UNIVERSITY OF CAMBRIDGE

Soft-constrained LASSO MPC for Robust LTI Tracking

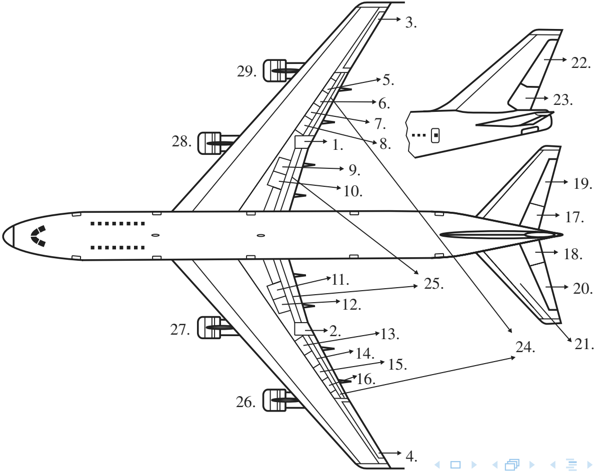
Marco Gallieri (mg574@cam.ac.uk)
with Jan M. Maciejowski (jmm@eng.cam.ac.uk)

Conference on Decision and Control – Firenze, 13 December 2013

Cambridge University Engineering Department

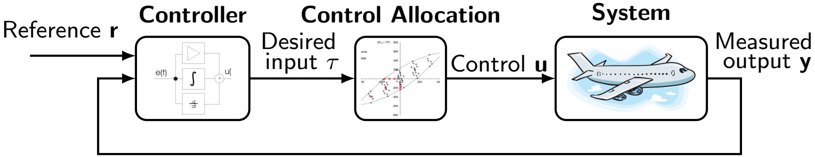
Multiply-actuated systems

Aircraft example: 16 states, nearly 30 actuators



LASSO Model Predictive Control (ℓ_{asso}-MPC)

Control of multiply-actuated systems – e.g. Aircraft $\approx$ 30 inputs



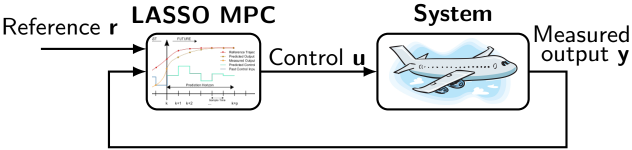


Figure: Integration of multiple control layers through LASSO MPC.

LASSO Model Predictive Control (ℓ_{asso}-MPC)

Preferred actuators

Cost function: $\ell(x_j, u_j) = x_j^T Q x_j + u_j^T R u_j + \|S u_j\|_1$

- Standard MPC moves all the actuators all of the time
- We may have *preferred* actuators:
  - Aircraft: Ailerons normally, spoilers only if necessary.
- We want *sparse control allocation*.
- LASSO: ℓ_1-norm regularisation.
- Quadratic cost on state
  $\implies$ smooth plant behaviour.

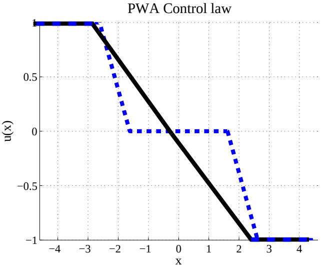


Figure: Primary (solid) and secondary actuator (dashed).

Who inspired our work ?

Authors who have previously investigated regularised MPC

- H. Ohlsson, F. Gustafsson, L. Ljung, and S. Boyd, 2010. "Trajectory generation using sum-of-norms regularization", IEEE CDC.
  - An MPC with sparse input derivatives (Δu). Reduce actuator's wear and tear.
- M. Nagahara and D.E. Quevedo, 2011. "Sparse representations for packetized predictive networked control", Proc. IFAC World Con.
  - Remote control. Compress control message. Send future N moves. Unconstrained boundedness and robustness to communication losses.

ℓ_{asso}-MPC (or ℓ_1-regularised MPC)

Let's start from a standard quadratic MPC

Control the constrained discrete-time LTI system

$$x(k+1) = Ax(k) + Bu(k) + B_w w(k),$$

$$x(k) \in \mathbb{X}, \;\; u(k) \in \mathbb{U}, \;\; \|w(k)\|_\infty \leq \mu, \; \forall k \geq 0,$$

ℓ_{asso}-MPC (or ℓ_1-regularised MPC)

Let's start from a standard quadratic MPC

Control the constrained discrete-time LTI system

$$x(k+1) = Ax(k) + Bu(k),$$

$$x(k) \in \mathbb{X},\ \ u(k) \in \mathbb{U},\ \ \forall k \geq 0,$$

ℓ_{asso}-MPC (or ℓ_1-regularised MPC)

Let's start from a standard quadratic MPC

Control the constrained discrete-time LTI system

$$x(k+1) = Ax(k) + Bu(k),$$

$$x(k) \in \mathbb{X},\ \ u(k) \in \mathbb{U},\ \ \forall k \geq 0,$$

by applying the first move of the optimal sequence

$$\underline{u}^\star = \arg\min_{\underline{u}} \hat{x}_N^T P \hat{x}_N \qquad + \sum_{j=0}^{N-1} \hat{x}_j^T Q \hat{x}_j + \hat{u}_j^T R \hat{u}_j \qquad \text{s.t. constraints.}$$

ℓ_{asso}-MPC (or ℓ_1-regularised MPC)

Version 1: Add a 1-norm penalty to obtain sparse u (or Δu)

Control the constrained discrete-time LTI system

$$x(k+1) = Ax(k) + Bu(k),$$

$$x(k) \in \mathbb{X}, \; u(k) \in \mathbb{U}, \; \forall k \geq 0,$$

by applying the first move of the optimal sequence

$$\underline{\mathrm{u}}^{\star} = \arg\min_{\underline{\mathrm{u}}} \hat{x}_N^T P \hat{x}_N \qquad + \sum_{j=0}^{N-1} \hat{x}_j^T Q \hat{x}_j + \hat{u}_j^T R \hat{u}_j + \|S\hat{u}_j\|_1 \quad \text{s.t. constraints.}$$

ℓ_{asso}-MPC (or ℓ_1-regularised MPC)

Version 2: Add a second 1-norm penalty for stability (difficult conditions)

Control the constrained discrete-time LTI system

$$x(k+1) = Ax(k) + Bu(k),$$

$$x(k) \in \mathbb{X},\ u(k) \in \mathbb{U},\ \forall k \geq 0,$$

by applying the first move of the optimal sequence[1]

$$\underline{u}^\star = \arg\min_{\underline{u}} \hat{x}_N^T P \hat{x}_N + \|Z\hat{x}_N\|_1 + \sum_{j=0}^{N-1} \hat{x}_j^T Q \hat{x}_j + \hat{u}_j^T R \hat{u}_j + \|S\hat{u}_j\|_1 \quad \text{s.t. constraints.}$$

[1] M. Gallieri and J. M. Maciejowski, *Stabilising Terminal Cost and Terminal Controller for !asso-MPC: Enhanced Optimality and Region of Attraction*, ECC 2013

ℓ_{asso}-MPC (or ℓ_1-regularised MPC)

Asymptotic Stability: How to choose the terminal cost?

Apply the first move of the optimal sequence

$$\underline{\mathrm{u}}^\star = \arg\min_{\underline{\mathrm{u}}} F(\hat{x}_N) + \sum_{j=0}^{N-1} \hat{x}_j^T Q \hat{x}_j + \hat{u}_j^T R \hat{u}_j + \|S \hat{u}_j\|_1$$

$$\text{s.t. } \hat{x}_j \in \mathbb{X},\ \hat{u}_j \in \mathbb{U},\ \forall j = 0,\ 1, \ldots, N-1,\ \hat{x}_N \in \mathbb{X}_f.$$

MPC stability assumption (Mayne et al., 2000)

- $F((A+BK)\hat{x}_N) - F(\hat{x}_N) \le -\ell(\hat{x}_N, K\hat{x}_N)$, for some K, $\forall \hat{x}_N \in \mathbb{X}_f$.
- $\mathbb{X}_f$ is *positively invariant* under $u(k) = Kx(k)$.

ℓ_{asso}-MPC (or ℓ_1-regularised MPC)

Asymptotic Stability: How to choose the terminal cost?

MPC stability assumption (Mayne et al., 2000)

- $F((A+BK)\hat{x}_N) - F(\hat{x}_N) \leq -\ell(\hat{x}_N, K\hat{x}_N)$, for some K, $\forall \hat{x}_N \in \mathbb{X}_f$.
- $\mathbb{X}_f$ is positively invariant under $u(k) = Kx(k)$.

Two possible strategies

- *Partial regularisation*: $S = [0 \;\; \bar{S}]$, $F(x) = x^T P x$. Prioritised actuators (first actuators stabilise the plant).
- *New terminal cost*: Using Minkowski functions. Multiple levels of priority.

MPC stability assumption (Mayne et al., 2000)

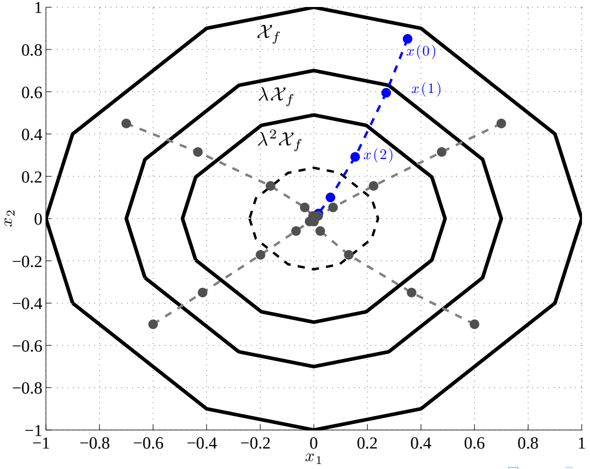
- $F((A+BK)\hat{x}_N) - F(\hat{x}_N) \leq -\ell(\hat{x}_N, K\hat{x}_N)$, for some K, $\forall \hat{x}_N \in \mathbb{X}_f$.
- $\mathbb{X}_f$ is positively invariant under $u(k) = Kx(k)$.

New terminal cost

- $F(\hat{x}_N) = \alpha\ \psi_{\mathbb{X}_f}(\hat{x}_N) + \beta\ \psi^2_{\mathbb{X}_f}(\hat{x}_N)$.
- $\psi_{\mathbb{X}_f}(x)$ is the Minkowski function of $\mathbb{X}_f$.
- $\mathbb{X}_f$ is λ-contractive : $\psi_{\mathbb{X}_f}((A+BK)\hat{x}_N) \leq \lambda\ \psi_{\mathbb{X}_f}(\hat{x}_N)$, $\forall \hat{x}_N \in \mathbb{X}_f$.
- $\alpha,\ \beta > 0$ are chosen to satisfy the required inequality.

Contractive sets

At the next time step the system state is in a scaled subset of $\mathbb{X}_f$



LASSO MPC for Tracking

Basic setting

Tracking MPC paradigm (after Ferramosca et al., Limon et al.)

- Virtual steady states: $z_s = (x_s, u_s) = \begin{bmatrix} M_x \\ M_u \end{bmatrix} \theta$.
- Penalise deviation from z_s, $\delta z = (\delta x, \delta u)$, in usual ℓ_{asso}-MPC cost.
- Add a reference penalty to the MPC cost, $V_O(z_r - z_s)$.
- Needs an *invariant set for tracking*.

Proposed terminal set for tracking

- Extended autonomous system: states $(\delta x, \theta)$. Assumes $u = K\delta x + M_u\theta$. Couplings in the constraints.
- Terminal set is "λ-contractive on δx". Obtained using $\frac{1}{\lambda}(A + BK)$.

Soft-constraints and robust stability

Softened constraints (after Zeilinger et al., Kerrigan et al.)

- The state constraints $\mathbb{X} = \{x : Lx \leq \underline{1}\}$ are "relaxed", by using slack variables. A penalty function is added to the MPC cost.

$$L\hat{x}_j \leq \underline{1} + s_j, \;\; V_s(\underline{s}) = \Sigma_{j=0}^{N-1} Q_s \|s_j\|_2^2 + R_s \|s_j\|_1.$$

- The MPC can be feasible outside $\mathbb{X}$. Recover from disturbance action. Input-to-State Stability (ISS).

Soft-constraints and robust stability

Proposed results

- Robust feasibility, bound for the disturbance $\|w\|_\infty \leq \mu$,

$$\mu \leq \frac{1-\lambda}{\|GA^{N-1}B_w\|_\infty}.$$

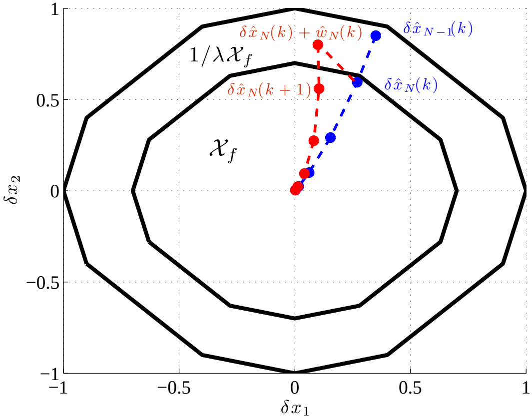
- Local ISS gain

$$V^\star_{\text{MPC}}(\delta x(k+1)) - V^\star_{\text{MPC}}(\delta x(k)) \leq -\bar{\alpha}(\|\delta x(k)\|) + \sigma(\|w(k)\|).$$

- DOA can be larger than using standard robust MPC design.

Robust feasibility - Basic idea

At the next time step the predictions can be steered back to $\mathbb{X}_f$



Robust terminal set for $\|w\|_\infty \leq \mu$

Restricting only $\delta x \Rightarrow$ tracking "all" steady states. Possible constraint violation

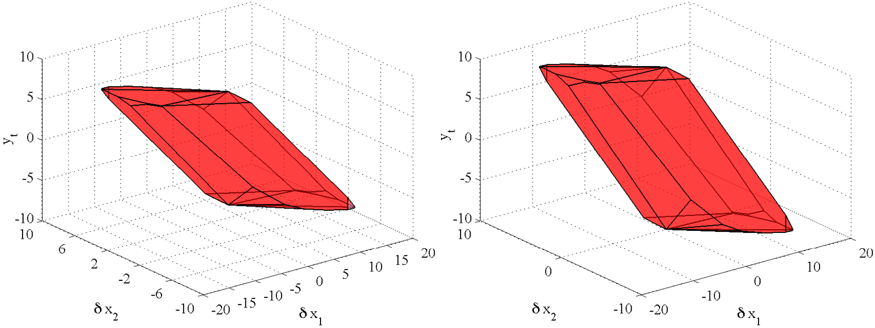
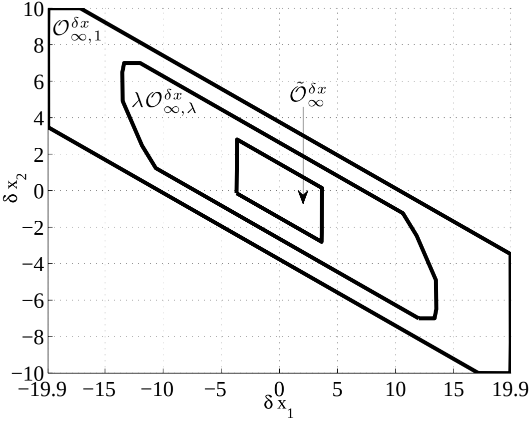


Figure: Proposed method (left). Restricting only δx (right).

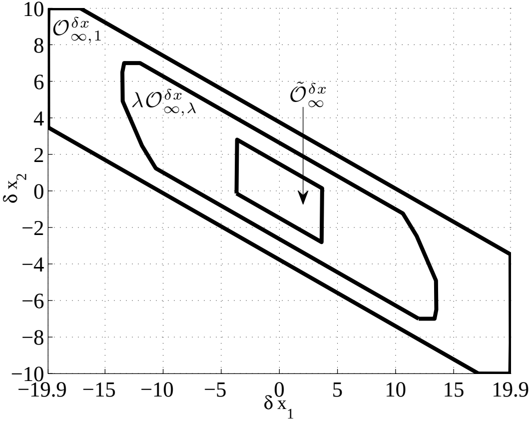
Robust terminal set for $\|w\|_\infty \leq \mu$

Comparison with constraints restriction



Robust terminal set for $\|w\|_\infty \leq \mu$

Comparison with constraints restriction



Constraints restriction

- $G_i x \leq 1 - \max G_i \phi_k(w)$.
- Input: μ, Output: $\mathbb{X}_f$.
- Robust constraints satisfaction.

Proposed approach

- Nominal complexity.
- In: λ, Out: $\mathbb{X}_f$, μ.
- Recovers from constraints violation.

Robust terminal set for $\|w\|_\infty \leq \mu$

Comparison with constraints restriction

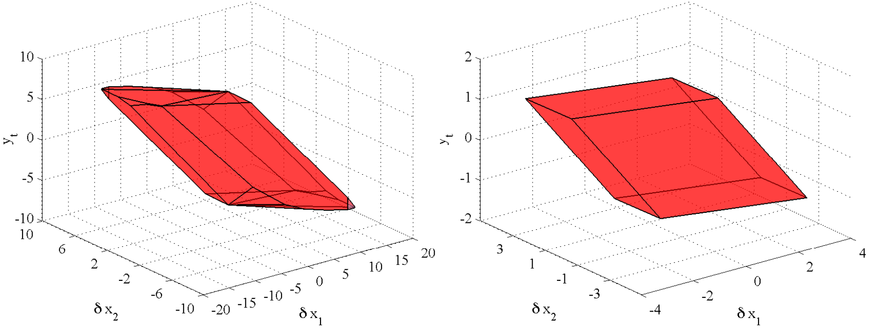


Figure: Proposed method (left). Constraints restriction (right).

Example: Double integrator

Comparison with constraints restriction

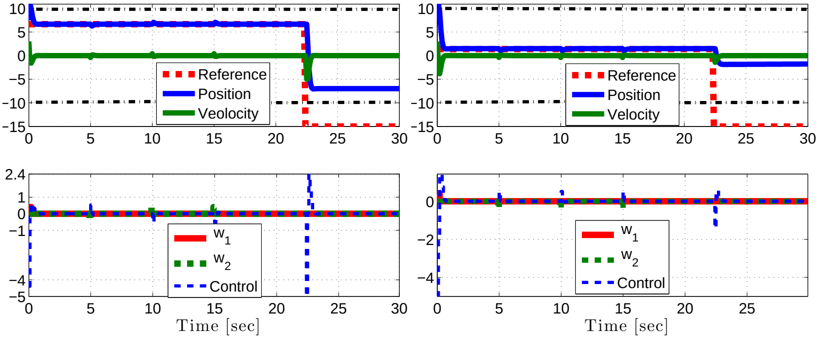


Figure: Proposed method (left). Constraints restriction (right).

Example: Double integrator

Disturbance rejection for different regularisation parameters ($\|Su\|_1$)

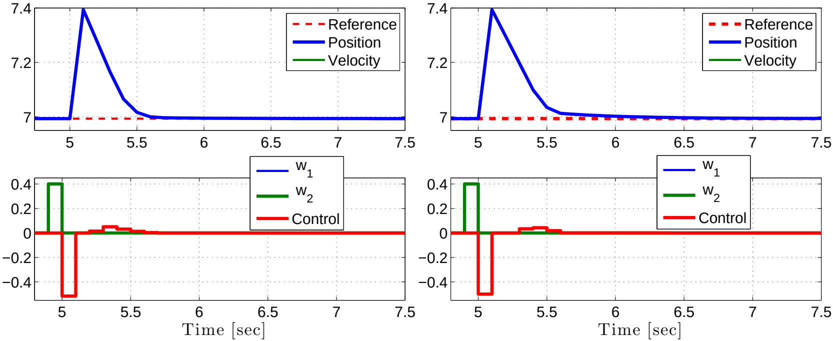


Figure: $S = 1 \Rightarrow$ fast (left). $S = 100 \Rightarrow$ cheap (right). $\|w\|_\infty = \mu$.

Robust terminal set for $\|w\|_\infty \leq \mu$

Restricting only $\delta x \Rightarrow$ tracking "all" steady states. Possible constraint violation

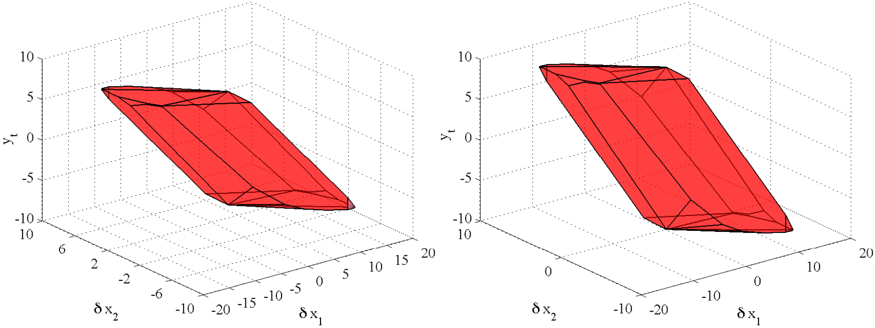


Figure: Proposed method (left). Restricting only δx (right).

Example: Double integrator

Response to constraint violation for different regularisation parameters ($\|Su\|_1$)

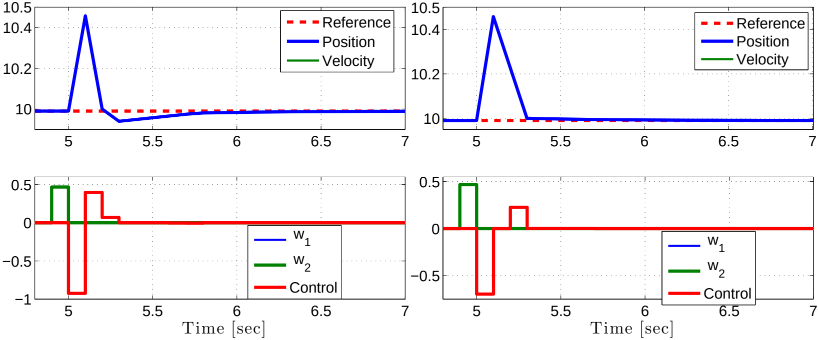


Figure: $S = 1 \Rightarrow$ fast (left). $S = 100 \Rightarrow$ cheap (right). $\|w\|_\infty = \mu$.

Thank you for the attention!
Any questions?