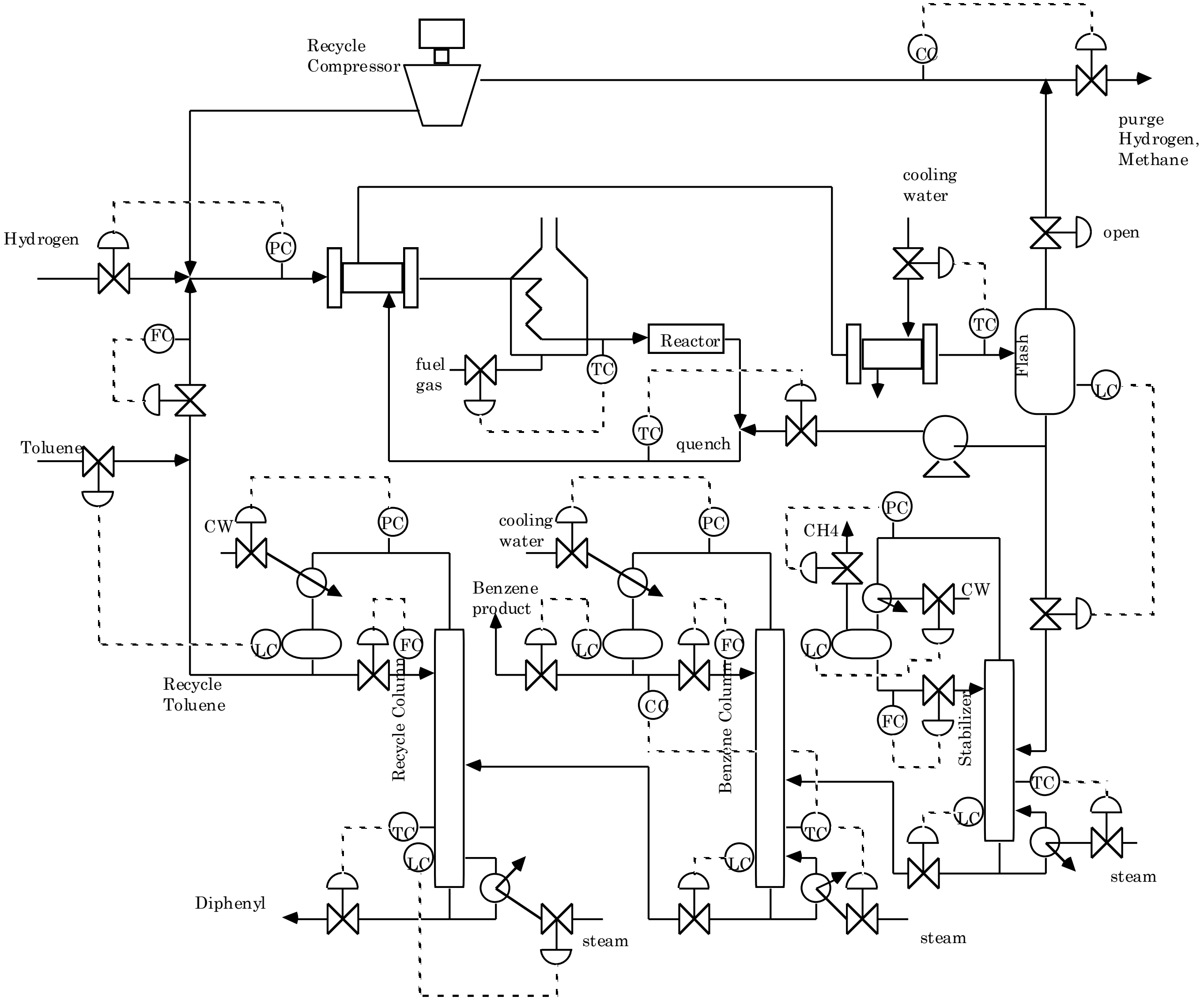
MPC Introduction

- Overview
  - Basic Concept of MPC
  - History
- Optimization Formulation
  - Models
  - Analytical Solution to Unconstrained Problem
- Summary
  - Limitations & a Look Ahead

B. Wayne Bequette

Rensselaer
Chemical and Biological Engineering

Motivation: Complex Processes

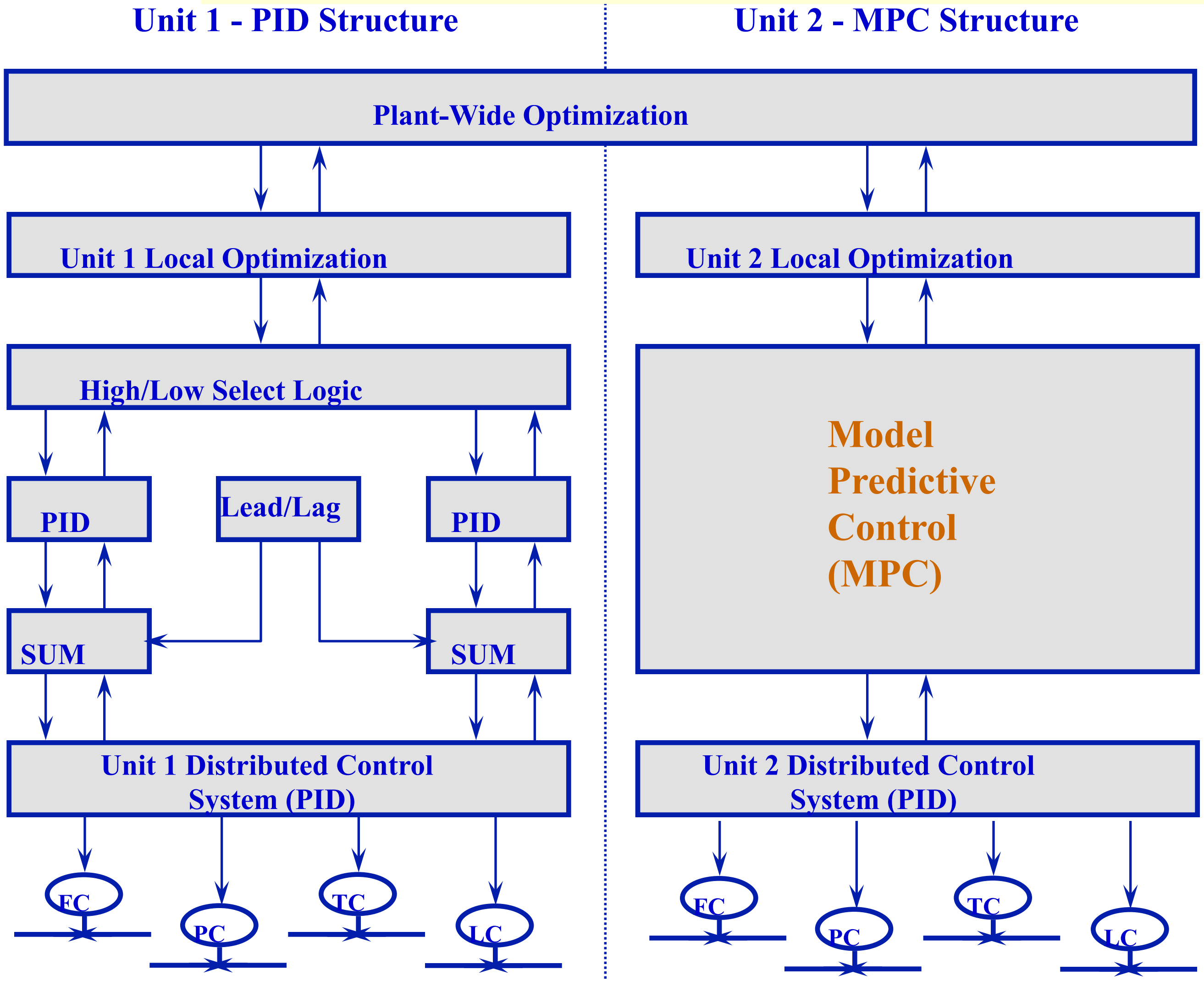


B. Wayne Bequette

Important Issues in Petroleum Refining

- Multivariable, Large Scale
  - Challenge to tune individual SISO controllers
- Operation at Constraints
  - Anti-reset windup and other strategies for PID
- Economic Payout for Advanced Control
  - Economic return justifies capital and on-going maintenance costs
- Model Predictive Control
  - Evolved independently in the US and France refining industry

How is MPC used?



Unit 1 - PID Structure

Unit 2 - MPC Structure

Plant-Wide Optimization

Unit 1 Local Optimization

High/Low Select Logic

PID

Lead/Lag

PID

SUM

SUM

Unit 1 Distributed Control System (PID)

FC

PC

TC

LC

Unit 2 Local Optimization

Model Predictive Control (MPC)

Unit 2 Distributed Control System (PID)

FC

PC

TC

LC

Global Steady-State Optimization (every day)

Local Steady-State Optimization (every hour)

Dynamic Constraint Control (every minute)

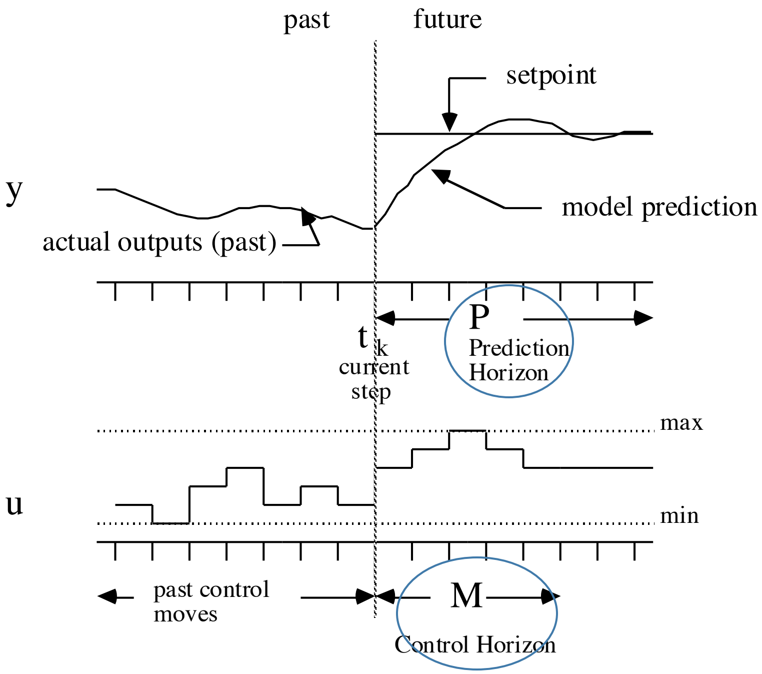
Supervisory Dynamic Control (every minute)

Basic Dynamic Control (every second)

Bequette

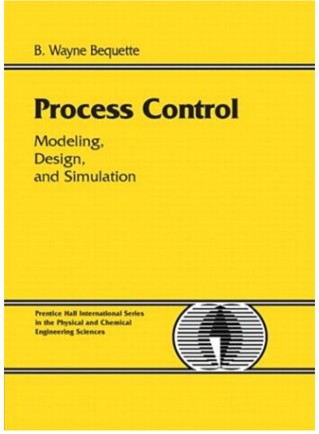
From Tom Badgwell, 2003 Spring AIChE Meeting, New Orleans

Model Predictive Control (MPC)



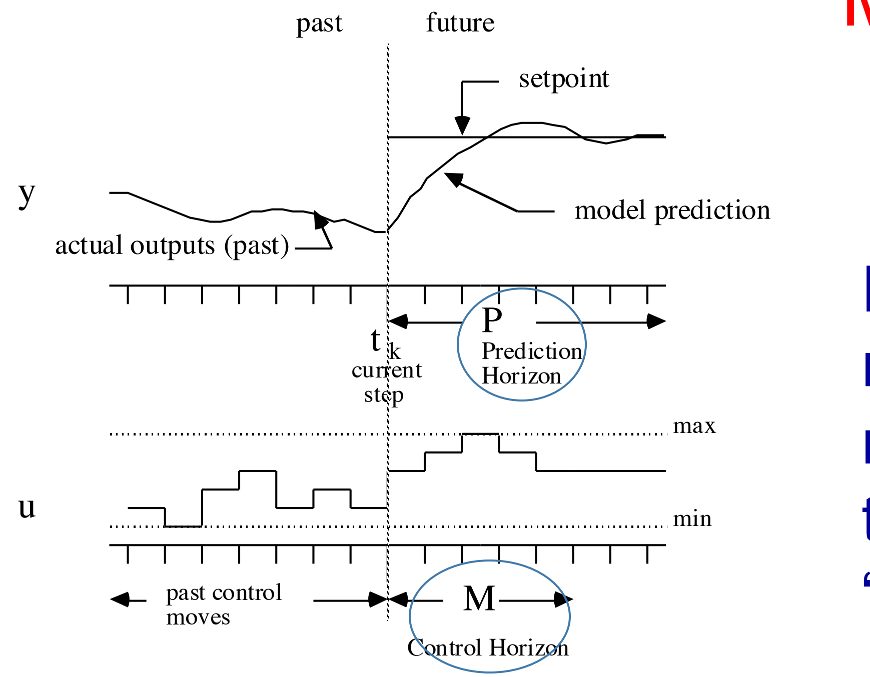
Find current and future manipulated inputs that best meet a desired future output trajectory. Implement first “control move”.

- Type of model for predictions?
- Information needed at step k for predictions?
- Objective function and optimization technique?
- Correction for model error?

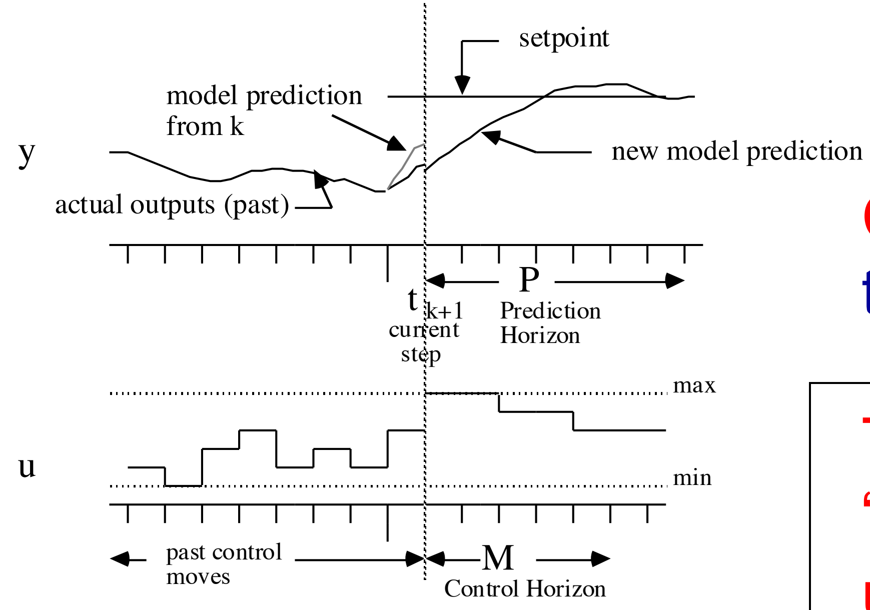


Chapter 16

Model Predictive Control (MPC)



Find current and future manipulated inputs that best meet a desired future output trajectory. Implement first “control move”.



At next sample time:

Correct for model mismatch, then perform new optimization.

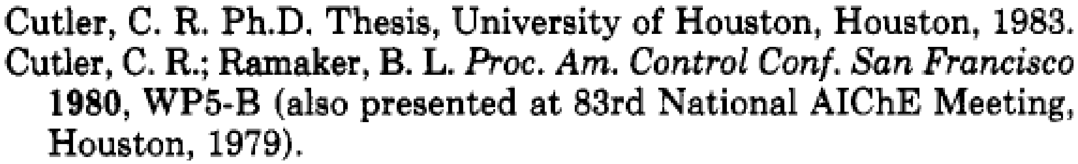
This is a major issue – “disturbances” vs. model uncertainty

B. Wayne Bequette

MPC History

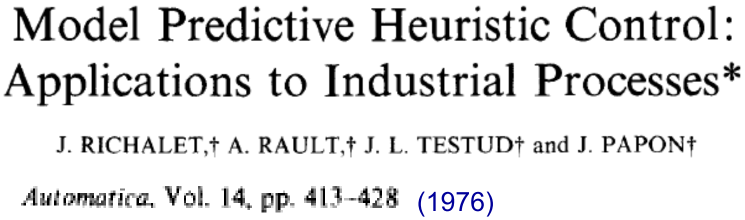
- Intuitive
  - Basically arose in two different "camps"
- Dynamic Matrix Control (DMC)
  - 1960's and 1970's – Shell Oil - US

Charlie Cutler



Cutler, C. R. Ph.D. Thesis, University of Houston, Houston, 1983.
Cutler, C. R.; Ramaker, B. L. *Proc. Am. Control Conf. San Francisco* **1980**, WP5-B (also presented at 83rd National AIChE Meeting, Houston, 1979).

- Related to techniques developed in France (IDCOM)
- Large-scale MIMO
- Formulation for constraints important

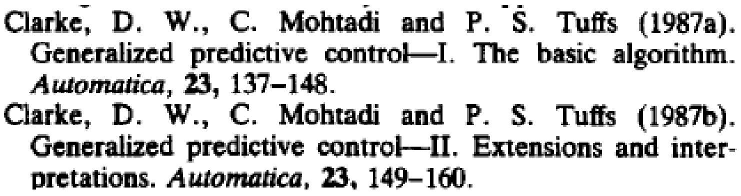


Model Predictive Heuristic Control: Applications to Industrial Processes*

J. RICHALET,† A. RAULT,† J. L. TESTUD† and J. PAPON†

Automatica, Vol. 14, pp. 413–428 (1976)

- Generalized Predictive Control (GPC)
  - Evolved from adaptive control
  - Focus on SISO, awkward for MIMO



Clarke, D. W., C. Mohtadi and P. S. Tuffs (1987a). Generalized predictive control—I. The basic algorithm. *Automatica*, **23**, 137–148.
Clarke, D. W., C. Mohtadi and P. S. Tuffs (1987b). Generalized predictive control—II. Extensions and interpretations. *Automatica*, **23**, 149–160.

B. Wayne Bequette

Objective Functions

Quadratic Objective Function, Prediction Horizon (P) = 3,
Control Horizon (M) = 2

$$J = \underbrace{\left(r_{k+1} - \hat{y}_{k+1}\right)^2 + \left(r_{k+2} - \hat{y}_{k+2}\right)^2 + \left(r_{k+3} - \hat{y}_{k+3}\right)^2}_{\text{3 steps into future}}$$

$$+ \underbrace{w\Delta u_k^2 + w\Delta u_{k+1}^2}_{\text{2 control moves}}$$

Weight

General Representation of a Quadratic Objective Function

$$J = \sum_{i=1}^{P} \left(r_{k+i} - \hat{y}_{k+i}\right)^2 + w \sum_{i=0}^{M-1} \Delta u_{k+i}^2$$

With linear models, results in analytical solution (w/o constraints)

Alternative Objective Functions

Penalize u rather than Δu

$$J = \sum_{i=1}^{P} \left(r_{k+i} - \hat{y}_{k+i}\right)^2 + w \sum_{i=0}^{M-1} u_{k+i}^2$$

Will usually result in "offset"

Sum of absolute values (results in LP)

$$J = \sum_{i=1}^{P} \left|r_{k+i} - \hat{y}_{k+i}\right| + w \sum_{i=0}^{M-1} \left|\Delta u_{k+i}\right|$$

Existing LP methods are efficient, but solutions hop from one constraint to another

Models

- State Space
- ARX (auto-regressive, exogenous input)
- Step Response
- Impulse (Pulse) Response

- Nonlinear, Fundamental (First-Principles)
- ANN (Artificial Neural Networks)
- Hammerstein (static NL with linear dynamics)
- Volterra
- Multiple Model

Discrete Linear Models used in MPC

$$x_{k+1} = \Phi x_k + \Gamma u_k$$

$$y_k = C x_k$$

State Space

Some texts/papers have different sign conventions

$$y_k = -a_1 y_{k-1} - a_2 y_{k-2} - \cdots - a_n y_{k-n} +$$

$$\enclose{circle}{b_0} u_k + b_1 u_{k-1} + b_2 u_{k-2} + \cdots + b_m u_{k-m}$$

Input-Output
(ARX)

usually $b_0 = 0$

$$y_k = \sum_{i=1}^{\infty} s_i \Delta u_{k-i}$$

$$= s_1 \Delta u_{k-1} + \cdots + s_N \Delta u_{k-N} + s_{N+1} \Delta u_{k-N-1} + \cdots + s_{N+\infty} \Delta u_{k-\infty}$$

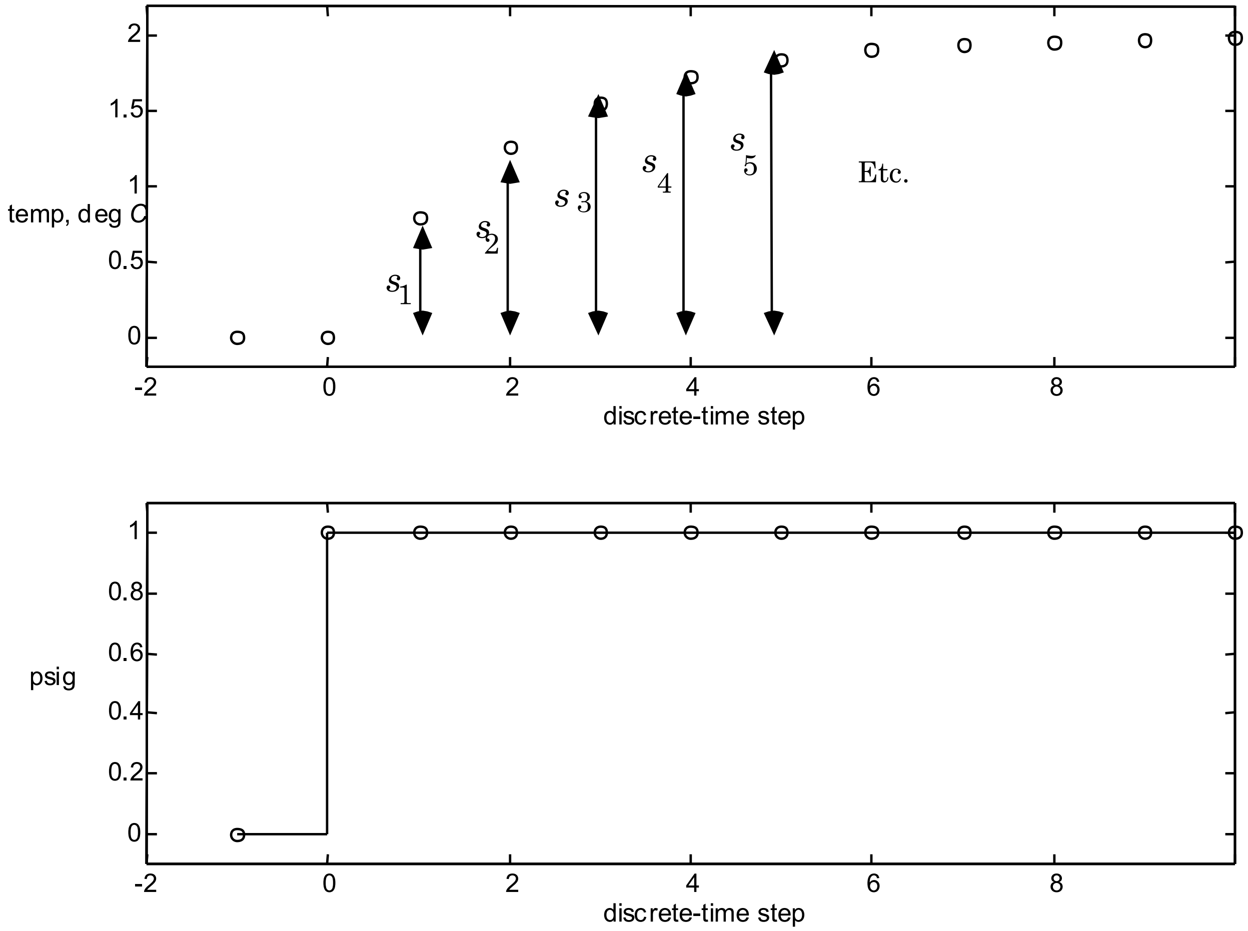
Step Response

$$y_k = \sum_{i=1}^{\infty} h_i u_{k-i}$$

$$= h_1 u_{k-1} + \cdots + h_N u_{k-N} + h_{N+1} u_{k-N-1} + \cdots + h_{N+\infty} u_{k-\infty}$$

Impulse Response

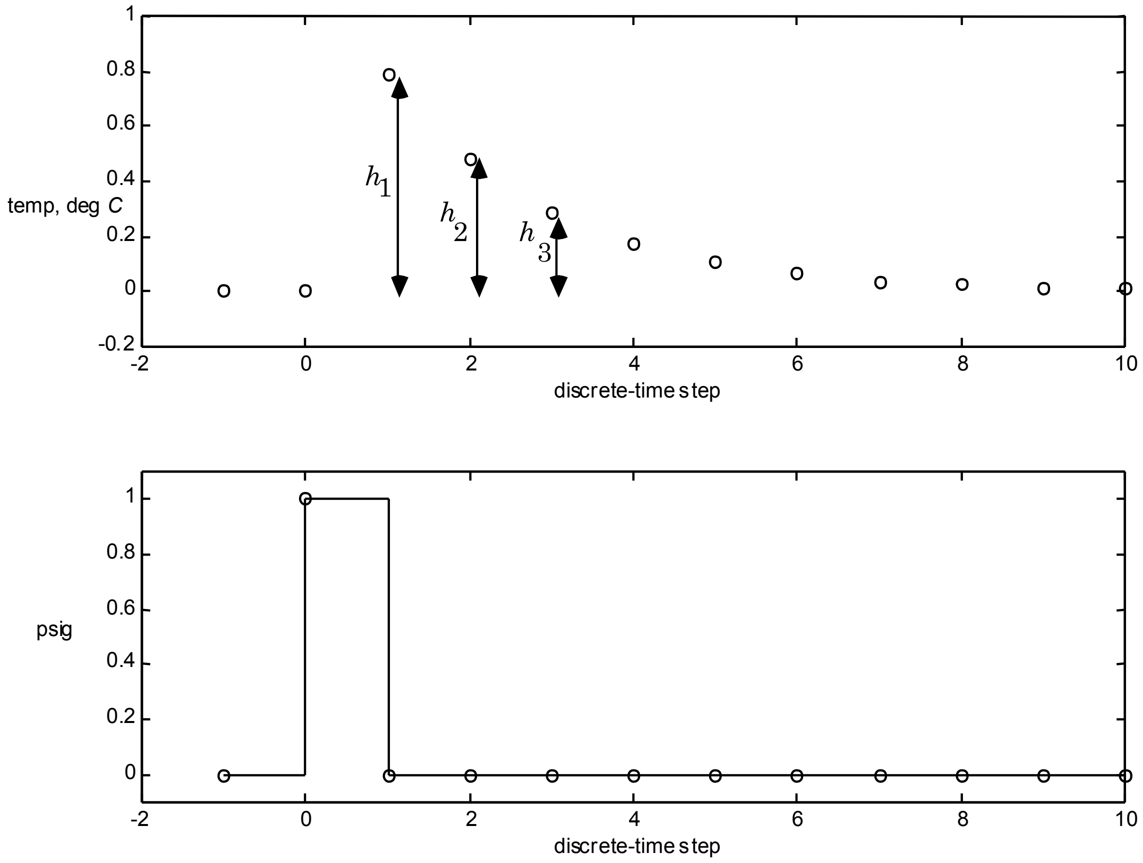
Example Step Response Model



Used in DMC

$$S = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & \mathrm{L} & s_N \end{bmatrix}^T$$

Example Impulse Response Model



Used in IDCOM

Impulse and step response coefficients are related

$$h_i = s_i - s_{i-1}$$

$$s_i = \sum_{j=1}^{i} h_j$$

Step & Impulse Models from State Space Models

$$x_{k+1} = \Phi x_k + \Gamma u_k$$
$$y_k = C x_k$$

$$H_i = C\Phi^{i-1}\Gamma$$

$$S_k = \sum_{i=1}^{k} C\Phi^{i-1}\Gamma = \sum_{i=1}^{k} H_i$$

MPC based on State Space Models

$$x_{k+1} = \Phi x_k + \Gamma u_k$$

$$y_k = C x_k$$

with known current state, easy to propagate estimates

$$x_{k+1} = \Phi x_k + \Gamma u_k$$

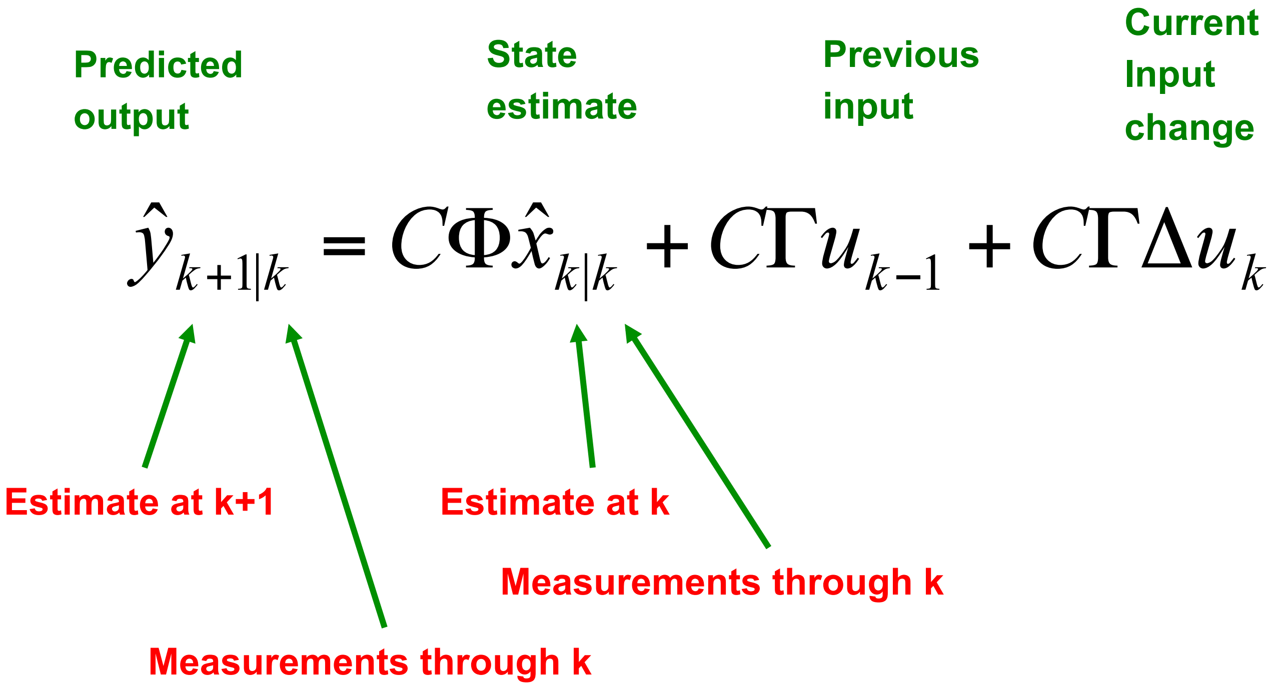
$$y_{k+1} = C x_{k+1} = C\Phi x_k + C\Gamma u_k$$

and, using control changes

$$u_k = u_{k-1} + \Delta u_k$$

$$y_{k+1} = C\Phi x_k + C\Gamma u_{k-1} + C\Gamma\Delta u_k$$

Use ^ notation for model states



Predicted output

State estimate

Previous input

Current Input change

$$\hat{y}_{k+1|k} = C\Phi\hat{x}_{k|k} + C\Gamma u_{k-1} + C\Gamma\Delta u_k$$

Estimate at k+1

Estimate at k

Measurements through k

Measurements through k

Now, propagate the prediction for P steps into the future

Output Predictions

$$\underbrace{\begin{bmatrix}\hat{y}_{k+1|k}\\ \hat{y}_{k+2|k}\\ \vdots \\ \hat{y}_{k+P|k}\end{bmatrix}}_{\hat{Y}} = \underbrace{\begin{bmatrix} C\Phi \\ C\Phi^{2} \\ \vdots \\ C\Phi^{P}\end{bmatrix}\hat{x}_{k|k} + \begin{bmatrix} C\Gamma \\ C\Phi\Gamma + C\Gamma \\ \vdots \\ \sum_{i=1}^{P} C\Phi^{i-1}\Gamma\end{bmatrix} u_{k-1}}_{\text{"free" or "unforced response" (if no more control moves are made)}}$$

f

$$\underbrace{\underbrace{\begin{bmatrix} C\Gamma & 0 & \cdots & 0 \\ C\Phi\Gamma + C\Gamma & C\Gamma & 0 & 0 \\ \vdots & \vdots & & \\ \sum_{i=1}^{P} C\Phi^{i-1}\Gamma & \sum_{i=1}^{P-1} C\Phi^{i-1}\Gamma & \cdots & \end{bmatrix}}_{S_f} \underbrace{\begin{bmatrix}\Delta u_{k} \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+M-1}\end{bmatrix}}_{\Delta u_f}}_{\text{"forced" response}}$$

B. Wayne Bequette

Output Predictions

$$\underbrace{\begin{bmatrix}\hat{y}_{k+1|k}\\ \hat{y}_{k+2|k}\\ \vdots\\ \hat{y}_{k+P|k}\end{bmatrix}}_{\hat{Y}} = \underbrace{\underbrace{\begin{bmatrix}C\Phi\\ C\Phi^{2}\\ \vdots\\ C\Phi^{P}\end{bmatrix}\hat{x}_{k|k} + \begin{bmatrix}S_1\\ S_2\\ \vdots\\ S_P\end{bmatrix}u_{k-1}}_{\text{"free" or "unforced response" (if no more control moves are made)}}}_{f}$$

Step response coefficients

$$\underbrace{\underbrace{\begin{bmatrix}S_1 & 0 & \cdots & 0\\ S_2 & S_1 & 0 & 0\\ \vdots & \vdots & & \\ S_P & S_{P-1} & \cdots & S_{P-M+1}\end{bmatrix}}_{S_f}\underbrace{\begin{bmatrix}\Delta u_k\\ \Delta u_{k+1}\\ \vdots\\ \Delta u_{k+M-1}\end{bmatrix}}_{\Delta u_f}}_{\text{"forced" response}}$$

B. Wayne Bequette

Optimization Problem

$$\min_{\Delta u_f} \quad J = \underbrace{\sum_{i=1}^{P}\left(r_{k+i|k} - \hat{y}_{k+i|k}\right)^T W^y \left(r_{k+i|k} - \hat{y}_{k+i|k}\right)}_{\hat{E}^T W^Y \hat{E}} + \underbrace{\sum_{i=0}^{M-1} \Delta u_{k+i}^T W^u \Delta u_{k+i}^T}_{\Delta u_f^T W^U \Delta u_f}$$

Where

$$W^Y = \begin{bmatrix} W^y & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & W^y \end{bmatrix} \qquad W^U = \begin{bmatrix} W^u & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & W^u \end{bmatrix}$$

future setpoints

and

$$\hat{E} = r - \hat{Y} = \underbrace{r - f}_{E} - S_f \Delta u_f$$

"unforced" (free response) error

so

$$\hat{E} = E - S_f \Delta u_f$$

B. Wayne Bequette

Optimization Problem

$$\hat{E}^T W^Y \hat{E} = \left(E - S_f \Delta u_f\right)^T W^Y \left(E - S_f \Delta u_f\right)$$

$$= E^T W^Y E - 2\Delta u_f^T S_f^T W^Y E + \Delta u_f^T S_f^T W^Y S_f \Delta u_f$$

so

$$\min_{\Delta u_f} \quad J = \hat{E}^T W^Y \hat{E} + \Delta u_f^T W^U \Delta u_f$$

can be written

$$\min_{\Delta u_f} \quad J = \Delta u_f^T \left(S_f^T W^Y S_f + W^U\right) \Delta u_f - 2\Delta u_f^T S_f^T W^Y E$$

and the unconstrained solution is found from

$$\partial J / \partial \Delta u_f = 0$$

B. Wayne Bequette

Unconstrained Solution

Analytical Solution for Unconstrained System

$$\Delta u_f = \left(S_f^T W^Y S_f + W^U\right)^{-1} S_f^T W^Y E$$

"unforced" error

In practice, do not actually invert a matrix. Solve as set of simultaneous equations (or use \ in MATLAB)

$$\Delta u_f = \left(S_f^T W^Y S_f + W^U\right) \backslash S_f^T W^Y E$$

Vector of Control Moves

$$\Delta u_f = \underbrace{\begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+M-1} \end{bmatrix}}_{\text{current and future moves}}$$

Although a set of control moves is computed, only the first move Δu_k is implemented

The next output at k+1 is obtained, then a new optimization problem is solved

MPC Tuning Parameters

- Prediction Horizon, P

- Control Horizon, M

- Manipulated Input Weighting, W^u

Usually, **P >> M** for robustness (less aggressive action). Sometimes M = 1, with P varied for desired performance.

Sometimes larger input weights for robustness

Pre-Summary

- Concise overview of MPC
- State space model, unconstrained solution
- Have not discussed
  - State estimation and "**corrected outputs**"
    - **The additive disturbance assumption of DMC is covered in the slides that follow (this is identical to the plant-model mismatch term in IMC)**
  - Disturbances
  - Constraints
  - Other model forms

Original DMC Approach to Plant-Model Mismatch

$$\hat{x}_{k|k-1} = \Phi \hat{x}_{k-1|k-1} + \Gamma u_{k-1}$$

$$\hat{y}_{k|k-1} = C \hat{x}_{k|k-1}$$

Prediction at step k, based on information at k-1

Measured output

$$\hat{p}_{k|k} = y_k - \hat{y}_{k|k-1}$$

Model output predicted from k-1

$$\hat{y}_{k|k} = \hat{y}_{k|k-1} + \hat{p}_{k|k}$$

“additive output” disturbance assumption (previously d$_k$)

Forces the model “corrected output” equal to measured output

Notice that Model States are Not “Corrected”

$$\hat{x}_{k|k} = \hat{x}_{k|k-1}$$

Model Prediction to k+1

$$\hat{x}_{k+1|k} = \Phi\hat{x}_{k|k} + \Gamma u_k$$

$$\hat{p}_{k+1|k} = \hat{p}_{k|k}$$

Assumes future corrections equal to current correction

$$\hat{y}_{k+1|k} = C\hat{x}_{k+1|k} + \hat{p}_{k+1|k} =$$

$$= C\Phi\hat{x}_{k|k} + C\Gamma u_k + \hat{p}_{k+1|k}$$

$$= C\Phi\hat{x}_{k|k} + C\Gamma u_{k-1} + C\Gamma\Delta u_k + \hat{p}_{k+1|k}$$

Continue Output Predictions

$$\underbrace{\begin{bmatrix} \hat{y}_{k+1|k} \\ \hat{y}_{k+2|k} \\ \vdots \\ \hat{y}_{k+P|k} \end{bmatrix}}_{\hat{Y}} = \underbrace{\begin{bmatrix} C\Phi \\ C\Phi^{2} \\ \vdots \\ C\Phi^{P} \end{bmatrix} \hat{x}_{k|k} + \begin{bmatrix} I \\ I \\ \vdots \\ I \end{bmatrix} \hat{p}_{k|k} + \begin{bmatrix} C\Gamma \\ C\Phi\Gamma + C\Gamma \\ \vdots \\ \sum_{i=1}^{P} C\Phi^{i-1}\Gamma \end{bmatrix} u_{k-1}}_{\text{"free" or "unforced response" (if no more control moves are made)}}$$

f

$$\underbrace{\begin{bmatrix} C\Gamma & 0 & \cdots & 0 \\ C\Phi\Gamma + C\Gamma & C\Gamma & 0 & 0 \\ \vdots & \vdots & & \\ \sum_{i=1}^{P} C\Phi^{i-1}\Gamma & \sum_{i=1}^{P-1} C\Phi^{i-1}\Gamma & \cdots & \end{bmatrix} \begin{bmatrix} \Delta u_{k} \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+M-1} \end{bmatrix}}_{\text{"forced" response}}$$

S_f

Δu_f

Output Predictions

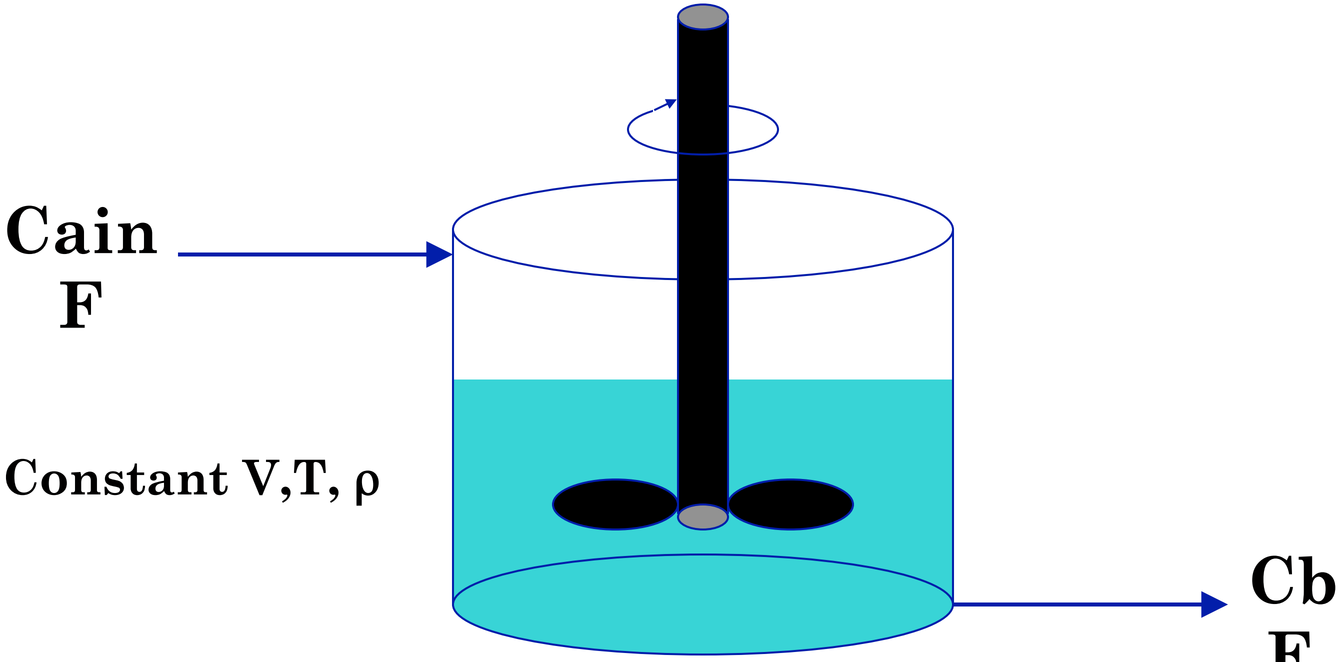
$$\underbrace{\begin{bmatrix} \hat{y}_{k+1|k} \\ \hat{y}_{k+2|k} \\ \vdots \\ \hat{y}_{k+P|k} \end{bmatrix}}_{\hat{Y}} = \underbrace{\begin{bmatrix} C\Phi \\ C\Phi^2 \\ \vdots \\ C\Phi^P \end{bmatrix} \hat{x}_{k|k} + \begin{bmatrix} I \\ I \\ \vdots \\ I \end{bmatrix} \hat{p}_{k|k} + \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_P \end{bmatrix} u_{k-1}}_{\text{"free" or "unforced response" (if no more control moves are made)}}$$

$$f$$

$$\underbrace{\begin{bmatrix} S_1 & 0 & \cdots & 0 \\ S_2 & S_1 & 0 & 0 \\ \vdots & \vdots & & \\ S_P & S_{P-1} & \cdots & S_{P-M+1} \end{bmatrix} \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+M-1} \end{bmatrix}}_{\text{"forced" response}}$$

$$S_f \qquad\qquad \Delta u_f$$

Example Inverse Response Process: Van de Vusse



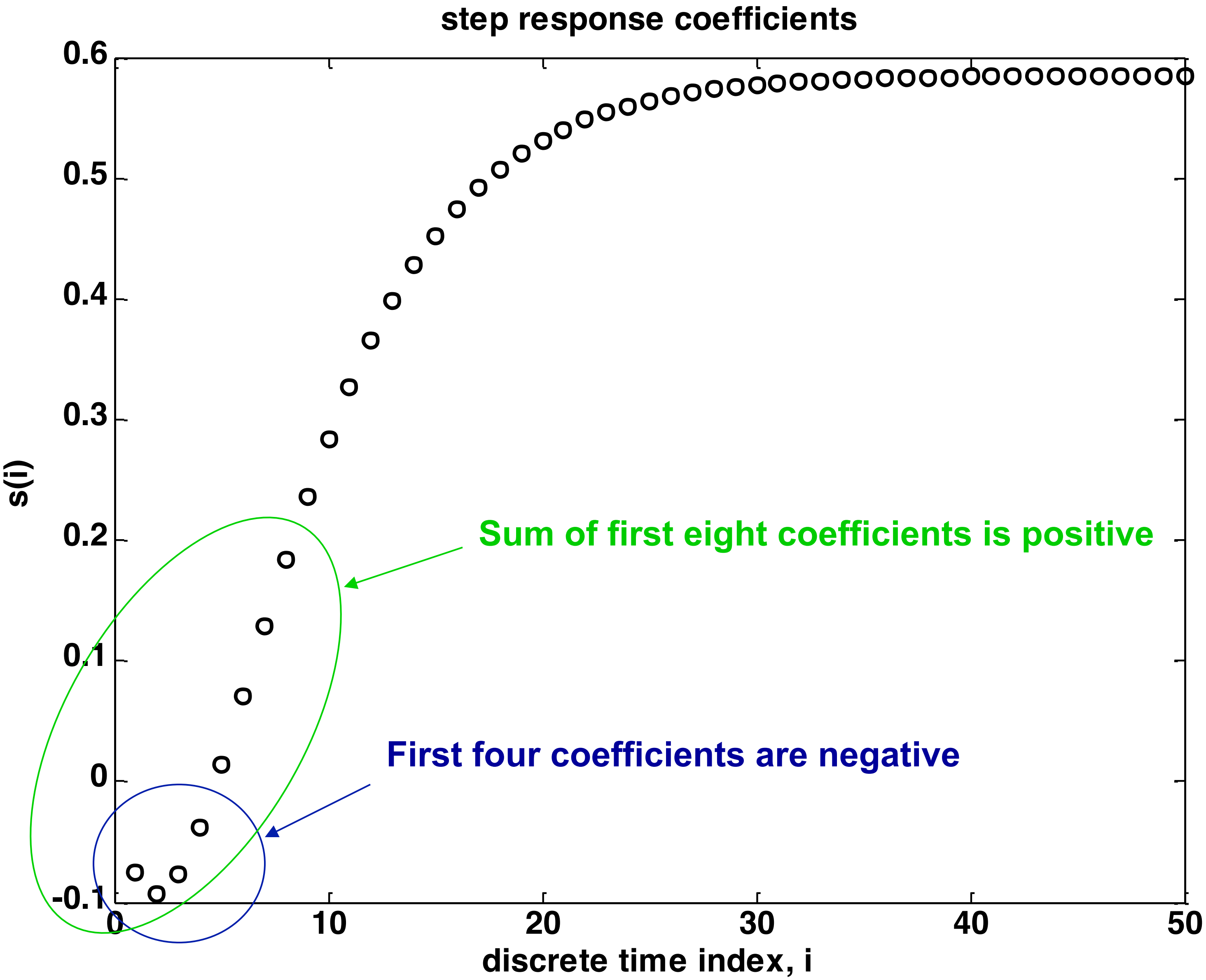
$$A \xrightarrow{k_1} B \xrightarrow{k_2} C \qquad A+A \xrightarrow{k_3} D$$

$$\frac{dCa}{dt} = -k_1Ca - k_3Ca^2 + (Cain - Ca)u$$

$$\frac{dCb}{dt} = k_1Ca - k_2Cb - Cbu \qquad \text{where } u = F/V$$

B. Wayne Bequette

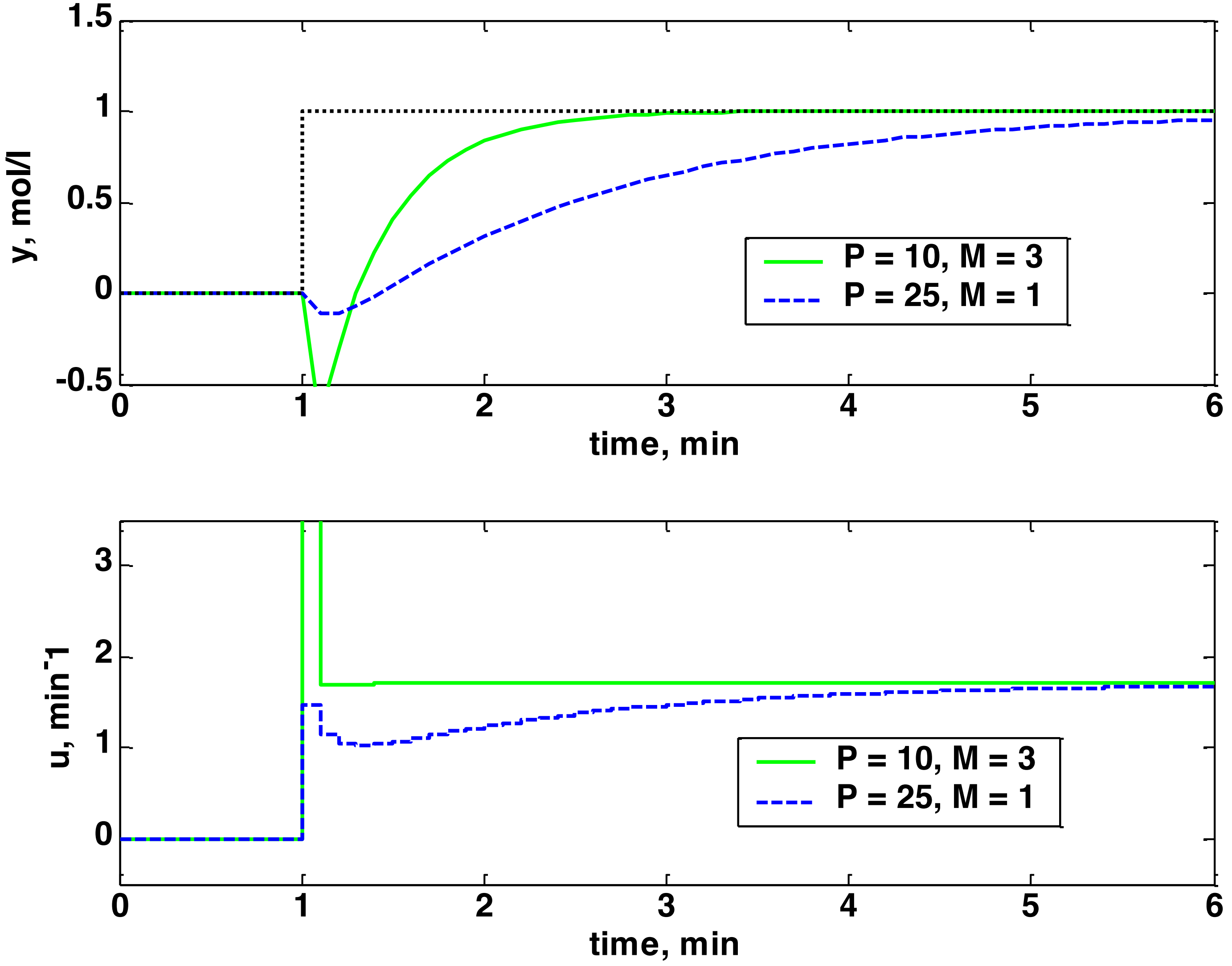
Example: Inverse Response Process



Sum of first eight coefficients is positive

First four coefficients are negative

Closed-Loop: Compare P=10, M=1 with P=25, M=1

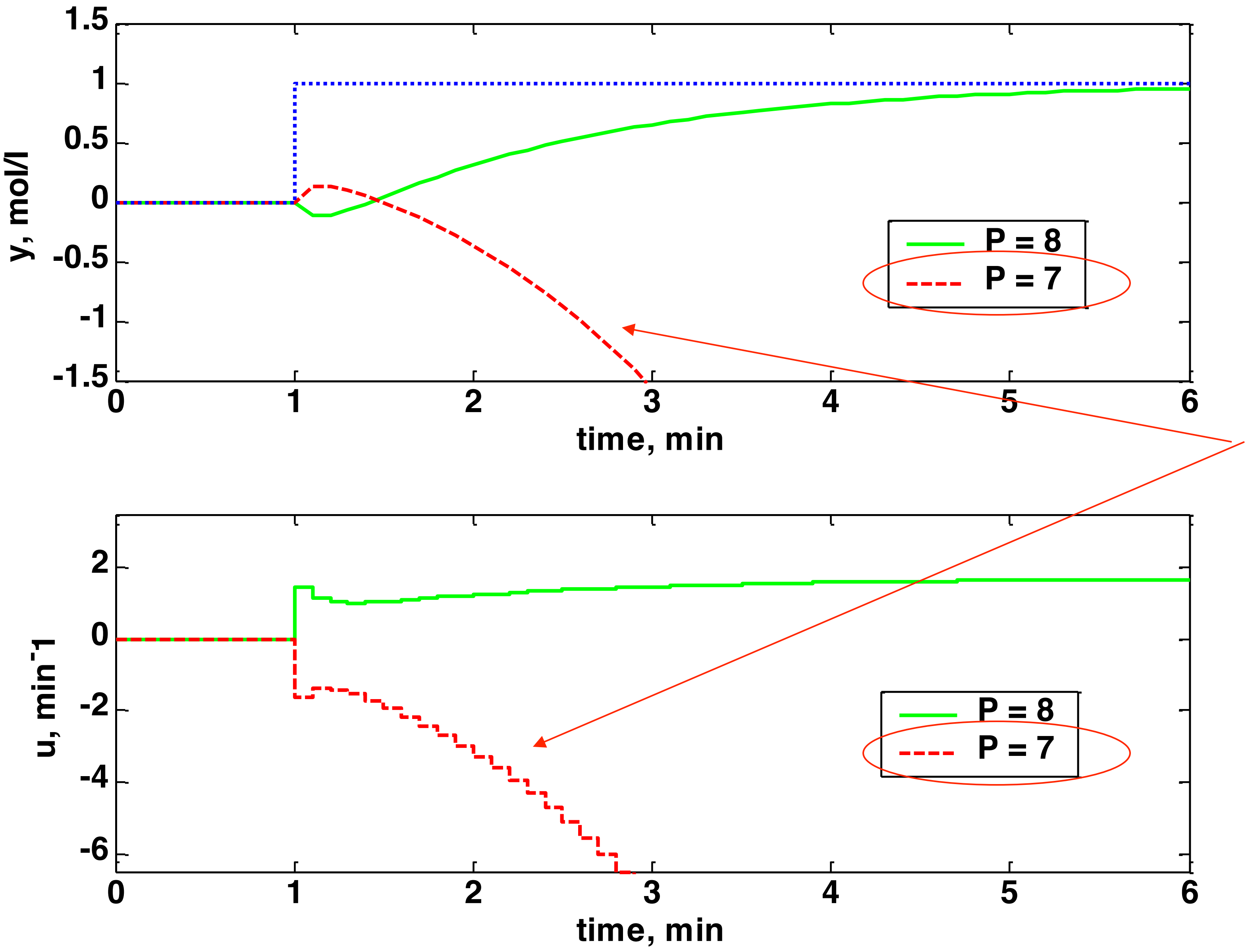


Short prediction horizons & long control horizons lead to more aggressive action

B. Wayne Bequette

Results

Control Horizon: M = 1, Weighting: W = 0



P=7:
Unstable

B. Wayne Bequette

Stability of Inverse Response Systems with DMC

- For a control horizon, M = 1, closed-loop MPC will be stable for a prediction horizon where the sum of the impulse response coefficients has the same sign as the process gain

$$\sum_{i=1}^{P_{\min}} s_i > 0$$

Predictive Controller Design for Single-Input/Single-Output (SISO) Systems

Paul R. Maurath,[†] Duncan A. Mellichamp, and Dale E. Seborg*

Department of Chemical and Nuclear Engineering, University of California, Santa Barbara, Santa Barbara, California 93106

Ind. Eng. Chem. Res. **1988**, *27*, 956–963

For the example, $P_{\min}$ = 8, so P = 7 = unstable

Summary

- Concise overview of MPC
- State space model, unconstrained solution
- The additive disturbance assumption of DMC is used for plant-model mismatch
- Upcoming topics
  - Disturbances and state estimation (Kalman filtering framework)
  - Constraints
  - Other model forms