Model Predictive Control
TECHNICAL OVERVIEW

Excerpted from: Wade, H.L., *Basic and Advanced Regulatory Control: System Design and Application.* ISA, 2004. Used by permission.

INTRODUCTION

Model predictive control (MPC) uses a sampled data form of process model to predicted future values of a process variable, based upon past values of controlled inputs (controller outputs). The predicted values are compared with future values of the set point to calculate predicted future error values, as well as a prediction of future encroachment on constraints. From these predictions, a sequence of controller output values is calculated which will minimize some function of the predicted error as well as avoid encroaching upon the constraint. Model predictive control is usually (but not always) applied to multiple-input, multiple-output processes, subject to numerous disturbances and dynamically varying constraints. The technology thus encompasses feedback, feedforward, decoupling and constraint control in one comprehensive package.

We will present here the essence of MPC, first for a simple single-input, single-output processes, both in mathematical terms and in graphical form. Then we will indicate how it can be extended to multiple-input, multiple-output applications. Later we will discuss additional issues regarding MPC. This material is excerpted from the referenced cited under the title, where additional details may be found:

For greater depth of coverage, see

Camancho, E. F, and C. Bordons, *Model Predictive Control.* Springer-Verlag, 1999.

Rossier, J. A., *Model-Based Predictive Control, A Practical Approach.* CRC Press, 2003.

To be consistent with terminology used in the MPC literature, we will use the terms "process variable," "control variable" and "CV" interchangeably. Likewise "process input," "controller output," "manipulated variable" and "MV" are interchangeable, as are "disturbance variable" and "DV." Auxiliary variables, usually associated with constraints, are termed "AVs."

SYMBOLS USED

$\mathbf{v}$	column vector of n elements (Size of $\mathbf{v}$ is $n \times 1$.)
$\mathbf{v}^\mathbf{T}$	row vector. (Superscript T indicates transpose of the vector.)
v_i	i^{th} element of vector $\mathbf{v}$. $i = 1,2, \ldots, n$.
$\mathbf{P}$	matrix of n rows and m columns (Size of $\mathbf{P}$ is $n \times m$.)
$\mathbf{P}^\mathbf{T}$	transpose of matrix $\mathbf{P}$. (Size of $\mathbf{P}^\mathbf{T}$ is $m \times n$.)
$\mathrm{P}ij$	element of matrix $\mathbf{P}$, in the i^{th} row, j^{th} column
x	actual value of a controlled variable
$\hat{x}$	predicted value of a controlled variable
CV	controlled variables

MV	manipulated variables
DV	disturbance variables
AV	auxiliary variables
K	control horizon (number of future samples to calculate control moves)
N	prediction horizon (number of future samples to predict values of CV)
λ	time constant for reference trajectory

For multiple input, multiple output processes:

R	number of CVs
S	number of MVs
T	number of DVs

UNCONSTRAINED MPC FOR SISO PROCESSES

Process Model

Model predictive control begins by maintaining a sampled data step response model of the process. Compare this with the first-order-lag plus dead-time (FOPDT) step response model frequently used for feedforward and decoupling control. To obtain a FOPDT model, a step change is made to the process input, the response is observed then approximated with three parameters, representing process gain, dead time, and time constant. With MPC, all of the sampled data collected as a result of the step testing is retained in a series of memory locations; this data is called a "vector." For example, in Figure 1, the sequence of values ($p_1, p_2, \text{---}, p_N$), resulting from a step input change of one unit would be retained as the step response model; this data vector is called **p** in equation 1.

$$\mathbf{p} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_N \end{bmatrix} \tag{1}$$

Note that this technique is as valid for irregular responses shown in Figure 2 as is it for the well behaved process shown in Figure 1. Also note that for a self-regulating process, the response reaches an equilibrium where there is no further change in the sampled values. Hence, only a finite number of samples need to be retained. With commercial packages, N can be as small as 30 or as large as 120.

Note also that if the step input change is something other than one unit, the data should be normalized so that it represents the response that would result from a one unit input change.

Finally, note that what we have called "process input" may in fact be a set point change to a lower level controller, such as a flow controller. In this case, we would be considering the flow loop merely as a part of the process.

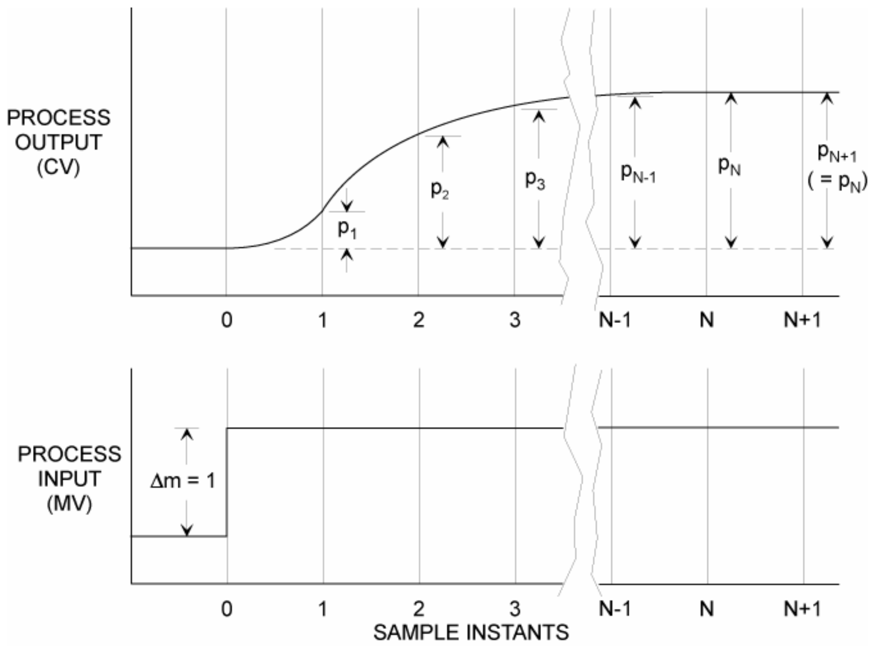

Figure 1
Step Response Model

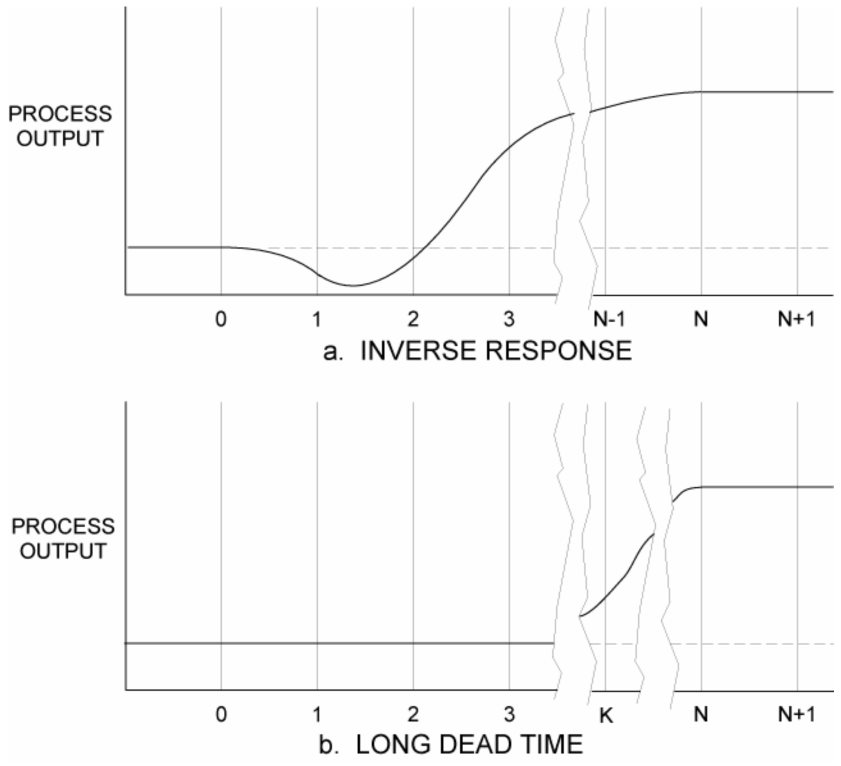

Figure 2
Step Response Model for Irregular Processes

While the procedure described above is valid in concept, in actual practice a more elaborate procedure may be employed for obtaining the step response model. For instance, the data may be prefiltered, there may be a series of alternate direction steps of varying lengths, etc. Once such test procedure is called a pseudo-random binary test sequence. This phase is called the "identification" phase, and may be a proprietary procedure for a particular commercial package.

Prediction

For the next step in our exposition, let us assume that during some sample and control period, we know the current value of the CV; call this x_0. Furthermore, assume that we also have a sequence of values, $\left(\hat{x}_{1,0},\ \hat{x}_{2,0},\ \cdots,\hat{x}_{N,0}\right)$, or collectively as vector $\hat{\mathbf{x}}_{\mathbf{0}}$. This represents our current prediction of what the CV will be for the next N sample periods, based upon prior values of the process input, and also with the assumptions that there will be no further changes in process input nor disturbances to the process. The maximum index, N, is the number of sample values in our step response model. This is called the "prediction horizon."

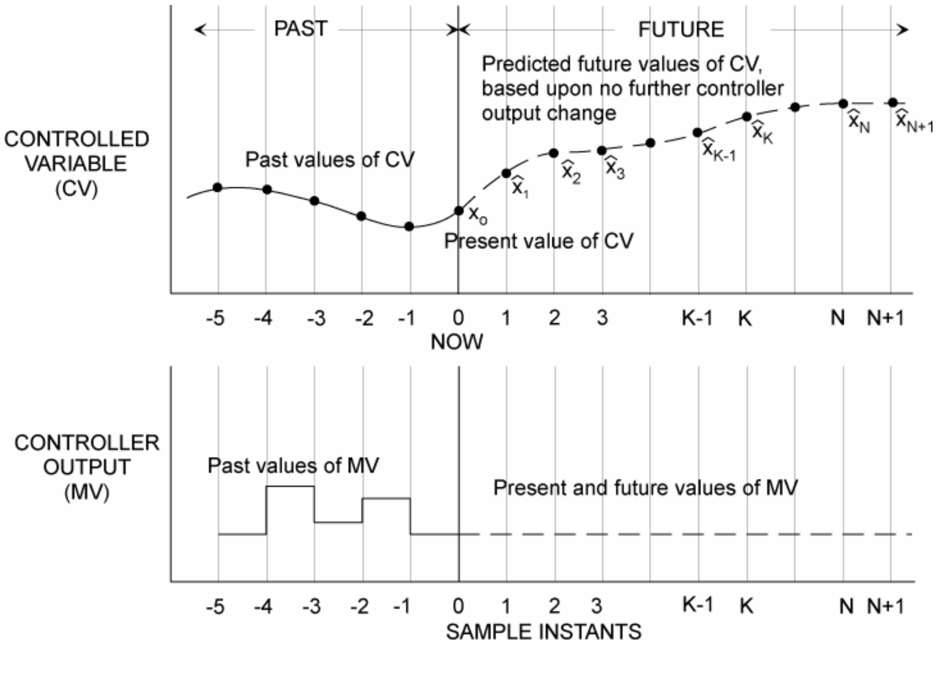

Figure 3
Predicted Profile, Based Only On Prior Control Moves

Suppose also that we know the next K controller output moves (changes in the MV) we intend to make. (Go along with us on this; don't worry about how we happen to know. Later on, we will see how these moves are determined. For now, just "suppose we know ….") Call this sequence of moves $\left(\Delta m_0, \Delta m_1, \cdots, \Delta m_{K-1}\right)$, or collectively as vector $\mathbf{\Delta m}$. K is called the "control horizon;" K is much less than N, perhaps 1/3 of N.

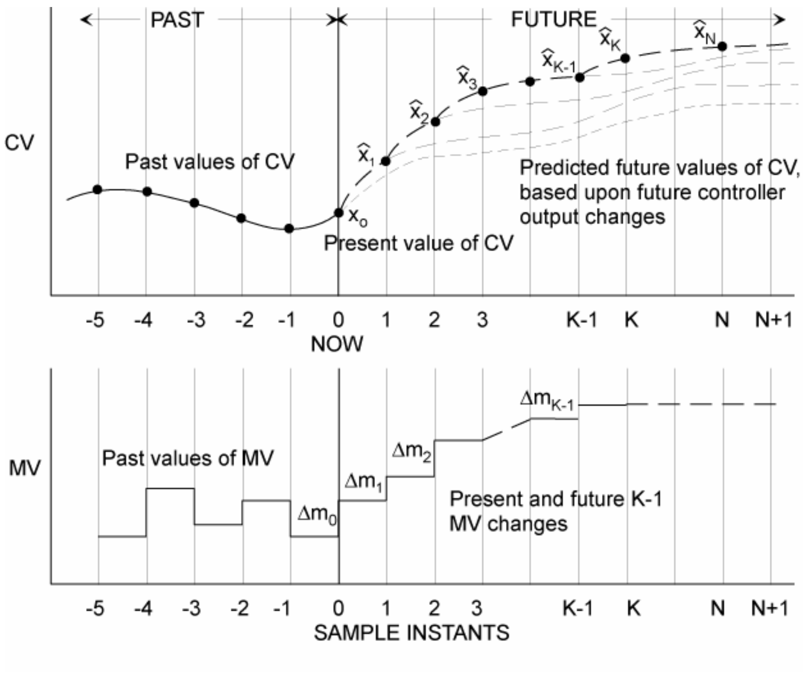

Figure 4
Modification of Predicted Profile by Current and Future Controller Moves

Each control move that is made will change our prediction of the future profile of the control variable. This is depicted by Figure 4, and shown by equation 2. By the principle of superposition, the change to the predicted profile will be the magnitude of the control move times the step response vector. (Recall that the step response vector is the step response to a process input change of 1 unit.) For the control horizon, the predicted values of CV are given by:

$$\begin{aligned}
\hat{x}_{1,K} &= \hat{x}_{1,0} + p_1\Delta m_0 \\
\hat{x}_{2,K} &= \hat{x}_{2,0} + p_2\Delta m_0 + p_1\Delta m_0 \\
\hat{x}_{3,K} &= \hat{x}_{3,0} + p_3\Delta m_0 + p_2\Delta m_1 + p_1\Delta m_3 \\
\vdots &= \qquad \vdots \\
\hat{x}_{K,K} &= \hat{x}_{K,0} + p_K\Delta m_0 + p_{K-1}\Delta m_1 + \dots + p_1\Delta m_{K-1} \\
\vdots &= \qquad \vdots \qquad\qquad \vdots \\
\hat{x}_{N,K} &= \hat{x}_{N,0} + p_N\Delta m_0 + p_{N-1}\Delta m_1 + \dots + \dots + p_{N-K+1}\Delta m_{K-1}
\end{aligned} \tag{2}$$

Equation 2 can be written in expanded vector-matrix notation, as shown in equation 3, or in compact vector-matrix notation as shown by equation 4[1].

[1] Readers unfamiliar with vector-matrix notations can consult any number of textbooks on this subject.

$$\begin{bmatrix} \hat{x}_{1,K} \\ \hat{x}_{2,K} \\ \hat{x}_{3,K} \\ \vdots \\ \hat{x}_{K,K} \\ \vdots \\ \hat{x}_{N,K} \end{bmatrix} = \begin{bmatrix} \hat{x}_{1,0} \\ \hat{x}_{2,0} \\ \hat{x}_{3,0} \\ \vdots \\ \hat{x}_{K,0} \\ \vdots \\ \hat{x}_{N,0} \end{bmatrix} + \begin{bmatrix} p_1 & 0 & 0 & \cdots & 0 \\ p_2 & p_1 & 0 & \cdots & 0 \\ p_3 & p_2 & p_1 & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ p_K & p_{K-1} & & p_2 & p_1 \\ \vdots & \vdots & & & \vdots \\ p_N & p_{N-1} & \cdots & \cdots & p_{N-K+1} \end{bmatrix} \begin{bmatrix} \Delta m_0 \\ \Delta m_1 \\ \Delta m_2 \\ \vdots \\ \Delta m_{K-1} \end{bmatrix} \qquad (3)$$

$$\mathbf{\hat{x}_K} = \mathbf{\hat{x}_0} + \mathbf{P}\Delta\mathbf{m} \qquad (4)$$

Calculation of Control Moves

We will now address the question of determining the current and future control moves. We assumed previously that we knew the control moves, hence we could correct the predicted profile of the control variable. If this were true, and if we knew the set point during the prediction horizon, as shown in Figure 5, then we could also predict the error values after K control moves in the future. Call the sequence of error values $\left(\hat{e}_{1,K}, \hat{e}_{2,K}, \cdots, \hat{e}_{N,K}\right)$, or collectively as the vector $\mathbf{\hat{e}_K}$.

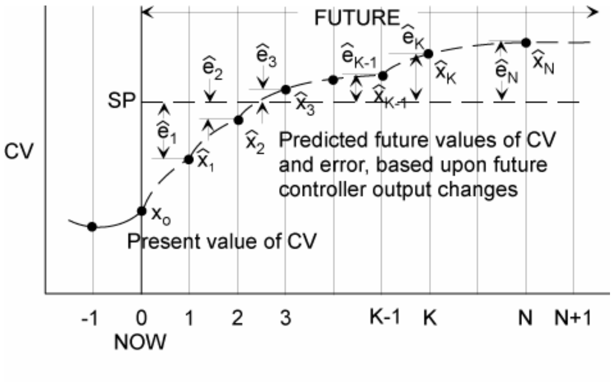

Figure 5
Predicted Error Profile

$$\hat{e}_{i,K} = x_{SP,i} - \hat{x}_{i,K}$$

Hence,

$$\begin{aligned} \mathbf{\hat{e}_K} &= \mathbf{x_{SP}} - \mathbf{\hat{x}_K} \\ &= \mathbf{x_{SP}} - \mathbf{\hat{x}_0} - \mathbf{P}\Delta\mathbf{m} \\ &= \mathbf{\hat{e}_0} - \mathbf{P}\Delta\mathbf{m} \end{aligned} \qquad (5)$$

where $\mathbf{\hat{e}_0} = \mathbf{x_{SP}} - \mathbf{\hat{x}_0}$

To calculate the control moves, we will minimize a cost functional, *J*, which is one-half of the sum of squares of the predicted errors.

$$J = \frac{1}{2}\sum_{i=1}^{N} \hat{e}_{i,K}{}^{2}$$
$$= \frac{1}{2}\hat{\mathbf{e}}_{\mathbf{K}}{}^{\mathrm{T}}\hat{\mathbf{e}}_{\mathbf{K}} \qquad (6)$$

After incorporation equation 5, this becomes:

$$\mathrm{J} = \frac{1}{2}\left[\hat{\mathbf{e}}_{\mathbf{0}}{}^{\mathrm{T}} - \Delta\mathbf{m}^{\mathrm{T}}\mathbf{P}^{\mathrm{T}}\right]\left[\hat{\mathbf{e}}_{\mathbf{0}} - \mathbf{P}\Delta\mathbf{m}\right]$$
$$= \frac{1}{2}\hat{\mathbf{e}}_{\mathbf{0}}{}^{\mathrm{T}}\hat{\mathbf{e}}_{\mathbf{0}} - \hat{\mathbf{e}}_{\mathbf{0}}{}^{\mathrm{T}}\mathbf{P}\Delta\mathbf{m} + \frac{1}{2}\Delta\mathbf{m}^{\mathrm{T}}\mathbf{P}^{\mathrm{T}}\mathbf{P}\Delta\mathbf{m} \qquad (7)$$

The usual minimization procedure is to set the derivative to zero, hence:

$$\frac{\partial J}{\partial \Delta\mathbf{m}} = \begin{bmatrix} \dfrac{\partial J}{\partial \Delta m_0} \\ \dfrac{\partial J}{\partial \Delta m_1} \\ \\ \dfrac{\partial J}{\partial \Delta m_{K-1}} \end{bmatrix}$$

$$= \mathbf{P}^{\mathrm{T}}\mathbf{P}\Delta\mathbf{m} - \mathbf{P}^{\mathrm{T}}\hat{\mathbf{e}}_{\mathbf{0}} \qquad (8)$$

Set the right hand side of this equation to zero and solve for **Δm**:

$$\Delta\mathbf{m} = \left[\mathbf{P}^{\mathrm{T}}\mathbf{P}\right]^{-1}\mathbf{P}^{\mathrm{T}}\hat{\mathbf{e}}_{\mathbf{0}} \qquad (9)$$

The matrix **P** is tall, slender matrix (size *N*x*K*) and cannot be inverted, but $\left[\mathbf{P}^{\mathrm{T}}\mathbf{P}\right]$ is a square matrix (*K*x*K*), hence, in general, can be inverted. The matrix **P** is determined initially by the process model, equation 1.

Inversion of $\left[\mathbf{P}^{\mathrm{T}}\mathbf{P}\right]$ is not required at each control sample period but only at the time **P** is determined. In fact, the entire matrix manipulation, $\left[\mathbf{P}^{\mathrm{T}}\mathbf{P}\right]^{-1}\mathbf{P}^{\mathrm{T}}$, can be performed at that time. Furthermore, let matrix **W** be a *K*x*N* matrix, defined by

$$\mathbf{W} = \left[\mathbf{P}^{\mathrm{T}}\mathbf{P}\right]^{-1}\mathbf{P}^{\mathrm{T}} \qquad (10)$$

Matrix **W** is comprised of a series of *K* row vectors, each of *N* elements:

$$\mathbf{W} = \begin{bmatrix} - & \mathbf{w}_1{}^{\mathrm{T}} & \rightarrow \\ - & \mathbf{w}_2{}^{\mathrm{T}} & \rightarrow \\ \vdots & \vdots & \vdots \\ - & \mathbf{w}_K{}^{\mathrm{T}} & \rightarrow \end{bmatrix}$$

The current control move to be made, Δm_0, is calculated using only the top row, $[\mathbf{w_1}]^{\mathrm{T}}$, of **W.**

$$\Delta m_0 = [\mathbf{w_1}]^{\mathrm{T}}\hat{\mathbf{e}}_{\mathbf{0}} \qquad (11)$$

The other control moves, $\Delta m_1, \Delta m_2, \cdots, \Delta m_{K-1}$ are not required, since after making the control move and correcting the predicted profile, we are going to step forward one sample period and repeat the procedure. This further reduces the computation burden at each calculation step.

The astute reader will observe that there are several missing ingredients in the development thus far:

Feedback has not been utilized;
No provisions for controller tuning have been presented;
Furthermore, there has been no utilization of the knowledge of measurable disturbances.

These omissions will now be corrected.

Incorporation of Feedback

After making the control move calculated by equation 11, the predicted profile must be corrected. This corrected profile includes a prediction of the value of the control variable at the next sample instant, $\hat{x}_1$. Once we have advanced to the next sample instant, that prediction becomes the prediction of the value of the control variable at the present time, $\hat{x}_0$. The actual value of the variable, x_0, and the difference between the actual and predicted values is calculated:

$$\Delta x_0 = x_0 - \hat{x}_0 \qquad (12)$$

The entire profile, including the predicted current value, is then shifted by this difference, as shown in Figure 6.

$$\textit{For } i = 1, 2, \cdots, N \qquad \hat{x}_i \leftarrow \hat{x}_i + \Delta x_0 \qquad (13)$$

where "←" means "replaced by." Academic research shows that this step is equivalent to adding an integrator into the control loop, thus assuring that the process variable will eventually come to set point.

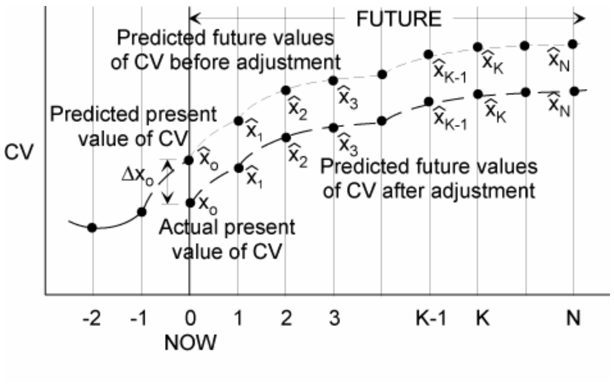

Figure 6
Incorporation of Feedback

Tuning

There are two common techniques for tuning MPC. Some commercial systems use one or the other; some use both. These techniques are:

Move suppression
Reference trajectory

Other design parameters that have an effect on performance, hence could be considered as tuning parameters include the sample time, the prediction horizon (*N* sample periods) and the control horizon (*K* sample periods).

Move Suppression.

With this technique, the cost functional *J*, given by equation 6, is augmented by the weighted sum of squares of the control moves.

$$J = \frac{1}{2}\left(\sum_{1}^{N} \hat{e}_{i,K}{}^{2} + \sum_{1}^{K} q_i \, \Delta m_{i-1}{}^{2} \right) \tag{14}$$

$$= \frac{1}{2} \hat{\mathbf{e}}_{\mathbf{K}}{}^{\mathrm{T}} \hat{\mathbf{e}}_{\mathbf{K}} + \frac{1}{2} \Delta \mathbf{m}^{\mathrm{T}} \mathbf{Q} \, \Delta \mathbf{m}$$

$$= \frac{1}{2} \left[\hat{\mathbf{e}}_{\mathbf{0}}{}^{\mathrm{T}} - \Delta \mathbf{m}^{\mathrm{T}} \mathbf{P}^{\mathrm{T}} \right] \left[\hat{\mathbf{e}}_{\mathbf{0}} - \mathbf{P} \Delta \mathbf{m} \right] + \frac{1}{2} \Delta \mathbf{m}^{\mathrm{T}} \mathbf{Q} \, \Delta \mathbf{m}$$

$$= \frac{1}{2} \hat{\mathbf{e}}_{\mathbf{0}}{}^{\mathrm{T}} \hat{\mathbf{e}}_{\mathbf{0}} - \hat{\mathbf{e}}_{\mathbf{0}}{}^{\mathrm{T}} \mathbf{P} \Delta \mathbf{m} + \frac{1}{2} \Delta \mathbf{m}^{\mathrm{T}} \left[\mathbf{P}^{\mathrm{T}} \mathbf{P} + \mathbf{Q} \right] \Delta \mathbf{m}$$

Equations 15, 16, and 17 are analogous to equations 8, 9 and 10:

$$\frac{\partial J}{\partial \mathbf{\Delta m}} = \begin{bmatrix} \dfrac{\partial J}{\partial \Delta m_0} \\ \dfrac{\partial J}{\partial \Delta m_1} \\ \\ \dfrac{\partial J}{\partial \Delta m_{K-1}} \end{bmatrix} = \left[\mathbf{P}^{\mathrm{T}}\mathbf{P} + \mathbf{Q}\right]\mathbf{\Delta m} - \mathbf{P}^{\mathrm{T}}\hat{\mathbf{e}}_{\mathbf{0}} \tag{15}$$

$$\mathbf{\Delta m} = \left[\mathbf{P}^{\mathrm{T}}\mathbf{P} + \mathbf{Q}\right]^{-1}\mathbf{P}^{\mathrm{T}}\hat{\mathbf{e}}_{\mathbf{0}} \tag{16}$$

$$\mathbf{W} = \left[\mathbf{P}^{\mathrm{T}}\mathbf{P} + \mathbf{Q}\right]^{-1}\mathbf{P}^{\mathrm{T}} \tag{17}$$

Note that equation 11 is still applicable for the calculation of Δm_0.

In practice, the q_i weights in equation 14 are usually selected to be the same value, q, thus leading to a single tuning parameter for move suppression. A larger value of q will lead to a more conservative controller.

<u>Reference Trajectory</u>

A reference trajectory for exponential return to set point from the present value is established by the specification of a time constant, λ. Pseudo-set point values are the values of this reference trajectory at the future sampling instances, using equation 18. The error vectors used in equation 6 or 14 are the differences between these pseudo-set point values and the predicted values for the PV. This technique provides an additional parameter for tuning; a small value for λ will cause the controller to be aggressive; a larger value will result in a more conservative controller.

$$x_{SP,i} = \left(x_{SP} - x_0\right)\left(1 - \exp\left(-i\Delta T / \lambda\right)\right) \tag{18}$$

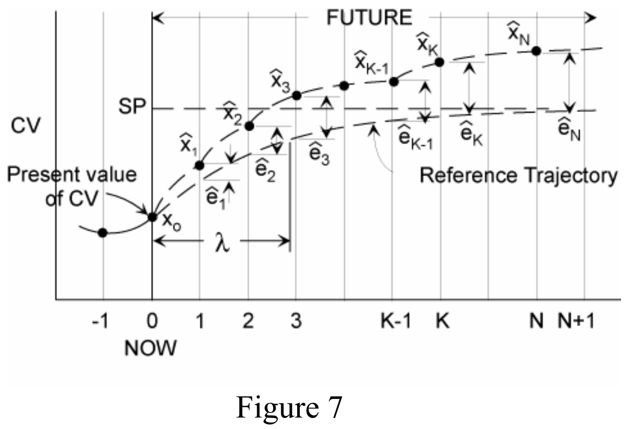

Figure 7

Reference Trajectory for Return to Set Point

Incorporation of Feedforward Control

If there is a measurable disturbance, then a unit step response model is determined for the effect of this disturbance on the process variable, similar to Figure 1. This disturbance model is characterized by a sequence of values, $\left(d_1, d_2, \cdots, d_N\right)$, or collectively by the vector, **d**. At the beginning of each calculation cycle, the disturbance variable is measured, and the change in the disturbance since the last sample instant, Δu, is determined. Then the predicted profile is corrected to account for this change. Equations 19, 20 and 21 are modifications of equations 3, 4 and 5 to incorporate this feature. The vector $\hat{\mathbf{e}}_{\mathbf{0}}$ can then be used in equation 9 or 16 to compute the control moves.

$$\begin{bmatrix} \hat{x}_{1,K} \\ \hat{x}_{2,K} \\ \hat{x}_{3,K} \\ \vdots \\ \hat{x}_{K,K} \\ \vdots \\ \hat{x}_{N,K} \end{bmatrix} = \begin{bmatrix} \hat{x}_{1,0} \\ \hat{x}_{2,0} \\ \hat{x}_{3,0} \\ \vdots \\ \hat{x}_{K,0} \\ \vdots \\ \hat{x}_{N,0} \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \\ d_K \\ \\ d_N \end{bmatrix} \Delta u + \begin{bmatrix} p_1 & 0 & 0 & \cdots & 0 \\ p_2 & p_1 & 0 & \cdots & 0 \\ p_3 & p_2 & p_1 & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ p_K & p_{K-1} & & p_2 & p_1 \\ \vdots & \vdots & & & \vdots \\ p_N & p_{N-1} & \cdots & \cdots & p_{N-K+1} \end{bmatrix} \begin{bmatrix} \Delta m_0 \\ \Delta m_1 \\ \Delta m_2 \\ \vdots \\ \Delta m_{K-1} \end{bmatrix} \tag{19}$$

$$\hat{\mathbf{x}}_{\mathbf{K}} = \hat{\mathbf{x}}_{\mathbf{0}} + \mathbf{d}\Delta \mathrm{u} + \mathbf{P}\Delta\mathbf{m} \tag{20}$$

$$\hat{\mathbf{e}}_{\mathbf{0}} = \mathbf{x}_{\mathbf{SP}} - \hat{\mathbf{x}}_{\mathbf{0}} - \mathbf{d}\Delta \mathrm{u} \tag{21}$$

Summary Diagram

A diagram showing in detail the computations made in one calculation cycle and the effect on the memory locations holding the data vector $\hat{\mathbf{x}}$ is shown in Figure 8. This diagram starts with the status of $\hat{\mathbf{x}}$ at the end of one calculation cycle, then proceeds to the beginning of the next calculation cycle and on through the completion of that cycle.

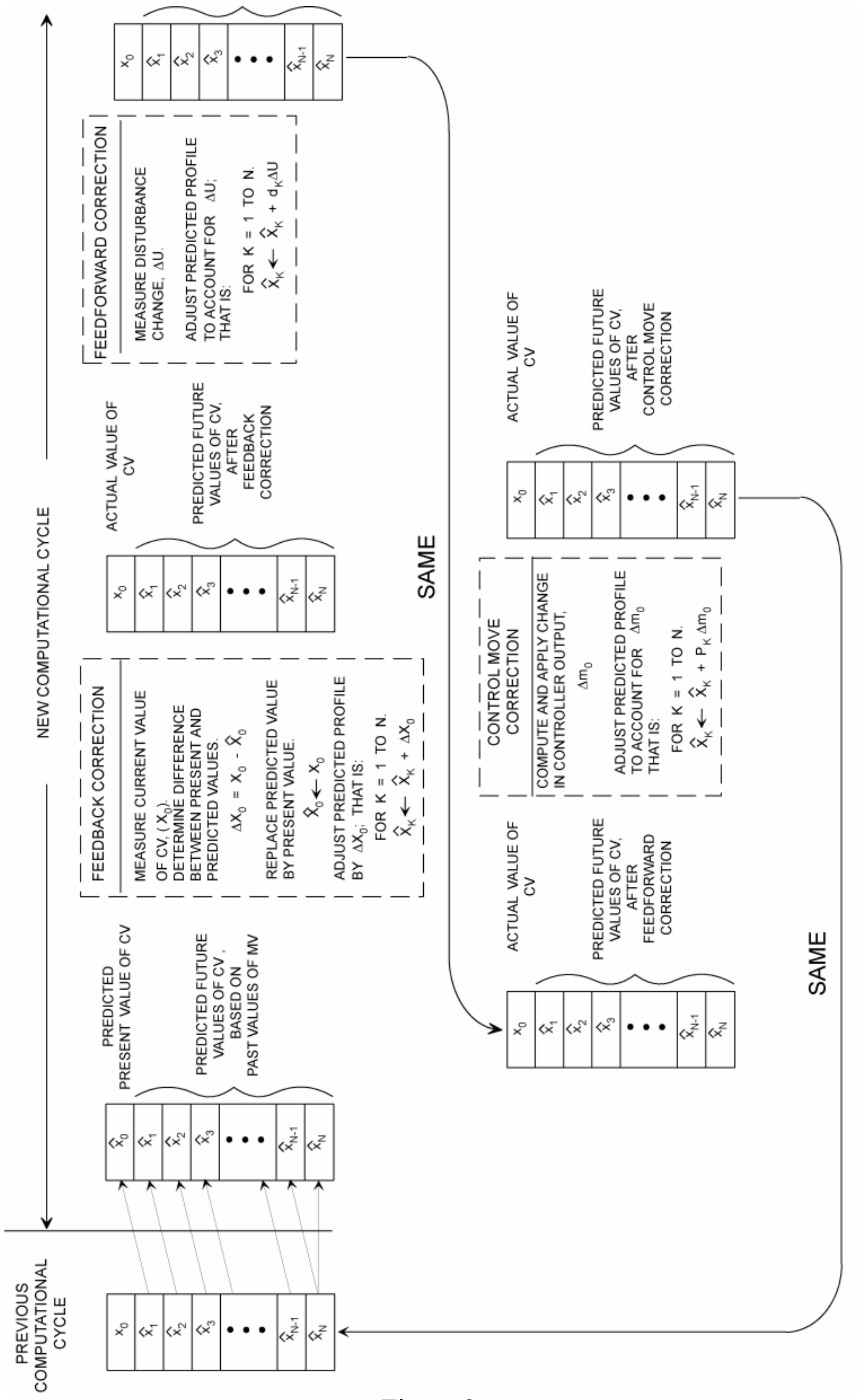
PREVIOUS COMPUTATIONAL CYCLE
NEW COMPUTATIONAL CYCLE
x_0 $\hat{x}_1$ $\hat{x}_2$ $\hat{x}_3$ • • • $\hat{x}_{N-1}$ $\hat{x}_N$
PREDICTED PRESENT VALUE OF CV
PREDICTED FUTURE VALUES OF CV, BASED ON PAST VALUES OF MV
$\hat{x}_0$ $\hat{x}_1$ $\hat{x}_2$ $\hat{x}_3$ • • • $\hat{x}_{N-1}$ $\hat{x}_N$
FEEDBACK CORRECTION
MEASURE CURRENT VALUE OF CV, (X_0). DETERMINE DIFFERENCE BETWEEN PRESENT AND PREDICTED VALUES.
$\Delta X_0 = X_0 - \hat{X}_0$
REPLACE PREDICTED VALUE BY PRESENT VALUE.
$\hat{X}_0 \leftarrow X_0$
ADJUST PREDICTED PROFILE BY ΔX_0; THAT IS:
FOR K = 1 TO N.
$\hat{X}_K \leftarrow \hat{X}_K + \Delta X_0$
ACTUAL VALUE OF CV
PREDICTED FUTURE VALUES OF CV, AFTER FEEDBACK CORRECTION
x_0 $\hat{x}_1$ $\hat{x}_2$ $\hat{x}_3$ • • • $\hat{x}_{N-1}$ $\hat{x}_N$
FEEDFORWARD CORRECTION
MEASURE DISTURBANCE CHANGE, ΔU.
ADJUST PREDICTED PROFILE TO ACCOUNT FOR ΔU; THAT IS:
FOR K = 1 TO N.
$\hat{X}_K \leftarrow \hat{X}_K + d_K \Delta U$
x_0 $\hat{x}_1$ $\hat{x}_2$ $\hat{x}_3$ • • • $\hat{x}_{N-1}$ $\hat{x}_N$
SAME
ACTUAL VALUE OF CV
PREDICTED FUTURE VALUES OF CV, AFTER FEEDFORWARD CORRECTION
x_0 $\hat{x}_1$ $\hat{x}_2$ $\hat{x}_3$ • • • $\hat{x}_{N-1}$ $\hat{x}_N$
CONTROL MOVE CORRECTION
COMPUTE AND APPLY CHANGE IN CONTROLLER OUTPUT,
Δm_0
ADJUST PREDICTED PROFILE TO ACCOUNT FOR Δm_0
THAT IS:
FOR K = 1 TO N.
$\hat{X}_K \leftarrow \hat{X}_K + P_K \Delta m_0$
ACTUAL VALUE OF CV
PREDICTED FUTURE VALUES OF CV, AFTER CONTROL MOVE CORRECTION
x_0 $\hat{x}_1$ $\hat{x}_2$ $\hat{x}_3$ • • • $\hat{x}_{N-1}$ $\hat{x}_N$
SAME

Figure 8.
Stored Values and Computation During a Typical MPC Sample and Calculation Cycle.

UNCONSTRAINED MPC FOR MIMO PROCESSSES

Conceptually, it is a simple matter to scale up from the single-input, single-output process used in the previous development to a process with multiple inputs, multiple outputs and multiple disturbances, as shown in Figure 9.

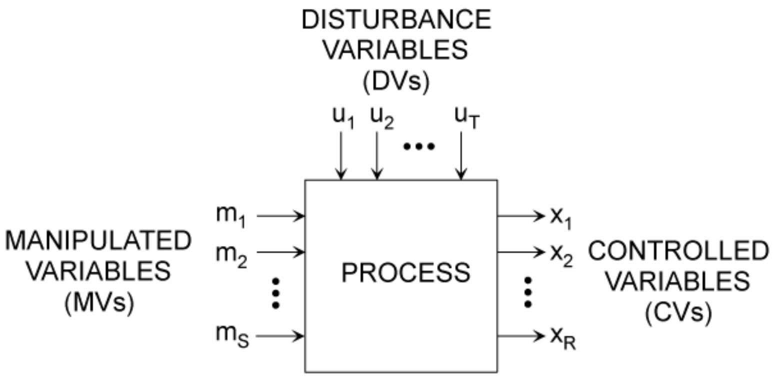
DISTURBANCE VARIABLES (DVs)
u_1 u_2 u_T
MANIPULATED VARIABLES (MVs)
m_1
m_2
m_S
PROCESS
x_1
x_2
x_R
CONTROLLED VARIABLES (CVs)

Figure 9
Multiple-Input, Multiple-Output Process.

In this section, we shall assume that

R = number of CVs
S = number of MVs
T = number of DVs

In practice, R may equal S; that is, the process control system may be "square" system. We will continue to let K represent the number of sample instances in our control horizon, and N the number of sample instances in our prediction horizon.

From each MV to each CV there will be a step response model similar to Figure 1. (Some may be null; that is, not every MV will affect all of the CVs.) These models are designated $\mathbf{p}_{ij}$, where subscript "i" represents "to CV" and subscript "j" represents "from MV". Hence

For $i = 1,\ldots,R;\ j = 1,\ldots,S$
$$\mathbf{p}_{ij} = \begin{bmatrix} p_{ij,1} \\ p_{ij,2} \\ \vdots \\ p_{ij,\mathrm{N}} \end{bmatrix} \qquad (22)$$

and

$$\mathbf{P}_{ij} = \begin{bmatrix} p_{ij,1} & 0 & 0 & \cdots & 0 \\ p_{ij,2} & p_{ij,1} & 0 & \cdots & 0 \\ p_{ij,3} & p_{ij,2} & p_{ij,1} & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ p_{ij,K} & p_{ij,K\text{-}1} & & & p_{ij,1} \\ \vdots & \vdots & & & \vdots \\ p_{ij,N} & p_{ij,N\text{-}1} & \cdots & \cdots & p_{ij,N\text{-}K+1} \end{bmatrix} \tag{23}$$

From each DV to each CV there will be a similar step response model (some may be null), designated $\mathbf{d}_{ik}$, where subscript "*i*" represents "to CV" and subscript "*k*" represents "from DV".

For $i = 1,\ldots, R;\ k = 1,\ldots,T$

$$\mathbf{d}_{ik} = \begin{bmatrix} d_{ik,1} \\ d_{ik,2} \\ \vdots \\ d_{ik,N} \end{bmatrix} \tag{24}$$

Vectors representing the current values and predicted profiles of the CVs are:

For $i = 1,\ldots, R$

$$\hat{\mathbf{x}}_{i,0} = \begin{bmatrix} \hat{x}_{i,1,0} \\ \hat{x}_{i,2,0} \\ \vdots \\ \hat{x}_{i,N,0} \end{bmatrix} \qquad \mathbf{x}_{i,K} = \begin{bmatrix} x_{i,1,K} \\ x_{i,2,K} \\ \vdots \\ x_{i,N,K} \end{bmatrix} \tag{25}$$

Vectors representing future control moves are

For $j = 1$ to S

$$\mathbf{\Delta m}_j = \begin{bmatrix} \Delta m_{j,0} \\ \Delta m_{j,1} \\ \\ \Delta m_{j,K-1} \end{bmatrix} \tag{26}$$

Using these definitions, the predicted profile for each of the CVs is given by the following equation:

For $i = 1$ to R

$$\hat{\mathbf{x}}_{i,K} = \hat{\mathbf{x}}_{i,0} + \sum_{k=1}^{\mathrm{T}} \mathbf{d}_{ik} \Delta u_k + \sum_{j=1}^{\mathrm{S}} \mathbf{P}_{ij} \mathbf{\Delta m}_j \tag{27}$$

Now define the following "super vectors" (vector of vectors) and "super matrix" (matrix of matrices). (The size of each vector or matrix is indicated.):

$$\hat{\mathbf{x}}_{\mathbf{K}} = \begin{bmatrix} \hat{\mathbf{x}}_{1,K} \\ \hat{\mathbf{x}}_{2,K} \\ \vdots \\ \hat{\mathbf{x}}_{R,K} \end{bmatrix} \quad \hat{\mathbf{x}}_0 = \begin{bmatrix} \hat{\mathbf{x}}_{1,0} \\ \hat{\mathbf{x}}_{2,0} \\ \vdots \\ \hat{\mathbf{x}}_{R,0} \end{bmatrix} \quad \Delta\mathbf{u} = \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \\ \vdots \\ \Delta u_T \end{bmatrix} \quad \Delta\mathbf{m} = \begin{bmatrix} \Delta\mathbf{m}_1 \\ \Delta\mathbf{m}_2 \\ \\ \Delta\mathbf{m}_S \end{bmatrix}$$

$$(RN \times 1) \qquad (RN \times 1) \qquad (T \times 1) \qquad (SK \times 1)$$

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \dots & \mathbf{P}_{1S} \\ \mathbf{P}_{21} & \mathbf{P}_{22} & \dots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{P}_{R1} & \mathbf{P}_{R2} & \dots & \mathbf{P}_{RS} \end{bmatrix} \quad \mathbf{D} = \begin{bmatrix} \mathbf{d}_{11} & \mathbf{d}_{12} & \cdots & \mathbf{d}_{1T} \\ \mathbf{d}_{21} & \mathbf{d}_{22} & \dots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{d}_{R1} & \mathbf{d}_{R2} & \cdots & \mathbf{d}_{RT} \end{bmatrix}$$

$$(RN \times SK) \qquad (RN \times T)$$

Analogous to equations 20 and 21 we have:

$$\hat{\mathbf{x}}_{\mathbf{K}} = \hat{\mathbf{x}}_0 + \mathbf{D}\Delta\mathbf{u} + \mathbf{P}\Delta\mathrm{m} \tag{28}$$

$$\hat{\mathbf{e}}_{\mathbf{0}} = \mathbf{x}_{\mathbf{SP}} - \hat{\mathbf{x}}_{\mathbf{0}} - \mathbf{D}\Delta\mathbf{u} \tag{29}$$

After minimizing the sum of squares of the errors, the vector of current and future control moves is given by:

$$\Delta\mathbf{m} = \mathbf{W}\mathbf{e}_{\mathbf{0}} \tag{30}$$

where
$$\mathbf{W} = \left[\mathbf{P}^{\mathrm{T}}\mathbf{P} + \mathbf{Q}\right]^{-1}\mathbf{P}^{\mathrm{T}} \tag{31}$$

Note that not every element of $\Delta\mathbf{m}$ needs to be computed; only the current move for each of the MVs is required. Therefore we can compute:

For $j = 1$ to S
$$\Delta m_{j,0} = \mathbf{w}_{\boldsymbol{j}}^{\mathrm{T}}\hat{\mathbf{e}}_{\mathbf{0}} \tag{32}$$

where $\mathbf{w}_{\boldsymbol{j}}^{\mathrm{T}}$ is the jth row vector of the matrix **W**.

CONSTRAINED MPC

In real-world applications, there will be constraints on process variables, manipulated variables and auxiliary variables. There may also be constraints on the rate of change of these variables. Furthermore, some of the constraints may be hard constraints; others may be soft constraints. Hard constraints are established by physical limits. Examples of hard constraints are valves which cannot go beyond saturation limits or controllers whose set point cannot be moved outside the measured range. Soft constraints are based process design, equipment limits and safety considerations. As long as there is a feasible solution (i.e., an operating point) that satisfies all constraints, then hard and soft constraints can be treated equally. However, there may be no feasible solution which satisfies all constraints simultaneously. If these if some of these limits are soft constraints, then one strategy is to permit violation of each of the soft constraints by a small amount. But rather than leave it up to the process operator to make an *ad hoc* decision as to how much each constraint can be violated, it may be preferable to have a "graceful violation" of each limit in a planned fashion (perhaps up to some other hard limit).

First, assume that there is a feasible solution which will satisfy all the constraints. The objective will be to minimize the functional J, subject to constraints:

$$\min_{\Delta \mathbf{m}} J = \frac{1}{2}\left(\hat{\mathbf{e}}^{\mathrm{T}}\hat{\mathbf{e}} + \Delta\mathbf{m}^{\mathrm{T}}\mathbf{Q}\Delta\mathbf{m}\right)$$

$$\text{Subject to:} \qquad \begin{aligned} \mathbf{X_L} &\le \hat{\mathbf{x}} \le \mathbf{X_U} \\ \mathbf{M_L} &\le \mathbf{m} \le \mathbf{M_U} \\ \mathbf{Y_L} &\le \hat{\mathbf{y}} \le \mathbf{Y_U} \end{aligned} \qquad (33)$$

where $\hat{\mathbf{y}}$ refers to the predicted value of auxiliary variables and the subscripts "L" and "U" refer to lower and upper limits for each class of variable. If there are rate of change limits, these should also be included in the constraint set.

A problem of the type described by equation 33 is called a "quadratic programming" (QP) problem. There are standard iterative techniques for solving problems of this type; most MPC vendors include a QP solver within their software package.

Another abnormal condition is when there are fewer MVs than there are CVs. There are not enough degrees of freedom to control all of the CVs to their set points. In this case, one possible strategy is to assign priority variables (between 0 and 1) to each of the CVs. Those with a higher priority will be controlled closer to their set points.