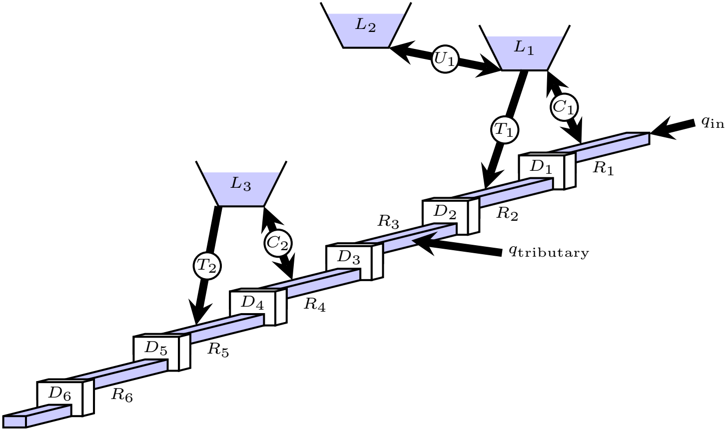
Carlo Savorgnan

Hydro Power Valley benchmark

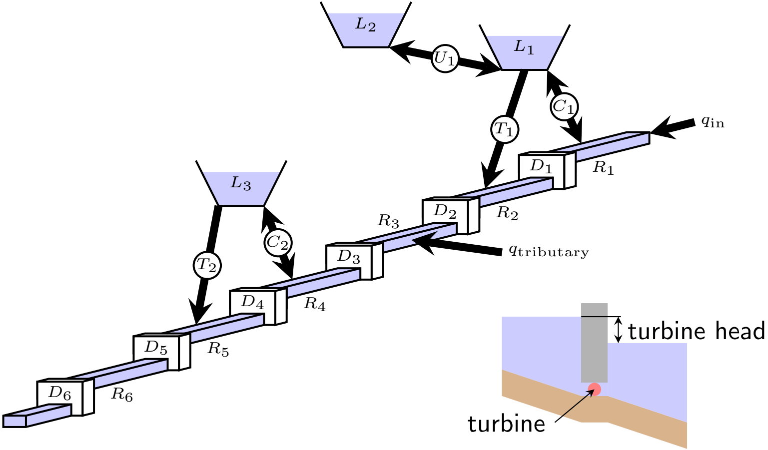
HD-MPC workshop – Leuven, 24 June 2011

HD–MPC

System overview



System overview



Reach model

based on the 1D Saint Venant equation

$$\begin{cases} \dfrac{\partial q(t,z)}{\partial z} + \dfrac{\partial s(t,z)}{\partial t} = 0 \\ \dfrac{1}{g}\dfrac{\partial}{\partial t}\left(\dfrac{q(t,z)}{s(t,z)}\right) + \dfrac{1}{2g}\dfrac{\partial}{\partial z}\left(\dfrac{q^2(t,z)}{s^2(t,z)}\right) + \dfrac{\partial h(t,z)}{\partial z} + I_{\mathrm{f}}(t,z) - I_0(z) = 0 \end{cases}$$

Reach model

based on the 1D Saint Venant equation

$$\begin{cases} \dfrac{\partial q(t,z)}{\partial z} + \dfrac{\partial s(t,z)}{\partial t} = 0 \\ \dfrac{1}{g}\dfrac{\partial}{\partial t}\left(\dfrac{q(t,z)}{s(t,z)}\right) + \dfrac{1}{2g}\dfrac{\partial}{\partial z}\left(\dfrac{q^2(t,z)}{s^2(t,z)}\right) + \dfrac{\partial h(t,z)}{\partial z} + I_{\mathrm{f}}(t,z) - I_0(z) = 0 \end{cases}$$

Lake model

$$\frac{\partial v(t)}{\partial t} = q_{\mathrm{in}}(t) - q_{\mathrm{out}}(t).$$

Duct model

$$q_{U_1}(t) = S_{U_1}\mathrm{sign}(h_{L_2}(t) - h_{L_1}(t) + \Delta h_{U_1})\sqrt{2g|h_{L_2}(t) - h_{L_1}(t) + \Delta h_{U_1}|}$$

Duct model

$$q_{U_1}(t) = S_{U_1} \text{sign}(h_{L_2}(t) - h_{L_1}(t) + \Delta h_{U_1}) \sqrt{2g|h_{L_2}(t) - h_{L_1}(t) + \Delta h_{U_1}|}$$

Turbine and Pump model

Turbine:

$$p_{\mathrm{t}}(t) = k_{\mathrm{t}} q_{\mathrm{t}}(t) \Delta h_{\mathrm{t}}(t)$$

Pump:

$$p_{\mathrm{p}}(t) = -k_{\mathrm{p}} q_{\mathrm{p}}(t) \Delta h_{\mathrm{p}}(t)$$

Power tracking

$$\min_{x_i,u_i} \int_0^T \gamma \, |e(t)| \, dt + \sum_{i=1}^{8} \int_0^T (x_i(t) - x_{ss,i})^T Q_i (x_i(t) - x_{ss,i}) dt$$

s.t.
$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_8(t) \end{bmatrix} = f\left(\begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_8(t) \end{bmatrix}, \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_8(t) \end{bmatrix} \right)$$

$$(x_i(t), u_i(t)) \in C_i \quad i = 1, \ldots, 8$$

$$x_i(t) = x_{i,0} \quad i = 1, \ldots, 8$$

$$e(t) = p_r(t) - \textstyle\sum_{i=1}^{8} p_i(x_i(t), u_i(t))$$

Power tracking

$$\min_{x_i, u_i} \int_0^T \gamma\,|e(t)|\,dt + \sum_{i=1}^{8} \int_0^T (x_i(t) - x_{ss,i})^T Q_i (x_i(t) - x_{ss,i}) dt$$

$$\text{s.t.} \quad \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_8(t) \end{bmatrix} = f\left(\begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_8(t) \end{bmatrix}, \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_8(t) \end{bmatrix} \right)$$

$$(x_i(t), u_i(t)) \in C_i \quad i = 1, \ldots, 8$$

$$x_i(t) = x_{i,0} \quad i = 1, \ldots, 8$$

$$e(t) = p_r(t) - \textstyle\sum_{i=1}^{8} p_i(x_i(t), u_i(t))$$

- reaches discretized using 20 cells $\rightarrow$ 249 states
- 10 inputs
- prediction horizon 24 h (86400 s), sampling time 30 min (1800 s)

$$
\begin{aligned}
\min_{\substack{x,u,z,\\ y,e}} \quad & \int_0^T \ell(e(t))dt + \sum_{i=1}^{M} \int_0^T \ell^i(x^i(t), u^i(t), z^i(t))dt \\
\text{s.t.} \quad & \dot{x}^i(t) = f^i(x^i(t), u^i(t), z^i(t)) \\
& y^i(t) = g^i(x^i(t), u^i(t), z^i(t)) \\
& x^i(0) = \bar{x}_0^i \\
& z^i(t) = \textstyle\sum_{j=1}^{M} A_{ij} y^j(t) \\
& e(t) = r(t) + \textstyle\sum_{i=1}^{M} B^i y^i(t) \\
& p^i(x^i(t), u^i(t)) \geq 0, \quad q(e(t)) \geq 0 \quad t \in [0, T]
\end{aligned}
$$

Problem

If we want to implement a shooting method in which every subsystem dynamics is integrated and linearized separately, we need to know $z^i(t)$.

Multiple shooting for distributed systems (MSD)

Key idea: The signals $z^i(t)$, $y^i(t)$ and $e(t)$ can be represented as a linear combination of orthogonal polynomials.

Multiple shooting for distributed systems (MSD)

Key idea: The signals $z^i(t)$, $y^i(t)$ and $e(t)$ can be represented as a linear combination of orthogonal polynomials.

$$
\begin{aligned}
\min_{\substack{u_n^i, x_n^i, \mathbf{z}_n^i, \\ \mathbf{y}_n^i, \mathbf{e}_n}} \quad & \sum_{n=0}^{N-1} \left(L_n(\mathbf{e}_n) + \sum_{i=1}^{M} L_n^i(x_n^i, u_n^i, \mathbf{z}_n^i) \right) \\
\text{s.t.} \quad & x_{n+1}^i = F_n^i(x_n^i, u_n^i, \mathbf{z}_n^i) \qquad n = 0, \dots, N-1 \\
& \mathbf{y}_n^i = G_n^i(x_n^i, u_n^i, \mathbf{z}_n^i) \qquad n = 0, \dots, N-1 \\
& x_0^i = \bar{x}_0^i \\
& \mathbf{z}_n^i = \textstyle\sum_{i=1}^{M} A_{ij} \mathbf{y}_n^j \\
& \mathbf{e}_n = \mathbf{r}_n + \textstyle\sum_{i=1}^{M} B_{ij} \mathbf{y}_n^j \\
& p^i(x_n^i, u_n^i) \geq 0, \qquad Q_n(\mathbf{e}_n) \geq 0
\end{aligned}
$$

Multiple shooting for distributed systems (MSD)

Key idea: The signals $z^i(t)$, $y^i(t)$ and $e(t)$ can be represented as a linear combination of orthogonal polynomials.

$$\min_{\substack{u_n^i, x_n^i, \mathbf{z}_n^i,\\ \mathbf{y}_n^i, \mathbf{e}_n}} \quad \sum_{n=0}^{N-1} \left(L_n(\mathbf{e}_n) + \sum_{i=1}^{M} L_n^i(x_n^i, u_n^i, \mathbf{z}_n^i) \right)$$

s.t.

$$\begin{aligned}
& x_{n+1}^i = F_n^i(x_n^i, u_n^i, \mathbf{z}_n^i) \qquad n = 0, \ldots, N-1 \\
& \mathbf{y}_n^i = G_n^i(x_n^i, u_n^i, \mathbf{z}_n^i) \qquad n = 0, \ldots, N-1 \\
& x_0^i = \bar{x}_0^i \\
& \mathbf{z}_n^i = \textstyle\sum_{i=1}^{M} A_{ij} \mathbf{y}_n^j \\
& \mathbf{e}_n = \mathbf{r}_n + \textstyle\sum_{i=1}^{M} B_{ij} \mathbf{y}_n^j \\
& p^i(x_n^i, u_n^i) \geq 0, \qquad Q_n(\mathbf{e}_n) \geq 0
\end{aligned}$$

This NLP can be solved using standard methods, e.g. SQP

Some simulation results

QP solution $\approx$ 9.85 s

Integration with sensitivities (1 shooting interval, 30 min)

- 1 subsystem $\approx$ 1.92 s
- whole system $\approx$ 33.96 s (4.25 s/subsystem)

Some simulation results

QP solution $\approx$ 9.85 s

Integration with sensitivities (1 shooting interval, 30 min)

- 1 subsystem $\approx$ 1.92 s
- whole system $\approx$ 33.96 s (4.25 s/subsystem)

Assuming we have an unlimited number of processors:

Some simulation results

QP solution ≈ 9.85 s

Integration with sensitivities (1 shooting interval, 30 min)

- 1 subsystem ≈ 1.92 s
- whole system ≈ 33.96 s (4.25 s/subsystem)

Assuming we have an unlimited number of processors:

- Single shooting: $1630 + 9.85$ s

Some simulation results

QP solution $\approx$ 9.85 s

Integration with sensitivities (1 shooting interval, 30 min)

- 1 subsystem $\approx$ 1.92 s
- whole system $\approx$ 33.96 s (4.25 s/subsystem)

Assuming we have an unlimited number of processors:

- Single shooting: 1630 + 9.85 s
- Multiple shooting: 33.96 + 9.85 s

Some simulation results

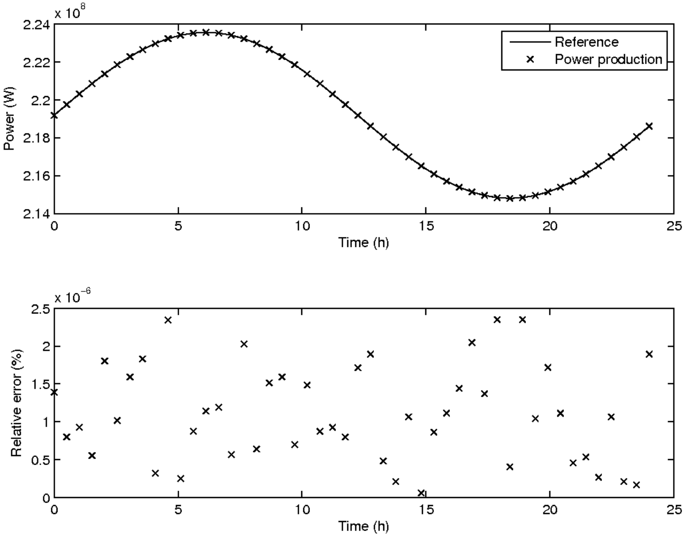
QP solution $\approx$ 9.85 s

Integration with sensitivities (1 shooting interval, 30 min)

- 1 subsystem $\approx$ 1.92 s
- whole system $\approx$ 33.96 s (4.25 s/subsystem)

Assuming we have an unlimited number of processors:

- Single shooting: 1630 + 9.85 s
- Multiple shooting: 33.96 + 9.85 s
- Multiple shooting for distributed systems: **1.92 + 9.85 s**


x 10^8
2.24
2.22
2.2
2.18
2.16
2.14
Power (W)
0
5
10
15
20
25
Time (h)
Reference
Power production
x 10^-6
2.5
2
1.5
1
0.5
0
Relative error (%)
0
5
10
15
20
25
Time (h)