SC42100 Networked and Distributed Control Systems

Assignment 4

TU Delft, 3ME, DCSC, Spring 2022

- The assignments are individual. You may consult and discuss with your colleagues, but the answers you provide, the code you use, and the report you hand in must be independent.
- Provide detailed answers, describing the steps you followed. Only end results of calculations are not sufficient.
- Provide a clear report as a PDF file, typed, preferably using LaTeX.
- Submit the report digitally via Brightspace before the deadline: 9:00, June 24, 2022.
- You may use Matlab/Python to solve the assignment.
- Include your code, for reproducibility of your results, in your Brightspace submission on a zip file, or through a link on your report.
- The code will only be used to verify reproducibility in case of doubts. The grading will be performed based on the documentation and results described in the report.

(4+1)p **Problem 1**

Similarly to what has been seen in the lectures, we would like to solve a multi-aircraft coordination problem, where each aircraft i can be described by a linear time-invariant dynamical system

$$x_i(t+1) = Ax_i(t) + Bu_i(t), \quad x_i(0) = x_{i,0}, \quad i = 1, \dots, 4, \quad t = 0, \dots, T_{\text{final}} - 1.$$

The objective is to find a decomposition-based algorithm for coordinating towards a common target state at T_{final}, i.e., to ensure that

$$x_i(T_{\text{final}}) = x_f, \quad \forall i$$

while satisfying a limitation on the total control input

$$|u_i(t)| \leq \frac{u_{\max}}{T_{\text{final}}}, \quad \forall i, t.$$

The objective function to be minimized is the following quadratic function

$$\sum_i \sum_t x_i(t)^T x_i(t) + u_i(t)^T u_i(t).$$

The optimization variables are the (private) controls u_i of the individual aircraft, as well as the (public) common terminal state x_f. A Matlab file is attached (`aircraft.m`) with the matrices A_i and B_i along with the initial states, control limit, and horizon length for a four-aircraft example.

1p (a) Derive a solution based on dual decomposition using the projected subgradient method and demonstrate it with the use of Matlab or Python (use of cvx toolbox is allowed). Show a plot that demonstrates the convergence of the optimization process (in terms of an error sequence). Show a plot of the resulting aircraft state trajectories after convergence, and compare with the centralized optimal solution.

1p (b) Investigate the effect of step size (for constant step) and step size update sequence (for variable step) on the convergence of the standard subgradient method. Show the results using an error sequence plot.

1p (c) Implement an accelerated version (e.g., Nesterov method) of the subgradient updates. Show the results using an error sequence plot.

1p (d) Consider the standard incremental subgradient method, and the following consensus matrix between the agents:

$$W = \begin{pmatrix} 0.75 & 0.25 & 0 & 0 \\ 0.25 & 0.5 & 0.25 & 0 \\ 0 & 0.25 & 0.5 & 0.25 \\ 0 & 0 & 0.25 & 0.75 \end{pmatrix}$$

Implement a combined consensus/incremental subgradient approach and investigate the effect of the number of consensus steps in each subgradient iteration on the convergence rate.

1p (e) For extra points (and a good learning experience), consider replacing the linear input constraint with the following quadratic limitation on the total control energy

$$\sum_t u_i(t)^T u_i(t) \leq u_{\max}^2, \quad \forall i.$$

You may need to have access to solvers that are able to solve the resulting quadratically constrained quadratic programs (QCQP), such as Yalmip using the Gurobi solver.

3p Problem 2

2p (a) Consider the multi-aircraft coordination formulation in Problem 1 and implement a consensus ADMM approach (see ADMM lecture slides) to solve this problem. Show a plot that demonstrates the convergence of the optimization process (in terms of an error sequence). Show a plot of the resulting aircraft state trajectories after convergence, and compare with the centralized optimal solution.

1p (b) Investigate the effect of the ρ parameter on the convergence rate. Show the results using an error sequence plot.