Alternating Direction Method of Multipliers (ADMM)

In this tutorial, we will briefly introduce the Alternating Direction Method of Multipliers (ADMM), explore its key concepts, and show how to implement this algorithm using Python programming. ADMM is an optimization algorithm designed to solve complex problems that can be broken into smaller, easier sub-problems. Due to its advantages, ADMM has been recognized as a powerful tool for distributed optimizations and has been widely used in multi-robot coordination, motion planning, and resource allocations [1,2]. In general, ADMM works by solving each part iteratively and then coordinating their solutions through Lagrange multipliers.

Let us consider the following optimization problem:

$$\begin{aligned} &\min_{x,z} \quad f(x) + g(z) \\ &\text{s.t.} \quad h(x,z) \triangleq Ax + Bz - C = 0 \end{aligned} \tag{1}$$

where $x, z \in \mathbb{R}^n$ are decision variables to be designed and $f(x) + g(z) \in \mathbb{R}$ denotes the cost function, and $h(x,z) \in \mathbb{R}^n$ denotes a constraint function with proper matrices. Note that $h(x,z)$ is a coupling constraint regarding x and z. In the multi-robot coordination example, this constraint can be utilized for ideal coordination between robots. Next, consider the augmented Lagrangian as follows:

$$\mathcal{L}_\rho(x,z,\lambda) = f(x) + g(z) + \lambda^T (Ax + Bz - C) + \frac{\rho}{2}||Ax + Bz - C||_2^2 \tag{2}$$

where $\lambda \in \mathbb{R}^n$ denotes the Lagrangian multiplier and $\rho \in \mathbb{R}$ denotes a constant for the penalty term (i.e., $||Ax + Bz - C||_2^2$). The ADMM algorithm is conducted iteratively in the following manner:

ADMM Algorithm:

$$\begin{cases} x^{k+1} = \arg\min_{x\in\mathbb{R}^n} \mathcal{L}_\rho(x, z^k, \lambda^k), \\ z^{k+1} = \arg\min_{z\in\mathbb{R}^n} \mathcal{L}_\rho(x^{k+1}, z, \lambda^k), \\ \lambda^{k+1} = \lambda^k + \rho\left(Ax^{k+1} + Bz^{k+1} - C\right) \end{cases} \tag{3}$$

where k is an index for iterations and we solve the algorithm in (3) until solutions converge to certain values. For more details (e.g., convergence proof), readers may refer to [1].

References

[1] S. Boyd, N. Parikh, E. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends,"*Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp.1-122, 2011.
[2] A. Carron, D. Saccani, L. Fagiano, and M. N. Zeilinger, “Multi-agent distributed model predictive control with connectivity constraint,"*IFAC-PapersOnLine*, vol. 56, no. 2, pp.3806-3811, 2023.