

# GA を用いた倒立振子の安定化制御

2001MM061 大平哲也

指導教員 高見勲

## 1 はじめに

従来の制御系設計方法では制御パラメータの調節には複雑な計算や、試行錯誤的な方法が必要であった。しかしソフトコンピューティングの登場により、パラメータを自律的に調整する事が可能となり、設計時の低コスト化に成功してきた。本研究では実際に代表的なソフトコンピューティング技法である遺伝的アルゴリズム [1](以下 GA) を用いて倒立振子の安定化制御系を設計することを通して、制御系設計における遺伝的アルゴリズムの有用性を検証する。

## 2 モデリングと線形化

制御対象の 2 リンク振上げ型倒立振子は、以下の図 1 のような様式である。リンク 1 の支点部分にモーターが

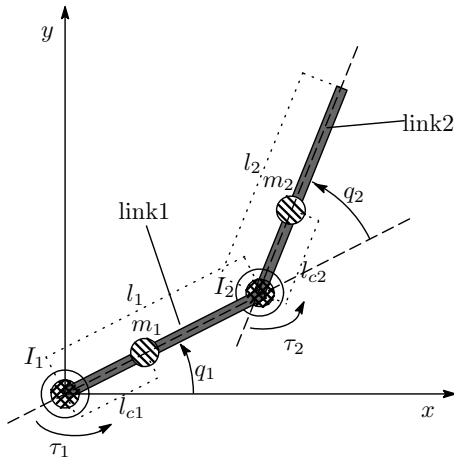


図 1 2 リンク振上げ型倒立振子のモデル

付けられており、それを動力とし回転する。リンク 2 の支点にはモータはなく、 $\tau_2=0$  となる。ラグランジュの運動方程式によりモデル化し、平衡点で線形化すると

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{\partial f_2}{\partial x_1} & 0 & \frac{\partial f_2}{\partial x_3} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{\partial f_4}{\partial x_1} & 0 & \frac{\partial f_4}{\partial x_3} & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ \frac{\partial f_2}{\partial u} \\ 0 \\ \frac{\partial f_4}{\partial u} \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

が得られる。但し多項式関数

$$f(x, u) = \begin{bmatrix} f_1(x, u) \\ f_2(x, u) \\ f_3(x, u) \\ f_4(x, u) \end{bmatrix} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix}$$

は倒立振子の運動方程式である [2]。

## 3 パラメータの自律調節

線形化で得られた状態方程式を用いて、図 3 のような状態フィードバック制御系を構成する。ここでは状態フィードバックゲイン  $K$  に対する指定極を GA を用いて探索することとする。

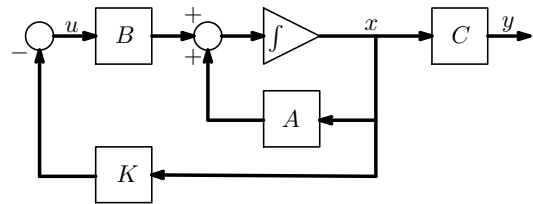


図 2 状態フィードバック制御のブロック線図

### 3.1 評価関数

評価関数として式 (1) を用いる。これは初期値応答  $x_1, x_3$  の 10 秒間の誤差面積  $S_{x_1}, S_{x_3}$  と応答、入力に関するペナルティ  $P_{x_1}, P_{x_3}, P_u$  の総和の逆数を表す。ペナルティ関数については  $P_{x_i}$  は平衡点からの変位が  $\frac{\pi}{6}$  を超える場合、 $P_u$  は入力の絶対値が 10 を超える場合、それぞれその超過分に比例する量のペナルティを課すこととした。これにより  $F$  を最大にする指定極が最適解となり、最終世代に生き残る事となる。

$$F = \frac{1}{S_{x_1} + S_{x_3} + P_{x_1} + P_{x_3} + P_u} \quad (1)$$

$$S_{x_i} = \int_0^{10} |x_i(t)| dt$$

$$P_{x_i} = \begin{cases} 0 & (\max(|x_i(t)|) \leq \frac{\pi}{6}) \\ W_{x_i} (\max(|x_i(t)|) - \frac{\pi}{6}) & (else) \end{cases}$$

$$P_u = \begin{cases} 0 & (\max(|u(t)|) \leq 10) \\ W_u (\max(|u(t)|) - 10) & (else) \end{cases}$$

$W_{x_i}, W_u$  :  $P_{x_i}, P_u$  に対する重み定数 ( $i = 1, 3$ )

### 3.2 コーディング方法

遺伝的処理を行うために、指定極を遺伝子型にコーディングする必要がある。交叉の都合上、支配極と非支

配極を区別した方が計算効率が良いので、求めるべき4つの指定極を

$$\mathbf{P} = \{-\alpha_1 \pm \omega_1 j, -\alpha_1 - \alpha_2 \pm \omega_2 j\}$$

とし、4つのパラメータ  $\alpha_1, \alpha_2, \omega_1, \omega_2$  をバイナリコードに変換することで遺伝子型とする。

### 3.3 探索結果

GAにより400世代目に状態フィードバックゲイン

$$\mathbf{K} = [87.917 \quad 8.896 \quad 116.548 \quad 14.031]$$

が得られた。

## 4 積分器拡大系

### 4.1 積分器の導入

3.3節で得られたゲインでは、応答にオフセットが残ってしまった。これを解消するため状態変数  $x_1$  に対する積分器を加えた拡大系を設計した[3]。新たに  $x_1$  を積分した値  $x_5$  を追加する。

$$x_5(t) = \int x_1(t) dt$$

### 4.2 コーディング方法

状態変数を追加したことにより新たに指定極を追加し、5つの極の最適な組合せを探索することとなるため、決定変数に一つの実数極を追加した。

$$\mathbf{P} = \{-\alpha_1, -\alpha_2 \pm \omega_1 j, -\alpha_2 - \alpha_3 \pm \omega_2 j\}$$

$\alpha_1, \alpha_2, \alpha_3, \omega_1, \omega_2$  がGAで求めるべきパラメータである。

### 4.3 探索結果

GAにより400世代目に状態フィードバックゲイン

$$\mathbf{K} = [169.816 \quad 17.767 \quad 220.032 \quad 28.365 \quad -24.080]$$

が得られた。図3(右)は得られたフィードバックゲインを用いた実験結果である。積分器を用いない場合と比べ、積分器を加えたことによりオフセットが解消されていることがわかる。

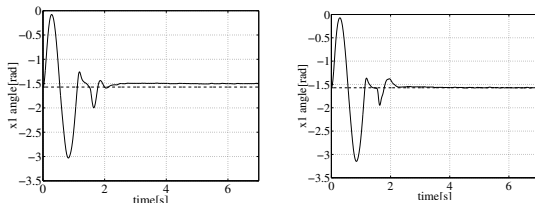


図3 積分器を用いなかった場合(左)と用いた場合(右)の応答 ( $x_1$ )

## 5 外乱耐性

以上では『良さ』の尺度として初期値応答の誤差面積を用いたがその他にも制御系を設計する際、『良さ』の測定は仕様により様々な物が考えられる。ここでは外乱に対する耐性を例に挙げ、トルク外乱に対する強さを尺度の基準とし、改めて設計することとする。

### 5.1 探索結果

GAにより400世代目に状態フィードバックゲイン

$$\mathbf{K} = [359.942 \quad 48.576 \quad 425.244 \quad 79.187 \quad -66.059]$$

が得られた。図4(右)は得られた制御系に整定状態から大きさ10[unit]のトルク外乱を加えた時の外乱応答である。4.3節のコントローラ(左)と比べ外乱による影響が抑えられていることがわかる。

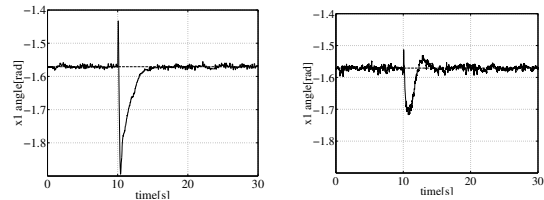


図4 4.3節のゲイン(左)と新たに得られたゲイン(右)を用いた際の外乱応答 ( $x_1$ )

## 6 おわりに

本研究では得られた成果と従来の方法との比較を行った結果、勝るとも劣らない性能のよいコントローラを設計することができた。比較対象の最適レギュレータでは、重みを決定する為の指標が無いので、ある程度試行錯誤的に調整して決めなければならない。それに対しGAを用いた方法では、応答の波形から直接評価し設計する事ができる為、設計の方法が直接的で分かりやすく、また仕様に合わせて柔軟な設計が可能だといえよう。

しかしGAにも計算コスト等の欠点がある。自律調整をするための評価方法の複雑さや個体数の大きさなどによっては莫大な計算コストがかかってしまう事がある。GAを用いる際にはこれらの欠点を踏まえた上で設計する必要がある。なお計算コストの問題に関しては、今後、計算機の処理能力の向上に伴い解消されることが予想できる。

### 参考文献

- [1] 三宮信夫、喜多一、玉置久、岩本貴司：遺伝アルゴリズムと最適化、朝倉書店(1998)
- [2] 坂野誠一、川津勇治：極配置によるペンダボットの制御系設計、南山大学数理情報学部数理科学科卒業論文(2003)
- [3] 岩井善太、石飛光章、川崎義則：制御工学、朝倉書店(1999)