CHEN4011
Advanced Modelling and COntrol

Curtin University

Dr. Ranjeet Utikar (RU)

Curtin University
Malaysia

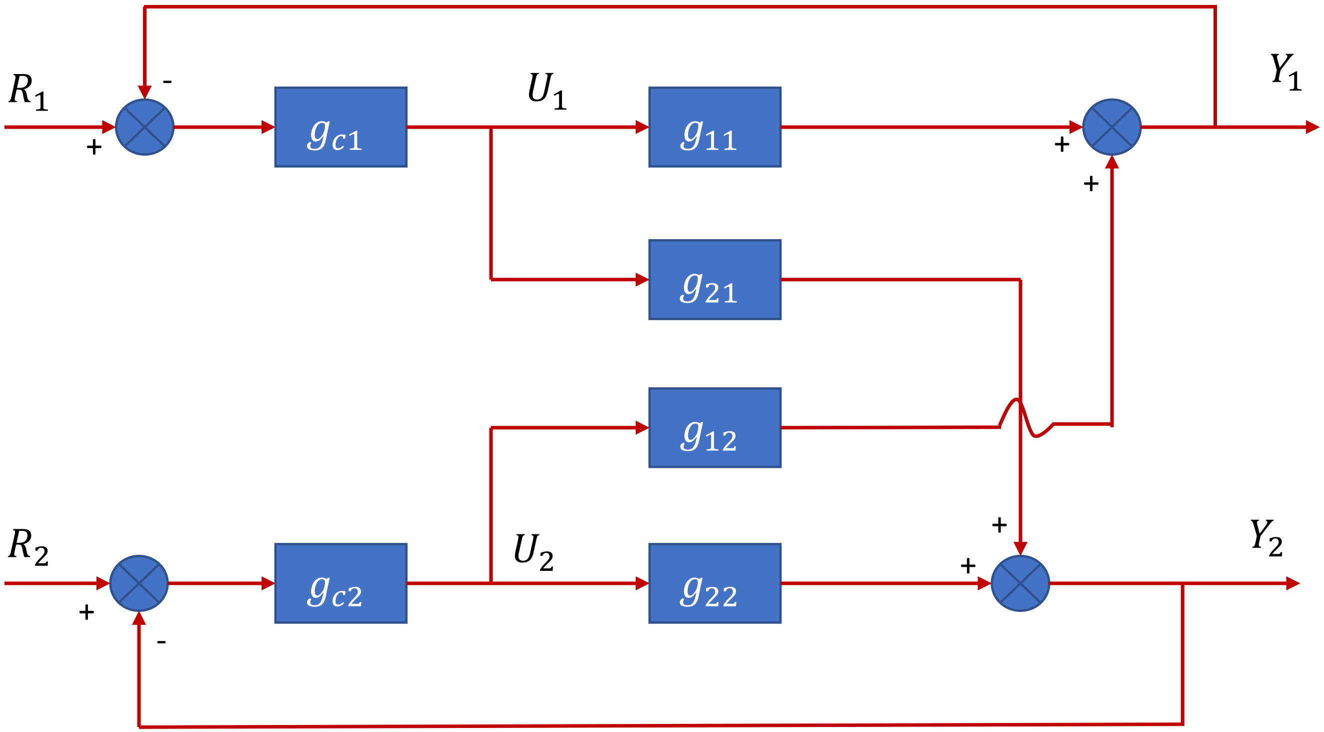
Dr. Jobrun Nandong (JN)

Multivariable Centralized Control: MPC

Outline

- ❑Decoupling control system
- ❑Multi-loop controllers vs. Multivariable Controllers
- ❑Model Predictive Control (MPC)
- ❑Dynamic Matrix Control (DMC)
- ❑Discrete-Time Step Response Model (DTSRM)
- ❑Moving Horizon Algorithms
- ❑Prediction Vector
- ❑DMC Control Law
- ❑Summary

Decentralized Control



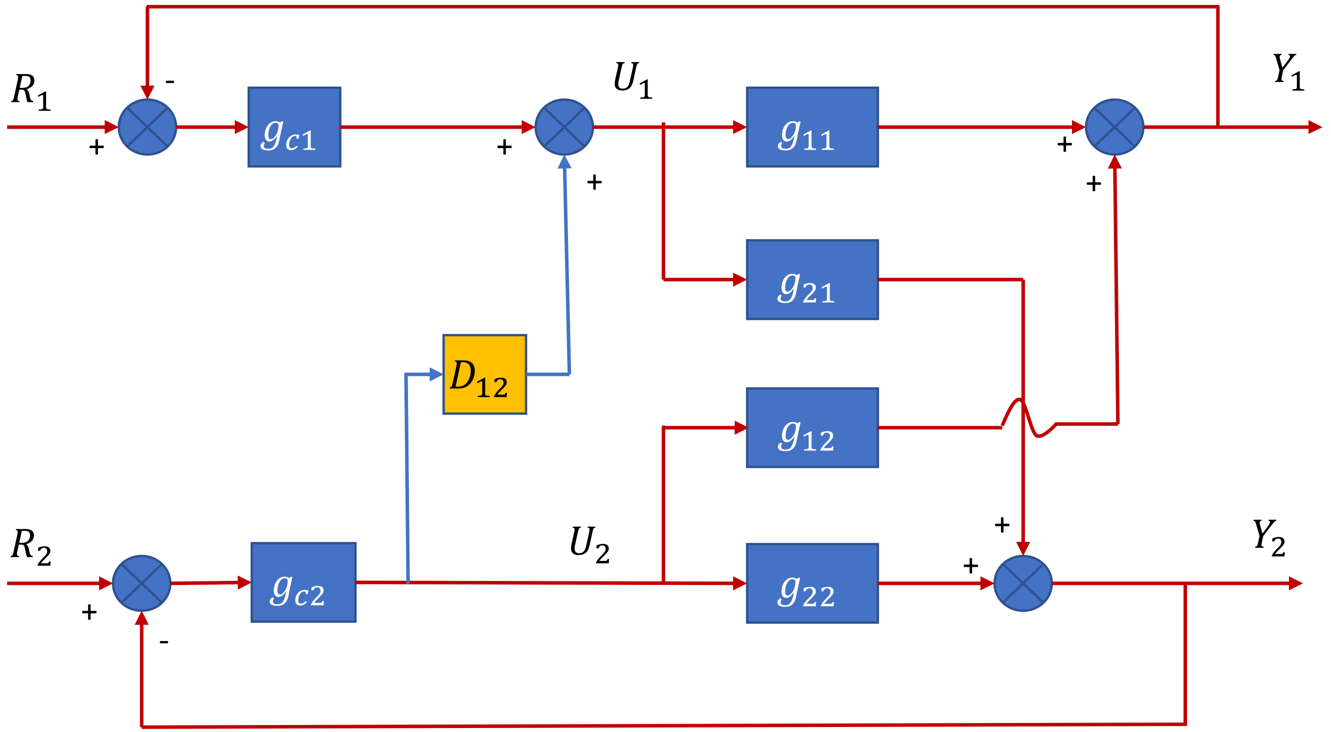
Major Challenge

- ✓ Process Interactions
- ✓ Loop 1 interacts with Loop 2
- ✓ Can lead to low control performance overall
- ✓ Adopt MPC to handle the interactions
- ✓ **Other methods ??**

Decoupling Controller

- Decoupling controller (or decoupler) can be added to the decentralized control in order to handle the process interaction
- Two types of decoupling control systems:
  1. Partial decoupling (one-way)
  2. Complete decoupling (two-way)
- A decoupler is designed for a pair of control loops which is to cancel out the interacting effect in a single direction from one loop to another – partial decoupling system
- Two decouplers can be designed for a pair of control loops so to calcel out the interactions between the two loops in both directions.

Decoupling Control: One-Way Decoupling



- ✓ To remove the coupling effect in one direction only
- ✓ D_{12} decoupling the effect of control loop 2 on control loop 1
- ✓ Decoupler improves mainly only the control loop 1
- ✓ Decoupler can be viewed as a feedforward control
- ✓ Disturbance is the coupling effect from control loop 2

Decoupler Design

- Assume **perfect cancellation** of coupling effect from loop 2 to loop 1

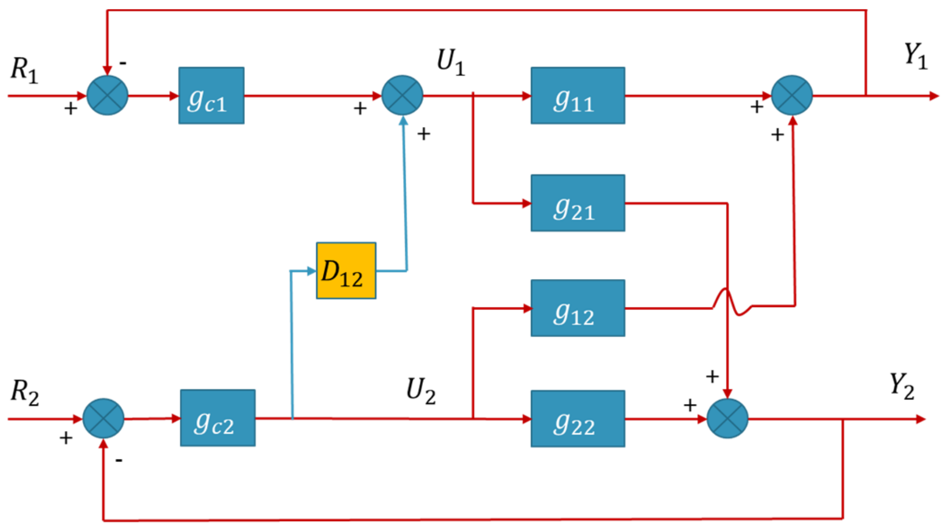
$$U_2 D_{12} g_{11} + U_2 g_{12} = 0$$

$$\therefore \ \boldsymbol{D_{12}} = -\frac{\boldsymbol{g_{12}}}{\boldsymbol{g_{11}}}$$

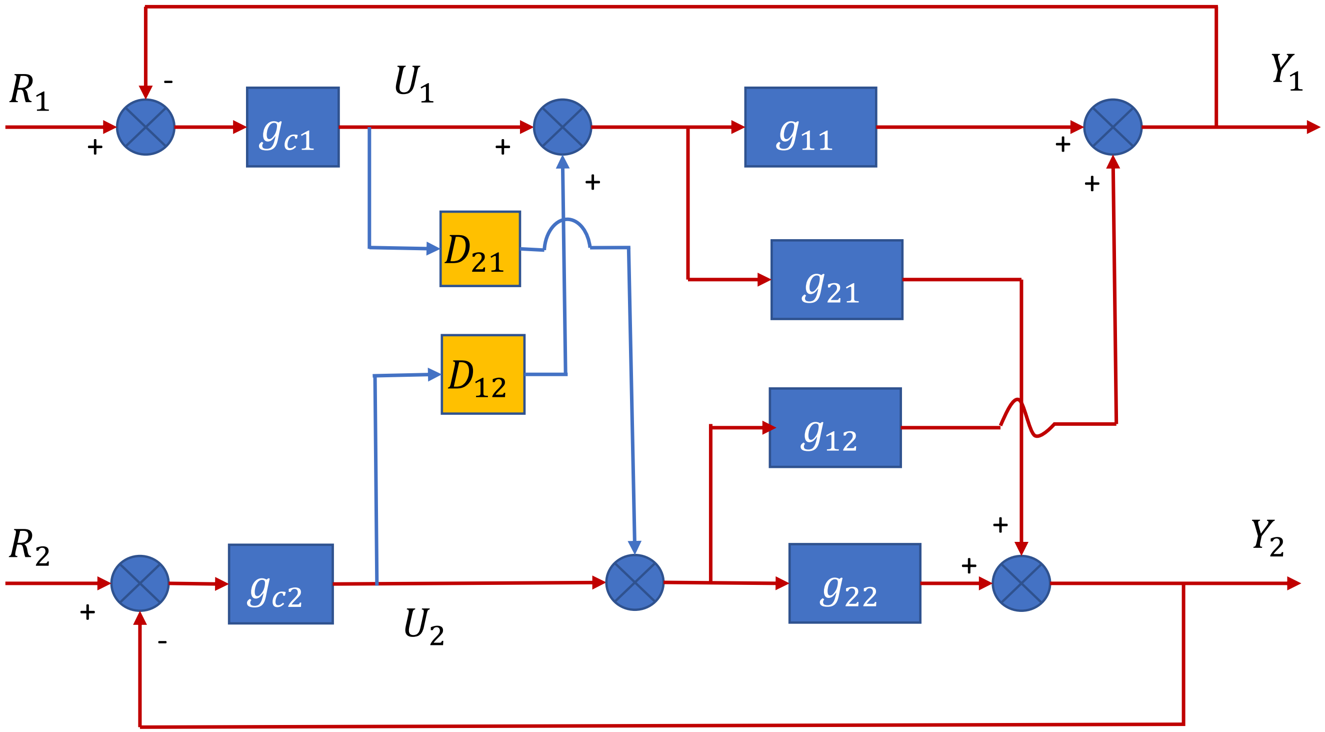
- General equation for decoupler D_{ij}

$$\therefore \ \boldsymbol{D_{ij}} = -\frac{\boldsymbol{g_{ij}}}{\boldsymbol{g_{ii}}}$$

- $\boldsymbol{D_{ij}}$ is to remove coupling effect from loop j to loop i



Two-Way Decoupling System



- ✓ D_{12} removes the coupling effect from loop 2 to loop 1
- ✓ D_{21} removes the coupling effect from loop 1 to loop 2
- ✓ This is a complete decoupling control system
- ✓ For $n \times n$ MIMO system, there will be $n(n-1)$ decouplers are required in a complete decoupling system
- ✓ Complete decoupling system may not be practical for a large MIMO
- ✓ Partial decoupling is often adopted.

Multi-loop Controllers for Distillation Column

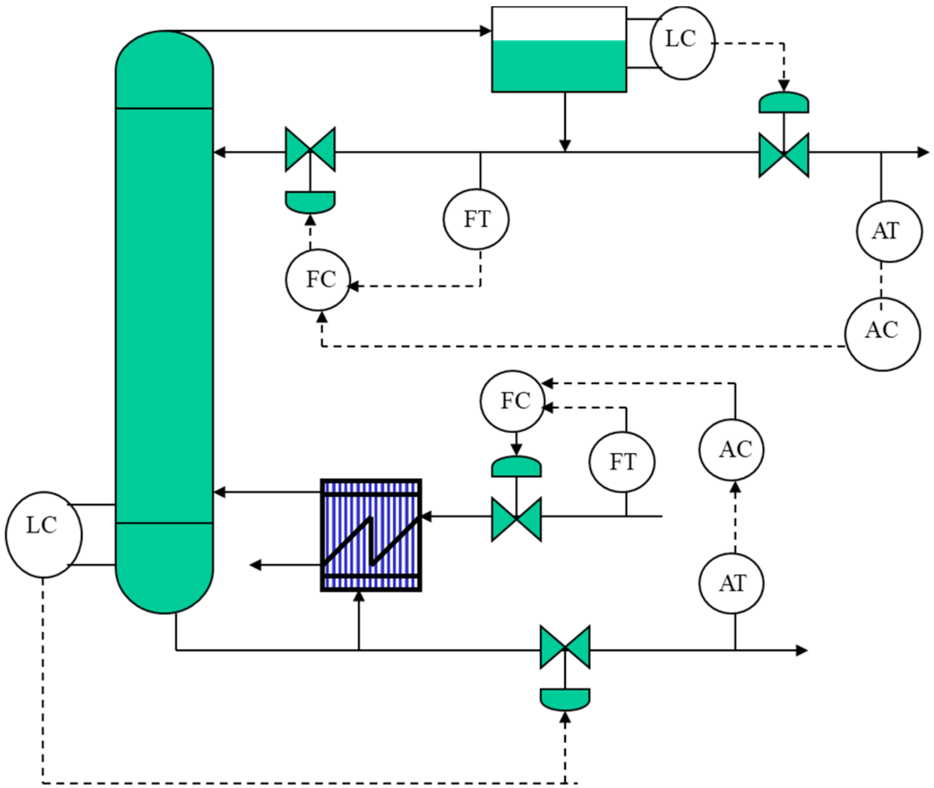
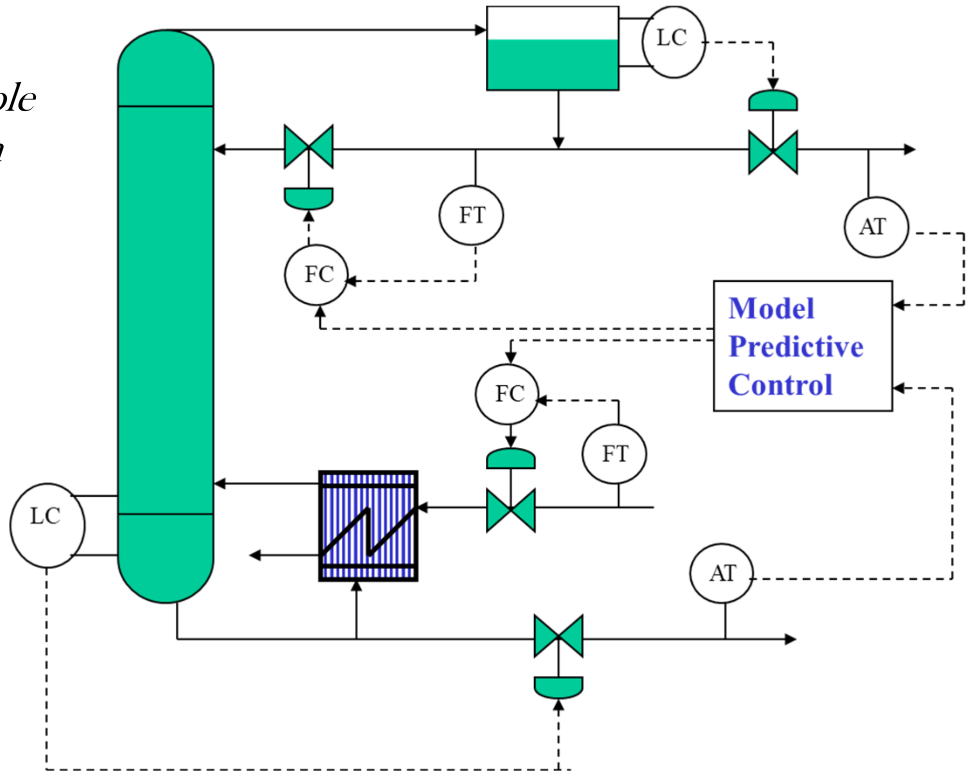


Figure 1. *Multi-loop control of distillation column*

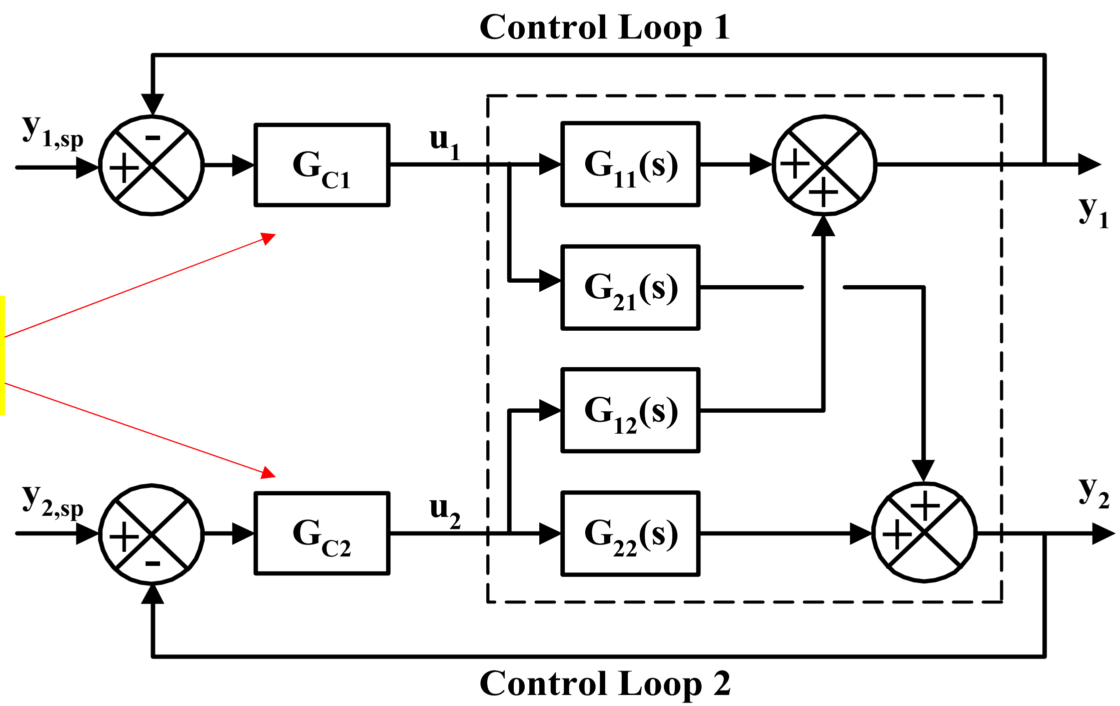
Multivariable Controller for Distillation Column

Figure 2. Multivariable control of distillation column

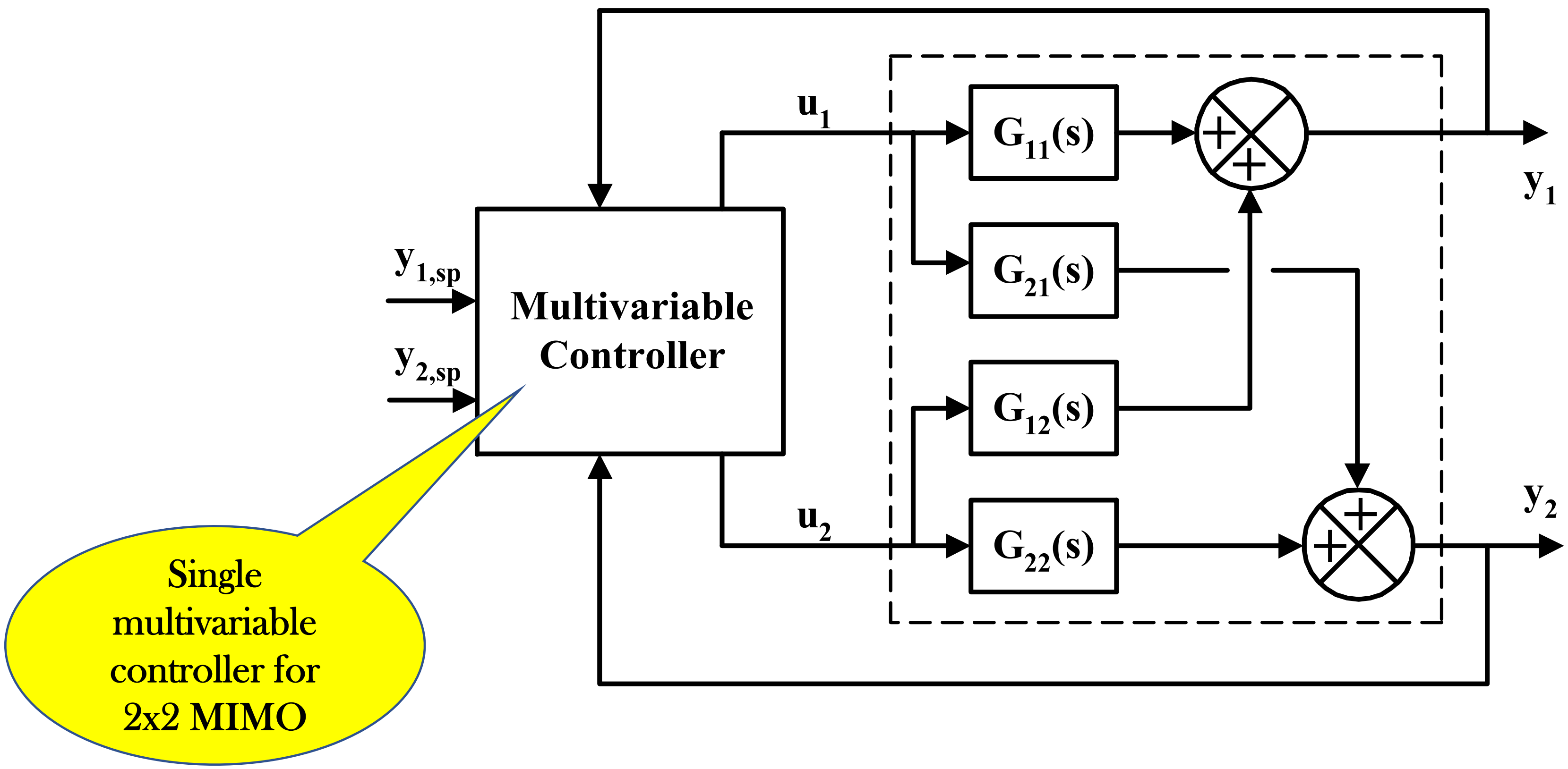


Multi-loop Control Block Diagram

SISO controllers



Multivariable Controller Block Diagram



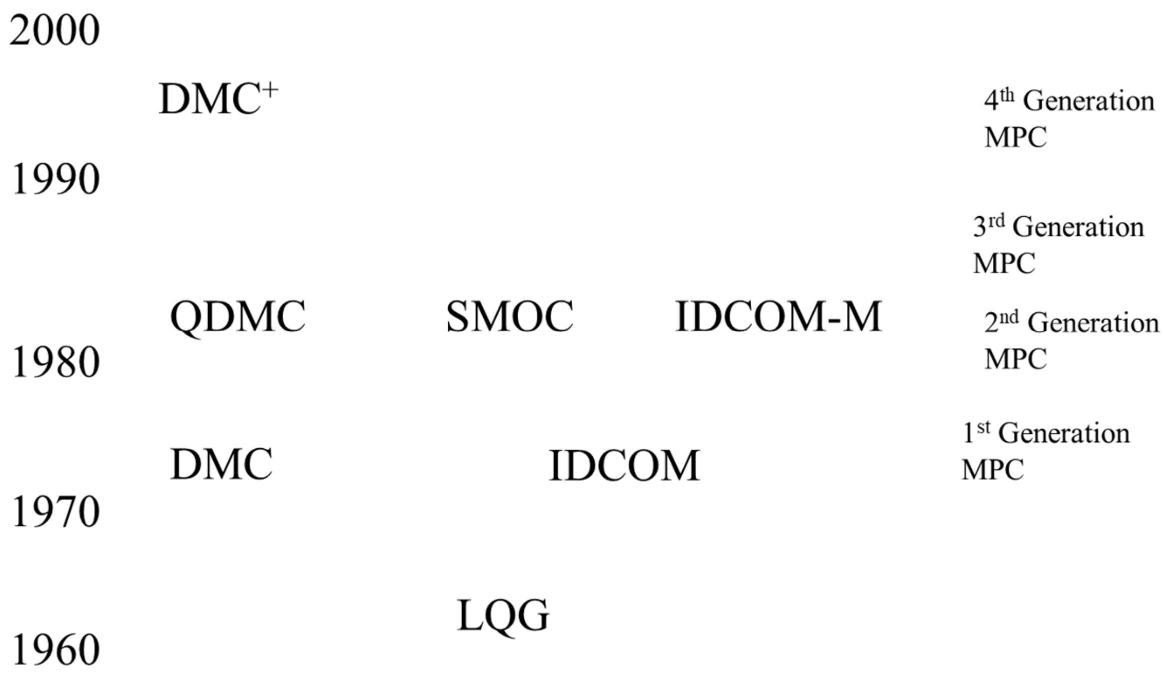
Model Predictive Control (MPC)

- Most **popular form of multivariable** control – MPC
- MPC is considered the "**gold standard**" of industrial controller used in process industry
- Can effectively handle several of soft and hard constraints.
- Conventional PID controller cannot handle a constraint.
- MPC can be optimized so that a process can be operated closed to the most profitable set of constraints without violating these constraints.
  - Critical safety issue related to the constraints can be addressed in the MPC
- Several types of industrial MPC but dynamic matrix control (DMC) is the most widely used form.

Advantages of MPC

1) Can provide **decoupling** to reduce the effect of process interactions
2) Can provide **feedforward compensation** for measured disturbances
3) Can directly **compensate** for the nonlinearity of the process if the model used is nonlinear.
4) Can **handle soft and hard constraints**

Genealogy of Linear MPC algorithms



Dynamic Matrix Control (DMC)

- First developed by engineers at Shell Oil and first applied in 1973.
- Key features of the **DMC control algorithm**:
- Linear **step response model** for the plant
- Quadratic **performance objective** over a finite prediction horizon
- Future value of controlled variable (CV) is driven to follow the setpoint as closely as possible.

Discrete-Time Step Response Model (DTSRM)

- **Dynamic Matrix Control (DMC)** requires a **dynamic model** of the given process.
- A **DTSRM** is used in DMC.
- **DTSRM** is required to calculate the **optimal control actions**.
- In conventional control algorithms, a transfer function ***Gp*** is often used to represent the effect of **MV on CV.**
- Similar information contained in ***Gp*** can also be represented using the DTSRM.

DTSRM developed from FOPDT

- Consider a FOPDT

$$\boldsymbol{G_p} = \frac{\boldsymbol{y(s)}}{\boldsymbol{u(s)}} = \frac{\mathbf{exp}(-\boldsymbol{s})}{\boldsymbol{s}+\mathbf{1}}, \qquad K_p = 1, \tau_p = 1, \theta_p = 1$$

- Apply a **unit step change** in input, $\boldsymbol{\Delta u(s) = \frac{1}{s}}$ **at** $\boldsymbol{t = t_0}$
- Choose a **fixed sampling time**, e.g., $T_s = 1$
- A generalized DTSRM can be obtained from the step test as follows

$$\boldsymbol{a_i} = \frac{\boldsymbol{y'(i)}}{\boldsymbol{\Delta u(t_0)}} \quad where\ y'(i) = y(i) - y(0) \qquad (1)$$

- A **set of coefficients** a_i for $i = 0, 1, 2, \ldots n$ can be obtained using Eqn. (1)

Example: Discrete-Time Step Response Model

t	i	Δu	$y(t)$	a_i
0	0	1	0	0
1	1	0	0	0
2	2	0	0.63	0.63
3	3	0	0.87	0.87
4	4	0	0.95	0.95
5	5	0	0.98	0.98
6	6	0	0.99	0.99
7	7	0	1.00	1.00
8	8	0	1.00	1.00

- ✓ Input step change at $t = 0$ by 1 unit
- ✓ Notice only 1 time of input change.
- ✓ Several input changes can be made at different t values
- ✓ A set of a_i for $i = 0, 1, \ldots 8$ are calculated.
- ✓ Ideally, n is chosen until the process reach a new steady state

Response of a Process to an Input Change

❑From Eqn. (1)

$$y(t_i) = y'(t_i) + y(t_0) = a_i \Delta u(t_0) \qquad (2)$$

❑Eqn. (2) can be used to predict the response $y(t)$ for several input changes, e.g., at time $t = t_0, t = t_1, t = t_2$

$$\begin{cases} y_1 - y_0 = y(t_1) - y(t_0) = a_1 \Delta u_0 = a_1 \Delta u(t_0) \\ y_2 - y_0 = a_2 \Delta u_0 + a_1 \Delta u_1 \\ y_3 - y_0 = a_3 \Delta u_0 + a_2 \Delta u_1 + a_1 \Delta u_3 \end{cases} \qquad (3)$$

❑In general, Eqn. (3) can be written as follows

$$y_n - y_0 = \sum_{i=1}^{n} a_i \Delta u(t_{n-i}) \qquad (4)$$

Response to Input Changes in Matrix Form

Matrix A

$$
\begin{bmatrix} \mathbf{y}'(\mathbf{t}_1) \\ \mathbf{y}'(\mathbf{t}_2) \\ \mathbf{y}'(\mathbf{t}_3) \\ \mathbf{y}'(\mathbf{t}_4) \\ . \\ . \\ \mathbf{y}'(\mathbf{t}_n) \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1 & 0 & 0 & 0 & \dots \\ \mathbf{a}_2 & \mathbf{a}_1 & 0 & 0 & \dots \\ \mathbf{a}_3 & \mathbf{a}_2 & \mathbf{a}_1 & 0 & \dots \\ \mathbf{a}_4 & \mathbf{a}_3 & \mathbf{a}_2 & \mathbf{a}_1 & \dots \\ . \\ . \\ .\mathbf{a}_m & \mathbf{a}_m & \mathbf{a}_m & \dots & \mathbf{a}_1 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u}(\mathbf{t}_0) \\ \Delta \mathbf{u}(\mathbf{t}_1) \\ \Delta \mathbf{u}(\mathbf{t}_2) \\ \Delta \mathbf{u}(\mathbf{t}_3) \\ \Delta \mathbf{u}(\mathbf{t}_4) \\ . \\ .\Delta \mathbf{u}(\mathbf{t}_{n-1}) \end{bmatrix}
$$

$$\mathbf{y}' = \mathbf{A}\,\Delta\mathbf{u}$$

- ✓ n is the prediction horizon
- ✓ m is the model horizon
- ✓ $n > m$
- ✓ Typically $n = 1.5m$
- ✓ Duration for a process to settle after an input step change is
  $$T_{stt} = mT_s$$
- ✓ Duration a process to settle if subjected to a series of input changes will be longer than T_{stt}

Previous example:
Use equation (4) to calculate $y(t_7)$

$$\boldsymbol{y_n - y_0 = \sum_{i=1}^{n} a_i \Delta u(t_{n-i})} \qquad (4)$$

- Initial value $y(0) = 1,\ u(0) = 1$
- 7 step changes in input u
- $n = 7$

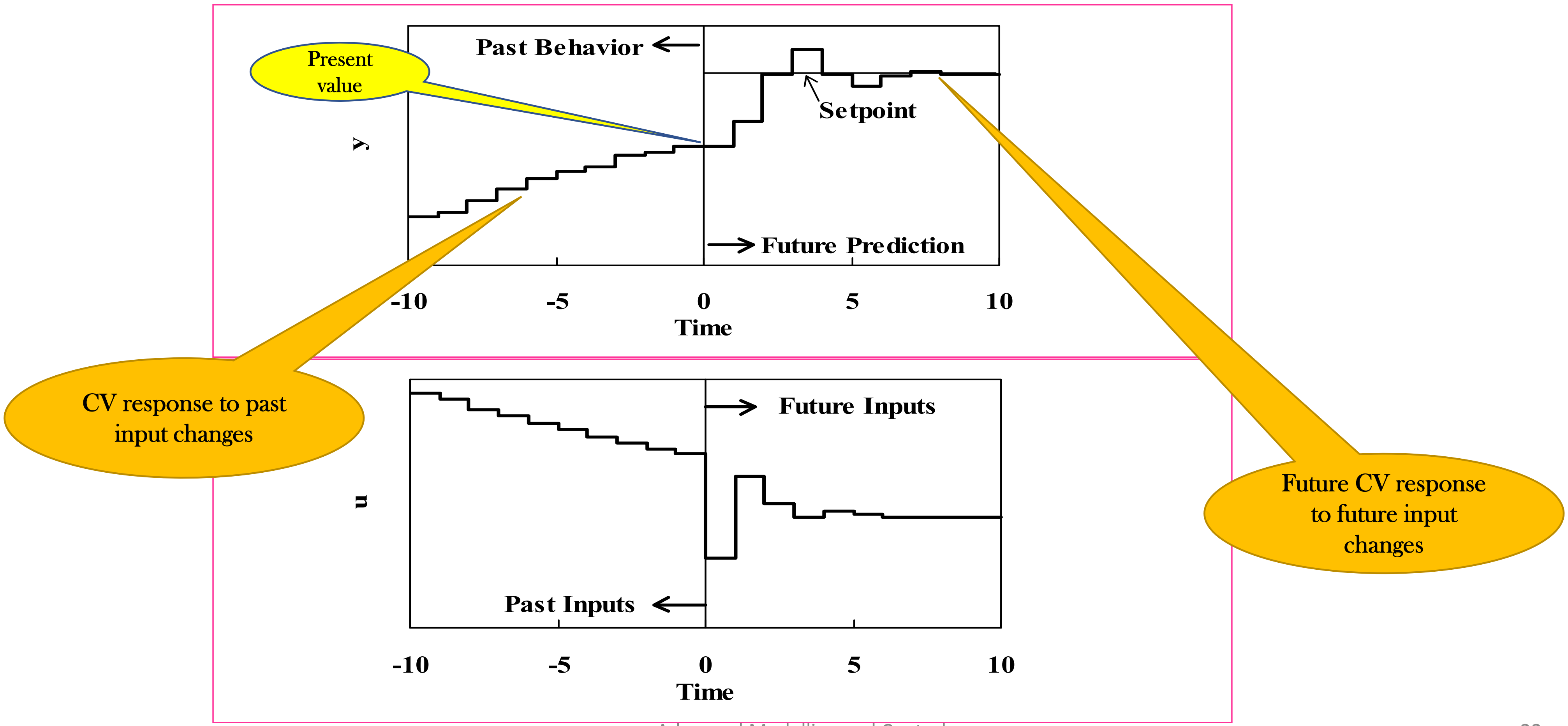
$$y_7 = y_0 + a_1\Delta u_6 + a_2\Delta u_5 + a_3\Delta u_4 + a_4\Delta u_3 + a_5\Delta u_2 + a_6\Delta u_1 + a_7\Delta u_0$$

$$\therefore\ \boldsymbol{y_7} = 1 + 0(-1) + 0.63(-1) + 0.87(0) + 0.95(-1) + 0.98(1) + 0.99(1) + 1.0(0) = \mathbf{1.39}$$

t	**i**	**Δu**	***y(t)***	$\boldsymbol{a_i}$
0	0	1	0	0
1	1	0	0	0
2	2	0	0.63	0.63
3	3	0	0.87	0.87
4	4	0	0.95	0.95
5	5	0	0.98	0.98
6	6	0	0.99	0.99
7	7	0	1.00	1.00
8	8	0	1.00	1.00

Time	Sample Number I	u(ti)	Δu(ti)
0	0	1	0
1	1	2	1
2	2	3	1
3	3	2	-1
4	4	2	0
5	5	1	-1
6	6	0	-1
7	7	1	1
8	8	2	1

Moving Horizon Algorithm



Moving Horizon Algorithms Concept

- **Choose future MV values** to regulate the CV to its **setpoint** using DTSRM and previous inputs.
- After one **control** interval has **expired**, **new (present value) of CV last change of MV value $\Delta u(t)$** are available.
- The **controller recalculates** the **sequence of MV values** into the future to meet the **control objective**.
- Then, only **the first move** is **actually implemented** before a new sequence of input values are recalculated.

Prediction Vector $\boldsymbol{y}^P$

- So far, we have assumed that $\boldsymbol{y}(\boldsymbol{t_0})$ **is at steady state** and the MV changes are made only for $t > t_0$.
- For control application, this is **not a realistic assumption** since MV changes for $t < t_0$ are likely to exist.
- Effect of the previous $\boldsymbol{\Delta u(t)}$ **for** $\boldsymbol{t < t_0}$ **must be taken into account** to properly model the future behavior (response) of the CV, i.e., $y(t)$ for $t > t_0$.
- **Prediction vector** $\boldsymbol{y}^P$ contains the effect of **previous MV changes** on CV for $t > t_0$, if **no future MV change** is made, i.e., $\Delta u(t) = 0$ for $t > t_0$.

Prediction Vector y^P

- ❑Assume the model horizon, $\boldsymbol{m}$ time steps, an input change has its total **steady-state** effect on the process.
- ❑Applying Eqn. (4) to calculate y^P at $t = t_1$: $\Rightarrow \boldsymbol{y_n} - \boldsymbol{y_0} = \sum_{i=1}^{n}[\boldsymbol{a_i}\Delta\boldsymbol{u}(\boldsymbol{t_{n-i}})]$

$$\boldsymbol{y^P(t_1) = y(t_{-m}) + a_m\Delta u(t_{-m}) + a_m\Delta u(t_{-m+1}) + a_{m-1}\Delta u(t_{-m+2}) + \cdots a_3\Delta u(t_{-2}) + a_2\Delta u(t_{-1})} \quad \textbf{(5)}$$

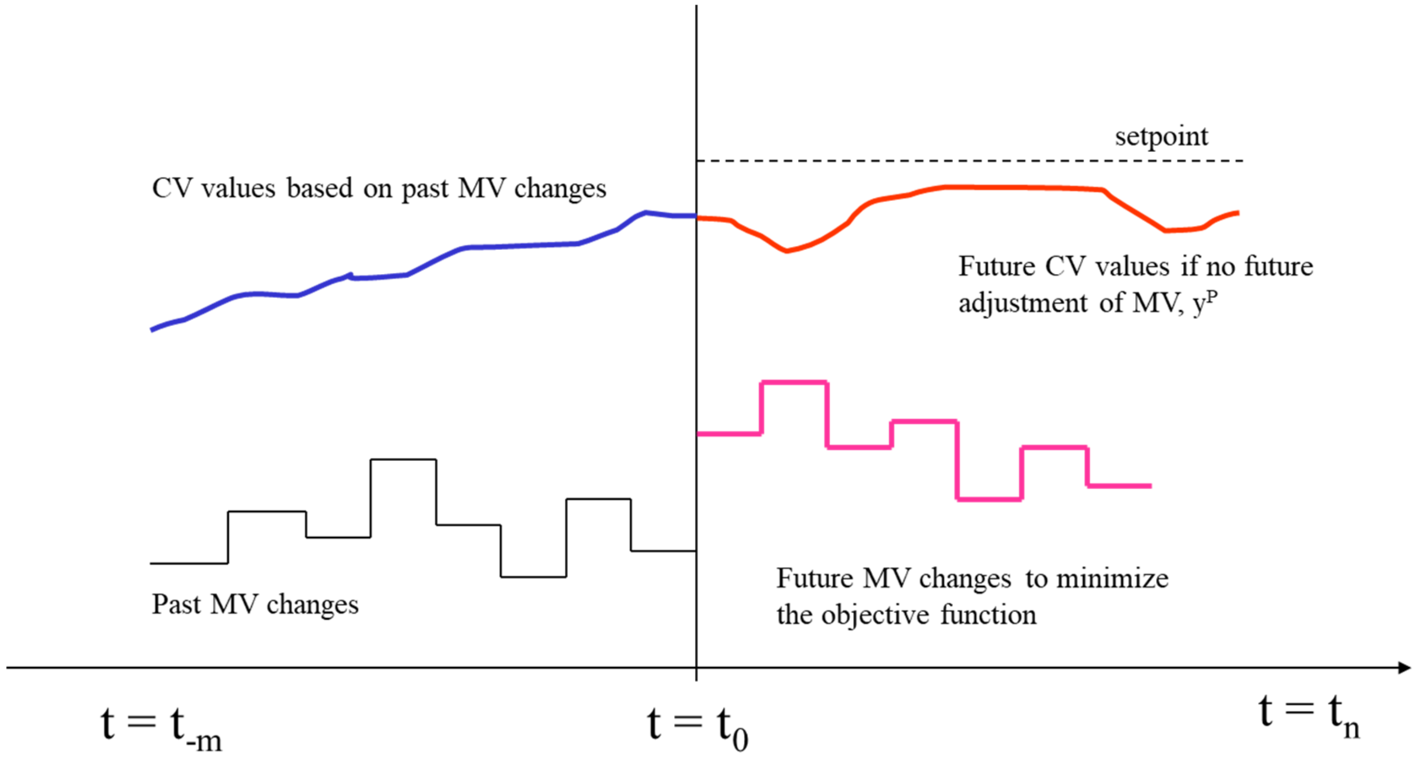
Note:

- $a_{-m} = a_m, a_{-m+1} = a_{m+1} \dots .$
- $a_{m+j} = a_m$ for $j = 1, 2, 3, \dots$ $\text{when total steady} - \text{state is achived with m steps}$
- Negative subscripts indicate the number of sampling intervals before t_0 and assuming the process is at steady state at $t = t_{-m}$

$\boldsymbol{y^P(t_1) = y_{-m} + \sum_{i=-m-1}^{n=-1}[a_i\Delta u(t_{m-i})]}$

$$\mathbf{y^P(t_n) = y(t_{-m}) + a_m\Delta u(t_{-m}) + a_m\Delta u(t_{-m+1}) + \dots.. + a_m\Delta u(t_{-2}) + a_m\Delta t(_{-1})}$$

Prediction Vector y^P



Matrix Form of y^P

Prediction matrix A^P

$$\begin{bmatrix} y^P(t_1) \\ y^P(t_2) \\ \cdot \\ \cdot \\ y^P(t_n) \end{bmatrix} = \begin{bmatrix} y(t_{-m}) \\ y(t_{-m}) \\ \cdot \\ \cdot \\ y(t_{-m}) \end{bmatrix} + \begin{bmatrix} a_m & a_m & a_{m-1} & a_{m-2} & \dots & a_3 & a_2 \\ a_m & a_m & a_m & a_{m-1} & \dots & a_4 & a_3 \\ \cdot & & & & & & \\ \cdot & & & & & & \\ a_m & a_m & a_m & a_m & \dots & a_m & a_m \end{bmatrix} \begin{bmatrix} \Delta u(t_{-m}) \\ \Delta u(t_{-m+1}) \\ \cdot \\ \cdot \\ \Delta u(t_{-1}) \end{bmatrix}$$

$$\mathbf{y}^P = \mathbf{y}(t_{-m}) + \mathbf{A}^P \Delta \mathbf{u}^P$$

- n denotes the number of time steps movement into the future that are model with n > m
- $y(t_{-k})$ denotes the value of CV at $t = t_0 - kT_s$ where k is the number of steps from the present to the historical past

Prediction values of $y(t)$ for $t > t_0$

❑Combining the effects of past input movements and future input movements on the future CV values:

$$\tilde{\boldsymbol{y}} = \boldsymbol{y}^{\boldsymbol{p}} + \boldsymbol{A}\Delta\boldsymbol{u} \qquad (6)$$

$$\therefore\ \tilde{\boldsymbol{y}} = \boldsymbol{y}(\boldsymbol{t}_{-\boldsymbol{m}}) + \boldsymbol{A}^{\boldsymbol{P}}\Delta\boldsymbol{u}^{\boldsymbol{P}} + \boldsymbol{A}\Delta\boldsymbol{u} \qquad (7)$$

❑**Accuracy** of the equation above depends on

- **Errors** in identifying the coefficients of DTSRM
- **Unmeasured** disturbances
- **Nonlinear** process behavior
- Non **steady-state behavior at $t = t_{-m}$**

Reducing Effects of Process/Model Mismatch

- To ensure the **reliability of MPC**, it is important that the **deviation** of the model from actual process is **small** – large process/model mismatch can lead to poor MPC performance
- Error between the measured value of $y(t_0)$ and the predicted value $y^P(t_0)$ can be used to adjust eqn. (6) to make it more accurate.
- Recall eqn. (6): $\tilde{\boldsymbol{y}} = \boldsymbol{y^p} + \boldsymbol{A}\Delta\boldsymbol{u}$
- Prediction error: $\boldsymbol{\varepsilon} = \boldsymbol{y^P}(\boldsymbol{t_0}) - \boldsymbol{y}(\boldsymbol{t_0})$
- Eqn. (6) is modified to: $\boldsymbol{y} = \boldsymbol{y^P} + \boldsymbol{A}\Delta\boldsymbol{u} + \boldsymbol{\phi^T}$
- Vector of error: $\boldsymbol{\phi} = [\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}, \ldots, \boldsymbol{\varepsilon}]$
- Number of rows of $\boldsymbol{\phi^T}$ is same as that of $\boldsymbol{y}$

DMC Control Law

- DMC control law is based on **minimizing the error** from setpoint E
- The **objective function, Φ** is the sum of the square of the errors from setpoint for the prediction horizon, n.

$$\Phi = \sum_{i=1}^{n} \left[y_{sp} - y(t_i) \right]^2$$

$$E(t_i) = y_{sp} - y^P(t_i) - \varepsilon$$

$$\therefore \quad \Phi = \sum \left[E(t_i) - y^C(t_i) \right]^2$$

$y^C(t_i)$: $A\Delta u(t)$

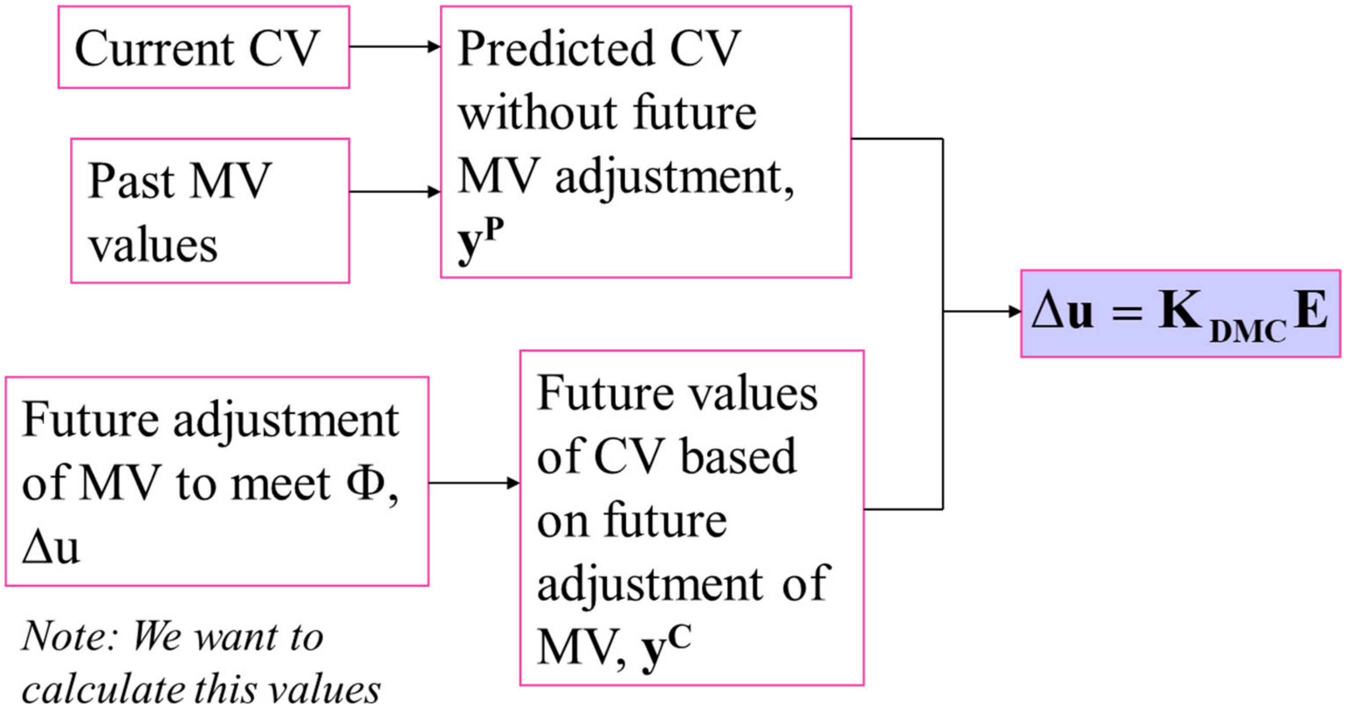
$$\frac{\partial \Phi}{\partial \Delta u} = A^T (A\Delta u - E) = 0$$

$$\Rightarrow \Delta u = K_{DMC} E$$

$$K_{DMC} = (A^T A)^{-1} A^T$$

$$\Delta u = K_{DMC} E$$

DMC computation



Move Suppression Factor Q

- Very aggressive control because based on **minimizing deviation from setpoint** without regard to the changes in the MV levels.
- Hence, **sharp changes in MV result**, which is **undesirable**.
- This problem is overcome by adding a diagonal matrix $\boldsymbol{Q^2}$ **to** $\boldsymbol{A^T A}$
- The larger the value of q, the more Δu is **penalized** for changes in MV.

$$\Delta \mathbf{u} = (\mathbf{A}^T \mathbf{A} + \mathbf{Q}^2)^{-1} \mathbf{A}^T \mathbf{E}$$

$$\mathbf{Q} = \begin{bmatrix} \mathbf{q} & 0 & 0 \ldots 0 \\ 0 & \mathbf{q} & 0 \ldots 0 \\ 0 & 0 & \mathbf{q} \ldots 0 \\ . & & \\ . & & \\ 0 & 0 & 0 \ldots \mathbf{q} \end{bmatrix}$$

Summary

- ❑ Decoupler can reduce control-loop interactions: can improve control performance overall
- ❑ Two control architectures used in industry – (1) decentralized, and (2) multivariable or centralized control
- ❑ Model Predictive Control (MPC) is the most widely used multivariable controller in process industry
- ❑ MPC has several different versions – one of the most common is DMC
- ❑ MPC requires a model of the process and an optimizer
- ❑ Advantages of MPC:
  - (1) able to handle constraints,
  - (2) feedforward capability for measured disturbances,
  - (3) able to handle process interactions, and
  - (4) can cope with nonlinearity if nonlinear model is used in the controller.
- ❑ Optimal future control actions can be calculated as $\boldsymbol{\Delta u} = \boldsymbol{K_{DMC}E}$