JOURNAL OF EARTHQUAKE ENGINEERING
https://doi.org/10.1080/13632469.2024.2333822

Taylor & Francis
Taylor & Francis Group

Unconditionally Stable Central Difference Dissipative Algorithm for Multi-Directional Real-Time Hybrid Simulations of Large Nonlinear Structural Systems

Safwan Al-Subaihawi [a], James M. Ricles [a], Spencer E. Quiel [a], and Thomas Marullo [b]

[a]Department of Civil and Environmental Engineering, Lehigh University, Bethlehem, Pennsylvania, USA; [b]ATLSS Engineering Research Center, Lehigh University, Bethlehem, Pennsylvania, USA

ABSTRACT

The central difference is a popular algorithm used to integrate the equations of motion, yet suffers from two drawbacks: (1) it is only conditionally stable and requires a small-time step to maintain numerical stability; (2) it is non-dissipative, and high-frequency spurious oscillations may appear and compromise the accuracy of the solution. These drawbacks are detrimental to applying the algorithm to the real-time hybrid simulation of large, complex nonlinear structural systems. In this paper, the conventional central difference algorithm is modified to overcome these drawbacks, and the modified algorithm is applied to the real-time hybrid simulation of complex structural systems.

ARTICLE HISTORY
Received 7 March 2023
Accepted 18 March 2024

KEYWORDS
3D real-time hybrid simulation; nonlinear multi-directional response; unconditionally stability; explicit formulation; controllable numerical damping; model-based explicit integration algorithm; discrete control theory

1. Introduction

When conducting a real-time hybrid simulation (RTHS), a structural system is discretized into an experimental and numerical substructure, where the former is used to physically model components for which no accurate numerical models exist, and the latter to model the remaining parts of the system for which accurate numerical models can be used. The substructures lead to the defining of degrees of fredome (DOF) associated with the equations of motion for the system. In a RTHS the equations of motion are integrated to obtain the displacement commands for the simulation that lead to the state determination of a structural system, including the determination of the restoring forces, velocities, and accelerations. For each time step, the displacement compatibility between the analytical and experimental substructure is enforced in real-time to ensure that the rate of loading is maintained in real time. As indicated in Fig. 1, the simulation coordinator integrates the equations of motion and the target displacement commands $\mathbf{x}^a$ and $\mathbf{x}^e$ are sent to the analytical and experimental substructures, respectively. The measured restoring forces $\mathbf{R}^a$ and $\mathbf{R}^e$ from the analytical and experimental substructures, respectively, are then obtained and used to complete the integration of the equations of motion for the given time step. This process is then repeated for each subsequent time step.

The Central Difference (CD) algorithm is a popular method for integrating the equations of motion in many applications such as wave propagation problems (Noh and Bathe 2013). The method is advantageous because it is computational efficient and the solution is determined explicitly, even for problems that exhibit nonlinear behavior. Thus, no iterations are required to satisfy equilibrium at the end of the integration time step. Another feature of the algorithm is that the calculation of the velocity and acceleration at the end of an integration time step is optional because the recursive relation of the

CONTACT Safwan Al-Subaihawi swa313@lehigh.edu Department of Civil and Environmental Engineering, Lehigh University, Lehigh University, 117 ATLSS Dr, Bethlehem, PA 18015

This article has been republished with minor changes. These changes do not impact the academic content of the article.

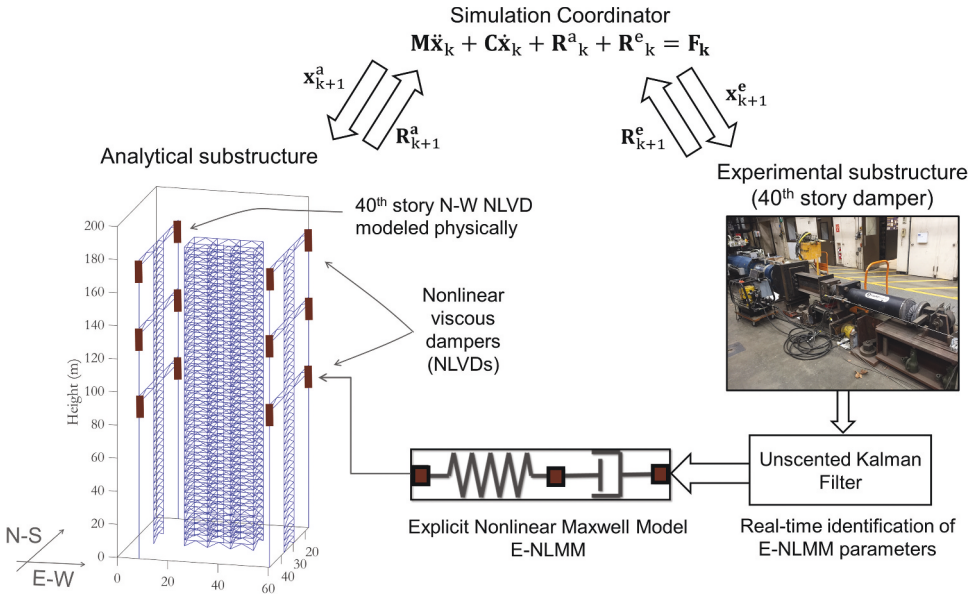

Figure 1. Representative RTHS of a 40-story tall building.

displacement depends only on the displacement of the previous two-time steps. These attributes are appealing when applied to RTHS. However, the velocity is implicit in nature since at time step i it is a function of the displacement at time step $i+1$ (see Eq. (4a,b)). Thus, to maintain the explicit form of the CD algorithm for both velocity and displacement, modifications are required when it is applied to RTHS. Wu et al. (2005) developed a modification of the CD algorithm to overcome this barrier, thereby enabling it to be applied to RTHS. Stability and accuracy analysis of the modified CD algorithm by Wu et al. (2005) found that the modified CD algorithm suffers a deterioration in its stability as the damping ratio of the experimental substructure increases. Wu et al. (2005) recommended that the algorithm was most suitable for an RTHS of systems that have a lightly damped experimental substructure and recomended that small time steps be used to ensure accuracy.

The CD algorithm, however, is only conditionally stable for linear and nonlinear problems (Chopra 2012), where the maximum permissible time step size to maintain numerical stability is dependent on the highest frequency of the system. It is therefore computationally expensive to integrate the equations of motion via the CD algorithm if the structural system possesses many DOFs, for a small time step is required. In a RTHS, the minimum time step is dictated by the servo controller's clock speed, digital controllers typically have a 1024 Hz clock speed and therefore restrict the minimum time step to be 1/1024 sec. For a RTHS's numerical substructure with many DOFs, the choice of the integration time step when using the CD algorithm can be governed by numerical stability rather than the required accuracy of the solution. The computational cost associated with a small time step can thus hinder the implementation of the CD algorithm for RTHS with large structural models.

Unconditionally stable algorithms assure a numerically stable solution of the integration process, irrespective of the selected integration time step. These types of algorithms offer the advantage that the selection of the time step is governed by the required accuracy of the solution instead of the numerical stability requirement (Hilber, Hughes, and Taylor 1977). This can be beneficial since the computational effort for integrating the equations of motion can be significantly reduced when a larger time step is used. Most well-known unconditionally stable algorithms are implicit in their formulation and require iterations to guarantee equilibrium at the end of each integration time step when solving nonlinear problems (Chung and Hulbert 1993; Hilber, Hughes, and Taylor 1977; Newmark 1959). Iterating in a RTHS is not

conducive, since the number of iterations needed to achieve equilibrium is not deterministic and can extend beyond the time allocated for such a process (which is a fraction of the time step for the simulation). Jung et al. (2007) developed an implicit time-integration method for real-time hybrid simulation, where a nonlinear solution strategy that combines Newton-type iteration with sub-increments based on the HHT-α method (Hilber, Hughes, and Taylor 1977). Chen and Ricles (2012) however determined using discrete control theory that the algorithm proposed by Jung et al. (2007) is only conditionally stable for nonlinear softening or hardening structures. Wu et al. (2007) proposed a real-time sub-structuring method called the equivalent force control (EFC) method that unified the effective force and hybrid simulation techniques in order to eliminate the numerical iteration process in implicit time integration. One of the important issues that Wu et al. (2007) noted is that restoring force measurement errors in the experimental substructure is an important issue for EFC. Noise resulting from measurement errors can develop that can affect the accuracy of the test results, depending on the contribution of the measured force to the total equivalent force. Wu et al. (2007) recommended that this phenomenon required further evaluation.

Conversely, as noted above, explicit integration algorithms do not require iterations, making them favorable when performing RTHS of nonlinear systems. Among the first unconditionally stable explicit algorithms is that developed by Chang for pseudo dynamic testing (Chang 2002). Chen and Ricles (2008a) later developed the unconditionally stable explicit CR algorithm using discrete control theory. Both Chang's and the CR algorithm are second order accurate, possess no numerical damping, and are only conditionally stable for hardening type nonlinear problems. Unconditionally stable explicit algorithms with controllable numerical damping have been developed, including the KR-α method by Kolay and Ricles (2014) and the algorithm developed by Chang (2014). The modified KR-α method was then proposed to improve the stability and the overshoot characteristics of the KR-α method (Kolay and Ricles 2019).

Another disadvantage of the CD algorithm is the presence of high frequency spurious oscillations that can appear in the solution (Noh and Bathe 2013). These oscillations are caused by various phenomena, including numerical rounding error, nonlinearities that can occur in the system within the time step (e.g. inelastic material response), and the inaccurate integration of the higher modes of vibration. One of the solutions to the spurious oscillations problem is to use numerical damping that can suppress these high frequency oscillations while leaving the lower modes of interest unaffected (Hilber, Hughes, and Taylor 1977). However, the CD algorithm is not dissipative, and these spurious oscillations can therefore be inherent in the solutions obtained using the algorithm.

In this paper, the conventional CD algorithm characteristics are reviewed using discrete control theory, which is then used to formulate a new algorithm that overcomes the disadvantages mentioned above. A modified version of the conventional CD algorithm (referred to as the *Model-based Central Difference*, MCD, algorithm) is shown to be unconditionally stable for linear and softening-type nonlinear problems. A single free parameter for controllable numerical damping is incorporated into the MCD algorithm to control spurious high-frequency oscillations. Numerical analyses of systems are performed to demonstrate the algorithm's characteristics, and the results are compared to solutions based on other algorithms and exact solutions to verify the MCD algorithm. To demonstrate the robustness and computational efficiency of the MCD algorithm for a RTHS, this paper concludes with a 3D RTHS of a tall building subjected to bidirectional earthquake ground motions, where the structure has a vast number of DOFs and undergoes nonlinear behavior during the simulation.

2. Analysis of Conventional CD Algorithm Using Discrete Control Theory

In control theory, the z-transform is used to find the transfer function of a discrete system given its difference equation. The transfer function of a linear system is defined as the ratio between the output

to the input of the system. The z-transform of a discrete time signal is mathematically defined as follows:

$$X(z) = \sum_{k=0}^{\infty} x(k) z^{-k} \tag{1}$$

where $x(k)$ is a discrete sample of the continuous function $x(t)$ at discrete points $k \in \{0, 1, ..\infty\}$ in time t. For example, the difference equation that describes the relation between the output $y(k)$ and the input $x(k)$ of a system is given as (Franklin, Powell, and Emami 2009):

$$y(k) = -a_1 y(k-1) - a_2 y(k-2) + b_o x(k) + b_1 x(k-1) + b_2 (k-2) \tag{2}$$

where a_1, a_2, b_o, b_1, and b_2 are constant coefficients. Taking the z-transform of Eq. (2) using the definition described in Eq. (1) and then solving for the transfer function (also known as the discrete transfer function) defined as $G(z) = Y(z)/X(z)$ results in

$$G(z) = \frac{Y(z)}{X(z)} = \frac{b_0 z^2 + b_1 z + b_2}{z^2 + a_1 z + a_2} \tag{3}$$

where $Y(z)$ and $X(z)$ are the z-transforms of $y(k)$ and $x(k)$.

The transfer function of a system provides valuable information about its behavior (Franklin, Powell, and Emami 2009). Specifically, the solution of the roots of the polynomial equation in the denominator of the transfer function when equated to zero provides the poles of the transfer function. The location of the poles in the complex z-plane not only effects how the system behaves due to an external disturbance, but also the tendency of the system to be stable. The roots to the numerator of the transfer function that make it equal to zero are called the zeros of the transfer function.

Figure 2 shows the expected response of a system based on the location of the poles in the complex z-plane (Franklin, Powell, and Emami 2009) (only the upper part of the plane is shown in order to to conserve space in the figure). Three important characteristics are shown in Fig. 2 (via the use of displacement time history x(t) plots): (1) the response of the poles outside the unit circle becomes unbounded (indicating instability); (2) the response of the poles on the unit circle is stable and possesses no damping; and (3) the response of the poles inside the unit circle is stable and possesses damping. Inside the unit circle, the magnitude of damping increases as the distance between the pole location and the origin is reduced.

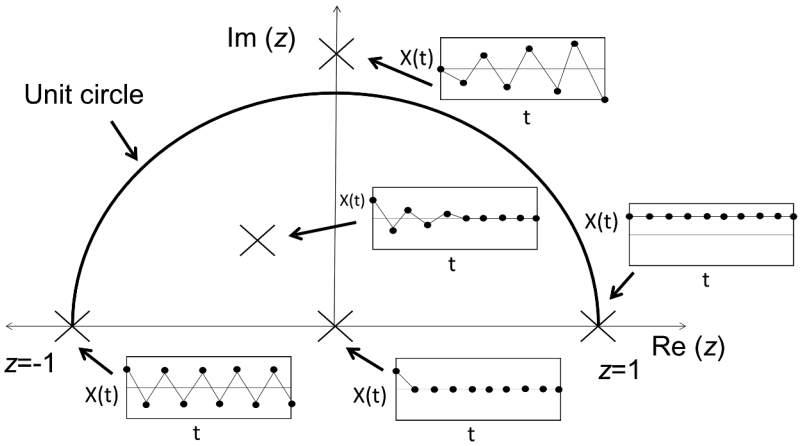

Figure 2. System response based on the location of the poles in the complex z-plane (only the upper part of the plane is shown in order to conserve space in the figure).

The procedure explained above can be used to obtain the discrete transfer function for the conventional CD algorithm (Chen and Ricles 2008a) and provide insight into the characteristics of the CD algorithm. The velocity and acceleration difference equations of the CD algorithm are given by Eq. (4a) and (5a). Taking the z-transform of these difference equations leads to the expressions in Eqs. (4b) and (5b).

$$\text{Velocity: } v_i = \frac{x_{i+1} - x_{i-1}}{2\Delta t}, \; V(z) = \frac{(z^2 - 1)}{2\Delta t z} X(z) \tag{4a, b}$$

$$\text{Acceleration: } a_i = \frac{x_{i+1} - 2x_i + x_{i-1}}{\Delta t^2}, \;\; A(z) = \frac{(z^2 - 2z + 1)}{\Delta t^2 z} X(z) \tag{5a, b}$$

Eqs. (4b) and (5b) can be substituted into the equation of motion for a single degree of freedom (SDOF) system (Eq. (6a), where Eq. (6b) is the associated z-transform form):

$$a_i \Delta t^2 + 2\Omega\xi\Delta t v_i + \Omega^2 x_i = f_i \Delta t^2, A(z)\Delta t^2 + 2\Omega\xi\Delta t V(z) + \Omega^2 X(z) = F(z)\Delta t^2 \tag{6a, b}$$

where f_i is the forcing function at time i divided by the mass m. Substituting Eq. 4(b) and 5(b) into 6(b) leads to the discrete transfer function $G(z) = X(z)/F(z)$:

$$G(z) = \frac{X(z)}{F(z)} = \frac{\Delta t^2 z}{(1 + \Omega\xi)z^2 + (\Omega^2 - 2)z - \Omega\xi + 1} \tag{7}$$

In Eqs. (6) and (7), $\Omega = \omega_n \Delta t$, where ω_n is the undamped natural frequency, Δt is the integration time step, and ξ is the viscous damping ratio. Eq. (7) represents an open loop system with no feedback and is therefore an open loop transfer function. The solution for z of the polynomial of the numerator of the transfer function when set equal to zero is associated with the open loop zero(s), while that of the denominator is associated with the open loop pole(s). The denominator of Eq. (7) is known as the characteristic equation (Franklin, Powell, and Emami 2009). By equating the characteristic equation to zero and subsequently solving for the two poles $z_{1,2}$ gives the open loop poles when $\Omega = 0$ and the open loop zeros when $\Omega = \infty$, where:

$$z_{1,2} = \frac{-\Omega^2 + 2 \pm \sqrt{\Omega^4 + 4\Omega^2(\xi^2 - 1)}}{2\Omega\xi + 2} \tag{8}$$

The location of the poles $z_{1,2}$ of the characteristic equation in the complex plane start from the open loop poles and terminate at the open loop zeros, unless an instability exists as Ω is increased from zero to infinity. Figure 3 shows the location of the poles $z_{1,2}$ as Ω increases for the case when $\xi = 0$ (referred as the root locus plot). The two root loci branches in Fig. 3 both begin at the open loop poles and travel around the non-dissipative region of the complex z-plane (i.e., along the circumference of the unit circle) indicating that the CD algorithm is non-dissipative. One of the branches terminates at the open loop zero while the other branch leaves the unit circle, as seen in Fig. 3. The latter is an indication of instability. The value of Ω that induces instability can be found by setting $z_2 = -1$ in Eq. (8) and solving for Ω which leads to $\Omega_{cr} = 2$. This result agrees with the well-known stability limit of $\omega_n \Delta t = 2$ for the CD algorithm.

Thus, the well-known characteristics for the CD algorithm of being non-dissipative and conditionally stable are confirmed by the root loci in Fig. 3. To obtain a dissipative, unconditional stable form of the CD algorithm, the poles of the algorithm must migrate inwards and remain within the unit circle. In the next section of this paper the MCD is developed based on this requirement.

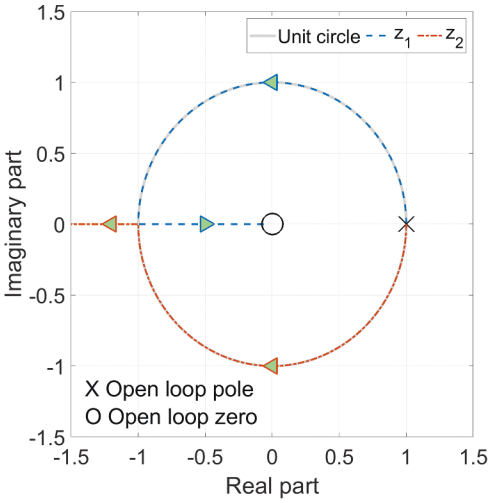

Figure 3. Closed loop poles of the conventional central difference algorithm transfer function.

3. Development of the MCD Algorithm

3.1. SDOF Systems

To overcome the shortcomings of the CD algorithm, the following two characteristics of the MCD algorithm are necessary: (1) unconditional stability; and (2) providing controlled numerical damping to remove oscillatory behavior at high frequencies (i.e., dissipative). The MCD algorithm is developed using control theory where a controller and compensator are incorporated into the design of the algorithm to enable it to possess these required characteristics. The formulation is based on the following weighted CD equations for velocity and acceleration:

$$v_i = \frac{(1-\gamma_1)x_{i+1} + \gamma_1 x_i - (1-\gamma_2)x_{i-1} - \gamma_2 x_i}{2\Delta t},$$
$$a_i = \frac{(1-\gamma_1)x_{i+1} + \gamma_1 x_i - 2x_i + (1-\gamma_2)x_{i-1} + \gamma_2 x_i}{\Delta t^2 \gamma_3} \quad (9a, b)$$

In Eq. (9) the parameters γ_1, γ_2 and γ_3 are feedback gains for the controller and compensator that enables the new algorithm to have the above characteristics. Taking the z-transform of Eq. (9) and substituting the result into the equation of motion leads to the following transfer function $G(z) = X(z)/F(z)$

$$G(z) = \frac{X(z)}{F(z)} = \frac{b_1 z}{a_0 z^2 + a_1 z + a_2} \quad (10)$$

where a_0, a_1, a_2, and b_1 are constant coefficients and defined in Table 1. The feedback gains γ_1, γ_2, and γ_3 that appear in the denominator (i.e., characteristic equation) of the transfer function control the location of the poles in the complex z-plane.

Table 1. Coefficients of the MCD transfer function.

Coefficient	Value
b_1	$\Delta t^2 \gamma_3$
a_0	$-\Omega\xi\gamma_1\gamma_3 + \Omega\xi\gamma_3 - \gamma_1 + 1$
a_1	$\Omega\xi\gamma_1\gamma_3 - \Omega\xi\gamma_2\gamma_3 + \Omega^2\gamma_3 + \gamma_1 + \gamma_2 - 2$
a_2	$\Omega\xi\gamma_2\gamma_3 - \Omega\xi\gamma_3 - \gamma_2 + 1$

Unconditional stability of the MCD algorithm is achieved through the controller gain γ_3 that makes the poles of the algorithm remain on the unit circle $\Omega\mathbb{R} \geq 0$. In addition, γ_3 helps to improve the overshoot properties of the MCD, which are presented later in the paper. The value of γ_3 is obtained by matching the poles of the transfer function for the non-dissipative form of the MCD algorithm to that of the unconditionally stable Newmark Constant Average Acceleration (CAA) algorithm (Newmark 1959). A similar pole matching technique was previously used in developing other integration algorithms such as the CR and KR-α methods (Chen and Ricles 2008a; Kolay and Ricles 2014). The compensator gains γ_1 and γ_2 are not used in this stage of the controller design and are temporarily set to zero. The poles for the non-dissipative unconditionally stable CAA are

$$z_{CAA1,2} = \frac{-\Omega^2 + 4 \pm 4\sqrt{\Omega^2\xi^2 - \Omega^2}}{\Omega^2 + 4\Omega\xi + 4} \tag{11}$$

while those of the non-dissipative form of the MCD algorithm are

$$z_{MCDnonDis1,2} \frac{-\Omega^2\gamma_3 + 2 \pm \sqrt{\Omega^4\gamma_3^2 + 4\Omega^2\xi^2\gamma_3^2 - 4\Omega^2\gamma_3}}{2\Omega\xi\gamma_3 + 2} \tag{12}$$

and the result for γ_3 is therefore

$$\gamma_3 = \frac{4}{\Omega^2 + 4} \tag{13}$$

The root loci of the MCD transfer function with $\gamma_1 = \gamma_2 = 0$ (i.e., Eq. (12)) end in the oscillatory region of the complex z-plane for the $\lim_{\Omega\to\infty} z_{\text{MCD nonDis1,2}} = -1$. The compensator is now incorporated into the derivation to eliminate the oscillatory behavior at high frequencies, making use of the feedback gains γ_1 and γ_2. The compensator adds two open loop zeros to the MCD transfer function located in the dissipative region of the complex z-plane at a distance of ρ_∞ from the origin, where the root loci terminate when $\Omega \to \infty$. ρ_∞ is analogous to the spectral radius of the amplification matrix at high frequencies (i.e., when $\Omega \to \infty$), where the spectral radius ρ is defined as the distance of the pole from the origin in the complex z-plane.

The compensator feedback gains γ_1 and γ_2 are found by matching the poles of the dissipative form of the MCD algorithm (i.e., when γ_1 and γ_2 are non-zero values) to a desired set of poles that are selected using discrete control theory to satisfy the following conditions: (1) convergence of $\lim_{T\to 0} z = e^{sT} = 1$, where T is the sampling period; (2) the root loci branches end in the dissipative region of the complex z-plane at the two open loop zeros of the MCD algorithm located at the distance $\rho_\infty \in [0, 1]$ from the origin; and (3) the root loci branches travel within the unit circle $|z_{1,2}| \leq 1$, $\Omega \in \mathbb{R} \geq 0$. The resulting two poles of the MCD are a function of γ_1, γ_2, and γ_3 and equal to

$$z_{MCD1,2} = \frac{c_1 \pm \sqrt{c_2}}{c_3} \tag{14}$$

where the coefficients c_1, c_2, and c_3 are given in Table 2. The set of desired poles having dissipative characteristics are derived in the Appendix based on satisfying the above conditions, and equal to

$$z_{1,2} = \frac{d_1 \pm \sqrt{d_2}}{d_3} \tag{15}$$

where the coefficients d_1, d_2, and d_3 are given in Table 2. Equating the two sets of poles to each other leads to

Table 2. Coefficients for the poles of $z_{MCD1,2}$ and $z_{1,2}$.

Coefficient	Values
c_1	$\Omega\xi\gamma_1\gamma_3 - \Omega\xi\gamma_2\gamma_3 + \Omega^2\gamma_3 + \gamma_1 + \gamma_2 - 2$
c_2	$(\Omega^2 + 2\Omega\xi(\gamma_1 - \gamma_2) + \xi^2(\gamma_1 + \gamma_2 - 2)^2)\Omega^2\gamma_3^2 + 2\Omega(\gamma_1 + \gamma_2 - 2)(\Omega + \xi(\gamma_1 - \gamma_2))\gamma_3 + (\gamma_1 - \gamma_2)^2$
c_3	$2(\Omega\xi\gamma_1\gamma_3 - \Omega\xi\gamma_3 + \gamma_1 - 1)$
d_1	$\rho_\infty + 1$
d_2	$((-\Omega\xi + \xi^2 - 1)\rho_\infty^2 + (-\Omega^2 + 2\xi^2 - 2)\rho_\infty + \xi^2 + \Omega\xi - 1)\Omega^2$
d_3	$\Omega^2 + \Omega\xi(\rho_\infty + 1) + \rho_\infty + 1$

$$\gamma_1 = \left(\frac{(\rho_\infty - 3)\Omega^2}{(\rho_\infty + 1)(\Omega^2 + 4\Omega\xi + 4)}\right), \gamma_2 = \left(\frac{(3\rho_\infty - 1)\Omega^2}{(\rho_\infty + 1)(-\Omega^2 + 4\Omega\xi - 4)}\right) \tag{16}$$

The two gains $\gamma_1$and γ_2 are functions of $\rho_\infty \in [0, 1]$ due to setting the poles equal to $\pm\sqrt{-\rho_\infty}$ as $\Omega \to \infty$, (see the Appendix). All of the feedback gains γ_1, γ_2 and γ_3 represent integration parameters and depend on the model properties of the dynamic system, and therefore the MCD algorithm is a model-based integration algorithm (Chang 2002, 2014; Chen and Ricles 2008a; Kolay and Ricles 2014).

3.2. Multi-Degree of Freedom (MDOF) Systems

For a multi-degree of freedom (MDOF) system, the algorithm is described by the following equations:

$$\begin{aligned} \mathbf{v}_i &= [2\Delta t]^{-1}\left[(\mathbf{I} - \gamma_1)\mathbf{x}_{i+1} + \gamma_1\mathbf{x}_i - (\mathbf{I} - \gamma_2)\mathbf{x}_{i-1} - \gamma_2\mathbf{x}_i\right], \\ \mathbf{a}_i &= \left[\Delta t^2\gamma_3\right]^{-1}\left[(\mathbf{I} - \gamma_1)\mathbf{x}_{i+1} + \gamma_1\mathbf{x}_i - 2\mathbf{x}_i + (\mathbf{I} - \gamma_2)\mathbf{x}_{i-1} + \gamma_2\mathbf{x}_i\right] \end{aligned} \tag{17a, b}$$

Moreover, the equations of motion are

$$\mathbf{M}\mathbf{a}_i + \mathbf{C}\mathbf{v}_i + \mathbf{K}\mathbf{x}_i = \mathbf{F}_i \tag{18}$$

where items in bold are matrices or vectors, and **M**, **C**, and **K** are the system's mass, damping, and stiffness matrices, respectively, of dimension NDOF x NDOF for an MDOF system, where NDOF is the number of DOF. Considering the damping matrix **C** to be classical and using the orthogonality of modes, Eqs. (17) and (18) can be written in modal coordinates:

$$\begin{aligned} \dot{\mathbf{Y}}_i &= [2\Delta t]^{-1}\left[(\mathbf{I} - \boldsymbol{\gamma}_1^*)\mathbf{Y}_{i+1} + \boldsymbol{\gamma}_1^*\mathbf{Y}_i - (\mathbf{I} - \boldsymbol{\gamma}_2^*)\mathbf{Y}_{i-1} - \boldsymbol{\gamma}_2^*\mathbf{Y}_i\right], \\ \ddot{\mathbf{Y}}_i &= \left[\Delta t^2\boldsymbol{\gamma}_3^*\right]^{-1}\left[(\mathbf{I} - \boldsymbol{\gamma}_1^*)\mathbf{Y}_{i+1} + \boldsymbol{\gamma}_1^*\mathbf{Y}_i - 2\mathbf{Y}_i + (\mathbf{I} - \boldsymbol{\gamma}_2^*)\mathbf{Y}_{i-1} + \boldsymbol{\gamma}_2^*\mathbf{Y}_i\right] \end{aligned} \tag{19a, b}$$

where

$$\mathbf{M}^*\ddot{\mathbf{Y}}_i + \mathbf{C}^*\dot{\mathbf{Y}}_i + \mathbf{K}^*\mathbf{Y}_i = \boldsymbol{\Phi}^T\mathbf{F}_i \tag{20}$$

In the aforementioned equations, $\boldsymbol{\Phi} = \left[\phi_1\phi_2 \ldots \phi_n\right]$ is the mode shape matrix, and ϕ_i the eigenvector for the i^{th} mode ($i \in 1 \ldots n$); $\mathbf{Y}, \dot{\mathbf{Y}}, \ddot{\mathbf{Y}}$ are the complete set of displacement, velocity, and accelerations in modal coordinates, respectively, which are related to physical coordinates by $\mathbf{x} = \boldsymbol{\Phi}\mathbf{Y}$, etc.; $\mathbf{M}^* = \boldsymbol{\Phi}^T\mathbf{M}\boldsymbol{\Phi}, \mathbf{C}^* = \boldsymbol{\Phi}^T\mathbf{C}\boldsymbol{\Phi}, \mathbf{K}^* = \boldsymbol{\Phi}^T\mathbf{K}\boldsymbol{\Phi}$ are the modal diagonal mass, damping, and stiffness matrices, respectively; and $\boldsymbol{\gamma}_1^* = \boldsymbol{\Phi}^T\boldsymbol{\gamma}_1\boldsymbol{\Phi}, \boldsymbol{\gamma}_2^* = \boldsymbol{\Phi}^T\boldsymbol{\gamma}_2\boldsymbol{\Phi}, \mathbf{and}\boldsymbol{\gamma}_3^* = \boldsymbol{\Phi}^T\boldsymbol{\gamma}_3\boldsymbol{\Phi}$ are diagonal matrices of the integration parameter. Eqs. (19) and (20) each represent a set of NDOF uncoupled equations. The integration parameters for any mode j can be determined based on the integration parameters for an SDOF system presented in Eq. (13) and (16), where:

$$\gamma_{1,j}^{*} = \left(\frac{(\rho_\infty - 3)\Omega_j^2}{(\rho_\infty + 1)\left(\Omega_j^2 + 4\Omega_j\xi_j + 4\right)} \right) = \frac{(\rho_\infty - 3)K_j^{*}\Delta t^2}{(\rho_\infty + 1)\left(K_j^{*}\Delta t^2 + 2C_j^{*}\Delta t + 4M_j^{*}\right)} \tag{21}$$

$$\gamma_{2,j}^{*} = \left(\frac{(3\rho_\infty - 1)\Omega_j^2}{(\rho_\infty + 1)\left(-\Omega_j^2 + 4\Omega_j\xi_j - 4\right)} \right) = \frac{(3\rho_\infty - 1)K_j^{*}\Delta t^2}{(\rho_\infty + 1)\left(-K_j^{*}\Delta t^2 + 2C_j^{*}\Delta t - 4M_j^{*}\right)} \tag{22}$$

$$\gamma_{3,j}^{*} = \frac{4}{\Omega_j^2 + 4} = \frac{4M_j^{*}}{K_j^{*}\Delta t^2 + 4M_j^{*}} \tag{23}$$

where $M_j^{*}, C_j^{*} = 2M_j^{*}\omega_j\xi_j$, and $K_j^{*} = M_j^{*}\omega_j^2$ are the modal mass, damping coefficient, and stiffness coefficient, respectively, for the j^{th} mode. The integration parameters in modal coordinates are the diagonal terms of the following matrices:

$$\boldsymbol{\gamma}_1^{*} = (\rho_\infty - 3)\left[(\rho_\infty + 1)[\mathbf{K}^{*}\Delta t^2 + 2\mathbf{C}^{*}\Delta t + 4\mathbf{M}^{*}]\right]^{-1}\mathbf{K}^{*}\Delta t^2 \tag{24}$$

$$\boldsymbol{\gamma}_2^{*} = (3\rho_\infty - 1)\left[(\rho_\infty + 1)[-\mathbf{K}^{*}\Delta t^2 + 2\mathbf{C}^{*}\Delta t - 4\mathbf{M}^{*}]\right]^{-1}\mathbf{K}^{*}\Delta t^2 \tag{25}$$

$$\boldsymbol{\gamma}_3^{*} = [\mathbf{K}^{*}\Delta t^2 + 4\mathbf{M}^{*}]^{-1}4\mathbf{M}^{*} \tag{26}$$

Pre- and post-multiplying the aforementioned equations by $\left(\boldsymbol{\Phi}^{\mathbf{T}}\right)^{-1}$ and $\boldsymbol{\Phi}^{-1}$, respectively, and using the relationships among the integration parameters in the physical and modal coordinate systems as presented earlier, the integration parameters in physical coordinates can be determined as

$$\boldsymbol{\gamma}_1 = \left(\boldsymbol{\Phi}^{\mathbf{T}}\right)^{-1}\boldsymbol{\gamma}_1^{*}\boldsymbol{\Phi}^{-1} = \left[(\rho_\infty + 1)\left(\mathbf{K}\Delta t^2 + 2\mathbf{C}\Delta t + 4\mathbf{M}\right)\right]^{-1}\left((\rho_\infty - 3)\mathbf{K}\Delta t^2\right) \tag{27}$$

$$\boldsymbol{\gamma}_2 = \left(\boldsymbol{\Phi}^{\mathbf{T}}\right)^{-1}\boldsymbol{\gamma}_2^{*}\boldsymbol{\Phi}^{-1} = \left[(\rho_\infty + 1)\left(-\mathbf{K}\Delta t^2 + 2\mathbf{C}\Delta t - 4\mathbf{M}\right)\right]^{-1}\left((3\rho_\infty - 1)\mathbf{K}\Delta t^2\right) \tag{28}$$

$$\boldsymbol{\gamma}_3 = \left(\boldsymbol{\Phi}^{\mathbf{T}}\right)^{-1}\boldsymbol{\gamma}_3^{*}\boldsymbol{\Phi}^{-1} = \left[\mathbf{K}\Delta t^2 + 4\mathbf{M}\right]^{-1}4\mathbf{M} \tag{29}$$

4. Characteristics of the MCD Integration Algorithm

The integration parameters γ_1, γ_2, and γ_3 are a function of one free parameter $\rho_\infty \in [0, 1]$, and therefore the MCD algorithm is a one parameter model-based algorithm. Figure 4 illustrates that the MCD algorithm is unconditionally stable for linear type problems with $\rho_\infty \in [0, 1]$. Observe in Fig. 4a that in the limit $\Omega \to \infty$, and $\rho_\infty \in [0, 1]$, that the MCD is mapped into $\bar{\Omega} \in [0, \pi/2]$, where $\bar{\Omega}$ is defined below in Eq. (31b). The value of ρ_∞ is equated to $\left(\rho^{*}{}_\infty\right)^2$ allowing the expression ρ_∞^{*} to then used to aesthetically present Fig. 4 with evenly spaced spectral radii.The root loci presented in Fig. 4a show that: (1) the poles of the MCD transfer function are moving either on or within the unit circle, (2) that the poles are complex conjugate pairs, and (3) that the spectral radius (Fig. 4b) of the MCD poles is always less than unity. As noted previously, the spectral radius is defined as the distance from the origin to the location of the pole in the complex z-plane, where a value larger than unity indicates an instability.

Numerical dispersion and energy dissipation of an integration algorithm are generally measured in terms of relative period error (PE) and equivalent damping ratio $\bar{\xi}$, respectively, where:

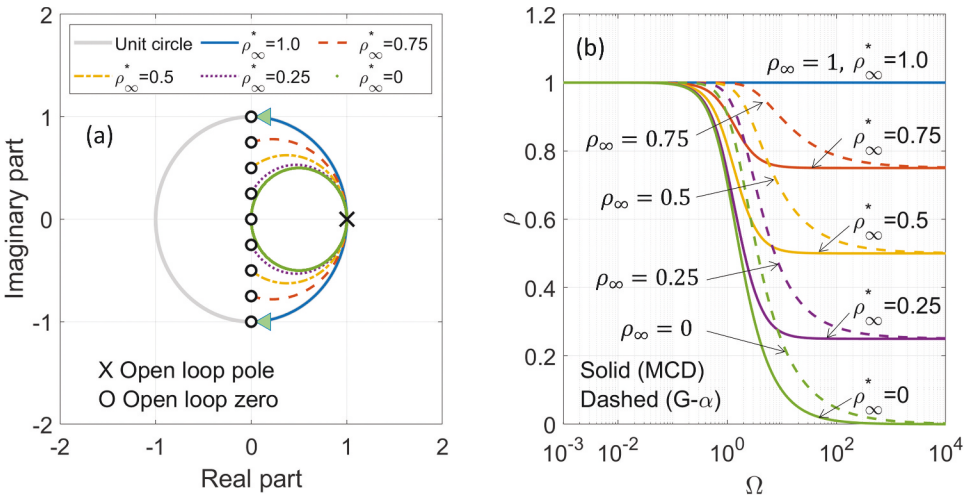

Figure 4. Characteristics of the MCD algorithm: (a) root loci branches; and, (b) spectral radius.

$$\mathrm{PE} = \frac{\bar{\mathrm{T}} - \mathrm{T}}{\mathrm{T}},\ \mathrm{T} = \frac{2\pi}{\omega},\ \bar{\mathrm{T}} = \frac{2\pi}{\bar{\omega}} \tag{30}$$

where $\bar{\omega}$ is defined in Eq. (31b). The equivalent damping ratio $\bar{\xi}$ and apparent frequency $\bar{\Omega}$ for an integration algorithm is determined from the position of the poles in the complex z-plane,

$$\bar{\xi} = -\frac{1}{2\bar{\Omega}}\ln\left(\sigma^2 + \varepsilon^2\right), \bar{\Omega} = \bar{\omega}\Delta t = \left(1 - \bar{\xi}^2\right)^{-0.5}\tan^{-1}\left(\frac{\varepsilon}{\sigma}\right) \tag{31a,b}$$

where $\sigma = \mathrm{Re}\{z_{1,2}\}$ and $\varepsilon = \mathrm{Im}\{z_{1,2}\}$, and $z_{1,2}$ are the pair of complex conjugate poles. The PE and equivalent damping ratio $\bar{\xi}$ are plotted in Fig. 5a,b, respectively, for various values of ρ_∞, and the PE compared to the Generalized-α method (G-α) method (Chung and Hulbert 1993). The G-α method is an implicit method that has controlled numerical damping, and is often used to compare with the controlled numerical damping properties of other algorithms. As illustrated in Fig. 5b maximum numerical damping for the higher frequencies is achieved when $\rho_\infty = 0$, while $\rho_\infty = 1$ changes the MCD algorithm into a non-dissipative scheme. Like the G-α method, it is seen in Fig. 5 that the PE and $\bar{\xi}$ are minimal at the lower frequencies for the MCD algorithm, before

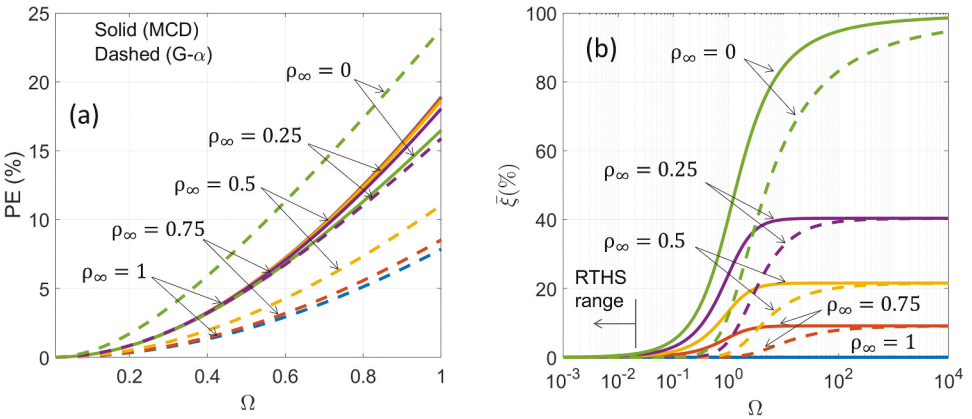

Figure 5. Characteristics of the MCD algorithm: (a) period elongation; and, (b) equivalent damping ratio.

they increase as Ω increases. Both the G-α and the MCD algorithms converge asymptotically to the same amount of numerical damping at the high frequencies as shown in Fig. 5b. The MCD algorithm experiences period elongation as Ω increases contrary to the conventional CD scheme, which exhibits period shortening at high Ω values. While the PE and $\bar{\xi}$ are minimal at the lower frequencies for the MCD algorithm, they are somewhat larger than that of the G-α algorithm. An exception is when $\rho_\infty = 0$ where it is found that the period elongation for the MCD algorithm is lower compared to the Generalized-α algorithm. The motivation for developing the MCD was to enable RTHS of systems with a large number of DOFs. For RTHS the time steps that are selected are a multiple of the clock speed of the servo controller, and typically range from $\Delta t = 1/1024$ sec to 6/1024 sec (Al-Subaihawi 2023; Kolay et al. 2015). Hence, Ω for RTHS typically is less than 0.02, where as shown in Fig. 5, the value of $\bar{\xi}$ is extremely small.

4.1. Special Starting Procedure

Like the CD algorithm, the MCD is not a self-starting algorithm. Considering a SDOF system, x_{-1} needs to be calculated based on the initial conditions x_0,v_0 and f_0. Setting i = 0 and solving for x_1 from Eq. (9a) and substituting the result into Eq. (9b) leads to the following expression for x_{-1}:

$$x_{-1} = x_0 + Gv_0 + Ha_0, \text{ where } G = \left(\frac{2\Delta t}{2\gamma_2 - 2}\right), H = \left(\frac{-\Delta t^2 \gamma_3}{2\gamma_2 - 2}\right) \tag{32a, b, c, d}$$

The initial acceleration a_0 is obtained directly from the equation of motion, Eq. 6(a), with i = 0. The quantity x_{-1} in Eq. (32a) is identical to that of the conventional CD algorithm when $\Omega = 0$, since $\lim_{\Omega\to 0} \gamma_1 = \lim_{\Omega\to 0} \gamma_2 = 0$, and $\lim_{\Omega\to 0} \gamma_3 = 1$. The denominator of Eq. (32b-c) is invertible for any underdamped system with a finite value of Ω. If the value for Ω is infinite, then the denominator is non-invertible when $\rho_\infty = 0$, hence caution must be used to avoid assigning $\rho_\infty = 0$ for the MCD. The determination of the displacement vector $\mathbf{x}_{-1}$ for MDOF systems is similar to Eq. (32), where G and H are replaced with their matrix equivalents with $\boldsymbol{\gamma}_2$ and $\boldsymbol{\gamma}_3$ determined from Eqs. (28) and (29).

4.2. Stability for Linear Systems

The unconditional stability of integration algorithms can be assessed by examining the eigenvalues of the amplification matrix (Hilber 1976) of a SDOF system subjected to free vibration. The spectral radius ρ of the eigenvalues λ needs to satisfy the requirement that the spectral radius $\rho = \max(|\lambda|) \leq 1$ in order to achieve unconditional stability. This section presents the amplification matrix of the MCD algorithm and shows that the algorithm is unconditionally stable for linear problems.

The recurrence relationship of the MCD algorithm for a SDOF system is expressed by Eq. (33), where $\mathbf{A}$ is the amplification matrix of the MCD algorithm, as given by Eq. (34a). The recurrence relationship for the displacement x_{i+1} is obtained by substituting the acceleration and velocity from Eq. (9) into the equation of motion, Eq. 6(a), and solving for the displacement x_{i+1}.

$$[x_{i+1} x_i]^T = \mathbf{A}[x_i x_{i-1}]^T + \mathbf{L} f_i \tag{33}$$

where

$$\mathbf{A} = \begin{bmatrix} \frac{2\rho_\infty + 2}{\Omega^2 + \Omega\xi\rho_\infty + \Omega\xi + \rho_\infty + 1} & \frac{-\Omega^2\rho_\infty + \Omega\xi\rho_\infty + \Omega\xi - \rho_\infty - 1}{\Omega^2 + \Omega\xi\rho_\infty + \Omega\xi + \rho_\infty + 1} \\ 1 & 0 \end{bmatrix}, L = \begin{bmatrix} \frac{(\rho_\infty + 1)\Delta t^2}{\Omega^2 + \Omega\xi\rho_\infty + \Omega\xi + \rho_\infty + 1} \\ 0 \end{bmatrix} \tag{34a, b}$$

The free vibration problem is associated with setting the forcing function f_i equal to 0 and specifying the initial displacement. The eigenvalues λ of $\mathbf{A}$ are the solution of the characteristic equation, Eq. (35), resulting from the eigenvalue problem applied to the amplification matrix. The eigenvalues are equivalent to the poles of the transfer function of the algorithm.

$$\lambda^2 + \left(\frac{-2\rho_\infty - 2}{\Omega^2 + \Omega\xi\rho_\infty + \Omega\xi + \rho_\infty + 1}\right)\lambda - \left(\frac{-\Omega^2\rho_\infty + \Omega\xi\rho_\infty + \Omega\xi - \rho_\infty - 1}{\Omega^2 + \Omega\xi\rho_\infty + \Omega\xi + \rho_\infty + 1}\right) = 0 \tag{35}$$

The solution for the eigenvalues is given by Eq. (36):

$$\lambda_{1,2} = \frac{\rho_\infty + 1 \pm \sqrt{-\Omega^4\rho_\infty + \Omega^3\left(\xi - \xi\rho_\infty^2\right) + \Omega^2\left(\xi^2\rho_\infty^2 + 2\xi^2\rho_\infty + \xi^2 - \rho_\infty^2 - 2\rho_\infty - 1\right)}}{\Omega^2 + \Omega\xi\rho_\infty + \Omega\xi + \rho_\infty + 1} \tag{36}$$

Taking the limits of Eq. (36) as $\Omega \to 0$ and $\Omega \to \infty$ leads to

$$\lim_{\Omega\to 0}\left(\lambda_{1,2}\right) = 1 \text{ and } \lim_{\Omega\to\infty}\left(\lambda_{1,2}\right) = \pm\sqrt{-\rho_\infty} \tag{37a, b}$$

Eq. (37) reveals that the eigenvalues start on the positive real axis with a value of 1 and end on the imaginary axis of the complex z-plane at $\pm\sqrt{-\rho_\infty}$, as expected and consistent with Fig. 4a. While it is shown that the eigenvalues of the MCD algorithm's transfer function start and end in the stable region of the complex z-plane, it is also of interest to investigate the stability in the intermediate range of $\Omega \in (0 \ldots \infty)$. This is accomplished by examining the maximum value for the spectral radius of the MCD algorithm's amplification matrix, which can be obtained by setting the partial derivatives of $\left|\lambda_{1,2}\right|$ from Eq. (36) with respect to Ω and ξ equal to zero. The resulting simultaneous equations have a solution at $\Omega = 0, \xi = 0$ which satisfies the requirement $\Omega, \quad \xi \in \mathbb{R} \geq 0$ and $\quad \rho_\infty \in [0, 1]$. Substituting $\Omega = 0, \xi = 0$ into Eq. (36) with $\rho_\infty = 1$ leads to a maximum value of the spectral radius ρ equal to unity. These results are consistent with the root locus plot shown in Fig. 4a and prove the unconditional stability of the MCD algorithm for linear problems.

4.3. Stability for Nonlinear Systems

The root locus method is utilized to investigate the stability of the MCD for nonlinear systems (Chen and Ricles 2008b). Discussion is limited to SDOF systems because the root locus method is applicable to Single-Input Single-Output (SISO) systems. Considering the equation of motion at time step i

$$ma_i + cv_i = p_i - kx_i = l_i \tag{38}$$

where the term $p_i - kx_i$ is defined as l_i. In the above equation of motion, m, c, and p_i are the mass, damping coefficient, and load at time step i, respectively, of the SDOF. The nonlinear stiffness between integration time steps is linearized to k_t, where k_t is the tangent stiffness matrix at time step i, in order to perform the root locus analysis where the incremental equation of motion becomes

$$m\Delta a_i + c\Delta v_i = \Delta p_i - k_t\Delta x_i = \Delta l_i \tag{39}$$

where $\Delta a_i = a_i - a_{i-1}$, $\Delta v_i = v_i - v_{i-1}$, $\Delta x_i = x_i - x_{i-1}$, and $\Delta p_i = p_i - p_{i-1}$. The block diagram of the MCD algorithm for nonlinear problems applied to the SDOF is shown in Fig. 6 where $F(z) = \sum_{k=-\infty}^{\infty} p(k)z^{-k}$, $G_1(z) = (1 - z^{-1})$, $G_3(z) = (1 - z^{-1})^{-1}$. The block diagram shown in Fig. 6 is a closed loop system due to the feedback associated with the gain of k_t.

The open loop transfer function $G_2(z)$ is obtained by taking the z-transform of Eq. (9), substituting the results into the incremental equation of motion, and solving for the resulting transfer function $G_2(z) = \Delta\, X(z)/\Delta\, L(z)$, whereby:

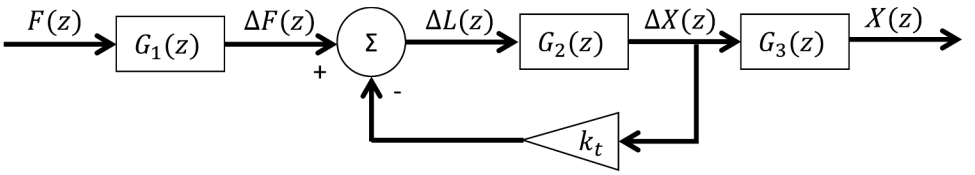

Figure 6. Closed loop block diagram of the MCD algorithm for nonlinear problems of an SDOF system.

Table 3. Coefficients of $G_2(z)$ transfer function.

Coefficient	Value
h_1	$2(\rho_\infty+1)\Delta t^2$
p_0	$2m(\rho_\infty+1)+c\Delta t(\rho_\infty+1)+2k_0\Delta t^2$
p_1	$-4m(\rho_\infty+1)-2k_0\Delta t^2(\rho_\infty+1)$
p_2	$2m(\rho_\infty+1)-c\Delta t(\rho_\infty+1)+2k_0\Delta t^2\rho_\infty$

$$G_2(z)=\frac{\Delta X(z)}{\Delta L(z)}=\frac{h_1 z}{p_0 z^2+p_1 z+p_2} \tag{40}$$

The parameters of the $G_2(z)$ transfer function of Eq. (40) are given in Table 3. Since the MCD algorithm is a model-based algorithm it is therefore assumed that the parameters γ_1, γ_2 and γ_3 are calculated based on the initial stiffness k_0 and mass m, where $\Omega=\sqrt{k_0/m}\Delta t$. The closed loop transfer function G^{NL}_{MCD} for nonlinear problems is determined using the block diagram of Fig. 6, whereby

$$G^{NL}_{MCD}(z)=\frac{X(z)}{F(z)}=G_1(z)\frac{G_2(z)}{1+k_t G_2(z)}G_3(z)=\frac{G_2(z)}{1+k_t G_2(z)} \tag{41}$$

The denominator of G^{NL}_{MCD} is the characteristic equation which provides the poles that govern the stability of the integration algorithm. The solution of the characteristic equation $z_{1,2}$ is dependent on the ratio k_t/k_0, and the location of the poles in the complex z-plane is dependent on the feedback gain k_t. If the poles associated for a specified k_t lie within the unit circle in the complex z-plane then the MCD is stable.

Consider an SDOF system with an initial elastic natural frequency associated with k_0 of $\omega_n=5\pi$, $m=1$, $c=0$, and $\Delta t=0.1$sec. The root loci of the G^{NL}_{MCD} transfer function is presented in Fig. 7 for three values of ρ_∞ for this SDOF, where the ratio k_t/k_0 was varied from 0 to 10 to generate the root loci. The poles $z_{1,2}$ are complex conjugate pairs with one of the branches of the root locus exiting the unit circle at $z=-1$ at a feedback gain of $k_t=3.62k_0$ for all three cases. Recall that instability occurs

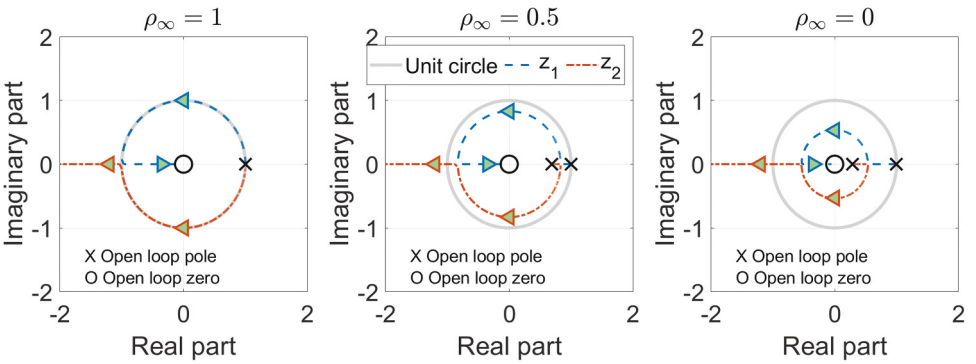

Figure 7. Root locus of an SDOF system for several ρ_∞ values.

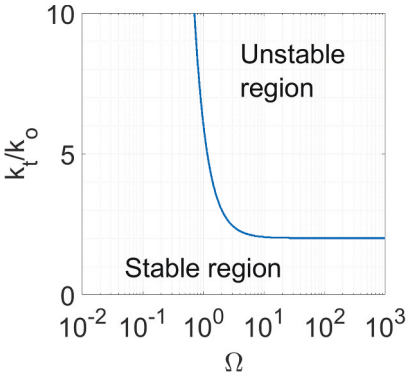

Figure 8. Stability region of the MCD algorithm for an SDOF system with $\omega_n = 5\pi$, m = 1, c = 0, and $\Delta t = 0.1$ sec.

when any of the branches exits the unit circle. Thus substituting $z = -1$ into the characteristic equation for G_{MCD}^{NL} and solving for k_t leads to

$$\frac{k_t}{k_0} \leq 2 + \frac{4}{\Omega^2} \tag{42}$$

The stability limit of the MCD algorithm expressed by Eq. (42) shows that the tangent stiffness can increase up to two times the initial stiffness used in determining the model parameters γ_1, γ_2 and γ_3, with an additional amount expressed by the second term in Eq. (42). Reducing the integration time step Δt is shown to increase the stability limit of the MCD algorithm. In addition, the amount of inherent damping or numerical damping does not influence the stability limit of the MCD algorithm, as it does not appear in Eq. (42). Eq. (42) is the same stability limit as that for the CR algorithm (Chen and Ricles 2008b). The MKR-α algorithm's stability limit is dependent on the numerical damping, where an increase in amount of numerical damping reduces the stability limit to be less than that of the MCD algorithm (Kolay and Ricles 2019).

The location of the poles in the complex z-plane was examined for Ω in the range of [0.01,1000] and k_t/k_0 in the range of [0, 10] where the resulting range of instability is determined using Eq. (42). The stability limit is shown in Fig. 8, where it is apparent that the lower bound value for k_t/k_0 is 2.0 and occurs when Ω exceeded a value of about 20. The contribution of the second term of Eq. (42) is small when the time step becomes larger. Figure 8 shows that reducing the size of the time step increases the stability limit and is associated with the second term in Eq. (42).

The above example can be applied to any SDOF, whereby the stability limit for k_t/k_0 is established. It can also be applied to multi-degree of freedom systems, as demonstrated by Chen and Ricles (2010), but requires expressing the proportionality of the tangent stiffnesses of each element of the structure to a common value in order to utilize the root locus. It is worth noting that demonstrating that the spectral radius of the amplification matrix or the poles of the transfer function are within the unit circle for nonlinear problems is necessary but not insufficient to guarantee unconditional stability for nonlinear problems (Liang and Mosalam 2016).

4.4. Consistency, Accuracy, and Convergence

This section investigates the consistency, accuracy, and convergence of the MCD algorithm. Applying Taylor Series expansion to the displacements $x_{t+\Delta t}$ and $x_{t-\Delta t}$ leads to

$$x_{t+\Delta t} = x_t + \left(\frac{dx}{dt}\right)\Delta t + \frac{1}{2}\left(\frac{d^2x}{dt^2}\right)\Delta t^2 + O(\Delta t^3), x_{t-\Delta t} = x_t - \left(\frac{dx}{dt}\right)\Delta t + \frac{1}{2}\left(\frac{d^2x}{dt^2}\right)\Delta t^2 + O(\Delta t^3) \tag{43a, b}$$

Eq. (43) along with γ_1, γ_2 and γ_3 defined by Eqs. (13) and (16) when substituted into Eqs. (9), where the notation i is replaced with t, i + 1 is replaced with t+Δt, and i-1 is replaced with *t*-Δt in Eq. (9), and then the result substituted into the equation of motion for an SDOF system, Eq. (6a), with no external force leads to the following expressions:

$$m\left(\frac{d^2x}{dt^2}\right) + c\left(\frac{dx}{dt}\right) + kx_t + E(\Delta t) = 0, E(\Delta t) = -k\left(\frac{1-\rho_\infty}{\rho_\infty+1}\right)\left(\frac{dx}{dt}\right)\Delta t + O(\Delta t^2), \lim_{\Delta t \to 0}(E(\Delta t)) = 0 \tag{44a, b, c}$$

The local truncation error $E(\Delta t)$ of the MCD algorithm is a function of the integration time step Δt. The MCD is therefore considered consistent because the error term $E(\Delta t)$ diminshes as the time step is decreased. The order of accuracy of the MCD algorithm is found to be second order when $\rho_\infty = 1$ (since the first term on the right-hand side of Eq. (44b) would equal zero) and first order accurate when $\rho_\infty < 1$.

A convergence study was performed of a SDOF system subjected to a unit initial velocity and displacement. The time step is systematically decreased and the displacement Root Mean Square Error (RMSE) between the MCD algorithm's solution and the analytical solution of the free vibration problem is evaluated. The results are plotted in Fig. 9 (where both axes are log-scale) for $\rho_\infty = 1$ and 0.5 for an undamped ($\xi = 0$) and damped ($\xi = 0.2$) SDOF system. The RMSE of the CD algorithm versus the analytical solution are also plotted for comparison. The slope of each curve is associated with the order of accuracy, which in turn defines the rate of convergence of the algorithm. In both plots, the CD algorithm and the MCD with $\rho_\infty = 1$ both have a slope of 2.0, while the MCD with $\rho_\infty = 0.5$ has a slope of 1.0. These results confirm the analytical proof of the MCD order of accuracy. Furthermore, the order of accuracy of the MCD algorithm is demonstrated to be independent of the inherent damping ξ of the system. Since the MCD algorithm is proven to be consistent and stable for linear systems, it is therefore concluded that the MCD is convergent according to the Lax Equivalence Theorem.

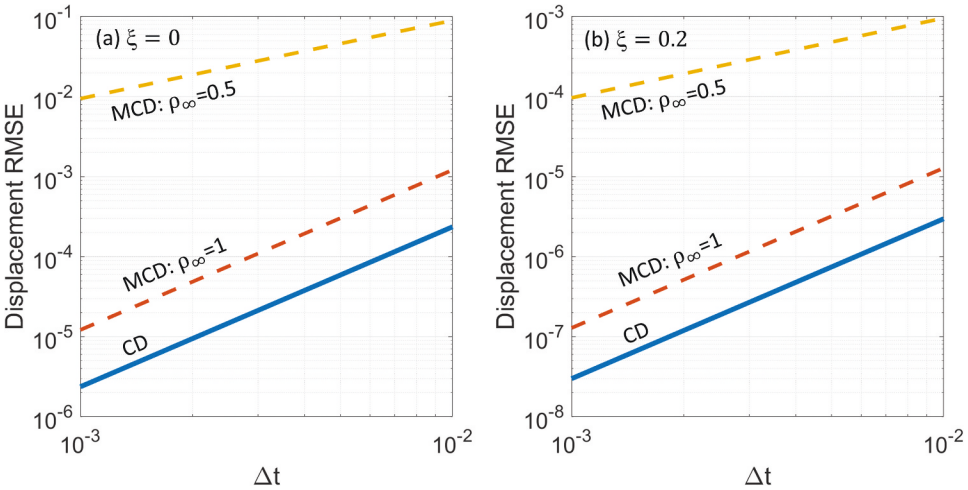

Figure 9. Rate of displacement convergence of the MCD and CD algorithms for an SDOF system subjected to $u_0 = 1, v_0 = 1$: (a) with; and, (b) without inherent damping, (k = 1, m = 1).

4.5. Overshoot Assessment

Overshoot is the tendency of an integration scheme to overestimate the exact solution of the initial value (i.e., free vibration) problem of the higher modes that are characterized by a large Ω value (Goudreau and Taylor 1973; Zhou and Tamma 2004). To assess the tendency of an integration algorithm to overshoot the exact solution, especially in the first few steps, one can determine the free vibration response of a SDOF system (Hilber and Hughes 1978). When $\Omega \to 0$, the MCD algorithm is convergent and there is no overshoot. On the other hand, $\Omega \to \infty$ gives an indication of the overshoot characteristics of the high-frequency modes present in a system. Although higher modes may have little contribution to the response in a structural dynamics problem, significant overshoot of the higher modes can influence the response of the lower modes of interest if non-classical damping is used.

The displacement $x_{i(MCD)}$ and velocity $v_{i(MCD)}$ of the MCD algorithm at time step i are obtained by applying the recurrence relationship of the MCD algorithm, Eq. (33), to the initial conditions and considering the value for x_{-1} from Eq. (32a). Applying $\boldsymbol{\xi} = 0$ to the result for the displacement and velocity as $\Omega \to \infty$ at the first-time step (i = 1) leads to

$$x_{1(MCD)} = \left(\frac{1-\rho_\infty}{2}\right)x_0 + \left(\frac{1+\rho_\infty}{4}\right)\Delta t v_0, \tag{45a}$$

$$v_{1(MCD)} = -\left(\frac{1+\rho_\infty}{\Delta t}\right)x_0 + \left(\frac{\rho_\infty - 1}{2}\right)v_0 \tag{45b}$$

where x_0 and v_0 are the initial displacement and initial velocity, respectively. The velocity of the first-time step (i = 1) in Eq. (45b) is calculated using the recurrence relationship and Eq. (9a). Eq. (45a) indicates that the displacement $x_{1(MCD)}$ is proportional to the initial displacement x_0 and linearly increases with respect to the integration time step Δt due to the initial velocity v_0. It is concluded from Eq. (45a) that the MCD overshoots the displacement of the first-time step linearly with respect to Ω when subjected to an initial velocity v_0. However, this is considered an improved overshoot response compared to the conventional CD algorithm response when Ω > 1, as shown in Eq. (46a). The inclusion of numerical damping alleviates the magnitude of the MCD displacement overshoot as noticed in Eq. (45a). The velocity $v_{1(MCD)}$ at the first-time step (i = 1) as given by Eq. (45b) indicates a linear increase with the inverse of the integration time step Δt due to an initial displacement x_0. A similar tendency is observed in the conventional CD algorithm as shown in Eq. (46b):

$$x_{1(CD)} = \left(1 - \frac{\Omega^2}{2}\right)x_0 + \Delta t v_0, v_{1(CD)} = \left(\frac{\Omega^4 - 4\Omega^2}{4\Delta t}\right)x_0 + \left(1 - \frac{\Omega^2}{2}\right)v_0 \tag{46a, b}$$

The displacement and velocity response of the MCD algorithm at time step i (with $\Omega \to \infty$, $\boldsymbol{\xi} = 0$, and $i \geq 1$) are shown in Eqs. (47) and (48):

$$\lim_{\Omega\to\infty}\left(x_{i(MCD)}\right) = \begin{cases} (-\rho_\infty)^{\frac{i-1}{2}}\left(\frac{1-\rho_\infty}{2}\right)x_0 + (-\rho_\infty)^{\frac{i-1}{2}}\left(\frac{1+\rho_\infty}{4}\right)\Delta t v_0, & \text{i odd} \\ (-\rho_\infty)^{\frac{i}{2}}x_0, & \text{i even} \end{cases} \tag{47}$$

$$\lim_{\Omega\to\infty}\left(v_{i(MCD)}\right) = \begin{cases} (-1)(-\rho_\infty)^{\frac{i-1}{2}}\left(\frac{1+\rho_\infty}{\Delta t}\right)x_0 + (-\rho_\infty)^{\frac{i-1}{2}}\left(\frac{\rho_\infty - 1}{2}\right)v_0, & \text{i odd} \\ (-\rho_\infty)^{\frac{i}{2}}v_0, & \text{i even} \end{cases} \tag{48}$$

Eqs. (47) and (48) are obtained using the following steps:

1. Substitute the velocity v_i and acceleration a_i from Eq. (9) into the equation of motion (Eq. (6a)) of a SDOF system subjected to an initial displacement x_0 and velocity v_0 without external loads (i.e., $f_i = 0$);
2. Solve for x_{i+1} and note the value for x_{-1} from Eq. (32a);
3. Solve for the velocity v_i from Eq. (9a);
4. Repeat Steps 2 and 3 recursively and take the limit of the computed values as $\Omega \rightarrow \infty$.

The displacement and velocity responses, $x_{i(MCD)}$ and $v_{i(MCD)}$, of the MCD algorithm shown in Eqs. (47) and (48) are bounded and do not grow with each time step. The inclusion of numerical damping (i.e., where $\rho_\infty < 1$) accelerates the rate of decay of the response, $x_{i(MCD)}$ and $v_{i(MCD)}$, since all terms in Eqss. (47) and (48) are multiplied by ρ_∞ raised to a power that includes the time step number i. The general displacement and velocity of the CD algorithm with $\Omega \rightarrow \infty$ are not shown here because the CD algorithm is only conditionally stable. Further discussion about the overshoot characteristics of the MCD are given below in the example where the algorithm is applied to the free vibration of an SDOF system. It should be noted that the overshoot characteristics of the MCD are influenced by γ_3, where by including it in the development of the MCD algorithm not only enables unconditional stability but also results in improved overshooting characteristics.

5. Implementation and Example Usage of the MCD Algorithm

The implementation of the MCD to numerically integrate the equations of motion for a RTHS is summarized in Table 4. The model parameters and the recurrence relation of the displacement are presented in terms of the model properties of the system, namely the initial stiffness matrix $\mathbf{K}_o$, mass matrix $\mathbf{M}$, and viscous damping matrix $\mathbf{C}$. The displacement vector $\mathbf{x}_{i+1}$ is obtained by substituting the acceleration and velocity vectors from Eq. (17) into the equations of motion, Eq. (18), and solving for the displacement $\mathbf{x}_{i+1}$. The restoring force vector in Table 4 is denoted as $\mathbf{R}_i$ and the applied load vector is $\mathbf{F}_i$. The integration parameters need to be initialized only once at the beginning of the

Table 4. Implementation of the MCD algorithm for integration of the equations of motion.

Step 1	Select time step and ρ_∞, initialize initial acceleration, integration parameters, and use special starting procedure: $\mathbf{a}_0 = \mathbf{M}^{-1}(\mathbf{F}_0 - \mathbf{C}\mathbf{v}_0 - \mathbf{K}_o\mathbf{x}_0)$ $\gamma_1 = ((\rho_\infty + 1)(\mathbf{K}_o\Delta t^2 + 2\mathbf{C}\Delta t + 4\mathbf{M}))^{-1}((\rho_\infty - 3)\mathbf{K}_o\Delta t^2)$ $\gamma_2 = ((\rho_\infty + 1)(-\mathbf{K}_o\Delta t^2 + 2\mathbf{C}\Delta t - 4\mathbf{M}))^{-1}((3\rho_\infty - 1)\mathbf{K}_o\Delta t^2)$ $\gamma_3 = (\mathbf{K}_o\Delta t^2 + 4\mathbf{M})^{-1}4\mathbf{M}$ $\mathbf{x}_{-1} = \mathbf{x}_0 + \mathbf{G}\mathbf{v}_0 + \mathbf{H}\mathbf{a}_0$ where, $\mathbf{I}$ which appears below is the identity matrix $\mathbf{G} = (2\Delta t)Z,$ $\mathbf{H} = (-\Delta t^2)Z\gamma_3,$ $Z = (2(\gamma_2 - \mathbf{I}))^{-1}$
Step 2	Select value for ρ_∞ and initialize Ψ, Ψ_1, Ψ_2, and Ψ_3 where, $\Psi = 2\mathbf{M}(\rho_\infty + 1) + \mathbf{C}\Delta t(\rho_\infty + 1) + 2\mathbf{K}_o\Delta t^2$ $\Psi_1 = (-2\mathbf{M}(\rho_\infty + 1) + \mathbf{C}\Delta t(\rho_\infty + 1) - 2\rho_\infty\mathbf{K}_o\Delta t^2),$ $\Psi_2 = (4\mathbf{M}(\rho_\infty + 1) + 2\mathbf{K}_o\Delta t^2(\rho_\infty + 1)),$ $\Psi_3 = (2\Delta t^2(\rho_\infty + 1))$
Step 3	For time step i, determine the displacement vector $\mathbf{x}_{i+1}$ $\Psi\mathbf{x}_{i+1} = (\Psi_1\mathbf{x}_{i-1} + \Psi_2\mathbf{x}_i + \Psi_3(\mathbf{F}_i - \mathbf{R}_i))$ where $\mathbf{R}_i$ is the restoring force vector at time step i obtained from the state determination of the element internal forces, and $\mathbf{F}_i$ is the external load vector
Step 4 (Optional)	Determine the velocity and acceleration vectors at time step i $\mathbf{v}_i = (2\Delta t)^{-1}((\mathbf{I} - \gamma_1)\mathbf{x}_{i+1} + \gamma_1\mathbf{x}_i - (\mathbf{I} - \gamma_2)\mathbf{x}_{i-1} - \gamma_2\mathbf{x}_i)$ $\mathbf{a}_i = (\Delta t^2\gamma_3)^{-1}((\mathbf{I} - \gamma_1)\mathbf{x}_{i+1} + \gamma_1\mathbf{x}_i - 2\mathbf{x}_i + (\mathbf{I} - \gamma_2)\mathbf{x}_{i-1} + \gamma_2\mathbf{x}_i)$
Step 5	Continue to the next integration time step: replace i with $i + 1$ and repeat Steps 3 through 4

analysis. Step 4 is optional and only required if the velocities $\mathbf{v}_i$ and accelerations $\mathbf{a}_i$ are values of interest since they are not used to compute $\mathbf{x}_{i+1}$.

The computational efficiency of the MCD algorithm depends on the sparsity of the **M**, **C**, and $\mathbf{K_o}$ matrices. While a lumped mass **M** and the initial stiffness $\mathbf{K_o}$ matrices are often sparse, the sparsity of the damping matrix **C** depends on the choice of damping model. If Rayleigh damping is used as a linear combination of a lumped mass and initial stiffness matrices, then the MCD algorithm is computationally efficient because the damping matrix **C** will be sparse and leads to a linear system of sparse equations. If the damping matrix **C** is formulated based on the superposition of modal damping ratios, then the damping matrix is dense when a consistent mass matrix is used (Chopra 2012). The MCD algorithm is not as computational efficient when the damping matrix **C** is dense.

For large structural dynamics problems where **M**, $\mathbf{K_o}$, and **C** are sparse, $\mathbf{\Psi}$, $\mathbf{\Psi}_1$, and $\mathbf{\Psi}_2$ are also sparse, leading to a linear system of sparse equations that can be solved efficiently during a RTHS where matrix operations are performed only on non-zero elements. The matrix $\mathbf{\Psi}$ would need to be decomposed into lower and upper triangular matrices to efficiently solve for the $\mathbf{x}_{i+1}$ vector in Table 4. Additional computational effort is required to compute the velocities and accelerations. However, unlike other RTHS algorithms that include the MKR-α algorithm, the computational of the velocities and accelerations are not required to be computed in the MCD algorithm in each time step during a RTHS and can be computed afterwards. This represents additional computational saving of the MCD algorithm over other RTHS integration algorithms.

When using the MCD algorithm, the linearized stiffness k and damping c of the analytical and experimental substructures need to be included in the system's initial stiffness matrix $\mathbf{K}_o$ and viscous damping matrix **C**, respectively, in order to determine the algorithm's integration parameters $\boldsymbol{\gamma}_1$, $\boldsymbol{\gamma}_2$, and $\boldsymbol{\gamma}_3$ (see Step 1 in Table 4) and $\mathbf{\Psi}$, $\mathbf{\Psi}_1$, $\mathbf{\Psi}_2$ in the recursive relationship (Step 4 in Table 4). An example of where nonlinear viscous dampers existed in the experimental substructure of a RTHS is given in Kolay et al. (2018), in which k and c for the nonlinear viscous damper were based on the equivalent Kelvin-Voigt model of a linearized Maxwell model. The determination of the restoring forces at time step i of any numerically-modeled velocity-dependent device (e.g. nonlinear viscous damper) that contributes to the structural system's restoring force vector $\mathbf{R}_i$ requires the velocity vector $\mathbf{v}_i$. Since $\mathbf{v}_i$ is not yet readily available until Step 4 of the algorithm (see Table 4) it is recommended to use the backward finite difference to determine the velocity in a device associated with its DOF at each time step in order to avoid iterating.

Four examples that illustrate the use of the MCD algorithm follow below. These examples illustrate the overshoot characteristics of the MCD algorithm associated with initial conditions (Example 1), the dissipative characteristics of the MCD algorithm (Example 2), the application and efficiency of the MCD algorithm when solving large dynamic analysis problems involving either linear or nonlinear response (Example 3), and the use of the MCD algorithm for conducting multi-natural hazard 3D RTHS of a tall building with nonlinear response (Example 4).

5.1. Example 1: Free Vibration of an SDOF System

This example illustrates the overshoot characteristics of the MCD algorithm associated with initial conditions. An SDOF system with $\xi = 0$ is subjected to two sets of initial conditions: (1) initial displacement $x_0 = 1meter$, initial velocity $v_0 = 0$; and (2) initial displacement $x_0 = 0$, initial velocity $v_0\Delta t = 1$ meter. The mass m and stiffness k of the SDOF are equal to 0.01 kN-s^2/m and 1 kN/m, respectively. The time step Δt is chosen such that $\Omega = 20\pi$. Computations are done for cases without ($\rho_\infty = 1$) and with numerical damping ($\rho_\infty = 0.75, 0.5, 0.25, 0$). The comparison with the conventional CD algorithm is not given herein because the CD algorithm is unstable under the selected time step. Instead, the MCD results are compared to the analytical solution via the time history of

normalized displacement ($x(t)/A^A$), velocity ($v(t)/\omega A^A$) and the normalized total energy ($E(t)/E_A$), where A^A and E_A are the exact displacement amplitude and total energy of the solution to the initial value problem of the homogeneous ODE, where

$$A^A = \sqrt{(x_0)^2 + \left(\frac{v_0 + \xi\omega x_0}{\omega_D}\right)^2}, \frac{E(t)}{E_A} = \frac{(\Delta t v(t))^2 + (\Omega x(t))^2}{(\Delta t v_0)^2 + (\Omega x_0)^2} \tag{49a, b}$$

Because of the normalization of the responses, $x(t)/A^A > 1$, $v(t)/\omega A^A > 1$, and $(E(t)/E_A) > 1$ indicates overshoot in displacement, velocity, and total energy, respectively.

The normalized displacement response under the first set of initial conditions $x_0 = 1, v_0 = 0$ is shown in Fig. 10. The MCD algorithm results are shown to be bounded by the exact solution without overshoot. Numerical damping is shown to damp out the response and reduce the displacement amplitude with each integration time step. The velocity response based on Eq. (9a) never overshoots the exact answer, and numerical damping decreases the amplitude quickly. The total energy is also shown to not overshoot. The trend in the MCD algorithm results is consistent with the closed form expressions for the displacement and velocity given by Eqs. (47) and (48) where the response remains bounded shows decay over time with smaller values of ρ_∞.

For the second set of initial conditions the MCD algorithm exhibits displacement overshoot but the response remains bounded as shown in Fig. 11. Incorporating numerical damping for the MCD algorithm not only accelerates the decay of the response, but also reduces the magnitude of overshoot for the first-time step. There is no overshoot in the velocity response as indicated in Fig. 11. The MCD overestimates the energy if an initial velocity is present (which as discussed above causes a displacement overshoot), while including numerical damping alleviates this overestimation. The trend in the MCD algorithm results are again consistent with the closed form expressions for the displacement and velocity given in Eqs. (47) and (48) where the response remains bounded and is reduced with smaller values of ρ_∞.

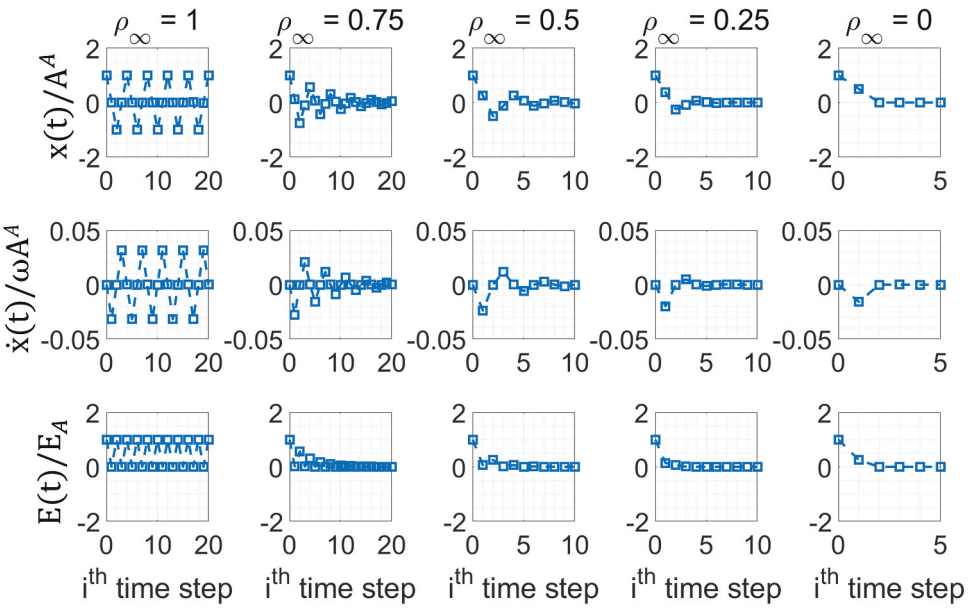

Figure 10. Free vibration response of SDOF system with $\Omega = 20\pi$ subjected to initial conditions $x_0 = 1, v_0 = 0$.

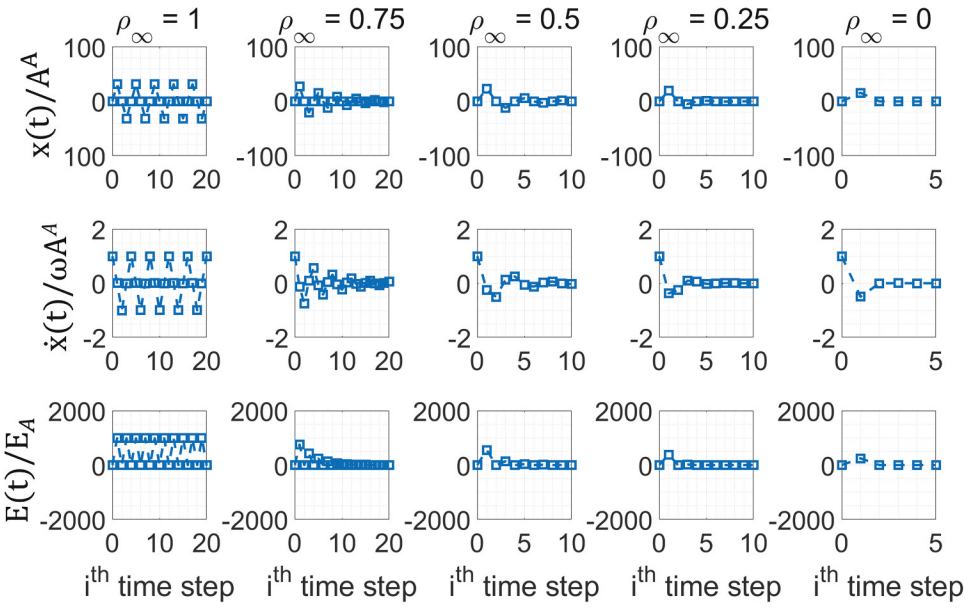

Figure 11. Free vibration response of SDOF system with $\Omega = 20\pi$ subjected to initial conditions $x_0 = 0, v_0\Delta t = 1$.

5.2. Example 2: Free Vibration of an MDOF System

This example illustrates the dissipative characteristics of the MCD algorithm of the free vibration response of an undamped two degrees of freedom system subjected to initial displacements. The properties of the system include

$$M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, K = \begin{bmatrix} 8100 & -8000 \\ -8000 & 8000 \end{bmatrix}, C = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{q}_1 = \begin{bmatrix} 0.746 \\ 0.751 \end{bmatrix}, \mathbf{q}_2 = \begin{bmatrix} 0.254 \\ -0.251 \end{bmatrix}, \text{where } \mathbf{q}_n = \phi_n \frac{\phi_n^T M u_0}{\phi_n^T M \phi_n} \tag{50}$$

The natural frequencies ω_n of the system are equal to 7.06 and 126.69 rad/sec for the 1st and 2nd modes of vibration, respectively. The modal expansion of the initial displacement vector $\mathbf{x}_0$ into $\mathbf{q}_1$ and $\mathbf{q}_2$ associated with the two modes are given in Eq. (50), where ϕ_n is the eigenvector for mode n. The system is subjected to the initial displacement $\mathbf{x}_0 = [1, 0.5]^T$, which contains a contribution from both modes of vibration. The second mode is intentionally set to have a high frequency to demonstrate the ability of the numerical damping to suppress its contribution to the response. The integration time step is chosen to be $\Delta t = 0.001$ sec. The first and second modes have $\Omega = 0.007$ and $\Omega = 0.126$, respectively.

The response of the CD and MCD algorithms with ($\rho_\infty = 0.5$) and without numerical damping ($\rho_\infty = 1$) is shown in Fig. 12. The exact solution shown plotted in Fig. 12 is obtained by solving the initial value problem involving the homogenous ODE. The solutions using the CD and the MCD algorithms coincide with the exact solution when $\rho_\infty = 1$ and the high frequency response of the second mode is prevalent in the displacement. Numerical damping is shown to suppress the contribution of the second mode to the response based on the MCD algorithm when $\rho_\infty = 0.5$. Modes 1 and 2 have a $\bar{\xi}$ of 0.1% and 2%, respectively, and 0.001% and 0.3% period elongation, respectively, with $\rho_\infty = 0.5$. Reducing the time step can reduce this period distortion because the MCD is proven to be convergent.

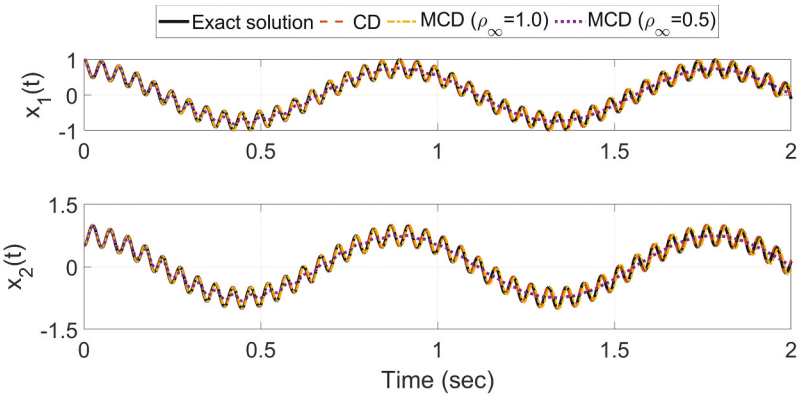

Figure 12. Free vibration response 2-DOF system.

5.3. Example 3: 2D Tall Building Subjected to Wind and Earthquake Natural Hazards

This example demonstrates the application and efficiency of the MCD algorithm when solving large dynamic analysis problems involving either linear or nonlinear response. The efficiency will be assessed by comparing the total central processing unit (CPU) time to complete the analysis using the MCD algorithm with the MKR-α and CCA algorithms, which are described below. The example consists of two analyses of a 40-story tall building that includes a strong windstorm and a strong earthquake. The building is taken from one of the designs of California Tall Building Initiative (Moehle et al. 2011). The lateral force resisting system of the building consists of six buckling-restrained braced frames (BRBFs) in the N-S and E-W directions and six outrigger trusses in the N-S direction as depicted in Fig. 13a. The columns of the BRBFs are composed of steel tubes filled with high strength concrete, while the outrigger columns consist of wide flange steel sections ranging from W14 × 283 at the 40th story to W14×455at the 1st story. The BRBFs have W16 × 100 steel beams that are attached to the columns via shear connections (i.e., no moment transfer). The buckling restrained braces (BRBs) of the BRBFs have a cross sectional area ranging from 0.011 m^2 at the 1st story level to 0.007 m^2 at the 40th story level, and are constructed using 262 MPa steel with a yieldable core over 70% of the brace length.

The building is modeled in 2D using HybridFEM-MH (Kolay, Marullo, and Ricles 2018), a computer program capable of performing nonlinear time history analysis of structures subjected to wind or

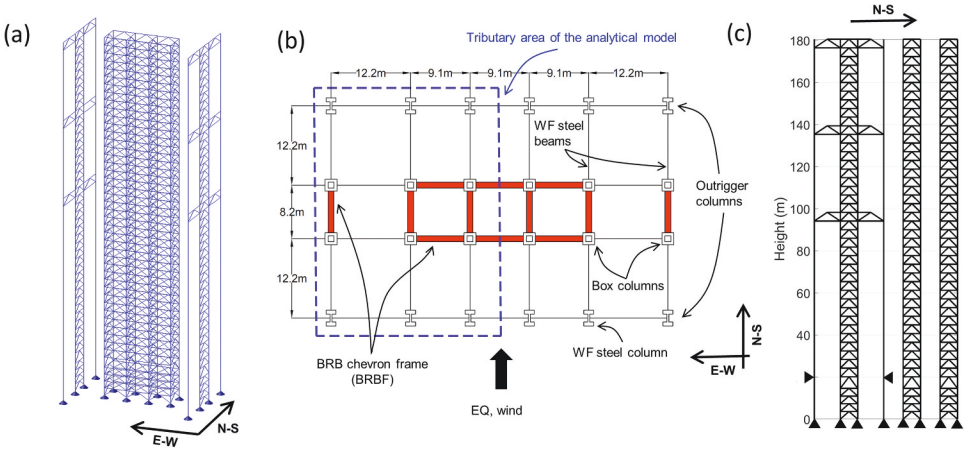

Figure 13. Tall building prototype per Moehle et al. (2011): (a) 3D finite element model of the tall building used in 3D RTHS in example 4; (b) typical floor plan; and, (c) planar 2D finite element model of the building used in example 3.

earthquake loading. A 3D RTHS of this structure was performed and is discussed in Example 4. The structure in this example is subjected to loading in the N-S direction, and hence symmetry is implemented to model only half of the tributary area of the floor in the N-S direction. The columns and beams are modeled using beam-column elements where the moment at the ends of the beam are released because the beam-to-column connections transfer shear and axial forces only. The outrigger columns are modeled using beam-column elements. A lean-on column is used to account for P-delta effects. The diagonal brace members are modeled using truss elements. The Giuffre-Menegotto-Pinto formulation is used to define the stress-strain curve to capture nonlinear behavior in the elements, where the final tangent modulus is 0.028 of the initial elastic modulus. A master node with an equal displacement constraint is applied to each floor of the BRBFs to simulate rigid floor diaphragm action of the building floor. The mass of the gravity load system is based on the dead load and lumped at the master node of each floor level, where the master node is located at the middle of the left-hand BRBF bay shown in Fig. 13c. The HybridFEM-MH model, shown in Fig. 13c, is a planar model that contains 1590 degrees of freedom and 902 elements. Complete details of the building design, member sizes, and the live and dead loads can be found in Moehle et al. (2011). The building was assigned 2% damping for modes 1 through 10 and stiffness proportional damping for modes 11 and beyond. A consistent mass matrix was used to model the mass of the members of the lateral load resisting frames (i.e., BRBFs and outriggers) and therefore the damping matrix is dense.

In the first analysis, the building is subjected to a 177 km/h wind loading having a 700-year Mean Recurrence Interval (MRI). The wind profile is based on the wind tunnel data available from the Aerodynamic Database of High-Rise Buildings of Tokyo Polytechnic University (Tokyo Polytechnic University 2019). In the second analysis the building is subjected to the Northridge earthquake, using the RSN1084_NORTHR_SCS052 component (PEER 2019). This record was scaled to the target uniform hazard spectrum for the MCE hazard level (2,475-year return period) over a period range of interest using the procedure outlined in the Tall Building Initiative of this same building by Moehle et al. (2011). The site of the building is Los Angeles. In the scaling procedure the scale factor for a pair of records is determined by minimizing the weighted sum of square errors between the target spectrum and the geometric mean spectrum of the pair over the period range of 0.5 to 10 s with an interval of 0.1 s. As given in the TBI case studies, the errors in the period ranges of 0.5–3.0, 3.0–7.0 and 7.0–10.0 sec were weighted 10%, 60%, and 30%, respectively. The same ground motion record was also used by SGH in the performance evaluation of the prototype building (Moehle et al. 2011).

It is impractical to solve this problem with the conventional CD algorithm, for the structure has high natural frequencies and therefore the required integration time step to guarantee a stable solution is unrealistically small. Instead, the solution of the MCD is compared to Newmark Constant Average Acceleration (CAA) (Newmark 1959) and MKR-α algorithms (Kolay and Ricles 2019), where the latter is another unconditionally stable explicit model-based algorithm that possesses controlled numerical damping. The details of the CAA algorithm for linear and nonlinear problems are presented in (Chopra 2012) and not repeated herein. The damping matrix **C** is dense; therefore, the CAA formulation involves the full matrix multiplication for both linear and nonlinear problems, respectively (Chopra 2012).

The MKR-α algorithm solves the following weighted equations of motion:

$$\mathbf{M}\ddot{\mathbf{x}}_{i+1-\alpha_3} + \mathbf{C}\dot{\mathbf{x}}_{i+1-\alpha_f} + \mathbf{K}\dot{\mathbf{x}}_{i+1-\alpha_f} = \mathbf{F}_{i+1-\alpha_f} \tag{51}$$

The MKR-α algorithm uses the forward difference approximation of the derivative where the displacement, velocity, and acceleration are obtained using (Kolay and Ricles 2019; Kolay et al. 2015):

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \Delta\dot{\mathbf{x}}_i + \left(\tfrac{1}{2}+\gamma\right)\Delta t\widehat{\ddot{\mathbf{x}}}_i, \dot{\mathbf{x}}_{i+1} = \dot{\mathbf{x}}_i + \widehat{\ddot{\mathbf{x}}}_i, \ddot{\mathbf{x}}_{i+1} = \mathbf{D}\widehat{\ddot{\mathbf{x}}}_{i+1} \tag{52a, b, c}$$

$$\widehat{\ddot{\mathbf{x}}}_{i+1} = \mathbf{A}\left(\mathbf{F}_{i+1-\alpha_f} - \mathbf{F}_{ID_{i+1-\alpha_f}} - \mathbf{R}_{i+1-\alpha_f} - \mathbf{B}\widehat{\ddot{\mathbf{x}}}_i\right), \mathbf{F}_{i+1-\alpha_f} = \alpha_f\mathbf{F}_{i+1} + (1-\alpha_f)\mathbf{F}_i,$$
$$\mathbf{F}_{ID_{i+1-\alpha_f}} = \mathbf{C}\left(\dot{\mathbf{x}}_i + (1-\alpha_f)\widehat{\ddot{\mathbf{x}}}_i\right) \tag{53a, b, c}$$

$$\mathbf{F}_{\mathrm{ID}_{i+1-\alpha_f}} = \mathbf{C}\left(\dot{\mathbf{x}}_i + (1-\alpha_f)\widehat{\dot{\mathbf{x}}}_i\right), \mathbf{R}_{i+1} = \alpha_f \mathbf{R}_{i+1} + (1-\alpha_f)\mathbf{R}_i \tag{53d, e}$$

where $\mathbf{F}_{i+1-\alpha_f}$, $\mathbf{F}_{\mathrm{ID}_{i+1-\alpha_f}}$, and $\mathbf{R}_{i+1-\alpha_f}$ are the weighted applied load vector, damping force vector, and restoring force vector, respectively, and

$$\mathbf{A} = \Delta t \boldsymbol{\alpha}_1 (\mathbf{M} - \mathbf{M}\boldsymbol{\alpha}_3)^{-1}, \mathbf{B} = \frac{1}{\Delta t}\mathbf{M}\boldsymbol{\alpha}_3\boldsymbol{\alpha}_1^{-1}, \mathbf{D} = \frac{1}{\Delta t}\boldsymbol{\alpha}_1^{-1} \tag{54a, b, c}$$

The model parameters are calculated as follows:

$$\boldsymbol{\alpha}_1 = \left(\mathbf{M} + \gamma\Delta t\mathbf{C} + \beta\Delta t^2\mathbf{K_o}\right)^{-1}\mathbf{M}, \quad \boldsymbol{\alpha}_2 = \left(\frac{1}{2} + \gamma\right)\boldsymbol{\alpha}_1,$$
$$\boldsymbol{\alpha}_3 = \left(\mathbf{M} + \gamma\Delta t\mathbf{C} + \beta\Delta t^2\mathbf{K_o}\right)^{-1}\left(\alpha_m\mathbf{M} + \alpha_f\gamma\Delta t\mathbf{C} + \alpha_f\beta\Delta t^2\mathbf{K_o}\right) \tag{55a, b, c}$$

where

$$\beta = \frac{1}{2}\left(\gamma + \frac{1}{2}\right), \gamma = \frac{1}{2} - \alpha_m + \alpha_f, \alpha_f = \frac{\rho_\infty}{\rho_\infty + 1}, \alpha_m = \frac{2\rho_\infty^3 + \rho_\infty^2 - 1}{\rho_\infty^3 + \rho_\infty^2 + \rho_\infty + 1} \tag{56a, b, c, d}$$

A and **B** are often dense matrices, making the reoccurrence relationship of the MKR-α algorithm expressed by Eq. (53a requiring more computational effort than the recurrence relationship for the MCD algorithm (Step 3 of Table 4).

The formulation of the CAA algorithm is documented in (Chopra 2012; Newmark 1959) and therefore not presented here. The Newton Raphson method (i.e., tangent stiffness iteration) with a convergence tolerance of $||\mathbf{R}^u||/\max|\mathbf{R}^u| \leq 0.001$ is used to check convergence, where $||.||$ represents the Euclidian norm of a vector and $\mathbf{R}^u$ is the unbalanced force vector; $\max|\mathbf{R}^u|$ is the maximum absolute unbalanced force in $\mathbf{R}^u$ during the first cycle of iteration. The integration time step is chosen to be 6/1024 seconds with $\rho_\infty = 0.86$ for the MCD and MKR-α algorithms.

The building response is linear under the wind loading and nonlinear under earthquake loading. Figure 14a–c show the calculated lateral displacement at the 40th, 25th, and 15th floors under the wind excitation, with the wind loads at these floor levels shown in Fig. 14d. The response based on the CAA, MKR-α, and MCD algorithms is essentially identical. Using the response based on the CAA as the reference solution, the MKR-α and MCD algorithms produced Normalized Mean Square Error

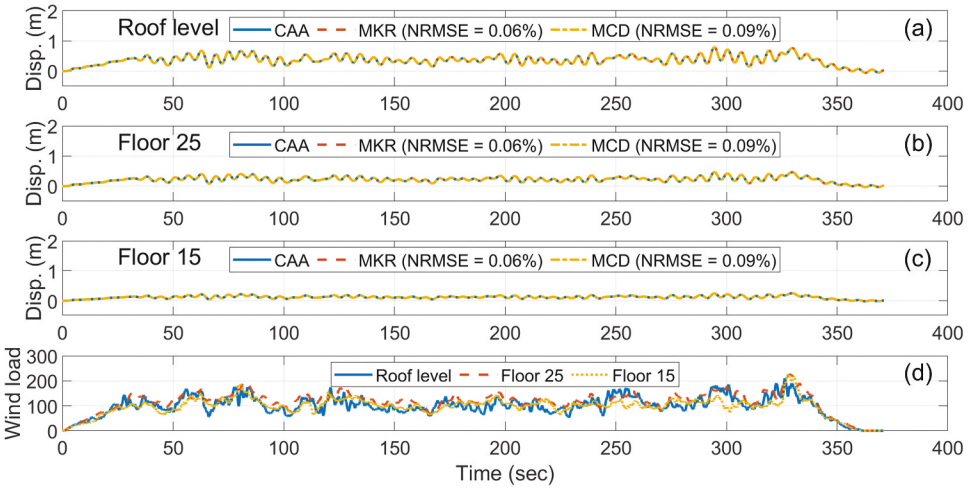

Figure 14. Comparison of the CAA, MKR-α, and MCD algorithms under wind loading: (a, b, and c) roof level, 25th, and 15th floor displacements response; and, (d) wind loads at selected floor levels.

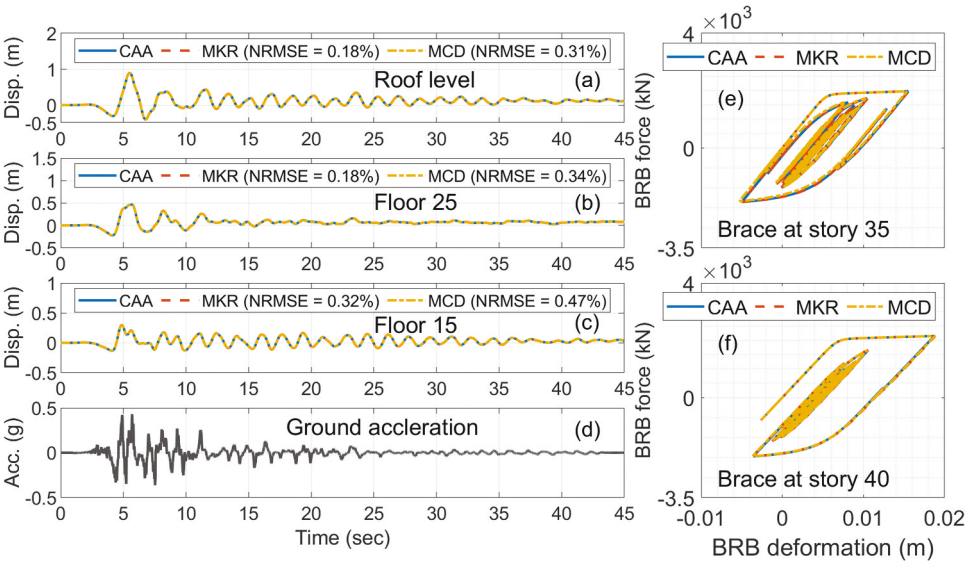

Figure 15. Comparison of the CAA, MKR-α, and MCD algorithms under seismic loading: (a, b, and c) roof level, 25th, and 15th floor displacements response; (d) ground acceleration; and, (e, f) brace axial force-deformation hysteresis at stories 35 and 40.

(NRMSE) values of 0.06% and 0.09%, respectively, for displacement at these selected floors. Other floors had results for the NRMSE that were less than or equal to these results. The NRMSE is defined as

$$\mathrm{NRMSE} = \frac{\sqrt{\frac{1}{N}\sum_{1}^{N}\left(x^{p}-x^{r}\right)^{2}}}{\max(\mathbf{x}^{r})-\min(\mathbf{x}^{r})} \tag{57}$$

where the x^p is the predicted solution, while x^r is the reference solution (i.e., the CAA solution). The building re-centered by the end of the windstorm as shown in Fig. 14, as it responded linearly to the event.

Figure 15a–c show the calculated lateral displacement of the 40th, 25th and 15th floors under the earthquake. The scaled ground accelerations are shown in Fig. 15d. Good agreement is shown to exist between the CAA, MKR-α, and MCD results, where the NRMSE is 0.18%, 0.18%, and 0.32% for the 40th, 25th, and 15th floors, respectively, for the MKR-α algorithm, and 0.31%, 0.34%, and 0.47%, respectively, for the MCD algorithm. Again, the response based on the CAA is considered the reference solution when calculating the NRMSE values. Other floors had results for the NRMSE where the maximum NRMSE was less than or equal to the above values. The hysteretic axial force-deformation response of a BRB at the 35th and 40th stories is shown in Figure 15e,f, where good agreement between the solutions using the CAA, MKR-α, and MCD algorithms can be seen. It is not possible to compute an NRMSE for the BRB force response since the associated BRB deformations obtained from the different integration methods do not coincide with the respective time steps for the CAA algorithm. Therefore, NRMSE values are not given. The degree of nonlinearity in these members is evident in this figure. Similar response occurred in the BRBs at other floors.

Table 5. Comparison of CPU time required for tall building analysis.

		Normalized time		NRMSE Roof Displacements	
Integration algorithm	Formulation type	Wind Analysis (linear elastic response)	Seismic Analysis (nonlinear response)	Wind	Earthquake
CAA	Implicit	1.96	2.85	–	–
MKR-α	Explicit	1.55	1.06	0.06%	0.18%
MCD	Explicit	1.0	1.0	0.09%	0.31%

Shown in Table 5 is the CPU time needed to obtain the solution for the CAA, MKR-α, and MCD algorithms for the analyses involving the two natural hazards. All cases in Table 5 were executed on the same Intel i9-9900X CPU having ten cores and 4.4 GHz maximum clock speed. For purposes of comparison the CPU times have been normalized by that of the MCD algorithm. The computational efficiency of the MCD algorithm is apparent under wind loading where linear response occurred, with CPU reductions by a factor of 1.96 and 1.55 compared to the CAA and MKR-α algorithms, respectively. For analysis of linear problems, the restoring force is obtained by multiplying the global structural stiffness by the displacement in lieu of a state determination process applied to each element in the model. For the earthquake loading, the MCD enables reductions by a factor of 2.85 and 1.06 in the computational time compared to the CAA and MKR-α algorithms, respectively. The latter comparison demonstrated smaller reductions versus the MKR-α algorithm because the CPU time for the seismic case is dominated by the time required to compute the nonlinear restoring forces $\mathbf{R}_i$. The NRMSE for the roof displacements for the wind and earthquake RTHS are also given in Table 5, where for the MCD algorithm they are shown to be equal to 0.09% and 0.31% for the wind and earthquake RTHS, respectively. The NRMSE values for the MKR-α algorithm are slightly smaller, being equal to 0.06% and 0.18% for the wind and earthquake RTHS, respectively. Although the MCD algorithm's NRMSE values are slightly larger compared to those for the MKR-α algorithm, the MCD NRMSE values are exceptionally small and provide evidence that the MCD algorithm gives accurate results. Overall, the MCD algorithm is shown in Table 5 to be accurate and more efficient than the CAA and MKR-α algorithms and therefore its use in solving structural dynamics problems is warranted.

This example uses a consistent mass matrix; therefore, the damping matrix $\mathbf{C}$ based on the superposition of modal damping ratios is dense. The recurrence relations of the CAA, MCD and MKR-α integration algorithms are therefore not sparse. The only exception is in the MCD algorithm where the parameter $\mathbf{\Psi}_2$ is not a function of the damping matrix $\mathbf{C}$ and the multiplication of the matrix $\mathbf{\Psi}_2$ and the vector $\mathbf{x}_i$ in Table 4 was performed only on the nonzero elements when computing the CPU time in Table 5. Hence, an additional reduction in the computational time required for the MCD algorithm could be achieved when a lumped mass matrix is used, for the resulting damping matrix would become less dense. The reason for using the consistent mass matrix in the above example is because the MRK-α algorithm requires the use of a consistent mass matrix, and hence the comparisons between the algorithms are based on using the same mass matrix formulation.

5.4. Example 4: 3D RTHS of a Tall Building Subjected to Multi-Natural Hazards

The computational efficiency of the MCD algorithm makes it well suited for performing 3D RTHS, where an increased number of DOF in 3D models of a system exists that requires more computational effort. The integration of the equations of motion must be completed within the time step of a RTHS, where time steps of adequate size (i.e., not too large) are required to ensure that the algorithm's results are accurate. Hence, computational efficiency of the integration algorithm is of utmost importance to enable 3D RTHS with complex models to be performed. The following example illustrates the use of the MCD algorithm to integrate the equations of motion during a 3D RTHS of the 40-story building discussed above. The 3D model shown in Fig. 13a with a lumped mass matrix RTHS are used. In the model each member of the structure is modeled, with each node having six DOF. Three 3-D RTHS were performed, each with a different value for ρ_∞ $(1.0, 0.86, \text{and } 0)$ to illustrate the effect of ρ_∞ on the results. The sparsity of the mass $\mathbf{M}$, damping $\mathbf{C}$, and intial stifness $\mathbf{K_o}$ associated with the equations of motion is exploited in the RTHS as described below. The building is modeled in HyCoM-3D (Ricles, Kolay, and Marullo 2020), a computer program capable of performing 3-D nonlinear RTHS of structures subjected to wind or earthquake loading. Only earthquake response is presented in this example. The building is supplemented with nonlinear viscous dampers between the outriggers and perimeter columns to improve building performance (Smith and Willford 2007). A full scale nonlinear viscous damper is used as the experimental substructure of the RTHS as shown in Fig. 16a. The

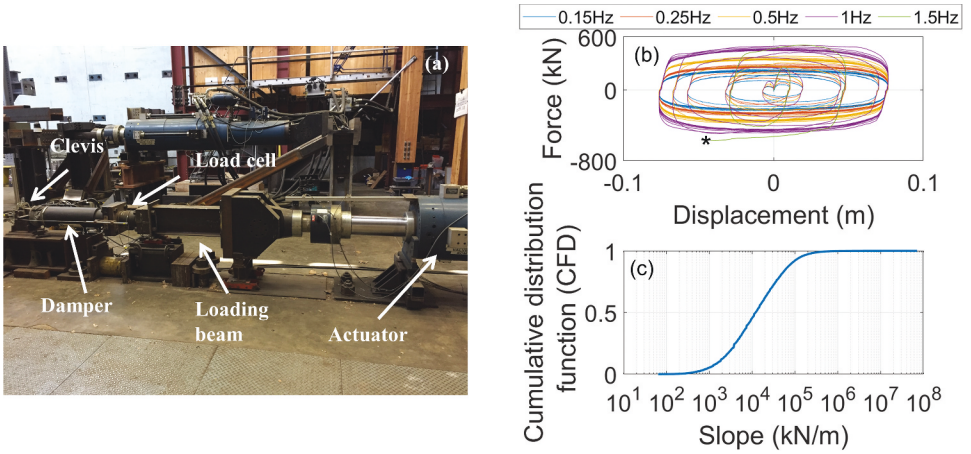

Figure 16. Full scale nonlinear viscous damper: (a) test setup for RTHS experimental substructure and damper characterization tests; (b) damper force-displacement response from characterization tests; and, (c) cumulative distribution function of the force-displacement slope from the characterization tests. (*damper force limit is reached at 1.5Hz).

remaining parts of the structure are modeled in the analytical substructure. The three-dimensional model of the building has the same element types used in Section 5.3; however, it includes a larger number of degrees of freedom (3,974), 1,828 nonlinear truss elements and 1,079 linear beam-column elements. The MKR-α algorithm cannot be used for this RTHS because the matrices are too large and dense (e.g. matrix **A** in Eq. 53a) that prevent the recursive relationship, Eq. (53a), from being calculated in real time within the time step. Note that the CR algorithm for RTHS (Chen and Ricles 2008a) is among the same family of algorithms as the MKR-α algorithm and equivalent when $\rho_\infty = 1.0$ is used. The CR algorithm lacks the dissipative characteristics necessary for a RTHS integration algorithm. Only the MCD algorithm can handle an analytical substructure model with these many DOFs and perform a RTHS.

The MKR-α algorithm requires statically condensing the column and beam elements in to order to reduce the NDOF in the model to 1680, limiting the ability for the model to capture any nonlinear behavior that occurs in the beams or columns of the building during a RTHS. The CAA algorithm cannot be used in a real-time hybrid simulation for it is an implicit integration algorithm which requires a nondeterministic number of iterations within a time step when nonlinearities occur. If convergence is not achieved within the time step the real-time hybrid simulation would then suffer a delay in the restoring force feedback, causing the simulation to become unstable.

Rigid floor diaphragms are used in the model, with a master node located at the centroid of each floor plan. The floor mass is lumped at the master node at each floor level. A lean-on-column is used that is loaded with the floor gravity loading to account for the P-delta effect. The inherent damping of the building is modeled by assigning 2% modal damping to modes 1 through 30 and a superimposed stiffness proportional damping for modes 31 and beyond. A lumped mass matrix formulation was used in order to reduce the density of the damping matrix and therefore as noted above make the recurrence relationship of the MCD more efficient. The building is subjected to the 1989 Loma Prieta earthquake, where the horizontal components RSN802-LOMAP-STG000 and RSN802_LOMAP_STG090 (PEER 2019) are scaled to the target uniform hazard spectrum for the MCE hazard level as described previously. These same ground motion records were also used by SGH in the performance evaluation of the prototype building with conventional outriggers (Moehle et al. 2011). In the building three nonlinear viscous dampers that act in parallel are located between each outrigger truss and perimeter column for a total of 36 dampers. Each damper has a nominal force capacity and stroke of 600 kN and ±125 mm, respectively. Only one damper is modeled experimentally (at the north-west corner of the building at the 40^{th} story) while the remaining ones were modeled analytically using an explicit-nonlinear Maxwell

model with an online model approach as indicated in Fig. 1 (see (Al-Subaihawi, Ricles, and Quiel 2022) for more on-line modeling details). The members of the outrigger trusses and perimeter columns, which are in the load path of the dampers, have their axial stiffness increased by a factor three to make the dampers more effective (Al-Subaihawi et al. 2020). Additional information regarding the modeling of the building is given in (Moehle et al. 2011).

In order to obtain the model integration parameters required by the MCD algorithm, the model stiffness k_0 of the nonlinear viscous damper needs to be identified. To achieve this, characterization tests were performed where the damper is subjected to a set of predefined displacement histories with an amplitude of 75 mm and frequency range from 0.15 Hz to 1.5 Hz which corresponds to the range of frequencies that participate in the building response. The displacement included two ramping up cycles, seven cycles with constant amplitude, and three ramping down cycles, all sampled at 1/1024 sec. The test setup for the characterization tests was the same as the RTHS experimental substructure, see Fig. 16a. The damper force-displacement response is given in Fig. 16b, and the instantaneous stiffness defined as the instantaneous stiffness of the force-displacement response is obtained by dividing the change in the measured damper force by the change in displacement. The cumulative distribution function of the instantaneous stiffness is plotted in Fig. 16c. Based on the results shown in Fig. 16c a value of $k_o = 7.1x10^5$kN/m was used for the damper stiffness, which is associated with 99.99% of all measured stiffness being less than this value. If an estimated damper stiffness is used that underestimates the maximum instantaneous stiffness an instability can occur in the RTHS.

The matrices $\boldsymbol{\Psi}$, $\boldsymbol{\Psi}_1$, and $\boldsymbol{\Psi}_2$ from Table 4 are formulated based on the model properties of the building. It is important to emphasize that the stiffness matrix $\mathbf{K}_0$ includes the initial stiffness of the building and the estimated model stiffness k_0 of the nonlinear viscous dampers. For RTHS it is ill-advisable to invert the matrix $\boldsymbol{\Psi}$ in Table 4 because the resulting $\boldsymbol{\Psi}^{-1}$ matrix will be dense. For RTHS, computational efficiency is critical. Therefore, the user is advised to keep $\boldsymbol{\Psi}$ on the left-hand side in order to solve the system of sparse matrices. The matrix $\boldsymbol{\Psi}$ has 58,622 non-zero elements, where the profile is shown in Fig. 17a and where white space is associated with zero elements in the matrix. To efficiently solve the system of equations, the matrix $\boldsymbol{\Psi}$ is decomposed into a lower ($\mathbf{L}$) triangular matrix such that $\mathbf{P}^{\mathbf{T}}\mathbf{S}\ \boldsymbol{\Psi}\ \mathbf{SP} = \mathbf{LDL}^{\mathbf{T}}$, where $\mathbf{P}$ and $\mathbf{S}$ are permutation and scaling matrices, respectively, to improve the numerical stability of the calculations with $\mathbf{S}$ and $\mathbf{D}$ both diagonal matrices (Anderson et al. 1999; Ashcraft, Grimes, and Lewis 1998; Duff 2004). $\boldsymbol{\Psi}$ is re-ordered using the reverse Cuthill-McKee (RCM) algorithm (George and Liu 1981; Gilbert, Moler, and Schreiber 1992) before it is decomposed to reduce the number of non-zero elements in the resulting matrix $\mathbf{L}$. If $\boldsymbol{\Psi}$ is decomposed without the RMC ordering, the resulting $\mathbf{L}$ matrix has the profile shown in Fig. 17b with 69,646 non-zero elements. Decomposing the $\boldsymbol{\Psi}$ after the RCM ordering results in $\mathbf{L}$ with the profile shown in Fig. 17c which has a smaller number of non-zero elements of 64,822. The RCM ordering is used during the RTHS. While decomposing $\boldsymbol{\Psi}$ results in a greater density in $\mathbf{L}$, which can be reduced by applying the RCM algorithm yet the result is still more dense than $\boldsymbol{\Psi}$, it is still more computationally efficient to perform the decomposition rather than invert $\boldsymbol{\Psi}$.

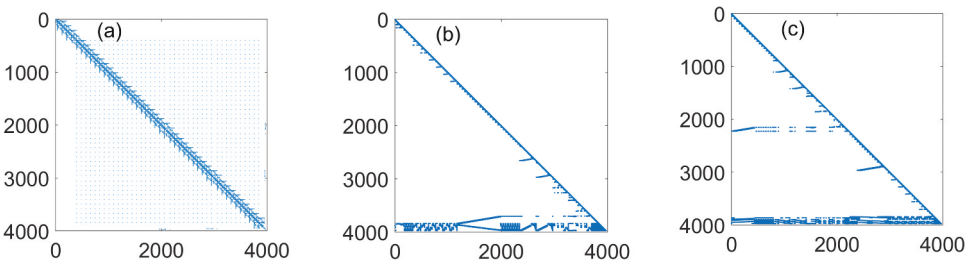

Figure 17. Profile of the matrix Ψ: (a) before RCM ordering; (b) resulting **L** matrix without RCM ordering; and, (c) resulting **L** matrix with RCM ordering.

The sparse matrices $\mathbf{L}$, $\mathbf{P}$, $\mathbf{\Psi}_1$, and $\mathbf{\Psi}_2$ are stored by index (i.e., the sparse matrix is stored in three one dimensional arrays representing the row location, column location, and the corresponding non-zero value of the element of the matrix). The recurrence relationship is recast where $\mathbf{\Psi}$ is on the left-hand side of the expression:

$$\mathbf{\Psi x}_{i+1} = \mathbf{\Psi}_1\mathbf{x}_{i-1} + \mathbf{\Psi}_2 x_i + \Psi_3(\mathbf{F}_i - \mathbf{R}_i), \text{ where } \mathbf{P^T S \Psi S P} = \mathbf{LDL^T} \tag{58}$$

Solving this recurrence relation during the RTHS involves two steps: Step 1: After decomposing $\mathbf{\Psi}$ into $\mathbf{LD}$ (which has to be done only once since elements in $\mathbf{\Psi}$ remain constant), solve for $\mathbf{Y}_1$ from Eq. (59). The size of the vector $\mathbf{Y}_1$ is NDOF x1 where NDOF is the number of degrees of freedom of 3,974.

$$\mathbf{LDY}_1 = \mathbf{P^T S b}, \text{where } \mathbf{b} = \mathbf{\Psi}_1\mathbf{x}_{i-1} + \mathbf{\Psi}_2\mathbf{x}_i + \Psi_3(\mathbf{F}_i - \mathbf{R}_i) \tag{59}$$

Obtaining the right-hand side $\mathbf{b}$ from Eq. (59) involves multiplying the NDOF x NDOF sparse matrices $\mathbf{\Psi}_1$ and $\mathbf{\Psi}_2$ by two NDOF x 1 vectors $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$. The multiplication is performed only on the non-zero elements of the matrices. The vector $\mathbf{b}$ needs to be arranged in accordance with the RCM re-numbering. The matrix $\mathbf{P^T S}$ is also sparse and therefore multiplication operations are performed only on the non-zero elements when multiplied by the vector $\mathbf{b}$. Solving for $\mathbf{Y}_1$ is performed by making use of the triangular shape and sparsity of the matrix resulting from the multiplication $\mathbf{LD}$.

Step 2: Solve for the vector $\mathbf{Y}_2$ from Eq. (60) by making use of the triangular shape and sparsity of the matrix $\mathbf{L^T}$. Then solve for $\mathbf{x}_{i+1}$ from Eq. (61) by making use of the fact that the matrix $\mathbf{SP}$ is sparse.

$$\mathbf{L^T Y}_2 = \mathbf{Y}_1 \tag{60}$$

$$\mathbf{x}_{i+1} = \mathbf{SPY}_2 \tag{61}$$

The resulting $\mathbf{x}_{i+1}$ needs to be re-arranged based on the RCM re-numbering. The sparsity of the model integration parameters is exploited during the RTHS where the multiplication operations are performed only on the non-zero elements. This results in a substantial savings in the cost of integrating the equations of motion during the RTHS, where a time step of 6/1024 sec was used.

Figure 18 shows the time history response of the roof displacement and torsional twist during the RTHS for different values of ρ_∞. The cases with $\rho_\infty = 1$ and $\rho_\infty = 0.86$ have comparable time histories because the adds numerical damping to the lower modes when $\rho_\infty = 0.86$ is negligible; however, using $\rho_\infty = 0$ imposes excessive numerical damping that not only suppressed the higher modes, as will be discussed later, but also affected the response of the lower modes to some extent as evident in Fig. 18. Recall that the purpose of the numerical damping is to suppress high frequency

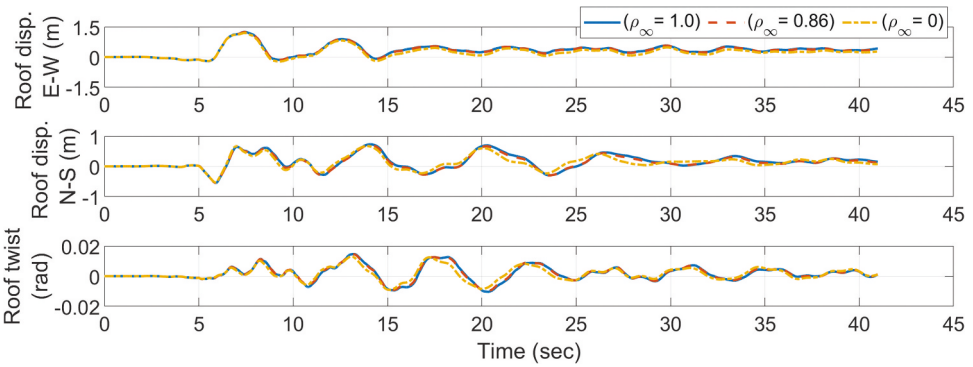

Figure 18. Roof displacement during the 3D real-time hybrid simulations for different ρ_∞ values in (a) the E-W, (b) N-S directions; and, (c) twist at the roof level.

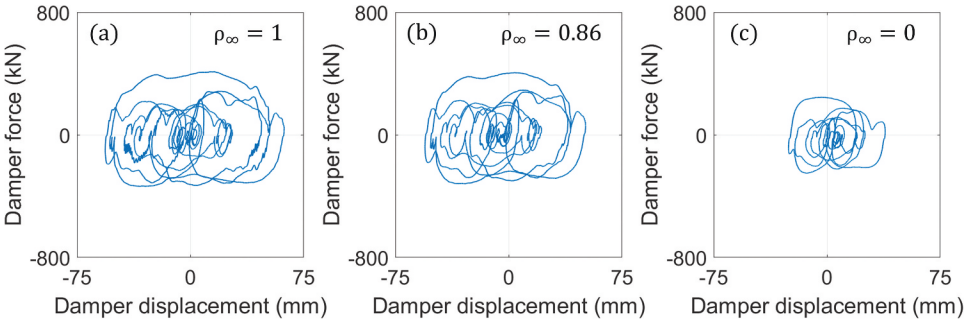

Figure 19. Damper force-displacement response during real-time hybrid simulations for different ρ_∞ values.

oscillations. The building experienced residual displacement at the end of the RTHS because of nonlinear behavior (yielding of the BRBs) in the analytical substructure.

Figure 19 shows the force-displacement response of the damper of the experimental substructure during the RTHS for the cases of $\rho_\infty = 1.0, 0.86,$ and0. A comparison of the results shows that the damper force-displacement response when $\rho_\infty = 1.0$ or $\rho_\infty = 0.86$ are comparable, with the exception of the removal of high frequency oscillations that occurred when $\rho_\infty = 1.0$. However, using $\rho_\infty = 0$ is shown to be inappropriate where the damper displacement decreases substantially because of the added numerical damping has greater influence on the lower modes of interest. Care must be taken to establish an appropriate value for ρ_∞. Figure 5 can be used to provide guidance for selecting the value of $\rho_\infty$and adjusting the time step Δt (i.e., the value of Ω) such that the frequency range of interest is not affected by numerical damping. Figure 20 shows the axial force-deformation response of different braces at stories 35 and 40 in the N-S and E-W directions for the case of $\rho_\infty = 0.86$. The BRBs are seen to exhibit nonlinear behavior in both directions of the building.

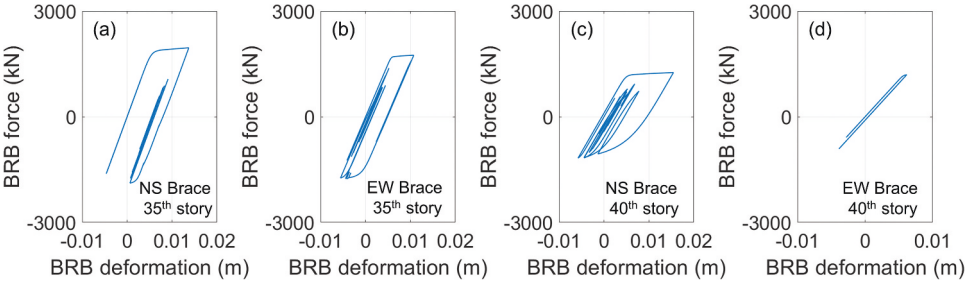

Figure 20. Brace axial force-deformation hysteresis at stories 35 and 40 with $\rho_\infty = 0.86$: (a) 35th story, NS direction, (b) 35th story, EW direction, (c) 40th story, NS direction; and, (d) 40th story, EW direction.

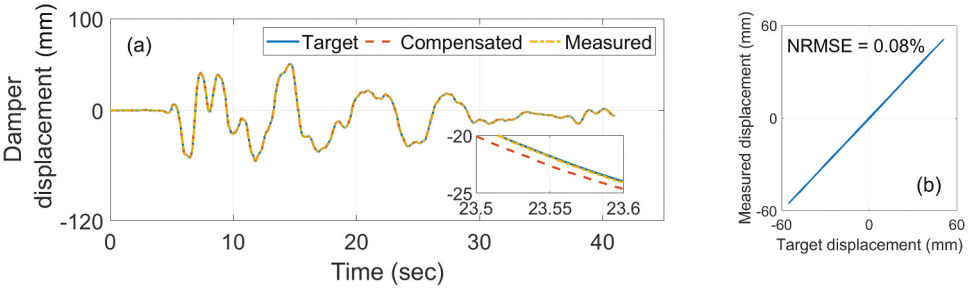

Figure 21. Experimental damper: (a) time history response of damper displacement; and, (b) synchronization subspace plot, $\rho_\infty = 0.86$.

The time history response of the experimental damper for the case of $\rho_\infty = 0.86$ is shown in Fig. 21a where the target displacement x^t is the motion required to be imposed on the damper and based on the result of integrating the equations of motion using the MCD algorithm. The compensated signal is the modified input to the actuator to compensate for amplitude error and delay associated with servo-hydraulic actuator dynamics (Chae, Kazemibidokhti, and Ricles 2013), and the measured damper displacement x^m is the measured displacement during the RTHS. In this study, the second-order adaptive time series compensator (Chae, Kazemibidokhti, and Ricles 2013) is used for each actuator target displacement x^t. The associated synchronization plot of measured displacement plotted against the target displacement is given in Fig. 21b. The NRMSE is 0.08%, and is considered small. The NRMSE is calculated using Eq. (57), where x^p and x^r are replaced by x^t and x^m, respectively. These results, which are representative of all of the three RTHS, show excellent actuator control is achieved and any delay and error amplitude is minimal. The time history of the adaptive compensator parameters is shown in Fig. 22, where the maximum compensated delay ranged from 13msec to 21msec and the maximum compensated amplitude error from 0.98 to 1.01 for the $\rho_\infty = 0.86$ RTHS. The other RTHS has similar results.

It is of interest to study the sensitivity of the RTHS to the model stiffness of the damper k_o. Hence, additional RTHS were performed, where the value for k_o was varied. The hysteretic response of the damper in the experimental substructure is plotted in Fig. 23 for several values of the damper model stiffness. The model stiffness k is normalized by the baseline value discussed previously where $k_o = 7.1 \times 10^5$kN/m, and the results are shown for normalized stiffness values of 0.75, 1.0, 1.5, and 2.5. A case of a normalized stiffness of 0.75 underestimates the maximum actual stiffness of the damper, while the later cases over-estimates it. The associated energy dissipated by the damper for these cases is summarized in Table 6. These results correspond to a reduction of 16.7%, 8.4%, and 22.2% in the total energy dissipation when the

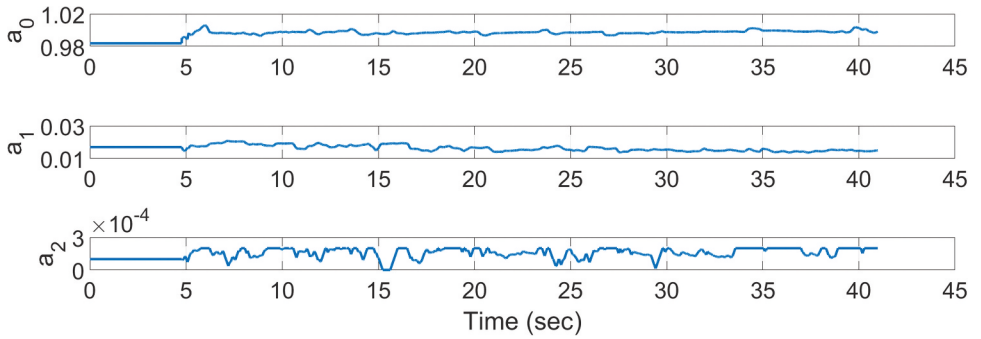

Figure 22. Time history of the adaptive time series compensator parameters, $\rho_\infty = 0.86$.

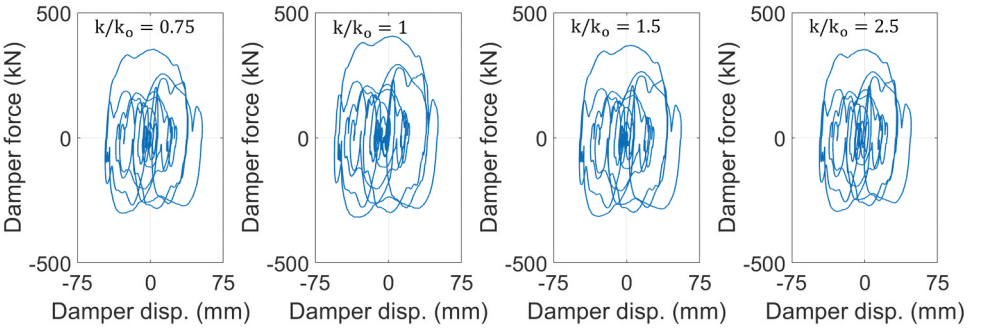

Figure 23. Effects of damper model stiffness on damper hysteretic behavior, $\rho_\infty = 0.86$.

Table 6. Effect of damper model stiffness k_o on energy dissipation of damper and accuracy of roof motions, $\rho_\infty = 0.86$.

Model stiffness ratio (k/k_o)	Total energy dissipation of damper (kN-m)	NRMSE of Roof Displacements (%)		NRMSE of Roof Twist (%)
		E-W	N-S	
0.75	138.3	0.22	1.26	1.37
1	166.1	-	-	-
1.5	152.0	0.31	1.77	1.92
2.5	129.1	0.62	3.36	3.52

model stiffness factor is 0.75, 1.5, and 2.5, respectively. Reducing the damper model stiffness to less than 0.75 of the baseline stiffness resulted in underestimating the maximum instantaneous stiffness which led to the RTHS becoming unstable. On the contrary, increasing the damper model stiffness to 2.5 times the baseline stiffness had the RTHS remain stable but at the expense of smaller energy dissipation compared to the baseline case (for the damper stiffness is overestimated which enables the poles to remain within the unit circle). The instability caused by underestimating the stiffness k_o of the experimental substructure also occurs in the MKR-α and MR-α algorithms (Kolay and Ricles 2019; Kolay et al. 2015). It is advised that the user of the MCD algorithm perform trial numerical simulations to explore the sensitivity of the stability of the RTHS to selected values of k_o to ensure that the actual RTHS will remain stable.

The effect of varying the value of k_o on the accuracy is reflected in the NRMSE values given in Table 6 for the roof bi-directional displacements and twist. The NRMSE for the E-W and N-S displacements range from 0.22% to 0.62% and 1.26% to 3.36%, respectively, while the NRMSE roof twist is shown to range from 1.27% to 3.52%. The error is considered to be small in terms of an experimental error, although the NRSME values become larger as the value for k_o is reduced. (i.e., the ratio k/k_o increases). It is also observed that the NRMSE is larger in the N-S direction due to the greater effectiveness of the dampers in the outrigger system influencing motions parallel to the plane of the outriggers. The NRMSE is also greater in the roof twist since there is motion in the N-S direction to accommodate the twisting motion of the building.

Like any direct integration algorithm, it is recommended to perform a convergence study in order to assure that the results using the MCD algorithm are accurate. It is recommended that users of the MCD algorithm perform such studies, where a systematic variation of the time step Δt, spectral radius ρ_∞, and the assumed values for the model-based parameters of initial stiffness k and damping c of the experimental substructure are systematically varied in order to assess the effects on the accuracy of the results. For these studies, unless it is convenient to incorporate the experimental substructure, these studies can be conducted using hydraulics-off mode where the complete system is modeled analytically.

6. Summary and Conclusions

This paper proposes the Model-based Central Difference (MCD) algorithm for use in real-time hybrid simulations. The MCD algorithm has the following characteristics: (1) explicit, where it does not require iterations to solve nonlinear dynamics problems; (2) unconditional stability for linear and softening-type nonlinear problems; (3) controllable numerical damping that is adjustable using a single free parameter; and (4) the displacement calculation is computationally efficient. The MCD is an efficient integration algorithm for large real-time hybrid simulation (RTHS) nonlinear structural problems with a sparse system of matrices. The reduction in the computational effort is achieved by performing the matrix operations only on the non-zero elements.

Control theory is used to develop the MCD algorithm. The feedback gain of the controller is found by mapping the poles of the transfer function for the non-dissipative form of the algorithm onto the circumference of the unit circle in the complex z-plane, which is in the region of stability. Numerical damping is then incorporated into the MCD algorithm by adding a compensator to the non-dissipative form of the algorithm and mapping the poles of the transfer function for the dissipative

form of the algorithm to lie within the unit circle, which is in the dissipative and stable region of the complex z-plane.

The MCD algorithm is shown to be consistent, convergent, and possess second order accuracy when $\rho_\infty = 1$ and first-order accuracy when $\rho_\infty \in [0, 1]$. The overshoot characteristics of the MCD algorithm are shown to be improved compared to the conventional CD algorithm. The displacement and velocity are shown to remain bounded and stable, with the inclusion of numerical damping reducing the amount of overshooting in the first time step (when the maximum overshoot would occur) for problems involving an initial velocity. The period elongation and the equivalent damping of the MCD algorithm are shown to be a function of Ω, which is associated with the product of the natural frequency and time step. It is found that the period elongation for the MCD algorithm is higher compared to the Generalized-α algorithm, except when $\rho_\infty = 0$. The equivalent damping of the MCD algorithm is a function of ρ_∞, and varies from zero damping (when $\rho_\infty = 1$) to asymptotic annihilation (when $\rho_\infty = 0$), that is 100% numerical dissipation at the high-frequency limit of $\Omega \to \infty$.

A closed-form derivation of the stability limit of the MCD algorithm for nonlinear problems is presented. If the feedback gains of the controller and the compensator are calculated based on the initial state of the structural system and kept constant, it is shown that instability may occur in hardening-type nonlinear SDOF problems when the structure stiffens by a factor of roughly two or more compared to the initial stiffness of the structure. This was found to occur in the RTHS when a value of the estimated damper model stiffness is used that is too small compared to the maximum instantaneous damper stiffness. However, there is some leeway in selecting the value as shown in the RTHS. The user will need to investigate what is a reasonable value for the equivalent stiffness of any experimental substructure component and use a conservative value, such as 1.5 times the estimated stiffness. It is also shown that the MCD algorithm is unconditionally stable for linear and softening type nonlinear problems that are associated with a reduction in structural stiffness.

The computational efficiency of the MCD algorithm is demonstrated by the analysis of a prototype tall building subjected to two natural hazards, namely a wind storm and strong earthquake ground motion. The MCD enables reductions in computational effort by a factor of 1.96 and 1.55 compared to the CAA and MKR-α algorithms, respectively, when the structural response is linear under wind loading, and 2.85 and 1.06 compared to the CAA and MKR-α algorithms, respectively, when the structural response is nonlinear under earthquake loading. The MCD algorithm is implemented in the real-time hybrid simulation (RTHS) of a tall 40 story building with 3,974 degrees of freedoms and a lumped mass matrix and subjected to an earthquake. The MCD algorithm enabled a 3D RTHS simulation to be successfully performed of this structure.

Like any direct integration algorithm, a convergence study should be performed in order to assure that the results using the MCD algorithm are accurate. It is recommended that users of the MCD algorithm perform such studies, where a systematic variation of the time step Δt, spectral radius ρ_∞, and the assumed values for the model-based parameters of initial stiffness k and damping c of the experimental substructure are systematically varied in order to assess the effects on the accuracy and stability of the results. For these studies, unless it is convenient to incorporate the experimental substructure into the simulation, the studies can be conducted using a hydraulics-off mode where the complete system is modeled analytically.

Acknowledgments

This research was performed at the NHERI Lehigh Experimental Facility, whose operation is supported by a grant from the National Science Foundation (NSF) under Cooperative Agreement No. CMMI-2037771. This material is based upon the research supported by the NSF under Grant No. 1463497. The support by NSF is gratefully acknowledged. Any opinions, findings, conclusions, or recommendations expressed herein are the authors and do not necessarily reflect the views or policies of the NSF. Many thanks to the staff of the NHERI Lehigh Experimental Facility for their continued support of this project.

Disclosure Statement

No potential conflict of interest was reported by the author(s).

Funding

The work was supported by the National Science Foundation.

ORCID

Safwan Al-Subaihawi http://orcid.org/0000-0002-7876-5738

References

Al-Subaihawi, S. 2023. "Real-Time Hybrid Simulation of Complex Structural Systems Subjected to Multi-natural Hazards." PhD diss., Bethlehem, PA: Department of Civil and Environmental Engineering, Lehigh University.

Al-Subaihawi, S., C. Kolay, T. Marullo, J. Ricles, and S. Quiel. 2020. "Assessment of Wind-Induced Vibration Mitigation in a Tall Building with Damped Outriggers Using Real-Time Hybrid Simulations." *Engineering Structure* 205:110044. https://doi.org/10.1016/j.engstruct.2019.110044.

Al-Subaihawi, S., J. Ricles, and S. Quiel. 2022. "Online Explicit Model Updating of Nonlinear Viscous Dampers for Real Time Hybrid Simulation." *Soil Dynamics and Earthquake Engineering* 154. https://doi.org/10.1016/j.soildyn.2021.107108.

Anderson, E., Z. Bai, C. Bischof, L. Blackford, J. Demmel, J. Dongarra, J. Croz, et al. 1999. *LAPACK Users' Guide*. 3rd ed. Software, Environments, Tools Philadelphia: Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9780898719604.

Ashcraft, C., R. Grimes, and J. Lewis. 1998. "Accurate Symmetric Indefinite Linear Equation Solvers." *SIAM Journal on Matrix Analysis and Applications* 20 (2): 513–561. https://doi.org/10.1137/S0895479896296921.

Chae, Y., K. Kazemibidokhti, and J. Ricles. 2013. "Adaptive Time Series Compensator for Delay Compensation of Servohydraulic Actuator Systems for Real-Time Hybrid Simulation." *Earthquake Engineering & Structural Dynamics* 42 (11): 1697–1715. https://doi.org/10.1002/eqe.2294.

Chang, S. 2002. "Explicit Pseudodynamic Algorithm with Unconditional Stability." *Journal of Engineering Mechanics* 128 (9): 935–947. https://doi.org/10.1061/(ASCE)0733-9399(2002)128:9(935).

Chang, S. 2014. "A Family of Noniterative Integration Methods with Desired Numerical Dissipation." *International Journal for Numerical Methods in Engineering* 100 (1): 62–86. https://doi.org/10.1002/nme.4720.

Chen, C., and J. Ricles. 2008a. "Development of Direct Integration Algorithms for Structural Dynamics Using Discrete Control Theory." *Journal of Engineering Mechanics* 134 (8): 676–683. https://doi.org/10.1061/(ASCE)0733-9399(2008)134:8(676).

Chen, C., and J. Ricles. 2008b. "Stability Analysis of Direct Integration Algorithms Applied to Nonlinear Structural Dynamics." *Journal of Engineering Mechanics* 134 (9): 703–711. https://doi.org/10.1061/(ASCE)0733-9399(2008)134:9(703).

Chen, C., and J. Ricles. 2010. "Stability Analysis of Direct Integration Algorithms Applied to MDOF Nonlinear Structural Dynamics." *Journal of Engineering Mechanics* 136 (4): 485–495. https://doi.org/10.1061/(ASCE)EM.1943-7889.0000083.

Chen, C., and J. Ricles. 2012. "Analysis of Implicit HHT-α Integration Algorithm for Real-Time Pseudodynamic Testing." *Earthquake Engineering & Structural Dynamics* 41 (5): 1021–1041. https://doi.org/10.1002/eqe.1172.

Chopra, A. 2012. *Dynamics of Structures*. 4th ed. Englewood Cliffs, NJ: Prentice Hall.

Chung, J., and G. Hulbert. 1993. "A Time Integration Algorithm for Structural Dynamics with Improved Numerical Dissipation: The Generalized-Alpha Method." *Journal of Applied Mechanics* 60 (2): 371–375. https://doi.org/10.1115/1.2900803.

Duff, I. 2004. "MA57—A Code for the Solution of Sparse Symmetric Definite and Indefinite Systems." *ACM Transactions on Mathematical Software* 30 (2): 118–144. https://doi.org/10.1145/992200.992202.

Franklin, G., J. D. Powell, and A. Emami. 2009. *Feedback Control of Dynamic Systems*. 6th ed. Upper Saddle River: Prentice hall.

George, A., and J. Liu. 1981. *Computer Solution of Large Sparse Positive Definite Systems*. Englewood Cliffs, New Jersey: Prentice-Hall.

Gilbert, J., C. Moler, and R. Schreiber. 1992. "Sparse Matrices in MATLAB: Design and Implementation." *SIAM Journal on Matrix Analysis and Applications* 13 (1): 333–356. https://doi.org/10.1137/0613024.

Goudreau, G., and R. Taylor. 1973. "Evaluation of Numerical Integration Methods in Elastodynamics." *Computer Methods in Applied Mechanics and Engineering* 2 (1): 69–97. https://doi.org/10.1016/0045-7825(73)90023-6.

Hilber, H. 1976. *Analysis and Design of Numerical Integration Methods in Structural Dynamics*. Rep. No. EERC 67-29. Berkeley, CA: Earthquake Engineering Resistant Center, University of California.

Hilber, H., and T. Hughes. 1978. "Collocation, Dissipation and [Overshoot] for Time Integration Schemes in Structural Dynamics." *Earthquake Engineering & Structural Dynamics* 6 (1): 99–117. https://doi.org/10.1002/eqe.4290060111.

Hilber, H., T. Hughes, and R. Taylor. 1977. "Improved Numerical Dissipation for Time Integration Algorithms in Structural Dynamics." *Earthquake Engineering & Structural Dynamics* 5 (3): 283–292. https://doi.org/10.1002/eqe.4290050306.

Jung, R.-Y., P. Benson Shing, E. Stauffer, and B. Thoen. 2007. "Performance of a Real-Time Pseudodynamic Test System Considering Nonlinear Structural Response." *Earthquake Engineering & Structural Dynamics* 36 (12): 1785–1809. https://doi.org/10.1002/eqe.722.

Kolay, C., T. Marullo, and J. Ricles. 2018. *HybridFEM-MH: A Program for Nonlinear Dynamic Analysis and Real-Time Hybrid Simulation of Civil Infrastructure Systems Subject to Multi-Hazards*. ATLSS Report No. 18-06. Bethlehem, PA: Lehigh University.

Kolay, C., and J. Ricles. 2014. "Development of a Family of Unconditionally Stable Explicit Direct Integration Algorithms with Controllable Numerical Energy Dissipation." *Earthquake Engineering & Structural Dynamics* 43 (9): 1361–1380. https://doi.org/10.1002/eqe.2401.

Kolay, C., and J. Ricles. 2019. "Improved Explicit Integration Algorithms for Structural Dynamic Analysis with Unconditional Stability and Controllable Numerical Dissipation." *Journal of Earthquake Engineering* 23 (5): 771–792. https://doi.org/10.1080/13632469.2017.1326423.

Kolay, C., J. Ricles, T. Marullo, A. Mahvashmohammadi, and R. Sause. 2015. "Implementation and Application of the Unconditionally Stable Explicit Parametrically Dissipative KR-α Method for Real-Time Hybrid Simulation." *Earthquake Engineering & Structural Dynamics* 44 (5): 735–755. https://doi.org/10.1002/eqe.2484.

Liang, X., and K. Mosalam. 2016. "Lyapunov Stability and Accuracy of Direct Integration Algorithms Applied to Nonlinear Dynamic Problems." *Journal of Engineering Mechanics* 142 (5): 04016022. https://doi.org/10.1061/(ASCE)EM.1943-7889.0001073.

Moehle, J., Y. Bozorgnia, N. Jayaram, P. Jones, M. Rahnama, N. Shome, et al. 2011. *Case Studies of the Seismic Performance of Tall Buildings Designed by Alternative Means*. Berkeley PEER: Pacific Earthquake Engineering Research Center College of Engineering University of California. Report 2011/5.

Newmark, N. 1959. "A Method of Computation for Structural Dynamics." *American Society of Civil Engineers* 85 (3): 67–94. https://doi.org/10.1061/JMCEA3.0000098.

Noh, G., and K. Bathe. 2013. "An Explicit Time Integration Scheme for the Analysis of Wave Propagations." *Computers & Structures* 129:178–193. https://doi.org/10.1016/j.compstruc.2013.06.007.

PEER Ground Motion Database. July 19, 2019. "Pacific Earthquake Engineering Research Center." University of California at Berkeley. https://peer.berkeley.edu/peer-strong-ground-motion-databases.

Ricles, J., C. Kolay, and T. Marullo. 2020. *HyCoM-3D: A Program for Multi-Hazard Nonlinear Dynamic Analysis and Real-Time Hybrid Simulation of 3-D Civil Infrastructural Systems (HyCom-3D) User's Manual*. ATLSS Report No. 20-02. Bethlehem, PA: Lehigh University.

Smith, R., and M. Willford. 2007. "The Damped Outrigger Concept for Tall Buildings." *The Structural Design of Tall & Special Buildings* 16 (4): 501–517. https://doi.org/10.1002/tal.413.

Tokyo Polytechnic University. January 19, 2019. "Aerodynamic Database of High-Rise Buildings." http://www.wind.arch.t-kougei.ac.jp/info_center/windpressure/highrise/Homepage/homepageHDF.htm.

Wu, B., H. Bao, J. Ou, and S. Tian. 2005. "Stability and Accuracy Analysis of the Central Difference Method for Real-Time Substructure Testing." *Earthquake Engineering and Structural Dynamics* 34 (7): 705–718. https://doi.org/10.1002/eqe.451.

Wu, B., Q. Wang, P. Benson Shing, and J. Ou. 2007. "Equivalent Force Control Method for Generalized Real-Time Substructure Testing with Implicit Integration." *Earthquake Engineering and Structural Dynamics* 36 (9): 1127–1149. https://doi.org/10.1002/eqe.674.

Zhou, X., and K. Tamma. 2004. "Design, Analysis, and Synthesis of Generalized Single Step Single Solve and Optimal Algorithms for Structural Dynamics." *International Journal for Numerical Methods in Engineering* 59 (5): 597–668. https://doi.org/10.1002/nme.873.

Appendix

The poles must satisfy the requirements of convergence, stability, and removal of the undesirable oscillatory behavior at high frequencies. The derivation starts with the following discrete weighted difference equations for velocity and acceleration:

$$v_i = \frac{(1-\psi)x_{i+1}+\psi x_i-(1-\phi)x_{i-1}-\phi x_i}{2\Delta t}, a_i = \frac{(1-\psi)x_{i+1}+\psi x_i-2x_i+(1-\phi)x_{i-1}+\phi x_i}{\Delta t^2} \quad (A1, 2)$$

Taking the z-transform of these equations and then substituting the result into the equation of motion, the resulting solution for the resulting transfer function produces the following characteristic equation:

$$(-\Omega\xi\psi + \Omega\xi - \psi + 1)z^2 + \left(\Omega^2 + \Omega\xi\psi - \Omega\xi\phi + \psi + \phi - 2\right)z + (\Omega\xi\phi - \Omega\xi - \phi + 1) = 0 \quad (A3)$$

The parameters ψ and ϕ are obtained by substituting the poles for the unconditionally stable Newmark Constant Average Acceleration method (see Eq. (11)) into Eq. (A3):

$$\psi = -\frac{\Omega^2}{4\Omega\xi+4}, \phi = \frac{\Omega^2}{4\Omega\xi-4} \quad (A4, 5)$$

To introduce dissipative characteristics into the transfer function, Eqs. (A1,2) are further weighted in order to shape the root loci of the transfer function such that the root loci branches terminate in the dissipative region of the z-plane, where:

$$v_i = \frac{(1-\alpha_1\psi)x_{i+1} + \alpha_1\psi x_i - (1-\alpha_2\phi)x_{i-1} - \alpha_2\phi x_i}{2\Delta t}, a_i = \frac{(1-\alpha_1\psi)x_{i+1} + \alpha_1\psi x_i - 2x_i + (1-\alpha_2\phi)x_{i-1} + \alpha_2\phi x_i}{\Delta t^2} \quad (A6a, b)$$

Solving for the resulting transfer function leads to the following characteristic equation:

$$(\Omega^2\alpha_1 + 4\Omega\xi + 4)z^2 + (-\Omega^2\alpha_1 - \Omega^2\alpha_2 + 4\Omega^2 - 8)z + \left(\Omega^2\alpha_2 - 4\Omega\xi + 4\right) = 0 \quad (A7)$$

To shape the root loci of the transfer function so that the branches terminate in the dissipative region of the z-plane when $\Omega \to \infty$, the poles of Eq. (A7) are equated to a pair of complex conjugate poles that lie on the imaginary axis having the values of $\pm\sqrt{-\rho_\infty}$. Taking the limit of $\Omega \to \infty$ in Eq. (A7) with the two poles leads to the following:

$$\alpha_1 = \frac{4}{\rho_\infty+1}, \alpha_2 = \frac{4\rho_\infty}{\rho_\infty+1} \quad (A8)$$

The desired set of poles are then obtained for a general value of Ω by substituting α_1 and α_2 from Eq. (A8) into (A7) and solving for the roots to the characteristic equation:

$$z_{1,2} = \frac{\rho_\infty+1\pm\sqrt{\left(\left(-\Omega\xi+\xi^2-1\right)\rho_\infty^2+\left(-\Omega^2+2\xi^2-2\right)\rho_\infty+\xi^2+\Omega\xi-1\right)\Omega^2}}{\Omega^2+\Omega\xi\left(\rho_\infty+1\right)+\rho_\infty+1} \quad (A9)$$

The above poles were produced using a modified form of the difference equations compared to Eq. (9a,b). This enabled an improvement in the overshooting properties of the MCD algorithm to be achieved. It is possible to arrive at an unconditionally stable and dissipative MCD algorithm using Eqs. (A1) and (A2) and the model parameters in Eqs. (A4), (A5); however, the resulting MCD algorithm tends to possess less favorable overshoot characteristics compared to the one presented within the body of the manuscript.