

# システム数理特別講義II

August 6, 2023

29C23002

石川健太郎

(1)

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ x(0) = x_0, \\ y(t) = Cx(t). \end{cases} \quad (1)$$

式 (1) から, 基準の時間を  $t_0$  として次が得られる.

$$x(t) = e^{A(t-t_0)}x(t_0) + \int_{t_0}^t e^{A(t-\tau)}Bu(\tau)d\tau \quad (2)$$

$T$  をサンプル時間,  $k$  を非負の整数とする.

$kT \leq t \leq (k+1)T$  で  $u(t) = u_k$  という定数の入力 (0 次ホールド) が加えられる.

$t_0 = kT$ ,  $t = (k+1)T$  として,  $x((k+1)T)$  を考える.

$$\begin{aligned} x((k+1)T) &= \exp(A((k+1)T - kT))x(kT) + u_k \int_{kT}^{(k+1)T} \exp(A((k+1)T - \tau))Bd\tau \\ &= e^{AT}x(kT) + u_k \int_0^T e^{A\tau}Bd\tau \quad (\text{変数変換 } \tau = (k+1)T - \tau) \end{aligned} \quad (3)$$

式 (1), (3) から, 次の式が得られる.

$$\begin{cases} x((k+1)T) = e^{AT}x(kT) + u_k \int_0^T e^{A\tau}Bd\tau, \\ y(kT) = Cx(kT). \end{cases} \quad (4)$$

式 (4) の係数を  $A_d := e^{AT}$ ,  $B_d := \int_0^T e^{A\tau}Bd\tau$ ,  $C_d := C$  と置き直し,  $x_k := x(kT)$ ,  $y_k := y(kT)$  とすると, 次の差分方程式が得られる.

$$\begin{aligned} x_{k+1} &= A_dx_k + B_d u_k, \\ y_k &= C_d x_k. \end{aligned} \quad (5)$$

(2)

行列  $D$  が正定であることを  $D \succ^p 0$ , 半正定であることを  $D \succeq^p 0$  と表記する.

$$x_{k+1} = Ax_k + Bu_k. \quad (6)$$

$$J = x_N^T P x_n + \sum_{k=0}^{N-1} (x_k^T Q x_k + u_k^T R u_k). \quad (7)$$

式 (6) のシステムのもとで、式 (7) の  $J$  を最小化するような制御則を求める。  
動的計画法に基づいて考えるので、最適性の必要条件である Bellman 方程式を用いる。  
ここでは  $k$  番目 (時系列では  $N - k$  番目) の価値関数を  $V_{N-k}(x_k)$  とする。

$$V_{N-k}(x_k) := \min_{u_k} \left\{ \underbrace{V_{N-(k+1)}(x_{k+1}) + x_k^T Q x_k + u_k^T R u_k}_{:= W_{N-k}(x_k)} \right\}, \quad k = 0, 1, \dots, N-1, \quad (8)$$

$$V_0(x_N) := x_N^T P x_N.$$

ただし、 $P$  はリカッチ方程式の解で、次を満たす。

$$P = A^T P A + Q - A^T P B (B^T P B + R)^{-1} B^T P A \quad (9)$$

さらに、 $P, Q, R \succ^p 0$  である。

まず、最初のステップの  $k = N - 1$  の場合について議論する。

$$\begin{aligned} W_1(x_{N-1}) &= V_0(x_N) + x_{N-1}^T Q x_{N-1} + u_{N-1}^T R u_{N-1} \\ &= x_N^T P x_N + x_{N-1}^T Q x_{N-1} + u_{N-1}^T R u_{N-1} \quad (\text{式 (8)}) \\ &= (A x_{N-1} + B u_{N-1})^T P (A x_{N-1} + B u_{N-1}) + x_{N-1}^T Q x_{N-1} + u_{N-1}^T R u_{N-1} \quad (\text{式 (6)}) \\ &= x_{N-1}^T (A^T P A + Q) x_{N-1} + x_{N-1}^T A^T P B u_{N-1} + u_{N-1}^T B^T P A x_{N-1} + u_{N-1}^T (B^T P B + R) u_{N-1} \\ &= (u_{N-1} + M x_{N-1})^T (B^T P B + R) (u_{N-1} + M x_{N-1}) + x_{N-1}^T (A^T P A + Q - M^T (B^T P B + R) M) x_{N-1} \\ &\quad \left( M := \underbrace{(B^T P B + R)^{-1}}_{\geq^p 0} \underbrace{B^T P A}_{>^p} \text{ として平方完成} \right) \end{aligned} \quad (10)$$

式 (10) より、 $u_{N-1} = -M x_{N-1}$  で  $W_1(x_{N-1})$  は最小値をとる。  
したがって、次が成り立つ。

$$V_1(x_{N-1}) = x_{N-1}^T (A^T P A + Q - M^T (B^T P B + R) M) x_{N-1} \quad (11)$$

式 (8) の  $k = 0, 1, \dots, N - 2$  においても、式 (10) を導いた議論と同様の議論ができる。

すなわち、次のように  $M_k$ ,  $k = 0, 1, \dots, N - 1$  および  $P_k$ ,  $k = 0, 1, \dots, N$  を定めると、式 (10) を導いた議論の  $P$  を  $P_k$  に、 $M$  を  $M_k$  に置き換えた議論ができ、 $W_k(x_k)$ ,  $k = 0, 1, \dots, N - 1$  が  $u_k = -M_k x_k$  で最小値をとることが分かる。

$$\begin{aligned} M_k &= (B^T P_{k+1} B + R)^{-1} B^T P_{k+1} A, \quad k = 0, 1, \dots, N - 1, \\ P_k &= A^T P A + Q - M_k^T (B^T P B + R) M_k, \quad k = 0, 1, \dots, N - 1, \\ P_N &= P. \end{aligned} \quad (12)$$

以上の議論をまとめて、次の制御則が得られる。

$$u_k = -M_k x_k, \quad k = 0, 1, \dots, N - 1. \quad (13)$$

### (3)

#### a )

システムの未来の出力をモデルによって推定して、それに基づいて制約条件などを定めて最適制御の入力を求める。

b )

- 複数入出力系を自然に扱うことができる.
- システムのもつ制約を最適化問題の制約として自然に扱うことができる.
- 状態空間やステップ応答など, 出力の推定に用いるモデルに制限がない.

c )

- 最適化問題を実時間で解く必要がある (オンライン計算が必要).
  - 要求される計算資源が大きい.
  - 最適化問題の計算 (反復解法) に時間がかかる.

d )

現在の状態がどの最適な入力に対応するのかを確かめるだけで入力を定めることができるため, 最適化問題をオンラインで解く必要がない. オフラインの計算によって状態と最適な入力の関係を調べておけばよい.