



OPEN ACCESS

EDITED BY

Mariantonieta Gutierrez Soto,
The Pennsylvania State University (PSU),
United States

REVIEWED BY

Cheng Chen,
San Francisco State University, United States
Elif Ecem Bas,
University of Nevada, Reno, United States

*CORRESPONDENCE

Zhen Wang,
✉ wang_zhen@whut.edu.cn

RECEIVED 29 February 2024

ACCEPTED 29 July 2024

PUBLISHED 14 August 2024

CITATION

Shangguan Y, Wang Z, Guo Y, Chen Y, Zeng Y
and Zhou H (2024) Adaptive sliding-mode delay
compensation for real-time hybrid simulations
with multiple actuators.
Front. Built Environ. 10:1393710.
doi: 10.3389/fbuil.2024.1393710

COPYRIGHT

© 2024 Shangguan, Wang, Guo, Chen, Zeng
and Zhou. This is an open-access article
distributed under the terms of the [Creative
Commons Attribution License \(CC BY\)](#). The use,
distribution or reproduction in other forums is
permitted, provided the original author(s) and
the copyright owner(s) are credited and that the
original publication in this journal is cited, in
accordance with accepted academic practice.
No use, distribution or reproduction is
permitted which does not comply with these
terms.

Adaptive sliding-mode delay compensation for real-time hybrid simulations with multiple actuators

Yuekun Shangguan¹, Zhen Wang^{2*}, Yu Guo², Yucai Chen¹,
Yunhai Zeng¹ and Huimeng Zhou¹

¹Engineering Seismic Research Center, Guangzhou University, Guangzhou, China, ²School of Civil Engineering and Architecture, Wuhan University of Technology, Wuhan, Hubei Province, China

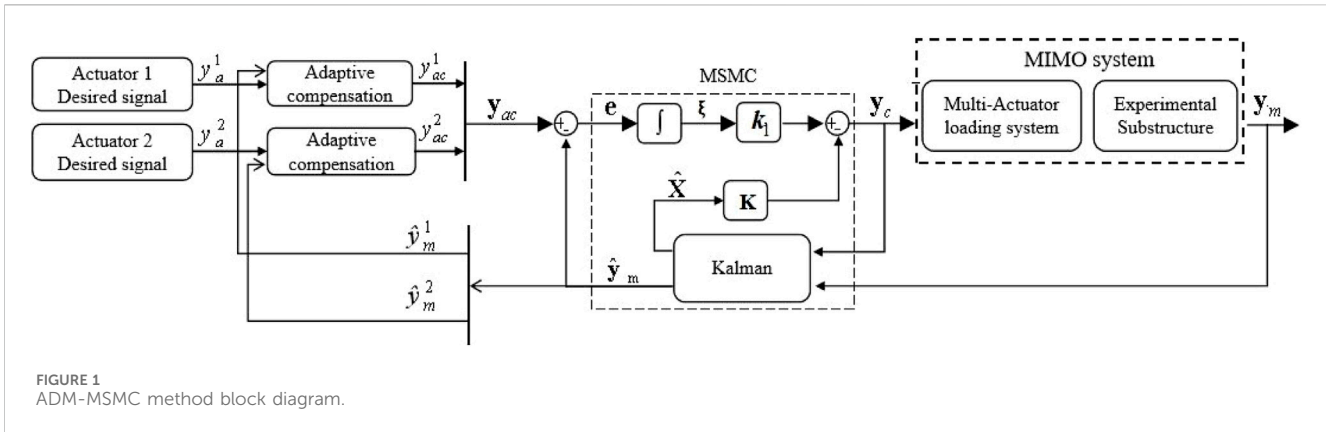
Real-time hybrid simulation (RTHS) is a widely applied test method in structural engineering, which is developed from pseudo-dynamic test. Much of the past work has been centered on one-dimensional RTHS using a single hydraulic actuator. When the complexity of the problem demands to increase the number of degrees of freedom to be enforced on the boundary conditions, more than one hydraulic actuator must be used. Multiple-actuator or multi-axial RTHS (maRTHS) requires that more than one hydraulic actuator exerts the required motion on experimental substructures demanding the implementation of multiple-input multiple-output (MIMO) control strategies. A new maRTHS benchmark control problem has been developed, focusing on a frame subjected to seismic load at the base, substantially transforming and intensifying the complexity of the problem. The time delay generated by the dynamic characteristics of the loading system and the transmission process as well as the high coupling between the hydraulic actuators and the nonlinear kinematics escalates the complexity of the actuator control tracking. A sliding mode adaptive delay compensation method suitable for maRTHS is proposed, which utilizes a MIMO sliding mode method to reduce the coupling effects of actuators and the adaptive compensation method to compensate the residual delay. The effectiveness of the method is verified by numerical simulating different working conditions in the Benchmark Problem Platform.

KEYWORDS

multi-axial real-time hybrid simulation, delay compensation method, decoupling, adaptive, sliding mode

1 Introduction

In recent years, the application of real-time hybrid simulation (RTHS) has become increasingly widespread in civil and other engineering (Stoten et al., 2016; Jiang et al., 2020; Liu, 2020; Yang et al., 2020). It divides the overall structure into numerically simulated substructures and experimentally loaded test substructures, combines real-time loading of physical specimens with computer numerical calculations. It requires the realization of boundary coordination between substructures, namely, force balance and deformation coordination at substructure boundaries. Therefore, substructure boundary coordination becomes critical to the success of RTHS (Horiuchi et al., 1999; Gao, 2012).

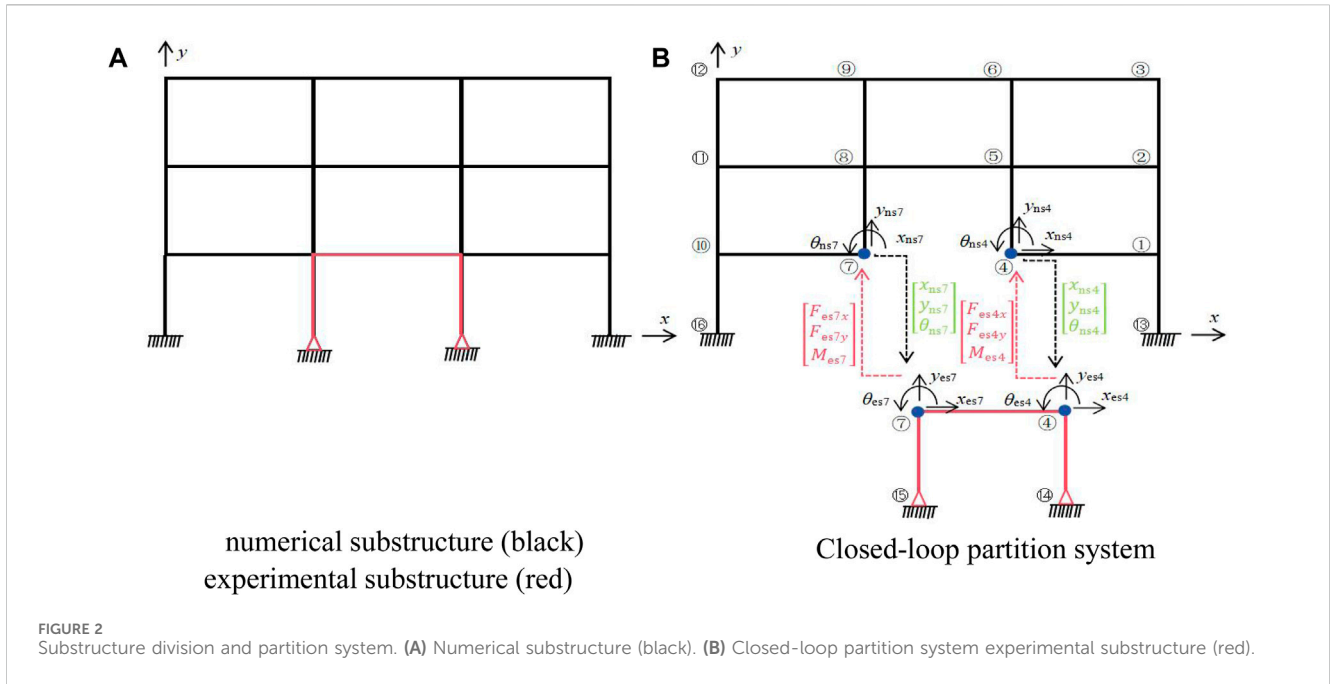


RTHS requires the experimental loading and data transmission to the numerical substructure in real-time, thus system delays will significantly affect the accuracy and stability of RTHS (Gao et al., 2013b). Additionally, the interactions between loading equipment and experimental substructure during tests will lead to the variations of the time delays. Therefore, the development of adaptive time delay compensation methods has been pursued to ensure the stability and precision of RTHS (Wallace et al., 2005; Philips and Spencer, 2011; Chen et al., 2012; Chae et al., 2013; Wu et al., 2013; Ou et al., 2015; Salvatore and Mario, 2016; Hayati and Song, 2017; Zhou et al., 2017; Chae et al., 2018; Alejandro and Mariantonieta, 2019; Wang et al., 2019; Zhou et al., 2019; Wang et al., 2020; Ning et al., 2023). Wang et al. (2019) proposes an adaptive Kalman-based noise filter and an adaptive two-stage delay compensation method, which achieves outstanding tracking performance and excellent robustness. Ning et al. (2023) added a feedback controller to the adaptive feedforward controller to reduce the dependency on the ADM method. Results of virtual and actual RTHS alongside five other compensation strategies revealed the superiority of the proposed compensation method. Wang et al. (2020) proposes an adaptive delay compensation method based on a discrete model (ADM) of the loading system. On the other side, nonlinear control methods especially sliding mode control (SMC) was used to improve the control accuracy (Wu and Zhou, 2014; Rajabi et al., 2018; Xu et al., 2019; Yang et al., 2021; Yang et al., 2023). Wu and Zhou (Wu and Zhou, 2014) used SMC in RTHS for single degree of freedom structure, incorporating the “internal model design” approach into the controller to enable asymptotic tracking of various reference input signals with zero steady-state error. Xu et al. (2019) combined SMC method with improved adaptive polynomial-based forward prediction to improve the robustness of RTHS system. Yang et al. (Yang et al., 2021; Yang et al., 2023) applied the slide mode control in acceleration control for shaking table test and shake table real-time hybrid simulation. Rajabi et al. (2018) combined slide mode control with online state estimations using EKF/UKF to control the shaking table. These adaptive time delay compensation methods and sliding mode control methods are mainly developed for single degree of freedom test, their control effects for maRTHS need to be studied.

Addressing the coupling effects through the intuitive approach of individually compensating for the dynamic characteristics of each actuator may not be very effective. Therefore, a control algorithm tailored for multi-input-multi-output (MIMO) systems is essential

in the context of collaborative loading in hybrid tests with multiple actuators. With the development of RTHS, the experimental substructures have become increasingly complex, often exhibiting strong non-linearity and multiple degrees of freedom characteristics. This necessitates the conduct of multi-directional control and delay compensation in RTHS (Gao et al., 2013a; Fermandois and Spencer, 2017; Sarebanha et al., 2019; Najafi and Spencer, 2021; Tian et al., 2022; Najafi et al., 2023). Fermandois and Spencer (Fermandois and Spencer, 2017) proposed model-based framework control method for multi-axial real-time hybrid simulation testing. Najafi et al. (Najafi and Spencer, 2021; Najafi et al., 2023) proposed a multi-axial real-time hybrid simulation framework that could achieve decoupling control for six actuators. This framework was applied to a small-scale specimen, which must be verified on a full-scale specimen. Tian et al. (2022) proposed an enhanced three variable control method to trace high-frequency force signals of a multi-degree-of-freedom (MDOF) boundary coordinating device. Sarebanha et al. (2019) applied adaptive time series compensator for real-time hybrid simulation of seismically isolated structures in three degree of freedom. Gao et al. (2013a) developed generalized robustness control procedure for MDOF real-time hybrid simulation. For such tests involving multiple actuators for cooperative control, there exists coupling between multiple actuators at the same control point. This means that during the control process of multiple degrees of freedom, a controlled quantity is influenced by multiple control variables. The presence of this coupling effects result in the response of a particular degree of freedom being influenced by multiple actuators, creating mutual interactions among different control points through the test specimen. When this coupling effects of actuators is strong and the load capacity of actuators is limited, it significantly diminishes the tracking control effectiveness of the actuators. Consequently, it becomes challenging to ensure boundary coordination between substructures, leading to a reduction in experimental accuracy and even test failure.

In response to the coupling issue arising from the collaborative loading of multiple actuators in RTHS, this paper develops a sliding mode adaptive time-delay compensation method applicable to MDOF loading. Expanding the single-degree-of-freedom sliding mode control method into vector form to accommodate MIMO systems, the MDOF sliding mode control (MSMC) method is employed to reduce or even eliminate coupling effects of



actuators, achieving similar effects to decoupling. Since the parameters of the sliding mode controller are fixed once set and cannot adaptively adjust their gains based on actual states, to optimize the compensation algorithm, allowing the controller to adaptively adjust its parameters based on operating conditions and responses, this paper integrates the ADM method for discrete model parameter identification with the MSMC method to reduce coupling effects of actuators. Based on the commands and feedback signals of the actuators, the model parameters are adaptively updated, and the controller gains are changed to reduce errors. Additionally, a distributed compensation strategy is adopted, applying ADM compensation to each degree of freedom's sliding mode controller to address its residual delay, further enhancing the tracking control performance of the actuators.

This paper combines ADM and MSMC to propose a MDOF sliding mode adaptive time-delay compensation (ADM-MSMC) method suitable for maRTHS, aiming to enhance the robustness and accuracy of RTHS. In Section 2, the principles of the ADM-MSMC method are primarily introduced. Section 3 defines the benchmark control problem of maRTHS and presents the simmlink diagram of the ADM-MSMC method based on the maRTHS Benchmark Problem Platform. Section 4 conducts RTHS of the proposed ADM-MSMC method based on the Benchmark Problem Platform to verify its feasibility and robust performance. The conclusions drawn from the numerical simulations are summarized in Section 5.

2 Adaptive sliding-mode delay compensation method with MDOF

In response to the issue of mutual coupling during simultaneous multi-actuator loading in RTHS, this section introduces an ADM-MSMC method suitable for MDOF loading. The SMC method is

expanded into a vector form to accommodate MIMO systems. The MSMC method is employed to reduce or eliminate coupling effects of actuators, achieving similar effects to decoupling. Building upon this, a distributed compensation strategy is adopted, applying ADM method individually to each actuator to compensate for residual delay, further enhancing the tracking control performance of the actuators. Thus, this section first outlines the principles of the ADM-MSMC method, followed by separate explanations of the MSMC and ADM method.

2.1 ADM-MSMC method

The ADM-MSMC method adopts the MSMC method to reduce or eliminate the coupling effects of actuators of the MIMO system. On this basis, a decentralized compensation strategy is adopted, and the discrete model parameter identification ADM method compensates the residual delay of each actuator separately, which alleviates the difficulty of the method to compensate individually and achieves better tracking control effects. The principle of this method is shown in the following figure (taking a two-degree-of-freedom system as an example).

In Figure 1, the superscripts of the variables represent the different actuators. The parameter representation in Figure 1 is as shown in Equation 1:

$$\begin{aligned} \mathbf{y}_a &= [y_a^1 \ y_a^2], \mathbf{y}_{ac} = [y_{ac}^1 \ y_{ac}^2]^T, \mathbf{y}_c = [y_c^1 \ y_c^2]^T = \mathbf{k}_1 \boldsymbol{\xi} - \mathbf{K} \hat{\mathbf{X}} \\ \mathbf{y}_m &= [y_m^1 \ y_m^2], \hat{\mathbf{y}}_m = [\hat{y}_m^1 \ \hat{y}_m^2]^T, \mathbf{e} = \mathbf{y}_{ac} - \hat{\mathbf{y}}_m, \boldsymbol{\xi} = \int \mathbf{e} dt \end{aligned} \quad (1)$$

where: \mathbf{y}_a is the target signal, y_a^i is the target signal of actuator i ; \mathbf{y}_{ac} is the actuator response signal, y_{ac}^i is the actuator i response signal; \mathbf{y}_c is the compensated command vector of actuator, y_c^i is the compensated command of actuator i ; \mathbf{y}_m is the measured actuator output response vector, y_m^i is the measured actuator i output response; $\hat{\mathbf{y}}_m$ is the observe response vector; \hat{y}_m^i is the

observe response of actuator i ; k_1 refers to the integral gain coefficient; \mathbf{K} represents the state feedback gain matrix; $\hat{\mathbf{X}}$ is the observed state vector.

2.2 MSMC method

SMC method is a discontinuous control method that introduces any initial state into a stable sliding surface in the phase plane. It allows the system, under certain conditions, to undergo small-amplitude, high-frequency oscillations along a specified state trajectory (the sliding surface), which is also the reason for its insensitivity to parameter variations. Generally, the design of sliding mode (abbreviated as sliding mode) control consists of the following two parts.

- (1) Sliding surface design, which ensures that the system's state trajectory exhibits asymptotically stable and other favorable dynamic characteristics after entering the sliding mode;
- (2) Sliding mode control law design, which involves selecting different reaching laws to drive the system's state trajectory onto the sliding surface within a finite time and maintain motion on it.

2.2.1 Sliding surface design

The SMC method first needs to determine the sliding surface. Let's denote the spatial state equation of the controlled object as follows:

$$\dot{\mathbf{Y}}(t) = \mathbf{A}^* \mathbf{Y}(t) + \mathbf{B}^* \mathbf{u} \tag{2}$$

where $\mathbf{Y}(t)$ represents the state vector matrix. \mathbf{u} is the input of the controlled system. Assuming the sliding surface is represented as $\mathbf{S} = \mathbf{P}\mathbf{Y}$, the sliding surface equation can be expressed as:

$$\mathbf{S} = \mathbf{P}\mathbf{Y} = \mathbf{0} \tag{3}$$

Applying a linear transformation to the system's state:

$$\boldsymbol{\eta} = \boldsymbol{\Gamma}\mathbf{Y} \tag{4}$$

where $\boldsymbol{\Gamma}$ is the state transformation matrix, represented as:

$$\boldsymbol{\Gamma} = \begin{bmatrix} \mathbf{I}_n & -\mathbf{B}_1 \mathbf{B}_2^{-1} \\ \mathbf{0} & \mathbf{I}_r \end{bmatrix}, \mathbf{B}^* = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix} \tag{5}$$

In Equation 5, \mathbf{B}_2 is required to be non-singular; the number of elements in \mathbf{B}_1 and \mathbf{B}_2 are denoted by n and r , respectively. Here, n represents the number of deterministic or linear elements, while r represents the number of uncertain or nonlinear elements. In the two-input-two-output servo-hydraulic system discussed in this chapter, $r = 2$, and n denotes the number of state vectors of the controlled object.

Substituting Equations 4, 5 into Equations 2, 3 respectively, one can derive the structural state equations and sliding surfaces denoted by $\hat{\boldsymbol{\eta}}$.

$$\dot{\hat{\boldsymbol{\eta}}} = \hat{\mathbf{A}}\hat{\boldsymbol{\eta}} + \hat{\mathbf{B}}\mathbf{u} \tag{6}$$

$$\mathbf{S} = \hat{\mathbf{P}}\hat{\boldsymbol{\eta}} = \mathbf{0} \tag{7}$$

where: $\hat{\mathbf{A}} = \boldsymbol{\Gamma}\mathbf{A}^*\boldsymbol{\Gamma}^{-1}$; $\hat{\mathbf{B}} = [\mathbf{0}, \mathbf{B}_2^T]^T$; $\hat{\mathbf{P}} = \mathbf{P}\boldsymbol{\Gamma}^{-1}$.

Decomposing Equations 6, 7 yields:

$$\hat{\boldsymbol{\eta}} = \begin{bmatrix} \boldsymbol{\eta}_1 \\ \boldsymbol{\eta}_2 \end{bmatrix}, \hat{\mathbf{A}} = \begin{bmatrix} \hat{\mathbf{A}}_{11} & \hat{\mathbf{A}}_{12} \\ \hat{\mathbf{A}}_{21} & \hat{\mathbf{A}}_{22} \end{bmatrix}, \hat{\mathbf{P}} = [\hat{\mathbf{P}}_1, \hat{\mathbf{P}}_2] \tag{8}$$

where $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$ are $n \times r$ and $r \times r$ matrices respectively, while the dimensions of other matrices can be determined based on the dimensions of $\boldsymbol{\eta}_1$ and $\boldsymbol{\eta}_2$.

By Equations 6-8, we can obtain:

$$\dot{\hat{\boldsymbol{\eta}}}_1 = \hat{\mathbf{A}}_{11}\boldsymbol{\eta}_1 + \hat{\mathbf{A}}_{12}\boldsymbol{\eta}_2 \tag{9}$$

$$\mathbf{S} = \hat{\mathbf{P}}_1\boldsymbol{\eta}_1 + \hat{\mathbf{P}}_2\boldsymbol{\eta}_2 = \mathbf{0} \tag{10}$$

For simplification of calculations, let $\hat{\mathbf{P}}_2 = \mathbf{I}_r$ be an identity matrix. From Equation 10, it follows that:

$$\boldsymbol{\eta}_2 = -\hat{\mathbf{P}}_1\boldsymbol{\eta}_1 \tag{11}$$

Substituting Equation 11 into Equation 9, we can derive the n th-order motion equation for the system when it moves on the sliding surface as shown in Equation 12:

$$\dot{\hat{\boldsymbol{\eta}}}_1 = (\hat{\mathbf{A}}_{11} - \hat{\mathbf{A}}_{12}\hat{\mathbf{P}}_1)\boldsymbol{\eta}_1 \tag{12}$$

Clearly, the design of the sliding surface \mathbf{S} entails determining \mathbf{P} . If $\hat{\mathbf{P}}_1$ can be reasonably determined to ensure the stability of the system's motion on the sliding surface, and with $\hat{\mathbf{P}}_2 = \mathbf{I}_r$, the \mathbf{P} matrix of the sliding surface can be determined. The principle of using the Linear Quadratic Regulator (LQR) method to determine the matrix \mathbf{P} is to minimize the integral quadratic performance index of the state vector, as shown in Equation 13:

$$J = \int_0^\infty \mathbf{Y}^T(t)\mathbf{Q}\mathbf{Y}(t)dt \tag{13}$$

where \mathbf{Q} is a positive definite matrix. Substituting Equation 8 into the above equation, we obtain the objective function represented by $\boldsymbol{\eta}$ as shown in Equation 14:

$$J = \int_0^\infty \left([\boldsymbol{\eta}_1^T \ \boldsymbol{\eta}_2^T] \mathbf{T} \begin{bmatrix} \boldsymbol{\eta}_1 \\ \boldsymbol{\eta}_2 \end{bmatrix} \right) dt \tag{14}$$

where the \mathbf{T} is as shown in Equation 15:

$$\mathbf{T} = (\boldsymbol{\Gamma}^{-1})^T \mathbf{Q} \boldsymbol{\Gamma}^{-1}; \mathbf{T} = \begin{bmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} \\ \mathbf{T}_{21} & \mathbf{T}_{22} \end{bmatrix} \tag{15}$$

To minimize the objective function of Equation 14 while satisfying the motion equation constraints of Equation 9, according to the maximum principle, we can obtain $\boldsymbol{\eta}_1$ with $\boldsymbol{\eta}_2$ as the feedback state.

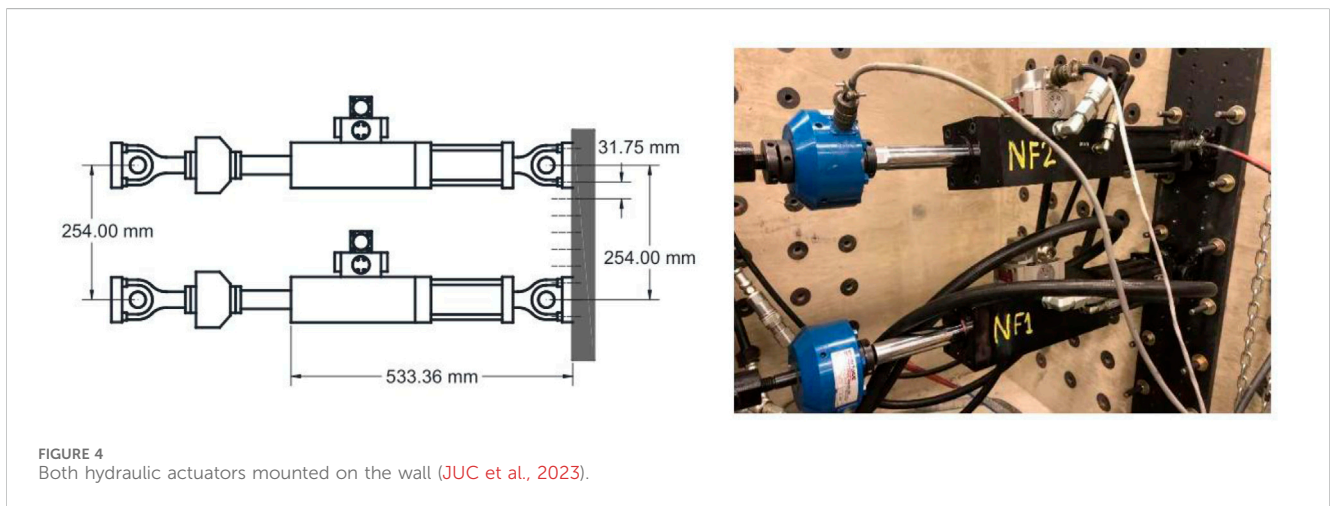
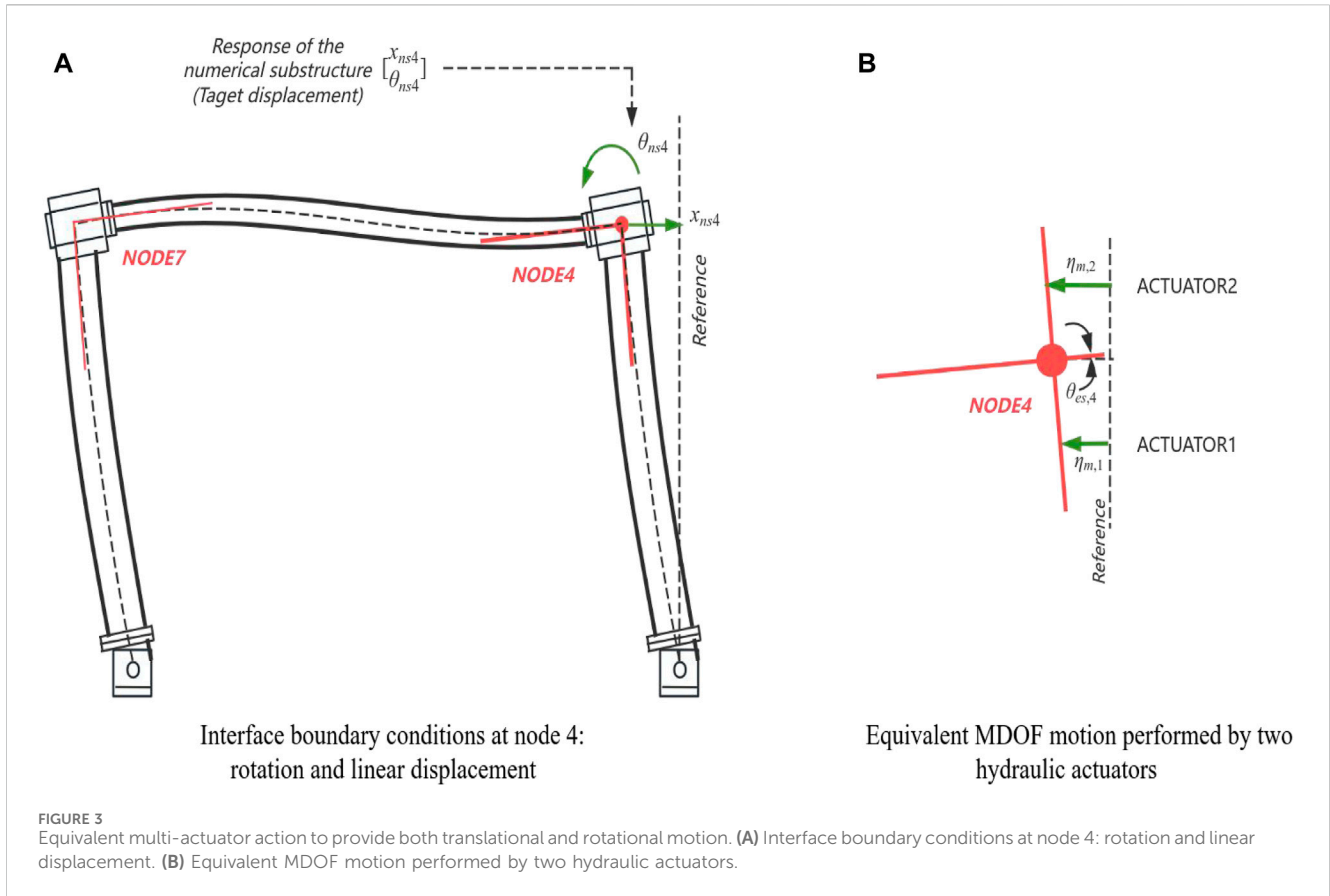
$$\boldsymbol{\eta}_2 = -0.5\mathbf{T}_{22}^{-1}(\hat{\mathbf{A}}_{12}^T \mathbf{H} + 2\mathbf{T}_{21})\boldsymbol{\eta}_1 \tag{16}$$

where \mathbf{H} is the solution to the following Riccati Equation 17:

$$\begin{aligned} \tilde{\mathbf{A}}^T \mathbf{H} + \mathbf{H} \tilde{\mathbf{A}} - 0.5\mathbf{H} \hat{\mathbf{A}}_{12} \mathbf{T}_{22}^{-1} \hat{\mathbf{A}}_{12}^T \mathbf{H} &= -2(\mathbf{T}_{11} - \mathbf{T}_{12} \mathbf{T}_{22}^{-1} \mathbf{T}_{12}^T) \tilde{\mathbf{A}} \\ &= \hat{\mathbf{A}}_{11} - \hat{\mathbf{A}}_{12} \mathbf{T}_{22}^{-1} \mathbf{T}_{21} \end{aligned} \tag{17}$$

The well-known Riccati equation can be solved using the LQR function in MATLAB software to obtain the matrix \mathbf{H} . Combining Equations 11, 16, we get the Equation 18:

$$\hat{\mathbf{P}}_1 = 0.5\mathbf{T}_{22}^{-1}(\hat{\mathbf{A}}_{12}^T \mathbf{H} + 2\mathbf{T}_{21}) \tag{18}$$



Considering Equation 8, we can get the Equation 19:

$$\hat{\mathbf{P}} = \begin{bmatrix} \hat{\mathbf{P}}_1 & \hat{\mathbf{P}}_2 \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{P}}_1 & \mathbf{I}_r \end{bmatrix} \quad (19)$$

At this point, $\hat{\mathbf{P}}$ is determined, and the matrix $\mathbf{P} = \hat{\mathbf{P}}\Gamma$ of the slip surface is also determined.

2.2.2 Sliding mode control law design

Using the Lyapunov method directly, the sliding mode control law is designed. Let the Lyapunov function be denoted as:

$$v = 0.5\mathbf{S}^T\mathbf{S} = 0.5\mathbf{Y}^T\mathbf{P}^T\mathbf{P}\mathbf{Y} \quad (20)$$

For $v \geq 0$, as $t \rightarrow \infty$, ensuring that the system motion on the sliding surface is asymptotically stable is a sufficient condition when $\dot{v} = \mathbf{S}^T\dot{\mathbf{S}} \leq 0$. Substituting Equation 2 into Equation 20 and differentiating, we obtain the Equation 21:

$$\begin{aligned} \dot{v} &= \mathbf{S}^T\mathbf{P}\dot{\mathbf{Y}} = \mathbf{S}^T\mathbf{P}(\mathbf{A}^*\mathbf{Y} + \mathbf{B}^*\mathbf{u}) = (\mathbf{P}\mathbf{Y})^T\mathbf{P}\mathbf{B}^*[(\mathbf{P}\mathbf{B}^*)^{-1}\mathbf{P}\mathbf{A}^*\mathbf{Y} + \mathbf{u}] \\ &= \mathbf{Y}^T\boldsymbol{\chi}(\mathbf{u} - \mathbf{G}_s\mathbf{Y}) \end{aligned} \quad (21)$$

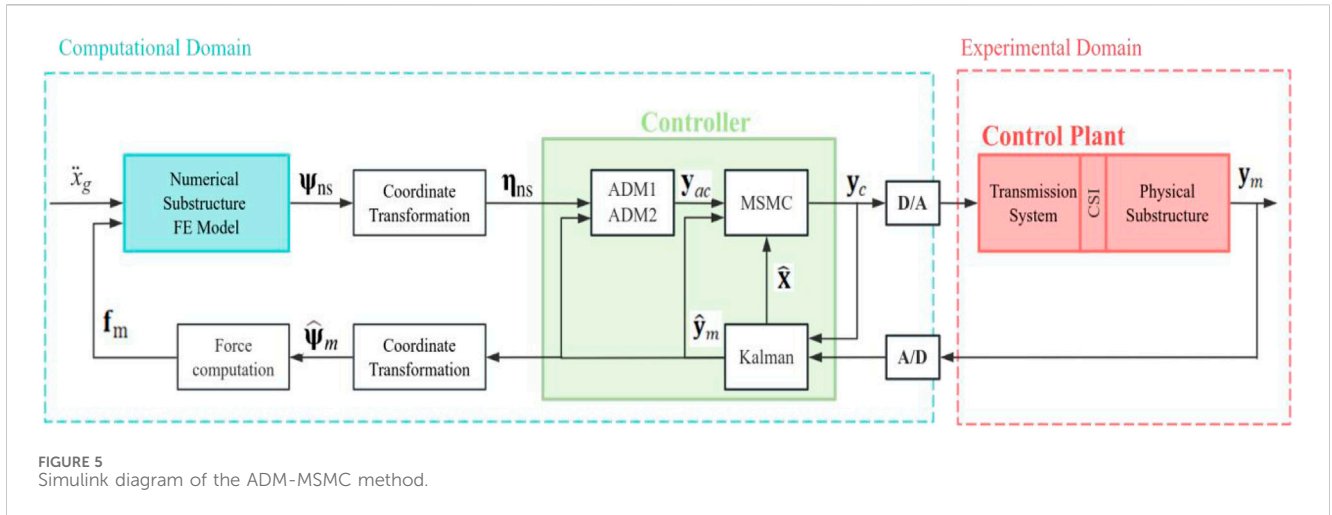


FIGURE 5 Simulink diagram of the ADM-MSMC method.

where $\chi = P^T P B^*$, $G_s = -(PB^*)^{-1} P A^*$

Using the continuous control law given by Equation 22 ensures the asymptotic stability of the system.

$$u = (G_s - \delta \chi^T) Y \tag{22}$$

where δ represents a specified positive constant known as the sliding tolerance. Consequently, $\dot{v} = -(Y^T \chi) \delta (Y^T \chi)^T \leq 0$ holds throughout the entire control process.

2.2.3 The principle of MSMC

In this benchmark control problem, only the uncertainty of the actuator-specimen system model is considered. A linear actuator-specimen transfer function model (JUC et al., 2023) is adopted herein for the convenience of the later analysis. For a MIMO system with multiple actuators collaborating in RTHS, the transfer function matrix of the system needs to be discussed. The expression for the transfer function relationship for a MIMO system is given by the Equation 23:

$$\begin{bmatrix} Y_m^1(s) \\ Y_m^2(s) \\ \vdots \\ Y_m^q(s) \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) & \dots & G_{1p}(s) \\ G_{21}(s) & G_{22}(s) & \dots & G_{2p}(s) \\ \vdots & \vdots & \dots & \vdots \\ G_{q1}(s) & G_{q2}(s) & \dots & G_{qp}(s) \end{bmatrix} \begin{bmatrix} Y_c^1(s) \\ Y_c^2(s) \\ \vdots \\ Y_c^p(s) \end{bmatrix} \tag{23}$$

where: $G_{ij}(s) (i = 1, 2, \dots, q; j = 1, 2, \dots, p)$ denotes the transfer function between the i th output quantity and the j th input quantity; The Y_c^i and Y_m^i is obtained by performing the Laplace transform of y_c^i and y_m^i .

As shown in Equation 23, the transfer function matrix of a coupled system is generally a non-diagonal array, where each input affects all outputs and each output is affected by all inputs. The presence of mutual coupling effects between actuators then implies that there are no zeros on the non-diagonal terms in the transfer function matrix.

The transfer function of the element of transfer function matrix $G_{ij}(s)$ is expressed as

$$G_{ij}(s) = \frac{Y_m(s)}{Y_c(s)} = \frac{b_n s^n + b_{n-1} s^{n-1} + \dots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0} \tag{24}$$

where: a_n and b_n are the parameters defined in (Silva et al., 2020). The variable z is defined such as the Equation 25

$$\begin{aligned} z^n + a_{n-1} z^{n-1} + \dots + a_1 z + a_0 z &= y_c \\ y_m &= \beta_{n-1} z^{n-1} + \dots + \beta_1 z + \beta_0 z \end{aligned} \tag{25}$$

Then the dynamic equations of the vector-matrix form of the transfer function Equation 24 are as the Equation 26

$$\dot{X}_{ij} = A_{ij} X_{ij} + B_{ij} y_c \quad y_m = C_{ij} X_{ij} \tag{26}$$

where:

$$X_{ij} = \begin{bmatrix} z \\ \dot{z} \\ \vdots \\ z^{n-1} \end{bmatrix}, A_{ij} = \begin{bmatrix} 0 & 1 & \dots & 0 \\ \vdots & 0 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & \vdots & \dots & \vdots \\ -a_0 & -a_1 & \dots & -a_{n-1} \end{bmatrix}, B_{ij} =$$

$$\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, C_{ij} = [\beta_0 \ \beta_1 \ \dots \ \beta_{n-1}],$$

$$\beta_0 = b_0 - a_0 b_n$$

$$\beta_1 = b_1 - a_1 b_n$$

$$\vdots$$

$$\beta_{n-1} = b_{n-1} - a_{n-1} b_n$$

Similarly, the MSMC method needs to extend the sliding mode method to a vector form method with MDOF. The following is an example of a two-degree-of-freedom system ($i = j = 2$) to introduce the MSMC method.

Transform the transfer function matrix into the state space equation, as the Equation 27:

$$\dot{X} = AX + B y_c \quad y_m = CX \tag{27}$$

where: $X = \begin{bmatrix} X_{11} \\ X_{12} \\ X_{21} \\ X_{22} \end{bmatrix}, A = \begin{bmatrix} A_{11} & 0 & 0 & 0 \\ 0 & A_{12} & 0 & 0 \\ 0 & 0 & A_{21} & 0 \\ 0 & 0 & 0 & A_{22} \end{bmatrix}, B = \begin{bmatrix} B_{11} & 0 \\ 0 & B_{12} \\ B_{21} & 0 \\ 0 & B_{22} \end{bmatrix}, y_c = \begin{bmatrix} y_c^1 \\ y_c^2 \end{bmatrix}, y_m = \begin{bmatrix} y_m^1 \\ y_m^2 \end{bmatrix}, C = \begin{bmatrix} C_{11} & C_{12} & 0 & 0 \\ 0 & 0 & C_{21} & C_{22} \end{bmatrix}.$

The corresponding state space coefficient matrix is built from a single transfer function $G_{ij}(s)$ using the following form A_{ij}, B_{ij}, C_{ij} .

In the MSMC method in this paper, the ‘internal mode design’ method is used to introduce the tracking error term into the state

TABLE 1 RTHS partitioning cases.

Partitioning configuration	Reference floor mass (kg)	Reference modal damping (%)
Case 1	1,000	5
Case 2	1,100	4
Case 3	1,300	3
Case 4	1,000	2

vector to construct a new system containing the error term, and the sliding mode control is performed on the new system, which successfully converts the state regulation problem of the sliding mode method into an error tracking problem and is successfully applied to the hydraulic servo system. MSMC control introduces the error matrix of a MDOF system into the state vector, and utilizes the sliding mode approach to regulate the state to the ‘zero state’ characteristic for error tracking control of the hydraulic servo system.

The tracking error after observation is as the Equation 28:

$$\dot{e} = \dot{y}_{ac} - \dot{y}_m = -C\dot{X} \tag{28}$$

where: $\dot{y}_{ac} = 0$.

Therefore, by introducing the tracking error term into the state vector, the new system model obtained is:

$$\begin{pmatrix} -\dot{e} \\ \ddot{X} \end{pmatrix} = \begin{pmatrix} \mathbf{0} & \mathbf{C} \\ \mathbf{0} & \mathbf{A} \end{pmatrix} \begin{pmatrix} -e \\ \dot{X} \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ \mathbf{B} \end{pmatrix} \dot{y}_c \tag{29}$$

$$\dot{y}_c = -[k_1 \mathbf{K}] \begin{bmatrix} -e \\ \dot{X} \end{bmatrix} \tag{30}$$

Such that Equation 29 is stable. This implies that the tracking error e is stable, the integral gain coefficient k_1 and the state feedback gain matrix \mathbf{K} will be found. The MSMC in the paper uses the LQR method (a pole configuration method) to obtain the integral and state feedback gains (Ou, 2003). Thus, we will have achieve the objective of asymptotic tracking with zero steady state error. The control input, found by integrating Equation 30, is:

$$y_c = k_1 \int edt - \mathbf{KX} = k_1 \xi - \mathbf{KX} \tag{31}$$

The state vector \mathbf{X} in Equation 31 is replaced by the observed state vector $\hat{\mathbf{X}}$ in actual controller, which will be discussed later.

Simultaneously, we can express Equation 29 in the following form:

$$\dot{\hat{\mathbf{X}}}(t) = \mathbf{A}^* \hat{\mathbf{X}}(t) + \mathbf{B}^* \dot{y}_c - \dot{y}_m = \mathbf{C}^* \hat{\mathbf{X}}(t) \tag{32}$$

where: $\dot{\hat{\mathbf{X}}}(t) = \begin{bmatrix} -\dot{e} \\ \ddot{X} \end{bmatrix}$, $\mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix}$, $\dot{y}_c = \begin{bmatrix} \dot{y}_c^1 \\ \dot{y}_c^2 \end{bmatrix}$, $\dot{y}_m = \begin{bmatrix} \dot{y}_m^1 \\ \dot{y}_m^2 \end{bmatrix}$, $\mathbf{A}^* = [\mathbf{0}, \mathbf{C}; \mathbf{0}, \mathbf{A}]$, $\mathbf{B}^* = [\mathbf{0}, \mathbf{B}]^T = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix}$.

Since the matrix \mathbf{B}_2 is required to be non-singular, it is necessary to modify the positions of the elements in the state vector of Equation 32, as the Equation 33:

$$\mathbf{B}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{33}$$

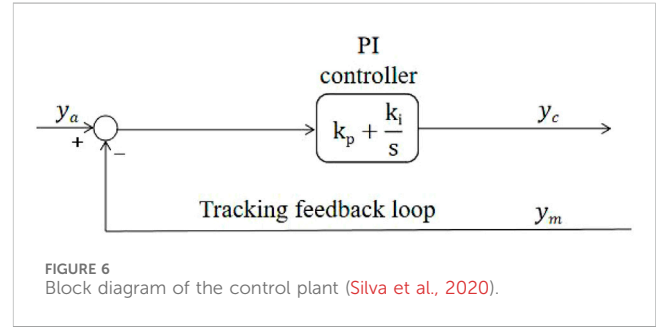


FIGURE 6 Block diagram of the control plant (Silva et al., 2020).

Therefore, \mathbf{B}_2 is a non-singular matrix, if it is a singular matrix, row and column transformations are used for \mathbf{B}^* to render \mathbf{B}_2 non-singular. The positions of the elements in \mathbf{A}^* and \mathbf{C}^* of the Equation 32 need to be conducted same row and column transformations.

Thus, the error tracking problem of the hydraulic servo system is transformed into a state regulation control problem using sliding. The control of coupled systems is complex, and in the field of automatic control, appropriate corrections are often introduced to diagonalize the transfer function matrix, known as decoupling, so that a certain output is controlled only by a certain input. The MSMC principle used in this section is not decoupled in the conventional sense and does not diagonalize the transfer function matrix. The design of the sliding surface is aimed at achieving the desired dynamic characteristics for the system. Therefore, the principle of MSMC to reduce or even eliminate the coupling effects of actuators is actually still through the pole configuration method.

In addition, ‘‘internal mode design’’ requires advance knowledge of the system’s state vector \mathbf{X} . However, since the state vector \mathbf{X} contains higher-order derivatives of measurement signals, many state variables cannot be directly measured by sensors. Higher order quantities obtained by direct derivation of the sensor measurement signal are then susceptible to noise. Therefore, the Kalman filtering (KF) method is employed as a state observer to estimate the state vector, obtaining more accurate state signals (the observed state vector $\hat{\mathbf{X}}$).

The equation for state estimation is as the Equation 34:

$$\dot{\hat{\mathbf{X}}} = \mathbf{A}\hat{\mathbf{X}} + \mathbf{B}y_c + \mathbf{K}_e(y_m - \mathbf{C}\hat{\mathbf{X}}) = (\mathbf{A} - \mathbf{K}_e\mathbf{C})\hat{\mathbf{X}} + \mathbf{B}y_c + \mathbf{K}_e y_m \tag{34}$$

where: \mathbf{K}_e is the observer feedback matrix, also known as the observation gain matrix.

When solving for the observer feedback matrix \mathbf{K}_e , the Matlab function *lqe2* can be employed, with the Equation 35:

$$\mathbf{K}_e = lqe2(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{Q}_e, \mathbf{R}_e) \tag{35}$$

where: \mathbf{Q}_e and \mathbf{R}_e are the covariance matrices of system noise and measurement noise, respectively.

2.3 ADM method

The ADM compensation method can adaptively adjust the parameter y_{ac}^i and the coefficient θ_i in the formula. Firstly, the system model is established, which establishes the relationship

TABLE 2 Parameters of the controllers.

Controller	Controller parameters					
	$\hat{\theta}_0$	P_0	Q	δ	Q_e	R_e
ADM	[100 4.5-4] [100-1.5 0.1]	$100^* I_{3 \times 3}$ $100^* I_{3 \times 3}$	NA	NA	NA	NA
MSMC	NA	NA	$\begin{bmatrix} 500^* I_{2 \times 2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0.002^* I_{14 \times 14} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 0.00004^* I_{2 \times 2} \end{bmatrix}$	220 70.2	0.01	1,000
ADM-MSMC	[165-2 -1.5] [98-1.5-7]	$100^* I_{3 \times 3}$ $100^* I_{3 \times 3}$	$\begin{bmatrix} 500^* I_{2 \times 2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0.002^* I_{14 \times 14} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 0.00004^* I_{2 \times 2} \end{bmatrix}$	220 70.2	0.01	1,000

between the input and output of the loading system, thereby predicting the response of the system to specified commands based on this foundation. The ADM method employs a difference model, namely:

$$y_{ac}^i = \Phi_i^T \theta_i \tag{36}$$

where: $\Phi_i^T = [y_{ac}^{i-1}, \dots, y_{ac}^{i-p}, y_m^i, y_m^{i-1}, \dots, y_m^{i-q+1}, \dots]$, $\theta_i = [\theta_{ac}^1, \dots, \theta_{ac}^p, \theta_m^1, \dots, \theta_m^q, \dots]$, θ_{ac}^p and θ_m^q are the respective model parameters relevant to the command y_{ac} and measured displacements y_m , with p and q denoting the numbers of parameters/terms.

After simplifying the experimental system into a discrete model, it is necessary to online estimate the parameters θ_i in Equation 36 during the experiment, and real-time track the changes in system characteristics. The parameters are estimated using a recursive least-squares algorithm (You et al., 2019) with a forgetting factor, which has the advantages of small storage requirements and computational simplicity. This algorithm is effective in overcoming the “data saturation” phenomenon and can be expressed as:

$$\hat{\theta}_i = \hat{\theta}_{i-1} + \frac{P_{i-1} \Phi_i}{\lambda + \Phi_i^T P_{i-1} \Phi_i} (y_{ac}^i - \Phi_i^T \hat{\theta}_{i-1}) \tag{37}$$

$$P_i = \frac{1}{\lambda} \left(I - \frac{P_{i-1} \Phi_i \Phi_i^T}{\lambda + \Phi_i^T P_{i-1} \Phi_i} \right) P_{i-1} \tag{38}$$

In Equation 37, 38: λ represents the forgetting factor, typically set to $0.9 \leq \lambda \leq 1$; I denotes the identity matrix. The initial covariance P_0 and parameter $\hat{\theta}_0$ are evaluated by the standard least-squares method with offline test data, as the Equations 39, 40

$$\hat{\theta}_0 = (\Phi^T \Phi)^{-1} \Phi^T Y \tag{39}$$

$$P_0 = (\Phi^T \Phi)^{-1} \tag{40}$$

with

$$\Phi = (\Phi_1, \Phi_2, \dots, \Phi_L)^T$$

$$Y = (y_{ac}^1, y_{ac}^2, \dots, y_{ac}^L)^T$$

where L indicates the length of the data.

The determination of initial values P_0 and $\hat{\theta}_0$, requires offline experimental loading, followed by parameter updates for a period of time, and then the converged values are taken as the initial values for the method. Otherwise, the estimated initial values may have significant errors, affecting the precision of the experiment.

The parameters estimated from Equation 37, 38 can reflect the current state of the system, including its time delay characteristics. If the established system model is effective, it will achieve good time delay compensation effects and automatically track changes in system time delay. The objective of time delay compensation control strategy is to make the measured signal y_m^i as close to the target signal y_a^i as possible, and in order to enhance the robustness of the method, the compensation method is as the Equation 41:

$$y_{ac}^{i+1} = \Phi_{i+1}^T \theta_{i+1} \tag{41}$$

where: $\Phi_{i+1}^T = [y_{ac}^i, \dots, y_{ac}^{i-p+1}, y_a^{i+1}, y_a^i, \dots, y_a^{i-q+2}, \dots]$.

Assuming that the model parameters do not vary significantly within $t_i \sim t_{i+1}$, we can get the Equation 42:

$$\theta_{i+1} \approx \hat{\theta}_i \tag{42}$$

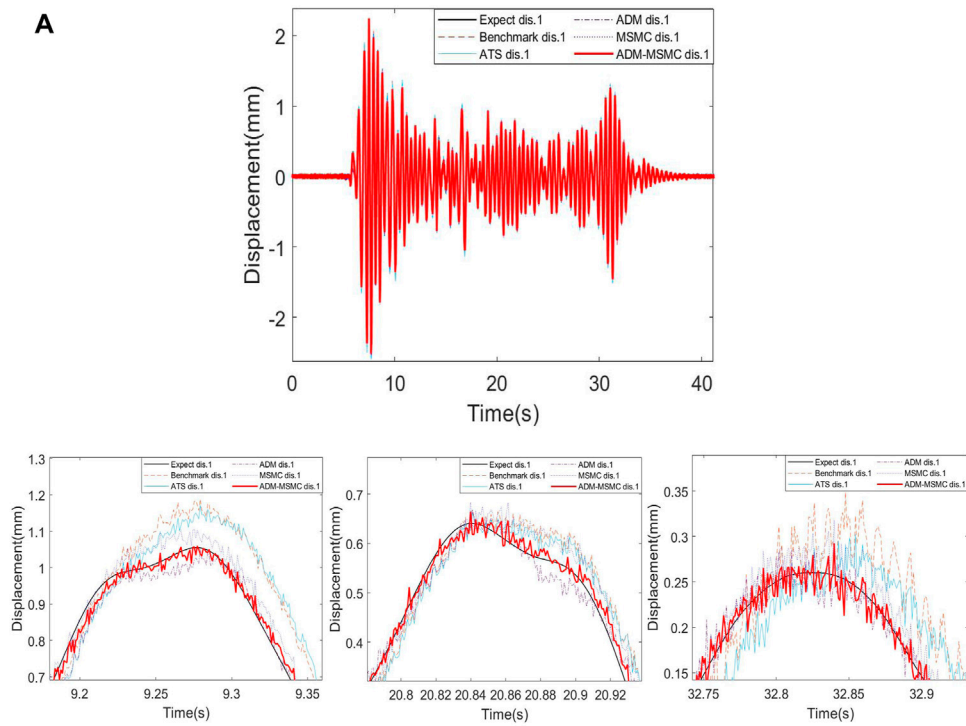
In conclusion, the ADM method assumes the system as a difference equation model. It predicts and updates parameters based on the displacement response of the actuator in the initial steps, then compensates by using the updated parameters to predict the next displacement command against the desired system displacement.

3 Virtual experiment platform

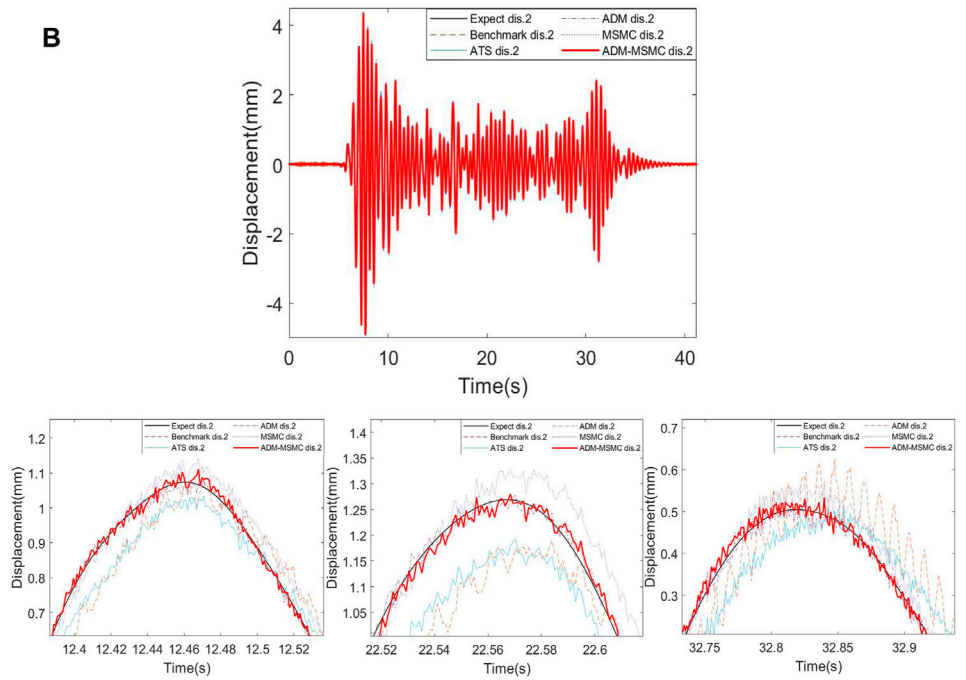
To better validate the feasibility and robustness of the proposed time-delay compensation method in this paper, it is necessary to apply the ADM-MSMC method to conduct RTHS on a Benchmark Problem Platform. Thus, in order to integrate the ADM-MSMC method with the Benchmark Problem Platform effectively, this section firstly introduces the partitioning of experimental and numerical substructures in the platform, along with the transmission system. Finally, the subsequent numerical simulations scheme for the maRTHS simulation is presented.

3.1 Structural division

A three-story, three-span steel structure is used as the reference model, divided into numerical substructures and experimental substructures. The middle span of one floor structure is designated as the experimental substructure, as shown in Figure 2A.



Desired displacement and actual displacement response of actuator 1

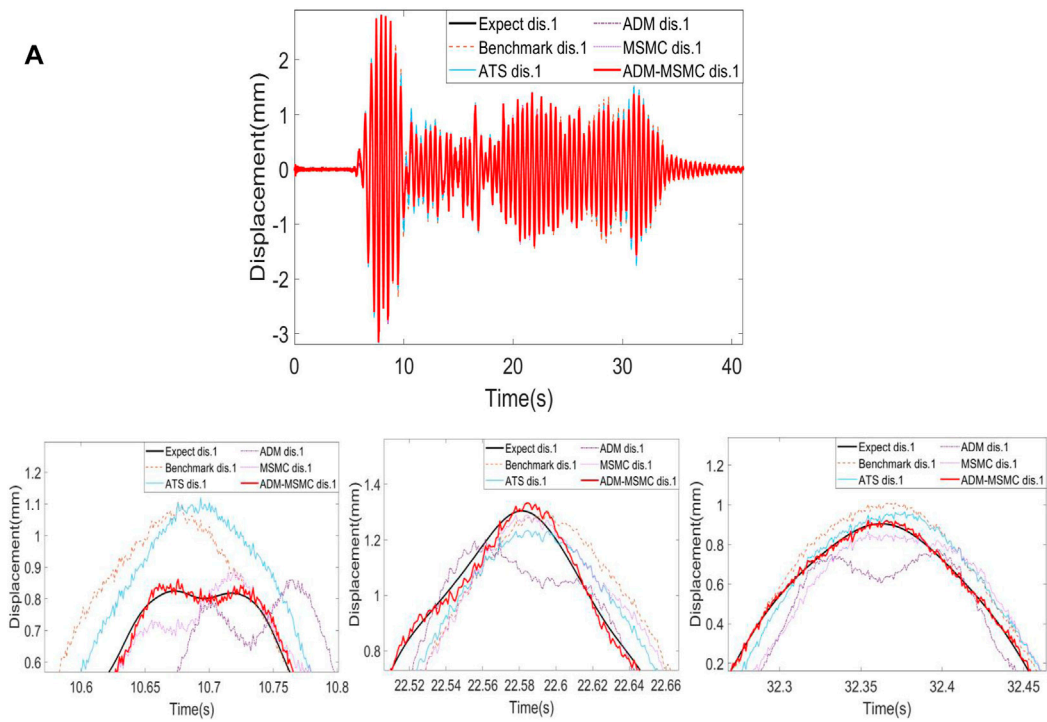


Desired displacement and actual displacement response of actuator 2

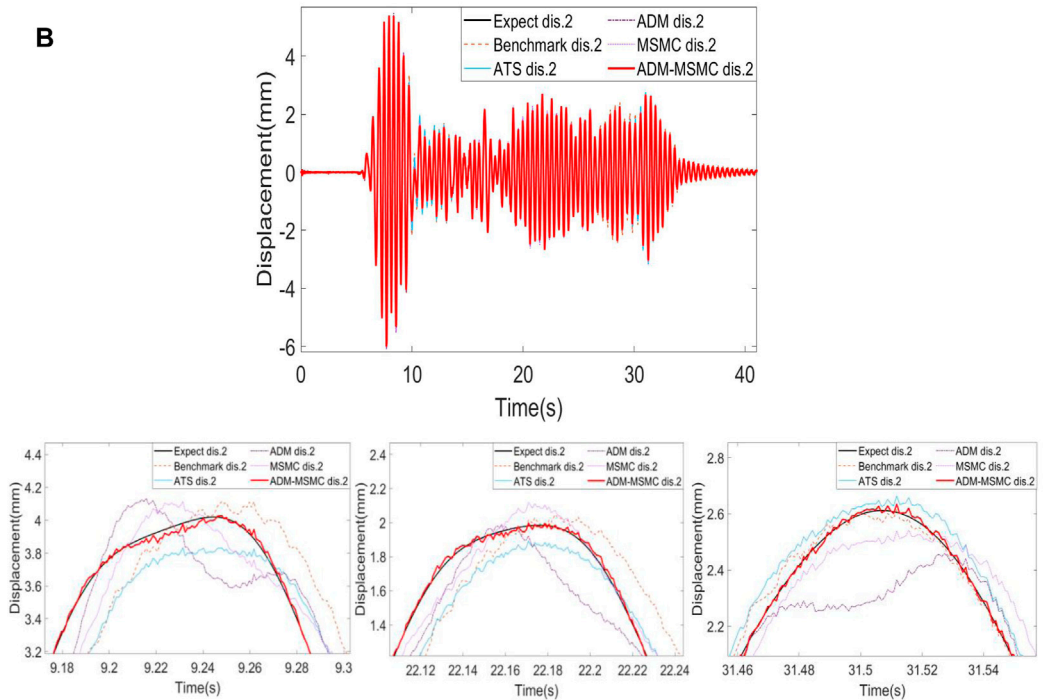
FIGURE 7 Displacement time-history comparison of Case 1. **(A)** Desired displacement and actual displacement response of actuator 1. **(B)** Desired displacement and actual displacement response of actuator 2.

Taking an example of a three-story experimental substructure, where the experimental substructure is the middle span of the first floor, and the rest is the numerical substructure, as shown in **Figure 2A**.

The numerical substructure and the experimental substructure are interconnected and synchronized through a feedback loop that allows them to share information at the common interface nodes at each time



Desired displacement and actual displacement response of actuator 1



Desired displacement and actual displacement response of actuator 2

FIGURE 8 Displacement time-history comparison of Case 4. (A) Desired displacement and actual displacement response of actuator 1. (B) Desired displacement and actual displacement response of actuator 2.

step during execution (Silva et al., 2020). Figure 2B shows the main degrees of freedom of the interface nodes and the signals transmitted between the numerical substructure and the experimental substructure.

Ideally, at each time interval, the numerical substructure is first excited, then the responses $\Psi_{ns4} [4, 16, 28] = [x_{ns4} \ y_{ns4} \ \theta_{ns4}]^T$ and $\Psi_{ns7} [7, 19, 31] = [x_{ns7} \ y_{ns7} \ \theta_{ns7}]^T$ of the interface nodes are excited,

their derivatives (velocity and acceleration) are added to the experimental substructure. Then, the generated experimental restoring forces $\mathbf{f}_{es4} = [F_{es4x} F_{es4y} M_{es4}]^T$ and $\mathbf{f}_{es7} = [F_{es7x} F_{es7y} M_{es7}]^T$ are measured and fed back to the numerical substructure. In addition, the more critical interface conditions in this paper are represented by vector $\Psi_{ns4} = [x_{ns4} y_{ns4} \theta_{ns4}]^T$, $\Psi_{ns7} = [x_{ns7} y_{ns7} \theta_{ns7}]^T$, $\Psi_{es4} = [x_{es4} y_{es4} \theta_{es4}]^T$, and $\Psi_{es7} = [x_{es7} y_{es7} \theta_{es7}]^T$.

The experimental substructure framework used in this paper consists of a horizontal beam and two vertical columns. The beam element is made of a 50 mm × 6 mm plate (web) and two 38 mm × 6 mm plates (flanges), and the column elements are made of A572 Grade 50 structural steel. This experimental substructure framework has been used in the research of Gao and Castaneda (Gao, 2012; Gao et al., 2013b), and Silva (Silva et al., 2020) and others have used this framework to study the benchmark control problem, then the feasibility of this device has been verified.

3.2 Transmission system

Due to the strong non-linearity and MDOF characteristics of the experimental substructure, RTHS with multi-directional loading are required. In addition, due to the high axial stiffness of the columns, the axial deformation of the columns will be neglected, and the vertical degrees-of-freedom along the entire coordinate y direction is not considered. To represent the MDOF response of the experimental substructure in a more realistic way and the hydraulic actuators only provide motion, two hydraulic actuators will be used in the transmission system to equivalently replace the motion of the original frame node 4, as shown in Figure 3. Figure 3A shows that node 4 of the experimental substructure is mainly affected by the rotation and horizontal displacement of the numerical substructure, and Figure 3B shows that at least two hydraulic actuators are required to provide equivalent translational and rotational motion for node 4.

To better apply these two hydraulic actuators to the experimental substructure, this paper adds a coupler attached to the frame between the actuators and the frame, the specific hydraulic actuator and coupler settings refer to the literature cited in (JUC et al., 2023). The two hydraulic actuators directly act horizontally on the coupler, causing it to rotate and translate, thereby inducing equivalent MDOF motions in the frame, as shown in Figure 4.

3.3 MaRTHS scheme

With the numerical simulations model provided by the Benchmark Problem Platform (Silva et al., 2020), maRTHS is carried out, and its simulink diagram is shown in Figure 5. The main task of this scheme is to design a control system to make the measured actuator displacement consistent with the response of the numerical substructure in the actuator coordinates, and to evaluate its tracking performance and the overall performance of RTHS (Silva et al., 2020).

Figure 5 shows that the numerical substructure outputs a displacement vector $\Psi_{ns4} [4, 28] = [x_{ns4} \theta_{ns4}]^T$ under the action of ground motion. It is necessary to use a coordinate transformation function to convert this displacement vector from “frame coordinates

” to “actuator coordinates η_{ns} ,” and then input the transformed actuator coordinates into the control system to obtain the input vector $\mathbf{y}_c = [y_c^1 y_c^2]^T$ for the control actuator. This signal is then transmitted to the corresponding hydraulic actuator, which drives the corresponding experimental substructure and obtains the multi-axis actuator displacement \mathbf{y}_m equivalent to the target displacement vector Ψ_{ns} of the frame node. The multi-axis actuator displacement \mathbf{y}_m is then fed back to the control system. $\hat{\Psi}_m$ is the vector of actuator responses transformed in frame coordinates. Similarly, this “actuator coordinate” is transformed by the coordinate transformation function into “frame coordinates,” and then the feedback force is calculated to obtain the experimental force vector \mathbf{f}_m required to satisfy the equilibrium condition at the interface nodes of the numerical substructure, and this feedback signal is fed back to the numerical substructure.

4 Numerical simulation

In this section, numerical simulations will be conducted using MATLAB/Simulink, with a primary focus on the time delay caused by actuators. In the simulation, the existing time delay compensation methods in the Benchmark Problem Platform are replaced with adaptive time series (ATS) (Chae et al., 2013), ADM, MSMC, and ADM-MSMC time delay compensation methods for RTHS. Experimental and numerical results are provided, followed by a comparison between the experimental and numerical results to validate the feasibility and robustness of the ADM-MSMC time delay compensation method.

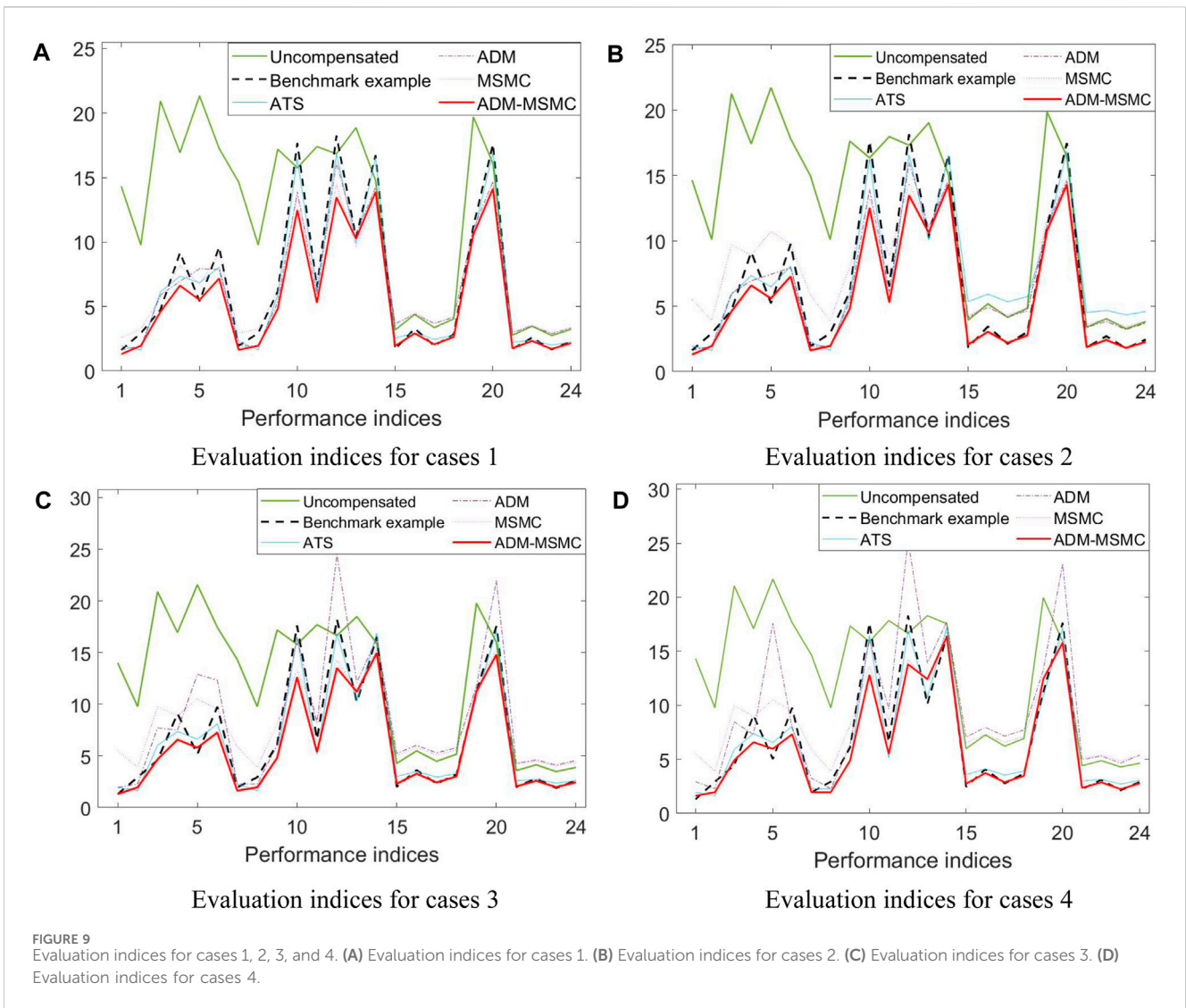
In the Benchmark Problem Platform, the inherent actuator model is represented by transfer functions. In order to evaluate the robustness and stability of the proposed control methods, this paper defines partitioning situations by altering the structural parameters of the reference structure, as shown in Table 1. The variation of modal damping and mass for each layer of the reference structure is considered, resulting in different stability and performance scenarios. The input to the reference and hybrid system for this simulation is the EI Centro earthquake historic record with a scaling factor of 0.40. As shown in Figure 6, the inner-loop PI controller serves as the controller for the original Benchmark Problem Platform, which is in the transmission system as shown in Figure 5. The proportional gain (k_p) and integral gain (k_i) of the PI controller are set to $k_p = 2$ and $k_i = 95$, for detailed definitions, please refer to the literature cited in (Silva et al., 2020).

Where s is Laplace operator.

The parameters used herein for these controllers are determined based on the parameters of the benchmark problem model, as summarized in Table 2 below. In this simulation, the solver was set to a fixed-step configuration with a base sample time of 1/1,024 s. The ODE4 (Runge-Kutta) method was chosen as the solver, providing a balanced approach between computational efficiency and accuracy for the simulation.

4.1 Displacement response

Due to space limitations, this paper only presents the displacement time-history comparison graphs for Case 1 and Case 4. Figures 7, 8 depict the displacement time-history curves



for the different compensation methods in Case 1 and Case 4, respectively. The curves represent a comparison between the displacement time-history curves in the original Benchmark Problem Platform and the ones obtained by replacing the time-delay compensation method with ATS, ADM, MSMC, and ADM-MSMC methods. From Figures 7, 8, it is evident that replacing the original time delay compensation method in the Benchmark Problem Platform with the ADM-MSMC compensation method results in the actual displacement responses of actuator 1 and actuator 2 closely matching the desired displacement responses. Furthermore, compared to replacing with ATS, ADM, and MSMC methods, the actual displacement response curve of the ADM-MSMC method aligns more closely with the desired displacement response curve. This indicates that the proposed ADM-MSMC method exhibits higher compensation accuracy in maRTHS.

4.2 Evaluation criteria values

To quantitatively assess the overall performance of maRTHS, this paper considers: 1) tracking control performance

(minimization of error between target displacement and measured displacement); and 2) global RTHS experimental performance (minimization of error between reference structural response and hybrid system response). This simulation instance involves a set of 10 evaluation criteria, with the J_1 to J_6 assessing the tracking performance of the control system, and the J_7 to J_{10} calculating the global performance of RTHS. The definitions and computation formulas of these criteria are detailed in reference (JUC et al., 2023).

The values of these ten evaluation criteria for four case are computed based on the numerical simulations responses, listed as A1-A4 in Supplementary Appendix SA, with the minimum value in each row bold for easier observation. This section presents line graphs of evaluation indicators analysis for four cases, as depicted in Figure 9, based on the four tables in Supplementary Appendix SA. In the graph, the horizontal axis ranging from 1 to 24 corresponds to performance indicators $J_{1,1}$ to $J_{10,27}$, with the specific numerical values of the performance indicators represented on the vertical axis. From Figure 9, it is evident that the values of J_1 to J_{10} are relatively large without

compensation, but show improvement after the application of compensation methods. Furthermore, it will be observed from **Figure 9** that, except for the MSMC method and without compensation, other compensation methods exhibit smaller time delay between the desired and actual actuator displacements (J_1). Additionally, the ADM-MSMC method demonstrates relatively smaller normalized tracking error (J_2) and maximum peak tracking error (J_3) compared to other methods, indicating that the tracking performance of the ADM-MSMC method is more reliable than the other compensation methods considered in this paper. Moreover, the values of J_4 to J_{10} for the ADM-MSMC method are smaller than those for the other four compensation methods. This suggests that the errors between the displacement responses of the ADM-MSMC method and the computed results of the reference model are also smaller, further validating the feasibility and robustness of the ADM-MSMC method.

5 Conclusion

This paper proposes an ADM-MSMC method to compensate for the delay inherent in maRTHS, applied to the benchmark control problem of maRTHS. The principle and design of the ADM-MSMC method is introduced, with a focus on integrating ADM with MSMC. Numerical simulations of RTHS are conducted on the Benchmark Problem Platform, comparing the results of the ADM-MSMC compensated system with those of the original Benchmark Problem Platform, as well as those compensated using ATS, ADM, and MSMC methods. The numerical and simulation results demonstrate that when the ADM-MSMC method is applied to the maRTHS, the responses after ADM-MSMC compensation are more accurate and closely match the desired responses. Furthermore, the ADM-MSMC method exhibits greater feasibility and robustness compared to the other four compensation methods. Therefore, this method demonstrates a certain level of effectiveness in maRTHS.

Data availability statement

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found in the article/**Supplementary Material**.

References

- Alejandro, P., and Mariantonieta, G. S. (2019). Adaptive tracking control for real-time hybrid simulation of structures subjected to seismic loading. *Mech. Syst. Signal Process.* 134 (12), 106345. doi:10.1016/j.ymssp.2019.106345
- Chae, Y., Kazemibidokhti, K., and Ricles, J. M. (2013). Adaptive time series compensator for delay compensation of servo-hydraulic actuator systems for real-time hybrid simulation. *Earthq. Engng Struct. Dyn.* 42 (11), 1697–1715. doi:10.1002/eqe.2294
- Chae, Y., Rabiee, R., Dursun, A., and Kim, C. Y. (2018). Real-time force control for servo-hydraulic actuator systems using adaptive time series compensator and compliance springs. *Earthq. Engng Struct. Dyn.* 47, 854–871. doi:10.1002/eqe.2994
- Chen, C., Ricles, J. M., and Guo, T. (2012). Improved adaptive inverse compensation technique for Real-Time Hybrid Simulation. *J. Eng. Mech.* 138 (12), 1432–1446. doi:10.1061/(asce)em.1943-7889.0000450

Author contributions

YS: Software, Writing–original draft, Writing–review and editing. ZW: Methodology, Writing–review and editing. YG: Methodology, Software, Data curation, Investigation, Project administration, Resources, Writing–review and editing. YC: Software, Writing–review and editing. YZ: Software, Writing–review and editing. HZ: Writing–review and editing.

Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. Thanks to the National Key Research and Development Program (2022YFC3801201), the National Natural Science Foundation of China (51878630, 52078398), and the Natural Science Foundation of Guangdong Province (2022A1515010500, 2022A1515010338).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fbuil.2024.1393710/full#supplementary-material>

- Hayati, S., and Song, W. (2017). An optimal discrete-time feedforward compensator for real-time hybrid simulation. *Smart Struct. Syst.* 20 (4), 483–498. doi:10.12989/sss.2017.20.4.483
- Horiuchi, T., Inoue, M., Konno, T., and Namita, Y. (1999). Real-time hybrid experimental system with actuator delay compensation and its application to a piping system with energy absorber. *Earthq. Engng Struct. Dyn.* 28, 1121–1141. doi:10.1002/(sici)1096-9845(199910)28:10<1121::aid-eqe858>3.3.co;2-f
- Jiang, Z., Jian, Z. Q., Cheng, G., Ma, J., and Dong, S. M. (2020). Three-dimensional scene construction based on missile semi-physical simulation. *Electron. Des. Eng.* 28 (15), 83–87. doi:10.14022/j.issn1674-6236.2020.15.019
- Juc, W., Manuel, S., Edwin, P., Montoya, H., Dyke, S. J., Silva, C. E., et al. (2023). Experimental benchmark control problem for multi-axial real-time hybrid simulation. *Front. Built Environ.* 9. doi:10.3389/fbuil.2023.1270996
- Liu, Z. (2020). *Real-time hybrid test method and hybrid simulation study of axle vibration system*. Harbin Institute of Technology.
- Najafi, A., Fernandois, G. A., Dyke, S. J., and Spencer, B. F. (2023). Hybrid simulation with multiple actuators: a state-of-the-art review. *Eng. Struct.* 276, 115284. doi:10.1016/j.engstruct.2022.115284
- Najafi, A., and Spencer, B. F. (2021). Multiaxial real-time hybrid simulation for substructuring with multiple boundary Points. *J. Struct. Eng.* 147 (11). doi:10.1061/(asce)st.1943-541x.0003138
- Ning, X., Huang, N. W., Xu, G. S., Wang, Z., Wu, B., Zheng, L., et al. (2023). A novel model-based adaptive feedforward-feedback control method for real-time hybrid simulation considering additive error model. *Struct. Control Health Monit.* 2023, 1–25. doi:10.1155/2023/5550580
- Ou, G., Ozdagli, A. I., Dyke, S. J., and Wu, B. (2015). Robust integrated actuator control: experimental verification and real-time hybrid-simulation implementation. *Earthq. Engng Struct. Dyn.* 44 (3), 441–460. doi:10.1002/eqe.2479
- Ou, J. P. (2003). *Structural vibration control - active, semi-active and intelligent control*. Beijing: Science Publishing House, 71–74.
- Philips, B. M., and Spencer, B. F. (2011). Model-based feedforward-feedback tracking control for real-time hybrid simulation. *NSEL Rep. Ser.* doi:10.1061/(ASCE)ST.1943-541X.0000606
- Rajabi, N., Abolmasoumi, H. A., and Soleymani, M. (2018). Sliding mode trajectory tracking control of a ball-screw-driven shake table based on online state estimations using EKF/UKF. *Struct. Control Health Monit.* 25 (4), e2133. doi:10.1002/stc.2133
- Salvatore, S., and Mario, T. (2016). Actuator dynamics compensation for real-time hybrid simulation: an adaptive approach by means of a nonlinear estimator. *Nonlinear Dyn.* 85 (4), 2353–2368. doi:10.1007/s11071-016-2831-0
- Sarebanha, A., Schellenberg, H. A., Schoettler, J. M., Mosqueda, G., and A. Mahin, S. (2019). Real-time hybrid simulation of seismically isolated structures with full-scale bearings and large computational models. *Comput. Model. Eng. Sci.* 120 (3), 693–717. doi:10.32604/cmescs.2019.04846
- Silva, C. E., Gomez, D., Maghareh, A., Dyke, S. J., and Spencer, B. F. (2020). Benchmark control problem for real-time hybrid simulation. *Mech. Syst. Signal Process.* 135, 106381. doi:10.1016/j.ymssp.2019.106381
- Stoten, D. P., Yamaguchi, T., and Yamashita, Y. (2016). Dynamically substructured system testing for railway vehicle pantographs. *J. Phys. Conf. Ser.* 744, 012204. doi:10.1088/1742-6596/744/1/012204
- Tian, Y. P., Wang, T., Zhou, H. M., and Du, C. (2022). Shaking-table substructure test using a MDOF boundary-coordinating device. *Earthq. Eng. Struct. Dyn.* 51 (15), 3658–3679. doi:10.1002/eqe.3741
- Wallace, M. I., Wagg, D. J., and Neild, S. A. (2005). An adaptive polynomial based forward prediction algorithm for multi-actuator real-time dynamic substructuring. *Proc. R. Soc. A* 461 (2064), 3807–3826. doi:10.1098/rspa.2005.1532
- Wang, Z., Ning, X., Xu, G., Zhou, H., and Wu, B. (2019). High performance compensation using an adaptive strategy for real-time hybrid simulation. *Mech. Syst. Signal Process.* 133, 106262. doi:10.1016/j.ymssp.2019.106262
- Wang, Z., Xu, G., Li, Q., and Wu, B. (2020). An adaptive delay compensation based on a discrete system model for real-time hybrid simulation. *Smart Struct. Syst.* 25 (5), 569–580. doi:10.12989/sss.2020.25.5.569
- Wu, B., Wang, Z., and Bursi, O. S. (2013). Actuator dynamics compensation based on upper bound delay for real-time hybrid simulation. *Earthq. Engng Struct. Dyn.* 42 (12), 1749–1765. doi:10.1002/eqe.2296
- Wu, B., and Zhou, H. (2014). Sliding mode for equivalent force control in real-time substructure testing. *Struct. Control Health Monit.* 21, 1284–1303. doi:10.1002/stc.1648
- Xu, D., Zhou, H., and Wang, T. S. (2019). Performance study of sliding mode controller with improved adaptive polynomial-based forward prediction. *Mech. Syst. Signal Process.* 133, 106263. doi:10.1016/j.ymssp.2019.106263
- Yang, B. Q., Ma, J., and Yao, Y. (2020). Progress and prospect of research on semi-physical simulation equipment for flight vehicles. *Astronautical J.* 41 (06), 657–665. doi:10.3873/j.issn.1000-1328.2020.06.003
- Yang, H. C., Tan, P., Yang, T. Y., and Zhou, F. L. (2021). Acceleration-based sliding mode hierarchical control algorithm for shake table tests. *Earthq. Eng. Struct. Dyn.* 50 (13), 3670–3691. doi:10.1002/eqe.3527
- Yang, H. C., Tan, P., Yang, T. Y., and Zhou, F. L. (2023). Shake table real-time hybrid testing for shear buildings based on sliding mode acceleration control method. *Structures* 52, 230–240. doi:10.1016/j.istruc.2023.03.140
- You, S., Gao, X., and Nelson, A. (2019). “Breaking the testing pyramid with virtual testing and hybrid simulation,” in *Fatigue of aircraft structures*, 1–10.
- Zhou, H., Wagg, D. J., and Li, M. (2017). Equivalent force control combined with adaptive polynomial-based forward prediction for real-time hybrid simulation. *Struct. Control Health Monit.* 24 (11), e2018. doi:10.1002/stc.2018
- Zhou, H., Xu, D., Shao, X., Ning, X., and Wang, T. (2019). A robust linear-quadratic-Gaussian controller for the real-time hybrid simulation on a benchmark problem. *Mech. Eng. Signal Process.* 133 (1), 106260. doi:10.1016/j.ymssp.2019.106260