Research Article

Fast Model Predictive Control Method for Large-Scale Structural Dynamic Systems: Computational Aspects

Haijun Peng,¹ Qiang Gao,¹ Zhigang Wu,² and Wanxie Zhong¹

¹ State Key Laboratory of Structural Analysis for Industrial Equipment, Department of Engineering Mechanics, Dalian University of Technology, Dalian 116024, China

² State Key Laboratory of Structural Analysis for Industrial Equipment, School of Aeronautics and Astronautics, Dalian University of Technology, Dalian 116024, China

Correspondence should be addressed to Qiang Gao; qgao@dlut.edu.cn

Received 20 April 2013; Revised 12 January 2014; Accepted 22 January 2014; Published 19 February 2014

Academic Editor: Hongyi Li

Copyright © 2014 Haijun Peng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A fast and accurate model predictive control method is presented for dynamic systems representing large-scale structures. The fast model predictive control formulation is based on highly efficient computations of the state transition matrix, that is, the matrix exponential, using an improved precise integration method. The enhanced efficiency for model predictive control is achieved by exploiting the sparse structure of the matrix exponential at each discrete time step. Accuracy is maintained using the precise integration method. Compared with the general model predictive control method, the reduced central processing unit (CPU) time required by the fast model predictive control scheme can result in a shorter control update interval and a lower online computational burden. Therefore, the proposed method is more efficient for large-scale structural dynamic systems.

1. Introduction

Model predictive control (MPC) is popular because it is one of the few control methodologies that can stabilize both linear and nonlinear systems [1]. In the past 20 years, applications of MPC have grown steadily, and MPC is now in widespread use, particularly in the aerospace field [2–5], the process industry [6–8], and mechanical systems [9–11]. The popularity of the method is the result of the strong theoretical background that has been developed over the past few years, the development of efficient optimization algorithms and codes, and the substantial increase in computational power [1]. However, MPC is used only for small- or medium-scale slow systems [12].

Recent advances in MPC have led to its implementation in faster, large-scale dynamic systems [1, 13, 14]. For large-scale systems, model-based control methods and data-driven control methods are the two most important frameworks in the control community. Data-driven control methods depend only on the measured process variables and are currently receiving increasing attention in the research field and in practical applications [15–17]. A recent study [16] provides

an excellent comparison of the basic data-driven methods for process monitoring and fault diagnosis. The results from this study can provide a reference for achieving successful process monitoring and fault diagnosis of large-scale industrial processes. In contrast to data-driven methods, MPC belongs to the class of model-based control methods. In this study, we are concerned with the computational aspects of large-scale control problems, such as structural dynamic systems.

The objective of MPC is to determine an optimal control sequence online that minimizes the cost of reaching a reference condition within a given prediction horizon given knowledge of the system dynamics and current information [2]. However, for high sampling rates and large-scale systems, the computational time required by MPC becomes a limitation of the method [1]. To overcome the limitations of MPC in a large-scale structural dynamic system, several alternatives that efficiently solve MPC problems have been proposed in the literature. Explicit MPC based on explicit solutions of a quadratic program (QP) is one method for implementing fast MPC [18, 19]. The main drawback of explicit MPC is that the calculation time and the memory requirements increase exponentially with the number of

states, controls, and constraints, so explicit MPC can be used at most for medium-scale problems. Other methods for fast implementations of MPC have involved the design of the optimization algorithm for online solutions such as custom integrated circuit architecture [20], the online active-set qpOASES method [21], the interior-point method, and fast gradient methods. Recently, an algebraic model predictive control (AMPC) algorithm using fewer prediction points was introduced to reduce the number of computations [2]. Based on the direct algebraic solution of the state transition matrix and using the eigenvalues and eigenvectors of the system, AMPC is an approximate method that is efficient and accurate.

The research effort of this study is inspired by the high efficiency of the AMPC algorithm. Two main contributions of the AMPC algorithm are the introduction of variable prediction time intervals and the computation of the matrix exponential for the discretization and construction of the prediction matrices. However, the control update depends on the computation of the state transition matrix. Both the eigenvalue-eigenvector method in AMPC and the Taylor/Padé series expansion method in general MPC for computing the matrix exponential of a large-scale system are time consuming.

In general, the dynamic equations of a structure are obtained using the finite element method, which can have thousands of degrees of freedom; thus, the dynamic equations typically form a large-scale system. Furthermore, the second-order ordinary differential equations for the dynamics can be transformed into first-order ordinary differential equations to form a linear, time-invariant system with no constraints, but the order of the system of equation increases. The numerical efficiency in obtaining solutions becomes a key problem when using the general MPC method and/or the AMPC method for controlling a structural dynamic system, especially for computing the matrix exponential of the system in the AMPC method.

In this study, we analyzed the important factors influencing the computational efficiency in large-scale structural dynamic systems and found that the computations of the feedback gain matrix and the control update have important consequences for the online implementation of unconstrained MPC. Essentially, the feedback gain matrix and the control update are functions of the state transition matrix (or matrix exponential). Therefore, the most critical task for fast online implementation of MPC is highly efficient computation of the state transition matrix. In this study, an efficient numerical method for computing the state transition matrix is proposed that significantly increases the computational efficiency of MPC for large-scale structural dynamic systems.

The contribution of this study lies in the application of the fast precise integration method (FPIM) to compute the matrix exponential arising in linear MPC problems for large-scale structural dynamic systems. Furthermore, we offer a new insight to the physical meaning of the matrix exponential and a fast model predictive control (FMPC) method to increase the efficiency without a loss of accuracy in computing the control law at each control update interval.

By exploiting the sparse structure of the matrix exponential based on the physical meaning of a large-scale structure, it will be shown that the prediction matrices can be computed efficiently. The proposed FMPC algorithm is compared with the general MPC method using a numerical simulation of a large-scale spring-mass system. The results are very encouraging, particularly for large-scale systems and long prediction horizons with many prediction points.

2. Model Predictive Control for Large-Scale Structural Dynamic Systems

Because the use of state-space models has gained increased attention for multivariable processes, a simple derivation of the state-space form for large-scale structural dynamic systems is given first. Then, the general MPC problem based on the state-space form and several relevant results are reviewed briefly in this section.

2.1. State-Space Form for Dynamic Systems. The nominal continuous-time, large-scale structural dynamic model can be defined as

$$\bar{\mathbf{M}}\ddot{\mathbf{q}} + \bar{\mathbf{C}}\dot{\mathbf{q}} + \bar{\mathbf{K}}\mathbf{q} = \bar{\mathbf{B}}\mathbf{u}, \quad (1)$$

where $\mathbf{q} \in \mathbb{R}^{n \times 1}$, $\dot{\mathbf{q}} \in \mathbb{R}^{n \times 1}$, and $\ddot{\mathbf{q}} \in \mathbb{R}^{n \times 1}$ are the displacement, velocity, and acceleration vectors, respectively, n is the total number of degrees of freedom, $\bar{\mathbf{M}} \in \mathbb{R}^{n \times n}$, $\bar{\mathbf{C}} \in \mathbb{R}^{n \times n}$, and $\bar{\mathbf{K}} \in \mathbb{R}^{n \times n}$ are the mass, damping, and stiffness matrices, respectively, $\mathbf{u} \in \mathbb{R}^{r \times 1}$ is the control input vector, r is the number of active control actuators, and $\bar{\mathbf{B}} \in \mathbb{R}^{n \times r}$ is the control input matrix. In general, the mass, damping, and stiffness matrices satisfy the following conditions: $\bar{\mathbf{M}} = \bar{\mathbf{M}}^T > 0$, $\bar{\mathbf{C}} = \bar{\mathbf{C}}^T \geq 0$, and $\bar{\mathbf{K}} = \bar{\mathbf{K}}^T \geq 0$.

Equation (1) is a general system of second-order dynamic equations, and there are many methods available for transforming (1) into a first-order state-space model. In this study, we introduce the new momentum vector as

$$\mathbf{p} = \bar{\mathbf{M}}\dot{\mathbf{q}}. \quad (2)$$

Substituting (2) into (1), the linear, time-invariant, controlled system can be expressed in the state-space form as follows:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad (3)$$

where

$$\mathbf{x} = \begin{bmatrix} \mathbf{q} \\ \mathbf{p} \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} \mathbf{0} & \bar{\mathbf{M}}^{-1} \\ -\bar{\mathbf{K}} & -\bar{\mathbf{C}}\bar{\mathbf{M}}^{-1} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{B}} \end{bmatrix}, \quad (4)$$

$\mathbf{x} \in \mathbb{R}^{2n \times 1}$ is the state vector, and $\mathbf{A} \in \mathbb{R}^{2n \times 2n}$ and $\mathbf{B} \in \mathbb{R}^{2n \times r}$ are the coefficient matrices. Equation (3) is the standard state-space form of a controlled large-scale structural dynamic system.

2.2. General Model Predictive Control Formulation. Combining the output equation of the controlled system with the

state-space model in (3) gives the system representation used in MPC:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad \mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}, \quad (5)$$

where $\mathbf{y} \in \mathbb{R}^{p \times 1}$ is the output vector, p is the number of outputs, and $\mathbf{C} \in \mathbb{R}^{p \times 2n}$ and $\mathbf{D} \in \mathbb{R}^{p \times r}$ are the coefficient matrices.

The future states resulting from control inputs given at the current time can be evaluated by applying a moving prediction horizon. These points in the moving prediction horizon will be referred to as prediction points [2]. General MPC formulations take the prediction horizon T as the length of the moving time horizon and predict future outputs and controls. For a numerical approach, the continuous time domain $\tau \in [t, t + T]$ is divided into N time intervals with equal time steps $\eta = T/N$; that is,

$$\begin{aligned} t_0 &= t, & t_1 &= t + \eta, \dots, \\ t_k &= t + k\eta, \dots, & t_N &= t + N\eta. \end{aligned} \quad (6)$$

Then, the state, control, and output vectors at time t_k are defined as

$$\mathbf{x}_k = \mathbf{x}(t_k); \quad \mathbf{u}_k = \mathbf{u}(t_k); \quad \mathbf{y}_k = \mathbf{y}(t_k). \quad (7)$$

The convolution integral formulation is used to evaluate the future states of a controlled system with a constant control input, which can be expressed as

$$\mathbf{x}_{k+1} = \Phi(t_{k+1}, t_k) \mathbf{x}_k + \int_{t_k}^{t_{k+1}} \Phi(t_{k+1} - \tau) \mathbf{B} \mathbf{u}_k d\tau, \quad (8)$$

where the state transition matrix can be expressed as $\Phi(t_{k+1}, t_k) = \exp(\mathbf{A}\eta)$ or $\Phi(t_{k+1} - \tau) = \exp(\mathbf{A}(t_{k+1} - \tau))$. Equation (8) shows that the predicted value of the state vector at the next time step, $\mathbf{x}_{k+1}$, can be obtained from the current value of the state vector, $\mathbf{x}_k$, and the current value of the control vector, $\mathbf{u}_k$, which is assumed to be constant. Furthermore, because the control vector $\mathbf{u}_k$ is constant in time interval $[t_k, t_{k+1}]$, (8) can be simplified as

$$\mathbf{x}_{k+1} = \Phi(t_{k+1}, t_k) \mathbf{x}_k + \Gamma(t_{k+1}, t_k) \mathbf{u}_k, \quad (9)$$

where

$$\Gamma(t_{k+1}, t_k) = \mathbf{A}^{-1} [\Phi(t_{k+1}, t_k) - \mathbf{I}] \mathbf{B}. \quad (10)$$

The predicted value of the output vector $\mathbf{y}_k$ can be expressed as a function of the current state and future control inputs in the following form:

$$\mathbf{y}_k = \mathbf{C}\Phi(t_k, t_0) \mathbf{x}_0 + \sum_{j=1}^k \mathbf{C}\Phi(t_k, t_j) \Gamma(t_j, t_{j-1}) \mathbf{u}_{j-1} + \mathbf{D}\mathbf{u}_k, \quad (11)$$

where $\mathbf{x}_0$ is the value of the state vector at time t_0 . The future values of the outputs of the controlled system at all of the prediction points in the prediction horizon T can be represented as a function of the current state and a set of control inputs as follows:

$$\begin{aligned} \mathbf{Y} &= \mathbf{F}\mathbf{x}_0 + \mathbf{G}\mathbf{U} \\ &= \begin{bmatrix} \mathbf{C}\Phi(t_1, t_0) \\ \mathbf{C}\Phi(t_2, t_0) \\ \vdots \\ \mathbf{C}\Phi(t_N, t_0) \end{bmatrix} \mathbf{x}_0 + \begin{bmatrix} \mathbf{C}\Gamma(t_1, t_0) & \mathbf{D} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{C}\Phi(t_2, t_1) \Gamma(t_1, t_0) & \mathbf{C}\Gamma(t_2, t_1) & \mathbf{D} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}\Phi(t_N, t_1) \Gamma(t_1, t_0) & \mathbf{C}\Phi(t_N, t_2) \Gamma(t_2, t_1) & \cdots & \mathbf{C}\Gamma(t_N, t_{N-1}) & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_N \end{bmatrix} \end{aligned} \quad (12)$$

in which $\mathbf{Y}$ is a vector of changes in the future outputs evaluated over the prediction horizon with N prediction points and $\mathbf{U}$ is a vector of control inputs over the prediction horizon.

With the predicted outputs in (12) for the given prediction horizon, the cost function for MPC can be defined as

$$J = \frac{1}{2} \mathbf{Y}^T \mathbf{Q} \mathbf{Y} + \frac{1}{2} \mathbf{U}^T \mathbf{R} \mathbf{U}, \quad (13)$$

where $\mathbf{Q} \in \mathbb{R}^{pN \times pN}$ and $\mathbf{R} \in \mathbb{R}^{(N+1)r \times (N+1)r}$ are the weighting matrices. The weighting matrix $\mathbf{Q}$ is a nonnegative definite symmetric matrix, and $\mathbf{R}$ is a positive definite symmetric matrix. The predicted outputs and the control inputs should be minimized over the prediction horizon.

Because there are no constraints, the control inputs can be obtained using the variational method; that is, the control law can be obtained analytically by minimizing the cost function in (13) with respect to the control input $\mathbf{U}$, which gives

$$\mathbf{U} = -(\mathbf{G}^T \mathbf{Q} \mathbf{G} + \mathbf{R})^{-1} \mathbf{G}^T \mathbf{Q} \mathbf{F} \mathbf{x}_0 = -\mathbf{K} \mathbf{x}_0. \quad (14)$$

In the preceding problem, the solution for the optimal control $\mathbf{U}$ in the interval $(t, t + T)$ is obtained with the current state $\mathbf{x}_0$ as the initial condition, but only the value of the control vector for the first step $\mathbf{u}_0$ (the first r rows of $\mathbf{U}$) is employed as the input to the system; the remainder of the control $\mathbf{U}$ is discarded. At the next time instant t , the solution process is repeated; that is, replacing t_0 with the current time t , a continuous-time feedback controller can be obtained from (14). It should be noted that, for a linear, time-invariant

system, the computation of the feedback gain matrix $\mathbf{K}$ must be performed only once, before control commences.

3. Fast Algorithm Based on Improved Precise Integration Method

The review of the general MPC algorithm in the previous section showed that the feedback gain matrix is mainly determined by the matrices $\mathbf{F}$ and $\mathbf{G}$ in (12), which are functions of the matrix exponential. As the number of degrees of freedom n and the number of prediction points N increase, the computation of the matrix exponential becomes increasingly time consuming. In this section, a formulation for computing matrix exponentials is introduced, and then the FPIM is developed to improve the computational efficiency for large-scale structural dynamic systems.

3.1. General Formulation for Matrix Exponentials. In principle, the exponential of a matrix can be computed in many ways. These methods involve series approximations, differential equations, matrix characteristic polynomials, matrix eigenvalues, and matrix decompositions. Moler and van Loan [22] gave a comprehensive review of the computation of matrix exponentials, but none of the methods reviewed is completely satisfactory. In the general MPC method, the value of a matrix exponential for a time-invariant system is typically obtained from a Taylor series expansion truncated to K th-order [2, 22–24]:

$$\Phi(t_{k+1}, t_k) = \exp(\mathbf{A}\eta) \approx \mathbf{I} + \mathbf{A}\eta + \frac{\mathbf{A}^2\eta^2}{2!} + \frac{\mathbf{A}^3\eta^3}{3!} + \cdots + \frac{\mathbf{A}^K\eta^K}{K!}. \quad (15)$$

In the computation of a matrix exponential employing equation (15), if the discretization step η is sufficiently small, the higher-order terms in the expansion are an acceptable approximation. However, if the discretization step η is not very small, the higher-order terms become more significant and affect the accuracy of the matrix exponential. Moler and van Loan [22] presented a better numerical method using a Padé approximation with scaling and squaring.

Once the matrix exponential is obtained, the prediction matrices $\mathbf{F}$ and $\mathbf{G}$ can be constructed. Because the computational procedure given by (15) and the scaling-and-squaring method involve numerous matrix multiplications, they are very time consuming for large-scale systems. Moreover, the length of the prediction horizon and the number of prediction points significantly affect the computation of the matrix exponential. Therefore, neither a Taylor series expansion nor the scaling-and-squaring method is satisfactory for computing the matrix exponential for large-scale dynamic systems.

3.2. Improved Fast Precise Integration Method for Matrix Exponentials. The precise integration method (PIM) was first proposed by Zhong and Williams [25] for the integration of the dynamic equations of structures. Because of the high

precision of the PIM for linear, time-variant systems, this method is now widely used [26–28]. The PIM for computing the matrix exponential in one time step η is as follows:

$$\Phi = \exp(\mathbf{A}\eta) = (\exp(\mathbf{A}\tau))^{\tau/\eta}, \quad (16)$$

where $\tau = \eta/2^M$ and M is an integer ($M = 20$ is recommended in the original PIM). Because the time step η is not very large, τ is extremely small; therefore, the 4th-order Taylor series approximation can be used to evaluate $\exp(\mathbf{A}\tau)$; that is,

$$\exp(\mathbf{A}\tau) \cong \mathbf{I} + \mathbf{T}_a, \quad (17)$$

where

$$\mathbf{T}_a = \mathbf{A}\tau + \frac{\mathbf{A}^2\tau^2}{2!} + \frac{\mathbf{A}^3\tau^3}{3!} + \cdots + \frac{\mathbf{A}^K\tau^K}{K!}. \quad (18)$$

Substituting (17) into (16) gives

$$\Phi = (\mathbf{I} + \mathbf{T}_a)^{2^M} = (\mathbf{I} + \mathbf{T}_a)^{2^{(M-1)}} \times (\mathbf{I} + \mathbf{T}_a)^{2^{(M-1)}}. \quad (19)$$

The incremental part $\mathbf{T}_a$ can be repeatedly computed from

$$\mathbf{T}_a \leftarrow 2\mathbf{T}_a + \mathbf{T}_a \times \mathbf{T}_a. \quad (20)$$

Finally, by computing (20) M times, the matrix exponential Φ is obtained from

$$\Phi = \mathbf{I} + \mathbf{T}_a. \quad (21)$$

From the above process, it can be observed that the original PIM is a simple algorithm. Initial values are chosen for the two integers, M and K , where these values ensure that the relative error of $\exp(\mathbf{A}\eta)$ is less than a predefined tolerance. Next, we let $\mathbf{A}' = \mathbf{A}\eta/2^M$ and compute $\mathbf{T}_a$ according to (18). Equation (20) is then solved M times. The final step is to add the identity matrix to Φ , which yields the required matrix exponential. However, the computational effort of the PIM is $O(n^3)$, which can therefore be very time consuming for large-scale systems. The choice of the parameters M and K in the PIM algorithm is very important. In this study, the parameters M and K are selected with an adaptive method that is based on the following theorem [22].

Theorem 1. For the matrices $\mathbf{A}$ and $\mathbf{E}$, if $\|\mathbf{X}\|/2^M \leq 1/2$, then $[\mathbf{T}_K(\mathbf{X}/2^M)]^{2^M} = \exp(\mathbf{X} + \mathbf{E})$, where

$$\frac{\|\mathbf{E}\|}{\|\mathbf{X}\|} \leq 8 \left(\frac{\|\mathbf{X}\|}{2^M} \right)^K \frac{1}{K+1} \quad (22)$$

in which $\|\mathbf{X}\|$ denotes the norm of the matrix $\mathbf{X}$ and

$$\mathbf{T}_K(\mathbf{A}) = \mathbf{I} + \mathbf{A} + \frac{1}{2!}\mathbf{A}^2 + \cdots + \frac{1}{K!}\mathbf{A}^K. \quad (23)$$

The above theorem can be used to determine the values of M and K in the following way. The relative error e is defined by

$$e = \frac{\left\| [\mathbf{T}_K(\mathbf{X}/2^M)]^{2^M} - \exp(\mathbf{X}) \right\|}{\|\exp(\mathbf{X})\|}. \quad (24)$$

Using the theorem, the following equation holds:

$$e = \frac{\|\exp(\mathbf{X}) \exp(\mathbf{E}) - \exp(\mathbf{X})\|}{\|\exp(\mathbf{X})\|} \leq \|\mathbf{E}\| \exp(\|\mathbf{E}\|) \leq \beta \|\mathbf{X}\| \exp(\beta \|\mathbf{X}\|), \quad (25)$$

where

$$\beta = 8 \left(\frac{\|\mathbf{X}\|}{2^M} \right)^K \frac{1}{K+1}. \quad (26)$$

Therefore, we can choose M and K in such a way to ensure that the relative error e is less than the given tolerance.

Given the appropriate parameters M and K , the original PIM, and the physical significance of the structural dynamic system, the sparse structure of the matrix exponential is analyzed, and then the FMPC is developed.

If there are no active control forces, (9) can be written in partitioned form as

$$\begin{aligned} \mathbf{q}_{k+1} &= \Phi_{11} \mathbf{q}_k + \Phi_{12} \mathbf{p}_k, \\ \mathbf{p}_{k+1} &= \Phi_{21} \mathbf{q}_k + \Phi_{22} \mathbf{p}_k. \end{aligned} \quad (27)$$

Equation (27) shows the physical meaning of the matrix exponential. That is, if (a) the initial displacement of the j th degree of freedom is one, (b) the initial displacements of the remaining degrees of freedoms are zero, and (c) the initial momentums of all of the degrees of freedoms are zero, then the displacement and momentum responses of the i th degree of freedom at $t = \eta$ are $\Phi_{11}(i, j)$ and $\Phi_{21}(i, j)$, respectively.

It is well known that the propagation speed of the energy in a one-dimensional rod is $\sqrt{E/\rho}$ (where E is Young's modulus and ρ is density) and therefore finite. The propagation speed of the energy in a structural dynamic system is similarly finite, and, within a small time step, the excitation of a specific degree of freedom can affect only the adjoining degrees of freedom of the system. Thus, according to the physical meaning of the matrix exponential given above, Φ_{11} and Φ_{12} must be sparse matrices. Similarly, the block matrices Φ_{21} and Φ_{22} are both sparse matrices. Despite the fact that the matrix exponential Φ is a sparse matrix for a small time step, the matrix $\mathbf{T}_a$ may be a dense matrix in the implementation of (18)–(20) because of the accumulation of computer round-off error. If we examine the values of the matrix $\mathbf{T}_a$ carefully, it can be observed that many elements of $\mathbf{T}_a$ are very close to zero. These extremely small elements of the matrix $\mathbf{T}_a$ must therefore be caused by numerical errors and hence should be zero. In the direct application of the PIM, many elements that should be zero are involved in the matrix multiplication, with a resulting waste of computational effort. This waste necessitates a procedure for transforming the dense matrix $\mathbf{T}_a$ to a sparse form, which in turn leads to an FPIM algorithm for computing the matrix exponential efficiently.

The procedure for transforming the dense matrix $\mathbf{T}_a$ to sparse form is as follows. The physical meaning of $\mathbf{T}_a$ enables it to be divided into the four submatrices Φ_{11} , Φ_{12} , Φ_{21} , and Φ_{22} . Calculate α_{11} , the maximum absolute value of all elements comprising Φ_{11} , and set a tolerance, typically

$\varepsilon = 10^{-25}$. Then, if the absolute value of any element in Φ_{11} is less than $\varepsilon \times \alpha_{11}$, set the element to zero. Using this procedure, Φ_{11} is transformed into a sparse matrix that can be stored in a sparse form. The matrices Φ_{12} , Φ_{21} , and Φ_{22} can be transformed in a similar fashion.

In this study, the FPIM is used to compute the matrix exponential in MPC, which results in an FMPC algorithm. The primary difference between the proposed FMPC method and the general MPC method is that the state transition matrix Φ and the matrix Γ are evaluated using the FPIM. As shown in previous sections, the main computational effort with MPC is the computation of the matrix exponential. The proposed FMPC method can significantly reduce the computational burden of obtaining the prediction matrices $\mathbf{F}$ and $\mathbf{G}$.

Based on the preceding analysis, the implementation of FMPC for a large-scale structural dynamic system is as follows.

- (1) Form the mass matrix $\bar{\mathbf{M}}$, the damping matrix $\bar{\mathbf{C}}$, and the stiffness matrix $\bar{\mathbf{K}}$ for the large-scale structural dynamic system, construct the system matrix $\mathbf{A}$ using (4), and store it in a sparse format.
- (2) Choose the prediction time horizon T and the number of prediction points N ; the time step is then given by $\eta = T/N$.
- (3) Let $\bar{\mathbf{A}} = \mathbf{A}\eta/2^M$ and determine the two parameters M and K according to the adaptive error theorem.
- (4) Compute the K th Taylor series approximation formulation

$$\mathbf{T}_a = \bar{\mathbf{A}} + \frac{(\bar{\mathbf{A}})^2}{2!} + \cdots + \frac{(\bar{\mathbf{A}})^K}{K!}. \quad (28)$$

- (5) Transform the matrix $\mathbf{T}_a$ to a sparse format.
- (6) Perform the following steps:

for iter = 1: M

$$\mathbf{T}_a = \mathbf{T}_a (\mathbf{T}_a + 2\mathbf{I}); \quad (29)$$

transform the matrix $\mathbf{T}_a$ to a sparse format;
end

- (7) Add matrix $\mathbf{T}_a$ to the identity matrix to obtain the matrix exponential Φ .
- (8) Compute the matrices $\mathbf{F}$ and $\mathbf{G}$ in (12) in association with (10)–(12).
- (9) The FMPC control law can be obtained from (14).

In this implementation of FMPC, it can be observed that the only difference is the addition of a procedure for transforming the matrix $\mathbf{T}_a$ to a sparse form. However, this simple procedure can significantly improve the computational efficiency, which will be shown by a comparison with general MPC given in Section 4.

FIGURE 1: Oscillating spring-mass model.

4. Numerical Simulations

In this section, the proposed fast MPC algorithm is applied to an oscillating spring-mass model with many degrees of freedom, as shown in Figure 1. This oscillating spring-mass model consists of a sequence of n (where $n = 300, 600$, and 900) masses connected by springs to either the adjacent masses or to the walls on either side. There are 30 actuators distributed evenly from the first mass to the last mass (n). Every mass has value of 1 ($m = 1$), and all springs have a spring constant of 1 ($k = 1$), so the mass matrix $\bar{\mathbf{M}}$ and the stiffness matrix $\bar{\mathbf{K}}$ are both known, and the damping matrix is chosen to be $\bar{\mathbf{C}} = 0.05\bar{\mathbf{K}}$. To capture all possible dynamics of the oscillating spring-mass model and to maintain simulation fidelity, the numerical simulation for the following analysis is executed at a frequency of 10 Hz, which means that the control update interval is 0.1 s. The output variables for MPC were the displacements of all the nodes, the control matrix was $\mathbf{D} = \mathbf{0}$, and the weighting matrices $\mathbf{Q}$ and $\mathbf{R}$ in (14) were both identity matrices.

The proposed algorithm and the general model predictive control algorithm were compared through numerical simulations. The effects of the prediction horizon, the number of prediction points, parametric errors, and nonlinearities in the spring stiffness were examined, and the computational precision and efficiency are discussed in detail.

4.1. Closed-Loop Simulation Results. In the first closed-loop simulation, the number of masses was $n = 300$, the prediction horizon was $T = 30$ seconds, and the number of prediction points was $N = 10$. The initial conditions of the oscillating spring-mass model were a unit displacement and velocity for every mass. The results obtained with the FMPC algorithm are presented in Figures 2–5.

The displacement responses of the first mass with and without FMPC are given in Figure 2, in which the results with FMPC and without FMPC are denoted by solid and dashed lines, respectively. Figure 2 gives the dynamic response with FMPC and the open-loop response. Figure 2 shows that FMPC is able to effectively control the vibrations of the masses, and the displacement is nearly zero after 200 seconds; that is, the oscillating spring-mass model can be stabilized within 200 seconds. The control inputs in Figure 3 support that conclusion.

Figures 4 and 5 illustrate the displacement and velocity responses for all masses at time $t = 1000$ s. The results with FMPC and without FMPC are denoted by triangles and circles, respectively. These results show that FMPC is effective in stabilizing the oscillating spring-mass model.

4.2. Computational Performance. There are two elements that affect the computational performance for a linear, time-invariant system: the formulation of the controller gains

FIGURE 2: Displacement response of the first mass with and without FMPC.

FIGURE 3: Control input of the first actuator with FMPC.

before control begins and the recurrent control evaluation is required at each control update interval [2]. From the formulation of the feedback gain matrix given by (14) and the recurrent control evaluation from (9), the key factor, especially for large-scale structural dynamic systems, is the computation of the matrix exponential. The proposed FMPC algorithm is based on the FPIM, whereas the general MPC algorithm is based on a Padé approximation with scaling and squaring [23] (the method used in the *expm* function in the

TABLE 1: Comparison of CPU times for oscillating spring-mass problem.

N	$n = 300$		$n = 600$		$n = 900$	
	FMPC	MPC	FMPC	MPC	FMPC	MPC
2	0.2327	0.2254	0.4894	6.1205	0.7810	74.9918
5	0.4717	1.4611	1.1203	78.3104	2.0233	233.8051
10	1.6026	3.8909	3.9019	216.8916	7.8787	640.0812

TABLE 2: Maximum response deviations of FMPC for oscillating spring-mass problem.

N	$n = 300$		$n = 600$		$n = 900$	
	$\max(\mathbf{x} - \mathbf{x}^*)$	$\max(\mathbf{u} - \mathbf{u}^*)$	$\max(\mathbf{x} - \mathbf{x}^*)$	$\max(\mathbf{u} - \mathbf{u}^*)$	$\max(\mathbf{x} - \mathbf{x}^*)$	$\max(\mathbf{u} - \mathbf{u}^*)$
2	$1.1554E - 11$	$5.0542E - 12$	$5.4040E - 11$	$2.1243E - 11$	$1.8816E - 10$	$3.7016E - 11$
5	$1.3988E - 11$	$4.7484E - 12$	$5.5522E - 11$	$2.3736E - 11$	$1.9629E - 10$	$5.2537E - 11$
10	$1.3181E - 11$	$6.1258E - 12$	$5.3232E - 11$	$3.0396E - 11$	$1.8866E - 10$	$5.8184E - 11$

FIGURE 4: Displacement responses of all masses at 1000 sec with and without FMPC.

FIGURE 5: Velocity responses of all masses at 1000 sec with and without FMPC.

programming environment MATLAB). To demonstrate the effectiveness of the proposed FMPC algorithm, a comparison of the computation time (CPU time) and the maximum response deviations for FMPC and general MPC is presented. The two methods were both implemented in MATLAB R2010a on a 2.66 GHz personal computer.

The CPU times for various numbers of degrees of freedom and various numbers of prediction points are given in Table 1 in units of seconds. Table 1 illustrates that the CPU times change with the number of prediction points N and the number of degrees of freedom n for both FMPC and general MPC. When $N = 2$ and $n = 300$, the CPU times for FMPC and general MPC are nearly the same. However, when $N = 2$ and $n = 900$, the CPU time for FMPC (0.7810 s) is much less than that for general MPC (74.9918 s); that is, general MPC takes 96.02 times longer to obtain a result similar to that of FMPC. Notably, when $N = 10$ and $n = 900$, the general MPC requires 640.0812 s to complete the computations. Therefore,

the proposed FMPC is computationally superior to general MPC for large-scale structural dynamic systems.

The maximum state response deviations and the maximum control input deviations are given in Table 2, in which $\mathbf{x}$ and $\mathbf{u}$ are the state response and control input given by FMPC and $\mathbf{x}^*$ and $\mathbf{u}^*$ are the state response and control input given by general MPC. Table 2 shows that for varying numbers of prediction points N and degrees of freedom n , the differences between FMPC and general MPC in the responses of the states and the control inputs are all very small, which shows that FMPC is very accurate.

Tables 1 and 2 show that FMPC gives nearly the same computational precision as general MPC using much less CPU time. This demonstrates the efficiency of FMPC in obtaining a satisfactory closed-loop response. Therefore, the recurrent computational work can be significantly reduced during each control evaluation for a large-scale dynamic system. Furthermore, the following important observations

FIGURE 6: Effect of prediction horizon on displacement response: (a) Case 1 ($T = 10$), (b) Case 2 ($T = 20$), and (c) Case 3 ($T = 30$).

can be made: (1) the reduced CPU times required by FMPC offer the opportunity to reduce the control update interval if desired; (2) the same closed-loop performance is achievable with a longer sampling period, so the online computational burden can be reduced further.

4.3. Prediction Horizons. There are many factors that influence the FMPC algorithm regarding system performance such as the weighting matrices $\mathbf{Q}$ and $\mathbf{R}$, the prediction horizon T , and the number of prediction points N . To study the influence of the prediction horizon T , the number of prediction points and the number of degrees of freedom were fixed at $N = 10$ and $n = 300$, respectively, and simulations were conducted for three different values of the prediction horizon: $T = 10, 20$, and 30 . Figures 6 and 7 show the displacement responses and the control inputs, respectively, for the three values of the prediction horizon.

From Figures 6 and 7, it can be observed that the state responses and the control inputs can be stabilized faster when the length of the prediction horizon is reduced, but the maximum values of the control inputs are larger when the prediction horizon is small. Therefore, shorter prediction horizons will generally shorten the transient response at the expense of increased control activity. The results additionally indicate that the transient response of the system can be changed by adjusting the length of the prediction horizon.

Furthermore, as discussed in Section 3, the important difference between the proposed FMPC and general MPC is the computation of the matrix exponential for a large-scale structural dynamic system. In FMPC, the matrix exponential is evaluated using the FPIM. The matrix exponential for a large-scale structural dynamic system has physical

FIGURE 7: Effect of prediction horizon on control inputs: (a) Case 1 ($T = 10$), (b) Case 2 ($T = 20$), and (c) Case 3 ($T = 30$).

significance and therefore has a sparse algebraic structure. Figure 8 gives the distribution of the nonzero elements of the matrix exponential for both FMPC and general MPC for different lengths of the prediction horizon. The upper part of Figure 8 shows that the matrix exponentials with FMPC for the various lengths of the prediction horizon are all sparse matrices with a high degree of sparsity, whereas the bottom part of Figure 8 shows that the matrix exponentials with the general MPC method for the various lengths of the prediction horizon are all full matrices. If we observe carefully these three matrix exponentials obtained from general MPC, it can be observed that many elements are very close to zero. If those nearly zero elements are deleted, the matrix exponentials with general MPC are nearly the same as the matrix exponentials with FMPC. Therefore, the computational efficiency of the proposed FMPC method can be increased significantly without a loss of computational precision.

4.4. Number of Prediction Points. Prediction points are specific points within the prediction horizon that can be used to shape the predicted response of tracked outputs resulting from a hypothetical set of control actions [2]. Therefore, these points can be considered tuning parameters, and the number of prediction points placed over the prediction horizon can influence the closed-loop system response. To study the influence of the number of prediction points, the length of the prediction horizon and the number of degrees of freedom were fixed at $T = 10$ and $n = 300$, respectively, and simulations were performed for three different values of the number of prediction points: $N = 5, 10$, and 15 . Figures 9 and 10

FIGURE 8: Effect of prediction horizon on the structure of the state matrix: FMPC (a) Case 1 ($N = 10$, $nz = 28800$), (b) Case 2 ($N = 20$, $nz = 35100$), and (c) Case 3 ($N = 30$, $nz = 40210$); MPC (a) Case 1 ($N = 10$, $nz = 359880$), (b) Case 2 ($N = 20$, $nz = 360000$), and (c) Case 3 ($N = 30$, $nz = 357655$).

show the displacement responses and the control inputs for the three values. From Figures 9 and 10, it can be observed that the state responses and the control inputs can be stabilized faster with a larger number of prediction points. However, the maximum values of the control inputs increase when the number of prediction points increases. Therefore, more prediction points will generally shorten the transient response at the expense of increasing the maximum values of the control inputs.

Figure 11 gives the distribution of the nonzero elements of the matrix exponential for both FMPC and general MPC. From the upper part of Figure 11, it may be observed that the matrix exponentials with the FMPC method for the various numbers of prediction points are all sparse matrices, whereas, from the bottom part of Figure 11, it may be observed that the matrix exponentials with the general MPC method for the various numbers of prediction points are all full matrices. Therefore, the same conclusion can be drawn as in the previous subsection: the FMPC algorithm has a more sparse structure than the general MPC algorithm. The computational efficiency of the proposed FMPC method can be increased substantially without losing computational precision by exploiting this sparse structure.

4.5. Influence of Parametric Errors and Nonlinear Spring Stiffness. In the previous sections, the algorithm parameters,

FIGURE 9: Effect of the number of prediction points on the displacement response: (a) Case 1 ($N = 5$), (b) Case 2 ($N = 10$), and (c) Case 3 ($N = 15$).

FIGURE 10: Effect of the number of prediction points on the control inputs: (a) Case 1 ($N = 5$), (b) Case 2 ($N = 10$), and (c) Case 3 ($N = 15$).

FIGURE 11: Effect of the number of prediction points on the structure of the state matrix: FMPC (a) Case 1 ($N = 5$, $nz = 35100$), (b) Case 2 ($N = 10$, $nz = 28800$), and (c) Case 3 ($N = 15$, $nz = 25338$); MPC (a) Case 1 ($N = 5$, $nz = 360000$), (b) Case 2 ($N = 10$, $nz = 359880$), and (c) Case 3 ($N = 15$, $nz = 359880$).

such as the prediction horizon T and the number of prediction points N , were investigated for their influence on the controlled system performance. However, errors in the system parameters in actual applications can affect the performance of the FMPC method, so the robustness of the FMPC algorithm must be evaluated. Therefore, this section first examines parametric errors in the mass and stiffness matrices of the linear spring-mass model, and then a nonlinear spring-mass model [29] is simulated with FMPC to validate the

presented methodology. In the numerical simulations in this section, the number of masses was $n = 300$, the prediction horizon length was $T = 10$ seconds, and the number of prediction points was $N = 15$. The initial conditions of the oscillating spring-mass model were a unit displacement for every mass.

To test the effects of mass and stiffness parametric errors, a model predictive controller that was designed with the nominal mass and stiffness parameters was used to close

FIGURE 12: Stochastic mass and stiffness distribution with 10% relative error for the spring-mass model with $n = 300$.

the loop around linear spring-mass models with stochastic mass and stiffness distributions having relative errors of 10% and 50%. The stochastic mass and stiffness distributions are shown in Figure 12. The displacement responses and control inputs for the first mass with FMPC are given in Figure 13, in which the results with the nominal parameters, 10% stochastic error and 50% stochastic error, are denoted by the solid, dotted, and dashed lines, respectively.

For the case of a nonlinear spring-mass model, we assumed that the restoring force f on each mass was $f = -q - \alpha q^3$, where the parameter α is a given coefficient and q is the displacement from the equilibrium position. All of the springs in this example were assumed to be nonlinear with this cubic model of the restoring force. The model predictive controller, which was designed for the nominal linear spring-mass model, was tested with the nonlinear spring-mass model. The displacement responses and control inputs for the first mass with FMPC are given in Figure 14, in which the results with the linear spring model and the cubic nonlinear spring stiffness with $\alpha = 1$ and $\alpha = 10$ are denoted by the solid, dotted, and dashed lines, respectively.

From Figures 13 and 14, we can observe that the displacement responses and the control inputs with 10% stochastic error (nonlinear spring stiffness, $\alpha = 1$) largely coincide with the results for ideal mass and stiffness parameters (linear spring stiffness, $\alpha = 0$). With the increased stochastic error, that is, 50%, and a nonlinear spring stiffness, $\alpha = 10$, the displacement responses and the control inputs obviously deviate from the results for the ideal mass and stiffness parameters (linear spring stiffness, $\alpha = 0$). However, the oscillating spring-mass model can nevertheless be controlled and reach the steady state. Therefore, the FMPC algorithm is robust with respect to the stochastic parameter errors and nonlinearities in the stiffness parameters.

FIGURE 13: Effect of mass and stiffness stochastic errors on the displacement response and the control input for mass 1.

FIGURE 14: Effect of nonlinear spring stiffness on the displacement response and the control input for mass 1.

5. Conclusions

This study introduced a fast and high-precision model predictive control computational method for large-scale structural dynamic systems. The fast model predictive control formulation is based on highly efficient computations of the state transition matrix using an improved precise integration method. Based on the physical significance of the dynamic system, we show that the state transition matrix, that is, the

matrix exponential, with an appropriately chosen time step has a sparse algebraic structure. Using the sparse structure of the matrix exponential in the precise integration method, the efficiency of the proposed method increased significantly. Furthermore, compared with general model predictive control, the proposed method is more efficient for large-scale structural dynamic systems. The most important conclusion is that the computational efficiency of the proposed FMPC algorithm can increase 1-2 orders of magnitude with the same computational precision. Therefore, for large-scale structural dynamic systems, the lower CPU times required by the proposed FMPC algorithm offer the opportunity to shorten the control update interval. Additionally, the same closed-loop performance is achievable with a longer sampling period, so the online computational burden can be reduced further.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Science Foundation of China (11102031, 11272076), the Program Funded by Liaoning Province Education Administration (L2013015), and the Fundamental Research Funds for the Central Universities (DUT13LK25).

References

- [1] P. Patrinos, P. Sopasakis, and H. Sarimveis, "A global piecewise smooth Newton method for fast large-scale model predictive control," *Automatica*, vol. 47, no. 9, pp. 2016–2022, 2011.
- [2] P. W. Gibbens and E. D. B. Medagoda, "Efficient model predictive control algorithm for aircraft," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 6, pp. 1909–1915, 2011.
- [3] E. N. Hartley, P. A. Trodden, A. G. Richards, and J. M. Maciejowski, "Model predictive control system design and implementation for spacecraft rendezvous," *Control Engineering Practice*, vol. 20, pp. 695–713, 2012.
- [4] P. Williams, "Dynamics and control of spinning tethers for rendezvous in elliptic orbits," *Journal of Vibration and Control*, vol. 12, no. 7, pp. 737–771, 2006.
- [5] J. Shi and A. G. Kelkar, "Feedback linearization based generalized predictive control of jupiter icy moons orbiter," *Journal of Dynamic Systems, Measurement and Control*, vol. 131, no. 1, pp. 1–10, 2009.
- [6] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, 2003.
- [7] A. G. Wills, D. Bates, A. J. Fleming, B. Ninness, and S. O. R. Moheimani, "Model predictive control applied to constraint handling in active noise and vibration control," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 1, pp. 3–12, 2008.
- [8] K. Huang, Y. C. Zhang, and F. Yu, "Predictive controller design for electromagnetic suspension based on mixed logical dynamical model," *Journal of Vibration and Control*, vol. 18, no. 8, pp. 1165–1176, 2011.
- [9] L. van den Broeck, M. Diehl, and J. Swevers, "A model predictive control approach for time optimal point-to-point motion control," *Mechatronics*, vol. 21, no. 7, pp. 1203–1212, 2011.
- [10] C. L. Bottasso, C.-S. Chang, A. Croce, D. Leonello, and L. Riviello, "Adaptive planning and tracking of trajectories for the simulation of maneuvers with multibody models," *Computer Methods in Applied Mechanics and Engineering*, vol. 195, no. 50–51, pp. 7052–7072, 2006.
- [11] P. Boscariol, A. Gasparetto, and V. Zanotto, "Active position and vibration control of a flexible links mechanism using model-based predictive control," *Journal of Dynamic Systems, Measurement and Control*, vol. 132, no. 1, pp. 1–4, 2010.
- [12] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 267–278, 2010.
- [13] J. L. Jerez, E. C. Kerrigan, and G. A. Constantinides, "A sparse and condensed QP formulation for predictive control of LTI systems," *Automatica*, vol. 48, no. 5, pp. 999–1002, 2012.
- [14] S. Hovland, J. T. Gravdahl, and K. E. Willcox, "Explicit model predictive control for large-scale systems via model reduction," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 4, pp. 918–926, 2008.
- [15] S. Ding, P. Zhang, E. Ding, P. Engel, and W. Gui, "A survey of the application of basic data-driven and model-based methods in process monitoring and fault diagnosis," in *Proceedings of the 18th World Congress of the International Federation of Automatic Control (IFAC '11)*, Milano, Italy, 2011.
- [16] S. Yin, S. Ding, A. Haghani, H. Hao, and P. Zhang, "A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process," *Journal of Process Control*, vol. 22, no. 9, pp. 1567–1581, 2012.
- [17] S. Yin, S. Ding, A. Haghani, and H. Hao, "Data-driven monitoring for stochastic systems and its application on batch process," *International Journal of Systems Science*, vol. 44, no. 7, pp. 1366–1376, 2013.
- [18] P. Tøndel, T. A. Johansen, and A. Bemporad, "An algorithm for multi-parametric quadratic programming and explicit MPC solutions," *Automatica*, vol. 39, no. 3, pp. 489–497, 2003.
- [19] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.
- [20] A. G. Wills, G. Knagge, and B. Ninness, "Fast linear model predictive control via custom integrated circuit architecture," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 1, pp. 59–71, 2012.
- [21] H. J. Ferreau, H. G. Bock, and M. Diehl, "An online active set strategy to overcome the limitations of explicit MPC," *International Journal of Robust and Nonlinear Control*, vol. 18, no. 8, pp. 816–830, 2008.
- [22] C. Moler and C. van Loan, "Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later," *SIAM Review*, vol. 45, no. 1, pp. 3–49, 2003.
- [23] N. J. Higham, "The scaling and squaring method for the matrix exponential revisited," *SIAM Journal on Matrix Analysis and Applications*, vol. 26, no. 4, pp. 1179–1193, 2005.
- [24] H. Demircioğlu and P. J. Gawthrop, "Continuous-time generalized predictive control (CGPC)," *Automatica*, vol. 27, no. 1, pp. 55–74, 1991.
- [25] W. X. Zhong and F. W. Williams, "A precise time step integration method," *Proceedings of the Institution of Mechanical Engineers C*, vol. 208, no. 6, pp. 427–430, 1994.

- [26] W. X. Zhong, "On precise integration method," *Journal of Computational and Applied Mathematics*, vol. 163, no. 1, pp. 59–78, 2004.
- [27] X. Lu, S. J. Tan, J. H. Lin, D. Kennedy, and F. W. Williams, "Discrete time-delay optimal control of structures under seismic excitations using the balanced reduction scheme," *Journal of Vibration and Control*, vol. 15, no. 3, pp. 465–478, 2009.
- [28] S.-J. Tan, Z.-G. Wu, and W.-X. Zhong, "Discrete-time H_∞ Full-information control of structural systems with control delay," *Journal of Vibration and Control*, vol. 16, no. 3, pp. 455–472, 2010.
- [29] M. Rafikov and J. M. Balthazar, "On control and synchronization in chaotic and hyperchaotic systems via linear feedback control," *Communications in Nonlinear Science and Numerical Simulation*, vol. 13, no. 7, pp. 1246–1255, 2008.