MODEL PREDICTIVE CONTROL

An Introduction

1. Introduction

Model predictive controller (MPC) is traced back to the 1970s. It started to emerge industrially in the 1980s as IDCOM (Richalet et. al.) and DMC (Cutler and Ramaker). The initial IDCOM and MPC algorithms represented the first generation of MPC technology. Generally, MPC is a family of controllers in which there is a direct use of an explicit identifiable model. It is also described as a class of computer control schemes that utilizes a process model for two central tasks:

- Explicit prediction of future plant behavior
- Computation of appropriate corrective control action required to drive the predicted output as close as possible to the desired target value.

Control design methods based on MPC concept have found wide acceptance in industrial applications and have been studied by academia. It is currently the most widely used of all advanced control methodologies in industrial applications. The reason for such popularity is the ability of MPC design to yield high performance control systems capable of operating without expert intervention for long periods of time.

2. Motivation

The typical goals of process control are:

- Disturbance rejection to decrease variability in the key variable
- Improve the operation of a process, the productivity of the plant, the quality of the product.
- Stable and safe operation.

While achieving the above tasks, consideration should be continuously given to the following issues:

- *dynamic and unpredictable marketplace conditions*: It is generally accepted that the most effective way to generate the most profit out of the plant while responding to marketplace variation with minimal capital investment is provided by the integration of all aspects of automation.
- *safety and environmental regulations*: Some process variables must not violate specified bounds for reasons of personnel or equipment safety or because of environmental regulation. Safety measures and environmental regulations are continuously changing.

To develop better, fast, accurate and robust process control, model-based modern control algorithms and efficient adaptive and learning techniques are required.

The requirement for consistent attainment of high product quality, more efficient use of energy, and an increasing awareness of environmental responsibilities have all combined to impose far restrict demand on control systems than can be met by traditional techniques alone. The industrial response to these challenges led to the development of the successful MPC algorithm. It had an enormous impact on industrial process control and served to define the industrial MPC paradigm.

3. Benefits of MPC

- Most widely used control algorithm in material and chemical processing industries.
- Increased consistency of discharge quality. Reduced off-specs products during grade changeover. Increased throughput. Minimizing the operating cost while meeting constraints (optimization, economic).
- Superior for processes with large number of manipulated and controlled variables (Multivariable, strong coupling)
- Allows constraints to be imposed on both MV and CV. The ability to operate closer to constraints. (constraints)
- Allow time delays, inverse response, inherent nonlinearities (difficult dynamics), changing control objectives and sensor failure. (predictive)

4. Industrial implementation

Thee are more than 2000 successful industrial implementations have been reported. According to recent reports, the number of applications is expected to continue to increase rapidly. Most of these applications (67%) are in refineries where MPC was originally born. However, significant number of applications can also be found in other areas. The following is an example of the areas where MPC has been active.

- Super heater, steam generator, utility boiler,
- FCCU (Fluid Catalytic Cracking Unit),
- batch reactor,
- Pulp and paper industries.
- Hydrocracker reactors
- Distillation columns
- Polymer extruder
- Olefin plants

5. Commercial MPC software

There are several versions (generations) of the MPC depending on the vendor or developer group. These different versions are very similar in principles. They may differ in implementation procedure, model type, objective function formulation, or method of solution. However, not all of these control methodologies has reached the level of

1. **IDCOM (Identification and Command), 1987**

- Marketed by Set point, Inc, USA
- Methodology: Model algorithmic control
- Model type: Impulse response
- Optimization is solved by QP approach
- Single tuning parameter "reference time"
- Available for direct hardware implementation to Honeywell DCS systems through a Phoenix interface written by Set point Inc.

2. **DMC (Dynamic Matrix Control), 1985**

- Developed by Shell Co. and Marketed by DMC corporation.
- Methodology: Dynamic Matrix control
- Model: Step response
- Optimization problem is solved by LP approach.
- Four tuning parameters with one major.
- Available for direct communication with Honeywell DCS systems through communication interface.

3. **OPC (Optimum Predictive Control), 1987**

- Marketed by Treiber Controls, Inc.
- Use step response, but with different identification method.
- Uses similar algorithm to that of DMC and solves LP problem.
- Can be implemented on Bailey's controller without communication interface.
- Model building, controller design and simulation tasks are carried out on Personal computers.

4. **PCT (Predictive Control Technology), 1984**

- Marketed by Profimatics, Inc.
- Combines the aspects of IDCOM and DMC
- The optimization problem is solved for only one control move.
- Implementation platform is similar to IDCOM and DMC.

5. **HMPC (Horizon multivariable Predictive Control), RMPCT (Robust Model Predictive Control Technology), 1991**

- Developed and marketed by Honeywell.
- Fundamentally different from that of other MPC schemes.
- For proprietary reasons, there are many aspects of the algorithm that are currently unavailable.

The above list includes some of the well-known software technologies. However, due to the success of MPC, the number of these control technologies is growing as each DSC system vendor started to develop their own MPC scheme.

Overall, DMC scheme is the most widely used industrially. It is reported that DMC has over 600 industrial applications.

6. Theory and Fundamentals

6.1 General concept

Block diagram for the MPC implementation is shown in Figure 1. As shown in the figure, a process model is used in parallel to the plant. MPC uses a dynamic model of the process in order to predict the controlled variable. The predicted controlled variable is fed back to the controller where it is used in an on-line optimization procedure, which minimizes an appropriate cost function to determine the manipulated variable. The controller output is implemented in real time and then the procedure is repeated every sampling time with actual process data. The difference between the plant measurement, y_p and the model output y_m is also fed to the controller to eliminate steady state offset. Usually the cost function depends on the quadratic error between the future reference variable and the future controlled variable within limited time horizon.

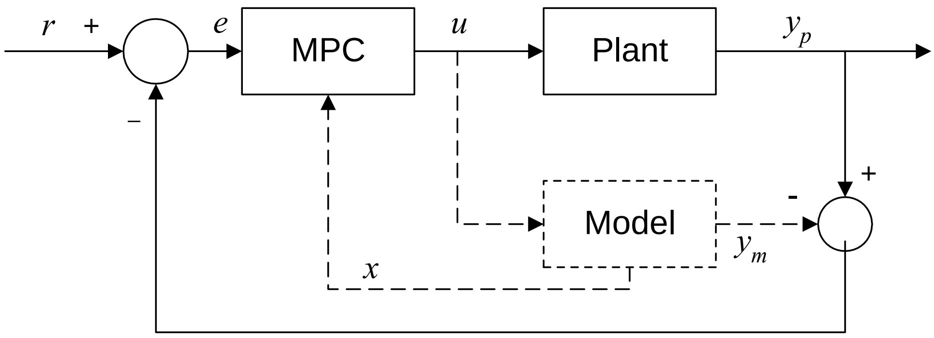

Figure 1: Block diagram for MPC implementation

6.2 The receding horizon concept

MPC is a digital controller, i.e. a discrete time technique. The control calculation is computed via a digital computer and the result is implemented online each sampling time. As mentioned earlier this procedure is repeated in a moving horizon approach. The concept of moving horizon can be understood from Figure 2. Assume we are at certain sampling time k. The past trend for the output (y) up to k and input (u) up to k-1 are known. The objective is then to find the future trend for the input (control actions) that moves the future trend of the output approaches the desired reference trajectory $r(k+1)$. The control actions are found through iteration. In fact, an optimization problem is solved to compute online and in real-time the open loop sequence of present and future control moves [$u(k/k)$, $u(k+1/k)$ … $u(k+M-1/k)$], such that the predicted outputs [$y(k+1/k)$ $y(k+2/k)$ … $y(k+P/k)$] follow the predefined trajectory. The optimization is solved taking into consideration constraints on the outputs and inputs. The first control action $u(k/k)$ is then picked and implemented on the real plant over the interval [$k,k+1$]. In the method, M is known as the control horizon and P as the prediction horizon.

At the next sampling time k+1, the prediction and control horizon are shifted ahead by one step and a new optimization problem is solved using updated measurements from the process. Thus, by repeatedly solve an open-loop optimization problem with every initial conditions updated at each time step, the model predictive control strategy results in a closed-loop constrained optimal control technique.

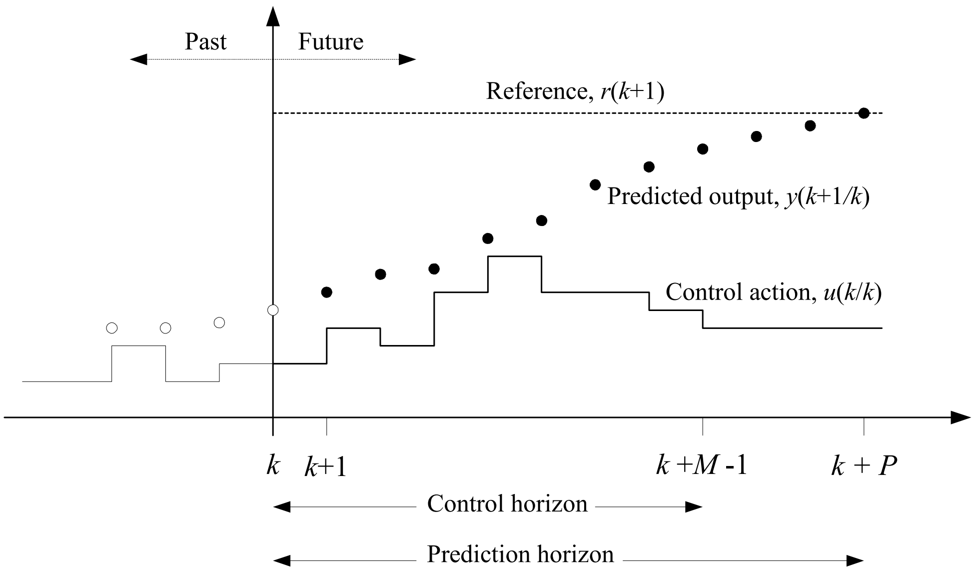

Figure 2: Receding horizon control scheme

6.3 Mathematical formulation

The above control techniques can also be formulated mathematically. The MPC concept is a computer-control technique. In this case, the control law is expressed mathematically by an optimization problem. The optimization problem is coded in a digital computer and solved numerically online at real time. The optimization problem is represented by the following objective function:

$$\min_{\Delta U(k/k)} [Y(k+1/k) - R(k+1/k)]^T \Gamma [Y(k+1/k) - R(k+1/k)] + [\Delta U(k/k)^T \Lambda \Delta U(k/k)] \qquad (1)$$

subject to:

$$U^l \le U(k) \le U^u \qquad (2)$$

$$\Delta U^l \le \Delta U(k) \le \Delta U^u \qquad (3)$$

Note that the last two equations represent the physical constraints on the input and their moves. They can be combined and formulated as constraints on the input move and then added to the optimization problem. This treatment is more suitable because the objective function contains only input moves. Inputs are not optimized because they create steady state offset.

In the above optimization problem, Γ *and* Λ are diagonal weight matrices. Their role will be explained under Tuning section. $R(k+1/k) = [r(k+1/k)\ r(k+2/k)\ \ldots\ r(k+P/k)]$ is a vector of dimension $ny \times P$, which contains the set point for all ny outputs over the prediction horizon P. In his case, $r(k+1)$ is a vector of all ny outputs at time $k+1$ Similarly, $\Delta U(k/k)$ is a vector of dimension $nu \times M$, which contains the input move for all nu inputs over the control horizon M and is defined as follows:

$$\Delta U(k/k) = [\Delta u(k/k)\ \Delta u(k+1/k)\ \ldots\ \Delta u(k+M-1/k)] \tag{3}$$

In this case, $\Delta u(k/k)$ is a vector of all nu inputs at time k. In fact, $\Delta U(k/k)$ is the unknown control actions that need to be found through solving the above optimization problem. $Y(k+1/k)$ is a vector of dimension $ny \times P$, which contains all ny outputs over the prediction horizon P and is defined as follows:

$$Y(k+1/k) = [y(k+1/k)\ y(k+2/k)\ \ldots\ y(k+P/k)]$$

Therefore, $y(k+1/k)$ is a vector of all ny outputs at time $k+1$. Generally, the output predictions Y is obtained from the recursive state equations:

$$Y(k+1/k) = M_p Y(k/k) + S_m^p \Delta U(k/k) \tag{4}$$

M_p is a constant matrix of dimension $ny \cdot P \times ny \cdot n$ and S_m^p is known as the dynamic matrix. The concept dynamic matrix and the truncation number, n are discussed in the next section.

Note that the second index (k) in all above equations and definitions denotes that these values are computed at real time k. this index changes with the moving horizon. According to the definition of the predicted output from equation 4, the optimization problem (Eqs. 1-3) can be written in terms of the unknown variable $\Delta U(k)$. Therefore, if the constraints (Eqns. 2-3) are excluded, the optimization problem can be solved analytically and closed-form solution can be written as follows:

$$\Delta U(k) = K_{mpc}[M_p Y(k) - R(k+1)] \tag{5}$$

Here, K_{mpc} is known as the MPC controller gain matrix, which is a combination of the dynamic matrix, and the weight matrices Γ and Λ. In this case, online solution of the optimization problem is not necessary. Instead, the controller gain can be computed offline and then equation (5) is executed online to determine the new values for the control action. However, in most cases the input constraints are included in the optimization. Therefore, analytical solution can not be found. Instead the optimization problem can be cast as Quadratic Programming (QP) or linear programming (LP) and then solved by available optimization software. It should be noted that no output constraints are imposed in the optimization problem. Although the MPC formulation allows using output constraints, it is usually excluded because it increases the computational effort and time. Moreover, inclusion of output constraints may cause the optimization problem to run into infeasible solution.

Output feedback:

In the mathematical treatment above, the concept of feedback was not highlighted explicitly. If the output prediction obtained from equation 4 is fed directly to equation 1, then the MPC control action depends only on the model outputs and not on the plant measurement. In due course, unless the model is extremely perfect, the control performance will be poor and even unstable. Nevertheless, the standard MPC concept allows for feedback correction. This means that the output prediction from equation 4 is corrected for model-plant mismatch before being implemented. In fact, the model states are first updated by the last control action as follows:

$$Y(k/k-1) = M_n Y(k-1/k-1) + S\Delta u(k-1/k-1) \quad (6)$$

The computed states are then corrected by the current plant measurement as follows:

$$Y(k/k) = Y(k/k-1) + I_n[y_p(k) - NY(k/k-1)] \quad (7)$$

The corrected output, $Y(k/k)$ is then inserted in equation 4. This procedure is repeated every sampling time to account for new plant information. In the above equations, M_n is a constant matrix of dimension $ny \cdot n \times ny \cdot n$, S is the step response coefficient matrix as will be explained in the next section, I_n is constant identity matrix used to project the output feedback term into n points, N is another constant matrix that extract the first ny elements out of the vector $Y(k/k-1)$, and y_p is the plant measurement. Note that since no future information about the plant measurement is available, the current output feedback, i.e. $[y_p(k) - NY(k/k-1)]$, is used to correct the future predictions. This treatment is a common concept for all MPC technologies.

Note that this feedback mechanism is a built-in feature of all MPC formulations. Inclusion of the output feedback mechanism is the main reason for repeating the overall MPC control calculation is each sampling time. It is also the main feature that differentiates the MPC from the classical optimal control theory.

In some cases, when the process is highly nonlinear or open-loop unstable, correcting the predictions by a constant output feedback may lead to poor performance. Consequently, researchers have developed different schemes to improve the robustness of MPC. For example, ramp-like estimates of the output feedback, kalman filtering, state estimation, and state observer can be used.

7. Developing models for MPC (identification)

7.1 MPC model forms

As shown earlier, the model predictive control relies on a process model. The type of process model used varies with the type and formulation of MPC technology. The concept of MPC is not limited to any particular model form, but the model form chosen, strongly affects the implementation and computation requirements. The step of choosing and designing appropriate process model is the heart of the MPC design. In fact, modeling effort can take up to 90% of the cost and time. Finite impulse response models FIR, Finite step response models FSR, nonlinear state space models, and neural network models can be used. Most of the commercial MPC software use convolution models such as FIR or FSR models. FIR and FSR are easier to develop than other types of models. In addition these convolution models have the following advantages:

- Model coefficients can be easily obtained from the experimental step test.
- No model structure or order needs to be assumed.
- Unusual dynamics can be modeled.
- Provide convenient way to design a controller based on the use of optimization.
- The model is linear which simplify the solution of the optimization problem.
- They can be deduced form other parametric model forms.

Despite the advantages of the convolution models some fundamental problems related to convolution models exist:

- Designing the Test signal to be used for stepping the process.
- Use too many parameters (overparameterization) which become even more pronounced in multivariable case.
- Integrator and unstable processes can be modeled.

It should be emphasized that the convolution models are linear, which allow simplifying the optimization problem into LP or QP. The latter can be solved easily or moderately online. However, when more complicated or nonlinear model forms are used in the MPC design, the objective function (Eq. 1) becomes nonlinear. As a result nonlinear programming is required to solve the optimization problem.

7.2 FIR and FSR model design

These models can be obtained via one of the following two common experimental test schemes:

- Step testing
- PRBS (pseudo random binary sequence)

Model identification by step testing:

Process modeling by step testing is shown in figure 3. Simply, the specific dynamic process unit undergoes a series of unit step changes each at a time. For example, the first process input (MV) can be stepped by unit value while the remaining inputs kept constant at their nominal values. No automatic control or manual operator intervention is allowed during the test. The process outputs (CV) responses are recorded with time till they reach a steady state. The step response coefficients can be deduced directly from the unit step response plot as shown in Figure 3. The step response vector for the first input can be developed as follows:

$$S_1 = \left[s_{0,1} s_{0,2} \cdots s_{0,ny} \cdots s_{1,1} s_{1,2} \cdots s_{1,ny} \cdots s_{n,1} s_{n,2} \cdots s_{n,ny} \right]^T$$

where $s_{0,i}$ are the initial step response values and they equal zero because deviation variables are used in linear models. The term n, is known as the truncation point and it is determined such that the full open-loop response has reached steady sate. It is also defined as the open-loop settling time. The above step test is repeated for each individual input and finally the full step response coefficient matrix can be developed as follows:

$$S = [S_1 S_2 \cdots S_{nu}]$$

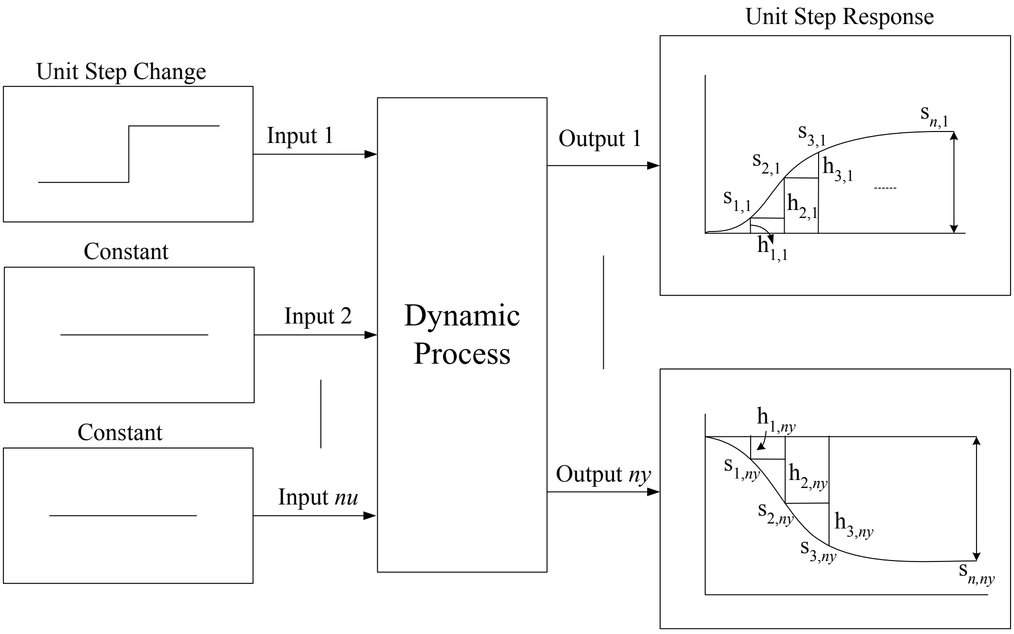

Figure 3: Step Testing procedure

This matrix has a dimension of $ny \cdot n \times nu$ and is used in equation 6. The dynamic matrix, S_m^p, has a dimension of $ny \cdot P \times nu \cdot M$ and it is a special arrangement of the matrix *S*. The dynamic matrix contains the first $ny \cdot P$ rows of the *S* matrix. The original *nu* columns are repeated *M* times with each time the step response coefficient are delayed by one unit and so on.

One can develop the FIR model from the FSR model directly noting the following relation:

$$h_{i,1} = s_{i,1} - s_{i-1,1}$$

Or otherwise, they can be deduced directly from the unit step response plot shown in Figure 3. Similar procedure to that used for developing FSR can be used to develop the overall dynamic matrix.

Problems related to step testing:

- The product quality may be disturbed by stepping the MVs.
- The test time is very long, which occupies much manpower and makes production planning difficult.

- The tests are done manually, which dedicates extremely high commitments of the engineers and operators.
- The overall multivariable model is generated by merging the individual SISO identification.

The SISO identification implies that the input channels are perturbed one at a time. The accuracy of the individual SISO models may not yield a multivariable model of required accuracy.

Identification by PBRS testing:

It appears that PBRS signals are the primary test signals used by identification packages. Usually a pseudo binary random sequence of step-like signals are generated for each input channel at a time (SISO) or to all input channels simultaneously (MIMO). These signals are applied to the plant and the output response is recorded as shown in Figure 4. The resulted output response can then be fitted to a parametric model from which an FIR or FSR models can be inferred. The main feature of the PBRS method is the MIMO identification where it fits a single model for all outputs simultaneously which account for any existing correlation. Other advantages of PBRS are reduction of disturbance to product quality and eliminate waiting for reaching steady state. However, some knowledge about time series analysis and identification theory is required.

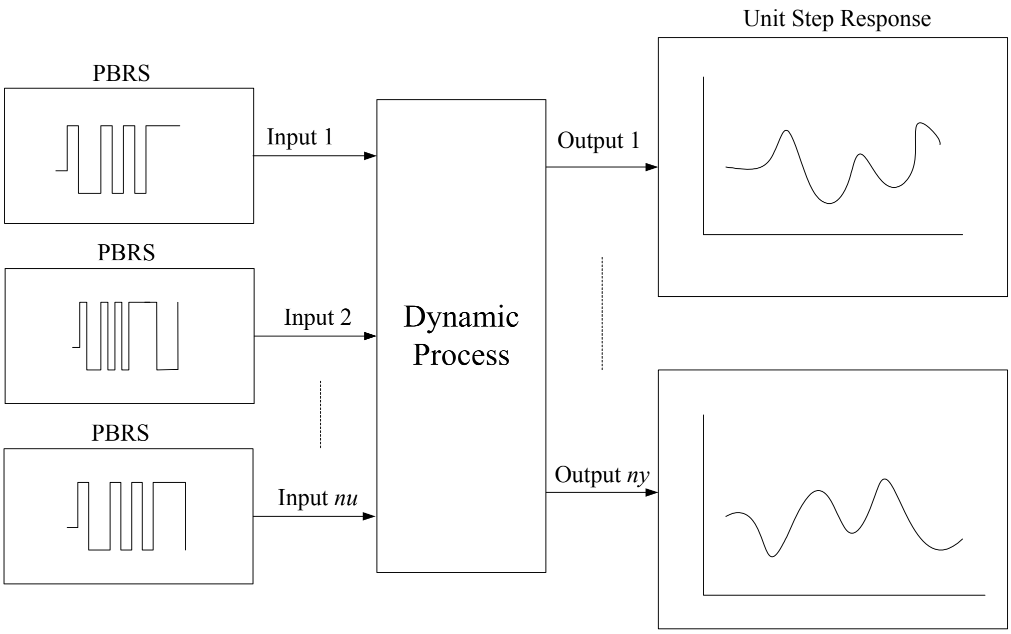

Figure 4: PBRS testing procedure

Without worrying about which identification method to be used, all the commercial MPC packages are equipped with their own identification software. The software can be used by engineers with the help of software specialist. Usually the plant test is run 24 hours a day with engineer monitoring the plant. Each MV is stepped 8 to 15 times with the output (CV) signal to noise ratio at least six. The whole plant test may take 10 to 15 days, depending on the time to steady state and number of variables of the unit.

8. **Tuning**

Basically, the MPC has four main tuning parameters namely; the control horizon *M*, the prediction horizon *P*, the output weight matrix Γ and the input weight matrix Λ. The last two matrices are diagonal ones. The sampling time has a strong impact on the control performance, however it is not used as a tuning parameter since it is often fixed based on the equipment at installation. Therefore, the total number of tuning parameters is $2+nu+ny$. These parameters have profound and somewhat overlapping effect on the closed-loop performance. The definition and function of each parameter is outlined next:

- The control horizon, *M* is the number of MV moves that MPC computes at each sampling time to eliminate the current prediction error. A large *M* has the advantage that it allows detection of constraints violation before they are reached, averages the control objective over time, and handle unknown variable time delays.

- The prediction horizon, *P,* represents the number of samples into the future over which MPC computes the predicted process variable profile and minimizes the prediction error. Usually $P>M$ is selected to avoid dead-band effect. Increasing *P*, result in a significant non-monotonic response ranging from damped to under-damped to damped.

- The move suppression factor (weight on the MV move, Λ). They indicate the trade-off between the amount of movement allowed in the manipulated variables and the rate at which the output deviation from set point is reduced over the prediction horizon. They serve dual purpose of suppressing aggressive control action and conditioning the system matrix before inversion.

- The weighting matrix Γ, is used primarily for scaling in the multivariable case; it permits the assignment of more or less weight for the objective of reducing the predicted error for the individual output variables. When a single γ is increased, the set point tracking response for the corresponding process variable has a faster rise time.

According to the understanding of the function of the MPC parameters, general guidelines are available (Moreshdi and Garcia, 1986; Garcia and Morari, 1982; Ohno et. al. 1988; Meadows and Rawlings, 1997). However, due to high nonlinearity of the process and/or the presence of modeling errors, MPC are commonly tuned by trail-and-error procedure.

In general, based on general reasoning, one can simplify the tuning problem to a single primary tuning parameter, which is Λ. Note that *P* can not be made independent of the sampling time. In addition, for stability reasons, *P* must be selected such that it includes the steady state effect of all past MV moves, i.e., it should equal the open-loop settling time (*n*) of the process in samples. Note also that the relative value of *M* to *P* is more important, therefore, *M* can be fixed at a small number to reduce computational effort. On the other hand, Γ do not affect the invertibility of the overall system matrix. Thus, they can be specified by the user on the basis of control objective priorities or saved for later fine-tuning.

More work on MPC tuning:
Recently, Shridhar and Cooper (1997,1998) developed an easy to use MPC tuning formula. However, their approach is limited to unconstrained MPC. Ali and Zafiriou (1993) presented an off-line optimization-based tuning procedure of MPC algorithms that utilizes non-linear process models. This procedure is computationally demanding. Al-Ghazzawi et. al. (2000) proposed an on-line adaptive strategy for constrained MPC. The algorithm utilizes the gradients of the closed-loop response with respect to the parameters to automatically determine on-line the new values of the parameters that drive the predicted output inside predefined time-domain constraints. This method is further modified by utilizing fuzzy logic theory (Ali and AlGhazzawi, 2001).

9. Real-time implementation structure

All MPC technologies can be easily implemented on fully installed DCS (Distributed Control systems) systems. The MPC can be applied in supervisory mode as shown in Figure 5. In this case, MPC is embedded in a hierarchy of control functions above a set of traditional PID loops. This is also known as cascade control configuration. Each slave loop uses PID, which acts directly on a specific final control element, i.e. valves, actuators. The master loop uses the MPC, which fixes the set point for the controlled variable of all the slave loops.

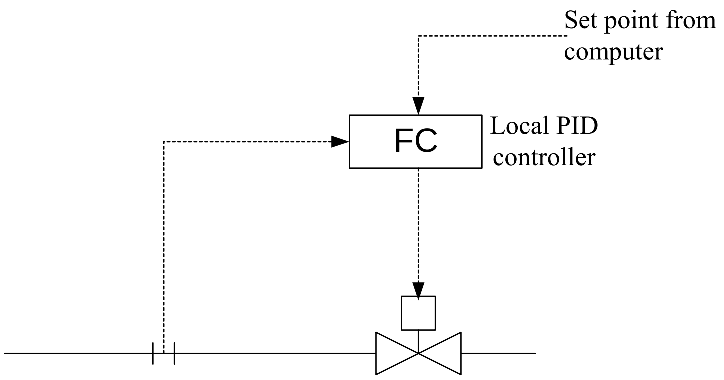

Figure 5: Example of supervisory control scheme

This implementation structure is very common for advanced industrial control applications. It allows for full integration of all control components of the plant. It allows for incorporating more than one advanced control algorithm and alternate between them. One can retrieve the original traditional control configuration any time by simply disabling the supervisory control system.

10. Simulation Example

A forced circulation evaporator is shown in figure 6. The process is originally proposed by Newell and Lee [see Ali & Ghazzawi, 2001] and is modeled as follows:

$$M_v \frac{dC_2}{dt} = F_1 C_1 - F_2 C_2$$
$$W \frac{dP_2}{dt} = F_4 - F_5$$

where C_1 and C_2 are the input and product compositions, respectively and P_2 is the operating pressure (kPa). F_1 and F_2 are the feed and product flow rates (kg/min), respectively. F_4 and F_5 are the vapor and condensate flow rates, respectively (kg/min). M_v is the liquid holdup in the evaporator (20 kg) and W is a constant (4 kg/kPa). The liquid level in the separator is considered well controlled by manipulating the product flow rate. Therefore, the flow rates are givens as follows:

$$F_2 = F_1 - F_4$$
$$F_4 = [0.16(F_1 + F_3)(-0.3126C_2 - 0.5616P_2 + 0.1538P_{100} + 41.57) -$$
$$F_1 C_p (0.3126C_2 + 0.5616P_2 + 48.43 - T_1)] / \lambda_{s1}$$
$$F_5 = \frac{2UA(0.507P_2 + 55 - T_{200})C_p F_{200}}{\lambda_{s2}(UA + 2C_p F_{200})}$$

where P_{100} is the steam pressure (kPa). F_{200} (kg/min) and T_{200} (^oC) are the flow and temperature of the cooling water, respectively. UA is the product of heat transfer coefficient and the transfer area (6.84 kW/K) and C_p is the heat capacity of the cooling water (0.07 kW/kg min). F_3 is the recycle flow rate (20 kg/min). λ_{s1} is the latent of steam at saturation condition (36.6 kW/Kg min) and λ_{s2} is the latent heat of evaporation of water (38.5 kW/kg min). F_1 is fixed at 10 kg/min and its temperature T_1 is fixed at 40 ^oC. The cooling water enters the cooler at 25 ^oC. The initial steady state value for the outputs is $C_2 = 0.1474$ and $P_2 = 32.109$ kPa which corresponds to $C_1 = 0.05$, $P_{100} = 200$ kPa and $F_{200} = 200$ kg/min. The control objective is to maintain the outputs within desired values using P_{100} and F_{200} as manipulated variables. This control problem is selected because it has strong cross-loop interaction. In addition, because mass fraction is used for C_1 and C_2, the process is ill conditioned. Both manipulated variables are constrained between 0 and 400.

A step response model is generated from the nonlinear process model using MATLB software with truncation number, $n = 30$ and sampling time $T_s = 0.5$ min. The step response model is used in the MPC controller algorithm. The full nonlinear model will be used to simulate the plant from which the output measurement will be obtained. This formulation generates model-plant mismatch, which makes the control problem more challenging.

Closed-loop simulations are carried out for demonstration purposes. Figure 7 shows the response for set point change for the product concentration from the initial steady state to a higher value of 0.18. The second loop is left open. The cooling water flow is used as the manipulated variable. It is obvious that the high product concentration is not achievable using one input. As the figure shows, the cooling water flow saturates at its maximum value to increase condensation. This situation led to a drop in the evaporator pressure. Since the steam pressure is not manipulated, the column pressure can not pick up its original condition. Figure 8 demonstrates how the closed-loop performance is improved when MIMO scheme is used. The set point for the column pressure is stepped by 1 kPa. The purpose of this simulation is to highlight the capability of MPC in handling multivariable interactive control loops. The MPC parameter values are $M = 1$, $P = 2$, $\Gamma = [1,1]$ and $\Lambda = [0,0]$. Figure 9 depicts the same set point change but for two different values for the prediction horizon while the rest of the MPC parameters are kept the same as before. It can be seen that increasing P creates slower response. In general, larger P leads to more robust and stable behavior. The effect of P can be considered marginal noting that the prediction horizon has increased by 25 folds. Figure 10 illustrates the effect of changing the other MPC parameter values. For example, the dashed line shows the performance becomes damped and sluggish for small increasing the move suppression. The dash-and-dot line, on the other hand, shows that increasing the weight on the first output, i.e. γ_1, the original performance is retrieved. This change in the control performance indicates clearly the overlapping effect of the MPC parameters.

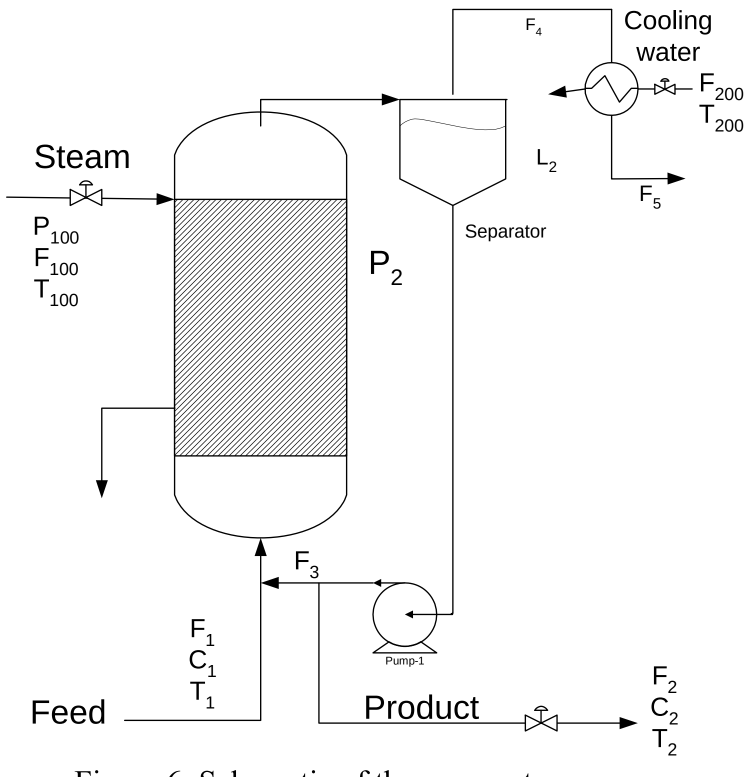

Figure 6: Schematic of the evaporator process

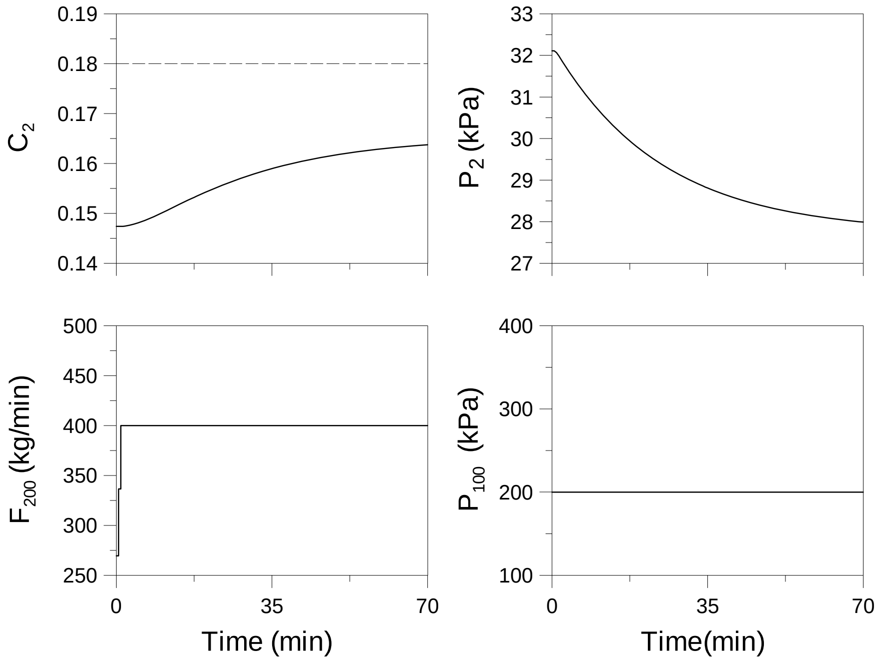

Figure 7: Set point change response for SISO control loop

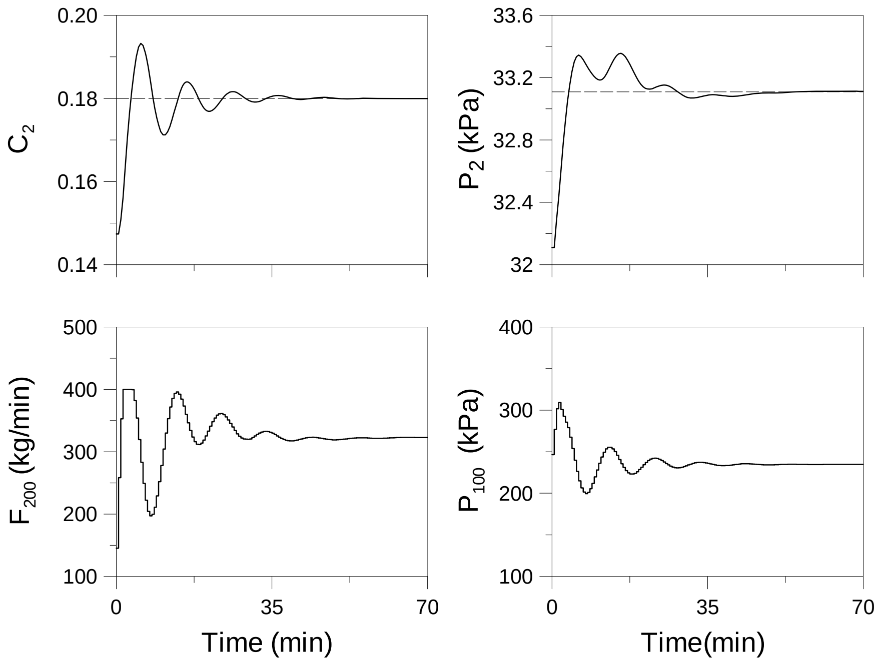

Figure 8: Set point change response for MIMO case

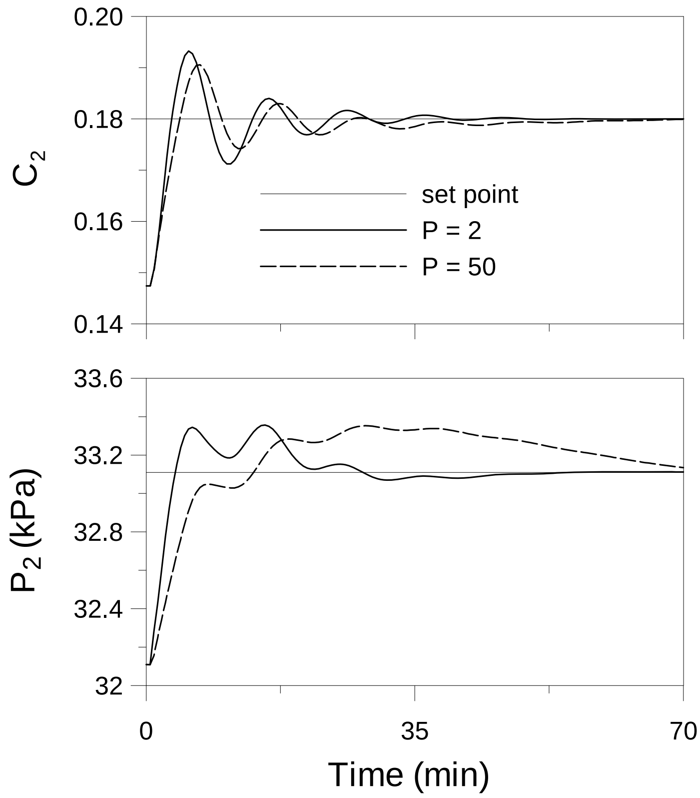

Figure 9: Set point change response for two different values of *P*.

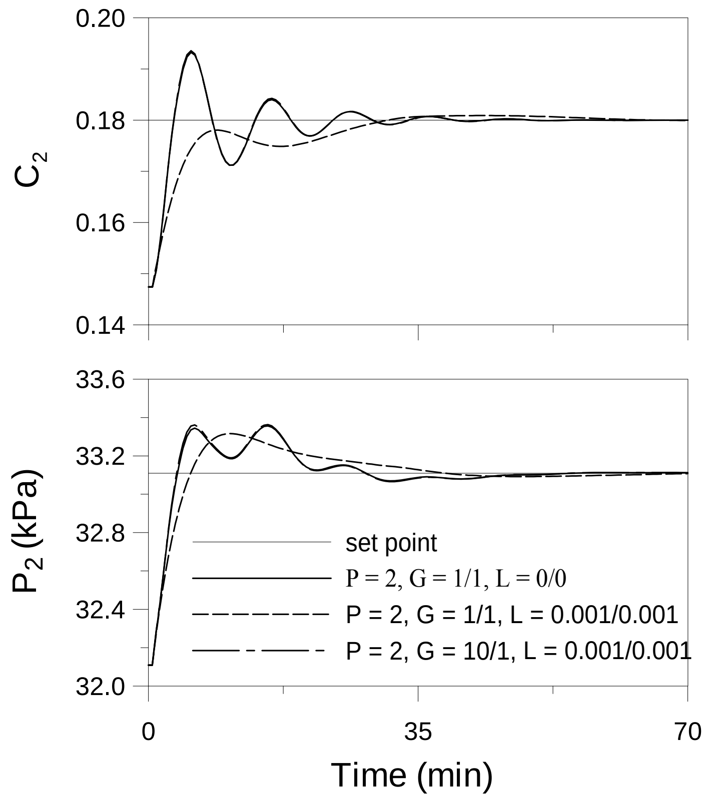

Figure 10: Set point change response for different values of the tuning parameters

References

Ali, E., and Zafiriou, E., "Optimization-based Tuning of Non-linear Model Predictive Control with State Estimation", *J. of Process Control,* **3,** 97-107, 1993.

Ali. E. and Ashraf Al-Ghazzawi, "On-Line Tuning of Model Predictive Controllers Using Fuzzy Logic", submitted to IEEE transaction on Fuzzy systems, 2001.

Al-Ghazzawi, A., Ali, E., Nouh, A., and Zafiriou, E., "On-line Tuning Strategy for Model Predictive Controllers", *Journal Process Control,* **11**(3), 265-284, 2001.

Cutler, C.R., and Ramaker, D. L., "Dynamic Matrix Control- A Computer Control Algorithm", Proceeding of JACC, San Francisco, CA , 1980.

Garcia, C., and Morari, M., "Internal Model Control 1: A Unifying Review and Some New Results", *Ind. Eng. Chem. Proc. Des. Dev*., **21**, 309-323, 1982.

Meadows, E. S., and Rawlings, J. B., "Model Predictive Control", In: Henson, M. A., and Seborg, D. E., (Eds), Nonlinear Process Control. Prentice-Hall, Englewood Cliff, NJ, 1997.

Morshedi, A., and Garcia, C., "Quadratic Programming Solution of Dynamic Matrix Control (QDMC)", *Chem. Eng. Comm.*, **46**, 73-87, 1986.

Ohno, H., Ohshima, M., and Hashimoto, I., "A Study on Robust Stability of Model Predictive Control", *IFAC Model Based Process Control*, GA, USA, 1988.

Richalet, J., Rault, A., Testud, J. and Papon, J., "Model Predictive /heuristic Control: Applications to Industrial Processes", *Automatica*, 14, 413-428, 1978.

Shridhar, R., and Cooper, D. J., "A Tuning Strategy for Unconstrained SISO Model Predictive Control", *Ind. & Eng. Chem. Res*., **36**, 729-746, 1997.

Shridhar, R., and Cooper, D. J., "A Tuning Strategy for Unconstrained Multivariable Model Predictive Control", *Ind. & Eng. Chem. Res*., **37**, 4003-4016, 1998.

Qin, S. J. and T. A. Badgwell, "An Overview of Industrial Model Predictive Control Technology", *Chem. Process Control*, Jan, 1-31, 1996.

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

Morari, M. and J. Lee, "Model Predictive Control: Past, Present and Future", *Comp. Chem. Eng*., **21**, 667-682, 1999.