Dynamic Real-Time Optimization: Concepts in Modeling, Algorithms and Properties

L. T. Biegler
Chemical Engineering Department
Carnegie Mellon University
Pittsburgh, PA

November 28, 2007

Dynamic Optimization Outline

I Introduction
   Typical Applications
   Problem Statement

II Dynamic Optimization
   Sequential Methods
   Multiple Shooting
   Simultaneous Methods

III Off-line Case Studies
   Unstable Grade Transitions
   Simulated Moving Beds
   Parameter Estimation – Reactor Models

IV On-line Optimization
   NMPC Case Study
   Advanced Step NMPC
   Moving Horizon Estimation

V Conclusions
   Summary
   References
DAE Models in Process Engineering

Differential Equations
Conservation Laws (Mass, Energy, Momentum)

Algebraic Equations
Constitutive Equations, Equilibrium (physical properties, hydraulics, rate laws)
Semi-explicit form
Assume to be index one (i.e., algebraic variables can be solved uniquely by algebraic equations)
If not, DAE can be reformulated to index one (see Ascher and Petzold)

Characteristics
Large-scale models – not easily scaled
Sparse but no regular structure
Direct linear solvers widely used
Coarse-grained decomposition of linear algebra

Batch Distillation Multi-product Operating Policies

• Run between distillation batches
• Treat as boundary value optimization problem
  • When to switch from A to offcut to B?
  • How much offcut to recycle?
  • Reflux?
  • Boilup Rate?
  • Operating Time?
Parameter Estimation

Catalytic Cracking of Gasoil (Tjoa, 1991)

\[ A \rightarrow Q, \quad Q \rightarrow S, \quad A \rightarrow S \]

\[ \dot{\alpha} = -(p_1 + p_3)a^2 \]

\[ \dot{q} = -p_2a^2 - p_3q \]

\[ a(0) = 1, \quad q(0) = 0 \]

number of states and ODEs: 2

number of parameters: 3

no control profiles

constraints: \( p_L \leq p \leq p_U \)

Objective Function: Ordinary Least Squares

\( (p_1, p_2, p_3)^0 = (6, 4, 1) \)

\( (p_1, p_2, p_3)^* = (11.95, 7.99, 2.02) \)

\( (p_1, p_2, p_3)_{true} = (12, 8, 2) \)

Batch Process Optimization

Optimization of dynamic batch process operation resulting from reactor and distillation column

DAE models:

\[ z' = f(z, y, u, p) \]

\[ g(z, y, u, p) = 0 \]

number of states and DAEs: \( n_z + n_y \)

parameters for equipment design

(reactor, column)

\( n_u \) control profiles for optimal operation

Constraints:

\[ u_L \leq u(t) \leq u_U \]

\[ y_L \leq y(t) \leq y_U \]

\[ p_L \leq p \leq p_U \]

Objective Function: amortized economic function at end of cycle time \( t \)
Nonlinear Model Predictive Control (NMPC)

\[ \min \sum \| y(t) - y^p \|^2 + \sum \| u(t) - u(t+1) \|^2 \]
\[ s.t. \]
\[ \dot{z}(t) = F(z(t), y(t), u(t), t) \]
\[ 0 = G(z(t), y(t), u(t), t) \]
\[ z(t) = z^{in} \]

Bound Constraint s
Other Constraint s

Dynamic Optimization Problem

\[ \min \Phi(z(t), y(t), u(t), p, t_f) \]
\[ s.t. \]
\[ \frac{dz(t)}{dt} = f(z(t), y(t), u(t), t, p) \]
\[ g(z(t), y(t), u(t), t, p) = 0 \]
\[ z^* = z(0) \]
\[ z^l \leq z(t) \leq z^u \]
\[ y^l \leq y(t) \leq y^u \]
\[ u^l \leq u(t) \leq u^u \]
\[ p^l \leq p \leq p^u \]

t, time
z, differential variables
y, algebraic variables
u, control variables
p, time independent parameters
Dynamic Optimization Approaches

**DAE Optimization Problem**
- Discretize controls
- Efficient for constrained problems
- Handles instabilities
- Larger NLPs

**Indirect/Variational**
- Pontryagin (1962)
- Inefficient for large, constrained problems

**Direct NLP solution**
- Sullivan (1977)
- Small NLP
- No instabilities

**Simultaneous Approach**
- Bock, Plitt (1984)
- Efficient for constrained problems
- Inefficient for large, constrained problems

**Multiple Shooting**
- Pontryagin (1962)
- Dense Sensitivity Blocks

**Collocation**
- Large/Sparse NLP

Sequential Approaches - Parameter Optimization

Consider a simpler problem without control profiles:

- e.g., equipment design with DAE models - reactors, absorbers, heat exchangers

\[
\begin{align*}
\text{Min} & \quad \Phi(z(t_f)) \\
\dot{z} & = f(z, p), \quad z(0) = z_0 \\
g(z(t_f)) & \leq 0, \quad h(z(t_f)) = 0
\end{align*}
\]

By treating the ODE model as a "black-box" a sequential algorithm can be constructed that can be treated as a nonlinear program.

**Task:** How are gradients calculated for optimizer?
Gradient Calculation

Perturbation
Sensitivity Equations
Adjoint Equations

**Perturbation**
Calculate approximate gradient by solving ODE model \((np + 1)\) times

Let \(\psi = \Phi, g\) and \(h\) (at \(t = t_f\))

\[
\frac{d\psi}{dp_i} = \left\{ \frac{\psi(p_i + \Delta p_i) - \psi(p_i)}{\Delta p_i} \right\}
\]

Very simple to set up

Leads to poor performance of optimizer and poor detection of optimum unless roundoff error \(O(1/\Delta p_i)\) and truncation error \(O(\Delta p_i)\) are small.

Work is proportional to \(np\) (expensive)

---

Direct Sensitivity

From ODE model:

\[
\frac{\partial}{\partial p} \{ \dot{x}' = f(z, p, t), z(0) = z_0(p) \}
\]

Define \(s_i(t) = \frac{\partial \psi(t)}{\partial p_i}, i = 1, \ldots, np\)

\[
\dot{s_i} = \frac{d}{dt}(s_i) = \frac{\partial f}{\partial p_i} + \frac{\partial f}{\partial z} s_i, \quad s_i(0) = \frac{\partial \psi(0)}{\partial p_i}
\]

\((nz \times np)\) sensitivity equations

- \(z\) and \(s_i, i = 1, \ldots, np\), can be integrated forward simultaneously.
- For implicit ODE solvers, \(s_i(t)\) can be carried forward in time after converging on \(z\)
- Linear sensitivity equations exploited in ODESSA, DASSAC, DASPK, DSL48s and a number of other DAE solvers

Sensitivity equations are efficient for problems with many more constraints than parameters \((1 + ng + nh > np)\)
Multiple Shooting for Dynamic Optimization

Divide time domain into separate regions

Integrate DAEs state equations over each region
Evaluate sensitivities in each region as in sequential approach wrt $u_i$, $p$ and $z_j$
Impose matching constraints in NLP for state variables over each region
Variables in NLP are due to control profiles as well as initial conditions in each region

Multiple Shooting Nonlinear Programming Problem

\[
\begin{align*}
\min_{u_{i,j},p} & \quad \psi(z(t_f), y(t_f)) \\
\text{s.t.} & \quad z(t_f, u_{i,j}, p, t_{j+1}) - z_{j+1} = 0 \\
& \quad z_k^l \leq z(t_j, u_{i,j}, p, t_k) \leq z^u_k \\
& \quad y_k^l \leq y(t_j, u_{i,j}, p, t_k) \leq y^u_k \\
& \quad u_i^l \leq u_{i,j} \leq u_i^u \\
& \quad p^l \leq p \leq p^u
\end{align*}
\]

\[
\frac{dz}{dt} = f(z, y, u_{i,j}, p) \quad z(t_j) = z_j \\
g(z, y, u_{i,j}, p) = 0 \\
z_0^o = z(0)
\]

Solved Implicitly
Dynamic Optimization – Multiple Shooting Strategies

Larger NLP problem $O(np+nu+NE\ nz)$
- Use SNOPT, MINOS, etc.
- Second derivatives difficult to get

Repeated solution of DAE model and sensitivity/adjoint equations, scales with nz and np
- Dominant computational cost
- May fail at intermediate points

Multiple shooting can deal with unstable systems with sufficient time elements.

Discretize control profiles to parameters (at what level?)

Path constraints are difficult to handle exactly for NLP approach

Block elements for each element are dense!

Extensive developments and applications by Bock and coworkers using MUSCOD code

Dynamic Optimization Approaches

Dynamic Optimization Approaches

DAE Optimization Problem
Indirect/Variational
- Inefficient for large, constrained problems

Single Shooting
Sullivan (1977)
+ Small NLP
- No instabilities

Multiple Shooting
Bock, Plitt (1984)
+ Embeds DAE Solvers/Sensitivity
- Dense Sensitivity Blocks

Simultaneous Approach
- Handles instabilities
- Larger NLPs

Collocation
Large/Sparse NLP
Nonlinear Dynamic Optimization Problem

Continuous variables

Nonlinear Programming Problem (NLP)

Collocation on finite Elements

Discretized variables

Nonlinear Programming Formulation

Discretization of Differential Equations
Orthogonal Collocation

Given: \( \frac{dz}{dt} = f(z, u, p), \) \( z(0)=\text{given} \)

Approximate \( z \) and \( u \) by Lagrange interpolation polynomials (order \( K+1 \) and \( K \), respectively) with interpolation points, \( t_k \)

\[
\begin{align*}
    z_{K+1}(t) &= \sum_{j=0}^{K} z_j \ell_j(t), \ell_j(t) = \prod_{\substack{j=0 \atop j \neq k}}^{K} \frac{t-t_j}{t_k-t_j} \implies z_{K+1}(t_k) = z_k \\
    u_K(t) &= \sum_{j=0}^{K} u_j \ell_j(t), \ell_j(t) = \prod_{\substack{j=0 \atop j \neq k}}^{K} \frac{t-t_j}{t_k-t_j} \implies u_K(t_k) = u_k
\end{align*}
\]

Substitute \( z_{K+1} \) and \( u_K \) into ODE and apply equations at \( t_k \).

\[
    r(t_k) = \sum_{j=0}^{K} z_j \ell_j(t_k) - f(z_k, u_k) = 0, \quad k = 1, \ldots K
\]
### Converted Optimal Control Problem Using Collocation

\[
\begin{aligned}
\text{Min} \quad & \phi(z(t)) \\
\text{s.t.} \quad & z^* = f(z, u, p), \ z(0) = z_0 \\
& g(z(t), u(t), p) \leq 0 \\
& h(z(t), u(t), p) = 0 \\
\end{aligned}
\]

to Nonlinear Program

\[
\begin{aligned}
\text{Min} \quad & \phi(z_f) \\
\sum & z_i^* (t_i) - f(z_i, u_i) = 0, \ z_0 = z(0) \\
& g(z_i, u_i) \leq 0 \\
& h(z_i, u_i) = 0 \\
\sum & z_i (t_i) - z_f = 0 \\
\end{aligned}
\]

How accurate is approximation

---

### Collocation on Finite Elements

- \( \frac{dz}{d\tau} = \frac{1}{h_i} \frac{dz}{d\tau} \)

- \( \frac{dz}{d\tau} = h_i f(z, u) \)

- \( t = \sum_{i=1}^{N-1} \frac{h_i + k \tau}{\tau} \in [0, 1) \)

- \( y(t) = \sum_{q=1}^{N} q_i(t) y_q \)

- \( u(t) = \sum_{q=1}^{N} q_i(t) u_q \)

- \( r(t_k) = 0 \quad k = 1, \ldots, N \)

- \( z(t) = \sum_{q=0}^{K} f_i(t) z_q \)

- \( q = 1, \ldots, \infty \)

- \( t_0, \ldots, t_f \)
Nonlinear Programming Problem

$$\min \; \psi(z_f)$$

s.t. \hspace{1cm} \sum_{j=0}^{K} (z_{ij} \dot{\ell}_j(z_k)) - h_j f(z_{ik}, u_{ik}, p) = 0$$

$$g(z_{ik}, y_{ik}, u_{ik}, p) = 0$$

$$\sum_{j=0}^{K} (z_{ij} \dot{\ell}_j(1)) - z_{io} = 0, \; i = 2,...,NE$$

$$\sum_{j=0}^{K} (z_{NE,j} \dot{\ell}_j(1)) - z_f = 0, \; z_{y0} = z(0)$$

$$z_{ij}' \leq z_{ij} \leq z_{ij}''$$

$$y_{i,j}' \leq y_{i,j} \leq y_{i,j}''$$

$$u_{ij}' \leq u_{ij} \leq u_{ij}''$$

$$p' \leq p \leq p''$$

Finite elements, $h_i$, can also be variable to determine break points for $u(t)$.

Add $h_2 \geq h_i \geq 0, \; \sum h_i = t_f$

Can add constraints $g(h, z, u) \leq \varepsilon$ for approximation error

Theoretical Properties of Simultaneous Method

A. Stability and Accuracy of Orthogonal Collocation

• Equivalent to performing a fully implicit Runge-Kutta integration of DAE models at Gaussian (Radau) points

• 2K order (2K-1) method which uses K collocation points

• Algebraically stable (i.e., possesses A, B, AN and BN stability)

B. Analysis of the Optimality Conditions (Kameswaran, B., 2007)

• An equivalence has been established between the KKT conditions of NLP and the variational necessary conditions

• Rates of convergence have been established for the NLP method
Example: Batch reactor - temperature profile

Maximize yield of B after one hour's operation by manipulating a transformed temperature, u(t).

\[ \Rightarrow \min -z_B(1.0) \]
\[ \text{s.t.} \]
\[ z_A' = -(u+u^2/2) z_A \]
\[ z_B' = u z_A \]
\[ z_A(0) = 1 \]
\[ z_B(0) = 0 \]
\[ 0 \leq u(t) \leq 5 \]

Optimality conditions:

\[ H = -\lambda_A (u+u^2/2) z_A + \lambda_B u z_A \]
\[ \partial H/\partial u = \lambda_A (1+u) z_A + \lambda_B z_A \]
\[ \lambda_A' = \lambda_A (u+u^2/2) - \lambda_B u, \quad \lambda_A(1.0) = 0 \]
\[ \lambda_B' = 0, \quad \lambda_B(1.0) = -1 \]

Optimal Profile, u(t)

\[
\begin{array}{c|c|c|c|c|c}
0 & 0.2 & 0.4 & 0.6 & 0.8 & 1.0 \\
\hline
0 & 2 & 4 & 6 & & \\
\end{array}
\]

Results:
Piecewise Linear Approximation with Variable Time Elements
Optimum B/A: 0.5726
Equivalent # of ODE solutions: 32
Batch Reactor Optimal Temperature Program
Indirect Approach

Results:
Control Vector Iteration with Conjugate Gradients
Optimum (B/A): \(0.5732\)
Equivalent # of ODE solutions: 58

Results of Optimal Temperature Program
Batch Reactor (Revisited)

- **Results** - NLP with Orthogonal Collocation
  - Optimum B/A: \(0.5728\)
  - # of ODE Solutions: 0.7 (Equivalent)
Dynamic Optimization Engines

Evolution of NLP Solvers:

→ for dynamic optimization, control and estimation

SQP

E.g., **NPSOL** and Sequential Dynamic Optimization - over 100 variables and constraints

E.g., **SNOPT** and Multiple Shooting - over 100 d.f.s but over $10^5$ variables and constraints
Dynamic Optimization Engines

Evolution of NLP Solvers:

\[ \text{for dynamic optimization, control and estimation} \]

\[ \text{SQP} \rightarrow \text{rSQP} \rightarrow \text{Full-space Barrier} \]

E.g., \textit{IPOPT} - Simultaneous dynamic optimization over 1,000,000 variables and constraints

Object Oriented Codes tailored to structure, sparse linear algebra and computer architecture (e.g., IPOPT 3.3)

---

Barrier Methods for Large-Scale Nonlinear Programming

\[
\begin{align*}
\text{Original Formulation} & \quad \min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad c(x) = 0, \quad a \leq x \leq b, \quad x \geq 0 \\
\text{Barrier Approach} & \quad \min_{x \in \mathbb{R}^n} \varphi_{\mu}(x) = f(x) - \mu \sum_{i=1}^{n} \ln x_i \\
& \quad \text{s.t.} \quad c(x) = 0 \\
\Rightarrow & \text{As } \mu \to 0, \quad x^*(\mu) \to x^* \quad \text{Fiacco and McCormick (1968)}
\end{align*}
\]
Solution of the Barrier Problem

⇒Newton Directions (KKT System)

\[
\nabla f(x) + A(x)\lambda - v = 0
\]
\[
Xv - \mu e = 0
\]
\[
e^T = [1, 1, 1, \ldots] 
\]
\[
X = \text{diag}(x)
\]

⇒Solve

\[
\begin{bmatrix}
W & A & -I \\
A^T & 0 & 0 \\
V & 0 & X
\end{bmatrix}
\begin{bmatrix}
d_x \\
d_\lambda \\
d_v
\end{bmatrix}
= 
\begin{bmatrix}
\nabla f + A\lambda - v \\
c \\
Xv - \mu e
\end{bmatrix}
\]

IPOPT Code – www.coin-or.org

⇒Reducing the System

\[
d_v = \mu X^{-1}e - v - X^{-1}V d_x
\]
IPOPT Algorithm – Features

Line Search Strategies for Globalization
- $l_2$ exact penalty merit function
- augmented Lagrangian merit function
- Filter method (adapted and extended from Fletcher and Leyffer)

Hessian Calculation
- BFGS (full/LM and reduced space)
- SR1 (full/LM and reduced space)
- Exact full Hessian (direct)
- Exact reduced Hessian (direct)
- Preconditioned CG

Algorithmic Properties
Globally, superlinearly convergent (Wächter and B., 2005)

Easily tailored to different problem structures

Freely Available
CPL License and COIN-OR distribution:
http://www.coin-or.org

IPOPT 3.x recently rewritten in C++

Solved on thousands of test problems and applications

Comparison of NLP Solvers: Data Reconciliation
(Poku, Kelly, B. (2004))

![Comparison of NLP Solvers](chart.png)
### Comparison of Computational Complexity

\((\alpha \in [2, 3], \beta \in [1, 2], n_w, n_u - \text{assume } N_m = O(N))\)

<table>
<thead>
<tr>
<th></th>
<th>Single Shooting</th>
<th>Multiple Shooting</th>
<th>Simultaneous</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAE Integration</td>
<td>(n_w^\beta N)</td>
<td>(n_w^\beta N)</td>
<td>---</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>((n_u N)(n_u N)) ((n_u N)(n_u + n_w)) (N(n_u + n_w))</td>
<td>((n_u N)(n_u N)) ((n_u N)(n_u + n_w)^2) (N(n_u + n_w))</td>
<td>---</td>
</tr>
<tr>
<td>Exact Hessian</td>
<td>((n_w N)(n_u N)^2)</td>
<td>((n_w N)(n_u N)^2) (N(n_u + n_w))</td>
<td>---</td>
</tr>
<tr>
<td>NLP Decomposition</td>
<td>---</td>
<td>(n_w^3 N)</td>
<td>---</td>
</tr>
<tr>
<td>Step Determination</td>
<td>((n_u N)^\alpha)</td>
<td>((n_u N)^\alpha) (((n_u + n_w)N)^\beta)</td>
<td>---</td>
</tr>
<tr>
<td>Backsolve</td>
<td>---</td>
<td>---</td>
<td>(((n_u + n_w)N)^\beta)</td>
</tr>
</tbody>
</table>

\[O((n_u N)^\alpha + N^2 n_w n_u + N^3 n_w n_u^2)\]

\[O((n_u N)^\alpha + N n_w^3)\]

\[O((n_u + n_w)N)^\beta\]

---

### Simultaneous DAE Optimization

**Case Studies**

- Reactor - Based Flowsheets
- Fed-Batch Penicillin Fermenter
- Temperature Profiles for Batch Reactors
- Parameter Estimation of Batch Data
- Synthesis of Reactor Networks
- Batch Crystallization Temperature Profiles
- Ramping for Continuous Columns
- Reflux Profiles for Batch Distillation and Column Design
- Air Traffic Conflict Resolution
- Satellite Trajectories in Astronautics
- Batch Process Integration
- Source Detection for Municipal Water Networks
- **Optimization of Simulated Moving Beds**
- Grade Transition of Polymerization Processes
- Parameter Estimation of Tubular Reactors
- Nonlinear MPC
Production of High Impact Polystyrene (HIPS)
Startup and Transition Policies (Flores et al., 2005a)

<table>
<thead>
<tr>
<th>Monomer, Transfer/Term. agents</th>
<th>Catalyst</th>
<th>Polymer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coolant</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Phase Diagram of Steady States

Transitions considered among all steady state pairs

System State
- Upper Steady-State
- Medium Steady-State
- Lower Steady-State

Bifurcation Parameter

Initiation reactions
- Thermal
  - \[ 3M + \text{Initiation} \rightarrow 2 \text{Polymer} + \text{Coolant} \]
- Chemical
  - \[ C + \text{Initiation} \rightarrow 2 \text{Polymer} \]
  - \[ R + M \rightarrow \text{Polymer} \]
  - \[ R + R \rightarrow \text{Polymer} \]
  - \[ R + \text{Monomer} \rightarrow \text{Polymer} \]
  - \[ R + \text{Transfer/Term. agents} \rightarrow \text{Polymer} \]

Propagation reactions
- \[ \text{Polymer} + \text{Monomer} \rightarrow \text{Polymer} \]
- \[ \text{Polymer} + \text{Transfer/Term. agents} \rightarrow \text{Polymer} \]
- \[ \text{Polymer} + \text{Coolant} \rightarrow \text{Polymer} \]

Definite termination reactions
- Homopolymer
  - \[ \text{Polymer} + \text{Polymer} \rightarrow \text{Polymer} \]
  - \[ \text{Polymer} + \text{Polymer} \rightarrow \text{Polymer} \]

Grafting
- \[ \text{Polymer} + \text{Polymer} \rightarrow \text{Polymer} \]
- \[ \text{Polymer} + \text{Polymer} \rightarrow \text{Polymer} \]

Crosslinking
- \[ \text{Polymer} + \text{Polymer} \rightarrow \text{Polymer} \]
- \[ \text{Polymer} + \text{Polymer} \rightarrow \text{Polymer} \]

Transfer reactions
- Monomer
  - \[ \text{Polymer} + \text{Monomer} \rightarrow \text{Polymer} \]
  - \[ \text{Polymer} + \text{Monomer} \rightarrow \text{Polymer} \]

Grafting sites
- \[ \text{Polymer} + \text{Polymer} \rightarrow \text{Polymer} \]
- \[ \text{Polymer} + \text{Polymer} \rightarrow \text{Polymer} \]
Phase Diagram of Steady States

Transitions considered among all steady state pairs

Startup to Unstable Steady State

- 926 variables
- 476 constraints
- 36 iters. / 0.95 CPU s (P4)
HIPS Process Plant (Flores et al., 2005b)

- Many grade transitions considered with stable/unstable pairs
- 1-6 CPU min (P4) with IPOPT
- Study shows benefit for sequence of grade changes to achieve wide range of grade transitions.

Simulated Moving Bed Optimization (Kawajiri, B., 2005-2007)

- Direction of liquid flow and valve switching
- Feed to Raffinate
- Extract to Desorbent
Simulated Moving Bed Optimization
(Kawajiri, B., 2005-2007)

Direction of liquid flow and valve switching
Repeat exactly the same operation (Symmetric)

Operating parameters:
4 Zone velocities + Step time

Zone 1
Zone 2
Zone 3
Zone 4
Formulation of Optimization Problem

Zone velocities

\[ \max_{\mathbf{u}(t)} \int_0^{t_{\text{step}}} \mathbf{u}(t) \, dt \]

(Maximize average feed velocity)

Product requirements

\[ \sum \frac{u_B(t) C_{B,i}(t)}{\bar{u}_B} \geq \text{Pur}_{\text{min}} \]

\[ \sum \frac{u_H(t) C_{H,i}(t)}{\bar{u}_H} \geq \text{Rec}_{\text{min}} \]

Bounds on liquid velocities

\[ u_l \leq \mathbf{u}(t) \leq u_u \quad m = I, II, III, IV \]

SMB model

\[ \frac{\partial C_i(x, t)}{\partial t} + \frac{\partial}{\partial x} \left( \mathbf{u}(x, t) C_i(x, t) \right) = 0 \]

CSS constraint

\[ C_i(x, t) = C_i(x, t + t_{\text{step}}) \]

Treatment of PDEs: Simultaneous Approach

(Orthogonal Collocation on Finite Elements)

\[ \min_{\mathbf{z}} \phi \left( \mathbf{z}(x, t) \right) \]

subject to:

\[ f_{i,j,k} \left( p, \frac{\Delta z}{\Delta x}, \frac{\partial z}{\partial t}, \frac{\partial^2 z}{\partial x^2} \right) = 0, \quad \frac{\Delta z}{\Delta x} \bigg|_{x=k \Delta x} = \frac{x_{k+1} - x_k}{h_x} \frac{z_{k+2} - z_k}{h_x} \]

\[ g \left( p, \mathbf{z}(x, t) \right) \leq 0 \]

\[ h \left( p, \mathbf{z}(x, t) \right) = 0 \]

PDE → Algebraic equations
Comparison of two approaches

<table>
<thead>
<tr>
<th></th>
<th># of variables</th>
<th>CPU Time*</th>
<th># of iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential Approach</td>
<td>644</td>
<td>111.8 min</td>
<td>49</td>
</tr>
<tr>
<td>Simultaneous Approach</td>
<td>33999</td>
<td>1.53 min</td>
<td>47</td>
</tr>
</tbody>
</table>

*(On Pentium IV 2.8GHz)

Sequential and Simultaneous methods find same optimal solution

- Initial feed velocity: 0.01 m/h
- Optimal feed velocity: 0.52 m/h

Nonstandard SMB: Addressed by Extended Superstructure NLP

- **Standard SMB**

- **VARICOL** (Asynchronous switching)

- **Three Zone** (Circulation loop is cut open)
Optimal Operating Scheme:
Result of Superstructure Optimization

CPU Time for optimization: 9.03 min
34098 variables, 34013 equations
*on Xeon 3.2 GHz

Large-Scale Parameter Estimation
Polymerization Reactor (Zavala, B., 2006)
Large-Scale Parameter Estimation

**Material & Energy**
\[
F_j \left[ \frac{dy_j(z)}{dz}, y_j(z), w_j(z), z, \pi_j, \Pi \right] = 0
\]

**Physical Properties**
\[
G_j \left[ y_j(z), w_j(z), z, \pi_j, \Pi \right] = 0
\]
\[
y_j(0) = \phi(y_{j-1}(z_{L_{j-1}}), F_{j-1})
\]
\[j \in \{1..NZ\}\]

**Zone Transitions**
- 500 ODEs
- 1000 AEs

\[ \times \text{ Stiffness} + \text{ Highly Nonlinear} + \text{ Parametric Sensitivity} + \text{ Algebraic Coupling} \]

---

Large-Scale Parameter Estimation

**Complex Kinetic Mechanisms**

- **Initiator decomposition**
  \[ I \xrightarrow{k_{i0}} 2R \quad i = 1,N_I \]

- **Chain initiation**
  \[ R + M_1 \xrightarrow{k_{p1}} P_{1,0} \]
  \[ R + M_2 \xrightarrow{k_{q1}} Q_{1,0} \]

- **Chain Propagation**
  \[ P_{1,0} + M_1 \xrightarrow{k_{p1}} P_{2,0} \]
  \[ P_{1,0} + M_2 \xrightarrow{k_{q1}} Q_{2,0} \]
  \[ Q_{1,0} + M_1 \xrightarrow{k_{q1}} P_{2,0} \]
  \[ Q_{1,0} + M_2 \xrightarrow{k_{q1}} Q_{2,0} \]

- **Chain Transfer to Monomer**
  \[ P_{1,0} + M_1 \xrightarrow{k_{p1}} P_{1,0} + M_1 \]
  \[ P_{1,0} + M_2 \xrightarrow{k_{q1}} Q_{1,0} + M_1 \]
  \[ Q_{1,0} + M_1 \xrightarrow{k_{q1}} P_{1,0} + M_1 \]
  \[ Q_{1,0} + M_2 \xrightarrow{k_{q1}} Q_{1,0} + M_1 \]

- **Chain Transfer to Solvent**
  \[ P_{1,0} + S \xrightarrow{k_{p1}} P_{1,0} + M_1 \]
  \[ Q_{1,0} + S \xrightarrow{k_{q1}} Q_{1,0} + M_1 \]

- **Chains to Polymer**
  \[ P_{1,0} + M_3 \xrightarrow{k_{p1}} P_{2,0} + M_3 \]
  \[ P_{1,0} + M_4 \xrightarrow{k_{q1}} Q_{1,0} + M_3 \]
  \[ Q_{1,0} + M_1 \xrightarrow{k_{q1}} P_{2,0} + M_3 \]
  \[ Q_{1,0} + M_2 \xrightarrow{k_{q1}} Q_{1,0} + M_3 \]

- **Termination by Combination**
  \[ P_{1,0} + P_{1,0} \xrightarrow{k_{p1}} M_{1+1,0} \]
  \[ P_{1,0} + Q_{1,0} \xrightarrow{k_{p1}} M_{1+1,0} \]
  \[ Q_{1,0} + Q_{1,0} \xrightarrow{k_{p1}} M_{1+1,0} \]

- **Termination by Disproportionation**
  \[ P_{1,0} + P_{1,0} \xrightarrow{k_{p1}} M_{1,0} + M_{1,0} \]
  \[ P_{1,0} + Q_{1,0} \xrightarrow{k_{p1}} M_{1,0} + M_{1,0} \]
  \[ Q_{1,0} + Q_{1,0} \xrightarrow{k_{p1}} M_{1,0} + M_{1,0} \]

- **Backbiting**
  \[ P_{1,0} \xrightarrow{k_{p1}} P_{1,0} \]

- **β-elimination**
  \[ P_{1,0} \xrightarrow{k_{p1}} M_{2,0} + P_{1,0} \]

The rate coefficient \( k \) is given by
\[ k = k_0 \exp \left( -\frac{E_a}{RT} \right) \]

\[ \sim 35 \text{ Elementary Reactions} \]
\[ \sim 100 \text{ Kinetic Parameters} \]
Large-Scale Parameter Estimation

- Parameter Estimation for Industrial Applications
  - Use Rigorous Model to Match Plant Data Directly
  - Start with Standard Least-Squares Formulation

\[
\begin{align*}
\min \quad & \sum_{k=1}^{Nk} \sum_{j=1}^{NZ} \sum_{i=1}^{NM(j)} (y_{k,j}(z_i) - y_{k,j,i}^M)^T \mathbf{V}_y^{-1} (y_{k,j}(z_i) - y_{k,j,i}^M) \\
+ & \sum_{k=1}^{Nk} (w_k - w_k^M)^T \mathbf{V}_w^{-1} (w_k - w_k^M) \\
n & \mathbf{F}_{k,j} \left[ \frac{d\mathbf{y}_{k,j}(z)}{dz}, \mathbf{y}_{k,j}(z), \mathbf{w}_{k,j}(z), z, \mathbf{\pi}_{k,j}, \mathbf{\Pi} \right] = 0 \\
& \mathbf{G}_{k,j} \left[ \mathbf{y}_{k,j}(z), \mathbf{w}_{k,j}(z), z, \mathbf{\pi}_{k,j}, \mathbf{\Pi} \right] = 0 \\
y_{k,j}(0) = \phi(y_{k,j-1}(z_{t_{k,j-1}}), \mathbf{F}_{k,j}) \\
j \in \{1...Nk\}, \quad k \in \{1...NS\}
\end{align*}
\]

- Least-Squares
- Rigorous Reactor Model

- Special Case of Multi-Stage Dynamic Optimization Problem
  - Solve using Simultaneous Collocation-Based Approach

<table>
<thead>
<tr>
<th>1 data set</th>
<th>6 data sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>500 ODEs</td>
<td>3000 ODEs</td>
</tr>
<tr>
<td>1000 AEs</td>
<td>6000 AEs</td>
</tr>
</tbody>
</table>

**Multi-Zone Tubular Reactor – Quasi Steady-State**

- Data Sets: Operating Conditions and Properties for Different Grades
- Match: Temperature Profiles and Product Properties
  - On-line Adjusting Parameters: Track Evolution of Disturbances
  - Kinetic Parameters: Development and Discrimination among Rigorous Models

**Results**

- Single Data Set (On-line Adjusting Parameters)

<table>
<thead>
<tr>
<th>Grade</th>
<th>Constraints</th>
<th>Parameters</th>
<th>LB</th>
<th>UB</th>
<th>Iterations</th>
<th>CPUs</th>
<th>NZJ</th>
<th>NZH</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>11955</td>
<td>32</td>
<td>374</td>
<td>361</td>
<td>11</td>
<td>17.03</td>
<td>166425</td>
<td>87954</td>
</tr>
<tr>
<td>B</td>
<td>11283</td>
<td>32</td>
<td>374</td>
<td>361</td>
<td>8</td>
<td>10.06</td>
<td>138666</td>
<td>76890</td>
</tr>
</tbody>
</table>

- Multiple Data Sets (On-line Adjusting Parameters + Kinetics)

<table>
<thead>
<tr>
<th>Data Sets</th>
<th>Constraints</th>
<th>Parameters</th>
<th>LB</th>
<th>UB</th>
<th>Iterations</th>
<th>CPUs</th>
<th>NZJ</th>
<th>NZH</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>33900</td>
<td>121</td>
<td>1240</td>
<td>1207</td>
<td>68</td>
<td>451.51</td>
<td>520275</td>
<td>552736</td>
</tr>
<tr>
<td>6</td>
<td>68421</td>
<td>217</td>
<td>2467</td>
<td>2389</td>
<td>58</td>
<td>900.21</td>
<td>1058412</td>
<td>1119258</td>
</tr>
</tbody>
</table>

Bottleneck (Memory Requirements)

Factorization Step

\[
\begin{bmatrix}
W(x_k, \lambda_k) \\
A(x_k) - F \\
V_k \\
X_k
\end{bmatrix} \begin{bmatrix}
\Delta x \\
\Delta \lambda \\
\Delta \nu
\end{bmatrix} = \begin{bmatrix}
\nabla f(x_k) + A(x_k)\lambda_k - v_k \\
(-1) \tilde{c}(x_k) \\
X_k V_k e - u_k
\end{bmatrix}
\]
Large-Scale Parameter Estimation

**Improved Match of Reactor Temperatures Profile**

![Graph showing improved match of reactor temperatures profile for Grades A and B.]

Industrial Case Study

- **Results - Reactor Overall Monomer Conversion**
  - (up to 20 Different Grades)

- **Graph showing predicted vs. plant conversion.**
  - Average Conversion Deviation:
    - Base Model: 12.1%
    - New Model: 2.5%
    - EVM Results: 0.12%
Parameter Estimation in Parallel Architectures

Exploit Structure of KKT Matrix – Laird, B. 2006

\[
\min_{n, n_1} \sum_{j=1}^{N} \sum_{k=1}^{N} \left( \frac{1}{2} (y_{kj}(t_k) - y_{kj}(t_k))^T v_j^{-1} (y_{kj}(t_k) - y_{kj}(t_k)) \right) + \sum_{i=1}^{N} \left( w_{i,k} - w_{i,k} \right)^T v_i^{-1} \left( w_{i,k} - w_{i,k} \right)
\]

\[
F_k \left[ \begin{array}{cccc}
\frac{\partial \phi_j(t_k)}{\partial x_j(t_k)} & \frac{\partial \phi_j(t_k)}{\partial x_j(t_k)} & \ldots & \frac{\partial \phi_j(t_k)}{\partial x_j(t_k)} \\
\frac{\partial \phi_j(t_k)}{\partial x_j(t_k)} & \frac{\partial \phi_j(t_k)}{\partial x_j(t_k)} & \ldots & \frac{\partial \phi_j(t_k)}{\partial x_j(t_k)} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial \phi_j(t_k)}{\partial x_j(t_k)} & \frac{\partial \phi_j(t_k)}{\partial x_j(t_k)} & \ldots & \frac{\partial \phi_j(t_k)}{\partial x_j(t_k)}
\end{array} \right] = 0
\]

Direct Factorization MA27
Memory Bottlenecks
Factorization Time Scales
Superlinearly with Data sets

Block-bordered Diagonal Structure
Coarse-Grained Parallelization using
Schur Complement Decomposition

IPOPT 3.x architecture supports tailored structured decompositions

Parameter Estimation in Parallel Architectures

(Zavala, Laird, B., 2007)

Computational Results – LDPE Reactor EVM Problem
Supply Chain, Planning and Scheduling
- Large LP and MILP models
- Many Discrete Decisions
- Few Nonlinearities
- Essential link needed to process models
- \textit{Decisions need to be feasible at lower levels}

Decision Pyramid for Process Operations
Real-time Optimization and Advanced Process Control
- Fewer discrete decisions
- Many nonlinearities
- Frequent, “on-line” time-critical solutions
- Higher level decisions must be feasible
- Performance communicated for higher level decisions
Dynamic Real-time Optimization

Integrate On-line Optimization/Control with Off-line Planning
- Consistent, first-principle models
- Consistent, long-range, multi-stage planning
- Increase in computational complexity
- Time-critical calculations

Applications
- Batch processes
- Grade transitions
- Cyclic reactors (cooking, regeneration…)
- Cyclic processes (PSA, SMB…)

Continuous processes are never in steady state:
- Feed changes
- Nonstandard operations
- Optimal disturbance rejections

Simulation environments and first principle dynamic models are widely used for off-line studies

Can these results be implemented directly on-line for large-scale systems?

Nonlinear Model Predictive Control (NMPC)

NMPC Estimation and Control

Why NMPC?
- Track a profile
- Severe nonlinear dynamics (e.g., sign changes in gains)
- Operate process over wide range (e.g., startup and shutdown)

NMPC Subproblem

\[
\begin{align*}
\min_x & \quad \sum ||y(t) - y^p||^2 + \sum ||u(t^k) - u(t^{k-1})||^2 \\
\text{s.t.} & \quad z'(t) = F(z(t), y(t), u(t), t) \\
& \quad 0 = G(z(t), y(t), u(t), t) \\
& \quad z(t) = z^m \\
& \quad \text{Bound} \quad \text{Constraint} \quad s \quad \text{Other} \quad \text{Constraint} \quad s
\end{align*}
\]
Nonlinear Model Predictive Control (NMPC)

\[ \min \sum ||y(t) - y_{sp}||^2_z + \sum ||u(t) - u(t-1)||^2_u, \]

s.t.
\[ z(t) = F(z(t), y(t), u(t), t) \]
\[ 0 = G(z(t), y(t), u(t), t) \]
\[ z(t) = z_{\text{min}} \]

Bound Constraint s
Other Constraint s

Tennessee Eastman Process
(Downs and Vogel, 1993)

Unstable Reactor
11 Controls; Product, Purge streams
Model extended with energy balances
Tennessee Eastman NMPC Model
(Jockenhövel, Wächter, B., 2003)

**Method of Full Discretization of State and Control Variables**
Large-scale Sparse block-diagonal NLP

**Case Study:**
Change Reactor pressure by 60 kPa

**Control profiles**
- All profiles return to their base case values
- Same production rate
- Same product quality
- Same control profile
- Lower pressure – leads to larger gas phase (reactor) volume
- Less compressor load
Case Study: Change Reactor Pressure by 60 kPa

Optimization with IPOPT
1000 Optimization Cycles
5-7 CPU seconds
11-14 Iterations

Optimization with SNOPT
Often failed due to poor conditioning
Could not be solved within sampling times
> 100 Iterations

Limitations to NMPC Implementation

*Issues:* time-critical, more complex models, fast NLP solvers.

*Computational delay* – between receipt of process measurement and injection of control, determined by cost of dynamic optimization

Leads to loss of *performance* and *stability* (see Findeisen and Allgöwer, 2004; Santos et al., 2001)

As larger NLPs are considered for NMPC, can computational delay be overcome?
Avoid computational delay due to on-line optimization?

Real-time Iteration
- preparation, feedback response and transition stages
- solve perturbed (linearized) problem on-line
  - Li, de Oliveira, Santos, B. (1990+)
  - Diehl, Findeisen, Bock, Allgöwer et al. (2000+)
  - > two orders of magnitude reduction in on-line computation
- solve complete NLP in background (‘between’ sampling times as part of preparation and transition stages

Based on NLP sensitivity for dynamic systems
- Extended to Simultaneous Collocation approach – Zavala et al. (2007)
- Develop Advanced Step NMPC
- Related to MPC with linearization constantly updated one step behind

Nonlinear Model Predictive Control – Parametric Problem (Zavala, Laird, B.)

\[
P(\bar{z}, N) \quad \min_{z_{k}} \quad J(z(k), N) = F(z_{k}, y_{a}) + \sum_{i=1}^{N-1} \psi(z_{a}, y_{a}) \\
\quad \text{s.t.:} \\
\quad z_{k+1} = f(z_{k}, y_{a}), \quad |k = k, \ldots, k + N - 1 \\
\quad z_{a} = z(k) = p_{0} \\
\quad z_{a} \in \mathbb{R}, y_{a} \in X, z_{a+1} \in X_{f}, y_{a} \in U.
\]
Nonlinear Model Predictive Control – Parametric Problem (Zavala, Laird, B.)

\[ P(x(k), N) \quad \min_{\eta_k} J(x(k), N) = F(\eta_k, \eta_k) + \sum_{i=k}^{L(N-1)} \psi(\eta_i, \eta_i) \]

s.t.:
\[ x_{i+1} = f(x_i, \eta_i), \quad i = k, ..., k + N - 1 \]
\[ \eta_i \in \mathbb{R}, \eta_{k+1} \in \mathbb{R}^n, \eta_k \in \mathbb{U}. \]

\[ P(x(k+1), N) \quad \min_{\eta_{k+1}} J(x(k+1), N) = F(\eta_{k+1} + \eta_{k+1}) + \sum_{i=k+1}^{L(N-1)} \psi(\eta_i, \eta_i) \]

s.t.:
\[ x_{i+1} = f(x_i, \eta_i), \quad i = k + 1, ..., k + N - 1 \]
\[ \eta_{k+1} = x(k+1) + \eta_{k+1} \]
\[ \eta_{k+1} \in \mathbb{R}, \eta_{k+1} \in \mathbb{R}^n, \eta_{k+1} \in \mathbb{U}. \]

NLP Sensitivity

Parametric Programming
\[ \min \quad f(x, p) \]
\[ \text{s.t.} \quad c(x, p) = 0 \]
\[ x \geq 0 \]

Solution Triplet
\[ s^*(p)^T = [x^T, s^T, \nu^T] \]

Optimality Conditions \[ P(p) \]
\[ \nabla_x f(x, p) + \nabla_x c(x, p) \lambda - \nu = 0 \]
\[ c(x, p) = 0 \]
\[ \lambda V_c = 0 \]

NLP Sensitivity \[ \rightarrow \] Rely upon existence and differentiability of \( s^*(p) \)
\[ \rightarrow \text{Main Idea: Obtain } \frac{\partial s^*}{\partial p} \text{ and find } s^*(p_1) \text{ by Taylor Series Expansion} \]

\[ s^*(p_1) \approx s^*(p_0) + \frac{\partial s^*}{\partial p} \bigg|_{p_0} (p_1 - p_0) \]
NLP Sensitivity

Optimality Conditions of $P(p)$

$$\begin{align*}
\nabla_x \mathcal{L} &= \nabla_x f(x, p) + \nabla_x c(x, p) \lambda - \nu = 0 \\
c(x, p) &= 0 \\
XV &= 0
\end{align*}$$

$$Q(s, p) = 0$$

Apply Implicit Function Theorem to

$$Q(s, p) = 0 \quad \text{around} \quad (p_0, s^*(p_0))$$

$$\frac{\partial Q(s^*(p_0), p_0)}{\partial s} \frac{\partial s}{\partial p} |_{p_0} + \frac{\partial Q(s^*(p_0), p_0)}{\partial p} = 0$$

KKT Matrix IPOPT

$$\begin{bmatrix}
W(x_k, \lambda_k) & A(x_k) & -I \\
A(x_k)^T & 0 & 0 \\
V_k & 0 & X_k
\end{bmatrix}
\begin{bmatrix}
\frac{\partial z}{\partial x} \\
\frac{\partial z}{\partial p} \\
\frac{\partial \mathcal{L}(s^*(p_0))}{\partial p}
\end{bmatrix}
+\begin{bmatrix}
\nabla_x c(x^*(p_0)) \\
0 \\
0
\end{bmatrix} = 0$$

Key Concept – Relate to Previous Horizon

Solutions to both problems are equivalent in nominal case (ideal plant model, no disturbances)
Advanced Step NMPC
Combine advanced step with sensitivity to solve NLP in background (between steps) – not on-line

\[ P(z) \]

\[ z_0 \]

\[ z_1 \]

\[ z_2 \]

\[ z_{N-1} \]

\[ z_N \]

\[ u_U \]

\[ u_L \]

\[ t_0 \]

\[ t_1 \]

\[ t_2 \]

\[ t_{N-1} \]

\[ t_N \]

\[ \min_{\delta} J(z(k), N) = F(z_{k+N}(k)) + \sum_{j=1}^{N-1} \psi(z_{k+j}, \nu_k) \]

s. t.: \[ z_{k+j} = f(z(k), \nu_k), \]

\[ z_{k+j} = f(z_{k+j}, \nu_{k+j}), \quad l = k+1, \ldots, k+N-1 \]

\[ z_{k+j} = 0 \]

\[ z_{k+j} \in X, \quad z_{k+j} \in X, \quad \nu_{k+j} \in U. \]

Solve \( P(z) \) in background (between \( t_0 \) and \( t_f \))

\[ \Delta = \begin{bmatrix} \Delta x \\ \Delta z \end{bmatrix} \]

\[ K = \begin{bmatrix} W \quad A_i \\ A_i^T \quad 0 \end{bmatrix} \]

\[ A_i = \begin{bmatrix} 0 & 0 & \Delta A \end{bmatrix} \]

\[ V = \begin{bmatrix} 0 & X_k \end{bmatrix} \]

\[ K = \begin{bmatrix} K & E_0 \\ E_{1-k} & 0 \end{bmatrix} \]

\[ \Delta u = - \begin{bmatrix} 0 \\ \Delta z \end{bmatrix} \]

Sensitivity to updated problem to get \( (z_0, u_0) \)
**Advanced Step NMPC**

Combine advanced step with sensitivity to solve NLP in background (between steps) – not on-line

Solve $P(z_{t+1})$ in background (between $t_i$ and $t_f$)

Sensitivity to updated problem to get $(z_0, u_0)$

Solve $P(z_{t+1})$ in background with new $(z_0, u_0)$

**AS-NMPC Stability Analysis**

Nominal NMPC stability proof

- Nominal case – no noise: perfect model
- General formulation with local asymptotic controller for $t \to \infty$
- Advanced step controller satisfies same relations, has same input sequence
  \( \Rightarrow \) shares identical stability property

Plant:

\[
\begin{align*}
  x_{k+1} &= \mathbf{f}(x_k, u_k) + g(x_k, u_k, w_k) \\
  \| g(x_k, u_k, w_k) \| &\leq L_f \| x_k \| + \sigma(\| w_k \|)
\end{align*}
\]

Model:

\[
\begin{align*}
  z_{t+1} &= \mathbf{f}(z_t, u_t) \\
  z_0 &= x_k
\end{align*}
\]

Robust Stability Margins

- Analysis similar to Limon, Alamo, Camacho (2004), Magni and Scattolini (2005)
- Advanced step NMPC is ISS and tolerates some model mismatch
- ISS property (Jiang and Wang, 2001; Magni and Scattolini, 2005)
- Advanced step NMPC has smaller margin than Ideal NMPC, \( \Rightarrow \) but can be implemented without computational delay
CSTR NMPC Example (Hicks and Ray)

\[
\begin{align*}
\text{min} & \quad \sum_{i=1}^{k+N-1} Q_i (s_i^*(k)) + Q_d (s_i^*(k)) + R (r_i(k))^2 \\
\text{s.t.} & \quad z_{i+1,k} = \frac{1}{\theta} (1 - (z_i^*(k) + z_{i+1,k}^*)) - \theta \exp \left( - \frac{E_r}{z_i^*(k)} \right) (z_i^*(k) + z_{i+1,k}^*) \\
& \quad z_{i+1,k} = \frac{1}{\theta} (1 - (z_i^*(k) + z_{i+1,k}^*)) + \theta \exp \left( - \frac{E_r}{z_i^*(k)} \right) z_i^*(k) - \alpha \left( z_i^*(k) \right) (z_i^*(k) + z_{i+1,k}^*) - u_i \\
& \quad z_{i+1,k} = z^*(k), \quad z_{i+1,k} = z^*(k) \\
& \quad \lambda_{i+k} = 0, \quad \lambda_{i+k} = 0, \quad u_i^B \leq u_i \leq u_i^U.
\end{align*}
\]

- Maintain unstable setpoint
- Close to bound constraint
- Final time constraint for stability

Effects of:
- Computational Delay
- Measurement Noise
- Model Mismatch
- Advanced Step NMPC

CSTR NMPC Example – Nominal Case

- NMPC applied with \( N = 10, \tau = 0.5 \) sampling time
- Stable \((z = 0)\) and unstable \((z = 0.1)\) steady states
- \( u_2^* \) close to upper bound
- Computational delay = 0.5, leads to instabilities
CSTR NMPC Example – Model Mismatch

Advanced Step NMPC not as robust as ideal - suboptimal selection of $u(k)$

Better than Direct Variant – due to better active set preservation

CSTR Example: Mismatch + Noise

Advanced Step NMPC not as robust as ideal - suboptimal selection of $u(k)$

Better than Direct Variant – due to better active set preservation
Industrial Case Study – Grade Transition Control

Process Model: 289 ODEs, 100 AEs

\[
\min \int_{t_f}^{t_f+\Delta t} \left( uC(t) - uC_a(t) \right)^2 + \left( FC_a(t) - FC(t) \right)^2 + \left( F_{nu}(t) - F_{nu}^* \right)^2 \, dt \\
\text{subject to} \\
PDEs + ODEs \\
\quad \sigma(t_{i-1}) = \sigma(t_i) \\
\quad \tau_i \leq t \leq \tau_{i+1} \\
\quad u_i \leq u \leq u_i \\
\quad u_i \leq u \leq u_i \\
\quad \text{Simultaneous Collocation-Based Approach} \\
\quad \min \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \left( \frac{\partial^2}{\partial t^2} \right) \mathbf{x}(t_i) \\
\quad \text{s.t.} \\
\quad \sigma_i = \sigma_{i+1} + \Delta t \frac{\partial}{\partial t} \mathbf{x}(t_i) \\
\quad \mathbf{x}(t_i) = \mathbf{x}(t_{i-1}) \\
\quad \mathbf{x}(t_i) = \mathbf{x}(t_{i+1}) \\
\quad \mathbf{x}(t_0) = \mathbf{x}(t_f) \\
\quad \mathbf{x}(t_{N-1}) = \mathbf{x}(t_f) \\
\quad \mathbf{F}(t) = \mathbf{F}(t_i) \\
\quad \mathbf{F}(t_{N-1}) = \mathbf{F}(t_f) \\
\quad \mathbf{F}(t_{N-1}) = \mathbf{F}(t_f) \\
\quad 27,135 \text{ constraints, 9630 LB & UB} \\
\]

Off-line Solution with IPOPT

<table>
<thead>
<tr>
<th>Algorithmic Step</th>
<th>CPU(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Solution (10 iterations)</td>
<td>351.5</td>
</tr>
<tr>
<td>Single Factorization of KKT Matrix</td>
<td>33.9</td>
</tr>
<tr>
<td>Step Computation (single backsolve)</td>
<td>0.94</td>
</tr>
<tr>
<td>Rest of Steps</td>
<td>0.12</td>
</tr>
</tbody>
</table>

Feedback Every 6 min

NMPC Case Study

- Optimal Feedback Policy → (On-line Computation 351 CPU s)

Ideal NMPC controller - computational delay not considered

Time delays as disturbances in NMPC
NMPC Case Study

Optimal Policy vs. NLP Sensitivity - Shifted \( \rightarrow \) (On-line Computation 1.04 CPU s)

Very Fast Close-to-Optimal Feedback
Large-Scale Rigorous Models

Moving Horizon Estimation

Linear Systems, No inequalities \( \rightarrow \) Kalman Filter for State Estimation
Nonlinear Systems: \( \rightarrow \) Extended Kalman Filters in Practice

Moving Horizon Estimation:

- + directly captures nonlinear dynamics, statistical behavior
- - need to solve NLP on-line
  Solution Time \( \rightarrow \) Order of Minutes

Computational delay – between receipt of process measurement, estimation and injection of new control
Leads to loss of performance and stability
Moving Horizon Estimation

Linear Systems, No inequalities → Kalman Filter for State Estimation
Nonlinear Systems: → Extended Kalman Filters in Practice

Moving Horizon Estimation:
+ directly captures nonlinear dynamics, statistical behavior
- need to solve NLP on-line

Solution Time → Order of Minutes

Computational delay – between receipt of process measurement, estimation and injection of new control
Leads to loss of performance and stability

\[
\min_{x_0, w_k} \left( x_0 - \bar{x}_0 \right)^T \Pi_0^{-1} \left( x_0 - \bar{x}_0 \right) + \sum_{k=0}^{N} L_k(v_k, w_k) \\
\text{s.t.} \quad x_{k+1} = f_k(x_k, w_k) \\
\bar{y}_{k+1}^{r-N} = h_k(x_k) + v_k
\]

Computational Delay - MHE Impractical
NLPs are Parametric
Solve \( P(\ell) \) in Background
Fast Approximation to \( P(\ell + 1) \) On-line

\[
P(\ell) \left\{ \begin{array}{l}
\min_{x_0, w_k} \left( x_0 - \bar{x}_0^{\ell+1} \right)^T \Pi_0^{-1} \left( x_0 - \bar{x}_0^{\ell+1} \right) + \sum_{k=0}^{N} L_k(v_k, w_k) \\
\text{s.t.} \quad x_{k+1} = f_k(x_k, w_k) \\
\bar{y}_{k+1}^{r-N+1} = h_k(x_k) + v_k
\end{array} \right.
\]
Fast Moving Horizon Estimation

1) Solve Extended Problem Between $t_{\ell}$ and $t_{\ell+1}$

$$ \begin{align*} 
& \min_{x_0, w_0, u_{\ell+1}} \quad (x_N - \bar{y}_N^+)^T \mathbf{P}_N^{-1} (x_N - \bar{y}_N^+) + \sum_{k=0}^{N} l_k(v_k, w_k) + l_{N+1} (v_{N+1}, w_{N+1}) \\
& \text{s.t.} \\
& \quad x_{k+1} = f_{k}(x_k, w_k), \quad k = 0, \ldots, N-1 \\
& \quad \bar{y}_{k+1} = h_{k}(x_k) + v_k, \quad k = 0, \ldots, N \\
& \quad x_{N+1} = f_N(x_N, w_N) \\
& \quad y_{N+1} = h_{N+1}(x_{N+1}) + v_{N+1} \\
& \quad \bar{y}_{N+1} = \text{Dummy Measurement} = \text{Model Prediction at } t_{\ell+1} \\
\end{align*} $$

Re-use KKT Matrix Available At Solution of $\bar{P}(\ell)$

Analyze Terms due to Extended Horizon

$$ \begin{align*} 
\Delta \lambda_{N+1} &= 0 \\
R_{N+1}^{-1} \Delta v_{N+1} + \Delta \lambda_{N+1} &= 0 \\
\Delta y_{N+1} - \nabla_s h_{N+1} \Delta x_{N+1} - \Delta v_{N+1} &= 0 \\
\end{align*} $$

Find perturbation $\Delta \varphi$ that enforces $K \Delta s = 0$

$$ \begin{align*} 
\varphi_{\ell+1} &= \bar{y}_{\ell+1} \\
K &\text{ is already factorized} \\
\text{Solve as Schur complement problem} \\
\text{On-line Cost is a simple backsolve} \\
\end{align*} $$

2) At $t_{\ell+1}$ once we know $\bar{y}_{\ell+1}$

KKT System At Solution of $\bar{P}(\ell)$

$$ \begin{align*} 
\begin{bmatrix} 
W(z_k, \lambda_k) & A(z_k) \\
A(z_k)^T & 0 \\
V_k & 0 \\
0 & X_k \\
0 & \Delta x \\
\end{bmatrix} \begin{bmatrix} 
\Delta x \\
\Delta \lambda \\
\Delta \varphi \\
\end{bmatrix} &= 0 \\
\end{align*} $$

Relax Multiplier

Force Dummy to Measurement

$$ \begin{align*} 
\Delta \lambda_{N+1} + \Delta \varphi &= 0 \\
\Delta y_{N+1} = (\bar{y}_{\ell+1} - \bar{y}_{\ell+1}) \\
\end{align*} $$

Augmented KKT System

$$ \begin{align*} 
\begin{bmatrix} 
K & E_P \\
E_P^T & 0 \\
\end{bmatrix} \begin{bmatrix} 
\Delta s \\
\Delta \varphi \\
\end{bmatrix} &= \begin{bmatrix} 
0 \\
\bar{y}_{\ell+1} - \bar{y}_{\ell+1} \\
\end{bmatrix} \\
\end{align*} $$
MHE Case Study (Zavala, Laird, B.)

Single Measurement - Composition of Recycle Gas
Sampling Time ~ 6 min
Estimation Horizon N = 15

NLP Simultaneous Approach

\[
\begin{align*}
\min_{x_k, w_k} & \quad (x_0 - \tilde{x}_0)^T \Pi_0^{-1} (x_0 - \tilde{x}_0) + \sum_{k=0}^{N} l_k(y_k, w_k) \\
\text{s.t.} & \quad x_{k+1} = f_k(x_k, w_k) \\
& \quad y_{k+1} = h_k(x_k) + v_k
\end{align*}
\]

27,121 Constraints, 9330 Bounds
294 Degrees of Freedom

Algorithmic Step | CPUs | Computational Delay
Full Solution (6 iterations) | 202.64 | On-Line
Single Factorization of KKT Matrix | 33.77 | Calculation
Step Computation (single backsolve) | 0.9-1.0 |
Rest of Steps | 0.936 |

\( \kappa \) has correct Inertia at Solution – System Locally Observable

MHE Case Study

Simulated Control Profiles

Measurement Profiles Gaussian Noise 5% SD

On-line NLP requires up to 4 CPU minutes
Leads to Feedback Delay in Controller
MHE Case Study

Simulated Control Profiles

Measurement Profiles Gaussian Noise 5% SD

On-line Update 1 Second
Sensitivity Errors Negligible

Summary: Dynamic Optimization

Sequential Approaches – Use DAE Integrators
- Parameter Optimization
  - Gradients by: Direct (and Adjoint) Sensitivity Equations
- Optimal Control (Profile Optimization)
  - Variational Methods
  - NLP-Based Methods - Single and Multiple Shooting
- Require Repeated Solution of Model
- State Constraints are Difficult to Handle

Simultaneous Collocation Approach
- Discretize ODE’s using orthogonal collocation on finite elements
- Straightforward addition of state constraints.
- Deals with unstable systems
- Solve model only once
- Avoid difficulties at intermediate points

Large-Scale Extensions
- Exploit structure of DAE discretization through decomposition
- Large problems solved efficiently with IPOPT
Summary: On-line Extensions

RTO and MPC widely used for refineries, ethylene and, more recently, chemical plants

- Inconsistency in models → operating problems?

Off-line dynamic optimization is widely used

- Polymer processes (especially grade transitions)
- Batch processes
- Periodic processes

NMPC provides link for off-line and on-line optimization

- Stability and robustness properties
- Advanced step controller leads to very fast calculations
  - Analogous stability and robustness properties
  - On-line cost is negligible

Multi-stage planning and on-line switches

- Avoids conservative performance
- Update model with MHE
- Evolve from regulatory NMPC to Large-scale DRTO

Acknowledgements

Funding

- Department of Energy
- National Science Foundation
- Center for Advanced Process Decision-making (CMU)

Research Colleagues

- Antonio Flores-Tlacuahuac
- Tobias Jockenhövel
- Shivakumar Kameswaran
- Yoshiaki Kawajiri
- Carl Laird
- Yi-dong Lang
- Andreas Wächter
- Victor Zavala

http://dynopt.cheme.cmu.edu
References – Dynamic Optimization


Biegler Homepage: http://dynopt.cheme.cmu.edu/papers.htm


Biegler Homepage: http://dynopt.cheme.cmu.edu/papers.htm


References – Recent DRTO Case Studies


