Monograph:

Rossiter, J.A. and Richalet, J. (2001) Sensitivity and a Preferable Alternative to Re-aligned Models for Prediction in MPC. Research Report. ACSE Research Report 808 . Department of Automatic Control and Systems Engineering

Sensitivity and a preferable alternative to re-aligned models for prediction in MPC

by J A Rossiter and J Richalet

RESEARCH REPORT NO. 808

October 2001

SENSITIVITY AND A PREFERABLE ALTERNATIVE TO RE-ALIGNED MODELS FOR PREDICTION IN MPC

J.A. Rossiter *

* *Dept. of Automatic Control & Systems Engineering, Sheffield University, Mappin Street, Sheffield, S1 3JD, email:* J.A.Rossiter@sheffield.ac.uk, *Tel. 44 114 2225685*

J. Richalet **

** *ADERSA, 10 Rue de la Croix Martre, 91873 PALAISEAU CEDEX, France, email:* jacques.richalet@adersa.com

Abstract: Most papers on predictive control use either state-space models with an observer or transfer function models with output realignment for prediction purposes. Here it is shown that the this approach can have weaknesses, especially with regard to noise rejection and the independent model approach should often be preferred.

Keywords: Predictive control, prediction models, sensitivity

1. INTRODUCTION

Model predictive control (MPC), e.g.(Garcia *et al.*, 1989) is a popular control strategy, however there is a significant discrepancy between the early variants such as Dynamic matrix control (DMC), (Cutler *et al.*, 1980), IDCOM ((Richalet *et al.*, 1978)) and IMC (Garcia *et al.*, 1982) and the algorithms common in the academic literature, for instance Generalized predictive control (GPC), (Clarke *et al.*, 1987). Historically the main difference is in the type of model used for computation of the system predictions; DMC and IDCOM use FIR models, IMC uses an internal model (often a FIR) whereas academia has favoured transfer function and state space models. More recently industrial vendor ADERSA in its product PFC (Predictive functional control) have favoured the use of independent models a concept that bears a strong resemblance to the internal model principle in IMC (Garcia *et al.*, 1982) but specifically is not restricted to FIR models. They argue that the use of transfer function models (or state space models with an estimator) in MPC realigns the model state on noisy data, hence giving poor predictions (Rossiter *et al.*, 2001). FIR models avoid this shortcoming by basing the predictions almost entirely on past input information with only an output correction to avoid offset, however they introduce bias errors due to truncation.

The independent (internal) model can overcome the shortcomings of the FIR models (truncation, large number of parameters) by being transfer function or state-space based while retaining the advantages of low prediction sensitivity to noise (by being based on inputs not outputs). It has been shown in an earlier paper (Rossiter *et al.*, 2001) that the structure of the prediction equations is such that one would expect the use of an independent model to give more reliable predictions than realigned models, that is less sensitive to noise. However, that paper did not consider the impact of such prediction errors on the resulting closed-loop control law. Hence the purpose of this paper is twofold.

- To show how predictive control laws are derived with realigned models and independent models and thus to derive the sensitivity functions of the closed-loop in each case.
- To demonstrate by way of example how sensitivity varies significantly depending on whether one uses the realigned model or the independent model.

The paper finishes with a discussion.

2. BACKGROUND AND CONTROL LAW STRUCTURE

In this section we give only a quick overview of how to compute the control laws; the reader is refered to the literature (e.g. (Clarke *et al.*, 1987)) for more details. As our only aim is to derive the sensitivity functions, we will not dwell on the most efficient realisation of these laws, but rather a convenient form that can be expressed using Z-transforms. The notation adopted is GPC for GPC with no T-filter (Yoon *et al.*, 1995), GPCT for GPC with a T-filter and GPCI for GPC with an independent model. Note GPC is equivalent to GPCT with $T(z) = 1$.

2.1 *Conventional GPC with a T-filter*

In GPC (Clarke *et al.*, 1987) the aim is, at each sample, to minimise a cost function of the form

$$J = \|W_y(R - Y)\|_2^2 + \|W_u \Delta U\|_2^2 \qquad (1)$$

where W_y, W_u are weighting matrices, Y is a vector of output predictions, R is a vector of future set points [1] and ΔU is a vector of future control increments. Assuming a model of the form

$$A(z) y_k = B(z) u_k + T(z) \frac{\zeta}{\Delta} \qquad (2)$$

where y, u, ζ are outputs, inputs and an unknown zero mean random disturbance respectively. Define filtered values $\tilde{y} = y/T$, $\tilde{u} = u/T$, then it is easy e.g. (Rossiter, 1993) to form predictions

$$Y = H\Delta U + P\Delta\tilde{U}_{past} + Q\tilde{Y}_{past}; \qquad (3)$$

where $\Delta\tilde{U}_{past}$, $\tilde{Y}_{past}$ are vectors of past filtered input/output values respectively and $Y, \Delta U$ are vectors of unfiltered future output/input predictions respectively. Substitution of (3) into (1), minimisation w.r.t ΔU and selection of only the first block element of the optimal ΔU gives rise to a control law of the form

$$\Delta u_k = -\check{D}_k \Delta\tilde{U}_{past} - \check{N}_k \tilde{y}_{past}$$
$$\begin{cases} \check{D}_k = [I, 0, 0, ...][H^T H + W_u]^{-1} H^T W_y P \\ \check{N}_k = [I, 0, 0, ...][H^T H + W_u]^{-1} H^T W_y Q \end{cases} \qquad (4)$$

This is easily rearranged into a more conventional form in terms of z-transforms:

$$D_k(z)\Delta\tilde{u} = -N_k(z)\tilde{y}$$
$$D_k(z) = T(z) + [I, \check{D}_k] \begin{bmatrix} I \\ z^{-1}I \\ \vdots \end{bmatrix}; \; N_k(z) = \check{N}_k \begin{bmatrix} I \\ z^{-1}I \\ \vdots \end{bmatrix} \qquad (5)$$

The argument $.(z)$ is dropped hereafter to improve clarity. Note that the corresponding D_k, N_k for GPC and GPCT will be different as T affects the definition of P, Q (Clarke *et al.*, 1987).

2.2 *GPC with an independent model*

In this case the predictions are slightly different from that adopted in (3). Simulate an independent model $\hat{A}\hat{y} = \hat{B}u$ in parallel with the plant [2] and use the measured offset $y_k - \hat{y}_k$ to correct predictions based on this model. The predictions are

$$Y = H\Delta U + \hat{P}U_{past} + \hat{Q}\hat{Y}_{past} + L(y_k - \hat{y}_k) \quad (6)$$

where $\hat{Y}_{past}$ is based on past outputs of the model (not process), U_{past} is past absolute inputs (not increments) and for example in the SISO case, L is a vector of ones. Substitution into (1) and minimisation w.r.t. ΔU gives a control law of the form

$$\Delta u_k = -\hat{D}_k U_{past} - \hat{N}_k \hat{y}_{past} - \hat{M}_k y_k$$
$$\begin{cases} \hat{D}_k = [I, 0, 0, ...][H^T H + W_u]^{-1} H^T W_y \hat{P} \\ \hat{N}_k = [I, 0, 0, ...][H^T H + W_u]^{-1} H^T W_y \hat{Q} - M_k \\ \hat{M}_k = [I, 0, 0, ...][H^T H + W_u]^{-1} H^T W_y L \end{cases} \qquad (7)$$

Again, as in (5), this is easily rearranged into a neat form based on Z-transforms:

$$D_k(z)u = -N_k(z)\hat{y} - M_k(z)y; \quad \hat{A}\hat{y} = \hat{B}u \quad (8)$$

where $D_k(z)$, $N_k(z)$, $M_k(z)$ depend upon the parameters in $\hat{D}_k$, $\hat{N}_k$, $\hat{M}_k$. Then (8) can be simplified to

$$D_i u = -M_k y; \quad D_i = [D_k + N_k \hat{A}^{-1}\hat{B}] \qquad (9)$$

2.3 *Summary of control laws*

The z-transform representation of the control laws for GPC, GPCT and GPCI are summarised in table 1. Again, it is emphasised that D_k, N_k for GPC and GPCT will be different in general.

Table 1: Control laws

GPC	$D_k \Delta u = -N_k y$
GPCT	$\frac{D_k}{T}\Delta u = -\frac{N_k}{T} y$
GPCI	$D_i u = -M_k y$

[1] Hereafter as it is not relevant, a zero setpoint is assumed.

[2] Clearly one chooses $\hat{A} = A$, $\hat{B} = B$ if possible

3. COMPUTATION OF SENSITIVITY

The following assumptions are made (different assumptions will give rise to different sensitivity functions): (i) the sensitivity to the two signals noise and disturbance captures a good range of possibilities and (ii) the plant model is given as

$$A(z)y_k = B(z)u_k + d_k; \quad w_k = y_k + v_k \qquad (10)$$

where d_k is a disturbance signal, v_k is output measurement noise and w_k is the measured output. The controller acts on w_k not on y_k.

The sensitivity functions, that is the transferences from d_k, v_k to y, u can be computed in a straightforward manner by solving for y_k, u_k in terms of d_k, v_k using eqn.(10) and the control laws of table 1. The notation adopted is that sensitivity of x wr.t f is denoted S_{fx}. The sensitivities for no parameter uncertainty are presented in tables 2-5.

Table 2: Output sensitivity to noise	
GPC	$S_{vy} = [A + B(D_k\Delta)^{-1}N_k]^{-1}A$
GPCT	$S_{vy} = [A + B(D_k\Delta)^{-1}N_k]^{-1}A$
GPCI	$S_{vy} = [A + BD_i^{-1}M_k]^{-1}A$

Table 3: Output sensitivity to disturbances	
GPC	$S_{dy} = [A + B(D_k\Delta)^{-1}N_k]^{-1}$
GPCT	$S_{dy} = [A + B(D_k\Delta)^{-1}N_k]^{-1}$
GPCI	$S_{dy} = [A + BD_i^{-1}M_k]^{-1}$

Table 4: Input sensitivity to noise	
GPC	$S_{vu} = [D_k\Delta + N_kA^{-1}B]^{-1}N_k$
GPCT	$S_{vu} = [D_k\Delta + N_kA^{-1}B]^{-1}N_k$
GPCI	$S_{vu} = [D_i + M_kA^{-1}B]^{-1}M_k$

Table 5: Input sensitivity to disturbances	
GPC	$S_{du} = [D_k\Delta + N_kA^{-1}B]^{-1}N_kA^{-1}$
GPCT	$S_{du} = [D_k\Delta + N_kA^{-1}B]^{-1}N_kA^{-1}$
GPCI	$S_{du} = [D_i + M_kA^{-1}B]^{-1}M_kA^{-1}$

The sensitivity to multiplicative model uncertainty for the nominal case is

$$S_g = [I + KG]^{-1}KG; \quad G = A^{-1}B \qquad (11)$$

The different controllers to be subsituted into this expression are summarised in table 6.

Table 6: Nominal control laws	
GPC	$K = [D_k\Delta]^{-1}N_k$
GPCT	$K = [D_k\Delta]^{-1}N_k$
GPCI	$K = D_i^{-1}M_k$

4. EXAMPLES

The effectiveness of the independent model approach is illustrated by way of examples. The information will be presented as Bode plots of the sensitivity functions as this shows the variation of sensitivity over the whole frequency range. Separate figures will give the sensitivity functions

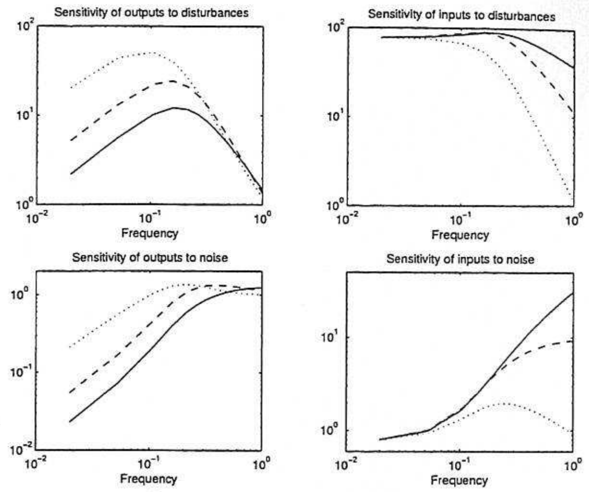

Fig. 1. Closed-loop sensitivies to noise and disturbances, example 1

S_{dy}, S_{vy}, S_{du}, S_{vu}, S_g with notation as in table 7. The frequency range is 0 to π.

Table 7: Notation of plots	
GPC	solid line
GPCT	dashed line
GPCI	dotted line

For disturbances, it might be argued that the focus should be on output sensitivity only for low frequencies as one would not normally expect high frequency disturbances. However, the integral action will deal with this which shifts the focus back to the transients in disturbances, that is high frequencies. Also one would expect noise to be mostly high frequency and hence one should focus mainly on the high frequency range of these bode plots.

4.1 *Example 1*

This is a SISO example. The controller is designed with $n_y = 30$, $n_u = 3$, $W_u = 1$. The corresponding sensitivity functions are displayed in figures 1,2.

$$\begin{aligned} A(z) &= 1 - 1.8z^{-1} + 0.81z^{-2} \\ B(z) &= 0.01z^{-1} + 0.003z^{-2} \\ T(z) &= 1 - 0.8z^{-1} \end{aligned} \qquad (12)$$

Clearly using an independent model has much reduced the input sensitivity to noise and disturbances (as well as multiplicative uncertainty) in the high frequency range. This could be construed as a good thing as one does not want the inputs chasing noise as can happen with realigned models (e.g. (Rossiter *et al.*, 2001)). The output variance is also smaller for high frequencies. The price is a larger variance of output at intermediate frequencies where one might consider noise/disturbances are less likely to occur. Clearly GPCT is better than GPC and more interestingly (as discussed in

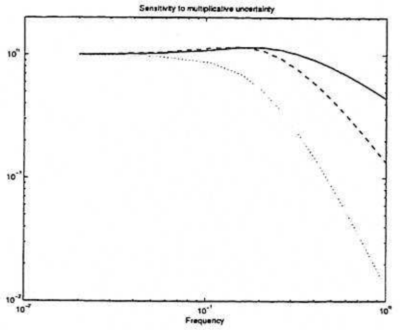

Fig. 2. Closed-loop sensitivity to multiplicative uncertainty, example 1

(Yoon *et al.*, 1995)), if $T(z) = A(z)$ the sensitivity plots of GPCT exactly replicate those of GPCI. The choice $T = A$ however may not always be a wise choice of filter. GPCI also has better robustness to model uncertainty (figure 2).

4.2 *Example 2*

This is a 2 by 2 plant with reasonably large interactions in the step response characteristics. However, the step responses are smooth with non-minimum phase characteristics.

$$A(z) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} -1.3 & 0 \\ 0 & -0.7 \end{bmatrix} z^{-1} + \begin{bmatrix} 0.4 & 0 \\ 0 & -0.18 \end{bmatrix} z^{-2} \tag{13}$$

$$B(z) = \begin{bmatrix} 0.5 & 0.2 \\ -0.6 & 1 \end{bmatrix} z^{-1} + \begin{bmatrix} -0.5 & 0.3 \\ 0.3 & 1 \end{bmatrix} z^{-2} + \begin{bmatrix} 2 & 0.5 \\ 0.6 & 0.5 \end{bmatrix} z^{-3} \tag{14}$$

With $T(z) = 1 - 0.8z^{-1}, n_u = 5,\ n_y = 30, W_u = 1$. The corresponding closed-loop sensitivity functions are plotted in figures 3-7 where the subplot position corresponds to the matrix position, that is row 'i', col 'j' of the figure corresponds to $S_{i,j}$. Here we see a similar trend to example 1. The independent model algorithm has the lowest input and output sensitivity for high frequencies, but poorer at intermediate frequencies. For robustness to multiplicative uncertainty the case is less clear cut though GPCI is clearly better than GPC.

4.3 *Example 3*

This is a 3 by 3 plant with large interation.

$$A(z) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} -0.88 & -0.16 & 0 \\ 0.32 & -0.96 & -0.8 \\ 0 & 0 & -0.72 \end{bmatrix} z^{-1} + \begin{bmatrix} 0.112 & 0.08 & 0 \\ 0 & 0.136 & 0.08 \\ 0.16 & 0 & -0.112 \end{bmatrix} z^{-2} + \begin{bmatrix} 0.064 & 0.016 & 0 \\ -0.08 & 0.072 & 0.24 \\ 0.16 & 0 & 0.0192 \end{bmatrix} z^{-3} \tag{15}$$

$$B(z) = \begin{bmatrix} 0.5 & 0.2 & -.5 \\ 2 & 0 & 0.3 \\ 0 & 0.9 & -0.4 \end{bmatrix} z^{-1} + \begin{bmatrix} 1 & 2 & 1 \\ -0.8 & 0.6 & 0.5 \\ 1 & 0.3 & 0.5 \end{bmatrix} z^{-2} \tag{16}$$

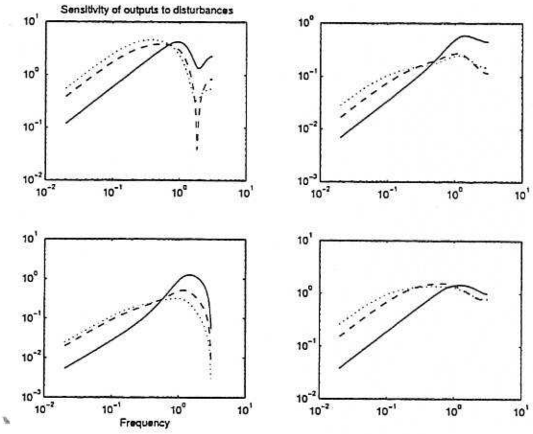

Fig. 3. Output sensitivity to disturbances, ex. 2

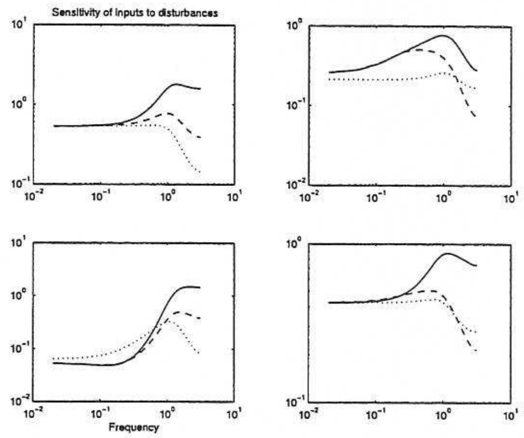

Fig. 4. Input sensitivity to disturbances, ex. 2

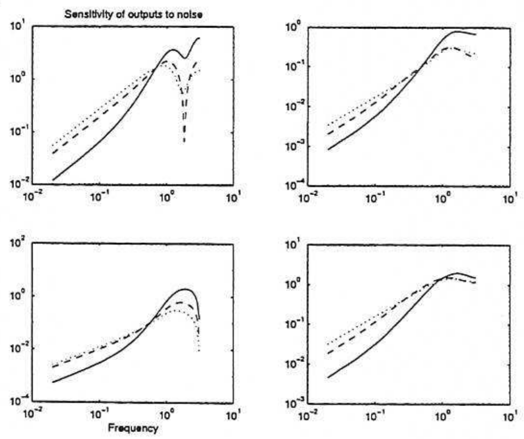

Fig. 5. Output sensitivity to noise, example 2

With $T(z) = 1 - 0.8z^{-1}, n_u = 15,\ n_y = 30, W_u = 1$. The corresponding closed-loop sensitivity functions are plotted in figures 8-12. Here we see that the results are less conclusive but one can still see a preference for the independent model at high frequencies.

4.4 *Discussion*

The purpose here was to compare 'simple' approaches to minimising sensitivity, without the

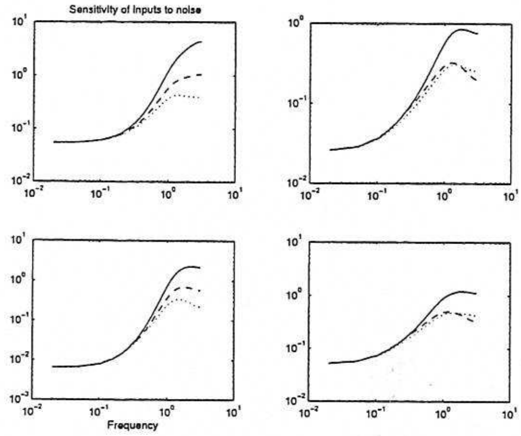

Fig. 6. Input sensitivity to noise, example 2

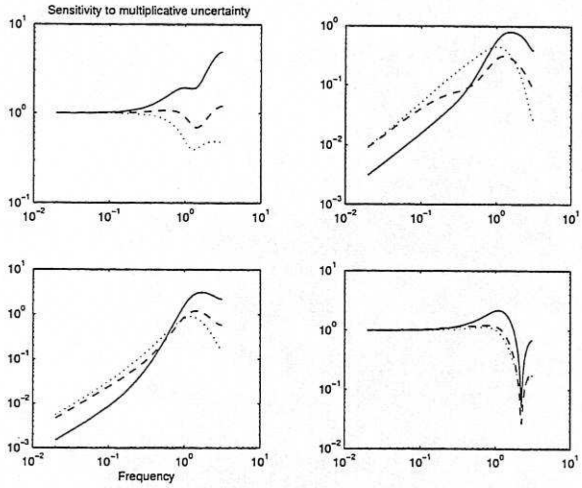

Fig. 7. Sensitivity to multiplicative uncertainty, example 2

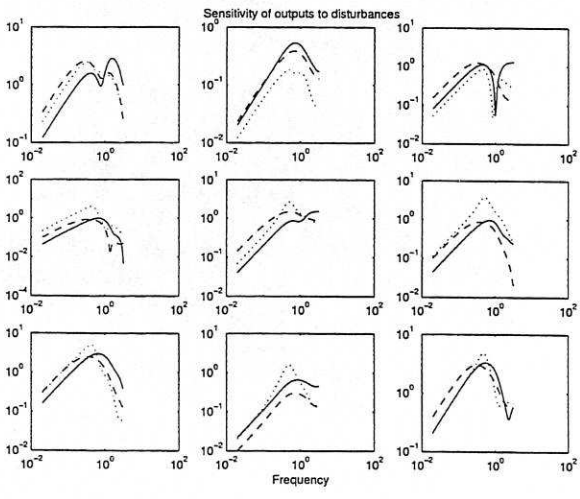

Fig. 8. Output sensitivity to disturbances, ex. 3

use of involved robust control theory. For the examples shown GPCI has outperformed GPC and also on average outperformed GPCT. One might argue that with a SISO case, one can always choose $T = A$ (making GPCT equivalent to GPCI) however this may not be desirable in general. Moroever, for multivariable systems the

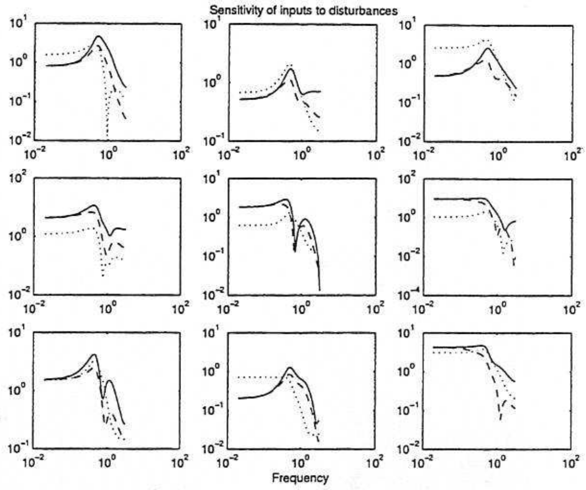

Fig. 9. Input sensitivity to disturbances, ex. 3

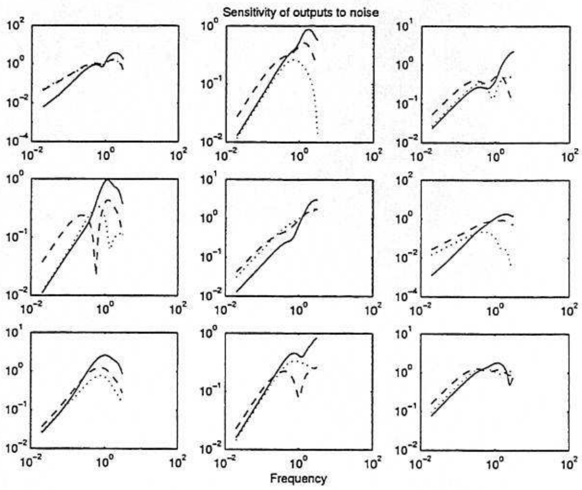

Fig. 10. Output sensitivity to noise, example 3

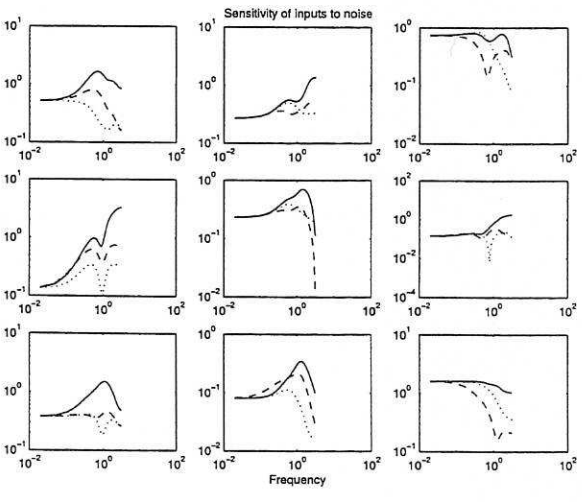

Fig. 11. Input sensitivity to noise, example 3

guidelines for choosing T are much less clear cut and then one can see that GPCI is likely to be the best. Conversely, with GPCI, there is one less control parameter to design (that is no T) and without loss of performance. Hence at the very least it is worth considering the use of an IM at the outset of a control design. To compare

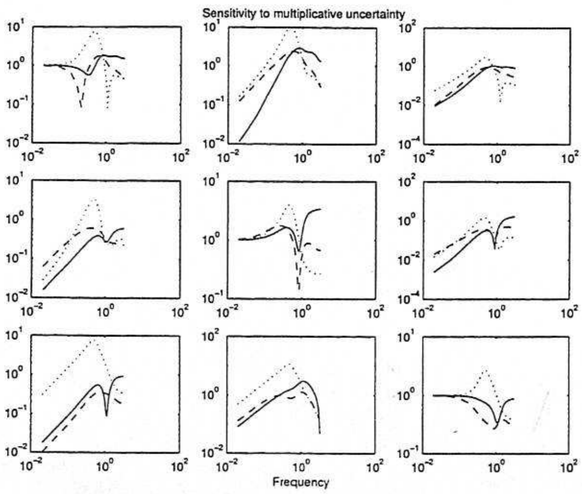

Fig. 12. Sensitivity to multiplicative uncertainty, example 3

the sensitivities arising from the use of different models is relatively trivial in simulation and there may be much to be gained. An offline case by case comparison is essential in general, because one cannot generalise. The conclusions will change for different models and moreover if the disturbance model differs from that in (10).

One might argue that, at least in the unconstrained case, that the Youla parameterisation can be used to adapt all controllers (for realigned and IM models) with the same nominal performance to have similar robustness and to give a convenient decoupling of performance objectives from robustness objectives (e.g. see (Kouvaritakis *et al.*, 1992),(Garcia *et al.*, 1982) for details). Hence does the model choice really matter ? To counter this, it should be emphasised that one strength of predictive control is the ability to do online constraint handling. Systematic extension of sensitivity functions to this case is non trivial and scenario dependent. However, to date no simple and systematic means of augmenting robustness, for instance via the Youla parameterisation, has been developed for the constraint handling case. Some ideas are presented in (Rossiter *et al.*, 1998) but need further work. In the meantime, one can argue that if the prediction structure gives low sensitivity in the nominal case, this is likely to carry over to the constrained case in general.

5. CONCLUSION

It has been shown that the typical academic practice of using realigned models in predictive control can lead to poor sensitivity with respect to noise. This is often corrected by the design of a T-filter, however such a process is not systematic beyond the guideline of using a low-pass filter with poles near those of the plant. Here some examples have shown that the alternative proposal of using an independent model (internal model) gives low sensitivity without the need for an extra design parameter. It is expected that such benefits will transfer to the constrained case where robust design approaches are not easily applicable.

6. REFERENCES

Clarke, D.W., C. Mohtadi and P.S. Tuffs (1987). Generalised predictive control, Parts 1 and 2, *Automatica*, 23, 137-160

Bemporad, A., A. Casavola and E. Mosca (1997). Non-linear control of constrained linear systems via predictive reference managment, *IEEE Trans. AC*, 42, 3, 340-349

Cutler, C.R. and B.L. Ramaker (1980), Dynamic matrix control - a computer control algorithm, *Proc. ACC*, San Fransisco

Garcia, C.E. and M. Morari (1982), Internal Model control 1. A unifying review and some new results, *I&EC Process Design and Development*, 21, 308-323

Garcia, C.E. , D.M. Prett and M. Morari (1989), Model predictive control: theory and practice, a survey, *Automatica*, 25, 335-348

Kouvaritakis, B., J.A. Rossiter and A.O.T.Chang (1992), Stable Generalized predictive control: an algorithm with guaranteed stability, *Proc IEE*, 139, 349-362

Richalet, J., A. Rault, J.L. Testud and J. Papon (1978), Model predictive heuristic control: applications to industrial processes, *Automatica*, 14, 413-428

Rossiter, J.A., M.J. Rice and B.Kouvaritakis (1998). A numerically robust state-space approach to stable predictive control strategies, *Automatica*, 38, 65-73

Rossiter, J.A., M.J. Rice, J.Schuurmanns and B.Kouvaritakis (1998). A computationally efficient constrained predictive control law, *Proc. ACC 98.*

Rossiter, J.A. (1993), Notes on multi-step ahead prediction based on the principle of concatenation, *Proc. IMechE*, 207, 261-263

Rossiter, J. A., and B. Kouvaritakis (1998), Youla parameter and robust predictive control with constraint handling, *Workshop on Non-linear Predictive control*, Ascona

Rossiter, J.A. and J. Richalet (2001), Re-aligned models for prediction in MPC: a good thing or not ? *APC6* (York, UK)

Scokaert, P.O.M. and J. B. Rawlings (1996). Infinite horizon linear quadratic control with constraints, *Proc. IFAC'96*, **vol. M**, 109-114

Yoon, T-W., and D.W. Clarke (1995), Observer design in receding horizon predictive control, *IJC*, 61, 171-191