Model Predictive Control

1. Theory

Niket S. Kaisare
Department of Chemical Engineering
Indian Institute of Technology - Madras

Summer School on Process Identification and Control
NIT-Tiruchirappalli, June 25th 2008
Hierarchy of Process Control

- **Regulatory Control**
- **Advanced Process Control**
- **Real-Time Optimizer**
- **Planning Scheduling**
- **Strategic Decisions**

Time-scale:
- **Seconds**
- **Minutes ~ Hours**
- **Hours ~ Day**
- **Week ~ Month**
- **Quarterly ~ Annual**
Example: Shell Heavy Oil Fractionator

- Hierarchical control strategy
  - PID loops for level & inventory control
  - Quality control loops
  - Minimise energy usage

- Several operating constraints
What is Model Predictive Control?
Exemplary MPC Algorithm

**Objective:**
\[
\max \sum \text{productivity}
\]

**Model:**
\[
\dot{x} = f(x, u, d)
\]

**Constraints:**
\[
x_{\min} \leq x \leq x_{\max}
\]

**Controller**

**Plant**

**Observer**

**Disturbances**
Dynamic Matrix Control:
Industrial Origin of MPC

- Initiated at Shell Oil and other refineries (1970s)
- Various commercial software
  - DMCplus - Aspen Tech
  - RMPCT - Honeywell
  - Dozen+ other players (e.g., 3DMPC-ABB)
- >4500 worldwide installations
- Predominant in oil and petrochemical industries
- The range of applications is expanding
## A Brief History of MPC

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Model</th>
<th>Objective</th>
<th>Pred. Horizon</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>LQG (1960)</td>
<td>L SS</td>
<td>min ISE IO</td>
<td>infinity</td>
<td>-</td>
</tr>
<tr>
<td>IDCOM (1976)</td>
<td>L conv</td>
<td>min ISE O</td>
<td>p</td>
<td>IO</td>
</tr>
<tr>
<td>DMC (1979)</td>
<td>L conv</td>
<td>min ISE IOM</td>
<td>p</td>
<td>IO</td>
</tr>
<tr>
<td>QDMC (1983)</td>
<td>L conv</td>
<td>min ISE IOM</td>
<td>p</td>
<td>IO</td>
</tr>
<tr>
<td>GPC (1987)</td>
<td>L ARMA</td>
<td>min ISE IO</td>
<td>p</td>
<td>-</td>
</tr>
<tr>
<td>SMOC (1988)</td>
<td>L SS</td>
<td>min ISE IO</td>
<td>p</td>
<td>IO</td>
</tr>
<tr>
<td>Rawlings and Muske (1993)</td>
<td>L SS</td>
<td>min ISE IO</td>
<td>infinity</td>
<td>IO</td>
</tr>
</tbody>
</table>
Multivariable Process Control

**Conventional Structure**

- Plant-Wide Optimization
- Unit 1 Local Optimization
  - High/Low Select Logic
  - PID
  - Lead/Lag
  - SUM
- Unit 1 Distributed Control System (PID)

**MPC Structure**

- Model Predictive Control (MPC)
- Unit 2 Local Optimization
  - PID
  - SUM
- Unit 2 Distributed Control System (PID)

**Optimization**

- Global SS Optimization
- Local SS Optimization
- Dynamic Supervisory/Constraint Control
- Basic Dynamic Control

**Control Strategies**

- FC
- PC
- TC
- LC
Almost all models used in MPC are typically empirical models “identified” through plant tests rather than first-principles models.
Model Predictive Control (MPC) refers to a class of algorithms that utilize an explicit process model to compute a manipulated variable profile that will optimize an open-loop performance objective over a future time interval.

The performance objective typically penalizes predicted future errors and manipulated variable movement subject to constraints.
Model Predictive Control (MPC) refers to a class of algorithms that utilize an explicit process model to compute a manipulated variable profile that will optimize an open-loop performance objective over a future time interval. The performance objective typically penalizes predicted future errors and manipulated variable movement subject to constraints.
Receding Horizon Control Concept

Past Future

$\text{setpoint}$

output

input
Analogy to Chess

Opponent (Plant)

Model Predictive Control

Opponent’s Move → New Piece Position

Me → Model Predictive Control
Major Components of MPC

1. Dynamic state (stored in memory)

   Predicted future outputs = Effect of current “state” (memory) + Future input adjustments + Feedforward measurement + Feedback correction

2. Multi-step prediction equation

3. Objectives and Constraints

4. Optimization solver (Linear / Quadratic Program)

5. State update scheme
General Setup

Future Input Moves (To Be Determined) → Prediction Model for Future Outputs → To Optimization

- **Current State**
- **Prediction**
- **Dynamic Model State:** Compact representation Of the past input record

- **Measurement Correction**
- Feedback / Feedforward Measurements
General Setup

Previous State (in Memory)

Current State

State Update

New Input Move (Just Implemented)

Prediction

Future Input Moves (To Be Determined)

Prediction Model for Future Outputs

To Optimization

Measurement Correction

Feedback / Feedforward Measurements
MPC Options

- **Model Types**
  - Finite Impulse Response Model or **Step Response Model**
  - State-Space Model
  - Linear or Nonlinear

- **Measurement Correction**
  - To the prediction (based on open-loop state calculation)
  - To the state (through state estimation)

- **Objective Function**
  - Linear or **Quadratic**
  - Constrained or Unconstrained
Step Response Model

Assumptions:
- No immediate effect: $S_0 = 0$
- Stable process settles after time $n$: $S_j = S_n$ for $j \geq n + 1$
Step Response Model: Superposition Principle

\[ v(k) = \sum_{i=0}^{k} \Delta v(i) \]
Step Response Model: Superposition Principle

\[ y(k) = \sum_{i=1}^{n-1} \Delta v(k-i)S_i + \Delta v(k-n)S_n + \Delta v(k-n-1)S_n + \cdots + v(0)S_n \]

1. Dynamic Model
Extension to Multivariate Systems

1. Dynamic Model

\[ S_i \triangleq \begin{bmatrix} S_{1,1,i} & S_{1,2,i} \\ S_{2,1,i} & S_{2,2,i} \end{bmatrix} \]
State Definition

Effect of past inputs is kept in memory as the system state

Possible choices of state:
- Past $n$ input moves, or
- Future $n$ output response assuming $v$ is held constant at $v(k)$

1b. Definition of “State”
Define:

1. \( y_i(k) = y(k + i) \) assuming \( \Delta v(j)=0 \) for

2. Effect of past inputs on n-step future predictions is the current state:

\[
x(k) = \begin{bmatrix} y_0^T(k) & y_1^T(k) & \cdots & y_{n-1}^T(k) \end{bmatrix}^T
\]
1. Dynamic State for Step Response Model

- **Step response model**
  \[ y(k) = \sum_{i=1}^{n} S_i \Delta v(k - i) + S_n v(k - n) \]

- **Definition of the state**
  - Stores n future output response in the memory assuming the input is kept constant at \( u(k-1) \)

\[ \tilde{Y}(k) = \begin{bmatrix} y_0(k) \\ y_1(k) \\ \vdots \\ y_{n-1}(k) \end{bmatrix} \]

\[ y_i(k) = y(k + i) \quad \text{with} \quad \Delta u(k) = \Delta u(k + 1) = \cdots = 0 \]
Main interest for control: predicting future output behavior.

→ Future output = function of (past input, hypothesized future input)

→ Dynamic state: memory about the effect of past input

→ Future output = function of (dynamic state, future input)
Multi-Step Prediction

\[ x(k) \]

Effect of future input moves (to be determined)
2. Multi-Step Prediction

\[
\begin{bmatrix}
y(k+1|k) \\
y(k+2|k) \\
\vdots \\
y(k+p|k)
\end{bmatrix}
= 
\begin{bmatrix}
y_1(k) \\
y_2(k) \\
\vdots \\
y_p(k)
\end{bmatrix}
\]

Effect of past inputs

\[
\begin{bmatrix}
S_1 \\
S_2 \\
\vdots \\
S_p
\end{bmatrix}
\Delta u(k) + 
\begin{bmatrix}
0 \\
S_1 \\
\vdots \\
S_{p-1}
\end{bmatrix}
\Delta u(k+1) + \cdots + 
\begin{bmatrix}
0 \\
0 \\
\vdots \\
S_{p-m+1}
\end{bmatrix}
\Delta u(k+m-1)
\]

Effect of current and future inputs

\[
\begin{bmatrix}
S_1^d \\
S_2^d \\
\vdots \\
S_p^d
\end{bmatrix}
\Delta d(k) + 
\begin{bmatrix}
0 \\
S_1^d \\
\vdots \\
S_{p-1}^d
\end{bmatrix}
\Delta d(k+1) + \cdots
\]

Effect of measured disturbances

\[
\begin{bmatrix}
e(k+1|k) \\
e(k+2|k) \\
\vdots \\
e(k+p|k)
\end{bmatrix}
\]

Unmeasured disturbances

2. Multi-step Prediction
2. Multi-Step Prediction

Assume measured disturbance remains constant: \( d(k) = d(k+1) = d(k+2) = \ldots \)
\( \Delta d(k+1) = \Delta d(k+2) = \ldots = 0 \)

Previous prediction error is added as a constant bias term
\( e(k+1) = e(k+2) = \ldots e(k+p) = y_{\text{measured}}(k) - y(k) \)

\[
\begin{bmatrix}
y(k+1|k) \\
y(k+2|k) \\
\vdots \\
y(k+p|k)
\end{bmatrix}
= \mathcal{M}\tilde{Y}(k) +
\begin{bmatrix}
S_1 & 0 & \cdots & 0 \\
S_2 & S_1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
S_p & S_{p-1} & \cdots & S_{p-m+1}
\end{bmatrix}
\begin{bmatrix}
\Delta u(k) \\
\Delta u(k+1) \\
\vdots \\
\Delta u(k+m-1)
\end{bmatrix}
\]
\[
\begin{bmatrix}
S_1^d \\
S_2^d \\
\vdots \\
S_p^d
\end{bmatrix}
\Delta d(k) +
\begin{bmatrix}
1 \\
1 \\
\vdots \\
1
\end{bmatrix}
(y(k) - \tilde{y}_0(k))
\]
3. Objective Function and Constraints

Constraint violation may not be acceptable

Shaded region is the setpoint tracking error

\[
\begin{align*}
\min_{\Delta u(k), \ldots, \Delta u(k+m-1)} & \left\{ \sum_{i=1}^{p} [y_{SP} - y(k+i|k)]^T Q [y_{SP} - y(k+i|k)] + \sum_{l=0}^{m-1} [\Delta u(k+l)]^T R [\Delta u(k+l)] \right\} \\
\end{align*}
\]

3. Objective function and constraints
Output Trajectories

3. Objective function
Constraints include

- **Input magnitude constraints:**  \( u_{\text{min}} \leq u(k+i) \leq u_{\text{max}} \)
- **Input rate constraints:**  \( |\Delta u(k+i)| \leq \Delta u_{\text{max}} \)
- **Output magnitude constraints:**  \( y_{\text{min}} \leq y(k+i) \leq y_{\text{max}} \)
3. Constraints

Constraint Formulations

Hard constraint

Soft constraint

Setpoint approximation of soft constraint

past future

past future

past future

past future
Optimisation in Receding Horizon Control

Past Future

setpoint

output
input

Which input move to choose?

$k$, $k+1$, $k+m-1$, $k+p-1$

3&4. Objective function and Optimisation
4. Optimization: Quadratic Program

Basic form of Quadratic Program:

$$\min_{\Delta u} \left( \Delta u^T H \Delta u - g^T \Delta u(k) \right)$$
$$C \Delta u \geq c$$

\(H\) : hessian matrix
\(g\) : gradient vector
\(C\) : constraint matrix
\(c\) : constraint vector
\(\Delta u\) : decision variable
3&4. Objective Function for QP
3&4. Constraint Formulation for QP

\[ u_{\text{min}} \leq u(k + \ell | k) \leq u_{\text{max}}, \quad \ell = 0, \ldots, m - 1 \]

\[ u(k - 1) + \sum_{i=0}^{\ell} \Delta u(k + i | k) \geq u_{\text{min}} \]

\[ -u(k - 1) - \sum_{i=0}^{\ell} \Delta u(k + i | k) \geq -u_{\text{max}} \]
3&4. Constraint Formulation in QP

**Manipulated Variable Rate Constraints**

\[
|\Delta u(k + \ell|k)| \leq \Delta u_{\text{max}}, \quad \ell = 0, \cdots, m - 1
\]

\[
\downarrow
\]

\[-\Delta u_{\text{max}} \leq \Delta u(k + \ell|k) \leq \Delta u_{\text{max}}
\]

\[
\downarrow
\]

\[
\Delta u(k + \ell|k) \geq -\Delta u_{\text{max}}
\]

\[-\Delta u(k + \ell|k) \geq -\Delta u_{\text{max}}
\]

\[
\downarrow
\]

\[
\begin{bmatrix}
I & 0 & \cdots & 0 \\
0 & I & 0 & \vdots \\
\vdots & \cdots & \ddots & 0 \\
0 & 0 & \cdots & I \\
I & 0 & \cdots & 0 \\
0 & I & 0 & \vdots \\
\vdots & \cdots & \ddots & 0 \\
0 & 0 & \cdots & I
\end{bmatrix}
\begin{bmatrix}
\Delta u(k|k) \\
\Delta u(k + 1|k) \\
\vdots \\
\Delta u(k + m - 1|k)
\end{bmatrix}
\geq
\begin{bmatrix}
\Delta u_{\text{max}} \\
\Delta u_{\text{max}} \\
\vdots \\
\Delta u_{\text{max}} \\
\Delta u_{\text{max}} \\
\vdots \\
\Delta u_{\text{max}}
\end{bmatrix}
\]
Output Magnitude Constraints

\[ y_{\text{min}} \leq y(k + j | k) \leq y_{\text{max}}, \quad j = 1, \ldots, p \]

\[ \Downarrow \]

\[ y(k + j | k) \geq y_{\text{min}} \]

\[ -y(k + j | k) \geq -y_{\text{max}} \]

\[ \Downarrow \]

\[ \begin{bmatrix} M_p \bar{Y}(k) + S^d \Delta d(k) + \mathcal{I}_p(y(k) - \bar{y}(k/k)) + S^{\text{dU}} \Delta \mathcal{U}(k) \\ -M_p \bar{Y}(k) - S^d \Delta d(k) - \mathcal{I}_p(y(k) - \bar{y}(k/k)) - S^{\text{dU}} \Delta \mathcal{U}(k) \end{bmatrix} \geq \begin{bmatrix} \mathcal{Y}_{\text{min}} \\ -\mathcal{Y}_{\text{max}} \end{bmatrix} \]
In summary, we have

\[
\begin{bmatrix}
I_L & -I_L \\
-I_L & I \\
S^d & -S^d \\
\end{bmatrix}
\Delta \mathcal{U}(k) \geq
\begin{bmatrix}
- \mathcal{Y}_\text{min} + M_p \bar{Y}(k) + \mathcal{S}^d \Delta d(k) + \mathcal{I}_p (y(k) - \bar{y}(k/k)) \\
- \mathcal{Y}_\text{max} + M_p \bar{Y}(k) + \mathcal{S}^d \Delta d(k) + \mathcal{I}_p (y(k) - \bar{y}(k/k))
\end{bmatrix}
- \begin{bmatrix}
\Delta u_{\text{min}} - u(k-1) \\
\vdots \\
\Delta u_{\text{min}} - u(k-1) \\
\Delta u_{\text{max}} \\
\vdots \\
\Delta u_{\text{max}} \\
\end{bmatrix}

The above is in the standard form of linear equality,

\[
C \Delta \mathcal{U}(k) \geq c(k)
\]
Receding Horizon Control Concept

Past  | Future
---|---
\(k\)  | \(k+1\)  | \(k+m-1\)  | \(k+p-1\)

Output
Input
Setpoint
5. State Update Scheme

Recall from previous lecture: State Update Equation

\[ x(k + 1) = M_1 x(k) + \underbrace{S}_{\text{shift}} \Delta v(k) \]

\[ \begin{bmatrix}
    y_0(k + 1) \\
    y_1(k + 1) \\
    \vdots \\
    y_{n-2}(k + 1) \\
    y_{n-1}(k + 1)
\end{bmatrix} = \begin{bmatrix}
    y_0(k) \\
    y_1(k) \\
    \vdots \\
    y_{n-2}(k) \\
    y_{n-1}(k)
\end{bmatrix} + \sum S_i \Delta v(k) \]
Model Predictive Control (MPC) refers to a class of algorithms that utilize an explicit *process model* to compute a manipulated variable profile that will **optimize** an open-loop performance objective over a future time interval.

The performance objective typically penalizes *predicted future errors* and manipulated variable movement subject to *constraints*. 
Model Predictive Control
2. Applications

Niket S. Kaisare
Department of Chemical Engineering
Indian Institute of Technology - Madras

Summer School on Process Identification and Control
NIT-Tiruchirappalli, June 25th 2008
Example 1: Blending System Control

- Control $r_A$ and $r_B$
- Control blend flow $q$
- Constraints on MV
Example 1: Blending System Control

- Control $r_A$ and $r_B$
- Control blend flow $q$
- Constraints on MV

Minimize deviation from the set-point for $p$ steps into the future.

$$
\min_{u_k, u_{k+1}, \ldots, u_{k+m-1}} \sum_{i=1}^{p} \left[ (r_A - r_A^{sp})^2 + (r_B - r_B^{sp})^2 \right] + \gamma (q - q^{sp})^2
$$

$u_{\min} \leq u_i \leq u_{\max}$
Example 2: Heavy Oil Fractionator

- **Objective**
  - Keep temperature $y_7$ above a minimum value
  - Keep $y_1$ and $y_2$ at setpoint
  - Minimize $u_3$ to maximize heat recovery

- **Several constraints**

- **Non-square system**
Example 3: Ethylene Plant

- Furnaces
- Primary Fractionator Tower
- Quench Tower
- Charge Gas Compressor
- Chilling
- Demethanizer
- Deethanizer
- Ethylene Fractionator
- Hydrogen
- Methane
- Ethylene
- Ethane
- Propylene
- Propane
- B - B
- Gasoline
- Fuel Oil
- Feedstock
- Naphtha
- Light H-C
- Debutanizer
- Propylene Fractionator
Typical Scale of Modeling and Control
Popularity of MPC
Systematic and Integrated Solution

More and more optimization is done at the MPC level.

Selectors, Switches, Delay Compensations, Anti-windups, Decouplers, etc.

Low-level PID Loops

To control valves

Advanced MV Control MPC

Low-level PID Loops
Popularity of MPC
Systematic and Integrated Solution

Advantages over traditional APC

- Integrated solution
  - automatic constraint handling
  - Feedforward / feedback
  - No need for decoupling or delay compensation

- Efficient Utilization of degrees of freedom
  - Can handle nonsquare systems (e.g., more MVs and CVs)
  - Assignable priorities, ideal settling values for MVs

- Consistent, systematic methodology

- Realized benefits
  - Higher on-line times
  - Cheaper implementation
  - Easier maintenance
Popularity of MPC

Emerging Popularity of On-line Optimization

- Process optimization and control are often conflicting objectives
  - Optimization pushes the process to the boundary of constraints.
  - Quality of control determines how close one can push the process to the boundary.

- Implications for process control
  - High performance control is needed to realize on-line optimization.
  - Constraint handling is a must.
  - The appropriate tradeoff between optimization and control is time-varying and is best handled within a single framework.
Popularity of MPC

Emerging Popularity of On-line Optimization

Conflict / Synergy Between Optimization and Control
Popularity of MPC

Emerging Popularity of On-line Optimization

Bi-level Optimization in MPC

Steady-State Optimization (LP)

Optimal setpoints

S.S. prediction model

Dynamic Optimization (QP)

Regulate low-level loops

Measurements (feedback update)

Economic optimization

Minimization of tracking error
Example: Model Predictive Control of Mammalian Cell Cultures

Niket S. Kaisare, Jay H. Lee
A. A. Namjoshi, D. Ramkrishna

Presented at AIChE Meeting, November 2000
Overview of the Talk

- Introduction
  - Hybridoma cell reactor
- Cybernetic Modeling
- Case Study
  - Simpler microbial cell system
  - Full hybridoma cell system
- Conclusions
Introduction and Motivation

- Hybridoma Cells
  - Hybrid mammalian cells
  - Produce monoclonal antibodies

- Applications of antibodies
  - Protein detection/purification
  - Immunotherapy
  - Tumor imaging

- B-lymphocytes
  - Produce antibodies
- Myeloma
  - Tumor cells
  - Rapid in-vitro growth

Hybridoma
Hybridoma Cell Reactor

- Glucose ($s_{1f}$)
- Glutamine ($s_{2f}$)
- Amino acids

Air ($O_2$)

- Unconsumed substrates
- Biomass w/ antibodies
- Waste metabolites (Lactate, Alanine etc.)
Three types of steady states observed for same input conditions.

Differ in the metabolic states of the cells.

Start up Experiments, Hu et al. (1998)
Hybridoma Reactor Control

■ Previous Work
  - Detremblay et al. (1993): Estimation and PI control
  - Lenas et al. (1997): Adaptive fuzzy control
  - Dhir et al. (2000)
    - Fuzzy update of model parameters
    - Heuristic random optimizer (HRO)
  - Europa et al. (2000): Heuristic scheme

■ Our focus: Model based optimal control
Bioreactor Modeling

- Monod type model
  \[ r_{S\to P} = \mu^{\text{max}} \frac{S}{K + S} \]
  - Phenomenological modifications possible
  - Does not capture cellular metabolic regulations

- Detailed description of extremely large number of metabolic reactions
  - Unsuitable for on-line optimization and control
Cybernetic Modeling (1)

Characteristics

- Captures internal cellular regulatory mechanisms
- Relatively fewer metabolic reactions
- Accurate description of average cell in a batch, fed-batch or continuous culture
  - Diauxie growth phenomenon
  - Mammalian cultures
  - PHB storage intermediate synthesis
Postulates

- Kuberne’tes (Greek): “Steersman”
- A Metabolic Network has physiological objectives
- Limited pool of resources
- A cell directs the synthesis and activity of enzymes in a metabolic competition
  - Mathematically, through cybernetic regulation variables
  - $u$ modifies enzyme synthesis
  - $v$ modifies enzyme activity

- Cells are optimal strategists
Analysis of a Simpler Model

- Single cybernetic competition

- Bacterial cell growth
  - Diauxie phenomenon (Kompala et al., 1986)
  - To understand cybernetic competitions

- Model Equations

\[
\frac{dS_i}{dt} = D(S_{if} - S_i) - (r_{iv_i})Y_i c \quad i = 1, 2
\]

\[
\frac{de_i}{dt} = (r_e u_i) - \beta e_i - r_{g} e_i + r_e^* \quad i = 1, 2
\]

\[
\frac{dc}{dt} = (r_g - D)c
\]

\[
r_i = \mu_{\text{max}} \frac{S_i}{K_i + S_i} \left[ \frac{e_i}{e_{i,\text{max}}} \right] \quad \nu_i^{sc} = \frac{r_i}{\max(r_1, r_2)}
\]
The Steady State Multiplicity

- Two stable steady states for certain range of substrate feeds

- Objectives are
  - Maintain the reactor at the steady state
  - Drive the reactor to other steady state

Bifurcation diagram for the system (Namjoshi et al., unpublished)
At steady state, we apply a series of step changes in $s_{2f}$

- Measurement noise

- The system drifts to the other steady state in open loop
Model Predictive Control

- **Dynamic Model**
  - Cybernetic model
- **Linear Approximation**
  - NLP - difficult to solve
  - Linearize in the horizon
- **Optimizer**
- **Measurement Correction**
  - Extended K.F.
  - Glucose and biomass feedback

\[
\min_{\Delta u} \left[ \| \Lambda^y \left( R_{k+1} - Y_{k+1|k} \right) \|_2^2 + \| \Lambda^u (\Delta U_k) \|_2^2 \right]
\]
Manipulated and Controlled Variables
(case A: Two M.V.s and two C.V.s)

- C.V.: biomass, $s_2$
- M.V.: D, $s_{1f}$

Controller is able to control the reactor outputs at the given set points.

However, the cells are in physiologically different state.
Case B: One M.V. and two C.V.s

- Changes implemented
  - Decreased weight on exit $s_2$ concentration
  - M.V.: Dilution rate $D$

- Controlled at the desired steady state

- Behavior is better than case 1
Varying the Steady State

- At \( t=50 \) hours, step change in reference trajectory to another steady state

- White measurement noise

The controller is able to reject disturbances and track the reference trajectory
\[ \frac{ds_1}{dt} = D(s_{1f} - s_1) - \left( r_1v_1^{sc}v_1^{sd} + r_2v_2^{sd} + r_1^m + r_2^m \right)c \]

- Set of 16 ODEs
- 4 sets of cybernetic regulation variables
- Result from 4 substitutable cybernetic competitions
**Step Response at the LOW State**

- We give a 10% step down change in D in our simulated reactor
- Highly nonlinear and stiff nature of the system
- Biomass increases with decrease in D
Control Actions: **LOW** to **HIGH** State

- Starting the reactor at the low biomass state
- Step change in the reference trajectory
- Controller drives reactor to the high biomass steady state
Step Response at the HIGH State

- We give a 10% step up change in D in our simulated reactor
- Biomass first increases with increase in D
- When metabolic change is triggered, there is a sudden decrease
- $S_1$ behavior is relatively monotonous
Control Actions: HIGH State

- Drive the reactor from high to intermediate biomass steady state

- Controlled Variable: c

- Prediction horizon is not long enough
Control Actions: **HIGH State**

- Same as previous case
- Controlled var.: $c$ and $s_1$
- Controller takes the reactor to the intermediate state
- We use monotonicity of $s_1$ advantageously
Conclusions

- MPC algorithms were used to drive biological reactors from one state to another "multiple" state.
- Successive linear approximation based controller was applied effectively to the cybernetic models displaying multiplicity features in biological systems.
- Necessary to look at the entire state matrix rather than just the biomass or nutrient conc.
- Non-differentiability of the cybernetic variable was overcome by using relevant approximations in different metabolic states.