プロジェクト実習II　システム・制御

注意

1. 実習開始までに最低限 3 までを熟読しておくこと．
2. ノート PC などを所有している受講生は，10.1 に記載されている数値計算ソフトウエア Scilab をインストールした上で，そのノート PC などを持参して使用してもかまわない．

1　目的と概要

様々な問題や事象を工学的に取り扱うとき，それらをシステムとして捉えて取り扱うという考え方は重要である．また，制御はシステムの代表的な応用例の一つとして知られている．そこで本実習では，システムの最適化，設計，解析，性能評価などの各手順を経験し，システム工学的なセンスを養う．また，これを通じて，システムと制御におけるいくつかの手法を体得する．

具体的には，まず線形最適化，システム，制御に関する基礎的な実習を行う．次に，ベンチマーク的な問題である倒立振子制御システムの設計を通して，より実践的な実習を行う．これらを通して，上記の目的を達成する．

2　線形最適化

何らかの設計をするとき，いくつかの変数の値を最適に決定することを通して設計することが多い．例えば燃費のよい車の形状を決定する場合，燃費の点で最適な車の高さや幅といった変数の値を決定することになる．このような問題は最適化問題と呼ばれており，その中でも下記の標準形で与えられる線形最適化問題では，現実的な問題を必ず解くことができる方法が開発されている．

$$
\begin{aligned}
\min \quad & f = c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
\text{s.t.} \quad & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\
& a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\
& \vdots \\
& a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \\
& x_i \geqq 0 \ (i = 1, 2, \cdots, n)
\end{aligned}
\tag{1}
$$

ここで，$x_1, x_2, \cdots, x_n$ は最適な値を決定したい変数であり，決定変数と呼ばれる．$c_1, c_2, \cdots, c_n$ は予め与えられる定数，f は最適化の目的を表す $x_1, x_2, \cdots, x_n$ の関数で，目的関数と呼ばれる．min は f の最小化を意味している．車の形状を決める問題では，燃費が目的関数，車の高さや幅が決定変数となる．$a_{11}, a_{12}, \cdots, a_{mn}, b_1, b_2, \cdots, b_m$ も定数である．2 行目の s.t. は subject to を省略したものであり，2 行目以降は決定変数に関する制約条件を表している．例えば，車の高さ x_1 と幅 x_2 の和が 4[m] であるという制約条件があるとすると，それは $x_1 + x_2 = 4$ で表現できる．以上より，線形最適化問題は 2 行目以降の制約条件式を満たす決定変数の値の中で，目的関数 f の値を最小にする決定変数の値を求める問題である．このときの決定変数の値を最適解という．

問題 (1) では目的関数の最小化であったり，制約条件式が等式であったりするが，目的関数の最大化であったり，制約条件式が不等式であったりしても，それと等価な標準形に変換できる．例えば，問題

$$
\begin{aligned}
\max \quad & f' = x_1 + 2x_2 \\
\text{s.t.} \quad & x_1 + 3x_2 \leqq 9 \\
& 2x_1 + x_2 \leqq 8 \\
& x_1 \geqq 0, \ x_2 \geqq 0
\end{aligned}
\tag{2}
$$

は，f' の最大化が $-f'$ の最小化と同じであることと，制約条件式 $x_1 + 3x_2 \leqq 9$ が新しい決定変数 x_3 を加えて $x_1 + 3x_2 + x_3 = 9, \ x_3 \geqq 0$ とできることなどから，標準形

$$
\begin{aligned}
\min \quad & f = -x_1 - 2x_2 \\
\text{s.t.} \quad & x_1 + 3x_2 + x_3 = 9 \\
& 2x_1 + x_2 + x_4 = 8 \\
& x_1 \geqq 0, \ x_2 \geqq 0, \ x_3 \geqq 0, \ x_4 \geqq 0
\end{aligned}
\tag{3}
$$

に変換できる．この新しく加えた変数（ここでは x_3 と x_4）をスラック変数という．

問題 (2) において制約条件式を満たす決定変数の値の範囲は，4 つの制約条件式で定まる四角形の辺と内部である．また，最適解は $x_1 = 3, \ x_2 = 2$ であり，これは四角形の頂点の 1 つである．一般の問題においても同様に制約条件式で定まる多角形の頂点の 1 つが最適解となるこ

とが知られている．また，各頂点は標準形 (1) で $n-m$ 個の決定変数の値を 0 と与え，残りの m 個の決定変数の値を制約条件式を満たす値で与えることによって求められる．例えば，標準形 (3) では $m=2,\ n=4$ であるので，$x_1=0,\ x_2=0,\ x_3=9,\ x_4=8$ としたものは，元の問題 (2) の制約条件式で定まる四角形の頂点の 1 つ $(x_1,x_2)=(0,0)$ となっている．このような頂点における変数の値を実行可能基底解という．また，0 と与える決定変数を非基底変数，それ以外の決定変数を基底変数という．

3　シンプレックス法

実行可能基底解の中の 1 つに最適解があるので，線形最適化問題を解く代表的な方法であるシンプレックス法では，この実行可能基底解を一つずつ調べることで最適解を求めようとする．シンプレックス法では，調べている実行可能基底解が最適解かどうかを判別することができ，また次の実行可能基底解を調べるときに目的関数の値が増加しない実行可能基底解を調べることができる．これらによって，効率的に最適解を発見することができる．シンプレックス法の流れを下記に示す．

1. 初期の実行可能基底解を与える．
2. 実行可能基底解が最適解かどうかを判別し，最適解なら終了する．
3. 別の実行可能基底解を生成し，2. へ戻る．

以下で各ステップを説明するが，理論的な部分の詳細は参考文献を参照すること．

まず，1. ではスラック変数を基底変数，その他の変数を非基底変数とすると容易に実行可能基底解を与えることができる場合は，そのように与える．例えば，標準形 (3) では $x_1=x_2=0$ とすると容易に $x_3=9,\ x_4=8$ と得られるので，これを初期の実行可能基底解とできる．このようにして初期の実行可能基底解を与えられない場合は 2 段階法を用いるが，その詳細は省略する．

2. と 3. の計算はタブローと呼ばれるものを用いると計算しやすい．初期のタブローは，標準形 (3) に対しては，

-1	-2	0	0	0
1	3	1	0	9
2	1	0	1	8

(4)

となる．1 行目は目的関数の係数 c_1,c_2,c_3,c_4 ($c_3=c_4=0$ に注意）と，初期の実行可能基底解に対する目的関数 f の値に -1 を乗じた値を書く．2 行目は最初の制約条件式の定数 $a_{11},a_{12},a_{13},a_{14},b_1$，3 行目は 2 番目の制約条件式の定数 $a_{21},a_{22},a_{23},a_{24},b_2$ を書く．このように m 行にわたって a_{ij} と b_j を書く．左の 4 列の各列は各決定変数 x_1,x_2,x_3,x_4 に，下の 2 行の各行は一つの基底変数にそれぞれ対応する．各列において 1 行だけ 1 となって他の行がいずれも 0 である列に対応する決定変数が基底変数であり，その値はその行の右端で示される値である．この例では，左から 3,4 列目がそれにあたり，したがって x_3,x_4 が基底変数である．x_3 が 2 行目，x_4 が 3 行目に対応しており，それらの値はそれぞれ 9,8 となっている．その他の決定変数（ここでは x_1,x_2）は非基底変数なので，これらの値は 0 である．

シンプレックス法の流れの 2. を説明する．現在の実行可能基底解が最適解かどうかはタブローの 1 行目の値（ただし右端を除く）に注目し，これらの全ての値が 0 以上ならば最適解が得られており，シンプレックス法を終了させる．標準形 (3) の例ではタブロー (4) において -1 と -2 があるので最適解は得られておらず，これは目的関数値が改善できることを示している．

流れの 3. では別の実行可能基底解を生成する．このために非基底変数 1 つを基底変数に，基底変数 1 つを非基底変数に変更する．非基底変数から基底変数に変更する決定変数は，先ほどの 2. で最適解とならない要因となった負の数に対応する決定変数とすればよく，その中でも最小の負の数に対応する決定変数とする．この決定変数に対応する列を j^* で表記する．タブロー (4) の例では，-2 の列に対応する x_2 が基底変数に変更されることになり，$j^*=2$ となる．

次に，基底変数から非基底変数に変更する決定変数を決める．それは，2 行目以降の各行 $i(\geq 2)$ に対して，その行の j^* 列の値で右端の値を除した値 q_i を算出し，それらの中で正で最小の数となる行に対応する基底変数とする．この行を i^* で表記する．タブロー (4) の例では，$q_2=9\div 3=3$，$q_3=8\div 1=8$ なので $i^*=2$ となり，この行に対応する x_3 が非基底変数となる．i^* は q_i が負数となる行とはならないことに注意する．

最後に基底変数の値を求めるためにタブローを更新する．まず i^* 行の各列の値を i^* 行 j^* 列の値で割る．この結果，i^* 行 j^* 列の値は 1 となる．タブロー (4) の例では，2 行目が $\frac{1}{3},\ 1,\ \frac{1}{3},\ 0,\ 3$ となる．次に i 行 j 列の値を T_{ij} としたとき，i^* 行以外の各 T_{ij} を

$$T_{ij}\leftarrow T_{ij}-T_{i^*j}T_{ij^*}\quad(\forall i,j;\ i\neq i^*)\tag{5}$$

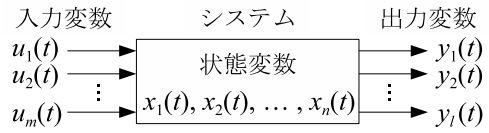


図 1: 多入力多出力システム

とする．この式の右辺で用いる T_{ij} の値は，いずれも更新後の値ではなく更新前の値であることに注意する．この更新の結果，j^* 列で i^* 行以外の各行の値は 0 となる．

タブロー (4) の例では，下記のようになる．

$-\frac{1}{3}$	0	$\frac{2}{3}$	0	6
$\frac{1}{3}$	1	$\frac{1}{3}$	0	3
$\frac{5}{3}$	0	$-\frac{1}{3}$	1	5

例えば，T_{31} は $T_{31} \leftarrow T_{31} - T_{21}T_{32} = 2 - \dfrac{1}{3} \times 1 = \dfrac{5}{3}$ と更新されている．この結果，新しい実行可能基底解は $x_1 = 0,\ x_2 = 3,\ x_3 = 0,\ x_4 = 5$ である．まだ，最適解が求められていないので，流れの 3. を再度実行し，タブローは下記のようになる．

0	0	$\frac{3}{5}$	$\frac{1}{5}$	7
0	1	$\frac{2}{5}$	$-\frac{1}{5}$	2
1	0	$-\frac{1}{5}$	$\frac{3}{5}$	3

これで問題 (2) の最適解 $x_1 = 3,\ x_2 = 2$ が求められた．

4　システムのモデル表現

ここではシステムの表現法について必要事項をまとめておく．

4.1　状態空間モデル

一般にシステムにおいては図 1 に示すように 3 種類の変数が存在し，これら 3 種類の変数によってシステムは記述，表現される．その一つはシステムの外部の変数である m 個の入力変数 $u_1(t), u_2(t), \cdots, u_m(t)$ で，これらの変数はシステムに影響を及ぼすが逆にシステムがこれらの変数に影響を及ぼすことのない変数である．また，他の一つはシステムの内部の状態を表す変数で，いくつか適当な数からなり，この数を n とすると $x_1(t), x_2(t), \cdots, x_n(t)$ と表され状態変数と呼ばれる．最後の一つは ℓ 個の出力変数 $y_1(t), y_2(t), \cdots, y_\ell(t)$ で，通常そのシステムを解析，設計するにあたって関心の対象となる変数，あるいは測定などにより設計者が入手できる変数である．なお，t は時間で，各変数が時間の関数であることを表している．

このようなシステムは次のモデルで表される．

$$\begin{aligned}
\dot{x}_1(t) &= f_1(x_1(t), x_2(t), \cdots, x_n(t), \\
&\qquad u_1(t), u_2(t), \cdots, u_m(t)) \\
\dot{x}_2(t) &= f_2(x_1(t), x_2(t), \cdots, x_n(t), \\
&\qquad u_1(t), u_2(t), \cdots, u_m(t)) \\
&\vdots \\
\dot{x}_n(t) &= f_n(x_1(t), x_2(t), \cdots, x_n(t), \\
&\qquad u_1(t), u_2(t), \cdots, u_m(t))
\end{aligned} \tag{6}$$

ここで，記号 $\dot{}$ は時間 t による微分 $\dfrac{d}{dt}$ を示す．また，$f_1, f_2, \cdots, f_n$ は状態変数と入力変数に関する関数である．通常多くのシステムにおいては，その状態変数の挙動，すなわち各状態変数が時間 t とともにどのように変化するのかが，入力変数を用いてこの一階の連立微分方程式で表される．出力変数は状態変数と入力変数に関する関数 $g_1, g_2, \cdots, g_\ell$ を用いて次のように表される．

$$\begin{aligned}
y_1(t) &= g_1(x_1(t), x_2(t), \cdots, x_n(t), \\
&\qquad u_1(t), u_2(t), \cdots, u_m(t)) \\
y_2(t) &= g_2(x_1(t), x_2(t), \cdots, x_n(t), \\
&\qquad u_1(t), u_2(t), \cdots, u_m(t)) \\
&\vdots \\
y_\ell(t) &= g_\ell(x_1(t), x_2(t), \cdots, x_n(t), \\
&\qquad u_1(t), u_2(t), \cdots, u_m(t))
\end{aligned} \tag{7}$$

(6) 式は状態方程式，また (7) 式は出力方程式とそれぞれ呼ばれ，これらを合わせたものがシステムの状態空間モデルである．この状態空間モデルは通常次のようにベクトルを用いて簡潔に表現される．

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \tag{8}$$

$$\boldsymbol{y}(t) = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \tag{9}$$

ただし，$\boldsymbol{x}$，$\boldsymbol{u}$，$\boldsymbol{y}$，$\boldsymbol{f}$，$\boldsymbol{g}$ は次のように表される．

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix},\ \boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix},\ \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_\ell \end{bmatrix},$$

$$\boldsymbol{f} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix},\ \boldsymbol{g} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_\ell \end{bmatrix}$$

また，ベクトルの微分 $\dot{\boldsymbol{x}}(t)$ は各成分の微分で定義される．

4.2 線形近似システムの導出

システムは線形と非線形に分類される．システムが線形であるとは，入力 $\boldsymbol{u}_1(t)$ に対する出力が $\boldsymbol{y}_1(t)$，$\boldsymbol{u}_2(t)$ に対する出力が $\boldsymbol{y}_2(t)$ であるときに，入力 $a\boldsymbol{u}_1(t) + b\boldsymbol{u}_2(t)$（$a, b$ は任意の実数）に対する出力が $a\boldsymbol{y}_1(t) + b\boldsymbol{y}_2(t)$ となることである．また，線形でないシステムを非線形システムといい，実際のシステムのほとんどは非線形システムとなる．ところが一般に非線形システムを解析，設計するのは非常に困難であるので，これを線形システムによって近似的に表現することが通常行われる．以下ではその線形近似システムの導出法について述べる．(8)(9) 式の状態空間モデルで表現されたシステムは，入力を時間に関して一定な入力 ($\boldsymbol{u}(t) = \boldsymbol{u}^*$) とすると，十分時間が経過した（理論的には $t \to \infty$）状態では，状態変数および出力変数も一定の値（その値をそれぞれ $\boldsymbol{x}^*$, $\boldsymbol{y}^*$ とする）をとり，この状態を平衡状態と呼ぶ．明らかにこの平衡状態 $\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{y}^*$ は次式を満足する．

$$\boldsymbol{0} = \boldsymbol{f}(\boldsymbol{x}^*, \boldsymbol{u}^*) \tag{10}$$

$$\boldsymbol{y}^* = \boldsymbol{g}(\boldsymbol{x}^*, \boldsymbol{u}^*) \tag{11}$$

ただし，$\boldsymbol{0}$ は n 次元の零ベクトルである．多くのシステムでは，ある平衡状態を動作点として，そこを基準とした振る舞いを考えることが多い．そのためその平衡状態からのずれ（偏差）を新しい変数として

$$\Delta\boldsymbol{x}(t) = \boldsymbol{x}(t) - \boldsymbol{x}^*,\ \Delta\boldsymbol{u}(t) = \boldsymbol{u}(t) - \boldsymbol{u}^*,\ \Delta\boldsymbol{y}(t) = \boldsymbol{y}(t) - \boldsymbol{y}^* \tag{12}$$

と定義し，また関数 $\boldsymbol{f}$ および $\boldsymbol{g}$ を連続微分可能として (8)(9) 式の右辺を平衡点 $(\boldsymbol{x}^*, \boldsymbol{u}^*)$ のまわりでテイラー展開する．たとえば (8) 式および (9) 式の第 k 行目の式は次のようになる．

$$\begin{aligned} \Delta\dot{x}_k(t) = {} & f_k(\boldsymbol{x}^*, \boldsymbol{u}^*) \\ & + a_{k1}\Delta x_1(t) + a_{k2}\Delta x_2(t) + \cdots + a_{kn}\Delta x_n(t) \\ & + b_{k1}\Delta u_1(t) + b_{k2}\Delta u_2(t) + \cdots + b_{km}\Delta u_m(t) \\ & + O(\Delta\boldsymbol{x}, \Delta\boldsymbol{u}) \end{aligned} \tag{13}$$

$$\begin{aligned} \Delta y_k(t) + y^* = {} & g_k(\boldsymbol{x}^*, \boldsymbol{u}^*) \\ & + c_{k1}\Delta x_1(t) + c_{k2}\Delta x_2(t) + \cdots + c_{kn}\Delta x_n(t) \\ & + d_{k1}\Delta u_1(t) + d_{k2}\Delta u_2(t) + \cdots + d_{km}\Delta u_m(t) \\ & + O(\Delta\boldsymbol{x}, \Delta\boldsymbol{u}) \end{aligned} \tag{14}$$

ただし

$$a_{ki} = \left.\frac{\partial f_k(\boldsymbol{x}, \boldsymbol{u})}{\partial x_i}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}^* \\ \boldsymbol{u}=\boldsymbol{u}^*}},\ b_{kj} = \left.\frac{\partial f_k(\boldsymbol{x}, \boldsymbol{u})}{\partial u_j}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}^* \\ \boldsymbol{u}=\boldsymbol{u}^*}}$$

$$c_{ki} = \left.\frac{\partial g_k(\boldsymbol{x}, \boldsymbol{u})}{\partial x_i}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}^* \\ \boldsymbol{u}=\boldsymbol{u}^*}},\ d_{kj} = \left.\frac{\partial g_k(\boldsymbol{x}, \boldsymbol{u})}{\partial u_j}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}^* \\ \boldsymbol{u}=\boldsymbol{u}^*}}$$

である．ここで，$O(\Delta\boldsymbol{x}, \Delta\boldsymbol{u})$ は $\Delta\boldsymbol{x}$，$\Delta\boldsymbol{u}$ に関する 2 次以上の高次の項を表す．また，記号 $\left|_{\substack{\boldsymbol{x}=\boldsymbol{x}^* \\ \boldsymbol{u}=\boldsymbol{u}^*}}\right.$ は偏微分した後に $\boldsymbol{x} = \boldsymbol{x}^*$, $\boldsymbol{u} = \boldsymbol{u}^*$ を代入することを表す．システムは平衡点の近傍で動作すると仮定すると，偏差は微少量と見なせるので，これに関する高次の項 $O(\Delta\boldsymbol{x}, \Delta\boldsymbol{u})$ は無視することができ，また平衡状態の条件 (10)(11) 式より

$$f_k(\boldsymbol{x}^*, \boldsymbol{u}^*) = 0,\ \ y_k^* = g_k(\boldsymbol{x}^*, \boldsymbol{u}^*) \tag{15}$$

であることに注意すると，(8)(9) 式は次のように近似できる．

$$\Delta\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\Delta\boldsymbol{x}(t) + \boldsymbol{B}\Delta\boldsymbol{u}(t) \tag{16}$$

$$\Delta\boldsymbol{y}(t) = \boldsymbol{C}\Delta\boldsymbol{x}(t) + \boldsymbol{D}\Delta\boldsymbol{u}(t) \tag{17}$$

ただし，$\boldsymbol{A}$，$\boldsymbol{B}$，$\boldsymbol{C}$，$\boldsymbol{D}$ は次の行列である．

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

$$\boldsymbol{B} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \vdots & \vdots & & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nm} \end{bmatrix}$$

$$\boldsymbol{C} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & & \vdots \\ c_{\ell 1} & c_{\ell 2} & \cdots & c_{\ell n} \end{bmatrix}$$

$$\boldsymbol{D} = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1m} \\ d_{21} & d_{22} & \cdots & d_{2m} \\ \vdots & \vdots & & \vdots \\ d_{\ell 1} & d_{\ell 2} & \cdots & d_{\ell m} \end{bmatrix}$$

以上で得られた (16)(17) 式が，(8)(9) 式で表される非線形システムの振る舞いを近似する線形システムの状態空間モデルである．

5　制御理論の基礎

この節では，制御理論のうち特に本実習で必要となる事項をまとめておく．詳細は末尾に挙げる参考文献などで各自勉強すること．以下では，状態空間モデル (16)(17) 式において偏差を表す記号 Δ を省略し，システムが

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \tag{18}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t) \tag{19}$$

なる状態空間モデルで表されるとして議論をすすめる．

5.1　状態方程式の解

ここでは，(18) 式の状態方程式の解を考える．このため，まず入力を $\boldsymbol{u}(t) = \boldsymbol{0}$，初期値を $\boldsymbol{x}_0$ として

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{20}$$

の解 $\boldsymbol{x}(t)$ を考える．この解はスカラーの微分方程式 $\dot{x}(t) = ax(t),\ x(0) = x_0$ の解が $x(t) = e^{at}x_0$ と表されるのと同様に次のように表される．

$$\boldsymbol{x}(t) = e^{\boldsymbol{A}t}\boldsymbol{x}_0 \tag{21}$$

ただし $e^{\boldsymbol{A}t}$ は状態遷移行列と呼ばれる $n \times n$ の行列で，次の様に行列 $\boldsymbol{A}$ の無限級数で定義される．

$$e^{\boldsymbol{A}t} = \boldsymbol{I} + \boldsymbol{A}t + \boldsymbol{A}^2\frac{t^2}{2!} + \boldsymbol{A}^3\frac{t^3}{3!} + \cdots + \boldsymbol{A}^k\frac{t^k}{k!} + \cdots \tag{22}$$

ここで，$\boldsymbol{I}$ は $n \times n$ の単位行列である．この状態遷移行列を用いて入力 $\boldsymbol{u}(t)$ がある場合の解は次の様に表される．

$$\boldsymbol{x}(t) = e^{\boldsymbol{A}t}\boldsymbol{x}_0 + \int_0^t e^{\boldsymbol{A}(t-\tau)}\boldsymbol{B}\boldsymbol{u}(\tau)d\tau \tag{23}$$

ここでベクトルの積分は各成分の積分で定義される．上式の右辺第 1 項は初期値に対する応答，また第 2 項は入力に対する応答を表す．

5.2　可制御性と可観測性

可制御性と可観測性はシステムの構造に関する基本的な性質で，可制御性は，システムの入力を操作する事によりシステムの内部の状態を自由に制御できるかどうか，また可観測性は，観測された出力よりシステムの内部の状態をすべて知り得るかどうかという性質のことである．状態空間モデルで制御システムの設計問題を考える上では，まずこれらの性質が前提となる．ここでは可制御性と可観測性の定義とそれらが成立するための必要十分条件を述べる．

[**可制御性**] ある任意の初期状態 $\boldsymbol{x}(0)$ が与えられた時，ある有限の時刻 t_f までの適当な入力 $\boldsymbol{u}(t),\ 0 \leqq t \leqq t_f$ を加える事によって，システムの状態を任意の状態 $\boldsymbol{x}(t_f)$ にすることができるならばこのシステムは可制御であるという．

また，システムが可制御であるための必要十分条件は行列 $\boldsymbol{M}_c$ を

$$\boldsymbol{M}_c = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{A}\boldsymbol{B} & \cdots & \boldsymbol{A}^{n-1}\boldsymbol{B} \end{bmatrix} \tag{24}$$

と定義した時，$\mathrm{rank}\boldsymbol{M}_c = n$ が成立する事である．

[**可観測性**] ある有限の区間 $0 \leqq t \leqq t_f$ におけるシステムの出力 $\boldsymbol{y}(t)$ の値のみから初期状態 $\boldsymbol{x}(0)$ の値を知ることができるならば，このシステムは可観測であるという．

また，システムが可観測であるための必要十分条件は行列 $\boldsymbol{M}_o$ を

$$\boldsymbol{M}_o = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{C}\boldsymbol{A} \\ \vdots \\ \boldsymbol{C}\boldsymbol{A}^{n-1} \end{bmatrix} \tag{25}$$

と定義した時，$\mathrm{rank}\boldsymbol{M}_o = n$ が成立する事である．

5.3　システムの安定性

システムの安定性は制御問題を考える上で最も重要となる概念で，いくつかの定義がある．システムの入力を考えない場合の内部状態の挙動に関するものと，ある条件を満たすような入力を加えた場合の出力の挙動に関するものに大きく分けられ，前者は内部安定性，後者は入出力安定性と呼ばれる．

(18) 式で表されるシステムに対し，入力を零として内部安定性の定義を与えよう．システム (18) の解は (21) 式となるが，任意の初期値 $\boldsymbol{x}_0$ に対して，$\boldsymbol{x}(t)$ が $t \to \infty$ で零に収束する ($\|\boldsymbol{x}(t)\| \to 0$) とき，このシステムは安定（正確には漸近安定）であるという．したがってシステムが漸近安定であるかどうかはシステムの行列 $\boldsymbol{A}$ のみの性質で決まり，次のような条件が知られている．

『システム (18)(19) が漸近安定であるための必要十分条

件は $\boldsymbol{A}$ の固有値の実部がすべて負であることである.』
$\boldsymbol{A}$ の固有値は s を複素数の変数とした方程式 $\det(s\boldsymbol{I}-\boldsymbol{A})=0$ の解として求められる．ここで，det は行列式を表す記号である．また，$\boldsymbol{A}$ の固有値はシステム (18)(19) の極とも呼ばれる．

次に入出力安定性について述べる．システム (18)(19) の初期状態が零である時に，任意の有界な入力，すなわちすべての $t \geqq 0$ に対して $\|\boldsymbol{u}(t)\| \leqq M_1$ となる定数 M_1 が存在する時，その出力 $y(t)$ もまた有界，すなわち $\|\boldsymbol{y}(t)\| \leqq M_2$ なる定数 M_2 がすべての $t \geqq 0$ について存在するとき，そのシステムは有界入力有界出力安定であるという．システム (18)(19) が可制御かつ可観測である場合は有界入力有界出力安定性と漸近安定性は等価となり，それらが成り立つための必要十分条件は一致することが知られている．

6　制御システム設計手順

一般に制御システムの設計は次のような手順で行われる．

Input: 制御対象システム，制御目標
Output: 制御器
Step 1 制御対象の数学モデルを導く（モデリング）.
Step 2 制御システムの設計のための設計仕様を決める.
Step 3 決定した設計仕様を考慮して，Step 1 で導いた数学モデルより制御対象の設計用のモデルを導く.
Step 4 制御方法を検討し，制御器を設計する.
Step 5 設計された制御器の性能を，解析的に評価したり，計算機シミュレーションで評価したりする．その結果，仕様を満たす性能を有していると判断できる場合は次のステップへ進む．そうでなければ必要に応じて Step 4 あるいは Step 2 あるいは Step 1 へ戻る.
Step 6 実機による実験により制御性能を評価する．その結果，仕様を満たしていれば設計を終了する．満たしていなければ必要に応じて Step 4 あるいは Step 2 あるいは Step 1 へ戻る.

7　最適レギュレータ理論による設計

ある平衡状態で動作していたシステムが外乱などによりその平衡状態からずれた場合を考え，そのずれた状態をすみやかに元の平衡状態に戻す制御器の設計問題を考える．このような問題は一般にレギュレータ問題と呼ばれ，それを実現する制御器はレギュレータと呼ばれる．レギュレータは，制御対象の出力を用いて計算された制御入力を制御対象に入力する制御器で，これにより図 2 のような閉ループシステムが構成される．このようにシステムの出力を入力にフィードバックする制御システムをフィードバック制御システムと呼ぶ.

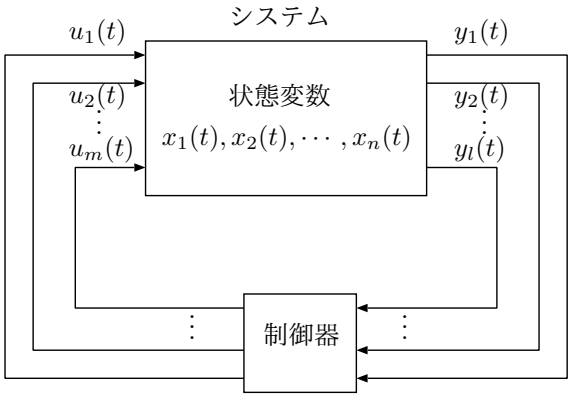


図 2: フィードバック制御システム

ここでは状態方程式 (18) で表されるシステムを考え，また状態 $\boldsymbol{x}$ がすべて直接出力 $\boldsymbol{y}$ として観測できるものとする．すなわち，出力方程式が次式で表されるとし，レギュレータの設計問題を考える.

$$\boldsymbol{y}(t) = \boldsymbol{x}(t) \tag{26}$$

この出力を用いて次のフィードバック入力を考える.

$$\boldsymbol{u}(t) = \boldsymbol{K}\boldsymbol{x}(t) \tag{27}$$

ここで $\boldsymbol{K}$ は $m \times n$ の行列でゲイン行列と呼ばれる．この式を (18) 式に代入すると次の式が得られる.

$$\dot{\boldsymbol{x}}(t) = (\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K})\boldsymbol{x}(t) \tag{28}$$

この様に状態をすべてフィードバックする制御を状態フィードバック制御と呼ぶ．$\boldsymbol{x}(t) = \boldsymbol{0}$ なる平衡状態をとっていたシステムが時間 $t = 0$ で外乱などのため $\boldsymbol{x}(0) = \boldsymbol{x}_0$ にずれたとする．(28) 式の解は $\boldsymbol{x}(t) = e^{(\boldsymbol{A}+\boldsymbol{B}\boldsymbol{K})t}\boldsymbol{x}_0$ と表されるのでこの閉ループシステムが安定であるならば $\|\boldsymbol{x}(t)\| \to 0$ となり，もとの平衡状態に戻る．この様に (27) 式の状態フィードバック制御による閉ループシステムの漸近安定性はシステムの極，すなわち行列 $\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}$ の n 個の固有値で決まる.

ゲイン $\boldsymbol{K}$ を求めるための制御法として，最適レギュレータ理論がある．この制御法では，システムの応答を評価するある適当な指標（評価関数）を設定し，これを最小と

するように制御システムを設計する．評価関数 J は次式で設定する．

$$J = \int_0^\infty \left[\boldsymbol{y}^T(t)\boldsymbol{Q}\boldsymbol{y}(t) + \boldsymbol{u}^T(t)\boldsymbol{R}\boldsymbol{u}(t)\right] dt$$
$$= \int_0^\infty \left[\boldsymbol{x}^T(t)\boldsymbol{Q}\boldsymbol{x}(t) + \boldsymbol{u}^T(t)\boldsymbol{R}\boldsymbol{u}(t)\right] dt \qquad (29)$$

ただし $\boldsymbol{Q}$，$\boldsymbol{R}$ は正定値である実対称行列であり（実対称行列 $\boldsymbol{Q}$ が正定であるとは，零ベクトルでない任意のベクトル $\boldsymbol{v}$ に対して $\boldsymbol{v}^T\boldsymbol{Q}\boldsymbol{v} > 0$ が成り立つことであり，$\boldsymbol{Q}$ が正定であるための必要十分条件は $\boldsymbol{Q}$ の固有値が全て正であることである），T は行列の転置を表す記号である．この様に定義された評価関数において，右辺第 1 項は出力変数および状態変数の平衡状態からの 2 乗積分偏差（より正確には行列 $\boldsymbol{Q}$ で重み付けされた 2 乗積分偏差）を表し，また第 2 項は入力の 2 乗積分（行列 $\boldsymbol{R}$ で重み付けされた 2 乗積分）すなわち制御に必要な入力のエネルギーに相当すると考えることができる．したがってこのように定義された評価関数 J をできるだけ小さくするようなゲインを求めることにより，出力および状態の偏差をできるだけ小さくし，かつそれに必要な入力のエネルギーをできるだけ小さくする制御入力を決めることができるようになる．この様に J を最小化する制御問題は最適レギュレータ問題と呼ばれ，J を目的関数，$\boldsymbol{K}$ を決定変数とすると最適レギュレータ問題は最適化問題の一種ととらえることができる（ただし，線形最適化問題ではないのでシンプレックス法は適用できない）．

最適レギュレータ問題は次のように解かれる．システムが可制御かつ可観測とするとき，(29) 式の評価関数 J を最小にするゲインは一意に決まり，次式として与えられる．

$$\boldsymbol{K} = -\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P} \qquad (30)$$

ただし $\boldsymbol{P}$ は次の代数方程式（リカッチ代数方程式と呼ばれる）

$$\boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P} + \boldsymbol{Q} = \boldsymbol{O} \qquad (31)$$

を満足する実正定行列である．上式において，$\boldsymbol{O}$ は零行列である．このとき，得られた閉ループシステム

$$\dot{\boldsymbol{x}}(t) = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P})\boldsymbol{x}(t) \qquad (32)$$

は漸近安定となる．

8　倒立振子のモデリング

本実習の制御システムの設計における制御対象は倒立振子である．その実験装置は，図 3 に示すように水平方向に移動可能な台車の上に棒状の振子が取り付けられており，この振子は台車の取付位置を中心として平面内を自由に回転できるような機構になっている．台車の車輪にはギヤー（歯車）を介して直流モータがとりつけられており，これに適当な電圧を印加する事により，台車を水平方向に移動させる．したがってこの電圧が制御入力となる．また，車輪および振子の台車上の回転軸にはエンコーダがとりつけられ，台車の位置および振子の回転角度が検出できる．従ってこれらが測定によって得られる出力となる．すなわち一入力二出力のシステムとなる．

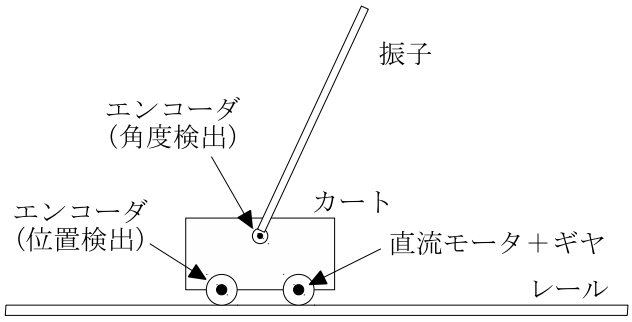


図 3: 倒立振子実験装置

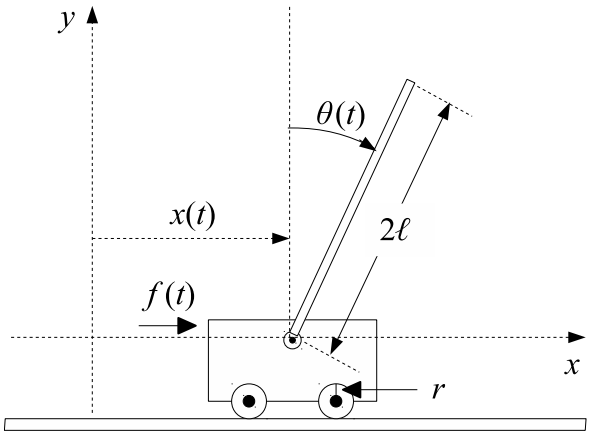


図 4: 倒立振子システムと変数の定義

8.1　モデル方程式

図 4 に示すように時間 t での基準位置（原点）からの台車の（振子取り付け）位置を $x(t)$，振子の鉛直上方からの角度（時計まわり）を $\theta(t)$，また直流サーボモータによる台車の水平方向の移動により台車に加えられる力を $f(t)$ とする．また，台車の質量を m_c，振子の長さを 2ℓ，振子の質量を m_p，振子の重心回りの慣性モーメントを J_p，台

車とレール間の摩擦係数を c_c，振子と台車の間の摩擦係数を c_p，重力加速度を g とすると，ラグランジュの運動方程式よりこのシステムの運動方程式は次のように得られる．

$$(m_c+m_p)\ddot{x}(t)+m_p\ell\ddot{\theta}(t)\cos\theta(t)+c_c\dot{x}(t) \\ -m_p\ell\dot{\theta}^2(t)\sin\theta(t)=f(t) \tag{33}$$

$$(J_p+m_p\ell^2)\ddot{\theta}(t)+m_p\ell\ddot{x}(t)\cos\theta(t)+c_p\dot{\theta}(t) \\ -m_pg\ell\sin\theta(t)=0 \tag{34}$$

ここで記号 ¨ は時間による 2 階微分を表す．上式において，$f(t)$ は台車にとりつけられた直流モータから台車へ伝達された力であり，実際の制御入力はモータへの入力電圧なのでさらに直流モータのモデルを考える必要がある．直流モータのモデルはその電機子に印加する入力電圧を $v_a(t)$，電機子電流を $i_a(t)$，電機子インダクタンスを L_a，電機子抵抗を R_a，電機子の回転角を $\theta_a(t)$，発生トルクを $\tau(t)$ とすると次式のように表される．

$$R_ai_a(t)+L_a\dot{i}_a(t)+K_E\dot{\theta}_a(t)=v_a(t) \tag{35}$$

$$\tau(t)=K_Ti_a(t) \tag{36}$$

ただし K_E，K_T は定数でそれぞれ逆起電力定数およびトルク定数と呼ばれる．また $\tau(t)$ と $f(t)$ との間に次の関係が成立する．

$$f(t)=\frac{K_g}{r}\tau(t) \tag{37}$$

ここで K_g はギヤ比，また r は台車の駆動車輪の半径である．さらにモータの回転角 $\theta_a(t)$ と台車の駆動車輪の回転角 $\theta_c(t)$ との間には $\theta_a(t)=K_g\theta_c(t)$ との関係があるので台車の位置 $x(t)$ と $\theta_a(t)$ との関係は次式で与えられる．

$$x(t)=r\theta_c(t)=\frac{r}{K_g}\theta_a(t) \tag{38}$$

したがって速度の関係は次式となる．

$$\dot{x}(t)=r\dot{\theta}_c(t)=\frac{r}{K_g}\dot{\theta}_a(t) \tag{39}$$

以上で得られた (33)〜(39) 式が本実習で対象とする倒立振子システムのモデル式である．

8.2 状態空間モデルの導出

前項で得られた (33)〜(39) 式より制御を考えるため，状態空間モデルを導く．まず (33)〜(39) 式のままでは非常に複雑で制御システムの設計が困難となるため，$J_p=0,\ c_c=0,\ c_p=0,\ L_a=0$ と仮定する．これらの仮定のもとでまず (35) 式の左辺第 3 項に (39) 式より得られる $\dot{\theta}_a=\dfrac{K_g}{r}\dot{x}$ を代入し，i_a について解くと次式が得られる．

$$i_a=\frac{1}{R_a}\left(v_a-\frac{K_EK_g}{r}\dot{x}\right) \tag{40}$$

この式を (36) 式へ代入し，さらに (37) 式を用いると f は次のように求まる．

$$f=\frac{K_TK_g}{R_ar}\left(v_a-\frac{K_EK_g}{r}\dot{x}\right)=c_1v_a-c_2\dot{x} \tag{41}$$

ただし，$c_1=\dfrac{K_TK_g}{R_ar}$，$c_2=\dfrac{K_TK_EK_g^2}{R_ar^2}$ と置いている．一方 $J_p=0$，$c_c=0$，$c_p=0$ を (33)(34) 式に代入し $\ddot{x}$ および $\ddot{\theta}$ について整理すると次式が得られる．

$$\begin{bmatrix}m_c+m_p & m_p\ell\cos\theta\\ m_p\ell\cos\theta & m_p\ell^2\end{bmatrix}\begin{bmatrix}\ddot{x}\\ \ddot{\theta}\end{bmatrix}=\begin{bmatrix}cm_p\ell\dot{\theta}^2\sin\theta+f\\ m_pg\ell\sin\theta\end{bmatrix} \tag{42}$$

これを $\ddot{x}$ と $\ddot{\theta}$ について解くと次のようになる．

$$\ddot{x}=\frac{1}{\Delta}\{m_p\ell^2(m_p\ell\dot{\theta}^2\sin\theta+f)-m_p^2\ell^2g\cos\theta\sin\theta\} \tag{43}$$

$$\ddot{\theta}=\frac{1}{\Delta}\{-m_p\ell\cos\theta(m_p\ell\dot{\theta}^2\sin\theta+f) \\ +(m_c+m_p)m_pg\ell\sin\theta\} \tag{44}$$

ただし，$\Delta=m_p\ell^2(m_c+m_p)-m_p^2\ell^2\cos^2\theta$ である．ここで (41) 式を (43)(44) 式に代入し，状態変数として $\boldsymbol{x}=\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix}=\begin{bmatrix}x\\\theta\\\dot{x}\\\dot{\theta}\end{bmatrix}$ と選び，入力変数として $u=v_a$ と選ぶと，次式が得られる．

$$\dot{x}_1=\ x_3 \tag{45}$$

$$\dot{x}_2=\ x_4 \tag{46}$$

$$\dot{x}_3=\ \frac{m_p\ell x_4^2\sin x_2-m_pg\cos x_2\sin x_2-c_2x_3}{m_c+m_p-m_p\cos^2x_2} \\ +\frac{c_1}{(m_c+m_p-m_p\cos^2x_2)}u \tag{47}$$

$$\dot{x}_4=\ \frac{-m_p\ell x_4^2\cos x_2\sin x_2+(m_c+m_p)g\sin x_2}{\ell(m_c+m_p)-m_p\ell\cos^2x_2} \\ +\frac{c_2x_3\cos x_2}{\ell(m_c+m_p)-m_p\ell\cos^2x_2} \\ -\frac{c_1\cos x_2}{\ell(m_c+m_p)-m_p\ell\cos^2x_2}u \tag{48}$$

ここで，u は 1 変数 v_a のみからなるので，スカラーとなっている．以上で得られた (45) 〜(48) 式が倒立振子システムの非線形の状態方程式であり，(6) 式に対応する．また，

出力変数として $\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x \\ \theta \end{bmatrix}$ と選ぶと出力方程式は次の様になる．

$$y_1 = x_1 \tag{49}$$

$$y_2 = x_2 \tag{50}$$

この出力方程式は (7) 式に対応する．

本実習においてはこのように得られた倒立振子の状態空間モデルを用いて制御システムの設計を行う．ただし，最適レギュレータ理論で得られる制御システムは状態フィードバックとして実現される．したがって，すべての状態が測定により得られフィードバックできるという事が前提となる．ところが本実習で用いる実験装置では，測定で得られるのは台車の位置 $x_1(= x)$ と振子の角度 $x_2(= \theta)$ のみである．したがって状態フィードバックをするためにはこれら測定で得られる変数より，他の測定できない変数を求める必要がある．測定できない変数は $x_3(= \dot{x})$, $x_4(= \dot{\theta})$ なので，それぞれ x_1, x_2 を時間微分する事で得ることができる．

9 実習項目

まず線形最適化と状態空間モデルに関する基礎実習を行い，次に倒立振子制御システムの設計を行う．

9.1 基礎実習

9.1.1 線形最適化

シンプレックス法の動作を理解し，線形最適化の基礎事項を確認するための実習を行う．

1. Scilab のエディタ SciNotes を用いて標準形 (1) に対するシンプレックス法のプログラムを作成する．プログラムの概要を下記に示す．

```
// 初期のタブローを設定
T = [-1 -2 0 0 0; 1 3 1 0 9; 2 1 0 1 8]
sizeT = size(T);
while min( T(1,1:sizeT(2)-1) ) < 0 do
  // まず j* を求める．
  // 次に i* を求める．
  // 最後にタブローを更新する．
  T // 更新したタブローを表示
end
```

ここで，「//」より右はコメントを示している．標準形 (3) を例題としてプログラムを実行し，プログラムに誤りがないことを確認する．

2. 問題

$$\begin{aligned} \max \quad & f' = 2x_1 + x_2 \\ \text{s.t.} \quad & x_1 + 5x_2 \leqq 50 \\ & 3x_1 + 4x_2 \leqq 60 \\ & 2x_1 + 3x_2 \leqq 34 \\ & x_1 + x_2 \leqq 14 \\ & 5x_1 + x_2 \leqq 50 \\ & x_1 \geqq 0,\ x_2 \geqq 0 \end{aligned} \tag{51}$$

を標準形に変換し，変換した標準形に対して 1. で作成したプログラムを実行して最適解を求める．

[課題] 問題 (51) の制約条件式を満たす決定変数の値の範囲，およびシンプレックス法で更新した最初から最後までの実行可能基底解（の x_1 と x_2 の値）を x_1-x_2 平面に図示せよ．

3. シンプレックス法の流れの 3. において，非基底変数から基底変数に変更する決定変数を，流れの 2. で最適解とならない要因となった負の数の中の最小値ではなく，2 番目に小さい負の数（ただし，負の数が 1 つしかない場合は，その 1 つしかない負の数を選ぶ）とするプログラムに変更して，そのプログラムを問題 (51) に対して実行する．

[考察] 今回の方法と 2. で用いた方法の実行結果を比較する．

9.1.2 状態空間モデル

一入力，二出力の非線形システム

$$\dot{x}_1 = x_2 + u_1 \tag{52}$$

$$\dot{x}_2 = 6 \sin x_1 - 5x_2 \tag{53}$$

$$y_1 = x_1 \tag{54}$$

$$y_2 = x_2 \tag{55}$$

を例として，状態空間モデルに関する基礎事項を確認するための実習を行う．

1. 入力を $u_1 = 0$ と一定にしたときの非線形システムの平衡状態 $\boldsymbol{x}^* = \begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix}$ を二つ求める．ただし，x_1^*

は $0 \leqq x_1^* < 2\pi$ の範囲とする．次に，Scilab のシステムシミュレータ Xcos を使用して $u_1 = 0$ とした非線形システムのシミュレータを作成し，x_1, x_2 の初期値を平衡状態としたシミュレーションを行う．シミュレーションは平衡状態ごとに行うこと．

[考察] 計算結果とシミュレーション結果から平衡状態に関する考察を行う．

2. 非線形システムを平衡状態のまわりで近似した線形システムを平衡状態ごとに求める．次に，(53) 式の右辺を f_2，すなわち $f_2 = 6\sin x_1 - 5x_2$ とし，f_2 を線形に近似したものを g_2, h_2 とするとき，数値計算ソフトウエア Scilab を用いて，f_2, g_2, h_2 のグラフを描く．報告書にはグラフの各軸の変数を明記するとともに，どのグラフが f_2, g_2, h_2 であるのかがわかるようにすること．

[考察] f_2 のグラフと g_2, h_2 のグラフを比較して，(1) グラフの形，(2) 平衡状態のまわりで g_2, h_2 が f_2 をどのように近似しているのか，に関する考察を行う．

3. $u_1(t) = 5\sin t + 2\sin 10t$ としたときの非線形システム (52)–(55) のシミュレータを Xcos で作成し，初期値を $x_1(0) = 1, x_2(0) = 0$ としたシミュレーションを行う．次に，非線形システムに $u_1(t) = 5\sin t$ を入力したときの出力（$y_1^a(t), y_2^a(t)$ とする）と，$u_1(t) = 2\sin 10t$ を入力したときの出力（$y_1^b(t), y_2^b(t)$ とする）をそれぞれ加えた信号 $y_1^a(t) + y_1^b(t), y_2^a(t) + y_2^b(t)$ をグラフとして出力するシミュレータを Xcos で作成し，初期値を $x_1(0) = 1, x_2(0) = 0$ としたシミュレーションを行う．さらに，2. で求めた二つの線形システムのうち，x_1^* の大きい方の平衡状態のまわりで線形化したシステムのシミュレータを作成し，非線形システムに対して行った 2 種類のシミュレーションを線形システムで行う．初期値は $\Delta x_1(0) = 1, \Delta x_2(0) = 0$ とする．

[考察] 非線形システムと線形システムのシミュレーション結果の相違点に関する考察を，線形システムの定義に基づいて行う．

4. 各線形システムの行列 $\boldsymbol{A}$ の固有値を求めることにより内部安定性を調べる．次に，線形システムのそれぞれに対して $u_1 = 0$ としたシミュレータを Xcos で作成し，初期値を $\Delta x_1(0) = 1, \Delta x_2(0) = 0$ としたシミュレーションを行う．

[考察] 調べた内部安定性とシミュレーション結果から線形システムの安定性に関する考察を行う．

[課題] 内部安定性を調べて安定となっている線形システムにおいて，シミュレーション結果の各グラフでの収束値を読み取り，それらの値に対応する x_1, x_2 の値を求めよ．

5. 1. で作成した非線形システムのシミュレータで初期値を $x_1(0) = 1, x_2(0) = 0$ としたシミュレーションを行う．

[考察] 1. で求めた二つの平衡状態のいずれか一方に収束し，もう一方には収束しない結果となる．この結果となる理由を，4. の二つのシミュレーション結果や 4. の課題の解答と関連づけて考察する．

9.2 倒立振子制御システムの設計

振子を倒立させる制御システムを最適レギュレータ理論により設計する．倒立時に振子をなるべく静止させるために，設計仕様は倒立時の台車の振れ幅（位置 x_1 の最大値から最小値を引いた値）が 5cm 以内となることとする．

1. 制御対象の状態空間モデル (45)–(50) を，平衡状態の一つである目標状態 $\boldsymbol{x}^* = \boldsymbol{0}, \boldsymbol{y}^* = \boldsymbol{0}, u^* = 0$ のまわりで線形化したシステム

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{b}u(t) \tag{56}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) \tag{57}$$

を求める．ここで，u はスカラーであるので，(16) 式の行列 $\boldsymbol{B}$ をベクトル $\boldsymbol{b}$ で表記している．次に，制御対象のパラメータに基づき，$\boldsymbol{A}, \boldsymbol{b}$ の各成分を算出する．パラメータの値は以下の通りである．

台車の質量	m_c=0.522	kg
トルク定数	K_T=0.00767	Nm/A
逆起電力定数	K_E=0.00767	Vsec/rad
電機子抵抗	R_a=2.6	Ω
駆動車輪半径	r=0.00635	m
ギア比	K_g=3.7	
振子の質量	m_p=0.212	kg
振子の長さの $\frac{1}{2}$	ℓ=0.305	m
重力加速度	g=9.8	m/s^2

2. Scilab を用いてシステム (56)(57) の可制御性，可観測性，安定性を調べる．

3. Scilab を用いて最適レギュレータ理論により，ゲイン $\boldsymbol{K}$ を一つ求め，制御器を設計する．その際，評価

関数の重み行列 $\boldsymbol{Q}, \boldsymbol{R}$ は適当に設定する．$\boldsymbol{Q}, \boldsymbol{R}$ は正定値である実対称行列でなければならないことに注意する．次に，他の $\boldsymbol{Q}, \boldsymbol{R}$ を適当に設定して，いくつかの $\boldsymbol{K}$ を求める．このとき，値が近い $\boldsymbol{Q}, \boldsymbol{R}$ を設定しないようにすること．次に，設計した制御器を用いた全ての閉ループシステムの極を求めて安定性を調べる．さらに，これらの極の位置を複素平面上に図示する．
[考察] 設定した $\boldsymbol{Q}, \boldsymbol{R}$ と，安定性を調べるときに求めた極の複素平面上の位置との関係を考察する．

4. Xcos を使用して倒立振子のフィードバック制御システムのシミュレータを作成し，3. で最初に求めた $\boldsymbol{K}$ を用いたシミュレーションを行う．フィードバック制御システムにおいて，制御対象は倒立振子の非線形システムであり，制御器は 3. で設計したものである．シミュレーションのグラフには x_1, x_2, x_3, x_4, u の 5 変数を表示するようにせよ．u をグラフに表示させるので，u を消去したシミュレータを作成しないように注意せよ．このために，最初に制御対象を描き，次に制御器を描くと良いだろう．シミュレータを作成したら，状態変数の初期値を平衡状態や他の適当な値とし，ゲインを $\boldsymbol{0}$ や適当な値としてシミュレーションを行い，正しくシミュレータが作成できていることを確認する．正しくシミュレータが作成できていれば，初期値を x_1=0.01[m]，x_2=0.08[rad]，x_3=0[m/s]，x_4=0[rad/s] とし，3. で最初に求めた $\boldsymbol{K}$ を用いてシミュレーションを実行する．

   [課題] Xcos で作成したシミュレータを示し，制御対象および制御器をそれぞれ点線で囲め．

5. 3. において求めた種々の $\boldsymbol{K}$ を用いて，4. のシミュレーションを行う．
   [考察] シミュレーション結果より，3. で求めた極の複素平面上の位置と制御性能の関係を考察する．

6. 望ましい応答のシミュレーション結果が得られたら，実機実験を行う．実験に用いた $\boldsymbol{K}$ とその算出に用いた重み行列 $\boldsymbol{Q}, \boldsymbol{R}$，および結果を記録する．結果として，倒立したかどうか，および倒立した場合の台車の位置の振れ幅を記録する．また，実機実験を行っていないときは，種々の重み行列に対するシミュレーションを行い，さらによい $\boldsymbol{K}$ を見つける．

10 数値計算ソフトウエア Scilab

Scilab はシステム・制御に関する種々の計算・シミュレーションを実行できるフリーウエアである．ここでは本実習に最低限必要な操作方法を概説する．

10.1 インストール・起動

Scilab は `https://www.scilab.org/` からダウンロードして個人所有の PC にインストールすることができる．実習室の PC にはすでにインストールされているが，最新バージョン以前のものがインストールされている場合がある．以下では，実習室の PC にインストールされているバージョンにおける操作方法を説明する．

10.2 基本的な計算方法

Scilab では，実数はもちろんのこと，ベクトルや行列などの種々の計算が容易に可能である．まず，タブロー (4) の値を並べた行列 $\boldsymbol{T} = \begin{bmatrix} -1 & -2 & 0 & 0 & 0 \\ 1 & 3 & 1 & 0 & 9 \\ 2 & 1 & 0 & 1 & 8 \end{bmatrix}$ を設定してみよう．

Scilab Desktop を起動すると，コンソールにプロンプト`-->`が現れるので，ここに種々のコマンドを入力していく．なお，↑キーでこれまでに入力してきたコマンドが表示されるので，この機能を利用して入力の手間を軽減することができる．また，入力したコマンドはコマンド履歴に表示され，コマンド履歴のコマンドをダブルクリックすることにより，これまでに入力してきたコマンドを再実行することができる．

```
-->T=[-1 -2 0 0 0; 1 3 1 0 9; 2 1 0 1 8]
```

`T` は変数で，`=`によって右辺の内容を左辺の変数に代入できる．行列は `[ ]` で表し，セミコロンが行の区切りを示す．また，行あるいは列の数を一つとすることで，ベクトルが設定できる．設定した変数は変数ブラウザに表示され，変数ブラウザの変数をダブルクリックすることにより，変数の内容を確認できる．あるいは，次のように変数名のみを入力しても内容を確認できる．

```
-->T
```

次に，行列やベクトルから一部の成分を取り出す演算を示す．

```
-->T(3,2)
-->v=T(1:3,1)
```

( ) を用いることで行列やベクトルの一部を取り出せる．行列の場合はカンマ (,) の両側に，取り出す行と列の範囲を示す．範囲に 3 や 2 といった数値を入れた場合はその数値に対応する行と列となる．したがって，T(3,2) は行列 $\boldsymbol{T}$ の 3 行 2 列の成分となる．また，範囲に 1:3 とすると 1 から 3 までを示すこととなる．したがって，T(1:3,1) は行列 $\boldsymbol{T}$ の第 1 列の第 1 行から第 3 行までを取り出したベクトルとなる．ここで，第 1 行から第 3 行までとは全ての行のことであり，このように全てを指定する場合は単に「:」だけでもよい．すなわち，T(1:3,1) は T(:,1) としてもよい．

逆に，次のように行列やベクトルの一部の成分のみに 1 つのコマンドで代入することができる．

```
-->T(:,5)=[2; 2; 2]
```

この例では行列 $\boldsymbol{T}$ の第 5 列のみに代入している．

ベクトルの成分中の最小値やその最小値をとる成分番号は

```
-->m1=min(v)
-->[m1,m2]=min(v)
```

と，min 関数を用いて求められる．1 行目の方法では最小値のみを求め，2 行目の方法では最小値を変数 m1 に，最小値をとる成分番号を m2 にそれぞれ代入する．最大値を求める場合は max 関数を用いる．

次に，四則演算の例をいくつか示す．

```
-->4/2+3*(3^2-8)
-->v*2
-->v+[2; 2; 4]
-->T(:,1)=T(:,2)*5+T(:,3)
```

1 行目はスカラーの四則演算であり，*は乗算，/は除算，^は累乗である．2 行目はベクトルとスカラーの乗算である．ベクトルとスカラーの演算はベクトルの各成分とスカラーとの演算となり，これは行列とスカラーの演算でも同様である．3 行目はベクトル同士の演算であり，行列同士の場合も同様に計算できる．4 行目の右辺では，行列 $\boldsymbol{T}$ の第 2 列（のベクトル）を 5 倍してから第 3 列（のベクトル）を加え，その演算結果を第 1 列に代入している．

次に，行列に関する演算を示す．

```
-->size(T)
-->rank(T)
-->spec(T(:,1:3))
```

size 関数は引数の行の数と列の数を求め，これらの 2 つの要素としたベクトルで返す．rank，spec 関数は，それぞれ引数のランク（階数）と固有値を求める．

次に，(31) 式で示されるリカッチ方程式を解いて，行列 $\boldsymbol{P}$ を求める方法を説明する．いま，各行列が次式で与えられるリカッチ方程式を解いてみよう．

$$\boldsymbol{A}=\begin{bmatrix}0 & 0\\ 0 & 1\end{bmatrix},\ \boldsymbol{B}=\begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix},\ \boldsymbol{Q}=\boldsymbol{R}=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} \tag{58}$$

まず，行列を設定する．

```
-->A=[0 0; 0 1]
-->B=[1 1; 0 1]
-->Q=diag([1 1])
-->R=Q
```

diag 関数は対角成分を [ ] 内で与えた数値とした対角行列を設定する．ここでは [1 1] としているので，単位行列となる．

Scilab は次式で表される形式のみのリカッチ方程式しか解くことができない．

$$\boldsymbol{A}^T\boldsymbol{P}+\boldsymbol{P}\boldsymbol{A}-\boldsymbol{P}\boldsymbol{B_2}\boldsymbol{P}+\boldsymbol{Q}=\boldsymbol{O} \tag{59}$$

そこで，まず $\boldsymbol{B_2}$ を求める．

```
-->B2=B*inv(R)*B'
```

inv 関数は引数の逆行列，’ は転置行列を求める．

```
-->riccati(A,B2,Q,'c')
```

riccati 関数は第一引数から第三引数で与えられた行列のリカッチ方程式を解く．なお，第四引数は連続時間領域で解くことを示す．その結果，$\boldsymbol{P}=\begin{bmatrix}1 & -1\\ -1 & 3.7320508\end{bmatrix}$ となる．

Scilab にはいくつかの定数が定義されており，%inf は無限大を表す．

[練習問題 1]

まず，行列 $\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{bmatrix}$ を変数 A1 に代入せよ．次に，第 1 行の各成分を 2 倍したものを第 2 行に代入する 1 つのコマンドを考え，そのコマンドを実行せよ．

[練習問題 2]

ある n 次元ベクトル $\boldsymbol{w}$ の要素数をコマンド

```
-->sizeofw=size(w)
```

で求めたとする．この後，第 1 成分から第 $(n-1)$ 成分までの間で最大値をとる成分の番号を求めるコマンドを考え，そのコマンドを $\boldsymbol{w} = (\ 3\ 8\ 6\ 2\ 10\ )$ に実行した結果を変数 A2 に代入せよ．

[練習問題 3]

(31)(58) 式のリカッチ方程式において，$\boldsymbol{B} = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}$ と置き換えた場合のリカッチ方程式を解き，その解を変数 A3 に代入せよ．

10.3 グラフの描画

最初に，$y = \sin t$ のグラフを描いてみる．まず，つぎのコマンドで横軸の値を設定する．

```
-->t=[-5:0.1:5];
```

ここでは変数 t に 1 行 101 列のベクトルを代入している．第 n 列の要素は $-5 + 0.1(n-1)$ である．すなわち，-5 から 0.1 ずつ増加して 5 まで存在する．最後の「;」はコマンドに関連する出力を Scilab ウィンドウに表示させないようにするものである．ここでは，101 個もの大量の数値があるので，それを表示させないようにしている．

次に，縦軸の値を設定する．

```
-->y=sin(t);
```

sin 関数は引数の正弦を求める．この結果，y も 1 行 101 列のベクトルとなっている．

最後に，つぎのコマンドでグラフを描く．

```
-->plot2d(t,y)
```

plot2d 関数は第一引数を横軸，第二引数を縦軸とした 2 次元のグラフを描く．

次に，2 変数関数 $y = \sin x_1 + \sin x_2$ のグラフを描いてみる．まず，つぎのコマンドで x_1, x_2 軸の値を設定する．

```
-->x1=[-5:0.1:5];
-->x2=[-5:0.1:5];
-->x1m=x1'*ones(1,length(x2));
-->x2m=ones(length(x1),1)*x2;
```

length 関数は引数の行列やベクトルの要素数を求める．また，ones 関数は要素がすべて 1 の行列を求める．第一引数に行数，第二引数に列数を指定する．次に，縦軸の値を設定する．

```
-->y=sin(x1m)+sin(x2m);
```

最後にグラフを描くが，以前にグラフを描いている場合は，clf コマンドで以前のグラフを消去してからグラフを描く．

```
-->clf
-->plot3d(x1,x2,y)
```

plot3d 関数は各引数を座標軸とした 3 次元のグラフを描く．

[練習問題 4]

2 変数関数 $y = e^{x_1} + e^{x_2}$ のグラフを描け．ここで，指数関数の値は exp 関数を用いて求められる．

10.4 保存，読み込み，終了方法

Scilab では設定した変数およびその内容を保存できる．保存は「ファイル→環境を保存」で行い，読み込みは「ファイル→環境をロード」で行う．ファイルの拡張子は sav であり，ファイル名に日本語は使用できないようである．終了は「ファイル→終了」または exit コマンドを実行する．

描画したグラフの画像ファイルへの変換は，グラフィック・ウィンドウ上の「ファイル→エクスポート」で行う．

10.5 エディタ

これまではコマンドを 1 つずつ実行したが，複数のコマンドをまとめて実行したり，繰り返しや条件分岐を行いたい場合はエディタ SciNotes を用いる．「アプリケーション」→「SciNotes」か，または下記のコマンドを入力すると SciNotes が起動する．

```
-->scinotes;
```

SciNotes のウィンドウ内で C 言語のプログラミングのように，コマンドを並べてプログラムを作成し，「実行する」→「... ファイルを実行（出力あり）」を選択すると，プログラムが実行される．

SciNotes では C 言語などと同様に，繰り返しを行う for や while，条件分岐を行う if などを用いることができる．まず，for は次のように用いる．

```
a=0
for i=1:5 do
    a=a+1
end
```

for と do の間に「変数=初期値:最終値」などと書き，初期値から値を 1 つずつ増やして最終値となるまで end までのコマンドが繰り返される．ここでは変数 i の値を 1 から 5 まで 1 つずつ増やして計 5 回変数 a の値を 1 増やしている．なお，行列やベクトルの行または列のすべての成分に同じような演算を行いたい場合は，for 文ではなく練習問題 1 のような 1 つのコマンドを用いる方が簡便である．

while は次のように用いる．

```
a=0
while a<3 do
    a=a+1
end
```

while の右に書かれた条件が満たされている間は end までのコマンドを繰り返す．<は小なりを意味し，その他に<=（以下），==（等しい），>（大なり），>=（以上），<>（等しくない）を用いることができる．また，これらを &（かつ）や |（または）と組み合わせて使用できる．while を使うときはループを必ず終了させるように注意すること．ここでは，a の値が 3 以上になるまで，a の値を 1 ずつ増やしている．

if は次のように用いる．

```
if a<3 then
    a=5
elseif a<5 then
    a=10
else
    a=15
end
```

if の右に書かれた条件が満たされていれば then と elseif の間を実行する．その条件が満たされておらず，elseif の右に書かれた条件が満たされていれば，elseif と else の間を実行する．その条件も満たされていなければ，else と end の間を実行する．elseif と else の部分は省略してもよい．ここでは，$a < 3$ ならば a の値を 5 に，$3 \leq a < 5$ ならば a の値を 10 に，$a \geq 5$ ならば a の値を 15 にしている．

SciNotes のプログラムの保存は「ファイル」→「保存」で，読み込みは「ファイル」→「開く」で行う．終了は「ファイル」→「Scinotes を終了」で行う．

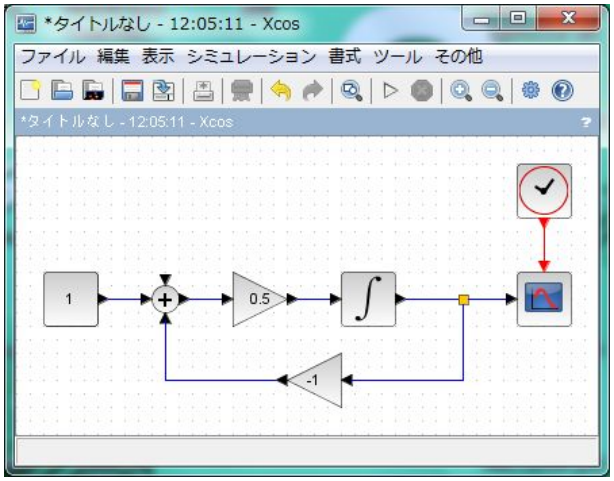


図 5: $\dot{y}(t) = 0.5\{u(t) - y(t)\}$ のシミュレータ

10.6 シミュレータ

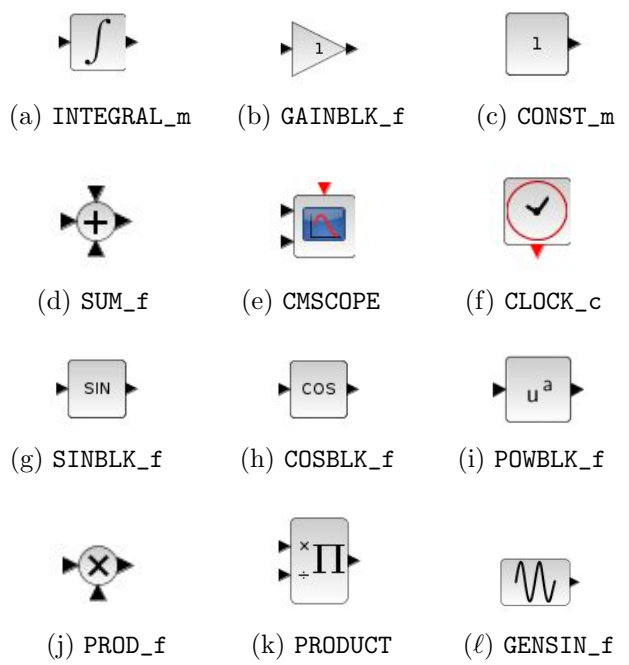
Scilab には，種々なパレットやリンクなどを配置することによって種々のシミュレーションを実行するツール Xcos が用意されている．「アプリケーション→ Xcos」か，または下記のコマンドを入力すると Xcos が起動する．

```
-->xcos;
```

Xcos を起動すると Xcos ウィンドウとパレットブラウザウィンドウが表示される．Xcos で配置したパレットやリンクなどの最初の保存は「ファイル→別名で保存」で行い，上書きの保存は「ファイル→保存」で行う．また，読み込みは「ファイル→開く」，画像ファイルへの変換は「ファイル→エクスポート」で行う．Xcos の終了は「ファイル→ Xcos を終了」で行う．

一入力一出力のシステム $\dot{y}_1(t) = 0.5\{u_1(t) - y_1(t)\}$ において，入力を単位ステップ関数 $u(t) = \begin{cases} 1 & (t > 0) \\ 0 & (t < 0) \end{cases}$ とした場合のシミュレータを作成してみよう．先に完成図を図 5 に示しておく．図 5 において，矢印は信号の流れを表し，楕円，三角形，四角形の図形は信号の何らかの処理を表している．このような図をブロック線図という．

システム $\dot{y}_1(t) = 0.5\{u_1(t) - y_1(t)\}$ は，$\dot{y}_1(t)$ を積分すると $y_1(t)$ となり，また $\dot{y}_1(t)$ は $u_1(t) - y_1(t)$ の 0.5 倍に等しいことを意味している．そこで，「$\dot{y}_1(t)$ を積分すると $y_1(t)$ となる」部分を作成するために，図 6(a) で示される INTEGRAL_m パレット（連続時間システムグループ内にある）をパレットブラウザウィンドウから Xcos ウィンドウにドラッグする．このように Xcos ではシミュレータに



(a) INTEGRAL_m (b) GAINBLK_f (c) CONST_m

(d) SUM_f (e) CMSCOPE (f) CLOCK_c

(g) SINBLK_f (h) COSBLK_f (i) POWBLK_f

(j) PROD_f (k) PRODUCT (ℓ) GENSIN_f

図 6: パレット

必要なパレットをパレットブラウザウィンドウから Xcos ウィンドウにドラッグしていく．INTEGRAL_m パレットは入力信号を積分するパレットであり，今回のシステムでは入力が $\dot{y}_1(t)$ に，出力が $y_1(t)$ に対応する．積分の初期値 Initial Condition をパラメータとして変更できるが，今はデフォルトの値 (0) でよい．

$u_1(t) - y_1(t)$ は $u_1(t)$ と $-y_1(t)$ の和と考え，まず $y_1(t)$ を -1 倍して $-y_1(t)$ とするために，図 6(b) で示される GAINBLK_f パレット（数値計算グループ）を用いる．これは入力を k 倍したものを出力とするパレットであり，k の値をパラメータ Gain として変更できる．パラメータの値を変更するためにはパレットをダブルクリックするか，「右クリック→ブロックパラメータ」を選択する．ここでは，値をデフォルトの 1 から-1 に変更しておく．この結果，このパレットの出力信号は $-y(t)$ を示すことになる．

次に $u_1(t)$ は図 6(c) で示される CONST_m パレット（信号源グループ）を用いる．これは定数値の信号を出力するパレットであり，このパレットの出力信号が $u_1(t)$ となる．$u_1(t)$ と $-y_1(t)$ の和は図 6(d) で示される SUM_f パレット（数値計算グループ）を用いる（SOM_f と間違わないように注意）．これは三つまでの入力の和を出力とするパレットであり，このパレットの出力が $u_1(t) - y_1(t)$ を示すことになる．

次に，$u_1(t) - y_1(t)$ を 0.5 倍するために GAINBLK_f パレットを用いる．もう一度パレットブラウザウィンドウからドラッグしてもよいが，先に配置したパレット上で「右クリック→コピー」し，パレットのないところで「右クリック→貼り付け」を選択するとパレットのコピー・ペーストができる．ペーストしたパレットのパラメータを 0.5 に変更する．このパレットの出力が $0.5\{u_1(t) - y_1(t)\}$ であり，$\dot{y}_1(t)$ に等しい．

配置したパレットは，コピーやペーストの他に種々の操作が可能である．例えば，ドラッグ・ドロップで自由に移動させることができる．パレット上で「右クリック→削除」により，そのパレットを消去できる．パレット上で「右クリック→書式→反転」で左右方向または上下方向に反転できるので，-1 倍の GAINBLK_f パレットを左右に反転させておく．また，「編集→元に戻す」で直前の操作を取り消せる．

次に，パレット間をリンクする．パレットの出力（または入力）部分の矢印上からドラッグし，リンクをはりたいパレットの入力（または出力）部分の矢印上でドロップすることによってリンクできる．パレットの入力（または出力）部分の矢印以外の場所でドロップすることにより，そこでリンクの線を折ることができる．一度配置したリンクの線も線上でダブルクリックすることにより，線を折ることができる．また，線上で「右クリック→削除」することにより，リンクを削除することができる．CONST_m パレットから SUM_f パレット，-1 倍の GAINBLK_f パレットから SUM_f パレット，SUM_f パレットから 0.5 倍の GAINBLK_f パレット，0.5 倍の GAINBLK_f パレットから INTEGRAL_m パレットにそれぞれリンクをはっておく．なお，INTEGRAL_m パレットから -1 倍の GAINBLK_f パレットへのリンクは後ではる．

以上でシミュレータは一応完成したが，グラフを描画する部分も作成する必要がある．そのために図 6(e)(f) で示される CMSCOPE パレット（出力/表示グループ）と CLOCK_c パレット（信号源グループ）を用いる．CMSCOPE パレットは複数のグラフをある離散時間間隔で描画するパレットであり，CLOCK_c パレットはその離散時間を出力するパレットである．CMSCOPE パレットでは種々のパラメータの値を変更することでグラフの表示方法を変更できる．まず Input ports sizes は入力数およびグラフ数を設定できる．初期設定では二つの要素を入力して二つのグラフを描かせるようになっている．これは「1」の個数で与えられる．すなわち，初期設定では 1 1 と「1」が二つ並んでいるので，入力数が二つとなっている．今は入力数を一つにするために 1 と変更せよ．もし入力数を五つに

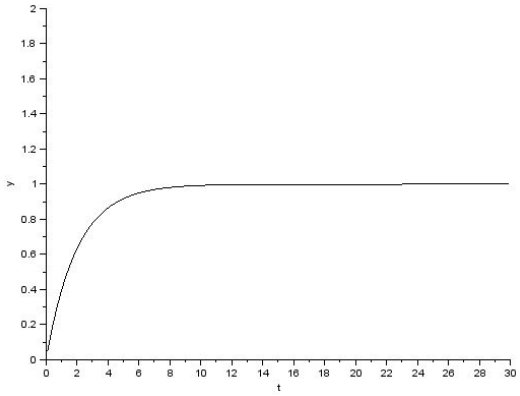


図 7: シミュレーション結果

したいなら，1 1 1 1 1 とすればよい．Ymin vector と Ymax vector は各グラフの縦軸の最小値と最大値を設定する．初期設定では最初のグラフの最小値が −1，つぎのグラフの最小値が −5 となっている．Ymin vector を 0，Ymax vector を 2 として，縦軸の最小値および最大値を 0 および 2 とせよ．Refresh period もグラフごとに横軸の長さを変更でき，今は 30 とせよ．また，今は変更しなくてよいが，CLOCK_c パレットではパラメータを変更して（時間）間隔と初期化時間を設定できる．

最後に，残りのリンクをはってシミュレータを完成させる．まず INTEGRAL_m パレットと CLOCK_c パレットから CMSCOPE パレットにリンクをはる．次に，INTEGRAL_m パレットから CMSCOPE パレットにはったリンク上にマウスをもっていくと線が緑になるので，そのときにドラッグして −1 倍の GAINBLK_f パレットにリンクをはる．リンクの線から分岐するリンクをはるにはこのように行う．

それでは，いよいよシミュレーションを行う．「シミュレーション→設定」でシミュレーションの設定ができる．「積分終了時間」はシミュレーション時間を示し，今は 30 に変更する．このように，シミュレーション時間を CMSCOPE パレットの Refresh period と同一にするべきである．次に，「シミュレーション→開始」でシミュレーションを実行でき，図 7 のグラフが描画される．

実習では図 6(g)–(ℓ) で示されるパレットも使用する．(ℓ) は信号源グループ，それ以外は数値計算グループにある．(g) および (h) はそれぞれ入力の正弦および余弦を出力とする．(i) は入力の p 乗を出力とし，to the power of でパラメータ p の値を変更できる．(j) および (k) はそれぞれ二つの入力の積および商を出力とする．(ℓ) は正弦波を入力するために用い，ゲインは正弦波の振幅，周波数は正弦波の角周波数（周波数ではないことに注意）を与える．

[練習問題 5]

微分方程式 $\dot{y}_1(t) = -0.3y_1(t) + 0.6u_1(t)$ で表されるシステムにおいて，入力が単位ステップ関数であるときのシミュレータを作成し，そのシミュレーション結果を示せ．ただし，グラフの横軸を 0 から 20 までにし，初期値を $y_1(0) = 4$ とすること．

11 倒立振子実験装置操作方法

倒立振子実験装置の PC において，ブロックダイアグラムウィンドウのブロック線図に示されている k1,k2,k3,k4 がフィードバックゲイン $\boldsymbol{k}$ の第 1 成分から第 4 成分であり，設計で求めた値をこれらに入力する．台車をレールの中央に置き，振子が倒れている状態で静止させた後，フロントパネルウィンドウのメニューの「操作」→「実行」で実験装置の動作を開始させる．このとき，応答データを記録するファイルを尋ねられるので，デスクトップの output.txt を選択する．振子を頂上付近まで回転させると自動的に制御が始まるので，振子の動きを見ながら手を離す．

振子が倒立しなかった場合，直ちにフロントパネルウィンドウの「操作」→「停止」により実験装置の動作を停止させる．

振子が倒立した場合，台車の振れ幅を調べるために応答データを保存する．このために，倒立してから数秒後に「操作」→「停止」により実験装置の動作を停止させる．応答データは先に選択した output.txt に自動的に保存される．このファイルには，0.002 秒間隔でデータが縦に並んでおり，各行には「入力電圧 台車の位置 振子の角度 台車の速度 振子の角速度」の順に値が記録されている．

参考文献

1. 福島：新版 数理計画入門，朝倉書店，2011
2. 近藤 編：基礎制御工学，森北出版，1977
3. 吉川，井村：現代制御論，昭晃堂，1994
4. 小郷，美多：システム制御理論入門，実数出版，1979
5. 平井：非線形制御，コロナ社，2003