IMPROVING THE PERFORMANCE OF DUAL RATE CONTROL IN THE ABSENCE OF A FAST RATE MODEL

J.A. Rossiter*, **T. Chen**+, **Sirish L. Shah**+ *

* *Sheffield University, UK. S1 3JD,* + *University of Alberta, Canada. T6G 2V4*

Abstract: This paper discusses issues related to model predictive control for dual rate systems in which the input is updated faster than the output. When fast-rate models are not available, a simple and effective approach is proposed for receding horizon control which uses just a single prediction equation to generate fast control moves.

Keywords: Multirate control, performance, predictive control, multirate systems.

1. INTRODUCTION

In many real systems the output can only be measured at a relatively slow sampling rate and certainly much slower than an achievable input sampling rate. In such cases it would seem wise to update the system input at a fast rate (FR) in order to improve performance. However, it is not obvious how a control law should be designed for such a case as most conventional design methods and models assume a single sampling rate. Moreover, as new information (that is measurements) come only at the slow rate (SR), one might wonder if there can be any benefits to updating control decisions at a FR as during intersample [1] periods. That is how can one improve on decisions without new information? This might suggest that one could just as well design a slow rate control law but that goes against the intuitive feel that surely one can get benefits by faster input updates if that is possible. The aim of this paper is to look at such issues in a given context, to be defined, and hence give some answers to what can be achieved.

[1] Intersample is used here to mean in between output measurements.

Many pieces of work on multi-rate systems assume that one has access to an underlying fast rate model (Lee *et al*, 1992). However such an assumption is simplistic as only SR output measurements are available and this places limitations on the identification. Recent work (Li *et al*, 2001) has shown that in some cases a FR model can be deduced but this is still an area of active research and the robustness and limitations of such models are still not well understood. Hence in this paper we take the more pragmatic approach that only a slow rate model is available. However such a model can nevertheless show the dependence on FR input sampling through the mechanism of lifting (Khargonekar *et al*, 1985; Kranc *et al*, 1957) whereby a multi-rate SISO system can be represented by a single rate MIMO system. This will be explained in more detail in section 2.

Given only a SR model there is no obvious mechanism for estimating outputs at intersample points. One common method is to use an internal model (Garcia *et al*, 1982) to estimate intersample outputs and use these in lieu of the actual measurements to allow the implementation of a FR control law. Here it is assumed that an internal model is not available so one might think that the best one can do is to use a SR control law based on lifting. The weakness of this is illustrated in (Rossiter *et al*, 2003).

This paper builds on (Rossiter *et al*, 2003) in that it proposes a simple solution without recourse to infinite horizons, that is it demonstrates how to use a slow rate lifted model and yet update the control law (optimisation) at the fast rate. We note that similar concepts were adopted in (Pan *et al*, 2003). One should also emphasise that a further aim of this paper is to propose only those solutions which gives significant improvements in performance for a fixed (small) computational load; hence the control horizon is restricted to small values.

The paper structured is: Section 2 gives background to multi-rate systems, lifting and predictive control; Section 3 discusses alternative algorithms and section 4 contains numerical comparisons.

2. BACKGROUND

2.1 Lifting for dual rate systems

Consider a system where the input u is updated every T seconds and the output y every $mT = T_m$ seconds. Use the index k for the slow sample rate so that y_k is the output at the k^{th} slow sampling period and use the index l for intersample periods so that $u_{k,l}$ is the l^{th} intersample value of the input during the k^{th} period. One could consider this as a single rate system if the intersample inputs $u_{k,l}$ were grouped as follows:

$$U_k = \begin{bmatrix} u_{k,1} \\ u_{k,2} \\ \vdots \\ u_{k,m} \end{bmatrix} \tag{1}$$

where if $u_{k,1}$ is implemented synchronous with the measurement of y_k, then $u_{k,i}$ is implemented $(i-1)T$ seconds later [2]. Then, using the z-transform operator z to denote the time-delay operator at the slow sample rate T_m, a SISO system could be represented as a MISO system

$$y(z) = G(z)U(z) = [g_1(z)\ g_2(z)\ \ldots\ g_m(z)]U(z) \tag{2}$$

where the output is still single dimensional but the input U is m dimensional. A state space model representation is:

$$x_{k+1} = Ax_k + BU_k; \quad y_k = Cx_k \tag{3}$$

[2] For convenience this paper will often use the equivalence $u_{k,m+i} = u_{k+1,i}$

The key point to remember here is that there is implied timing within the components of U_k. As a consequence, when designing for multi-rate systems, care has to be taken in the set up of the control law to respect the so called causality constraint (Chen *et al*, 1994; Meyer *et al*, 1990; Sheng *et al*, 2002). That is one cannot have current inputs dependent on future inputs. In this paper, because we restrict ourselves to dual rate systems, causality is not an issue.

2.2 Predictive control

For the purposes of this paper the illustrations will be based on a generalised pedictive control (GPC, (Clarke *et al*, 1987)) algorithm. Future work will consider algorithms with guaranteed stability (e.g.(Kouvaritakis *et al*, 1992; Scokaert *et al*, 1998)) but it should be noted that the means of performing such an extension is not necessarily obvious or straightforward for MR systems.

As predictive control (MPC) is by now well known, here only the key points are presented. First there is a need for prediction equations. For state space models (3) these can take the form:

$$\underset{\rightarrow k+1}{y} = P_x x_k + H\underset{\rightarrow k}{U}$$
$$\underset{\rightarrow k+1}{y} = \begin{bmatrix} y_{k+1} \\ y_{k+2} \\ \vdots \\ y_{k+n_y} \end{bmatrix} \quad \underset{\rightarrow k}{U} = \begin{bmatrix} U_k \\ U_{k+1} \\ \vdots \\ U_{k+n_y-1} \end{bmatrix} \tag{4}$$

where x_k is the state, y_k the output and U_k the input, all at sampling instant k and P_x, H are defined as:

$$P_x = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{n_y} \end{bmatrix}; \ H = \begin{bmatrix} B & 0 & \ldots & 0 \\ AB & B & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ A^{n_y-1}B & A^{n_y-2}B & \ldots & B \end{bmatrix} \tag{5}$$

Remark 2.1. It is usual to limit the number of d.o.f. in the optimisation of J, hence typically one will minimise over the first n_u components of $\underset{\rightarrow k}{U}$ (that is $u_{k,1}, ..., u_{k,n_u}$) and assume that the remaining components (that is $u_{k,n_u+1}, ..., u_{k+n_y-1,m}$ are given by u_{ss}, the expected steady-state value of u. Hence one partitions the part of the prediction equation (4) dependent on future inputs as follows:

$$H\underset{\rightarrow k}{U} = \underbrace{[H_f|H_{ff}]}_{H} \left.\begin{bmatrix} \underset{\rightarrow f}{U} \\ \hline Lu_{ss} \end{bmatrix}\right\} \underset{\rightarrow k}{U}; \ \underset{\rightarrow f}{U} = \begin{bmatrix} u_{k,1} \\ \vdots \\ u_{k,n_u} \end{bmatrix} \tag{6}$$

where L is a vector of ones.

For convenience and to avoid unnecessary details here, one can assume $u_{ss} = 0$ in the initial design and then introduce offset free tracking in a later stage. Hence the term $H_{ff} L u_{ss}$ is ignored.

In GPC it is usual to minimise a 2-norm performance index. A typical choice for the regulation problem is is

$$\min_{\underset{\to f}{U}} \quad J = \|\underset{\to}{y}\|_2^2 + \lambda \|\underset{\to f}{U}\|_2^2 \tag{7}$$

where λ is an input weight. Subsituting predictions (4) into cost (7) and minimising wrt to $\underset{\to f}{U}$ gives the optimal set of predicted future inputs as

$$\begin{aligned} \underset{\to f}{U} &= -\underbrace{[H_f^T H_f + \lambda I]^{-1} H_f^T P_x}_{K} x_k \\ &= -K x_k \end{aligned} \tag{8}$$

In order to introduce tracking and to allow for offset free disturbance rejection (Muske *et al*, 1993) the control law is implemented as

$$\begin{aligned} \underset{\to f}{U} - L_{n_u} u_{ss} &= -K[x_k - x_{ss}] \\ &\Downarrow \\ \underset{\to f}{U} &= -K x_k + f_k; \end{aligned} \tag{9}$$

where $f_k = K x_{ss} + L_{n_u} u_{ss}$, L_{n_u} is an n_u vector of ones and $u_{ss},\ x_{ss}$ are the expected steady-state input and state giving no steady-state offset, that is $r = y,\ u = u_{ss},\ x = x_{ss}$ are consistent (r is the set point). The reader is refered to (Muske *et al*, 1993) for details of how to estimate $x_{ss},\ u_{ss}$ and here we note only that it depends on state estimation and the set point but is otherwise straightforward.

In GPC it is usual to take only the first component $u_{k,1}$ of the optimal control trajectory $\underset{\to f}{U}$ and then recompute the optimum at each sampling instant. For a dual rate system one would implement the first component of $\underset{\to k}{U}$, that is U_k which the reader will recall actually has m seperate moves spread over for the intersample period. Should $n_u < m$, then, as mentioned in remark 2.1, $\underset{\to f}{U}$ is padded with u_{ss} $m - n_u$ times as this is consistent with the prediction assumption. Hence the first m control moves, those defined in U_k, are given from

$$U_k = \begin{bmatrix} -K x_k + f_k \\ u_{ss} \\ u_{ss} \\ \vdots \end{bmatrix} \tag{10}$$

3. AN ALTERNATIVE APPROACH TO GPC FOR DUAL RATE SYSTEMS

3.1 The important property of predictions in GPC

Although very succesful in practice (DMC (Cutler *et al*, 1980) is essentially the same algorithm), GPC has one major weakness – there is no apriori stability guarantee for the general case. Although solutions to this exist (e.g. (Kouvaritakis *et al*, 1992; Scokaert *et al*, 1998)), it is pertinent to understand from an intuitive rather than mathematical viewpoint why problems can arise. The answer is very simple and given next.

First let us assume no model mismatch so that the predictions are exact. Nevertheless in GPC the closed-loop behaviour and the open-loop predictions used in the minimisation of J do not match. That is one is minimising predicted performance with a certain assumption (that the inputs become fixed after n_u steps) when what actually happens is something quite different. It is this mismatch that causes the problem, that is the optimisation is illposed as it is based on erroneous assumptions that do not match reality [3]. Whether this causes stability or performance problems depends upon how great the mismatch is. Fortunately for many processes the resulting closed-loop behaviour is quite close to the predictions and good performance results.

Secondly let us look in more detail at the prediction assumption that the input assume a fixed value after n_u steps. This assumption is taken to ensure computational tractability but clearly does not match the expected closed-loop behaviour where the input would be expected to change continuously, at least over the process rise time. MPC strategies deal with this mismatch by continuously updating the assumption, that is the so called receding horizon (RH). At each sampling instant a new d.o.f. (e.g. u_{k+n_u}) is introduced and the optimum input trajectory is recomputed.

So in summary, GPC does have a weakness in the prediction assumption adopted. However in practice, use of the receding horizon concept is sufficient to recover reasonable closed-loop performance. There are of course well documented exceptions where this mismatch can have catastrophic consequences. The next section will demonstrate that multi-rate systems are one area where this weakness cannot be overlooked.

[3] The work of (Scokaert *et al*, 1998) provides an ideal solution when tractable as it removes this mismatch.

3.2 Does the receding horizon work well for dual rate systems ?

The problem with using the receding horizon to rectify what could be considered as a limited prediction class is that it can work well given a fast update of the optimisation, that is where the receding horizon concept is deployed every sampling instant. However, for dual rate systems the control law (10) is updated only every m^{th} sample (of the faster sample rate) so one has to live with the previously computed optimum for m samples before it can be improved. In this case if the prediction class is not close to the desired closed-loop dynamic, the RH update is too infrequent to correct the behaviour. This is particularly marked when $m > n_u$, an observation which the example section will illustrate.

Remark 3.1. This paper takes the assumption that n_u should be small, say 1 or 2 as that is common in practical implementations. Clearly for large n_u the issues are much less significant but the implied computational load may be considered intractable for some cases.

3.3 Improving the GPC algorithm for lifted systems

The normal expectation for GPC implementations is to use the receding horizon concept at the fast sampling rate, that is at the sample rate at which the input is updated. Here it will be shown how that can be achieved effectively in the lifted framework despite the lack of fast rate output measurements and the lack of a fast rate model. It is noted that the existence or not of new output measurements does not affect nominal performance, as for the certain case predictions are exact. In fact output measurements are needed primarily to give robust performance and disturbance rejection.

The key issue in the receding horizon is the introduction of a new d.o.f. It will now be shown that in the dual rate scenario it is straightforward to introduce new d.o.f. at intersample periods and hence to update the predictions at a fast rate. Define the future input trajectory $\underset{\rightarrow k}{U}$ to have 3 parts, a past (that is inputs that have already been implemented), d.o.f. (that is available to the optimisation) and a steady state:

$$\underset{\rightarrow k}{U}^T = [U^T_{k|\text{past}}, U^T_{k|\text{d.o.f.}}, U_{k|ss}] \tag{11}$$

The structure of the components of input trajectory $\underset{\rightarrow k}{U}$ at the l^{th} intersample period is given as

$$\begin{aligned} U_{k|\text{past}} = U_{k,l|p} &= \begin{bmatrix} u_{k,1} \\ \vdots \\ u_{k,l-1} \end{bmatrix} \\ U_{k|\text{d.o.f.}} = U_{k,l|f} &= \begin{bmatrix} u_{k,l} \\ \vdots \\ u_{k,l+n_u-1} \end{bmatrix} \\ U_{k|ss} = U_{k,l|ss} &= \begin{bmatrix} u_{ss} \\ u_{ss} \\ \vdots \end{bmatrix} \end{aligned} \tag{12}$$

Hence at the (k, l) sampling instant (that is the l^{th} intersample period of the k^{th} sample period), the optimisation should be set up as follows:

$$\min_{U_{k,l|f}} \quad J = \|\underset{\rightarrow}{y}\|_2^2 + \lambda\|\underset{\rightarrow}{U}\|_2^2 \tag{13}$$

where again for convenience it is assumed that $u_{ss} = 0$. Partition H analogously to the partition of $\underset{\rightarrow k}{U}$ (which clearly depends upon l) and but ignore (ignore $U_{k,l|ss}$ (as $u_{ss} = 0$), so that

$$H\underset{\rightarrow k}{U} = H_{p|l}U_{k,l|p} + H_{f|l}U_{k,l|f} \tag{14}$$

Hence substituting (14) into (13) and minimising gives the optimal control law as

$$\begin{aligned} U_{k,l|f} &= -[H^T_{f|l}H_{f|l} + \lambda I]^{-1}H^T_{f|l} \\ &\qquad [P_x x_k + H_{p|l}U_{k,l|p}] \\ &= -K_l x_k + M_l U_{k,l|p} \end{aligned} \tag{15}$$

where $P_L = [H^T_{f|l}H_{f|l}+\lambda I]^{-1}H^T_{f|l}$, $K_l = P_l P_x$, $M_l = P_l H_{p|l}$. One can introduce tracking and offset free disturbance rejection as in (9) so that the optimal control trajectory is given from

$$\begin{aligned} U_{k,l|f} - L_{n_u}u_{ss} &= -K_l(x_k - x_{ss}) \\ &\qquad + M_l(U_{k,l|p} - L_l u_{ss}) \end{aligned} \tag{16}$$

where L_i is a i-dimensional vector of ones. As one is doing a fast RH update, only teh first element, that is $u_{k,l}$, is implemented.

Remark 3.2. Control law (16) is time varying within the intersample periods, that is the law depends upon l. This is an inevitable consequence of lifting (Khargonekar *et al*, 1985; Kranc *et al*, 1957; Sheng *et al*, 2002) but nevertheless it appears as a time invariant control law at the slow sampling rate. There can be problems with intersample ripple (Tangirala *et al*, 2001) due to this time varying nature but this should not occur in the presence of integral action which is deployed here.

Remark 3.3. The beauty of the approach proposed here is that it is based on a single prediction model

(4). The only change for intersample periods is how the matrix H is partitioned in (14).

4. NUMERICAL ILLUSTRATIONS

In this section it will be shown how the combination of lifting (which is necessary when a FR model does not exist) and small n_u render more typical GPC algorithms ineffective. By adopting the modification suggested in this paper one can recover the nominal performance to that achievable should the fast rate output measurements be available.

Consider the simple example with a fast rate state space model

$$x_{k,l+1} = \begin{bmatrix} 0.3 & 0.5 \\ 0.1 & 0.9 \end{bmatrix} x_{k|l} + \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix} u_{k|l}; \qquad (17)$$
$$y_k = [1 \quad 0] x_k$$

Recall that this model is given for the readers benefit but the assumption is made that only the equivalent model of form (3) is known. Assume that the output is sampled 4 times slower than the input, i.e. $m = 4$. The GPC algorithms of (10) and (16) are implemented for $n_u = 1, 2, 3, 4$ with $n_y = 8, \quad \lambda = 1$. The simulations are displayed in figures 1-4 for $n_u = 1, ..., 4$ respectively; circles and dotted lines are used for the fast RH algorithm of (16) and crosses and solid lines are used for the slow RH update algorithm of (10). The x-axis has units of the fast sample rate so new output measurements are given only every 4th sample. The corresponding closed-loop runtime costs are given in table 1.

RH Algorithm	$n_u = 1$	$n_u = 2$	$n_u = 3$	$n_u = 4$
Slow (eqn.(10))	3.18	2.65	2.33	2.14
Fast (eqn.(16))	2.21	2.17	2.14	2.11

Table 1: Closed-loop runtime costs J

It is clear both from the table and the figures that the proposed algorithm using a fast RH update based on an identical prediction model has vastly outperformed the nominal lifted algorithm when n_u is small, even though there has been no new output measurements. The limitation of the prediction assumption is very clear in figures 1-3 where it can be seen that the input moves to a poor default value, that is u_{ss}, during the later intersample periods. If one uses a fast RH law (as iin (16)) the negative effects of this poor assumption can be alleviated, but not if one uses a slow RH (as in 10). Naturally as n_u becomes larger (e.g. fig. 4) then the issue becomes less significant but there is a consequent increase in the online computational burden.

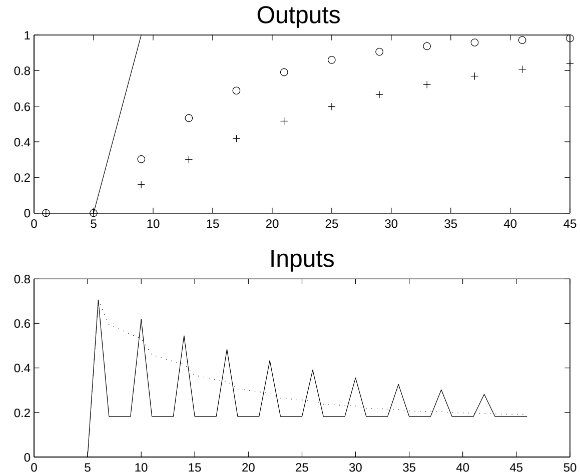

Fig. 1. Simulations with $n_u = 1$

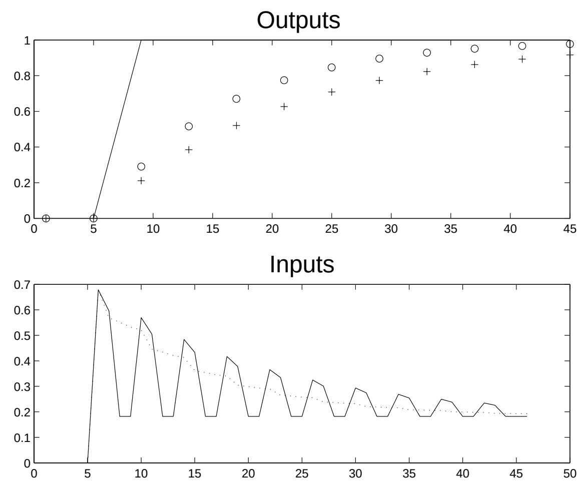

Fig. 2. Simulations with $n_u = 2$

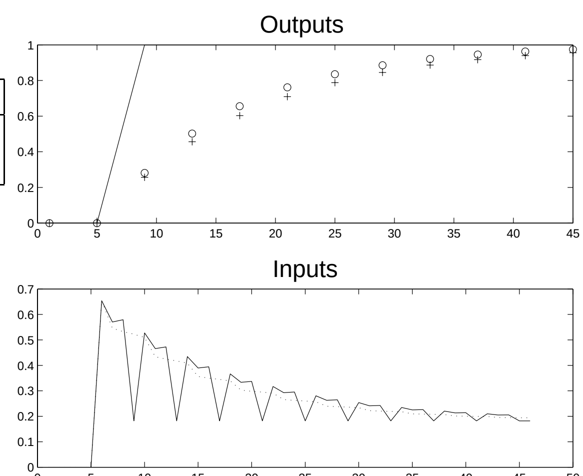

Fig. 3. Simulations with $n_u = 3$

5. CONCLUSIONS AND FURTHER WORK

This paper builds further on (Rossiter *et al*, 2003) which illustrated that in the context of dual rate systems, purely using lifting and a slow control law update, has severe and unnecessary limitations. It is shown here that even though no new obser-

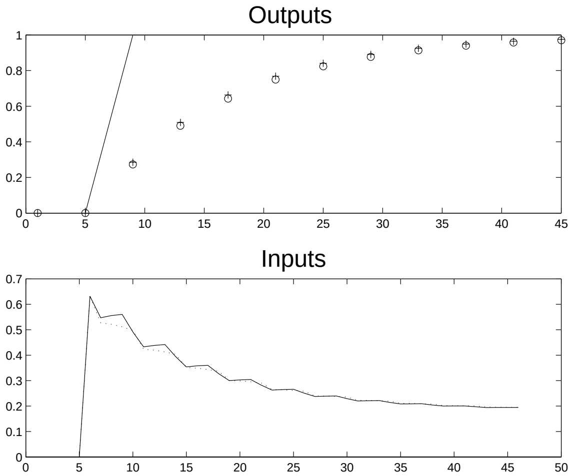

Fig. 4. Simulations with $n_u = 4$

vations are available during intersample periods, the predictive control algorithm is at its best only when used in a receding horizon sense. Hence this receding horizon property should be utilised to its maximum, even during intersample periods. Here a means of doing this has been presented which does not require more than one model (Pan*et al*, 2003) unlike other approaches (Sheng *et al*, 2002) which require different models for each intersample point. Rather it has been shown that one can simply use different partitions of a single prediction equation in order to find the m control laws most relevant to the m intersample periods. The benefits of this simple approach are evident from the examples.

Future work will look at the context of algorithms with guaranteed stability, e.g. (Scokaert *et al*, 1998), and investigate whether the same concept is equally applicable. Allied to this is the issue of constraint handling which was omitted here as secondary to the key contribution. Do constraints bring any new facets to the problem? Moreover there is a need to look at multi-rate systems which are not just dual rate as for these the partitioning of the intersample period is more complex. It should be re-emphasised here that the focus here has not been *what is the best one can do with arbitrary computing power?* but rather *what can one achieve with a relatively simple and undemanding algorithm?*

REFERENCES

Chen, T., and L. Qiu, $\mathcal{H}_\infty$ design of general multirate sampled-data control systems, *Automatica*, 30(7), pp. 1139-1152, 1994.

Cutler, C.R., and B.L. Ramaker, Dynamic matrix control - A computer control algorithm, *Proc. Joint Automatic Control Conference*, San Francisco, U.S.A., 1980.

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

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

Khargonekar, P.P., K. Poolla, and A. Tannenbaum, Robust control of linear time-invariant plants using periodic compensation, *IEEE Trans. AC*, 30, pp. 1088-1096, 1985.

Kouvaritakis, B., J.A. Rossiter, and A.O.T. Chang, Stable generalized predictive control: an algorithm with guaranteed stability, *IEE Proc. Pt. D*, 140, pp. 364-372, 1992.

Kranc, G.M., Input output analysis of multi-rate feedback system, *IRE Trans, Automatic Control*, 3, pp. 21-28, 1957.

Lee, J.H., and M. Morari, Robust inferential control of multi-rate sampled data systems, *Chem. Eng Sci.*, 47, pp. 865-885, 1992.

Li, D., S.L. Shah, and T. Chen, Identification of fast-rate models from multirate data, *Int. Journal of Control*, 74(7), pp. 680-689, 2001.

Meyer, D.G., A parametrization of stabilizing controllers for multirate sampled-data systems, *IEEE Trans. AC*, 35, pp. 233-236, 1990.

Muske, K.R., and J. R. Rawlings, Model predictive control with linear models, *AIChE Journal*, 39(2), pp. 262-287, 1993.

Pan, Y., and J.H. Lee, Identification and control of processes with periodic operations or disturbances, INd Eng. Chem. Res., 42, 1938-1947, 2003.

Rossiter, J.A., T.Chen and S.L.Shah, Developments in Multi-rate Predictive Control, ADCHEM03, 2004

Scokaert, P.O.M., and J.B. Rawlings, Constrained linear quadratic regulation, *IEEE Trans AC*, 43(8), pp. 1163-1168, 1998.

Sheng, J., T. Chen, and S.L. Shah, Generalized predictive control for non-uniformly sampled systems, *Journal of Process Control*, (12), pp. 875-885, 2002.

Tangirala, A.K., D. Li, R.S. Patwardhan, S.L. Shah, and T. Chen, Ripple free conditions for lifted multi-rate control systems, *Automatica*, 37, pp. 1637-1645, 2001.