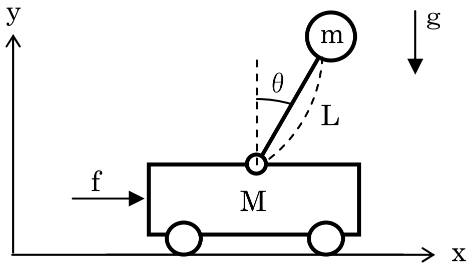
ソフトコンピューティング演習

第２回　ファジー制御の構築

講義用 HP：http://www.robot.t.u-tokyo.ac.jp/dcm/lec/softcomp.htm

2.1 倒立振り子の運動方程式

下図は倒立振り子と台車で構成されるシステムである．但し，振り子の棒の質量・慣性モーメントは考えないものとする．



m	：	おもりの質量
L	：	振り子の長さ
θ	：	振り子の角度
M	：	台車の質量
f	：	台車に加わる力
g	：	重力加速度

図１　倒立振り子システム

ラグランジュ法によりこのシステムの運動方程式を求める．このシステムのポテンシャルエネルギ U と運動エネルギ K は,

$$U = mgL\cos\theta$$

$$K = \frac{1}{2}M\,\dot{x}^2 + \frac{1}{2}m\left(\dot{x}^2 + 2L\dot{x}\cos\theta\cdot\dot{\theta} + L^2\dot{\theta}^2\right)$$

となる．ラグランジュ方程式

$$\frac{d}{dt}\frac{\partial K}{\partial \dot{x}} - \frac{\partial K}{\partial x} + \frac{\partial U}{\partial x} = f \qquad \cdots \text{台車}$$

$$\frac{d}{dt}\frac{\partial K}{\partial \dot{\theta}} - \frac{\partial K}{\partial \theta} + \frac{\partial U}{\partial \theta} = 0 \qquad \cdots \text{振り子}$$

に代入すると，以下の運動方程式を得ることができる．

$$(M+m)\ddot{x} + mL\cos\theta\cdot\ddot{\theta} - mL\sin\theta\cdot\dot{\theta}^2 = f$$

$$mL\cos\theta\cdot\ddot{x} + mL^2\ddot{\theta} - mgL\sin\theta = 0$$

以上の 2 つの運動方程式を用いて角加速度およびお加速度を求めることができる．

$$\ddot{\theta} = -\frac{\cos\theta}{L\left(M + m\sin^2\theta\right)}f + \frac{(m+M)\sin\theta}{L\left(M + m\sin^2\theta\right)}g - \frac{mL\sin\theta\cos\theta}{L\left(M + m\sin^2\theta\right)}\dot{\theta}^2$$

$$\ddot{x} = \frac{1}{M + m\sin^2\theta}f - \frac{m\sin\theta\cos\theta}{M + m\sin^2\theta}g + \frac{mL\sin\theta}{M + m\sin^2\theta}\dot{\theta}^2$$

以上の二式が倒立振り子の挙動を示す方程式となる．

2007 年 12 月 5 日

2.2 微分方程式の数値解法（オイラー法）：漸化式およびフローチャート

倒立振り子シミュレーション（数値解析）を構成するために必要である漸化式およびフローチャートを以下に示す.

漸化式

経過時間： $t_n = t_{n-1} + dt$

運動方程式１： $\ddot{x}_n = \frac{1}{M + m\sin^2\theta_{n-1}} f - \frac{m\sin\theta_{n-1}\cos\theta_{n-1}}{M + m\sin^2\theta_{n-1}} g + \frac{mL\sin\theta_{n-1}}{M + m\sin^2\theta_{n-1}} \dot{\theta}_{n-1}^{\ 2}$

差分方程式１： $\dot{x}_n = \dot{x}_{n-1} + \ddot{x}_n * dt$

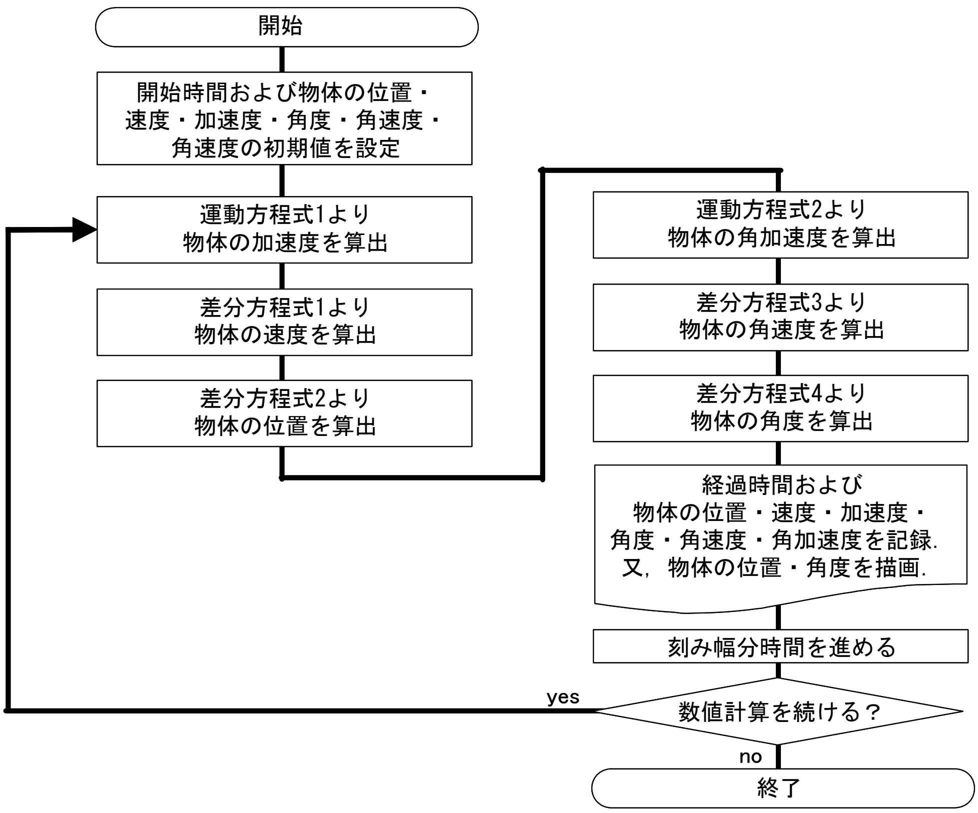
差分方程式２： $x_n = x_{n-1} + \dot{x}_n * dt$

運動方程式２： $\ddot{\theta}_n = -\frac{\cos\theta_{n-1}}{L\left(M + m\sin^2\theta_{n-1}\right)} f + \frac{(m+M)\sin\theta_{n-1}}{L\left(M + m\sin^2\theta_{n-1}\right)} g - \frac{mL\sin\theta_{n-1}\cos\theta_{n-1}}{L\left(M + m\sin^2\theta_{n-1}\right)} \dot{\theta}_{n-1}^{\ 2}$

差分方程式３： $\dot{\theta}_n = \dot{\theta}_{n-1} + \ddot{\theta}_n * dt$

差分方程式４： $\theta_n = \theta_{n-1} + \dot{\theta}_n * dt$

フローチャート



［課題１］上記されている漸化式およびフローチャートにもとづいて，数値解析プログラムを完成させなさい.

```
// 変数宣言
double th, dth, ddth;       // 振り子の角度，角速度，角加速度
double m1, m2, l;           // 振り子の質量，台車の質量，振り子棒の長さ
double f, g;                // 台車に加わる力，重力加速度
double x, dx, ddx;          // 台車の位置，速度，加速度
double t, dt;       // シミュレーション時間，時刻の刻み幅
main{
th=3.14/6.;     dth=0;      ddth=0;
m1=1;           m2=2;       l=1.5;
x=0;            dx=0;       ddx=0;
f=0;            g=9.8;
t=0;            dt=0.001;

// 数値解析
while(true){
// 台車の加速度(a)を求める方程式
(a)______________________________________

// オイラー法による台車の速度(b)および位置(c)の近似
(b)______________________________________
(c)______________________________________

// 振り子の角加速度(d)を求める方程式
(d)______________________________________

// オイラー法による角速度(e)および角度(f)の近似
(e)______________________________________
(f)______________________________________

～～～ 省略：制御値 f の決定．～～～
        t=t+dt;
        }
}
```

2007 年 12 月 5 日

2.3 ファジー制御

本演習では，振り子が台車に加わる力 f のみで制御されるものとし，振り子が垂直状態に収束するファジー制御を構築する．ファジーとは，複雑なシステムを「曖昧」にとらえることで最適に制御するアルゴリズムであり，1965 年に L.A.Zadeh によって提唱された境界がはっきりしない集合（ファジー集合）に帰属する度合をメンバーシップ関数として表すことで曖昧な主観を表現する方法である．そのため，多くの変数からなる複雑な系を扱うのに有効となる．

シミュレーションの初期設定は以下に記述したものを適用

dth=0;	// 角速度	m1=1;	// 重りの質量
ddth=0;	// 角加速度	m2=2;	// 台車の質量
x=0;	// 台車の位置	l=1.5;	// 振り子の長さ
dx=0;	// 台車の速度	g=9.8;	// 重力加速度
ddx=0;	// 台車の加速度	dt=0.001;	// 時間刻み幅

［課題２］まず初めに，振り子を安定させるため台車に加わる力が振り子の角度に比例する制御関数を作成する．

(a) 図 1 に示される制御関数 f=1000*th を用いると，振り子の初期角度が-PI/2 から+PI/2 までの範囲で始まる場合に限り安定となる．シミュレーションを作成し，どのような安定が得られるか確認して下さい．

(b) 次に，振り子の初期角度の範囲を-PI から+PI まで広げたとき，どのような制御関数に拡張すれば課題[1a]と同じような結果を得られるのか確認してください．また，その制御関数を図 2 に描画してください．

ヒント：[-PI<=th<-PI/2], [-PI/2<=th<0], [0<=th<+PI/2], [+PI/2<=th<+PI]の範囲ごとに振り子を立ち上げるための力の向きは違う．プログラムでは，条件 if で範囲分けする．

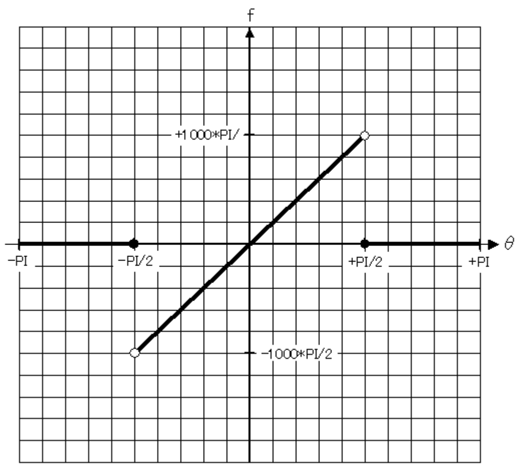


図 2　課題[2a]の制御関数

f=1000*th [-PI/2<th<+PI/2]

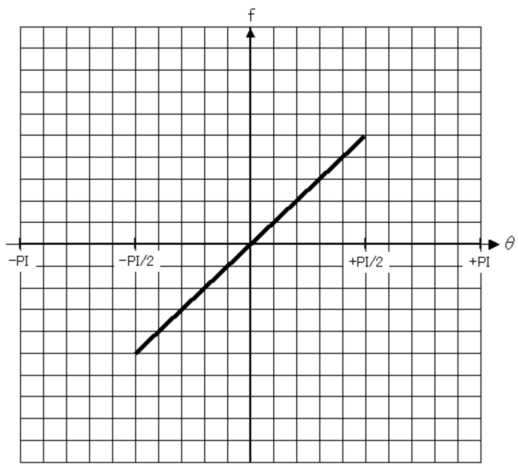


図 3　課題[2b]の制御関数

2007 年 12 月 5 日

[課題３]　次に，制御関数として比例関数を用いるのではなく，非線形関数であるガウス分布（正規分布）を用いて振り子を安定させてみましょう．

正規分布（ガウス分布）は，カール・フリードリヒ・ガウスが発見した次の式で表される確率密度関数を持つ確率分布です．μ =0，σ =1 のとき標準正規分布と呼ばれます．

$$y(x)=\frac{1}{\sqrt{2\pi\sigma}}e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

平均 : μ　　　分散 : σ^2

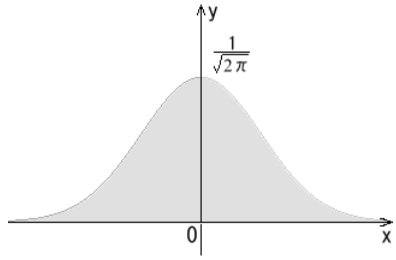


図 4 ガウス分布（μ =0，σ =1）

(a) まず，１つのガウス分布を用いて初期角度が-PI/2 から+PI/2 までの範囲で開始される条件のシミュレーションを作成して下さい．そして，ガウス分布の分散を変化することで，振り子の安定状態はどのように変わるのか確認してください．

<u>プログラム言語によるガウス分布表記</u>

```
mu=0;                // 平均μ
sig=1;               // 分散σ
f=(1/Math.sqrt(2*Math.PI*sig)*Math.exp(-(th-mu)* (th-mu)/(2*sig*sig)));
```

(b) 次にガウス分布を複数用いて，初期角度が-PI から+PI までの範囲で開始され安定となるような制御関数を作成してください．なお，下式は図５を示しています．

$$f(\theta)=-\frac{1}{\sqrt{2\pi\sigma}}e^{\frac{-(\theta-\mu)^2}{2\sigma^2}}+\frac{1}{\sqrt{2\pi\sigma}}\cdots(0\le\theta<+\frac{\pi}{2}),\quad f(\theta)=+\frac{1}{\sqrt{2\pi\sigma}}e^{\frac{-(\theta-\mu)^2}{2\sigma^2}}-\frac{1}{\sqrt{2\pi\sigma}}\cdots(-\frac{\pi}{2}<\theta<0)$$

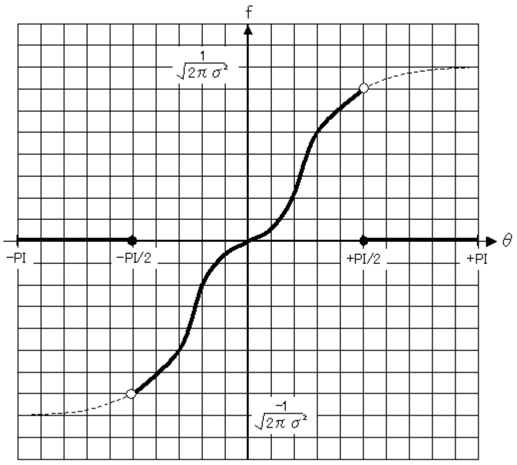


図５　課題[3a]の制御関数

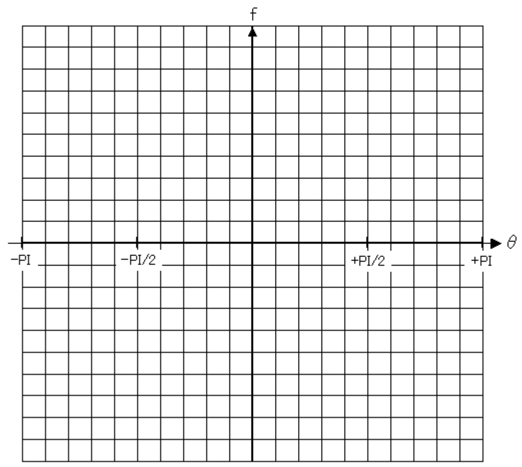


図６　課題[3b]の制御関数

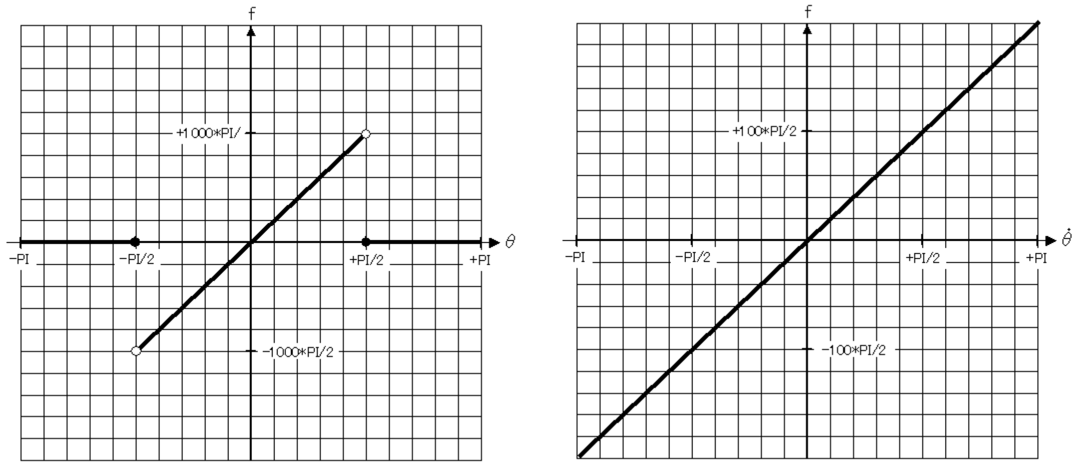
****** 補足１　******

ガウス分布により制御関数を作成すると倒立振り子が収束する場合があります．この収束は制御によるものではなく，数値計算の誤差の蓄積により起こっているとみなすことができます．これを解決するには刻み幅をより小さくする必要があります．

2007 年 12 月 5 日

[課題４] これまでの課題では，安定解は周期的な振動を出力するのみであり，角度が 0 度に収束することが基本的に見られませんでした．この倒立振り子システムにおいて摩擦を考慮していないため，振り子の速度に対しての減衰の項が無いのでこのような現象がおきたと考えられます．この課題では，制御入力変数（フィードバック変数）を振り子の角度１つのみではなく，制御入力変数を増やして倒立振り子の安定を求めます．

(a) まず，振り子の角度に以外に振り子の角速度を制御入力変数として用います．図 7 に示すような制御関数 f=1000*th+100*dth［-PI/2<th<+PI/2］を使用してシミュレーションを実行し，どのような挙動が得られるか述べよ．

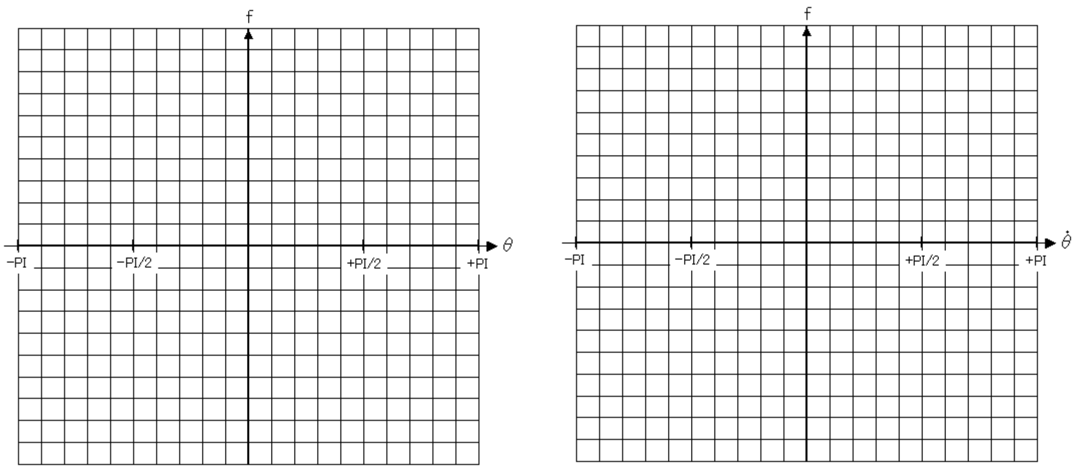


(i) 振り子の角度の制御関数 f=1000*th

(ii) 振り子の角速度の制御関数 f=100*dth

図 7 課題[4a]の制御関数 f=1000*th+100*dth; ［-PI/2<th<+PI/2］

(b) 振り子の初期角度が-PI から+PI までの範囲においても収束するように，制御関数を作成してください．

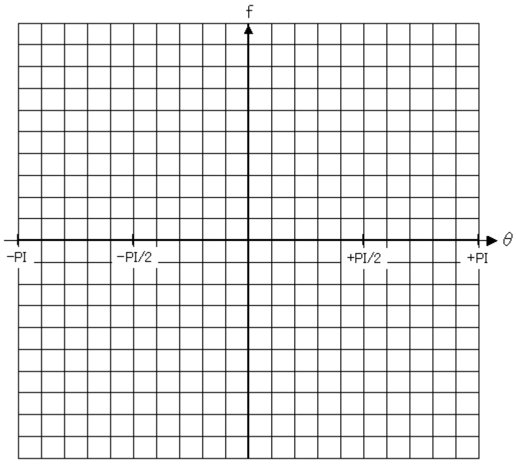


(i) 振り子の角度の制御関数

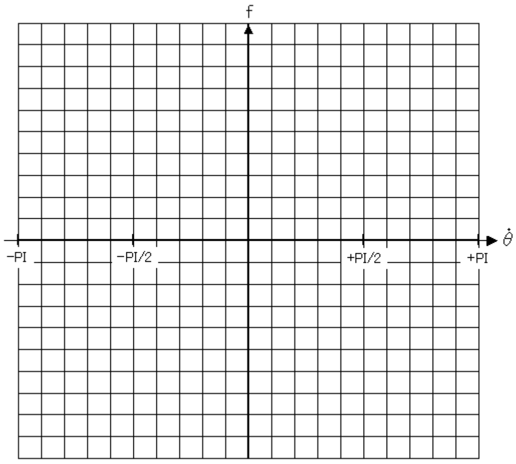
(ii) 振り子の角速度の制御関数

図 8 課題[4b]の制御関数 f=(i)+(ii)

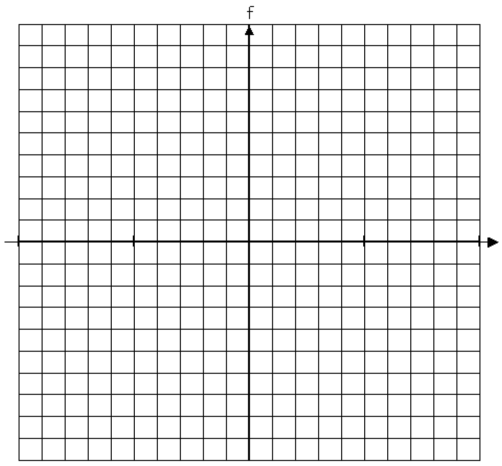
(c) また振り子の角度が収束したときに台車も静止するように，3 制御入力変数を用いて制御関数を作成してください．



(i) 振り子の角度の制御関数



(ii) 振り子の角速度の制御関数



(iii) 「　　　　　」制御関数

図 9 課題[4c]の制御関数 f=(i)+(ii)+(iii)

****

[レポート課題 1]

出来うる限り短時間で収束すると思われる倒立振り子の制御関数（例：ガウス分布など複数用いる）を作成し考察せよ．レポートには制御関数の式および図，プログラムも記述してください．

提出方法：横井教官宛（soft_computing@robot.t.u-tokyo.ac.jp）にメールで送付

提出ファイル名：kadai1_xxxxxx.doc，または kadai1_xxxxxx.pdf （※xxxxxx は学籍番号）

URL：http://www.robot.t.u-tokyo.ac.jp/dcm/lec/softcomp.htm

2007 年 12 月 5 日