

# ファジィ制御

## はじめに

本稿ではファジィ制御について簡単に紹介しようと思う。シミュレーションプログラムによる実例を示す。また PID 制御との比較も実施する。筆者自身、製品開発では PID 制御の経験はあるが、ファジィ制御については初学レベルである。あくまでも紹介程度の内容となる。

## PID 制御

まず PID の制御系を示す(図 1)。

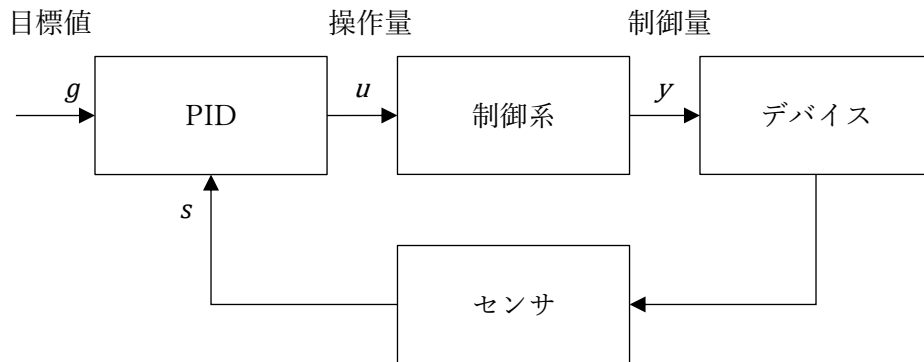


図 1 制御系(PID)

PID 制御の数式は(1)である。

$$u(g, s) = k_p d(g, s) + k_d \frac{d}{dt} d(g, s) + k_i \int_{\tau=0}^t d(g, s(\tau)) d\tau \quad (1)$$

$k_p$ : 比例ゲイン

$k_d$ : 微分ゲイン

$k_i$ : 積分ゲイン

$\tau$ : 経過時間

$g$ : 目標値

$s(t)$ : 観測値

$d(g, s) = g - s(t)$ : 偏差

$y = u$ : 簡単な制御系

目標値が定数ならば, (1)は(2)となる.

$$u(g, s) = k_p d(g, s) - k_d \frac{d}{dt} s(t) + k_i \int_{\tau=0}^t d(g, s(\tau)) d\tau \quad (2)$$

## 簡単な温度制御モデル

シミュレーションの前に制御対象のデバイス(ハードウェア)が必要である. この部分をシミュレーション(ソフトウェア)で用意することとする. 題材としては簡単な熱伝導モデルである (図2).

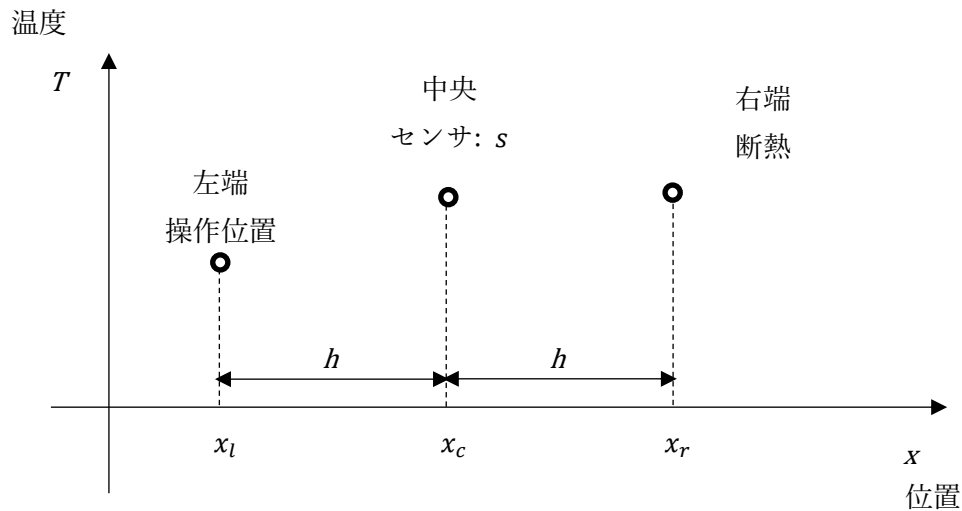


図2 温度制御のデバイスのシミュレーション

まず1次元熱伝導方程式は(3)である.

$$\frac{dT}{dt} = c \frac{d^2T}{dx^2} \quad (3)$$

$T$ : 温度(スカラー: 1次元)

シミュレーションとしては(4)である.

$$s \rightarrow g \quad (4)$$

観測信号は観測誤差があれば (5)と置き換えられる.

$$s(t) = s(t) \pm \sigma_s \quad (5)$$

$$\sigma_s: \text{観測誤差}$$

$$\sigma_s = 0 \text{ならば } s = x_c$$

さて、時間更新を(6)とすれば、(3)の微分方程式を差分近似して時間更新は(7)となる。この導出は容易である。詳細は専門書[2]等、参考いただきたい。

$$dt: j \rightarrow j + 1 \quad (6)$$

$$T_{x_{i-1},t_{j+1}} - pT_{x_i,t_{j+1}} + T_{x_{i+1},t_{j+1}} = -T_{x_{i-1},t_j} + qT_{x_i,t_j} - T_{x_{i+1},t_j} \quad (7)$$

$$p = 2(1 + 1/r)$$

$$q = 2(1 - 1/r)$$

$$r = \frac{c \cdot dt}{h^2}$$

(7)は有限差分法で用いられるクラック・ニコルソンの更新式である。本シミュレーションでは有限差分法で物性を表現するわけである。ここで $r$ を下記の定数で無次元化する(8)。

$$r = 0.4 \quad (8)$$

次に位置を(9)~(11)で定義して(7)に代入すれば(12)となる。

$$x_{i-1} = x_l \quad (9)$$

$$x_i = x_c \quad (10)$$

$$x_{i+1} = x_r \quad (11)$$

$$T_{x_c,t_{j+1}} = \frac{-T_{x_l,t_j} - T_{x_l,t_{j+1}} + qT_{x_c,t_j} - T_{x_r,t_j} - T_{x_r,t_{j+1}}}{-p} \quad (12)$$

以上より観測信号の更新は(13)となる。

$$s(t) = T_{x_c,t_{j+1}} \pm \sigma_s \quad (13)$$

右端(境界条件)については断熱として(14)で定義する。

$$T_r = T_{x_r,t_j} = T_{x_r,t_{j+1}} \quad (14)$$

左端では操作量 $y = u$ として(15)で定義する。これにより左端の時間更新は(16)となる。

$$u = T_{x_l,t_{j+1}} - T_{x_l,t_j} \quad (15)$$

$$T_{x_l,t_{j+1}} = T_{x_l,t_j} + u \quad (16)$$

(15)だが、一般には微分値により熱量(湧出)を与える。本シミュレーションでは簡単のため、直接、温度で与えることとした。

## ファジィ制御

ファジィ制御は PID ほど簡単ではなく、一定の学習期間を要する。理論については参考文献[1]を参照されたい。ただ、設計の流れを追うだけでも感触は伝わるのではないと思う。まず入力を2変数とする(17), (18)。

A,  $d(g, s)$  (17): 目標値と中央 $x_c$ における現在の温度との差(偏差)

B,  $\Delta s = \frac{d}{dt}s(t) = s(t_j) - s(t_{j-1})$  (18): 中央 $x_c$ における直近の温度変化

次にファジィ制御では**あいまいな命題**と**推論規則**を設計する。温度制御モデルでは**表 1**である。

A(入力) $\mu_{A'}(v)$ ( $v = d$ )	B(入力) $\mu_{B'}(w)$ ( $w = \Delta s$ )	C(出力): $\mu_{C'}(z)$ $C' = C_1 \cup C_2 \cup C_3 = D \cup E \cup F$
偏差 $d$ が小	温度変化 $\Delta s$ が小	熱量を小とせよ: $D (= C_1)$ $\bar{A}', \bar{B}' \rightarrow D'$ $\mu_{D'}(z) = \mu_{\bar{A}'}(v) \mu_{\bar{B}'}(w)$
	温度変化 $\Delta s$ が大	熱量を中とせよ: $E (= C_2)$ $\bar{A}', B' \rightarrow E'$ $\mu_{E'}(z) = \mu_{\bar{A}'}(v) \mu_{B'}(w)$
偏差 $d$ が大	温度変化 $\Delta s$ が小	熱量を大とせよ: $F (= C_3)$ $\bar{A}' \rightarrow F'$ $\mu_{F'} = \mu_{A'}(v)$
	温度変化 $\Delta s$ が大	

表 1 あいまいな命題と推論規則

一般に多重ファジィ推論ではマックスミニ合成よりもマックス積合成が用いられる。本稿でもこれを用いる。筆者は、独立事象の想定であればマックス積合成で十分と考える。他に計算コストからもマックス積合成になるのかと思う。ここで2重のファジィ推論のマックス積合成は(19)である。

$$\mu_{C'}(z) = \bigvee_v (\mu_{A'}(v) \cdot \mu_{A_i}(v)) \cdot \bigvee_w (\mu_{B'}(w) \cdot \mu_{B_i}(w)) \cdot \mu_{C_i}(z) \quad (19)$$

$A_i$ :  $A$ の要素

$B_i$ :  $B$ の要素

ダッシュ(′)の表記はその関数がとりうる要素の集合を表し,  $i$ は集合の要素を表す. マックス積合成については読み飛ばしてもらってかまわない. 次に非ファジィ化(あいまいな命題と推論規則を数値化)するためのメンバシップ関数を定義する(20)~(26)

$$\mu_A(d) = \left(\frac{1}{10}|d(g,s)|\right) \wedge 1 \quad (20)$$

$$\mu_{\bar{A}}(d) = 1 - \mu_A(d) \quad (21)$$

$$\mu_B(\Delta s) = \left(\frac{1}{10}|\Delta s|\right) \wedge 1 \quad (22)$$

$$\mu_{\bar{B}}(\Delta s) = 1 - \mu_B(\Delta s) \quad (23)$$

$$\mu_D(z) = (1 - |z - 1|) \vee 0 \quad (24)$$

$$\mu_E(z) = (1 - |z - 2|) \vee 0 \quad (25)$$

$$\mu_F(z) = (1 - |z - 3|) \vee 0 \quad (26)$$

入力のメンバシップ関数を示す(図3).

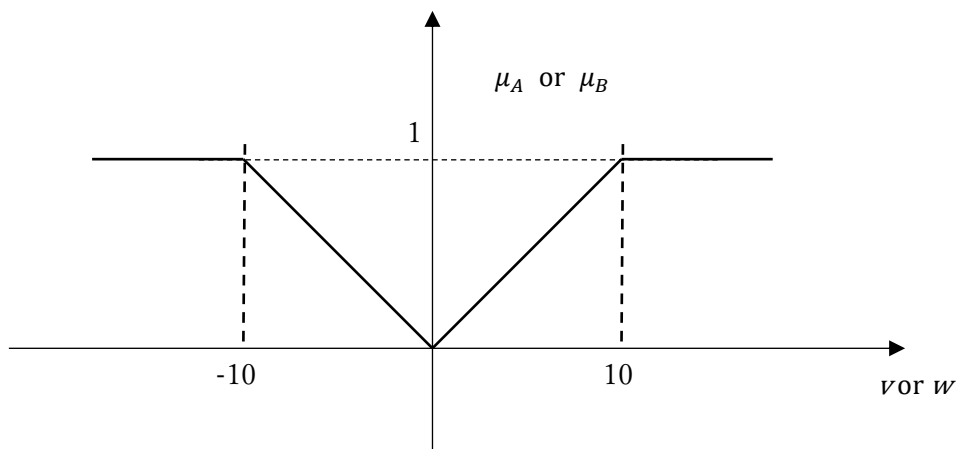


図3 入力のメンバシップ関数

出力のメンバシップ関数を示す(図 4).

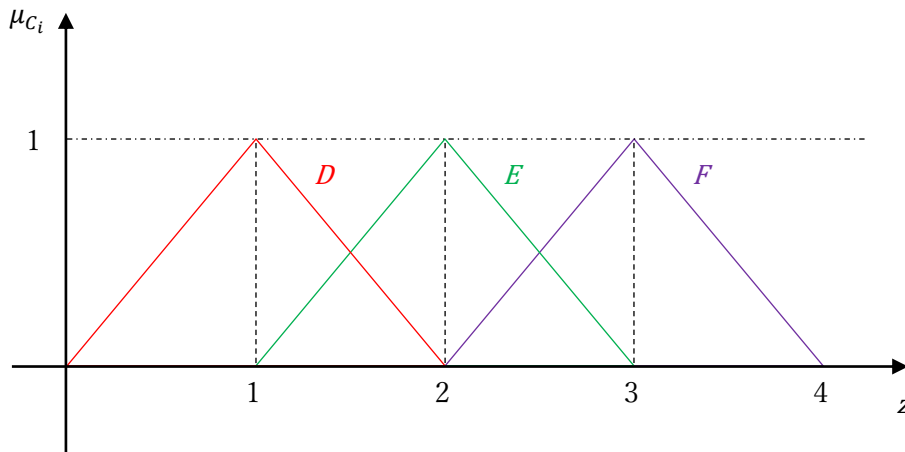


図 4 出力のメンバシップ関数

出力の合成(非ファジィ化)は重心法か面積法であるが, 本稿では図 4 に対応した面積法を用いる(27).

$$\mu = 1\mu_{D'} + 2\mu_{E'} + 3\mu_{F'} = \mu_{A'}\mu_{B'} + 2\mu_{A'}\mu_{B'} + 3\mu_{A'} \quad (27)$$

操作量  $u = \mu$  として, 温度制御したいわけだが, 命題を簡単(命題が少なすぎる)にしすぎたため, このままでは制御できない. 理由は  $u$  に符号がないからである. そこで図 1 の制御量  $y$  を用いることとする(28).

$$y = \text{sign}(d) \times \alpha u \quad (28)$$

$$\text{sign}(x) = \begin{cases} 1(x \geq 0) \\ -1(x < 0) \end{cases}$$

$\alpha > 0$ : 適当な定数

以上, 全てをファジィ制御とせず制御量  $y$  で拡張することとした.

## 温度制御のシミュレーション

初期条件, 目標値, 計測誤差を下記とする.

$$\text{左端: } T_{x_l, t_0} = 15^\circ\text{C}$$

$$\text{中央: } T_{x_c, t_0} = 11^\circ\text{C}$$

$$\text{右端: } T_{x_r, t_0} = 10^\circ\text{C}$$

$$\text{目標値: } g = 20^\circ\text{C}$$

$$\text{計測誤差: } \sigma_s = 0$$

## PID 制御のシミュレーション例

PID 制御のシミュレーション例を示す(図 5~図 8).

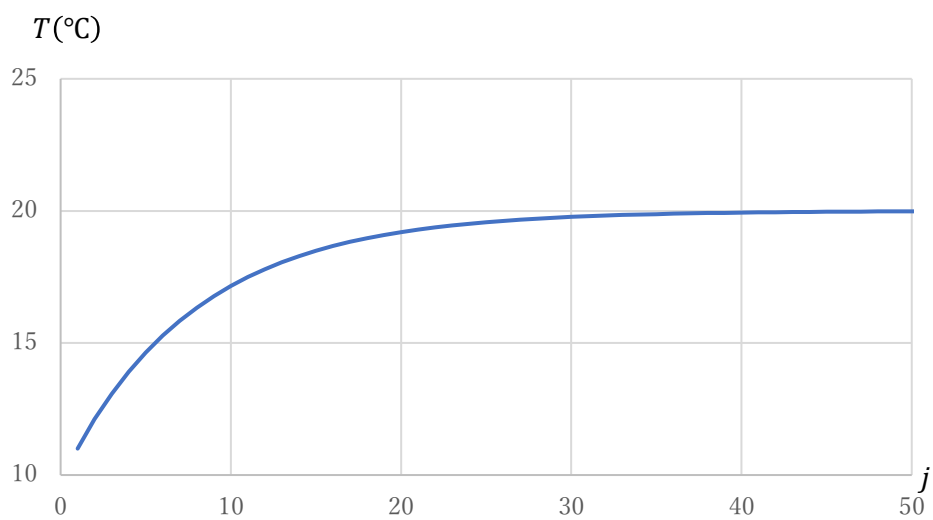


図 5  $k_p = 0.2, k_d = 0, k_i = 0$



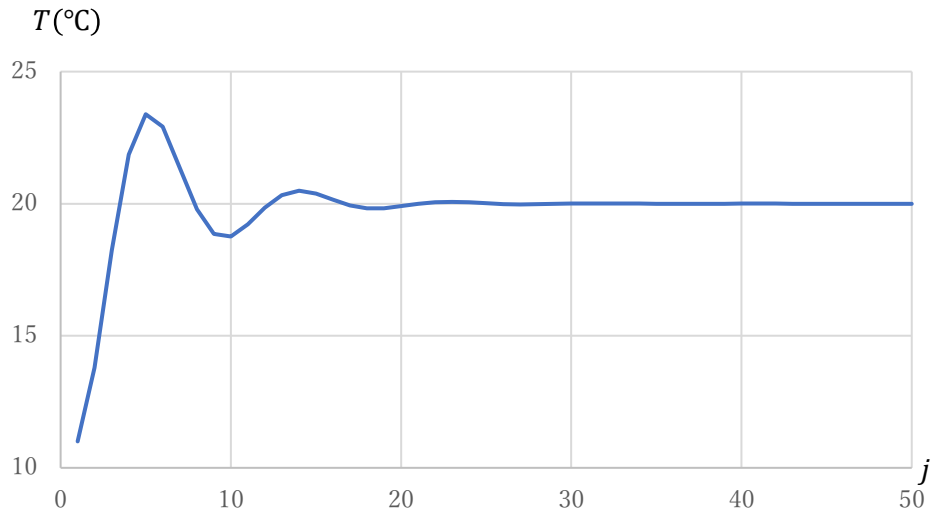


图 6  $k_p = 1.5, k_d = 0, k_i = 0$

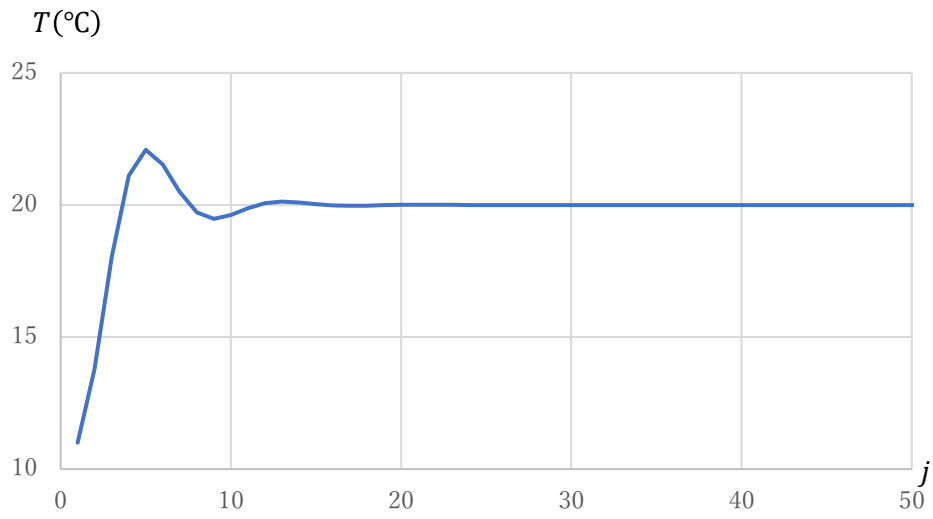


图 7  $k_p = 1.5, k_d = 0.5, k_i = 0$

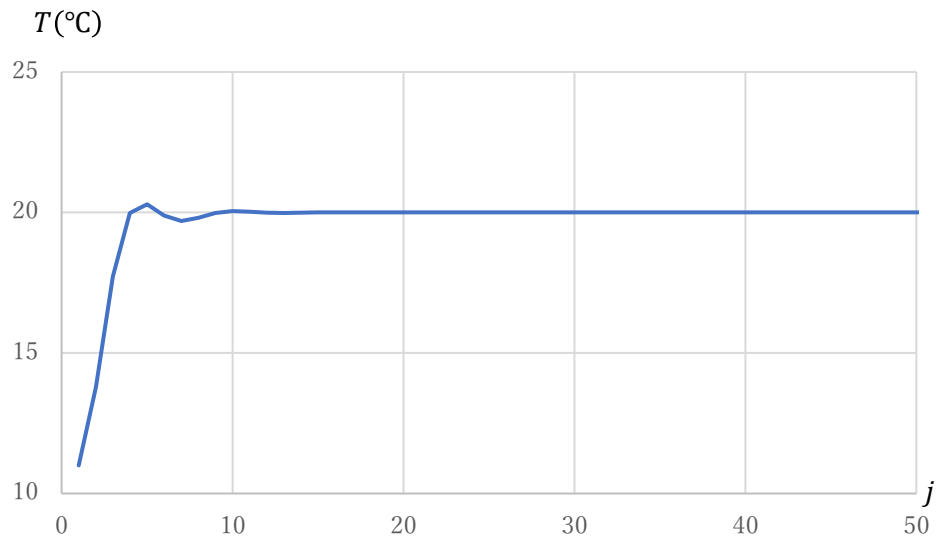


図8  $k_p = 1.5, k_d = 1.3, k_i = 0$

制御時間を短縮するため比例係数を大きくするとリンギングが発生する。微分項により、リンギングを抑えることができる。筆者は積分項を使ったことがないが PID の解説は多くみられる。他の URL なり文献を参照されたい。

## ファジィ制御のシミュレーション例

ファジィ制御のシミュレーション例を示す(図9, 図10).

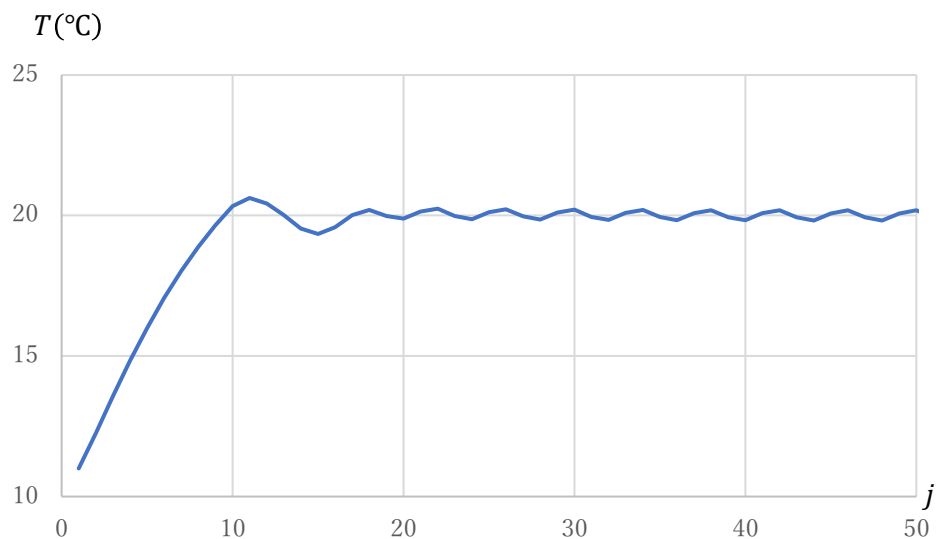


図9  $\alpha = 1$

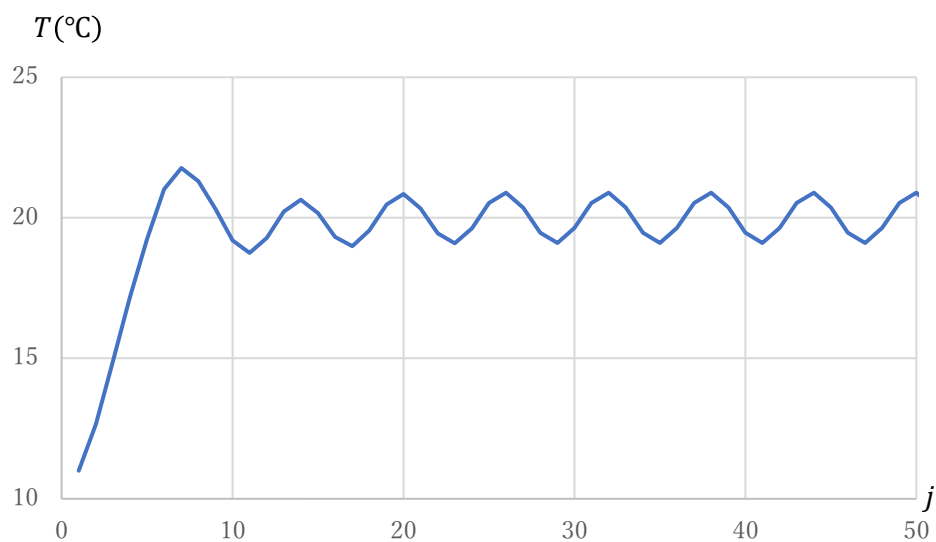


図10  $\alpha = 2$

制御時間を短縮しようとするともリングングが大きくなる。これは設計の問題で、振幅を小さくすることは可能である。リングングはリミットサイクルと呼ばれ、ファジィ制御に特有のものである。設計方法から自明であるが、これを完全に防ぐことは難しい。

## まとめ

PID 制御とファジィ制御, それぞれの特徴をまとめる.

### PID 制御

- \* デバイス制御で広く使われ, 確立された手法  
温度制御, モーター制御など
- \* 設計, プログラミングが容易  
単純でライブラリ化も容易
- \* 多変数への対応が難しい?  
筆者は多変数の PID について知見なし. 線形なら(1)の偏差を(29)として機能するか? といった処

$$d(g, s_1, s_2) = g - (\beta_1 s_1 + \beta_2 s_2) \quad (29)$$

### ファジィ制御

- \* 過去, 家電製品などで一世風靡したこともある. 起源は古く 1965~
- \* 設計, プログラミングが複雑化しやすいが自由度は高い. 非線形問題に強い  
ライブラリ化し難い
- \* 多変数の対応が容易  
変数間の相関を推論規則で表現可能.
- \* リミットサイクルが発生する
- \* 制御だけでなく, ファジィグラフやクラスター分析といった特徴抽出に役立つ体系もある[1]. 応用面で広大

## 参考文献

- [1] 山下元, 瀧澤武信. ファジィ理論 基礎と応用. 共立出版. 2010.
- [2] 堀之内總一, 酒井幸吉, 榎園茂 ANSIC による 数値計算法入門 第2版. 森北出版. 2002.