

# On Time Series Forecasting Error Measures for Finite Horizon Control

Farzad Noorian, *Member, IEEE*, 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 paper, 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 to conventional error measures, model validation using this measure can improve the controller’s performance.

**Index Terms**—Time series forecasting, Model selection, Finite horizon control, Linear quadratic regulator

## I. INTRODUCTION

**F**INITE 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 *predictive modeling* of the system’s future states to any input. Based on this model, controllable inputs of the system are determined such that the objective cost function is minimized, and thus the system is controlled towards 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 *non-measurable 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 series* 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 paper, we advocate using  $\Delta J$ , 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  $\Delta J$  for the case of discrete LQ FHC systems.
- 2) We apply dimensionality reduction techniques to the closed form solution, reducing the computational cost of  $\Delta J$  with minimal loss of accuracy. This leads to a significant speed-up in 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  $\Delta J$  instead of MSE as a forecasting model selection criterion can improve the system’s performance.

The rest of this paper is as follows. In Section II, the FHC notation used in this paper is introduced. Section III discusses the effect of additive noise in 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]:

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

Here,  $\mathbf{x}(t) \in \mathbb{R}^k$  is the state vector at time  $t \in [0, 1, \dots, n]$ ,  $\mathbf{u}(t) \in \mathbb{R}^l$  is the vector of controllable inputs,  $\mathbf{v}(t) \in \mathbb{R}^m$  is the vector of exogenous inputs, and  $\mathbf{A} \in \mathbb{R}^{k \times k}$ ,  $\mathbf{B} \in \mathbb{R}^{k \times l}$

This research was supported under Australian Research Council’s Linkage Projects funding scheme (project numbers LP110200413 and LP130101034), Westpac Banking Corporation, and Zomojo Pty Ltd.

F. Noorian and P.H.W. Leong are with the School of Electrical and Information Engineering, The University of Sydney, NSW, 2006 Australia (e-mail: {farzad.noorian, philip.leong}@sydney.edu.au).

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,  $\mathbf{u}(t)$ , to minimize a quadratic cost function defined by

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

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  $\mathbf{x}(t)$  and  $\mathbf{u}(t)$  respectively.

### A. Matrix Form Solution

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

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

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

where

$$\begin{aligned} \mathbf{X} &= [\mathbf{x}(1)^T \ \mathbf{x}(2)^T \ \cdots \ \mathbf{x}(n)^T]^T, \\ \mathbf{U} &= [\mathbf{u}(0)^T \ \mathbf{u}(1)^T \ \cdots \ \mathbf{u}(n-1)^T]^T, \\ \mathbf{V} &= [\mathbf{v}(0)^T \ \mathbf{v}(1)^T \ \cdots \ \mathbf{v}(n-1)^T]^T, \\ \mathbf{S}_A &= [\mathbf{A}^T \ (\mathbf{A}^2)^T \ \cdots \ (\mathbf{A}^n)^T]^T, \\ \mathbf{S}_B &= \begin{bmatrix} \mathbf{B} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{AB} & \mathbf{B} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}^{n-1}\mathbf{B} & \mathbf{A}^{n-2}\mathbf{B} & \cdots & \mathbf{B} \end{bmatrix}, \\ \mathbf{S}_C &= \begin{bmatrix} \mathbf{C} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{AC} & \mathbf{C} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}^{n-1}\mathbf{C} & \mathbf{A}^{n-2}\mathbf{C} & \cdots & \mathbf{C} \end{bmatrix}, \end{aligned}$$

$$\begin{aligned} \bar{\mathbf{Q}} &= \text{blockdiag}\{\mathbf{Q}_1, \mathbf{Q}_2, \cdots, \mathbf{Q}_n\}, \text{ and} \\ \bar{\mathbf{P}} &= \text{blockdiag}\{\mathbf{P}_0, \mathbf{P}_1, \cdots, \mathbf{P}_{n-1}\}. \end{aligned}$$

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

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

Here,  $\dagger$  is the Moore–Penrose pseudoinverse operator.

Let

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

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

$$\mathbf{U}^* = \mathbf{H} \mathbf{Y}. \quad (4)$$

Derivation of this equation is described in Appendix A.

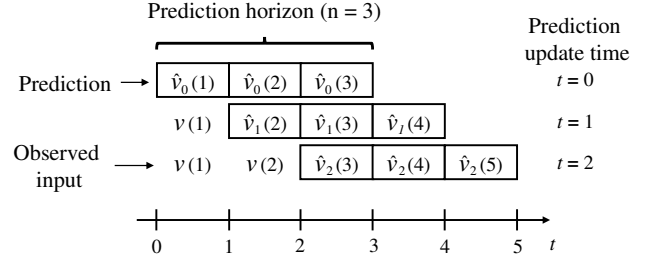


Fig. 1. Prediction updates in FHC. To differentiate between predictions performed at different times, the estimation of  $v(t+h)$  predicted at time-step  $t$  is denoted with  $\hat{v}_t(t+h)$ .

### B. Finite Horizon Control

The goal of an FHC regulator is to dynamically minimize  $J$  over control period  $t \in [0, 1, \dots, N]$ . 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  $\mathbf{U}$ , 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 better control of the system in 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 paper, both methodologies are studied.

### III. CLOSED FORM ANALYSIS OF THE FINAL COST

With the availability of the exogenous inputs  $\mathbf{V}$ , the optimal controllable inputs  $\mathbf{U}^*$  and consequently the optimal (i.e., minimum) cost  $J^* = J(\mathbf{U}^*)$  can be obtained from (4). In real-world applications, the true value of  $\mathbf{V}$  is not known in advance, and has to be estimated with analytical or numeric models. 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 form of deviation of the cost from its optimal, i.e.,

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

#### A. Prediction Error

In a real system, the exogenous inputs  $\mathbf{v}(t)$  are not observed before time  $t$  and must be substituted by  $\hat{\mathbf{v}}(t) = \mathbf{v}(t) + \boldsymbol{\epsilon}(t)$ , where  $\hat{\mathbf{v}}(t)$  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  $t$ ,  $\hat{\mathbf{v}}(t+h)$  is re-estimated (i.e., predicted again using the latest available information) for  $h \in [1, 2, \dots, n]$ . We denote these re-estimations and their prediction error by  $\hat{\mathbf{v}}_t(t+h)$  and  $\boldsymbol{\epsilon}_t(t+h)$  respectively (Fig. 1).

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

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

To denote all prediction errors over all control horizons,

$$\mathbf{E} = [\mathbf{E}_0^T \ \mathbf{E}_1^T \ \dots \ \mathbf{E}_{N-1}^T]^T$$

is defined.

### B. Optimal Input in Presence of Prediction Error

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

$$\mathbf{U} = \Psi \mathbf{Y} + \Phi \mathbf{E} \quad (5)$$

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

$$\Phi_i = \sum_{j=0}^{N-1} \mathbf{M}_\Phi(j) \mathbf{H}_j \Gamma(j, i), \quad (6)$$

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

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

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

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

$n_t$  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  $f \times g$  sub-block 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  $\Psi$  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,  $\Psi$  incorporates the effects of horizon length variation. This is because the goal of (5) is not *control*, 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  $\mathbf{U}^* = \Psi \mathbf{Y}$  be the optimal input. The increase in cost from optimal due to prediction error  $\mathbf{E}$  is given by

$$\Delta J = \mathbf{E}^T \Theta \mathbf{E} + \mathbf{E}^T \Omega \mathbf{Y} \quad (7)$$

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  $h \in [1, 2, \dots, n]$  using available data, i.e.,  $\hat{v}_t(t+h) = p(v(t), v(t-1), v(t-2), \dots)$ .

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 is 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 performance of different models and hyperparameters are compared across a separate set of testing data [13].

Typically, when applying cross-validation, MSE error (i.e.,  $\mathbf{E}^T \mathbf{E}$ ) 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 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 prediction error in a single step.

### A. Cost Matrix Dimensionality Reduction

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

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

In other cases, assuming repeated evaluation on a fixed system, one can pre-compute  $\Theta$  and  $\Omega \mathbf{Y}$ , and calculate  $\Delta J$  efficiently for different values of  $\mathbf{E}$ , the latter coming from different prediction models.

With this assumption, matrix decomposition can also be used to further reduce computation 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 size of the inputs vector, i.e.,  $M = m \sum_i n_i$ .

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

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

$$\Delta J' = \mathbf{E}^T \mathbf{W}' \mathbf{W}'^T \mathbf{E} + \mathbf{E}^T \Omega \mathbf{Y}. \quad (8)$$

Exploiting the symmetric structure of the new cost function and assuming precomputed  $\mathbf{W}'$  and  $\Omega \mathbf{Y}$ , 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  $\text{tr}(\Sigma') \geq \lambda \text{tr}(\Sigma)$ , where  $\text{tr}$  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  $\Delta J$  is compared with MSE for predictor selection. 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 performance results of models selected using  $\Delta J$  and MSE are compared.

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

### A. Pre-ordering 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 pre-order 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 pre-order  $\kappa$  steps ahead, where at each time-step a discount of  $d$  is applied.
- 3) A pre-order 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 over-supplied orders (i.e., discarded perishables).
- 4) Pre-orders are denoted with  $\mathbf{u}(t) = [u_{t+1}(t) \cdots u_{t+\kappa-1}(t) \ u_{t+\kappa}(t)]^T$ , 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 form of the vector  $\mathbf{x}(t) = [x_t(t) \ x_{t+1}(t) \ \cdots \ x_{t+\kappa}(t)]^T$ , where  $x_{t+h}(t)$  is the total of orders expected at time  $t$  to be delivered at time  $t+h$ .

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

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 pre-ordering (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

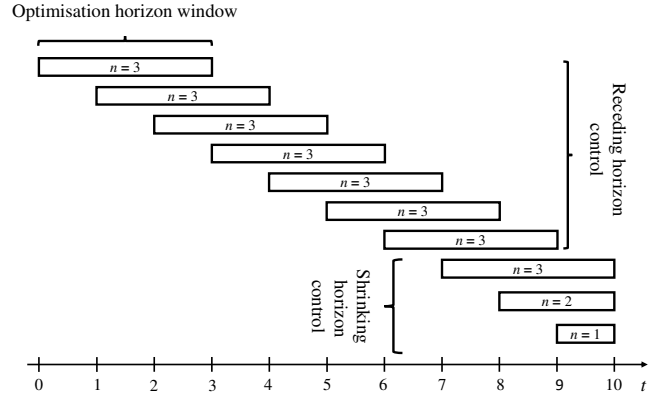


Fig. 2. A hybrid receding-shrinking horizon control scheme, with a maximum horizon length of 3.

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

Eq. (2) is used as the cost function, with the stage cost of  $\mathbf{u}(t)$  (i.e., pre-ordering prices and discounts), and the stage cost of  $\mathbf{x}(t)$  (i.e., the penalty for unmet demand), defined as

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

respectively.

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  $\Theta$  and  $\Omega$  from (7) for the current problem reveals the matrix elements associated with prediction of the current demand, and prediction of steps more than  $\kappa$  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  $\kappa$  steps is not used for ordering in the current horizon. Consequently, choosing a horizon length beyond the number of pre-orders is redundant.

Furthermore,  $\Theta$  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  $\Delta J$  to a weighted MSE.

3) *Simulation and Results:* The simulation was set up to run for 10 time-steps, with pre-ordering allowed for three steps ahead (i.e.,  $N = 10$  and  $\kappa = 3$ ). Based on the analysis of  $\Theta$  for this problem, the control horizon  $n$  was limited to the number of pre-orders  $\kappa$  (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  $h$ th step in the future,  $v(t+h)$ , was predicted using an AR model,

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

where  $q$  is the model order, and  $\phi_i$  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  $q \in [1, 2, \dots, 8]$ .

To simulate real-world demand, time series from the M3 competition dataset [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  $\Delta J$ .

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

- 1) A *random search* was performed to select the best orders using 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  $\Delta J$  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 in-sample MSE through an exhaustive search. The model orders for  $h = 2$  and  $h = 3$  were selected using in-sample  $\Delta J$  through a random search.

The first two methods compare MSE and  $\Delta J$  model selection when the computational resources are limited. The design rationale of the third technique is based on the  $\Theta$  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 dependency between the error at  $h = 1$  and other steps, the  $\Delta J$  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 each approach. For the hybrid approach, this translated to eight evaluations exhaustively searching for the best first model order, and eight additional random evaluations for the second and third model orders.

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  $\Delta J$  are not directly comparable. Consequently, these errors were normalized to the MSE and  $\Delta J$  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  $\Delta J$ , considering the diagonal nature of  $\Theta$ , (7) was also numerically implemented as a weighted MSE. TABLE I compares the run-times of different methods. It is observed that computing  $\Delta J$  using (7), even in its full matrix form, offers a significant speed-up compared

TABLE I  
RUN-TIME AND SPEED-UP COMPARISON FOR THE PRE-ORDERING PROBLEM.

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

TABLE II  
MEAN NORMALIZED PREDICTION ERROR FOR DIFFERENT MODEL SELECTION METHODS IN THE PRE-ORDERING PROBLEM.

| Selection Method                     | In-sample MSE | In-sample $\Delta J$ | Out-of-sample MSE | Cost Performance |
|--------------------------------------|---------------|----------------------|-------------------|------------------|
| Random Search (using MSE)            | <b>0.9078</b> | 0.8953               | <b>0.9261</b>     | 0.9208           |
| Random Search (using $\Delta J$ )    | 0.937         | 0.867                | 0.941             | 0.9109           |
| Hybrid Search (based on $\Delta J$ ) | 0.9081        | <b>0.8533</b>        | 0.9321            | <b>0.9072</b>    |

to a step-by-step FHC simulation. Furthermore, the diagonal only implementation is almost as efficient as an ordinary MSE, achieving a speed-up of more than 2500 $\times$ .

In TABLE II, prediction errors for models selected using MSE and  $\Delta J$  are summarized. While the models selected using MSE offered a better in-sample and out-of-sample MSE, they were outperformed over the cost performance by the models selected using  $\Delta J$ . 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  $\Delta J$ . 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 can not 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, inter-dealer 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 inter-dealer 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 inter-dealer 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 10 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 5th order autoregressive (AR) model,

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

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  $\Delta J$ . The test was repeated 10 times.

Our first concern is the dimensionality reduction for  $\Theta \in \mathbb{R}^{40 \times 40}$ . Despite its simpler formulation compared to the problem of Section V-A,  $\Theta$  is not diagonal, and consequently the eigenvalue method has to be used. The 10 largest eigenvalues hold more than 99.9% of eigenvalue energy, and thus a fourfold order reduction is possible using  $\mathbf{W}' \in \mathbb{R}^{40 \times 10}$ .

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

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

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

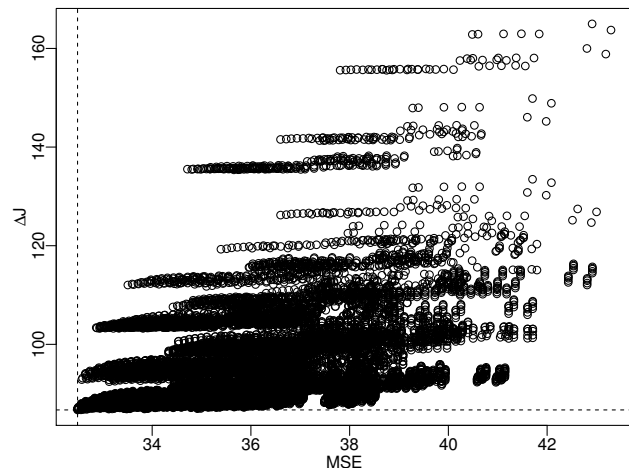


Fig. 3. MSE versus  $\Delta J$  for different model orders in the stocks portfolio problem.

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

An exhaustive search was performed to cross-validate different model orders over the in-sample period using MSE and  $\Delta J$ . 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  $\Delta J$  as a model selection measure has reduced the controller's cost compared to selections using the ordinary MSE measure. While this reduction is not significant for in-sample  $\Delta J$  results, the controller's cost performance improvement, from 98.18 to 94.79, is considerable. Additionally, 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 $\Delta J$ | Out-of-sample MSE | Cost Performance |
|-----------------------|----------------|----------------------|-------------------|------------------|
| MSE                   | <b>32.4888</b> | 86.9027              | <b>36.3447</b>    | 98.1846          |
| $\Delta J$            | 32.4892        | <b>86.7015</b>       | 38.0916           | <b>94.7870</b>   |

## VI. A 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  $\Delta J$  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 multi-parametric 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 piece-wise affine solution. Considering the current affine form of  $H$  and quadratic form of  $\Delta J$ , one would expect a piece-wise quadratic form for the constrained  $\Delta J$ . 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 to repeated runs of quadratic programming in FHC simulation.

Implementing and benchmarking this multi-parametric approach will be subject of a future study.

## VII. CONCLUSION

This paper 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,  $\Delta J$  accurately describes the effects of prediction on the objective cost function. In addition, techniques were presented to reduce computational costs of this measure, in some cases even comparable to that of MSE, and thus making  $\Delta J$  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

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

The cost function can be expanded and rewritten as

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

Using the optimality condition, namely  $\frac{dJ}{d\mathbf{U}} = 0$ , the solution to argmin  $J$  can be derived:

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

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

## APPENDIX B

### DERIVATION OF INPUT IN REPEATED APPLICATION OF THE CONTROL LAW

In finite horizon control, controllable inputs are re-evaluated at each time-step. For example, at time-step  $t$ ,

$$\mathbf{U}_t = \mathbf{H}_t (\mathbf{Y}_t + \mathbf{E}_t) = \mathbf{H}_t \hat{\mathbf{Y}}_t, \quad (11)$$

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

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

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

At any time-step,  $\mathbf{Y}_{t+1}$  can be computed by accumulating the *applied* input controls  $\mathbf{u}_i$  for  $i < t$ , in to the  $\mathbf{Y}_{t+1}$ :

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

Here,  $\mathbf{M}_U$  is used to extract  $\mathbf{u}_i$  by separating the first vector of inputs from  $\mathbf{U}_i$ ; 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  $\mathbf{Y} = \mathbf{S}_C^\dagger \mathbf{S}_A \mathbf{x}(0) + \mathbf{V}$ , to reshape the resulting new states into the new uncontrollable inputs vector.

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

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

which can be expanded recursively and then factored to

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

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

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

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

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

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

$$\mathbf{U} = \Psi \mathbf{Y} + \Phi \mathbf{E}. \quad (13)$$

APPENDIX C  
DERIVATION OF COST DEVIATION

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

$$J(U) = UJ_AU + 2UJ_BY + J_C.$$

Let  $U_e = U - U^* = \Phi E$  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  $U^*$  (i.e.,  $J(U^*)$ ) and the cost with prediction error using  $U$  (i.e.,  $J(U)$ ), is given by

$$\begin{aligned} \Delta J &= J(U) - J(U^*) \\ &= U^T J_A U + 2U J_B Y - U^{*T} J_A U^* - 2U^* J_B Y. \end{aligned}$$

Expanding  $U = U^* + U_e$  results in

$$\begin{aligned} \Delta J &= (U^* + U_e)^T J_A (U^* + U_e) - U^{*T} J_A U^* \\ &\quad + 2(U^* + U_e)^T J_B Y - 2U^* J_B Y \\ &= U_e^T J_A U_e + 2U_e^T J_A U^* + 2U_e^T J_B Y. \end{aligned}$$

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

$$\begin{aligned} \Delta J &= U_e^T J_A U_e + 2U_e^T J_A (\Psi Y) + 2U_e^T J_B Y \\ &= U_e^T J_A U_e + 2U_e^T (J_A \Psi + J_B) Y. \end{aligned}$$

Let  $\Theta = \Phi^T J_A \Phi$  and  $\Omega = 2\Phi^T (J_A \Psi + J_B)$ . Replacing  $U_e = \Phi E$ , the above transforms to

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

REFERENCES

- [1] G. C. Goodwin, M. M. Seron, and J. A. De Doná, *Constrained Control and Estimation: An Optimisation Approach*. Springer, 2006.
- [2] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control*. in preparation, 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," *Elect. 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, 2012.
- [5] G. C. Calafiore and F. Kharaman, "Multi-period asset allocation with lower partial moments criteria and affine policies," in *IEEE Symp. Computational Intelligence for Financial Engineering & Economics (CIFER)*, London, UK, 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, 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, Sept 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 and 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, 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, 1936.
- [15] T. Miller, *Hierarchical Operations and Supply Chain Planning*. 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, 2011.
- [18] S. Makridakis and M. Hibon, "The M3-competition: Results, conclusions and implications," *Int. J. Forecasting*, vol. 16, no. 4, pp. 451–476, 2000.
- [19] A. Bemporad, L. Puglia, and T. Gabbriellini, "A stochastic model predictive control approach to dynamic option hedging with transaction costs," in *American 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 *IEEE Conf. Computational Intelligence for Financial Engineering & Economics (CIFER)*, London, UK, 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, 2016.
- [22] H. Markowitz, "Portfolio selection," *J. Finance*, vol. 7, no. 1, pp. 77–91, March 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.



**Farzad Noorian** (M'16) received the Ph.D. degree in engineering from the School of Electrical and Information Engineering, the University of Sydney, Australia.

He is currently a research associate at the Computer Engineering Lab, the University of Sydney.

His research concentrates on data-driven risk management, and his interests mainly include machine learning and model predictive control, with a focus on their applications in financial engineering and renewable energies.



**Philip H. W. Leong** (SM'02) received the B.Sc., B.E., and Ph.D. degrees from the University of Sydney, Australia.

From 1997–2009 he was with the Chinese University of Hong Kong. He is currently Professor of Computer Systems in the School of Electrical and Information Engineering at the University of Sydney, Visiting Professor at Imperial College, Visiting Professor at Harbin Institute of Technology, and Chief Technology Advisor to Cluster Technology.

Prof Leong was co-founder and program co-chair of the International Conference on Field Programmable Technology (FPT); program co-chair of the International Conference on Field Programmable Logic and Applications (FPL) and Senior Associate Editor for ACM Transactions on Reconfigurable Technology and Systems (TRETs).