確率・統計処理 & 真値推定

カルマン・フィルタ 入門

Part 4: 現代制御理論

リニア・テック　別府 伸耕

linear tec : Nobuyasu Beppu

現代制御理論

Sample

1. イントロ
2. 状態方程式
3. 可制御性と可観測性
4. 安定性
5. 状態フィードバック
6. オブザーバ
7. 最適制御

線形微分方程式：linear differential equation

Sample

システムの挙動は，時間tの関数“$x(t)$”によって表されるとする.
次式のように，関数 $x(t)$ の導関数 “$x^{(1)}(t), x^{(2)}(t), \cdots, x^{(n)}(t)$” の「線形結合」を考える.
（“$a_0, a_1, \cdots, a_n$” は定数，“$b(t)$” は何らかの関数とする）

$$\boxed{a_n x^{(n)}(t) + a_{n-1} x^{(n-1)}(t) + \cdots + a_1 x^{(1)}(t) + a_0 x(t) = b(t)}$$

上式のような微分方程式を「線形微分方程式」という.
特に，上式は第n階の導関数まで使われているため，「n階線形微分方程式」と呼ばれる.

※ n階線形微分方程式は，フーリエ変換やラプラス変換によって単なる「代数方程式」に変形できる.

線形微分方程式でその挙動が表されるシステムを「線形システム」（linear system）という.
線形システムは特性が素直で扱いやすい.

特に，1階の導関数“ $\dot{x}(t)$ ”だけを含む線形微分方程式を「1階線形微分方程式」と呼ぶ.
1階線形微分方程式は，ラプラス変換などに頼らなくとも簡単に解ける.

$$\boxed{\dot{x}(t) = a \cdot x(t) + b \cdot u(t)}$$

状態方程式：state equation

Sample

一般のn階線形微分方程式で表された線形システムの挙動を表すために，
次のように関数 “$x(t), x^{(1)}(t), \cdots, x^{(n-1)}(t)$” を束ねた「$n$次元ベクトル “$\boldsymbol{x}(t)$”」を考える．

$$\boldsymbol{x}(t) = \begin{pmatrix} x(t) \\ x^{(1)}(t) \\ \vdots \\ x^{(n-1)}(t) \end{pmatrix}$$

n階の線形微分方程式は，係数行列 “A” および “B” と，
時間tの関数をまとめたn次元ベクトル “ $\boldsymbol{u}(t)$” を使って，「1階微分方程式のように」書ける．

$$\boxed{\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + B\boldsymbol{u}(t)}$$

上式は線形システムを表現する微分方程式の一般形であり「状態方程式」と呼ばれる．
各項の意味は次のとおり．

- ベクトル $\boldsymbol{x}(t)$：「状態ベクトル」（state vector）．システムの状態を表す．
- ベクトル $\boldsymbol{u}(t)$：「入力ベクトル」（input vector）．システムに対する入力を表す．
- 行列 A：「状態行列」（state matrix）．システムの挙動を表す．
- 行列 B：「入力行列」（input matrix）．入力信号$\boldsymbol{u}(t)$がシステムに及ぼす影響を表す．

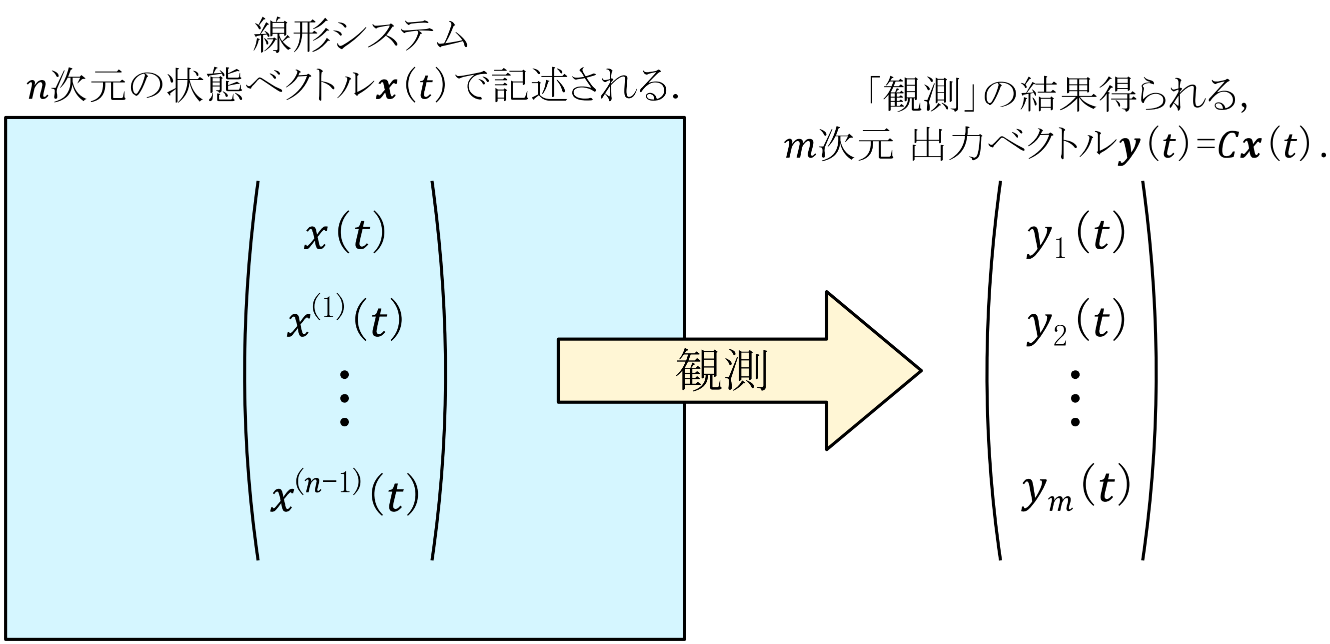
出力方程式: output equation

Sample

状態ベクトル$\boldsymbol{x}(t)$で表されるシステムの状況はセンサなどで「観測」する.
しかし，すべての情報を入手できるとは限らない.

実際に「観測」によって入手できるシステムの情報を「出力ベクトル“$\boldsymbol{y}(t)$”」で表す.
状態ベクトル“$\boldsymbol{x}(t)$”と出力ベクトル“$\boldsymbol{y}(t)$”との関係を「出力方程式」で表す.
なお，行列“C”は$m \times n$行列であり「出力行列」（output matrix）と呼ばれる.

$$\boldsymbol{y}(t) = C\boldsymbol{x}(t)$$

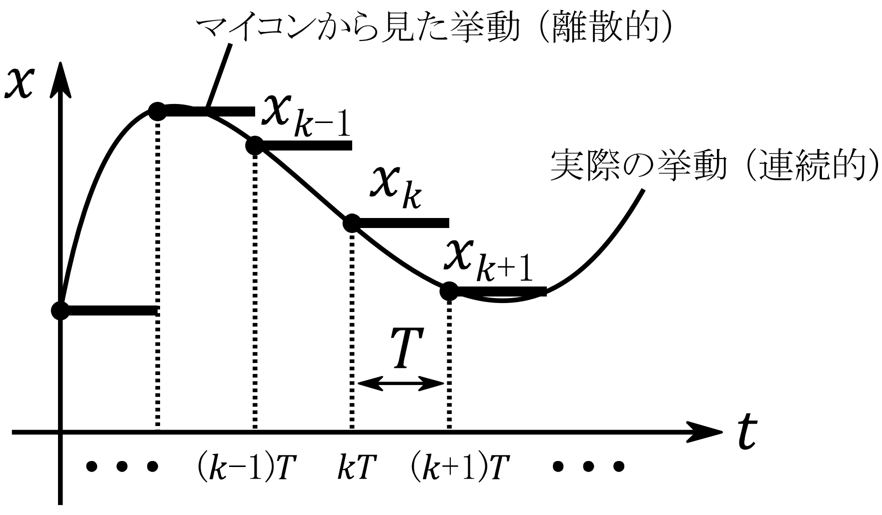


離散時間システム：discrete time system

実際にロボットなどを制御する場合は，マイコンなどを使用することになる．
マイコンでセンサ値をサンプリングしたり，制御電圧を出力したりする場合，
（タイマ割込みなどを使って）一定の時間間隔 “T” で処理を行う．
このTを「制御周期」と呼ぶ．

時間間隔Tで時間的に「ぶつ切り」にして扱うシステムのことを「離散時間システム」という．
逆に，いままで考えてきた連続時間tの流れに沿ったシステムを「連続時間システム」という．

マイコンを使った制御システムでは，一定の時間間隔でセンサ値を取得する．
次の測定タイミングまで前のタイミングで取得したセンサ値がずっと続いていると見なす．
このような考えを「ゼロ次ホールド」（zero order hold）という．

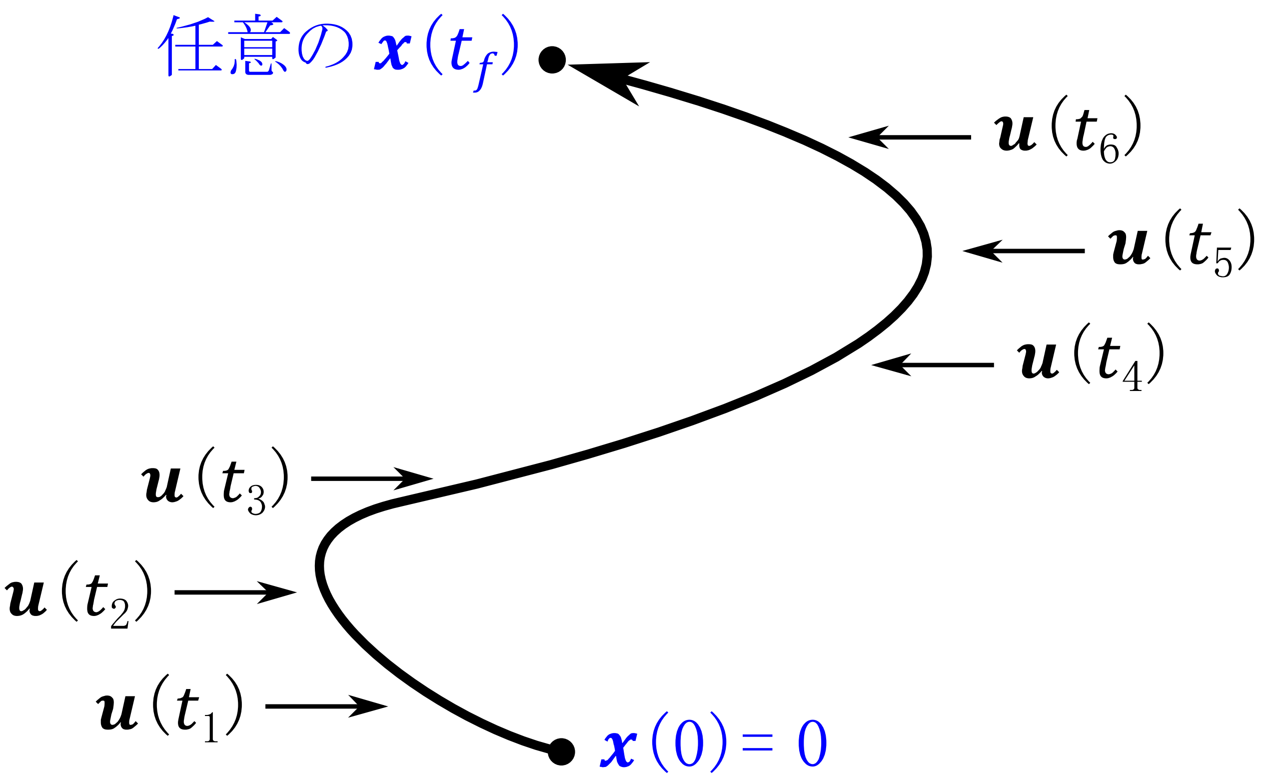


可制御性：controllability

Sample

システムの「可制御性」は次のように定義される.

初期状態 “$\boldsymbol{x}(0) = \boldsymbol{0}$” に対して，“$0 \leq t \leq t_f$” の間に適切な入力 “$\boldsymbol{u}(t)$” を印加すれば，有限の時刻 “$t_f$” において任意の終状態 “$\boldsymbol{x}(t_f)$” に遷移できる場合，そのシステムは「可制御」(controllable) であるという.

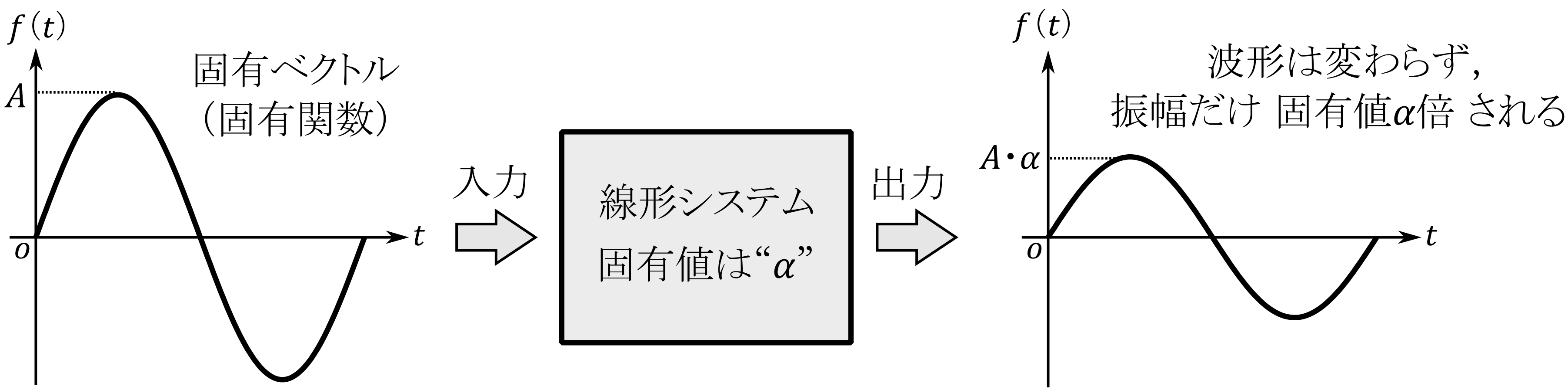


システムの安定判別法 （離散時間システム）

入力を常に "$\boldsymbol{u}_k = \boldsymbol{0}$" とした離散時間システム "$\boldsymbol{x}_{k+1} = A_d\boldsymbol{x}_k$" の安定性について，次の定理が成り立つ.

> 離散時間システムが安定であるための必要十分条件は，
> 行列 "A_d" のすべての**固有値**の絶対値が "1" より小さいことである.

※ 次図は「**線形システム**」と「**固有値**」の直感的なイメージ.



"$\boldsymbol{x}_{k+1} = A_d\boldsymbol{x}_k$" という関係式は，前時刻の出力を入力に入れ続ける様子を意味する.

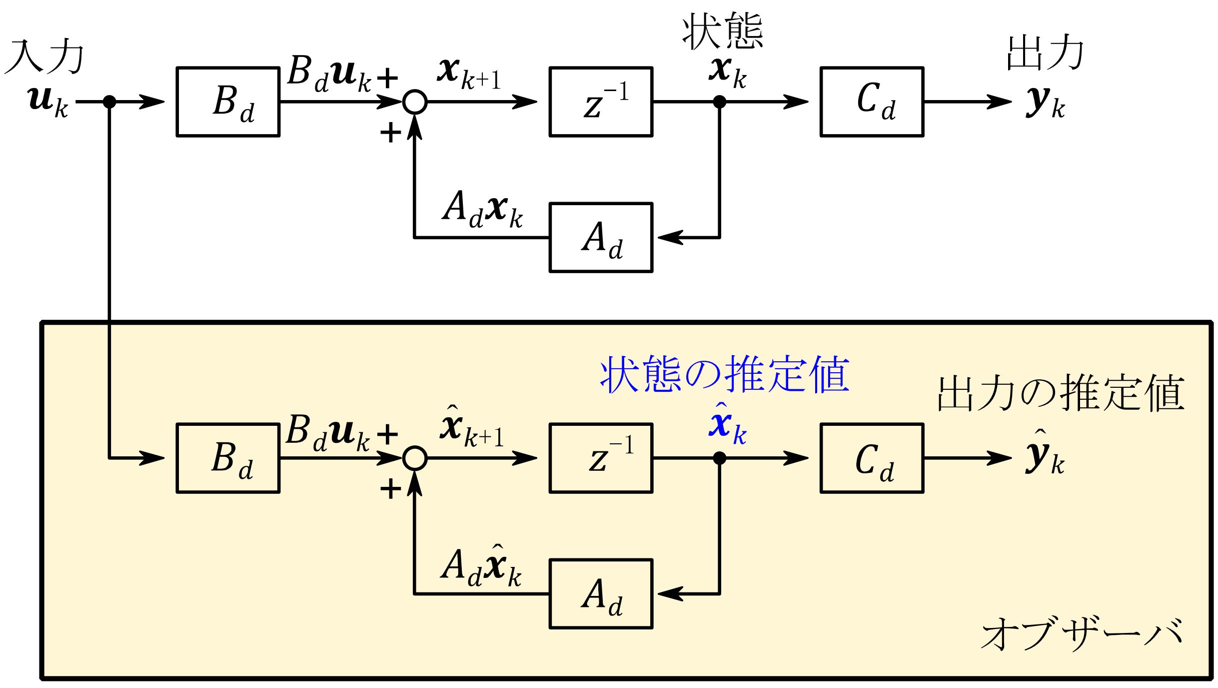
最も単純なオブザーバ (1)

Sample

「状態フィードバック」を行うには，システムの状態 $\boldsymbol{x}_k$ のすべての情報を必要とする．
しかし，システムによっては 状態 $\boldsymbol{x}_k$ の一部の成分しか 「観測」 できないことがある．

$$（例）\quad \boldsymbol{y}_k = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = C_d \boldsymbol{x}_k = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{k1} \\ x_{k2} \\ x_{k3} \end{pmatrix} = \begin{pmatrix} x_{k1} \\ x_{k3} \end{pmatrix}$$

出力 $\boldsymbol{y}_k$ の情報が足りないときに，状態 $\boldsymbol{x}_k$の「推定値 “$\hat{\boldsymbol{x}}_k$”」を求める手法が考案された．
これを「オブザーバ」（observer）あるいは「状態観測器」という．



最適レギュレータと評価関数

Sample

システムの状態を一定値（通常は $\boldsymbol{x}_k = \boldsymbol{0}$ ）に安定させるための制御装置を，
「レギュレータ」（regulator） あるいは「調整器」という.
特に，制御の「効率」が最も良いレギュレータを「最適レギュレータ」という.

「効率の良さ」を定量化するために，次の「評価関数」“J_d” を考える.
この評価関数はベクトルの2次形式を使っているので，「2次形式評価関数」と呼ばれる.
この “J_d” の値（これはスカラ）が小さいほど，「効率が良い制御」であると言える.

$$J_d = \sum_{k=0}^{\infty} \left(\boldsymbol{x}_k^T Q_d \boldsymbol{x}_k + \boldsymbol{u}_k^T R_d \boldsymbol{u}_k \right)$$

上式に含まれる “Q_d” および “R_d” は，通常は次のような対角行列を使う.
ただし，すべての成分は正の値とする.

$$Q_d = \begin{pmatrix} q_1 & 0 & \cdots & 0 \\ 0 & q_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & q_n \end{pmatrix} \qquad R_d = \begin{pmatrix} r_1 & 0 & \cdots & 0 \\ 0 & r_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & r_n \end{pmatrix}$$

上記の行列 Q_d および R_d は，優先するパラメータを指定するための「重み行列」である.