On Time Series Forecasting Error Measures for Finite Horizon Control

Farzad Noorian, *Member, IEEE*, and Philip H. W. Leong, *Senior Member, IEEE*

***Abstract*— Time series forecasting is routinely utilized to improve regulation in finite horizon control (FHC) problems by forecasting the system's uncontrollable inputs. In this brief, we propose a novel measure for validating forecasting models for FHC applications. Specifically, for the case of linear-quadratic time-invariant systems, we derive a closed-form equation for the increase in cost due to forecast error, present techniques for reducing its computational cost, and demonstrate that compared with conventional error measures, model validation using this measure can improve the controller's performance.**

***Index Terms*— Finite horizon control (FHC), linear-quadratic (LQ) regulator, model selection, time series forecasting.**

I. Introduction

FINITE HORIZON CONTROL (FHC) is the process of determining control policies by solving an optimal control problem over a finite time horizon. The FHC methodology is well studied [1], [2], especially for linear-quadratic (LQ) systems, where closed-form or efficient numerical solutions exist. FHC regulators, commonly implemented in discrete time using digital computers, have been successfully applied to many applications including electric power systems [3], inventory management [4] and finance [5].

In FHC, as with other optimal control techniques, a state-space model is defined to allow the predictive modeling of the system's future states to any input. Based on this model, the controllable inputs of the system are determined such that the objective cost function is minimized, and thus the system is controlled toward the desirable outcome.

Classic FHC formulations model the system's uncontrollable exogenous inputs analytically as a part of the state-space equations. This is performed by dividing these inputs into measurable and nonmeasurable external disturbances, and using an analytic model to describe them [2]. However, due to the complexity of real-world applications, developing an accurate analytic model for the system's exogenous inputs is not always possible.

One relatively recent solution is the use of statistical models and machine learning (ML) algorithms as a part of the predictive model, where the time-varying exogenous inputs are treated as a time and estimated via a forecasting algorithm [6]–[8]. In time series forecasting, error measures, such as mean square error (MSE) or mean absolute error (MAE) are used for model validation, by comparing the effectiveness of different forecasting techniques and their parameters for a certain prediction task [9]. However, these assume that the error of each sample is independent of other time-steps, which is not generally true. To obtain full accuracy for model selection, one must consider the control law while validating prediction models over the control horizon. Alternative techniques exist, such as including the conditional distribution of the exogenous input in the dynamic programming problem [10]. Unfortunately, none of these approaches are computationally efficient.

In this brief, we advocate using ΔJ, i.e., the increase in the cost function of an FHC system due to forecast error, for model validation instead of conventional forecasting error measures. The contributions are threefold.

1) We derive an exact closed-form solution to efficiently compute ΔJ for the case of discrete LQ FHC systems.
2) We apply dimensionality reduction techniques to the closed-form solution, reducing the computational cost of ΔJ with minimal loss of accuracy. This leads to a significant speedup in the cases, where several forecasting algorithms and parameters need to be tested for fixed system parameters.
3) We demonstrate via two case studies that by considering the effects of prediction error on FHC dynamics, using ΔJ instead of MSE as a forecasting model selection criterion can improve the system's performance.

The rest of this brief is organized as follows. In Section II, the FHC notation used in this brief is introduced. Section III discusses the effect of additive noise in the FHC formulation and offers a closed-form solution for quantifying it. Time series prediction requirements for FHC regulation are analyzed in Section IV, and a dimensionality reduction technique for reducing cross-validation computational cost is presented accordingly. In Section V, two inventory and portfolio management examples are studied to demonstrate the effectiveness of the proposed formulation. Finally, in Section VI, limitations of this formulation are discussed and possible solutions are reviewed.

II. Background

A discrete linear time invariant (LTI) system can be described using the following state-space model [2]:

$$\boldsymbol{x}(t+1) = \mathbf{A}\boldsymbol{x}(t) + \mathbf{B}\boldsymbol{u}(t) + \mathbf{C}\boldsymbol{v}(t). \tag{1}$$

Here, $\boldsymbol{x}(t) \in \mathbb{R}^k$ is the state vector at time $t \in [0, 1, \ldots, n]$, $\boldsymbol{u}(t) \in \mathbb{R}^l$ is the vector of controllable inputs, $\boldsymbol{v}(t) \in \mathbb{R}^m$

Manuscript received August 30, 2015; revised December 19, 2015 and March 5, 2016; accepted April 23, 2016. Date of publication June 9, 2016; date of current version February 8, 2017. Manuscript received in final form May 19, 2016. This work was supported in part by the Australian Research Council's through the Linkage Projects under Grant LP110200413 and Grant LP130101034, in part by Westpac Banking Corporation, and in part by Zomojo Pty Ltd and Exablaze Pty Ltd. Recommended by Associate Editor A. Alessandri.

The authors are with the School of Electrical and Information Engineering, The University of Sydney, Sydney, NSW 2006, Australia (e-mail: farzad.noorian@sydney.edu.au; philip.leong@sydney.edu.au).

Digital Object Identifier 10.1109/TCST.2016.2571661

is the vector of exogenous inputs, and $\mathbf{A} \in \mathbb{R}^{k \times k}$, $\mathbf{B} \in \mathbb{R}^{k \times l}$, and $\mathbf{C} \in \mathbb{R}^{k \times m}$ are the system and input matrices, respectively, and are controllable.

The objective is to find controllable inputs, $\boldsymbol{u}(t)$, to minimize a quadratic cost function defined by

$$J = \sum_{t=1}^{n} \boldsymbol{x}(t)^T \mathbf{Q}_t \boldsymbol{x}(t) + \sum_{t=0}^{n-1} \boldsymbol{u}(t)^T \mathbf{P}_t \boldsymbol{u}(t) \tag{2}$$

where positive-semidefinite matrix $\mathbf{Q}_t \in \mathbb{R}^{k \times k}$ and positive-definite matrix $\mathbf{P}_t \in \mathbb{R}^{l \times l}$ are stage costs of $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$, respectively.

A. Matrix Form Solution

A common way to find the solution to argmin J is to explicitly express $\boldsymbol{x}(t), \forall t > 0$ as a function of inputs through matrix form, and then minimize J to find all $\boldsymbol{u}(t)$ in a batch approach [2].

In matrix form, the state-space equation (1) and the cost function (2) are represented by

$$\begin{aligned} \boldsymbol{X} &= \mathbf{S}_A \boldsymbol{x}(0) + \mathbf{S}_B \boldsymbol{U} + \mathbf{S}_C \boldsymbol{V} \\ J &= \boldsymbol{X}^T \bar{\mathbf{Q}} \boldsymbol{X} + \boldsymbol{U}^T \bar{\mathbf{P}} \boldsymbol{U} \end{aligned}$$

where

$$\begin{aligned} \boldsymbol{X} &= [\boldsymbol{x}(1)^T \ \boldsymbol{x}(2)^T \ \cdots \ \boldsymbol{x}(n)^T]^T \\ \boldsymbol{U} &= [\boldsymbol{u}(0)^T \ \boldsymbol{u}(1)^T \ \cdots \ \boldsymbol{u}(n-1)^T]^T \\ \boldsymbol{V} &= [\boldsymbol{v}(0)^T \ \boldsymbol{v}(1)^T \ \cdots \ \boldsymbol{v}(n-1)^T]^T \\ \mathbf{S}_A &= [\mathbf{A}^T \ (\mathbf{A}^2)^T \ \cdots \ (\mathbf{A}^n)^T]^T \\ \mathbf{S}_B &= \begin{bmatrix} \mathbf{B} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{AB} & \mathbf{B} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}^{n-1}\mathbf{B} & \mathbf{A}^{n-2}\mathbf{B} & \cdots & \mathbf{B} \end{bmatrix} \\ \mathbf{S}_C &= \begin{bmatrix} \mathbf{C} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{AC} & \mathbf{C} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}^{n-1}\mathbf{C} & \mathbf{A}^{n-2}\mathbf{C} & \cdots & \mathbf{C} \end{bmatrix} \\ \bar{\mathbf{Q}} &= \text{blockdiag}\{\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_n\} \\ \bar{\mathbf{P}} &= \text{blockdiag}\{\mathbf{P}_0, \mathbf{P}_1, \ldots, \mathbf{P}_{n-1}\}. \end{aligned}$$

We define $\boldsymbol{Y}$ as the vector of uncontrollable variables, i.e., the accumulated initial state and exogenous inputs, such that $\boldsymbol{X} = \mathbf{S}_B \boldsymbol{U} + \mathbf{S}_C \boldsymbol{Y}$

$$\boldsymbol{Y} = \mathbf{S}_C^{\dagger} \mathbf{S}_A \boldsymbol{x}(0) + \boldsymbol{V}. \tag{3}$$

Here, † is the Moore–Penrose pseudoinverse operator.

Let

$$\begin{aligned} \mathbf{J}_A &= \bar{\mathbf{P}} + \mathbf{S}_B^T \bar{\mathbf{Q}} \mathbf{S}_B \\ \mathbf{J}_B &= \mathbf{S}_B^T \bar{\mathbf{Q}} \mathbf{S}_C \\ \mathbf{J}_C &= \boldsymbol{Y}^T \mathbf{S}_C^T \bar{\mathbf{Q}} \mathbf{S}_C \boldsymbol{Y} \\ \mathbf{H} &= -\mathbf{J}_A^{\dagger} \mathbf{J}_B. \end{aligned}$$

The optimal control input, $\boldsymbol{U}^*$, is given by

$$\boldsymbol{U}^* = \mathbf{H} \boldsymbol{Y}. \tag{4}$$

Derivation of (4) is described in Appendix A.

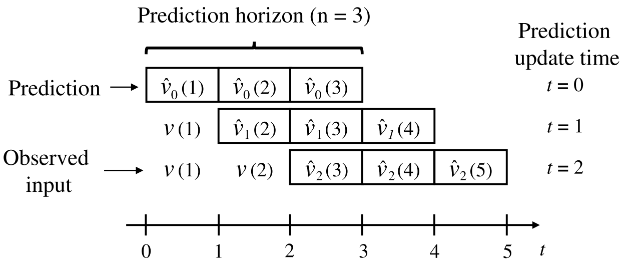

Fig. 1. Prediction updates in FHC. To differentiate between predictions performed at different times, the estimation of $v(t+h)$ predicted at time-step t is denoted with $\hat{v}_t(t+h)$.

B. Finite Horizon Control

The goal of an FHC regulator is to dynamically minimize J over control period $t \in [0, 1, \ldots, N]$. Due to practical considerations, at each time-step, (4) is solved over a finite horizon of length $n < N$ to obtain the optimum control sequence $\boldsymbol{U}$, and only the first action of this sequence is applied. As time moves forward, the model is updated based on observations, and the finite horizon optimization is repeated. This update allows the better control of the system in the presence of external disturbances and model misspecification, at the expense of computational power required for repeated optimization at each time-step.

Updating the horizon at the next time-step takes two major forms in FHC: 1) the horizon is either moved forward, becoming the receding horizon control (RHC) or 2) the same termination time is held, resulting in the shrinking horizon control (SHC). In this brief, both methodologies are studied.

III. Closed-Form Analysis of the Final Cost

With the availability of the exogenous inputs $\boldsymbol{V}$, the optimal controllable inputs $\boldsymbol{U}^*$ and consequently the optimal (i.e., minimum) cost $J^* = J(\boldsymbol{U}^*)$ can be obtained from (4). In real-world applications, the true value of $\boldsymbol{V}$ is not known in advance, and has to be estimated with analytical or numeric models. The inaccuracies of these models, which manifest as prediction error, result in suboptimal control. In this section, we formulate a closed-form equation to measure this suboptimality, in the form of deviation of the cost from its optimal value, that is

$$\Delta J = J(\boldsymbol{U}) - J(\boldsymbol{U}^*).$$

A. Prediction Error

In a real system, the exogenous inputs $\boldsymbol{v}(t)$ are not observed before time t and must be substituted by $\hat{\boldsymbol{v}}(t) = \boldsymbol{v}(t) + \boldsymbol{\epsilon}(t)$, where $\hat{\boldsymbol{v}}(t)$ is their prediction and $\boldsymbol{\epsilon}(t)$ is the additive prediction error.

In a finite horizon approach (either shrinking or receding), at each time-step t, $\hat{\boldsymbol{v}}(t+h)$ is reestimated (i.e., predicted again using the latest available information) for $h \in [1, 2, \ldots, n]$. We denote these reestimation and their prediction errors by $\hat{\boldsymbol{v}}_t(t+h)$ and $\boldsymbol{\epsilon}_t(t+h)$, respectively (Fig. 1).

Similarly, we define the matrix form counterpart of $\boldsymbol{\epsilon}_t(t+h)$ for the tth horizon as

$$\boldsymbol{E}_t = [\boldsymbol{\epsilon}_t(t+1)^T \ \boldsymbol{\epsilon}_t(t+2)^T \ \cdots \ \boldsymbol{\epsilon}_t(t+n)^T]^T.$$

To denote all prediction errors over all control horizons

$$\boldsymbol{E} = \left[\boldsymbol{E}_0^T \ \boldsymbol{E}_1^T \ \ldots \ \boldsymbol{E}_{N-1}^T\right]^T$$

is defined.

B. Optimal Input in the Presence of Prediction Error

The applied controllable input $\boldsymbol{U}$, obtained from the FHC regulator, is a linear function of the accumulated exogenous inputs $\boldsymbol{Y}$ and their prediction errors $\boldsymbol{E}$

$$\boldsymbol{U} = \Psi \boldsymbol{Y} + \Phi \boldsymbol{E} \tag{5}$$

where $\Phi = [\Phi_0 \ \Phi_1 \ \cdots \ \Phi_{N-1}]$ $\Psi = \sum_{i=0}^{N-1} \Phi_i \mathbf{M}_Y(i)$

$$\Phi_i = \sum_{j=0}^{N-1} \mathbf{M}_\Phi(j)\mathbf{H}_j\Gamma(j,i)$$

$$\Gamma(t,\tau) = \begin{cases} t > \tau & \mathbf{0}_{kn_t \times kn_\tau} \\ t = \tau & \mathbf{I}_{kn_t \times kn_t} \\ t < \tau & \displaystyle\sum_{i=0}^{t-1} \mathbf{C}^\dagger \mathbf{A}^{t-i}\mathbf{B}\mathbf{M}_U(i)\mathbf{H}_i\Gamma(i,\tau) \end{cases}$$

$$\mathbf{M}_\Phi(t) = [\mathbf{K}_{i,j}]_{N \times n_t}, \begin{cases} i = t+1, j = 1 & \mathbf{K}_{i,j} = \mathbf{I}_{l \times l} \\ \text{else} & \mathbf{K}_{i,j} = \mathbf{0}_{l \times l} \end{cases}$$

$$\mathbf{M}_U(t) = [\mathbf{K}_{i,j}]_{n_t \times 1}, \begin{cases} i = j = 1 & \mathbf{K}_{i,j} = \mathbf{I}_{l \times l} \\ \text{else} & \mathbf{K}_{i,j} = \mathbf{0}_{l \times l} \end{cases}$$

$$\mathbf{M}_Y(t) = [\mathbf{K}_{i,j}]_{n_t \times N}, \begin{cases} j - i = t & \mathbf{K}_{i,j} = \mathbf{I}_{k \times k} \\ i = 1, j \le t & \mathbf{K}_{i,j} = \mathbf{C}^\dagger \mathbf{A}^{t+1-j}\mathbf{C} \\ \text{else} & \mathbf{K}_{i,j} = \mathbf{0}_{k \times k} \end{cases} \tag{6}$$

n_t is the length of horizon at time-step t, $\mathbf{0}_{f \times g}$ and $\mathbf{I}_{f \times g}$ are $f \times g$ zero and identity matrices, respectively, and $[\mathbf{K}_{i,j}]_{f \times g}$ denotes a block matrix of $f \times g$ subblock matrices. Derivation of (5) is detailed in Appendix B.

Here, $\mathbf{H}_t$ is the matrix used to derive the control law, akin to $\mathbf{H}$ in (4), but considering a time-varying horizon length. For this, the horizon length has been denoted with n_t, in line with the horizon for $\mathbf{H}$, which was denoted with n. This formulation allows (5) to be used with receding horizon, shrinking horizon, or hybrid control schemes. For example, for a receding horizon scheme, where the horizon length remains the same, $\mathbf{H}_t = \mathbf{H}$, while for a shrinking horizon scheme, at each step $\mathbf{H}_t$ has to be recomputed by decrementing n_t.

Note that Ψ in (5) is an extended form of $\mathbf{H}$ in (4). In Section II, $\mathbf{H}$ was designed for a constant horizon length. In contrast, Ψ incorporates the effects of horizon length variation. This is because the goal of (5) is not tocontrol, but an accurate estimation of what would have happened if the control was to be performed using (4) with either a receding or shrinking horizon. For shrinking only horizons (i.e., $n_{t+1} = n_t - 1$), due to the principle of optimality, $\Psi = \mathbf{H}$.

C. Effects of Prediction Error on Cost

Let $\boldsymbol{U}^* = \Psi \boldsymbol{Y}$ be the optimal input. The increase in cost from optimal due to prediction error $\boldsymbol{E}$ is given by

$$\Delta J = \boldsymbol{E}^T \Theta \boldsymbol{E} + \boldsymbol{E}^T \Omega \boldsymbol{Y} \tag{7}$$

where $\Theta = \Phi^T \mathbf{J}_A \Phi$ and $\Omega = 2\Phi^T(\mathbf{J}_A \Psi + \mathbf{J}_B)$. Derivation of (7) is detailed in Appendix C.

IV. Time Series Prediction and FHC

The objective of time series prediction is to find a function $p(\cdot)$ to estimate the future of time series $v(t)$ over prediction horizon $h \in [1, 2, \ldots, n]$ using available data, i.e., $\hat{v}_t(t+h) = p(v(t), v(t-1), v(t-2), \ldots)$.

Time series prediction techniques are usually carried out in two phases: 1) a learning phase, where for a chosen model and its hyperparameters, the training data are fitted to minimize a p-norm error (i.e., $||v(t) - \hat{v}_t(t)||_p$) [11], [12] and 2) a cross-validation phase, where the performance of different models and hyperparameters are compared with a separate set of testing data [13].

Typically, when applying cross validation, an MSE error (i.e., $\boldsymbol{E}^T\boldsymbol{E}$) is used as a measure of comparison. This ignores possible dependencies between errors and also between the errors and the inputs. To include this knowledge of the FHC controller, a step-by-step simulation of the FHC has to be performed: for each discrete time step, the control law is first applied based on predictions, and then the system states and costs are updated according to the actual exogenous inputs. Equivalently, (7) can be used to evaluate the final cost of the prediction error in a single step.

A. Cost Matrix Dimensionality Reduction

Computing ΔJ using (7) is of $O(N^2)$ time complexity, compared with $O(N)$ of using an MSE. Consequently, the efficiency of computing ΔJ has to be improved before being used in data-intensive problems.

In many real-world problems, matrices Θ and Ω prove to be sparse, or even diagonal. In such cases, numerical techniques can be used to improve the efficiency of computing (7).

In other cases, assuming repeated evaluation on a fixed system, one can precompute Θ and $\Omega \boldsymbol{Y}$, and calculate ΔJ efficiently for different values of $\boldsymbol{E}$, the latter coming from different prediction models.

With this assumption, matrix decomposition can also be used to further reduce computational complexity by approximating matrix $\Theta \in \mathbb{R}^{M \times M}$ with a matrix of lower rank [14]. Here, M is the total number of predictions and is obtained from the sum of horizon lengths and the size of the inputs vector, i.e., $M = m \sum_i n_i$.

Let $\Theta = \boldsymbol{V}\Sigma\boldsymbol{V}^T$, where Σ is the diagonal matrix of eigenvalues and $\boldsymbol{V}$ is the matrix of eigenvectors of Θ. By only keeping the $0 < L < M$ largest eigenvalues of Θ and their corresponding eigenvectors, Θ can be approximated with $\Theta' = \boldsymbol{V}'\Sigma'\boldsymbol{V}'^T$, where $\boldsymbol{V}' \in \mathbb{R}^{M \times L}$ and $\Sigma' \in \mathbb{R}^{L \times L}$.

Let $\mathbf{W} = \boldsymbol{V}'\sqrt{\Sigma'}$, where $\sqrt{\Sigma'}$ is the root square of the diagonal eigenvalue matrix. An approximation to (7) is

$$\Delta J' = \boldsymbol{E}^T \mathbf{W}'\mathbf{W}'^T \boldsymbol{E} + \boldsymbol{E}^T \Omega \boldsymbol{Y}. \tag{8}$$

Exploiting the symmetric structure of the new cost function and assuming precomputed $\mathbf{W}'$ and $\Omega \boldsymbol{Y}$, evaluating (8) is reduced to time complexity O(ML).

The choice of L is problem-dependent. A general guideline is to select L such that $\mathrm{tr}(\Sigma') \geq \lambda \mathrm{tr}(\Sigma)$, where tr is the trace operator and $0 \leq \lambda \leq 1$ determines how much of the matrix's energy is to be conserved. In practice, $\lambda > 0.99$ is commonly used.

V. Numerical Examples and Simulation Results

In this section, two finite horizon problems with time series forecasting are analyzed, and the proposed error measure ΔJ is compared with the MSE for predictor selection. The MSE was chosen as it offers a quadratic error function, similar to the controller's cost function, and is not scaled against the magnitude of inputs.

For each problem, first the system dynamics are defined and used to formulate an FHC control problem. The best predictor models are then selected over a set of training data using different error measures. Consequently, a simulation is performed over a separate set of testing data, and the performance results of models selected using ΔJ and the MSE are compared.

Implementations of these examples, written in the *R* programming language, are available from http://sydney.edu.au/engineering/electrical/cel/farzad/TCST16.

A. Preordering Problem

Inventory management and supply chain planning techniques play an essential role in managing supply and demand, and are widely studied and used in practice [15]. Recent developments in this regard have shown that forward-looking optimization-based policies, such as using optimal control in combination with forecasting, significantly outperform other rule-based decision policies [6], [16].

In this example, we study the problem of meeting a fluctuating demand for a perishable commodity, similar to the problem discussed in [17]. In this task, one can either preorder the perishable item with different lead times and discounts, or buy it on the spot market at a higher price. The objective is to minimize the ordering costs by utilizing prediction.

1) Problem Formulation: We formalize this problem as follows.

1) The demand is denoted with $v(t) \in \mathbb{R}^{+}$, and the spot price is depicted with p.
2) It is possible to preorder κ steps ahead, where at each time-step a discount of d is applied.
3) A preorder can be adjusted at any time before delivery; however, an adjustment penalty equal to the discounted price is applied. Similarly, a penalty is applied for oversupplied orders (i.e., discarded perishables).
4) Preorders are denoted with $\boldsymbol{u}(t) = [u_{t+1}(t) \ \cdots \ u_{t+\kappa-1}(t) \ u_{t+\kappa}(t)]^T$, where $u_{t+h}(t)$ is the order (or adjustment to the order) at time t for delivery at time $t+h$.
5) An order book is maintained, in the form of the vector $\boldsymbol{x}(t) = [x_t(t) \ x_{t+1}(t) \ \cdots \ x_{t+\kappa}(t)]^T$, where $x_{t+h}(t)$ is the total of orders expected at time t to be delivered at time $t+h$.

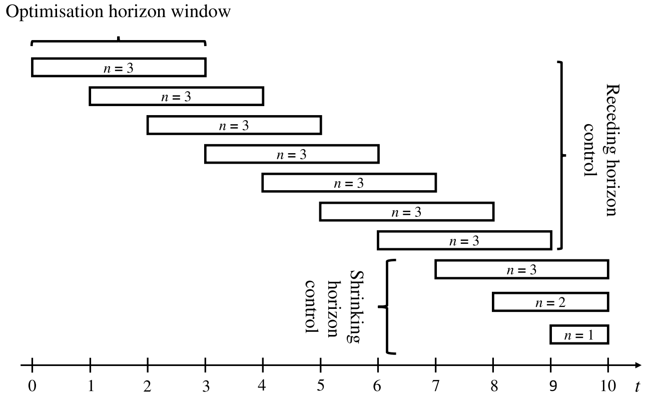

Fig. 2. Hybrid RHC–SHC scheme, with a maximum horizon length of 3.

Notice that $\boldsymbol{u}(t) \in \mathbb{R}^{\kappa}$, and $\boldsymbol{x}(t) \in \mathbb{R}^{\kappa+1}$ as it includes a state for the delivery at time-step t in addition to the future preorders.

The problem can be formulated using the state-space equation (1), as updating the order book using $x_{t+h}(t+1) = x_{t+h}(t) + u_{t+h}(t)$ when preordering (i.e., $h > 1$), and $x_{t+1}(t+1) = x_{t+1}(t) - v(t)$ during delivery.

Assuming $\kappa = 3$, the system dynamics in matrix form are

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Equation (2) is used as the cost function, with the stage cost of $\boldsymbol{u}(t)$ (i.e., preordering prices and discounts), and the stage cost of $\boldsymbol{x}(t)$ (i.e., the penalty for unmet demand), defined, respectively, as

$$\mathbf{P}_t = \begin{bmatrix} d^3 p & 0 & 0 \\ 0 & d^2 p & 0 \\ 0 & 0 & dp \end{bmatrix} \quad \mathbf{Q}_t = \begin{bmatrix} p & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

For numeric simulation, the spot price was set to $p = 4$, and the discount to $d = 0.7$.

2) Analysis of the Cost Equation: An analysis of Θ and $\boldsymbol{\Omega}$ from (7) for the current problem reveals the matrix elements associated with the prediction of the current demand, and the prediction of steps more than κ step ahead, are zero. The former is a result of the observed demand (and not the prediction) being used for spot market ordering. The latter is because any prediction beyond κ steps is not used for ordering in the current horizon. Consequently, choosing a horizon length beyond the number of preorders is redundant.

Furthermore, Θ is a diagonal matrix for optimizations ending with shrinking horizons (e.g., as in Fig. 2). In addition, the diagonal elements for each set of predictions decay exponentially. For example, for $\kappa = 3$ and a horizon of $n = 5$, the diagonal associated with the first block of predictions is [0 2.98 0.35 0.13 0]. This simplifies ΔJ to a weighted MSE.

3) Simulation and Results: The simulation was set up to run for ten time-steps, with preordering allowed for three steps ahead (i.e., $N = 10$ and $\kappa = 3$). Based on the analysis of Θ

for this problem, the control horizon n was limited to the number of preorders κ (i.e., $n_t \leq \kappa = 3, \forall t$). A finite horizon controller governs the system, with seven receding windows of length 3, followed by three shrinking windows (Fig. 2).

The hth step in the future, $v(t+h)$, was predicted using an auto-regressive (AR) model

$$\hat{v}_t(t+h) = \sum_{i=0}^{q} \phi_i v(t-i) \tag{9}$$

where q is the model order, and values of ϕ_i are the model parameters. For each future step in a horizon, a different model was selected and fitted to data using ordinary least squares, resulting in a total of three models. The range of orders was limited to $q \in [1, 2, \ldots, 8]$.

To simulate real-world demand, time series from the M3 competition data set [18] were used. Only time series longer than 100 samples were selected for simulation. For each time series, the first 80 values were assigned to in-sample testing, with the starting 60 time-steps used for training the predictor, and the next 20 values for model validation. The last 20 were used to report out-of-sample prediction errors, as well as the controller's cost performance, i.e., the out-of-sample ΔJ.

Three different approaches were used for model order selection through cross-validation.

1) A random search was performed to select the best orders using an in-sample MSE. A subset of the models was sampled from the model space using a uniform distribution, and the model with the least cross-validation error was selected.
2) A random search using in-sample ΔJ as the selection criterion was performed.
3) A hybrid search was undertaken. The best model for the first step (i.e., $h = 1$) was chosen independently, based on the in-sample MSE through an exhaustive search. The model orders for $h = 2$ and $h = 3$ were selected using in-sample ΔJ through a random search.

The first two methods compare MSE and ΔJ model selection when the computational resources are limited. The design rationale of the third technique is based on the Θ weights, where the first step contributes 84% of the total error. Hence, half of the computational capacity is solely allocated to the first predictor's model selection.

Also note that in the hybrid search, the first step was chosen using MSE. Considering the lack of dependence between the error at $h = 1$ and other steps, the ΔJ for only $h = 1$ is equivalent to a weighted MSE with a constant weight, i.e., 2.98 according to the analysis of Section V-A2. We simplified this further by removing the constant, resulting in an ordinary MSE measure.

For a fair comparison, only 16 model evaluations were allowed per approach. For the hybrid approach, this translated into eight evaluations, exhaustively searching for the best first model order, and eight additional random evaluations for the second and third model orders.

TABLE I
RUN-TIME AND SPEEDUP COMPARISON FOR THE PREORDERING PROBLEM

Measurement Technique	Run-time (s)	Speed-up
Step-by-step simulation	2340.7	1
Closed form ΔJ using (7)	11.730	199.5
Diagonal ΔJ	0.9130	2563.7
MSE	0.8940	2619.0

TABLE II
MEAN NORMALIZED PREDICTION ERROR FOR DIFFERENT MODEL SELECTION METHODS IN THE PREORDERING PROBLEM

Selection Method	In-sample MSE	In-sample ΔJ	Out-of-sample MSE	Cost Performance
Random Search (using MSE)	**0.9078**	0.8953	**0.9261**	0.9208
Random Search (using ΔJ)	0.937	0.867	0.941	0.9109
Hybrid Search (based on ΔJ)	0.9081	**0.8533**	0.9321	**0.9072**

The test was repeated for each of the 1020 selected time series. As each time series exhibits different mean-variance characteristics, the resulting MSE and ΔJ are not directly comparable. Consequently, these errors were normalized to the MSE and ΔJ of a naïve predictor, respectively, where the predictor simply repeats the last observed value, i.e., $\hat{v}_t(t+h) = v(t), \forall h$. The normalized results were then averaged and reported.

To reduce the computation time of ΔJ, considering the diagonal nature of Θ, (7) was also numerically implemented as a weighted MSE. Table I compares the run-times of different methods. It is observed that computing ΔJ using (7), even in its full matrix form, offers a significant speedup compared with a step-by-step FHC simulation. Furthermore, the diagonal only implementation is almost as efficient as an ordinary MSE, achieving a speedup of more than 2500×.

In Table II, prediction errors for models selected using the MSE and ΔJ are summarized. While the models selected using the MSE offered a better in-sample and out-of-sample MSEs, they were outperformed over the cost performance by the models selected using ΔJ. Overall, a cost improvement of 1.5% was obtained simply by prioritizing the first step's model selection over other steps. A paired sample t-test of results rejected the null hypothesis of improvement not being significant with p value of 0.0027.

Considering the independence of the three model orders and their errors, the best global model would have minimized both the MSE and ΔJ. This example demonstrates that when a global search is not possible (e.g., limited computational resources), the search can be focused on the most influential factor, as analyzed by the proposed cost measure, to improve model selection.

B. Stock Portfolio Management

We extend the previous example to financial markets, where a dealer keeps a portfolio of stocks to trade on behalf of his or her clients. The dealer wishes to reduce the costs associated with: 1) the market risk, i.e., the loss of portfolio value due to market price changes and 2) trading with other dealers, when the client's requests cannot be fulfilled using what is available in the portfolio. The first issue forces the dealer to minimize the inventory to avoid risk, while the second obligates keeping all client trades in the portfolio, such that opposing client trades (i.e., buys and sells) are neutralized without referring to other dealers.

The problem of finding the optimal portfolio subject to cost and risk considerations has been extensively researched. Recent studies of this problem for stock options and FX market, using FHC but neglecting the effects of time series prediction error, can be found in [19]–[21].

1) Problem Formulation: We simplify the problem by assuming a single-stock inventory with the following rules and notations.

1) $x(t)$ denotes the dealer's inventory, interdealer trades are determined using $u(t)$, and the demand is denoted with $v(t)$.
2) Short-selling is allowed.
3) To consider market impact, the interdealer brokering cost is modeled using a quadratic function of trades, $P_t u^2(t)$.
4) The risk is modeled using $Q_t x^2(t)$, where Q_t is the market volatility, i.e., the variance of the price process as used in modern portfolio optimization [22].

The dealer's dynamics can be formulated using the state-space equation (1), with $A = 1$, $B = 1$, and $C = 1$. The dealer begins with a zero balance, i.e., $x(0) = 0$. To concentrate on demand prediction, we assume the cost of trading with interdealer brokers and the market volatility are available and constant in time, $P_t = 1$ and $Q_t = 1$, respectively. The objective is to minimize the overall cost, as defined by (2).

2) Simulation and Results: Similar to Section V-A, a receding/shrinking horizon controller was designed to govern the system for ten time-steps. The controller uses five receding windows of length 5 followed by five windows with shrinking lengths.

To predict v_t, separate linear models described by (9) were considered for each future time-step in the horizon. The order of each model was limited to $q \in [2, 3, \dots, 8]$.

The clients' trades were simulated via a fifth order AR model

$$v_{t+1} = 2.76v_t - 3.13v_{t-1} + 1.79v_{t-2} - 0.50v_{t-3} + 0.05v_{t-4} + \epsilon_t \tag{10}$$

where $\epsilon_t \sim N(0, 1)$ is a zero-mean unit-variance Gaussian noise.

For each test, a time series of length 100 was generated. The first 80 values were assigned to in-sample testing, with the starting 60 time-steps used for training the predictor, and the next 20 values for model validation. The last 20 were used to report out-of-sample prediction errors, as well as the controller's cost performance, i.e., the out-of-sample ΔJ. The test was repeated ten times.

Our first concern is the dimensionality reduction for $\Theta \in \mathbb{R}^{40\times40}$. Despite its simpler formulation compared with the problem of Section V-A, Θ is not diagonal, and consequently the eigenvalue method has to be used. The ten largest eigenvalues hold more than 99.9% of eigenvalue energy, and thus a fourfold order reduction is possible using $\mathbf{W}' \in \mathbb{R}^{40\times10}$.

TABLE III

RUN-TIME AND ACCURACY COMPARISON FOR THE STOCK PORTFOLIO MANAGEMENT PROBLEM

Measurement Technique	Run-time (s)	Speed-up	Measured ΔJ
Step-by-step simulation	1427.664	1	96.8476
Closed form ΔJ using (7)	12.136	117.6	96.8476
Approximated ΔJ using (8)	2.826	598.1	96.8476
MSE	1.228	1376.0	N/A

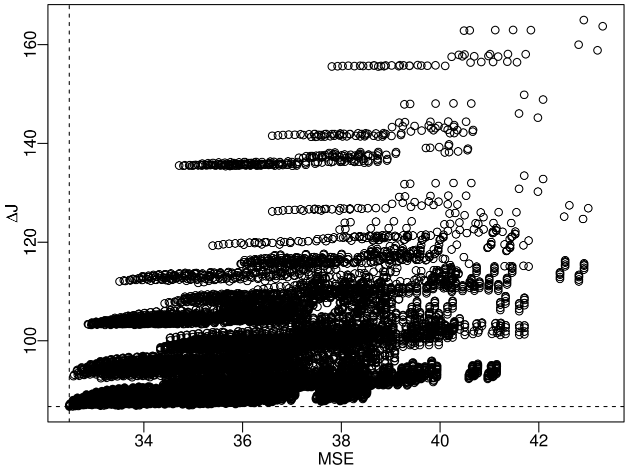

Fig. 3. MSE versus ΔJ for different model orders in the stocks portfolio problem.

Table III compares the time required for computing each error measure and the aggregated value of errors over all tests. It is observed that the proposed dimensionality reduction technique results in near 5× speedup, with <0.001% loss of accuracy. Comparing run-times of Table III, an overall 598× speedup over a step-by-step FHC simulation is observed. While the MSE is still 2.3× faster, the difference in execution is 1.6 s, which is negligible compared with the time spent for training the prediction model.

Fig. 3 compares the in-sample MSE error with ΔJ for all models. It is evident that these errors are not strongly correlated. The numerical value of the correlation coefficient for MSE and ΔJ was measured to be 0.423, compared with that of the MSE and the MAE being 0.893. As a result, model selection using MSE does not necessarily improve ΔJ.

An exhaustive search was performed to cross validate different model orders over the in-sample period using MSE and ΔJ. For each of these measures, the model with the least in-sample error was selected, and its performance results for both the in-sample and out-of-sample periods were summarized in Table IV. It can be seen that using ΔJ as a model selection measure has reduced the controller's cost compared with selections using the ordinary MSE measure. While this reduction is not significant for in-sample ΔJ results, the controller's cost performance improvement, from 98.18 to 94.79, is considerable. In addition, a paired sample t-test rejected the null hypothesis of improvement not being significant with $p = 0.0094$.

TABLE IV

MEAN PREDICTION ERROR FOR DIFFERENT MODEL SELECTION MEASURES IN THE STOCK PORTFOLIO MANAGEMENT PROBLEM

Measure for selection	In-sample MSE	In-sample ΔJ	Out-of-sample MSE	Cost Performance
MSE	**32.4888**	86.9027	**36.3447**	98.1846
ΔJ	32.4892	**86.7015**	38.0916	**94.7870**

VI. Discussion on Problems With Constraints

The proposed error measure, as formulated in (7), is limited to LQ systems without constraints. In real-world applications, however, constraints are prevalent.

The easiest method for extending ΔJ to such applications is relaxing the constraints in J. Nevertheless, this will be at the cost of reduced accuracy, as the results will no longer match the original cost function formulation. An alternative approach is to use multiparametric programming to obtain an explicit solution for the FHC cost function with constraints [23]. For an LTI LQ system, this transforms the quadratic programming problem to a piecewise affine solution. Considering the current affine form of H and the quadratic form of ΔJ, one would expect a piecewise quadratic form for the constrained ΔJ. While obtaining the solution in this form is computationally intensive and may require massive memory storage, it will be more efficient in evaluating several forecasting models' results compared with the repeated runs of quadratic programming in FHC simulation.

Implementing and benchmarking this multiparametric approach will be the subject of a future study.

VII. Conclusion

This brief presented several observations regarding time series prediction used in LQ FHC regulators. Using the proposed error measure, better model selection results were obtained, since in contrast to MSE, ΔJ accurately describes the effects of prediction on the objective cost function. In addition, techniques were presented to reduce the computational costs of this measure, in some cases even comparable to that of MSE, and thus making ΔJ a viable replacement in data-intensive applications.

Appendix A
Derivation of Optimal Input in Presence of Error

The system state in the presence of prediction error is obtained by

$$\begin{aligned} \boldsymbol{X} &= \mathbf{S}_A \boldsymbol{x}(0) + \mathbf{S}_B \boldsymbol{U} + \mathbf{S}_C(\boldsymbol{V} + \boldsymbol{E}) \\ &= \mathbf{S}_B \boldsymbol{U} + \mathbf{S}_C \boldsymbol{Y} + \mathbf{S}_C \boldsymbol{E}. \end{aligned}$$

The cost function can be expanded and rewritten as

$$\begin{aligned} J &= \boldsymbol{X}^T \bar{\mathbf{Q}} \boldsymbol{X} + \boldsymbol{U}^T \bar{\mathbf{P}} \boldsymbol{U} \\ &= (\mathbf{S}_B \boldsymbol{U} + \mathbf{S}_C \boldsymbol{Y} + \mathbf{S}_C \boldsymbol{E})^T \bar{\mathbf{Q}} (\mathbf{S}_B \boldsymbol{U} + \mathbf{S}_C \boldsymbol{Y} + \mathbf{S}_C \boldsymbol{E}) \\ &\quad + \boldsymbol{U}^T \bar{\mathbf{P}} \boldsymbol{U} \\ &= \boldsymbol{U}^T \mathbf{J}_A \boldsymbol{U} + 2\boldsymbol{U}^T \mathbf{J}_B (\boldsymbol{Y} + \boldsymbol{E}) \\ &\quad + \left(\boldsymbol{Y}^T \mathbf{S}_C^T \bar{\mathbf{Q}} \mathbf{S}_C \boldsymbol{Y} + 2\boldsymbol{E}^T \mathbf{S}_C^T \bar{\mathbf{Q}} \mathbf{S}_C \boldsymbol{Y} + \boldsymbol{E}^T \mathbf{S}_C^T \bar{\mathbf{Q}} \mathbf{S}_C \boldsymbol{E}\right). \end{aligned}$$

Using the optimality condition, namely $(dJ/d\boldsymbol{U}) = 0$, the solution to argmin J can be derived

$$\boldsymbol{U} = -\mathbf{J}_A^{\dagger} \mathbf{J}_B (\boldsymbol{Y} + \boldsymbol{E}) = \mathbf{H}(\boldsymbol{Y} + \boldsymbol{E}).$$

When $\boldsymbol{E} = 0$, the equation simplifies to $\boldsymbol{U}^* = \mathbf{H}\boldsymbol{Y}$.

Appendix B
Derivation of Input in the Repeated Application of the Control Law

In FHC, controllable inputs are reevaluated at each time-step. For example, at time-step t

$$\boldsymbol{U}_t = \mathbf{H}_t(\Upsilon_t + \boldsymbol{E}_t) = \mathbf{H}_t \hat{\Upsilon}_t \tag{11}$$

where $\boldsymbol{U}_t$ is the controllable input, optimized using information available at time t, and $\Upsilon_t \in \mathbb{R}^{n_t m}$ is constructed by accumulating previous exogenous inputs [i.e., $\boldsymbol{v}(i), i < t$] and the initial state $\boldsymbol{x}(0)$ into the first value of $\boldsymbol{Y}_t$, the tth horizon window of $\boldsymbol{Y}$. $\boldsymbol{Y}_t$ can be obtained by the affine mapping

$$\boldsymbol{Y}_t = \mathbf{M}_Y(t)\boldsymbol{Y}. \tag{12}$$

$\hat{\Upsilon}_t$ is the estimate (i.e., prediction) of Υ_t and includes the prediction error at time t, i.e., $\hat{\Upsilon}_t = (\Upsilon_t + \boldsymbol{E}_t)$.

At any time-step, Υ_{t+1} can be computed by accumulating the applied input controls $\boldsymbol{u}_i$ for $i < t$, into the $\boldsymbol{Y}_{t+1}$

$$\hat{\Upsilon}_{t+1} = \boldsymbol{Y}_{t+1} + \boldsymbol{E}_{t+1} + \mathbf{C}^{\dagger} \sum_{i=0}^{t} \mathbf{A}^{t+1-i} \mathbf{B} \mathbf{M}_U(i) \boldsymbol{U}_i.$$

Here, $\mathbf{M}_U$ is used to extract $\boldsymbol{u}_i$ by separating the first vector of inputs from $\boldsymbol{U}_t$; then $\mathbf{A}^{(t+1)-i}\mathbf{B}$ is used in a recursive approach (similar to applying $\mathbf{S}_B$) to accumulate its effect on the system's states. Finally $\mathbf{C}^{\dagger}$ is applied, similar to $\mathbf{S}_C^{\dagger}$ in $\boldsymbol{Y} = \mathbf{S}_C^{\dagger}\mathbf{S}_A \boldsymbol{x}(0) + \boldsymbol{V}$, to reshape the resulting new states into the new uncontrollable input vector.

The equation above can be rewritten in a recursive form by replacing $\boldsymbol{U}_i$ with (11)

$$\hat{\Upsilon}_{t+1} = \boldsymbol{Y}_{t+1} + \boldsymbol{E}_{t+1} + \sum_{i=0}^{t} \mathbf{C}^{\dagger} \mathbf{A}^{t+1-i} \mathbf{B} \mathbf{M}_U(i) \mathbf{H}_i \hat{\Upsilon}_i$$

which can be expanded recursively and then factored to

$$\hat{\Upsilon}_{t+1} = \sum_{i=0}^{t+1} \Gamma(t+1, i)(\boldsymbol{Y}_i + \boldsymbol{E}_i).$$

The final $\boldsymbol{U}$ is constructed using $\mathbf{M}_\Phi(j)$, which puts the first inputs of each $\boldsymbol{U}_j$ into the jth position of $\boldsymbol{U}$

$$\begin{aligned} \boldsymbol{U} &= \sum_{j=0}^{N-1} \mathbf{M}_\Phi(j) \boldsymbol{U}_j \\ &= \sum_{j=0}^{N-1} \mathbf{M}_\Phi(j) \mathbf{H}_j \sum_{i=0}^{N-1} \Gamma(j, i)(\boldsymbol{Y}_i + \boldsymbol{E}_i). \end{aligned}$$

By letting $\Phi_i = \sum_{j=0}^{N-1} \mathbf{M}_\Phi(j) \mathbf{H}_j \Gamma(j, i)$, substituting $\boldsymbol{Y}_i$ in (12), and considering that $\mathbf{M}_U$ is nonzero only for $i = j$, this simplifies to

$$\boldsymbol{U} = \sum_{i=0}^{N-1} \Phi_i (\mathbf{M}_Y(i)\boldsymbol{Y} + \boldsymbol{E}_i).$$

Let $\Psi = \sum_{i=0}^{N-1} \Phi_i \mathbf{M}_Y(i)$ and $\Phi = [\Phi_0 \;\; \Phi_1 \;\; \cdots \;\; \Phi_{N-1}]$. The vector $\boldsymbol{U}$ can be reorganized as

$$\boldsymbol{U} = \Psi \boldsymbol{Y} + \Phi \boldsymbol{E}. \tag{13}$$

Appendix C
Derivation of Cost Deviation

The objective cost of the system as a function of controllable and exogenous inputs $\boldsymbol{U}$ and $\boldsymbol{Y}$ is realized by

$$J(\boldsymbol{U}) = \boldsymbol{U}\mathbf{J}_A\boldsymbol{U} + 2\boldsymbol{U}\mathbf{J}_B\boldsymbol{Y} + \mathbf{J}_C.$$

Let $\boldsymbol{U}_e = \boldsymbol{U} - \boldsymbol{U}^* = \Phi\boldsymbol{E}$ be the difference of the controllable input computed based on predictions, and the optimal input (i.e., without prediction error). The cost difference between minimum cost using $\boldsymbol{U}^*$ [i.e., $J(\boldsymbol{U}^*)$] and the cost with prediction error using $\boldsymbol{U}$ [i.e., $J(\boldsymbol{U})$], is given by

$$\begin{aligned}\Delta J &= J(\boldsymbol{U}) - J(\boldsymbol{U}^*)\\ &= \boldsymbol{U}^T\mathbf{J}_A\boldsymbol{U} + 2\boldsymbol{U}\mathbf{J}_B\boldsymbol{Y} - \boldsymbol{U}^{*T}\mathbf{J}_A\boldsymbol{U}^* - 2\boldsymbol{U}^*\mathbf{J}_B\boldsymbol{Y}.\end{aligned}$$

Expanding $\boldsymbol{U} = \boldsymbol{U}^* + \boldsymbol{U}_e$ results in

$$\begin{aligned}\Delta J &= (\boldsymbol{U}^* + \boldsymbol{U}_e)^T\mathbf{J}_A(\boldsymbol{U}^* + \boldsymbol{U}_e) - \boldsymbol{U}^{*T}\mathbf{J}_A\boldsymbol{U}^*\\ &\quad + 2(\boldsymbol{U}^* + \boldsymbol{U}_e)^T\mathbf{J}_B\boldsymbol{Y} - 2\boldsymbol{U}^{*T}\mathbf{J}_B\boldsymbol{Y}\\ &= \boldsymbol{U}_e^T\mathbf{J}_A\boldsymbol{U}_e + 2\boldsymbol{U}_e^T\mathbf{J}_A\boldsymbol{U}^* + 2\boldsymbol{U}_e^T\mathbf{J}_B\boldsymbol{Y}.\end{aligned}$$

By replacing $\boldsymbol{U}^*$ with its definition from (13), considering $\boldsymbol{E} = 0$, we can simplify the above to

$$\begin{aligned}\Delta J &= \boldsymbol{U}_e^T\mathbf{J}_A\boldsymbol{U}_e + 2\boldsymbol{U}_e^T\mathbf{J}_A(\Psi\boldsymbol{Y}) + 2\boldsymbol{U}_e^T\mathbf{J}_B\boldsymbol{Y}\\ &= \boldsymbol{U}_e^T\mathbf{J}_A\boldsymbol{U}_e + 2\boldsymbol{U}_e^T(\mathbf{J}_A\Psi + \mathbf{J}_B)\boldsymbol{Y}.\end{aligned}$$

Let $\Theta = \Phi^T\mathbf{J}_A\Phi$ and $\Omega = 2\Phi^T(\mathbf{J}_A\Psi + \mathbf{J}_B)$. Replacing $\boldsymbol{U}_e = \Phi\boldsymbol{E}$, the above transforms to

$$\begin{aligned}\Delta J &= \boldsymbol{E}^T(\Phi^T\mathbf{J}_A\Phi)\boldsymbol{E} + 2\boldsymbol{E}^T\Phi^T(\mathbf{J}_A\Psi + \mathbf{J}_B)\boldsymbol{Y}\\ &= \boldsymbol{E}^T\Theta\boldsymbol{E} + \boldsymbol{E}^T\Omega\boldsymbol{Y}.\end{aligned}$$

References

[1] G. C. Goodwin, M. M. Seron, and J. A. De Doná, *Constrained Control and Estimation: An Optimisation Approach*. London, U.K.: Springer, 2006.

[2] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control*. 2015. [Online]. Available: http://www.mpc.berkeley.edu/mpc-course-material

[3] H.-S. Ko, J. Jatskevich, G. Dumont, and G.-G. Yoon, "An advanced LMI-based-LQR design for voltage control of grid-connected wind farm," *Electr. Power Syst. Res.*, vol. 78, no. 4, pp. 539–546, 2008.

[4] P. Ignaciuk and A. Bartoszewicz, "Linear-quadratic optimal control of periodic-review perishable inventory systems," *IEEE Trans. Control Syst. Technol.*, vol. 20, no. 5, pp. 1400–1407, Sep. 2012.

[5] G. C. Calafiore and F. Kharaman, "Multi-period asset allocation with lower partial moments criteria and affine policies," in *Proc. IEEE Symp. Comput. Intell. Financial Eng. Econ. (CIFEr)*, London, U.K., Mar. 2014, pp. 100–106.

[6] P. Doganis, E. Aggelogiannaki, and H. Sarimveis, "A combined model predictive control and time series forecasting framework for production-inventory systems," *Int. J. Prod. Res.*, vol. 46, no. 24, pp. 6841–6853, 2008.

[7] K. Patan, "Neural network-based model predictive control: Fault tolerance and stability," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 3, pp. 1147–1155, May 2015.

[8] M. Liu, Y. Shi, and F. Fang, "Load forecasting and operation strategy design for CCHP systems using forecasted loads," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 5, pp. 1672–1684, Sep. 2015.

[9] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *Int. J. Forecasting*, vol. 22, no. 4, pp. 679–688, 2006.

[10] J. Skaf, S. Boyd, and A. Zeevi, "Shrinking-horizon dynamic programming," *Int. J. Robust Nonlinear Control*, vol. 20, no. 17, pp. 1993–2002, 2010.

[11] G. P. Zhang and M. Qi, "Neural network forecasting for seasonal and trend time series," *Eur. J. Oper. Res.*, vol. 160, no. 2, pp. 501–514, 2005.

[12] L. Cao, "Support vector machines experts for time series forecasting," *Neurocomputing*, vol. 51, pp. 321–339, Apr. 2003.

[13] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*. OTexts, 2013. [Online]. Available: https://www.otexts.org/fpp

[14] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, Sep. 1936.

[15] T. C. Miller, *Hierarchical Operations and Supply Chain Planning*. London, U.K.: Springer-Verlag, 2002.

[16] J. D. Schwartz, W. Wang, and D. E. Rivera, "Simulation-based optimization of process control policies for inventory management in supply chains," *Automatica*, vol. 42, no. 8, pp. 1311–1320, 2006.

[17] J. Mattingley, Y. Wang, and S. Boyd, "Receding horizon control: Automatic generation of high-speed solvers," *IEEE Control Syst. Mag.*, vol. 31, no. 3, pp. 52–65, Jun. 2011.

[18] S. Makridakis and M. Hibon, "The M3-competition: Results, conclusions and implications," *Int. J. Forecasting*, vol. 16, no. 4, pp. 451–476, Dec. 2000.

[19] A. Bemporad, L. Puglia, and T. Gabbriellini, "A stochastic model predictive control approach to dynamic option hedging with transaction costs," in *Proc. Amer. Control Conf. (ACC)*, 2011, pp. 3862–3867.

[20] F. Noorian and P. H. W. Leong, "Dynamic hedging of foreign exchange risk using stochastic model predictive control," in *Proc. IEEE Conf. Comput. Intell. Financial Eng. Econ. (CIFEr)*, London, U.K., Mar. 2014, pp. 441–448.

[21] F. Noorian, B. Flower, and P. H. W. Leong, "Stochastic receding horizon control for short-term risk management in foreign exchange," *J. Risk*, vol. 18, no. 5, pp. 1–38, 2016.

[22] H. Markowitz, "Portfolio selection," *J. Finance*, vol. 7, no. 1, pp. 77–91, 1952.

[23] 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.