A DYNAMIC PLS FRAMEWORK FOR CONSTRUANED MODEL PREDICTIVE CONTROL

S. Lakshminarayanan  Rohit S. Patwardhan  Sirish L. Shah  K. Nandakumar

Department of Chemical and Materials Engineering, University of Alberta, Edmonton, Canada T6G 2G6. Email: Sirish.Sham@UAlberta.Ca

Abstract. This paper demonstrates the constrained predictive control of linear multivariable systems based on models identified using a Dynamic Projection to Latent Structures (Partial Least Squares or PLS) algorithm. Though conventional control of systems using such models have been reported elsewhere (Kaspar and Ray (1992, 1993), Lakshminarayanan et al. (1996)), they are not suited for practical applications owing to their inability to handle constraints. A simple modification of the constraints provides a framework wherein the PLS based models can be incorporated in existing model based predictive control algorithms. The theory is supported using simulation as well as laboratory experiments.

Keywords. Model Predictive Control, Partial Least Squares, Constraint Mapping

1. INTRODUCTION

Model Predictive Control (MPC) schemes are gaining increasingly wide acceptance in the chemical process industries. Various forms of model predictive control schemes have been reported in the last decade owing to great interest evinced both by industrial practitioners and academic researchers. Besides their ability to handle large multivariable systems, the key feature that makes these algorithms attractive for industrial applications is their ability to handle constraints that arise in any real life process control situation. To achieve higher profits, the supervisory control layer often forces the process to operate at the intersection of constraints. MPC schemes incorporate the constraints explicitly in the control design strategy by solving a constrained optimization problem at each sampling instant and implementing the control moves in a receding horizon fashion. A variety of process descriptions ranging from step/impulse response coefficients, discrete transfer function models to state space models are employed by these MPC schemes - each having its merits and drawbacks.

A complete process model must be available in order to implement any of these control schemes. The multivariable process model is usually obtained empirically through an identification experiment. In the identification of MIMO processes, a high degree of correlation is often observed between process variables. In such cases, use of identification software based on the ordinary least squares (OLS) technique will result in parameter estimates with large variances owing to the ill-conditioned nature of the problem. Multivariate statistical techniques such as PLS have proved to be robust when presented with such correlated data. Consequently PLS and related methods (e.g. Principal Components Analysis (PCA), Ridge Regression (RR)) have found applications in the area of process monitoring, fault detection and identification (Kresta (1992), Wise (1991), Qin and McAvoy (1992), Qin (1993), Kaspar and Ray (1992,1993), Nomikos and MacGregor (1995)).
In this paper, we combine the PLS based multivariable modelling strategy proposed by Lakshminarayan et al. (1996) with the popular and powerful DMC algorithm (Cutler and Ramaker, 1980) in order to provide constrained control of multivariable systems. Since PLS and DMC represent mature statistics and control algorithms, we will review them only briefly. Different ways of dealing with process constraints will be explored - their merits and drawbacks will be pointed out. Illustrative examples involving the simulation of the Wood-Berry column and an experimental laboratory stirred tank heater will be presented.

2. DYNAMIC MODELLING AND CONTROL USING PLS

Earlier attempts to utilize the robust nature of the PLS regression in determining dynamic models from process data use PLS as a substitute for the OLS algorithm. Kaspar and Ray (1992, 1993) describe a method wherein the PLS algorithm is utilized in a manner that results in a convenient model structure. The resulting controller is diagonal and can be designed using the theory developed for SISO systems. Lakshminarayan et al. (1996) developed a related modelling algorithm that can be used to model and control (including feedback control) even nonlinear systems. This approach is described in some detail below.

Let us assume that the X and Y blocks consist of nx and ny variables respectively. For practical applications of the PLS algorithm, it may be necessary to scale the X and Y blocks suitably in view of the fact that the measurement units can be grossly different. The scaled X and Y blocks i.e., $X S^{-1}_{x}$ and $Y S^{-1}_{y}$ are processed by the PLS algorithm (with diagonal matrices $S_{x}$ and $S_{y}$ denoting the scaling matrices for the X and Y blocks respectively).

In PLS, the X and Y data are decomposed as a sum of a series of rank 1 matrices as follows (Wise, 1991):

$$X = t_{1}p_{1}^{T} + t_{2}p_{2}^{T} + \cdots + t_{n}p_{n}^{T} + E = TP^{T} + E$$ (1)

$$Y = u_{1}q_{1}^{T} + u_{2}q_{2}^{T} + \cdots + u_{q}q_{q}^{T} + F = UQ^{T} + F$$ (2)

In the above representation, T and U represent the matrices of scores while P and Q represent the loading matrices for the X and Y blocks. The first set of loading vectors (direction cosines of the dominant directions within the data set), $p_{1}$ and $q_{1}$, is obtained by maximizing the covariance between X and Y. Projection of the X and Y data respectively onto $p_{1}$ and $q_{1}$ gives the first set of scores vectors $t_{1}$ and $u_{1}$. This procedure is often referred to as the outer model. The matrices X and Y are now indirectly related through their scores by the inner model which is just a linear regression of $t_{1}$ on $u_{1}$ yielding $\hat{u}_{1} = t_{1}b_{1}$. $\hat{u}_{1}q_{1}^{T}$ can be interpreted as the part of the Y data that has been predicted by the first PLS dimension; in doing so, $t_{1}p_{1}^{T}$ portion of X data has been used up. Denoting $E_{1} = X$ and $F_{1} = Y$, the residuals at this stage are computed via the deflation process:

$$E_{2} = X - t_{1}p_{1}^{T} = E_{1} - t_{1}p_{1}^{T}$$

$$F_{2} = Y - \hat{u}_{1}q_{1}^{T} = Y - b_{1}t_{1}q_{1}^{T} = F_{1} - b_{1}t_{1}q_{1}^{T}$$

The procedure of determining the scores and loading vectors and the inner relation is continued (with the residuals computed at each stage) until the required number of PLS dimensions ($n$) are extracted. In practice, the number of PLS dimensions is determined based on the percentage of variance explained or by the use of statistically sound approaches such as cross validation. The directions considered irrelevant in the datasets (such as noise and redundancies) are confined to the error matrices E and F.

From a practical viewpoint, PLS can be considered as a technique that breaks up a multivariate regression problem into a series of univariate regression problems. The original regression problem is handled by constructing n inner relationship models (usually, $n \ll nx$). In addition to the PLS outer model (equations 1 & 2), we can write the following equation for describing the inner model of the PLS technique:

$$Y = TBQ^{T} + F$$ (3)

Lakshminarayan et al. (1996) proposed a dynamic extension of the PLS algorithm that is based on the direct modification of the PLS inner relation. Instead of relating the input and output scores (i.e. $t_{i}$ and $u_{i}$) using a static linear model, we relate them via a dynamic component (ARX or ARMAX structures). For the modelling procedure based on incorporation of the linear dynamic relationship (linear systems) in the PLS inner model, the decomposition of the X block is as given by equation (1). The dynamic analog of equation (2) is given by

$$Y = G_{1}(t_{1})q_{1}^{T} + G_{2}(t_{2})q_{2}^{T} + \cdots + G_{n}(t_{n})q_{n}^{T} + F = Y_{1}^{exp} + Y_{2}^{exp} + \cdots + Y_{n}^{exp} + F$$

Here, the $G_{i}$’s denote the linear dynamic models (eg. ARX) identified at each stage and $G_{i}(t_{i})q_{i}^{T}$ quantifies the measure of Y space explained by the i-th PLS dimension ($Y_{i}^{exp}$). For the model identified using the dynamic PLS algorithm, we can express the transfer function relating input j to output i as...
3. CONSTRAINED MODEL PREDICTIVE CONTROL IN LATENT SPACES

Dynamic Matrix Control

The DMC algorithm has been extensively described in the literature (García et al., 1990). Utilizing step response data, the DMC algorithm is designed on the basis of a multistep objective function subject to input amplitude, rate and output constraints. The objective function is usually a quadratic function of: (1) the weighted deviation of plant outputs from their targets over the prediction horizon \(N_2\) and (2) the weighted control action over the control horizon \(N_c\). At each sampling instant, several control moves are computed but only the first control move is implemented. Under this scheme, the control law portrays a nonlinear nature since different sets of constraints may be active at any sampling instant. For the unconstrained case, there exists an analytical solution for the optimal control move to be made at each sampling instant. When constraints do exist, use of numerical optimization codes such as Quadratic Program SOLver (QPSOL) becomes mandatory. Rather than getting into the mathematical details, we focus attention on the geometry of the constraints in the original and latent spaces.

In the case of linear systems, equation (4) provides a model in terms of the original variables. For the unconstrained case, the dynamic inner models can be used in two ways: (1) Each inner model can be used to develop SISO DMC controllers and (2) A MIMO DMC controller which utilizes all the \(n\) inner models together. If constraints are imposed on the manipulated variables, the constraints in the latent space are coupled (as described below). If strategy (1) is employed, then the controllers must act in a co-ordinated fashion. Otherwise, constraints on the original variables will be violated. With strategy (2), a one-time transformation of the constraints is adequate for efficient implementation of the DMC algorithm.

In the original space, the constraints are represented as:

\[
\begin{align*}
\underline{x}_{\text{min}} \leq x & \leq \overline{x}_{\text{max}} \\
\Delta \underline{x}_{\text{min}} \leq \Delta x & \leq \Delta \overline{x}_{\text{max}} \\
\underline{u}_{\text{min}} \leq \dot{u} & \leq \overline{u}_{\text{max}}
\end{align*}
\]

For a case involving two manipulated variables, the amplitude constraints are shown in figure 2(a) (mathematically expressed in equation (5)). In terms of the latent space variables and the PLS matrices, the above equation may be written as:

\[
\begin{align*}
\underline{x}_{\text{min}} \leq P S_x T & \leq \overline{x}_{\text{max}}
\end{align*}
\]
A graphical plot of the constraints in terms of the input space latent variables (T) is depicted in figure 2(b). Use of the PLS inner models with the constraints as given in equation (8), will ensure the satisfaction of constraints in the original space. Such a mapping is one to one - each point in the constrained original space has a unique image in the constrained latent space and vice versa. The outcome of transforming the original constraints into latent space constraints is that the constraints that were decoupled in the original space become coupled in the latent space. A similar analysis holds for the rate constraints as well. Hence a multivariate approach to controller design is mandatory.

Let us pose the problem of mapping constraints differently. The constraints are now posed in the latent space in a decoupled form. We now seek the maximum and minimum values in the t-space, \( L_{\text{max}} \) and \( L_{\text{min}} \), such that the constraints in the original space are satisfied i.e., find \( L_{\text{min}} \leq \mathbf{L} \leq L_{\text{max}} \) such that \( \mathbf{L}_{\text{min}} \leq \mathbf{L} \leq \mathbf{L}_{\text{max}} \). This approach results in original space constraints that are coupled. When the constrained regions of the previous approach and this approach are plotted together, we notice the suboptimality of this approach (see figure 3) - not all of the original constraint region is utilized (space bounded by broken lines) because we seek to match only the necessary conditions i.e. the maximum and minimum values. The constraints are satisfied but the controller does not use some permitted regions in the input space implying that some set points cannot be reached and some disturbances will not be rejected completely.

The objective function is given by

\[
\min J = \Delta x^T (G^T + A') \Delta x - 2(r - \dot{f} - A')^T, G \Delta x
\]

subject to \( A \Delta x + B \geq 0 \)

where \( G \) is the dynamic matrix comprising the step response coefficients, \( f \) is the free response vector, \( d \) is the estimate of the disturbance vector and \( r \) is the setpoint trajectory that we want to achieve. For a more detailed account of the terms involved in the MPC formulation, see García et al. (1990). In the latent space the same problem can be restated as

\[
\min J = \Delta u^T (S^T + A') \Delta t - 2(r' - \dot{f}' - A')^T, S \Delta t
\]

subject to \( A' \Delta t + B' \geq 0 \)

where the primed quantities are the corresponding expressions in terms of the latent variables and \( S \) is the dynamic matrix obtained from the PLS dynamic model in the latent space. The free response and the disturbance estimate are also obtained from the PLS model. The constraints are posed in the original space and transformed as per the previous section.

Note that the latent variables are scaled variables, hence the control and output weightings have to be chosen accordingly. The time scales however, are invariant to the transformations and therefore the choice of \( N_1, N_2 \& N_x \) will remain the same in both the original and the latent spaces.

**Example 1: Simulation of the Wood-Berry Column**

Wood and Berry (1973) reported the following transfer functions for methanol-water separation in a distillation column. The composition of the top \( (y_1) \) and bottom \( (y_2) \) products expressed in weight % of methanol are the controlled variables. The reflux \( (x_1) \) and the reboiler steam \( (x_2) \) flowrates are the manipulated inputs expressed in lb/min. Time is in minutes.

\[
\begin{bmatrix}
  y_1(s) \\
  y_2(s)
\end{bmatrix} = \begin{bmatrix}
  12.8e^{-s} & -18.9e^{-3s} \\
  16.7s + 1 & 21s + 1 \\
  6.6e^{-7s} & -19.4e^{-3s} \\
  10.9s + 1 & 14.4s + 1
\end{bmatrix} \begin{bmatrix}
  x_1(s) \\
  x_2(s)
\end{bmatrix}
\]

\[(9)\]
The above model serves as the plant in the simulation runs. For this system, Lakshminarayanan et al. (1996) developed a dynamic PLS model using input-output data generated from equation (9). The PLS model is:

\[
\begin{align*}
S_x &= \begin{bmatrix} 0.4770 & 0 \\ 0 & 0.6374 \end{bmatrix}; S_y &= \begin{bmatrix} 12.7577 & 0 \\ 0 & 12.2821 \end{bmatrix} \\

P &= \begin{bmatrix} 0.3228 & 0.9455 \\ -0.9455 & 0.3256 \end{bmatrix}; R &= \begin{bmatrix} 0.3256 & 0.9465 \\ -0.9455 & 0.3228 \end{bmatrix} \\

Q &= \begin{bmatrix} 0.6972 & 0.7397 \\ 0.7169 & -0.6503 \end{bmatrix} \\

G_1 &= \frac{0.1417 z^{-5}}{1 - 0.4305 z^{-1} - 0.4706 z^{-2}} \\

G_2 &= \frac{0.0529 z^{-5} + 0.0291 z^{-6}}{1 - 0.2336 z^{-1} - 0.2321 z^{-2}}
\end{align*}
\]

The comparison of the conventional and the PLS based MPC (figure 4) clearly indicates the effect of scaling on the control weighting \( \Lambda \). \( \Lambda = 100 I \) in the original space yields the same performance as \( \Lambda' = 0.05 I \) in the latent space (ISE values were compared for servo as well as several regulatory runs). Since the variables are well scaled in the latent space and the dynamics are decoupled, the choice of tuning parameters - , \( \Lambda \) is easier.

In figure 5, we compare the performance of the MPC to that of the Vogel-Edgar digital controller. Both the controllers were implemented in the latent space. The results indicate the superior performance of the MPC controller.

Figure 6 highlights the geometry of the constraints in the original (X) and latent spaces (T). The crosses (x) indicate the constraint region posed in the control design \((-0.06 \leq \Delta x \leq 0.06, -0.3 \leq x \leq 0.3\) and the circles (o) are the values from the simulation run.
Example 2: Real-Time Control of the Laboratory Stirred Tank Heater

The laboratory stirred tank heater is a cylindrical tank of uniform cross section where two streams of water (one hot and the other cold) are mixed. The contents of the tank exit through a long and winding copper tube. The flow rates of the hot and cold water streams serve as manipulated variables to control the level of water in the tank and the temperature of the exit stream. Facilities exist to introduce disturbances in the steam flow through the heater coils, the inlet temperature of the hot water stream etc. The control algorithm was implemented using a personal computer running Real-Time Matlab/Simulink.

The results of the laboratory run showing servo and regulatory responses is depicted in figure 7. Two set point changes in each output variable were made. An unknown amount of cold water was dumped into the vessel around the 1575th sample point. Satisfactory servo and regulatory responses were obtained by employing $N_1 = 1$, $N_x = 2$, $N_2 = 20$ and $\lambda' = 10I$ as the controller parameters. The constraints placed on the amplitude (0-100\%) and rate ($\pm 10\%$) of the manipulated variables remained inviolate over the entire duration of the experiment.

4. CONCLUSIONS

Constrained control of multivariable processes using dynamic PLS models has been demonstrated in this paper. The approach presented here has been extended to a class of nonlinear systems with promising results (Patwardhan et al., 1996). Synthesis of such predictive controllers in the latent variable subspace is expected to provide more insight into the use and role of different tuning parameters. Work is in progress to investigate the usefulness of the proposed strategy in the tuning of control loops.

5. REFERENCES


