Subspace-based Model Predictive Control with Data Prefiltering

Noor A. Mardi * **Liuping Wang** *

* *School of Electrical and Computer Engineering, Royal Melbourne Institute of Technology (RMIT) University, Melbourne, Victoria 3000 Australia (e-mail: n.mardi@student.rmit.edu.au)*

Abstract: Subspace-based model-free predictive control algorithms directly estimate the relevant components of a predictive controller. Due to disturbances and noise in the measured data, the estimation results were often poor, which limited the applications of subspace-based model-free predictive controllers. By assuming a priori knowledge of the disturbance characteristics, this paper proposes a subspace-based model-free predictive control algorithm that utilizes the noise model for the estimation of the predictive control gain matrices. Simulation results show improved control results.

Keywords: Subspace; Model Predictive Control; Filtering

1. INTRODUCTION

Traditionally, the strategy employed in an attempt to control dynamical systems is a two step process. It starts with the identification of the system that is to be controlled, followed by the development of the desired control system. Only after obtaining the suitable system model can the control strategy be developed, according to the requirements and limitations present.

However, an innovative technique of combining the two steps of system identification and control into a single-step implementation has been introduced, for example in Favoreel et al. [1999, 2000]. The essence of this approach is based on that using input and output data from a plant, the appropriate control response will be determined directly to drive the output to the desired values, without the need to know the exact plant model.

In order to achieve this, some of the ideas and tools that arise from subspace-based system identification (SSID) theory was utilized. Subspace-based method can be seen as a combination of system theory, linear numerical algebra and geometry as described by Favoreel et al. [2000]. Its main attraction is that subspace technique is known for its adaptability to systems with many system variables as stated in Barry [2004]. Furthermore, since no nonlinear computation is performed, in addition the fact that canonical parameterization is not used, subspace method is found to be numerically reliable. Moreover, the computational complication of this technique is quite reasonable compared to other prediction error techniques shown in Viberg [1995]. The algorithms in subspace identification method include "Numerical algorithms for Subspace State Space System IDentification" (N4SID) by Overschee and Moor [1994a], and "Multivariable Output-Error State space" (MOESP) by Verhaegen and Dewilde [1992]. Although some of these algorithms use different weighting matrices for its spaces, they are found to be quite similar in principle, see Overschee and Moor [1994b].

Model Predictive Control (MPC) is usually defined as a category of control systems that determine the control trajectory that will result in an optimized future behavior of a plant, see Wang [2003]. The determination of the control trajectory is usually done within a fixed time-frame and the whole optimization procedure can be subjected to constraints in the input or output variables, see Maciejowski [2002] for a review of MPC.

Efforts to implement subspace-based identification ideas in predictive control applications have been investigated before, for example in Favoreel and Moor [1998], Ruscio [1997], Wang et al. [2007], Barry and Wang [2004]. Another example is in Favoreel et al. [1999], where the typically three-step required in classical linear quadratic Gaussian controller design (system identification, state estimation and LQ-controller determination) was able to be reduced into a 1-step QR and SV-decomposition. The fundamental principle that allows the combination of subspace-based tools into MPC is the derivation of a linear predictor, for instance in papers by Favoreel and Moor [1998], Barry [2004], Kadali et al. [2003].

However, one drawback of using subspace-based identification tools in the implementation of model-free predictive control is its sensitivity to disturbance. Therefore in this paper, with the a priori knowledge of the noise model, a prefiltering step in the control process is attempted. With this prefiltering in place, the cumulative effect of disturbance in process control can be overcome.

2. SUBSPACE-BASED MODEL PREDICTIVE CONTROL WITH DATA PREFILTERING OF KNOWN DISTURBANCE MODEL

2.1 Problem Definition

A linear, time-invariant system described in discrete state-space form is defined as follows:

$$x_{k+1} = Ax_k + Bu_k$$
$$y_k = Cx_k + Du_k + d_k \tag{1}$$

where $u_k \in \mathbb{R}^m$ is the control input vector, $y_k \in \mathbb{R}^l$ is the output vector, $x_k \in \mathbb{R}^n$ is the internal state vector and $d_k \in \mathbb{R}^l$ is the contribution from the external disturbances. The matrices $A \in \mathbb{R}^{n\times n}$ represents the state matrix, $B \in \mathbb{R}^{n\times m}$ is the input matrix, $C \in \mathbb{R}^{l\times n}$ is the output matrix while $D \in \mathbb{R}^{l\times m}$ is the feedthrough matrix. We will further model the external disturbance, d_k as a first-order difference model as described in the equation,

$$d_k = \frac{1-\alpha z^{-1}}{1-z^{-1}} e = Fe \tag{2}$$

where e is zero-mean white Gaussian noise signal, z^{-1} is the backshift operator and F is the disturbance model, $F = \frac{1-\alpha z^{-1}}{1-z^{-1}}$, with α as the parameter that defines the numerator of the disturbance model.

With the definition of the system model, we can then briefly review the ideas of Model-based Predictive Control (MPC). MPC algorithm works in such a way that it tries to find the control input that will drive the output of the system to the desired values by minimizing a given cost function. The cost function can consist of the input and output variables, as well as 'soft' variables such as monetary costs, labor charges etc. However, in this application of MPC, the terms in the cost function depend only on u and y, with no crossed terms. Therefore, define a typical cost function in MPC as

$$J = \sum_{k=1}^{N_p} \left(\widehat{y}_k - r_k\right)^T Q \left(\widehat{y}_k - r_k\right) + \sum_{k=1}^{N_c} \left(u_k^T R u_k\right) \tag{3}$$

with $\widehat{y}_k$ is the predicted outputs at instance k, r_k is the desired setpoint values and u_k is the control input variables. N_p is defined as the prediction horizon, which determines how far ahead the cost function sees the contribution of the predicted outputs. N_c on the other hand is called the control horizon, which limits the number of control action steps after the current time step k. The control horizon limits the amount of steps the controller can act after the current time step k. Finally, Q and R are user-defined positive definitive or semi-definite weighting matrices that set the weights of the outputs and inputs to the overall cost function.

We introduce the problem statement as follows. With the definition of our system model in (1), with external disturbance defined in (2) where α value is known beforehand, it is assumed that the input data u_k and output data y_k for $k \in \{k = -N+1, -N+2, \ldots, -1, 0\}$ is available, with N represents the size of the data. Therefore, using the available data, Subspace-based Model Predictive Control (SMPC) algorithm will determine the most optimum control inputs u_k that will give the most minimum cost defined in (3), with the awareness of the effects of the non-stationary disturbance on the output of the system.

2.2 Subspace-based Linear Predictor

The SMPC algorithm was developed from the combination of theories from Subspace-based system identification, signal processing and MPC. As the result from the 'contamination' of the input and output data with the non-stationary disturbance, it is evident that the effects of the disturbance on the data must first be neutralized before further control implementation is formulated. After that, an expression of the predicted output values has to be developed from the available data. Finally, with the predicted output values at disposal, we can then optimize the control input that will give the minimum cost defined in the cost function formulation.

Firstly, in order to overcome the effects of the non-stationary disturbance in the dataset, the input and output data is filtered with the inverse disturbance model as follows,

$$\bar{y} = F^{-1}y = \frac{1-z^{-1}}{1-\alpha z^{-1}} y, \quad \bar{u} = F^{-1}u = \frac{1-z^{-1}}{1-\alpha z^{-1}} u \tag{4}$$

Note the bar (e.g. $\bar{y}$) denotes filtered data. As a result of filtering, we now have a modified state-space description of the system. Similar to the definition in (1), a new description is obtained as below. As with previous model description, the input and output data u_k and y_k for $k \in \{-N+1, -N+2, \ldots, -1, 0\}$ are assumed available.

$$\bar{x}_{k+1} = \bar{A}\bar{x}_k + \bar{B}u_k \tag{5}$$
$$\bar{y}_k = \bar{C}\bar{x}_k + \bar{D}\bar{u}_k$$

Typical in SSID, define block Hankel data matrices for filtered past and future input data as shown below.

$$\bar{U}_p \triangleq \begin{pmatrix} \bar{u}_{-N+1} & \bar{u}_{-N+2} & \ldots & \bar{u}_{-2M+1} \\ \bar{u}_{-N+2} & \bar{u}_{-N+3} & \ldots & \bar{u}_{-2M+2} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{u}_{-N+M} & \bar{u}_{-N+M+1} & \ldots & \bar{u}_{-M} \end{pmatrix} \tag{6}$$

$$\bar{U}_f \triangleq \begin{pmatrix} \bar{u}_{-N+M+1} & \bar{u}_{-N+M+2} & \ldots & \bar{u}_{-M+1} \\ \bar{u}_{-N+M+2} & \bar{u}_{-N+M+3} & \ldots & \bar{u}_{-M+2} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{u}_{-N+2M} & \bar{u}_{-N+2M+1} & \ldots & \bar{u}_0 \end{pmatrix} \tag{7}$$

and the filtered past and future output values are defined in a similar way. Note that the subscript p denotes 'past' data and subscript f denotes 'future' data. The past and future states block matrices are then given by,

$$\bar{X}_p \triangleq [\bar{x}_{-N+1} \;\; \bar{x}_{-N+2} \;\; \bar{x}_{-N+3} \;\; \ldots \;\; \bar{x}_{-2M+1}]$$
$$\bar{X}_f \triangleq [\bar{x}_{-N+M+1} \;\; \bar{x}_{-N+M+2} \;\; \bar{x}_{-N+M+3} \;\; \ldots \;\; \bar{x}_{-M+1}] \tag{8}$$

With these definitions of 'past' and 'future' data matrices, an equation that relates the future filtered output values $\bar{Y}_f$ with future filtered input $\bar{U}_f$ and past data $\bar{W}_p$ can be written as,

$$\bar{Y}_f = \Gamma_M L_p \bar{W}_p + H_M \bar{U}_f \tag{9}$$

with

$$L_p = \begin{pmatrix} \bar{A}^M \Gamma_M^\dagger \\ \Delta_M - \bar{A}^M \Gamma_M^\dagger H_M \end{pmatrix}^T, \bar{W}_p = \begin{pmatrix} \bar{Y}_p \\ \bar{U}_p \end{pmatrix} \tag{10}$$

$$\Gamma_M \triangleq \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{M-1} \end{pmatrix} \tag{11}$$

$$H_M \triangleq \begin{pmatrix} D & 0 & 0 & \ldots & 0 \\ CB & D & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{M-2}B & CA^{M-3} & CA^{M-4} & \ldots & D \end{pmatrix} \tag{12}$$

$$\Delta_M \triangleq \left(A^{M-1}B \ A^{M-2}B \ \ldots \ AB \ B \right) \in \mathbb{R}^{n\times Mm} \quad (13)$$

Γ_M is the extended observability matrix and H_M is the lower triangular Toeplitz matrix that holds the impulse response coefficients of the deterministic input. Δ_M on the other hand is the reversed extended controllability matrix. Note that $(.)^\dagger$ signifies the Moore-Penrose pseudoinverse of (.).

A least-squares estimate of $\bar{Y}_f$ can be derived by writing $\bar{Y}_f$ as a linear combination of 'past' data $\bar{W}_p$ and 'future' data $\bar{Y}_f$ and $\bar{U}_f$ as shown below.

$$\bar{Y}_f = \left(L_w \ L_u \right) \begin{pmatrix} \bar{W}_p \\ \bar{U}_f \end{pmatrix} \quad (14)$$

Therefore, the estimate of $\bar{Y}_f$ is the least-squares solution for the equation above, by finding the parameters L_w and L_u that will give the least sum of squared error for the linear equation (14) above. This is expressed as,

$$\min_{Lw,Lu} \left\| \bar{Y}_f - \left(L_w \ L_u \right) \begin{pmatrix} \bar{W}_p \\ \bar{U}_f \end{pmatrix} \right\|_F^2 \quad (15)$$

An approximate solution to equation (15) is found by Favoreel and Moor [1998] to be the orthogonal projection of row spaces of $\bar{Y}_f$ into the row spaces of the composite matrix $\begin{pmatrix} W_p \\ U_f \end{pmatrix}$ and denoted by the expression

$$\hat{\bar{Y}}_f = \bar{Y}_f / \begin{pmatrix} \bar{W}_p \\ \bar{U}_f \end{pmatrix} \quad (16)$$

with A/B denotes the projection of row spaces of A into the row space of B, which equals to $AB^\dagger B$. To compute this orthogonal projection, an RQ-decomposition is performed on the composite matrix $\left[\bar{W}_p \ \bar{U}_f \ \bar{Y}_f \right]^T$ to give us a lower-triangular matrix R and an orthogonal matrix Q as shown below, cf. Favoreel and Moor [1998].

$$\begin{bmatrix} \bar{W}_p \\ \bar{U}_f \\ \bar{Y}_f \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 \\ R_{21} & R_{22} & 0 \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \\ Q_3^T \end{bmatrix}$$

Thus, from equation (16), we will arrive at,

$$\begin{aligned} \hat{\bar{Y}}_f &= \bar{Y}_f \begin{pmatrix} \bar{W}_p \\ \bar{U}_f \end{pmatrix}^\dagger \begin{pmatrix} \bar{W}_p \\ \bar{U}_f \end{pmatrix} \\ &= \left[R_{31} \ R_{32} \right] \begin{bmatrix} R_{11} & 0 \\ R_{21} & R_{22} \end{bmatrix}^\dagger \begin{pmatrix} \bar{W}_p \\ \bar{U}_f \end{pmatrix} \end{aligned}$$

By defining a matrix L as below, we can then separate L into two subspace-based linear predictor coefficients L_w and L_u as follows.

$$L = \left[R_{31} \ R_{32} \right] \begin{bmatrix} R_{11} & 0 \\ R_{21} & R_{22} \end{bmatrix}^\dagger$$

with,

$$\begin{aligned} L = \left[L_w \ L_u \right] \text{ and } L_w \in \mathbb{R}^{M(2m+l)\times M(m+l)}&, \\ L_u \in \mathbb{R}^{M(2m+l)\times Mm}& \end{aligned} \quad (17)$$

Therefore, we can write equation (9) as

$$\hat{\bar{Y}}_f = L_w \bar{W}_p + L_u \bar{U}_f \quad (18)$$

In this section, we have arrived at a formulation of the 'future' filtered output values in therms of the system's 'past' filtered input and output data embodied in $\bar{W}_p$, and in the filtered 'future' input data. L_w and L_u are known in the literature as the subspace-based linear predictor coefficients, and equation (18) is called the linear predictor equation.

2.3 Unconstrained SMPC

The core idea in the implementation of MPC is finding the control input signals that will give the minimum value for an objective function. In this paper, consider the following cost function, J.

$$J = \sum_{k=1}^{M} \left((\hat{y}_k - r_k)^T Q (\hat{y}_k - r_k) + (\bar{u}_k^T R \bar{u}_k) \right) \quad (19)$$

Notice that the input signal that is being optimized $\bar{u}_k$ is the filtered input signal defined in equation (4). For this implementation of MPC, N_p and N_c from equation 3 are set to be equal to the order of the data Hankel matrices, M, thus giving the cost equation above.

To use the result from section (2.2), we need an equation that relates the unfiltered predicted y-values, $\hat{y}_k \in \{k = 1, 2, ..., M\}$ in equation (19) above to $\bar{y}_f$ which is the 'future' filtered y-data. Thus, referring to definition of filtered y-data in (4), we can write the following relation in matrix form (with the assumption $\bar{y}_0 = 0$), bearing in mind of the number of output variables manifested in the sizes of column vector $\hat{y}_k$ and $\bar{y}_k$:

$$\underbrace{\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_M \end{bmatrix}}_{\hat{y}_f} = \underbrace{\begin{bmatrix} I_l & 0 & 0 & \ldots & 0 \\ I_{1-\alpha}^l & I_l & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I_{1-\alpha}^l & I_{1-\alpha}^l & I_{1-\alpha}^l & \ldots & I_l \end{bmatrix}}_{\Gamma_l} \underbrace{\begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \\ \vdots \\ \bar{y}_M \end{bmatrix}}_{\bar{y}_f} + \underbrace{\begin{bmatrix} I_l \\ I_l \\ \vdots \\ I_l \end{bmatrix}}_{F_l} y_0 \quad (20)$$

with,

$$I_l = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \in \mathbb{R}^{l\times l} \quad (21)$$

$$I_{1-\alpha}^l = \begin{bmatrix} 1-\alpha & 0 & \cdots & 0 \\ 0 & 1-\alpha & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1-\alpha \end{bmatrix} \in \mathbb{R}^{l\times l} \quad (22)$$

$$\hat{y}_f = \Gamma_l \ \bar{y}_f + F_l y_0 \quad (23)$$

and y_0 is the value of y at the current time. Using the derivation of equation (23), we can then update the cost function in (19) to become equation (26) as follows:

$$\hat{y}_f = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_M]^T \qquad \bar{y}_f = [\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_M]^T \quad (24)$$

$$\bar{u}_f = [\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_M]^T \qquad r_f = [r_1, r_2, \ldots, r_M]^T \quad (25)$$

$$\begin{aligned} J &= (\hat{y}_f - r_f)^T \, Q \, (\hat{y}_f - r_f) + \bar{u}_f^T R \bar{u}_f \\ &= (\Gamma_l \, \bar{y}_f + F_l y_0 - r_f)^T Q (\Gamma_l \, \bar{y}_f + F_l y_0 - r_f) \\ &\quad + \bar{u}_f^T R \bar{u}_f \end{aligned} \quad (26)$$

Before applying the linear predictor equation in (18) to the cost function (26) above, it can be seen that the filtered input and output values Hankel matrices, that is $\bar{Y}_f$, $\bar{W}_p$ and $\bar{U}_f$, contain redundant information. This is because only the first column of these matrices are needed in our updated cost function. Therefore, by retaining the first columns of $\bar{Y}_f$, $\bar{W}_p$ and $\bar{U}_f$ and adjusting their indices, we can get a streamlined version of the subspace-based linear predictor, without needing to adjust the linear predictor coefficients L_w and L_u. This streamlined linear predictor will give the predicted filtered y values up to M steps ahead in the MPC implementation, based on M past filtered data and M future filtered input. The new subspace-based linear predictor is given by the equation,

$$\bar{y}_f = L_w\bar{w}_p + L_u\bar{u}_f \tag{27}$$

Substituting $\bar{y}_f$ in (26) by (27), removing unrelated parameters and writing the resulting equation in quadratic form will then give,

$$J = \frac{1}{2}\bar{u}_f^T H\bar{u}_f + \bar{u}_f^T f \tag{28}$$

with

$$\begin{aligned} H &= (\Gamma_l L_u)^T Q(\Gamma_l L_u) + R \\ f &= (\Gamma_l L_u)^T Q(\Gamma_l L_w\bar{w}_p + F_l y_0 - r_f) \end{aligned}$$

Thus using the equation (28) above, a solution for the predictive control implementation can be found by setting the derivative of the cost function, J, with respect to the filtered future input values $\bar{u}_f$ to zero and solving for $\bar{u}_f$ giving us,

$$\begin{aligned} \bar{u}_f &= \left(R + (\Gamma_l L_u)^T Q(\Gamma_l L_u)\right)^{-1} \\ &\times(\Gamma_l L_u)^T Q(r_f - \Gamma_l L_w\bar{w}_p - F_l y_0) \end{aligned} \tag{29}$$

With the equation above, the filtered future input values $\bar{u}_f$ can be solved and the appropriate unfiltered future input values u_f are then calculated using the formulation $u = F\bar{u} = \frac{1-\alpha z^{-1}}{1-z^{-1}}\bar{u}$. After u_f has been calculated, only the first value which corresponds to u_1 is fed to the system that is to be controlled. Subsequently during the next time step, the variables $\bar{w}_p$, r_f and y_0 are updated before the next input can be calculated. This routine is repeated until the desired behavior in the system is obtained.

2.4 Constrained SMPC

In real world situations there are physical limits on the control variable, as well as the output variables. Therefore, in this subsection we will introduce constraints in the control routine by implementing a quadratic programming procedure to the cost function in (28).

We will start with the introduction of the constraints as follows. Let the unfiltered output signals y, unfiltered inputs u and their rates of change, Δy and Δu subjected to minimum and maximum limits. Writing the constraints for M-future input-output variables will then give,

$$F_l y_{min} \le y_f \le F_l y_{max} \tag{30a}$$
$$F_l\Delta y_{min} \le \Delta y_f \le F_l\Delta y_{max} \tag{30b}$$
$$F_m u_{min} \le u_f \le F_m u_{max} \tag{30c}$$
$$F_m\Delta u_{min} \le \Delta u_f \le F_m\Delta u_{max} \tag{30d}$$

with

$$\begin{aligned} F_m &= [\, I_m\ I_m\ \ldots\ I_m \,]^T \\ I_m &= \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \in \mathbb{R}^{m\times m} \\ y_f &= [y_1\ y_2\ \cdots\ y_M]^T \\ u_f &= [u_1\ u_2\ \cdots\ u_M]^T \\ \Delta y_f &= [\, 0\ y_1 - y_2\ \ldots\ y_M - y_{M-1} \,]^T \\ \Delta u_f &= [\, 0\ u_1 - u_2\ \ldots\ u_M - u_{M-1} \,]^T \end{aligned} \tag{31}$$

To implement the optimization of the cost function, we first need to express the constraints in terms of $\bar{u}_f$. Therefore, substitute y_f with equation (23) and (27) to obtain (by taking $y_f = \hat{y}_f$),

$$-\Gamma_l L_u\bar{u}_f \le F_l y_0 + \Gamma_l L_w\bar{w}_p - F_l y_{min} \tag{32}$$

Similarly for the second constraint $y_f \le F_l y_{max}$, we will arrive at

$$\Gamma_l L_u\bar{u}_f \le F_l y_{max} - F_l y_0 - \Gamma_l L_w\bar{w}_p \tag{33}$$

For the next set of constraints in (30b), we need to formulate these in terms of $\bar{u}_f$. Thus from the definition of filtered y values in (4), we can derive the following relation between the rate of change of y and the filtered 'future' y values. With $\bar{y} = \frac{1-z^{-1}}{1-\alpha z^{-1}}y$, and with the assumption $\bar{y}_0 = 0$ we can write,

$$\underbrace{\begin{bmatrix} \Delta y_1 \\ \Delta y_2 \\ \Delta y_3 \\ \vdots \\ \Delta y_M \end{bmatrix}}_{\Delta y_f} = \underbrace{\begin{bmatrix} I_l & 0 & 0 & 0 & \cdots & 0 \\ I_{-\alpha} & I_l & 0 & 0 & \cdots & 0 \\ 0 & I_{-\alpha} & I_l & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & I_{-\alpha} & I_l \end{bmatrix}}_{\Psi_l} \underbrace{\begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \\ \bar{y}_3 \\ \vdots \\ \bar{y}_M \end{bmatrix}}_{\bar{y}_f}$$

$$\Delta y_f = \Psi_l\,\bar{y}_f \tag{34}$$

with

$$I_{-\alpha} = \begin{bmatrix} -\alpha & 0 & \cdots & 0 \\ 0 & -\alpha & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -\alpha \end{bmatrix} \in \mathbb{R}^{l\times l}$$

Using equation (34) and (27), we can write the constraints in (30b) as follows.

$$-\Psi_l\, L_u\bar{u}_f \le \Psi_l\, L_w\bar{w}_p - F_l\Delta y_{min} \tag{35}$$
$$\Psi_l\, L_u\bar{u}_f \le F_l\Delta y_{max} - \Psi_l\, L_w\bar{w}_p \tag{36}$$

Similar to the derivations in (23) and (34), relations between Δu_f, $\bar{y}_f$, u_f and $\bar{u}_f$ can be derived easily. Variables Γ_m, Ψ_m and F_m are constructed in a similar fashion as its counterparts Γ_l, Ψ_l and F_l. Summarizing all the constraints, we arrive at the following inequality:

$$\underbrace{\begin{bmatrix} \Gamma_l L_u \\ -\Gamma_l L_u \\ \Psi_l\, L_u \\ -\Psi_l\, L_u \\ \Gamma_m \\ -\Gamma_m \\ \Psi_m \\ -\Psi_m \end{bmatrix}}_{A_{QP}} \bar{u}_f \le \underbrace{\begin{bmatrix} F_l y_{max} - \Gamma_l L_w\bar{w}_p - F_l y_0 \\ \Gamma_l L_w\bar{w}_p + F_l y_0 - F_l y_{min} \\ F_l\Delta y_{max} - \Psi_l\, L_w\bar{w}_p \\ \Psi_l\, L_w\bar{w}_p - F_l\Delta y_{min} \\ F_m u_{max} - F_m u_0 \\ F_m u_0 - F_m u_{min} \\ F_m\Delta u_{max} \\ -F_m\Delta u_{min} \end{bmatrix}}_{b_{QP}} \tag{37}$$

Thus, we have arrived to a quadratic programming problem of equation (28), subject to constraints in (37). In order to solve this, we use a quadratic programming function solver to arrive at the most optimum $\bar{u}_f$. After the optimum $\bar{u}_f$ is calculated, it is then converted into the unfiltered future input values, u_f using the definition in (4). Only the value at the next instance (i.e. at u_1) is fed to the system. At the next iteration, $\bar{w}_p$, y_0, and u_0 in equations (28) and (37) are updated, before the next value of control input is calculated. This step is repeated until the appropriate reference output values r_f are achieved.

2.5 Laguerre function parameterization

In order to reduce the number of variables passed into the quadratic programming algorithm, we parameterize $\bar{u}_f$ in the form of Laguerre orthonormal basis functions. By using the Laguerre orthonormal functions, $\bar{u}_f$ can be equated as

$$\bar{u}_f = c_1 l_1(k) + c_2 l_2(k) + \ldots + c_i l_i(k) \tag{38}$$

with c_i as the coefficients for the Laguerre orthonormal function set, and $l_1, l_2, ..., l_i$ is the set of Laguerre functions. In order to construct the discrete Laguerre function, we use a difference equation used in Wang [2004], and is given by the following.

$$L(k+1) = \Omega L(k) \tag{39}$$

where

$$L(k) = [\, l_1(k)\ l_2(k)\ \ldots\ l_{N_L}(k)\,]^T \tag{40}$$

$$\Omega = \begin{bmatrix} a & 0 & 0 & \ldots & 0 \\ a^2-1 & a & 0 & \ldots & 0 \\ a(a^2-1) & a^2-1 & a & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a^{N_L-2}(a^2-1) & a^{N_L-3}(a^2-1) & a^{N_L-4}(a^2-1) & \ldots & a \end{bmatrix} \tag{41}$$

with the inital condition,

$$L(0) = \sqrt{1-a^2}[\, 1\ a\ a^2\ a^3\ \ldots\ a^{N_L-1}\,]^T \tag{42}$$

Note that N_L here is the order of the Laguerre functions used in the parameterization, and a is the scaling factor with the condition $|a| < 1$.

Therefore, parameterizing $\bar{u}_f$ in terms of Laguerre function will give us

$$\underbrace{\begin{bmatrix} \bar{u}_1 \\ \bar{u}_2 \\ \vdots \\ \bar{u}_M \end{bmatrix}}_{\bar{u}_f} = \underbrace{\begin{bmatrix} l_1(0) & l_2(0) & \ldots & l_{N_L}(0) \\ l_1(1) & l_2(1) & \ldots & l_{N_L}(1) \\ \vdots & \vdots & \ddots & \vdots \\ l_1(M) & l_2(M) & \ldots & l_{N_L}(M) \end{bmatrix}}_{\Phi} \underbrace{\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_L \end{bmatrix}}_{\eta} \tag{43}$$

Thus the cost function in (28) can now be rewritten as

$$J = \frac{1}{2}(\Phi\eta)^T H(\Phi\eta) + (\Phi\eta)^T f \tag{44}$$

$$= \frac{1}{2}\eta^T H_n \eta + \eta^T f_n \tag{45}$$

where

$$H_n = (\Gamma_l L_u \Phi)^T Q (\Gamma_l L_u \Phi) + \Phi^T R \Phi \tag{46}$$

$$f_n = (\Gamma_l L_u \Phi)^T Q (\Gamma_l L_w \bar{w}_p + F_l y_0 - r_f) \tag{47}$$

with the updated constraints

$$\underbrace{\begin{bmatrix} \Gamma_l L_u \Phi \\ -\Gamma_l L_u \Phi \\ \Psi_l\, L_u \Phi \\ -\Psi_l\, L_u \Phi \\ \Gamma_m \Phi \\ -\Gamma_m \Phi \\ \Psi_m\, \Phi \\ -\Psi_m\, \Phi \end{bmatrix}}_{A_{QP_L}} \eta \leq \underbrace{\begin{bmatrix} F_l y_{max} - \Gamma_l L_w \bar{w}_p - F_l y_0 \\ \Gamma_l L_w \bar{w}_p + F_l y_0 - F_l y_{min} \\ F_l \Delta y_{max} - \Psi_l\, L_w \bar{w}_p \\ \Psi_l\, L_w \bar{w}_p - F_l \Delta y_{min} \\ F_m u_{max} - F_m u_0 \\ F_m u_0 - F_m u_{min} \\ F_m \Delta u_{max} \\ -F_m \Delta u_{min} \end{bmatrix}}_{b_{QP_L}} \tag{48}$$

By solving the quadratic cost function (44), subject to the inequality constraints in (48) using a quadratic programming algorithm, we will arrive to an optimized η vector. After the optimization of η is complete, the corresponding value of the filtered inputs $\bar{u}_f$ can be calculated from equation (43). The unfiltered input is then calculated from equation (4) and the first value of the unfiltered input u_1 is fed into the system. After that, similar to the unconstrained solution, the corresponding outputs are recorded and the variables $\bar{w}_p$, y_0, and u_0 in the cost function and inequality constraints are updated accordingly. The optimization is the repeated until the desired setpoint value has been reached.

3. IMPLEMENTATION

The proposed filtered unconstrained SMPC algorithm was tested on a real 2-input 2-output servo motor system, where its inputs are the voltage applied to the servos while the outputs are the position of its respective motors. A sampling period of 0.02s was used throughout the experiment. Figure (1) below shows the identification data that was corrupted with non-stationary disturbances $d1$ and $d2$, with $y1$ corrupted by $d1$ while $y2$ corrupted by $d2$. Both $d1$ and $d2$ have α values of 0.9, but are different because of the different seeds for the white noise. The signal to noise ratio for y1 was 9.60 dB, while for y2 was -4.74 dB, indicating the influence of the disturbance especially for y2.

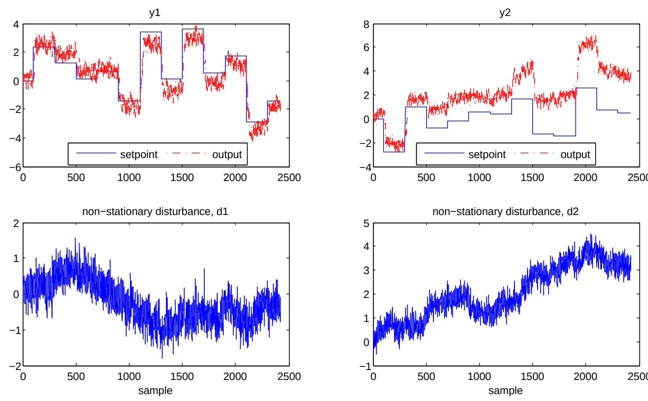

Fig. 1. Identification data

Using the identification data shown in Figure (1), subspace-based linear predictor coefficients L_w and L_u were calculated using the filtered data as shown in subsection (2.2). The unconstrained SMPC algorithm was then applied to the system using the steps in subsection (2.3). In order to compare the performance of the filtering algorithm developed in this paper, an unfiltered implementation of subspace-based MPC (for example in Favoreel and Moor

[1998] ,Barry [2004]) was also tested on the system, using the same identification data.

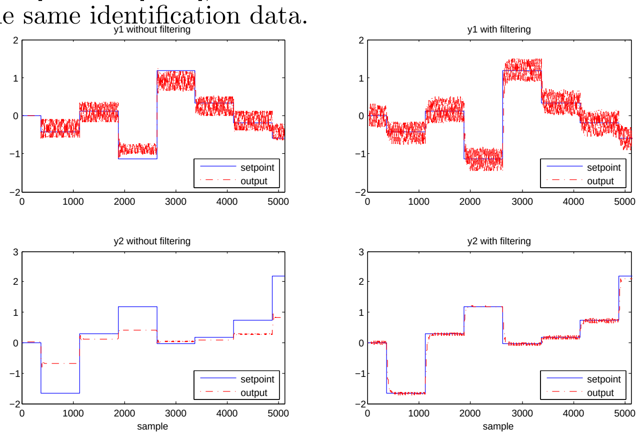

Fig. 2. Comparison of outputs with and without filtering

Method	Sum of Squared Error (SSE)
Unfiltered MfPC	2114.4084
Filtered MfPC	536.0936

Table 1: Sum of squared error

Figure (2) shows the result from the control implementation of the motor system. In this implementation, it is assumed that no external disturbance enters the system. The graphs on the left side of Figure (2) are the outputs $y1$ and $y2$ without filtering action, while the graphs on the right shows the result from the use of the proposed data-prefiltering SMPC. Figure (3) shows the input signals u_1 and u_2 that were applied during the experiment, for both filtered and unfiltered implementations. From the graphs, we can see that the filtered control strategy shows a better control performance compared to without filtering. This is confirmed by the sum of squared error values for both variables y_1 and y_2 as shown in Table 1. Therefore, we can conclude that the filtering action has reduced the effect of the non-stationary disturbance in the derivation of the linear predictor coefficients, thus improving the offset error of the control. Besides that, integrator is naturally embedded in this implementation of predictive control system.

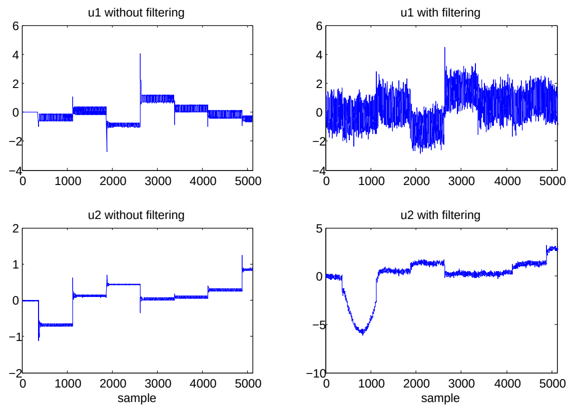

Fig. 3. Control input signals applied with and without filtering

4. CONCLUSIONS

In this paper, a comparison of subspace-based model free predictive control with and without filtering of the data has been shown. It can be seen that without the filtering of data in the implementation of the control action, the effects of the non-stationary disturbance entering the system cannot be handled with satisfactory results. Therefore, with the knowledge of a priori non-stationary disturbance model, the effects of the disturbance can be overcome thus improving on the control implementation of SMPC. Additionally, an integrator was naturally embedded into the system.

REFERENCES

T. Barry. A data driven approach to constrained control. Master's thesis, School of Electrical and Computer Engineering, RMIT University, 2004.

T. Barry and L. Wang. A model-free predictive controller with laguerre polynomials. In *5th Asian Control Conference*, volume 1, pages 177–184, 2004.

W. Favoreel and B. De Moor. SPC: Subspace predictive control. Technical report, Katholieke Universiteit, Leuven, 1998.

W. Favoreel, B. De Moor, P. Van Overschee, and M. Gevers. Model-free subspace-based LQG-design. In *Proc. of the American Control Conference*, pages 3372–3376, San Diago, California, 1999.

W. Favoreel, B. De Moor, and P. Van Overschee. Subspace state-space system identification for industrial processes. *Journal of Process Control*, 10:149–155, 2000.

R. Kadali, B. Huang, and A. Rossiter. A data driven subspace approach to predictive controller design. *Control engineering practice*, 11:261–278, 2003.

J. M. Maciejowski. *Predictive Control With Constraints*. Prentice Hall, 2002.

P. Van Overschee and B. De Moor. N4SID: Subspace algorithms for the identification of combined deterministic and stochastic systems. *Automatica*, 30(1):75–93, 1994a.

P. Van Overschee and B. De Moor. A unifying theorem for three subspace system identification algorithms. In *Proc. of the American Control Conference*, volume 2, pages 1645–1649, Baltimore, Maryland, 1994b.

D.D. Ruscio. Model based predictive control: An extended state space approach. In *Proceedings of the 36th Conference on Decision & Control*, pages 3210–3217, San Diego, USA, 1997.

M. Verhaegen and P. Dewilde. Subspace identification part I: The output-error state space model identification class of algorithms. *International Journal of Control*, 56: 1187–1210, 1992.

M. Viberg. Subspace-based methods for the identification of linear time-invariant systems. *Automatica*, 31(12): 1835–1851, 1995.

L. Wang. A tutorial on model predictive control using a linear velocity form model. School of Electrical and Computer Engineering, RMIT University, Australia, 2003.

L. Wang. Discrete model predictive controller design using laguerre functions. *Journal of Process Control*, 14:131–142, 2004.

X. Wang, B. Huang, and T. Chen. Data-driven predictive control for solid oxide fuel cells. *Journal of Process Control*, 17:103–114, 2007.