Controller Coefficient Truncation Using Lyapunov Performance Certificates

Stephen Boyd       Joëlle Skaf

Information Systems Laboratory
Electrical Engineering Department
Stanford University
Specifier/implementer interface

• specifier (control/system designer) and implementer (hardware/software/network implementer) interact through an interface

• specifier provides *reference design* and *performance certificate*

• implementer uses any method, provided implementation passes certification.

• system designer warrants that overall system will work, if implemented controller passes certification

• this work: implementing controller with fixed-point coefficients, with Lyapunov/ LMI certificate
State feedback controller with LQR cost specification

• plant: \( x(t + 1) = Ax(t) + Bu(t) \); controller: \( u(t) = Kx(t) \)

• reference controller \( K^{\text{nom}} \) is LQR optimal

• goal: minimize \( \Phi(K) \), total number of bits in \( K \), while guaranteeing LQR cost is \( \epsilon \)-suboptimal

• controller \( K \) passes certification if

\[
(1 + \epsilon)(A + BK)^T P^{\text{nom}}(A + BK) - (1 + \epsilon)P^{\text{nom}} + Q + K^T R K \succeq 0
\]

where \( P^{\text{nom}} \) is solution of DARE

\[
(A + BK^{\text{nom}})^T P(A + BK^{\text{nom}}) - P + Q + K^{\text{nom}} R K^{\text{nom}} = 0
\]
Algorithm

- initialize with nominal design (truncated to 40-bit coefficients)
- at each step choose an index pair \((i, j)\) at random; fix all other entries in \(K\)
- use Lyapunov certificate to find an interval \([l, u]\) of admissible values for \(K_{ij}\) (by solving convex optimization problem)
- truncate \(K_{ij}\) to value in interval with fewest bits
- stop when design does not change over one pass over all coefficients
- run algorithm several times; take best design found as final choice
Example run

10 states, 5 inputs; certificate requires 15\% suboptimality

The final controller is 4.1\%-suboptimal, with 6.3 bits/coeff.
Final design

best design in 100 runs is 14.9%-suboptimal, has only 1.5 bits/coeff.
Conclusions & observations

the method described

• extends to many other problems
  – dynamic controllers
  – nonlinear effects (saturation, underflow)
  – filters

• often achieves extremely aggressive coefficient truncation, with little
decrease in performance (when implementation is over parametrized)

next: handling timing errors (jitter, late/lost packets, . . . )