PURDUE UNIVERSITY

ECE 68000: MODERN AUTOMATIC CONTROL

Professor Stan Żak

A Lagrangian Algorithm for Inequality Constraints

A Lagrangian Algorithm for Inequality Constraints

- A first-order Lagrangian algorithm for the optimization problem involving inequality constraints,

$$\begin{array}{ll} \text{minimize} & f(\boldsymbol{x}) \\ \text{subject to} & \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{0}, \end{array}$$

  where $\boldsymbol{g} : \mathbb{R}^N \to \mathbb{R}^P$
- The Lagrangian function is

$$l(\boldsymbol{x}, \boldsymbol{\mu}) = f(\boldsymbol{x}) + \boldsymbol{\mu}^\top \boldsymbol{g}(\boldsymbol{x})$$

Iterative first-order Lagrangian algorithm

- The first-order Lagrangian algorithm for the above optimization problem involving minimizing f subject to the inequality constraints, $\boldsymbol{g}(\boldsymbol{x}) \leq \mathbf{0}$,

$$\begin{aligned}
\boldsymbol{x}^{[k+1]} &= \boldsymbol{x}^{[k]} - \alpha_k \left(\nabla f\left(\boldsymbol{x}^{[k]}\right) + D\boldsymbol{g}\left(\boldsymbol{x}^{[k]}\right)^\top \boldsymbol{\mu}^{[k]} \right) \\
\boldsymbol{\mu}^{[k+1]} &= \left[\boldsymbol{\mu}^{[k]} + \beta_k \boldsymbol{g}\left(\boldsymbol{x}^{[k]}\right)\right]_+ ,
\end{aligned}$$

  where the operation $[\cdot]_+ = \max(\cdot, 0)$ is applied component-wise

Using the Lagrangian algorithm in MPC implementation

- In our application to the MPC construction, the Lagrangian function is

$$l(\Delta \boldsymbol{U}, \boldsymbol{\mu}) = J\left(\Delta \boldsymbol{U}\right) + \boldsymbol{\mu}^{\top} \boldsymbol{g}\left(\Delta \boldsymbol{U}\right).$$

- The gradient of J with respect to $\Delta \boldsymbol{U}$

$$\nabla J\left(\Delta \boldsymbol{U}\right) = -\boldsymbol{Z}^{\top} \boldsymbol{Q}\left(\boldsymbol{r}_p - \boldsymbol{W}\boldsymbol{x}_a - \boldsymbol{Z}\Delta \boldsymbol{U}\right) + \boldsymbol{R}\Delta \boldsymbol{U}.$$

- Suppose that we impose constraints on the plant output
- Then, the function $\boldsymbol{g}$ that represents these inequality constraints takes the form

$$\boldsymbol{g}\left(\Delta \boldsymbol{U}\right) = \begin{bmatrix} -\boldsymbol{Z} \\ \boldsymbol{Z} \end{bmatrix} \Delta \boldsymbol{U} - \begin{bmatrix} -\boldsymbol{Y}^{\min} + \boldsymbol{W}\boldsymbol{x}_a[k] \\ \boldsymbol{Y}^{\max} - \boldsymbol{W}\boldsymbol{x}_a[k] \end{bmatrix}$$

Algorithm implementation

- The gradient of $\boldsymbol{\mu}^\top \boldsymbol{g}$ with respect to $\Delta \boldsymbol{U}$ is

$$\nabla \left(\boldsymbol{\mu}^\top \boldsymbol{g}\right) = \begin{bmatrix} -\boldsymbol{Z} \\ \boldsymbol{Z} \end{bmatrix}^\top \boldsymbol{\mu}.$$

- The first-order Lagrangian algorithm takes the form

$$\begin{aligned} \Delta \boldsymbol{U}^{(i+1)} &= \Delta \boldsymbol{U}^{(i)} - \alpha_i \left(\nabla J \left(\Delta \boldsymbol{U}^{(i)}\right) + \nabla \left(\boldsymbol{\mu}(\boldsymbol{i})^\top \boldsymbol{g}(\boldsymbol{i})\right)\right) \\ \boldsymbol{\mu}^{(i+1)} &= \left[\boldsymbol{\mu}^{(i)} + \beta_i \boldsymbol{g}\left(\Delta \boldsymbol{U}^{(i)}\right)\right]_+ \end{aligned}$$