Users' Guide for Optimal-Distributed-Control Solver

Ghazal Fazelnia and Javad Lavaei

1 Optimal Distributed Control Solver

This solver aims to solve a convex relaxation for three types of the Optimal Distributed Control (ODC) problem: finite-horizon deterministic ODC, infinite-horizon deterministic ODC, and stochastic ODC. The finite-horizon deterministic ODC is defined as follows:

Minimize

$$\sum_{\tau=0}^{p}(x[\tau]^T Q x[\tau] + u[\tau]^T R u[\tau]) + \alpha \text{ trace}\{KK^T\} \tag{1}$$

subject to:

$$\begin{aligned} &\begin{cases} x[\tau+1] = Ax[\tau] + Bu[\tau] \\ y[\tau] = Cx[\tau] \end{cases} \qquad \tau = 0, 1, ..., p \\ &K \in \mathcal{K} \\ &\text{trace}\{KK^T\} \leq \beta \end{aligned} \tag{2}$$

where $x[\tau]$, $y[\tau]$ and $u[\tau]$ show the state, output and input of the system at time τ, respectively. In this formulation , $(\cdot)^T$ represents the transpose operator (please refer to [1, 2] for more details). This problem aims to find a static controller K such that $u[\tau] = Ky[\tau]$ for given $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{r\times n}$, $x[0] \in \mathbb{R}^n$, and terminal time p. The control structure is imposed by the subspace $\mathcal{K}$ which can be specified by a zero-one matrix L in the solver.

The above problem is called infinite-horizon deterministic ODC if $p = \infty$. The stochastic ODC is defined as below:

Minimize

$$\lim_{\tau\to+\infty} \mathcal{E}\left(x[\tau]^T Q x[\tau] + u[\tau]^T R u[\tau]\right) + \alpha \text{ trace}\{KK^T\} \tag{3}$$

subject to:

$$\begin{aligned} &\begin{cases} x[\tau+1] = Ax[\tau] + Bu[\tau] + Ed[\tau] \\ y[\tau] = Cx[\tau] + Fv[\tau] \end{cases} \qquad \tau = 0, 1, 2, ... \\ &K \in \mathcal{K} \\ &\text{trace}\{KK^T\} \leq \beta \end{aligned} \tag{4}$$

where $d[\tau]$ and $v[\tau]$ denote the input disturbance and measurement noise, which are assumed to be zero-mean white-noise random processes. The solver requires two input matrices defined as follows:

$$\Sigma_{\text{disturbance}} = \mathcal{E}\{Ed[0]d[0]^T E^T\}, \quad \Sigma_{\text{noise}} = \mathcal{E}\{Fv[0]v[0]^T F^T\} \tag{5}$$

where $\mathcal{E}\{\cdot\}$ denotes the expectation operator.

Given one of the above ODC problems, our solver designs a semidefinite programming (SDP) relaxation for the problem, from which a lower bound on the globally optimal cost of the ODC problem is found. The solution of the SDP relaxation, denoted as W, needs to have the lowest rank possible in order for the relaxation to be exact. To compensate for the rank of this matrix, the objective of the SDP relaxation can be penalized as ϵ trace$\{W\}$, where the value of ε is controlled by the user.

Our solver also attempts to design a near-global solution from the SDP solution. This will be achieved using two possible techniques named Direct Method and Indirect Method. Direct Method recovers the matrix K immediately from the SDP solution, while Indirect Method solves a second convex optimization based on the SDP solution from which a near-optimal controller is retreived. For more details, please refer to the papers [1, 2].

To use the ODC solver, you need to install CVX (http://cvxr.com/cvx/), load your data in a file named 'System_Description', and then run the file 'Solver_ODC'. You will be asked to provide a 1×4 matrix M whose entries can be interpreted as follows:

- $M(1)$ specifies the type of the ODC problem to be solved. Please set $M(1)$ to 0 for finite-horizon ODC, 1 for infinite-horizon ODC and 2 for stochastic ODC.
- $M(2)$ specifies the recovery method. Please set $M(2)$ to 0 for Direct Method and 1 for Indirect Method.
- $M(3)$ specifies the type of the SDP solver. Please set $M(3)$ to 0 for MOSEK, 1 for SDP3 and 2 for SeDuMi. For a large-scale system, it is recommend to use MOSEK (this needs a CVX professional).
- $M(4)$ specifies the coefficient ϵ for the penalization of the trace of the SDP solution.

Please open the files 'Example_MassSpring' and 'Example_PowerSystem' to test out our solver on two physical systems.

2 Four Other Convex Relaxations

Four different relaxations are presented in [3] for a finite-horizon ODC, which are associated with four different quadratic formulations of this problem. Note that the relaxations explained in the previous part are computationally cheap, while the four relaxations given in [3] may be computationally intense. If you would like to try these 4 different relaxations, you can use the code 'QCQP_four_ Relaxations' and specify the required values.

References

[1] G. Fazelnia, R. Madani, A. Kalbat, and J. Lavaei, "Convex relaxation for optimal distributed control problempart I: Time-domain formulation," *http: //www.ee.columbia.edu/~lavaei/Dec_Control_2014_PartI.pdf*, 2014.

[2] G. Fazelnia, R. Madani, A. Kalbat, and J. Lavaei, "Convex Relaxation for Optimal Distributed Control ProblemPart II: Lyapunov Formulation and Case Studies," *http://www.ee.columbia.edu/~lavaei/Dec_Control_ 2014_PartII.pdf*, 2014.

[3] G. Fazelnia, R. Madani, and J. Lavaei, "http://www.ee.columbia.edu/ ~lavaei/CDC_Decentralized_2014.pdfConvex Relaxation for Optimal Distributed Control Problem," *IEEE Conference on Decision and Control (CDC)*, 2014.