Sparsity Methods for Systems and Control

Numerical Optimization by Time Discretization

Masaaki Nagahara[1]

[1]The University of Kitakyushu
nagahara@ieee.org

Table of Contents

Table of Contents

L^1-optimal control problem

L^1-optimal control problem

For the linear time-invariant system

$$\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + \boldsymbol{b}u(t), \quad t \geq 0, \quad \boldsymbol{x}(0) = \boldsymbol{\xi} \in \mathbb{R}^d, \tag{A}$$

find a control $\{u(t) : t \in [0, T]\}$ with $T > 0$ that minimizes

$$J_1(u) = \|u\|_1 = \int_0^T |u(t)|dt \tag{B}$$

subject to

$$\boldsymbol{x}(T) = \boldsymbol{0}, \tag{C}$$

and

$$\|u\|_\infty \leq 1. \tag{D}$$

Time discretization of control problem

$$\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + \boldsymbol{b}u(t), \quad t \geq 0, \quad \boldsymbol{x}(0) = \boldsymbol{\xi} \in \mathbb{R}^d, \tag{A}$$

$$J_1(u) = \|u\|_1 = \int_0^T |u(t)| dt \tag{B}$$

$$\boldsymbol{x}(T) = \boldsymbol{0}, \tag{C}$$

$$\|u\|_\infty \leq 1. \tag{D}$$

- Discretize the time axis (time discretization)
- Obtain a discrete-time system representation (A′) for (A)
- Discretize the objective function (B)
- Represent terminal condition (C) for the discrete-time system (A′)
- Obtain magnitude condition (D) for the discrete-time system (A′)

Time discretization of control problem

$$\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + \boldsymbol{b}u(t), \quad t \geq 0, \quad \boldsymbol{x}(0) = \boldsymbol{\xi} \in \mathbb{R}^d, \tag{A}$$

$$J_1(u) = \|u\|_1 = \int_0^T |u(t)|dt \tag{B}$$

$$\boldsymbol{x}(T) = \boldsymbol{0}, \tag{C}$$

$$\|u\|_\infty \leq 1. \tag{D}$$

- Discretize the time axis (time discretization)
- Obtain a discrete-time system representation (A′) for (A)
- Discretize the objective function (B)
- Represent terminal condition (C) for the discrete-time system (A′)
- Obtain magnitude condition (D) for the discrete-time system (A′)

Time discretization of control problem

$$\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + \boldsymbol{b}u(t), \quad t \geq 0, \quad \boldsymbol{x}(0) = \boldsymbol{\xi} \in \mathbb{R}^d, \tag{A}$$

$$J_1(u) = \|u\|_1 = \int_0^T |u(t)| dt \tag{B}$$

$$\boldsymbol{x}(T) = \boldsymbol{0}, \tag{C}$$

$$\|u\|_\infty \leq 1. \tag{D}$$

- Discretize the time axis (time discretization)
- Obtain a discrete-time system representation (A′) for (A)
- Discretize the objective function (B)
- Represent terminal condition (C) for the discrete-time system (A′)
- Obtain magnitude condition (D) for the discrete-time system (A′)

Time discretization of control problem

$$\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + \boldsymbol{b}u(t), \quad t \geq 0, \quad \boldsymbol{x}(0) = \boldsymbol{\xi} \in \mathbb{R}^d, \tag{A}$$

$$J_1(u) = \|u\|_1 = \int_0^T |u(t)|dt \tag{B}$$

$$\boldsymbol{x}(T) = \boldsymbol{0}, \tag{C}$$

$$\|u\|_\infty \leq 1. \tag{D}$$

- Discretize the time axis (time discretization)
- Obtain a discrete-time system representation (A′) for (A)
- Discretize the objective function (B)
- Represent terminal condition (C) for the discrete-time system (A′)
- Obtain magnitude condition (D) for the discrete-time system (A′)

Time discretization of control problem

$$\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + \boldsymbol{b}u(t), \quad t \geq 0, \quad \boldsymbol{x}(0) = \boldsymbol{\xi} \in \mathbb{R}^d, \tag{A}$$

$$J_1(u) = \|u\|_1 = \int_0^T |u(t)| dt \tag{B}$$

$$\boldsymbol{x}(T) = \boldsymbol{0}, \tag{C}$$

$$\|u\|_\infty \leq 1. \tag{D}$$

- Discretize the time axis (time discretization)
- Obtain a discrete-time system representation (A′) for (A)
- Discretize the objective function (B)
- Represent terminal condition (C) for the discrete-time system (A′)
- Obtain magnitude condition (D) for the discrete-time system (A′)

Time discretization of control problem

$$\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + \boldsymbol{b}u(t), \quad t \geq 0, \quad \boldsymbol{x}(0) = \boldsymbol{\xi} \in \mathbb{R}^d, \tag{A}$$

$$J_1(u) = \|u\|_1 = \int_0^T |u(t)|dt \tag{B}$$

$$\boldsymbol{x}(T) = \boldsymbol{0}, \tag{C}$$

$$\|u\|_\infty \leq 1. \tag{D}$$

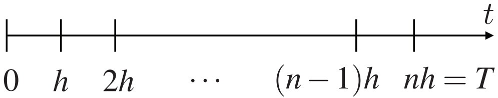
- Discretize the time axis (time discretization)
- Obtain a discrete-time system representation (A′) for (A)
- Discretize the objective function (B)
- Represent terminal condition (C) for the discrete-time system (A′)
- Obtain magnitude condition (D) for the discrete-time system (A′)

Time discretization

- Discretize the time interval $[0, T]$ into n subintervals as

$$[0, T] = [0, h) \cup [h, 2h) \cup \cdots \cup [nh - h, nh],$$

- $h > 0$ is the sampling time and $n \in \mathbb{N}$ is the number of subintervals such that $T = nh$.

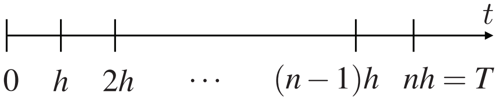


Time discretization

- Discretize the time interval $[0, T]$ into n subintervals as

$$[0, T] = [0, h) \cup [h, 2h) \cup \cdots \cup [nh - h, nh],$$

  - $h > 0$ is the sampling time and $n \in \mathbb{N}$ is the number of subintervals such that $T = nh$.



Zero-order hold assumption

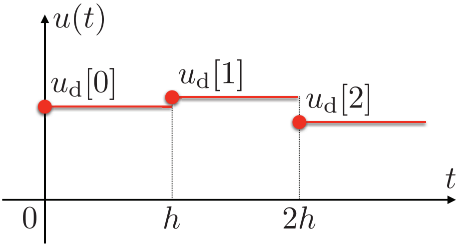
- We assume that on each subinterval, we assume the control $u(t)$ is constant.
- Namely, we assume

$$u(t) = u(kh) = u_{\mathrm{d}}[k], \quad t \in [kh, (k+1)h), \quad k = 0, 1, 2, \dots, n-1.$$

- This is the output of the zero-order hold of a discrete-time signal

$$u_{\mathrm{d}} \triangleq \{u_{\mathrm{d}}[0], u_{\mathrm{d}}[1], \dots, u_{\mathrm{d}}[n-1]\}.$$

- This assumption is reasonable for digital control systems.



Zero-order hold assumption

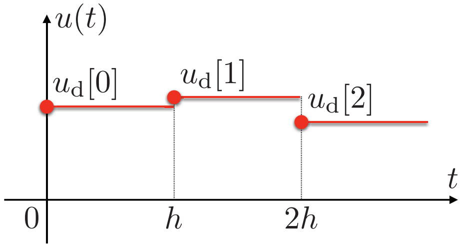
- We assume that on each subinterval, we assume the control $u(t)$ is constant.
- Namely, we assume

$$u(t) = u(kh) = u_{\mathrm{d}}[k], \quad t \in [kh, (k+1)h), \quad k = 0, 1, 2, \dots, n-1.$$

- This is the output of the zero-order hold of a discrete-time signal

$$u_{\mathrm{d}} \triangleq \{u_{\mathrm{d}}[0], u_{\mathrm{d}}[1], \dots, u_{\mathrm{d}}[n-1]\}.$$

- This assumption is reasonable for digital control systems.



Zero-order hold assumption

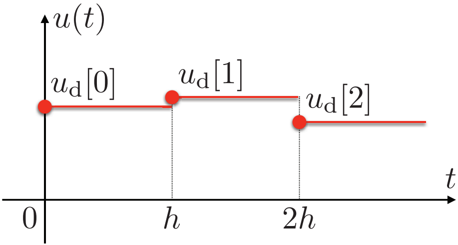
- We assume that on each subinterval, we assume the control $u(t)$ is constant.
- Namely, we assume

$$u(t) = u(kh) = u_{\mathrm{d}}[k], \quad t \in [kh, (k+1)h), \quad k = 0, 1, 2, \ldots, n-1.$$

- This is the output of the zero-order hold of a discrete-time signal

$$u_{\mathrm{d}} \triangleq \{u_{\mathrm{d}}[0], u_{\mathrm{d}}[1], \ldots, u_{\mathrm{d}}[n-1]\}.$$

- This assumption is reasonable for digital control systems.



Discretizing state-space equation

- The solution of the state-space equation is given by

$$\boldsymbol{x}(t_1) = e^{A(t_1-t_0)}\boldsymbol{x}(t_0) + \int_{t_0}^{t_1} e^{A(t_1-\tau)}\boldsymbol{b}u(\tau)d\tau,$$

  where $0 \leq t_0 \leq t_1$.

- Take

$$t_0 = kh, \quad t_1 = kh + h, \quad k \in \{0, 1, 2, \ldots, n-1\}.$$

- Then we have

$$\begin{aligned}\boldsymbol{x}(kh+h) &= e^{Ah}\boldsymbol{x}(kh) + \int_{kh}^{kh+h} e^{A(kh+h-\tau)}\boldsymbol{b}u(\tau)d\tau \\ &= e^{Ah}\boldsymbol{x}(kh) + \int_{0}^{h} e^{A(h-t)}\boldsymbol{b}u(t+kh)dt.\end{aligned}$$

Discretizing state-space equation

- The solution of the state-space equation is given by

$$\boldsymbol{x}(t_1) = e^{A(t_1-t_0)}\boldsymbol{x}(t_0) + \int_{t_0}^{t_1} e^{A(t_1-\tau)}\boldsymbol{b}u(\tau)d\tau,$$

  where $0 \leq t_0 \leq t_1$.
- Take

$$t_0 = kh, \quad t_1 = kh + h, \quad k \in \{0, 1, 2, \dots, n-1\}.$$

- Then we have

$$\begin{aligned}\boldsymbol{x}(kh+h) &= e^{Ah}\boldsymbol{x}(kh) + \int_{kh}^{kh+h} e^{A(kh+h-\tau)}\boldsymbol{b}u(\tau)d\tau \\ &= e^{Ah}\boldsymbol{x}(kh) + \int_{0}^{h} e^{A(h-t)}\boldsymbol{b}u(t+kh)dt.\end{aligned}$$

Discretizing state-space equation

- The solution of the state-space equation is given by

$$\boldsymbol{x}(t_1) = e^{A(t_1-t_0)}\boldsymbol{x}(t_0) + \int_{t_0}^{t_1} e^{A(t_1-\tau)}\boldsymbol{b}u(\tau)d\tau,$$

where $0 \leq t_0 \leq t_1$.

- Take

$$t_0 = kh, \quad t_1 = kh + h, \quad k \in \{0, 1, 2, \dots, n-1\}.$$

- Then we have

$$\begin{aligned}\boldsymbol{x}(kh+h) &= e^{Ah}\boldsymbol{x}(kh) + \int_{kh}^{kh+h} e^{A(kh+h-\tau)}\boldsymbol{b}u(\tau)d\tau \\ &= e^{Ah}\boldsymbol{x}(kh) + \int_0^h e^{A(h-t)}\boldsymbol{b}u(t+kh)dt.\end{aligned}$$

Discretizing state-space equation

- Define

$$\boldsymbol{x}_{\mathrm{d}}[k] \triangleq \boldsymbol{x}(kh), \quad u_{\mathrm{d}}[k] \triangleq u(kh), \quad k = 0, 1, \dots, n-1,$$

and

$$\boldsymbol{x}_{\mathrm{d}}[n] \triangleq \boldsymbol{x}(T).$$

- From the zero-order-hold assumption, we have

$$\boldsymbol{x}_{\mathrm{d}}[k+1] = e^{Ah}\boldsymbol{x}_{\mathrm{d}}[k] + \left(\int_0^h e^{A(h-t)}\boldsymbol{b}\,dt\right) u_{\mathrm{d}}[k].$$

- or

$$\boldsymbol{x}_{\mathrm{d}}[k+1] = A_{\mathrm{d}}\boldsymbol{x}_{\mathrm{d}}[k] + \boldsymbol{b}_{\mathrm{d}} u_{\mathrm{d}}[k], \quad k = 0, 1, \dots, n-1,$$

- This is the zero-order hold discretization of the continuous-time state-space equation.

Discretizing state-space equation

- Define

$$\boldsymbol{x}_{\mathrm{d}}[k] \triangleq \boldsymbol{x}(kh), \quad u_{\mathrm{d}}[k] \triangleq u(kh), \quad k = 0, 1, \dots, n-1,$$

and

$$\boldsymbol{x}_{\mathrm{d}}[n] \triangleq \boldsymbol{x}(T).$$

- From the zero-order-hold assumption, we have

$$\boldsymbol{x}_{\mathrm{d}}[k+1] = e^{Ah}\boldsymbol{x}_{\mathrm{d}}[k] + \left(\int_0^h e^{A(h-t)} \boldsymbol{b}\, dt \right) u_{\mathrm{d}}[k].$$

- or

$$\boldsymbol{x}_{\mathrm{d}}[k+1] = A_{\mathrm{d}}\boldsymbol{x}_{\mathrm{d}}[k] + \boldsymbol{b}_{\mathrm{d}} u_{\mathrm{d}}[k], \quad k = 0, 1, \dots, n-1,$$

- This is the zero-order hold discretization of the continuous-time state-space equation.

Discretizing state-space equation

- Define

$$x_d[k] \triangleq x(kh), \quad u_d[k] \triangleq u(kh), \quad k = 0, 1, \ldots, n-1,$$

and

$$x_d[n] \triangleq x(T).$$

- From the zero-order-hold assumption, we have

$$x_d[k+1] = e^{Ah} x_d[k] + \left(\int_0^h e^{A(h-t)} b \, dt \right) u_d[k].$$

- or

$$x_d[k+1] = A_d x_d[k] + b_d u_d[k], \quad k = 0, 1, \ldots, n-1,$$

- This is the zero-order hold discretization of the continuous-time state-space equation.

Discretizing state-space equation

- Define

$$\boldsymbol{x}_{\rm d}[k] \triangleq \boldsymbol{x}(kh), \quad u_{\rm d}[k] \triangleq u(kh), \quad k = 0, 1, \ldots, n-1,$$

and

$$\boldsymbol{x}_{\rm d}[n] \triangleq \boldsymbol{x}(T).$$

- From the zero-order-hold assumption, we have

$$\boldsymbol{x}_{\rm d}[k+1] = e^{Ah}\boldsymbol{x}_{\rm d}[k] + \left(\int_0^h e^{A(h-t)}\boldsymbol{b}\,dt\right) u_{\rm d}[k].$$

- or

$$\boldsymbol{x}_{\rm d}[k+1] = A_{\rm d}\boldsymbol{x}_{\rm d}[k] + \boldsymbol{b}_{\rm d}u_{\rm d}[k], \quad k = 0, 1, \ldots, n-1,$$

- This is the zero-order hold discretization of the continuous-time state-space equation.

Terminal constraint

- Define the control vector

$$\boldsymbol{u} \triangleq \begin{bmatrix} u_{\mathrm{d}}[0] \\ u_{\mathrm{d}}[1] \\ \vdots \\ u_{\mathrm{d}}[n-1] \end{bmatrix} \in \mathbb{R}^n.$$

- Then the terminal state $\boldsymbol{x}(T)$ of the continuous-time state-space equation is now represented by

$$\boldsymbol{x}(T) = \boldsymbol{x}_{\mathrm{d}}[n] = -\boldsymbol{\zeta} + \Phi \boldsymbol{u}.$$

where

$$\Phi \triangleq \begin{bmatrix} A_{\mathrm{d}}^{n-1}\boldsymbol{b}_{\mathrm{d}} & A_{\mathrm{d}}^{n-2}\boldsymbol{b}_{\mathrm{d}} & \dots & \boldsymbol{b}_{\mathrm{d}} \end{bmatrix}, \quad \boldsymbol{\zeta} \triangleq -A_{\mathrm{d}}^n \boldsymbol{\xi}.$$

- Therefore, the terminal constraint is

$$-\boldsymbol{\zeta} + \Phi \boldsymbol{u} = \boldsymbol{0}.$$

Terminal constraint

- Define the control vector

$$\boldsymbol{u} \triangleq \begin{bmatrix} u_{\mathrm{d}}[0] \\ u_{\mathrm{d}}[1] \\ \vdots \\ u_{\mathrm{d}}[n-1] \end{bmatrix} \in \mathbb{R}^n.$$

- Then the terminal state $\boldsymbol{x}(T)$ of the continuous-time state-space equation is now represented by

$$\boldsymbol{x}(T) = \boldsymbol{x}_{\mathrm{d}}[n] = -\zeta + \Phi \boldsymbol{u}.$$

where

$$\Phi \triangleq \begin{bmatrix} A_{\mathrm{d}}^{n-1}\boldsymbol{b}_{\mathrm{d}} & A_{\mathrm{d}}^{n-2}\boldsymbol{b}_{\mathrm{d}} & \dots & \boldsymbol{b}_{\mathrm{d}} \end{bmatrix}, \quad \zeta \triangleq -A_{\mathrm{d}}^{n}\boldsymbol{\xi}.$$

- Therefore, the terminal constraint is

$$-\zeta + \Phi \boldsymbol{u} = \mathbf{0}.$$

Terminal constraint

- Define the control vector

$$\boldsymbol{u} \triangleq \begin{bmatrix} u_{\mathrm{d}}[0] \\ u_{\mathrm{d}}[1] \\ \vdots \\ u_{\mathrm{d}}[n-1] \end{bmatrix} \in \mathbb{R}^n.$$

- Then the terminal state $\boldsymbol{x}(T)$ of the continuous-time state-space equation is now represented by

$$\boldsymbol{x}(T) = \boldsymbol{x}_{\mathrm{d}}[n] = -\boldsymbol{\zeta} + \Phi \boldsymbol{u}.$$

where

$$\Phi \triangleq \begin{bmatrix} A_{\mathrm{d}}^{n-1}\boldsymbol{b}_{\mathrm{d}} & A_{\mathrm{d}}^{n-2}\boldsymbol{b}_{\mathrm{d}} & \dots & \boldsymbol{b}_{\mathrm{d}} \end{bmatrix}, \quad \boldsymbol{\zeta} \triangleq -A_{\mathrm{d}}^{n}\boldsymbol{\xi}.$$

- Therefore, the terminal constraint is

$$-\boldsymbol{\zeta} + \Phi \boldsymbol{u} = \boldsymbol{0}.$$

Magnitude constraint of control and cost function

- The magnitude constraint $\|u\|_\infty \le 1$ is equivalently described under the zero-order hold assumption as

$$\|\boldsymbol{u}\|_{\ell^\infty} \le 1.$$

- The L^1 cost function is also described as

$$J_1(u) = \int_0^T |u(t)|dt = \sum_{k=0}^{n-1} \int_{kh}^{(k+1)h} |u(t)|dt = \sum_{k=0}^{n-1} |u_{\mathrm{d}}[k]|h = h\|\boldsymbol{u}\|_{\ell^1}.$$

Magnitude constraint of control and cost function

- The magnitude constraint $\|u\|_\infty \leq 1$ is equivalently described under the zero-order hold assumption as

$$\|\boldsymbol{u}\|_{\ell^\infty} \leq 1.$$

- The L^1 cost function is also described as

$$J_1(u) = \int_0^T |u(t)|dt = \sum_{k=0}^{n-1} \int_{kh}^{(k+1)h} |u(t)|dt = \sum_{k=0}^{n-1} |u_{\mathrm{d}}[k]|h = h\|\boldsymbol{u}\|_{\ell^1}.$$

Discretized optimization problem

- Finally, we obtain an optimization problem described by

$$\underset{\boldsymbol{u}\in\mathbb{R}^n}{\text{minimize}} \ \|\boldsymbol{u}\|_{\ell^1} \ \text{ subject to } \ \Phi\boldsymbol{u} = \zeta, \ \|\boldsymbol{u}\|_{\ell^\infty} \leq 1.$$

- This is an ℓ^1 optimization problem, which can be solved by ADMM!

Discretized optimization problem

- Finally, we obtain an optimization problem described by

$$\underset{\boldsymbol{u}\in\mathbb{R}^n}{\text{minimize}} \ \|\boldsymbol{u}\|_{\ell^1} \ \text{subject to} \ \Phi\boldsymbol{u} = \zeta, \ \|\boldsymbol{u}\|_{\ell^\infty} \leq 1.$$

- This is an ℓ^1 optimization problem, which can be solved by ADMM!

Table of Contents

ADMM formulation

ℓ^1 optimization problem

$$\underset{\boldsymbol{u}\in\mathbb{R}^n}{\text{minimize}} \ \|\boldsymbol{u}\|_{\ell^1} \ \text{subject to} \ \Phi\boldsymbol{u} = \zeta, \ \|\boldsymbol{u}\|_{\ell^\infty} \leq 1.$$

- Let's first consider the two constraints.
- Define

$$C_1 \triangleq \{\boldsymbol{u} \in \mathbb{R}^n : \|\boldsymbol{u}\|_{\ell^\infty} \leq 1\}$$
$$C_2 \triangleq \{\zeta\}$$

- Indicator functions:

$$I_{C_1}(\boldsymbol{u}) \triangleq \begin{cases} 0, & \text{if } \|\boldsymbol{u}\|_{\ell^\infty} \leq 1, \\ \infty, & \text{if } \|\boldsymbol{u}\|_{\ell^\infty} > 1, \end{cases}$$

$$I_{C_2}(\boldsymbol{x}) \triangleq \begin{cases} 0, & \text{if } \boldsymbol{x} = \zeta, \\ \infty, & \text{if } \boldsymbol{x} \neq \zeta. \end{cases}$$

ADMM formulation

ℓ^1 optimization problem

$$\underset{\boldsymbol{u}\in\mathbb{R}^n}{\text{minimize}} \ \|\boldsymbol{u}\|_{\ell^1} \ \text{subject to} \ \Phi\boldsymbol{u} = \zeta, \ \|\boldsymbol{u}\|_{\ell^\infty} \leq 1.$$

- Let's first consider the two constraints.
- Define

$$C_1 \triangleq \{\boldsymbol{u} \in \mathbb{R}^n : \|\boldsymbol{u}\|_{\ell^\infty} \leq 1\}$$
$$C_2 \triangleq \{\zeta\}$$

- Indicator functions:

$$I_{C_1}(\boldsymbol{u}) \triangleq \begin{cases} 0, & \text{if } \|\boldsymbol{u}\|_{\ell^\infty} \leq 1, \\ \infty, & \text{if } \|\boldsymbol{u}\|_{\ell^\infty} > 1, \end{cases}$$

$$I_{C_2}(x) \triangleq \begin{cases} 0, & \text{if } x = \zeta, \\ \infty, & \text{if } x \neq \zeta. \end{cases}$$

ADMM formulation

ℓ^1 optimization problem

$$\underset{\boldsymbol{u}\in\mathbb{R}^n}{\text{minimize}} \ \|\boldsymbol{u}\|_{\ell^1} \ \text{subject to} \ \Phi\boldsymbol{u} = \zeta, \ \|\boldsymbol{u}\|_{\ell^\infty} \leq 1.$$

- Let's first consider the two constraints.
- Define

$$C_1 \triangleq \{\boldsymbol{u} \in \mathbb{R}^n : \|\boldsymbol{u}\|_{\ell^\infty} \leq 1\}$$
$$C_2 \triangleq \{\zeta\}$$

- Indicator functions:

$$I_{C_1}(\boldsymbol{u}) \triangleq \begin{cases} 0, & \text{if } \|\boldsymbol{u}\|_{\ell^\infty} \leq 1, \\ \infty, & \text{if } \|\boldsymbol{u}\|_{\ell^\infty} > 1, \end{cases}$$

$$I_{C_2}(\boldsymbol{x}) \triangleq \begin{cases} 0, & \text{if } \boldsymbol{x} = \zeta, \\ \infty, & \text{if } \boldsymbol{x} \neq \zeta. \end{cases}$$

ADMM formulation

ℓ^1 optimization problem

$$\underset{u \in \mathbb{R}^n}{\text{minimize}} \ \|u\|_{\ell^1} \ \text{subject to} \ \Phi u = \zeta, \ \|u\|_{\ell^\infty} \leq 1.$$

- Then the optimization problem becomes

$$\underset{u \in \mathbb{R}^n}{\text{minimize}} \ \left\{ \|u\|_{\ell^1} + I_{C_1}(u) + I_{C_2}(\Phi u) \right\}.$$

- Define new variables $z_0, z_1 \in \mathbb{R}^n$, $z_2 \in \mathbb{R}^d$ by $z_0 = z_1 = u$, $z_2 = \Phi u$.
- Then we obtain

$$\underset{u \in \mathbb{R}^n, z \in \mathbb{R}^\nu}{\text{minimize}} \ \left\{ \|z_0\|_{\ell^1} + I_{C_1}(z_1) + I_{C_2}(z_2) \right\} \ \text{subject to} \ z = \Psi u,$$

where $\nu \triangleq 2n + d$, and

$$z \triangleq \begin{bmatrix} z_0 \\ z_1 \\ z_2 \end{bmatrix} \in \mathbb{R}^\nu, \quad \Psi \triangleq \begin{bmatrix} I \\ I \\ \Phi \end{bmatrix} \in \mathbb{R}^{\nu \times n}.$$

ADMM formulation

ℓ^1 optimization problem

$$\underset{\boldsymbol{u}\in\mathbb{R}^n}{\text{minimize}}\ \|\boldsymbol{u}\|_{\ell^1}\ \text{subject to}\ \Phi\boldsymbol{u}=\zeta,\ \|\boldsymbol{u}\|_{\ell^\infty}\leq 1.$$

- Then the optimization problem becomes

$$\underset{\boldsymbol{u}\in\mathbb{R}^n}{\text{minimize}}\ \left\{\|\boldsymbol{u}\|_{\ell^1}+I_{C_1}(\boldsymbol{u})+I_{C_2}(\Phi\boldsymbol{u})\right\}.$$

- Define new variables $z_0, z_1 \in \mathbb{R}^n$, $z_2 \in \mathbb{R}^d$ by $z_0 = z_1 = \boldsymbol{u}$, $z_2 = \Phi\boldsymbol{u}$.
- Then we obtain

$$\underset{\boldsymbol{u}\in\mathbb{R}^n, z\in\mathbb{R}^\nu}{\text{minimize}}\ \left\{\|z_0\|_{\ell^1}+I_{C_1}(z_1)+I_{C_2}(z_2)\right\}\ \text{subject to}\ z=\Psi\boldsymbol{u},$$

where $\nu \triangleq 2n+d$, and

$$z \triangleq \begin{bmatrix} z_0 \\ z_1 \\ z_2 \end{bmatrix} \in \mathbb{R}^\nu, \quad \Psi \triangleq \begin{bmatrix} I \\ I \\ \Phi \end{bmatrix} \in \mathbb{R}^{\nu\times n}.$$

ADMM formulation

ℓ^1 optimization problem

$$\underset{\boldsymbol{u}\in\mathbb{R}^n}{\text{minimize}} \;\; \|\boldsymbol{u}\|_{\ell^1} \;\; \text{subject to} \;\; \Phi\boldsymbol{u} = \zeta, \; \|\boldsymbol{u}\|_{\ell^\infty} \leq 1.$$

- Then the optimization problem becomes

$$\underset{\boldsymbol{u}\in\mathbb{R}^n}{\text{minimize}} \;\; \left\{\|\boldsymbol{u}\|_{\ell^1} + I_{C_1}(\boldsymbol{u}) + I_{C_2}(\Phi\boldsymbol{u})\right\}.$$

- Define new variables $\boldsymbol{z}_0, \boldsymbol{z}_1 \in \mathbb{R}^n$, $\boldsymbol{z}_2 \in \mathbb{R}^d$ by $\boldsymbol{z}_0 = \boldsymbol{z}_1 = \boldsymbol{u}$, $\boldsymbol{z}_2 = \Phi\boldsymbol{u}$.
- Then we obtain

$$\underset{\boldsymbol{u}\in\mathbb{R}^n, \boldsymbol{z}\in\mathbb{R}^\nu}{\text{minimize}} \;\; \left\{\|\boldsymbol{z}_0\|_{\ell^1} + I_{C_1}(\boldsymbol{z}_1) + I_{C_2}(\boldsymbol{z}_2)\right\} \;\; \text{subject to} \;\; \boldsymbol{z} = \Psi\boldsymbol{u},$$

where $\nu \triangleq 2n + d$, and

$$\boldsymbol{z} \triangleq \begin{bmatrix} \boldsymbol{z}_0 \\ \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \end{bmatrix} \in \mathbb{R}^\nu, \quad \Psi \triangleq \begin{bmatrix} I \\ I \\ \Phi \end{bmatrix} \in \mathbb{R}^{\nu\times n}.$$

ADMM formulation

- Define

$$f_1(\boldsymbol{u}) \triangleq 0, \quad f_2(\boldsymbol{z}) \triangleq \|\boldsymbol{z}_0\|_{\ell^1} + I_{C_1}(\boldsymbol{z}_1) + I_{C_2}(\boldsymbol{z}_2)$$

- Then we obtain the standard optimization problem for ADMM

$$\underset{\boldsymbol{u}\in\mathbb{R}^n, \boldsymbol{z}\in\mathbb{R}^\nu}{\text{minimize}} \ f_1(\boldsymbol{u}) + f_2(\boldsymbol{z}) \ \text{subject to} \ \boldsymbol{z} = \Psi\boldsymbol{u},$$

- ADMM algorithm:

$$\begin{aligned}
\boldsymbol{u}[k+1] &:= \underset{\boldsymbol{u}\in\mathbb{R}^n}{\arg\min}\left\{f_1(\boldsymbol{u}) + \frac{1}{2\gamma}\left\|\Psi\boldsymbol{u} - \boldsymbol{z}[k] + \boldsymbol{v}[k]\right\|_{\ell^2}^2\right\},\\
\boldsymbol{z}[k+1] &:= \text{prox}_{\gamma f_2}\big(\Psi\boldsymbol{u}[k+1] + \boldsymbol{v}[k]\big),\\
\boldsymbol{v}[k+1] &:= \boldsymbol{v}[k] + \Psi\boldsymbol{u}[k+1] - \boldsymbol{z}[k+1].
\end{aligned}$$

ADMM formulation

- Define

$$f_1(\boldsymbol{u}) \triangleq 0, \quad f_2(\boldsymbol{z}) \triangleq \|\boldsymbol{z}_0\|_{\ell^1} + I_{C_1}(\boldsymbol{z}_1) + I_{C_2}(\boldsymbol{z}_2)$$

- Then we obtain the standard optimization problem for ADMM

$$\underset{\boldsymbol{u}\in\mathbb{R}^n, \boldsymbol{z}\in\mathbb{R}^\nu}{\text{minimize}} \ f_1(\boldsymbol{u}) + f_2(\boldsymbol{z}) \ \text{subject to} \ \boldsymbol{z} = \Psi\boldsymbol{u},$$

- ADMM algorithm:

$$\begin{aligned}
\boldsymbol{u}[k+1] &:= \underset{\boldsymbol{u}\in\mathbb{R}^n}{\arg\min} \left\{ f_1(\boldsymbol{u}) + \frac{1}{2\gamma}\|\Psi\boldsymbol{u} - \boldsymbol{z}[k] + \boldsymbol{v}[k]\|_{\ell^2}^2 \right\}, \\
\boldsymbol{z}[k+1] &:= \operatorname{prox}_{\gamma f_2}(\Psi\boldsymbol{u}[k+1] + \boldsymbol{v}[k]), \\
\boldsymbol{v}[k+1] &:= \boldsymbol{v}[k] + \Psi\boldsymbol{u}[k+1] - \boldsymbol{z}[k+1].
\end{aligned}$$

ADMM formulation

- Define

$$f_1(\boldsymbol{u}) \triangleq 0, \quad f_2(\boldsymbol{z}) \triangleq \|\boldsymbol{z}_0\|_{\ell^1} + I_{C_1}(\boldsymbol{z}_1) + I_{C_2}(\boldsymbol{z}_2)$$

- Then we obtain the standard optimization problem for ADMM

$$\underset{\boldsymbol{u}\in\mathbb{R}^n, \boldsymbol{z}\in\mathbb{R}^\nu}{\text{minimize}} \; f_1(\boldsymbol{u}) + f_2(\boldsymbol{z}) \; \text{subject to} \; \boldsymbol{z} = \Psi\boldsymbol{u},$$

- ADMM algorithm:

$$\begin{aligned}
\boldsymbol{u}[k+1] &:= \underset{\boldsymbol{u}\in\mathbb{R}^n}{\arg\min} \left\{ f_1(\boldsymbol{u}) + \frac{1}{2\gamma}\left\|\Psi\boldsymbol{u} - \boldsymbol{z}[k] + \boldsymbol{v}[k]\right\|_{\ell^2}^2 \right\}, \\
\boldsymbol{z}[k+1] &:= \mathrm{prox}_{\gamma f_2}\big(\Psi\boldsymbol{u}[k+1] + \boldsymbol{v}[k]\big), \\
\boldsymbol{v}[k+1] &:= \boldsymbol{v}[k] + \Psi\boldsymbol{u}[k+1] - \boldsymbol{z}[k+1].
\end{aligned}$$

ADMM formulation

- Since $f_1 = 0$, the first term becomes

$$\begin{aligned} \boldsymbol{u}[k+1] &= \arg\min_{\boldsymbol{u}\in\mathbb{R}^n} \left\{ \frac{1}{2\gamma} \|\Psi \boldsymbol{u} - \boldsymbol{z}[k] + \boldsymbol{v}[k]\|_{\ell^2}^2 \right\} \\ &= (\Psi^\top \Psi)^{-1} \Psi^\top (\boldsymbol{z}[k] - \boldsymbol{v}[k]). \end{aligned}$$

- For the proximal operator $\mathrm{prox}_{\gamma f_2}$ with

$$f_2(z) \triangleq \|z_0\|_{\ell^1} + I_{C_1}(z_1) + I_{C_2}(z_2)$$

we can safely split it into prox's of $\|z_0\|_{\ell^1}$, $I_{C_1}(z_1)$, and $I_{C_2}(z_2)$, since z_1, z_2, and z_3 are independent.

ADMM formulation

- Since $f_1 = 0$, the first term becomes

$$\begin{aligned} \boldsymbol{u}[k+1] &= \operatorname*{arg\,min}_{\boldsymbol{u}\in\mathbb{R}^n} \left\{ \frac{1}{2\gamma} \|\Psi\boldsymbol{u} - \boldsymbol{z}[k] + \boldsymbol{v}[k]\|_{\ell^2}^2 \right\} \\ &= (\Psi^\top\Psi)^{-1}\Psi^\top(\boldsymbol{z}[k] - \boldsymbol{v}[k]). \end{aligned}$$

- For the proximal operator $\mathrm{prox}_{\gamma f_2}$ with

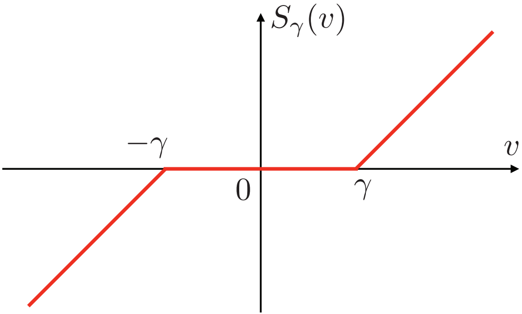
$$f_2(\boldsymbol{z}) \triangleq \|\boldsymbol{z}_0\|_{\ell^1} + I_{C_1}(\boldsymbol{z}_1) + I_{C_2}(\boldsymbol{z}_2)$$

we can safely split it into prox's of $\|\boldsymbol{z}_0\|_{\ell^1}$, $I_{C_1}(\boldsymbol{z}_1)$, and $I_{C_2}(\boldsymbol{z}_2)$, since $\boldsymbol{z}_1$, $\boldsymbol{z}_2$, and $\boldsymbol{z}_3$ are independent.

ADMM formulation

- The proximal operator of $\|\boldsymbol{z}_0\|_{\ell^1}$ is the soft-thresholding operator, that is,

$$\left[\text{prox}_{\gamma\|\cdot\|_{\ell^1}}(\boldsymbol{u})\right]_i = [S_\gamma(\boldsymbol{u})]_i \triangleq \begin{cases} u_i - \gamma, & u_i \geq \gamma, \\ 0, & |u_i| < \gamma, \\ u_i + \gamma, & u_i \leq -\gamma, \end{cases}$$



ADMM formulation

- The proximal operator of the indicator function $I_{C_1}(\boldsymbol{z}_1)$ is the projection Π_{C_1} onto $C_1 = \{\boldsymbol{u} \in \mathbb{R}^n : \|\boldsymbol{u}\|_{\ell^\infty} \leq 1\}$
- The projection onto the ℓ^∞ ball in $\mathbb{R}^n$ is given by

$$\Pi_{C_1}(\boldsymbol{u}) = \begin{bmatrix} \mathrm{sat}(u_1) \\ \vdots \\ \mathrm{sat}(u_n) \end{bmatrix}, \quad \mathrm{sat}(u) \triangleq \mathrm{sgn}(u) \min\{|u|, 1\},$$

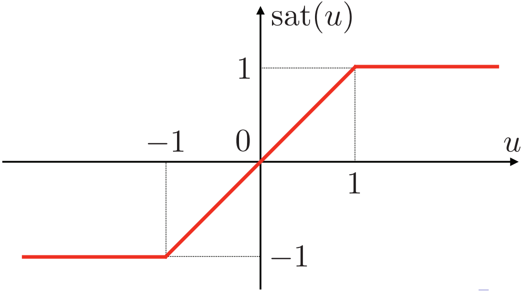
where the function $\mathrm{sat}(\cdot)$ is called the saturation function.

ADMM formulation

- The proximal operator of the indicator function $I_{C_1}(\boldsymbol{z}_1)$ is the projection Π_{C_1} onto $C_1 = \{\boldsymbol{u} \in \mathbb{R}^n : \|\boldsymbol{u}\|_{\ell^\infty} \leq 1\}$
- The projection onto the ℓ^∞ ball in $\mathbb{R}^n$ is given by

$$\Pi_{C_1}(\boldsymbol{u}) = \begin{bmatrix} \mathrm{sat}(u_1) \\ \vdots \\ \mathrm{sat}(u_n) \end{bmatrix}, \quad \mathrm{sat}(u) \triangleq \mathrm{sgn}(u) \min\{|u|, 1\},$$

where the function $\mathrm{sat}(\cdot)$ is called the saturation function.



ADMM formulation

- The proximal operator of the indicator function $I_{C_2}(z_2)$ is the projection Π_{C_2} onto $\{\zeta\}$, which is simply given by

$$\Pi_{C_2}(z_2) \triangleq \zeta.$$

- Finally, we have the second step of ADMM

$$z[k+1] = \begin{bmatrix} S_\gamma(u[k+1] + v_0[k]) \\ \Pi_{C_1}(u[k+1] + v_1[k]) \\ \zeta \end{bmatrix}$$

- The third step of ADMM is just a linear transformation

$$v[k+1] = v[k] + \Psi u[k+1] - z[k+1]$$

ADMM formulation

- The proximal operator of the indicator function $I_{C_2}(\boldsymbol{z}_2)$ is the projection Π_{C_2} onto $\{\zeta\}$, which is simply given by

$$\Pi_{C_2}(\boldsymbol{z}_2) \triangleq \zeta.$$

- Finally, we have the second step of ADMM

$$\boldsymbol{z}[k+1] = \begin{bmatrix} S_\gamma(\boldsymbol{u}[k+1] + \boldsymbol{v}_0[k]) \\ \Pi_{C_1}(\boldsymbol{u}[k+1] + \boldsymbol{v}_1[k]) \\ \zeta \end{bmatrix}$$

- The third step of ADMM is just a linear transformation

$$\boldsymbol{v}[k+1] = \boldsymbol{v}[k] + \Psi\boldsymbol{u}[k+1] - \boldsymbol{z}[k+1]$$

ADMM formulation

- The proximal operator of the indicator function $I_{C_2}(z_2)$ is the projection Π_{C_2} onto $\{\zeta\}$, which is simply given by

$$\Pi_{C_2}(z_2) \triangleq \zeta.$$

- Finally, we have the second step of ADMM

$$z[k+1] = \begin{bmatrix} S_\gamma(u[k+1] + v_0[k]) \\ \Pi_{C_1}(u[k+1] + v_1[k]) \\ \zeta \end{bmatrix}$$

- The third step of ADMM is just a linear transformation

$$v[k+1] = v[k] + \Psi u[k+1] - z[k+1]$$

ADMM formulation

ADMM algorithm to solve the ℓ^1 optimization

Initialization: give initial vectors $\boldsymbol{z}[0], \boldsymbol{v}[0] \in \mathbb{R}^\nu$, and $\gamma > 0$
Iteration: for $k = 0, 1, 2, \dots$ do

$$\boldsymbol{u}[k+1] = M(\boldsymbol{z}[k] - \boldsymbol{v}[k])$$

$$\boldsymbol{z}[k+1] = \begin{bmatrix} S_\gamma(\boldsymbol{u}[k+1] + \boldsymbol{v}_0[k]) \\ \Pi_{C_1}(\boldsymbol{u}[k+1] + \boldsymbol{v}_1[k]) \\ \zeta \end{bmatrix}$$

$$\boldsymbol{v}[k+1] = \boldsymbol{v}[k] + \Psi\boldsymbol{u}[k+1] - \boldsymbol{z}[k+1], \quad k = 0, 1, 2, \dots$$

- $M \triangleq (\Psi^\top\Psi)^{-1}\Psi^\top$ can be computed offline.
- S_γ is the soft-thresholding function.
- Π_{C_1} is the saturation function.
- The algorithm can be implemented easily and fast!

ADMM formulation

ADMM algorithm to solve the ℓ^1 optimization

Initialization: give initial vectors $\boldsymbol{z}[0], \boldsymbol{v}[0] \in \mathbb{R}^\nu$, and $\gamma > 0$

Iteration: for $k = 0, 1, 2, \ldots$ do

$$\boldsymbol{u}[k+1] = M(\boldsymbol{z}[k] - \boldsymbol{v}[k])$$

$$\boldsymbol{z}[k+1] = \begin{bmatrix} S_\gamma(\boldsymbol{u}[k+1] + \boldsymbol{v}_0[k]) \\ \Pi_{C_1}(\boldsymbol{u}[k+1] + \boldsymbol{v}_1[k]) \\ \zeta \end{bmatrix}$$

$$\boldsymbol{v}[k+1] = \boldsymbol{v}[k] + \Psi \boldsymbol{u}[k+1] - \boldsymbol{z}[k+1], \quad k = 0, 1, 2, \ldots$$

- $M \triangleq (\Psi^\top \Psi)^{-1} \Psi^\top$ can be computed offline.
- S_γ is the soft-thresholding function.
- Π_{C_1} is the saturation function.
- The algorithm can be implemented easily and fast!

ADMM formulation

ADMM algorithm to solve the ℓ^1 optimization

Initialization: give initial vectors $\boldsymbol{z}[0], \boldsymbol{v}[0] \in \mathbb{R}^\nu$, and $\gamma > 0$
Iteration: for $k = 0, 1, 2, \ldots$ do

$$\boldsymbol{u}[k+1] = M(\boldsymbol{z}[k] - \boldsymbol{v}[k])$$

$$\boldsymbol{z}[k+1] = \begin{bmatrix} S_\gamma(\boldsymbol{u}[k+1] + \boldsymbol{v}_0[k]) \\ \Pi_{C_1}(\boldsymbol{u}[k+1] + \boldsymbol{v}_1[k]) \\ \boldsymbol{\zeta} \end{bmatrix}$$

$$\boldsymbol{v}[k+1] = \boldsymbol{v}[k] + \Psi \boldsymbol{u}[k+1] - \boldsymbol{z}[k+1], \quad k = 0, 1, 2, \ldots$$

- $M \triangleq (\Psi^\top \Psi)^{-1} \Psi^\top$ can be computed offline.
- S_γ is the soft-thresholding function.
- Π_{C_1} is the saturation function.
- The algorithm can be implemented easily and fast!

ADMM formulation

ADMM algorithm to solve the ℓ^1 optimization

Initialization: give initial vectors $\boldsymbol{z}[0], \boldsymbol{v}[0] \in \mathbb{R}^{\nu}$, and $\gamma > 0$

Iteration: for $k = 0, 1, 2, \dots$ do

$$\boldsymbol{u}[k+1] = M(\boldsymbol{z}[k] - \boldsymbol{v}[k])$$

$$\boldsymbol{z}[k+1] = \begin{bmatrix} S_{\gamma}(\boldsymbol{u}[k+1] + \boldsymbol{v}_0[k]) \\ \Pi_{C_1}(\boldsymbol{u}[k+1] + \boldsymbol{v}_1[k]) \\ \zeta \end{bmatrix}$$

$$\boldsymbol{v}[k+1] = \boldsymbol{v}[k] + \Psi\boldsymbol{u}[k+1] - \boldsymbol{z}[k+1], \quad k = 0, 1, 2, \dots$$

- $M \triangleq (\Psi^{\top}\Psi)^{-1}\Psi^{\top}$ can be computed offline.
- S_{γ} is the soft-thresholding function.
- Π_{C_1} is the saturation function.
- The algorithm can be implemented easily and fast!

MATLAB implementation

Please download the MATLAB program

`chap9_sparse_control_ADMM.m`

for the ADMM computation from

`https://nagahara-masaaki.github.io/spm_en.html`

Table of Contents

Conclusion

- By time discretization, the L^1-optimal control problem (L^1-OPT) is reduced to a finite-dimensional ℓ^1 optimization problem.
- In time discretization, the control is assumed to be piecewise constant by a zero-order hold.
- The reduced ℓ^1 optimization can be efficiently solved by ADMM.

Conclusion

- By time discretization, the L^1-optimal control problem (L^1-OPT) is reduced to a finite-dimensional ℓ^1 optimization problem.
- In time discretization, the control is assumed to be piecewise constant by a zero-order hold.
- The reduced ℓ^1 optimization can be efficiently solved by ADMM.

Conclusion

- By time discretization, the L^1-optimal control problem (L^1-OPT) is reduced to a finite-dimensional ℓ^1 optimization problem.
- In time discretization, the control is assumed to be piecewise constant by a zero-order hold.
- The reduced ℓ^1 optimization can be efficiently solved by ADMM.