UNIVERSITY OF CAMBRIDGE

$\ell_{asso} - MPC$ for Over-actuated Systems

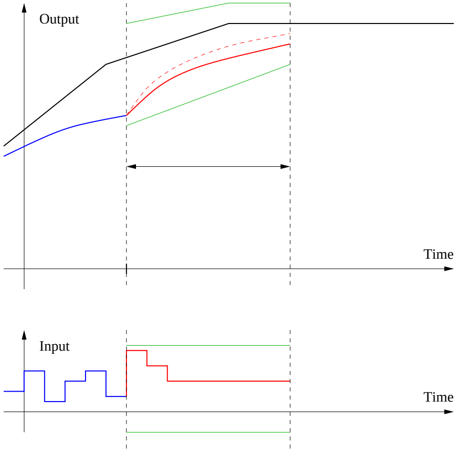
Jan Maciejowski and Marco Gallieri

Workshop on the Control of Cyber-Physical Systems
Notre Dame London, 20 October 2012

Cambridge University Engineering Department

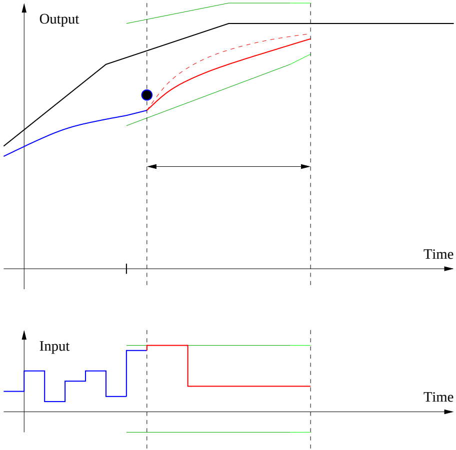
Model Predictive Control (MPC) — the basic idea

Plan over a future horizon



Model Predictive Control (MPC) — the basic idea

The receding horizon concept



The optimisation problem

If:

Model: *Linear* & **Cost:** *Convex* & **Constraints:** *Convex*

Then:

Convex optimisation problem (QP, LP, . . .)

The optimisation problem

If:

Model: *Linear* & **Cost:** *Convex* & **Constraints:** *Convex*

Then:

Convex optimisation problem (QP, LP, . . .)

Else:

Non-convex optimisation — local minima

The optimisation problem

If:

Model: *Linear* & **Cost:** *Convex* & **Constraints:** *Convex*

Then:

Convex optimisation problem (QP, LP, . . .)

Else:

Non-convex optimisation — local minima

But:

Must be solved 'quickly'.

The optimisation problem

If:

Model: *Linear* & **Cost:** *Convex* & **Constraints:** *Convex*

Then:

Convex optimisation problem (QP, LP, . . .)

Else:

Non-convex optimisation — local minima

But:

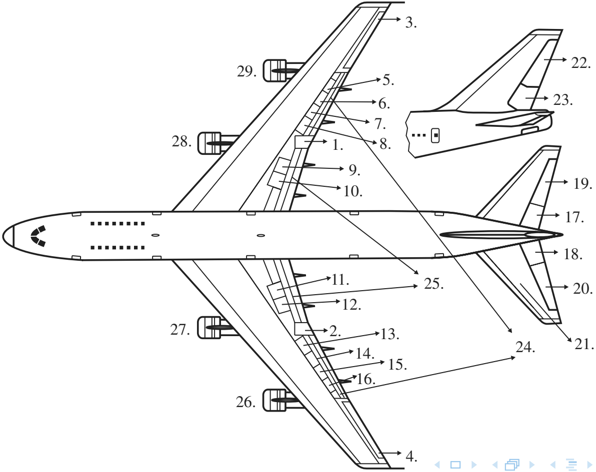
Must be solved 'quickly'.

So:

Formulate convex problem if possible.

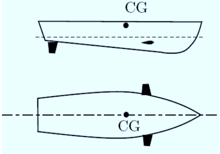
Over-actuated systems

Aircraft example: 12 states, nearly 30 actuators



Over-actuated systems

Ship roll stabilisation: fins and rudder



Cruise ship
Michelangelo
(1962)

Over-actuated systems

- Standard MPC moves all the actuators all of the time (like LQR).

Over-actuated systems

- Standard MPC moves all the actuators all of the time (like LQR).
- We may have *preferred* actuators:

Over-actuated systems

- Standard MPC moves all the actuators all of the time (like LQR).
- We may have *preferred* actuators:
  - Aircraft: Ailerons normally, spoilers only if necessary.

Over-actuated systems

- Standard MPC moves all the actuators all of the time (like LQR).
- We may have *preferred* actuators:
  - Aircraft: Ailerons normally, spoilers only if necessary.
  - Ship roll control: Stabilisers normally, rudder only if necessary.

Over-actuated systems

- Standard MPC moves all the actuators all of the time (like LQR).
- We may have *preferred* actuators:
  - Aircraft: Ailerons normally, spoilers only if necessary.
  - Ship roll control: Stabilisers normally, rudder only if necessary.
- Actuators may ‘fight’ each other to get differential action.

Over-actuated systems

- Standard MPC moves all the actuators all of the time (like LQR).
- We may have *preferred* actuators:
  - Aircraft: Ailerons normally, spoilers only if necessary.
  - Ship roll control: Stabilisers normally, rudder only if necessary.
- Actuators may ‘fight’ each other to get differential action.
- So we may want sparse solutions.

Over-actuated systems

- Standard MPC moves all the actuators all of the time (like LQR).
- We may have *preferred* actuators:
  - Aircraft: Ailerons normally, spoilers only if necessary.
  - Ship roll control: Stabilisers normally, rudder only if necessary.
- Actuators may 'fight' each other to get differential action.
- So we may want sparse solutions.
- If control actions are expensive, we may want sparse in time solutions — like *Statistical Process Control*.

How to get sparse solutions?

'Regularise' by adding $\|u\|_q$ (or $\|\Delta u\|_q$) penalty term

$$\min_{\mathbf{u}} F(x_N) + \sum_{k=0}^{N-1} \left(x_k^T Q x_k + u_k^T R u_k \right) \qquad \text{subject to constraints.}$$

How to get sparse solutions?

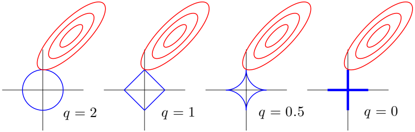
'Regularise' by adding $\|u\|_q$ (or $\|\Delta u\|_q$) penalty term

$$\min_{\mathbf{u}} F(x_N) + \sum_{k=0}^{N-1} \left(x_k^T Q x_k + u_k^T R u_k \right) + \lambda \|u_k\|_q \quad \text{subject to constraints.}$$

How to get sparse solutions?

'Regularise' by adding $\|u\|_q$ (or $\|\Delta u\|_q$) penalty term

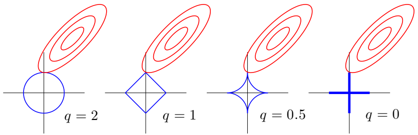
$$\min_{\mathbf{u}} F(x_N) + \sum_{k=0}^{N-1} \left(x_k^T Q x_k + u_k^T R u_k \right) + \lambda \|u_k\|_q \quad \text{subject to constraints.}$$



How to get sparse solutions?

'Regularise' by adding $\|u\|_q$ (or $\|\Delta u\|_q$) penalty term

$$\min_{\mathbf{u}} F(x_N) + \sum_{k=0}^{N-1} \left(x_k^T Q x_k + u_k^T R u_k\right) + \lambda \|u_k\|_q \quad \text{subject to constraints.}$$



$q = 1$ is the smallest q that gives a convex problem.

ℓ_{asso}-MPC gives sparse solutions for large enough λ

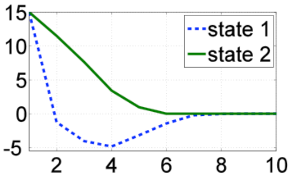
Example: Unstable toy plant

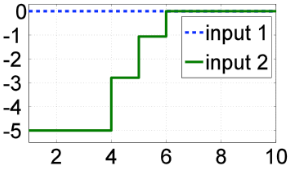
$$A = \begin{bmatrix} 0.15 & 0.1 \\ 0 & 1.1 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

$$Q = \begin{bmatrix} 20 & 0 \\ 0 & 60 \end{bmatrix} \quad R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

$$\|x\|_\infty \leq 20 \qquad \|u\|_\infty \leq 5$$

$$\lambda = 300$$





ℓ_{asso}-MPC gives sparse solutions for large enough λ

Example: Ship roll control

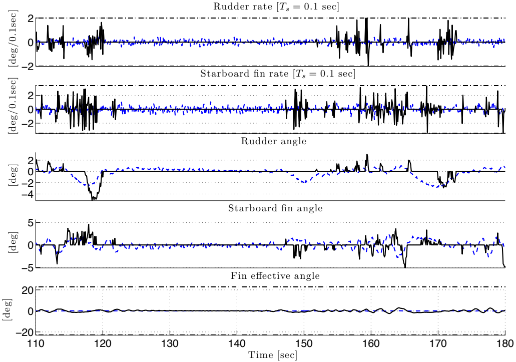


Figure: Solid: ℓ_{asso}-MPC ($\lambda = 1.8$). Dashed: Standard MPC.

Why 'ℓ_{asso}-MPC'?

- *LASSO*: L_1-constrained fitting for statistics and data mining.

Why 'ℓ_{asso}-MPC'?

- *LASSO*: L_1-constrained fitting for statistics and data mining.
- A method for variable selection —
  which are the important explanatory variables?

Why 'ℓ_{asso}-MPC'?

- *LASSO*: L_1-constrained fitting for statistics and data mining.
- A method for variable selection —
  which are the important explanatory variables?
- Has been used in 'sparse regression', data compression,
  wavelet expansions, . . .

Why ‘ℓ_{asso}-MPC’?

- *LASSO*: L_1-constrained fitting for statistics and data mining.
- A method for variable selection —
  which are the important explanatory variables?
- Has been used in ‘sparse regression’, data compression,
  wavelet expansions, . . .
- It’s not ‘ℓ_2-MPC’ or ‘ℓ_1-MPC’.

ℓ_{asso}-MPC: Anything new here?

Is control different from modelling/statistics?

- Stability

ℓ_{asso}-MPC: Anything new here?

Is control different from modelling/statistics?

- Stability
  1. *Dual-mode* approach: Switch to regular LQR when close to set-point. But then sparseness is lost in the terminal set.

ℓ_{asso}-MPC: Anything new here?

Is control different from modelling/statistics?

- Stability
  1. *Dual-mode* approach: Switch to regular LQR when close to set-point. But then sparseness is lost in the terminal set.
  2. *Contractive terminal set* approach: $\hat{x}_{k+N} \in \{x : F(x) \leq c_k\}$, with $\{c_k\}$ decreasing according to LQR 'cost-to-go'. ($F(x)$ is terminal cost.)

ℓ_{asso}-MPC: Anything new here?

Is control different from modelling/statistics?

- Stability
  1. *Dual-mode* approach: Switch to regular LQR when close to set-point. But then sparseness is lost in the terminal set.
  2. *Contractive terminal set* approach: $\hat{x}_{k+N} \in \{x : F(x) \leq c_k\}$, with $\{c_k\}$ decreasing according to LQR 'cost-to-go'. ($F(x)$ is terminal cost.)
  3. *New terminal cost* which preserves sparseness in the terminal set.

ℓ_{asso}-MPC: Anything new here?

Is control different from modelling/statistics?

- Stability
  1. *Dual-mode* approach: Switch to regular LQR when close to set-point. But then sparseness is lost in the terminal set.
  2. *Contractive terminal set* approach: $\hat{x}_{k+N} \in \{x : F(x) \leq c_k\}$, with $\{c_k\}$ decreasing according to LQR 'cost-to-go'. ($F(x)$ is terminal cost.)
  3. *New terminal cost* which preserves sparseness in the terminal set.
- Robustness: Add/tighten constraints to get *recursive feasibility* with model $x_{k+1} = Ax_k + Bu_k + w_k, \quad w_k \in W$.

- Stability
  1. *Dual-mode* approach: Switch to regular LQR when close to set-point. But then sparseness is lost in the terminal set.
  2. *Contractive terminal set* approach: $\hat{x}_{k+N} \in \{x : F(x) \leq c_k\}$, with $\{c_k\}$ decreasing according to LQR 'cost-to-go'. ($F(x)$ is terminal cost.)
  3. *New terminal cost* which preserves sparseness in the terminal set.
- Robustness: Add/tighten constraints to get *recursive feasibility* with model $x_{k+1} = Ax_k + Bu_k + w_k, \quad w_k \in W$.
- Offset-free tracking: Use disturbance estimator and target calculator (modified for ℓ_1 term).

Conclusions

- MPC is very successful, in great demand.

Conclusions

- MPC is very successful, in great demand.
- Over-actuated systems or expensive controls: ℓ_{asso}-MPC.
  (An example of What not How.)

Conclusions

- MPC is very successful, in great demand.
- Over-actuated systems or expensive controls: ℓ_{asso}-MPC.
  (An example of What not How.)
- Simultaneous regulation and control allocation.

Conclusions

- MPC is very successful, in great demand.
- Over-actuated systems or expensive controls: ℓ_{asso}-MPC.
  (An example of What not How.)
- Simultaneous regulation and control allocation.
- Further developments:

Conclusions

- MPC is very successful, in great demand.
- Over-actuated systems or expensive controls: ℓ_{asso}-MPC.
  (An example of What not How.)
- Simultaneous regulation and control allocation.
- Further developments:
  - Tuning to select preferred actuators, enhance pre-existing controllers.

Conclusions

- MPC is very successful, in great demand.
- Over-actuated systems or expensive controls: ℓ_{asso}-MPC.
  (An example of What not How.)
- Simultaneous regulation and control allocation.
- Further developments:
  - Tuning to select preferred actuators, enhance pre-existing controllers.
  - Design to maximise region of attraction.

References

- M. Gallieri and J.M. Maciejowski, The ℓ_{asso} MPC: Smart regulation of over-actuated systems, *Proc. American Control Conference*, Montreal, July 2012.
- H. Ohlsson, F. Gustafsson, L. Ljung, and S. Boyd, Trajectory generation using sum-of-norms regularization, *Proc. IEEE Conference on Decision and Control*, Atlanta, December 2010.