



This is a repository copy of *Robust model predictive control for dynamics compensation in real-time hybrid simulation*.

White Rose Research Online URL for this paper:
<http://eprints.whiterose.ac.uk/163961/>

Version: Published Version

Article:

Tsokanas, N., Wagg, D. orcid.org/0000-0002-7266-2105 and Stojadinovic, B. (2020) Robust model predictive control for dynamics compensation in real-time hybrid simulation. *Frontiers in Built Environment*, 6. 127. ISSN 2297-3362

<https://doi.org/10.3389/fbuil.2020.00127>

Reuse

This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here:
<https://creativecommons.org/licenses/>

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



eprints@whiterose.ac.uk
<https://eprints.whiterose.ac.uk/>



Robust Model Predictive Control for Dynamics Compensation in Real-Time Hybrid Simulation

Nikolaos Tsokanas^{1*}, David Wagg² and Božidar Stojadinović¹

¹ Institute of Structural Engineering, Department of Civil, Environmental and Geomatic Engineering, ETH, Zurich, Switzerland,

² Department of Mechanical Engineering, University of Sheffield, Sheffield, United Kingdom

OPEN ACCESS

Edited by:

Wei Song,
University of Alabama, United States

Reviewed by:

Richard Christenson,
University of Connecticut,
United States

Ehsan Noroozinejad Farsangi,
Graduate University of Advanced
Technology, Iran

Tao Wang,
China Earthquake Administration,
China

*Correspondence:

Nikolaos Tsokanas
tsokanas@ibk.baug.ethz.ch

Specialty section:

This article was submitted to
Computational Methods in Structural
Engineering,
a section of the journal
Frontiers in Built Environment

Received: 30 April 2020

Accepted: 13 July 2020

Published: 14 August 2020

Citation:

Tsokanas N, Wagg D and
Stojadinović B (2020) Robust Model
Predictive Control for Dynamics
Compensation in Real-Time Hybrid
Simulation. *Front. Built Environ.* 6:127.
doi: 10.3389/fbuil.2020.00127

Hybrid simulation is an efficient method to obtain the response of an emulated system subjected to dynamic excitation by combining loading-rate-sensitive numerical and physical substructures. In such simulations, the interfaces between physical and numerical substructures are usually implemented using transfer systems, i.e., an arrangement of actuators. To guarantee high fidelity of the simulation outcome, conducting hybrid simulation in hard real-time is required. Albeit attractive, real-time hybrid simulation comes with numerous challenges, such as the inherent dynamics of the transfer system used, along with communication interrupts between numerical and physical substructures, that introduce time delays to the overall hybrid model altering the dynamic response of the system under consideration. Hence, implementation of adequate control techniques to compensate for such delays is necessary. In this study, a novel control strategy is proposed for time delay compensation of actuator dynamics in hard real-time hybrid simulation applications. The method is based on designing a transfer system controller consisting of a robust model predictive controller along with a polynomial extrapolation algorithm and a Kalman filter. This paper presents a proposed tracking controller first, followed by two virtual real-time hybrid simulation parametric case studies, which serve to validate the performance and robustness of the novel control strategy. Real-time hybrid simulation using the proposed control scheme is demonstrated to be effective for structural performance assessment.

Keywords: real-time hybrid simulation, model predictive control, actuator dynamics, dynamic response, polynomial extrapolation, Kalman filter, uncertainty propagation

1. INTRODUCTION

Hybrid simulation (HS), also known as hardware-in-the-loop (HIL), online computer-controlled testing technique or model-based simulation, is a dynamic response simulation method. It is based on a step-by-step numerical solution of the governing equations of motions for a model that consolidates both numerical and physical substructures (Schellenberg et al., 2009). It is an efficient technique, since it merges the advantages of numerical simulations with the verisimilitude of experimental testing to form a high-fidelity tool for studying the dynamic response of systems whose size and complexity exceed the capacity of typical testing laboratories. Furthermore, substructures that are complex to model numerically can be tested physically, allowing for real measurements of the output quantities of interest (QoI). Moreover, substructures whose dynamic response is sensitive to the rate of loading can be tested in real-time,

in the so-called real-time hybrid simulation (RTHS). In this way, many assumptions and model distortions made in the process of modeling complicated systems are avoided, increasing the fidelity of and trust in simulation outcomes.

In every HS time step, the numerical substructure generates a command that needs to be followed by the physical substructure to maintain continuity of forces and displacements at the interface. In control engineering, this is known as reference tracking, since the output of the system under control (control plant hereafter) should follow the reference signal (the command). In HS the commanded signal is then transferred to the physical substructure through a transfer system. In most cases, this is an arrangement of linear hydraulic or linear electric actuators. During the test, the dynamic response of the physical substructure is measured and fed back to the numerical substructure, completing the unknown terms of the governing equations of motion of the hybrid model needed to compute the following command for the next time step of the simulation. This feedback loop continues until the end of the HS process. If the command signal is displacement or force, then HS is conducted under displacement or force control, respectively. In structural RTHS applications, displacement and/or force command signals are usually used. However, velocity or acceleration control can also be employed, depending on the application needs.

HS is often conducted on a distorted time scale, with the rate of physical substructure testing slowed down to accommodate the power of the transfer system. Such HSs are the so-called pseudodynamic test (Thewalt and Mahin, 1987). Real-time hybrid simulation (RTHS) is an extension of HS, in which the dynamic boundary conditions at the interfaces between numerical and physical substructures are being synchronized in real-time (Nakashima et al., 1992). Albeit attractive, RTHS comes with numerous challenges. The inherent dynamics of the transfer system used, along with interruptions to the communication between numerical and physical substructures, introduce time delays into the hybrid model, altering the dynamic response of the tested system (Gao and You, 2019). As a result, implementation of adequate control techniques to compensate for such time delays is necessary.

Recently, several control approaches have been proposed to compensate for time delays in RTHS. A selection of these approaches is highlighted below. Horiuchi developed a compensation technique using a polynomial extrapolation methodology to overcome time delays (Horiuchi, 1996), which was later modified into an adaptive scheme (Wallace M. et al., 2005; Wallace M. I. et al., 2005). Phase-lead compensators were also proposed by several authors. These work by compensating for the phase shift of the transfer system (Zhao et al., 2003; Gawthrop et al., 2007; Jung et al., 2007). Another popular compensation method was inverse compensation, in which an inverse model of the transfer function is used as a feedforward compensator—see, for example, Chen and Ricles (2009) and references therein. Following the initial work of Wagg and Stoten (2001) and Neild et al. (2005), adaptive compensation strategies were employed to improve the robustness of RTHS by online estimation of controller parameters (Chae et al., 2013; Chen et al., 2015). Many authors adapted general control methods to RTHS.

For example, Carrion and Spencer developed a method using model-based and LQG algorithms (Carrion and Spencer, 2007). Phillips and Spencer further enhanced this method by adding feedforward and feedback terms, accounting for multi-actuator schemes as well (Phillips and Spencer, 2013a,b). H_∞ loop shaping controller designs were also proposed as an additional technique to improve the performance and robustness of RTHS under the presence of uncertainties in the experimental procedure (Gao et al., 2013; Ou et al., 2015; Ning et al., 2019). Lately, a self-tuning nonlinear controller based on a combined robust-adaptive scheme was proposed, aiming at capturing nonlinearities of the dynamic interaction between transfer systems and physical substructures (Maghareh et al., 2020). Recently, Condori et al. (2020) proposed a robust control approach with a nonlinear Bayesian estimator to address uncertain nonlinear systems.

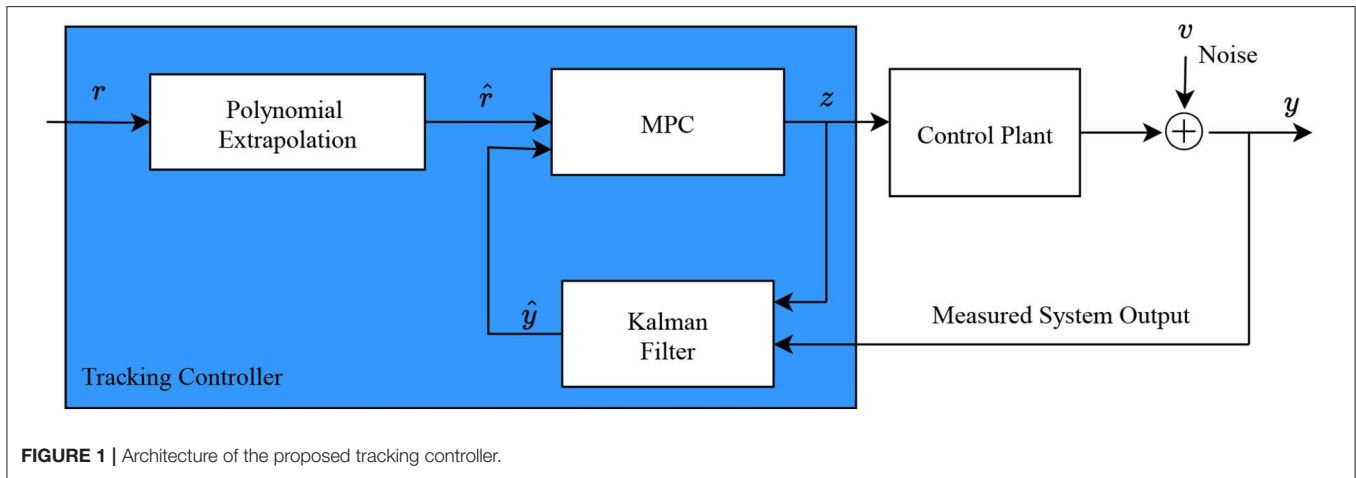
In this study, a novel control method is proposed, in which the tracking controller consists of a robust model predictive controller (MPC) along with a polynomial extrapolation algorithm and a Kalman filter. One important advantage of MPC is its capability to adapt the control law online, compensating for time delays and uncertainties for a set of specific simulation time steps. This is of significant importance for RTHS, since experimental errors and actuator dynamics introduce arbitrarily delays in the system, which need to be compensated for online. Another significant advantage of MPC is the fact that it can perform online optimization, handling at the same time constraints of the system under consideration. Following the design formulation of the proposed tracking controller, two virtual RTHS (vRTHS) parametric case studies are examined in order to validate the performance and robustness of the proposed control scheme. Variations in the parameters of the hybrid model will prove the robustness of the proposed controller to uncertainties introduced throughout the RTHS procedure. RTHS using the proposed control scheme is demonstrated to be effective for structural seismic performance assessment.

2. THE TRACKING CONTROLLER

In this section, the architecture of the proposed tracking controller is explained. The controller consists of a robust MPC along with a polynomial extrapolation algorithm and a Kalman filter. In **Figure 1**, the tracking controller's block diagram is shown. In the following sections, the main parts of the controller are described in detail. The control plant corresponds to the system under consideration, namely the actuator in series with the physical substructure used within the RTHS framework. The subsequent vRTHS case studies will give more insight into the control plant dynamics and architecture.

2.1. Model Predictive Control

Model Predictive Control (MPC) is a control strategy in which the ongoing control law is adapted by computing, at every control interval, a finite horizon optimization problem, applying the ongoing state of the control plant as the initial state. The optimization generates an optimal control sequence consisting of a series of individual control laws, out of which the first one is applied to the control plant for the current control



interval (Mayne et al., 2000). The control interval is defined as a sampling instant or, in other words, as a set of continuous time steps of the simulation, serving as an internal time step for the MPC in order to gather sufficient feedback measurements to accurately predict future control plant outputs and to advance the optimization to the next control sequence. In MPC, a standard finite horizon optimal control problem is being solved similar to H_∞ and LQR control approaches. In H_∞ and LQR control, the optimal problem could be of infinite horizon as well, while that's not the case for MPC. What differs, nonetheless, is the fact that in MPC new control laws are computed in each control interval, whilst in classical control theory a single control law, which is computed offline prior to the simulation, is used for the whole duration of the simulation. This is the fundamental difference between MPC and classical control theory. Online control law derivation is also a feature of adaptive control theory. However, in the latter, conducting system identification needed for the adaptive controller can cause numerical delays, whilst in MPC the model used in the controller remains the same and therefore no identification is needed. Changing online the parameters of the model used in MPC would result in the so-called adaptive MPC, but this is not part of this study.

Every application imposes mandatory (hard) constraints. For example: (i) actuators are of limited stroke/capacity meaning that the produced displacement/force is limited; and/or (ii) safety limits are applied in almost every experimental setup. The problem of meeting hard constraints in control applications is well established in the literature. MPC has proven to be one of the few adequate control methodologies to suitably satisfy constraints on the inputs, states and/or outputs of the system under consideration, maintaining concurrently the desired performance (Zafiriou, 1990).

The proposed tracking MPC controller consists of four elements; (a) the prediction model, (b) the performance index or cost function, (c) the constraints, and (d) a solver to derive the control laws. The prediction model serves as the core of MPC since it is responsible for the future predictions of the control plant outputs, taking into account the past and present values of the computed optimal control laws. The prediction model should

be as accurate as possible in order to be able to sufficiently capture the control plant dynamics and its behavior. Therefore, a detailed prediction model could improve MPC performance. However, there is a trade-off between the complexity of the prediction model and the computational power needed to compute it at every control interval. Care must be taken in order not to introduce delays due to numerical calculations, especially in real-time applications, as RTHS, in which timing is crucial.

The MPC methodology used in this study is described below and illustrated in **Figure 2B**. In **Figure 2A**, the structure of the MPC controller is shown. At each control interval k , MPC optimizes the control plant outputs y_j . Namely, the future outputs $\hat{y}_j(k+i|k)$, for $k=0, \dots, P$ of a predefined prediction horizon P are predicted at each control interval k using the prediction model. The i -th prediction horizon step is a time instant of the current control interval k . The latter depends on the known values up to this k and on the future control laws $u(k+i|k)$, for $k=0, \dots, P-1$. The control sequence $z_k^T = [u(k|k)^T \dots u(k+i|k)^T \dots u(k+P-1|k)^T]$ consists of a sequence of control laws $u(k+i|k)$. It is calculated by optimizing a quadratic cost function at each k . The cost function embodies the tracking error, i.e., the error between the reference trajectory and the predicted output values of the control plant, and is expressed as follows:

$$J^*(\hat{r}_k, \hat{y}_k, z_k) = \sum_{j=1}^{n_y} \sum_{i=1}^P \{w^{y_j} [\hat{r}_j(k+i|k) - \hat{y}_j(k+i|k)]\}^2 + \sum_{j=1}^{n_u} \sum_{i=0}^{P-1} \{w^{u_j} [u_j(k+i|k) - u_j(k+i-1|k)]\}^2 \quad (1)$$

where n_y correspond to the number of control plant outputs, n_u the number of control plant inputs, $\hat{r}_j(k+i|k)$ the reference value to be tracked at the i -th prediction horizon step from the j -th control plant output, $\hat{y}_j(k+i|k)$ the predicted value of the j -th control plant output at the i -th prediction horizon step, $u_j(k+i|k)$ the j -th control plant input at the i -th prediction horizon step, w^{y_j} the tuning weight of the j -th control plant output and w^{u_j} the

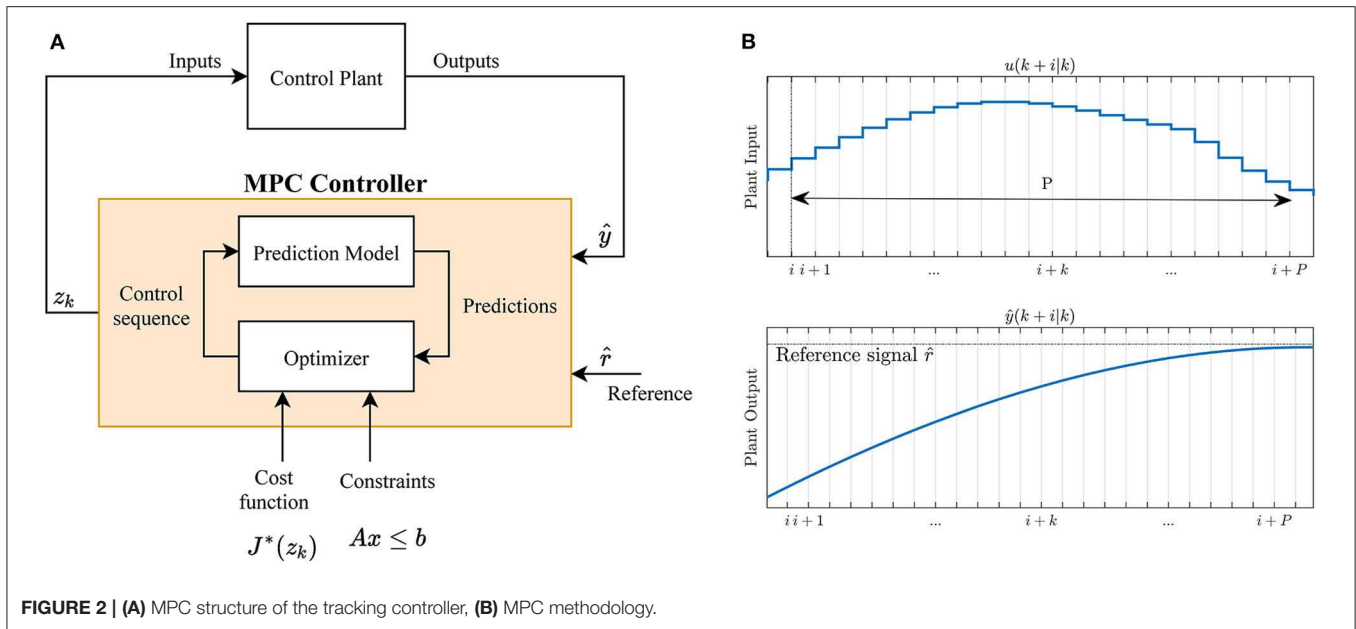


FIGURE 2 | (A) MPC structure of the tracking controller, (B) MPC methodology.

tuning weight of the j -th control plant input. $(i|k)$ represents the current step i of the prediction horizon P at the control interval k .

Additionally, in the proposed tracking controller an output disturbance model w is used as described in Equation (2). The input of the model u_w , is assumed to be white noise and the disturbance w is additive to the control plant outputs. The disturbance model is used to include potential unmeasured noise that could occur during RTHS, e.g., experimental measurement errors. In the proposed design, the disturbance model follows:

$$\begin{aligned} x_w(k+1) &= A_w x_w(k) + B_w u_w(k) \\ w(k) &= C_w x_w(k) + D_w u_w(k) \end{aligned} \quad (2)$$

where A_w, B_w, C_w , and D_w are matrices associated with the disturbance w .

The k, P, w^{ij} , and w^{ij} parameters from Equation (1) remain constant for the entire RTHS. MPC constantly receives reference trajectories, $\hat{r}_j(k+i|k)$, for the whole prediction horizon P , which in RTHS corresponds to the outputs of the numerical substructure and uses the prediction model along with the Kalman filter (see section 2.2) to predict the control plant outputs, $\hat{y}_j(k+i|k)$, which depend on the control sequence z_k , the disturbance $w(k)$ and the Kalman filter's estimates. The control sequence z_k is computed in the optimizer (see Figure 2A), which takes into account the cost function (and in essence the tracking error, as it's embedded in the cost) and the constraints. The quadratic cost function of Equation (1) can be transformed into a Quadratic Programming (QP) problem (Delbos and Gilbert, 2003; Tøndel et al., 2003) and this is what is essentially being solved in the optimizer. The QP problem is formulated as follows:

$$\min_x \left(\frac{1}{2} x^T H x + f^T x \right) \quad (3)$$

$$\text{subject to } Ax \leq b \quad (4)$$

where the $Ax \leq b$ inequality corresponds to the constraints applied, x is the solution vector, H the Hessian matrix, A is a matrix of linear constraints coefficients, b is a vector relevant with the constraints, and f is a vector obtained by:

$$f = Kx^T x_{kf}(k|k) + Kr(k|k)^T r(k|k) + Ku(k|k-1)^T u(k|k-1) \quad (5)$$

where $x_{kf}^T = [x_p^T \ x_w^T]$ is the vector corresponding to the states of the Kalman filter (see section 2.2) and consists of the control plant states x_p and of the disturbance w states x_w , $r(k|k)$ is the reference signal at the current control interval, $u(k|k-1)$ is the control law applied to the control plant in the previous control interval and K a weighting factor.

In the proposed tracking controller, an active-set solver applying the KWIK algorithm (Schmid and Biegler, 1994) is used for solving the QP problem. This is a built-in QP solver from the Model Predictive Toolbox of MATLAB, used in this study to derive the control law sequence.

The MPC algorithm used in the proposed tracking controller can be summarized as follows:

1. Assuming the output disturbance model from Equation (2), consider a discrete-time multiple-input-multiple-output (MIMO), linear time invariant (LTI) system, representing a linearized model of the control plant:

$$\begin{aligned} x_p(k+1) &= A_p x_p(k) + B_p u(k) \\ y(k) &= C_p x_p(k) + D_p u(k) + D_{pw} w(k) \end{aligned} \quad (6)$$

where A_p, B_p, C_p, D_p , and D_{pw} are matrices corresponding to the control plant. This is the prediction model used along with estimates from the Kalman filter (see section 2.2) to provide MPC with predictions of future control plant outputs.

2. MPC performs the optimization at every control interval $k = 0, 1, \dots$:

$$\min_{z_k} \mathbf{J}^*(\hat{r}_k, \hat{y}_k, z_k) \quad (7)$$

$$\text{subject to } u_{j_{\min}} \leq u_j \leq u_{j_{\max}} \quad \text{and} \quad (8)$$

$$\dot{u}_{j_{\min}} \leq \dot{u}_j \leq \dot{u}_{j_{\max}} \quad (9)$$

where the above constraints correspond to the physical limitations of the actuator regarding displacement and velocity capacity and $\mathbf{J}^*(\hat{r}_k, \hat{y}_k, z_k)$ to the cost of Equation (1). The above limitations/capacities of the actuator are implemented as internal hard constraints of MPC. As a result, MPC guarantees, for the case studies addressed in section 3, that the performance of the controller is not affected by how close the actuator is to its limits.

3. The control sequence $z_k^T = [u(k|k)^T \dots u(k+i|k)^T \dots u(k+P-1|k)^T]$ is obtained for every control interval k :

$$z_k := \arg \min \mathbf{J}^*(\hat{r}_k, \hat{y}_k, z_k) \quad (10)$$

from the QP solver and it's applied to the control plant.

4. Steps 1–3 are repeated until the end of the RTHS.

In RTHS, the uncertainties and experimental errors are neither constant nor predictable. MPC enables computing a new control law for every control interval within the simulation time, making it possible to compensate specifically for the incurred time delays, uncertainties and/or experimental errors that are introduced in RTHS at each specific control interval. In contrast, classical control techniques utilize a single pre-computed control law that is robust enough to compensate for all the delays coming into play in the entire simulation process. In addition, RTHS always involves experimental equipment, which has physical boundaries, e.g., limited actuator force capacity. Hence, the command signals must be limited to satisfy these boundaries. MPC can solve optimization problems and concurrently satisfy hard constraints, which in the RTHS case, can be laboratory limitations. The aforementioned points make MPC desirable and suitable for RTHS applications. In the case studies presented in the following sections, the selection of the control interval k , the prediction horizon P , and the weights w^{y_j} and w^{u_j} is made through trial and error as there exists a trade-off between optimal controller performance and computational effort. Selection of the above MPC parameters is case study dependent. However, for control interval k and prediction horizon P , the following rules may be applied as a first trial (Bemporad et al., 2020):

- obtain each k at a sampling rate T_s , between 10 and 25% of the minimum desired closed-loop response time. A radical decrease of T_s will result in computational effort increase. T_s cannot be smaller than the sampling rate of RTHS.
- set P such that the desired closed-loop response time T , is approximately equal to PT_s , and the controller is internally stable.
- further optimization of the controller should be done through tuning of the weight coefficients w , but not through tuning of P .

MPC theory is quite extensive, covering various subjects (e.g., convex optimization, optimal control theory, computational solvers) that are taken into account during the design and implementation process of MPC and are not described in full detail in this paper. For a more comprehensive literature in MPC, the reader is encouraged to consult (Bitsoris, 1988; Rossiter, 2003; Boyd and Vandenberghe, 2004; Camacho and Bordons, 2007; Rawlings et al., 2017).

2.2. Kalman Filter

As mentioned above, good accuracy of the predicted control plant outputs is significant as it affects the performance of MPC. In order to improve the predictions' accuracy, a Kalman filter is implemented to estimate the future control plant output values. The purpose of the Kalman filter is to estimate how the current control law will alter the future control plant outputs and use these estimations to optimize the control sequence. The Kalman filter state-space formulation used in MPC follows:

$$x_{kf}(k+1) = A_{kf}x_{kf}(k) + B_{kf}u_{kf}(k) \quad (11)$$

$$\hat{y}(k) = C_{kf}x_{kf}(k) + D_{kf}u_{kf}(k)$$

where

$$A_{kf} = \begin{bmatrix} A_p & 0 \\ 0 & A_w \end{bmatrix}, \quad B_{kf} = \begin{bmatrix} B_p & 0 \\ 0 & B_w \end{bmatrix} \quad (12)$$

$$C_{kf} = [C_p \ C_w], \quad D_{kf} = [0 \ D_w] \quad \text{and}$$

$$u_{kf}^T = [z_k^T \ u_w(k)^T]$$

The weighting coefficients for the Kalman filter are derived from the following expectations:

$$Q = \mathbb{E}[B_{kf}u_{kf}u_{kf}^T B_{kf}^T], \quad R = \mathbb{E}[D_{kf}u_{kf}u_{kf}^T D_{kf}^T], \\ N = \mathbb{E}[B_{kf}u_{kf}u_{kf}^T D_{kf}^T] \quad (13)$$

In **Figure 1** is illustrated how the Kalman filter is integrated within the proposed tracking controller. More specifically, in the beginning of each control interval k , the state of the Kalman filter, $x_{kf}^T = [x_p^T \ x_w^T]$, is estimated for the next interval as follows:

- $x_{kf}(k|k)$ is updated based on the latest measurements:

$$x_{kf}(k|k) = x_{kf}(k|k-1) + M[y(k) - C_{kf}x_{kf}(k|k-1)] \quad (14)$$

- The state for the next, $k+1$, control interval is estimated as:

$$x_{kf}(k+1|k) = A_{kf}x_{kf}(k|k-1) + B_p u(k) \\ + L[y(k) - C_{kf}x_{kf}(k|k-1)] \quad (15)$$

where L , M are the Kalman filter gain matrices and $u(k)$ the optimal control law assumed to be used from the control interval $(k-1)$ until k . $y(k)$ is the measured control plant output at the control interval k .

Once the state for the $k + 1$ interval is estimated, the values of the control plant output at this interval can be predicted as follows:

- For any successive step, $i = 1 : P$, within the ongoing control interval k , the next state is estimated as:

$$x_{kf}(k + i|k) = A_{kf}x_{kf}(k + i - 1|k) + B_p u(k + i - 1|k) \quad (16)$$

- Hence, the predicted control plant output value is calculated as:

$$\hat{y}(k + i|k) = C_{kf}x_{kf}(k + i|k), \quad \text{for } i = 1, \dots, P \quad (17)$$

where i corresponds to the prediction horizon step.

2.3. Polynomial Extrapolation

MPC can guarantee adequate tracking performance and robustness under uncertainties and disturbances. However, since in RTHS even small tracking errors can significantly alter the simulation outcome, a fourth-order polynomial extrapolation (Horiuchi, 1996; Wallace M. et al., 2005; Wallace M. I. et al., 2005; Ning et al., 2019) is integrated in the tracking controller as illustrated in **Figure 1**, in order to further compensate for time delays and additionally improve the MPC performance. Its formulation follows:

$$\hat{r}_k = a_0 r_{(0,k)} - a_1 r_{(1,k)} + a_2 r_{(2,k)} - a_3 r_{(3,k)} + a_4 r_{(4,k)} \quad (18)$$

where $r_{(i,k)} = r(t_k - iT_d)$ is the discrete reference signal by adding shifts of a pure time delay T_d by integer values of i . The polynomial coefficients $a_0 - a_4$ are obtained using the Lagrange basis function by trial and error.

3. CASE STUDIES

The following two virtual RTHS parametric case studies (CS) serve as validation for the performance and robustness of the proposed tracking controller. The case studies are virtual in that both the physical substructures of the hybrid models are implemented numerically in software, not physically as specimens in a laboratory. This was done to facilitate the development and testing of the proposed MPC. For each case study, the dynamics of the tested system are explained, then the tracking controller design properties are addressed and finally, results are presented. Since the goal of each case study is to examine the behavior of the tracking controller, the outputs of the hybrid models are exclusively related to the controller's performance. The outputs will be:

1. Tracking time-delay, defined as:

$$J_1 = \left(\arg \max_k (\text{Corr}(r(i), y(i - k))) \right) f_{RTHS} \quad [\text{msec}] \quad (19)$$

where f_{RTHS} is the sampling frequency of RTHS.

2. Normalized Root Mean Square (NRMS) of the tracking error, defined as:

$$J_2 = \sqrt{\frac{\sum_{i=1}^N [y(i) - r(i)]^2}{\sum_{i=1}^N [r(i)]^2}} \times 100 \quad [\%] \quad (20)$$

3. Peak Tracking Error (PTE), defined as:

$$J_3 = \frac{\max |y(i) - r(i)|}{\max |r(i)|} \times 100 \quad [\%] \quad (21)$$

J_1 is established as the maximum cross-correlation between the reference and the measured signal, multiplied by the sampling frequency of RTHS. It is a metric of how different in time these two signals are. The cross-correlation describes how many time steps the measured signal should be shifted in order to match the reference. When $J_1 > 0$ the measured signal is delayed with respect to the reference (tracking delay), whilst when $J_1 < 0$, the measured signal is leading the reference (overcompensation). The desire is to have zero time tracking delay, meaning the value of J_1 to be as close to zero as possible, without overcompensating. J_2 represents how quantitatively different the reference and measured signals are accounting for the whole simulation period, whilst J_3 accounts only for the maximum value of the tracking error. The performance of the tracking controller is assessed by how close to zero J_1 , J_2 , and J_3 are (Silva et al., 2020).

3.1. CS1: vRTHS of a Structure With an Attached Pendulum

3.1.1. Problem Formulation

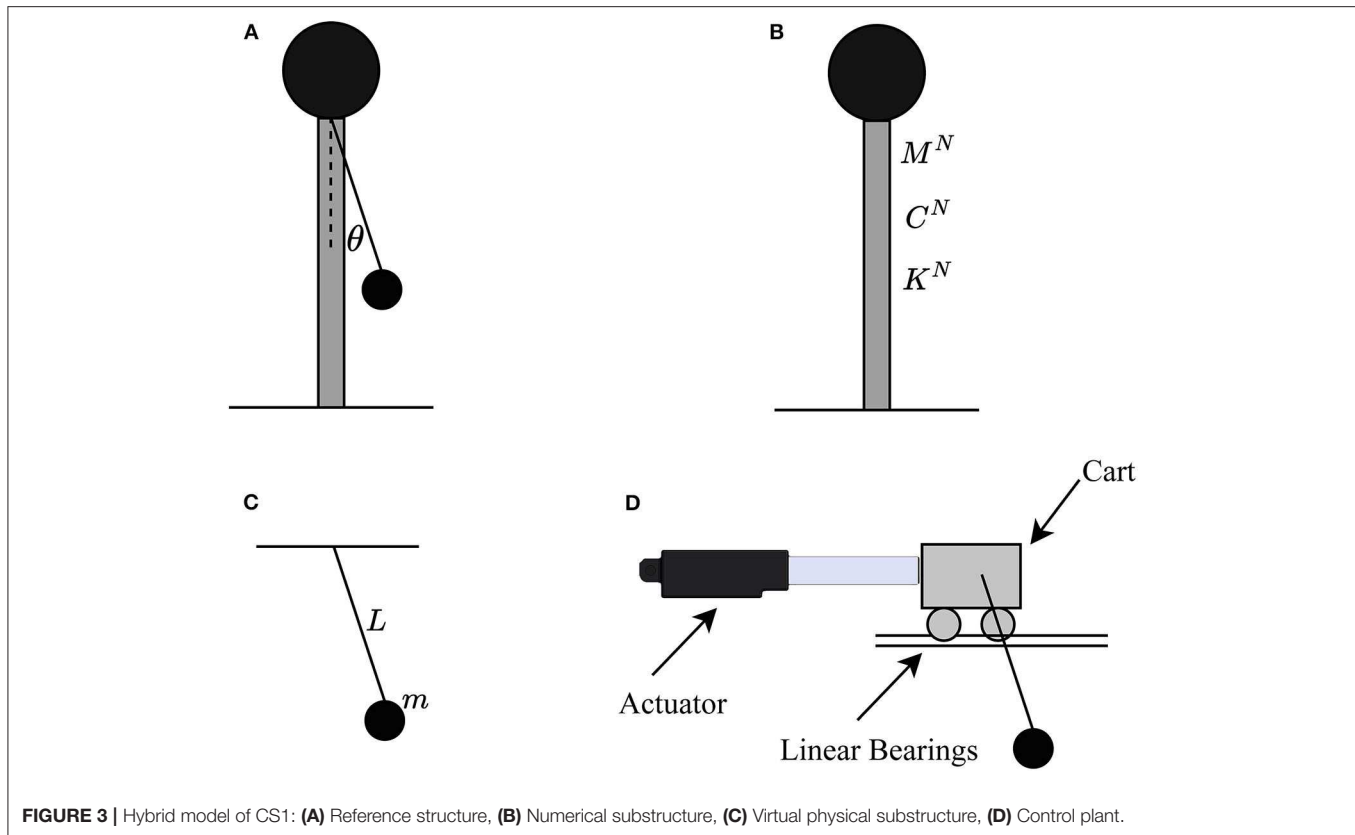
The reference system under consideration for CS1 corresponds to a vertical cantilever beam with mass concentrated at its top, and a pendulum attached to the center of gravity of the cantilever mass, as shown in **Figure 3A**. The numerical substructure is the cantilever beam (**Figure 3B**), described by Equation (22), while the virtual physical substructure is the pendulum (**Figure 3C**).

The Equation of Motion (EoM) for the reference structure follows:

$$M^N \ddot{x} + C^N \dot{x} + K^N x = -M^N \ddot{x}_g + f^P \quad (22)$$

where \ddot{x} , \dot{x} , and x correspond to acceleration, velocity and displacement of the numerical substructure relative to the ground, $M^N = 100$ [Kg], $C^N = 100$ [$\frac{Ns}{m}$], and $K^N = 10e4$ [$\frac{N}{m}$] are mass, damping and stiffness of the numerical substructure respectively, \ddot{x}_g is the ground motion applied to the hybrid model and f^P the force measured from the virtual physical substructure.

The virtual physical substructure corresponds only to the pendulum. However, to move the pendulum pivot point horizontally in a lab, an actuator could be attached to a cart mounted on a horizontal rail. Thus, the cart and the actuator would be the transfer system. As a result, the cart dynamics and its interaction with the pendulum are taking into account for solving the equations for the virtual physical substructure. The virtual physical substructure is described by the Equations (23) and (24). Moreover, in order to reduce as much as possible the



friction due to the cart movement μ , low friction linear bearings are assumed to be implemented. The friction at the pendulum pivot, b , is assumed to be small.

The EoM for the cart with the pendulum follows:

$$\ddot{x} = \frac{mL^2\dot{\theta}^2 \sin(\theta) + mLg \sin(\theta) \cos(\theta) + \dot{\theta}b \cos(\theta) - \mu L\dot{x} + f^P L}{L(m \sin^2(\theta) + M)} \quad (23)$$

$$\ddot{\theta} = \frac{-mL^2\dot{\theta}^2 \sin(\theta) \cos(\theta) - \dot{\theta}b(1 + \frac{M}{m}) - gL(M + m) \sin(\theta) + \dot{x}L\mu \cos(\theta) + f^P L \cos(\theta)}{L^2(m \sin^2(\theta) + M)} \quad (24)$$

where the parameters from Equations (23) and (24) correspond to:

- Pendulum angle, angular velocity, angular acceleration $\rightarrow \theta, \dot{\theta}, \ddot{\theta}$, respectively
- Cart position, velocity, acceleration $\rightarrow x, \dot{x}, \ddot{x}$, respectively
- Force generated from the pendulum $\rightarrow f^P$
- Pendulum mass $\rightarrow m = 0.15$ [Kg]
- Cart mass $\rightarrow M = 2$ [Kg]
- Rod length $\rightarrow L = 0.7$ [m]
- Cart friction coefficient $\rightarrow \mu = 0.001$ [-]
- Pendulum friction coefficient $\rightarrow b = 0.0001$ [-]
- Acceleration of gravity $\rightarrow g = 9.81$ [$\frac{m}{s^2}$].

Since this is a virtual simulation, an actuator model needs to be implemented representing the dynamics of the real actuator.

For CS1, a linear hydraulic actuator was chosen. Its model consists of three transfer functions; (i) G_{sv} represents the servo-valve dynamics as in Equation (25), (ii) G_a the actual actuator dynamics as in Equation (26), and (iii) G_{CSI} the control-structure-interaction (CSI) (Dyke et al., 1995) as in Equation (27). The way these transfer functions are interconnected is shown using a block diagram of the actuator model in Figure 4. Taking the above into consideration, the control plant for CS1, corresponds to the actuator model along with the cart and the pendulum. A graphical representation of the control plant is illustrated in Figure 3D and its block diagram in Figure 4. The control plant is a single-input-multiple-output (SIMO) model with input the displacement of the actuator z and four outputs; x, \dot{x}, θ , and $\dot{\theta}$. In the tracking controller, only the first output, the cart position x is used, described by a single-input-single-output (SISO) transfer system as in Equation (28).

$$G_{sv} = \frac{2.128e13}{s^2 + 425s + 99976} \quad (25)$$

$$G_a = \frac{1}{s + 3.3} \quad (26)$$

$$G_{CSI} = 7.26e5s \quad (27)$$

$$G_{cp} = \frac{0.5s^2 + 0.0007823s + 8.058}{s^4 + 0.001963s^3 + 15.07s^2 + 0.007s} \quad (28)$$

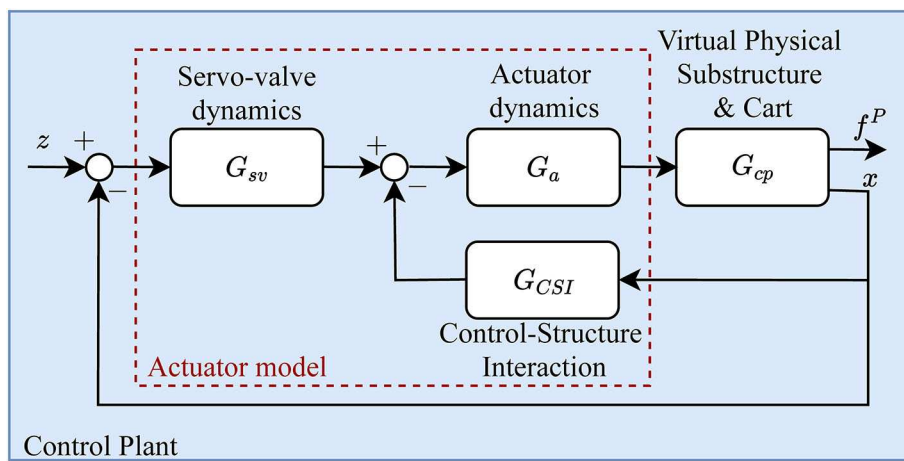


FIGURE 4 | Block diagram of the control plant in CS1.

The block diagram of the overall hybrid model is presented in **Figure 5A**. It consists of the numerical and virtual physical substructures, the proposed tracking controller and the control plant. The RTHS is conducted in displacement control; in every time step the measured displacement of the cart (same as the horizontal displacement of the pendulum), x , is fed back in the tracking controller, while the measured force generated from the movement of the pendulum, f^P is fed back to the numerical substructure to compute the next displacement command r . The coupling of the two substructures is achieved through force f^P . Furthermore, apart from the disturbance described in Equation (2), in order to capture even more realistic results, additional white noise v is added in the calculated displacement x and force f^P , representing measurement noise from the displacement and force sensors, respectively. The displacement and force sensor noise is modeled with two correlated standard Gaussian distributions, generated at the same frequency as the sampling of RTHS and amplified each by $1.5e-7$ m and $6e-5$ N, respectively, which approximately equals to 0.01% of the respective full spans. The sampling frequency of RTHS was set to $f_{RTHS} = 4,096$ Hz. For the numerical integration scheme, the RK4 (fourth-order Runge–Kutta) method is used with a fixed time step of $1/4,096$ s. The reference ground motion of the hybrid model \ddot{x}_g is a historical acceleration record from the El Centro 1940 earthquake downscaled by 0.4, as shown in **Figure 6A**. In **Figure 6B**, the power spectral density of the respective record is illustrated.

3.1.2. Tracking Controller Design Properties

The prediction model used in MPC for CS1 is a linearized model of the control plant. Since MPC functions in discrete time, the linearized model of the control plant is discretized with the sampling frequency of RTHS, f_{RTHS} . Essentially it's a discrete LTI SISO model described using the state-space formulation as follows:

$$\begin{aligned} x(k+1) &= A_p x(k) + B_p u(k) \\ y(k) &= C_p x(k) + D_p u(k) + D_{pw} w(k) \end{aligned} \quad (29)$$

where A_p , B_p , C_p , D_p , and $D_{pw} = 1$ are the prediction model matrices, equal to:

$$\begin{aligned} A_p &= \begin{bmatrix} 0.4762 & -0.3993 & -0.2475 & -0.2307 & -0.2279 & -0.0009 & -0.0143 \\ 0.7532 & 0.7912 & -0.1315 & -0.1222 & -0.1272 & -0.0005 & -0.008 \\ 0.2103 & 0.4645 & 0.9775 & -0.0209 & -0.0223 & 0 & -0.0014 \\ 0.0184 & 0.0603 & 0.2486 & 0.9987 & -0.0014 & 0 & 0 \\ 0.0012 & 0.0051 & 0.0312 & 0.2499 & 1 & 0 & 0 \\ 0 & 0 & 0.0002 & 0.002 & 0.0156 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0.0001 & 0.0156 & 1 \end{bmatrix}, \\ B_p &= \begin{bmatrix} 0.0118 \\ 0.0066 \\ 0.0012 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \\ C_p &= [0 \ 0 \ 0 \ 0 \ 19.86 \ 0.002 \ 1.25], D_p = [0] \end{aligned} \quad (30)$$

The disturbance model used, expressed by Equation (2), is added to the control plant output and its model follows:

$$\begin{aligned} x_w(k+1) &= x_w(k) + 0.0009766u_w(k) \\ w(k) &= x_w(k) \end{aligned} \quad (31)$$

with $A_w = 1$, $B_w = 0.0009766$, $C_w = 1$, and $D_w = 0$. The Kalman filter gain matrices follow:

$$\begin{aligned} L &= [0.0096 \ -0.0062 \ -0.0269 \ 0.0021 \ 0.0148 \ 0.0004 \ -0.0003 \ 0.0008]^T \\ M &= [0.0071 \ -0.0147 \ -0.0215 \ 0.0082 \ 0.0135 \ 0.0002 \ -0.0003 \ 0.0008]^T \end{aligned} \quad (32)$$

Starting with Equations (30), (31), and (32) the derivation of the Kalman filter formulation in Equation (11) is straightforward. The MPC weight coefficients used in Equation (1) are selected

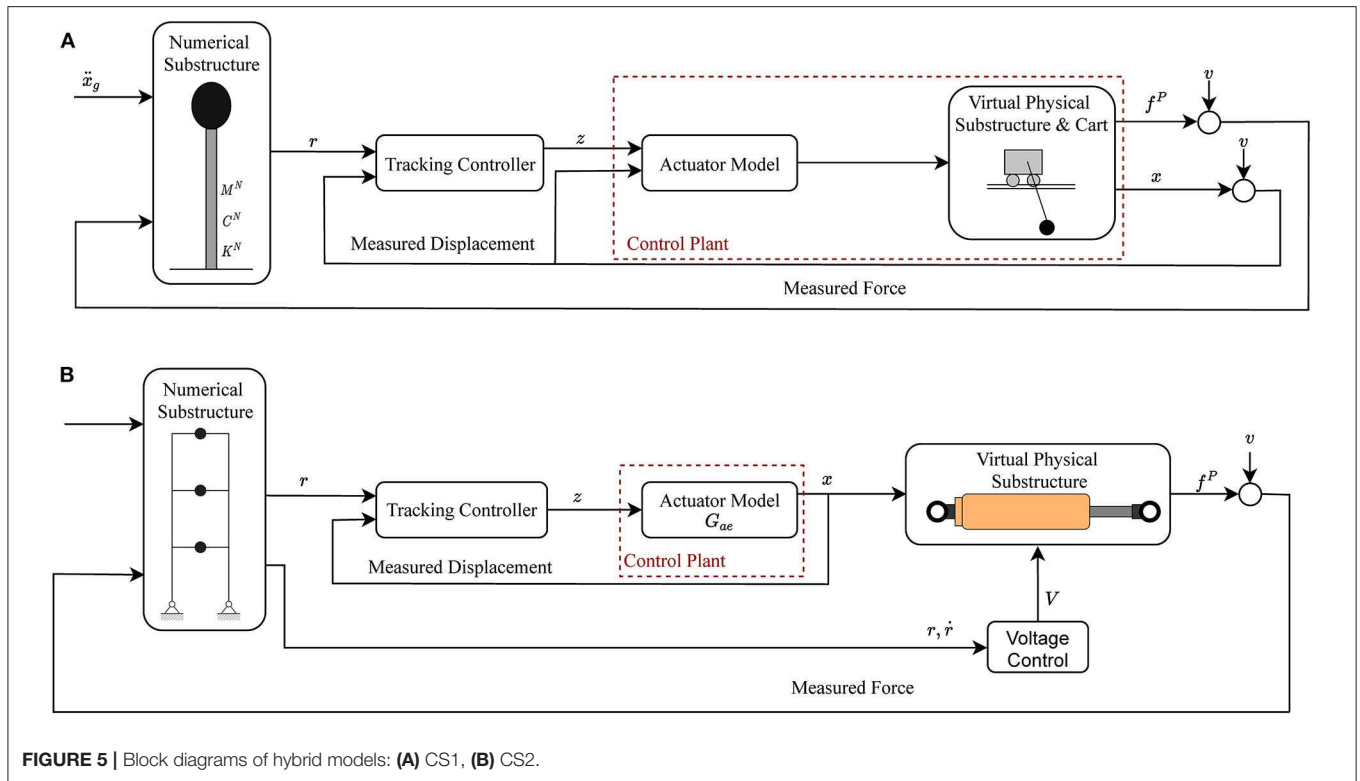


FIGURE 5 | Block diagrams of hybrid models: (A) CS1, (B) CS2.

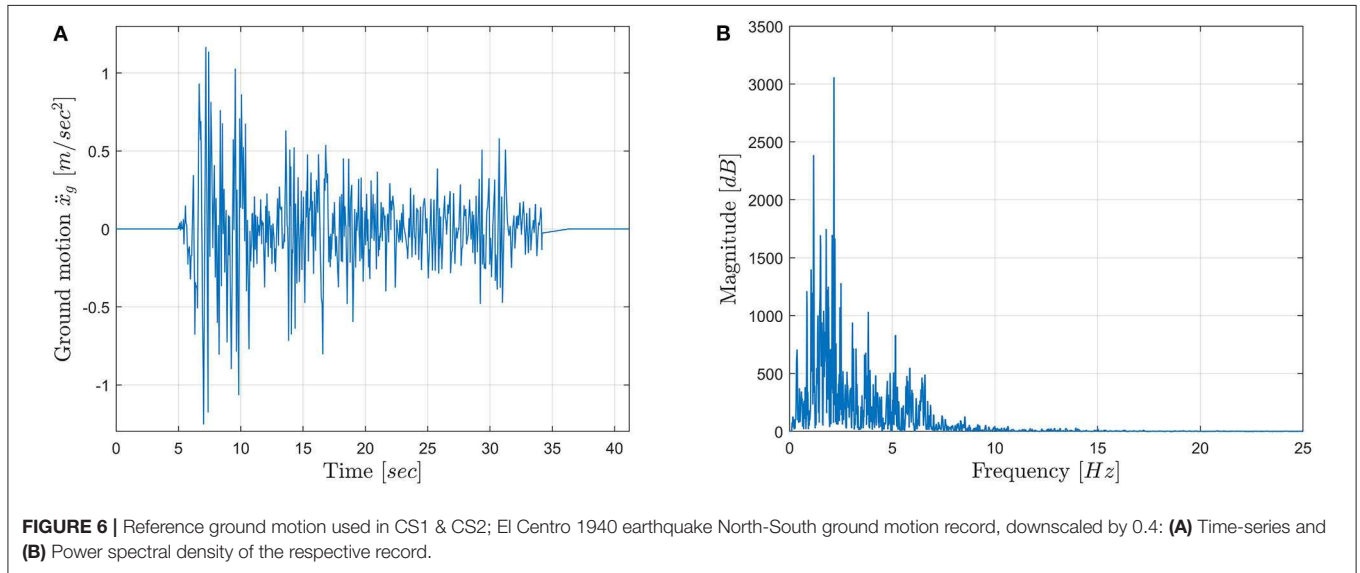


FIGURE 6 | Reference ground motion used in CS1 & CS2; El Centro 1940 earthquake North-South ground motion record, downscaled by 0.4: (A) Time-series and (B) Power spectral density of the respective record.

to be $w^y = 15.26$ and $w^u = 0.63$. The number of control plant outputs n_y is 1 and the number of control plant inputs n_u , is also 1. The prediction horizon was set to $P = 8$ and each control interval k was obtained at a sampling frequency of $1,024\text{Hz}$, one fourth of the RTHS sampling rate. The constraints applied represent the physical limitations of the actuator to provide bounded displacements and velocity. It's assumed that the virtual actuator has a maximum stroke of $\pm 250\text{ [mm]}$ and maximum velocity of $\pm 100\text{ [mm/sec]}$. So the

constraints follow:

$$\begin{aligned} -250 \leq \hat{z}(k+i|k) &\leq 250, & \left[\begin{matrix} mm \\ \end{matrix} \right] & \text{ for } i = 1, \dots, P & (33) \\ -100 \leq \dot{\hat{z}}(k+i|k) &\leq 100, & \left[\begin{matrix} mm \\ sec \end{matrix} \right] & \text{ for } i = 1, \dots, P \end{aligned}$$

The polynomial extrapolation coefficients used in CS1 for the proposed tracking controller follow:

$$\hat{r}_k = 5r_{(0,k)} - 10r_{(1,k)} + 10r_{(2,k)} - 5r_{(3,k)} + r_{(4,k)} \quad (34)$$

TABLE 1 | CS1: Variation of the random parameters in the hybrid model.

Input variable	Probability distribution	Mean value (μ)	Stand. dev. (σ)	Coeff. of variat. (CV) (%)	Parameter description	Units
M	Lognormal	2	0.4	20	Cart mass	Kg
m	Lognormal	0.15	0.03	20	Pendulum mass	Kg
L	Lognormal	0.7	0.21	30	Rod length	m
M^N	Lognormal	100	20	20	Cantilever beam mass	Kg
C^N	Lognormal	100	30	30	Cantilever beam damping	Ns/m
K^N	Lognormal	10e4	3e4	30	Cantilever beam stiffness	N/m

3.1.3. CS1 Results

In order to test the robustness of the proposed tracking controller, six dominant parameters of the hybrid model were chosen to vary. The first three parameters originate from the control plant and correspond to its M , m , and L , while the remaining three originate from the numerical substructure and correspond to its M^N , C^N , and K^N . These parameters are treated as random with known probability distributions. Their distribution characteristics are described in **Table 1**.

Using the Latin Hypercube Sampling (LHS) methodology, 200 samples were generated from all six parameters and 200 runs of the vRTHS were conducted using combinations of all parameters in each iteration. The tracking controller was kept the same for each one of the 200 runs. The simulation of the 200 vRTHS runs is referred as stochastic vRTHS hereafter. The resulting J_1 , J_2 , and J_3 outputs of the nominal and the stochastic vRTHS are shown in **Table 2** for both CS1 and CS2 for brevity. The nominal values correspond to the parameters used in Equations (22), (23), and (24). The normalized histograms of the J_1 , J_2 , and J_3 out of the 200 vRTHS are shown in **Figure 7**. The aforementioned histograms are a more comprehensive, graphical representation of the values presented in **Table 2**, illustrating the mean values as well as the deviations from them. It is also a metric of robustness; more robust tracking controllers would result in lower deviations in the histograms.

Analysis of the results using uncertainty quantification techniques indicated that 200 runs were sufficient to unveil how the tracking controller performance is affected by parameter variations. Specifically, surrogate models were developed to replicate the response of the CS as the number of runs (in the surrogate training data set) was increasing. With a training data set of 200 samples, validation errors of the surrogate models were <5%. No new runs were added to the data set as this error was deemed to be sufficiently small.

To check if the proposed tracking controller remains stable as the hybrid model parameters vary, vRTHS simulations using the minimum and the maximum values of the random variables were conducted first. No instabilities were observed. Furthermore, none of the conducted 200 simulations was unstable. The same holds for CS2.

The reference, command and measured signals of the hybrid model in the nominal vRTHS are illustrated in **Figure 8**. The reference signal corresponds to the displacement response of the reference model (one with integrated physical and

TABLE 2 | Tracking controller performance and robustness results for CS1 and CS2.

	CS1			CS2		
	Nominal	Stochastic		Nominal	Stochastic	
		Mean values	Stand. dev.		Mean values	Stand. dev.
J_1 [msec]	0.24	0.21	0.21	0	0	0
J_2 [%]	1.88	2.04	0.67	1.89	2.95	0.65
J_3 [%]	1.86	2.01	0.57	2.29	3.04	0.63

CS2 is addressed in the next section 3.2. CS2 results are presented here for brevity and for simpler comparison with results from CS1.

numerical substructures). The command signal corresponds to the displacement response r computed from the numerical substructure at each given time step of vRTHS and is the one that should be followed from the control plant. Finally, the measured signal corresponds to the measured displacement response x of the virtual physical substructure. An ideal tracking controller should be able to compensate the hybrid model in such a way that the command r and measured x to be identical. As it's shown from **Figure 8**, those two signals are, indeed, very close. The comparison with the reference signal is provided in order to validate the fidelity of the hybrid model with respect to the reference structure.

In **Figure 9**, the performance index of the MPC versus time for the nominal case is displayed. This graph illustrates how well MPC managed to minimize the given objective cost function of Equation (1) in every time step of the simulation. A zero value would mean that the cost function was minimized as desired and the "best" optimal control sequence was computed for the given time step. From **Figure 9**, we can observe that the performance index is almost zero during the entire vRTHS, while it is not zero in the time steps in which the highest peaks of the reference signal are attained. This is expected, as the peaks of the command signal are approached, the controller is challenged more and more and has to adapt.

Since the performance of the tracking controller is assessed by how close to zero J_1 , J_2 , and J_3 are, it's clear from **Table 2** and **Figures 7–9** that the proposed tracking controller can provide the desired performance under the presence of any combination of all six random parameters of the hybrid model

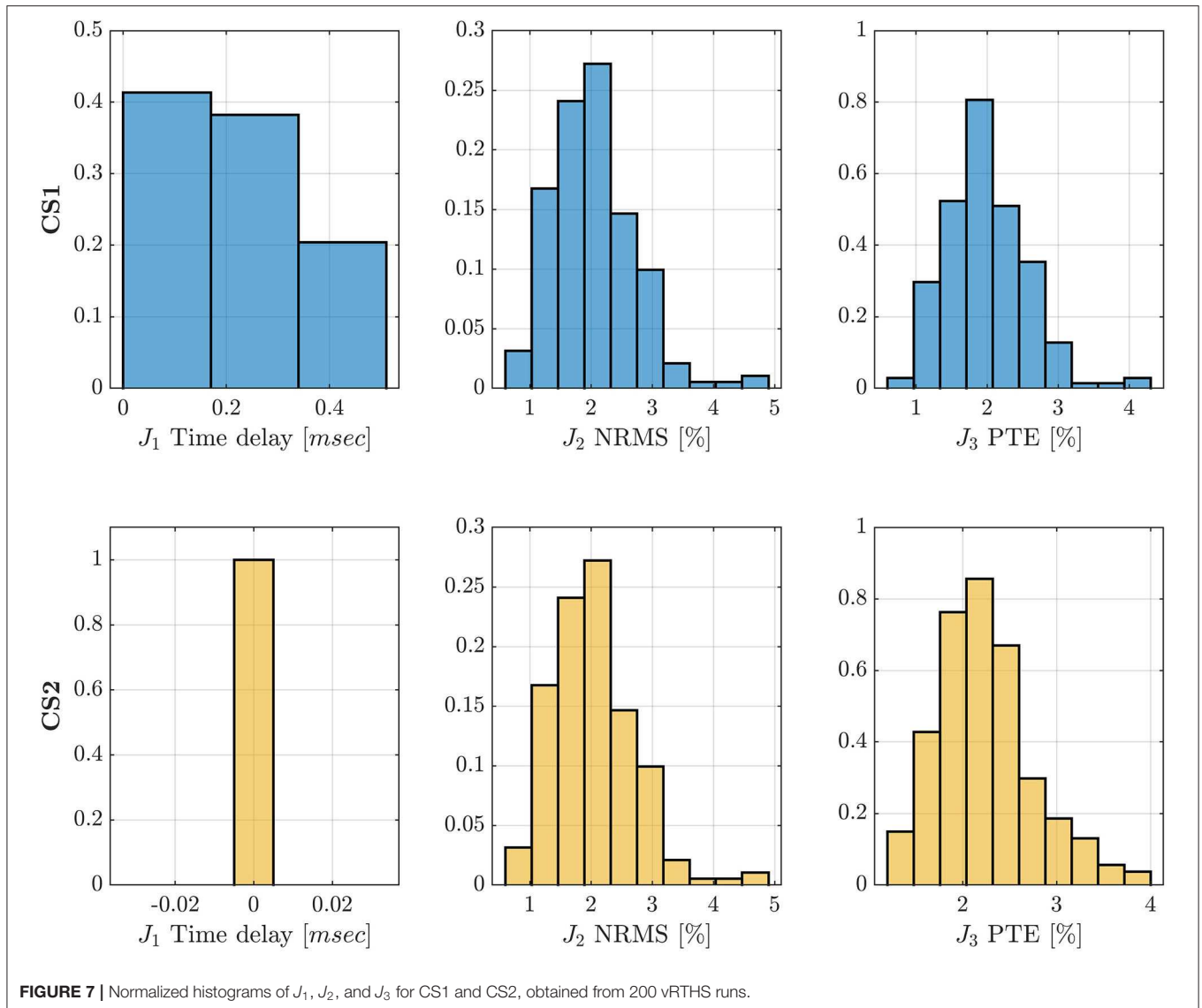


FIGURE 7 | Normalized histograms of J_1 , J_2 , and J_3 for CS1 and CS2, obtained from 200 vRTHS runs.

chosen here, which also demonstrates its robustness. The effects of these stochastic parameters could represent the effect of potential uncertainties (aleatory and/or epistemic) that could be introduced during RTHS. On top of that, it should be pointed out that the controller maintains its performance even in the presence of the additional noise ν and disturbance w that were added in the hybrid model.

3.2. CS2: vRTHS of a Magnetorheological Damper Attached to a 3-Story Structure

3.2.1. Problem Formulation

The reference structure in CS2 corresponds to a 3-story structure equipped with a magnetorheological damper (MRD), installed between the ground and first floor (Dyke et al., 1998) as shown in **Figure 10A**. The numerical substructure corresponds to the 3-story structure (**Figure 10B**), while the virtual physical to the MRD (**Figure 10C**).

The EoM of the reference model reads:

$$\mathbf{M}^N \ddot{\mathbf{x}} + \mathbf{C}^N \dot{\mathbf{x}} + \mathbf{K}^N \mathbf{x} = -\mathbf{M} \Lambda \ddot{x}_g + \mathbf{\Gamma} f^P \quad (35)$$

where $\mathbf{x} = [x_1, x_2, x_3]^T$, $\dot{\mathbf{x}} = [\dot{x}_1, \dot{x}_2, \dot{x}_3]^T$, and $\ddot{\mathbf{x}} = [\ddot{x}_1, \ddot{x}_2, \ddot{x}_3]^T$ correspond to the displacement, velocity, and acceleration relative to the ground, \ddot{x}_g is the ground motion and f^P corresponds to the force generated from the MRD. The $\mathbf{M}^N, \mathbf{C}^N, \mathbf{K}^N$ matrices represent the mass, damping and stiffness of the 3-story structure, respectively, as follows:

$$\mathbf{M}^N = \begin{bmatrix} 1,000 & 0 & 0 \\ 0 & 1,000 & 0 \\ 0 & 0 & 1,000 \end{bmatrix} [\text{Kg}], \quad (36)$$

$$\mathbf{C}^N = 1e4 * \begin{bmatrix} 1.408 & -0.787 & 0.044 \\ -0.787 & 1.494 & -0.635 \\ 0.044 & -0.635 & 0.722 \end{bmatrix} \left[\frac{Ns}{m} \right],$$

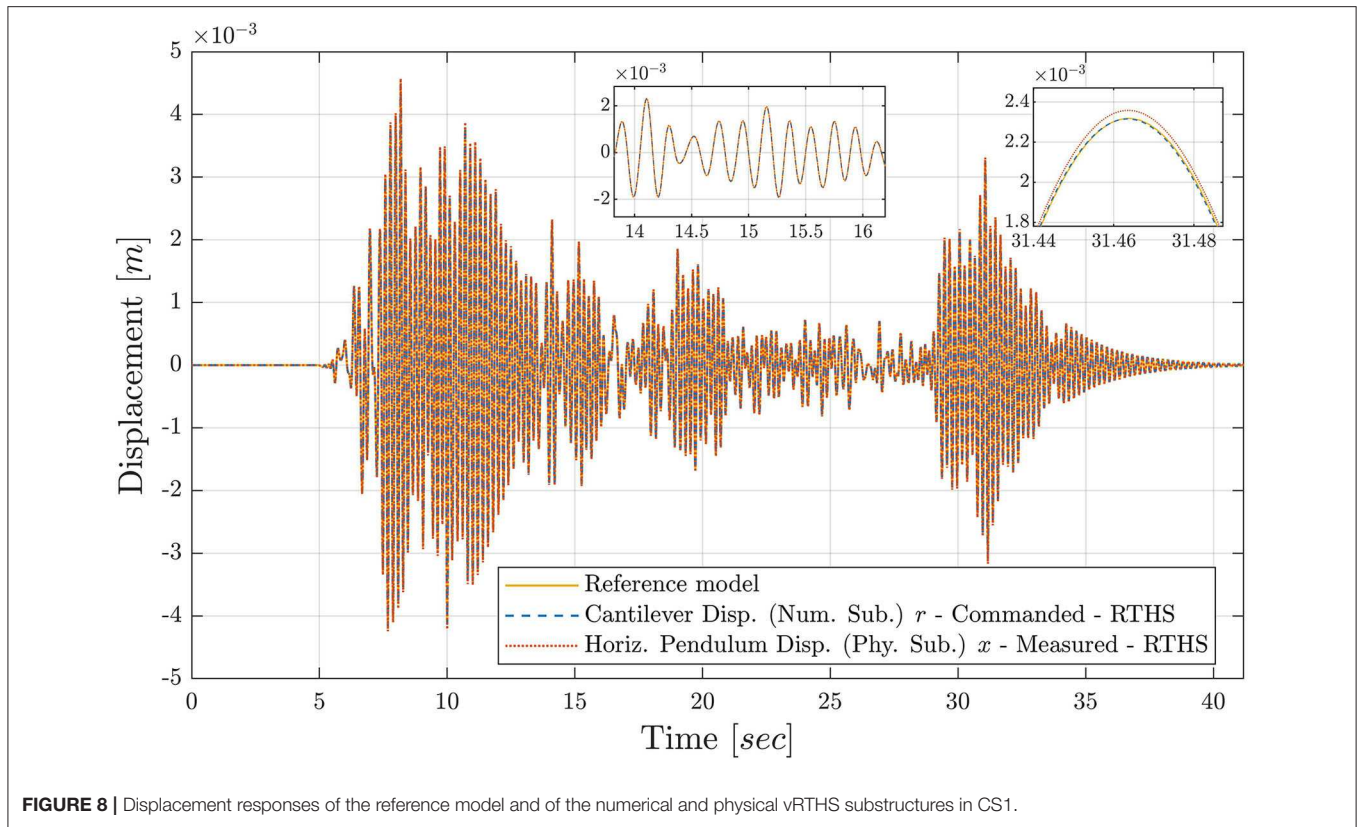


FIGURE 8 | Displacement responses of the reference model and of the numerical and physical vRTHS substructures in CS1.

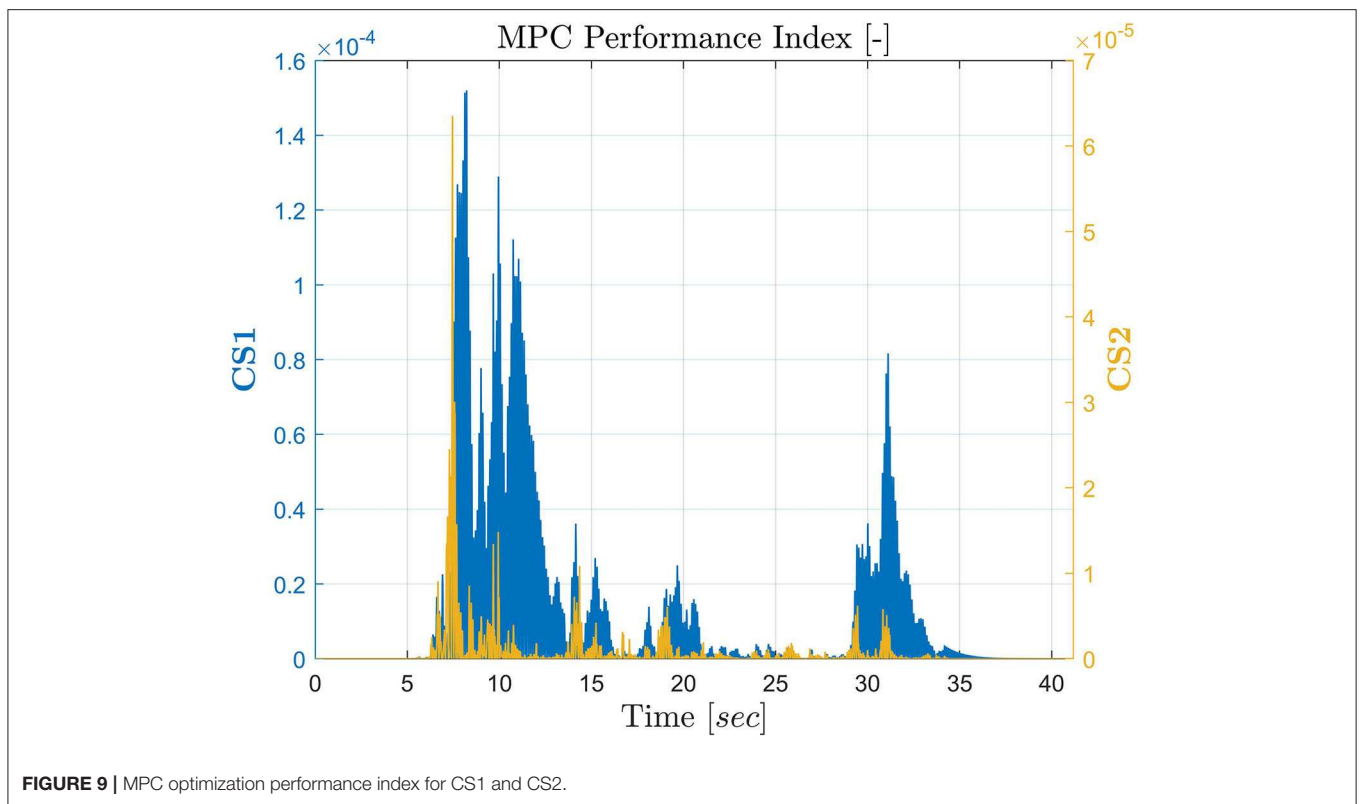
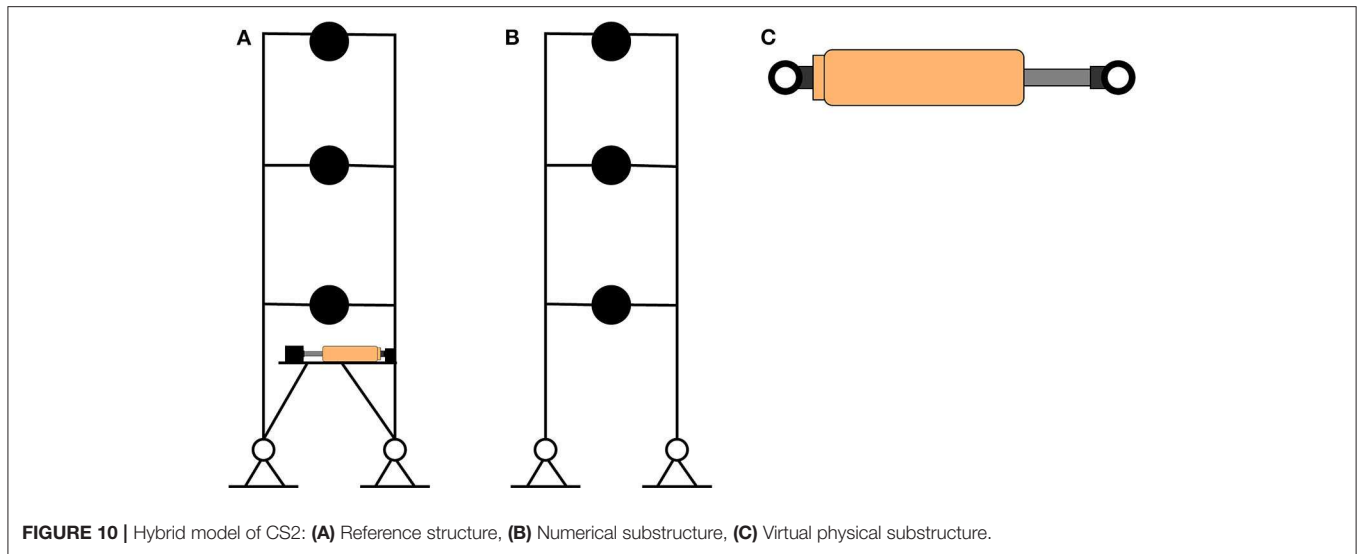


FIGURE 9 | MPC optimization performance index for CS1 and CS2.



$$\mathbf{K}^N = 1e7 * \begin{bmatrix} 2.605 & -2.313 & 0.594 \\ -2.313 & 3.256 & -1.442 \\ 0.594 & -1.442 & 0.927 \end{bmatrix} \begin{bmatrix} N \\ - \\ m \end{bmatrix}$$

Vector $\mathbf{\Lambda} = [1 \ 1 \ 1]^T$ is the ground motion influence vector, while vector $\mathbf{\Gamma} = [1 \ 0 \ 0]^T$ represents the effect of the MRD on the structure. In RTHS, a state-space representation of the Equation (35) is used, which follows:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y} &= \mathbf{Cx} + \mathbf{Du} \end{aligned} \quad (37)$$

where $\mathbf{x} = [x_1, x_2, x_3, \dot{x}_1, \dot{x}_2, \dot{x}_3]^T$, $\mathbf{u} = [\ddot{x}_g, f^P]^T$, $\mathbf{y} = [x_1, x_2, x_3, \dot{x}_1, \dot{x}_2, \dot{x}_3, \ddot{x}_1, \ddot{x}_2, \ddot{x}_3]^T$, and

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{N-1} \mathbf{K}^N & -\mathbf{M}^{N-1} \mathbf{C}^N \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ -\mathbf{\Lambda} & -\mathbf{M}^{N-1} \mathbf{\Gamma} \end{bmatrix} \\ \mathbf{C} &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \\ -\mathbf{M}^{N-1} \mathbf{K}^N & -\mathbf{M}^{N-1} \mathbf{C}^N \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ -\mathbf{\Lambda} & -\mathbf{M}^{N-1} \mathbf{\Gamma} \end{bmatrix} \end{aligned} \quad (38)$$

The block diagram of the hybrid model of CS2 is shown in **Figure 5B**. The reference signal r , in **Figure 5B**, corresponds to the displacement of the first story x_1 . Respectively, $\dot{r} = \dot{x}_1$. RTHS is conducted in displacement control, as in CS1. The ground motion applied to the hybrid model is the same as in CS1, a historical acceleration record from the El Centro 1940 earthquake downscaled by 0.4. As in CS1, apart from the additive disturbance described in Equation (2), additional white noise v is added in the calculated force from MRD f^P , which represents measurement noise from the load cell. The load cell measurement noise is modeled with a standard Gaussian distribution, generated at the same frequency as the sampling of RTHS and amplified by 0.15 N, which approximately equals to 0.01% of the load cell full span. The sampling frequency of RTHS was set to $f_{RTHS} = 4,096$ Hz.

For the numerical integration scheme, the RK4 method is used with a fixed time step of 1/4,096 s.

To model the virtual physical substructure, the MRD in CS2, the Viscous + Dahl model (Ikhouane and Dyke, 2007) was employed. It's dynamics are described as follows:

$$\begin{aligned} f^P(t) &= [k_{x_a} + k_{x_b} V(t)] \dot{x}(t) + [k_{w_a} + k_{w_b} V(t)] W(t) \\ \dot{W}(t) &= \rho (\dot{x}(t) - |\dot{x}| W(t)) \\ W(0) &= \frac{f^P(0) - [k_{x_a} + k_{x_b} V(0)] \dot{x}(0)}{k_{w_a} + k_{w_b} V(0)} \end{aligned} \quad (39)$$

where $\dot{x}(t)$ denotes the MRD piston velocity, $V(t)$ the voltage input command, $f^P(t)$ the damping force, W the damper's nonlinear behavior, k_{x_a} and k_{x_b} the viscous friction coefficient, k_{w_a} and k_{w_b} the dry friction coefficient and t refers to the simulation time. The parameter ρ is calculated as in Tsouroukdissian et al. (2008) and selected to be $\rho = 4,795$ (m^{-1}). The friction parameters are calculated from linear regression as $k_{x_a} = 978$ (Nsm^{-1}), $k_{x_b} = 4,075$ ($Nsm^{-1}V^{-1}$), $k_{w_a} = 160.11$ (N), and $k_{w_b} = 500.78$ (NV^{-1}). The inputs of the MRD model are the displacement $x(t)$ and the voltage $V(t)$, while the output is the force f^P . The latter is the variable that couples the two substructures.

In a MRD, a relatively small electric current applied to the MR valve can change the behavior from very high to very low resistance to motion over a very short time period. In order to ensure optimal response, a bang-bang voltage controller is designed and implemented as illustrated in **Figure 5B**. More specific, when $\text{sgn}(r(t)) = \text{sgn}(\dot{r}(t))$ then the controller provides the MRD with the maximum input voltage, resulting in maximum MRD force f^P . Otherwise, the MRD force is minimum. This bang-bang controller is part of the MRD and it's exclusively responsible for the internal behavior of the MRD.

In CS2, a different approach of the control plant is investigated compared to CS1, since in this case the control plant corresponds

TABLE 3 | CS2: Variation of the random parameters in the hybrid model.

Input variable	Probability distribution	Mean value (μ)	Stand. dev. (σ)	Coeff. of variat. (CV) (%)	Parameter description	Units
L_b	Lognormal	0.762	0.1524	20	Beam length	m
L_c	Lognormal	0.635	0.127	20	Column length	m
M^N	Lognormal	1,000	200	20	Floor mass	kg
Z	Lognormal	0.05	0.01	20	Damping ratio	-
K_{x_a}	Lognormal	978	195.6	20	Viscous friction coef. of MR	Ns/m
K_{w_a}	Lognormal	160.11	32.022	20	Dry friction coef. of MR	N

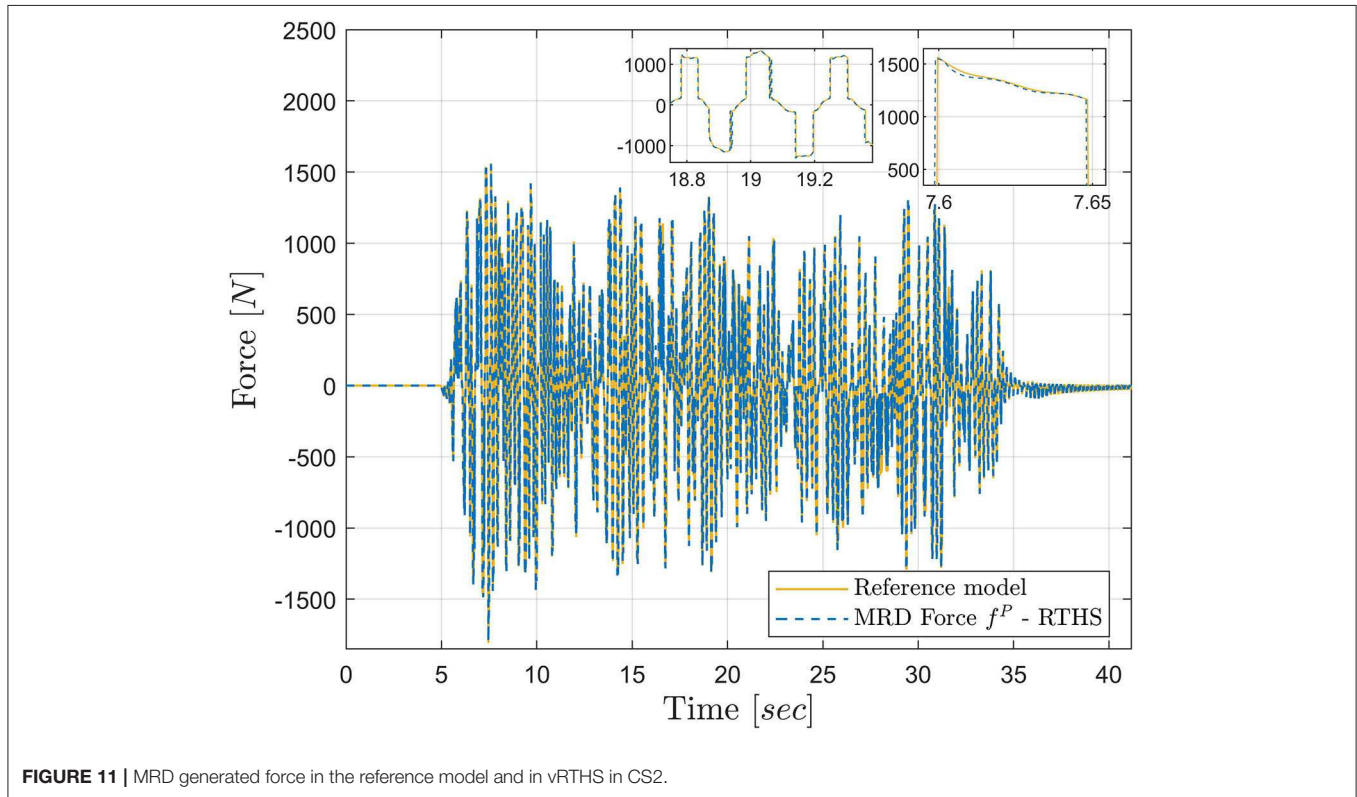


FIGURE 11 | MRD generated force in the reference model and in vRTHS in CS2.

only to the actuator model. In contrast with the control plant represented the actuator model in series with the virtual physical substructure in CS1. Results presented later on prove that the compensation of time delays is sufficient and the performance of RTHS is as desired, when this approach is followed. Moreover, in this way, the dynamics of the control plant are much simpler. Hence, the complexity of the tracking controller is reduced significantly as well. This can be observed by comparing the prediction models used in the two case studies (Equations 30, 40). Therefore, in CS2 the control plant is a SISO model described by Equation (40) with input the desired displacement of the actuator z and output x , the achieved displacement of the actuator. So, in the J_1, J_2 , and J_3 criteria the measured signal $y(i)$ of Equations (19), (20), and (21) corresponds to the actuator achieved displacement x . Furthermore, in order to try the proposed tracking controller under different actuator scenarios, the actuator model used in CS2 corresponds to an

electric actuator, represented by a second-order transfer function, G_{ae} , described by the dynamics of Equation (40).

$$G_{ae} = \frac{3060}{s^2 + 267s + 3060} \quad (40)$$

3.2.2. Tracking Controller Design Properties

As in CS1, the prediction model used in MPC is the control plant, discretized by the sampling frequency of RTHS. The state-space formulation of the discretized model follows (Equation 29) with:

$$A_p = \begin{bmatrix} 0.7693 & -0.0411 \\ 0.055 & 1 \end{bmatrix}, B_p = \begin{bmatrix} 0.0069 \\ 0.0002 \end{bmatrix}, C_p = [0 \ 5.98], D_p = 0 \quad (41)$$

The disturbance model is the same as in Equation (31). The Kalman filter gain matrices in this case are:

$$L = [-0.001 \ 0.723 \ 0.974]^T * 1e(-3) \quad (42)$$

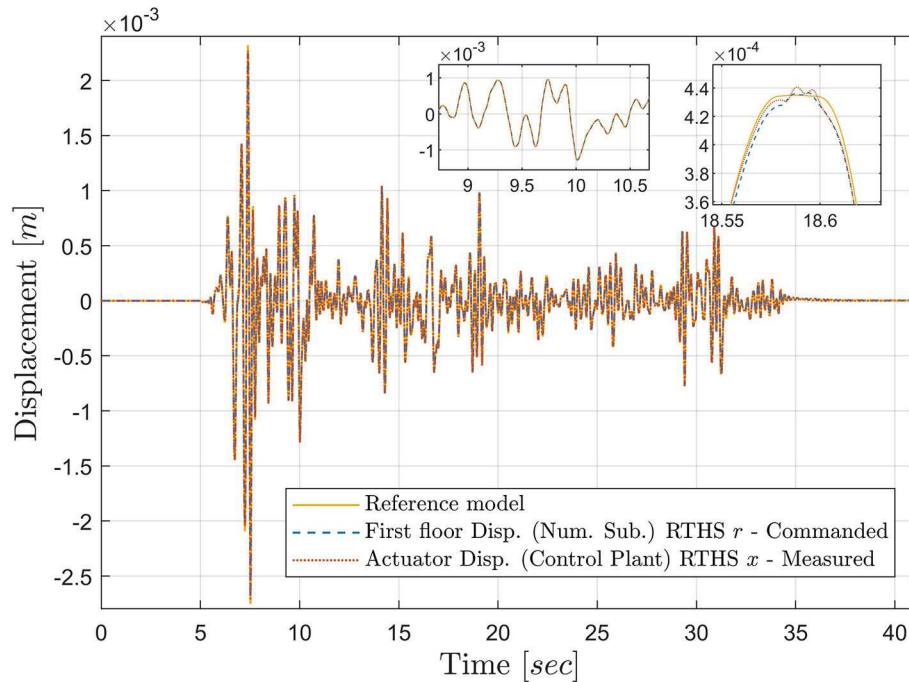


FIGURE 12 | Displacement responses of the reference model, the numerical substructure, and control plant in CS2.

$$M = [0.037 \ 0.721 \ 0.974]^T * 1e(-3)$$

In CS2, the MPC parameters are selected as follows:

- $n^y = 1$ and $n^u = 1$
- $w^y = 64.073$ and $w^u = 0.002$
- $P = 10$
- Each control interval k is obtained on a sampling frequency of 1,024 Hz
- The constraints remain the same with Equation (33).

The polynomial extrapolation coefficients are the same as in Equation (34).

3.2.3. CS2 Results

As in CS1, in order to test the robustness of the tracking controller, six dominant parameters are selected to be random variables with known probability distributions. The first four originate from the numerical substructure:

- Beam length $\rightarrow L_b$
- Column length $\rightarrow L_c$
- Floor mass $\rightarrow M^N$
- Damping ratio $\rightarrow Z$.

The remaining two parameters correspond to the virtual physical substructure and more specific to K_{x_a} and K_{w_a} . These parameters are of particular importance for the MRD model since they are responsible for its nonlinear behavior. All six parameters along with their distribution characteristics are displayed in **Table 3**. As in the previous case study, 200 samples are generated with the LHS method from

the six parameters, and 200 vRTHS runs are conducted accounting for the variability of all parameters in each run. Again the tracking controller was kept the same in all vRTHSs. The nominal case for CS2 are the parameter values from Equations (37) and (39). The arithmetic results for J_1 , J_2 , and J_3 can be found in **Table 2**. Their corresponding normalized histograms for the stochastic vRTHS are illustrated in **Figure 7**.

In **Figure 11**, the force generated by the MRD in the reference model is compared against the one obtained from the vRTHS framework. We can observe that the forces are almost identical. This serves as a demonstration that, although the virtual physical substructure was not included in the control plant, since the latter consists only of the actuator model, its response is compensated sufficiently from time delays and tracking errors. In **Figure 12**, a comparison between three displacement responses is shown; the displacement response of the reference model, the displacement response of the first floor of the numerical substructure r (this is the command signal to the control plant), and the displacement response x , measured from the control plant [this should track r]. The latter two signals prove that the performance of the tracking controller is as desired, as **Figure 12** serves as a graphical illustration of the nominal results shown in **Table 2**. We can observe that due to the proposed controller, x follows the commanded r with minimum delay and tracking error. Finally, in **Figure 9**, the performance index of MPC for CS2 is illustrated.

As in the previous case study, from the **Table 2**, **Figures 7, 12**, it's shown that the controller performance does not get affected by the presence of the introduced random variables, and it provides the requested performance in all considered cases.

4. CONCLUSIONS

In this study, a novel control method to develop a time delay and experimental error compensation strategy in RTHS is presented. The proposed tracking controller aims to conduct RTHS in hard-real time while compensating for potential time delays and tracking errors, under the uncertainties that may arise during RTHS. The tracking controller consists of a robust MPC along with a polynomial extrapolation algorithm and a Kalman filter. The fact that MPC can solve optimization problems online, adapt the new control laws during RTHS using the same model of the system, and simultaneously handle constraints for the system under consideration, indicates that the proposed novel control method is promising for RTHS applications. Polynomial extrapolation was employed to further assist MPC performance, as even small tracking errors can alter the hybrid model's dynamic response. A Kalman filter was used so as to provide MPC with future estimations of the system, in order to derive optimal control laws.

In this paper, the proposed tracking controller formulation was addressed first, followed by two virtual RTHS parametric case studies to assess the performance and robustness of the tracking controller. Dominant parameters of the hybrid model in both case studies were selected and given random perturbations via prescribed probability distributions, varied with at least a 20% coefficient of variation. In each case study, 200 samples were generated from the random parameters and 200 RTHS runs were conducted in order to verify if the proposed tracking controller was robust enough to maintain the desired performance under the introduced uncertainties. Such parameter variations represent potential uncertainties that could be present in real RTHSs. Furthermore, a random disturbance was added in the hybrid model loop along with additional white noise additive to the measured signals. The added disturbance and noise represent systematic or random errors occurring in a real experiment. Since the two case studies were virtual, actuator

models had to be developed in order to simulate actuator dynamics. Two different actuators models were employed in order to assess the tracking controller performance in a wider range of potential experimental equipment. Results from the two case studies illustrate that the proposed tracking controller can guarantee very small time delays and tracking errors under uncertainties that may be introduced in RTHS. Notably, the delays and errors were very close to zero in both case study reference models. Therefore, RTHS using the proposed tracking controller scheme is demonstrated to be effective for structural performance assessment. Ongoing work is focused on implementing the presented case studies in a laboratory and conducting real RTHS using the proposed tracking controller.

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 below: ETH Research Collection, <https://www.research-collection.ethz.ch/handle/20.500.11850/424317>, <https://doi.org/10.3929/ethz-b-000424317>.

AUTHOR CONTRIBUTIONS

NT: conceptualization, methodology, writing. DW and BS: supervision, funding acquisition, and writing. All authors contributed to the article and approved the submitted version.

FUNDING

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 764547. The sole responsibility of this publication lies with the author(s). The European Union is not responsible for any use that may be made of the information contained herein.

REFERENCES

- Bemporad, A., Ricker, N. L., and Morari, M. (2020). *Model Predictive Control Toolbox: Getting Started Guide*. The Math Works.
- Bitsoris, G. (1988). Positively invariant polyhedral sets of discrete-time linear systems. *Int. J. Control* 47, 1713–1726. doi: 10.1080/00207178808906131
- Boyd, S. P., and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge; New York, NY: Cambridge University Press. doi: 10.1017/CBO9780511804441
- Camacho, E. F., and Bordons, C. (2007). "Model predictive control," in *Advanced Textbooks in Control and Signal Processing*, eds M. J. Grimble and M. A. Johnson (London: Springer London), 405. doi: 10.1007/978-0-85729-398-5
- Carrion, J. E., and Spencer, B. F. (2007). *Model-Based Strategies for Real-Time Hybrid Testing*. Technical Report NSEL-006, Newmark Structural Engineering Laboratory. University of Illinois at Urbana-Champaign.
- 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. Eng. Struct. Dyn.* 42, 1697–1715. doi: 10.1002/eqe.2294
- Chen, C., and Ricles, J. M. (2009). Analysis of actuator delay compensation methods for real-time testing. *Eng. Struct.* 31, 2643–2655. doi: 10.1016/j.engstruct.2009.06.012
- Chen, P.-C., Chang, C.-M., Spencer, B. F., and Tsai, K.-C. (2015). Adaptive model-based tracking control for real-time hybrid simulation. *Bull. Earthq. Eng.* 13, 1633–1653. doi: 10.1007/s10518-014-9681-2
- Condori, J., Maghareh, A., Orr, J., Li, H.-W., Montoya, H., Dyke, S., et al. (2020). Exploiting parallel computing to control uncertain nonlinear systems in real-time. *Exp. Techn.* doi: 10.1007/s40799-020-00373-w
- Delbos, F., and Gilbert, J. C. (2003). *Global Linear Convergence of an Augmented Lagrangian Algorithm for Solving Convex Quadratic Optimization Problems*. Technical Report 00071556, INRIA.
- Dyke, S. J., Spencer, B. F., Quast, P., and Sain, M. K. (1995). Role of control-structure interaction in protective system design. *J. Eng. Mech.* 121, 322–338. doi: 10.1061/(ASCE)0733-9399(1995)121:2(322)
- Dyke, S. J., Spencer, B. F., Sain, M. K., and Carlson, J. D. (1998). An experimental study of MR dampers for seismic protection. *Smart Mater. Struct.* 7, 693–703. doi: 10.1088/0964-1726/7/5/012
- Gao, X., Castaneda, N., and Dyke, S. J. (2013). Real time hybrid simulation: from dynamic system, motion control to experimental error. *Earthq. Eng. Struct. Dyn.* 42, 815–832. doi: 10.1002/eqe.2246
- Gao, X. S., and You, S. (2019). Dynamical stability analysis of MDOF real-time hybrid system. *Mech. Syst. Signal Process.* 133:106261. doi: 10.1016/j.ymssp.2019.106261

- Gawthrop, P. J., Wallace, M. I., Neild, S. A., and Wagg, D. J. (2007). Robust real-time substructuring techniques for under-damped systems. *Struct. Control Health Monit.* 14, 591–608. doi: 10.1002/stc.174
- Horiuchi, T. (1996). “Development of a realtime hybrid experimental system with actuator delay compensation,” in *Proc. 11th World Conference on Earthquake Engineering (Acapulco)*.
- Ikhrouane, F., and Dyke, S. J. (2007). Modeling and identification of a shear mode magnetorheological damper. *Smart Mater. Struct.* 16, 605–616. doi: 10.1088/0964-1726/16/3/007
- Jung, R.-Y., Benson Shing, P., Stauffer, E., and Thoen, B. (2007). Performance of a real-time pseudodynamic test system considering nonlinear structural response. *Earthq. Eng. Struct. Dyn.* 36, 1785–1809. doi: 10.1002/eqe.722
- Maghareh, A., Dyke, S. J., and Silva, C. E. (2020). A Self-tuning Robust Control System for nonlinear real-time hybrid simulation. *Earthq. Eng. Struct. Dyn.* 49, 695–715. doi: 10.1002/eqe.3260
- Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O. M. (2000). Constrained model predictive control: stability and optimality. *Automatica* 36, 789–814. doi: 10.1016/S0005-1098(99)00214-9
- Nakashima, M., Kato, H., and Takaoka, E. (1992). Development of real-time pseudo dynamic testing. *Earthq. Eng. Struct. Dyn.* 21, 79–92. doi: 10.1002/eqe.4290210106
- Neild, S. A., Stoten, D. P., Drury, D., and Wagg, D. J. (2005). Control issues relating to real-time substructuring experiments using a shaking table. *Earthq. Eng. Struct. Dyn.* 34, 1171–1192. doi: 10.1002/eqe.473
- Ning, X., Wang, Z., Zhou, H., Wu, B., Ding, Y., and Xu, B. (2019). Robust actuator dynamics compensation method for real-time hybrid simulation. *Mech. Syst. Signal Process.* 131, 49–70. doi: 10.1016/j.ymssp.2019.05.038
- 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. Eng. Struct. Dyn.* 44, 441–460. doi: 10.1002/eqe.2479
- Phillips, B., and Spencer, B. (2013a). Model-based feedforward-feedback actuator control for real-time hybrid simulation. *J. Struct. Eng.* 139, 1205–1214. doi: 10.1061/(ASCE)ST.1943-541X.0000606
- Phillips, B., and Spencer, B. Jr. (2013b). Model-based multiactuator control for real-time hybrid simulation. *J. Eng. Mech.* 139, 219–228. doi: 10.1061/(ASCE)EM.1943-7889.0000493
- Rawlings, J. B., Mayne, D. Q., and Diehl, M. M. (2017). *Model Predictive Control: Theory, Computation, and Design, 2nd Edn.* Madison, WI: Nob Hill Publishing.
- Rossiter, J. A. (2003). *Model-Based Predictive Control: A Practical Approach.* Control series. Boca Raton, FL: CRC Press.
- Schellenberg, A. H., Mahin, S. A., and Fenves, G. L. (2009). *Advanced Implementation of Hybrid Simulation.* Technical Report PEER 2009/104, Pacific Earthquake Engineering Research Center, University of California, Berkeley, CA.
- Schmid, C., and Biegler, L. T. (1994). Quadratic programming methods for reduced hessian SQP. *Comput. Chem. Eng.* 18, 817–832. doi: 10.1016/0098-1354(94)E0001-4
- 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
- Thewalt, C. R., and Mahin, S. A. (1987). *Hybrid Solution Techniques for Generalized Pseudodynamic Testing.* Technical Report UCB/EERC-87/09, Earthquake Engineering Research Center.
- Tøndel, P., Johansen, T. A., and Bemporad, A. (2003). An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica* 39, 489–497. doi: 10.1016/S0005-1098(02)00250-9
- Tsouroukdissian, A. R., Ikhrouane, F., Rodellar, J., and Luo, N. (2008). Modeling and identification of a small-scale magnetorheological damper. *J. Intell. Mater. Syst. Struct.* 20, 825–835. doi: 10.1177/1045389X08098440
- Wagg, D. J., and Stoten, D. P. (2001). Substructuring of dynamical systems via the adaptive minimal control synthesis algorithm. *Earthq. Eng. Struct. Dyn.* 30, 865–877. doi: 10.1002/eqe.44
- Wallace, M., Wagg, D., and Neild, S. (2005). An adaptive polynomial based forward prediction algorithm for multi-actuator real-time dynamic substructuring. *Proc. R. Soc. A* 461, 3807–3826. doi: 10.1098/rspa.2005.1532
- Wallace, M. I., Sieber, J., Neild, S. A., Wagg, D. J., and Krauskopf, B. (2005). Stability analysis of real-time dynamic substructuring using delay differential equation models. *Earthq. Eng. Struct. Dyn.* 34, 1817–1832. doi: 10.1002/eqe.513
- Zafiriou, E. (1990). Robust model predictive control of processes with hard constraints. *Comput. Chem. Eng.* 14, 359–371. doi: 10.1016/0098-1354(90)87012-E
- Zhao, J., French, C., Shield, C., and Posbergh, T. (2003). Considerations for the development of real-time dynamic testing using servo-hydraulic actuation. *Earthq. Eng. Struct. Dyn.* 32, 1773–1794. doi: 10.1002/eqe.301

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.

Copyright © 2020 Tsokanas, Wagg and Stojadinović. 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.