the remaining three. Canonical units are used in this example: the length unit (LU) is chosen to be the radius of the Earth; the time unit (TU) is chosen such that the gravitational parameter $\mu = 1$.

To construct the initial dictionary, the set $\mathbb{R}_3[\boldsymbol{r}]$ consisting of 20 "monomials" up to order 3 in x, y, and z is defined:

$$\mathbb{R}_3[\boldsymbol{r}] = \{1 \quad x \quad x^2 \quad x^3 \quad y \quad xy \quad x^2y \quad y^2 \quad xy^2 \quad y^3 \quad z \quad xz \quad x^2z \quad yz \quad xyz \quad y^2z \quad z^2 \quad xz^2 \quad yz^2 \quad z^3\} \tag{46}$$

Furthermore, sets $\mathbb{R}_{3,1}[\boldsymbol{r}, r]$, $\mathbb{R}_{3,2}[\boldsymbol{r}, r]$, and $\mathbb{R}_{3,3}[\boldsymbol{r}, r]$ are defined to consist of the aforementioned monomials divided by r, r^2, and r^3, i.e.,

$$\mathbb{R}_{3,1}[\boldsymbol{r}, r] = \mathbb{R}_3[\boldsymbol{r}]/r, \qquad \mathbb{R}_{3,2}[\boldsymbol{r}, r] = \mathbb{R}_3[\boldsymbol{r}]/r^2,$$
$$\mathbb{R}_{3,3}[\boldsymbol{r}, r] = \mathbb{R}_3[\boldsymbol{r}]/r^3 \tag{47}$$

This class of basis functions is motivated by the general $1/r^n$ forms that are often seen in force interactions, especially for conservative force fields including gravity. The final dictionary consists of a total of 80 basis functions constructed as the union of $\mathbb{R}_3[\boldsymbol{r}]$, $\mathbb{R}_{3,1}[\boldsymbol{r}, r]$, $\mathbb{R}_{3,2}[\boldsymbol{r}, r]$, and $\mathbb{R}_{3,3}[\boldsymbol{r}, r]$, i.e.,

$$\mathcal{D} = \mathbb{R}_3[\boldsymbol{r}] \cup \mathbb{R}_{3,1}[\boldsymbol{r}, r] \cup \mathbb{R}_{3,2}[\boldsymbol{r}, r] \cup \mathbb{R}_{3,3}[\boldsymbol{r}, r] \tag{48}$$

The second-order formulation described in Sec. III.B is used to identify the inherent true central force field dynamics, with coefficients of the low-pass filters chosen to be $\lambda_1 = \lambda_2 = 10$. Figure 7 presents the coefficients found using the least-squares solution and the coefficients from the sparse approximation method. Once again the least-squares method fails to clearly identify the true participating basis functions. Instead, a nonminimal combination of basis functions in the dictionary is selected to minimize the mean squared error. On the contrary, the sparse approximation approach clearly identifies the one basis function corresponding to inverse square law as reported in Table 9. These converged coefficients agree with the true value of μ with almost machine precision. Figure 8 shows the error resulting in the propagation on the training orbits using dynamics identified by the least-squares and sparse methods. The error in the propagation is around 10^{-4} LU depending on the type of orbit for the least-squares method, whereas an average absolute error for sparse approximation is on the order of 10^{-13} LU. The final error resulting from the least-squares solution is not satisfactory due to the excitation of basis functions that are not participating in the true dynamics.

3. *Comparison with the Deep Learning Approach*

To show the relative performance of the sparse learning methodology presented in this paper, a multilayer NN learning-based approach is also considered. An NN can be seen as a complex nonlinear mapping between some given input and output data. Mathematically speaking, if E and F are two topological spaces, an NN is a mapping $\mathcal{M}{:}E \to F$ such that

$$\mathcal{M}{:}\boldsymbol{x} \mapsto \boldsymbol{y} = \mathcal{M}(\boldsymbol{x}) \tag{49}$$

where $\boldsymbol{x}$ is the input and $\boldsymbol{y}$ the output of the NN. The mapping $\mathcal{M}$ is generally nonlinear and a function of a set of parameters $\boldsymbol{\alpha}$:

$$\mathcal{M} = \boldsymbol{\alpha}_{\mathcal{M}} \tag{50}$$

Along with the specific structure of the mapping $\mathcal{M}$, the set of parameters $\boldsymbol{\alpha}$ defines an NN uniquely. The notation $\boldsymbol{\alpha}$ is not chosen randomly: this set of parameters is analogous to the matrix of coefficients $\boldsymbol{\alpha}$ defined Eq. (5). These unknown parameters are found by minimizing the loss function

$$L(\mathcal{M}_{\boldsymbol{\alpha}}, \mathcal{S}) = \sum_{k=1}^{N} MSE(k),$$
$$MSE(k) = \frac{1}{m}\sum_{i=1}^{m}(y_k(i) - \tilde{y}_k(i))^2 = \frac{1}{m}\|\boldsymbol{y}_k - \tilde{\boldsymbol{y}}_k\|_2^2 \tag{51}$$

where $\tilde{\boldsymbol{y}}_k \in \mathbb{R}^m$ represents the output of the network, i.e., $\tilde{\boldsymbol{y}}_k = \mathcal{M}_{\boldsymbol{\alpha}}(\boldsymbol{x}_k)$, and the $\boldsymbol{y}_k \in \mathbb{R}^m$ represent the true measurements for the output vector. Figure 9 shows that how the solution from the NN and the true solution are generated to compute the loss function to find the unknown network parameters $\boldsymbol{\alpha}$. Unfortunately, the resulting optimization is nonlinear in nature with multiple local minima. In our prior work [40,41], an extensive study is conducted to understand the

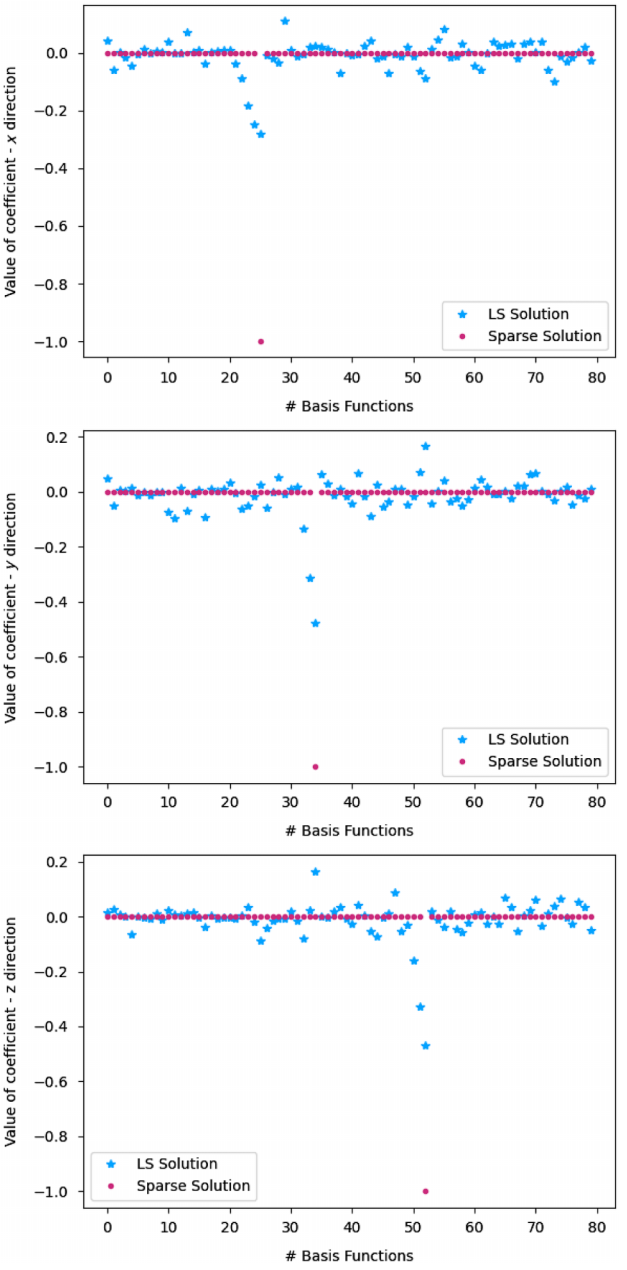


Fig. 7 Two-body problem: value of the coefficients for the basis functions.

Table 9 Value of the coefficients for the sparse solution vs the true coefficients

Basis function	No. in the dictionary	True value	Relative error %
$\boldsymbol{r} \mapsto x/r^3$	25	−1	$2.45 \cdot 10^{-11}$
$\boldsymbol{r} \mapsto y/r^3$	34	−1	$-7.21 \cdot 10^{-12}$
$\boldsymbol{r} \mapsto z/r^3$	52	−1	$-1.93 \cdot 10^{-11}$

learning capabilities of NNs to identify the Keplerian dynamics. The goal is to examine whether the specific structure of NNs can learn the inherent dynamics of the two-body problem and examine whether an NN-trained model can reproduce the well-known characteristics of Keplerian dynamics such as conservation of energy and angular momentum. Several test cases are considered to assess the learning capability of the converged NN. Three different network architectures (feed-forward, residual, and deep residual) are considered in addition to studying the impact of the size of training data size on the network approximation. Table 10 summarizes the best results on

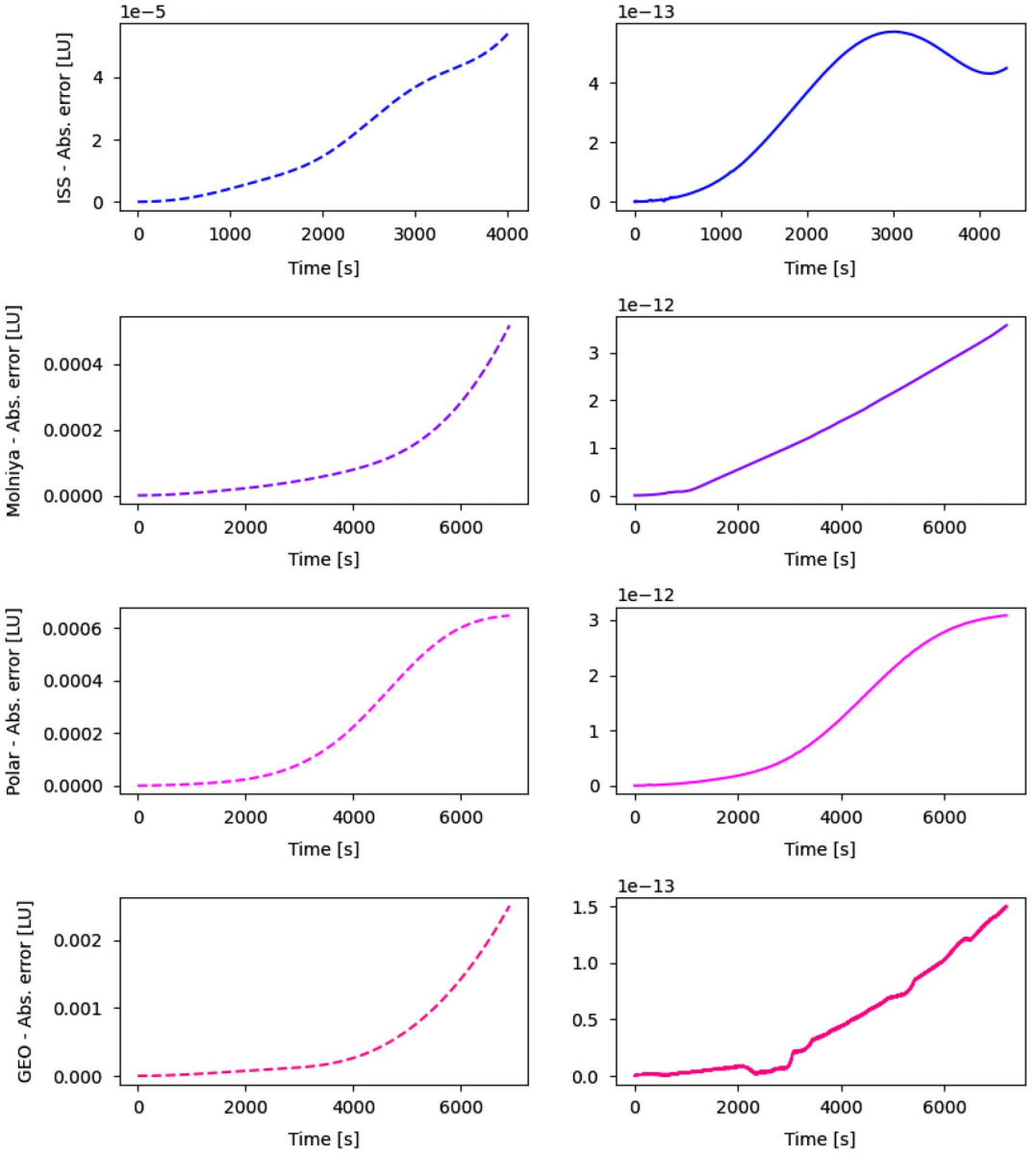


Fig. 8 Two-body problem: absolute relative error for radial component.

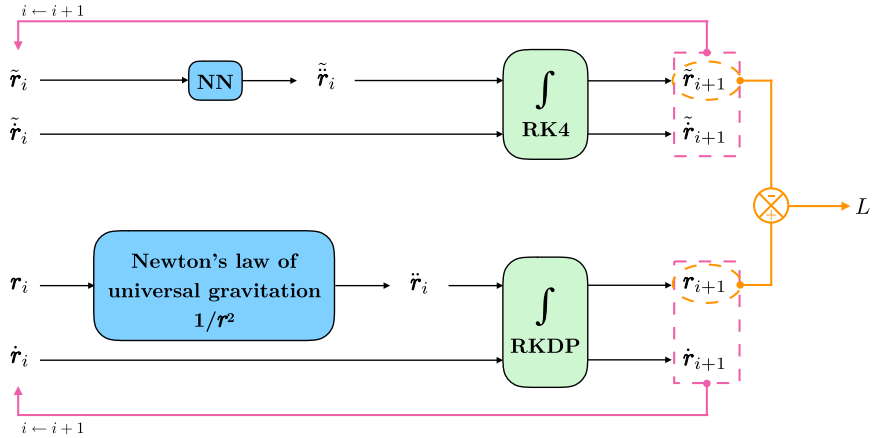


Fig. 9 The computation of loss function for NN approximation: the upper part illustrates how the NN is used to approximate the dynamics along with the Runge–Kutta fixed-size step algorithm; the bottom part is the classical generation of the true solution with the known dynamics and a Dormand–Prince integration algorithm.

approximating the Keplerian motion using different architectures of NN along with the results obtained with the least-squares and sparse approximation in this work. It should be mentioned that the training data set for the NN is much larger than the training data set for the least-squares and sparse approximation. The training data set for the NN approximation consists of 20 random orbits with multiple

Table 10 Comparison of different methods to approximate the Keplerian dynamics

Method	Feed-forward NN	Residual NN	Deep residual NN	Least-squares	Sparse
Average error [LU]	$7 \cdot 10^{-8}$	$4 \cdot 10^{-8}$	$3 \cdot 10^{-8}$	$2 \cdot 10^{-4}$	$4 \cdot 10^{-13}$
No. of parameters	2070	480	7720	80	3

revolutions. It has been shown that the three structures are able to provide accurate results for orbit prediction considering a large data set. Although it seems that the NN-learned model can be trained to approximate the Keplerian dynamics to good accuracy, the complexity of the learned model is an issue. The resulting NN model is a profligate model for the Keplerian dynamics as compared with Newton's law of gravitation. While the DeepResNet structure considered provides the most accurate results, the ResNet architecture shows very similar performance with fewer parameters. However, none of these architectures is able to identify the parsimonious structure of the governing dynamics as identified by the sparse approximation algorithm. This is because methods like NN improve the approximation accuracy by increasing the complexity of the model (defined by the parameters of the model) while fixing the basic building blocks or the basis function, whereas the developed methodology along with other variants such as [36] improves the approximation capability by judiciously selecting the basis functions.

V. Conclusions

This work presents a convex-optimization-based approach for nonlinear system identification from state and control input time histories. The proposed methodology expands the unknown nonlinearities in system dynamics in terms of basis functions consisting of monomials of various orders. An integral form of the underlying nonlinear ordinary differential equations is considered to solve for the unknown coefficients for the basis functions. While conventional methods for nonlinear system identification rely on improving the approximation accuracy by increasing the number of basis functions and hence the parameters of the model, the developed approach exploits recent advances in sparse approximation to automatically select the appropriate structure for the inherent nonlinearities. Hence, the developed methodology chooses building blocks for accurate and efficient construction of the I/O map. Though the main ideas are developed for a first-order system to identify the I/O map from the time histories for full state (i.e., position and velocity) as well as control input vector, an extension is developed for the identification of second-order systems where nonlinearities are a function of only position-level state variables from time histories of only position-level state variables and control input. Three numerical examples are presented to provide evidence in support of the efficacy of the proposed approach. The unique agreement in correctly identifying the true dynamics for all the three problems provides a strong basis for optimism in demonstrating the utility of the approach for identifying the inherent physics-based map from given data. Future work will concentrate on developing methods to combine sparse approximation algorithms with linear system identification procedures to automatically select the dimension of the hidden state-space model.

Appendix: Approach for a General *d*th-order system

This section aims to extend even further the mathematical development and show how to identify the dynamics of a general dth-order physical system where only position measurements are accessible and the governing dynamics rely only upon position knowledge. For $d \in \mathbb{N}$, this system is represented as a general dth-order nonlinear dynamic system:

$$\boldsymbol{x}^{(d)}(t) = \boldsymbol{f}(\boldsymbol{x}(t)) + G\boldsymbol{u}(t) \tag{A1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$ represents the state of the system and $\boldsymbol{u}(t) \in \mathbb{R}^n$ is the control action at time t and $G \in \mathbb{R}^{n\times r}$ is the constant-time input influence matrix. The nonlinear function $\boldsymbol{f}:\mathbb{R}^n \to \mathbb{R}^n$ represents the dynamics constraints that define the equations of motion of the system and is unknown. Some important results useful for further developments will be presented first. In the following, $\lambda_1, \lambda_2, \cdots$ are real positive numbers.

A. Mathematical Development

For $(i, j) \in \mathbb{N}^2$, let us define the function ϕ such that $\phi(i, j) = 1$ if $i = 0$ or $j = 0$ and

$$\phi(i,j) = (-1)^j \sum_{k=1}^{i} \sum_{\substack{(l_1,l_2,\ldots,l_k)\in 1,i \\ n_{i_{l_1}}+n_{i_{l_2}}+\cdots+n_{i_{l_k}}=j \\ l_1<l_2<\cdots<l_k}} \lambda_{l_1}^{n_{l_1}} \lambda_{l_2}^{n_{l_2}} \ldots \lambda_{l_k}^{n_{l_k}} \quad \text{otherwise} \tag{A2}$$

For $(i, j) \in \mathbb{N}^2$, it can be shown that the number of terms in the series $\phi(i, j)$ is 1 if i or j is 0 and

$$\sum_{k=1}^{j} \binom{i}{k} \binom{j-1}{k-1} \quad \text{otherwise} \tag{A3}$$

Similarly, one has the following relationship for $(i, j) \in \mathbb{N}^2$:

$$\phi(i, j) - \lambda_{i+1}\phi(i + 1, j - 1) = \phi(i + 1, j) \tag{A4}$$

For any $d \in \mathbb{N}^*$ and $x \in \mathbb{R}$, it is possible to write

$$\begin{aligned} x^d &= a_0 \prod_{m=1}^{d}(x + \lambda_m) + a_1 \prod_{m=1}^{d-1}(x + \lambda_m) + \cdots + a_{d-1}(x + \lambda_1) + a_d \\ &= \sum_{i=0}^{d-1}\left[a_i \prod_{m=1}^{d-i}(s + \lambda_m)\right] + a_d \end{aligned} \tag{A5}$$

The coefficients are

$$a_i = \phi(d - i + 1, i) \tag{A6}$$

This last assumption is a central result and can be proved by induction.

Before proceeding, consider a set of basis functions $\{\phi_i\}_{i=1\ldots\infty}$, $\phi_i:\mathbb{R}^n \to \mathbb{R}$, so that one can approximate $\boldsymbol{f}$ as a linear combination of finite number of N basis functions:

$$\boldsymbol{f}(\boldsymbol{x}) \approx \sum_{i=1}^{N} \boldsymbol{\alpha}_i \phi_i(\boldsymbol{x}) = \boldsymbol{\alpha}^T \boldsymbol{\phi}(\boldsymbol{x}) \tag{A7}$$

where $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1 \ \boldsymbol{\alpha}_2 \ \cdots \ \boldsymbol{\alpha}_N]^T \in \mathbb{R}^{N\times n}$ and $\boldsymbol{\phi}(\boldsymbol{x}) = [\phi_1(\boldsymbol{x}) \ \phi_2(\boldsymbol{x}) \ \cdots \ \phi_N(\boldsymbol{x})]^T \in \mathbb{R}^{N\times 1}$. Hence Eq. (A1) can be rewritten as

$$\boldsymbol{x}^{(d)}(t) = \sum_{i=1}^{N} \boldsymbol{\alpha}_i \phi_i(\boldsymbol{x}) + G\boldsymbol{u}(t) \tag{A8}$$

From now on, consider a one-dimensional system; that is, $n = 1$ and $\boldsymbol{x} = x$ is a scalar variable. If $n > 1$, it is sufficient to work along the dimension of $\boldsymbol{x}$ and repeat the steps along each dimension. The Laplace transform of the left-hand side of Eq. (A8) is

$$\mathcal{L}\{x^{(d)}(t)\} = s^d X(s) - \sum_{k=1}^{d} s^{d-k} x^{(k-1)}(0) \tag{A9}$$

For $\lambda_i \in \mathbb{R}_+^*$, let us consider the Laplace filtering operator

$$\mathcal{I}_{\lambda_i}: \mathbb{R} \to \mathbb{R}, \quad \bullet \mapsto \frac{\bullet}{s+\lambda_i} \tag{A10}$$

Applying successively $\mathcal{I}_{\lambda_1}, \mathcal{I}_{\lambda_2}, \ldots, \mathcal{I}_{\lambda_d}$ to Eq. (A9) gives

$$X^{df}(s) = \mathcal{I}_{\lambda_d} \circ \mathcal{I}_{\lambda_{d-1}} \circ \cdots \circ \mathcal{I}_{\lambda_1}[\mathcal{L}\{x^{(d)}(t)\}] = \frac{s^d X(s) - \sum_{k=1}^{d} s^{d-k} x^{(k-1)}(0)}{\prod_{m=1}^{d}(s+\lambda_m)} = X_1^{df}(s) - X_2^{df}(s) \tag{A11}$$

where

$$X_1^{df}(s) = \frac{s^d X(s)}{\prod_{m=1}^{d}(s+\lambda_m)} \text{ and } X_2^{df}(s) = \frac{\sum_{k=1}^{d} s^{d-k} x^{(k-1)}(0)}{\prod_{m=1}^{d}(s+\lambda_m)} \tag{A12}$$

Using previous developments, it is possible to write $X_1^{df}(s)$ as

$$X_1^{df}(s) = \frac{s^d X(s)}{\prod_{m=1}^{d}(s+\lambda_m)} = X(s) + \sum_{i=1}^{d} \frac{\phi(d-i+1,i)}{\prod_{m=0}^{i-1}(s+\lambda_{d-m})} X(s) \tag{A13}$$

Similarly, the second term $X_2^{df}(s)$ is

$$\begin{aligned} X_2^{df}(s) &= \frac{\sum_{k=1}^{d} s^{d-k} x^{(k-1)}(0)}{\prod_{m=1}^{d}(s+\lambda_m)} = \frac{\sum_{k=1}^{d-1}(\sum_{i=0}^{d-k-1} \phi(d-k-i+1,i) \prod_{m=1}^{d-k-i}(s+\lambda_m) + \phi(1,d-k)) x^{(k-1)}(0)}{\prod_{m=1}^{d}(s+\lambda_m)} + \frac{x^{(d-1)}(0)}{\prod_{m=1}^{d}(s+\lambda_m)} \\ &= \sum_{k=1}^{d-1} \sum_{j=k}^{d-1} \frac{\phi(d-j+1,j-k)}{\prod_{m=0}^{j-1}(s+\lambda_{d-m})} x^{(k-1)}(0) + \sum_{k=1}^{d} \frac{\phi(1,d-k)}{\prod_{m=0}^{d-1}(s+\lambda_{d-m})} x^{(k-1)}(0) \\ &= \sum_{j=1}^{d} \sum_{k=1}^{j} \frac{\phi(d-j+1,j-k)}{\prod_{m=0}^{j-1}(s+\lambda_{d-m})} x^{(k-1)}(0) \end{aligned} \tag{A14}$$

Finally, the compact expression for X^{df} is

$$X^{df}(s) = X(s) + \sum_{j=1}^{d} \frac{\phi(d-j+1,j) X(s) - \sum_{k=1}^{j} \phi(d-j+1,j-k) x^{(k-1)}(0)}{\prod_{m=0}^{j-1}(s+\lambda_{d-m})} \tag{A15}$$

This expression of X^{df} is easily expressed in the time domain. Performing the same operations on the right-hand side of Eq. (A7) leads to

$$\Phi_i^{df}(s) = \mathcal{I}_{\lambda_d} \circ \mathcal{I}_{\lambda_{d-1}} \circ \cdots \circ \mathcal{I}_{\lambda_1}[\mathcal{L}\{\phi_i(x)\}] = \frac{\Phi_i(s)}{\prod_{m=1}^{d}(s+\lambda_m)}, \quad i = 1, 2, \ldots, N \tag{A16}$$

and

$$U^{df}(s) = \mathcal{I}_{\lambda_d} \circ \mathcal{I}_{\lambda_{d-1}} \circ \cdots \circ \mathcal{I}_{\lambda_1}[\mathcal{L}\{u(t)\}] = \frac{U(s)}{\prod_{m=1}^{d}(s+\lambda_m)} \tag{A17}$$

where $\mathcal{L}\{\phi_i(\boldsymbol{x}(t))\} = \Phi_i(s)$ and $\mathcal{L}\{u(t)\} = U(s)$. Finally, the Laplace filtered version of Eq. (A7) is

$$X^{df}(s) = \sum_{i=1}^{N} \alpha_i \Phi_i^{df}(s) + G U^{df}(s) \tag{A18}$$

with X^{df}, Φ_i^{df}, and U^{df} defined in Eqs. (A14–A16). Going back in the time domain, one can write the corresponding differential equations for Φ^{df} and U^{df} immediately. For Φ_i^{df}, first rewrite Eq. (A15) as

$$\begin{aligned} \Phi_i^{df}(s) = \frac{\Phi_i(s)}{\prod_{m=1}^{d}(s+\lambda_m)} &\Leftrightarrow \Phi_i^{df}(s) \prod_{m=1}^{d}(s+\lambda_m) = \Phi_i(s) \\ &\Leftrightarrow s^d \Phi_i^{df}(s) = -\sum_{k=0}^{d-1} a_k s^k \Phi_i^{df}(s) + \Phi_i(s) \end{aligned} \tag{A19}$$

Considering zero initial conditions, $\phi_i^{df(k)}(0) = 0$ for $k = 0, 1, \ldots, d$, one obtains the differential equation:

$$\phi_i^{df(d)}(t) = -\sum_{k=0}^{d-1} a_k \phi_i^{f(k)}(t) + \phi_i(x), \quad i = 1, 2, \ldots, N \tag{A20}$$

The same procedure for the input signal leads to the differential equation

$$u^{df(d)}(t) = -\sum_{k=0}^{d-1} a_k u^{df(k)}(t) + u(t) \tag{A21}$$

with similar zero initial conditions. Notice that Eqs. (A14), (A19), and (A20) lead to Eqs. (11), (14), and (15) for $d = 1$ and Eqs. (31), (35), and (36) for $d = 2$.

Acknowledgment

This material is based upon work supported jointly by the Air Force Office of Scientific Research grants FA9550-15-1-0313 and FA9550-17-1-0088.s.

References

[1] Juang, J.-N., *Applied System Identification*, Prentice-Hall, Englewood Cliffs, NJ, 1994.

[2] Nelles, O., *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*, Springer Science, Berlin, 2000.

[3] Gilbert, E. G., "Controllability and Observability in Multivariable Control Systems," *Journal of the Society for Industrial and Applied Mathematics, Series A Control*, Vol. 1, No. 2, 1963, pp. 128–151. https://doi.org/10.1137/0301009

[4] Ho, B. L., and Kálmán, R. E., "Effective Construction of Linear State-Variable Models from Input/Output Functions," *Regelungstechnik*, Vol. 14, No. 12, 1966, pp. 545–548. https://doi.org/10.1524/auto.1966.14.112.545

[5] Alvin, K. F., and Park, K. C., "Second Order Structural Identification Procedure via State-Space Based System Identification," *AIAA Journal*, Vol. 32, No. 2, 1994, pp. 397–406. https://doi.org/10.2514/3.11997

[6] Anderson, B. D. O., and Skelton, R. E., "The Generation of All q-Markov Covers," *IEEE Transactions on Circuits and Systems*, Vol. 46, 1988, pp. 351–356. https://doi.org/10.1016/S1474-6670(17)55525-1

[7] Juang, J.-N., Cooper, J. E., and Wright, J. R., "An Eigensystem Realization Algorithm Using Data Correlation (ERA/DC) for Modal Parameter Identification," *Control Theory and Advanced Technology*, Vol. 4, No. 1, 1988, pp. 5–14.

[8] Liu, K., Skelton, R. E., and Sharkey, J. P., "Modeling Hubble Space Telescope Flight Data by Q-Markov Cover Identification," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 2, 1994, pp. 250–256. https://doi.org/10.23919/ACC.1992.4792462

[9] Skelton, R. E., and Shi, G., "Iterative Identification and Control Using a Weighted q-Markov Cover with Measurement Noise," *Signal Processing*, Vol. 52, No. 2, 1996, pp. 217–234. https://doi.org/10.1016/0165-1684(96)00055-2

[10] Juang, J.-N., "Generalized Bilinear System Identification," *Journal of the Astronautical Sciences*, Vol. 57, Nos. 1–2, 2009, pp. 261–273. https://doi.org/10.1007/BF03321504

[11] Juang, J.-N., "Continuous-Time Bilinear System Identification," *Nonlinear Dynamics*, Vol. 39, 2005, pp. 79–94. https://doi.org/10.1007/s11071-005-1915-z

[12] Juang, J.-N., and Pappa, R. S., "An Eigensystem Realization Algorithm (ERA) for Modal Parameter Identification and Model Reduction," *Journal of Guidance, Control, and Dynamics*, Vol. 8, No. 5, 1985, pp. 620–627. https://doi.org/10.2514/3.20031

[13] Juang, J.-N., and Pappa, R. S., "Effects of Noise on Modal Parameters Identified by the Eigensystem Realization Algorithm," *Journal of Guidance, Control, and Dynamics*, Vol. 9, No. 3, 1986, pp. 294–303. https://doi.org/10.2514/3.20106

[14] Juang, J.-N., Phan, M., Horta, L. G., and Longman, R. W., "Identification of Observer/Kalman Filter Markov Parameters: Theory and Experiments," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 2, 1993, pp. 320–329. https://doi.org/10.2514/3.21006

[15] Juang, J.-N., and Longman, R. W., "Optimized System Identification," NASA TR 1999-209711, 1999.

[16] Verhaegen, M., and Yu, X., "A Class of Subspace Model Identification Algorithms to Identify Periodically and Arbitrarily Time Varying Systems," *Automatica*, Vol. 31, No. 2, 1995, pp. 201–216. https://doi.org/10.1016/0005-1098(94)00091-V

[17] Majji, M., Juang, J.-N., and Junkins, J. L., "Time-Varying Eigensystem Realization Algorithm," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 1, 2010, pp. 13–28. https://doi.org/10.2514/1.45722

[18] Majji, M., Juang, J.-N., and Junkins, J. L., "Observer/Kalman-Filter Time-Varying System Identification," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 3, 2010, pp. 887–900. https://doi.org/10.2514/1.45768

[19] Rugh, W. J., *Nonlinear System Theory: The Volterra/Wiener Approach*, Johns Hopkins Univ. Press, Baltimore, MD, 1981.

[20] Kvaternik, R. G., and Silva, W. A., "A Computational Procedure for Identifying Bilinear Representations of Nonlinear Systems Using Volterra Kernels," NASA, Langley Research Center TR 2008-215320, Hampton VA, 2008.

[21] Singla, P., and Junkins, J. L., *Multi-Resolution Methods for Modeling and Control of Dynamical Systems*, Chapman and Hall CRC Applied Mathematics and Nonlinear Science, Chapman and Hall, London, U.K., 2008.

[22] Crutchfield, J. P., and McNamara, B. S., "Equations of Motion from a Data Series," *Complex Systems*, Vol. 1, 1987, pp. 417–452.

[23] Narendra, K. S., and Parthasarathy, K., "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Transactions on Neural Networks*, Vol. 1, No. 1, 1990, pp. 4–27. https://doi.org/10.1109/72.80202

[24] Haykin, S., *Neural Networks: A Comprehensive Foundation*, Prentice-Hall, Upper Saddle River, NJ, 1998.

[25] Jin, C., Singla, P., and Singh, T., "A Multi-Resolution Approach with Sparseness Property for Input-Output Approximation," *AIAA Guidance, Navigation and Control Conference*, AIAA Paper 2010-8156, 2010.

[26] Jin, C., Singla, P., and Singh, T., "A Multi-Resolution Approach for Tumor Motion Modeling," IEEE, New York, 2010, pp. 1248–1253.

[27] Schmidt, M. D., Vallabhajosyula, R. R., Jenkins, J. W., Hood, J. E., Soni, A. S., Wikswo, J. P., and Lipson, H., "Automated Refinement and Inference of Analytical Models for Metabolic Networks," *Physical Biology*, Vol. 8, No. 5, 2011, Paper 055011. https://doi.org/10.1088/1478-3975/8/5/055011

[28] Daniels, B. C., and Nemenman, I., "Efficient Inference of Parsimonious Phenomenological Models of Cellular Dynamics Using S-Systems and Alternating Regression," *PLoS ONE*, Vol. 10, No. 3, 2015, Paper e0119821. https://doi.org/10.1371/journal.pone.0119821

[29] Daniels, B. C., and Nemenman, I., "Automated Adaptive Inference of Phenomenological Dynamical Models," *Nature Communications*, Vol. 6, No. 1, 2015, p. 8133. https://doi.org/10.1038/ncomms9133

[30] Brunton, S. L., Proctor, J. L., and Kutz, J. N., "Discovering Governing Equations from Data: Sparse Identification of Nonlinear Dynamical Systems," *Proceedings of the National Academy of Sciences*, Vol. 113, No. 15, 2016, pp. 3932–3937. https://doi.org/10.1073/pnas.1517384113

[31] Proctor, J. L., Brunton, S. L., Brunton, B. W., and Kutz, J. N., "Exploiting Sparsity and Equation-Free Architectures in Complex Systems," *European Physical Journal Special Topics*, Vol. 223, 2014, pp. 2665–2684. https://doi.org/10.1140/epjst/e2014-02285-8

[32] AlMomani, A. A. R., Sun, J., and Bollt, E., "How Entropic Regression Beats the Outliers Problem in Nonlinear System Identification," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, Vol. 30, No. 1, 2020, Paper 013107. https://doi.org/10.1063/1.5133386

[33] Zhang, S., and Lin, G., "Robust Subsampling-Based Sparse Bayesian Inference to Tackle Four Challenges (Large Noise, Outliers, Data Integration, and Extrapolation) in the Discovery of Physical Laws from Data," arXiv:1907.07788v2, 2019.

[34] Zhang, S., and Lin, G., "SubTSBR to Tackle High Noise and Outliers for Data-Driven Discovery of Differential Equations," *Journal of Computational Physics*, Vol. 428, 2021, Paper 109962. https://doi.org/10.1016/j.jcp.2020.109962

[35] Pantazis, Y., and Tsamardinos, I., "A Unified Approach for Sparse Dynamical System Inference from Temporal Measurements," *Bioinformatics*, Vol. 35, No. 18, 2019, pp. 3387–3396. https://doi.org/10.1093/bioinformatics/btz065

[36] Schaeffer, H., and McCalla, S. G., "Sparse Model Selection via Integral Terms," *Physical Review E*, Vol. 96, No. 2, 2017, Paper 023302. https://doi.org/10.1103/PhysRevE.96.023302

[37] Guého, D., Singla, P., and Melton, R. G., "Data-Driven Sparse Approximation for the Identification of Nonlinear Dynamical Systems: Applications in Astrodynamics," *Spaceflight Mechanics 2020*, Univelt Inc., Escondido, CA, 2020.

[38] Adurthi, N., Singla, P., and Majji, M., "Sparse Approximation-Based Collocation Scheme for Nonlinear Optimal Feedback Control Design," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, 2017, pp. 248–264. https://doi.org/10.2514/1.G001755

[39] Boyd, S., and Vandenberghe, L., *Convex Optimization*, Cambridge Univ. Press, Cambridge, England, U.K., 2004, Chap. 4.

[40] Guého, D., Singla, P., and Melton, R. G., "Learning Capabilities of Neural Networks and Keplerian Dynamics," *AAS/AIAA Astrodynamics Specialist Conference, 2018*, Univelt Inc., Escondido, CA, 2018, pp. 2293–2310.

[41] Guého, D., Singla, P., and Melton, R. G., "Investigation of Different Neural Network Architectures for Dynamic System Identification: Applications to Orbital Mechanics," *Spaceflight Mechanics 2019*, Univelt Inc., Escondido, CA, 2019, pp. 1789–1803.