Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Sound and Vibration 294 (2006) 82–96
JOURNAL OF SOUND AND VIBRATION
www.elsevier.com/locate/jsvi

Real-time implementation of adaptive feedback and feedforward generalized predictive control algorithm

Suk-Min Moon, Daniel G. Cole, Robert L. Clark*

Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC 27708, USA

Received 11 May 2004; received in revised form 7 October 2005; accepted 25 October 2005
Available online 4 January 2006

Abstract

The adaptive generalized predictive control (GPC), which combines the process of system identification using recursive least-squares (RLS) algorithm and the process of generalized predictive feedback control design, has been presented and successfully implemented on testbeds. In this paper, the adaptive GPC algorithm is extended when the disturbance measurement signal is available for feedforward control. First, the adaptive feedback and feedforward GPC algorithm is presented when the disturbance is stochastic or random. Second, the adaptive algorithm is further extended when the disturbance is deterministic or periodic. In the second case, measured disturbance signals are used to estimate future disturbance values, which are used in the control design. The proposed adaptive GPC design algorithm with/without the future disturbance estimation is implemented in real time and demonstrated for the application of acoustic noise control, structural vibration control, and optical jitter control.

1. Introduction

Generalized predictive control (GPC) with fixed gains has been of popular use and its application can be found in a wide variety of engineering disciplines. The basic theory and algorithm of GPC is presented along with some interpretations by Clarke and his co-workers [1,2]. The continuous generalized predictive control (CGPC) algorithm implements the discrete-time GPC in a continuous-time setting [3]. The stable continuous generalized predictive control (SCGPC) algorithm guarantees closed-loop stability [4]. The continuous version of GPC algorithm solves problems with discrete-time methods such as numerical sensitivity, sample rate selection, and non-minimum phase zeros [5–7].

Recent developments based on GPC concepts are related to adaptive control. There are many different ways to achieve an adaptive control process. In the early literature on adaptive GPC algorithms, the direct adaptive GPC algorithm was referred to as an adaptive algorithm [8,9]. The direct algorithm estimates the controller parameters directly from measured and known data sets. The indirect algorithm, however, estimates the controller parameters from the process (model) parameters that are estimated from the data set. An indirect algorithm is characterized by two complementary processes: model identification and control

*Corresponding author. Tel.: +919 660 5375; fax: +919 660 5409.
E-mail address: rclark@duke.edu (R.L. Clark).

0022-460X/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jsv.2005.10.017

Nomenclature

$\mathbf{A}_c$	normalized coefficient matrix for past output
$\mathbf{B}_c$	normalized coefficient matrix for past input
$\mathbf{B}_d$	normalized coefficient matrix for past disturbance
$\mathbf{d}_f$	future disturbance vector
$\mathbf{d}_p$	past disturbance vector
$\mathbf{F}$	coefficient matrix for past output
$\mathbf{G}$	coefficient matrix for future output
$\mathbf{H}$	coefficient matrix for past input
h_p	output prediction step
$\tilde{h}_p$	disturbance prediction step
$\mathbf{J}$	coefficient matrix for future input
$\mathbf{M}$	coefficient matrix for past disturbance
m	number of system outputs
$\mathbf{N}$	coefficient matrix for future disturbance
n	number of system inputs
n_d	number of disturbance measurements
p	model/controller order
$\tilde{p}$	disturbance estimation model order
$\mathbf{T}_c$	normalized coefficient matrix for future output
$\mathbf{T}_d$	normalized coefficient matrix for future disturbance
$\mathbf{u}_f$	future input vector
$\mathbf{u}_p$	past input vector
$\mathbf{y}_f$	future output vector
$\mathbf{y}_p$	past output vector
α_i	output parameter, $i = 0, \ldots, p$
β_i	input parameter, $i = 0, \ldots, p$
γ_i	disturbance parameter, $i = 0, \ldots, p$
$\tilde{\gamma}_i$	disturbance estimation model parameter, $i = 1, \ldots, \tilde{p}$
θ	system parameter matrix
θ_d	disturbance estimation model parameter vector
λ	input weighting factor

design. Model identification is the experimental determination of the dynamic behavior of processes. Control design is the design process of controllers based on the identified model and control/performance objectives.

Adaptive control can be viewed as an automation of plant modeling and controller design in which the plant model and controller are updated during each sampling period. Many proposed adaptive GPC algorithms are, however, limited to theoretical development, resulting in difficulty and complexity for real-time application [10–13]. The adaptive GPC, which combines the process of system identification using recursive least-squares (RLS) algorithm and the process of generalized predictive feedback control design, has been presented and successfully implemented on testbeds [14,15]. In this paper, the adaptive algorithm is extended to the case when the disturbance measurement signal is available for feedforward control. Furthermore, the adaptive feedback and feedforward GPC algorithm is extended to the case when disturbance is deterministic or periodic. In such a case, the measured disturbance signals are used to estimate the future disturbance values, which are used in the control design. The proposed feedback and feedforward GPC algorithm is implemented in real time and applied to an optical jitter testbed, a structural system, and an acoustic system.

2. Recursive least-squares (RLS) model estimation

The principle of least squares is that the unknown parameters of a mathematical model should be chosen in such a way that the sum of the squares of the differences between the actually observed and the computed values, multiplied by numbers that measure the degree of precision, is a minimum [16]. It is applied to a mathematical model written in the form

$$y(t) = \theta\varphi^{\mathrm{T}}(t) + e(t), \tag{1}$$

where y is the observed variable, θ is the parameter of the model to be determined, and e is the prediction error. The variable, φ, is called the regression variable [16]. The index, t, denotes discrete time. In an experimental application of the least-squares method, the observed variable, $y(t)$, and the regression variable, $\varphi(t)$, are obtained from an experimental process.

The problem of the system identification is to minimize the prediction error, $e(t)$ [16–20]. The recursive parameter estimation algorithm using the least-squares method is obtained by

$$\theta(t) = \theta(t-1) + \{y(t) - \theta(t-1)\varphi^{\mathrm{T}}(t)\}L(t), \tag{2}$$

where the correcting factor, $L(t)$, is

$$L(t) = \frac{\varphi(t)\mathbf{P}(t-1)}{\rho + \varphi(t)\mathbf{P}(t-1)\varphi^{\mathrm{T}}(t)} \tag{3}$$

and

$$\mathbf{P}(t) = \frac{1}{\rho}\,\mathbf{P}(t-1)[\mathbf{I} - \varphi^{\mathrm{T}}(t)L(t)]. \tag{4}$$

The parameter $0<\rho\leqslant 1$ is the forgetting factor or discounting factor [16,20,21]. In real-time application, it is convenient to start the recursion with zero initial condition for $\theta(0)$ and $\mathbf{P}_0 = \alpha\mathbf{I}$, where $\alpha>0$ and the matrix $\mathbf{I}$ is an identity matrix.

One property of the RLS algorithm is that it allows to omit some data during operation. In the discrete-time application, for instance, if the data sampling rate is too fast to perform all computations in a data sampling period, the recursion can be applied in a slower sampling rate. This is because the recursion does not rely on the relation between two successive regression variables, $\varphi(t)$. By doing this, however, a longer time is needed before a good model is obtained.

2.1. System model estimation

When disturbance measurements are available in a system, a system model can be described as

$$y(t) = \sum_{i=1}^{p} -\alpha_i y(t-i) + \sum_{i=1}^{p} \beta_i u(t-i) + \sum_{i=1}^{p} \gamma_i\, d(t-i), \tag{5}$$

where $y(t)$ is the measured system output, $u(t)$ is the system input, and $d(t)$ is the disturbance measurement [17,18,22]. In order to estimate the model parameters, α_i, β_i, and γ_i, $(i = 1,\ldots,p)$, using the RLS algorithm, Eq. (5) is rewritten as

$$y(t) = \sum_{i=1}^{p} -\alpha_i y(t-i) + \sum_{i=1}^{p} \eta_i v(t-i), \tag{6}$$

where

$$\eta_i = \begin{bmatrix}\beta_i & \gamma_i\end{bmatrix}, \quad v(t) = \begin{bmatrix} u(t) \\ d(t)\end{bmatrix}. \tag{7}$$

The input parameter, β_i, and the disturbance parameter, γ_i, are grouped together to consider the disturbance measurements as an extra input of an estimated model.

Eq. (6) clearly corresponds to Eq. (1) with

$$\theta = [\alpha_1 \;\; \cdots \;\; \alpha_p \;\; \eta_1 \;\; \cdots \;\; \eta_p] \tag{8}$$

and

$$\varphi(t) = \left[-y^{\mathrm{T}}(t-1) \;\; \cdots \;\; -y^{\mathrm{T}}(t-p) \;\; v^{\mathrm{T}}(t-1) \;\; \cdots \;\; v^{\mathrm{T}}(t-p)\right]^{\mathrm{T}}. \tag{9}$$

By rewriting a model given in Eq. (5) into the form given in Eq. (6), the model parameters can be recursively estimated using the RLS algorithm given in Eq. (2).

3. Generalized predictive control

With a set of model parameters given in Eq. (8), a generalized predictive controller can be designed. Predictive control, including GPC, is based upon the following steps: output prediction, control calculation, and feedback implementation [8,10,23,24]. The first two steps are performed open-loop to minimize error between projected output and reference point. Future system outputs are estimated using model parameters and available past data. The control signal is then calculated to enable the predicted output to be as close as possible to the desired future output. The loop is then closed by applying the calculated control signal to the system.

3.1. Output prediction

A model given in Eq. (5) can be written in matrix form:

$$\mathbf{A}\mathbf{y}_p = \mathbf{B}\mathbf{u}_p + \mathbf{D}\mathbf{d}_p, \tag{10}$$

where the output parameter matrix, $\mathbf{A} = [\alpha_0, \alpha_1, \ldots, \alpha_p]$, the input parameter matrix, $\mathbf{B} = [\beta_0, \beta_1, \ldots, \beta_p]$, and disturbance parameter matrix, $\mathbf{D} = [\gamma_0, \gamma_1, \ldots, \gamma_p]$ are coefficient matrices of past output, input, and disturbance data, respectively. Parameters with subscript zero are used for convenience. For m outputs and n inputs system with n_d disturbance measurements, $\alpha_0 = \mathbf{I}_m$, $\beta_0 = \mathbf{0}_{(m,n)}$, and $\gamma_0 = \mathbf{0}_{(m,n_d)}$, where $\mathbf{I}_m$ is an $m \times m$ identity matrix and $\mathbf{0}_{(m,n)}$ is an $m \times n$ zero matrix.

Using Eq. (10), the set of h_p-step output predictions can be written as

$$\mathbf{y}_f(t) = \mathbf{A}_c\mathbf{y}_p(t) + \mathbf{T}_c\mathbf{u}_f(t) + \mathbf{B}_c\mathbf{u}_p(t) + \mathbf{T}_d\mathbf{d}_f(t) + \mathbf{B}_d\mathbf{d}_p(t), \tag{11}$$

where the future output vector, $\mathbf{y}_f(t)$, and the past output history vector, $\mathbf{y}_p(t)$, are written as

$$\mathbf{y}_f(t) = \begin{bmatrix} y(t) \\ \vdots \\ y(t+h_p) \end{bmatrix}, \quad \mathbf{y}_p(t) = \begin{bmatrix} y(t-p) \\ \vdots \\ y(t-1) \end{bmatrix} \tag{12}$$

and the input past history vector, $\mathbf{u}_p(t)$, the future input vector, $\mathbf{u}_f(t)$, the past disturbance history vector, $\mathbf{d}_p(t)$, and the future disturbance vector, $\mathbf{d}_f(t)$, can be obtained similarly. Each normalized coefficient matrix is summarized in Appendix A.

3.2. Control input

The future input is a sequence of control signals. The generalized predictive controller applies only the first one and a new sequence of control signals is calculated when a new measurement is obtained [12,24,25]. The control input at time t is defined by the first n rows of the future input, i.e.,

$$\mathbf{u}(t) = \text{the first } n \text{ rows of } \{-[\mathbf{T}_c^{\mathrm{T}}\mathbf{T}_c + \lambda\mathbf{I}]^{-1}\mathbf{T}_c^{\mathrm{T}}\}(\mathbf{B}_c\mathbf{u}_p(t) + \mathbf{A}_c\mathbf{y}_p(t) - \mathbf{y}_r(t) + \mathbf{A}_d\mathbf{d}_p(t) + \mathbf{T}_d\mathbf{d}_f(t)), \tag{13}$$

where λ is the input weighting factor or control penalty, which is a positive scalar [1,2,24,26]. The last two terms in Eq. (13) are the feedforward terms. The term with $\mathbf{d}_p(t)$ is the past disturbance term and the term with $\mathbf{d}_f(t)$ is the future disturbance term. The past disturbance term can be obtained easily. On the other hand, when disturbance is considered to be random, the future disturbance is hard to estimate. In such a case the future disturbance term is dropped, i.e.,

$$\mathbf{u}(t) = \text{the first } n \text{ rows of } \{-[\mathbf{T}_c^{\mathrm{T}}\mathbf{T}_c + \lambda\mathbf{I}]^{-1}\mathbf{T}_c^{\mathrm{T}}\}(\mathbf{B}_c\mathbf{u}_p(t) + \mathbf{A}_c\mathbf{y}_p(t) + \mathbf{A}_d\mathbf{d}_p(t)). \tag{14}$$

But, when the disturbance is deterministic such as a periodic signal, the future disturbance can be estimated and fills the future disturbance term.

4. Disturbance estimation

When the disturbance is deterministic or periodic, a model that characterizes the measured disturbance signal can be designed. In this research, a model is estimated using the RLS algorithm described in Section 2. A disturbance estimation model is written in the following finite difference equation form:

$$d(t) = \sum_{i=1}^{\tilde{p}} \tilde{\gamma}_i d(t-i), \tag{15}$$

where $\tilde{\gamma}_i$ $(i = 1, \ldots, \tilde{p})$ is the estimated disturbance model parameter and $\tilde{p}$ is the order of the disturbance model. The disturbance model parameter $\tilde{\gamma}_i$ must be distinguished from the system model parameter, γ_i, in Eq. (5).

The parameter vector, θ_d, and the regression vector, $\varphi_d(t)$, are written as

$$\theta_d = \begin{bmatrix} \tilde{\gamma}_1 & \tilde{\gamma}_2 & \cdots & \tilde{\gamma}_{\tilde{p}} \end{bmatrix}, \tag{16}$$

$$\varphi_d(t) = \begin{bmatrix} d^{\mathrm{T}}(t-1) & d^{\mathrm{T}}(t-2) & \cdots & d^{\mathrm{T}}(t-\tilde{p}) \end{bmatrix}^{\mathrm{T}}. \tag{17}$$

It is noted that the order of a disturbance model, $\tilde{p}$, is independent of the order of a system model, p. The subscript d is used for the disturbance model estimation process.

Once the disturbance model parameters given in Eq. (16) are estimated, the future disturbance vector, written as

$$\mathbf{d}_f(t) = \begin{bmatrix} d^{\mathrm{T}}(t) & d^{\mathrm{T}}(t+1) & \cdots & d^{\mathrm{T}}(t+h_p) \end{bmatrix}^{\mathrm{T}}, \tag{18}$$

can be calculated by

$$d(t+j) = \tilde{\gamma}_1 d(t+j-1) + \tilde{\gamma}_2 d(t+j-2) + \cdots + \tilde{\gamma}_{\tilde{p}} d(t+j-\tilde{p}), \tag{19}$$

where the index j goes from 0 to the prediction horizon, h_p.

Eq. (18) assumes that the disturbance prediction step, $\tilde{h}_p$, is equal to the system output prediction step, h_p. The disturbance prediction step, $\tilde{h}_p$, however, can be assigned such that $\tilde{h}_p \leqslant h_p$. In such a case, the normalized coefficient matrix, $\mathbf{T}_d$ in Eq. (13), can be obtained by reducing the size of the coefficient matrix $\mathbf{N}$ in Appendix A to be $\mathbf{N}(1 : m(h_p + p), 1 : n_d(\tilde{h}_p + p))$.

5. Adaptive control

The block diagram shown in Fig. 1 illustrates the proposed adaptive feedback and feedforward GPC algorithm with future disturbance estimation. The overall algorithm consists of two processes: adaptive feedback/feedforward GPC design and future disturbance estimation.

In the process of the adaptive GPC design, the fundamental steps of system identification, GPC design, and stability test are combined into a single process to yield a controller that can adapt; RLS system identification determining system models, GPC design building controllers using the most up-to-date system information, and the stability test of a closed-loop system determining suitable controllers for application.

In addition to the fundamental adaptive controller design steps, a time-varying input weighting factor algorithm is added in order to design aggressive controllers [27]. The input weighting factor is updated based on the stability of a closed-loop system model. When a stable closed-loop model is obtained, the smaller value of input weighting factor is assigned in a new controller design process. Since the input weighting factor is updated after the stability test, application of controllers resulting in unstable closed-loop models must be avoided. If a controller causing an unstable closed-loop model is detected by the stability test, the previous controller resulting in a stable closed-loop model is implemented instead. Moreover, a large initial value of the input weighting factor allows it to avoid a large magnitude of the initial control input, which may cause an overload in the experimental hardware.

In the process of future disturbance estimation, a mathematical representation of disturbance signal is obtained using the RLS algorithm and a set of future disturbances is computed based on the disturbance

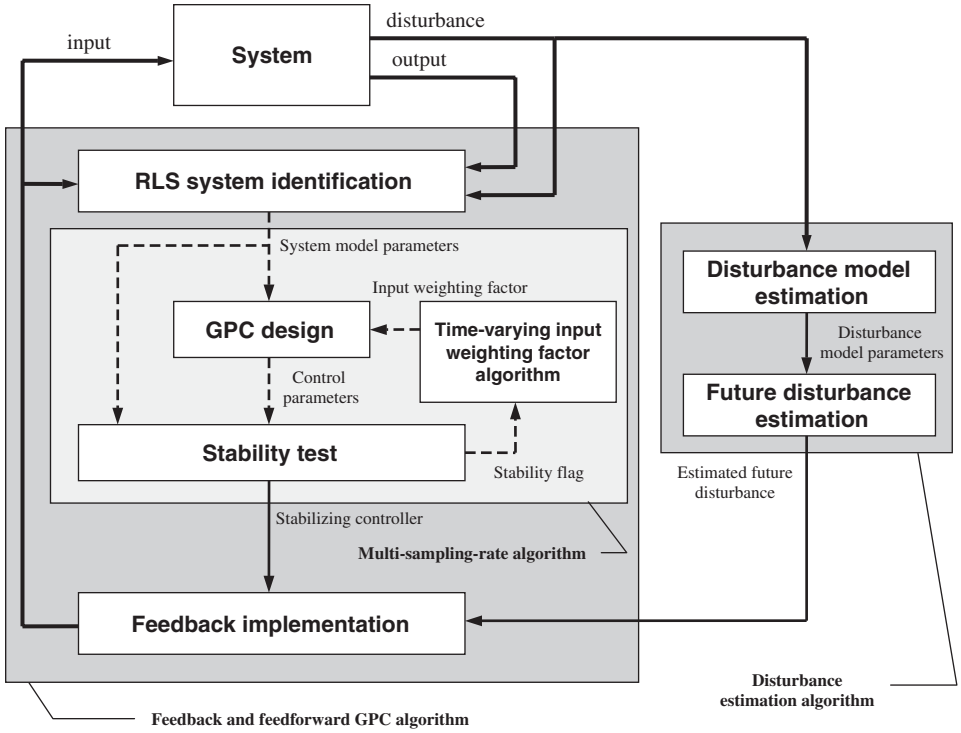

Fig. 1. Adaptive feedback and feedforward GPC design with future disturbance estimation algorithm.

model parameters and measured disturbance data. The future disturbance estimation process is useful when the disturbance is periodic or deterministic.

In order to design higher-order controllers and/or higher-order disturbance models, a multi-sampling rate is applied in the real-time implementation of the overall algorithm. First, each RLS algorithm in the adaptive GPC design process and the disturbance estimation process can be applied in a slower sampling rate than the data sampling rate, t_s, because, as mentioned in Section 2, the recursion in the RLS algorithm does not rely on the relation between two successive data histories. Second, assuming that model parameters do not change significantly, a slower sampling rate than the rate for the RLS algorithm can be applied for the controller design. The controller is fixed until a new controller is computed and updated. The control input and the future disturbance is computed with the measured and stored data in the same sampling rate as the data sampling rate. Graphical implementation for multi-sampling-rate algorithm is illustrated in Fig. 1.

6. Experimental results

The proposed adaptive algorithm is applied to three different testbeds to demonstrate the performance in the application of acoustic noise control [28], structural vibration control [27], and jitter control [14,25].

The discrete-time algorithm is implemented for real-time application using MATLAB/Simulink and the interface between the MATLAB/Simulink model and I/O board (National Instrument PCI-6024E data acquisition board) is performed using xPC Target and Real-time Workshop (RTW). It is an environment in which a desktop computer serving as a host PC generates executable code using RTW and a C/C++ compiler, and xPC Target downloads the executable code to a second PC acting as a target PC, which runs an application model in real time. In the experimental demonstration, a computer with an AMD 1.4 GHz processor and 512 MB memory is used for a target PC.

6.1. Acoustic system: acoustic enclosure

The proposed control algorithm is applied to the acoustic enclosure shown in Fig. 2. The enclosure is configured with two acoustic loudspeakers placed at each end. One of the loudspeakers is used for the control source and the other loudspeaker is used for the primary disturbance. The microphone placed above the control loudspeaker diaphragm is used as the feedback error sensor and the microphone above the disturbance speaker is used as the disturbance measurement sensor. Both microphones are fixed at the center of the square cross-section to minimize the acoustic reflection by the wall. The control objective is to minimize the sound pressure level around the error microphone sensor.

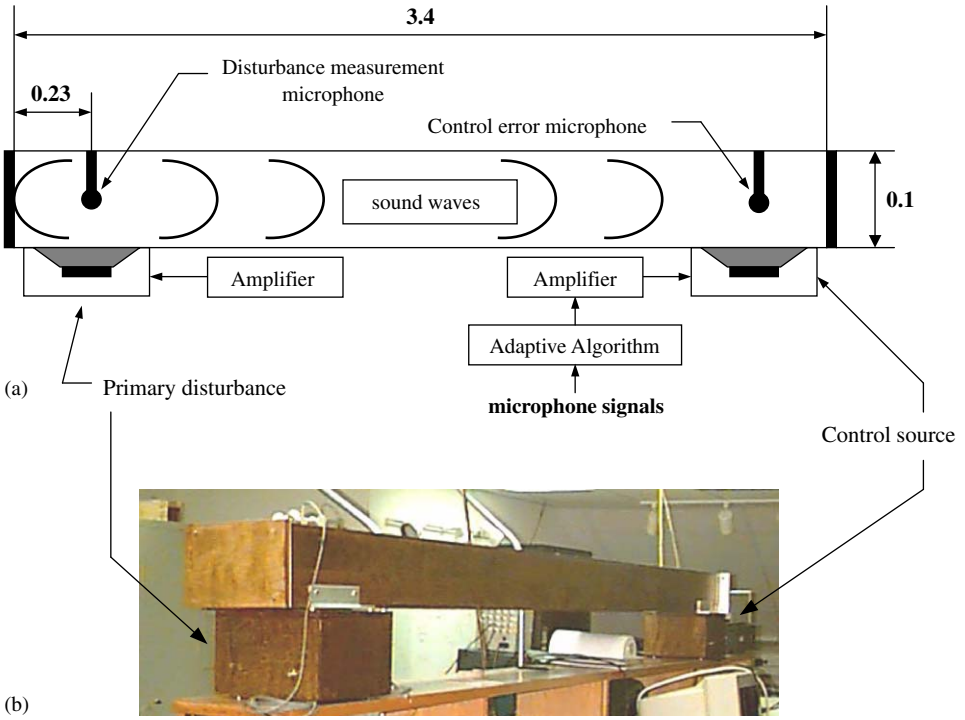

Fig. 2. Acoustic enclosure: (a) schematic diagram; (b) picture of testbed.

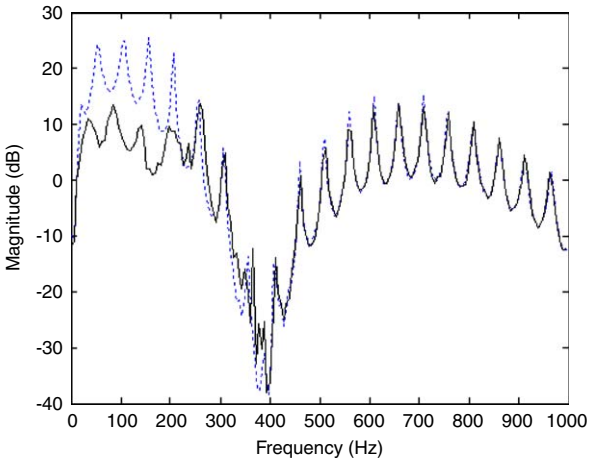

Fig. 3. Frequency response function (FRF) magnitude plot of the error microphone signal for closed–closed acoustic enclosure: open-loop response (dotted line) and closed-loop response (solid line).

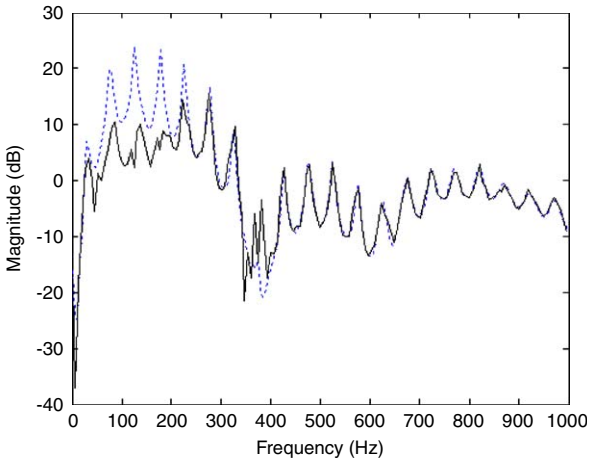

Fig. 4. Frequency response function (FRF) magnitude plot of the error microphone signal for open–closed acoustic enclosure: open-loop response (dotted line) and closed-loop response (solid line).

First, the adaptive control algorithm without the future disturbance estimation algorithm is implemented in real time while the band-limited (500 Hz) random signal is applied to the disturbance loudspeaker. The control algorithm is executed while both endcaps are covered, which yields a closed–closed acoustic enclosure. The disturbance and error sensor signals are measured and stored using a Siglab spectrum analyzer. While the control algorithm for the closed–closed acoustic enclosure is running, the endcap on the control loudspeaker side is opened manually to yield an open–closed acoustic enclosure configuration. The endcap is removed slowly (~2 s) to prevent the overload that may happen to the experimental hardware.

Fig. 3 shows the open- and closed-loop frequency response function (FRF) between the stored disturbance signal and the measured sensor signal for the closed–closed acoustic enclosure configuration. Fig. 4 shows the FRF for the open–closed acoustic enclosure configuration. The integrated response is attenuated by 10.8 dB at frequencies between 50 and 250 Hz for the closed–closed acoustic enclosure configuration and 8.2 dB for the open–closed configuration. No increase is observed in response outside of the bandwidth (>250 Hz, data acquisition at 500 Hz).

Second, the adaptive control algorithm with the future disturbance estimation algorithm is implemented with the following periodic disturbance signal,

$$d(t) = \sin(\Omega_1 t) + \sin(\Omega_2 t) + 0.1w(t), \tag{20}$$

where $\Omega_1 = 30$ Hz, $\Omega_2 = 40$ Hz, and $w(t)$ is uniformly distributed and bounded random signal, $|w(t)| < 1$. System models and disturbance models are estimated at the same sampling rate as data acquisition, 200 Hz. The controller is updated at $\frac{1}{3}$ the speed of the estimation process, i.e., 66.7 Hz. The order of the system model, p in Eq. (6), is chosen to be 16 and the order of the disturbance model, $\tilde{p}$ in Eq. (15), is to be 24. The disturbance prediction horizon, $\tilde{h}_p$ in Eq. (18), is to be 6, while the output prediction horizon, h_p in Eq. (11), is to be 24. The longer prediction horizon and the higher model order improve the performance but they are limited by the target PC in the experiments [27].

Fig. 5 shows the auto-spectrum estimation of the error sensor signal. In order to observe the contribution of the future disturbance estimation algorithm, the closed-loop response, solid line in Fig. 5, is compared with the open-loop response (dotted line) and the closed-loop response without the future disturbance estimation algorithm (dashed line). When the disturbance estimation algorithm is applied with the control algorithm, the better disturbance rejection is achieved at 30 and 40 Hz. The acoustic pressure reduction is summarized in Table 1. The sound pressure level is attenuated by 68% at 30 Hz and by 95% at 40 Hz with the future disturbance estimation algorithm, while 12% reduction at 30 Hz and 52% reduction at 40 Hz is obtained without the future disturbance estimation algorithm.

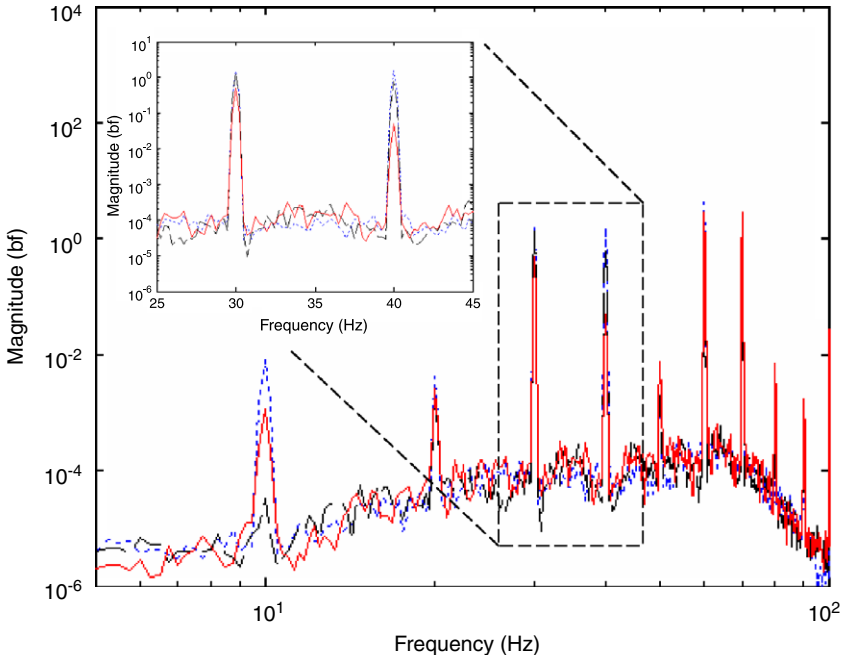

Fig. 5. Auto-spectrum estimation of acoustic error sensor signal: open-loop response (dotted line), closed-loop response without future disturbance estimation (dashed line), and closed-loop response with future disturbance estimation (solid line).

Table 1
Sound pressure reduction on the sound enclosure by the adaptive algorithm

Frequency (Hz)	Open-loop (lbf)	Adaptive algorithm without future disturbance estimation (lbf)	Adaptive algorithm with future disturbance estimation (lbf)
30	0.25	0.22	0.08
40	0.25	0.12	0.008

Fig. 6 shows the FRF between the stored disturbance signal and the measured error sensor signal. Both closed-loop responses with/without future disturbance estimation algorithm achieved similar level reductions at around 10 Hz, which signal is presumed to be the acoustic reflection by the wall.

6.2. Structural vibration control: cantilevered beam

The application of the proposed control algorithm is used for structural vibration control. The cantilevered beam, shown in Fig. 7, is configured with two piezoelectric devices and two accelerometer sensors mounted on the beam. One of the piezoelectric patches is used to disturb the beam and the other patch is used as a control source. One of the accelerometer sensors is used as an error sensor and the other is used for the disturbance measurement.

In order to demonstrate the adaptive control algorithm with future disturbance estimation, the following disturbance signal is applied:

$$d(t) = \sin(\omega_1 t) + 0.1w(t), \tag{21}$$

where ω_1 is the first (measured) natural frequency of the beam, $\omega_1 = 32.5$ Hz, and $w(t)$ is uniformly distributed and bounded random signal, $|w(t)| < 1$. System models and disturbance models are estimated at the same sampling rate of data acquisition, 200 Hz. Controllers are updated at 1/3 the speed of the data acquisition, i.e., 66.7 Hz. The order of system models and disturbance models are chosen to be 16 and 24, respectively.

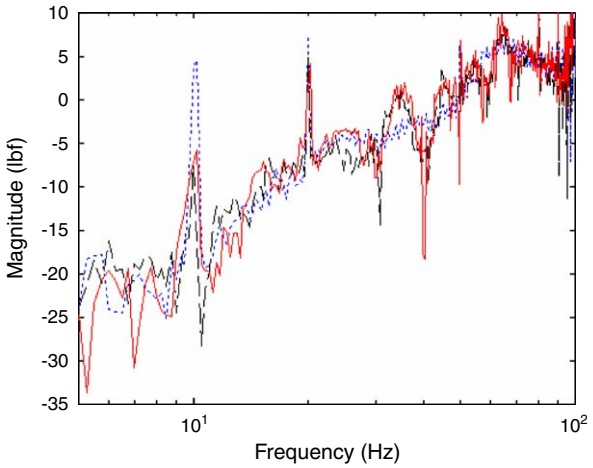

Fig. 6. Frequency response function (FRF) magnitude plot between disturbance signal and error sensor signal: open-loop response (dotted line), closed-loop response without future disturbance estimation (dashed line), and closed-loop response with future disturbance estimation (solid line).

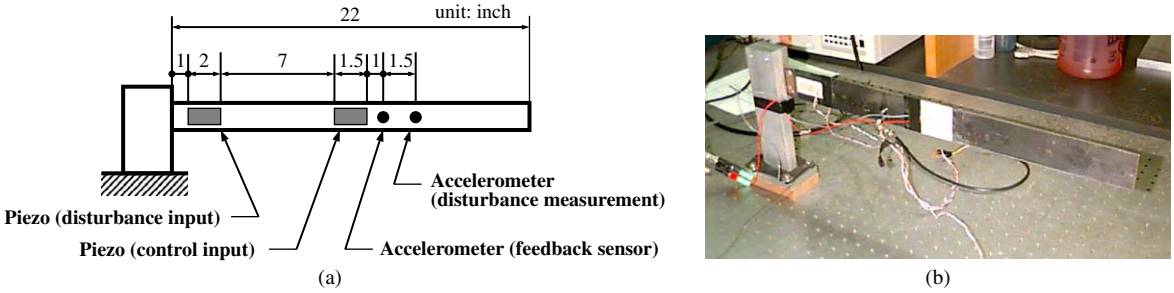

Fig. 7. Cantilevered beam: (a) schematic diagram; (b) picture of testbed.

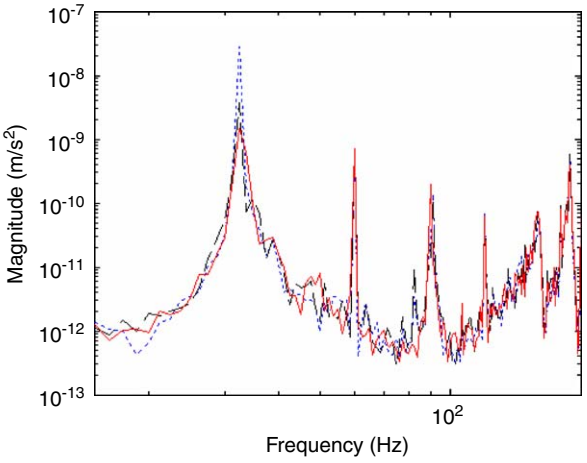

Fig. 8. Auto-spectrum estimation of feedback error sensor signal: open-loop response (dotted line), closed-loop response without future disturbance estimation (dashed line), and closed-loop response with future disturbance estimation (solid line).

Fig. 8 shows the auto-spectrum estimation of the error sensor signals. When the future disturbance estimation algorithm is combined with the adaptive control algorithm, the magnitude at frequency ω_1 is reduced from $27.7 \times 10^{-9}\,\mathrm{m/s^2}$ to $1.4 \times 10^{-9}\,\mathrm{m/s^2}$, while magnitude of $3.7 \times 10^{-9}\,\mathrm{m/s^2}$ at ω_1 is measured without future disturbance estimation algorithm. Although more reduction at ω_1 is achieved with the disturbance estimation algorithm, an increase is observed at higher frequencies, ~90 Hz, because the future disturbance estimation algorithm identifies the disturbance model and estimates the future disturbance based on the periodicity of the disturbance signal.

6.3. Jitter control: optical jitter testbed

The proposed adaptive algorithm is performed on the optical jitter testbed shown in Fig. 9. The control objective is to minimize the acoustically induced jitter using a fast steering mirror (FSM). The test bed is built in an anechoic chamber at Duke University. The purpose of an anechoic chamber is to minimize the jitter effect by extraneous, ambient, acoustic disturbance. Two optical benches are used. The laser source and the position sensing detector (PSD) are mounted on one optical bench and a flat turning mirror and a FSM are mounted on the other optical bench to maximize the beam length. A laser shines a beam on the turning flat mirror, reflects off of the FSM, and shines on the PSD, which provides a measure of the acoustically induced jitter. Jitter is induced acoustically by acoustic loudspeakers, part (F) in Fig. 9, arranged around the flat turning mirror and the FSM. Acoustic microphone sensors are used for acoustic disturbance measurements. Microphones are placed near the flat turning mirror and the FSM.

The data sampling rate in jitter control experiments is set to be 600 Hz because jitter effect occurs below 300 Hz [29]. A low-pass filter, set the cutoff frequency to 250 Hz, and a high-pass filter, set at 10 Hz, is used for the PSD signal. A high-pass filter is used to remove the DC offset in the PSD signal to prevent the static position of the laser. In the application of the adaptive GPC algorithm, model parameters are updated at the same sampling rate as the data sampling rate and controllers are updated at a four times slower sampling rate than the data acquisition, i.e., 150 Hz. The order of a model and controller is set to be 16.

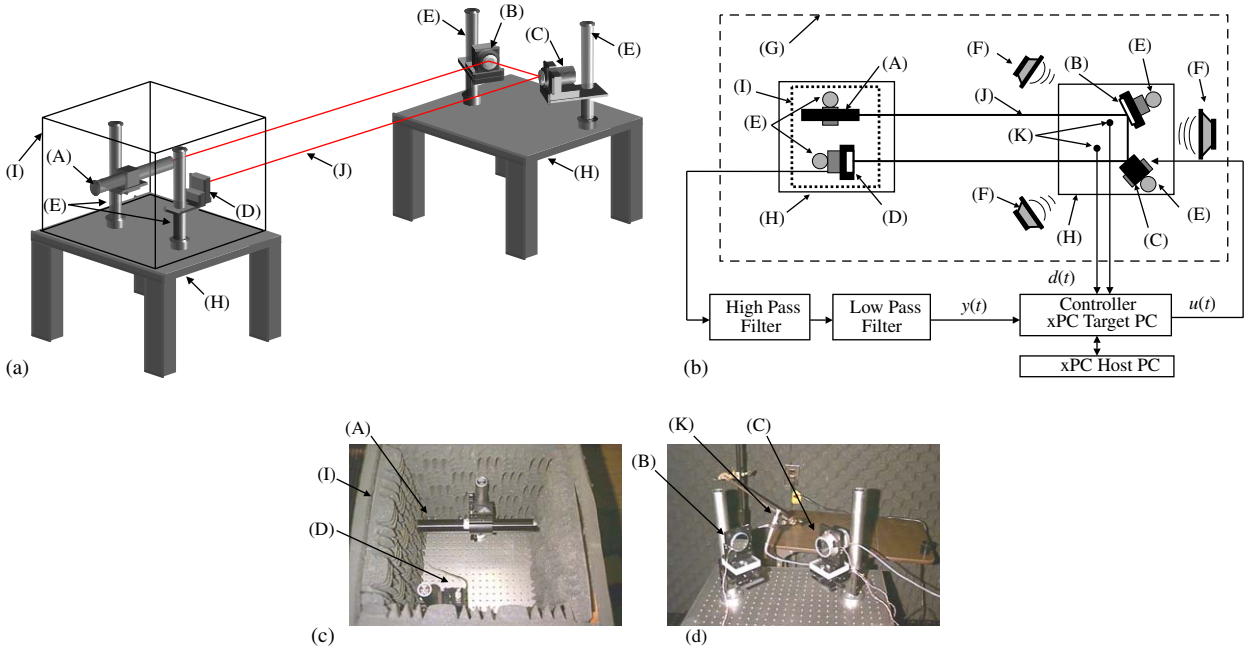

Fig. 9. Experimental system setup of an optical jitter suppression testbed: (a) schematic diagram of jitter testbed; (b) top view of schematic diagram; (c) picture of laser source and PSD; (d) picture of turning mirror and FSM. Experimental parts: (A) HeNe laser, (B) turning flat mirror, (C) fast steering mirror, (D) position sensing detector, (E) mounting rods, (F) disturbance speakers, (G) anechoic chamber, (H) optics benches, (I) enclosure, (J) laser beam (K) microphone.

Experiments are performed with several different environments: (1) a single loudspeaker for disturbance source and a single microphone for disturbance measurement; (2) three loudspeakers and a microphone; (3) a loudspeaker and two microphones; and (4) three speakers and two microphones. For a single disturbance source, a loudspeaker placed behind the optical bench is used (see Fig. 9). When three acoustic loudspeakers are used, each loudspeaker is driven by an band-limited random disturbance, uncorrelated with other speakers. For a single disturbance measurement, a microphone sensor is placed near the FSM. The second microphone sensor is placed near the turning flat mirror.

In the application of the adaptive control algorithm, the future disturbance estimation process is not applied because the random disturbance signal is hard to estimate. Fig. 10 shows the power spectral density of the PSD signal. The dashed lines show the uncontrolled jitter level and the solid lines show the jitter level when the control algorithm is applied. The reduction in the root-mean square (RMS) microradian level for each

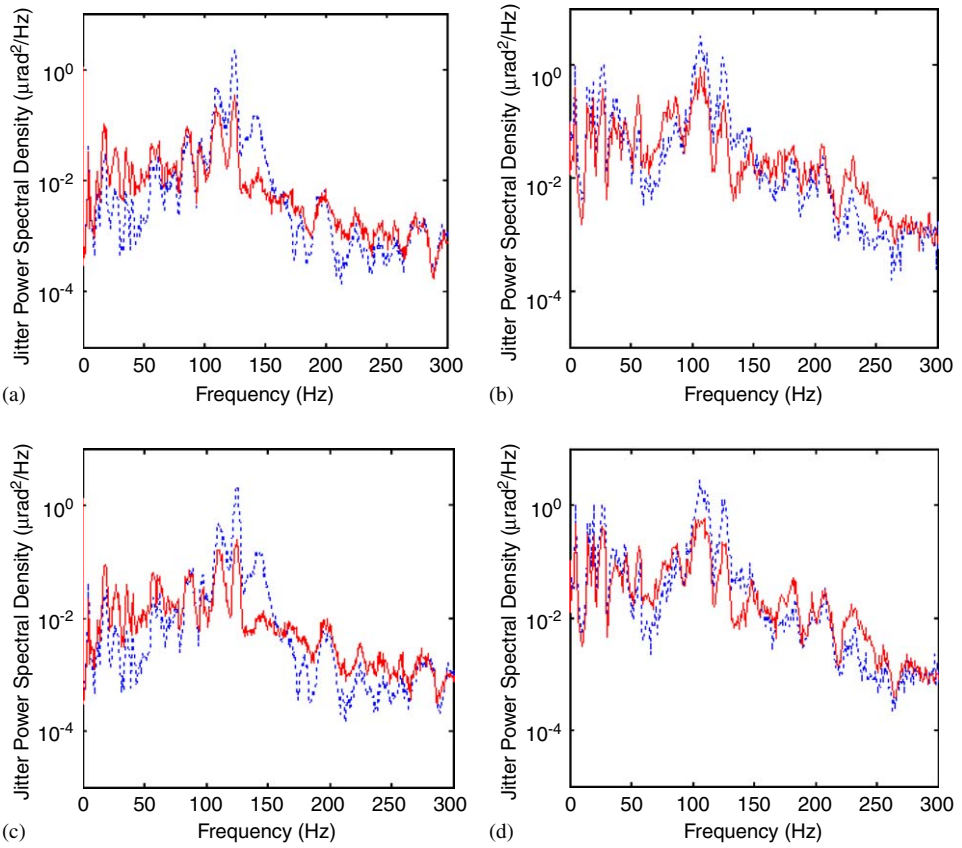

Fig. 10. Auto-spectrum estimation for open-loop (dashed line) and closed-loop (solid line) systems: (a) case 1: a disturbance speaker and a microphone sensor; (b) case 2: three disturbance speakers and a microphone sensor; (c) case 3: a disturbance speaker and two microphone sensors; (d) case 4: three disturbance speakers and two microphone sensors.

Table 2
RMS level reduction on the optical jitter testbed by the adaptive algorithm

Case	No. of speaker	No. of microphone	RMS level reduction (%)
1	1	1	42
2	3	1	43
3	1	2	46
4	3	2	42

configuration is summarized in Table 2. In all cases, the proposed adaptive control algorithm achieved more than 40% reduction from the uncontrolled jitter level.

7. Discussion

The adaptive control algorithm outlined herein extends the RGPC algorithm, which combines the processes of system identification and feedback controller design in a single process [27], for the case when the disturbance measurement is available for the feedforward control. This algorithm is intended for real-time applications and the experimental demonstrations discussed previously demonstrate its application and feasibility.

Before implementation, various control system parameters must be specified; the choice of these parameters depends upon the system to be controlled, its dynamics, and time varying characteristics. One of parameter is the sampling rate. As described earlier, each process of system model identification, disturbance model identification, and controller design can be processed at rate slower than the data acquisition sampling rate. A slower rate, however, can result in a longer convergence time [18,20]. Hence, a sampling rate for the system identification process must be chosen to be faster than temporal variations of system dynamics and disturbances. In experimental trials with the acoustic enclosure in a time-varying configuration (see Section 6.1), unstable responses (in particular, an overload in the experimental hardware) were observed when the endcap of the enclosure was removed too fast, faster than the typical convergence rate of the algorithm. Typically, the RLS algorithm converges in about $2p$ iterations, where p is the number of model parameters [20].

Other parameters to be specified are the number of system and disturbance model parameters, p and $\tilde{p}$, respectively, and the output and disturbance prediction horizons, h_p and $\tilde{h}_p$, respectively. In general, larger models (more parameters) enable more accurate models, and longer prediction horizons improve the control performance; however, this results in greater computational cost at each sampling period [18,30], leading to longer convergence rates and increased difficulty in tracking system changes with time.

As a rule of thumb, the number of system model parameters can be chosen to be 3–4 times of the number of distinct peaks observed in the FRF of the system being investigated. Each peak corresponds to a second-order response, and this choice provides for double the number of second-order poles and effective model prediction in practice [20,31]. In the experiments using the acoustic enclosure, four distinct peaks were observed below 250 Hz (500 Hz sampling rate) (see Figs. 3 and 4); accordingly, the system model size was chosen to be $p = 16$.

Although a longer prediction horizon improves the control performance, it results in increased computational cost. A suitable choice for the prediction horizon is about twice the number of system model parameters [27]. For increasing prediction horizon, this choice results in a smaller effect on the control signal. In our experiments, the output prediction horizon was chosen to be as large as possible provided a computational overload in the DSP system has not occurred.

The disturbance is estimated from the past disturbance measurements and is expressed in a finite impulse response (FIR) filter (see Eq. (15)). In this case, more parameters enable a more accurate model, but this may be limited by the processing capabilities of the DSP system. For the cases of a deterministic or periodic disturbance, we can define the disturbance prediction horizon. For a periodic disturbance, disturbance rejection was achieved with a disturbance prediction horizon shorter than the necessary output prediction horizon.

Given these specifications, the proposed control algorithm designs a controller using the most up-to-date system model and input weighting factor based on model predicted closed-loop stability. This is in contrast to other adaptive predictive control algorithms that use only a current system model and do not adjust the input weight factor to compensate for allowable changes in controller design [4,9,32]. Changes in the input weighting factor can result in better closed-loop performance than would result from changes in other parameters such as the prediction horizon. Furthermore, changes in the input weighting factor minimally affects the computational cost in digital signal processing.

From the viewpoint of adaptive controller design, this algorithm can be applied when there are variations in process dynamics and in the disturbance. It constructs model estimates from real-time data and the controller is updated in the presence of a changing operating environment.

Appendix A. Normalized coefficient matrix

From Eq. (10), the following matrix equation is obtained by constructing the h_p-step predictor at time t and partitioning into past and future parts:

$$[\mathbf{F} \mid \mathbf{G}]\begin{bmatrix}\mathbf{y}_p(t)\\ \mathbf{y}_f(t)\end{bmatrix} = [\mathbf{H} \mid \mathbf{J}]\begin{bmatrix}\mathbf{u}_p(t)\\ \mathbf{u}_f(t)\end{bmatrix} + [\mathbf{M} \mid \mathbf{N}]\begin{bmatrix}\mathbf{d}_p(t)\\ \mathbf{d}_f(t)\end{bmatrix}, \tag{A.1}$$

where

$$[\mathbf{F} \mid \mathbf{G}] = \left[\begin{array}{ccc|cccccc} \alpha_p & \cdots & \alpha_1 & \alpha_0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ \vdots & 0 & \alpha_p & \vdots & \alpha_1 & \alpha_0 & 0 & \cdots & 0 \\ \vdots & \vdots & 0 & \alpha_p & \vdots & \alpha_1 & \alpha_0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \alpha_p & \cdots & \alpha_1 & \alpha_0 \end{array}\right], \tag{A.2}$$

$$[\mathbf{H} \mid \mathbf{J}] = \left[\begin{array}{ccc|cccccc} \beta_p & \cdots & \beta_1 & \beta_0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ \vdots & 0 & \beta_p & \vdots & \beta_1 & \beta_0 & 0 & \cdots & 0 \\ \vdots & \vdots & 0 & \beta_p & \vdots & \beta_1 & \beta_0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \beta_p & \cdots & \beta_1 & \beta_0 \end{array}\right], \tag{A.3}$$

$$[\mathbf{M} \mid \mathbf{N}] = \left[\begin{array}{ccc|cccccc} \gamma_p & \cdots & \gamma_1 & \gamma_0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots \\ \vdots & 0 & \gamma_p & \vdots & \gamma_1 & \gamma_0 & 0 & \cdots & 0 \\ \vdots & \vdots & 0 & \gamma_p & \vdots & \gamma_1 & \gamma_0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \gamma_p & \cdots & \gamma_1 & \gamma_0 \end{array}\right]. \tag{A.4}$$

Eq. (11) is obtained by solving Eq. (A.1) for the future outputs and normalizing the coefficients by multiplying the inverse of the coefficient matrix of the future output. The normalized coefficient matrices are summarized in Table 3.

Table 3
The normalized coefficient matrix in Eq. (11)

Matrix	Size
$\mathbf{T}_c = \mathbf{G}^{-1}\mathbf{J}$	$m(h_p+p) \times n(h_p+p)$
$\mathbf{B}_c = \mathbf{G}^{-1}\mathbf{H}$	$m(h_p+p) \times np$
$\mathbf{A}_c = -\mathbf{G}^{-1}\mathbf{F}$	$m(h_p+p) \times mp$
$\mathbf{T}_d = \mathbf{G}^{-1}\mathbf{N}$	$m(h_p+p) \times n_d(h_p+p)$
$\mathbf{B}_d = \mathbf{G}^{-1}\mathbf{M}$	$m(h_p+p) \times n_d p$

References

[1] D.W. Clarke, C. Mohtad, P.S. Tuffs, Generalized predictive control—part I, The basic algorithm, *Automatica* 23 (2) (1987) 137–148.
[2] D.W. Clarke, C. Mohtad, P.S. Tuffs, Generalized predictive control—part II, Extensions and interpretations, *Automatica* 23 (2) (1987) 149–160.
[3] H. Demircloglu, P.J. Gawthrop, Continuous-time generalized predictive control (CGPC), *Automatica* 27 (1) (1991) 55–74.
[4] J.R. Gossner, B. Kouvaritakis, J.A. Rossiter, Cautious stable predictive control: a guaranteed stable predictive control algorithm with low input activity and good robustness, *International Journal of Control* 67 (5) (1997) 675–697.
[5] K.J. Åström, P. Hagander, J. Sternby, Zeros of sampled systems, *Optimal Control Applications and Methods* 3 (4) (1982) 399–414.
[6] D.W. Clarke, Self-tuning control of nonminimum-phase systems, *Automatica* 20 (5) (1984) 501–518.
[7] N.K. Sinha, S. Puthenpura, Choice of the sampling interval for the identification of continuous-time systems from samples of input/output data, *Proceedings of the Institution of Electrical Engineers* 132(6) (1985) 263–267.
[8] J.-N. Juang, M. Phan, Deadbeat predictive controllers, Technical Report TM-112862, NASA, May 1997.
[9] W. Wang, A direct adaptive generalized predictive control algorithm for MIMO systems, *International Journal of Control* 60 (6) (1994) 1371–1381.
[10] H. Demircloglu, E. Karasu, Generalized predictive control: a practical application and comparison of discrete- and continuous-time versions, *IEEE Control Systems Magazine* 20 (5) (2000) 36–47.
[11] W. Wang, R. Henriksen, Direct adaptive generalized predictive control, in: *Proceedings of the American Control Conference*, Vol. 3, June 1992, pp. 2402–2406.
[12] D. Soloway, P.J. Haley, Neural generalized predictive control: a Newton–Raphson implementation. Technical Report TM-110244, NASA, February 1997.
[13] R.M.C. De Keyser, A.R. Van Cauwenberghe, Extended prediction self-adaptive control, in: *Proceedings of the IFAC Identification and System Parameter Estimation*, Vol. 122, 1975, pp. 929–934.
[14] S.-M. Moon, R.L. Clark, Advanced methods for optical jitter suppression using acoustic actuators, in: *Proceedings of the SPIE: Smart Structures and Materials, Damping and Isolation*, Vol. 4331, 2001, pp. 72–81.
[15] S.-M. Moon, R.L. Clark, D.G. Cole, The theory and experiments of recursive generalized predictive control, *IMECE'2002*, *2002 ASME International Mechanical Engineering Congress and Exposition*, November 2002, New Orleans, Lovisiana United States of America, IMECE2002-33351.
[16] K.J. Åström, B. Wittenmark, *Adaptive Control*, second ed., Addison-Wesley Publishing Company, Inc., Reading, MA, 1995.
[17] J.-N. Juang, *Applied System Identification*, Prentice-Hall, Englewood Cliffs, NJ, 1994.
[18] L. Ljung, *System Identification: Theory for the User*, second ed., Prentice-Hall, Englewood Cliffs, NJ, 1999.
[19] S.M. Kuo, D.R. Morgan, *Active Noise Control Systems: Algorithms and DSP Implementation*, Wiley, New York, 1996.
[20] S. Haykin, *Adaptive Filter Theory*, second ed., Prentice-Hall, Englewood Cliffs, NJ, 1991.
[21] G.V. Moustakides, Study of the transient phase of the forgetting factor RLS, *IEEE Transaction on Signal Processing* 45 (10) (1997) 2468–2476.
[22] A.V. Oppenheim, R.W. Schafer, J.R. Buck, *Discrete-Time Signal Processing*, second ed., Prentice-Hall, Englewood Cliffs, NJ, 1998.
[23] M.Q. Phan, J.-N. Juang, Predictive controllers for feedback stabilization, *Journal of Guidance, Control, and Dynamics* 21 (5) (1998) 747–753.
[24] J.-N. Junag, K.W. Eure, Predictive feedback and feedforward control for systems with unknown disturbance. Technical Report TM-208744, NASA, December 1998.
[25] S.-M. Moon, R.L. Clark, D.G. Cole, Recursive methods for optical jitter suppression using acoustic actuators, in: *Proceedings of the SPIE: Smart Structures and Materials, Damping and Isolation*, Vol. 4697, 2002, pp. 193–204.
[26] D.W. Clarke, C. Mohtadi, Properties of generalized predictive control, *Automatica* 25 (6) (1989) 859–875.
[27] S.-M. Moon, R.L. Clark, D.G. Cole, The recursive generalized predictive feedback control: theory and experiments, *Journal of Sound and Vibration* 279 (2005) 171–199.
[28] S.-M. Moon, R.L. Clark, D.G. Cole, The on-line generalized predictive control combined with a fast transversal filter, in: *Proceedings of the SPIE: Smart Structures and Materials, Damping and Isolation*, Vol. 5049, August 2003, pp. 589–599.
[29] R.M. Glaese, E.H. Anderson, P.C. Janzen, Active suppression of acoustically induced jitter for the airborne laser, in: *Proceedings of the SPIE*, Vol. 4034, 2000, pp. 151–164.
[30] R. Isermann, K.-H. Lachmann, D. Matko, *Adaptive Control Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1992.
[31] R.L. Clark, G.P. Gibbs, W.R. Saunders, *Adaptive Structures, Dynamics and Control*, Wiley, New York, 1998.
[32] E. Mosca, *Optimal, Predictive and Adaptive Control*, Prentice-Hall, Englewood Cliffs, NJ, 1995.