Robust computation of the observability of large linear systems with unknown parameters

X. Shi, M.N. Chatzis & M.S. Williams
University of Oxford, Oxford, UK

ABSTRACT: The concept of observability plays an important role in structural health monitoring as it can be utilized to theoretically predict whether the system identification of a structure of interest may be successful. The Observability Rank Condition (ORC) is an efficient tool used for analyzing the observability of dynamical systems. However, there are practical constraints, in terms of processing speed and computer memory, in implementing the ORC for large systems, or systems with complicated equations. In this work, a robust approach to compute the observability matrix of linear systems with unknown parameters is proposed. The newly developed algorithm is capable of handling a linear system of substantially larger order than the original algorithm.

1 INTRODUCTION

Identifying the structural parameters of a system, such as mass, damping and stiffness, is an important objective for reducing the pertaining uncertainty. A means of achieving so is suggested through using system identification algorithms on dynamic measurements obtained from the structure [1,2]. A related concept that of observability of the dynamical system can be used to reveal whether the parameters and dynamic states of a system under a given measurement setup can be identified [3]. Lack of observability guarantees that no identification algorithm would be successful in estimating the related state/parameter accurately. Thus one of the most important roles of observability is in determining proper sensor placement strategies that will lead to 'observable' output measurements. Avoiding 'unobservable' sensor setups before conducting measurement campaigns can result in savings of both time and money. Several works [4-7] have made contributions to the analyses of the observability of linear systems, while others [8] proposed suitable methods directly on the selection of the optimal sensor configurations. The observability properties of non-smooth nonlinear systems were discussed in [3]. This work illustrated that such systems, often related to damage, involve parameters whose identifiability property changes over time. This further led to the Discontinuous modification of Online Kalman Filters presented in [9,10].

Hermann and Krener introduced an observability method applicable to differentiable nonlinear systems, called the Observability Rank Condition (ORC) [11]. In implementation of the ORC, an observability matrix consisting of symbolic Lie derivatives of the system and their gradients must be constructed. However, the related computational cost leads to practical limitations when implementing the ORC to systems with many degrees of freedom (e.g. a long-span bridge) or systems with complicated equations. The amount and complexity of symbolic operations involved for such systems result in the physical memory requirements growing rapidly and often exceeding the computer's capacity [12,13]. Such are also encountered in other fields of engineering and biology when monitoring multiple DOF systems, such as aircrafts or complicated biological systems based on biochemical reaction networks [14]. Several researchers have made efforts to address the issue by either implementing symbolic computations more efficiently or exploring numerical and semi-numerical solutions [15-17].

The objective of this work is to explore a computationally efficient way for calculating the observability matrix. The method utilizes the idea of expressing the gradients of Lie derivatives as products between easier to calculate tensor gradients. This paper further focuses on the nonlinear augmented systems occurring from linear systems whose states and parameters are to be estimated. The form of the resulting observability matrix is provided and an efficient recursion algorithm to compute the observability matrix numerically is developed. The new algorithm implemented on a standard desktop computer is capable of handling a linear civil engineering system with substantially more degrees of freedom than the original algorithm.

2 TENSOR AND MATRIX

In this section the notation and convention used for tensor are introduced. An n^{th} order tensor $\mathbf{A}$ can be written in the form of the Einstein Summation notation as, $\mathbf{A} = A_{i_1 i_2 \dots i_n} \boldsymbol{e}_{\boldsymbol{i_1}} \otimes \boldsymbol{e}_{\boldsymbol{i_2}} \otimes \dots \otimes \boldsymbol{e}_{\boldsymbol{i_n}}$, where $A_{i_1 i_2 \dots i_n}$ refers to the coefficient of the tensor, $\boldsymbol{e}_{\boldsymbol{i_j}}$ is the j^{th} unit vector in a multi-dimensional space, and

$\otimes$ is the tensor product symbol. The coefficients of a low order tensor ($n \leq 2$) can easily be represented in a matrix form, such as a scalar (0^{th} order), a row or a column vector (1^{st} order) and a matrix (2^{nd} order).

For the remaining of the paper it is important to distinguish between expressing a tensor in its tensor form, or expressing its coefficients in a matrix form. Let a bold letter with a box bracket denote a $(1 \times \alpha)$ row vector, e.g. $[\mathbf{b}]$, and a bold letter with a curly bracket denote a $(\alpha \times 1)$ column vector, e.g. $\{\mathbf{b}\}$, both of which are the matrix representations of the coefficients of a first order tensor (a vector), $\mathbf{b} = b_j \boldsymbol{e_j}, j \in [1, \alpha]$. For any tensor $\mathbf{A} = A_{i_1 i_2 \dots i_n} \boldsymbol{e_{i_1}} \otimes \boldsymbol{e_{i_2}} \otimes \dots \otimes \boldsymbol{e_{i_n}}$, $i_1 \in [1, \beta]$, $i_2 \in [1, \gamma]$, …, $i_n \in [1, \delta]$, $[\mathbf{A}]$ corresponds to the matrix representation of its coefficients and is a multi-dimensional matrix of size $(\beta \times \gamma \times \dots \times \delta)$.

The correspondence between matrix multiplication and tensor dot product is also reminded. For example, a right multiplication of matrix $[\mathbf{M}]$ by column vector $\{\mathbf{b}\}$ is equivalent to the dot product of the corresponding second order tensor and vector: $[\mathbf{M}]\{\mathbf{b}\} = \mathbf{M} \cdot \mathbf{b}$. It is noted that the tensor dot product in the Einstein Summation notation follows the rule of simple contraction, that is, $\mathbf{M} \cdot \mathbf{b} = \left(M_{i_1 i_2} \boldsymbol{e_{i_1}} \otimes \boldsymbol{e_{i_2}}\right) \cdot \left(b_j \boldsymbol{e_j}\right) = M_{i_1 i_2} b_{i_2} \boldsymbol{e_{i_1}}$, where the last dimension of the first tensor and the first dimension of the second tensor are contracted.

3 THE OBSERVABILITY RANK CONDITION (ORC)

The state and measurement equations of a nonlinear SISO dynamical system can be written in the matrix form:

$$\{\dot{\boldsymbol{x}}\} = \{\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})\}, \quad y = h(x) \tag{1}$$

where $[\boldsymbol{x}]$, of size $(1 \times n)$, is the state vector, u is a known single-input and y is the single-output respectively. $\{\boldsymbol{f}\}$ and h are nonlinear functions. If the time-invariant parameters of the system are to be estimated in an identification problem, they are treated as additional states such that $[\boldsymbol{x}]$ becomes a state of the augmented form:

$$[\boldsymbol{x}] = \left[[\boldsymbol{x_t}], [\boldsymbol{\theta}]\right], \quad \left\{\dot{\boldsymbol{\theta}}\right\} = \{\mathbf{0}\} \tag{2}$$

where $[\boldsymbol{x_t}]$, of size $(1 \times n_t)$, is a row vector of time-variant states, and $[\boldsymbol{\theta}]$, of size $(1 \times n_\theta)$, is a row vector of time-invariant parameters. In general, a system is considered observable when the evolution of all time-variant states can be tracked and the values of all parameters can be estimated given the output measurements. If $\{\boldsymbol{f}\}$ and h are analytic, i.e. differentiable functions, then the ORC can be used to perform the observability analysis. In implementing the ORC, an observability matrix $O(x)$ consisting of the gradients of the measurement and its Lie derivatives is constructed. For an autonomous system, the form of $O(x)$ is:

$$O(x) = \begin{bmatrix} [\nabla h(x)] \\ [\nabla L_f h(x)] \\ [\nabla L_f^2 h(x)] \\ \vdots \\ [\nabla L_f^k h(x)] \\ \vdots \\ [\nabla L_f^{n-1} h(x)] \end{bmatrix} \tag{3}$$

where $L_f h(x) = [\nabla h(x)]\{\boldsymbol{f}(\boldsymbol{x})\}$ is the Lie derivative of the scalar function $h(x)$ along the vector field $\{\boldsymbol{f}(\boldsymbol{x})\}$, and ∇ denotes the gradient with respect to the state $[\boldsymbol{x}]$. Higher order Lie derivatives are defined as:

$$L_f^k h(x) = \left[\nabla L_f^{k-1} h(x)\right]\{\boldsymbol{f}(\boldsymbol{x})\},$$

$$\text{with } L_f^0 h(x) = h(x) \tag{4}$$

System (1) is observable, if $O(x)$ is a full rank matrix, i.e. $rank\big(O(x)\big) = n$. If the condition is not satisfied, one can detect whether the i^{th} state/parameter is unobservable/unidentifiable state/parameter by deleting the corresponding column of $O(x)$. If the rank of the occurring matrix remains the same then that state/parameter is unobservable/unidentifiable [3].

4 THE PRODUCT RULE FOR TENSOR DOT PRODUCTS

In calculus, it is well known that the derivative of product of scalar functions can be expanded following the product rule. This section illustrates the extension of that principle to the gradient of the dot product between tensors.

4.1 *The gradient of a tensor*

The gradient of an n^{th} order tensor $\mathbf{A} = A_{i_1 i_2 \dots i_n} \boldsymbol{e_{i_1}} \otimes \boldsymbol{e_{i_2}} \otimes \dots \otimes \boldsymbol{e_{i_n}}$ with respect to a vector $\boldsymbol{z} = z_m \boldsymbol{e_m}$ is defined as:

$$\begin{aligned} \nabla_{\boldsymbol{z}} \mathbf{A} &= \frac{\partial \mathbf{A}}{\partial z_m} \otimes \boldsymbol{e_m} \\ &= \frac{\partial A_{i_1 i_2 \dots i_n}}{\partial z_m} \boldsymbol{e_{i_1}} \otimes \boldsymbol{e_{i_2}} \otimes \dots \otimes \boldsymbol{e_{i_n}} \otimes \boldsymbol{e_m} \end{aligned} \tag{5}$$

where the Einstein Summation notation is used. For notational convenience in this work, let the Nabla

symbol ∇ denote the gradient with respect to the state $\boldsymbol{x}$, while the Nabla symbol with a subscript $\boldsymbol{z}$, $\nabla_{\boldsymbol{z}}$, denotes the gradient with respect to a specific vector $\boldsymbol{z}$.

4.2 *Tensor transposition*

For a tensor $\mathbf{A}$, ^{T_t}, denotes a mode of transpositions used for swapping the last two dimensions of the tensor:

$$\mathbf{A}^{T_t} = A_{i_1 i_2 \dots i_n} \boldsymbol{e_{i_1}} \otimes \boldsymbol{e_{i_2}} \otimes \dots \otimes \boldsymbol{e_{i_n}} \otimes \boldsymbol{e_{i_{n-1}}} \quad (6)$$

For a second order tensor the previous mode of transpostions simply corresponds to the standard transpose ^{T}.

4.3 *The product rule for tensor dot products*

Consider an n^{th} order tensor $\mathbf{A} = A_{i_1 i_2 \dots i_n} \boldsymbol{e_{i_1}} \otimes \boldsymbol{e_{i_2}} \otimes \dots \otimes \boldsymbol{e_{i_n}}$, and a vector $\mathbf{b} = b_j \boldsymbol{e_j}$, satisfying $i_n \in [1, N]$, $j \in [1, N]$, then the gradient of dot product between $\mathbf{A}$ and $\mathbf{b}$ with respect to a vector $\boldsymbol{z} = z_m \boldsymbol{e_m}$ can be expanded as:

$$\nabla_{\boldsymbol{z}}(\mathbf{A} \cdot \mathbf{b}) = (\nabla_{\boldsymbol{z}} \mathbf{A})^{T_t} \cdot \mathbf{b} + \mathbf{A} \cdot \nabla_{\boldsymbol{z}} \mathbf{b} \quad (7)$$

Proof: The left hand side of (7) is according to the definition of the gradient of a tensor:

$$\begin{aligned} \nabla_{\boldsymbol{z}}(\mathbf{A} \cdot \mathbf{b}) &= \frac{\partial \left[\left(A_{i_1 i_2 \dots i_n} \boldsymbol{e_{i_1}} \otimes \boldsymbol{e_{i_2}} \otimes \dots \otimes \boldsymbol{e_{i_n}} \right) \cdot \left(b_j \boldsymbol{e_j} \right) \right]}{\partial z_m} \otimes \boldsymbol{e_m} \\ &= \frac{\partial \left(A_{i_1 i_2 \dots i_n} b_{i_n} \right)}{\partial z_m} \boldsymbol{e_{i_1}} \otimes \boldsymbol{e_{i_2}} \otimes \dots \otimes \boldsymbol{e_{i_{n-1}}} \otimes \boldsymbol{e_m} \end{aligned} \quad (8)$$

The product rule for scalar functions can be applied to expand the partial derivatives in the above expression:

$$\begin{aligned} \nabla_{\boldsymbol{z}}(\mathbf{A} \cdot \mathbf{b}) &= \left(\frac{\partial A_{i_1 i_2 \dots i_n}}{\partial z_m} b_{i_n} + A_{i_1 i_2 \dots i_n} \frac{\partial b_{i_n}}{\partial z_m} \right) \\ &\quad \boldsymbol{e_{i_1}} \otimes \boldsymbol{e_{i_2}} \otimes \dots \otimes \boldsymbol{e_{i_{n-1}}} \otimes \boldsymbol{e_m} \\ &= \frac{\partial A_{i_1 i_2 \dots i_n}}{\partial z_m} b_{in} \boldsymbol{e_{i_1}} \otimes \boldsymbol{e_{i_2}} \otimes \dots \otimes \boldsymbol{e_{i_{n-1}}} \otimes \boldsymbol{e_m} + \\ &\quad A_{i_1 i_2 \dots i_n} \frac{\partial b_{in}}{\partial z_m} \boldsymbol{e_{i_1}} \otimes \boldsymbol{e_{i_2}} \otimes \dots \otimes \boldsymbol{e_{i_{n-1}}} \otimes \boldsymbol{e_m} \end{aligned} \quad (9)$$

Next the right hand side of (7) is calculated:

$$\begin{aligned} &(\nabla_{\boldsymbol{z}} \mathbf{A})^{T_t} \cdot \mathbf{b} + \mathbf{A} \cdot \nabla_{\boldsymbol{z}} \mathbf{b} = \\ &\left(\frac{\partial A_{i_1 i_2 \dots i_n}}{\partial z_m} \boldsymbol{e_{i_1}} \otimes \boldsymbol{e_{i_2}} \otimes \dots \otimes \boldsymbol{e_m} \otimes \boldsymbol{e_{i_n}} \right) \cdot \left(b_j \boldsymbol{e_j} \right) + \\ &\left(A_{i_1 i_2 \dots i_n} \boldsymbol{e_{i_1}} \otimes \boldsymbol{e_{i_2}} \otimes \dots \otimes \boldsymbol{e_{in}} \right) \cdot \left(\frac{\partial b_j}{\partial z_m} \boldsymbol{e_j} \otimes \boldsymbol{e_m} \right) \\ &= \frac{\partial A_{i_1 i_2 \dots i_n}}{\partial z_m} b_{in} \boldsymbol{e_{i_1}} \otimes \boldsymbol{e_{i_2}} \otimes \dots \otimes \boldsymbol{e_{i_{n-1}}} \otimes \boldsymbol{e_m} + \\ &A_{i_1 i_2 \dots i_n} \frac{\partial b_{in}}{\partial z_m} \boldsymbol{e_{i_1}} \otimes \boldsymbol{e_{i_2}} \otimes \dots \otimes \boldsymbol{e_{i_{n-1}}} \otimes \boldsymbol{e_m} \end{aligned} \quad (10)$$

From (9) and (10) being equal the proof for (7) is complete.

5 OBSERVABILITY MATRIX

Now, consider the case of a linear underlying system with parameters to be identified. The state space equations (1) for such a system can be re-written as:

$$\{\dot{\boldsymbol{x}}_{\boldsymbol{t}}\} = [\boldsymbol{A}(\boldsymbol{\theta})]\{\boldsymbol{x}_{\boldsymbol{t}}\} + \{\boldsymbol{B}(\boldsymbol{\theta})\}u, \quad y = [\boldsymbol{C}(\boldsymbol{\theta})]\{\boldsymbol{x}_{\boldsymbol{t}}\}$$

$$[\boldsymbol{x}] = \left[[\boldsymbol{x}_{\boldsymbol{t}}], [\boldsymbol{\theta}] \right], \quad \{\dot{\boldsymbol{\theta}}\} = \{\mathbf{0}\},$$

$$\{\boldsymbol{f}(\boldsymbol{x})\} = \begin{Bmatrix} [\boldsymbol{A}(\boldsymbol{\theta})]\{\boldsymbol{x}_{\boldsymbol{t}}\} \\ \{\mathbf{0}\} \end{Bmatrix} \quad (11)$$

where $[\boldsymbol{A}]$ is a $(n_t \times n_t)$ matrix, $\{\boldsymbol{B}\}$ is a $(n_t \times 1)$ column vector and $[\boldsymbol{C}]$ is a $(1 \times n_t)$ row vector, all of which are dependent on the unknown parameters $[\boldsymbol{\theta}]$. It should be noted that the augmented system in (11) is nonlinear.

Using equations (3) and (4), $[\nabla L_f^{k-1} h(x)]$ is the k^{th} row of the observability matrix $O(x)$. This is a gradient with respect to the state $\boldsymbol{x}$. The occurring row vector can be separated into two sub-gradients, that are, a gradient with respect to the time-variant state $\boldsymbol{x_t}$ and a gradient with respect to parameters $\boldsymbol{\theta}$, denoted by $\nabla_{\boldsymbol{x_t}}$ and $\nabla_{\boldsymbol{\theta}}$ respectively. Therefore, it is possible to write $[\nabla L_f^{k-1} h(x)]$ as:

$$[\nabla L_f^{k-1} h(x)] = \left[[\nabla_{\boldsymbol{x_t}} L_f^{k-1} h(x)], [\nabla_{\boldsymbol{\theta}} L_f^{k-1} h(x)] \right] \quad (12)$$

Thus for the examined system, $L_f^k h(x)$ can be expressed as:

$$\begin{aligned} &L_f^k h(x) \\ &= [[\nabla_{\boldsymbol{x_t}} L_f^{k-1} h(x)], [\nabla_{\boldsymbol{\theta}} L_f^{k-1} h(x)]] \begin{Bmatrix} [\boldsymbol{A}(\boldsymbol{\theta})]\{\boldsymbol{x_t}\} \\ \{\mathbf{0}\} \end{Bmatrix} \\ &= [\nabla_{\boldsymbol{x_t}} L_f^{k-1} h(x)] ([\boldsymbol{A}(\boldsymbol{\theta})]\{\boldsymbol{x_t}\}) \end{aligned} \quad (13)$$

Based on the correspondences between the matrices and tensors in section 2, (13) can be transformed to the following tensor form:

$$L_f^k h(x) = \nabla_{\boldsymbol{x_t}} L_f^{k-1} h(x) \cdot (\boldsymbol{A}(\boldsymbol{\theta}) \cdot \boldsymbol{x_t}) \quad (14)$$

It should be noted that $\nabla_{\boldsymbol{x_t}} L_f^{k-1} h(x)$ is a vector, $\boldsymbol{A}(\boldsymbol{\theta})$ is a second order tensor, $\boldsymbol{x_t}$ is a vector and the dot product of $\boldsymbol{A}(\boldsymbol{\theta})$ and $\boldsymbol{x_t}$ is a vector.

To explore the recursion relation of the sub-gradient, the gradient of both sides of (14) with respect to $\boldsymbol{x_t}$ is taken:

$$\nabla_{x_t} L_f^k h(x) = \nabla_{x_t}(\nabla_{x_t} L_f^{k-1} h(x) \cdot (\boldsymbol{A}(\boldsymbol{\theta}) \cdot \boldsymbol{x_t})) \quad (15)$$

Since both $\nabla_{x_t} L_f^{k-1} h(x)$ and $\boldsymbol{A}(\boldsymbol{\theta}) \cdot \boldsymbol{x_t}$ are vectors, the right hand side of (15) can be expanded according to (7):

$$\begin{aligned}\nabla_{x_t} L_f^k h(x) = &\left(\nabla_{x_t}\left(\nabla_{x_t} L_f^{k-1} h(x)\right)\right)^T \cdot (\boldsymbol{A}(\boldsymbol{\theta}) \cdot \boldsymbol{x_t}) \\ &+ \nabla_{x_t} L_f^{k-1} h(x) \cdot \nabla_{x_t}(\boldsymbol{A}(\boldsymbol{\theta}) \cdot \boldsymbol{x_t})\end{aligned} \quad (16)$$

Furthermore, since $\boldsymbol{x_t}$ is a vector, $\nabla_{x_t}(\boldsymbol{A}(\boldsymbol{\theta}) \cdot \boldsymbol{x_t})$ in (16) can also be expanded following (7):

$$\begin{aligned}\nabla_{x_t} L_f^k h(x) = &\left(\nabla_{x_t}\left(\nabla_{x_t} L_f^{k-1} h(x)\right)\right)^T \cdot (\boldsymbol{A}(\boldsymbol{\theta}) \cdot \boldsymbol{x_t}) \\ &+ \nabla_{x_t} L_f^{k-1} h(x) \cdot ((\nabla_{x_t} \boldsymbol{A}(\boldsymbol{\theta}))^{T_t} \cdot \boldsymbol{x_t} \\ &+ \boldsymbol{A}(\boldsymbol{\theta}) \cdot \nabla_{x_t} \boldsymbol{x_t})\end{aligned} \quad (17)$$

Because $\boldsymbol{A}(\boldsymbol{\theta})$ is not dependent on $\boldsymbol{x_t}$, $\nabla_{x_t} \boldsymbol{A}(\boldsymbol{\theta}) = \boldsymbol{0_3}$, where $\boldsymbol{0_n}$ is the n^{th} order zero tensor. $\nabla_{x_t} \boldsymbol{x_t}$ is a second order identity tensor, leading (17) to:

$$\begin{aligned}\nabla_{x_t} L_f^k h(x) = &\left(\nabla_{x_t}\left(\nabla_{x_t} L_f^{k-1} h(x)\right)\right)^T \cdot (\boldsymbol{A}(\boldsymbol{\theta}) \cdot \boldsymbol{x_t}) \\ &+ \nabla_{x_t} L_f^{k-1} h(x) \cdot \boldsymbol{A}(\boldsymbol{\theta})\end{aligned} \quad (18)$$

A recursion relation between $\nabla_{x_t} L_f^k h(x)$ and $\nabla_{x_t} L_f^{k-1} h(x)$ is obtained in (18), with the initial term:

$$\nabla_{x_t} L_f^0 h(x) = \nabla_{x_t} h(x) = \nabla_{x_t}(\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{x_t}) \quad (19)$$

Expanding (19) gives:

$$\begin{aligned}\nabla_{x_t} L_f^0 h(x) &= \nabla_{x_t}(\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{x_t}) \\ &= (\nabla_{x_t} \boldsymbol{C}(\boldsymbol{\theta}))^T \cdot \boldsymbol{x_t} + \boldsymbol{C}(\boldsymbol{\theta}) \cdot \nabla_{x_t} \boldsymbol{x}_t \\ &= \boldsymbol{C}(\boldsymbol{\theta})\end{aligned} \quad (20)$$

The second term of $\nabla_{x_t} L_f^k h(x)$, i.e. $\nabla_{x_t} L_f h(x)$, can be calculated by substituting $\nabla_{x_t} L_f^0 h(x)$ to (18):

$$\begin{aligned}\nabla_{x_t} L_f h(x) = &(\nabla_{x_t} \boldsymbol{C}(\boldsymbol{\theta}))^T \cdot (\boldsymbol{A}(\boldsymbol{\theta}) \cdot \boldsymbol{x_t}) + \boldsymbol{C}(\boldsymbol{\theta}) \\ &\cdot \boldsymbol{A}(\boldsymbol{\theta})\end{aligned} \quad (21)$$

Since $\boldsymbol{C}(\boldsymbol{\theta})$ is not dependent on $\boldsymbol{x_t}$, $\nabla_{x_t} \boldsymbol{C}(\boldsymbol{\theta}) = \boldsymbol{0_2}$, and thus:

$$\nabla_{x_t} L_f h(x) = \boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A}(\boldsymbol{\theta}) \quad (22)$$

Similarly, $\nabla_{x_t} L_f^2 h(x)$ is calculated by substituting $\nabla_{x_t} L_f h(x) = \boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A}(\boldsymbol{\theta})$, $\nabla_{x_t}(\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A}(\boldsymbol{\theta}))$ is $\boldsymbol{0_2}$ since both $\boldsymbol{C}(\boldsymbol{\theta})$ and $\boldsymbol{A}(\boldsymbol{\theta})$ are not dependent on $\boldsymbol{x_t}$, so it yields the result $\nabla_{x_t} L_f^2 h(x) = \boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A^2}(\boldsymbol{\theta})$. In fact, since the initial term $\nabla_{x_t} L_f^0 h(x)$ and $\boldsymbol{A}(\boldsymbol{\theta})$ are not dependent on $\boldsymbol{x_t}$, $\nabla_{x_t}\left(\nabla_{x_t} L_f^{k-1} h(x)\right) = \boldsymbol{0_2}$ for all $k \geq 1$. Therefore, (18) can be simplified to the recursive form:

$$\nabla_{x_t} L_f^k h(x) = \nabla_{x_t} L_f^{k-1} h(x) \cdot \boldsymbol{A}(\boldsymbol{\theta}) = \boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A^k}(\boldsymbol{\theta}) \quad (23)$$

The k^{th} order Lie derivative can be calculated by substituting (23) to (14):

$$\begin{aligned}L_f^k h(x) &= (\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A^{k-1}}(\boldsymbol{\theta})) \cdot (\boldsymbol{A}(\boldsymbol{\theta}) \cdot \boldsymbol{x_t}) \\ &= (\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A^k}(\boldsymbol{\theta})) \cdot \boldsymbol{x_t}\end{aligned} \quad (24)$$

$\nabla_{\boldsymbol{\theta}} L_f^k h(x)$ is then written as:

$$\nabla_{\boldsymbol{\theta}} L_f^k h(x) = \nabla_{\boldsymbol{\theta}}((\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A^k}(\boldsymbol{\theta})) \cdot \boldsymbol{x_t}) \quad (25)$$

With the expressions of $\nabla_{x_t} L_f^k h(x)$, $L_f^k h(x)$ and $\nabla_{\boldsymbol{\theta}} L_f^k h(x)$ having been derived in (23), (24) and (25) in the tensor form, the corresponding matrix forms are:

$$\begin{aligned}\left[\nabla_{x_t} L_f^k h(x)\right] &= [\boldsymbol{C}(\boldsymbol{\theta})]\left[\boldsymbol{A^k}(\boldsymbol{\theta})\right], \\ L_f^k h(x) &= [\boldsymbol{C}(\boldsymbol{\theta})]\left[\boldsymbol{A^k}(\boldsymbol{\theta})\right]\{\boldsymbol{x_t}\}, \\ \left[\nabla_{\boldsymbol{\theta}} L_f^k h(x)\right] &= \left[\nabla_{\boldsymbol{\theta}}\left([\boldsymbol{C}(\boldsymbol{\theta})]\left[\boldsymbol{A^k}(\boldsymbol{\theta})\right]\{\boldsymbol{x_t}\}\right)\right]\end{aligned} \quad (26)$$

A special form of the observability matrix $O_l(x)$ for linear systems with unknown parameters is obtained:

$$\begin{aligned}O_l(x) &= \begin{bmatrix} [\nabla([\boldsymbol{C}(\boldsymbol{\theta})]\{\boldsymbol{x_t}\})] \\ [\nabla([\boldsymbol{C}(\boldsymbol{\theta})][\boldsymbol{A}(\boldsymbol{\theta})]\{\boldsymbol{x_t}\})] \\ [\nabla([\boldsymbol{C}(\boldsymbol{\theta})][\boldsymbol{A^2}(\boldsymbol{\theta})]\{\boldsymbol{x_t}\})] \\ \vdots \\ \left[\nabla([\boldsymbol{C}(\boldsymbol{\theta})][\boldsymbol{A^k}(\boldsymbol{\theta})]\{\boldsymbol{x_t}\})\right] \\ \vdots \\ [\nabla([\boldsymbol{C}(\boldsymbol{\theta})][\boldsymbol{A^{n-1}}(\boldsymbol{\theta})]\{\boldsymbol{x_t}\})] \end{bmatrix} \\ &= \begin{bmatrix} [\boldsymbol{C}(\boldsymbol{\theta})], & [\nabla_{\boldsymbol{\theta}}([\boldsymbol{C}(\boldsymbol{\theta})]\{\boldsymbol{x_t}\})] \\ [\boldsymbol{C}(\boldsymbol{\theta})][\boldsymbol{A}(\boldsymbol{\theta})], & [\nabla_{\boldsymbol{\theta}}([\boldsymbol{C}(\boldsymbol{\theta})][\boldsymbol{A}(\boldsymbol{\theta})]\{\boldsymbol{x_t}\})] \\ [\boldsymbol{C}(\boldsymbol{\theta})][\boldsymbol{A^2}(\boldsymbol{\theta})], & [\nabla_{\boldsymbol{\theta}}([\boldsymbol{C}(\boldsymbol{\theta})][\boldsymbol{A^2}(\boldsymbol{\theta})]\{\boldsymbol{x_t}\})] \\ \vdots & \vdots \\ [\boldsymbol{C}(\boldsymbol{\theta})][\boldsymbol{A^k}(\boldsymbol{\theta})], & [\nabla_{\boldsymbol{\theta}}([\boldsymbol{C}(\boldsymbol{\theta})][\boldsymbol{A^k}(\boldsymbol{\theta})]\{\boldsymbol{x_t}\})] \\ \vdots & \vdots \\ [\boldsymbol{C}(\boldsymbol{\theta})][\boldsymbol{A^{n-1}}(\boldsymbol{\theta})], & [\nabla_{\boldsymbol{\theta}}([\boldsymbol{C}(\boldsymbol{\theta})][\boldsymbol{A^{n-1}}(\boldsymbol{\theta})]\{\boldsymbol{x_t}\})] \end{bmatrix}\end{aligned} \quad (27)$$

6 ROBUST COMPUTATION OF THE OBSERVABILITY MATRIX

In this section, a recursion algorithm is developed to efficiently compute the observability matrix $O_l(x)$ row by row. The algorithm aims to recursively calculate the $(k+1)^{th}$ row of the observability matrix, $[\nabla L_f^k h(x)]$, using the k^{th} row, $[\nabla L_f^{k-1} h(x)]$, based on the basic arithmetic operations such as summations and products of matrices, without requiring symbolic differentiations within step k, for $k > 0$. As a result, numerical evaluation of $O_l(x)$ becomes possible for a specific realization of the state vector $[\boldsymbol{x}] = [\boldsymbol{x_0}]$.

While the recursion for $\nabla_{\boldsymbol{x_t}} L_f^k h(x)$ has been derived in (23), the main challenge arises from calculating $\nabla_{\boldsymbol{\theta}} L_f^k h(x) = \nabla_{\boldsymbol{\theta}}((\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A^k}(\boldsymbol{\theta})) \cdot \boldsymbol{x_t})$ in (25) without using the gradient operation. To overcome this difficulty, $\nabla_{\boldsymbol{\theta}} L_f^k h(x)$ is firstly expanded by (7), and as both $(\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A^k}(\boldsymbol{\theta}))$ and $\boldsymbol{x_t}$ are vectors, their dot product is communitive, i.e. $\nabla_{\boldsymbol{\theta}}\big((\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A^k}(\boldsymbol{\theta})) \cdot \boldsymbol{x_t}\big) = \nabla_{\boldsymbol{\theta}}(\boldsymbol{x_t} \cdot (\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A^k}(\boldsymbol{\theta})))$. Therefore:

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} L_f^k h(x) &= \nabla_{\boldsymbol{\theta}}(\boldsymbol{x_t} \cdot (\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A^k}(\boldsymbol{\theta}))) \\ &= (\nabla_{\boldsymbol{\theta}} \boldsymbol{x_t})^T \cdot (\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A^k}(\boldsymbol{\theta})) + \boldsymbol{x_t} \\ &\cdot \nabla_{\boldsymbol{\theta}}(\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A^k}(\boldsymbol{\theta}))\end{aligned} \tag{28}$$

Since $\boldsymbol{x_t}$ is not dependent on $\boldsymbol{\theta}$, $\nabla_{\boldsymbol{\theta}} \boldsymbol{x_t} = \boldsymbol{0_2}$, and thus:

$$\nabla_{\boldsymbol{\theta}} L_f^k h(x) = \boldsymbol{x_t} \cdot \nabla_{\boldsymbol{\theta}}(\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A^k}(\boldsymbol{\theta})) \tag{29}$$

$\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A^k}(\boldsymbol{\theta})$ is then written as $\boldsymbol{C}(\boldsymbol{\theta}) \cdot \Big(\boldsymbol{A^{k-1}}(\boldsymbol{\theta}) \cdot \boldsymbol{A}(\boldsymbol{\theta})\Big) = \Big(\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A^{k-1}}(\boldsymbol{\theta})\Big) \cdot \boldsymbol{A}(\boldsymbol{\theta}) = \boldsymbol{A}^T(\boldsymbol{\theta}) \cdot (\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A^{k-1}}(\boldsymbol{\theta}))$. As $\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A^{k-1}}(\boldsymbol{\theta})$ is a vector, it is possible to further expand the expression of $\nabla_{\boldsymbol{\theta}}\Big(\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A^k}(\boldsymbol{\theta})\Big)$ using (7) as:

$$\begin{aligned}&\nabla_{\boldsymbol{\theta}}\Big(\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A^k}(\boldsymbol{\theta})\Big) = \\ &(\nabla_{\boldsymbol{\theta}} \boldsymbol{A}^T(\boldsymbol{\theta}))^{T_t} \cdot (\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A^{k-1}}(\boldsymbol{\theta})) + \boldsymbol{A}^T(\boldsymbol{\theta}) \cdot \\ &\nabla_{\boldsymbol{\theta}}\Big(\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A^{k-1}}(\boldsymbol{\theta})\Big)\end{aligned} \tag{30}$$

(30) is a recursion relation between $\nabla_{\boldsymbol{\theta}}(\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A^k}(\boldsymbol{\theta}))$ and $\nabla_{\boldsymbol{\theta}}(\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A^{k-1}}(\boldsymbol{\theta}))$ that does not involve differentiations at step $k > 0$. $\boldsymbol{A}^T(\boldsymbol{\theta})$ and $(\nabla_{\boldsymbol{\theta}} \boldsymbol{A}^T(\boldsymbol{\theta}))^{T_t}$ are known or calculated at the initial step, and $\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A^{k-1}}(\boldsymbol{\theta}) = \nabla_{\boldsymbol{x_t}} L_f^{k-1} h(x)$ is calculated during the previous step.

The expressions in (29) and (30) in the tensor form are transformed to those in the matrix form for ease of computation:

$$[\nabla_{\boldsymbol{\theta}} L_f^k h(x)] = [\boldsymbol{x_t}]\Big[\nabla_{\boldsymbol{\theta}}\Big(\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A^k}(\boldsymbol{\theta})\Big)\Big],$$

$$\begin{aligned}&\Big[\nabla_{\boldsymbol{\theta}}\Big(\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A^k}(\boldsymbol{\theta})\Big)\Big] = \\ &[(\nabla_{\boldsymbol{\theta}} \boldsymbol{A}^T(\boldsymbol{\theta}))^{T_t}]\{\nabla_{\boldsymbol{x_t}} L_f^{k-1} h(x)\} \\ &+ [\boldsymbol{A}^T(\boldsymbol{\theta})]\big[\nabla_{\boldsymbol{\theta}}(\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A^{k-1}}(\boldsymbol{\theta}))\big]\end{aligned} \tag{31}$$

A complete algorithm to compute the observability matrix $O_l(x)$ can be outlined as follows:

Let $[\boldsymbol{a_k}] = \Big[\nabla_{\boldsymbol{\theta}}\Big(\boldsymbol{C}(\boldsymbol{\theta}) \cdot \boldsymbol{A^k}(\boldsymbol{\theta})\Big)\Big]$, $[\boldsymbol{b_k}] = \big[\nabla_{\boldsymbol{x_t}} L_f^k h(x)\big]$, $\{\boldsymbol{b_k}\} = \big\{\nabla_{\boldsymbol{x_t}} L_f^k h(x)\big\}$, $[\boldsymbol{c_k}] = [\nabla_{\boldsymbol{\theta}} L_f^k h(x)]$, $[\boldsymbol{d}] = [(\nabla_{\boldsymbol{\theta}} \boldsymbol{A}^T(\boldsymbol{\theta}))^{T_t}]$ and $[\boldsymbol{e}] = [\boldsymbol{A}(\boldsymbol{\theta})]$

1. Initial point for $k = 0$: $[\boldsymbol{a_0}] = [\nabla_{\boldsymbol{\theta}} \boldsymbol{C}(\boldsymbol{\theta})]$, $[\boldsymbol{b_0}] = [\boldsymbol{C}(\boldsymbol{\theta})]$, $\{\boldsymbol{b_0}\} = \{\boldsymbol{C}(\boldsymbol{\theta})\}$, $[\boldsymbol{c_0}] = [\boldsymbol{x_t}][\boldsymbol{a_0}]$ and the first row of the matrix $O_l(x)$ is $\big[[\boldsymbol{b_0}], [\boldsymbol{c_0}]\big]$.
2. From (31), $[\boldsymbol{a_{k+1}}] = [\boldsymbol{d}]\{\boldsymbol{b_k}\} + [\boldsymbol{e}]^T[\boldsymbol{a_k}]$
3. From (23), $[\boldsymbol{b_{k+1}}] = [\boldsymbol{b_k}][\boldsymbol{e}]$
4. From (31), $[\boldsymbol{c_{k+1}}] = [\boldsymbol{x_t}][\boldsymbol{a_{k+1}}]$
5. The $(k+2)^{th}$ row of the matrix $O_l(x)$ is $\big[[\boldsymbol{b_{k+1}}], [\boldsymbol{c_{k+1}}]\big]$.
6. If $k = n - 2$, end; if not, $k = k + 1$ and go to step 2.

It is noted that the element $[(\nabla_{\boldsymbol{\theta}} \boldsymbol{A}^T(\boldsymbol{\theta}))^{T_t}]$ is a 3-dimensional array, so users should treat the term $[\boldsymbol{d}]\{\boldsymbol{b_k}\} = [(\nabla_{\boldsymbol{\theta}} \boldsymbol{A}^T(\boldsymbol{\theta}))^{T_t}]\big\{\nabla_{\boldsymbol{x_t}} L_f^k h(x)\big\}$ as a multi-dimensional array multiplication, or alternatively, one can calculate the term by unfolding the multi-dimensional array into a 2-dimensional matrix.

7 RESULTS

Consider the mass-spring model of a shear-type building system with n floors, i.e. n DOFs, as shown in Figure 1. An additional spring is connected between the ground and the second mass to illustrate a deviation of the system from the shear building approximation. For simplicity all masses are considered to be unit masses, i.e. $m_i = 1$. The displacement, velocity and acceleration of mass m_i are denoted as x_i, $\dot{x}_i$ and $\ddot{x}_i$ respectively. The parameters to be identified are the stiffness of springs k_i and the response of the system is to be tracked. Therefore, the row vector of parameters $[\boldsymbol{\theta}]$ in this case is:

$$[\boldsymbol{\theta}] = [k_1, k_2, \dots, k_{n+1}] \tag{32}$$

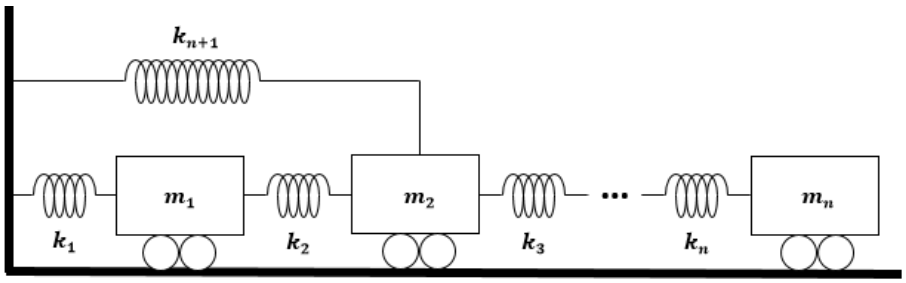


Figure 1. The mass-spring model with n degrees of freedom

and the row vector of dynamic states $[\boldsymbol{x_t}]$ is:

$$[\boldsymbol{x_t}] = [x_1, x_2, \dots, x_n, \dot{x}_1, \dot{x}_2, \dots, \dot{x}_n] \tag{33}$$

The measurement is chosen as the displacement of the first mass, i.e. $y = x_1$, for all observability tests in this section.

To implement the algorithm derived in section 6, users need to obtain or pre-calculate the invariant elements: $[\boldsymbol{e}] = [\boldsymbol{A}(\boldsymbol{\theta})]$ and $[\boldsymbol{d}] = [(\nabla_{\boldsymbol{\theta}}\boldsymbol{A}^T(\boldsymbol{\theta}))^{T_t}]$, and initial values of the recursive elements: $[\boldsymbol{a_0}] = [\nabla_{\boldsymbol{\theta}}\boldsymbol{C}(\boldsymbol{\theta})]$, $[\boldsymbol{b_0}] = [\boldsymbol{C}(\boldsymbol{\theta})]$ and $[\boldsymbol{c_0}] = [\boldsymbol{x_t}][\boldsymbol{a_0}]$, from the state space representation (11) of the system. The matrix $[\boldsymbol{A}(\boldsymbol{\theta})]$ of this system has the following form:

$$[\boldsymbol{A}(\boldsymbol{\theta})] = \begin{bmatrix} [\boldsymbol{0}]_{n\times n} & [\boldsymbol{I}]_{n\times n} \\ -[\boldsymbol{K}] & [\boldsymbol{0}]_{n\times n} \end{bmatrix} \tag{34}$$

where $[\boldsymbol{K}]$ is the stiffness matrix in the form:

$$[\boldsymbol{K}] = \begin{bmatrix} k_1 + k_2 & -k_2 & 0 & \cdots & 0 \\ -k_2 & k_2 + k_3 + k_{n+1} & -k_3 & \cdots & 0 \\ 0 & -k_3 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & k_{n-1} + k_n & -k_n \\ 0 & 0 & \cdots & -k_n & k_n \end{bmatrix} \tag{35}$$

Since the measurement is x_1, $[\boldsymbol{C}(\boldsymbol{\theta})]$ can be obtained as:

$$[\boldsymbol{C}(\boldsymbol{\theta})] = \left[1, [\boldsymbol{0}]_{1\times(2n-1)}\right] \tag{36}$$

and

$$[\nabla_{\boldsymbol{\theta}}\boldsymbol{C}(\boldsymbol{\theta})] = [\boldsymbol{0}]_{2n\times(n+1)} \tag{37}$$

While calculating $[(\nabla_{\boldsymbol{\theta}}\boldsymbol{A}^T(\boldsymbol{\theta}))^{T_t}]$, it is more convenient to firstly calculate the $[\nabla_{\boldsymbol{\theta}}\boldsymbol{A}(\boldsymbol{\theta})]$ without consideration of the transpositions, and then rearrange its sequence of dimensions following (6). The 3-dimensional array $[(\nabla_{\boldsymbol{\theta}}\boldsymbol{A}^T(\boldsymbol{\theta}))]$ is presented here in the MATLAB notation, where ':' corresponds to all the elements in that dimension :

$$[(\nabla_{\boldsymbol{\theta}}\boldsymbol{A}^T(\boldsymbol{\theta}))](:,:,1) = \begin{bmatrix} [\boldsymbol{0}]_{n\times n} & [\boldsymbol{0}]_{n\times n} \\ -\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}_{n\times n} & [\boldsymbol{0}]_{n\times n} \end{bmatrix}^T$$

$$[(\nabla_{\boldsymbol{\theta}}\boldsymbol{A}^T(\boldsymbol{\theta}))](:,:,2) = \begin{bmatrix} [\boldsymbol{0}]_{n\times n} & [\boldsymbol{0}]_{n\times n} \\ -\begin{bmatrix} 1 & -1 & \cdots & 0 \\ -1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}_{n\times n} & [\boldsymbol{0}]_{n\times n} \end{bmatrix}^T$$

$$\vdots$$

$$[(\nabla_{\boldsymbol{\theta}}\boldsymbol{A}^T(\boldsymbol{\theta}))](:,:,n+1) = \begin{bmatrix} [\boldsymbol{0}]_{n\times n} & [\boldsymbol{0}]_{n\times n} \\ -\begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}_{n\times n} & [\boldsymbol{0}]_{n\times n} \end{bmatrix}^T \tag{38}$$

$[(\nabla_{\boldsymbol{\theta}}\boldsymbol{A}^T(\boldsymbol{\theta}))^{T_t}]$ can be determined by swapping the last two dimensions of $[(\nabla_{\boldsymbol{\theta}}\boldsymbol{A}^T(\boldsymbol{\theta}))]$, i.e. $[(\nabla_{\boldsymbol{\theta}}\boldsymbol{A}^T(\boldsymbol{\theta}))^{T_t}](:,j,i) = \left[\left(\nabla_{\boldsymbol{\theta}}\boldsymbol{A}^T(\boldsymbol{\theta})\right)\right](:,i,j)$.

7.1 *A three degrees of freedom system*

The observability of a three degrees of freedom system, i.e. $n = 3$, is tested in MATLAB using the original ORC described in section 3 and the algorithm developed in section 6 (the modified ORC) respectively.

First, both the original ORC and the modified ORC are implemented symbolically to check whether they can produce the same symbolic expressions for observability matrix. For reasons of brevity, only the symbolic expression of the 4^{th} row of the ma-

trix, i.e., $\left[\nabla L_f^3 h(x)\right]$, is shown. Both methods produce the following line:

$$\left[\nabla L_f^3 h(x)\right] = [0,0,0,-k_1 - k_2, k_2, 0, -\dot{x}_1, \dot{x}_2 - \dot{x}_1, 0,0] \quad (39)$$

The methods agree entirely for any of the other rows as expected. Additionally, however the suggested algorithm can be implemented entirely numerically as stated earlier. There are two ways of numerical implementation in symbolic computations: in the first a realization of the states is used using symbolic numbers (e.g. $1/3$); in the second the states are evaluated with floating-point precision (e.g. $0.333\ldots$). The difference between symbolic numbers and floating-point numbers is that the arithmetics involving the first are exact, while floating-point numbers result in round-off error but can be processed much more efficiently. In this paper the modified ORC is implemented using only symbolic numbers.

It should be noted that while the symbolic expressions of the observability matrix from the original ORC, and the modified ORC, can be calculated first symbolically and then evaluated at a symbolic numerical realization of the states, only the second method can be used evaluating all the arrays at step 0 at a symbolic numerical realization of the states. The observability results obtained using the latter approach will be referred to as using symbolic numbers for the modified ORC and are compared versus the observability estimates obtained from the original ORC in Table 1.

Method	**The original ORC**	**The modified ORC**
Computation	Symbol	Symbolic number
Rank	9	9
Observable states	$x_1, \dot{x}_1$	$x_1, \dot{x}_1$
Unobservable states	$x_2, \dot{x}_2, x_3, \dot{x}_3$ k_1, k_2, k_3, k_4	$x_2, \dot{x}_2, x_3, \dot{x}_3$ k_1, k_2, k_3, k_4

Table 1. Observability of the three degrees of freedom system

As can be seen from the above results, the original ORC and the modified ORC using symbolic numbers can provide users with exactly the same observability information of a system. That is true if the symbolic realization of the states does not happen to be one of the roots of the determinant of the system, which is an unlikely scenario. The modified ORC can hence accurately predict the observability properties of the system, without the need to perform any operation using symbolic operations between the states but rather a symbolic number realization of them.

To show that the modified ORC can also be used for an observable system, the same 3 d.o.f. system without the $(n+1)^{th}$ spring is considered, i.e., a truly shear building. Thus the parameters to be identified are: k_1, k_2, k_3. The results of observability are shown in Table 2 where again the modified ORC using symbolic numbers agrees with the results of the original method:

Method	**The original ORC**	**The modified ORC**
Computation	Symbol	Symbolic number
Rank	9	9
Observable states	$x_1, \dot{x}_1, x_2, \dot{x}_2$ $x_3, \dot{x}_3, k_1, k_2$ k_3	$x_1, \dot{x}_1, x_2, \dot{x}_2$ $x_3, \dot{x}_3, k_1, k_2$ k_3

Table 2. Observability of the observable three degrees of freedom system

7.2 *Large systems*

The observability tests are performed on the system shown in Figure 1 with increased degrees of freedom: $n = 20, 40, 60, 80, 100, 120$, to compare capability and efficiency of the original ORC to the modified ORC using symbolic numbers. Both methods are implemented in MATLAB on a desktop computer with 16 GB of RAM and an i7 quad core cpu. The elapsed time of simulation is recorded for each test in seconds. If a test is not completed within two days, it is recorded as an unsuccessful test.

	The original ORC	**The modified ORC**
$\boldsymbol{n}$	Time (sec.)	Time (sec.)
20	Unsuccessful	12.04
40	-	152.43
60	-	735.53
80	-	2380.26
100	-	5962.65
120	-	12592.24

Table 3. Elapsed time (seconds) of simulation for large systems

As can be seen from Table 3, the original ORC is unable to give results of the observability for a system with 20 degrees of freedom within two days. In comparison, the modified ORC using symbolic numbers is capable of dealing with a large system with more than a hundred degrees of freedom within an acceptable period of about three and a half hours, by offering accurate observability information of the system to users.

The performance of the algorithm can further be improved by using floating point instead of symbolic numbers. In that case however robust numerical techniques, such as the Singular Value Decomposition (SVD), should be used to determine the numerical rank of a matrix. This will be examined as a future direction.

8 CONCLUSIONS

In this work, a special structure of the Observability Rank Condition (ORC) matrix for linear systems with unknown parameters is derived, and a robust algorithm is developed to efficiently compute the matrix. It is shown that the newly developed algorithm has superior performance on analyzing the observability of large linear systems. The concentration of the furture work will be placed on extending the method, that is, expanding Lie derivatives to the products of tensor gradients, for more general systems with nonlinear state space equations as well as using floating point arithmetics.

9 REFERENCES

1. Beck, J.L., Katafygiotis, L.S. 1998. Updating models and their uncertainties. i: Bayesian statistical framework. Journal of Engineering Mechanics; 124(4):455-461.
2. Chatzis, M.N., Chatzi, E.N., Smyth, A.W. 2014. An experimental validation of time domain system identification methods with fusion of heterogeneous data. Earthquake Engineering and Structural Dynamics; 44(4):523-547.
3. Chatzis, M.N., Chatzi, E.N., Smyth, A.W. 2015. On the observability and identifiability of nonlinear structural and mechanical systems. Structural Control and Health Monitoring; 22:574-593.
4. Udwadia, F., Sharma, D. 1978. Some uniqueness results related to building structural identification. SIAM Journal on Applied Mechanics; 34(1):104-118.
5. Franco, G., Betti, R., Longman, R.W. 2006. On the uniqueness of solutions for the identification of linear structural systems. Journal of Applied Mechanics; 73(1):153-162.
6. Mukhopadhyay, S., Luay, H., Betti, R. 2014. Modal parameter based structural identification using input-output data: minimal instrumentation and global identifiability issues. Mechanical Systems and Signal Processing; 45(2):283-301.
7. Katafygiotis, L.S., Beck, J.L. 1998. Updating models and their uncertainties. ii: model identifiability. Journal of Engineering Mechanics; 124(4):463-467.
8. Papadimitriou, C. 2004. Optimal sensor placement methodology for parametric identification of structural systems. Journal of Sound and Vibration; 278(4):923-947.
9. Chatzis, M.N., Chatzi, E.N., Triantafyllou, S.P. 2017. A Discontinuous Extended Kalman filter for non-smooth dynamic problems. Mechanical Systems and Signal Processing. 92:13-29.
10. Chatzis, M.N., Chatzis E.N. 2017. A Discontinuous Unscented Kalman Filter for Non-Smooth Dynamic Problems, Frontiers in Built Environment.
11. Hermann, R., Krener, A.J. 1977. Nonlinear controllability and observability. Automatic Control, IEEE Transactions; 22(5):728-740.
12. De Jager, B. 1995. The use of symbolic computation in nonlinear control: Is it visible? IEEE Trans. Autom. Control; 40(1):84-89.
13. De Jager, B. 1999. Symbolic computation in nonlinear control system modelling and analysis. 10^{th} IEEE Int. Symp. on Computer Aided Control System Design jointly with the 1999 Conference on Control Applications (1999 IEEE CCA/CACSD), Kohala Coast, Island of Hawaii, USA.
14. Farina, M., Findeisen, R., Bullinger, E. 2006. Results towards identifiability properties of biochemical reaction networks. Proc. 45th IEEE Conf. Decision Control; 2104-2109.
15. Kugi, A., Schlacher, K., Novak, R. 1999. Symbolic computation for the analysis and synthesis of nonlinear control systems. International conference on software for electrical engineering analysis and design; 255-264.
16. Sedoglavic, A. 2002. A probabilistic algorithm to test local algebraic observability in polynomial time. Journal of Symbolic Computation; 33:735-755.
17. Robenack, K., Reinschke, K.J. 2004. The computation of Lie derivatives and Lie brackets based on automatic differentiation. Z. Angew. Math. Mech; 84(2):114-123.