Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Closed-loop Subspace Predictive Control for Fault Tolerant MPC Design

Jianfei Dong * **Michel Verhaegen** * **Edward Holweg** **

* *DCSC, TU-Delft, Mekelweg 2, 2628 CD Delft, The Netherlands. (e-mail: j.dong@tudelft.nl, m.verhaegen@moesp.org).*
** *Product & Systems Development, SKF Automotive Division, 3430 DT Nieuwegein, The Netherlands. (e-mail: edward.holweg@skf.com).*

Abstract: Subspace predictive control (SPC) is recently seen in the literature for joint system identification and control design. This combination enables automatically tuning the parameters in conventional model predictive control (MPC); and therefore provides a solution to the problem of fault tolerant MPC design. The existing SPCs either deal with open-loop data or depend on the information of the controller in a closed loop. In this paper we introduce a new closed-loop SPC method, which is independent of any controller information. Both the analytic solution to the unconstrained case and the quadratic programming problem for the constrained case are formulated. A recursive solution for updating the SPC control law is proposed. A fault tolerant MPC scheme is then developed based on the recursive algorithm, whose effectiveness is demonstrated on tolerating a fault in a steer-by-wire actuator.

1. INTRODUCTION

Conventional model predictive control (MPC) is based on system models either built from physical principles or identified; or a sequence of impulse or step response coefficients. The subspace predictive control (SPC) approaches as recently seen in Favoreel et al. (1998, 1999); Woodley (2001), circumvents the modeling step, and directly seeks the predictors of future outputs from data. The N4SID algorithm, van Overschee and de Moor (1994), is applied to estimate such predictors in both studies. The SPC algorithms do not proceed further to recover the state space matrices from the estimated state sequences as does in N4SID, which is again an estimation problem and may need to choose model order. Favoreel et al. (1998) focuses on LQG design; while Woodley (2001) extends SPC to $\mathcal{H}_\infty$ control. However, these algorithms are based on open-loop identification. In Favoreel et al. (1999), the authors extends their previous results to handle closed-loop data using the closed-loop identification of van Overschee and de Moor (1996), which requires computing instrumental variables based on the impulse response coefficients of the controller, and is thus more prune to errors and fails to work in case the controller is not LTI, e.g. a constrained MPC controller. To overcome these drawbacks, the updating of a predicted controller is renewed based on the VARX (vector autoregressive with exogenous inputs) algorithm, developed in Chiuso (2007a,b) for closed-loop identification.

The capability of MPC in reconfiguring controllers is recognized in Maciejowski (1997, 1998), when there are actuator redundancies in a system. As also argued in Maciejowski (1997), to handle partial actuator failures, automatic re-estimation of the system model is necessary, which is known as the FDI (fault detection and isolation) step in fault tolerant control (FTC). The combination of closed-loop subspace identification and predictive control provides such an opportunity for online reconfiguring MPC controllers when a system is time varying or subject to faults. It is thus the purpose of this paper to provide a solution to this problem.

The paper is organized as follows. We introduce the necessary technical details of the VARX algorithm in Section 2. The main results of this paper is presented in Section 3, where the analytical solution for the unconstrained problem is derived; and the constrained problem is formulated. The recursive solution is developed in Section 4, where the computational complexity of the recursive algorithm is also analyzed. Based on this recursive solution, a fault tolerant MPC scheme is then proposed. The application of this scheme on a steer-by-wire actuator is presented as a case study in Section 5. The paper concludes in Section 6 with the direction for future research.

2. CLOSED-LOOP SUBSPACE IDENTIFICATION

Consider the innovation type state space model:

$$x(k+1) = Ax(k) + Bu(k) + Ke(k) \quad (1)$$

$$y(k) = Cx(k) + e(k), \quad (2)$$

where $e(k)$ is assumed to be a zero-mean white noise with a non-singular variance of $\mathcal{E}\mathcal{E}^T$. The dimensions are assumed to be $x(k) \in \mathbb{R}^n$, $y(k) \in \mathbb{R}^l$, and $u(k) \in \mathbb{R}^m$. We make the following assumption on the plant, which is commonly assumed in subspace identification.

Assumption 1. $\Phi \triangleq A - KC$ is stable, and the system is minimal. □

In the sequel, we denote by s, f respectively the past and future horizon ($s \geq f$), in both identification and control. N represents the number of columns in the identification data matrices. Let t be the current time instant in the formulation of the identification problem. We shall denote by k the current time instant in the control formulation.

The first step in identifying the state space model (1) and (2) is to solve the following least square problem,

$$Y_t = C\Phi^s X_{t-s} + \Xi_0 \mathcal{Z}_{[t-s,t)} + E_t. \tag{3}$$

$\Xi_0 \triangleq \big[C\Phi^{s-1}B \ C\Phi^{s-1}K \ \cdots \ CB \ CK\big]$ contains the Markov parameters of the system. $Y_t = \big[y(t) \ y(t{+}1) \ \cdots \ y(t{+}N{-}1)\big]$ and $E_t = \big[e(t) \ e(t{+}1) \ \cdots \ e(t{+}N{-}1)\big]$ are respectively the future output and innovation sequence. The past I/O data are collected in the matrix

$$\mathcal{Z}_{[t-s,t)} = \begin{bmatrix} u(t-s) & u(t-s+1) & \cdots & u(t-s+N-1) \\ y(t-s) & y(t-s+1) & \cdots & y(t-s+N-1) \\ u(t-s+1) & u(t-s+2) & \cdots & u(t-s+N) \\ y(t-s+1) & y(t-s+2) & \cdots & y(t-s+N) \\ \vdots & \vdots & & \vdots \\ u(t-1) & u(t) & \cdots & u(t+N-2) \\ y(t-1) & y(t) & \cdots & y(t+N-2) \end{bmatrix}.$$

The subscript "$[t-s,t)$" stands for the range of the time index along the first column of the $\mathcal{Z}$ matrix; while the number of columns, N, is omitted to simplify the notations.

$X_{t-s} = \big[x(t-s) \ x(t-s+1) \ \cdots \ x(t-s+N-1)\big]$ is the sequence of the unknown initial states, which induces a bias to the identification of Ξ_0. Due to the scaling by $C\Phi^s$, this bias can in fact be made arbitrarily small by choosing a relatively large s. Note that in (3), each column of E_t is at least one step ahead in the future of the I/Os along the corresponding column of $\mathcal{Z}_{[t-s,t)}$. Due to the causality of the system, $\lim_{N\to\infty} E_t \cdot \mathcal{Z}^T_{[t-s,t)} = 0$. This requires to take a large N. We assume N, s are so large that both the two sources of bias are negligible. We shall refer to Chiuso (2007b) for the asymptotic behavior of the subspace identification; while postpone the rigorous treatment of the finite-horizon case (finite N, s, f) in Dong and Verhaegen (2008). We shall only focus on this paper to formulate the nominal SPC and its recursive solution for fault tolerant MPC design.

Now, with the biases neglected, solving (3) in a least square (LS) sense results in a consistent estimate of Ξ_0, no matter the data are from open or closed loop experiments; i.e.

$$\hat{\Xi}_0 = Y_t \cdot \mathcal{Z}^{\dagger}_{[t-s,t)}, \tag{4}$$

where "$\dagger$" stands for pseudo inverse. To ensure $\mathcal{Z}^{\dagger}_{[t-s,t)}$ is unique, we need to make the following assumption, which is the necessary and sufficient condition for persistently exciting the system of any order.

Assumption 2. Chiuso (2007b), the spectrum of the joint input and output signals $\mathbf{z}(k) = \big[u^T(k) \ y^T(k)\big]^T$ (denoted by $\mathbf{\Phi_z}$) is bounded and bounded away from zero on the unit circle, i.e. $\exists 0 < c \le M < \infty$, s.t. $cI \le \mathbf{\Phi_z}(e^{j\omega}) \le MI$, $\forall \omega \in [0, 2\pi)$. □

With the identified Markov parameters in $\hat{\Xi}_0$, we are able to proceed to develop the new closed-loop SPC in the following sections.

3. CLOSED-LOOP SUBSPACE PREDICTIVE CONTROL

3.1 The output predictor

Let the prediction horizon be f. To distinguish with $\mathcal{Z}_{[t-s,t)}$ in the identification problem, we use $\bar{Z}_{[k-s,k)} = \big[u(k{-}s)^T \ y(k{-}s)^T \ \cdots \ u(k{-}1)^T \ y(k{-}1)^T\big]^T$ to represent the past I/Os in the control problem. It is shown in Chiuso (2007b) that the future f step ahead output predictor takes the following form,

$$\hat{\mathbf{y}}_{[k,k+f)} \approx \begin{bmatrix} \Xi_0 \\ \Xi_1 \\ \vdots \\ \Xi_{f-1} \end{bmatrix} \bar{Z}_{[k-s,k)} + \begin{bmatrix} 0 & & & & \\ \Psi_1 & 0 & & & \\ \vdots & \vdots & \ddots & & \\ \Psi_{f-1} & \Psi_{f-2} & \cdots & \Psi_1 & 0 \end{bmatrix} \cdot \begin{bmatrix} u(k) \\ y(k) \\ \vdots \\ u(k+f-1) \\ y(k+f-1) \end{bmatrix} + \underbrace{\begin{bmatrix} C\Phi^s x(k-s) \\ C\Phi^{s+1} x(k-s) \\ \vdots \\ C\Phi^{s+f-1} x(k-s) \end{bmatrix}}_{\mathbf{b}_x}, \tag{5}$$

where $\Psi_\tau \triangleq C\Phi^{\tau-1}\big[B \ K\big]$, $\tau = 1, \cdots, f-1$; and $\Xi_i = \big[0_{l\times i(m+l)} \ C\Phi^{s-1}B \ C\Phi^{s-1}K \ \cdots C\Phi^i B \ C\Phi^i K\big]$ is simply a right-shifted and zero-padded version of Ξ_0. Here we use $0_{m\times n}$ to represent an m-by-n zero matrix; and I_m an m-dimensional identity matrix. The predictor can therefore be derived based on the estimate $\hat{\Xi}_0$.

Remark 1. Note that the above approximate equality is due to ignoring the higher order terms, $C\Phi^{s+\tau}\big[B \ K\big], 0 \le \tau < i$ in $\Xi_i, \forall 0 < i < f$. In fact, since

$$\sum_{\tau=0}^{i-1} C\Phi^{s+\tau} \cdot [\, B \ K \,] \cdot \begin{bmatrix} u(k-s+i-1-\tau) \\ y(k-s+i-1-\tau) \end{bmatrix} + C\Phi^{s+i} x(k-s) = C\Phi^s \cdot x(k-s+i),$$

the approximate equality can be replaced with an equality by changing $\mathbf{b}_x$ to $\big[(C\Phi^s x(k-s))^T, \ (C\Phi^s x(k-s+1))^T, \ (C\Phi^s x(k-s+f-1))^T\big]^T$. If the closed-loop system is internally stable before the time instant k, by choosing a large enough s, this unknown bias term can be neglected, based on Assumption 1. □

Denote $\hat{\mathbf{y}}_{[k,k+f)} \triangleq \big[\hat{y}^T(k) \ \hat{y}^T(k+1) \ \cdots \ \hat{y}^T(k+f-1)\big]^T$; and by $\mathbf{u}_{[k,k+f-1)} \triangleq \big[u^T(k) \ u^T(k{+}1) \ \cdots \ u^T(k{+}f{-}2)\big]^T$ the future control inputs. The following lemma constructs the output predictor from the identified Markov parameters in $\hat{\Xi}_0$.

Lemma 1. If the closed-loop system is internally stable before the time instant k and s is so large that $\|\Phi^s\|_2 \ll 1$, then the future f outputs are predicted by

$$\hat{\mathbf{y}}_{[k,k+f)} = \underbrace{\begin{bmatrix} \Gamma_0 \\ \Gamma_1 \\ \Gamma_2 \\ \vdots \\ \Gamma_{f-1} \end{bmatrix}}_{\Gamma} \bar{Z}_{[k-s,k)} + \left[\begin{array}{c} 0 \\ \hline \underbrace{\begin{bmatrix} \Lambda_1 & & & \\ \Lambda_2 & \Lambda_1 & & \\ \vdots & \vdots & \ddots & \\ \Lambda_{f-1} & \Lambda_{f-2} & \cdots & \Lambda_1 \end{bmatrix}}_{\Lambda} \end{array}\right] \cdot \mathbf{u}_{[k,k+f-1)}, \tag{6}$$

where the parameters, $\{\Gamma_i,\ \Lambda_j | i, j = 1, \cdots, f-1\}$, are derived from

$$\begin{aligned}\Gamma_i &= \hat{\Xi}_i + \sum_{\tau=0}^{i-1} \overline{C\Phi^{i-\tau-1}K} \cdot \Gamma_\tau, \qquad\qquad \Gamma_0 = \hat{\Xi}_0, \\ \Lambda_j &= \overline{C\Phi^{j-1}B} + \sum_{\tau=1}^{j-1} \overline{C\Phi^{j-\tau-1}K} \cdot \Lambda_\tau,\ \Lambda_1 = \overline{CB},\end{aligned} \tag{7}$$

where $\overline{C\Phi^\tau B}$ and $\overline{C\Phi^\tau K}$, $\tau = 0, \cdots, f-2$, are the block elements of the estimate, $\hat{\Xi}_0$; and $\hat{\Xi}_i = \left[0_{l\times i(m+l)}\ \overline{C\Phi^{s-1}B}\ \overline{C\Phi^{s-1}K}\ \cdots \overline{C\Phi^i B}\ \overline{C\Phi^i K}\right]$. □

Proof See Appendix A for a sketch of proof. □

Since $\hat{y}(k)$ does not depend on u, the first block row in (6) can be deleted, which can then be reduced to

$$\hat{\mathbf{y}}_{[k+1,k+f)} = \tilde{\Gamma} \cdot \bar{Z}_{[k-s,k)} + \Lambda \cdot \mathbf{u}_{[k,k+f-1)}, \tag{8}$$

where $\tilde{\Gamma} = \left[\, \Gamma_1^T\ \Gamma_2^T\ \cdots\ \Gamma_{f-1}^T \,\right]^T$.

3.2 Unconstrained closed-loop SPC

Consider first the unconstrained SPC problem. Assume the future reference signals, $\mathbf{r}_{[k+1,k+f)} = [r^T(k+1)\ r^T(k+2)\ \cdots\ r^T(k+f-1)]^T$, are known. Define a quadratic cost function,

$$J(k+1) = \|\mathbf{r}_{[k+1,k+f)} - \hat{\mathbf{y}}_{[k+1,k+f)}\|_Q^2 + \|\Delta\mathbf{u}_{[k,k+f-1)}\|_R^2, \tag{9}$$

where $\Delta u(k+i) = u(k+i) - u(k+i-1)$ is the input change at instant $k+i$. $Q, R \succ 0$ are weighting matrices. $\|v\|_Q^2 \triangleq v^TQv$ defines a weighted 2-norm. Define the following structure matrices,

$$S_\Delta = \begin{bmatrix} I_m & & & \\ -I_m & I_m & & \\ & \ddots & \ddots & \\ & & -I_m & I_m \end{bmatrix}, \text{and}$$

$$S_{k-1} = \begin{bmatrix} 0_{m\times(s-1)(m+l)} & I_m & 0_{m\times l} \\ 0_{(f-2)m\times(s-1)(m+l)} & 0_{(f-2)m\times m} & 0_{(f-2)m\times l} \end{bmatrix}.$$

Then one can write the $f-1$ control changes as

$$\Delta\mathbf{u}_{[k,k+f-1)} = S_\Delta \cdot \mathbf{u}_{[k,k+f-1)} - S_{k-1} \cdot \bar{Z}_{[k-s,k)}. \tag{10}$$

The unconstrained closed-loop SPC design problem reads

$$\mathbf{u}_{[k,k+f-1)} = \arg \min_{\mathbf{u}_{[k,k+f-1)}} J(k+1), \tag{11}$$

whose solution is derived in the following theorem.

Theorem 1. The analytic solution to the unconstrained closed-loop SPC design problem, (11), is

$$\begin{aligned}\mathbf{u}^*_{[k,k+f-1)} = &\left[\Lambda^TQ\Lambda + S_\Delta^TRS_\Delta\right]^{-1}\left[\Lambda^TQ\big(\mathbf{r}_{[k+1,k+f)}\right. \\ &\left.-\tilde{\Gamma}\bar{Z}_{[k-s,k)}\big) + S_\Delta^TRS_{k-1}\bar{Z}_{[k-s,k)}\right].\end{aligned} \tag{12}$$

□

Note that since $Q, R \succ 0$, $\Lambda^TQ\Lambda \succeq 0$ and $S_\Delta^TRS_\Delta \succeq 0$. Furthermore, since S_Δ is a full-rank square matrix, $S_\Delta \cdot v \neq 0$, $\forall\, v \in \mathbb{R}^{(f-1)m} \neq 0$; and therefore $S_\Delta^TRS_\Delta \succ 0$. This leads to the conclusion that $\Lambda^TQ\Lambda + S_\Delta^TRS_\Delta$ is positive definite, and hence invertible.

3.3 Constrained closed-loop SPC

When inequality constraints have to be imposed to Problem (11), no analytic solution exists. Furthermore, the controller is no more a linear function of $\mathbf{r}_{[k+1,k+f)}$ and $\bar{Z}_{[k-s,k)}$, depending on whether the constraints are active or not. In this case, an online optimization problem has to be solved. In this paper, we consider two types of constraints, the limits on inputs and outputs. Other linear constraints such as the input incremental constraints can be incorporated in the same way.

Denote $\alpha(k+1) = \tilde{\Gamma}\bar{Z}_{[k-s,k)} - \mathbf{r}_{[k+1,k+f)}$ and $\beta(k) = S_{k-1}\bar{Z}_{[k-s,k)} = \left[u^T(k-1)\, 0 \cdots 0\right]^T$. Then the quadratic cost function in (9) can be rewritten in a standard quadratic form as

$$\begin{aligned}J(k+1) = &\,\mathbf{u}^T_{[k,k+f-1)} \cdot [\Lambda^TQ\Lambda + S_\Delta^TRS_\Delta] \cdot \mathbf{u}_{[k,k+f-1)} \\ &+[2\alpha^T(k+1)Q\Lambda - 2\beta^T(k)RS_\Delta] \cdot \mathbf{u}_{[k,k+f-1)} \\ &+\alpha^T(k+1)Q\alpha(k+1) + \beta^T(k)R\beta(k).\end{aligned} \tag{13}$$

Suppose the input limits are $u_{\min} \leq u(k+i) \leq u_{\max}$; and the outputs constraints are $y_{\min} \leq y(k+i) \leq y_{\max}$. Then the quadratic programming (QP) problem is summarized in the following proposition.

Proposition 1. The linear inequality constrained closed-loop SPC design is the following convex QP problem,

$$\min_{\mathbf{u}_{[k,k+f-1)}} \ (13) \quad \text{subject to} \tag{14}$$

$$\begin{bmatrix} \Lambda \\ -\Lambda \\ I_{(f-1)m} \\ -I_{(f-1)m} \end{bmatrix} \cdot \mathbf{u}_{[k,k+f-1)} \leq \begin{bmatrix} \mathbf{y}^{f-1}_{max} - \tilde{\Gamma}\bar{Z}_{[k-s,k)} \\ \tilde{\Gamma}\bar{Z}_{[k-s,k)} - \mathbf{y}^{f-1}_{min} \\ \mathbf{u}^{f-1}_{\max} \\ -\mathbf{u}^{f-1}_{\min} \end{bmatrix} \tag{15}$$

where $\mathbf{y}^{f-1}_{\max} \triangleq \left[\, y^T_{\max} \cdots\ y^T_{\max} \,\right]^T$ (same definition for $\mathbf{u}^{f-1}_{\min}$, $\mathbf{u}^{f-1}_{\max}$, and $\mathbf{y}^{f-1}_{\min}$. The operator "$\leq$" is element-wise. □

To see the problem is indeed convex, note that $\Lambda^TQ\Lambda + S_\Delta^TRS_\Delta \succ 0$; and the linear constraints in (15) define a convex set.

Now, we are ready to compare the new closed-loop SPC design with Favoreel et al. (1999). We shall denote the two algorithms respectively by CLSPC and FDGV99 in the sequel.

3.4 A Comparison between CLSPC and FDGV99

As a comparison, we use the same example as in Favoreel et al. (1999); except that we impose a saturation unit to the initial linear controller in the loop; i.e. $u(k) = 1$ if $u(k) \geq 1$, and $u(k) = -1$ if $u(k) \leq -1$. The saturation changes the initial linear controller to a nonlinear one. Since the saturation does not change the impulse response coefficients of the initial controller itself, it leaves the identified plant Markov parameters in FDGV99 unchanged, as if the saturation unit is absent. To identify the Markov parameters, the system is excited by a reference signal $r(k)$, which is a zero-mean white noise with variance 1.

The noise $e(k)$ is a zero-mean white noise with a variance of 1/9.

To test the two control algorithms[1], we set the reference to a constant, $r(k) \equiv 100$. The past and future horizons are respectively $s = 20, f = 10$. The weighting matrices are chosen as $Q = 10I_9$ and $R = I_9$. The simulation results are shown in figure 1. It is clearly seen that, in this constrained case, FDGV99 fails. This can be attributed to the fact that, the identified Markov parameters in FDGV99 are biased, since they are identified from the closed-loop experiment, where the controller is nonlinear, instead of LTI. This bias leads to a remarkable model-plant mismatch, which then results in biased output prediction in the implementation of FDGV99.

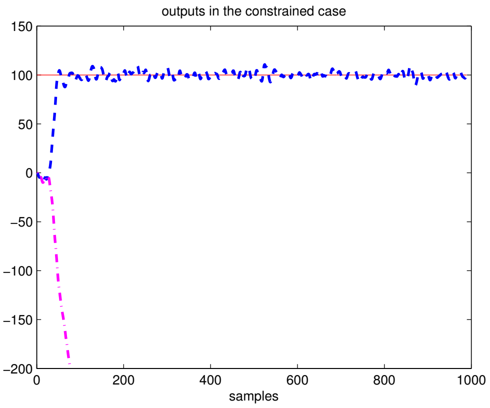

Fig. 1. A comparison between CLSPC and FDGV99 (solid red: reference; dashed blue: CLSPC; dash-dotted purple: FDGV99). The first $s = 20$ I/Os initialize the SPCs.

4. A FAULT TOLERANT MPC SCHEME

Based on a recursive implementation of the new closed-loop SPC, a fault tolerant MPC design is developed in this section. Specifically, at each sampling instant, the plant Markov parameters are recursively identified; the output predictor is updated; and finally the optimal control inputs are computed.

4.1 Recursive identification

It is well known in the literature that the full rank least square problem, (4), can be solved by QR factorization, Golub and van Loan (1996). In fact, one can compute the following LQ factorization to compress the data matrices,

$$\begin{bmatrix} \mathcal{Z}_{[t-s,t)} \\ Y_t \end{bmatrix} = \underbrace{\begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}}_{L} \cdot \underbrace{\begin{bmatrix} O_1 \\ O_2 \end{bmatrix}}_{O}, \tag{16}$$

where L is lower triangular and O is orthonormal. Then (4) can simply be solved as

$$\hat{\Xi}_0 = L_{21} \cdot L_{11}^{-1}, \tag{17}$$

where $L_{11} \in \mathbb{R}^{s(m+l)\times s(m+l)}$ and $L_{21} \in \mathbb{R}^{l\times s(m+l)}$. $O \in \mathbb{R}^{N\times N}$ is not needed. Obviously, L_{11} has much smaller dimension than $\mathcal{Z}_{[t-s,t)}$, because the data number N must be much larger than $s(m+l)$ to make $\mathcal{Z}_{[t-s,t)}$ full row rank. The number N is usually chosen to cover a duration as long as at least 10 times the longest time constant of a system, Verhaegen and Verdult (2007). It is clear that when N must be large, a large memory in the on-board computer is required to store $\mathcal{Z}_{[t-s,t)}$ and Y_t. In addition, the computation time to solve (16) and (17) increases with the dimension of $\mathcal{Z}_{[t-s,t)}$. For large systems, a stand-alone computing unit may be necessary to fulfill the real-time requirement of the overall system. The recursive solution to (16) and (17) can overcome these two drawbacks.

We shall refer to Hallouzi and Verhaegen (2007) for the detailed implementation of the recursive LQ factorization in SPCs. We only mention the key steps in this paper for completeness. Assume the a priori knowledge of the L matrix is $L^{(k-1)}$, where k represents the current time instant. At time k, the new output measurement, $y(k)$, is available. Then the one-step Givens rotation for updating $L^{(k-1)}$ can be computed as

$$\left[\begin{array}{c|c|c} L_{11}^{(k)} & 0 & \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \\ \hline L_{21}^{(k)} & L_{22}^{(k)} & * \end{array}\right] = \left[\begin{array}{c|c|c} \sqrt{\lambda} L_{11}^{(k-1)} & 0 & \begin{bmatrix} u(k-s) \\ y(k-s) \\ \vdots \\ u(k-1) \\ y(k-1) \end{bmatrix} \\ \hline \sqrt{\lambda} L_{21}^{(k-1)} & \sqrt{\lambda} L_{22}^{(k-1)} & y(k) \end{array}\right] \cdot P^{(k)}. \tag{18}$$

$\lambda \in [0.97, 1]$ is a forgetting factor. The orthonormal matrix $P^{(k)}$ is the product of a sequence of Givens matrices, eliminating the "new" data vector $\bar{Z}_{[k-s,k)} \triangleq \left[u^T(k-s)\ y^T(k-s)\ \cdots\ u^T(k-1)\ y^T(k-1)\right]^T$. The $*$ in the equation is not needed to compute.

At time instant k, the Markov parameters are updated by

$$\hat{\Xi}_0^{(k)} = L_{21}^{(k)} \cdot (L_{11}^{(k)})^{-1}, \tag{19}$$

where the inverse can be computed by the forward substitution algorithm, due to the lower triangular structure of $L_{11}^{(k)}$. The update of the predictor parameters can be subsequently performed according to (7). Now, the control inputs can be computed by solving (14) using a QP solver.

4.2 Computational complexity analysis

As a measure of the computational complexity, we count the number of "flops", as defined in Golub and van Loan (1996); i.e. a flop is a floating point operation, including $+, -, \times, \div$. We use the notation $O(n^m)$ to represent terms of order n^m and lower. Without going into details, we summarize the complexity of each computation step in the following table, where $\bar{n} \triangleq s(m+l)$ for simplicity. Obviously, the overall computational complexity is dominated by computing (19).

	Flops	*Order*
(18)	$\frac{7}{2}\bar{n}^2 + 7\bar{n}l + \frac{7}{2}\bar{n} + \frac{1}{2}(l^2 + l)$	$O(\frac{7}{2}\bar{n}^2)$
(19)	$\frac{\bar{n}(\bar{n}+1)(2\bar{n}+1)}{6} + l\bar{n}(\bar{n}+1)$	$O(\frac{\bar{n}^3}{3})$
(7)	$(m+l)l(f-1)[s(lf+1) - \frac{f}{2}]+$ $lm(f-1)[l(f-2)+1]$	$O(f^2l^2\bar{n})+$ $O(f^2l^2m)$
total	-	$O(\frac{\bar{n}^3}{3})$

Table 1. Computational complexity analysis

[1] FDGV99 has been modified to a constrained QP problem following the same procedures in Section 3.3.

```
Algorithm 1. (Fault tolerant MPC based on Recursive CLSPC).

Inputs: $s, f, Q, R, u_{\min}, u_{\max}, y_{\min}, y_{\max}, \lambda$, reference signal,
    and the $L$ matrix of the nominal system, denoted as $L^{(s)}$.
    Set $k = s$.
Step 1. Set $k \leftarrow k+1$, collect the current output $y(k)$.
Step 2. Run the Givens rotation algorithm, (18), to compute
    $L^{(k)}$.
Step 3. Run the forward substitution algorithm to compute
    $(L_{11}^{(k)})^{-1}$.
Step 4. Compute the matrix multiplication (19) to get $\hat{\Xi}_0^{(k)}$.
Step 5. Update the predictor matrices $\tilde{\Gamma}$ and $\Lambda$ according to
    (7).
Step 6. Solve (14) for $\mathbf{u}_{[k,k+f-1)}$ using a QP solver.
Output: $u(k)$ at the current sampling time.
```

On the other hand, according to Golub and van Loan (1996), the Givens QR factorization in the batch mode of (16) has a complexity of

$$3 \cdot (\bar{n}+l)^2 \cdot (N - \frac{\bar{n}+l}{3}) = O(3\bar{n}^2 N) \text{ flops,} \qquad (20)$$

which is usually much larger than that of the one-step update, since $N \gg \bar{n}$. For the simulation case in Section 5, where $s = f = 10, N = 2000$, the recursive and batch mode of solving (17) require respectively 6.2×10^3 and 2.4×10^6 flops.

4.3 The fault tolerant MPC scheme

The recursive solution to the closed-loop SPC combines predictive control and automatic tuning of system parameters, and consequently provides an adaptive way to control a time varying system or a system subject to faults. The information needed to tune the predictor parameters is only the I/O signals from a running process, provided the input signals can persistently excite the system. We propose the fault tolerant MPC scheme in Algorithm 1. The closed-loop control strategy is illustrated in figure 2, where the following matrices are used to define some of its blocks.

$$S_Z = \begin{bmatrix} 0_{(s-1)(m+l)\times(m+l)} & I_{(s-1)(m+l)} \\ 0_{(m+l)\times(m+l)} & 0_{(m+l)\times(s-1)(m+l)} \end{bmatrix},$$

$$B_u = \begin{bmatrix} 0_{(s-1)(m+l)\times m} \\ I_m \\ 0_{l\times m} \end{bmatrix}, \; B_y = \begin{bmatrix} 0_{((s-1)(m+l)+m)\times l} \\ I_l \end{bmatrix},$$

$$T_Z = \begin{bmatrix} I_{s(m+l)} \\ 0_{l\times s(m+l)} \end{bmatrix}, \; T_y = \begin{bmatrix} 0_{s(m+l)\times l} \\ I_l \end{bmatrix}, \; S_u = \begin{bmatrix} I_m & 0_{m\times(f-2)m} \end{bmatrix}.$$

5. CASE STUDY OF A STEER-BY-WIRE ACTUATOR

5.1 The steer-by-wire actuator

As a case study, we consider an *SKF* steer-by-wire (SbW) system, as schematically shown in figure 3. The actuating module is a steering rack (ball screw) driven by two brushless DC motors (BDCMs). The control objective is to follow the rack position setpoints given by a driver via the hand wheel. The inputs to the system are the currents to the two BDCMs. The current magnitude cannot exceed $40A$. The rack cannot displace more than $80mm$ from its central position to both sides.

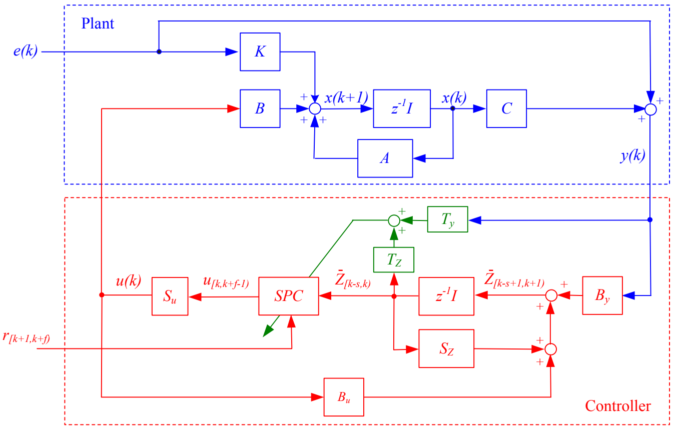

Fig. 2. The recursive Closed-loop SPC Scheme. The SPC is a function of $\mathbf{r}_{[k+1,k+f)}$ and $\bar{Z}_{[k-s,k)}$, parameterized by $\tilde{\Gamma}, \Lambda, Q, R$. The recursive tuning of the SPC takes as input the signal vector, stacking $\bar{Z}_{[k-s,k)}$ on top of $y(k)$.

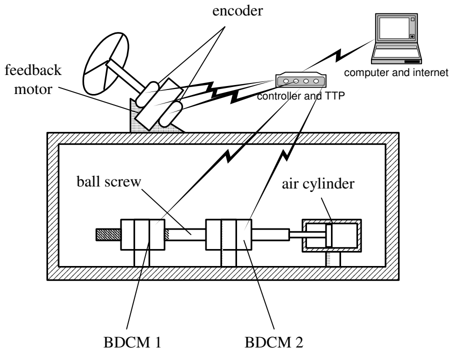

Fig. 3. The *SKF* Steer-by-Wire System.

The physical modeling of such a system is studied in Park et al. (2005). Although the lateral forces acting on the front tyres of a vehicle are nonlinear (and time variant), they can still be well approximated by springs and dampers connected to the rack. It is not the interest of this paper to go through this modeling and estimate the parameters such as the lateral stiffness of the front suspension. We shall, instead, implement the proposed fault tolerant MPC scheme on this system, which is in fact a model-free approach. To make the experiment realistic, we incorporate the SbW actuating module into a *Dymola* vehicle model. *Dymola* simulates the overall vehicle dynamics, where the tyre-road interaction is computed by the Magic formula. For our experiment, only the rack displacement (the output of the SbW system) and the current fed to the BDCMs (the output of the CLSPC algorithm) are assumed measurable. The parameters in the simulation model are all hidden. To mimic the real situation, the measurement noise is assumed to be white with a zero mean and a standard deviation of $0.007mm$, due to the limited resolution of the position sensor.

The maneuver of the car is designed as follows. A 2-ton car is driven on a dry asphalt road, at a constant longitudinal speed of $30m/s$. During the first 2 seconds, the car runs along a straight line. From the third to the sixth second, the car is steered to the right, following a trapezoidal rack position command with a magnitude of $20mm$. The car continues running along a straight line during the seventh second. Then from the eighth to the tenth second, the car is steered to the left, following a sinusoidal rack position command with a magnitude of $40mm$ and a frequency of $0.1Hz$. See figure 4.

We consider the loss of one BDCM after the first second. This failure is injected to the simulation model by disabling the motor. The fault tolerant MPC scheme in Algorithm 1 runs during the entire simulation. To ensure that the persistent excitation condition holds, as in Assumption 2, we impose a white noise signal to the rack position command; i.e. to ensure that the reference signal persistently excites the closed-loop system to a sufficient order; i.e.

$$5 \times 10^{-4} n(t) \quad [m] \tag{21}$$

where $n(t)$ has zero mean and unit variance. This noise can be viewed as an emulation of the small perturbations from the driver and the road. The parameters in the algorithm are chosen as $Q = 10^6 I, R = 10^{-3} I, s = f = 10, \lambda = 0.99$. The outputs of the closed-loop system are shown as the red dash-dotted curve in Fig. 4. The fault tolerating performance is compared against a PID controller (blue dashed curve), with carefully tuned anti-windup parameters. It can be clearly seen that the recursive closed-loop SPC algorithm outperforms the PID controller, within the limit of the remaining BDCM.

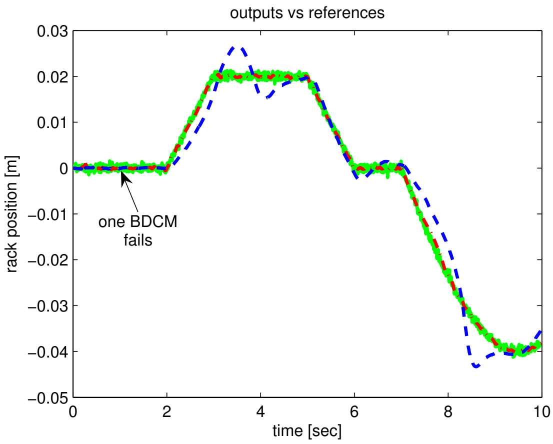

Fig. 4. Comparison of the performance of the PID controller and the fault tolerant MPC scheme, where one BDCM fails at the first second.

6. CONCLUSION

We present in this paper a new closed-loop SPC algorithm, together with its recursive solution for online implementations. A fault tolerant MPC scheme is then developed and tested on a steer-by-wire system. The future research shall be focused on ensuring the closed-loop stability and considering the persistent excitation condition in designing the control law.

Appendix A. PROOF OF LEMMA 1

It is straightforward to verify that when $i = 0, 1$, Lemma 1 holds. Now suppose $\forall 0 \leq i < f - 1$,

$$\hat{\mathbf{y}}_{[k,k+i+1)} = \begin{bmatrix} \Gamma_0 \\ \Gamma_1 \\ \vdots \\ \Gamma_i \end{bmatrix} \bar{Z}_{[k-s,k)} + \begin{bmatrix} 0 & \cdots & 0 \\ \Lambda_1 & & \\ \vdots & \ddots & \\ \Lambda_i & \cdots & \Lambda_1 \end{bmatrix} \cdot \mathbf{u}_{[k,k+i)}.$$

Substitute $\hat{\mathbf{y}}_{[k,k+i+1)}$ into the prediction of $y(k+i+1)$,

$$\hat{y}(k+i+1) = \hat{\Xi}_{i+1} \bar{Z}_{[k-s,k)} + \left[\, \overline{C\Phi^i B} \;\cdots\; \overline{CB} \,\right] \cdot \mathbf{u}_{[k,k+i)} \\ + \left[\, \overline{C\Phi^i K} \;\cdots\; \overline{CK} \,\right] \cdot \hat{\mathbf{y}}_{[k,k+i+1)}.$$

Then Lemma 1 follows by arranging the terms on the right hand side.

ACKNOWLEDGEMENTS

This research is jointly funded by the grant of the European AI4IA project (Contract No. 514510) under the Marie Curie FP6 Framework, and *SKF ERC* in the Netherlands. We would also like to thank F. Pepe, A. Carlevaris, and their colleagues in *SKF Airasca* in Italy for the valuable discussions about the steer-by-wire system.

REFERENCES

A. Chiuso. The role of vector autoregressive modeling in predictor based subspace identification. *Automatica*, 43: 1034–1048, 2007a.

A. Chiuso. On the relation between CCA and predictor-based subspace identification. *IEEE Transactions on Automatic Control*, 52:1795–1812, 2007b.

J. Dong and M. Verhaegen. Cautious $\mathcal{H}_2$ optimal control using uncertain markov parameters identified in closed loop. *submitted for publication*, 2008.

W. Favoreel, B. de Moor, M. Gevers, and P. van Overschee. Model-free subspace-based LQG-design. *Tech. Rep. ESAT-SISTA/TR 1998-34, KU Leuven, Leuven, Belgium*, 1998.

W. Favoreel, B. de Moor, M. Gevers, and P. van Overschee. Closed-loop model-free subspace-based LQG-design. In *the Proceedings of the 7th Mediterranean Conference on Control and Automation.* Haifa, Israel, 1999.

G.H. Golub and C.F. van Loan. *Matrix Computations, 3rd Edition.* Johns Hopkins University Press, 1996.

R. Hallouzi and M. Verhaegen. Reconfigurable fault tolerant control of a boeing 747 using subspace predictive control. In *AIAA Guidance, Navigation and Control Conference and Exhibit.* South Carolina, US, 2007.

J.M. Maciejowski. Reconfigurable control using constrained optimization. In *Semi-Plenary paper for the European Control Conference.* 1997.

J.M. Maciejowski. The implicit daisy-chaining property of constrained predictive control. *Applied Mathematics and Computer Science*, 8:695–711, 1998.

T.J. Park, C.S. Han, and S.H. Lee. Development of the electronic control unit for the rack-actuating steer-by-wire using the hardware-in-the-loop simulation system. *Mechatronics*, 15:899–918, 2005.

P. van Overschee and B. de Moor. N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 36:75–93, 1994.

P. van Overschee and B. de Moor. Closed-loop subspace identification. *Tech. Rep. ESAT-SISTA/TR 1996-52I, KU Leuven, Leuven, Belgium*, 1996.

M. Verhaegen and V. Verdult. *Filtering and System Identification: A Least Squares Approach.* Cambridge University Press, 2007.

B.R. Woodley. *Model free subspace based $\mathcal{H}_\infty$ control, PhD Dissertation.* Stanford University, 2001.