

A robust algorithm to test the observability of large linear systems with unknown parameters

X. Shi^a, M.S. Williams^b and M.N. Chatzis^{c,*}

^a*Ph.D. Candidate, Department of Engineering Science, University of Oxford, Oxford, UK*

^b*Professor, Department of Engineering Science, University of Oxford, Oxford, UK*

^c*Associate Professor, Department of Engineering Science, University of Oxford, Oxford, UK*

Abstract

This work proposes a robust algorithm to examine the observability of linear systems whose dynamic states and parameters are to be identified. The observability of a dynamical system plays a fundamental role in predicting whether it would be successful to use system identification methods to estimate the dynamic states and parameters of the system from a given set of input-output measurements. The motivation of the development of the suggested algorithm arises from the need to address the significant physical memory requirements of the standard implementation of the Observability Rank Condition (ORC). The high computational cost of the ORC substantially limits its applicability to real-world engineering systems, even when the underlying dynamics can be reasonably approximated as linear. The framework of the algorithm is obtained through the derivation of a recursive formula for the computation of the observability matrix of linear systems with unknown parameters. To further improve the efficiency, robust numerical implementations of the algorithm are achieved through random realizations of the dynamic states and parameters and the use of modular operations. The superior performance of the algorithm is demonstrated using several examples of large linear dynamical models of engineering systems containing up to thousands of dynamic states and parameters.

Keywords: Observability, Identifiability, Observability Rank Condition,

¹*Corresponding author. E-mail address: manolis.chatzis@eng.ox.ac.uk

1. Introduction

Due to recent advances in the development and improvement of sensor technologies and estimation methodologies, system identification has become a more mature technique by means of which engineers are able to gain accurate knowledge of the properties and behavior of real engineering systems. Before planning a system identification campaign, a priori observability assessment of a dynamical system is often necessary. If the system under a specific input-output measurement scheme is observable, the states of the system can theoretically be inferred from the input-output measurements. If not, it is then impossible to properly estimate the values of the states which are deemed as unobservable using any identification method. For the systems with unobservable states, a practical advantage of using the observability assessment is that it provides guidance to engineers on how to adjust sensor placement, or to reformulate the mathematical model of the system so that the system under study would become observable. Performing such changes before the experimental campaign starts brings the benefits of savings of cost and time, as the observability assessment is purely theoretical and analytical requiring no real measured data.

A large amount of works have been devoted to investigating the observability properties of dynamical systems. Initiating with the linear observability defined by Kalman [1], the differential-geometric definition of nonlinear observability given by Hermann and Krener [2] and its algebraic counterpart given by Diop and Fliess [3], many observability approaches were derived to test whether a system is observable or not. These include the power series method in [4], the local state variable isomorphism method in [5], the methods relied on characteristic set or standard bases computations in [6], the elimination method in differential algebra in [7], etc.

Udwadia and Sharma [8], Udwadia et al. [9] and Franco et al. [10] placed concentrations on the global identifiability, i.e. the observability of parameters,

of civil and mechanical engineering systems. The works discussed the existence
 30 of unique solutions for the unknown physical properties of shear-type structures based on a minimum amount of excitation-sensor instruments. Review and comparison of various observability methods were provided in Chatzis et al. [11] where the authors also extended the use of the existing methods to non-smooth systems which are often encountered in phenomena associated with
 35 damage, sliding and impact. The investigation into the observability of non-smooth systems leads to the discontinuous modifications of the Kalman filters that improve the identification of the systems with states having varying observability properties throughout their evolutions [12, 13]. The aforementioned approaches are all based on the assumption of all inputs applied to a system
 40 being measured. Recent extensions of the classical observability theory allow for checking the observability of the states of nonlinear analytic systems driven by unmeasured inputs [14, 15].

The concept of observability discussed herein, more strictly speaking, is theoretical observability or called structural observability in the field of systems
 45 biology [16, 17] which is not identical to practical observability that takes into account the real-life measured and noise-contaminated data during a dynamic test as in [16, 18, 19]. Unlike practical observability, theoretical observability is an inherent property of a system which is not affected by the quality of the measurements obtained or the type of the identification method used.
 50 A theoretically unobservable system will not become observable by changing the identification method used, using sensors of better quality, or repeating the tests. The remainder of this paper will keep referring to theoretical observability simply as observability.

The Observability Rank Condition (ORC) was introduced by Hermann and
 55 Krener [2] as a criterion to determine whether a nonlinear system is locally weakly observable or not. By definition, a system is said to be locally weakly observable if the states of the system can be instantaneously distinguished from their neighbors based on the input-output mapping. The implementation of the ORC can be achieved by a simple algorithm based on symbolic computations

60 of the Lie derivatives of outputs and their gradients (e.g. as in [20]). However, when the ORC is implemented for real-world engineering systems, often with a large number of states and complicated equations, the complexity of symbolic computations rapidly becomes overwhelming, resulting in significant growth of the physical memory requirements that are hard to be met in a standard com-
65 puter and increase of the computational cost [21]. In structural engineering applications, for example, a finite element model of a structure may have up to thousands of degrees of freedom and contain equally many dynamic states and parameters to be identified. The standard implementation of the ORC, as well as the other observability approaches mentioned before, are far from being able
70 to handle systems of such size.

Computationally efficient observability algorithms have been explored in several works [16, 17, 22, 23, 24]. A semi-numerical algorithm was developed by Sedoglavic [23] for robust testing of rational and polynomial nonlinear systems by making use of a numerical realized variational system and the method of
75 Newton’s iteration. The objective of this paper is to propose an efficient algorithmic implementation of the ORC for linear systems whose dynamic states and unknown parameters are to be estimated. This category of systems covers a wide range of civil and mechanical systems. The paper derives an explicit expression of the observability matrix of linear systems with unknown param-
80 eters. It shows the viability of computing the observability matrix recursively by introducing a product rule involving three-dimensional matrices. The computationally expensive symbolic differentiations are replaced by matrix additions and multiplications. This further allows for numerical realization of the algorithm, and in addition the application of modular operations makes it possible to use
85 **integers in floating-point format** which substantially enhances the computation efficiency. A MATLAB implementation of the algorithm is freely available at the following link ². Applications and performance of the algorithm are demonstrated through several engineering examples.

²www2.eng.ox.ac.uk/nldyn/on-line-resources/robust-linear-observability

2. The Observability Rank Condition (ORC)

90 In Section 2, the ORC is reviewed with a focus on its implementation while the details of theory and derivation can be found in [2, 20, 25]. Consider a nonlinear system with affine inputs written as the following state-space representation:

$$\begin{aligned}\dot{\mathbf{x}}_t &= \mathbf{F}(\mathbf{x}_t, \boldsymbol{\theta}) + \sum_{i=1}^m \mathbf{G}_i(\mathbf{x}_t, \boldsymbol{\theta}) u_i \\ \mathbf{y} &= \mathbf{h}(\mathbf{x}_t, \boldsymbol{\theta})\end{aligned}\tag{1}$$

where $\mathbf{x}_t \in \mathbb{R}^{n_t}$ denotes the time-variant dynamic states, $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ the time-invariant parameters, u_1, u_2, \dots, u_m the measured independent inputs and $\mathbf{y} \in \mathbb{R}^p$ the output measurement vector. $\mathbf{F}, \mathbf{G}_1, \mathbf{G}_2, \dots, \mathbf{G}_m$ and \mathbf{h} are vectors of nonlinear smooth functions. If the time-invariant parameters of the system are unknown, they can be treated as additional states with zero dynamics, i.e. $\dot{\boldsymbol{\theta}} = \mathbf{0}_{n_\theta \times 1}$. This leads to a state augmentation by including both the dynamic states and the parameters in a common state vector $\mathbf{x} \in \mathbb{R}^n$:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_t \\ \boldsymbol{\theta} \end{bmatrix}, \quad n = n_t + n_\theta\tag{2}$$

and the system in equation (1) may be rewritten with respect to the state vector as:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}) + \sum_{i=1}^m \mathbf{g}_i(\mathbf{x}) u_i \\ \mathbf{y} &= \mathbf{h}(\mathbf{x})\end{aligned}\tag{3}$$

where

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} \mathbf{F}(\mathbf{x}_t, \boldsymbol{\theta}) \\ \mathbf{0}_{n_\theta \times 1} \end{bmatrix}, \quad \mathbf{g}_i(\mathbf{x}) = \begin{bmatrix} \mathbf{G}_i(\mathbf{x}_t, \boldsymbol{\theta}) \\ \mathbf{0}_{n_\theta \times 1} \end{bmatrix}\tag{4}$$

The (local weak) observability of the above system defined in the equations (3) and (4) can be examined by the ORC. When focusing on whether the unknown parameters can be identified, their observability properties are equivalent to their identifiability properties [11, 16, 17]. From a practical standpoint and for the purposes of this paper, further distinctions between the concepts of local

weak observability, local observability, weak observability and global observability, as in [2], are not considered herein. Instead the term observability in this paper will be used to refer to local weak observability. To implement the ORC, an observability matrix consisting of the gradients of the Lie derivatives of the output functions \mathbf{h} with respect to the state vector \mathbf{x} is built. The Lie derivative of a vector of scalar functions, $\boldsymbol{\lambda} = [\lambda_1(x) \quad \dots \quad \lambda_p(x)]^T$, along a vector field, $\mathbf{v} = [v_1(x) \quad \dots \quad v_n(x)]^T$, is defined as:

$$\mathbf{L}_v \boldsymbol{\lambda} = \frac{\partial \boldsymbol{\lambda}}{\partial \mathbf{x}} \mathbf{v} \quad (5)$$

where $\frac{\partial \boldsymbol{\lambda}}{\partial \mathbf{x}}$ is a Jacobian matrix:

$$\frac{\partial \boldsymbol{\lambda}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \lambda_1}{\partial x_1} & \frac{\partial \lambda_1}{\partial x_2} & \dots & \frac{\partial \lambda_1}{\partial x_n} \\ \frac{\partial \lambda_2}{\partial x_1} & \frac{\partial \lambda_2}{\partial x_2} & \dots & \frac{\partial \lambda_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \lambda_p}{\partial x_1} & \frac{\partial \lambda_p}{\partial x_2} & \dots & \frac{\partial \lambda_p}{\partial x_n} \end{bmatrix} \quad (6)$$

For the ORC, the Lie derivatives are repeatedly computed along the system functions \mathbf{f} and \mathbf{g}_i , yielding $\mathbf{h}, \mathbf{L}_f \mathbf{h}, \mathbf{L}_{g_i} \mathbf{h}, \mathbf{L}_f \mathbf{L}_f \mathbf{h}, \mathbf{L}_{g_i} \mathbf{L}_f \mathbf{h}$, etc. A simple algorithm described in [11, 20] computes the Lie derivatives in a recursive way, and successively arranges the gradients of these Lie derivatives to output the observability matrix. The algorithm is presented in the following:

Algorithm I

Input to the algorithm: the system functions $\mathbf{f}, \mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_m$, the output functions \mathbf{h} and the system states \mathbf{x}

Output from the algorithm: the observability matrix $\boldsymbol{\Omega}$

Initialization: Let $k = 0$, $\boldsymbol{\lambda}_0 = \mathbf{h}$, $\boldsymbol{\Omega}_0 = \frac{\partial \boldsymbol{\lambda}_0}{\partial \mathbf{x}}$

1. Compute the Lie derivatives $\boldsymbol{\lambda}_{k+1} = \left[(\mathbf{L}_f \boldsymbol{\lambda}_k)^T \quad (\mathbf{L}_{g_1} \boldsymbol{\lambda}_k)^T \quad (\mathbf{L}_{g_2} \boldsymbol{\lambda}_k)^T \quad \dots \quad (\mathbf{L}_{g_m} \boldsymbol{\lambda}_k)^T \right]^T$
2. Compute the rows of the gradient of the Lie derivatives, and arrange them into the observability matrix $\boldsymbol{\Omega}_{k+1} = \boldsymbol{\Omega}_k \cup \frac{\partial \boldsymbol{\lambda}_{k+1}}{\partial \mathbf{x}}$
3. Compute the rank, if $\text{rank}(\boldsymbol{\Omega}_{k+1}) = \text{rank}(\boldsymbol{\Omega}_k)$, or $\text{rank}(\boldsymbol{\Omega}_{k+1}) = n$, or $k = n - 2$, end and output $\boldsymbol{\Omega} = \boldsymbol{\Omega}_{k+1}$

4. Let $k = k + 1$ and go to step 1

135 It should be noted that the computations involved in Algorithm I are symbolic as the gradient operations are analytic and thus must be implemented by differentiating symbolically the associated expressions. Once the observability matrix is obtained, the observability of the system in the equations (3) and (4) is analyzed through rank testing. The system satisfies the observability rank
140 condition so that it is observable if $\mathbf{\Omega}$ has a rank equal to the dimension of \mathbf{x} , n . The observability of the l^{th} ($l = 1, \dots, n$) state of \mathbf{x} can be detected by removing the l^{th} column of $\mathbf{\Omega}$: if the rank of the resulting matrix is smaller than the rank of $\mathbf{\Omega}$, then the l^{th} state is observable; otherwise it is unobservable. If that state is further a parameter it will be correspondingly deemed as identifiable or
145 unidentifiable. If and only if all of the dynamic states are observable and all of the parameters are identifiable, then the system is said to be observable. Furthermore, the transcendence degree of the system, that is calculated as the rank deficiency $n - \text{rank}(\mathbf{\Omega})$, can be used to account for the number of unobservable states for which modelling assumptions need to be made in order to reduce an
150 unobservable system to become observable [11, 25].

3. Observability matrix of linear systems with unknown parameters

It is worth mentioning the observability matrix of linear systems with *known* parameters has been well-explored [1]. This work focuses on linear underlying systems with *unknown* parameters $\boldsymbol{\theta}$ which are described by:

$$\begin{aligned} \dot{\mathbf{x}}_t &= \mathbf{A}(\boldsymbol{\theta})\mathbf{x}_t + \sum_{i=1}^m \mathbf{B}_i(\boldsymbol{\theta})u_i, & \dot{\boldsymbol{\theta}} &= \mathbf{0}_{n_\theta \times 1} \\ \mathbf{y} &= \mathbf{C}(\boldsymbol{\theta})\mathbf{x}_t \end{aligned} \tag{7}$$

155 where $\mathbf{A}(\boldsymbol{\theta})$, $\mathbf{B}_i(\boldsymbol{\theta})$ and $\mathbf{C}(\boldsymbol{\theta})$ are the matrices with elements functions of the unknown parameters. Strictly speaking, the system in equation (7) is linear with respect to the dynamic states and the inputs, but overall nonlinear due to the terms of the products between $\mathbf{A}(\boldsymbol{\theta})$, $\mathbf{B}_i(\boldsymbol{\theta})$, $\mathbf{C}(\boldsymbol{\theta})$ and \mathbf{x}_t , u_i . Following the

state augmentation $\mathbf{x} = \begin{bmatrix} \mathbf{x}_t^T & \boldsymbol{\theta}^T \end{bmatrix}^T$ for the purpose of parameter estimation,
 160 the system is rewritten as the form of (3), and the corresponding \mathbf{f} and \mathbf{g}_i can
 be obtained as:

$$\mathbf{f} = \begin{bmatrix} \mathbf{A}(\boldsymbol{\theta})\mathbf{x}_t \\ \mathbf{0}_{n_\theta \times 1} \end{bmatrix}, \quad \mathbf{g}_i = \begin{bmatrix} \mathbf{B}_i(\boldsymbol{\theta}) \\ \mathbf{0}_{n_\theta \times 1} \end{bmatrix} \quad (8)$$

Algorithm I is applicable to test the observability of the system in the equa-
 tions (7) and (8). However, as noted in the Introduction, the symbolic computa-
 tions are cumbersome especially when the augmented state vector \mathbf{x} is of large
 165 size and the system functions \mathbf{f} and \mathbf{g}_i are complicated. In order to develop a
 more computationally efficient algorithm, this section aims to derive an explicit
 expression of the observability matrix of the system. The key idea is to separate
 all the Lie derivatives of the system into three subsets:

$$\{\mathbf{L}_f^k \mathbf{h} : k = 0, 1, \dots, n-1\} \quad (9)$$

$$\{\mathbf{L}_{g_i} \mathbf{L}_f^k \mathbf{h} : k = 0, 1, \dots, n-2\} \quad (10)$$

170

$$\{\mathbf{L}_{v_r} \dots \mathbf{L}_{v_1} \mathbf{L}_{g_i} \mathbf{L}_f^k \mathbf{h} : k = 0, 1, \dots, n-3, r = 1, 2, \dots, n-k-2\} \quad (11)$$

where \mathbf{v}_r is any of the vector fields belonging to the set $\{\mathbf{f}, \mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_m\}$.
 $\mathbf{L}_f^k \mathbf{h}$ is the k^{th} order Lie derivative, i.e. $\underbrace{\mathbf{L}_f \dots \mathbf{L}_f}_k \mathbf{h}$, and this can be calculated
 recursively by $\mathbf{L}_f^k \mathbf{h} = \mathbf{L}_f \mathbf{L}_f^{k-1} \mathbf{h}$ with $\mathbf{L}_f^0 \mathbf{h} = \mathbf{h}$. The expression of each subset
 of the Lie derivatives is derived as follows.

175 In (9), the k^{th} order Lie derivative of \mathbf{h} along \mathbf{f} is expressed as:

$$\mathbf{L}_f^k \mathbf{h} = \mathbf{C} \mathbf{A}^k \mathbf{x}_t \quad (12)$$

A proof by induction for equation (12) follows: for $k = 0$, $\mathbf{L}_f^0 \mathbf{h} = \mathbf{h} = \mathbf{C} \mathbf{x}_t$,
 which indeed satisfies (12). If one assumes that equation (12) holds for $k-1$,
 i.e. $\mathbf{L}_f^{k-1} \mathbf{h} = \mathbf{C} \mathbf{A}^{k-1} \mathbf{x}_t$, the equality will be shown to hold for k . Using
 equation (5) to compute $\mathbf{L}_f^k \mathbf{h} = \mathbf{L}_f \mathbf{L}_f^{k-1} \mathbf{h}$, it is necessary to compute $\frac{\partial \mathbf{L}_f^{k-1} \mathbf{h}}{\partial \mathbf{x}} =$

180 $\frac{\partial CA^{k-1}x_t}{\partial x}$. The gradient with respect to the state vector, x , can be separated into a sub-gradient with respect to the dynamic states, x_t , and a sub-gradient with respect to the parameters, θ , giving $\frac{\partial CA^{k-1}x_t}{\partial x} = \begin{bmatrix} \frac{\partial CA^{k-1}x_t}{\partial x_t} & \frac{\partial CA^{k-1}x_t}{\partial \theta} \end{bmatrix}$. Since CA^{k-1} is a matrix depending on θ but not on x_t , $\frac{\partial CA^{k-1}x_t}{\partial x_t} = CA^{k-1}$. Thus taking the expression of f in equation (8) and substituting it into (5),
 185 $L_f^k h$ is computed as:

$$L_f^k h = L_f L_f^{k-1} h = \begin{bmatrix} CA^{k-1} & \frac{\partial CA^{k-1}x_t}{\partial \theta} \end{bmatrix} \begin{bmatrix} Ax_t \\ \mathbf{0}_{n_\theta \times 1} \end{bmatrix} = CA^k x_t \quad (13)$$

which coincides with equation (12) for k , and therefore concludes the proof by induction of (12).

Having obtained the expression of $L_f^k h$, it proceeds to derive the Lie derivative of $L_f^k h$ along g_i using the equations (8) and (5) in (10):

$$L_{g_i} L_f^k h = \begin{bmatrix} CA^k & \frac{\partial CA^k x_t}{\partial \theta} \end{bmatrix} \begin{bmatrix} B_i \\ \mathbf{0}_{n_\theta \times 1} \end{bmatrix} = CA^k B_i \quad (14)$$

190 To evaluate the remaining Lie derivatives $L_{v_r} \dots L_{v_1} L_{g_i} L_f^k h$ appearing in (11), $L_{v_1} L_{g_i} L_f^k h$ corresponding to either $L_f L_{g_i} L_f^k h$ or $L_{g_j} L_{g_i} L_f^k h$ ($j = 1, \dots, m$) is first computed using the equations (8) and (5). $\frac{\partial L_{g_i} L_f^k h}{\partial x} = \frac{\partial CA^k B_i}{\partial x}$ based on equation (14) is written as $\frac{\partial L_{g_i} L_f^k h}{\partial x} = \begin{bmatrix} \frac{\partial CA^k B_i}{\partial x_t} & \frac{\partial CA^k B_i}{\partial \theta} \end{bmatrix}$. Since $CA^k B_i$ is a vector of parameters but not of dynamic states, $\frac{\partial CA^k B_i}{\partial x_t}$ is a $p \times n_t$ zero matrix. Therefore,
 195

$$\begin{aligned} L_f L_{g_i} L_f^k h &= \begin{bmatrix} \mathbf{0}_{p \times n_t} & \frac{\partial CA^k B_i}{\partial \theta} \end{bmatrix} \begin{bmatrix} Ax_t \\ \mathbf{0}_{n_\theta \times 1} \end{bmatrix} = \mathbf{0}_{p \times 1} \\ L_{g_j} L_{g_i} L_f^k h &= \begin{bmatrix} \mathbf{0}_{p \times n_t} & \frac{\partial CA^k B_i}{\partial \theta} \end{bmatrix} \begin{bmatrix} B_j \\ \mathbf{0}_{n_\theta \times 1} \end{bmatrix} = \mathbf{0}_{p \times 1} \end{aligned} \quad (15)$$

The zero Lie derivatives $L_{v_1} L_{g_i} L_f^k h$ lead to zero higher order Lie derivatives, i.e. $L_{v_r} \dots L_{v_1} L_{g_i} L_f^k h = \mathbf{0}_{p \times 1}$, which can be ignored when constructing the observability matrix since they do not deliver any observability information of the system.

200 Thanks to the equations (12) and (14), where all the nonzero Lie derivatives have been expressed in terms of the system matrices \mathbf{A} , \mathbf{B}_i , \mathbf{C} and the dynamic states \mathbf{x}_t , the expression of the observability matrix can be readily obtained by evaluating the gradients of the Lie derivatives. At the k^{th} iteration of the algorithm, one obtains:

$$\begin{aligned}\frac{\partial L_f^k h}{\partial \mathbf{x}} &= \begin{bmatrix} \mathbf{C} \mathbf{A}^k & \frac{\partial \mathbf{C} \mathbf{A}^k \mathbf{x}_t}{\partial \boldsymbol{\theta}} \end{bmatrix} \\ \frac{\partial L_{g_i} L_f^{k-1} h}{\partial \mathbf{x}} &= \begin{bmatrix} \mathbf{0}_{p \times n_t} & \frac{\partial \mathbf{C} \mathbf{A}^{k-1} \mathbf{B}_i}{\partial \boldsymbol{\theta}} \end{bmatrix}\end{aligned}\quad (16)$$

205 Then for $k = 0, \dots, n-1$ the full observability matrix of the system in the equations (7) and (8) attains the following form:

$$\mathbf{\Omega} = \begin{bmatrix} \mathbf{C} & \frac{\partial \mathbf{C} \mathbf{x}_t}{\partial \boldsymbol{\theta}} \\ \mathbf{C} \mathbf{A} & \frac{\partial \mathbf{C} \mathbf{A} \mathbf{x}_t}{\partial \boldsymbol{\theta}} \\ \mathbf{0}_{p \times n_t} & \frac{\partial \mathbf{C} \mathbf{B}_1}{\partial \boldsymbol{\theta}} \\ \vdots & \vdots \\ \mathbf{0}_{p \times n_t} & \frac{\partial \mathbf{C} \mathbf{B}_m}{\partial \boldsymbol{\theta}} \\ \vdots & \vdots \\ \mathbf{C} \mathbf{A}^k & \frac{\partial \mathbf{C} \mathbf{A}^k \mathbf{x}_t}{\partial \boldsymbol{\theta}} \\ \mathbf{0}_{p \times n_t} & \frac{\partial \mathbf{C} \mathbf{A}^{k-1} \mathbf{B}_1}{\partial \boldsymbol{\theta}} \\ \vdots & \vdots \\ \mathbf{0}_{p \times n_t} & \frac{\partial \mathbf{C} \mathbf{A}^{k-1} \mathbf{B}_m}{\partial \boldsymbol{\theta}} \\ \vdots & \vdots \\ \mathbf{C} \mathbf{A}^{n-1} & \frac{\partial \mathbf{C} \mathbf{A}^{n-1} \mathbf{x}_t}{\partial \boldsymbol{\theta}} \\ \mathbf{0}_{p \times n_t} & \frac{\partial \mathbf{C} \mathbf{A}^{n-2} \mathbf{B}_1}{\partial \boldsymbol{\theta}} \\ \vdots & \vdots \\ \mathbf{0}_{p \times n_t} & \frac{\partial \mathbf{C} \mathbf{A}^{n-2} \mathbf{B}_m}{\partial \boldsymbol{\theta}} \end{bmatrix} \quad (17)$$

4. Recursive computation

This section aims to compute the observability matrix of equation (17) in a recursive way. A key objective of the recursive formulation is to avoid the use

210 of symbolic differentiations and instead compute the submatrices involved in Ω
 by the operations of matrix addition and multiplication. While it is straight-
 forward to compute the submatrix \mathbf{CA}^k recursively as $\mathbf{CA}^{k-1}\mathbf{A}$, the recursive
 computations of $\frac{\partial \mathbf{CA}^k \mathbf{x}_t}{\partial \boldsymbol{\theta}}$ and $\frac{\partial \mathbf{CA}^{k-1} \mathbf{B}_i}{\partial \boldsymbol{\theta}}$ are realized by introducing a product
 rule to expand the gradient of matrix multiplication, which is analogous to the
 215 well known product rule for the derivative of product of two scalar functions:
 $(fg)' = f'g + fg'$.

4.1. A product rule for the gradient of matrix multiplication

Let \mathbf{U} be a 2-dimensional matrix with its $(i_1, i_2)^{th}$ element denoted by $U_{i_1 i_2}$,
 and \mathbf{z} be a vector with its l^{th} element z_l . The gradient of \mathbf{U} with respect to
 220 \mathbf{z} , $\mathbf{V} = \frac{\partial \mathbf{U}}{\partial \mathbf{z}}$, is a 3-dimensional matrix where the $(i_1, i_2, l)^{th}$ element of \mathbf{V} is
 calculated by $V_{i_1 i_2 l} = \frac{\partial U_{i_1 i_2}}{\partial z_l}$.

Let \mathbf{U} and \mathbf{M} be matrices with dimensions $\alpha \times \beta$ and $\beta \times \gamma$ respectively,
 and \mathbf{z} be a vector with dimension $\nu \times 1$. A product rule for the gradient of the
 matrix multiplication \mathbf{UM} with respect to \mathbf{z} is introduced as follows:

$$\frac{\partial \mathbf{UM}}{\partial \mathbf{z}} = \left(\frac{\partial \mathbf{U}}{\partial \mathbf{z}} \right)^{T_t} \mathbf{M} + \mathbf{U} \frac{\partial \mathbf{M}}{\partial \mathbf{z}} \quad (18)$$

225 where $\frac{\partial \mathbf{UM}}{\partial \mathbf{z}}$ is an $\alpha \times \gamma \times \nu$ matrix, $\frac{\partial \mathbf{U}}{\partial \mathbf{z}}$ is an $\alpha \times \beta \times \nu$ matrix and $\frac{\partial \mathbf{M}}{\partial \mathbf{z}}$ is
 a $\beta \times \gamma \times \nu$ matrix. The notation ' T_t ' over a 3-dimensional matrix is used to
 denote the transpose between the second dimension and the third dimension
 of the matrix. For example, \mathbf{V} under the transpose, \mathbf{V}^{T_t} , is a 3-dimensional
 matrix with its $(i_1, l, i_2)^{th}$ element equal to the $(i_1, i_2, l)^{th}$ element of \mathbf{V} , i.e.
 230 $(\mathbf{V}^{T_t})_{i_1 l i_2} = V_{i_1 i_2 l}$. The multi-dimensional matrix multiplication follows the
 rule of single contraction between the last dimension of the first matrix and the
 first dimension of the second matrix. For example, if $\mathbf{W} = \mathbf{U} \frac{\partial \mathbf{M}}{\partial \mathbf{z}}$, then \mathbf{W} is
 an $\alpha \times \gamma \times \nu$ matrix with its $(i_1, j_2, l)^{th}$ element calculated by:

$$W_{i_1 j_2 l} = \sum_{i_2=1}^{\beta} U_{i_1 i_2} \left(\frac{\partial \mathbf{M}}{\partial \mathbf{z}} \right)_{i_2 j_2 l} \quad (19)$$

Similarly, if \mathbf{M} is a $\beta \times 1$ vector, the product rule is given as:

$$\frac{\partial \mathbf{UM}}{\partial \mathbf{z}} = \left(\frac{\partial \mathbf{U}}{\partial \mathbf{z}} \right)^{T_t} \mathbf{M} + \mathbf{U} \frac{\partial \mathbf{M}}{\partial \mathbf{z}} \quad (20)$$

235 where $\frac{\partial \mathbf{U}\mathbf{M}}{\partial \mathbf{z}}$ is an $\alpha \times \nu$ Jacobian matrix and $\frac{\partial \mathbf{M}}{\partial \mathbf{z}}$ is a $\beta \times \nu$ Jacobian matrix.

Proof: Equation (18) is proved in the following relying on the concept of tensors. Referring to the Chapter 1 of [26], a matrix of any number of dimensions has a corresponding tensor form of the same number of orders. The matrix \mathbf{U} may be written as a second order tensor using the Einstein summation
 240 notation: $U_{i_1 i_2} \mathbf{e}_{i_1} \otimes \mathbf{e}_{i_2} (i_1 = 1, \dots, \alpha, i_2 = 1, \dots, \beta)$, where $U_{i_1 i_2}$ is the $(i_1, i_2)^{th}$ element of the tensor, \mathbf{e}_{i_1} and \mathbf{e}_{i_2} are directional unit vectors, \otimes is the tensor product. Similarly, \mathbf{M} and \mathbf{z} can be written as $M_{j_1 j_2} \mathbf{e}_{j_1} \otimes \mathbf{e}_{j_2} (j_1 = 1, \dots, \beta, j_2 = 1, \dots, \gamma)$ and $z_l \mathbf{e}_l (l = 1, \dots, \nu)$ respectively. The correspondence of the matrix multiplication, e.g. $\mathbf{U}\mathbf{M}$, in the tensor field is namely the dot product following
 245 the rule of tensor single contraction:

$$(U_{i_1 i_2} \mathbf{e}_{i_1} \otimes \mathbf{e}_{i_2}) \cdot (M_{j_1 j_2} \mathbf{e}_{j_1} \otimes \mathbf{e}_{j_2}) = U_{i_1 i_2} M_{i_2 j_2} \mathbf{e}_{i_1} \otimes \mathbf{e}_{j_2} \quad (21)$$

The gradient of the resulting tensor (21), which is the tensor form of the left hand side of equation (18) i.e. $\frac{\partial \mathbf{U}\mathbf{M}}{\partial \mathbf{z}}$, can then be written as:

$$\frac{\partial U_{i_1 i_2} M_{i_2 j_2} \mathbf{e}_{i_1} \otimes \mathbf{e}_{j_2}}{\partial z_l} \otimes \mathbf{e}_l = \frac{\partial U_{i_1 i_2} M_{i_2 j_2}}{\partial z_l} \mathbf{e}_{i_1} \otimes \mathbf{e}_{j_2} \otimes \mathbf{e}_l \quad (22)$$

Applying the standard product rule for the derivative of product between scalars yields:

$$\frac{\partial U_{i_1 i_2} M_{i_2 j_2}}{\partial z_l} \mathbf{e}_{i_1} \otimes \mathbf{e}_{j_2} \otimes \mathbf{e}_l = \left(\frac{\partial U_{i_1 i_2}}{\partial z_l} M_{i_2 j_2} + U_{i_1 i_2} \frac{\partial M_{i_2 j_2}}{\partial z_l} \right) \mathbf{e}_{i_1} \otimes \mathbf{e}_{j_2} \otimes \mathbf{e}_l \quad (23)$$

250 Next, it aims to follow the above tensor operations to calculate the tensor form of the right hand side of equation (18):

$$\left(\left(\frac{\partial U_{i_1 i_2}}{\partial z_l} \mathbf{e}_{i_1} \otimes \mathbf{e}_{i_2} \otimes \mathbf{e}_l \right)^{T_t} \cdot M_{j_1 j_2} \mathbf{e}_{j_1} \otimes \mathbf{e}_{j_2} \right)^{T_t} + U_{i_1 i_2} \mathbf{e}_{i_1} \otimes \mathbf{e}_{i_2} \cdot \frac{\partial M_{j_1 j_2}}{\partial z_l} \mathbf{e}_{j_1} \otimes \mathbf{e}_{j_2} \otimes \mathbf{e}_l \quad (24)$$

The earlier defined transpose ' T_t ' is used as:

$$\left(\frac{\partial U_{i_1 i_2}}{\partial z_l} \mathbf{e}_{i_1} \otimes \mathbf{e}_{i_2} \otimes \mathbf{e}_l \right)^{T_t} = \frac{\partial U_{i_1 i_2}}{\partial z_l} \mathbf{e}_{i_1} \otimes \mathbf{e}_l \otimes \mathbf{e}_{i_2} \quad (25)$$

Thus continuing from expression (24), the tensor form of the right hand side of

(18) can be simplified as:

$$\begin{aligned}
& \left(\frac{\partial U_{i_1 i_2}}{\partial z_l} \mathbf{e}_{i_1} \otimes \mathbf{e}_l \otimes \mathbf{e}_{i_2} \cdot M_{j_1 j_2} \mathbf{e}_{j_1} \otimes \mathbf{e}_{j_2} \right)^{T_t} + \\
& U_{i_1 i_2} \mathbf{e}_{i_1} \otimes \mathbf{e}_{i_2} \cdot \frac{\partial M_{j_1 j_2}}{\partial z_l} \mathbf{e}_{j_1} \otimes \mathbf{e}_{j_2} \otimes \mathbf{e}_l \\
& = \left(\frac{\partial U_{i_1 i_2}}{\partial z_l} M_{i_2 j_2} \mathbf{e}_{i_1} \otimes \mathbf{e}_l \otimes \mathbf{e}_{j_2} \right)^{T_t} + U_{i_1 i_2} \frac{\partial M_{i_2 j_2}}{\partial z_l} \mathbf{e}_{i_1} \otimes \mathbf{e}_{j_2} \otimes \mathbf{e}_l \\
& = \left(\frac{\partial U_{i_1 i_2}}{\partial z_l} M_{i_2 j_2} + U_{i_1 i_2} \frac{\partial M_{i_2 j_2}}{\partial z_l} \right) \mathbf{e}_{i_1} \otimes \mathbf{e}_{j_2} \otimes \mathbf{e}_l
\end{aligned} \tag{26}$$

255 which coincides with expression (23). Therefore the tensor forms of the *L.H.S* and *R.H.S* of equation (18) are equal and thus (18) is proved. Following the same way, equation (20) when \mathbf{M} is a $\beta \times 1$ vector can also be proved to hold. The tensor form of the left hand side of (20) is written as:

$$\frac{\partial U_{i_1 i_2} M_{i_2}}{\partial z_l} \mathbf{e}_{i_1} \otimes \mathbf{e}_l = \left(\frac{\partial U_{i_1 i_2}}{\partial z_l} M_{i_2} + U_{i_1 i_2} \frac{\partial M_{i_2}}{\partial z_l} \right) \mathbf{e}_{i_1} \otimes \mathbf{e}_l \tag{27}$$

While the tensor form of the right hand side of (20) becomes:

$$\left(\frac{\partial U_{i_1 i_2}}{\partial z_l} \mathbf{e}_{i_1} \otimes \mathbf{e}_{i_2} \otimes \mathbf{e}_l \right)^{T_t} \cdot M_{j_1} \mathbf{e}_{j_1} + U_{i_1 i_2} \mathbf{e}_{i_1} \otimes \mathbf{e}_{i_2} \cdot \frac{\partial M_{j_1}}{\partial z_l} \mathbf{e}_{j_1} \otimes \mathbf{e}_l \tag{28}$$

260 Expression (28) can be simplified as:

$$\begin{aligned}
& \frac{\partial U_{i_1 i_2}}{\partial z_l} \mathbf{e}_{i_1} \otimes \mathbf{e}_l \otimes \mathbf{e}_{i_2} \cdot M_{j_1} \mathbf{e}_{j_1} + U_{i_1 i_2} \mathbf{e}_{i_1} \otimes \mathbf{e}_{i_2} \cdot \frac{\partial M_{j_1}}{\partial z_l} \mathbf{e}_{j_1} \otimes \mathbf{e}_l \\
& = \frac{\partial U_{i_1 i_2}}{\partial z_l} M_{i_2} \mathbf{e}_{i_1} \otimes \mathbf{e}_l + U_{i_1 i_2} \frac{\partial M_{i_2}}{\partial z_l} \mathbf{e}_{i_1} \otimes \mathbf{e}_l \\
& = \left(\frac{\partial U_{i_1 i_2}}{\partial z_l} M_{i_2} + U_{i_1 i_2} \frac{\partial M_{i_2}}{\partial z_l} \right) \mathbf{e}_{i_1} \otimes \mathbf{e}_l
\end{aligned} \tag{29}$$

which coincides with expression (27). Therefore the tensor forms of the *L.H.S* and *R.H.S* of equation (20) are equal and thus (20) is proved.

4.2. Recursive computation of the observability matrix

Consider the submatrices $\frac{\partial \mathbf{C} \mathbf{A}^k \mathbf{x}_t}{\partial \boldsymbol{\theta}}$ and $\frac{\partial \mathbf{C} \mathbf{A}^{k-1} \mathbf{B}_i}{\partial \boldsymbol{\theta}}$, since $\mathbf{C} \mathbf{A}^k$ is a 2-dimensional
265 matrix, \mathbf{x}_t is a vector and $\boldsymbol{\theta}$ is a vector, $\frac{\partial \mathbf{C} \mathbf{A}^k \mathbf{x}_t}{\partial \boldsymbol{\theta}}$ is expanded according to equation (20):

$$\frac{\partial \mathbf{C} \mathbf{A}^k \mathbf{x}_t}{\partial \boldsymbol{\theta}} = \left(\frac{\partial \mathbf{C} \mathbf{A}^k}{\partial \boldsymbol{\theta}} \right)^{T_t} \mathbf{x}_t + \mathbf{C} \mathbf{A}^k \frac{\partial \mathbf{x}_t}{\partial \boldsymbol{\theta}} \tag{30}$$

$\frac{\partial \mathbf{x}_t}{\partial \boldsymbol{\theta}}$ is a zero matrix due to the dynamic states and the parameters being independent, and therefore equation (30) becomes:

$$\frac{\partial \mathbf{C} \mathbf{A}^k \mathbf{x}_t}{\partial \boldsymbol{\theta}} = \left(\frac{\partial \mathbf{C} \mathbf{A}^k}{\partial \boldsymbol{\theta}} \right)^{T_t} \mathbf{x}_t \quad (31)$$

Expressing $\mathbf{C} \mathbf{A}^k$ as $\mathbf{C} \mathbf{A}^{k-1} \mathbf{A}$, $\frac{\partial \mathbf{C} \mathbf{A}^k}{\partial \boldsymbol{\theta}}$ is expanded based on equation (18):

$$\frac{\partial \mathbf{C} \mathbf{A}^k}{\partial \boldsymbol{\theta}} = \left(\left(\frac{\partial \mathbf{C} \mathbf{A}^{k-1}}{\partial \boldsymbol{\theta}} \right)^{T_t} \mathbf{A} \right)^{T_t} + \mathbf{C} \mathbf{A}^{k-1} \frac{\partial \mathbf{A}}{\partial \boldsymbol{\theta}} \quad (32)$$

270 Taking the transpose ‘ T_t ’ of the both sides of equation (32) gives:

$$\left(\frac{\partial \mathbf{C} \mathbf{A}^k}{\partial \boldsymbol{\theta}} \right)^{T_t} = \left(\frac{\partial \mathbf{C} \mathbf{A}^{k-1}}{\partial \boldsymbol{\theta}} \right)^{T_t} \mathbf{A} + (\mathbf{C} \mathbf{A}^{k-1} \frac{\partial \mathbf{A}}{\partial \boldsymbol{\theta}})^{T_t} \quad (33)$$

Similarly, $\frac{\partial \mathbf{C} \mathbf{A}^{k-1} \mathbf{B}_i}{\partial \boldsymbol{\theta}}$ is expanded using (20):

$$\frac{\partial \mathbf{C} \mathbf{A}^{k-1} \mathbf{B}_i}{\partial \boldsymbol{\theta}} = \left(\frac{\partial \mathbf{C} \mathbf{A}^{k-1}}{\partial \boldsymbol{\theta}} \right)^{T_t} \mathbf{B}_i + \mathbf{C} \mathbf{A}^{k-1} \frac{\partial \mathbf{B}_i}{\partial \boldsymbol{\theta}} \quad (34)$$

The equations (31), (33) and (34) can then be used to recursively update the submatrices $\frac{\partial \mathbf{C} \mathbf{A}^k \mathbf{x}_t}{\partial \boldsymbol{\theta}}$ and $\frac{\partial \mathbf{C} \mathbf{A}^{k-1} \mathbf{B}_i}{\partial \boldsymbol{\theta}}$, i.e. $\frac{\partial \mathbf{C} \mathbf{A}^k \mathbf{x}_t}{\partial \boldsymbol{\theta}}$ at step k is computed from $(\frac{\partial \mathbf{C} \mathbf{A}^k}{\partial \boldsymbol{\theta}})^{T_t}$ at step k based on equation (31); $(\frac{\partial \mathbf{C} \mathbf{A}^k}{\partial \boldsymbol{\theta}})^{T_t}$ at step k and $\frac{\partial \mathbf{C} \mathbf{A}^{k-1} \mathbf{B}_i}{\partial \boldsymbol{\theta}}$ at
275 step $k-1$ are computed from $(\frac{\partial \mathbf{C} \mathbf{A}^{k-1}}{\partial \boldsymbol{\theta}})^{T_t}$ at step $k-1$ and $\mathbf{C} \mathbf{A}^{k-1}$ at step $k-1$ based on the equations (33) and (34) respectively. The constant terms during the recursion, including $\frac{\partial \mathbf{A}}{\partial \boldsymbol{\theta}}$ and $\frac{\partial \mathbf{B}_i}{\partial \boldsymbol{\theta}}$, can be pre-computed at the initialization step.

For clarity of the presentation, the symbols \mathbf{a}_k , \mathbf{b}_k , \mathbf{c}_{k-1} and \mathbf{d}_k are introduced to denote $\mathbf{C} \mathbf{A}^k$, $\frac{\partial \mathbf{C} \mathbf{A}^k \mathbf{x}_t}{\partial \boldsymbol{\theta}}$, $\frac{\partial \mathbf{C} \mathbf{A}^{k-1} \mathbf{B}_i}{\partial \boldsymbol{\theta}}$ and $(\frac{\partial \mathbf{C} \mathbf{A}^k}{\partial \boldsymbol{\theta}})^{T_t}$ respectively. At step
280 k , the matrices in equation (16) are computed by:

$$\begin{aligned} \frac{\partial \mathbf{L}_f^k \mathbf{h}}{\partial \mathbf{x}} &= \begin{bmatrix} \mathbf{a}_k & \mathbf{b}_k \end{bmatrix} \\ \frac{\partial \mathbf{L}_{g_i} \mathbf{L}_f^{k-1} \mathbf{h}}{\partial \mathbf{x}} &= \begin{bmatrix} \mathbf{0}_{p \times n_t} & \mathbf{c}_{k-1} \end{bmatrix} \end{aligned} \quad (35)$$

where \mathbf{a}_k is computed by:

$$\mathbf{a}_k = \mathbf{a}_{k-1} \mathbf{A} \quad (36)$$

\mathbf{b}_k is computed by equation (31):

$$\mathbf{b}_k = \mathbf{d}_k \mathbf{x}_t \quad (37)$$

d_k is computed by equation (33):

$$d_k = d_{k-1}A + (a_{k-1} \frac{\partial A}{\partial \theta})^{T_t} \quad (38)$$

285 and c_{k-1} is computed by equation (34):

$$c_{k-1} = d_{k-1}B_i + a_{k-1} \frac{\partial B_i}{\partial \theta} \quad (39)$$

with $a_0 = C$ and $d_0 = (\frac{\partial C}{\partial \theta})^{T_t}$ at the initial step. A flowchart diagram is presented in Figure 1 showing the recursive formulation can be used as an algorithm to output the observability matrix of the system in the equations (7) and (8).

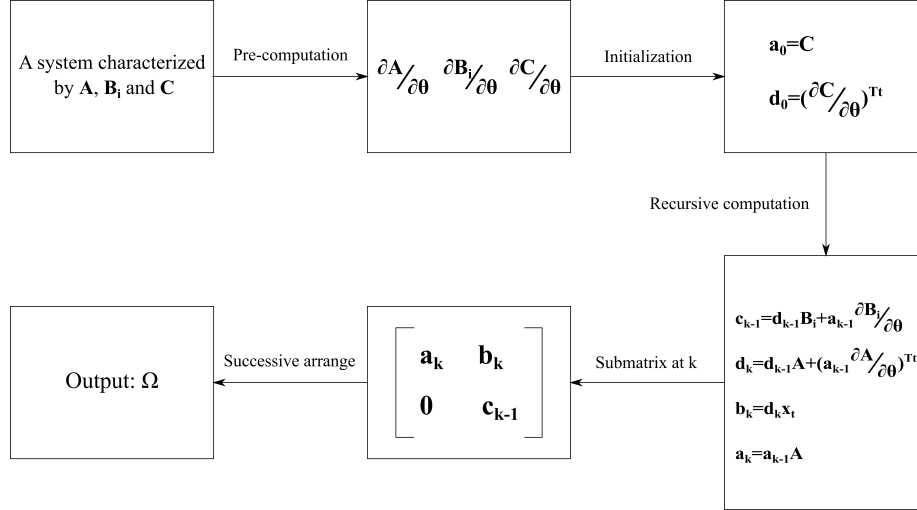


Figure 1: Recursive computation of the observability matrix

290 5. Robust algorithm

5.1. Numerical realization

The previously derived algorithm can be implemented symbolically using symbolic computing tools such as MATLAB symbolic toolbox [27], Mathematica [28] and Maple [29]. It should be noted that introducing such a robust recursive evaluation of the observability matrix, and more importantly the involved Lie

295

derivatives in symbolic form, without requiring higher-order differentiations is crucial and would find wide applications in dynamic problems related to Lie algebra, although those applications beyond observability is not within the scope of this work. On the other hand, an advantage of avoiding symbolic differentia-

300 tions, with the exception of the pre-computation in Figure 1, is that numerical realization of the algorithm becomes possible. This can be achieved by specializing the dynamic states and the parameters on some randomly chosen values: $\mathbf{x}_t = \tilde{\mathbf{x}}_t$, $\boldsymbol{\theta} = \tilde{\boldsymbol{\theta}}$. As a result, $\mathbf{A}(\tilde{\boldsymbol{\theta}})$, $\mathbf{B}_i(\tilde{\boldsymbol{\theta}})$, $\mathbf{C}(\tilde{\boldsymbol{\theta}})$, $\frac{\partial \mathbf{A}}{\partial \boldsymbol{\theta}}(\tilde{\boldsymbol{\theta}})$, $\frac{\partial \mathbf{B}_i}{\partial \boldsymbol{\theta}}(\tilde{\boldsymbol{\theta}})$ and $\frac{\partial \mathbf{C}}{\partial \boldsymbol{\theta}}(\tilde{\boldsymbol{\theta}})$ at the initialization are all numerical matrices, the following matrix operations are
305 all numerical, and consequently the algorithm is expected to output a numerical observability matrix $\boldsymbol{\Omega}(\tilde{\mathbf{x}}_t, \tilde{\boldsymbol{\theta}})$.

For observability problems, one is interested in the generic rank rather than reduced ranks of the observability matrix that might occur at certain realizations of the states corresponding to singular points. Such singular points are
310 more relevant for applications involving controllers [15]. Given that it is highly unlikely to pick those singular points in a random process, the rank of $\boldsymbol{\Omega}(\tilde{\mathbf{x}}_t, \tilde{\boldsymbol{\theta}})$ is expected to be the same as the generic rank of the corresponding symbolic matrix $\boldsymbol{\Omega}(\mathbf{x}_t, \boldsymbol{\theta})$. A similar discussion can be found in [23] for development of a semi-numerical algorithm to test the observability of nonlinear algebraic
315 systems, where the initial conditions of dynamic states and parameters of the systems are specialized on random integers. To summarize, a complete description of the proposed algorithm with numerical realizations is presented in the following as Algorithm II.

Algorithm II

320 **Input to the algorithm:** the matrices $\mathbf{A}, \mathbf{B}_1, \dots, \mathbf{B}_m$ and \mathbf{C} , the dynamic states \mathbf{x}_t and the parameters $\boldsymbol{\theta}$ of the system in equation (7)

Output from the algorithm: the observability matrix $\boldsymbol{\Omega}$

Preprocessing: Compute $\frac{\partial \mathbf{A}}{\partial \boldsymbol{\theta}}, \frac{\partial \mathbf{B}_1}{\partial \boldsymbol{\theta}}, \dots, \frac{\partial \mathbf{B}_m}{\partial \boldsymbol{\theta}}$ and $\frac{\partial \mathbf{C}}{\partial \boldsymbol{\theta}}$ symbolically

Initialization: Set $k = 0$. Choose random numerical realizations for $\mathbf{x}_t = \tilde{\mathbf{x}}_t$,
325 $\boldsymbol{\theta} = \tilde{\boldsymbol{\theta}}$ where $\tilde{\mathbf{x}}_t, \tilde{\boldsymbol{\theta}} \in \mathbb{R}$. Let $\mathbf{a}_0 = \mathbf{C}$, $\mathbf{d}_0 = (\frac{\partial \mathbf{C}}{\partial \boldsymbol{\theta}})^T$, $\mathbf{b}_0 = \mathbf{d}_0 \mathbf{x}_t$, $\boldsymbol{\Omega}_0 = \begin{bmatrix} \mathbf{a}_0 & \mathbf{b}_0 \end{bmatrix}$

1. From Eq. (39), compute $\mathbf{c}_{k,1} = \mathbf{d}_k \mathbf{B}_1 + \mathbf{a}_k \frac{\partial \mathbf{B}_1}{\partial \boldsymbol{\theta}}$, ..., $\mathbf{c}_{k,m} = \mathbf{d}_k \mathbf{B}_m + \mathbf{a}_k \frac{\partial \mathbf{B}_m}{\partial \boldsymbol{\theta}}$

2. From Eq. (38), compute $\mathbf{d}_{k+1} = \mathbf{d}_k \mathbf{A} + (\mathbf{a}_k \frac{\partial \mathbf{A}}{\partial \boldsymbol{\theta}})^{T_t}$

3. From Eq. (37), compute $\mathbf{b}_{k+1} = \mathbf{d}_{k+1} \mathbf{x}_t$

4. From Eq. (36), compute $\mathbf{a}_{k+1} = \mathbf{a}_k \mathbf{A}$

330 5. Arrange the submatrices from steps 1, 2, 3 and 4 in the observability matrix

$$\mathbf{\Omega}_{k+1} = \mathbf{\Omega}_k \cup \begin{bmatrix} \mathbf{a}_{k+1} & \mathbf{b}_{k+1} \\ \mathbf{0}_{p \times n_t} & \mathbf{c}_{k,1} \\ \vdots & \vdots \\ \mathbf{0}_{p \times n_t} & \mathbf{c}_{k,m} \end{bmatrix}$$

6. Compute the rank, if $\text{rank}(\mathbf{\Omega}_{k+1}) = \text{rank}(\mathbf{\Omega}_k)$, or $\text{rank}(\mathbf{\Omega}_{k+1}) = n$, or $k = n - 2$, end and output $\mathbf{\Omega} = \mathbf{\Omega}_{k+1}$

7. Let $k = k + 1$ and go to step 1

335

5.2. Symbolic numbers vs. Floating-point numbers

In terms of the type of numerical operations in a computing system, using floating-point numbers [30] is the most efficient way to process the algorithm. However, a drawback is that floating-points are of limited precision to represent real numbers. The errors brought by the lost precision would propagate and accumulate through the recursive computation and can eventually exert a significant impact on the numerical rank of the observability matrix. Moreover, floating-point numbers have to be limited within a range defined by the maximum and minimum representable numbers. When the algorithm is implemented for large systems, the occurring numbers during the recursion can easily exceed those limits.

Symbolic numbers: A considerable alternative is to use symbolic numbers instead of floating-point numbers as symbolic numbers are not only always exact but also infinitely representable. Operations between symbolic numbers are substantially cheaper than operations between symbols. Nevertheless, processing of symbolic numbers is less efficient than that of floating point numbers especially for large systems.

Integers in floating-point format with application of modular op-

erations: The drawbacks of using floating-point numbers, related to the issues of limited precision and number growth, can be effectively alleviated by using integers, i.e. $\tilde{\mathbf{x}}_t, \tilde{\boldsymbol{\theta}} \in \mathbb{Z}$, with application of modular operations. The modular operation of a rational number a , i.e. an integer or a fraction of two integers, using a positive prime number p is denoted by $a \bmod p$. The rule of modular arithmetic can be simply described as follows [31]: assuming a is an integer, $a \bmod p$ takes the remainder of a divided by p , and the result is a positive integer between 0 and $p - 1$; if the greatest common divisor of a and p is 1, then based on the rule of modular multiplicative inverse, $a^{-1} \bmod p$ is a positive integer b between 1 and $p - 1$ such that $ab \bmod p = 1$. To perform the modular operations on the algorithm, a large prime number p is selected and a set of integers between 1 and $p - 1$ are randomly chosen for $\tilde{\mathbf{x}}_t$ and $\tilde{\boldsymbol{\theta}}$. Then all the involved computations in the algorithm are operated by $\bmod p$. Let $\mathbf{a}_{k,p}$, $\mathbf{b}_{k,p}$, $\mathbf{c}_{k,p}$ and $\mathbf{d}_{k,p}$ denote the recursive terms after applying the modular operations to the equations (39), (38), (37) and (36),

$$\begin{aligned}
\mathbf{c}_{k-1,p} &= ((\mathbf{d}_{k-1,p} \mathbf{B}_i) \bmod p + (\mathbf{a}_{k-1,p} \frac{\partial \mathbf{B}_i}{\partial \boldsymbol{\theta}}) \bmod p) \bmod p \\
\mathbf{d}_{k,p} &= ((\mathbf{d}_{k-1,p} \mathbf{A}) \bmod p + (\mathbf{a}_{k-1,p} \frac{\partial \mathbf{A}}{\partial \boldsymbol{\theta}})^{T_t} \bmod p) \bmod p \\
\mathbf{b}_{k,p} &= (\mathbf{d}_{k,p} \mathbf{x}_t) \bmod p \\
\mathbf{a}_{k,p} &= (\mathbf{a}_{k-1,p} \mathbf{A}) \bmod p
\end{aligned} \tag{40}$$

where the modulo operations are applied to the matrices element-wise.

The following distributive law [31] is used to account for the usefulness of the modular operations on the algorithm,

$$\begin{aligned}
(E \bmod p + F \bmod p) \bmod p &= (E + F) \bmod p \\
(E \bmod p)(F \bmod p) \bmod p &= EF \bmod p
\end{aligned} \tag{41}$$

where the law holds when E and F are either rational numbers or matrices with

rational elements. Based on equation (41), equation (40) is equivalent to:

$$\begin{aligned}
c_{k-1,p} &= (d_{k-1,p}B_i + a_{k-1,p} \frac{\partial B_i}{\partial \theta}) \mod p \\
d_{k,p} &= (d_{k-1,p}A + (a_{k-1,p} \frac{\partial A}{\partial \theta})^{T_t}) \mod p \\
b_{k,p} &= (d_{k,p}x_t) \mod p \\
a_{k,p} &= (a_{k-1,p}A) \mod p
\end{aligned} \tag{42}$$

and consequently, it can be deduced that the occurring observability matrix from
 375 the algorithm using modular operations, denoted by Ω_p , satisfies the following
 property:

$$\Omega_p = \Omega \mod p \tag{43}$$

The observability of a tested system is then determined by evaluating the rank
 of Ω_p over a finite field \mathbb{F}_p through applying modular operations to Gaussian
 elimination [23, 31, 32, 33]. As discussed in [23], if Ω_p is a full-rank matrix, Ω
 380 must also be full-rank, both observability matrices agreeing to imply the tested
 system is observable. If Ω_p is not of full-rank, it is highly likely that Ω is not
 of full-rank either. An exception where Ω would be of full-rank despite Ω_p not
 being of full-rank may occur when the nonzero determinant of the largest square
 submatrix of Ω vanishes modulo p (if Ω is a square matrix, for example, the
 385 case would be $\det(\Omega) \mod p = 0$). [23] suggested an approach to estimate the
 upper bound of the probability of this case where Ω_p and Ω deliver different
 rank information, and it was shown that the likelihood of this occurring becomes
 practically negligible if the prime number p is selected to be large. Therefore,
 Ω_p is capable of accurately describing the observability of a tested system with
 390 a very high probability of success.

The usefulness of the modular operations allows for proposing Algorithm III
 which makes use of **integers in floating-point format** for numerical realizations.

Algorithm III

Input to the algorithm: the matrices A, B_1, \dots, B_m and C , the dynamic
 395 states x_t and the parameters θ of the system in equation (7)

Output from the algorithm: the observability matrix Ω_p

Preprocessing: Compute $\frac{\partial \mathbf{A}}{\partial \boldsymbol{\theta}}, \frac{\partial \mathbf{B}_1}{\partial \boldsymbol{\theta}}, \dots, \frac{\partial \mathbf{B}_m}{\partial \boldsymbol{\theta}}$ and $\frac{\partial \mathbf{C}}{\partial \boldsymbol{\theta}}$ symbolically

Initialization: Set $k = 0$. Select a large positive prime number p . Choose random numerical realizations for $\mathbf{x}_t = \tilde{\mathbf{x}}_t$, $\boldsymbol{\theta} = \tilde{\boldsymbol{\theta}}$ where $\tilde{\mathbf{x}}_t, \tilde{\boldsymbol{\theta}} \in \{1, \dots, p-1\}$

400 in floating-point format. Let $\mathbf{a}_{0,p} = \mathbf{C}$, $\mathbf{d}_{0,p} = (\frac{\partial \mathbf{C}}{\partial \boldsymbol{\theta}})^{T_t}$, $\mathbf{b}_{0,p} = \mathbf{d}_{0,p} \mathbf{x}_t$, $\boldsymbol{\Omega}_{0,p} = \begin{bmatrix} \mathbf{a}_{0,p} & \mathbf{b}_{0,p} \end{bmatrix}$. All the matrices *mod* p

1. Based on Eq.(40), compute

$$\mathbf{c}_{k,p,i} = ((\mathbf{d}_{k,p} \mathbf{B}_i) \bmod p + (\mathbf{a}_{k,p} \frac{\partial \mathbf{B}_i}{\partial \boldsymbol{\theta}}) \bmod p) \bmod p \quad (i = 1, \dots, m)$$

$$2. \mathbf{d}_{k+1,p} = ((\mathbf{d}_{k,p} \mathbf{A}) \bmod p + (\mathbf{a}_{k,p} \frac{\partial \mathbf{A}}{\partial \boldsymbol{\theta}})^{T_t} \bmod p) \bmod p$$

405 3. $\mathbf{b}_{k+1,p} = (\mathbf{d}_{k+1,p} \mathbf{x}_t) \bmod p$

$$4. \mathbf{a}_{k+1,p} = (\mathbf{a}_{k,p} \mathbf{A}) \bmod p$$

5. Arrange the submatrices from steps 1, 2, 3 and 4 in the observability matrix

$$\boldsymbol{\Omega}_{k+1,p} = \boldsymbol{\Omega}_{k,p} \cup \begin{bmatrix} \mathbf{a}_{k+1,p} & \mathbf{b}_{k+1,p} \\ \mathbf{0}_{p \times n_t} & \mathbf{c}_{k,p,1} \\ \vdots & \vdots \\ \mathbf{0}_{p \times n_t} & \mathbf{c}_{k,p,m} \end{bmatrix} \bmod p$$

6. Compute the rank over \mathbb{F}_p , if $\text{rank}(\boldsymbol{\Omega}_{k+1,p}) = \text{rank}(\boldsymbol{\Omega}_{k,p})$, or $\text{rank}(\boldsymbol{\Omega}_{k+1,p}) =$

410 n , or $k = n - 2$, end and output $\boldsymbol{\Omega}_p = \boldsymbol{\Omega}_{k+1,p}$

7. Let $k = k + 1$ and go to step 1

Note that Algorithm II through the use of symbolic numbers is more computationally expensive than Algorithm III, although the latter comes at the price of a very small probability of predicting a wrong rank for the observability matrix. In any practical cases, this probability is negligibly small as has been explained earlier, and to further exclude the unlikely occurrence of the failure of the algorithm users could repeat the numerical realizations using different values of $\tilde{\mathbf{x}}_t$ and $\tilde{\boldsymbol{\theta}}$ and the selection of p , e.g. one could further cross-examine the results versus another random realization of the dynamic states and parameters, and another selection of a larger prime number. A second limitation of Algorithm III is that the elements of the matrices \mathbf{A} , \mathbf{B}_i and \mathbf{C} must be rational functions of $\boldsymbol{\theta}$ i.e. polynomial functions or fractions of polynomial functions, while for Algorithm II those functions can be analytic with respect to $\boldsymbol{\theta}$.

420

6. Examples

6.1. Example 1: N degrees of freedom mass-spring system

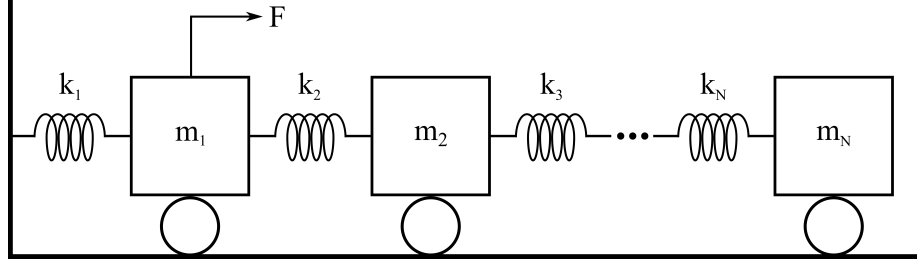


Figure 2: N degrees of freedom mass-spring system

The purpose of this example is to illustrate the use of the proposed Algorithm II and Algorithm III, and compare their efficiency versus the standard implementation of the ORC i.e. Algorithm I. The dynamical system shown in Figure 2 is an N degrees of freedom (DOFs) mass-spring system with the i^{th} mass and spring stiffness denoted by m_i and k_i respectively. The displacement and velocity of m_i are denoted by x_i and \dot{x}_i respectively. The dynamical system is subject to a measured force F applied on the first mass. Suppose all of the masses and stiffness are to be identified, and the displacements and velocities of the system are to be tracked using a set of measurements. θ is then written as:

$$\theta = \begin{bmatrix} k_1 & k_2 & \dots & k_N & m_1 & m_2 & \dots & m_N \end{bmatrix}^T \quad (44)$$

and \mathbf{x}_t is:

$$\mathbf{x}_t = \begin{bmatrix} x_1 & x_2 & \dots & x_N & \dot{x}_1 & \dot{x}_2 & \dots & \dot{x}_N \end{bmatrix}^T \quad (45)$$

The system matrix \mathbf{A} is of the form:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{N \times N} & \mathbf{I}_{N \times N} \\ -\mathbf{M}^{-1}\mathbf{K} & \mathbf{0}_{N \times N} \end{bmatrix} \quad (46)$$

where \mathbf{M} is the diagonal mass matrix and \mathbf{K} is the stiffness matrix:

$$\mathbf{M} = \begin{bmatrix} m_1 & 0 & \dots & 0 \\ 0 & m_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & m_N \end{bmatrix}, \mathbf{K} = \begin{bmatrix} k_1 + k_2 & -k_2 & 0 & \dots & 0 \\ -k_2 & k_2 + k_3 & -k_3 & \dots & 0 \\ 0 & -k_3 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & k_{N-1} + k_N & -k_N \\ 0 & 0 & \dots & -k_N & k_N \end{bmatrix} \quad (47)$$

440 Taking into account there is only a single force applied on m_1 , the input matrix \mathbf{B}_i with $i = 1$ is obtained as:

$$\mathbf{B}_1 = \begin{bmatrix} \mathbf{0}_{1 \times N} & \frac{1}{m_1} & \mathbf{0}_{1 \times (N-1)} \end{bmatrix}^T \quad (48)$$

The form of the output matrix \mathbf{C} depends on what the output measurements are. For example, if the measurement is the displacement of m_i , i.e. x_i , then \mathbf{C} is expressed as:

$$\mathbf{C} = \begin{bmatrix} \mathbf{0}_{1 \times (i-1)} & 1 & \mathbf{0}_{1 \times (2N-i)} \end{bmatrix} \quad (49)$$

445 *Three degrees of freedom system*

First, the observability of a three degrees of freedom system, i.e. $N = 3$, is tested. This system has already been studied in [11] using Algorithm I and the results are used for comparison herein. The observability of the system is investigated for three different measurement scenarios where in each the displacement of a single mass is measured, i.e. x_1 , x_2 and x_3 . The output results from Algorithm II and Algorithm III are presented in Table 1 where the observability and identifiability of the individual state is detected through rank testing after obtaining the observability matrix, as described in Section 2. The results from Algorithm I, Algorithm II and Algorithm III are in full agreement and hence provide users with exactly the same observability information of the system. In the MATLAB implementation available at the link in footnote 2, the authors have included an example file that allows the reader to use the code for the cases of the observability problem studied in Table 1.

	Algorithm II			Algorithm III		
Measurement	x_1	x_2	x_3	x_1	x_2	x_3
Rank(Ω)	12	11	10	12	11	10
Observable states	$x_1, x_2, x_3,$ $\dot{x}_1, \dot{x}_2, \dot{x}_3,$ $k_1, k_2, k_3,$ m_1, m_2, m_3			$x_1, x_2, x_3,$ $\dot{x}_1, \dot{x}_2, \dot{x}_3,$ $k_1, k_2, k_3,$ m_1, m_2, m_3		
Unobservable states	$x_1, \dot{x}_1, k_2,$ $k_3, m_1, m_2,$ m_3			$x_1, \dot{x}_1, k_2,$ $k_3, m_1, m_2,$ m_3		

Table 1: The observability of the 3 DOFs mass-spring system using Algorithm II and Algorithm III

Large systems

460 Having built the confidence in the proposed algorithms, the observability tests are performed on the system in Figure 2 with an increasing number of masses: $N = 7, 20, 40, 60, 80, 100, 200, 500$, in order to compare the capability and efficiency of Algorithm I, Algorithm II and Algorithm III. All the algorithms are implemented in MATLAB R2016a on a desktop computer with the Intel Core i7-6700 CPU (3.40GHz) and 16GB RAM. The elapsed time of implementation 465 is recorded in seconds for each test, as shown in Table 2. The elapsed time does not count the time of preprocessing since the symbolic computations of $\frac{\partial A}{\partial \theta}$, $\frac{\partial B_1}{\partial \theta}$ and $\frac{\partial C}{\partial \theta}$ are common for the proposed algorithms. If a test is not completed within 48 hours or the computer runs out of memory, the result is 470 indicated by the notation ‘-’.

The displacement x_1 is used as the measurement for all the systems and the force is always assumed to be applied on the first mass. It is worth mentioning that Franco et al. [10] have theoretically proved that a mass-spring system with any number of DOFs is observable when i) the excitation is located at the first mass m_1 and the measurement is the displacement x_1 , ii) the excitation 475 is located at the last mass m_N and the measurement is the displacement x_N or iii) the excitation is located at any mass m_i and the measurements are the

N	Algorithm I	Algorithm II	Algorithm III
7	5698.73	0.46	0.02
20	-	6.57	0.09
40	-	45.84	1.18
60	-	154.49	6.19
80	-	393.71	19.11
100	-	884.04	46.74
200	-	16888.09	733.56
500	-	-	23014.22

Table 2: The elapsed time (seconds) of observability testing for large systems

displacements x_i and x_{i-1} . The scenarios considered herein fall within case i).

The observability results of all successful implementations are that the system is observable as expected for this case. As can be seen in Table 2, Algorithm I successfully tests a system of 7 DOFs, but it is unable to give any results of observability for the systems with 20 and hence more DOFs within 48 hours. In comparison, the proposed algorithm either using symbolic numbers or floating-point integers is capable of dealing with a large system with several hundreds of DOFs within an acceptable period of time. When implementing Algorithm II using symbolic numbers, the elapsed time grows significantly with the size of the system. Algorithm III is more efficient due to fast processing of floating-points in a computer, and therefore it is applicable to larger systems.

The reported efficiency of implementations might be further improved e.g. by parallelizing the algorithms, but the focus of this study is on the relative differences between the algorithms. It should be mentioned that the main limitation with Algorithm I is its high RAM requirements. The implementations of Algorithm II and Algorithm III are efficient in the usage of RAM and thus can avoid problems with exceeding the RAM available to MATLAB. This then allows Algorithm II and Algorithm III to improve their efficiency by parallelly

making use of more cores in the computer. However, it is difficult for Algorithm I to improve if the RAM requirements are the critical limiting factor.

Using this example, the capabilities of Algorithm II and Algorithm III are also compared to two additional existing observability algorithms in the literature: the Observability Test implemented in Maple [23] and STRIKE-GOLDD implemented in MATLAB [17]. All the algorithms in comparison are implemented on the same desktop computer to examine the maximum number of DOFs they can handle for the system in Figure 2, and the results are presented in Table 3. The STRIKE-GOLDD algorithm using the default options is capable of handling shear systems with less than 10 DOFs. This is expected, as STRIKE-GOLDD is an implementation of the ORC using the variation originally presented in [2], rather than exploiting the separation of Lie derivatives used for affine-input systems that for example was used in Isidori’s implementation [20], and also implanted in this work for Algorithm I. Other implementations that are very closely related to the ORC are also expected to have a similar performance to that of Algorithm I, as illustrated with the investigation of the performance of STRIKE-GOLDD in Table 3.

The upper limit of the Observability Test is around 60 DOFs depending on the available computer memory resources to Maple. The results are also affected by the efficiency of Maple, the computer architecture, the specifications of the workstation and the operating system used. The effects of the aforementioned factors on the performance of the algorithm are beyond the scope of this paper. As such, implementations of the Observability Test in other languages such as the Mathematica implementation suggested in [34] are not examined in this paper. Such variations can improve the performance, but as the underlying algorithm remains the same, the order of the largest examined system is expected to be similar and somewhere in the range of up to 100 DOFs. In comparison, as shown in Table 2, Algorithm II and Algorithm III are capable of handling systems with order of larger than 200 and 500 DOFs respectively.

However it should be noted that the Observability Test and the original ORC, e.g. as represented by Algorithm I, are originally designed for more gen-

Algorithms	Max. N	Applicable dynamics
Observability Test	~ 60	rational nonlinear
STRIKE-GOLDD	~ 10	analytic nonlinear

Table 3: The capabilities of the Observability Test and STRIKE-GOLDD compared to Algorithm II and Algorithm III

eral nonlinear dynamical systems, hence the superior performance of the proposed algorithms achieved herein is a consequence of focusing on linear systems with unknown parameters. Additionally, it should be highlighted that even for linear systems the relative performance of the suggested algorithms to the Observability Test is dependent on the computer architecture and the handling of symbolic maths, both of which may rapidly change over the next decades. However, it is important to notice that the suggested algorithms in this paper offer a means of testing large linear systems with unknown parameters, with what would be considered to be a computer of modest specifications at the time of writing this paper.

Viscously damped system

In this part, the existence of rate-proportional viscous damping forces associated with a corresponding damping matrix \mathbf{C}_d is considered for the system in Figure 2. \mathbf{A} matrix of such a mass-spring-damper system becomes:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{N \times N} & \mathbf{I}_{N \times N} \\ -\mathbf{M}^{-1}\mathbf{K} & -\mathbf{M}^{-1}\mathbf{C}_d \end{bmatrix} \quad (50)$$

Algorithm II and Algorithm III are applied to examine the observability of the system with different forms of the damping matrix, including i) a full symmetric \mathbf{C}_d , ii) \mathbf{C}_d corresponding to having viscous dashpots between successive masses and iii) a Rayleigh, mass and stiffness proportional damping matrix. In case i),

545 the damping matrix can be generally written as:

$$\mathbf{C}_d = \begin{bmatrix} c_{11} & \dots & c_{1N} \\ & \ddots & \vdots \\ sym. & & c_{NN} \end{bmatrix} \quad (51)$$

where c_{11}, \dots, c_{NN} are the damping parameters to be identified and thus included in the vector of parameters $\boldsymbol{\theta}$. Case ii) results in:

$$\mathbf{C}_d = \begin{bmatrix} c_1 + c_2 & -c_2 & 0 & \dots & 0 \\ -c_2 & c_2 + c_3 & -c_3 & \dots & 0 \\ 0 & -c_3 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & c_{N-1} + c_N & -c_N \\ 0 & 0 & \dots & -c_N & c_N \end{bmatrix} \quad (52)$$

where the coefficients of the viscous dashpots c_1, \dots, c_N are the parameters to be identified. The last case focuses on the form:

$$\mathbf{C}_d = \alpha \mathbf{M} + \beta \mathbf{K} \quad (53)$$

550 where α and β are scalar parameters to be identified.

If the system has 100 masses, i.e. $N = 100$, the occurring 100 DOFs system can be viewed as the model of a 100-floor shear building. For simplicity, it is still assumed that a single force is applied on the first floor. Note that because of the inclusion of a damping matrix whose parameters are to be identified, the conclusions on the observability of the system in [10] are not applicable anymore. 555 Suppose ten sensors are installed at the floors following two different sensor configurations where in each five displacements and five accelerations are measured: configuration 1) the sensors are installed at the top ten floors measuring $x_{100}, x_{99}, x_{98}, x_{97}, x_{96}, \ddot{x}_{95}, \ddot{x}_{94}, \ddot{x}_{93}, \ddot{x}_{92}, \ddot{x}_{91}$ and configuration 2) the sensors are 560 installed at every ten floors measuring $x_{100}, \ddot{x}_{90}, x_{80}, \ddot{x}_{70}, x_{60}, \ddot{x}_{50}, x_{40}, \ddot{x}_{30}, x_{20}, \ddot{x}_{10}$. Using the two sensor setups the observability of the system is examined under the three different considerations of damping. As is often the case in practice, there is also a consideration of whether the masses of the system are deemed as

	Sensor configuration 1		Sensor configuration 2	
Damping matrix	Mass parameters	Observability	Mass parameters	Observability
Case i)	known	unobservable	known	unobservable
	unknown	unobservable	unknown	unobservable
Case ii)	known	observable	known	observable
	unknown	observable	unknown	observable
Case iii)	known	observable	known	observable
	unknown	unobservable	unknown	observable

Table 4: The observability of a 100 DOFs mass-spring-damper system

known or unknown. The observability results of all the scenarios are presented
565 in Table 4, where a cell gives the observability property of the system under one
of the sensor configurations, one of the assumed damping matrices and either
the masses being considered as known or unknown.

As shown in Table 4, the system with case i) a full symmetric damping matrix
is highly unobservable regardless of whether the masses are assumed to be known
570 or unknown and under any of the two sensor setups. It implies that a method
that aims to identify the general matrices of mass, stiffness and damping, as for
example the subspace identification method would attempt to do [35], would
not succeed under certain sensor setups and applied measured force(s). On
the contrary, case ii) the damping matrix makes the system observable for all
575 the scenarios. Interestingly, this is typically the form of damping that one
would assume when using for example the methods of nonlinear Kalman filters
[36]. The system with case iii) the Rayleigh damping matrix is observable
when the masses are known but unobservable when the masses are unknown
given the sensor configuration 1. A further observability analysis of this case
580 shows that, when the mass parameters are unknown, both the Rayleigh damping
coefficients α and β are identifiable but the stiffness and the masses are not
separately identifiable. This is an indication that while one would not be able
to identify those parameters, the combined $\mathbf{M}^{-1}\mathbf{K}$ and hence the frequencies
of the system might be identified successfully. Unlike the sensor configuration

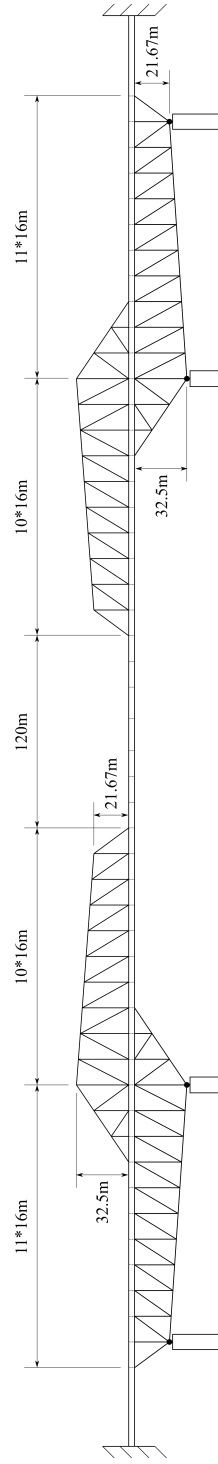
585 1, the Rayleigh damped system is observable for both the mass scenarios under
the sensor configuration 2 where the locations of sensors are distributed evenly
along the building.

This example demonstrates that for a given sensor setup the assumptions
on the form of damping used may have significant effects on the observability
590 of the system. Interestingly, the most general assumption of case i) would lead
to inability of identifying the system for the examined sensor setups. While
using a specific damping as in the cases ii) and iii) improves the observability of
the system, one should also realize that the results obtained are possible only
as a consequence of introducing those assumptions. Moreover, using a different
595 sensor configuration or introducing more sensors would also help improve ob-
servability. The observability tools proposed in this work allow users to achieve
such investigations without concerning about the limitations associated with
Algorithm I.

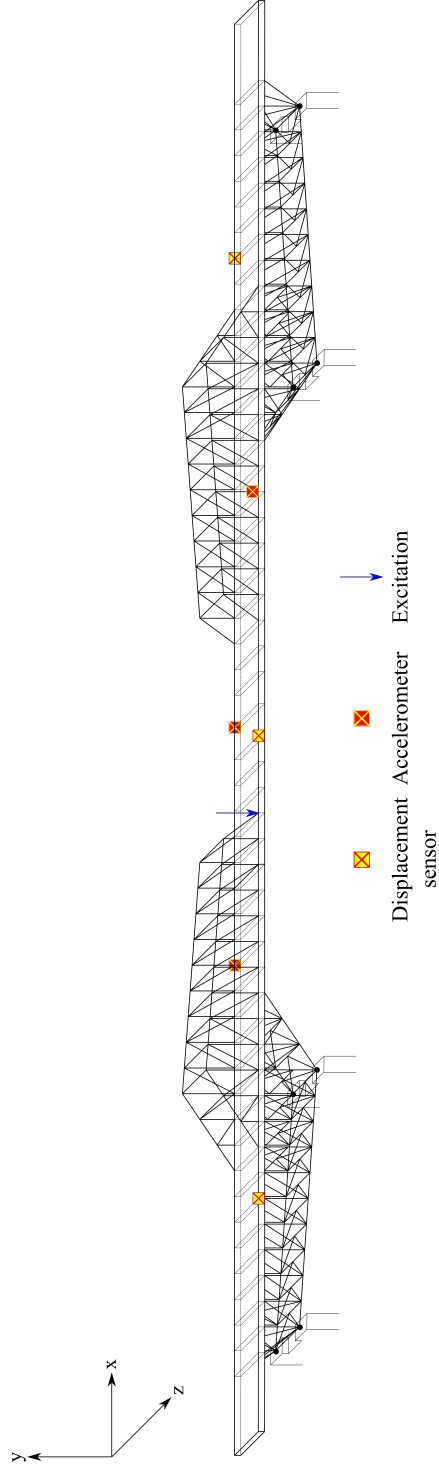
6.2. Example 2: 3D finite element model of a truss-beam bridge

600 Application of Algorithm II and Algorithm III to a 3D finite element (FE)
model of a truss-beam bridge whose geometry is inspired by the design of Tokyo
Gate Bridge is demonstrated in this example. The model under investigation
contains more than two thousand states, and its size does not allow for using
Algorithm I in practice due to the associated physical memory requirements.
605 Tokyo Gate Bridge is a recently built (2011) steel bridge located across the
Tokyo Bay in Tokyo. The main bridge is a three-span composite structure of
trusses and a steel girder. A schematic diagram to show the geometry of the
bridge model under investigation is in Figure 3(a), while the detailed informa-
tion about the construction of Tokyo Gate Bridge can be seen in [37]. This
610 example is chosen to illustrate that the proposed observability algorithms can
be implemented in a realistic scenario where a researcher would be investigat-
ing the viability of various sensor patterns for estimating the properties of the
bridge.

The bridge is modelled using linear frame-type macro-elements. The vector



(a)



(b)

Figure 3: (a) The geometry of the bridge model under study (b) The 3D FE model of the bridge with excitation-sensor setups

of dynamic states \mathbf{x}_t of the model includes the displacements and the corresponding velocities at the nodes of the discretization shown in Figure 3(b). The system matrix \mathbf{A} is of the form (46), where \mathbf{M} and \mathbf{K} are now the mass and stiffness matrices of the model that are obtained from assembling the elemental stiffness and mass matrices. The steel deck of the bridge is modelled using beam elements, while the other pin-jointed members are modelled as truss elements. For truss elements, each truss node has 3 DOFs, i.e. the translational motions along x , y and z axes. Each truss element between nodes i and j would contribute to the translational components of the mass matrix at the nodes. In the following the nodal mass from all truss elements lumped at the node i is defined as m_i . The axial stiffness of the truss element between nodes i and j is k_{ij} , and given its rotation relative to horizon, the stiffness matrix of the element can thus be derived in a global coordinate system. The deck is treated as a geometrically uniform Euler-Bernoulli beam. The 792m long beam is discretized into 49 segments including 47 segments of 16m and 2 segments of 20m as illustrated in Figure 3(a) and 3(b). Each beam node has 5 DOFs, as shown in Figure 4(a), assuming the axial deformations are neglected. The consistent mass matrix and the stiffness matrix of a beam segment are given as:

$$\begin{aligned}
\mathbf{M}_{beam} &= \frac{\rho A L}{420} \begin{bmatrix} 156 & 0 & 0 & 0 & 22L & 54 & 0 & 0 & 0 & -13L \\ 0 & 156 & 0 & -22L & 0 & 0 & 54 & 0 & 13L & 0 \\ 0 & 0 & \frac{140Ix}{A} & 0 & 0 & 0 & 0 & \frac{70Ix}{A} & 0 & 0 \\ 0 & -22L & 0 & 4L^2 & 0 & 0 & -13L & 0 & -3L^2 & 0 \\ 22L & 0 & 0 & 0 & 4L^2 & 13L & 0 & 0 & 0 & -3L^2 \\ 54 & 0 & 0 & 0 & 13L & 156 & 0 & 0 & 0 & -22L \\ 0 & 54 & 0 & -13L & 0 & 0 & 156 & 0 & 22L & 0 \\ 0 & 0 & \frac{70Ix}{A} & 0 & 0 & 0 & 0 & \frac{140Ix}{A} & 0 & 0 \\ 0 & 13L & 0 & -3L^2 & 0 & 0 & 22L & 0 & 4L^2 & 0 \\ -13L & 0 & 0 & 0 & -3L^2 & -22L & 0 & 0 & 0 & 4L^2 \end{bmatrix} \\
\mathbf{K}_{beam} &= \begin{bmatrix} \frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} & \frac{-12EI_z}{L^3} & 0 & 0 & 0 & \frac{6EI_z}{L^2} \\ 0 & \frac{12EI_y}{L^3} & 0 & \frac{-6EI_y}{L^2} & 0 & 0 & \frac{-12EI_y}{L^3} & 0 & \frac{-6EI_y}{L^2} & 0 \\ 0 & 0 & \frac{GJ}{L} & 0 & 0 & 0 & 0 & \frac{-GJ}{L} & 0 & 0 \\ 0 & \frac{-6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 \\ \frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L} & \frac{-6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} \\ \frac{-12EI_z}{L^3} & 0 & 0 & 0 & \frac{-6EI_z}{L^2} & \frac{12EI_z}{L^3} & 0 & 0 & 0 & \frac{-6EI_z}{L^2} \\ 0 & \frac{-12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 & 0 & \frac{12EI_y}{L^3} & 0 & \frac{6EI_y}{L^2} & 0 \\ 0 & 0 & \frac{-GJ}{L} & 0 & 0 & 0 & 0 & \frac{GJ}{L} & 0 & 0 \\ 0 & \frac{-6EI_y}{L^2} & 0 & \frac{2EI_y}{L} & 0 & 0 & \frac{6EI_y}{L^2} & 0 & \frac{4EI_y}{L} & 0 \\ \frac{6EI_z}{L^2} & 0 & 0 & 0 & \frac{2EI_z}{L} & \frac{-6EI_z}{L^2} & 0 & 0 & 0 & \frac{4EI_z}{L} \end{bmatrix}
\end{aligned} \tag{54}$$

where A is the cross section area and L is the length of the beam segment.

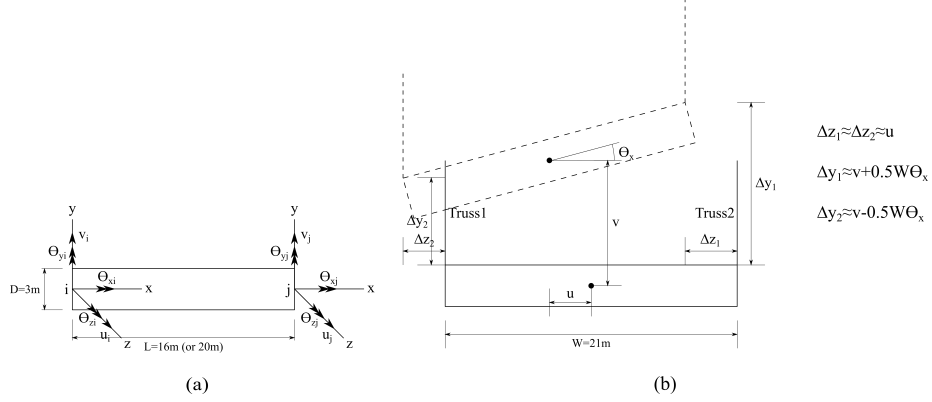


Figure 4: (a) The beam segment with 2 nodes having 10 DOFs (b) The translational and rotational motions of a rigid beam cross-section with truss elements connected at two sides

I_y and I_z are the second moments of area of the cross section about y and z axes respectively, $I_x = I_y + I_z$ is the polar moment of inertia and J is the torsion constant. E and G are the Young's modulus and the shear modulus respectively of the beam segment, and ρ is the density. Assume the Young's modulus and the shear modulus of the beam segment between nodes i and j are E_{ij} and G_{ij} respectively, and all the segments share the common density ρ . As the beam cross section remains plane after deformation, at the nodes of contact between the truss elements and the deck, a rigid element transformation relates the truss nodal DOFs to the beam nodal DOFs as shown in Figure 4(b). The rigidity transformations are applied on the elemental mass and stiffness matrices to express the corresponding truss DOFs at those nodes in terms of the beam nodal DOFs. The assembly of the elemental matrices can be found in [38]. The truss elements are connected with moment-free joints to the deck and the piers; the ends of the deck are assumed fixed. The following properties are considered unknown: 508 axial stiffness of the truss elements k_{ij} , 148 lumped masses of the truss nodes m_i , 49 Young's modulus E_{ij} and 49 shear modulus G_{ij} of the beam segments and the density of the deck ρ ; the model totally contains 755

unknown parameters to be identified and 1340 dynamic states corresponding to 670 DOFs.

Figure 3(b) gives an overview of the 3D FE model of the bridge with the proposed excitation-sensor setups. The bridge is excited by a load applied vertically at one side of the deck. In practice, such a load may be provided by a hammer or an actuator. Three displacement sensors and three accelerometers are installed on the sides of the deck to record its vertical and transversal displacement and acceleration response. The observability results obtained from the implementations using Algorithm II and Algorithm III are in absolute agreement: there are 36 unidentifiable parameters and 240 unobservable dynamic states. The detailed results are schematically presented in Figure 5 with the unobservable dynamic states and the unidentifiable parameters highlighted. Interestingly, the observability of the states related to the left half of the bridge is the same as that of the right half, while it should be noted that the geometry of the bridge is symmetric but the locations of the excitation and sensors are not. As can be seen, the axial stiffness of the lateral truss elements on the top of the bridge are unidentifiable, and their motions (displacement and velocity) toward z direction are unobservable. A further test shows that if extra sensors are installed on the top to measure the lateral vibrations of the truss nodes under e.g. wind load, those dynamic states and parameter become observable. Despite the observability of the truss elements, the parameters and dynamic states related to the beam segments are all observable, and therefore it can be concluded that, under the proposed excitation-measurement scheme, it is feasible to estimate the properties of the deck and simultaneously track its vibrations using system identification methods. In practice, more sensors are often needed to install in order to minimize the influence of measurement noises on the identification quality, that is, to enhance the so called practical observability. However, the lack of observability of the truss elements might affect the ability to properly identify some of the overall structural modes without introducing further assumptions.

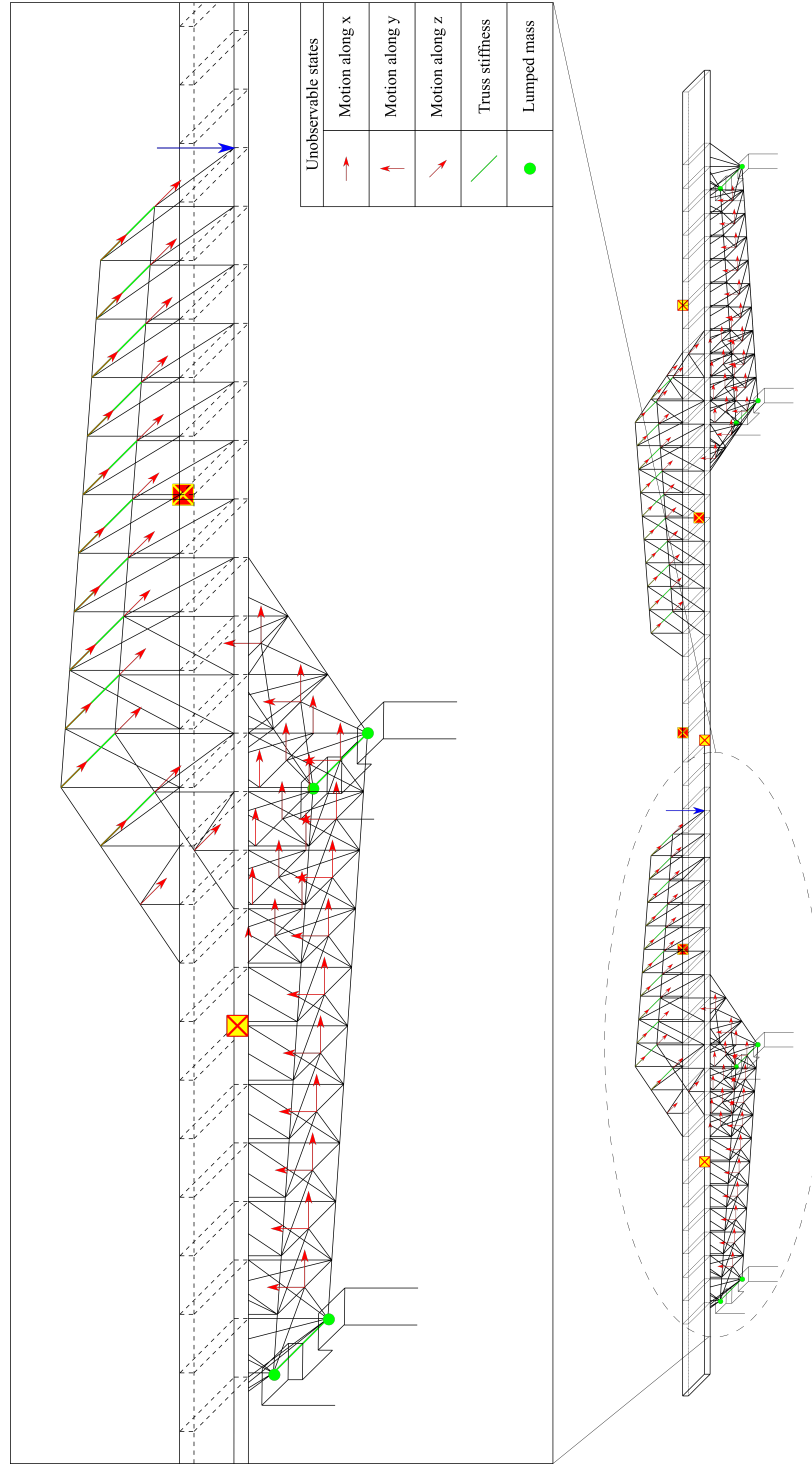


Figure 5: The observability of the FE model with the unobservable dynamic states and parameters highlighted by red arrows and green elements respectively

680 7. Conclusions

In this paper, a robust algorithm is developed to implement the ORC for the observability of the dynamic states and the identifiability of the parameters of linear dynamical systems (Algorithm II and Algorithm III can be seen as two versions of operations of the same algorithm). To derive the framework of the algorithm, an explicit expression of the observability matrix of linear systems with unknown parameters is obtained, and a product rule for the gradient of matrix multiplication is introduced to expand the elements involved in the observability matrix. This allows for obtaining a recursive evaluation of those elements removing the need to use symbolic differentiations after the initialization of the algorithm. Additionally, it opens up the route of efficient numerical implementations. The introduction of modular operations allows for obtaining a floating-point implementation that can handle robustly very large linear systems. Applications and superior performance of the algorithm are shown by successfully testing several examples of large engineering systems, including a 100-floor high-rise shear building and a 3D FE model of a large truss-beam bridge. To the best of authors' knowledge, none of the existing observability methods have the ability to deal with the problems of such sizes and complexities. The observability results provided by the algorithm are practically useful to give guidance for engineers on sensor placement and model adjustment in system identification campaigns.

To seek efficient algorithmic solutions for more general nonlinear systems, a forthcoming work will be devoted to the development of a fast algorithm to test the observability of rational nonlinear systems based on the computational scheme found in [23]. This algorithm will be further extended to account for unmeasured inputs applied on a dynamical system [14, 15].

Acknowledgement

The first author would like to gratefully acknowledge the financial support from the China Scholarship Council (CSC).

References

- 710 [1] R. Kalman, On the General Theory of Control Systems, Proceedings of
the 1st International IFAC Congress on Automatic and Remote Control,
Moscow, USSR 1 (1960) 491–502.
- [2] R. Hermann, A. Krener, Nonlinear Controllability and Observability, IEEE
Transactions on Automatic Control AC-22 (5) (1977) 728–740.
- 715 [3] S. Diop, M. Fliess, Nonlinear observability, identifiability, and persistent
trajectories, Proceedings of the 30th Conference on Decision and Control,
Brighton, UK (1991) 714–719.
- [4] H. Pohjanpalo, System Identifiability Based on the Power Series Expansion
of the Solution, Mathematical Biosciences 41 (1978) 21–33.
- 720 [5] S. Vajda, K. Godfrey, H. Rabitz, Similarity transformation approach to
identifiability analysis of nonlinear compartmental models, Mathematical
Biosciences 93 (1989) 217–248.
- [6] L. Ljung, T. Glad, On global identifiability for arbitrary model
parametrizations, Automatica 30 (1994) 265–276.
- 725 [7] F. Boulier, D. Lazard, F. Ollivier, M. Petitot, Computing representations
for radicals of finitely generated differential ideals, Applicable Algebra in
Engineering, Communication and Computing 20 (2009) 73–121.
- [8] F. Udwadia, D. Sharma, Some uniqueness results related to building struc-
tural identification, SIAM Journal on Applied Mathematics 34 (1) (1978)
730 104–118.
- [9] F. Udwadia, D. Sharma, P. Shah, Uniqueness of damping and stiffness
distributions in the identification of soil and structural systems, Journal of
Applied Mechanics 45 (1) (1978) 181–187.

- 735 [10] G. Franco, R. Betti, R. Longman, On the uniqueness of solutions for the identification of linear structural systems, *Journal of Applied Mechanics* 73 (1) (2005) 153–162.
- [11] M. Chatzis, E. Chatzi, A. Smyth, On the observability and identifiability of nonlinear structural and mechanical systems, *Structural Control and Health Monitoring* 22 (2015) 574–593.
- 740 [12] M. Chatzis, E. Chatzi, S. Triantafyllou, A discontinuous extended Kalman filter for non-smooth dynamic problems, *Mechanical Systems and Signal Processing* 92 (2017) 13–29.
- [13] M. Chatzis, E. Chatzi, A discontinuous unscented Kalman filter for non-smooth dynamic problems, *Frontiers in Built Environment* 3 (56) (2017) 1–15.
- 745 [14] A. Martinelli, Nonlinear unknown input observability: extension of the observability rank condition, *IEEE Transactions on Automatic Control* 64 (1) (2019) 222–237.
- [15] K. Maes, M. Chatzis, G. Lombaert, Observability of nonlinear systems with unmeasured inputs, *Mechanical Systems and Signal Processing* 130 (2019) 378–394.
- 750 [16] A. Raue, C. Kreutz, T. Maiwald, J. Bachmann, M. Schilling, U. Klingmuller, J. Timmer, Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood, *Bioinformatics* 25 (15) (2009) 1923–1929.
- 755 [17] A. Villaverde, A. Barreiro, A. Papachristodoulou, Structural identifiability of dynamic systems biology models, *PLOS Computational Biology* 12 (10) (2016) 1–22.
- [18] L. Katafygiotis, J. Beck, Updating models and their uncertainties. II: model identifiability, *Journal of Engineering Mechanics* 124 (4) (1998) 463–467.
- 760

- [19] C. Papadimitriou, Optimal sensor placement methodology for parametric identification of structural systems, *Journal of Sound and Vibration* 278 (4) (2004) 923–947.
- [20] A. Isidori, *Nonlinear Control Systems*, 3rd Edition, Springer-Verlag London, 1995.
- [21] B. De Jager, The use of symbolic computation in nonlinear control: is it viable?, *IEEE Transactions on Automatic Control* 40 (1) (1995) 84–89.
- [22] A. Kugi, K. Schlacher, R. Novak, Symbolic computation for the analysis and synthesis of nonlinear control systems, *WIT Press Transactions on Engineering Sciences* 22 (1999) 255–264.
- [23] A. Sedoglavic, A Probabilistic Algorithm to Test Local Algebraic Observability in Polynomial Time, *Journal of Symbolic Computation* 33 (2002) 735–755.
- [24] G. Bellu, M. Saccomani, S. Audoly, L. D’Angio, DAISY: A new software tool to test global identifiability of biological and physiological systems, *Computer Methods and Programs in Biomedicine* 88 (2007) 52–61.
- [25] M. Anguelova, Observability and identifiability of nonlinear systems with applications in biology, Ph.D. thesis, Department of Mathematical Sciences, Division of Mathematics, Chalmers University of Technology and Goteborg University (2007).
- [26] E. Chaves, *Notes on continuum mechanics*, 1st Edition, Springer Netherlands, 2013.
- [27] MATLAB, version R2016a, The MathWorks Inc., Natick, Massachusetts, 2016.
- [28] Wolfram, *Mathematica*, Wolfram Research Inc., Champaign, Illinois, 2019.
- [29] Maple, *Maplesoft*, a division of Waterloo Maple Inc., Maplesoft, Waterloo, Ontario, 2019.

- 790 [30] J. Muller, N. Brunie, F. de Dinechin, C. Jeannerod, M. Joldes, V. Lefevre, G. Melquiond, N. Revol, S. Torres, Handbook of Floating-Point Arithmetic, Springer International Publishing, 2018.
- [31] V. Shoup, A Computational Introduction to Number Theory and Algebra, 2nd Edition, Cambridge University Press, 2008.
- 795 [32] C. Bouillaguet, C. Delaplace, Sparse Gaussian Elimination Modulo p : an Update, International Workshop on Computer Algebra in Scientific Computing (2016) 101–116.
- [33] I. Shparlinski, Finite fields: theory and computation: the meeting point of number theory, computer science, coding theory and cryptography, 1st Edition, Springer Netherlands, 1999.
- 800 [34] J. Karlsson, M. Anguelova, M. Jirstrand, An Efficient Method for Structural Identifiability Analysis of Large Dynamic Systems, IFAC Proceedings Volumes 45 (16) (2012) 941–946.
- [35] P. Van Overschee, B. De Moor, N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems, Automatica 30 (1) (1994) 75–93.
- 805 [36] M. Chatzis, E. Chatzi, A. Smyth, An experimental validation of time domain system identification methods with fusion of heterogeneous data, Earthquake Engineering and Structural Dynamics 44 (2015) 523–547.
- 810 [37] T. Yoneyama, Y. Fujii, Fabrication and erection of Tokyo Gate Bridge, IABSE-JSCE Joint Conference on Advances in Bridge Engineering-III (2015) 268–277.
- [38] A. Chopra, Dynamics of Structures: Theory and Applications to Earthquake Engineering, 4th Edition, Pearson Education, 2012.