

差分進化アルゴリズムによるファジィ制御器の最適化

星野 孝総

(受領日：2011年5月21日)

高知工科大学システム工学群

〒782-8502 高知県香美市土佐山田町宮ノ口185

E-mail:hoshino.yukinobu@kochi-tech.ac.jp

要約：本研究では、ライントレースカーの操舵角とモータの制御に簡略型推論法を用いてジグザグ走行をせず、滑らかに走行させるように制御することを目的とする。ライントレースカーは、走行中に読み取ったセンサの値を入力とし、操舵角と左右の後輪モータ出力を出力して走行するシステム構成になっている。本研究で設計したファジィ制御器は操舵角の制御器をモータ出力の制御器の入力としてあつかう多段ファジィ制御を採用している。推論規則の獲得やメンバーシップ関数のチューニングは複雑となり、人がチューニングするためには多大な労力と時間が必要となる。そこでわれわれは、ファジィ制御器に用いるファジィ推論器の調整パラメータが連続空間上にあることに着目した。多次元の連続関数最適化問題として、ファジィ推論器の自動チューニングを Differential Evolution (DE) を用いて行った。本報告では多変数のパラメータ自動チューニング結果について報告する。

1. 緒言

最適化問題は、オペレーティングリサーチの分野において古くから研究されている。特に数理的計画法や線形計画法を用いた研究は現在でも盛んに研究されている。また、人工知能や知能システムの分野においても進化計算などさまざまなアプローチが多く研究されている。最適化問題を解く場合、対象問題の背景に十分注意を払いながら、制約条件や目的関数の評価基準を定め、数式もしくはグラフ等のモデルにより問題の単純化を行なう。単純化された問題に対して処理アルゴリズムを適用し、シミュレーションを行うことによって最適化問題の解を得る。プロセスの制御や生産管理にコンピュータが導入される現在では、このような最適化問題を実システムで適用するが一般化している。さらに、コンピュータの処理能力向上やインターネットの急速な発展により、人間の意思決定の支援に役立たせようとする試みも多くなった。与えられた問題に対し、「なるべく良好な手段で解決したい。」という要求がある場合、必ず直面する課題である。そのため最適化問題は、あらゆる工学モデルの基礎となる重要な研究分野として認知されている⁽¹⁾⁽²⁾。

組合せ最適化問題の近似解法として、生物の情報処理にヒントを得た遺伝的アルゴリズムやホップフィールドニューラルネットワークなどが研究されてきた。遺伝子表現として、実数値ベクトルを用いた実数値 GA が連続関数の最適化手法として研究されている⁽³⁾。また、連続関数の最適化手法として、実数値 GA に代わる新しい手法が多く提案されている。そのような手法の代表的なものに Particle Swarm Optimization (PSO)⁽²⁰⁾⁽²¹⁾ や K.Price、R.Storn らによる Differential Evolution (DE) があげられる。これらの手法は、多点探索法であることと直接探索法であることから実数値 GA と同様に進化的計算に分類されている。

また一方、ファジィ理論は 1968 年に Zadeh によって提唱されて以来、私たちの身の回りのあいまいな情報を取り扱うための理論として研究されてきた。特徴は、設計者の持つ知識をコンピュータに取り込む際に、人の感覚に似た自然な表現のままに、知識を記述できることである。現在では広く産業界に定着しており一般化している。心理学、経済学などの文系の分野から、プラント、自動車、家電製品の制御など、工学分野まで幅広く適用されている⁽⁴⁾⁻⁽⁸⁾。また、太田らは、試作

したファジィ LSI を悪路耐久運転試験に適用し、ファジィ操舵制御で人間の操舵と同等の操舵補正ができることを実証している⁽⁹⁾。現在、対象システムの大規模化や複雑化が進んでいる。この場合、容易に設計できないほどの規則数となり、推論規則の獲得やメンバーシップ関数のチューニングを最適化問題として扱う場合が多い。自動チューニングや自動生成に関する研究は現在でも盛んに行われている⁽¹⁰⁾⁻⁽¹³⁾。

われわれはこれまでの研究において、ライトレースカーの操舵角とモータ出力の制御に簡略型推論法を用いることで、ジグザグ走行をせずに、滑らかに制御することに成功している^{(4), (16)-(19)}。そこでは走行中に読み取ったセンサの値のみを入力とし、ファジィ推論を用いて操舵角を求め、その後、出力された操舵角を入力とし、2段目のファジィ推論を用いて左右のモータ出力を求め、操舵角の制御器が1入力、モータ出力の制御器が2入力、合計出力が3出力の2段ファジィ制御である。ここでは、推論規則の獲得やメンバーシップ関数の調整を設計者が主観的に決定していた。そこで本研究では、ファジィ推論のパラメータ群が多次元連続空間上にあることに着目し、Differential Evolution (DE) をファジィ推論器の自動チューニングに適用した。そのチューニング方法とシミュレーション結果について本稿で報告する。

2. システムの構成

これまでの研究でわれわれは、1入力での多段ファジィ制御器をライトレースカーに実装し、実走行に成功してきた^{(4), (16)-(18)}。そして、ファジィ規則の設計などを、設計者の知識や経験が容易に反映でき、その有効性を示した。しかし、車などの運転を例に挙げると、人は、カーブの状況に応じてアクセルの踏み込みや緩めなどの行為をおこなう。1入力の制御器の場合、現在の操舵角の状況に対してのみモータの電力を出力している。現在の操舵角 θ_t と操舵角の変化量 $\Delta\theta_t = \theta_t - \theta_{t-1}$ の2つの入力から、モータの制御をおこなうことにより、ライトレースカーの減加速を考慮した制御器を試みた。操舵角制御については、従来の制御器と同じであるため、モータ出力の制御器についてのみチューニングを行なった。

研究で使用するライトレースカーについて述べる。ライトレースカーとは、白のラインをセンサで読み取り走行するもので、マイコンを

使って制御する。本研究では、Japan Micom Car Rally の規定 (<http://www.mcr.gr.jp>) にそったライントレースカーを用いて実験した。本研究で使用するライトレースカーの概観を Fig.1 に示す。

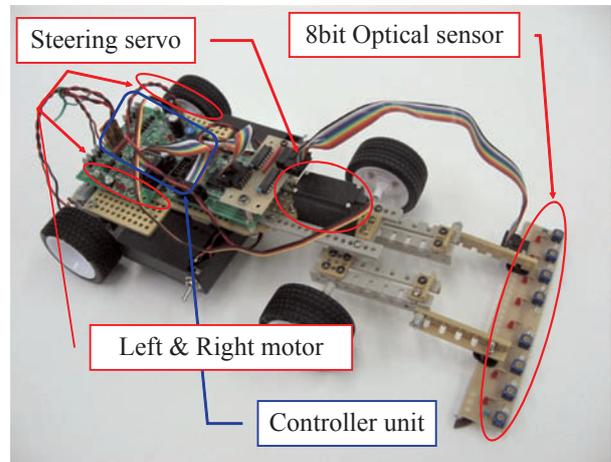


Fig.1 ライトレースカー

このライトレースカーは、HITACHI 製のマイコンカー製作キット Vol.3 を組み立てたものである。ギヤボックスには、ハイスピードギヤボックス HE を採用し、モータには RC-260RA18130 6V カーボンブラシを採用している。また、CPU には H8/3048F-ONE (HD64F3048BF25) を採用している。H8 の動作クロックは 24.576 [MHz] である。ライトレースカーは走行中に読み取ったセンサの値より制御量決定して制御を行う。制御する対象は、ライトレースカーの操舵用サーボモータの操舵角度と後輪左右モータの電力出力である。研究で用いる制御システムの構成では、操舵角を θ と左右のモータ出力を LM , RM とおいて制御し、ライトレースカーを動作させる。センサは白色を感知する 8 つの光センサがついている。この 8 つのセンサを 1 つのセンサ入力値 P にまとめ、1 入力 3 出力の制御器となるようにする。次に、ライトレースカーのセンサ入力値 P について述べる。手順は以下のようになっている。

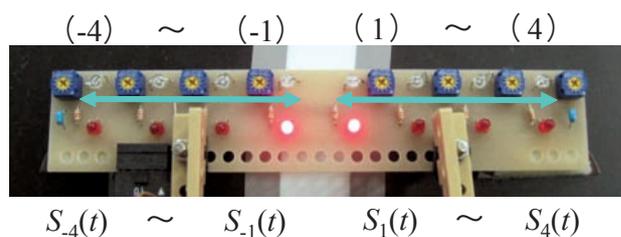


Fig.2 センサの値

Fig.2のようにセンサには(-4)から(-1)と(1)から(4)までの値を与える。点灯しているセンサの値の総和を出すために、センサ a のON、OFFの関数を $S_a(t)$ とし、(1)式に与える。

$$S_a(t) = \begin{cases} 1 & \text{センサ}a\text{がONの時} \\ 0 & \text{センサ}a\text{がOFFの時} \end{cases} \quad (1)$$

この総和は時刻 t において点灯しているセンサの個数となる。その時にあらかじめ与えておいた各センサ a の値は関数 $N_a(t)$ で与えられる。従って、次式のような関係になる。値 P は式 (2) (3) に示すように、 $N_a(t)$ の総和を $S_a(t)$ の総和で除算したものになる。式(4)参照。

$$S_a(t) = 0 \quad \leftrightarrow \quad N_a(t) = 0 \quad (2)$$

$$S_a(t) = 1 \quad \leftrightarrow \quad N_a(t) = a \quad (3)$$

$$P = \frac{\sum_{a \in A} N_a(t)}{\sum_{a \in A} S_a(t)} \quad (4)$$

但し、 $A = \{-4, -3, -2, -1, 1, 2, 3, 4\}$

そこで、本研究において制御対象のライトレースカーを制御する場合は、8つ(8bit)のセンサ値を P 値に変換してから制御量を計算する。したがって、制御器は P 値を入力、操舵角 θ と左右

のモータ出力 LM 、 RM とした1入力3出力制御器となる。ここでシステム構成図を Fig.3 に示す。

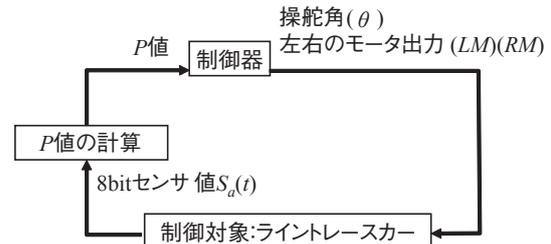


Fig.3 システム構成図

3. ライトレースカーのファジィ制御システム

ライトレースカーの制御器を Fig.4 に示す。制御器は2段の多段ファジィ制御器となっている。この制御器は、Fuzzy Controller1 の出力が Fuzzy Controller2 の入力になっているところが最大の特徴である。また、Fuzzy Controller1 のファジィ推論で推論された操舵角がそのまま出力され、さらに Controller1 で計算された操舵角 $\Delta\theta_t$ が操舵角 θ_t 共に Fuzzy Controller2 の入力になっている。ここで t は、システムの時刻をあらわしている。多段ファジィ制御を用いることで目的に合わせた制御器を組み合わせることが可能となり、全体で大きな制御器を設計するより、部分的に設計

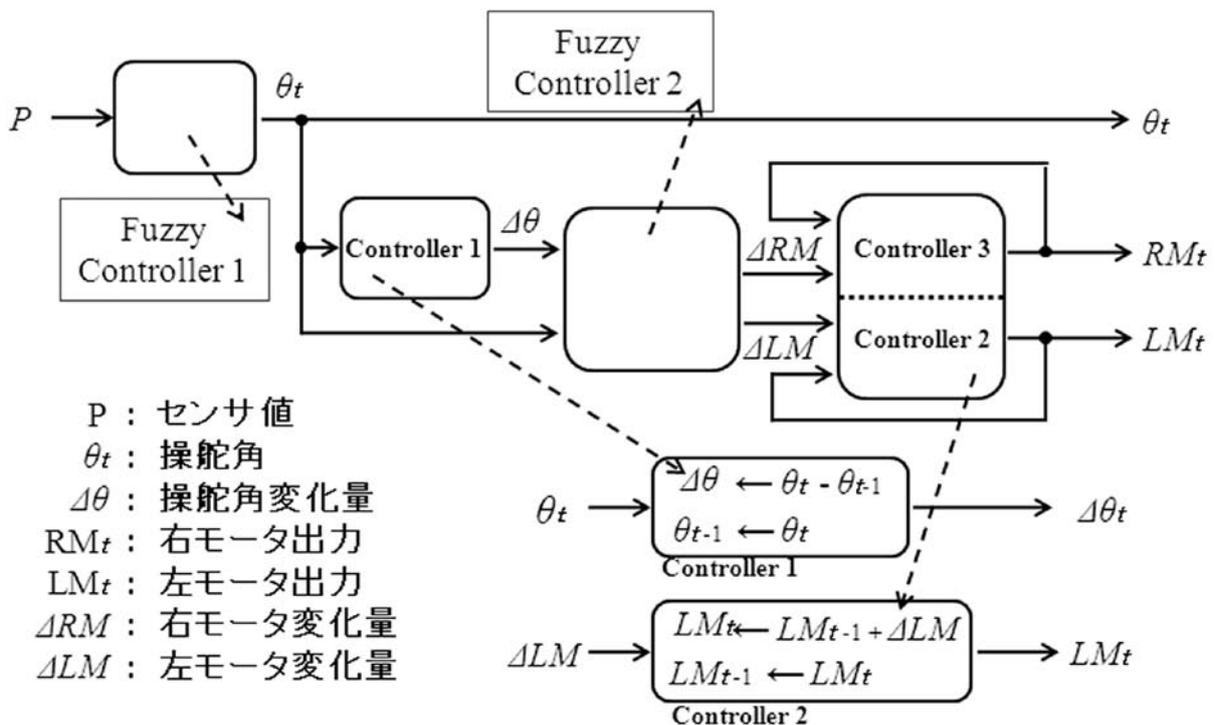


Fig.4 ライトレースカーのファジィ制御システム

できる。したがって、ファジィ変数とファジィ推論規則の組み合わせを少なくすることができる。Fuzzy Controller2 のファジィ推論規則を次に示す。

- R^1 : IF x_1 is A_1^1 and x_2 is A_2^1
then y_1 is B_1^1 , y_2 is B_2^1
- R^2 : IF x_1 is A_1^1 and x_2 is A_2^2
then y_1 is B_1^2 , y_2 is B_2^2
- R^3 : IF x_1 is A_1^1 and x_2 is A_2^3
then y_1 is B_1^3 , y_2 is B_2^3
- R^4 : IF x_1 is A_1^2 and x_2 is A_2^1
then y_1 is B_1^4 , y_2 is B_2^4
- R^5 : IF x_1 is A_1^2 and x_2 is A_2^2
then y_1 is B_1^5 , y_2 is B_2^5
- R^6 : IF x_1 is A_1^2 and x_2 is A_2^3
then y_1 is B_1^6 , y_2 is B_2^6
- R^7 : IF x_1 is A_1^3 and x_2 is A_2^1
then y_1 is B_1^7 , y_2 is B_2^7
- R^8 : IF x_1 is A_1^3 and x_2 is A_2^2
then y_1 is B_1^8 , y_2 is B_2^8
- R^9 : IF x_1 is A_1^3 and x_2 is A_2^3
then y_1 is B_1^9 , y_2 is B_2^9

ここで x_1 , x_2 は前件部変数となり制御対象の状態変数をあらわし、ファジィ制御器の入力変数となる。 y_1 , y_2 は後件部変数となり、制御対象の操作量であり、ファジィ制御器の出力変数となる。前件部変数 x_1 は操舵角 θ_t を表しており、 $x_1 \in \{A_1^1, A_1^2, A_1^3\}$ となる。一般に設計者が主観的に設計する場合は、右、中央、左のように言語表現を用い、これを言語ラベルと言う。また、この規則を言語規則と一般的に呼ぶ。しかし、今回の場合は自動チューニングを行なうため、言語表現を主観的に表現してもチューニングによって変わる可能性もある。このため具体的な言語ラベル表現をさげ、全体的に統一したファジィ変数をベースとした規則表現にした。この時の、 A_1^1 は操舵角度右側を表すファジィ変数であり、 A_1^2 は操舵角度中央を表すファジィ変数であり、 A_1^3 は操舵角度左側を表すファジィ変数としている。 x_2 は操舵角の変化量 $\Delta\theta_t = \theta_t - \theta_{t-1}$ がどのくらい変化したかを表しており、 $x_2 \in \{A_2^1, A_2^2, A_2^3\}$ とした。この時の、

A_2^1 は変化量増加を表すファジィ変数であり、 A_2^2 は変化量零を表すファジィ変数であり、 A_2^3 は変化量減少を表すファジィ変数である。また、後件部変数 y_1 は左モータ出力であり、 $y_1 \in \{B_1^1, B_1^2, B_1^3, B_1^4, B_1^5, B_1^6, B_1^7, B_1^8, B_1^9\}$ となる。また、 y_2 は右モータ出力であり、 $y_2 \in \{B_2^1, B_2^2, B_2^3, B_2^4, B_2^5, B_2^6, B_2^7, B_2^8, B_2^9\}$ としている。このようにファジィ変数表現で統一しておくことで、最適化問題にもちこみやすいようにしている。これらがファジィ制御の設計変数となっていく。設計変数を調整する場合、変数が多くなれば人が容易にファジィ規則を設計できない。それは組み合わせ数の問題に加え、制御規則の複雑さが問題になっていると考えられる。われわれは文献(16)及び文献(17)において単純な制御器を設計者により設計し、ライトレースカーを実走行させることに成功した。この時は単純化された規則であったため、直感的に設計できる程度の組み合わせであった。

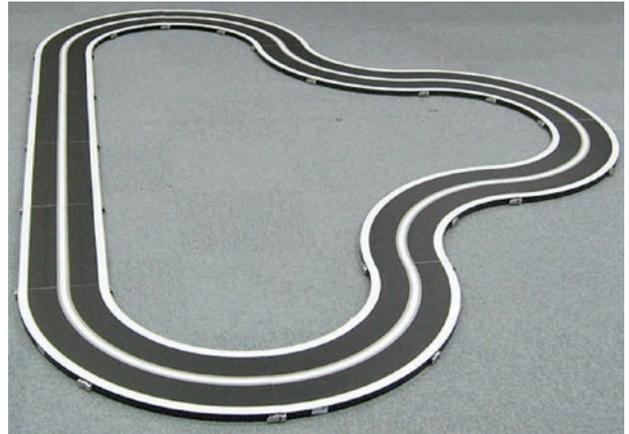


Fig.5 実験コース

本報告で用いた制御器はかなり複雑であり、人が調整するのは非常に困難となる。設計には多大な時間がかかる。そこで、各パラメータの変数は連続値をとるため、多次元の連続関数最適化問題として扱い、各変数を DE で自動チューニングさせる。

今回は実走行に成功してきた文献(4)及び文献(16)-(18)の結果から、Fig.5に示すようなコースを走行した時のセンサ値 P とモータ出力、操舵角度のデータからそれぞれの教師データを作成した。コースは、全長は約 10m。直線 600m(T600) × 4 枚、カーブ半径 450m(R450) × 16 枚を使用した。オーバルコースのコースにくらべ、左右のカーブが存在するコースである。このとき、フレーム毎の教師データを変数 TD と置く。推論結

果のモータ出力を変数 RO と置く。推論結果のモータ出力 RO を教師データ TD に近づけるような各パラメータ変数のチューニングをする最適化問題を解くこととした。具体的には、 TD と RO の誤差関数 E を最小化する。この時、調整するパラメータは以下のように 24 変数となる。

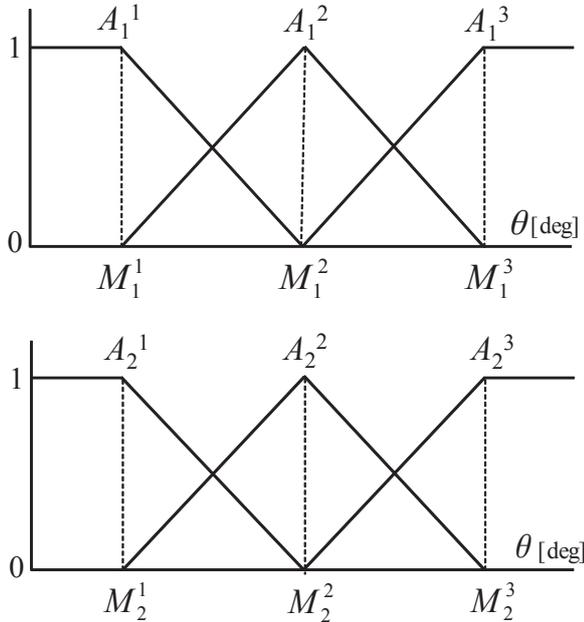


Fig.6 入力のパラメータ変数
(上：操舵角 下：操舵角変化量)

Fig.6 上の操舵角のファジイ変数 $\{A_1^1, A_1^2, A_1^3\}$ の位置を示す $\{M_1^1, M_1^2, M_1^3\}$ はメンバーシップ関数のパラメータ変数である。また、Fig.6 下の操舵角変化量のファジイ変数 $\{A_2^1, A_2^2, A_2^3\}$ の位置を示す $\{M_2^1, M_2^2, M_2^3\}$ は変化量のメンバーシップ関数のパラメータ変数である。

Fig.7 上にある左モータ出力のファジイ変数は、 $\{B_1^1, B_1^2, B_1^3, B_1^4, B_1^5, B_1^6, B_1^7, B_1^8, B_1^9\}$ とし、 $\{S_1^1, S_1^2, S_1^3, S_1^4, S_1^5, S_1^6, S_1^7, S_1^8, S_1^9\}$ はそれら位置を示す変数であり、シングルトン関数のパラメータ変数である。9 段階に分かれている。また、Fig.7 下にある右モータのファジイ変数は、 $\{B_2^1, B_2^2, B_2^3, B_2^4, B_2^5, B_2^6, B_2^7, B_2^8, B_2^9\}$ とし、 $\{S_2^1, S_2^2, S_2^3, S_2^4, S_2^5, S_2^6, S_2^7, S_2^8, S_2^9\}$ はそれら位置を示す変数であり、シングルトン関数のパラメータ変数である。

TD と RO は左右のモータ出力がある。教師データの左モータ出力を TDL 、教師データの右データ出力を TDR と定義し、推論結果の左モータ出力を ROL 、推論結果の右モータ出力を ROR と定義する。最小にする値を e と定義し、式(5)に示す。

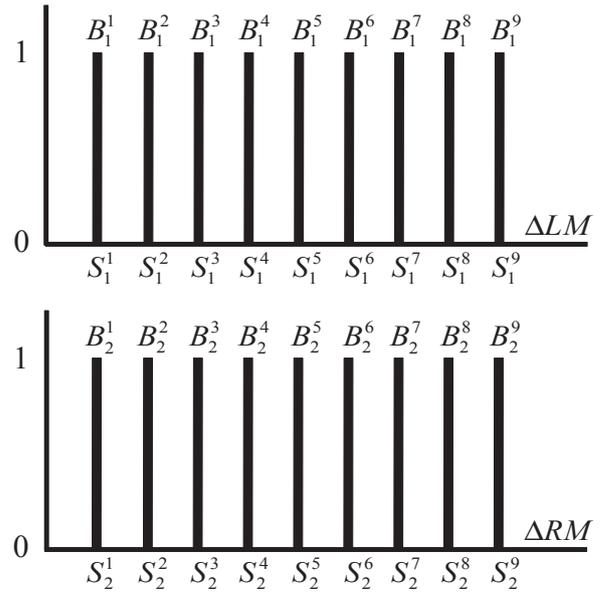


Fig.7 出力のパラメータ変数
(左：左モータ 右：右モータ)

$$e_i = TD_i - RO_i \quad (5)$$

今回の実験では過去に走行したデータを用いてその誤差を最小化する。その走行データは、コース一週分あり、一周で 202 回センサを読み取って走行したデータを用いた。したがって、最小化する誤差関数は各モータの 2 乗誤差、202 回分の総合計を用いることとした。そこで目的関数は式(6)のようになる。

$$E(x) = \sum_{i=1}^{202} \{(TDL_i - ROL_i)^2 + (TDR_i - ROR_i)^2\} \quad (6)$$

上記の誤差関数 E を目的関数として、 DE を用いて誤差最小値探索問題を解く。最適化関数を式(7)に示す。

$$\begin{aligned} & \text{minimize} && E(x) \\ & \text{subject to} && M_1^1 < 0, \quad M_1^3 > 0 \\ & && M_2^1 < 0, \quad M_2^3 > 0 \end{aligned} \quad (7)$$

なお、本来 ROL の出力を得るのに、用いられる変数は、 M_1 が 3 変数、 M_2 が 3 変数、 S_1 が 9 変数の計 15 変数とる。また、 ROR の出力を得るのに、用いられる変数は、 M_1 が 3 変数、 M_2 が 3 変数、 S_2 が 9 変数の計 15 変数となるため、合計 30 変数となる。その内、入力のパラメータ変数 $M_1^1 \sim M_1^3$ と $M_2^1 \sim M_2^3$ は同一になるため、重複する 6

変数を考慮する必要がないので、24 変数となる。さらに、 M_1^1 と M_2^2 はメンバーシップ関数の性質より変数は 0 とするよう設計したため、22 変数の最小値探索問題とする。

4. 差分進化アルゴリズムの適用

差分進化 (Differential Evolution ; DE) は K.Price、R.Storn らによって提案された。DE は確率的な探索法であり、解集団を用いた多点探索をおこなう。DE の重要な特徴としては、突然変異率などのパラメータを制御する必要がなく、単純な数学的演算で計算できるが挙げられる。このため、制御パラメータの数が少なく設定が容易であり問題への実装も比較的容易におこなえる^{(2) (5) (15)}。

DE 世代交代モデルを Fig.8 に示す。今回用いた DE は最もポピュラーなアルゴリズムである。差分変位親を作る際に元となる基本個体をランダムに選ぶ、差分をとる際に選ばれる個体対は 1 組とする。交叉方法は Fig.9 のように 2 点交叉を用いた。 N_D 次元の実数値空間、 N_p 個の個体 $x_i (i=1, 2, \dots, N_p)$ を与えた場合、個体の選び方がランダム、差分を取る際に選ばれる個体対が 1 組、交叉の長さの決定方法が指数的な二点交叉である DE アルゴリズムについて述べる。

Step1 : N_p 個の個体を、各次元の定義域においてランダムに生成して世代 $g=1$ とする。また、最終世代、収束の条件に設定する。

Step2 : 各個体の関数値を計算する。

Step3 : 差分変位親個体 v_i を生成する。

Step3-1 : 対象親個体 x_i と異なる 3 個体 $x_{p1,i}$ 、 $x_{p2,i}$ 、 $x_{p3,i}$ を同じ個体がかさらないようにランダムで選択する。

Step3-2 : Fig.8 中の式によって差分変位親個体 v_i を生成する。ここで S は差分の伸縮を表すスケールリングファクタである。 S の値が大きいほど差分変位親個体に対する差分の影響が大きく現れる。

Step4 : Fig.9 のように両親の交叉によって子個体 u_i を生成する。最初に交叉の始点を決める。次に、繰り返し発生させた 0 から 1 の大きさの乱数が連続で交叉率 C_R 以下である回数を l として、始点を含めず l 個目までの要素を交叉により交換する。このとき、第 1 要素に戻って同様の操作を続ける。子個体の成分としては、始点を含めて $(l+1)$ 個の差分変位親個体 u_i の成分が用いられ、それ以外の $(N_D - (l+1))$ 個の成分は対象親個体の x_i 成分が用い

られる。

Step5 : 対象親個体 x_i と子個体の関数値を比較し、良い方を次世代の x_i として残す。

Step6 : 終了条件を満たしていなければ、 $g=g+1$ として Step3 に戻る。

これらの Step における世代交代モデルを Fig.8 に示す。また DE の差分操作により生成される差分変位親個体の存在可能領域は、個体集団の存在領域を外側に拡張したものとなる。差分変位親個体と世代交代の対象となる対象親個体を交叉させてできる子個体も同様に、親個体集団の存在領域を外側に拡張した領域に存在し得る。これにより個体集団の存在領域の外側を探索することができる。Fig.10 に探索空間の推移を示す。DE を用いて、チューニングしている際、メンバーシップ関数のパラメータ変数より、シングルトンのパラメータ変数に依存すると考えた。設計者の知識を反映させることにより、メンバーシップ関数のパラメータ変数、 $M_1^1=30$ 、 $M_1^2=30$ に、また $M_2^1=20$ 、 $M_2^2=20$ に固定して目的関数を設計した。これにより、18 変数の最小値探索問題として扱う。

Table 2 DE パラメータ設定

DEのパラメータ	DE
個体数	100
最大探索回数	15000
試行回数	10
突然変異率	F=0.5
交叉率	CR=0.9

本実験は差分変位親個体を作る際に元となる基本個体をランダムで選び、差分をとる際に選ばれる子個体が 1 であり、指数関数的に減少する確率で交叉をおこなう手法を用いた。Table .2 に数値実験の設定について示す。個体数は 100 個体、最大探索回数 15000 回、試行回数は 10 試行、突然変異率 0.5、交叉率 0.9 に設定している。これは知能情報通信研究所の伊藤らの研究報告を参考にして実験をおこなった。Fig.11 に 1 試行分の誤差最小値探索問題における目的関数の値の変化を示す。縦軸は目的関数の値を表し、横軸は評価回

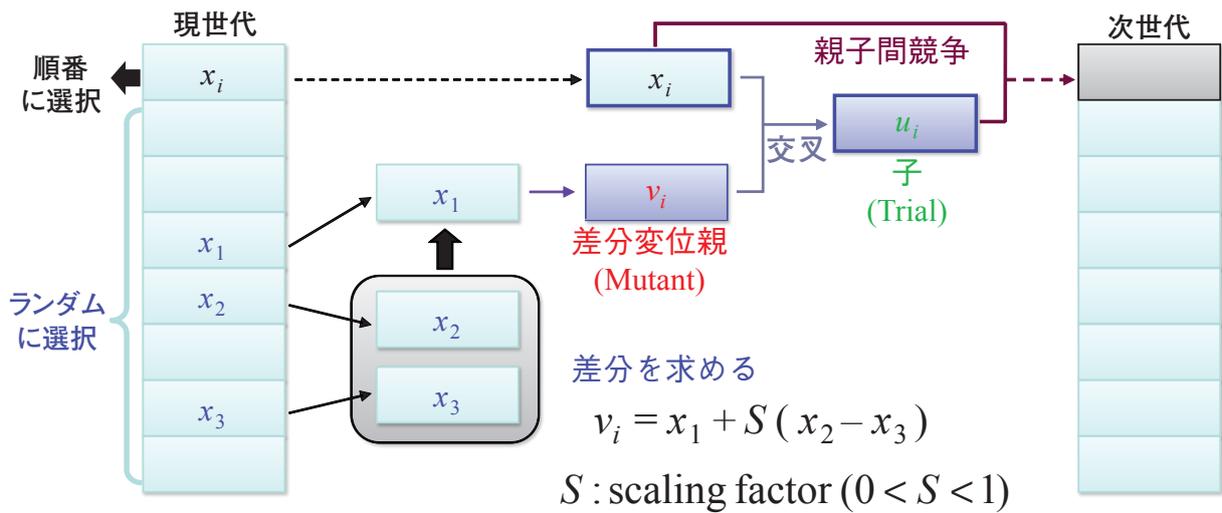


Fig.8 DE 世代交代モデル

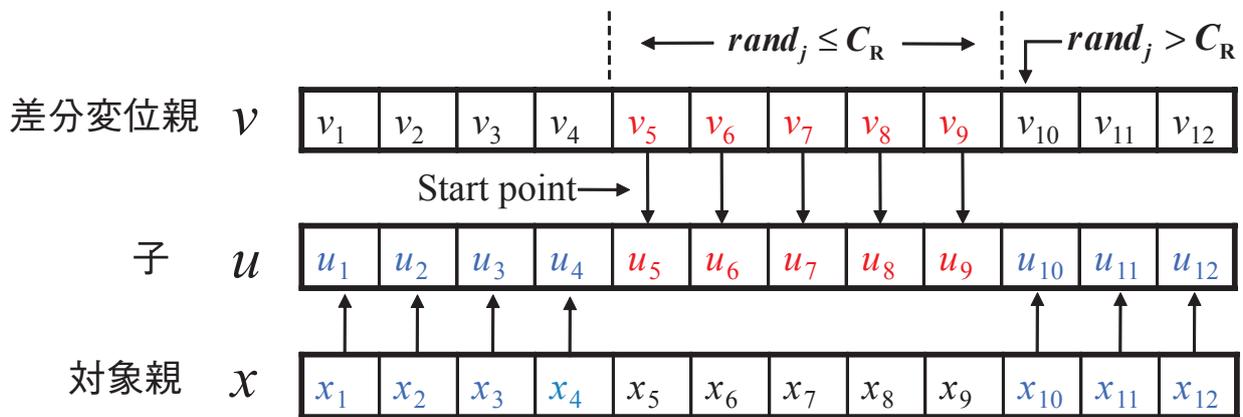


Fig.9 DE の二点交叉

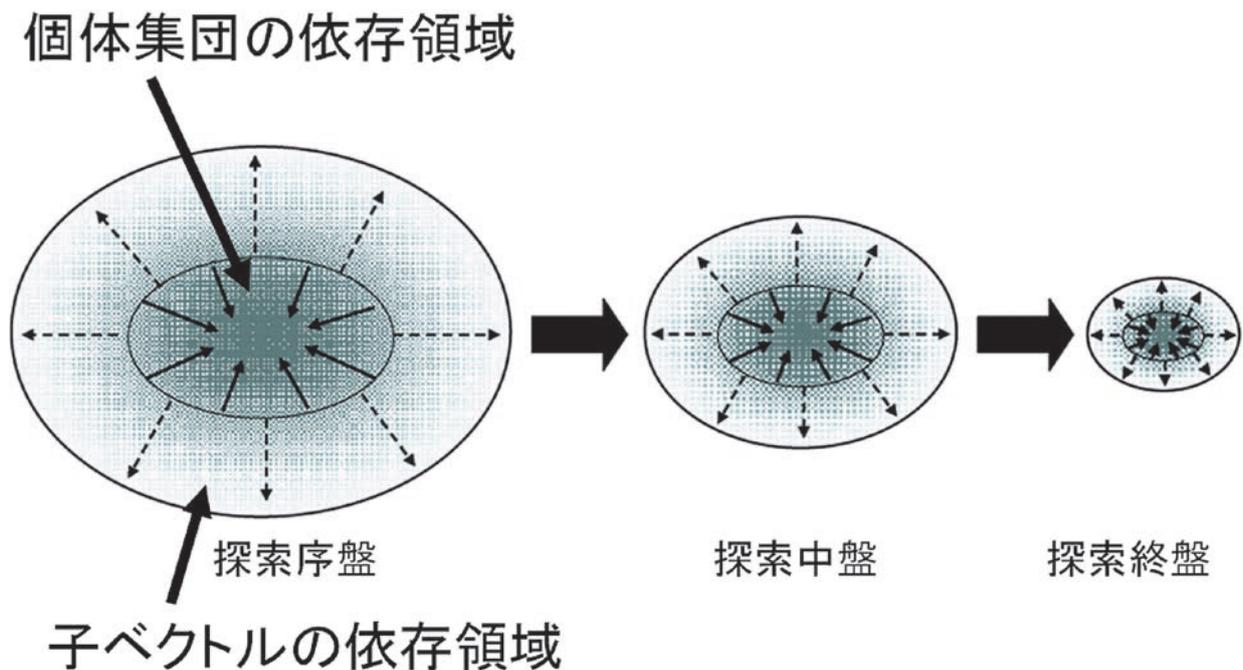


Fig.10 探索空間の推移

数を表す。グラフ上では、評価回数 4000 程度で徐々に収束し始め、評価回数 9487 で完全に収束した。10 試行中 10 試行、最適解を求めており、評価回数は平均 10326 回で収束した。このシミュレーション結果より求められたパラメータ変数を Table 3 に示す。

Table3 DE により求められたパラメータ

変数	B_1	変数	B_2
S_1^1	-40.355	S_2^1	-35.117
S_1^2	-6.339	S_2^2	19.252
S_1^3	37.184	S_2^3	32.853
S_1^4	0.027	S_2^4	0.068
S_1^5	0.072	S_2^5	-0.026
S_1^6	-0.432	S_2^6	0.179
S_1^7	19.576	S_2^7	12.820
S_1^8	25.291	S_2^8	6.037
S_1^9	-37.788	S_2^9	-56.658

DE により得られた結果の中で最も良いものを Fig.12 に示し考察してみる。得られたファジィ推論器の結果による左右のモータ出力と教師データでの左右のモータ出力 (Fig.13) を比較してみると、かなり類似しており、H8 マイコンに実装した場合同程度の走行性能が出ると考えられる。今回の実験では、DE での自動チューニング性能を検証するため Table2 のパラメータのみをつかって数十回程度実行した。全て同様の特性を示しており DE が多段ファジィ制御器の自動チューニングに有効的な手法になると考えられる。今後はさまざまなパラメータで検証する必要があると考えられる。実験に用いた計算機は、市販の PC で Windows 7 Professional edition 32bits を搭載し、Intel(R) Core(TM) i5 M560 2.67GHz 2.67GHz、メモリ 2.00GB である。開発には Microsoft VC/C++ 2010 Express Edition を用いた。実験システムは C 言語で作成して実験を行なった。制御器に変化量を加えたことにより、推論規則の獲得が困難となる問題が起きた。これを人間が調節するには膨大な時間がかかる。DE を用いてチューニングをおこなった場合、チューニング時間は約 1 分程度となり、DE による効率性とその有効性を示せた。

4. 結言

本研究ではライントレースカーの制御方法に現在の操舵角と操舵角の変化量を入力とし、モータの制御をおこなう制御器の設計を検討した。ライントレースカーの減加速を考慮した場合、変化量を考慮する必要があると考えたためである。この場合、ファジィ推論規則の設計や調整しようとする、複雑で膨大な時間がかかってしまうことがわかった。そのため、ファジィ推論器の自動チューニングを多次元の連続関数最適化問題として考え、最適解を DE によって求めた。実験では DE を用いた自動チューニングを数回行なった。その結果チューニングされた制御器の出力結果からチューニングの効率性とその有効性を示せた。今後の課題としては、実装実験を行ない、ライントレースカーの挙動を検証する予定である。

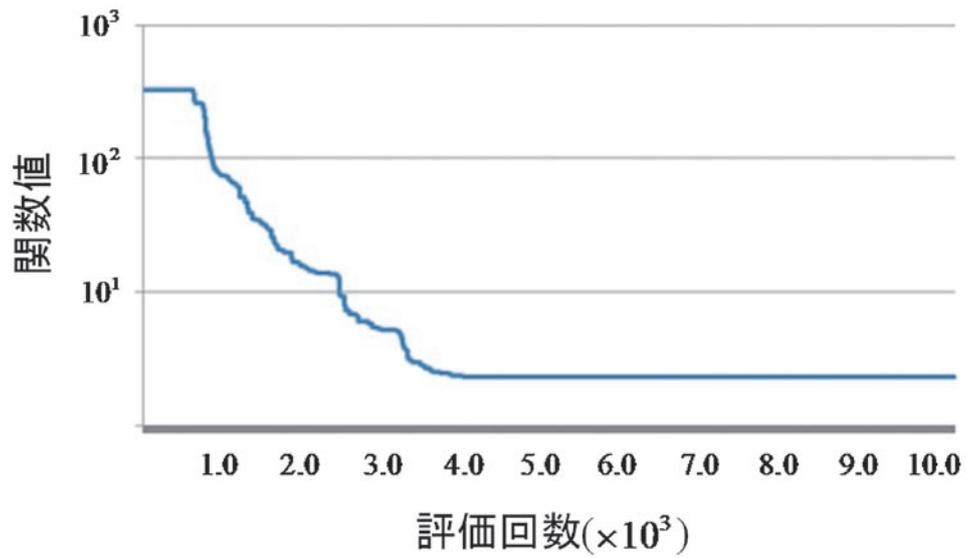


Fig.11 1 試行ごとの評価関数値の変化

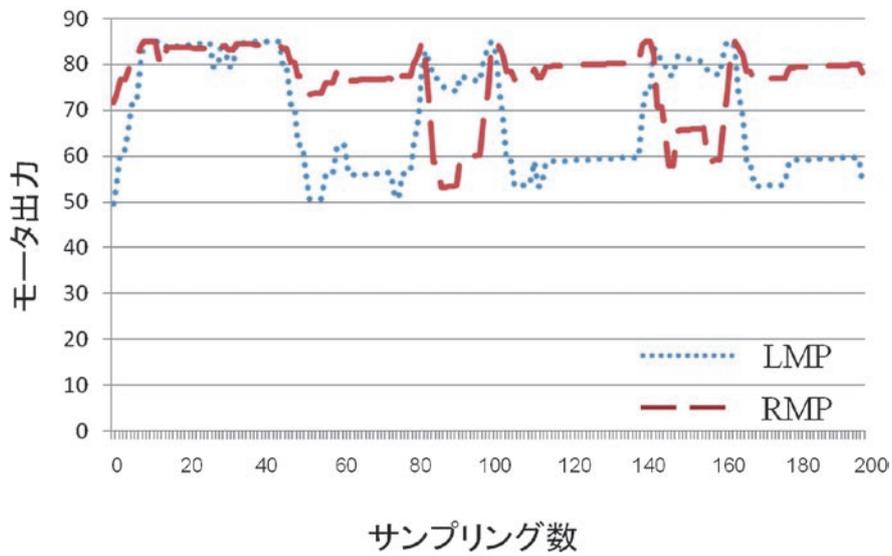


Fig.12 ミュレート実験結果(左右のモータ出力)

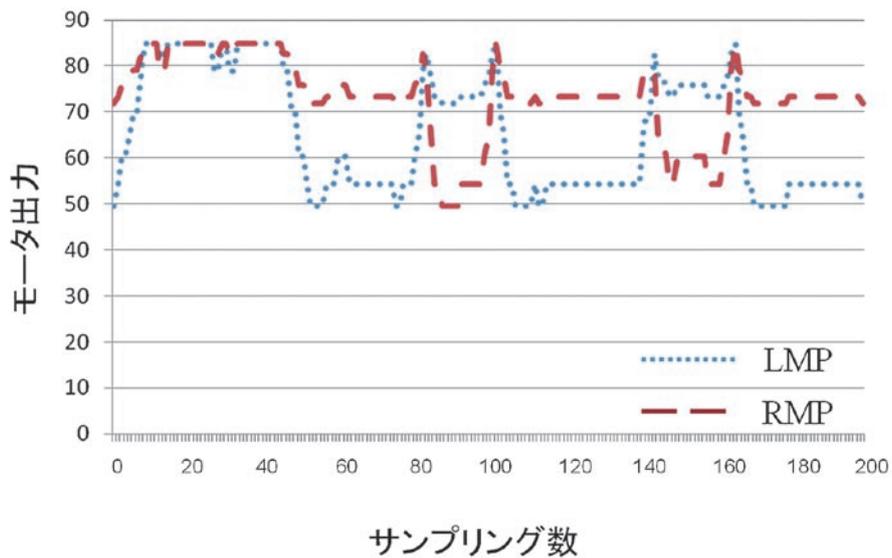


Fig.13 教師データでの左右のモータ出力

文献

- (1) 北野宏明：遺伝的アルゴリズム，産業図書株式会社(1993)
- (2) 伊藤稔：関数最適化のための Particle Swarm Optimization, Differential Evolution, 実数値遺伝的アルゴリズムの探索性能に関する検討，甲南大学紀要。理工学編, Vol.52, No.1, pp. 125-135, (2005)
- (3) 三宮信夫：遺伝的アルゴリズムと最適化，朝倉書店(1998)
- (4) Hiroshi TAKIMOTO, Yukinobu HOSHINO : Development of Fuzzy controller for the steering angle and the motor power control application to the line trace car, *Proceedings of FUZZ-IEEE 2009*. CD-ROM #2072. (2009)
- (5) R.M. Storn, K.V. Price, "Differential Evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces," Technical Report, International Computer Science Institute, Berkeley. (1995)
- (6) 萩原将文：ニューロ・ファジィ・遺伝的アルゴリズム，産業図書(1994)
- (7) 安部重夫：ニューラルネットとファジィシステム，近代科学社(1995)
- (8) 時永祥三：第3回 ファジィ推論システムの最適構成と推定への応用，オペレーションズ・リサーチ 1月号, Vol.47, No.1, pp.32-37 (2002)
- (9) 太田則一，原田義久，早川清春：自動車用ファジィ LSI. 豊田中央研究所 R&D レビュー Vol.30, No.2, pp.45-55 (1995)
- (10) 石淵久生：パレード最適ルールを候補ルールとして用いる遺伝的ファジィルール選択 日本知能情報ファジィ学会誌 Vol.20, No2, pp.231-243 (2008)
- (11) 吉田均：PSO を用いた構造物のファジィ振動制御装置の配置ルールの同時最適化 日本知能情報ファジィ学会誌 Vol.20, No6 pp.921-933 (2008)
- (12) 井上博行，亀井且有，井上和夫，遺伝的アルゴリズムと超円錐形メンバーシップ関数によるファジィルール自動生成手法の提案，日本ファジィ学会誌, Vol.8, No.6, pp.1104-1115 (1996)
- (13) 中岡伊織，亀井且有，金子隆臣：ニューロ・ファジィによる無リスク/有リスク資産の投資割合の最適化，第18回ファジィシステムシンポジウム講演論文集, pp.397-400 (2002)
- (14) 矢部博：工学基礎最適化とその応用，数理工学社(2006)
- (15) 串田淳一，大場和久，亀井且有：REAL : Differential Evolution における関数評価回数削減の提案，進化計算学会論文誌 Vol.1, No.1, pp.79-88 (2010)
- (16) 瀧本浩志，星野孝総：ライントレースカーにおける操舵角とモータ出力のファジィ制御器の開発，第53回システム制御情報学会研究発表講演会の講演論文集 CD-ROM, F41-1 (2009)
- (17) 瀧本浩志，星野孝総：操舵角とモータ出力における多段ファジィ制御のライントレースカーへの実装と検証，第25回ファジィシステムシンポジウム演会論文集 CD-ROM, 1E2-03 (2009)
- (18) 瀧本浩志，星野孝総：ライントレースカーにおける多段ファジィ制御器の開発，54回システム制御情報学会研究発表講演会論文集 CD-ROM, W26-3 (2010)
- (19) 瀧本浩志，岸雅基，木屋亮，福岡佑太，星野孝総：ライントレースカーにおける多段ファジィ制御法の提案，第26回ファジィシステムシンポジウム演会論文集 CD-ROM, WG3-1 (2010)
- (20) J. Kennedy and R. C. Eberhart, "Particle swarm optimization," *Proceedings of IEEE International Conference on Neural Networks*, Piscataway, NJ. pp.1942-1948 (1995)
- (21) J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," *Proceedings of IEEE international conference. Systems, Man, and Cybernetics: Computational Cybernetics and Simulation*, Vol. 5, pp.4104-4108 (1997)

Optimization of a fuzzy controller using a differential evolution algorithm

Yukinobu HOSHINO*

(Received : May 21st, 2011)

*School of Engineering, Kochi University of Technology
185 Miyanokuchi, Tosayamada, Kami city, Kochi 782-8502

E-mail:* hoshino.yukinobu @kochi-tech.ac.jp

Abstract: In this research, a fuzzy controller system is used as a key technique to design a machine controller. A fuzzy system can be designed using fuzzy sets, which can be represented using linguistic descriptions of natural language. These natural languages are able to capture human knowledge in fuzzy rules. A fuzzy controller system consists of fuzzy rules and the fuzzy inference techniques. Recently, research on fuzzy control has demonstrated the benefits of machine control using fuzzy rules. Through our research, we were able to verify that fuzzy control can be applicable to a line tracing car, which is running at a high speed. When carrying out our research, the adjustments and setup of the fuzzy controller required much time. In order to apply another Fuzzy controller to our line tracing car, we designed a simple inference system, and setup a Differential Evolution (DE) as a search algorithm. The DE was able to search a candidate solution of the fuzzy label. In addition, the auto-tuned fuzzy inference system can be controlled smoothly. In this paper, we present the search results of the simulation experiment and compare the results between the fuzzy controller and the auto-tuned controller using a DE.