第3回MPC勉強会

鶴原康太

November 6, 2023

今回の目標

$$-\frac{\partial V}{\partial t}(x,t) = \min_u H\left(x, u, \left(\frac{\partial V}{\partial x}\right)^T(x,t), t\right)$$

- **前回のおさらい**
- **動的計画法**
- HJB **方程式**
- **最小原理**
- MPC **導入**

前回のおさらい

微分法

関数の勾配を考える
停留するとある関数の最大 or 最小
点の変動を考える

→

変分法

汎関数 (関数の関数) の勾配を考える
停留するとある関数の全体を最大 or 最小
関数の変動を考える

- 微分法ではある曲線を最小化 (最大化) することを考えていたけど、変分法ではある評価関数に基づいて関数全体を最小化 (最大化) する関数を求める
- 偏微分と似た考え方をする
- 二点境界値問題の解
- Euler-Lagrange 方程式を満たす
- 制約を含んだ場合

最適性条件まとめ

時間を考慮しない

離散時間 MPC **では予め時間を考慮して出した式から出た最適化問題を解いたから最適化問題自体は時間を考慮しなくていい**
Lagrange **の未定乗数法**

$$L(x, \lambda) = f(x) + \lambda^T g(x)$$

KKT **条件**

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) + \sum_{j=1}^{n} \mu_j^* \nabla h_j(x^*) = 0$$

$$\begin{aligned} g_i(x^*) &\leq 0 \quad \text{for all } i \\ \lambda_i^* &\geq 0 \quad \text{for all } i \\ \lambda_i^* g_i(x^*) &= 0 \quad \text{for all } i \\ h_j(x^*) &= 0 \quad \text{for all } j \end{aligned}$$

時間を考慮

Euler-Lagrange **方程式**

$$\frac{\partial L}{\partial x} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) = 0$$

最適化の多様な定式

Q. **なぜ似たようなものがいくつもあるのか ??**
A. **同時期 (冷戦期) に 2 人の天才によって最適制御が定式化されたから**

Bellman

Pontryagin

Bellman **によって動的計画法** (Bellman **方程式**)
Pontryagin **によって最小原理** (**最大原理**)

動的計画法 1

最適制御

$$\dot{x}(t) = f(x(t), u(t), t)$$
$$J = \psi(x(t_f)) + \int_{t_0}^{t_f} L(x(t), u(t), t)\, dt$$

別の表現を見てみる

価値関数 $V(x, t)$: **評価関数** J **を最小にする値**

$$V(x, t) = \min_{u[t, t_f]} \left(\psi(x(t_f)) + \int_{t}^{t_f} L(x(\tau), u(\tau), \tau) d\tau \right)$$

最適化を価値関数を探すことだと考える

動的計画法 2

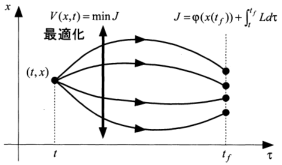
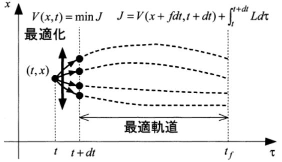


図 2　評価区間 $[t, t_f]$ の最適制御問題



大塚: 非線形最適制御入門

動的計画法は大きな問題を小さな部分問題に分解し、それぞれ解くことで全体の解を得る手法

- 最適的な性質を持つ
- 計算結果を保存して再利用 (メモ化)

動的計画法 3

$$
\begin{aligned}
V(x,t) &= \min_{u[t,t_f]} \left(\psi(x(t_f)) + \int_t^{t_f} L(x(\tau),u(\tau),\tau)d\tau \right) \\
&= \min_{u[t,t_f]} \left(\int_t^{t+dt} L(x(\tau),u(\tau),\tau)d\tau + \psi(x(t_f)) + \int_{t+dt}^{t_f} L(x(\tau),u(\tau),\tau)d\tau \right) \\
&= \min_{u[t,t+dt]} \left(\int_t^{t+dt} L(x(\tau),u(\tau),\tau)d\tau + \min_{u[t+dt,t_f]} \left(\psi(x(t_f)) + \int_{t+dt}^{t_f} L(x(\tau),u(\tau),\tau)d\tau \right) \right) \\
&= \min_{u[t,t+dt]} \left(\int_t^{t+dt} L(x(\tau),u(\tau),\tau)d\tau + V\left(x + \int_t^{t+dt} f(x,u,\tau)d\tau,\, t+dt \right) \right)
\end{aligned}
$$

Bellman 方程式

$$
\begin{aligned}
V(x,t) &= \min_{u[t,t+dt]} \left(L(x,u,t)dt + V\left(x + f(x,u,t)dt,\, t+dt\right) \right) \\
V(x,t_f) &= \psi(x(t_f))
\end{aligned}
$$

Bellman **方程式が解ければ最適制御ができる**!!!(**もともとの評価関数を満たす解を求めるのと一緒**)
目標の状態 $x(t_f)$ **が分かっていれば解けそう**??

HJB 方程式

Bellman 方程式

$$V(x,t) = \min_{u[t,t+dt]} \left(L(x,u,t)dt + V\left(x + f(x,u,t)dt, t+dt\right)\right)$$

次元の呪い

次元の呪いは、状態空間や行動空間の次元数が増加するにつれて、必要な計算量やメモリが指数的に増加する現象

2 次元の格子が 10 の場合、合計 100 のセルが存在します。しかし、10 次元の格子が各次元に 10 のセルを持つ場合、合計で $10^10 = 10,000,000,000$ のセルが存在します。このように、次元が増加するにつれて格子の数が指数的に増加し、それに伴い計算量も指数的に増加します。

次元の呪いのグラフを貼る

現在位置

- 前回のおさらい
- 動的計画法
- HJB 方程式
- 最小原理
- MPC 導入

Bellman 方程式から HJB 方程式への変形

$$V(x,t) = \min_u \left(L(x,u,t)dt + V\left(x + f(x,u,t)dt, t + dt\right)\right)$$

$$-\frac{\partial V}{\partial t}(x,t) = \min_u H\left(x, u, \left(\frac{\partial V}{\partial x}\right)^T (x,t), t\right)$$

HJB 方程式

$V(x,t) = \min_u \left(L(x,u,t)dt + V\left(x + f(x,u,t)dt, t+dt\right)\right)$

$V\left(x + f(x,u,t)dt, t+dt\right)$ を (x,t) 周りで Taylor 展開をする

$$V\left(x + f(x,u,t)dt, t+dt\right) \simeq V(x,t) + \frac{\partial V}{\partial x}(x,t)\, f(x,u,t)\, dt + \frac{\partial V}{\partial t}(x,t)\, dt$$

$$\begin{aligned}
V(x,t) &= \min_{u[t,t+dt]} \left(L(x,u,t)dt + V(x,t) + \frac{\partial V}{\partial x}(x,t)\, f(x,u,t)\, dt + \frac{\partial V}{\partial t}(x,t)\, dt \right) \\
0 &= \min_{u[t,t+dt]} \left(L(x,u,t)dt + \frac{\partial V}{\partial x}(x,t)\, f(x,u,t)\, dt + \frac{\partial V}{\partial t}(x,t)\, dt \right) \\
H(x,u,\lambda,t) &= L(x,u,t) + \lambda^T f(x,u,t) \\
H\left(x, u, \left(\frac{\partial V}{\partial x}\right)^T (x,t), t\right) &= L(x,u,t) + \frac{\partial V}{\partial x}(x,t)\, f(x,u,t)
\end{aligned}$$

$-\frac{\partial V}{\partial t}(x,t) = \min_u H\left(x, u, \left(\frac{\partial V}{\partial x}\right)^T (x,t), t\right)$

HJB 方程式の解法

直接法
間接法
シューティング法とかの図をここに貼る

現在位置

- 前回のおさらい
- 動的計画法
- HJB 方程式
- 最小原理
- MPC 導入

別の観点から最適制御を考える

Pontryagin

最小原理 1

最小原理 2

もちろん最小原理から HJB **方程式を導ける**

最適性条件まとめ

ラグランジュの未定乗数法
KKT **条件**
オイラーラグランジュ方程式
動的計画法 (new)
HJB **方程式** (new)
最小原理 (new)
動的計画法から HJB **方程式を導いたが、最小原理からも導ける**
ここに各関係の図を貼る

現在位置

- 前回のおさらい
- 動的計画法
- HJB 方程式
- 最小原理
- MPC 導入

MPC

Euler-Lagrange 方程式では入力に状態を含んでいないので、状態が変化 (外的要因によって) する現実システムだと扱いにくい

HJB 方程式は入力に状態を含んでいるが、偏微分方程式なので扱いにくい

現実問題に落とし込んだのが MPC

評価区間が無限の MPC の解は HJB 方程式と一致するはず (たぶん...)

参考資料

書籍

- 非線形最適制御入門 (名著です)
- しっかり学ぶ数理最適化 (最適化全般について)
- はじめての最適化 (変分法の説明が分かりやすいです)