PythonとCasADiで学ぶモデル予測制御

1.3以降

1.3.1 PID制御

1.3.1 PID制御

PID制御

- **フィードバック制御**
- 現在最も広く普及している制御手法
- 我々も倒立振子の時など，お世話になってきた

1.3.1 PID制御

観測と状態変数の関係

制御対象のシステムからセンサを通じて直接取得できる量：$y(t)$
→つまり，観測

観測$y(t)$と状態変数$\mathbf{x}(t)$を完全に等しいと見なせば

$$y(t) = \mathbf{x}(t)$$

また，観測ノイズ$v(t)$が加わると

$$y(t) = \mathbf{x}(t) + v(t)$$

または，

$$y(t) = g(\mathbf{x}(t))$$

1.3.1 PID制御

誤差について

観測の目標値を$y_{ref}(t)$とする.
観測とその目標値との誤差を

$$e(t) = y(t) - y_{ref}(t) \tag{1.13}$$

とする.
この誤差$e(t)$を用いて，制御入力$u(t)$を次の式のように計算する.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \dot{e}(t) \tag{1.14}$$

1.3.1 PID制御

式(1.14)の各項とゲインについて

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \dot{e}(t) \tag{1.14}$$

- 第1項
  $K_p e(t)$は誤差に比例しているため**比例項**と呼ばれる
- 第2項
  $K_i \int_0^t e(\tau) d\tau$は誤差の積分を利用していることから**積分項**と呼ばれる
- 第3項
  $K_d \dot{e}(t)$は誤差の微分を利用していることから**微分項**と呼ばれる.

1.3.1 PID制御

PID制御の特徴

- 1入力1出力システムの場合，各項の影響を把握することは比較的簡単
  （制御入力$u(t)$と観測$y(t)$がそれぞれ1次元であるシステム）
- 実際の制御対象を動作させながらPIDゲインを調整し，制御設計できる
  （学部でやった倒立振子など）
- 状態方程式モデルや伝達関数モデルを用意することなく使える
- 枯れた技術であるため，PID制御が今もなお採用され続けている

1.3.2 LQ制御

1.3.2 LQ制御

最適制御において最も基礎的な方法： **LQ(linear quadratic)制御**
離散時間で無限ホライズンの場合のLQ制御を説明する.

状態方程式モデルとして，次の線形システムを考える.

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k \tag{1.15}$$

ここで，$A \in \mathbb{R}^{n_x \times n_x}$と$B \in \mathbb{R}^{n_x \times n_u}$は定数行列である.

1.3.2 LQ制御

評価関数として，次式を考える．

$$\sum_{k=k_s}^{\infty}(\mathbf{x}_k^{\mathrm{T}}Q\mathbf{x}_k + \mathbf{u}_k^{\mathrm{T}}R\mathbf{u}_k) \tag{1.16}$$

ここで，$Q \in \mathbb{R}^{n_x \times n_x}$と$R \in \mathbb{R}^{n_x \times n_u}$は正定値行列である．
この行列は，状態変数$\mathbf{x}_k$と制御入力$\mathbf{u}_k$が0から離れていることに関するコストのバランスを調整する行列である．

→無限の期間で$\mathbf{u}_k$を可能な限り小さく，$\mathbf{x}_k$を0に収束させたいという制御の目標を定量化している

1.3.2 LQ制御

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k \tag{1.15}$$

$$\sum_{k=k_s}^{\infty} (\mathbf{x}_k^{\mathrm{T}} Q \mathbf{x}_k + \mathbf{u}_k^{\mathrm{T}} R \mathbf{u}_k) \tag{1.16}$$

式(1.15)の$\mathbf{x}_k, \mathbf{u}_k$の挙動の望ましさを，式$(1.16)$の評価関数で定義する場合，
この評価関数を最小化する制御入力は次式のような制御則として与えられる．

$$\mathbf{u}_k = -(R + B^{\mathrm{T}} P B)^{-1} B^{\mathrm{T}} P A \mathbf{x}_k \tag{1.17}$$

ここで，$P \in \mathbb{R}^{n_x \times n_x}$は次式（離散時間リッカチ方程式）を満たす行列である．

$$P = A^{\mathrm{T}} P A - A^{\mathrm{T}} P B (R + B^{\mathrm{T}} P B)^{-1} B^{\mathrm{T}} P A + Q \tag{1.18}$$

1.3.2 LQ制御： 式(1.17)の導出（$\mathbf{u}_k = -(R + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PA\mathbf{x}_k$）

1. **問題の設定**
   - 状態変数： $\mathbf{x} \in \mathbb{R}^n$
   - 制御入力： $\mathbf{u} \in \mathbb{R}^m$
   - 状態方程式

     $\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}(k)$： $\mathbf{x}_{k+1}$が現在の状態$\mathbf{x}_k$と制御入力$\mathbf{u}_k$によって決定される

2. **評価関数**

   $J = \sum_{k=k_s}^{\infty}(\mathbf{x}_k^{\mathrm{T}}Q\mathbf{x}_k + \mathbf{u}_k^{\mathrm{T}}R\mathbf{u}_k)$： ある初期状態$\mathbf{x}_k$から終端までの累積最小コスト

   なお，$\mathbf{x}_k^{\mathrm{T}}Q\mathbf{x}_k + \mathbf{u}_k^{\mathrm{T}}R\mathbf{u}_k$ が状態と制御入力に依存する(ある時刻：kの)**ステージコスト**

1.3.2 LQ制御： 式(1.17)の導出（$\mathbf{u}_k = -(R + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PA\mathbf{x}_k$）

評価関数： $J = \sum_{k=k_s}^{\infty}(\mathbf{x}_k^{\mathrm{T}}Q\mathbf{x}_k + \mathbf{u}_k^{\mathrm{T}}R\mathbf{u}_k)$

3. **価値関数の定義**
   価値関数$V(\mathbf{x}_k)$は，現在の状態$\mathbf{x}_k$から**最適な**制御入力を取った時に期待される最小コスト

$$V(\mathbf{x}_k) = \min_{\mathbf{u}_\mathrm{k}} \sum_{k=k_s}^{\infty}(\mathbf{x}_k^{\mathrm{T}}Q\mathbf{x}_k + \mathbf{u}_k^{\mathrm{T}}R\mathbf{u}_k) \tag{1-1}$$

   →無限に続く計算を有限の連立方程式に変換していく

1.3.2 LQ制御： 式(1.17)の導出（$\mathbf{u}_k = -(R + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PA\mathbf{x}_k$）

価値関数： $V(\mathbf{x}_k) = \min_{\mathbf{u}_k} \sum_{k=k_s}^{\infty} (\mathbf{x}_k^{\mathrm{T}} Q \mathbf{x}_k + \mathbf{u}_k^{\mathrm{T}} R \mathbf{u}_k)$

4. **ベルマン最適方程式の導出**

   価値関数の再帰的表現

   ここで，最適な場合のステージコスト $(\mathbf{x}_k^{\mathrm{T}} Q \mathbf{x}_k + \mathbf{u}_k^{\mathrm{T}} R \mathbf{u}_k)$を$C_k$と定義する．

   その場合，価値関数は

$$V(\mathbf{x}_k) = \min_{\mathbf{u}_k} [C_k + C_{k+1} + C_{k+2} + \cdots] \tag{1-2}$$

   また，式(1-2)のkのところに$k+1$を代入すると

$$V(\mathbf{x}_{k+1}) = \min_{\mathbf{u}_k} [C_{k+1} + C_{k+2} + C_{k+3} + \cdots] \tag{1-3}$$

1.3.2 LQ制御： 式(1.17)の導出（$\mathbf{u}_k = -(R + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PA\mathbf{x}_k$）

4. **ベルマン最適方程式の導出**

   価値関数の再帰的表現

   $V(\mathbf{x}_k) = \min_{\mathbf{u}_k}[C_k + C_{k+1} + C_{k+2} + \cdots]$ (1-2)

   $V(\mathbf{x}_{k+1}) = \min_{\mathbf{u}_k}[C_{k+1} + C_{k+2} + C_{k+3} + \cdots]$ (1-3)

   式(1-3)は，時刻$k+1$以降に得られる最小コストの和になる．
   このことを踏まえて，式(1-2)を変形すると次のようになる．

$$V(\mathbf{x}_k) = \min_{\mathbf{u}_k}[C_k + V(\mathbf{x}_{k+1})] \tag{1-4}$$

1.3.2 LQ制御： 式(1.17)の導出（$\mathbf{u}_k = -(R + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PA\mathbf{x}_k$）

4. **ベルマン最適方程式の導出**

   価値関数の再帰的表現

   $C_k = (\mathbf{x}_k^{\mathrm{T}}Q\mathbf{x}_k + \mathbf{u}_k^{\mathrm{T}}R\mathbf{u}_k)$

   $V(\mathbf{x}_k) = \min_{\mathbf{u}_k}[C_k + V(\mathbf{x}_{k+1})]$ (1-4)

   つまり，価値関数$V(\mathbf{x}_k)$は，次のように表すことができる．

$$V(\mathbf{x}_k) = \min_{\mathbf{u}_k}[(\mathbf{x}_k^{\mathrm{T}}Q\mathbf{x}_k + \mathbf{u}_k^{\mathrm{T}}R\mathbf{u}_k) + V(\mathbf{x}_{k+1})] \qquad (1\text{-}5)$$

   ここで，

   - $\mathbf{x}_k^{\mathrm{T}}Q\mathbf{x}_k + \mathbf{u}_k^{\mathrm{T}}R\mathbf{u}_k$ は現時刻のステージコスト
   - $V(\mathbf{x}_{k+1})$ は次の状態での将来の累積最小コスト

1.3.2 LQ制御： 式(1.17)の導出（$\mathbf{u}_k = -(R + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PA\mathbf{x}_k$）

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k \tag{1.15}$$

4. **ベルマン最適方程式の導出**
   価値関数の二次形式仮定
   $V(\mathbf{x}_{k+1})$を簡単にするため，価値関数が二次形式であると仮定し，
   次のように表す．

$$V(\mathbf{x}_{k+1}) = \mathbf{x}_{k+1}^{\mathrm{T}} P \mathbf{x}_{k+1} \tag{1-6}$$

   なお，$P \in \mathbb{R}^{n_x \times n_x}$は正定値対称行列である．
   これを用いると，ベルマン方程式は以下のようになる．

$$V(\mathbf{x}_k) = \min_{\mathbf{u}_\mathrm{k}} [\mathbf{x}_k^{\mathrm{T}} Q \mathbf{x}_k + \mathbf{u}_k^{\mathrm{T}} R \mathbf{u}_k + (A\mathbf{x}_k + B\mathbf{u}_k)^{\mathrm{T}} P (A\mathbf{x}_k + B\mathbf{u}_k)] \tag{1-7}$$

1.3.2 LQ制御： 式(1.17)の導出（$\mathbf{u}_k = -(R + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PA\mathbf{x}_k$）

5. **最適制御則の導出**
   ベルマン方程式を$\mathbf{u}_k$について微分し，最小化条件を求める．

   微分の準備

$$\frac{\partial}{\partial \mathbf{u}_k}(\mathbf{x}_k^{\mathrm{T}}Q\mathbf{x}_k + \mathbf{u}_k^{\mathrm{T}}R\mathbf{u}_k + (A\mathbf{x}_k + B\mathbf{u}_k)^{\mathrm{T}}P(A\mathbf{x}_k + B\mathbf{u}_k)) = 0$$

   各項の微分

   - $\mathbf{x}_k^{\mathrm{T}}Q\mathbf{x}_k$は$\mathbf{u}_k$に依存しないため，微分すると0
   - 二次形式の微分公式より，$\mathbf{u}_k^{\mathrm{T}}R\mathbf{u}_k$の微分は $2R\mathbf{u}_k$
   - ベクトル微分の公式より，$(A\mathbf{x}_k + B\mathbf{u}_k)^{\mathrm{T}}P(A\mathbf{x}_k + B\mathbf{u}_k)$ の微分は $2B^{\mathrm{T}}P(A\mathbf{x}_k + B\mathbf{u}_k)$

1.3.2 LQ制御： 式(1.17)の導出（$\mathbf{u}_k = -(R + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PA\mathbf{x}_k$）

5. **最適制御則の導出**

最適正条件

全て合わせると以下のようになる

$$2R\mathbf{u}_k + 2B^{\mathrm{T}}P(A\mathbf{x}_k + B\mathbf{u}_k) = 0 \tag{1-8}$$

これを整理すると，

$$(R + B^{\mathrm{T}}PB)\mathbf{u}_k = -B^{\mathrm{T}}PA\mathbf{x}_k$$

最適制御則

$\mathbf{u}_k$ について解くと，最適制御則が得られる．

$$\mathbf{u}_k = -(R + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PA\mathbf{x}_k \tag{1.17}$$

1.3.2 LQ制御： 式(1.18)の導出（$P = A^{\mathrm{T}}PA - A^{\mathrm{T}}PB(R + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PA + Q$）

価値関数の二次形式仮定： $V(\mathbf{x}_{k+1}) = \mathbf{x}_{k+1}^{\mathrm{T}} P \mathbf{x}_{k+1}$

ベルマン方程式： $V(\mathbf{x}_k) = \min_{\mathbf{u}_k} [\mathbf{x}_k^{\mathrm{T}} Q \mathbf{x}_k + \mathbf{u}_k^{\mathrm{T}} R \mathbf{u}_k + (A\mathbf{x}_k + B\mathbf{u}_k)^{\mathrm{T}} P (A\mathbf{x}_k + B\mathbf{u}_k)]$

最適制御則： $\mathbf{u}_k = -(R + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PA\mathbf{x}_k$

リカッチ方程式の導出

最適制御則をベルマン方程式に代入し，整理する．

$$\mathbf{x}_k^{\mathrm{T}} P \mathbf{x}_k = \mathbf{x}_k^{\mathrm{T}} Q \mathbf{x}_k + \mathbf{u}_k^{\mathrm{T}} R \mathbf{u}_k + (A\mathbf{x}_k + B\mathbf{u}_k)^{\mathrm{T}} P (A\mathbf{x}_k + B\mathbf{u}_k)$$

$$\mathbf{x}_k^{\mathrm{T}} P \mathbf{x}_k = \mathbf{x}_k^{\mathrm{T}} Q \mathbf{x}_k + (-(R + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PA\mathbf{x}_k)^{\mathrm{T}} R (-(R + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PA\mathbf{x}_k) + (A\mathbf{x}_k + B(-(R + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PA\mathbf{x}_k))^{\mathrm{T}} P (A\mathbf{x}_k + B(-(R + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PA\mathbf{x}_k))$$

なんやかんやすると...（複雑すぎて諦めた）

$$P = A^{\mathrm{T}}PA - A^{\mathrm{T}}PB(R + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PA + Q \qquad (1.18)$$

1.3.2 LQ制御 （再掲）

$$\sum_{k=k_s}^{\infty}(\mathbf{x}_k^{\mathrm{T}}Q\mathbf{x}_k + \mathbf{u}_k^{\mathrm{T}}R\mathbf{u}_k) \tag{1.16}$$

式(1.16)の評価関数で定義する場合，
この評価関数を最小化する制御入力は次式のような制御則として与えられる．

$$\mathbf{u}_k = -(R + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PA\mathbf{x}_k \tag{1.17}$$

ここで，$P \in \mathbb{R}^{n_x \times n_x}$は次式（離散時間リッカチ方程式）を満たす行列である．

$$P = A^{\mathrm{T}}PA - A^{\mathrm{T}}PB(R + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PA + Q \tag{1.18}$$

式(1.17)の最適制御入力は，初期時刻における初期状態$\bar{\mathbf{x}}_{\mathrm{init}}$に依存しない．
→どんな初期状態だったとしても使える

1.3.2 LQ制御

$$\mathbf{u}_k = -(R + B^{\mathrm{T}} P B)^{-1} B^{\mathrm{T}} P A \mathbf{x}_k \tag{1.17}$$

$$P = A^{\mathrm{T}} P A - A^{\mathrm{T}} P B (R + B^{\mathrm{T}} P B)^{-1} B^{\mathrm{T}} P A + Q \tag{1.18}$$

LQ制御を適用する流れ

1. 制御適用開始前
    i. 式(1.18)の離散時間リカッチ方程式を解いてPを計算する
    ii. 式(1.17)に基づいて制御則のゲイン $\sum = (R + B^{\mathrm{T}} P B)^{-1} B^{\mathrm{T}} P A$ を求める
2. 制御適用中

    制御則$\mathbf{u}_k = -\sum \mathbf{x}_k$ に従って制御入力を計算し，制御対象へ適用する.

1.3.3 動的計画法と強化学習

1.3.3 動的計画法と強化学習

この項について

1.3.2では，
「LQ制御は初期状態が任意である問題に対して，**制御則**の形式で最適制御を求める」と紹介

1.3.3では，

- LQ制御の適用範囲を広げる
- 非線形の評価関数と状態方程式モデルの問題に対して制御則の形式で最適制御を求める

1.3.3 動的計画法と強化学習

動的計画法

最適制御則の評価関数が満たすべき**式**（離散時間問題の場合はベルマン方程式，
連続時間問題の場合はハミルトン・ヤコビ・ベルマン方程式）を解くことを考える．

> なお，動的計画法の文脈では，制御則と評価関数は，それぞれ方策と価値関数

LQ制御などに帰着できる特別な場合を除き，この問題は解析的に解くことが
できない．
そのため，実際には方策や価値関数をパラメトライズして最適化する．

1.3.3 動的計画法と強化学習

動的計画法

アプローチ

- 最適制御則の評価関数を状態空間方向に離散化して，
  （ハミルトン・ヤコビ・）ベルマン方程式を数値的に解くことが挙げられる．
- データを用いてベルマン方程式をサンプル近似してパラメータを最適化
  →**強化学習**もサンプル近似の一つ

1.3.3 動的計画法と強化学習

強化学習

アプローチ

- **モデルフリー強化学習**
  状態方程式モデルを陽に利用せず，制御対象を動作させて得た現実のデータで
  パラメータを最適化する
- **モデルベース強化学習**
  状態方程式モデルを使用し，事前に与えられていない場合には現実のデータから
  同定しながらパラメータを最適化する

2章でこれらとモデル予測制御との関係を説明