カオス結合系を用いたGPによるファジーロボットの障害物回避モデル

3P−3　村井　保之*　松村　幸輝**　巽　久行*　徳増　眞司*
神奈川工科大学*　跡見学園女子大学**

1．はじめに

環境から得られた情報に基づいて自律的に障害物回避が遂行できるようなエージェント指向型ロボットの構築が望まれている(1)．これを実現する一方法として，ファジイ制御が有効であると考えられる．この場合，良好な制御を行うためには適切なメンバシップ関数の設定とチューニングがきわめて重要となる．

この観点から，筆者らは，進化システムのひとつである遺伝的プログラミング(2)（以下，ＧＰ）を用いてメンバシップ関数を自動的に生成する方法を提案した(3)．本論文では，この方法をカオス結合系を用いてＧＰのパラメータを動的に操作することで，メンバシップ関数の生成効率を改良することを試みた．

2．遺伝的プログラミング

ここでは，ＧＰの基本的な表現方法をそのまま適用し，数式で構成した木構造を用いてメンバシップ関数を表現し，交叉，突然変異などの遺伝的操作を施して，プログラム個体の進化を試みる．

3．障害物回避モデル

表１に，回避シミュレーションを行う作業空間の例を示す．作業空間には移動障害物と目標物が存在し，ロボットが移動障害物を回避しながら，スタート位置から目標物に到達するように動いていく．ロボットの基本的な動きと障害物回避方法を次のようにモデル化する．

(1)目標物の位置および障害物の位置や動き方はロボットにはあらかじめ与えず，センサで認識するものとする．センサの位置はロボットの前方中央位置とする．

(2)目標物および障害物の情報（距離と対象物に対する位置）をもとに移動方向を決定する．

(3)ロボットは，ｘ軸の正の方向へは常に一定の速度で移動するものとし，障害物回避および目標物接近のためにｙ軸方向方向にのみ加速できるものとする．

4．ファジイ制御による軌道修正

ロボットの軌道修正は，障害物および目標点との位置関係に基づき，ファジイ制御でｙ軸方向の速度を変化させることによって行う．

ＧＰにより生成されたメンバシップ関数の例を図１に示す．図１(a)は，入力関数でロボットと対象物（障害物あるいは目標点）との距離をラベル「近い，遠い」の２つで表したものである．図１(b)は，ロボットと対象物に対する相対位置（ロボットの中心と対象物の中心の差）をラベル「左側，やや左，中央，やや右，右側」の５つで表した．図１(c)は，出力関数で，ロボットのｙ軸方向の移動量（移動方向と速度）を「左，中央，右」の３つで表すものとする．

表１：作業空間と作業条件

パラメータ	設定値
作業空間の大きさ	640×400
障害物	円（半径 40）
目標物の位置	(560,200)
ロボットスタート位置	(60,200)

5．遺伝的プログラミングによるメンバシップ関数の生成

生成される関数は 10 個の関数が全て独立に生成されるものとした．また，データ構造として，個々の関数は二分木構造をもつものとしたが，常に 10 個の関数でひとつのグループとして捉え，一連の入出力関数で１個体を構成するものとした．そのため，交叉処理では，関数個々ではなくグループ単位で選択され，そのグループ間の同じ配列番号をもつ関数同士で遺伝操作が行われるものとした．また，適応度も個々の関数としてではなく，グループごとに設定した．これは，回避シミュレーションによる適応度は 10 個の関数を同時に用いて得られた走行結果から計算されるからである．そして，この回避シミュレーションによる適応度に，関数の形態に基づく個々の適応度を加えたものを全体の適応度とした．上述の個体を乱数で 100 個生成し初期集団とした．この集団に対し適応度に基づくルーレット法で交叉を行う．交叉により新たに発生した個体は元の集団

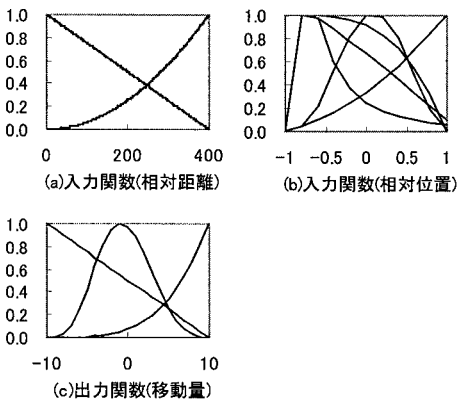

図１　生成したメンバシップ関数の例

Obstacle avoidance model of the fuzzy robot by the GP that used a Coupled Chaotic System

Yasuyuki Murai*, Koki Matsumura**, Hisayuki Tatsumi*
Shinji Tokumasu*,
Kanagawa Institute of Technology*
Atomi University**

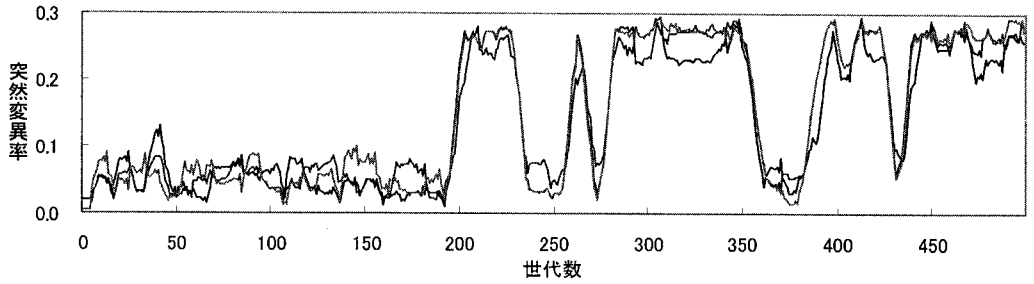

図２　カオス結合系のグラフ

に追加する．ただし，集団の数が 100 を越えた場合は適応度が最も低い個体と入れ替え，一定の個体数を保つようにした．

まず，回避シミュレーションに基づく適応度は，ロボットが障害物に接触せずに，かつ走行距離が短いものほど高くなるようにした．適応度が最も高くなる場合はロボットがスタート位置から目標点まで一直線に走行した場合とする(障害物が無い場合)．また，適応度が最も低くなる場合は，最悪の経路を通り目標点に到達しなかった場合とする．この処理を，１つの個体について，障害物の位置を変えて所定の回数だけ行い，この合計を最終的な回避シミュレーションの適応度とした．

次に，関数の形態に基づく適応度を設定する．ここで，メンバシップ関数の形態は，ファジイ制御の原理から考えると，それぞれの入出力関数が増加関数になるか減少関数になるか，あるいは凸型関数になるか，といったような基本的な事項は明確であって，むしろそのような条件を考えないで設計はできない．したがって，関数を自動生成する場合にも上のような条件が必要となる．この条件を満たせば適応度が高くなるような設定をする．最終的に，得られた回避シミュレーションの適応度と関数の形態の適応度を加算し個体の適応度とした．

ただし，進化の初期段階では，ある程度メンバーシップ関数としての形態が整っていない状態で回避シミュレーションを行っても多大な時間を要することになるので，回避シミュレーションを行わずに進化させ，関数の形態に関する適応度が一定以上になった後に回避シミュレーションを行うようにした．

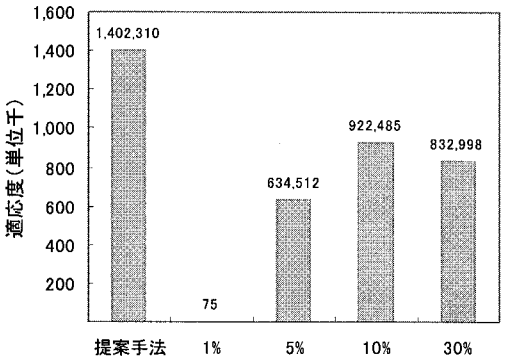

図３　突然変異率別平均適応度

６．カオス結合系を用いた突然変異率の操作

ＧＡやＧＰでは，交差率や突然変異率の設定が重要になってくる．本論文では，不動点の転移が見られるカオス結合系を用いて，突然変異率を動的に変化させることを試みた．突然変異率の低い状態で最高適応度の変化のない世代が続く場合，交叉の影響で集団中の個体の多様性が薄れてくる．そこで，カオス結合系の不動点の転移により，突然変異率を自動的に高くし個体の多様性を増し，より良い個体の出現を期待する．しかし，突然変異率を高いままにしておくと，交叉により適応度の高くなる可能性のある個体も突然変異してしまう可能性が高くなる．そこで，さらに一定期間最高適応度の変化がない場合は，カオス結合系の不動点の転移により，突然変異率を自動的に低くする．このように突然変異率を動的に変化させることで，解の探索効率が高くなると考えた．

７．実験による評価

提案手法と，突然変異率を固定した方法をそれぞれ，15 回づつ乱数の種をかえて実行しその結果を比較した．カオス結合系は図 2 に示すものでロジスティック関数を用いたものを使用し，値が 0～0．3 の範囲になるようにし，移動平均をとり調整した．突然変異率固定の場合は，突然変異率を 1%，5%，10%，30%と変え実験した．なお，交叉率は 5%，遺伝操作回数は 4000 回，個体数は 100 個体とした．実験の結果を図 3 に示す．この図は，指定した回数だけの遺伝操作が終了した時点での各実験の最高適応度の平均を求めたものである．この図から提案手法が，突然変異率を固定した場合にくらべ平均適応度が高くなっており，探索効率が向上していることがわかる．

８．むすび

以上，カオス結合系を用いてＧＰで生成されるメンバシップ関数を改良する手法について検討した．

これより，カオス結合系による突然変異率の動的変化はメンバシップ関数の効率的な生成に有効に働くものと考えられた．

参考文献

(1)松村:電学論,119-C,5,603-614(1999).

(2)Koza,J.:Genetic Programming, MIT Press(1992).

(3)松村,村井:信学論,J83-A,12,1539-1551(2000).

(4)金子,池上:複雑系の進化的シナリオ,朝倉書店(1998).