Multiple Shooting with Collocation: An Efficient Approach to Dynamic Optimization, with Possibilities for Integration in OpenModelica

Ines Mynttinen, Jasem Tamimi, Pu Li

Simulation and Optimal Processes Group (SOP)
Institute of Automation and Systems Engineering
Ilmenau University of Technology
Contents

1. Motivation

2. Multiple Shooting

3. Collocation on Finite Elements

4. A Case Study

5. Classification and Implementation

6. Conclusions
Motivation: NMPC using multiple shooting with collocation

NMPC is superior to traditional control in many cases

Critical issues:
- Accuracy
- Problem size - computational effort
- Safety specifications – path constraints

Fast algorithm is needed
Nonlinear optimal control problem

\[
\min_{u(t), x(t), y(t), p} \left\{ J = \Phi(x_f, y_f, t_f, p) + \int_{t_0}^{t_f} L(x(t), y(t), u(t), p) \, dt \right\}
\]

s.t. \[
\begin{align*}
\dot{x}(t) &= f(x(t), y(t), u(t), t, p) \quad x(t_0) = x_0 \\
0 &= g(x(t), y(t), u(t), t, p) \quad t \in [t_0, t_f] \\
0 &\leq h(x(t), y(t), u(t), t, p) \\
0 &= r_e(x(t), y(t), p) \\
0 &\leq r_i(x(t), y(t), p)
\end{align*}
\]

Bounds of state variables, controls and parameters
Multiple shooting

NMPC problem

problem transformation

• discretization
• parametrization
• initial value problem (IVP)
• continuity equations

problem solution

• initial guess
• objective function
• simulation
• continuity
• path constraints
Multiple shooting – control parametrization

\[ v_N = v_{N-1} \]

\[ u(t) \]

\[ v_0, v_1, v_2 \]

\[ t_0, t_1, t_2, t_N \]
Multiple shooting – state parametrization

\[ s_0^x - x_0 = 0 \]

\[ x_{upper}(t; s, v) \]

\[ x_{lower}(t; s, v) \]

\[ h(s_i, v_i) \geq 0 \]
Finite dimensional NLP problem

\[ \min_{v_0, \ldots, v_{N-1}, s_0, \ldots, s_{N-1}} \left\{ J = \Phi(s_N, p, t_N) + \sum_{i=0}^{N-1} L(s_i, v_i, p) \right\} \]

s.t. 
\[ 0 = s_0 - x_0 \]
\[ 0 = s_{i+1} - x_i(s_i, v_i, p, t_{i+1}) \quad i = 0, 1, \ldots, N-1 \]
\[ 0 = g(s_i, v_i, p) \quad i = 0, 1, \ldots, N \]
\[ 0 \leq h(s_i, v_i, p) \]
\[ 0 = r_e(s_N, p) \]
\[ 0 \leq r_i(s_N, p) \]
Multiple shooting

**Problem formulation**

| $u(t), t \in [t_i, t_{i+1}] \rightarrow v_i$ | $\dot{x} = f(x, u, t)$ |
| $x(t_i) \rightarrow s_i$ | $x(t_i) = x_i$ |
| $i = 0...N$ | $t \in [t_i, t_{i+1}]$ |

**Simulator**

$\boldsymbol{x}^{(k)}(v_i^{(k)}, t_{i+1})$

$\frac{dx^{(k)}(t_{ij})}{dv_i^{(k)}}$

$\frac{dx^{(k)}(t_{ij})}{ds_i^{(k)}}$

**Optimizer**

$\boldsymbol{v}_i^{(0)}, s_i^{(0)}$

$\boldsymbol{v}_i^{(0)}, s_i^{(0)}$

$\boldsymbol{v}_i^{(k)}, s_i^{(k)}$

$\boldsymbol{v}_i^{(k)}, s_i^{(k)}$

$\boldsymbol{v}_i^{*}, s_i^{*}$

$\boldsymbol{v}_i^{*}, s_i^{*}$
Multiple shooting - problem solution

\[
(s_3^x)^* - x_2(t_3; s_2^x, v_2) = 0
\]

\[
(s_3^x)^* - x_2(t_3; s_2^x, v_2) = 0
\]

\[
(s_3^x)^* - x_2(t_3; s_2^x, v_2) = 0
\]

\[
(s_3^x)^* - x_2(t_3; s_2^x, v_2) = 0
\]

\[
(s_3^x)^* - x_2(t_3; s_2^x, v_2) = 0
\]

\[
(s_3^x)^* - x_2(t_3; s_2^x, v_2) = 0
\]

\[
(s_3^x)^* - x_2(t_3; s_2^x, v_2) = 0
\]

\[
(s_3^x)^* - x_2(t_3; s_2^x, v_2) = 0
\]

\[
(s_3^x)^* - x_2(t_3; s_2^x, v_2) = 0
\]

\[
(s_3^x)^* - x_2(t_3; s_2^x, v_2) = 0
\]

\[
(s_3^x)^* - x_2(t_3; s_2^x, v_2) = 0
\]

\[
(s_3^x)^* - x_2(t_3; s_2^x, v_2) = 0
\]

\[
(s_3^x)^* - x_2(t_3; s_2^x, v_2) = 0
\]

\[
(s_3^x)^* - x_2(t_3; s_2^x, v_2) = 0
\]

\[
(s_3^x)^* - x_2(t_3; s_2^x, v_2) = 0
\]

\[
(s_3^x)^* - x_2(t_3; s_2^x, v_2) = 0
\]

\[
(s_3^x)^* - x_2(t_3; s_2^x, v_2) = 0
\]

\[
(s_3^x)^* - x_2(t_3; s_2^x, v_2) = 0
\]

\[
(s_3^x)^* - x_2(t_3; s_2^x, v_2) = 0
\]

\[
(s_3^x)^* - x_2(t_3; s_2^x, v_2) = 0
\]

\[
(s_3^x)^* - x_2(t_3; s_2^x, v_2) = 0
\]

\[
(s_3^x)^* - x_2(t_3; s_2^x, v_2) = 0
\]

\[
(s_3^x)^* - x_2(t_3; s_2^x, v_2) = 0
\]

\[
(s_3^x)^* - x_2(t_3; s_2^x, v_2) = 0
\]
Collocation on finite elements

\[ x_i = x_{i-1} + \Delta t_i \]

\[ t_0 \quad x_{(i-1)0} \quad x_{(i-1)1} \quad x_{(i-1)2} \quad x_i \quad x_{i+1} \quad x_{(i+1)1} \quad x_{(i+1)2} \quad t_f \]

\[ x_{(i-1)M} = x_i \quad x_{iM} = x_{(i+1)0} \]
Collocation on finite elements

\[ x(t) = \sum_{k=0}^{K} x_{ik} l_k(\tau) \]

\[ \dot{x}(t) = \frac{1}{h_i} \sum_{k=0}^{K} x_{ik} \frac{d l_k(\tau)}{d\tau} \]

\[ l_k(\tau) = \prod_{k \neq j=0}^{K} \frac{\tau - \tau_j}{\tau_k - \tau_j} \]

\[ t \in [t_i, t_{i+1}] \]

\[ t = t_i + h_i\tau \]

\[ \dot{x}(t) = f(x, t) \]
Advantages of collocation on finite elements

- **Single-step method** → smooth profiles required only *within* the finite elements, discontinuity of control profiles at $t_i, i = 1, ..., N$ possible
- **High-order implicit method** → provides accurate profiles with relatively few finite elements, i.e. reduces problem size
- No stability limitations on element size for stiff systems
- Path constraints can be held inside the elements
Case study - CSTR

\[ \min_{x,u} \int_{t_0}^{t_f} \sum_{i=1}^{N_x} w_i (x_i - x_i^{(ref)})^2 + \sum_{i=1}^{N_u} w_i (u_i - u_i^{(ref)})^2 \, dt \]

s.t.  
- model equations
- variable bounds

Using 50 subintervals, IPOPT 3.4.0 optimizer.
CPU time is 0.953 sec.
Case study - CSTR

![Graph showing Molar concentration and Tank level over time]
Case study - CSTR

![Graph showing outlet flow rate and coolant temperature over time](image-url)
## Classification of methods – consequences for the optimization framework

<table>
<thead>
<tr>
<th></th>
<th>Single shooting</th>
<th>Multiple shooting</th>
<th>Simultaneous approach</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Optimization variables</strong></td>
<td>( v_i ) ( i = 0, \ldots, N - 1 )</td>
<td>( v_i, s_i ) ( i = 0, \ldots, N - 1 )</td>
<td>( v_i, x_i ) ( i = 0, \ldots, N \times K )</td>
</tr>
<tr>
<td><strong>Problem size</strong></td>
<td>small</td>
<td>intermediate</td>
<td>large</td>
</tr>
<tr>
<td><strong>Simulation</strong></td>
<td>1 IVP</td>
<td>( N ) IVP`s</td>
<td>None</td>
</tr>
<tr>
<td><strong>Path constraints</strong></td>
<td>none</td>
<td>fulfilled at ( t_i, i = 0, \ldots, N \times K )</td>
<td>fulfilled at ( t_i, i = 0, \ldots, N \times K )</td>
</tr>
</tbody>
</table>
Implementation in OpenModelica

What is needed?

Model

- Collocation
- Sensitivity equations

Optimization problem

- objective
- opt. variables
- constraints

Discretizer

Simulator

$V_i^{(k)}, S_i^{(k)}$

Interface:

- f, gradient, g, Jacobian

Optimizer

Page 18
Summary

Collocation method: fast accurate simulator

Multiple shooting: reduced problem size, path constraints incorporated

Multiple shooting with collocation: high accuracy and reduced computational effort

Structure of the algorithm: reveals what is needed for implementation in OpenModelica
Many thanks for your attention!
References