Controller Coefficient Truncation Using Lyapunov Performance Certificate

Joëlle Skaf   Stephen Boyd

Information Systems Laboratory
Electrical Engineering Department
Stanford University

European Control Conference, Kos, Greece, July 2007

Control system description

Consider a discrete-time linear time-invariant control system, with plant

$$\begin{aligned} x_p(t+1) &= A_p x_p(t) + B_1 w(t) + B_2 u(t) \\ z(t) &= C_1 x_p(t) + D_{11} w(t) + D_{12} u(t) \\ y(t) &= C_2 x_p(t) + D_{21} w(t) \end{aligned}$$

and controller

$$\begin{aligned} x_c(t+1) &= A_c x_c(t) + B_c y(t) \\ u(t) &= C_c x_c(t) + D_c y(t) \end{aligned}$$

Nominal and acceptable controllers

- design parameters or coefficients in controller $\theta \in \mathbf{R}^N$ (typically entries of A_c, B_c, C_c and D_c)

- given:
  - set of acceptable controller designs $\mathcal{C} \subseteq \mathbf{R}^N$ (controllers that achieve given performance specifications)
  - *nominal* controller design $\theta^{\mathrm{nom}} \in \mathcal{C}$

Controller (coefficient) complexity

- $\Phi(\theta)$ is *complexity* of controller described by θ

$$\Phi(\theta) = \sum_{i=1}^{N} \phi_i(\theta_i),$$

  where $\phi_i(\theta_i)$ gives the complexity of the ith coefficient of θ

- $\phi(z)$ is number of bits needed to express the fractional part of the binary expansion of z

The controller coefficient truncation problem

- *controller coefficient truncation problem* (CCTP): find lowest complexity controller among acceptable designs

$$\begin{array}{ll} \text{minimize} & \Phi(\theta) \\ \text{subject to} & \theta \in \mathcal{C} \end{array}$$

- very difficult to solve
- can be cast as combinatiorial optimization problem
- global optimization techniques ($e.g.$, branch-and-bound) can only solve small CCTP
- need an efficient method that can handle large problems

The algorithm

- initialize algorihm with nominal design
- at each step, choose an index i randomly and fix all parameters except θ_i
- use subroutine $\mathbf{interv}$ to find an interval $[l, u]$ of acceptable values for θ_i
- use subroutine $\mathbf{trunc}$ to find a value of θ_i in $[l, u]$ with lower complexity
- repeat until there is no change in θ
- run the algorithm several times, with the best controller coefficient vector found taken as the final choice

- $\mathbf{interv}(\theta, i)$ takes as input coefficient vector $\theta \in \mathcal{C}$, and coefficient index i

- it returns an interval $[l, u]$ with $\theta_i \in [l, u]$ and

$$(\theta_1, \ldots, \theta_{i-1}, z, \theta_{i+1}, \ldots, \theta_N) \in \mathcal{C} \text{ for } z \in [l, u].$$

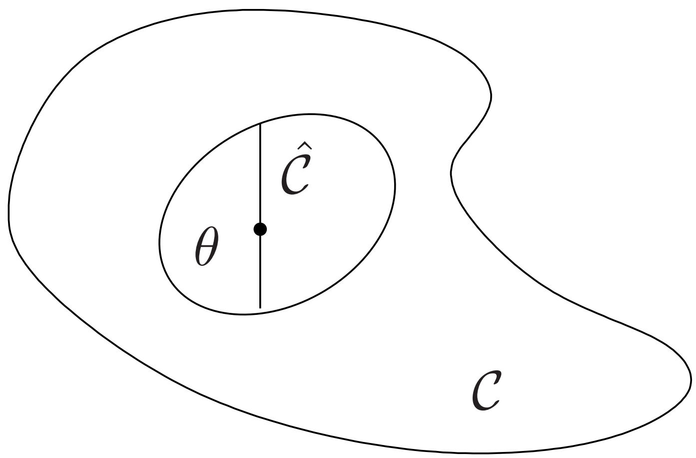
- simplest choice, always valid: return $l = u = \theta_i$

- other extreme: return largest valid interval that contains θ_i

- typical implementation of $\mathbf{interv}$: return reasonably large interval in $\mathcal{C}$, using linear matrix inequalities (LMIs)

Interval computation via Lyapunov performance certificate

given $\theta \in \mathcal{C}$, find a *convex* set $\hat{\mathcal{C}}$ such that $\theta \in \hat{\mathcal{C}} \subseteq \mathcal{C}$



- take

$$l = \inf\{z \mid (\theta_1, \ldots, \theta_{i+1}, z, \theta_{i+1}, \ldots, \theta_N) \in \hat{\mathcal{C}}\},$$
$$u = \sup\{z \mid (\theta_1, \ldots, \theta_{i+1}, z, \theta_{i+1}, \ldots, \theta_N) \in \hat{\mathcal{C}}\}.$$

- since $\hat{\mathcal{C}}$ is convex,

$$(\theta_1, \ldots, \theta_{i+1}, z, \theta_{i+1}, \ldots, \theta_N) \in \hat{\mathcal{C}} \subseteq \mathcal{C} \text{ for } z \in [l, u].$$

- use a *Lyapunov performance certificate* to find $\hat{\mathcal{C}}$

$$\theta \in \hat{\mathcal{C}} \iff \exists \nu \; L(\theta, \nu) \succeq 0$$

- L is a bi-affine function in ν and θ

$$L(\theta, \nu) = L_0 + \sum_{i=1}^{N} \theta_i L_i$$

- given $\theta \in \mathcal{C}$, compute ν such that $L(\theta,\nu) \succeq 0$

  (typically by maximizing minimum eigenvalue of $L(\theta,\nu)$ or maximizing $\det L(\theta,\nu)$)

- fix ν and take
$$\hat{\mathcal{C}} = \{\theta \mid L(\theta,\nu) \succeq 0\}$$
  ($\mathcal{C}$ is convex because described by an LMI in θ)

- need to minimize or maximize scalar variable over an LMI to find l and u

- can be reduced to an eigenvalue computation, carried out efficiently

$$\begin{aligned} l &= \theta_i - \min\{1/\lambda_i \mid \lambda_i > 0\} \\ u &= \theta_i - \max\{1/\lambda_i \mid \lambda_i < 0\} \end{aligned}$$

  where λ_i are the eigenvalues of $L(\theta,\nu)^{-1/2} L_i L(\theta,\nu)^{-1/2}$

State feedback controller with LQR cost specification

- plant is given by: $x(t+1) = Ax(t) + Bu(t), \quad x(0) = x_0$
- controlled by a state feedback gain controller: $u(t) = Kx(t)$
- design variables are entries of the matrix K
- performance measure is LQR cost

$$J(K) = \mathbf{E}\left[\sum_{t=0}^{\infty} x(t)^T Q x(t) + u(t)^T R u(t)\right] = \mathbf{Tr}(\Sigma P)$$

where P is unique solution to

$$(A+BK)^T P(A+BK) - P + Q + K^T R K = 0$$

- nominal design $K^{\rm nom}$ is the optimal state feedback controller

- set of acceptable controller designs is set of ϵ-suboptimal designs

$$\mathcal{C} = \{K \mid J(K) \leq (1+\epsilon) J^{\rm nom}\}$$

- Lyapunov performance certificate:

$$\begin{array}{rcl} P - (A+BK)^T P (A+BK) & \succeq & Q + K^T R K \\ \mathbf{Tr}(\Sigma P) & \leq & (1+\epsilon) J^{\rm nom} \\ P & \succeq & 0 \end{array}$$

- given $K \in \mathcal{C}$, take P to be the solution of

$$\begin{array}{ll} \text{maximize} & \lambda_{\min}(L(K,P)) \\ \text{subject to} & L(K,P) \succeq 0. \end{array}$$

- for a particular choice P,

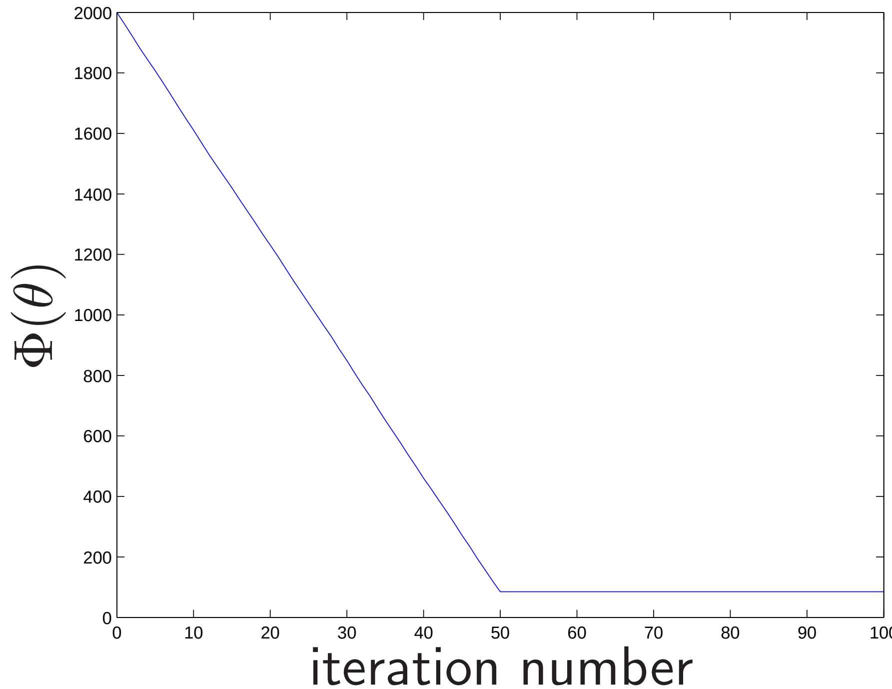
$$\hat{\mathcal{C}} = \{K \mid (A+BK)^T P(A+BK) - P + Q + K^T RK \preceq 0\}$$

- easy to show that $\hat{\mathcal{C}} \subseteq \mathcal{C}$

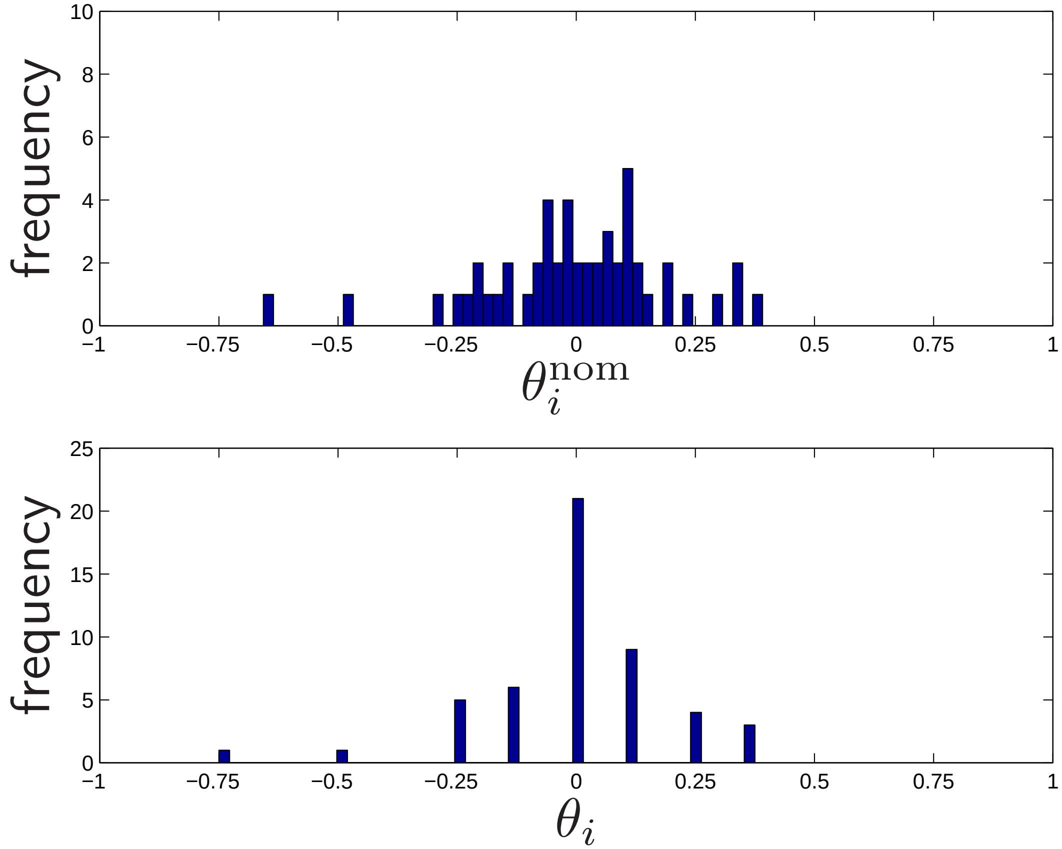
Numerical instance

- $A \in \mathbf{R}^{10\times 10}$ and $B \in \mathbf{R}^{10\times 5}$ randomly generated
- $\Sigma = I$, $Q = I$, $R = I$
- fractional part of each entry of $K^{\rm nom}$ expressed with 40 bits;
  $\Phi(K) = 2000$ bits.
- $\epsilon = 15\%$ *i.e.*, acceptable feedback controllers are up to 15%-suboptimal

total number of bits required to express K versus iteration number in a sample run of the algorihm



algorithm converges to a complexity of 85 bits in 50 iterations best design afer 100 random runs of the algorithm achieves $\Phi(K) = 75$ bits (1.5 bits per coefficient)



very aggressive coefficient truncation!

Dynamic controller with decay rate specification

- plant is given by

$$x_p(t+1) = A_p x_p(t) + B_p u(t), \quad y(t) = C_p x_p(t)$$

- controlled by a dynamic controller

$$x_c(t+1) = A_c x_c(t) + B_c y(t), \quad u(t) = C_c x_c(t)$$

- closed-loop system given by $x(t+1) = Ax(t)$ where

$$x(t) = \begin{bmatrix} x_p(t) \\ x_c(t) \end{bmatrix}, \quad A = \begin{bmatrix} A_p & B_p C_c \\ B_c C_p & A_c \end{bmatrix}$$

- design variables are entries of the controller matrices A_c, B_c and C_c
- performance measure is the decay rate of the closed-loop system: $J(A_c, B_c, C_c) = \rho(A)$
- given a nominal design $(A_c^{\rm nom}, B_c^{\rm nom}, C_c^{\rm nom})$
- set of acceptable controller designs

$$\mathcal{C} = \{(A_c, B_c, C_c) \mid J(A_c, B_c, C_c) \leq \alpha\},$$

where $\alpha = (1+\epsilon) J(A_c^{\rm nom}, B_c^{\rm nom}, C_c^{\rm nom})$

- Lyapunov performance certificate

$$\left[\begin{array}{cc} \alpha^2 P - A^T P A & 0 \\ 0 & P \end{array} \right] \succeq 0$$

- for $(A_c, B_c, C_c) \in \mathcal{C}$, take P to be the solution of

$$\begin{array}{ll} \text{maximize} & \lambda_{\min}(L(A_c, B_c, C_c, P)) \\ \text{subject to} & L(A_c, B_c, C_c, P) \succeq 0 \\ & \mathbf{Tr}(P) = 1. \end{array}$$

- for a fixed choice of P,

$$\hat{\mathcal{C}} = \{(A_c, B_c, C_c) \mid A^T P A \leq \alpha^2 P\}$$

- easy to show that $\hat{\mathcal{C}} \subseteq \mathcal{C}$

Numerical instance

- plant is given by

$$x_p(t+1) = A_p x_p(t) + B_p u(t) + w(t), \quad y(t) = C_p x_p(t) + v(t),$$

  where $w(t) \sim \mathcal{N}(0, I)$ is input noise and $v(t) \sim \mathcal{N}(0, I)$ is measurement noise

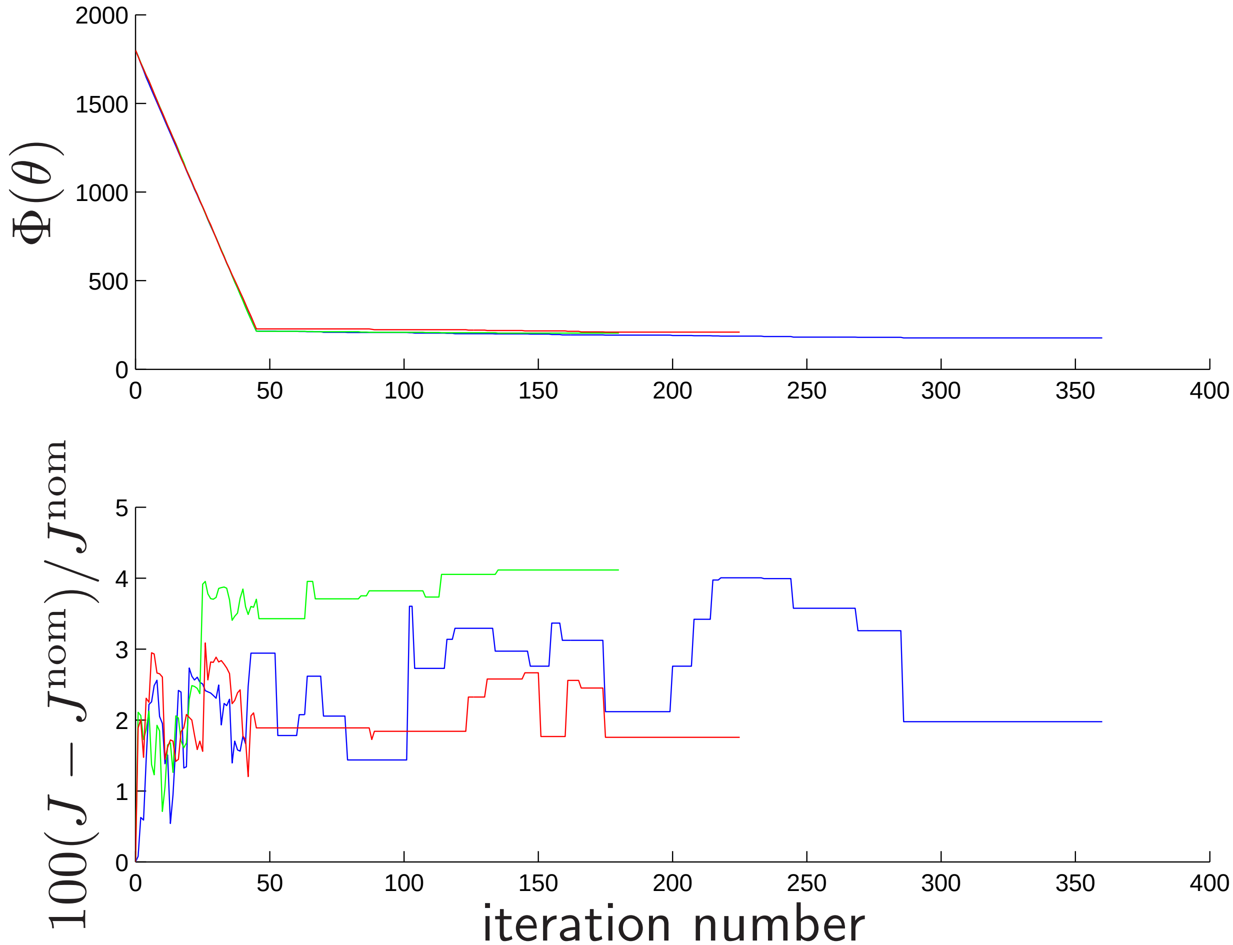
- $A_p^{\in}\mathbf{R}^{5\times 5}$, $B_p \in \mathbf{R}^{5\times 2}$, $C_p \in \mathbf{R}^{2\times 5}$ generated randomly

- plant controlled by an LQG controller with $Q = I$, $R = I$.

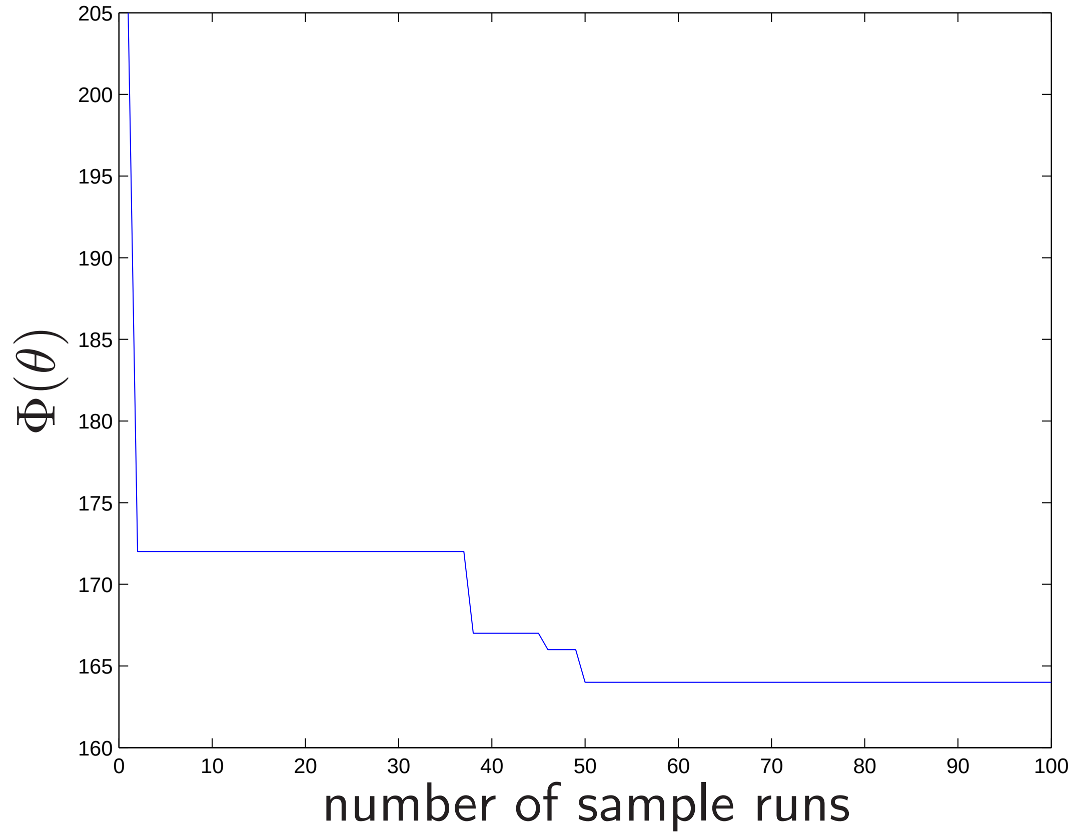
- fractional part of each entry of A_c^{nom}, B_c^{nom} and C_c^{nom} is expressed with 40 bits; $\Phi(\theta^{\mathrm{nom}}) = 1800$ bits

- $\epsilon = 5\%$

progress of the complexity $\Phi(\theta)$ and percentage deterioration in performance $100(J - J^{\text{nom}})/J^{\text{nom}}$ during 3 sample runs of the algorithm



best design complexity versus number of sample runs of the algorithm



best design after 100 random runs of the algorithm achieves a complexity of $\Phi(\theta) = 164$ bits and $J(\theta) = 1.0362 J(\theta^{\text{nom}})$