Multiplexed Model Predictive Control

K.V. Ling, J.M. Maciejowski, B. Wu

ekvling@ntu.edu.sg

jmm@eng.cam.ac.uk

bingfang_wu@pmail.ntu.edu.sg

Cambridge University **NTU Singapore**

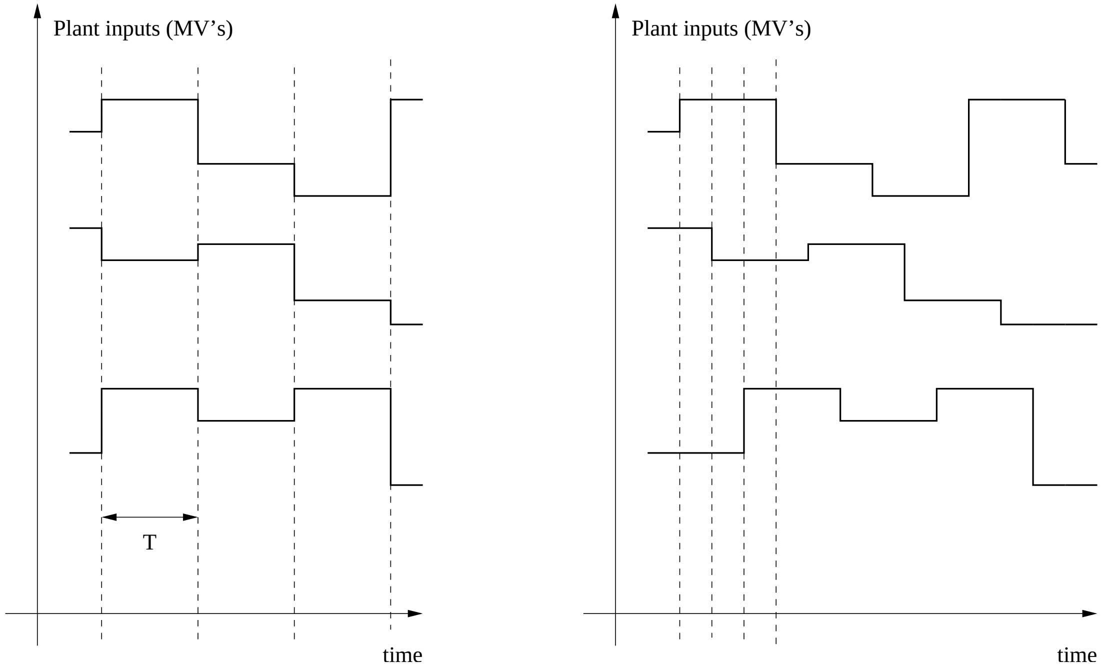
Model Predictive Control (MPC)

Control by using on-line optimization (QP or LP)

- Increasingly used for fast systems
- *MPC on a Chip* or FPGA — size limits
- Reduce complexity of optimization problem
- Various decentralised schemes proposed — but all with synchronous control updates.

Multiplexed MPC — multi-input systems

Sequential updates of control inputs



Left: Conventional MPC, Right: Multiplexed MPC

Assumptions

- Only one input updated at each time step (at time kT/m), in sequence.
- Measurements of state vector are made at intervals of T/m.
- Current state x_k is known when deciding the update of each input. *x_k is known to each controller.*
- *Scheme 1:* Optimise all inputs over future horizon at each step. *Can be thought of as* 1 *controller.*
- *Scheme 2:* Optimise only one input over future horizon at each step. *Can be thought of as m controllers.*

Update one input at a time

Plant: If only one input is updated at each k then

$$\begin{aligned} x_{k+1} &= Ax_k + \sum_{j=1}^{m} B_j \Delta u_{j,k} \\ &= Ax_k + B_{\sigma(k)} \Delta \tilde{u}_k \end{aligned}$$

where

$$\sigma(k) = (k \mod m) + 1$$

is a **periodic switching function**: $\sigma(k+m) = \sigma(k)$

So multi-input LTI plant looks like **periodic single-input** plant.

Stability of Scheme 1

Cost function:

$$\begin{aligned} J_k &= \sum_{i=0}^{\infty}(\|x_{k+i+1}\|_q^2 + \|\Delta u_{k+i}\|_r^2) \\ &= \sum_{i=0}^{N-1}\left(\|x_{k+i+1}\|_q^2 + \|\Delta u_{k+i}\|_r^2\right) + x_{k+N+1}^T P_{k+N+1} x_{k+N+1} \end{aligned}$$

for suitable P_{k+N+1}.

Infinite horizon $\Rightarrow$ closed-loop stable (if feasible).

Constraints can be imposed during first N steps of horizon.

Periodic Riccati Equation

$$P_k = A^T P_{k+1} A - A^T P_{k+1} B_{\sigma(k)} (B_{\sigma(k)}^T P_{k+1} B_{\sigma(k)} + r)^{-1} B_{\sigma(k)}^T P_{k+1} A + q$$

This converges to a periodic solution (given suitable final condition).

State feedback — after end of prediction horizon:

$$K_k = -(B_{\sigma(k)}^T P_{k+1} B_{\sigma(k)} + r)^{-1} B_{\sigma(k)}^T P_{k+1} A$$

Stability of *Scheme 1* follows easily — if feasible.

Scheme 2

- Controller j decides future sequence of j'th input only.
- Other inputs are treated as known disturbances.
- Assume that controller j knows the future plans of the other controllers, and assumes $\Delta u_{\sigma(k),k+i} = K_{\sigma(k)} x_{k+i}$ beyond the planning horizon.
- Stability proof idea: J_k is a *Lyapunov function*, if problem is feasible.

$$\Delta\vec{U}_k = \begin{bmatrix} \boldsymbol{\Delta u_{1,k}} \\ \Delta u_{2,k+1} \\ \boldsymbol{\Delta u_{1,k+2}} \\ \vdots \\ \boldsymbol{\Delta u_{1,k+2N-4}} \\ \Delta u_{2,k+2N-3} \\ \boldsymbol{\Delta u_{1,k+2(N-1)}} \\ K_2 x_{k+2N-1} \\ K_1 x_{k+2N} \\ K_2 x_{k+2N+1} \\ K_1 x_{k+2N+2} \\ \vdots \end{bmatrix}, \quad \Delta\vec{U}_{k+1} = \begin{bmatrix} \boldsymbol{\Delta u_{2,k+1}} \\ \Delta u_{1,k+2} \\ \boldsymbol{\Delta u_{2,k+3}} \\ \vdots \\ \boldsymbol{\Delta u_{2,k+2N-3}} \\ \Delta u_{1,k+2(N-1)} \\ \boldsymbol{\Delta u_{2,k+2N-1}} \\ K_1 x_{k+2N} \\ K_2 x_{k+2N+1} \\ K_1 x_{k+2N+2} \\ \vdots \end{bmatrix}$$

Pattern of control updates in *Scheme 2*.

Entries in **bold** get updated.

Scheme 2: How to compute the cost?

Assume that, after end of prediction horizon (of length N):

$$\Delta u_{\sigma(k),k+i} = K_{\sigma(k)} x_{k+i}$$

Let

$$\Phi_j = A + B_j K_j$$

Monodromy matrices:

$$\begin{aligned} \Psi_1 &= \Phi_m \Phi_{m-1} \ldots \Phi_2 \Phi_1 \\ \Psi_2 &= \Phi_1 \Phi_m \ldots \Phi_3 \Phi_2 \\ &\vdots \\ \Psi_m &= \Phi_{m-1} \Phi_{m-2} \ldots \Phi_1 \Phi_m \end{aligned}$$

Stability condition: $|\lambda_j(\Psi_i)| < 1$ for all j, for any i.

Scheme 2: Cost function J_k . . .

Paper has errors here! Details wrong, Idea OK.

At time k, controller $\sigma(k)$ evaluates J_k as:

$$J_k = \sum_{i=0}^{m(N-1)} \left(\|x_{k+i+1}\|_q^2 + \|\Delta u_{k+i}\|_r^2\right) + x_{k+m(N-1)+1}^T P_{\sigma(k)} x_{k+m(N-1)+1}$$

What is $P_{\sigma(k)}$?

The *"tail"* of J_k is

$$J_{k+m(N-1)+1} = \sum_{i=m(N-1)+1}^{\infty} \left(\|x_{k+i+1}\|_q^2 + \|\Delta u_{k+i}\|_r^2\right)$$

$$= \sum_{i=2}^{m} \|x_{k+m(N-1)+i}\|_q^2 + \sum_{i=N-1}^{\infty} \left(\|\mathcal{X}_{k+m(i+1)+1}\|_Q^2 + \|\Delta \mathcal{U}_{k+mi+1}\|_R^2\right)$$

where

$$\mathcal{X}_k = \begin{bmatrix} x_k \\ x_{k+1} \\ \vdots \\ x_{k+m-1} \end{bmatrix}, \ \Delta\mathcal{U}_k = \begin{bmatrix} \Delta u_{\sigma(k),k} \\ \Delta u_{\sigma(k+1),k+1} \\ \vdots \\ \Delta u_{\sigma(k+m-1),k+m-1} \end{bmatrix}$$

and $Q = \mathrm{diag}[q, \ldots, q]$, $R = \mathrm{diag}[r, \ldots, r]$.

State transition equation — if $i > m(N-1)$:

$$\mathcal{X}_{k+1+m(N+i+1)} = \begin{bmatrix} \Psi_{\sigma(k+1)} & 0 & \cdots & 0 \\ 0 & \Psi_{\sigma(k+2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Psi_{\sigma(k)} \end{bmatrix} \mathcal{X}_{k+1+m(N+i)}$$

Recall:

If $x_{i+1} = \Phi x_i$ and $u_i = K x_i$ and $J = \sum_{i=0}^{\infty}(\|x_{i+1}\|_Q^2 + \|u_i\|_R^2)$
then $J = x_0^T P x_0$ where $P = \Phi^T P \Phi + \Phi^T Q \Phi + K^T R K$.

Applying this to our problem:

$$J_{k+m(N-1)+1} = \mathcal{X}_{k+m(N-1)+1}^T \times \operatorname{diag}[\Pi_{\sigma(k+1)}, \ldots, \Pi_{\sigma(k+m)}] \mathcal{X}_{k+m(N-1)+1}$$

where

$$\Pi_\ell = \Psi_\ell^T \Pi_\ell \Psi_\ell + \Psi_\ell^T q \Psi_\ell + K_\ell^T r K_\ell \quad \text{for } \ell = 1, 2, \ldots, m$$

But

$$\begin{aligned} x_{k+m(N-1)+2} &= \Phi_{\sigma(k+1)} x_{k+m(N-1)+1} \\ x_{k+m(N-1)+3} &= \Phi_{\sigma(k+2)} \Phi_{\sigma(k+1)} x_{k+m(N-1)+1} \\ &\text{etc} \end{aligned}$$

so we obtain

$$J_{k+m(N-1)+1} = x^T_{k+m(N-1)+1} P_{\sigma(k)} x_{k+m(N-1)+1}$$

where

$$\begin{aligned} P_{\sigma(k)} = \Pi_{\sigma(k+1)} + \Phi^T_{\sigma(k+1)} \Pi_{\sigma(k+2)} \Phi_{\sigma(k+1)} + \cdots \\ + [\Phi^T_{\sigma(k+1)} \ldots \Phi^T_{\sigma(k+m-1)} \Pi_{\sigma(k+m)} \Phi_{\sigma(k+m-1)} \ldots \Phi_{\sigma(k+1)}] \end{aligned}$$

Scheme 2 **stability proof**

Standard *monotonically decreasing cost* argument:

- If, at step $k+1$, controller $\sigma(k+1)$ leaves the moves $\Delta u_{\sigma(k+1),k+1}, \ldots, \Delta u_{\sigma(k+1),k+m(N-1)+1}$ unchanged, then

$$J_{k+1} = J_k^o - \|x_{k+1}\|_q^2 - \|\Delta u_k\|_r^2 < J_k^o$$

- But $J_{k+1}^o \leq J_{k+1}$ by optimality.
- Hence $J_{k+1}^o \leq J_k^o$, (equality only if $x_k = 0$).

Hence J_k^o is a Lyapunov function.

***Scheme 2* is stable — if feasible.**

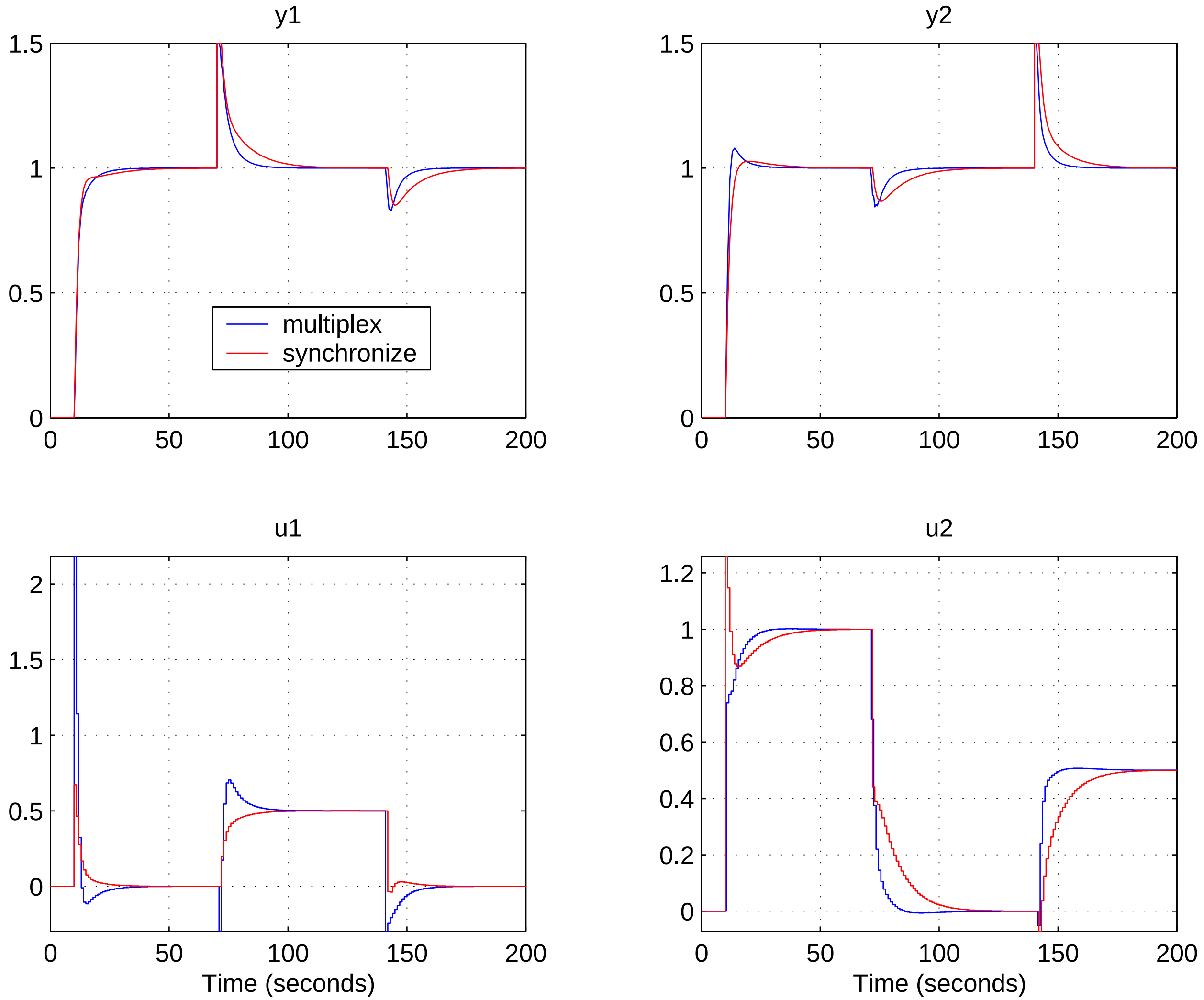
Example

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} \frac{1}{7s+1} & \frac{1}{3s+1} \\ \frac{2}{8s+1} & \frac{1}{4s+1} \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix}$$

$m = 2, \qquad T = 1$ sec, $\qquad T/m = 0.5$ sec.

No constraints.

Step disturbance on y_1 at $t = 70.1$ sec
Step disturbance on y_2 at $t = 140.1$ sec


y1
y2
u1
u2
multiplex
synchronize
Time (seconds)
Time (seconds)

Conclusions

- Multiplexed MPC updates one input at a time.
- *Do something sooner* can be better than *Do optimal thing later*.
- Extension of Chmielewski-Manousiouthakis approach using periodic systems theory.
- Constraints, feasibility etc not addressed yet.
- Complexity reduction requires constraint decoupling too.
- **Generalisations:** Unequal intervals; Groups of inputs.