

UC San Diego

UC San Diego Previously Published Works

Title

On choosing state variables for piecewise-smooth dynamical system simulations

Permalink

<https://escholarship.org/uc/item/1js9n19g>

Journal

Nonlinear Dynamics, 95(2)

ISSN

0924-090X

Authors

Pei, Jin-Song

Wright, Joseph P

Gay-Balmaz, François

et al.

Publication Date


2019

DOI

10.1007/s11071-018-4622-2

Peer reviewed

On choosing state variables for piecewise-smooth dynamical system simulations

Jin-Song Pei  · Joseph P. Wright ·
François Gay-Balmaz · James L. Beck ·
Michael D. Todd

Received: 11 May 2018 / Accepted: 2 October 2018
© Springer Nature B.V. 2018

Abstract Choosing state variables in a state-space representation of a nonlinear dynamical system is a nonunique procedure for a given input–output relationship and therefore a potentially challenging task. It can be even more challenging when there are piecewise-defined restoring forces, as in bilinear hysteresis or Bouc–Wen models, which are just two of many such engineering mechanics models. Using various piecewise-smooth models, we make use of flow- and effort-controlled system concepts, common to bond graph theory, to initiate our state variable selection task, and we view numerical simulation as being within the framework of hybrid dynamical systems. In

order to develop accurate and efficient time integration, we incorporate MATLAB’s state event location algorithm, which is a mathematically sound numerical solver that deserves to be better known in the engineering mechanics community. We show that different choices of state variables can affect state event implementation, which in turn can significantly affect accuracy and efficiency, as judged by tolerance proportionality and work–accuracy diagrams. Programming details of state event location are included to facilitate application to other models involving piecewise-defined restoring forces. In particular, one version of the Bouc–Wen–Baber–Noori (BWBN) class of models is implemented as a demonstration.

J.-S. Pei (✉)
School of Civil Engineering and Environmental Science,
University of Oklahoma, Norman, Oklahoma 73019-1024,
USA
e-mail: jspei@ou.edu

J. P. Wright
Division of Applied Science, Weidlinger Associates Inc.,
New York, NY 10005, USA

F. Gay-Balmaz
CNRS LMD IPSL, Ecole Normale Supérieure de Paris, 24
Rue Lhomond, 75005 Paris, France

J. L. Beck
Department of Computing and Mathematical Sciences,
California Institute of Technology, Mail Code 9-94,
Pasadena, CA 91125, USA

M. D. Todd
Department of Structural Engineering, University of
California, San Diego, 9500 Gilman Drive,
Mail Code 0085, La Jolla, CA 92093, USA

Keywords Restoring force model · Bilinear hysteresis model · Bouc–Wen model · BWBN model · Hybrid dynamical system · State event location algorithm · Bond graph theory · Flow-controlled system · Effort-controlled system

1 Introduction

1.1 Motivating example: state event location algorithm

This study builds on previous work developing accurate and efficient numerical methods for simulating nonlinear dynamical systems that include piecewise-defined restoring forces [34]. Throughout this paper, the fol-

lowing equation of motion is used based on Newton’s second law:

$$m\ddot{x} + r = u \tag{1}$$

where $x = x(t)$ is the displacement of the mass m at time t , \ddot{x} is the acceleration, r is the restoring force generated by the elements connected to the mass that tends to restore it to its neutral position and $u = u(t)$ is an arbitrary excitation force applied to the mass. For example, Eq. (1) could describe steady-state or transient dynamic response due to either force or base excitation. For simplicity in this study, we set mass $m = 1$, $x(0) = 0$, $\dot{x}(0) = 0$ unless otherwise specified.

In this study, $r = r(x)$ is a piecewise function of x , making the dynamical system a hybrid dynamical system, meaning a system with interactions of both continuous and discrete state/algebraic variables. We assume that the displacement, velocity and restoring force are continuous functions of time, meaning the transition from one piece to the next is continuous, and that they are differentiable within each piece. However, the instantaneous stiffness, defined piecewise as dr/dx , may change abruptly from piece to piece. As needed herein, we will review ideas and supply the required definitions as they apply to a specific hybrid dynamical system (e.g., [11,30]).

In [34], we focused on the Ramberg–Osgood model which has a piecewise-smooth algebraic form $r = r(x)$. In this study, more complicated models such as bilinear hysteresis or Bouc–Wen models [4,6,7,31] will be investigated, where r is defined piecewise in either algebraic or differential format.

As a motivating example involving “state event location” [27], we present results of a single-degree-of-freedom (SDOF) simulation with the trilinear elastic model shown in Fig. 1. The force–displacement relation, r versus x , is defined by three linear pieces joined at two points (x_y, r_y) and $(-x_y, -r_y)$, with $x_y = 0.75$ and $r_y = 0.75$ where x_y denotes yield displacement and r_y denotes yield force. Thus, the slope of the central piece defines a spring stiffness $k = 1$. The two other pieces have equal stiffness values $\alpha k = 0.3$.

We utilize the state event algorithm in [27] for accuracy and efficiency reasons. This algorithm ends time stepping “exactly” (within an approximation error) at each time when yield displacement occurs, such a time being called an “event,” and then restarts there with accurately calculated values of all time-dependent

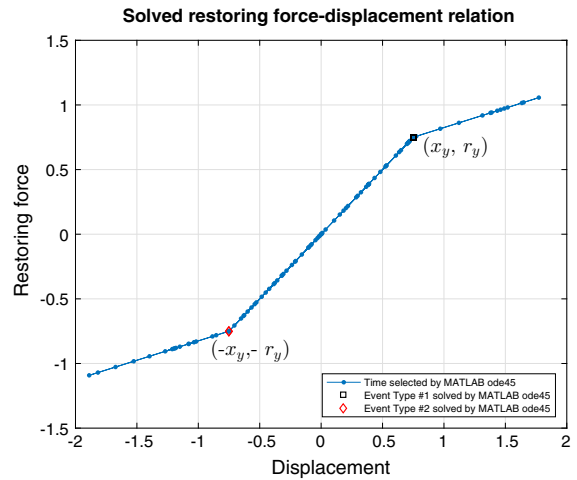


Fig. 1 Trilinear elastic model: central piece has stiffness $k = 1$; the two other pieces have $\alpha k = 0.3$; this result comes from SDOF subjected to one cycle of sinusoidal excitation force $u(t)$ (shown at top of Fig. 2); simulation used “event option” under MATLAB ode45 with RelTol = 10^{-3} and AbsTol = 10^{-12} . Event Types #1 and 2 are defined in Eqs. (2) and (3), respectively

quantities (displacement, velocity and restoring force in this example). This algorithm was implemented by combining the event option with ode45 in MATLAB. For the trilinear elastic model in Fig. 1, the event option needs to implement two *event functions*:

$$\text{Event Type \#1 : } x(t) - x_y = 0 \tag{2}$$

$$\text{Event Type \#2 : } x(t) + x_y = 0 \tag{3}$$

which are solved for the time instances when these event functions are satisfied so that the numerical integration will stop and then restart at all time instances that are discovered (i.e., not known *a priori*) in the process of numerical simulation. These events are called state events as they are functions of the state variables (x in this case).

Figure 2 presents simulation results for the trilinear elastic model. Time instances corresponding to the two types of event are highlighted in both Figs. 1 and 2. Although this is a simple model, with the excitation force $u(t)$ being the smooth sinusoid shown at the top of Fig. 2, the seemingly simple idea of state event location requires care to be taken before, during and after any event occurs. The procedure was discussed in [34] using both the Ramberg–Osgood and Bingham models, and will be further elucidated in this study using other

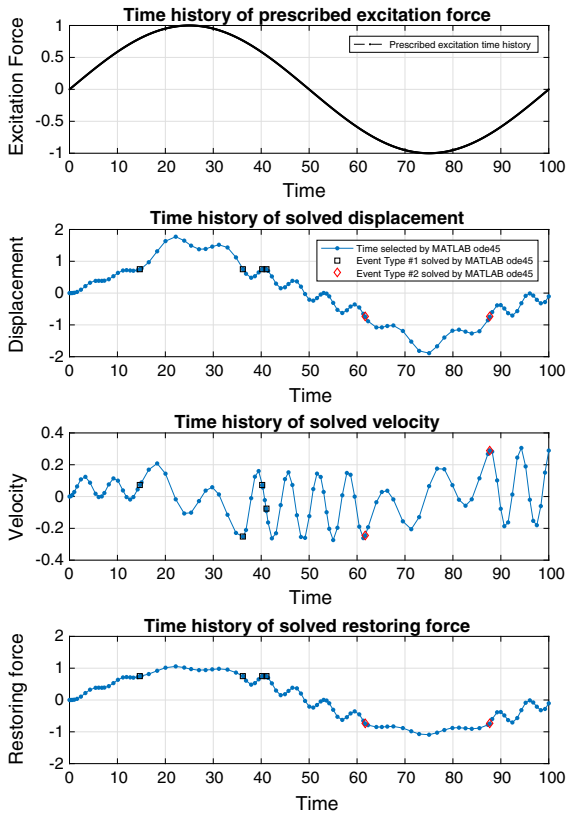


Fig. 2 Results from SDOF simulation with trilinear elastic model shown in Fig. 1; simulation used sinusoidal excitation force $u(t)$ shown at top and “event option” under MATLAB ode45 with $RelTol = 10^{-3}$ and $AbsTol = 10^{-12}$. Event Types #1 and 2 are defined in Eqs. (2) and (3), respectively

more complicated models. More information concerning the algorithm itself is given in Sect. 2.1.

1.2 Motivating example: choices of state variables

State-space representation has been the cornerstone for both theoretical and numerical studies of nonlinear dynamical systems. Loosely speaking, state variables for a dynamical system can be used to model the system as a set of first-order ordinary differential equations (ODE) called state equations:

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, \mathbf{u}, t) \tag{4}$$

where \mathbf{u} is an input vector (includes initial conditions) and \mathbf{y} is a state vector containing all state variables. For example, displacement and velocity are commonly

chosen as state variables when formulating the state equations corresponding to Eq. (1):

$$\mathbf{y} = \begin{Bmatrix} y(1) \\ y(2) \end{Bmatrix} = \begin{Bmatrix} x \\ \dot{x} \end{Bmatrix} \tag{5}$$

$$\dot{\mathbf{y}} = \begin{Bmatrix} \dot{y}(1) \\ \dot{y}(2) \end{Bmatrix} = \begin{Bmatrix} \dot{x} \\ \ddot{x} \end{Bmatrix} = \begin{Bmatrix} y(2) \\ \frac{1}{m}(u - r(y(1))) \end{Bmatrix} = \mathbf{f}(\mathbf{y}, \mathbf{u}, t) \tag{6}$$

This choice of state variables and corresponding pair of first-order ODEs was used when implementing the state event location algorithm for the example just presented. In particular, the displacement $x(t)$ was used in defining the two event functions, which satisfies a MATLAB requirement, namely that event functions need to be defined as (simple) functions of the state variables.

For piecewise restoring force models that are more complicated, appropriate choices of state variables are not always this straightforward. Reference [33] vividly acknowledges this fundamental challenge concerning state-space formulation: “It is, however, impossible to deduce *a priori*, in physical terms, what will be the state. This, indeed, is a very difficult problem even for relatively simple systems, and it appears to be the cause for much of the reluctance of introducing this concept in physics.”

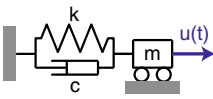
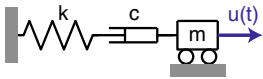
Table 1 offers motivating examples of four ways of choosing state variables for two different basic (linear) dynamical systems: (a) a mass in series with a Kelvin model (also called Kelvin–Voigt) and (b) a mass in series with a Maxwell model. A Kelvin model has a spring and dashpot in parallel, whereas a Maxwell model has a spring and dashpot in series.

Often when modeling a dynamical system, numerous choices of state variables are possible, although many are not straightforward, nor even physically or computationally convenient. The text and “Appendix A” provide additional discussion and examples, particularly Sect. 2.2 (flow- and effort-controlled systems) and Sect. 4 (Bouc–Wen model), where the first and second columns in Table 1 will be elaborated in Sect. 2.2.

1.3 Contributions and structure of this paper

This paper builds on previous work [34] promoting the use of hybrid dynamical systems for nonlinear hysteresis modeling. The utilization of a state event location algorithm is validated here for various piecewise-

Table 1 Possible choices of state variables for basic (linear) dynamical systems

Choice	State variable	(a) Flow-controlled: mass + kelvin	(b) Effort-controlled: Mass + Maxwell
			
1 (F)	$y_1 = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$	$\dot{y}_1 = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} y_1 + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u$	$\dot{y}_1 = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{k}{c} \end{bmatrix} y_1 + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} \int u dt$
2 (E)	$y_2 = \begin{bmatrix} p \\ r \end{bmatrix}$	$\dot{y}_2 = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} y_2 + \begin{bmatrix} 0 \\ \frac{c}{m} \end{bmatrix} u + \begin{bmatrix} 0 \\ \frac{k}{m} \end{bmatrix} \int u dt$	$\dot{y}_2 = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{k}{c} \end{bmatrix} y_2 + \begin{bmatrix} 0 \\ \frac{k}{m} \end{bmatrix} \int u dt$
3 (M)	$y_3 = \begin{bmatrix} x \\ p \end{bmatrix}$	$\dot{y}_3 = \begin{bmatrix} 0 & -\frac{1}{m} \\ k & -\frac{c}{m} \end{bmatrix} y_3 + \begin{bmatrix} \frac{1}{m} \\ \frac{c}{m} \end{bmatrix} \int u dt$	$\dot{y}_3 = \begin{bmatrix} 0 & -1 \\ k & -\frac{k}{c} \end{bmatrix} y_3 + \begin{bmatrix} \frac{1}{m} \\ 0 \end{bmatrix} \int u dt$
4 (M)	$y_4 = \begin{bmatrix} \dot{x} \\ r \end{bmatrix}$	$\dot{y}_4 = \begin{bmatrix} 0 & -\frac{1}{m} \\ k & -\frac{c}{m} \end{bmatrix} y_4 + \begin{bmatrix} \frac{1}{m} \\ \frac{c}{m} \end{bmatrix} u$	$\dot{y}_4 = \begin{bmatrix} 0 & -1 \\ k & -\frac{k}{c} \end{bmatrix} y_4 + \begin{bmatrix} \frac{1}{m} \\ 0 \end{bmatrix} u$

defined constitutive models. Furthermore, bond graph theory is introduced to directly benefit the implementation of the state event location algorithm. More specifically, flow- versus effort-controlled system classification is applied to systematically generate different choices of state variables, which are further tested through programming and quantitative measures of computational efficiency. Even though the primary focus is on the state event location algorithm, this study not only exercises the flow- versus effort-controlled systems classification, but also provides theoretical discussion of them. For both simulation and classification, there exist nonunique choices of state variables for piecewise-defined models.

Section 1, where a simple trilinear elastic model was presented to illustrate state event location, serves as an introduction. Section 2 discusses quantitative measures of computational performance (tolerance proportionality, work–accuracy diagrams), flow- and effort-controlled systems, and the essential elements in hybrid dynamical system modeling that are relevant to this study. Sections 3 and 4 focus on a comprehensive study of bilinear hysteresis and Bouc–Wen models, respectively. Section 5 is dedicated to guidelines on choices of state variables. Section 6 concludes this study.

2 Theoretical issues

2.1 Discontinuity sticking correction and performance measures for state event location algorithm

MATLAB’s ode45 uses a pair of Runge–Kutta formulas due to Dormand and Prince [8]. Numerical integra-

tion is done by computing a mesh of time points that are generated adaptively, normally not at a fixed time step. Great care and significant effort were taken to correct *discontinuity sticking*, which is a numerical problem caused by algebraic variable(s) not being updated properly at a state event location [34].

Following [34], the accuracy of our numerical solutions is assessed by studying the global error (GE) of the displacement at a specific time t_n :

$$GE(t_n) = |\hat{x}(t_n) - x(t_n)| \tag{7}$$

where $x(t_n)$ is the exact displacement at t_n and $\hat{x}(t_n)$ is the approximated displacement at that time. Since the exact displacement is unknown (in most cases), a highly converged numerical solution is obtained by using a very small value of the relative tolerance parameter called RelTol in MATLAB. In order to control approximation errors and assess accuracy in our numerical work, we fix the value of AbsTol to a very small value while allowing RelTol to take on a wide range of values down to the fixed value of AbsTol.

The algorithm behind the event option under ode45 is considered quite robust according to some studies [9,21] and reviewed in [34]. For example, continuing our discussion of the trilinear elastic model in Fig. 1, Fig. 3 shows results for a wide range of RelTol values, with algorithmic robustness being evident for RelTol < 10⁻³ (approx.) based on the identified number of Event #1 and #2 defined earlier and convergence of the identified time of each event. The results in Fig. 2 correspond to RelTol = 10⁻³.

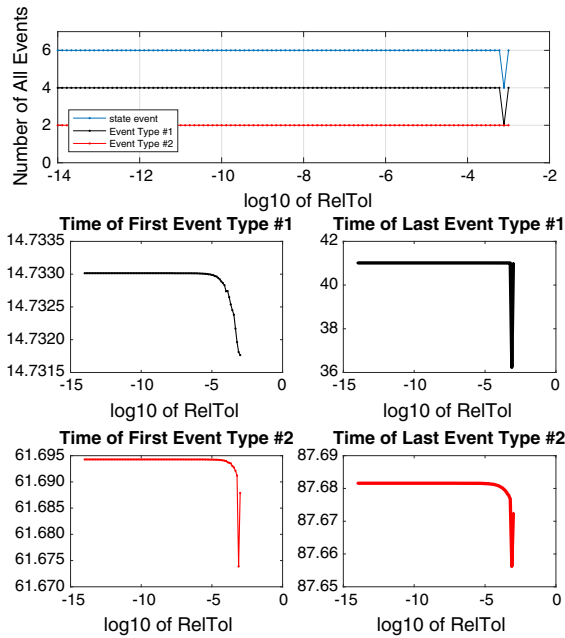


Fig. 3 Number of state events and their event time values plotted against MATLAB ode45 RelTol for the trilinear elastic model in Fig. 1. Event Types #1 and 2 are defined in Eqs. (2) and (3), respectively

Following our work in [34], *tolerance proportionality (TP)* and *work–accuracy diagrams* are also provided here to demonstrate the computational stability and efficiency of state event location. TP measures the relationship between GE and RelTol for explicit, adaptive Runge–Kutta time-stepping methods, while work–accuracy diagram measures computational stability. For example, above the upper left plot in Fig. 4, the TP slope is 0.965, obtained by a straight line fit to the TP data. This slope is quite close to one, which indicates very good computational stability for this series of trilinear elastic model simulations. Also, the work–accuracy diagrams for elapsed time, successful and unsuccessful function evaluations (FE) are all quite smooth, which again affirms good computational stability.

2.2 Flow- versus effort-controlled systems

The state variable alternatives in Table 1 are obtained, first, by applying bond graph theory, especially recent research on flow- and effort-controlled system classification (e.g., the memristor, memcapacitor and their

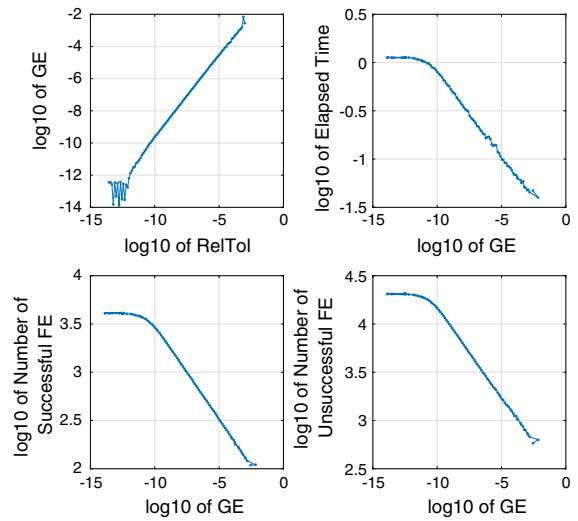


Fig. 4 Tolerance proportionality (upper left) and work–accuracy diagrams (the rest) for the trilinear elastic model in Fig. 1, where GE and FE stand for global error and function evaluation, respectively

advanced systems, e.g., [24,25]). Following the format of Fig. 1 in [29], Fig. 5 depicts Paynter’s tetrahedron of state [22,23] for (a) electrical circuits, which can be translated to mechanical systems using (b) force–voltage analogy, and (c) force–current analogy.

For mechanical systems, a commonly chosen pair of state variables is velocity (called *flow* in bond graph theory) and displacement (the time integral of flow), as in Eq. (5) or Choice No. 1 in Table 1. An alternative pair of state variables is restoring force $r(t)$ (called *effort* in bond graph theory) and corresponding “momentum” p (e.g., [16]) where

$$p(t) = \int_{-\infty}^t r(\tau)d\tau \tag{8}$$

as in Choice No. 2 in Table 1. The name momentum for $p(t)$ is in accord with [16]. Not to be confused with momentum $m\dot{x}$, p will be renamed as “g-momentum” herein, where “g” is short for “generalized” following [22]. For the autonomous case ($u = 0$) of Eq. (1), Eq. (8) holds with $p = -m\dot{x}$ if the system is initially at rest but at a displaced position.

Kelvin and Maxwell models, the two systems in Table 1, form a dual in the sense that the connectivity of elements in parallel and series corresponds, respectively, to flow- and effort-controlled classification of

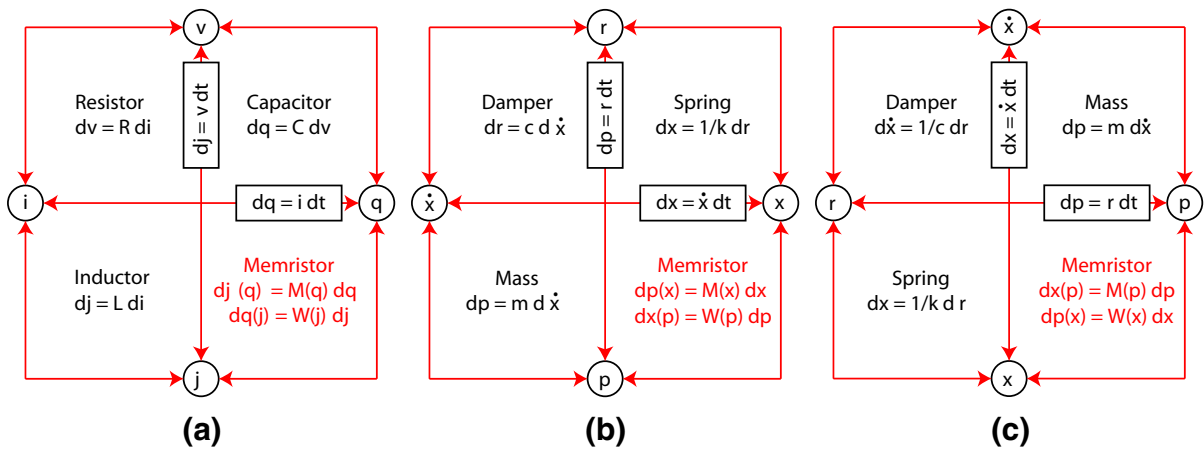


Fig. 5 Planar depiction of Paynter’s tetrahedron of state for the four fundamental **a** electrical circuit elements following Fig. 1 in [29], **b** mechanical system elements using force–voltage analogy and **c** mechanical system elements using force–current analogy

systems. In (a) flow-controlled system, a mass plus a Kelvin model, the natural state variables are displacement x and velocity \dot{x} . These state variables should be solved (or calculated) first from force equilibrium, which is typically expressed as an ODE involving these state variables. In contrast, in (b) effort-controlled system, a mass plus a Maxwell model, the natural state variables are g-momentum p and restoring force r . These state variables should be solved (or calculated) first from deformation compatibility, which is often expressed as another ODE involving the state variables. These two choices, which are labeled as (a) and (b) in Table 1, are from the perspective of flow- and effort-controlled classification, and are so indicated by (F) and (E), respectively.

There is a well-established procedure under bond graph theory on how to assign and then (if possible) reduce state variables in writing up system equations for numerical simulation (or even skipping system equations) [16, 22, 23]. The key idea in bond graph theory is to identify energy storage elements, masses and springs, in a system, and then assign energy storage variables, namely g-momenta and displacements for masses and springs, respectively, before other details are carried out. This means that the first choices for state variables would be the integral of the internal forces and displacements, a mixture of effort- and flow-controlled variables. Hence, this choice of state variables leads to a so-called *mixed* (M) system. Returning to the examples in Table 1, the state variables would be g-momentum p and displacement x , which is Choice No. 3 in Table 1.

Choice No. 4, which is used, for example, in the Bouc–Wen model [3], does not fit naturally into bond graph theory in the form given, and is a mixed (M) system. However, the Bouc–Wen model can be written in effort-controlled form, consistent with bond graph theory [24].

The motivating examples in Table 1, albeit being linear, reveal that various possibilities do exist to select state variables and their corresponding state equations. In particular, there are alternatives to displacement and velocity which are often used in practice.

To complement Table 1, additional examples of nonunique state-space representations are given in Tables 2 and 3 in “Appendix A.” For brevity, derivations for all these expressions are not provided in this paper; however, key technical details start with manipulating Eq. (1) as follows:

$$m\ddot{x} = u - r \implies \int m\ddot{x}dt = \int (u - r)dt \quad m = \text{a constant} \implies \underbrace{m\dot{x}(t)}_{\text{momentum}} = m\dot{x}(0) + \underbrace{\int udt}_{\text{impulse}} - \underbrace{\int rdt}_{\text{g-momentum}} \quad (9)$$

Equation (9) indicates that p , a kinetic quantity, and \dot{x} , a kinematic quantity, are related (principle of impulse–momentum) and so are not independent, which has two implications:

1. A state variable vector \mathbf{y} does not need to have all four vertices in Paynter’s tetrahedron of state, x, \dot{x}, p, r , the fact of which can be verified in

Tables 1, 2 and 3. To obtain state equations with full rank, one cannot have \dot{x} and p coexisting in one state variable vector as long as Eq. (1) holds.

2. All state equations in Table 1 can be made equivalent. We have the following pairs:

Choice #1 (F) \leftrightarrow #3 (M) by replacing \dot{x} with p
 Choice #2 (E) \leftrightarrow #4 (M) by replacing p with \dot{x}
 Choice #3 (M) \leftrightarrow #4 (M) by adjusting the order of state variables

With the equivalency of these pairs in place, it can be seen that Choices #1 (F) and #2 (E) can be made equivalent, leading ultimately to the equivalency of all choices in Table 1, meaning that the solution of any choice is mathematically the same as any other. However, for nonlinear models, one choice of state variables may be superior to another, based on physical reasoning, interpretation of experimental results, and even on numerical convenience (including accuracy and efficiency).

2.3 Proposed components for hybrid systems

Hybrid dynamical systems, or hybrid systems in short, deal with the interaction between continuous and discrete time variables. Different research communities approach hybrid systems from different angles and with different agendas. In [30], a hybrid automaton is defined formally by a septuple and includes the use of differential–algebraic equations (DAEs). The definition in [11], on the other hand, seems simpler and involves six components with differential inclusion utilized. We will focus on the following components for a typical hybrid dynamical system by applying the definition in [11] but with some amendments:

1. *Mode* A piecewise-defined $r = r(x)$ model corresponds to a collection of discrete characteristic statuses that the model has, called modes. For example, in the trilinear elastic example, one can say that there are two modes when either $k = 1$ or $\alpha k = 0.3$ applies. Normally, a *transition diagram* is used to illustrate the relationships among multiple modes.
2. *Domain* Within a mode, there are state variable(s) and perhaps algebraic variable(s). A domain is a region where the time evolution of the state variable(s) and perhaps algebraic variables' values are given by a specified mode.
3. *Flow map* Within a mode, the state variables are continuous and smooth when the state equation(s)

are the governing equation. This state equation is called a flow map. Algebraic equation(s) may be needed as well so the flow map consists of differential–algebraic equations (DAEs).

4. *Event* An event is when the mode makes a transition from one status to another. It is also called an edge, or a jump. It is normally a state event defined by state variables—sometimes, algebraic variable(s), as well.
5. *Event Function* An event function defines a criterion for an event's occurrence. It is also called a guard. It is defined by using state variables—and sometimes, algebraic variable(s), as well.
6. *Reset Map* A reset map defines mathematically how the process of reset takes place right after an event. In general, a reset map can include both state and algebraic variables.

These components are applied to both bilinear hysteresis and Bouc–Wen models in Sects. 3 and 4, respectively.

3 Bilinear hysteresis model

3.1 Bilinear hysteresis model: analysis

Bilinear hysteresis models (e.g., [6, 7]) are popular for modeling plastic deformation. The four *modes*, I to IV, defined in [15] are illustrated first by using a simulated example in Fig. 6a to replace the cartoon in Fig. 1(a) in [15], and also by a mode transition diagram extended slightly from Fig. 1(b) in [15]—with details substantiated in Table 4 in “Appendix B.” These details closely follow the hybrid system components elaborated above so that the hybrid system concept directly facilitates the understanding and programming. The four algebraic equations given in Eq. (2) in [15] are rewritten in a slightly more general form as follows:

$$\text{Mode I: } r = k(x - x_0) \tag{10}$$

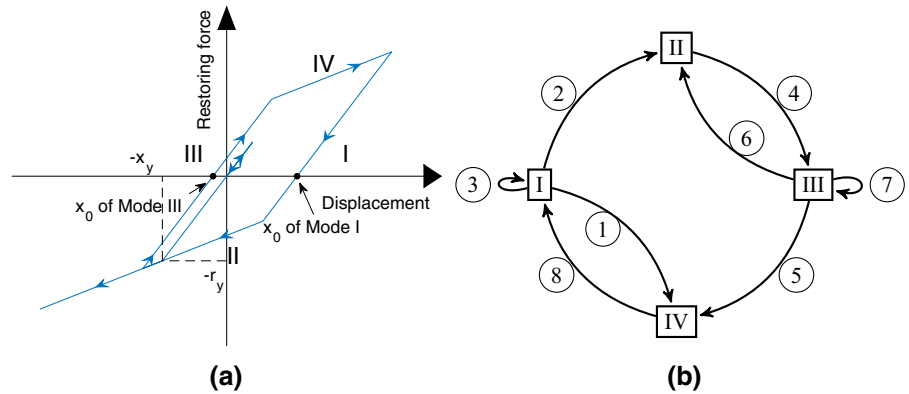
$$\text{Mode II: } r = \alpha kx - (1 - \alpha)r_y \tag{11}$$

$$\text{Mode III: } r = k(x - x_0) \tag{12}$$

$$\text{Mode IV: } r = \alpha kx + (1 - \alpha)r_y \tag{13}$$

where k , r_y and α are system properties similar to the trilinear example in Fig. 1, and $x_0 = x_0(t)$ is a quantity that changes as the simulation proceeds, starting with $x_0(0) = 0$. It is the reset map that updates the

Fig. 6 Bilinear hysteretic model following a flow-controlled formulation. **a** Illustration of hysteresis loops using a simulated result. **b** Mode transition diagram; see Table 4 for ① to ⑧



value of x_0 through the memory parameters H_l , H_u and O in programming, details of which are given in “Appendix B.”

As shown in Fig. 6a, Modes II and IV are the low and high envelope as defined in Eqs. (11) and (13), respectively. Modes I and III, however, cannot be grasped comprehensively from Fig. 6a, which does not cover all possible behaviors of these two modes.

The comprehensive depiction of Modes I and III is in Fig. 6b in conjunction with Table 4. Mode I can both stem from ⑧ and return to ① Mode IV; however, Mode I can only jump to ② Mode II but Mode II cannot jump to Mode I. When Mode I has an unfulfilled return to Mode IV, an internal loop (still on the same straight line) forms ③. Conversely, Mode III can both stem from ④ and return to ⑥ Mode II; however, Mode III can only jump to ⑤ Mode IV but Mode IV cannot jump to Mode III. When Mode III has an unfulfilled return to Mode II, an internal loop (still on the same straight line) forms ⑦. Therefore, Modes I and III are not the same – even when Eqs. (10) and (12) appear the same (where x_0 is actually being reset).

It is tempting to think that these four piecewise-defined formulas for $r(t)$ could be readily embedded in a flow-controlled setting with displacement and velocity as state variables, such as Eqs. (5) and (6), and implemented as was done for the Ramberg–Osgood model in [34]. However, bilinear hysteresis has more complicated events, event functions (for MATLAB implementation) and reset map, leading to an implementation similar in kind, but not in detail, to the Ramberg–Osgood model. Due to this complexity, we examined several formulations and ultimately implemented two: (1) a flow-controlled formulation (F) and (2) a mixed flow- and effort-controlled formulation (M), herein

called “partially effort-controlled.” The two formulations have their own state variables, event functions, flow map and reset map, but the modes and events are the same for the same applied force $u(t)$. The partially effort-controlled scheme is simpler to implement and more efficient in performance.

Regarding the mixed (M) formulation, the bilinear hysteresis model can be treated as a “Jenkin’s element” and a linear spring connected in parallel, whose response is illustrated in blue in Fig. 7a and decomposed in Fig. 7b. The linear spring is used to capture the hardening effect (whose response is plotted in red in Fig. 7b), while the Jenkin’s element is another linear spring and a Coulomb damper connected in series, an effort-controlled device (whose response is plotted in black in Fig. 7b). Thus, the bilinear hysteresis model can be implemented as a mixed device (flow- and effort-controlled mixture), called partially effort-controlled.

When the bilinear model is treated as a “Jenkin’s element” and a linear spring (for hardening effect) connected in parallel, we have:

$$r = r_1 + r_2 = r_1 + \alpha kx \tag{14}$$

where r_1 and r_2 represent the contribution to the restoring force from the Jenkin’s element and linear spring, respectively.

3.2 Bilinear hysteresis model: programming

Equations (5) and (6) are rewritten into the following:

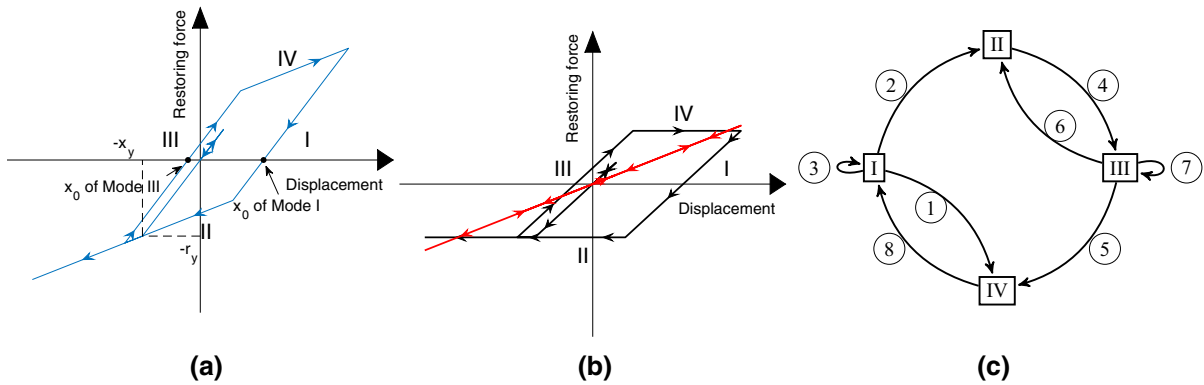


Fig. 7 Bilinear hysteretic model following a partially effort-controlled formulation. **a** Illustration of hysteresis loops using a simulated result. **b** An equivalent sum for hysteresis loops. **c** Mode transition diagram; see Table 4 for ① to ⑧

$$y_F = \begin{Bmatrix} y_F(1) \\ y_F(2) \end{Bmatrix} = \begin{Bmatrix} x \\ \dot{x} \end{Bmatrix}, \tag{15}$$

$$\dot{y}_F = \begin{Bmatrix} \dot{y}_F(1) \\ \dot{y}_F(2) \end{Bmatrix} = \begin{Bmatrix} \dot{x} \\ \ddot{x} \end{Bmatrix} = \begin{Bmatrix} y_F(2) \\ \frac{1}{m}(u - r) \end{Bmatrix} \tag{16}$$

where $r = r(x, x_0)$ and the subscript F is for flow-controlled. To complete the *flow map*, the algebraic equations corresponding to Eqs. (10)–(13) are given in Table 5 in “Appendix B,” using tag_2 , a mode indicator established in [34], and O , a programming notation for one of the memory parameters in this study (see “Appendix B”).

For the mixed (M) formulation, the *flow map* is given by:

$$y_M = \begin{Bmatrix} y_M(1) \\ y_M(2) \\ y_M(3) \\ y_M(4) \\ y_M(5) \end{Bmatrix} = \begin{Bmatrix} x \\ \dot{x} \\ p = p_1 + p_2 \\ r_1 \\ r_2 \end{Bmatrix}, \tag{17}$$

$$\dot{y}_M = \begin{Bmatrix} \dot{y}_M(1) \\ \dot{y}_M(2) \\ \dot{y}_M(3) \\ \dot{y}_M(4) \\ \dot{y}_M(5) \end{Bmatrix} = \begin{Bmatrix} \dot{x} \\ \ddot{x} \\ r = r_1 + r_2 \\ \dot{r}_1 \\ \dot{r}_2 \end{Bmatrix}$$

$$= \begin{Bmatrix} y_M(2) \\ \frac{1}{m}(u(t) - y_M(4) - y_M(5)) \\ y_M(4) + y_M(5) \\ (1 - tag_2)(1 - \alpha)ky_M(2) \\ \alpha ky_M(2) \end{Bmatrix} \tag{18}$$

where tag_2 is the same mode indicator explained in Table 5 with the differential equations corresponding to Eqs. (10)–(13).

For bilinear hysteresis, three types of events and event functions are needed, two of which involve displacement $x(t)$ while the third type is a so-called velocity-turning point, meaning a time when the velocity is zero:

$$\text{Event Type \#3 : } \dot{x}(t) = 0 \tag{19}$$

The F and M *events* (including Event Types #1 and #2 that are not presented here), *event functions* and *reset maps* are given in Table 4 in “Appendix B” covering all allowed mode transitions, from I to II, I to IV, II to III, III to IV, III to II and IV to I. The columns “direction” contain the information on *domain* for programming. The F and M reset maps are more complicated than the reset map for the trilinear elastic model. In the mixed (M) formulation, the switching values of tag_2 are fewer and the reset map is significantly simpler (without involving updating the memory parameters H_l , H_u and O as indicated in [†] in Table 4), in comparison with the flow-controlled (F) formulation.

For both the F and M formulations (flow-controlled and partially effort-controlled) and as illustrated in Fig. 8, *main.m* runs the ode45 solver, where the option of “options” is activated so that *Events.m* and *OutputFcn.m* are invoked. The three event functions are defined inside *Events.m*, while all outputs are recorded by using *OutputFcn.m* whenever there is a successful time step under ode45. For the F formulation, the right-hand side of Eq. (16) including the piecewise algebraic

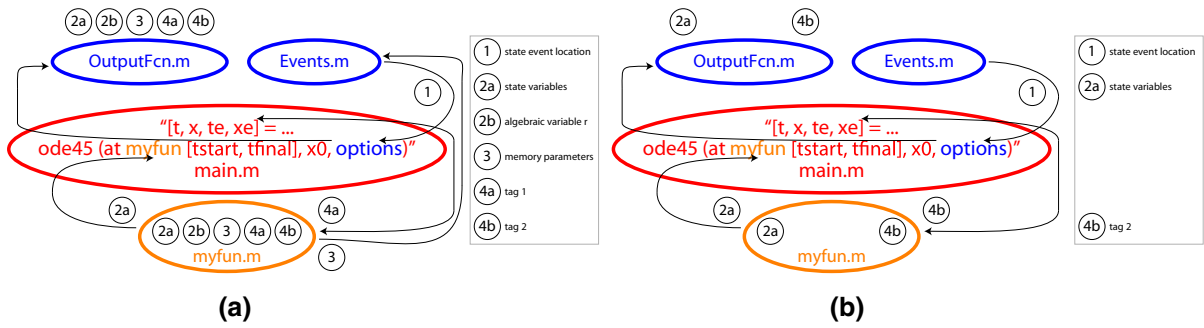


Fig. 8 Illustrations of the interactions among all MATLAB m-files. **a** Flow-controlled. **b** Partially effort-controlled

equations given in Table 5 in “Appendix B” is computed in myfun.m. For the M formulation, the right-hand side of Eq. (18) is computed in myfun.m.

As an adaptive time-stepping scheme, ode45 in MATLAB does not have fixed time steps. Control of time stepping is done through specifying RelTol—and—MaxStep and InitialStep via odeset. While the former has been investigated using tolerance proportionality (TP), the effect of the latter two become essential when simulating more complicated transient response of the bilinear hysteresis model, e.g., Fig. 22 in [15].

For both formulations, the system parameters are m, k, α, x_y and r_y (with $r_y = kx_y$). Additional programming details are given in “Appendix B” using the programming jargon and notation established in [34].

3.3 Bilinear hysteresis model: comparison between F and M formulations

The choice of state variables is straightforward for the flow-controlled formulation (F), although enabling the event functions and reset maps under the existing MATLAB ode45 setting turns out to be nontrivial, as illustrated in Fig. 8a.

For the flow-controlled formulation (F), there is strong coupling among all four MATLAB m-files, making the programming somewhat challenging. Figure 8a uses arrows to show the data flow. In contrast, the programming of the partially effort-controlled formulation (M) is simpler than the flow-controlled formulation (F). There is no strong coupling among all four MATLAB m-files, making the programming straightforward. Figure 8b uses arrows to show the data flow.

Two numerical examples with their performance measures are presented in Figs. 9, 10, 11 and 12. It can be seen that the partially effort-controlled formulation (M) reduces computational time, while other performance measures are similar, showing computational robustness in both cases.

4 Bouc–Wen model

4.1 Bouc–Wen model: analysis

The formula for the Bouc–Wen model is given as follows:

$$\dot{r} = A\dot{x} - \gamma|\dot{x}||r|^{n-1}r - \beta\dot{x}|r|^n = h(\dot{x}, r) \tag{20}$$

where A, γ, β and n are model parameters (see Eq.(5.24) in [13], which is based on [32]). Note that any elastic term, such as kx , added to r just adds k to A in Eq. (20) and so can be absorbed in A without any loss of generality. Because the absolute-value expressions $|\dot{x}|$ and $|r|$ in Eq. (20) induce discontinuities, they are treated as state events $\dot{x}(t) = 0$ and $r(t) = 0$ in this study. For comparison purposes, we also present simulation results where these discontinuities are ignored—by omitting state event coding—leading to reduced accuracy and computational efficiency.

Using Eq. (9) and the sign function (sgn), Eq. (20) can be rewritten as follows:

$$\begin{aligned} \dot{r} &= (A - [\gamma\text{sgn}(\dot{x}r) + \beta]|r|^n)\dot{x}, \\ \dot{x} &= \frac{1}{m} \left[\int_0^t u(\tau)d\tau - p \right] + \dot{x}(0), \text{ where } p = \int_0^t r(\tau)d\tau \end{aligned} \tag{21}$$

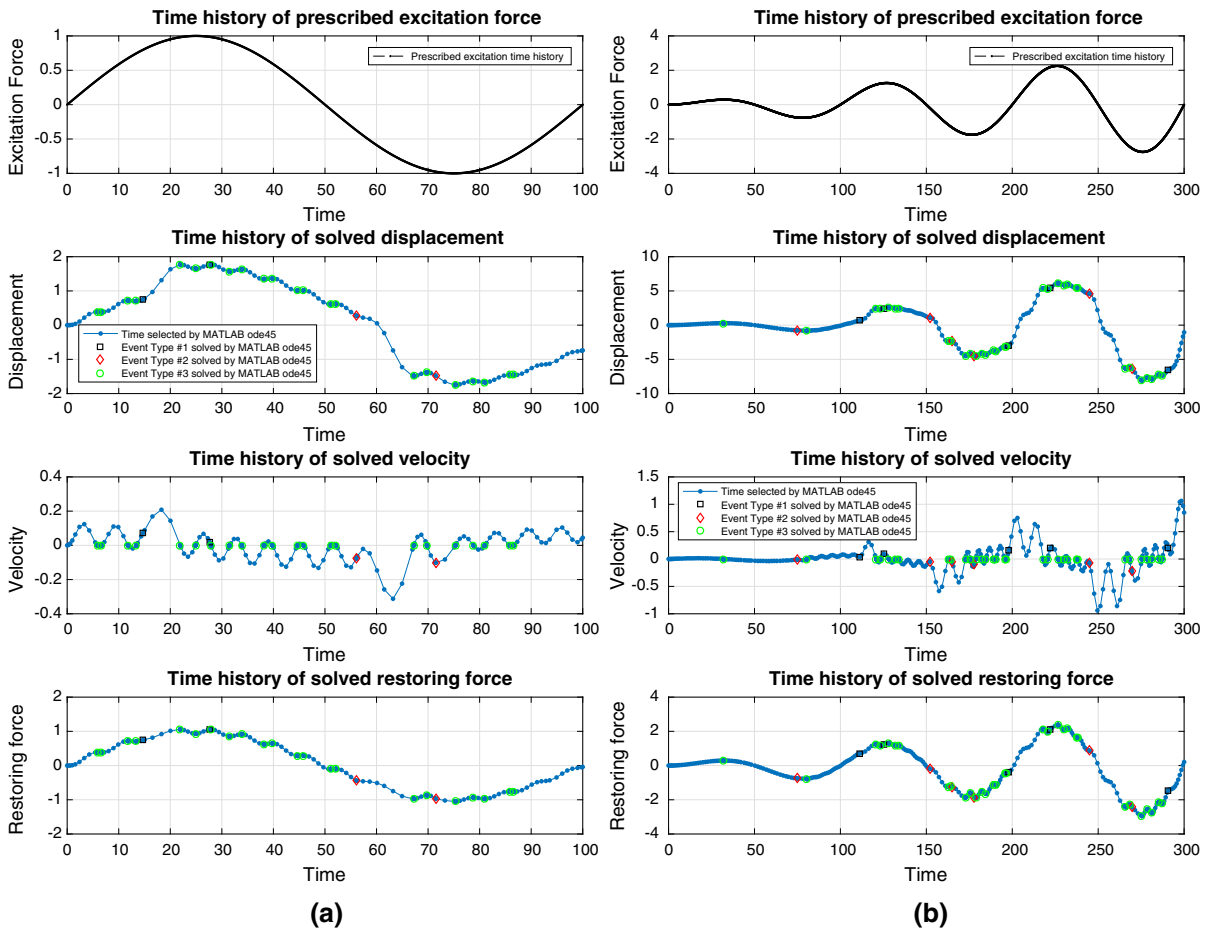


Fig. 9 Bilinear hysteresis model with $k = 1$, $\alpha = 0.3$ and $r_y = 0.75$ subject to different excitations and simulated using the “event option” under MATLAB ode45, where RelTol = 10^{-3} and AbsTol = 10^{-12} ; time histories. The flow-controlled results

(F) are presented here; the partially effort-controlled results (M) are the same. Event Types #1 to 3 are defined in Table 4 in “Appendix B.” **a** One cycle of sinusoidal excitation. **b** An amplitude-modulated sine excitation

Thus, \dot{r} is now expressed as a function of p and r . Written this way, it is evident that the Bouc–Wen element can be viewed as an effort-controlled device.

Following [25], we have the following pair of equations:

$$\begin{aligned}
 x &= \int \frac{1}{A - \gamma|r|^{n-1}r - \beta|r|^n} dr + x_+(0) \\
 &\triangleq F_+(r) + x_+(0), \quad \text{when } \dot{x} > 0
 \end{aligned}
 \tag{22}$$

$$\begin{aligned}
 x &= \int \frac{1}{A + \gamma|r|^{n-1}r - \beta|r|^n} dr + x_-(0) \\
 &\triangleq F_-(r) + x_-(0), \quad \text{when } \dot{x} < 0
 \end{aligned}
 \tag{23}$$

where $F_+(r)$ and $F_-(r)$ are the integrals with zero initial conditions. A qualitative discussion of the pair of Eqs. (22) and (23) is given as follows:

- A load versus displacement (r vs. x) plot of the Bouc–Wen model can have a piecewise definition following the sign of \dot{x} . The stiffness is entirely a function of r . This implies first the following fact: At any point on the loading path (x_0, r_0), the stiffness is always calculated by either Eqs. (22) or (23) depending on the sign of \dot{x} . This means that there are only two stiffness values for (x_0, r_0): one for all loading paths and the other for all unloading paths. This also implies that *parallel loading or unload-*

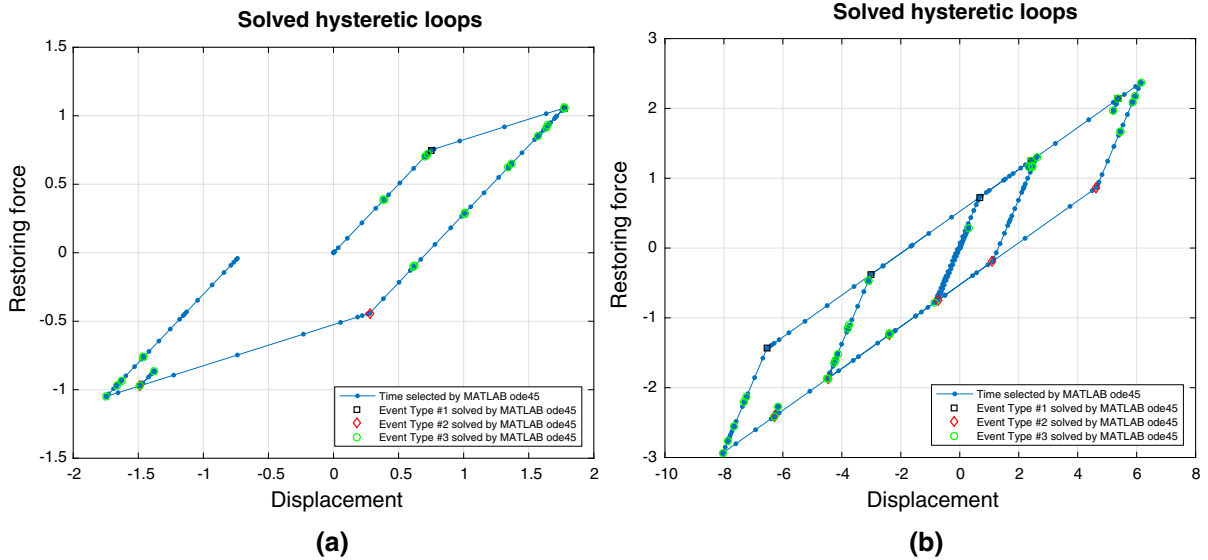


Fig. 10 Bilinear hysteresis model with $k = 1$, $\alpha = 0.3$ and $r_y = 0.75$ subject to different excitations and simulated using the “event option” under MATLAB ode45, where $RelTol = 10^{-3}$ and $AbsTol = 10^{-12}$: hysteresis loops. The flow-controlled

results (F) are presented here; the partially effort-controlled results (M) are the same. Event Types #1 to 3 are defined in Table 4 in “Appendix B.” **a** One cycle of sinusoidal excitation. **b** An amplitude-modulated sine excitation

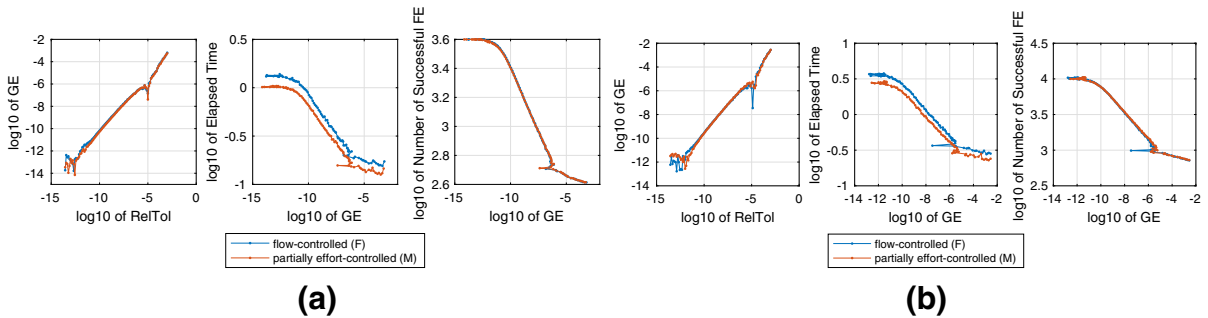


Fig. 11 Bilinear hysteresis model with $k = 1$, $\alpha = 0.3$ and $r_y = 0.75$ subject to different excitations and simulated using the “event option” under MATLAB ode45, where $RelTol = 10^{-3}$ and $AbsTol = 10^{-12}$: tolerance proportionality (TP) and work–accuracy diagrams. The results from flow-controlled (F) and

partially effort-controlled (M) simulations are compared here, where GE and FE stand for global error and function evaluation, respectively. **a** One cycle of sinusoidal excitation. **b** An amplitude-modulated sine excitation

ing paths are within the same range of r but for different ranges of x .

- The original Bouc–Wen formula has now been rewritten into a piecewise expression for the stiffness plus $\dot{x} = 0 \implies \dot{r} = 0$. This means that, essentially, the Bouc–Wen model represents a non-linear spring whose stiffness is piecewise-defined with the velocity being the switching mechanism.

- There are no viscous damping terms, linear or non-linear. Having said this, the Bouc–Wen model is rate-independent, which is often used in conjunction with viscous damping, i.e., to combine with rate-dependent components. Rate independency of the Bouc–Wen model is consistent with what is claimed in [20].

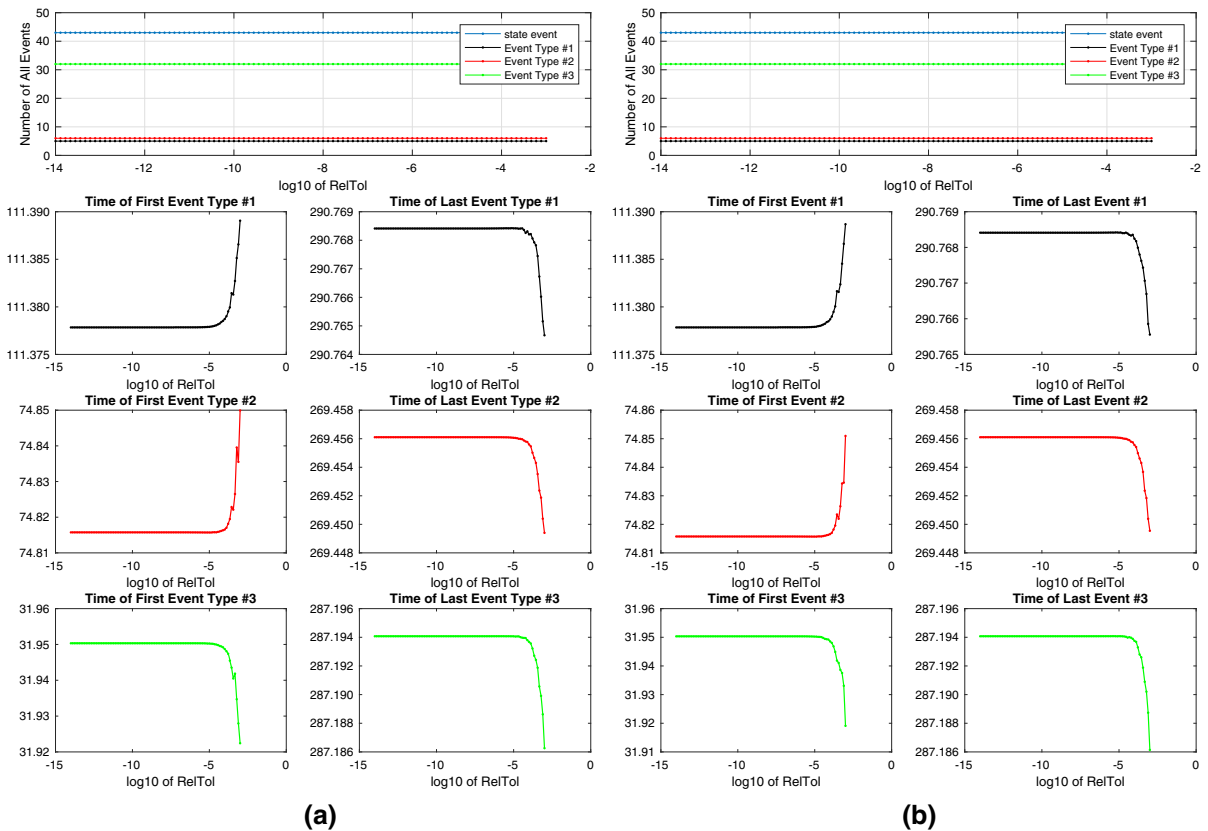


Fig. 12 Bilinear hysteresis model with $k = 1$, $\alpha = 0.3$ and $r_y = 0.75$ subject to the amplitude-modulated sine excitation and simulated using the “event option” under MATLAB ode45, where $\text{AbsTol} = 10^{-12}$: number of state events and their event

time values plotted against MATLAB ode45 RelTol. Event Types #1 to 3 are defined in Table 4 in “Appendix B.” **a** Flow-controlled. **b** Partially effort-controlled

4.2 Bouc–Wen model: programming

One way of programming the Bouc–Wen model with no state events considered at all is as follows, a version of which is denoted by “A” herein:

$$\begin{aligned}
 y_A &= \begin{Bmatrix} y_A(1) \\ y_A(2) \\ y_A(3) \end{Bmatrix} = \begin{Bmatrix} x \\ \dot{x} \\ r \end{Bmatrix}, \\
 \dot{y}_A &= \begin{Bmatrix} \dot{y}_A(1) \\ \dot{y}_A(2) \\ \dot{y}_A(3) \end{Bmatrix} = \begin{Bmatrix} \dot{x} \\ \ddot{x} \\ \dot{r} \end{Bmatrix} \\
 &= \left. \begin{Bmatrix} y_A(2) \\ \frac{1}{m}(u(t) - y_A(3)) \\ A y_A(2) - \gamma |y_A(2)| |y_A(3)|^{n-1} y_A(3) - \beta y_A(2) |y_A(3)|^n \end{Bmatrix} \right\} \quad (24)
 \end{aligned}$$

When considering state events, the *modes* and *domain* are illustrated in Fig. 13:

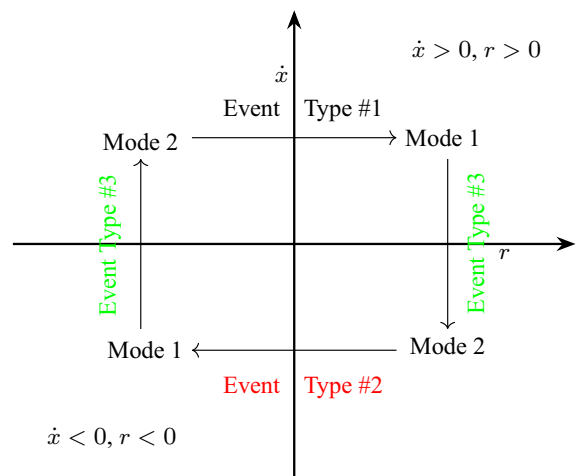


Fig. 13 Modes and domains for the Bouc–Wen model to be associated with Eq. (26)

Equation (21) indicates that only p and r are needed as state variables. Assuming $y(1) = p$ and $y(2) = r$, the state event associated with r can then be taken care of easily, whereas the state event associated with \dot{x} would meet a challenge by utilizing Eq. (9):

$$\dot{x} = 0 \implies \frac{1}{m} \int (u - r) dt = 0 \implies y(1) = \int u dt$$

This, however, goes beyond the allowed forms of event functions in MATLAB’s state event location algorithm according to [27]. Having said this, \dot{x} will be included as a state variable instead. The last state variable x is simply added to obtain the displacement. The proposed revision is given as follows, an option of which is denoted as “B” herein.

The *flow map* under Option B is given as follows:

$$y_B = \begin{Bmatrix} y_B(1) \\ y_B(2) \\ y_B(3) \end{Bmatrix} = \begin{Bmatrix} x \\ \dot{x} \\ r \end{Bmatrix}, \quad \dot{y}_B = \begin{Bmatrix} \dot{y}_B(1) \\ \dot{y}_B(2) \\ \dot{y}_B(3) \end{Bmatrix} = \begin{Bmatrix} \dot{x} \\ \ddot{x} \\ \dot{r} \end{Bmatrix}$$

$$= \begin{Bmatrix} y_B(2) \\ \frac{1}{m}(u(t) - y_B(3)) \\ A y_B(2) - \gamma \text{tag}_2 y_B(2) |y_B(3)|^{n-1} y_B(3) - \beta y_B(2) |y_B(3)|^n \end{Bmatrix} \quad (25)$$

with the *events*, *event functions* and *reset map* given as follows:

$$\begin{cases} \text{Event Type \#1 : } y_B(3) = 0 \text{ when } y_B(3) \text{ is ascending} & \text{tag}_3 = 1 \\ \text{Event Type \#2 : } y_B(3) = 0 \text{ when } y_B(3) \text{ is descending} & \text{tag}_3 = -1 \\ \text{Event Type \#3 : } y_B(2) = 0 \text{ when } y_B(2) \text{ is either ascending or descending} & \text{tag}_2 = -\text{tag}_2 \end{cases} \quad (26)$$

One drawback in automation is to assign initial values to tag_2 and tag_3 . Note that tag_1 is used for correcting discontinuity sticking, which will not be used for state variables.

To achieve a better understanding of the importance of all state events, we intentionally ignore the first two restoring force events. When we do so, we have the following option denoted as “C”:

$$y_C = \begin{Bmatrix} y_C(1) \\ y_C(2) \\ y_C(3) \end{Bmatrix} = \begin{Bmatrix} x \\ \dot{x} \\ r \end{Bmatrix},$$

$$\dot{y}_C = \begin{Bmatrix} \dot{y}_C(1) \\ \dot{y}_C(2) \\ \dot{y}_C(3) \end{Bmatrix} = \begin{Bmatrix} \dot{x} \\ \ddot{x} \\ \dot{r} \end{Bmatrix}$$

$$= \begin{Bmatrix} y_C(2) \\ \frac{1}{m}(u(t) - y_C(3)) \\ A y_C(2) - \gamma \text{tag}_2 y_C(2) |y_C(3)|^{n-1} y_C(3) - \beta y_C(2) |y_C(3)|^n \end{Bmatrix} \quad (27)$$

with an event function as follows:

$$y_C(2) = 0 \quad (28)$$

when $y_C(2)$ is either ascending or descending, belonging to the third type of state event, and $\text{tag}_2 = -\text{tag}_2$.

4.3 Bouc–Wen model: results and comparisons

Figures 14 and 15 present a set of simulation results under Option B. The same amplitude-modulated sine excitation was used. All three types of state events were detected.

A comparison is made among three options: A. no state events considered at all, B. all three types of state events considered, and C. only one out of three types of state events considered; the results are presented in Fig. 16 using tolerance proportionality and work–accuracy diagrams. It can be seen that, first, the utilization of state event location algorithm does improve computational accuracy and reduce computational effort measured by both number of function evaluation and computational time—when compared to omitting the state event location algorithm. This is the very motivation of our work. It can be seen that,

next, when not all types of state events are considered, the TP and work–accuracy diagrams may not change obviously from those—except with small GE values—when all types of state events are correctly considered. This indicates the importance of having a good understanding and doing a thorough job when implementing state event location algorithm because mistakes made may not be easy to detect.

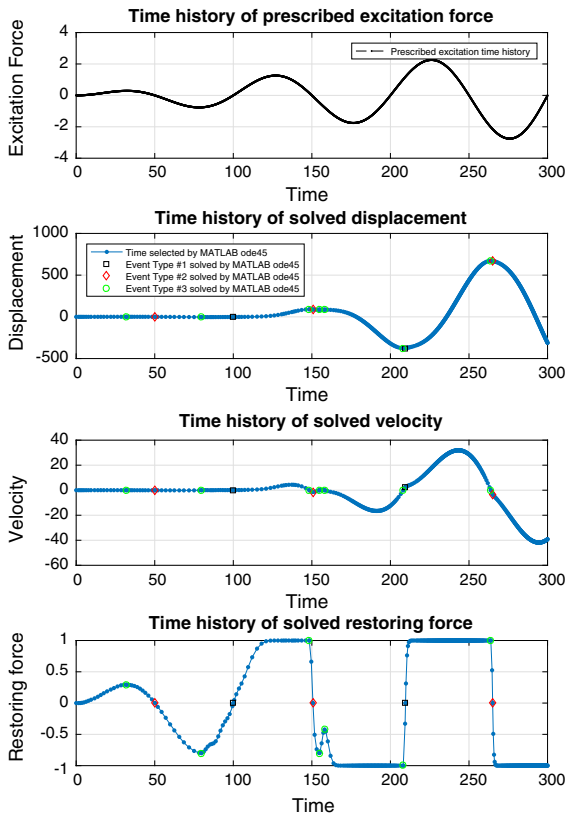


Fig. 14 Bouc–Wen (BW) model with $A = 0.5$, $\gamma = 0.25$, $\beta = 0.25$ and $n = 1$ subject to identical amplitude-modulated sine and simulated under Option B using the “event option” under MATLAB ode45, where $\text{AbsTol} = 10^{-12}$ and $\text{RelTol} = 10^{-3}$: time histories. Event Types #1 to 3 are defined in Eq. (26)

4.4 Bouc–Wen–Baber–Noori Model

The Bouc–Wen–Baber–Noori (BWBN) model has been popular. There are slightly different versions used and here we follow that in [10], where the model is considered to be a parallel connection between a linear spring with stiffness αk and a hysteresis model. The total restoring force is thus a sum of the linear spring restoring force αkx and the nonlinear restoring force of the hysteresis model, the latter of which is scaled by using the coefficient $(1 - \alpha)k$, as shown in Eq. (29).

$$r = \alpha kx + (1 - \alpha)kz \tag{29}$$

where α and k are model parameters with fixed values. z is a state variable having the same physical unit as the displacement:

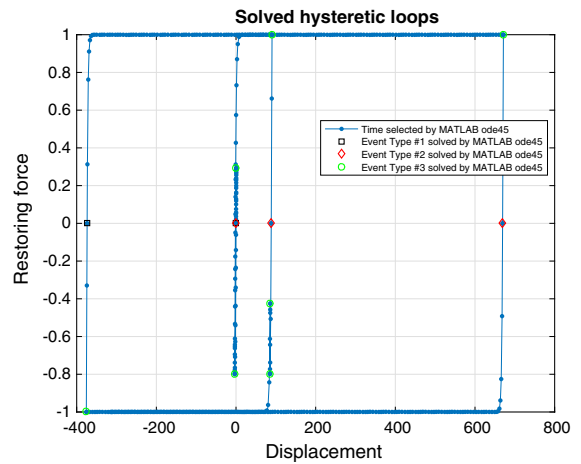


Fig. 15 Bouc–Wen (BW) model with $A = 0.5$, $\gamma = 0.25$, $\beta = 0.25$ and $n = 1$ subject to the amplitude-modulated sine excitation and simulated under Option B using the “event option” under MATLAB ode45, where $\text{AbsTol} = 10^{-12}$ and $\text{RelTol} = 10^{-3}$: hysteresis loops. Event Types #1 to 3 are defined in Eq. (26)

$$\dot{z} = h(z) \left\{ \frac{\dot{x} - v (\beta |\dot{x}| |z|^{n-1} z + \gamma \dot{x} |z|^n)}{\eta} \right\} \tag{30}$$

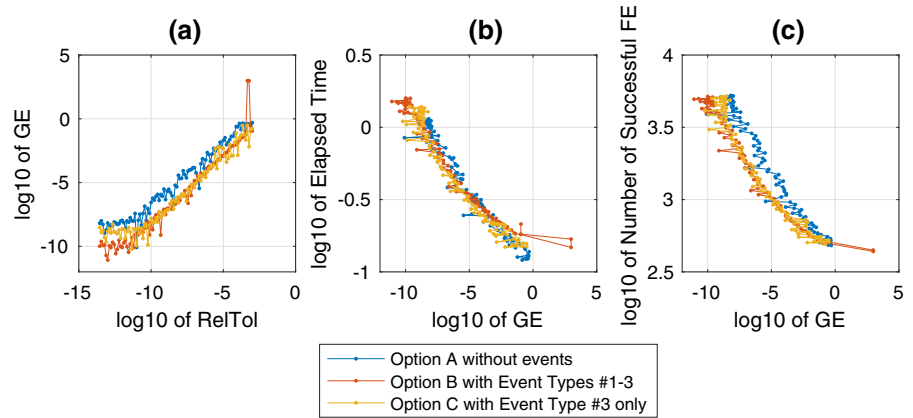
where the quantities h , η and v are time varying and defined further using equations to be incorporated later on, while β , n and γ are model parameters with fixed values. It should be noted that, for historical reasons, the β and γ parameters in this subsection, starting at Eq. (30), are reversed from those of the Bouc–Wen model in Sect. 4.1, starting at Eq. (20).

Connecting a BWBN model to a mass to form a SDOF system, an elegant choice is to formulate the equations as differential–algebraic equations (DAEs) specified below. Equation (31) is a state equation, while Eq. (32) is a coupled algebraic equation—both are in vector form.

$$\mathbf{y} = \begin{Bmatrix} y(1) \\ y(2) \\ y(3) \\ y(4) \end{Bmatrix} = \begin{Bmatrix} x \\ \dot{x} \\ z \\ \varepsilon \end{Bmatrix}, \quad \dot{\mathbf{y}} = \begin{Bmatrix} \dot{y}(1) \\ \dot{y}(2) \\ \dot{y}(3) \\ \dot{y}(4) \end{Bmatrix} = \begin{Bmatrix} \dot{x} \\ \ddot{x} \\ \dot{z} \\ \dot{\varepsilon} \end{Bmatrix}$$

$$= \begin{Bmatrix} y(2) \\ \frac{1}{m}(u(t) - w(1)) \\ \frac{w(7)}{w(2)} [y(2) - w(3) (\beta \text{tag}_2 y(2)(\text{tag}_3 y(3))^{n-1} y(3) + \gamma y(2)(\text{tag}_3 y(3))^n)] \\ (1 - \alpha)ky(2)y(3) \end{Bmatrix} \tag{31}$$

Fig. 16 Bouc–Wen (BW) model subject to identical amplitude-modulated sine excitation and simulated under Options A–C in terms of tolerance proportionality (a) and work–accuracy diagrams (b, c), where GE and FE stand for global error and function evaluation, respectively. Only time histories under Option B are presented in Fig. 14



$$\mathbf{w} = \begin{Bmatrix} w(1) \\ w(2) \\ w(3) \\ w(4) \\ w(5) \\ w(6) \\ w(7) \end{Bmatrix} = \begin{Bmatrix} r \\ \eta \\ v \\ z_u \\ \zeta_1 \\ \zeta_2 \\ h \end{Bmatrix} = \begin{Bmatrix} \alpha kx + (1 - \alpha)kz \\ 1 + \delta_\eta \varepsilon \\ 1 + \delta_v \varepsilon \\ \left(\frac{1}{v(\beta + \gamma)}\right)^{\frac{1}{n}} \\ \zeta_{10} (1 - e^{-p\varepsilon}) \\ (\Psi_0 + \delta_\Psi \varepsilon) (\lambda + \zeta_1) \\ \frac{-[\text{sgn}(\dot{x}) - qz_u]^2}{\zeta_2^2} \\ 1 - \zeta_1 e \end{Bmatrix} \tag{32}$$

$$= \begin{Bmatrix} \alpha ky(1) + (1 - \alpha)ky(3) \\ 1 + \delta_\eta y(4) \\ 1 + \delta_v y(4) \\ \left(\frac{1}{w(3)(\beta + \gamma)}\right)^{\frac{1}{n}} \\ \zeta_{10} (1 - e^{-py(4)}) \\ (\Psi_0 + \delta_\Psi y(4)) (\lambda + w(5)) \\ \frac{-[y(3)\text{tag}_2 - qw(4)]^2}{w(6)^2} \\ 1 - w(5)e \end{Bmatrix}$$

where ε stands for hysteretic energy, defined as $\varepsilon(t) = (1 - \alpha)k \int_0^t z(\tau)\dot{x}(\tau)d\tau$; r stands for restoring force; η and v are time-varying stiffness and strength degradation parameters, respectively, and h is a pinching function, which requires time-varying quantities z_u , ζ_1 and ζ_2 to define it. Overall, α , k , β , n , γ , δ_η , δ_v , q , ζ_{10} , p , Ψ_0 , δ_Ψ , λ are system parameters.

In terms of implementing this model under MATLAB ode45 solver, all codes for the BW model under Option “B” can be adopted as is, except for two updates: First, Eq. (31) is keyed into myfun.m after Eq. (32). Discontinuity sticking is corrected as verified. Next, following the recommendation in MATLAB, it is of critical importance to put the algebraic variable \mathbf{w} at the end of the argument lists of the four m-files as follows:

```

[t,y,te,ye] = ode45(@myfun,[tstart,tfinal],y0,
options,w)
[value, isterminal, direction] = Events(t,y,w)
ydot = myfun(t,y,w)
status = OutputFcn(t,y,flag,w)

```

Figures 17 and 18 present a set of simulation results using the system parameter values in Fig. 10(b) of [10] except for k . Their performance measures are presented in Figs. 19 and 20. Defining MaxStep becomes essential in this numerical example.

5 Discussion

5.1 Guidelines

Some guidelines concerning choices of state variables are summarized here and will be further validated in future work:

1. To apply Paynter’s tetrahedron of state using energy/power variables, we have the following two chains of kinematic and kinetic variables aligned with each other:

$$\begin{aligned}
 &\text{acceleration, } \ddot{x} \xrightarrow{\int dt} \text{velocity,} \\
 &\dot{x} \xrightarrow{\int dt} \text{displacement, } x \xrightarrow{\int dt} \text{absement, } a \tag{33} \\
 &\text{rate of change in restoring force,} \\
 &\dot{r} \xrightarrow{\int dt} \text{restoring force, } r \xrightarrow{\int dt} \text{g-momentum, } p \\
 &\xrightarrow{\int dt} \text{time integral of g-momentum, } \rho \tag{34}
 \end{aligned}$$

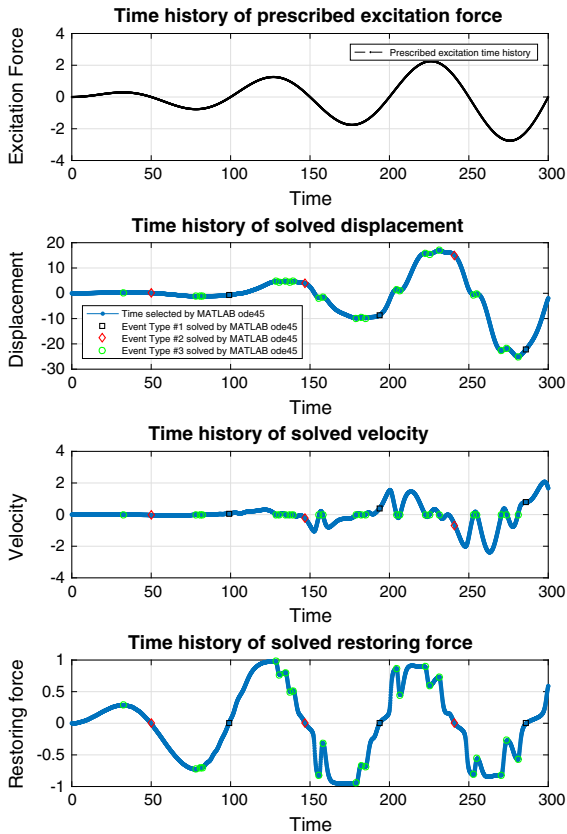


Fig. 17 Bouc–Wen–Baber–Noori (BWBN) models with $\alpha = 0.1$, $k = 1$, $\beta = 1.5$, $n = 1$, $\gamma = -0.5$, $\delta_\eta = 0.05$, $\delta_v = 0.005$, $q = 0.1$, $\zeta_{10} = 0.97$, $p = 1$, $\Psi_0 = 0.2$, $\delta_\psi = 0.002$ and $\lambda = 0.1$ subject to an amplitude-modulated sine and simulated using the “event option” under MATLAB ode45, where AbsTol = 10^{-12} , RelTol = 10^{-3} and MaxStep = 10^{-1} : time histories. Event Types #1 to 3 are defined in Eq. (26)

where absence a and time integral of g -momentum ρ are not used in this study. Subsequently, the equation of motion can be integrated step-by-step as follows:

$$m\ddot{x} = \dot{u} - \dot{r} \xrightarrow{\int dt} m\dot{x} = u - r \xrightarrow{\int dt} m\dot{x} = \int u dt - p \xrightarrow{\int dt} mx = \int \int u dt dt - \rho \xrightarrow{\int dt} ma = \int \int \int u dt dt dt - \int \rho dt \quad (35)$$

We make the following additional comments:

- The variable with the lowest order time derivative in the system dynamics that we care about in modeling should be included in the state variable vector.
- The variable with the highest order time derivative in the system dynamics that we care about

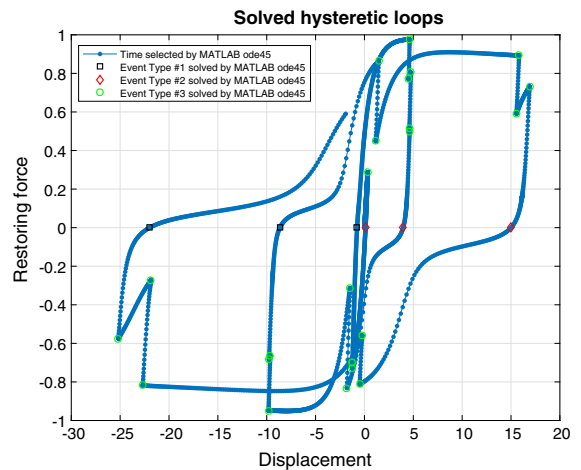


Fig. 18 Bouc–Wen–Baber–Noori (BWBN) models with $\alpha = 0.1$, $k = 1$, $\beta = 1.5$, $n = 1$, $\gamma = -0.5$, $\delta_\eta = 0.05$, $\delta_v = 0.005$, $q = 0.1$, $\zeta_{10} = 0.97$, $p = 1$, $\Psi_0 = 0.2$, $\delta_\psi = 0.002$ and $\lambda = 0.1$ subject to an amplitude-modulated sine and simulated using the “event option” under MATLAB ode45, where AbsTol = 10^{-12} , RelTol = 10^{-3} and MaxStep = 10^{-1} : hysteresis loops. Event Types #1 to 3 are defined in Eq. (26)

in modeling should be included in the state variable derivative vector.

- When we utilize the state event location algorithm, we need to make the variable that defines the event function a state variable—not a time derivative of a state variable. This means that we cannot utilize the variable with the highest order in the system dynamics to define an event function.
- Whenever \dot{x} is a state variable, then p does not need to be a state variable, and vice versa. This comes from Eq. (9) which shows that \dot{x} and p are not independent state variables, based on Newton’s second law in Eq. (1).

2. It is important to reiterate the need to effectively address discontinuities in a model, whenever applicable. Velocity \dot{x} is often required for the state event location for piecewise-defined loading and unloading branches in the $r-x$ plane. Given Eq. (9), whenever there is only one mass connected in series with the rest of the assembly, \dot{x} and p cannot coexist in

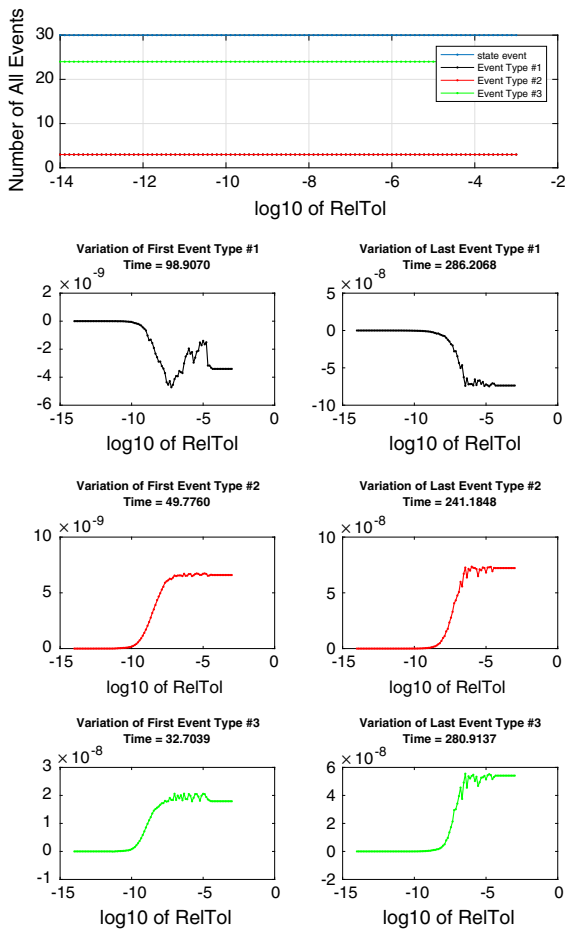


Fig. 19 Number of state events and the event time errors plotted against MATLAB ode45 RelTol for the BWBN model in Fig. 18, where there are three, three and 24 Types #1, 2 and 3 events throughout, respectively, to make a total of 30 events throughout. Event Types #1 to 3 are defined in Eq. (26)

the state variable vector for the state equations to be independent, so p will be replaced with \dot{x} and thus will not appear in the state variable vector in models with discontinuity. This explains, to some extent, why p is not popular for the choice of state variables.

3. The force-state mapping comes in handy for flow-controlled systems to map some kinematic quantities to some kinetic ones. When there is a need for the inverse map (kinetic quantities to kinematic ones) as in effort-controlled systems, it would not be rational if the force-state mapping technique is still adopted. Similarly, when dealing with mixed systems, mapping kinematic and kinetic quantities into each other, it may not be proper to use the

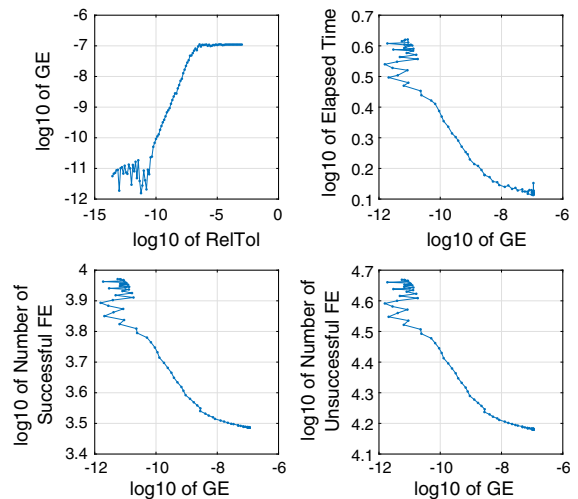


Fig. 20 Tolerance proportionality (upper left) and work-accuracy diagrams (the rest) for the BWBN model in Fig. 18, where GE and FE stand for global error and function evaluation, respectively

force-state mapping alone due to the mismatch of the nature of the inputs and outputs.

4. Conversions of state variables can be carried out as follows:

$$\begin{bmatrix} x \\ \dot{x} \end{bmatrix} \xrightarrow{\text{Eq. (9)}} \begin{bmatrix} x \\ p \end{bmatrix} \xrightarrow{\frac{d}{dt}} \begin{bmatrix} \dot{x} \\ r \end{bmatrix} \xrightarrow{\text{Eq. (9)}} \begin{bmatrix} p \\ r \end{bmatrix} \quad (36)$$

implies that if \dot{x} and r can serve as state variables, then the flow-controlled system can be considered effort-controlled. To do so, as indicated in the equation above, the rate of change seems essential for this system, given the use of time differentiation for state variables.

$$\begin{bmatrix} p \\ r \end{bmatrix} \xrightarrow{\text{Eq. (9)}} \begin{bmatrix} \dot{x} \\ r \end{bmatrix} \xrightarrow{\int dt} \begin{bmatrix} x \\ p \end{bmatrix} \xrightarrow{\text{Eq. (9)}} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \quad (37)$$

implies that if x and p can serve as state variables, then the effort-controlled system can be considered flow-controlled. To do so, as indicated in the equation above, a memory effect seems essential for this system, given the use of time integral for state variables.

5. If an algebraic variable is needed to define an event function, then the algebraic variable may be implemented via a global variable under MATLAB. All algebraic variables should be made into an algebraic vector that is carried over through all MAT-

LAB m-files as demonstrated in Sect. 4.4 for the BWBN model.

5.2 The $\dot{r} = h(\dot{x}, r)$ formulation

First, some review of the history is given here: [1,2] especially [3] using the Bouc–Wen and a “slip-lock” model to capture pinching hysteresis are frequently cited for the origin of this formulation. In fact, the theoretical justification cited in [3] is [5], which actually gives a more general formulation making \dot{r} a function of x, \dot{x}, r and a dynamic internal variable vector used to capture the path dependency of the model.

One major application of $\dot{r} = h(\dot{x}, r)$ is the Bouc–Wen model. A review paper [17] on system identification for nonlinear systems introduces this formulation for the Bouc–Wen model. In this study, we showed that the Bouc–Wen model is an effort-controlled model with p and r as its state variables, but it has two types of discontinuities, leading to selecting \dot{x} as an additional state variable. Given the dependence between \dot{x} and p shown in Eq. (9), the necessary state variables for the Bouc–Wen model then become \dot{x} and r . Consequently, the governing state equation is in the form of $\dot{r} = h(\dot{x}, r)$. This is one rationale behind the $\dot{r} = h(\dot{x}, r)$ formulation of the Bouc–Wen model. Another explanation follows from bond graph theory, which says to use x and p . Instead, we differentiate them once and use \dot{x} and r as state variables.

A more general application of $\dot{r} = h(\dot{x}, r)$ may be deduced as follows using the pair of flow- and effort-controlled formulations:

$$\text{Force-state mapping } r = f(x, \dot{x}) \tag{38}$$

$$\text{Formulation under discussion } \dot{r} = h(\dot{x}, r) \tag{39}$$

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \mathbf{f} \left(\begin{bmatrix} x \\ \dot{x} \end{bmatrix}, t \right), \text{ flow-controlled} \tag{40}$$

$$\begin{bmatrix} r \\ \dot{r} \end{bmatrix} = \mathbf{g} \left(\begin{bmatrix} p \\ r \end{bmatrix}, t \right), \text{ effort-controlled} \tag{41}$$

By using Eq. (1), it can be seen that Eq. (38) is consistent with the flow-controlled formulation Eq. (40). By using Eq. (9), it can be seen that Eq. (39) is consistent with Eq. (41). Equation (21) is such an example.

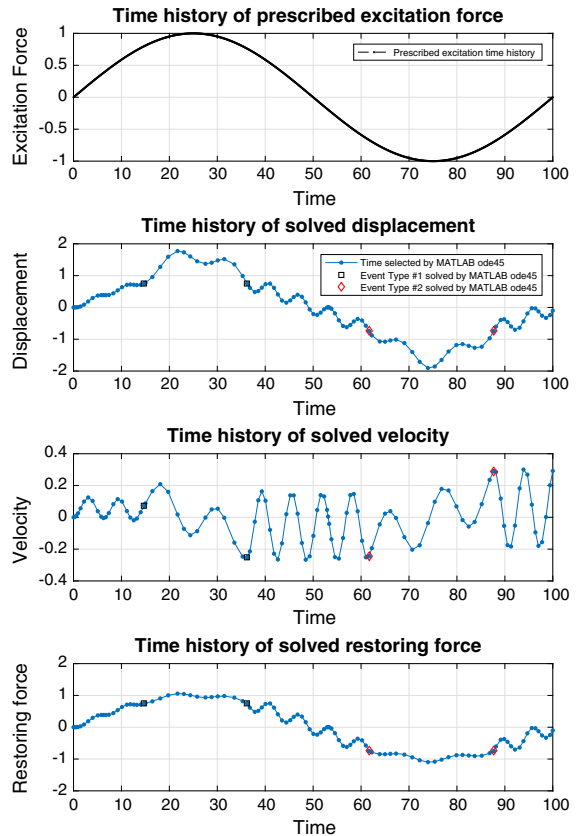


Fig. 21 To reveal the insight to the peculiar point in Fig. 3 and as a counterpart for Fig. 2; simulation used sinusoidal excitation force $u(t)$ shown at top and “event option” under MATLAB ode45 with RelTol = 0.00077426 and AbsTol = 10^{-12} . Event Types #1 and 2 are defined in Eqs. (2) and (3), respectively

5.3 Complications with state event location algorithm

Examination of Fig. 3 shows that a particular value of RelTol caused the number of state events to reduce by two. For this RelTol value (0.00077426), the time histories are presented in Fig. 21 showing the loss of two Type #1 events. One remedy is to introduce MaxStep = 0.1. As Fig. 3 shows, this loss of two events did not happen with other selected values of RelTol, larger or smaller. To be studied in future work, this is an example of a numerical difficulty analogous to grazing bifurcation, mentioned in [34].

5.4 Physically consistent models

Nonlinear constitutive models are notorious for allowing physically inconsistent solutions to dynamic prob-

lems, with nonuniqueness being just one of the underlying issues [26]. Various authors have addressed physical inconsistencies in the Bouc–Wen family of models; e.g., [12–14, 18, 28]. Before embarking on computations, with or without state event location techniques, it is crucial to start with a well-posed model since otherwise the resulting simulations will have no practical predictive value.

6 Conclusions

The need for a sufficient and efficient problem formulation of a system’s dynamics is of critical importance in many engineering mechanics applications, e.g., system identification, damage detection and earthquake response simulation, among many others. Two characteristics of some commonly used models have not been researched thoroughly: One is their flow-/effort-controlled nature and the other, discontinuity in their state questions.

These two characteristics are coupled in numerical simulation of these models. Normally, one would decide the state variables based on the flow-/effort-controlled nature, when algebraic variables might exist as well. The “switching rules” and “switching timing” (i.e., the state event locations) can be decided efficiently when we properly treat the discontinuity in the simulation. Without a good understanding of these two characteristics, inefficiency, inaccuracy and incorrectness in problem formulation of a system’s dynamics and model class building can be experienced. This is the motivation for us to study these two characteristics. In particular, the proper treatment of discontinuities has been explored by us starting with [34]. We also explored the flow-/effort-controlled nature of some models in [24, 25].

Choices for state variables are not unique as demonstrated in this study. This is not often made clear in applying state-space representations. Computationally, some choices are better than others in order to implement the state event location algorithm. There are two questions of particular interest to us:

- How can we select state variables that are not only theoretically sound but also computationally fruitful? Herein, a successful application of the state event location algorithm and the ode45 solver has been presented for some challenging piecewise-defined restoring force models.
- Does the choice of state variables for these models imply a classification of these models, or should it be vice versa?

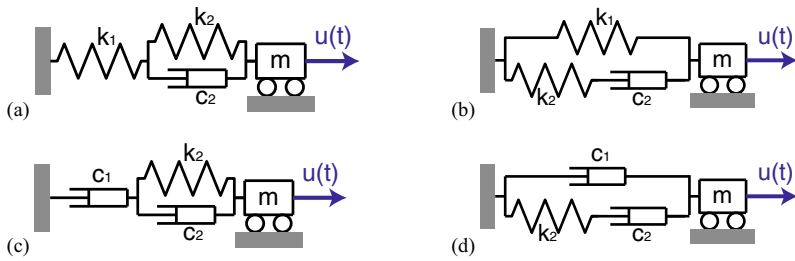
There is no simple or one-size-fits-all answer to these fundamental questions. Different facets of these questions have been demonstrated through case studies provided in this work and general guidelines are also provided. The Bouc–Wen model and other models discussed herein (and elsewhere) can be implemented without the state event location algorithm by choosing the algorithmic time steps to be sufficiently small. A key conclusion from our work is that the option of implementing the state event location algorithm is beneficial in reducing the computational burden while providing the same level of accuracy. In the spirit of reproducible computational research [19], the MATLAB code used in this work will be available upon request to the first author.

Acknowledgements Dr. Pei would like to acknowledge the hospitality of California Institute of Technology during her sabbatical leave for the completion of this study. Dr. Gay-Balmaz is partially supported by the ANR project GEOMFLUID, ANR-14-CE23-0002-01.

A Appendix for Sect. 2.2

Tables 2 and 3 list nonunique state-space representations of three- and four-parameter linear models, respectively.

Table 2 State variables for the three-element models each connected with a mass



(a) Standard solid model #1 connected in series with a mass

$$y = \begin{bmatrix} x \\ \dot{x} \\ r \end{bmatrix}, \quad \dot{y} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{m} \\ \frac{k_1 k_2}{c_2} & k_1 & \frac{k_1 + k_2}{c_2} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ r \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \\ 0 \end{bmatrix} u$$

Alternatively by replacing \dot{x} with p equivalently

$$y = \begin{bmatrix} x \\ p \\ r \end{bmatrix}, \quad \dot{y} = \begin{bmatrix} \dot{x} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{m} & 0 \\ 0 & 0 & 1 \\ \frac{k_1 k_2}{c_2} & -\frac{k_1}{m} & \frac{k_1 + k_2}{c_2} \end{bmatrix} \begin{bmatrix} x \\ p \\ r \end{bmatrix} + \begin{bmatrix} \frac{1}{m} \\ 0 \\ \frac{k_1}{m} \end{bmatrix} f u dt$$

(b) Standard solid model #2 connected in series with a mass

$$y = \begin{bmatrix} x \\ \dot{x} \\ r \end{bmatrix}, \quad \dot{y} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{m} \\ \frac{k_1 k_2}{c_2} & k_1 + k_2 & -\frac{k_2}{c_2} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ r \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \\ 0 \end{bmatrix} u$$

Alternatively by replacing \dot{x} with p equivalently

$$y = \begin{bmatrix} x \\ p \\ r \end{bmatrix}, \quad \dot{y} = \begin{bmatrix} \dot{x} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{m} & 0 \\ 0 & 0 & 1 \\ \frac{k_1 k_2}{c_2} & -\frac{k_1 + k_2}{m} & -\frac{k_2}{c_2} \end{bmatrix} \begin{bmatrix} x \\ p \\ r \end{bmatrix} + \begin{bmatrix} \frac{1}{m} \\ 0 \\ \frac{k_1 + k_2}{m} \end{bmatrix} f u dt$$

(c) Standard fluid model #1 connected in series with a mass

$$y = \begin{bmatrix} r \\ \dot{x} \end{bmatrix}, \quad \dot{y} = \begin{bmatrix} \dot{r} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} -\frac{k_2}{c_1 + c_2} & -\frac{1}{m} \frac{c_1 c_2}{c_1 + c_2} & \frac{k_2 c_1}{c_1 + c_2} \\ -\frac{1}{m} & 0 & 0 \end{bmatrix} \begin{bmatrix} r \\ \dot{x} \end{bmatrix} + \begin{bmatrix} \frac{1}{m} \frac{c_1 c_2}{c_1 + c_2} \\ \frac{1}{m} \end{bmatrix} u$$

Alternatively by lowering the order of both y and \dot{y}

$$y = \begin{bmatrix} p \\ x \end{bmatrix}, \quad \dot{y} = \begin{bmatrix} \dot{r} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -\frac{k_2}{c_1 + c_2} & -\frac{1}{m} \frac{c_1 c_2}{c_1 + c_2} & \frac{k_2 c_1}{c_1 + c_2} \\ -\frac{1}{m} & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ x \end{bmatrix} + \begin{bmatrix} \frac{1}{m} \frac{c_1 c_2}{c_1 + c_2} \\ \frac{1}{m} \end{bmatrix} f u dt$$

Alternatively in an entirely effort-controlled form

$$y = \begin{bmatrix} p \\ r \end{bmatrix}, \quad \dot{y} = \begin{bmatrix} \dot{r} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k_2}{m} \frac{c_1}{c_1 + c_2} & -\frac{k_2}{c_1 + c_2} - \frac{1}{m} \frac{c_1 c_2}{c_1 + c_2} \end{bmatrix} \begin{bmatrix} p \\ r \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{k_2}{m} \frac{c_1}{c_1 + c_2} f u dt + \frac{1}{m} \frac{c_1 c_2}{c_1 + c_2} u \end{bmatrix}$$

Alternatively in an entirely flow-controlled form

$$y = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}, \quad \dot{y} = \begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k_2}{m} \frac{c_1}{c_1 + c_2} & -\left(\frac{k_2}{c_1 + c_2} + \frac{1}{m} \frac{c_1 c_2}{c_1 + c_2}\right) \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{k_2}{m(c_1 + c_2)} f u dt + \frac{u}{m} \end{bmatrix}$$

(d) Standard fluid model #2 connected in series with a mass

$$y = \begin{bmatrix} r \\ \dot{x} \end{bmatrix}, \quad \dot{y} = \begin{bmatrix} \dot{r} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} -\frac{k_2}{c_2} - \frac{c_1}{m} \frac{k_2}{c_2} (c_1 + c_2) & 0 \\ -\frac{1}{m} & 0 \end{bmatrix} \begin{bmatrix} r \\ \dot{x} \end{bmatrix} + \begin{bmatrix} \frac{c_1}{m} \\ \frac{1}{m} \end{bmatrix} u$$

Alternatively by lowering the order of both y and \dot{y}

$$y = \begin{bmatrix} p \\ x \end{bmatrix}, \quad \dot{y} = \begin{bmatrix} \dot{r} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -\frac{k_2}{c_2} - \frac{c_1}{m} \frac{k_2}{c_2} (c_1 + c_2) & 0 \\ -\frac{1}{m} & 0 \end{bmatrix} \begin{bmatrix} p \\ x \end{bmatrix} + \begin{bmatrix} \frac{c_1}{m} \\ \frac{1}{m} \end{bmatrix} f u dt$$

Table 2 continued

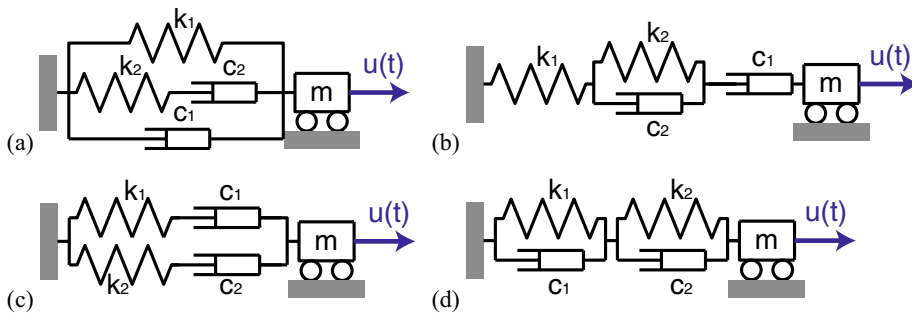
Alternatively in an entirely effort-controlled form

$$y = \begin{bmatrix} p \\ r \end{bmatrix}, \quad \dot{y} = \begin{bmatrix} \dot{r} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k_2}{m} \frac{c_1+c_2}{c_2} - \frac{k_2}{c_2} - \frac{c_1}{m} \end{bmatrix} \begin{bmatrix} p \\ r \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{k_2}{m} \frac{c_1+c_2}{c_2} \int u dt + \frac{c_1}{m} u \end{bmatrix}$$

Alternatively in an entirely flow-controlled form

$$y = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}, \quad \dot{y} = \begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k_2}{m} \frac{c_1+c_2}{c_1} - \left(\frac{k_2}{c_2} + \frac{c_1}{m}\right) \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{k_2}{m c_2} \int u dt + \frac{u}{m} \end{bmatrix}$$

Table 3 State variables for the four-element models each connected with a mass



(a) A generalized Maxwell model connected in series with a mass

$$y = \begin{bmatrix} x \\ \dot{x} \\ r \end{bmatrix}, \quad \dot{y} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{m} \\ \frac{k_1 k_2}{c_2} & k_2 \left(1 + \frac{k_1}{k_2} + \frac{c_1}{c_2}\right) & -\frac{k_2}{c_1} - \frac{c_1}{m} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ r \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \\ \frac{c_1}{m} \end{bmatrix} u$$

Alternatively by replacing \dot{x} with p equivalently

$$y = \begin{bmatrix} x \\ p \\ r \end{bmatrix}, \quad \dot{y} = \begin{bmatrix} \dot{x} \\ r \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{m} & 0 \\ 0 & 0 & 1 \\ \frac{k_1 k_2}{c_2} & -\frac{k_2}{m} \left(1 + \frac{k_1}{k_2} + \frac{c_1}{c_2}\right) & -\frac{k_2}{c_1} - \frac{c_1}{m} \end{bmatrix} \begin{bmatrix} x \\ p \\ r \end{bmatrix} + \begin{bmatrix} \frac{1}{m} \\ 0 \\ \frac{c_1}{m} u + \frac{k_2}{m} \left(1 + \frac{k_1}{k_2} + \frac{c_1}{c_2}\right) \int u dt \end{bmatrix}$$

(b) A generalized Kelvin chain connected in series with a mass

$$y = \begin{bmatrix} x \\ \dot{x} \\ r \end{bmatrix}, \quad \dot{y} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{m} \\ \frac{k_1 k_2}{c_2} & k_1 + m \frac{k_1 k_2}{c_1 c_2} & \frac{k_1}{c_2} \left(1 + \frac{k_2}{k_1} + \frac{c_2}{c_1}\right) \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ r \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \\ -\frac{k_1 k_2}{c_1 c_2} \int u dt \end{bmatrix}$$

Alternatively by replacing \dot{x} with p equivalently

$$y = \begin{bmatrix} x \\ p \\ r \end{bmatrix}, \quad \dot{y} = \begin{bmatrix} \dot{x} \\ r \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{m} & 0 \\ 0 & 0 & 1 \\ \frac{k_1 k_2}{c_2} & -\frac{k_1}{m} - \frac{k_1 k_2}{c_1 c_2} & \frac{k_1}{c_2} \left(1 + \frac{k_2}{k_1} + \frac{c_2}{c_1}\right) \end{bmatrix} \begin{bmatrix} x \\ p \\ r \end{bmatrix} + \begin{bmatrix} \frac{1}{m} \\ 0 \\ \frac{k_1}{m} \end{bmatrix} \int u dt$$

(c) Two Maxwell models connected in parallel—before being connected in series with a mass

$$y = \begin{bmatrix} x \\ \dot{x} \\ r \end{bmatrix}, \quad \dot{y} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{m} \\ k_1 k_2 \left(\frac{1}{c_1} + \frac{1}{c_2}\right) & k_1 + k_2 - m \frac{k_1 k_2}{c_1 c_2} & -\frac{k_1}{c_1} - \frac{k_2}{c_2} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ r \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \\ \frac{k_1 k_2}{c_1 c_2} \int u dt \end{bmatrix}$$

Table 3 continued

Alternatively by replacing \dot{x} with p equivalently

$$y = \begin{bmatrix} x \\ p \\ r \end{bmatrix}, \quad \dot{y} = \begin{bmatrix} \dot{x} \\ r \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{m} & 0 \\ 0 & 0 & 1 \\ k_1 k_2 \left(\frac{1}{c_1} + \frac{1}{c_2}\right) - \frac{k_1+k_2}{m} & -\frac{k_1 k_2}{c_1 c_2} & -\frac{k_1}{c_1} - \frac{k_2}{c_2} \end{bmatrix} \begin{bmatrix} x \\ p \\ r \end{bmatrix} + \begin{bmatrix} \frac{1}{m} \\ 0 \\ \frac{k_1+k_2}{m} \end{bmatrix} \int u dt$$

(d) Two Kelvin models connected in series—before being connected in series with a mass

$$y = \begin{bmatrix} x \\ \dot{x} \\ r \end{bmatrix}, \quad \dot{y} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{m} \\ \frac{k_1 k_2}{c_1+c_2} & \frac{k_1 c_2+k_2 c_1}{c_1+c_2} & -\frac{k_1+k_2}{c_1+c_2} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ r \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{m} \end{bmatrix} u$$

Alternatively by replacing \dot{x} with p equivalently

$$y = \begin{bmatrix} x \\ p \\ r \end{bmatrix}, \quad \dot{y} = \begin{bmatrix} \dot{x} \\ r \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{m} & 0 \\ 0 & 0 & 1 \\ \frac{k_1 k_2}{c_1+c_2} & -\frac{1}{m} \frac{k_1 c_2+k_2 c_1}{c_1+c_2} & -\frac{k_1+k_2}{c_1+c_2} \end{bmatrix} \begin{bmatrix} x \\ p \\ r \end{bmatrix} + \begin{bmatrix} \frac{1}{m} \\ 0 \\ \frac{1}{m} \frac{k_1 c_2+k_2 c_1}{c_1+c_2} \end{bmatrix} \int u dt$$

B Appendix for Sect. 3

For the F formulation (flow-controlled) and according to the programming framework in [34], our previous work, global variables are utilized for passing values of restoring force r , memory parameters H_l , H_u and O , and tags tag_1 and tag_2 among different MATLAB m-files. In particular, the memory parameters are obtained

and updated by tagging the entire solution history H in order to switch according to the correct logic and to correct discontinuity sticking. H_l , H_u and O are for the lower (Mode II), upper (Mode IV) bound values in the event functions as in Table 4, and a constant in the algebraic equations as in Table 5, respectively. tag_1 is used to mark each state event for correcting discontinuity sticking, while tag_2 is an indicator for the mode.

Table 4 Events, event functions and reset maps for F and M formulations

Transition	Event Type	Flow-controlled formulation (F)		Partially effort-controlled formulation (M)	
		Event function	Direction	Event function	Direction
① I → IV	1	$y_F(1) - H_u(\text{end}) = 0$	+ 1	$y_M(4) - (1 - \alpha)r_y = 0$	+ 1
② I → II	2	$y_F(1) - H_l(\text{end}) = 0$	- 1	$y_M(4) + (1 - \alpha)r_y = 0$	- 1
③ I → I	3	$y_F(2) = 0$	0	$y_M(2) = 0$	0
④ II → III	3	$y_F(2) = 0$	0 [†]	$y_M(2) = 0$	0
⑤ III → IV	1	$y_F(1) - H_u(\text{end}) = 0$	+ 1	$y_M(4) - (1 - \alpha)r_y = 0$	+ 1
⑥ III → II	2	$y_F(1) - H_l(\text{end}) = 0$	- 1	$y_M(4) + (1 - \alpha)r_y = 0$	- 1
⑦ III → III	3	$y_F(2) = 0$	0	$y_M(2) = 0$	0
⑧ IV → I	3	$y_F(2) = 0$	0 [†]	$y_M(2) = 0$	0

[†]Need to reset H_l , H_u and O

Table 5 Algebraic and state equations in flow maps obtained from piecewise restoring force expressions for F and M formulations, respectively

Mode	Flow-controlled (F)	Partially effort-controlled (M)
I	$r = k(y_F(1) - O(\text{end}))$, with $tag_2 = 1$	$\dot{y}_M(3) = (1 - tag_2)(1 - \alpha)k y_M(2) + \alpha k y_M(2)$, with $tag_2 = 0$
II	$r = -r_y + \alpha k(y_F(1) + x_y)$, with $tag_2 = 2$	$\dot{y}_M(3) = (1 - tag_2)(1 - \alpha)k y_M(2) + \alpha k y_M(2)$, with $tag_2 = 1$
III	$r = k(y_F(1) - O(\text{end}))$, with $tag_2 = 3$	$\dot{y}_M(3) = (1 - tag_2)(1 - \alpha)k y_M(2) + \alpha k y_M(2)$, with $tag_2 = 0$
IV	$r = r_y + \alpha k(y_F(1) - x_y)$, with $tag_2 = 4$	$\dot{y}_M(3) = (1 - tag_2)(1 - \alpha)k y_M(2) + \alpha k y_M(2)$, with $tag_2 = 1$

In terms of variables, $x(t)$, $\dot{x}(t)$ and $r(t)$ are continuous variables, while H_I , H_U , O , tag_1 and tag_2 are discrete variables.

For the M formulation (mixed, partially effort-controlled) and as in our previous work, a global variable is utilized for passing values of tag_2 among different MATLAB m-files. tag_2 is an indicator for the mode and is the only discrete variable to be taken care of.

References

1. Baber, T.T., Noori, M.N.: Random vibration of degrading, pinching systems. *ASCE J. Eng. Mech.* **111**(8), 1010–1026 (1985)
2. Baber, T.T., Noori, M.N.: Modeling general hysteresis behavior and random vibration application. *ASME J. Vib. Acoust. Stress Reliab. Des.* **108**, 411–420 (1986)
3. Benedettini, F., Capecchi, D., Vestroni, F.: Identification of hysteretic oscillators under earthquake loading by nonparametric models. *ASCE J. Eng. Mech.* **121**(5), 606–612 (1995)
4. Bouc, R.: Forced vibration of mechanical systems with hysteresis. In: *Proceedings of 4th Conference on Nonlinear Oscillations* (1967)
5. Capecchi, D.: Accurate solutions and stability criterion for periodic oscillations in hysteretic systems. *Meccanica* **25**, 159–167 (1990)
6. Caughey, T.K.: Random excitation of a system with bilinear hysteresis. *J. Appl. Mech.* **27**, 649–652 (1960a)
7. Caughey, T.K.: Sinusoidal excitation of a system with bilinear hysteresis. *J. Appl. Mech.* **27**, 640–643 (1960b)
8. Dormand, J.R., Prince, P.J.: A family of embedded Runge–Kutta formulae. *J. Comput. Appl. Math.* **6**(1), 19–26 (1980)
9. Esposito, J.M., Kumar, V.: A state event detection algorithm for numerically simulating hybrid systems with model singularities. *ACM Trans. Model. Comput. Simul.* **17**(1), 1–22 (2007)
10. Foliente, G.C.: Hysteresis modeling of wood joints and structural systems. *ASCE J. Struct. Eng.* **121**(6), 1013–1022 (1995)
11. Goebel, R., Sanfelice, R., Teel, A.: Hybrid dynamical systems. *IEEE Control Syst. Mag.* **28**, 28–93 (2009)
12. Ikhrouane, F., Rodellar, J.: Physical consistency of the hysteretic Bouc-Wen model. *IFAC Proc. Vol.* **38**(1), 874–879 (2005)
13. Jayakumar, P.: Modeling and identification in structural dynamics. Ph.D. thesis, California Institute of Technology, Pasadena, CA (1987)
14. Jayakumar, P., Beck, J.L.: System identification using nonlinear structural models. In: Nake, H.G., Yao, J.T.P. (eds.) *Structural Safety Evaluation Based on System Identification Approaches*, Friedr. Vieweg & Sohn Braunschweig/Wiesbaden, Vieweg International Scientific Book Series, vol. Proceedings of the Workshop at Lambrecht/Pfalz, pp. 82–102 (1988)
15. Kalmár-Nagy, T., Shekhawat, A.: Nonlinear dynamics of oscillators with bilinear hysteresis and sinusoidal excitation. *Phys. D* **238**, 1768–1786 (2009)
16. Karnopp, D.C., Margolis, D.L., Rosenberg, R.C.: *System Dynamics: Modeling, Simulation, and Control of Mechatronics Systems*, 5th edn. Wiley, Hoboken (2012)
17. Kerschen, G., Worden, K., Vakakis, A.F., Golinval, J.C.: Past, present and future of nonlinear system identification in structural dynamics. *Mech. Syst. Signal Process.* **20**, 505–592 (2006)
18. Kottari, A.K., Charalampakis, A.E., Koumousis, V.K.: A consistent degrading Bouc-Wen model. *Eng. Struct.* **60**, 235–240 (2014)
19. LeVeque, R.J.: Wave propagation software, computational science, and reproducible research. In: *Proceedings of the International Congress of Mathematicians, European Mathematical Society, Madrid, Spain*, pp. 1227–1253 (2006)
20. Oh, J., Bernstein, D.S.: Semilinear duhem model for rate-independent and rate-dependent hysteresis. *IEEE Trans. Autom. Control* **50**(5), 631–645 (2005)
21. Park, T., Barton, P.L.: State event location in differential-algebraic models. *ACM Trans. Model. Comput. Simul.* **6**(2), 147–165 (1996)
22. Paynter, H.M.: *Analysis and Design of Engineering Systems: Class Notes for M.I.T. Course 2.751*. M.I.T. Press, Cambridge (1961)
23. Paynter, H.M.: The gestation and birth of bond graphs. [https://www.me.utexas.edu/~longoria/paynter/hmp/Bond graphs.html](https://www.me.utexas.edu/~longoria/paynter/hmp/Bond%20graphs.html) (2000)
24. Pei, J.S., Wright, J.P., Todd, M.D., Masri, S.F., Gay-Balmaz, F.: Understanding memristors and memcapacitors for engineering mechanical applications. *Nonlinear Dyn.* **80**(1), 457–489 (2015)
25. Pei, J.S., Gay-Balmaz, F., Wright, J.P., Todd, M.D., Masri, S.F.: Dual input–output pairs for modeling hysteresis inspired by mem-models. *Nonlinear Dyn.* **88**(4), 2435–2455 (2017)
26. Sandler, I.S.: On the uniqueness and stability of endochronic theories of materials behavior. *ASME J. Appl. Mech.* **45**, 263–266 (1978)
27. Shampine, L., Gladwell, I., Brankin, R.: Reliable solution of special event location problems for ODEs. *ACM Trans. Math. Softw.* **17**(1), 11–25 (1991)
28. Sivaselvan, M.V., Reinhorn, A.M.: Hysteretic models for deteriorating inelastic structures. *ASCE J. Eng. Mech.* **126**(6), 633–640 (2000)
29. Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, R.S.: The missing memristor found. *Nature* **453**, 80–83 (2008)
30. van der Schaft, A., Schumacher, H.: *An Introduction to Hybrid Dynamical Systems*. Lecture Notes in Control and Information Sciences (Book 251). Springer, Berlin (1999)
31. Wen, Y.K.: Method for random vibration of hysteretic systems. *ASCE J. Eng. Mech.* **102**(2), 249–263 (1976)
32. Wen, Y.K.: Equivalent linearization for hysteretic systems under random excitation. *ASME J. Appl. Mech.* **47**, 150–154 (1980)
33. Willems, J.C.: Dissipative dynamical systems part I: general theory. *Arch. Ration. Mech. Anal.* **45**, 321–351 (1972)
34. Wright, J.P., Pei, J.S.: Solving dynamical systems involving piecewise restoring force using state event location. *ASCE J. Eng. Mech.* **138**(8), 997–1020 (2012)