計測自動制御学会東北支部　第 273 回研究集会(2012.6.29)
資料番号 273-10

ニューラルネットワークを用いた ナビゲーション手法による移動ロボットの遠隔操作支援

Supporting Remote Control of a Mobile Robot by Navigation Method Using Neural Network

○手塚洸*　　釜谷博行**　　工藤憲昌**
○Ko Tezuka*　　Hiroyuki Kamaya**　　Norimasa Kudoh**
*八戸工業高等専門学校 専攻科　**八戸工業高等専門学校
*Hachinohe National College of Technology Advanced Engineering Course
**Hachinohe National College of Technology

キーワード：ニューラルネットワーク(Neural Network), ロボット制御(Robot Control), Android 端末(Android Devices), 障害物回避(Obstacle Avoidance)

連絡先：〒039-1192 八戸市田面木字上野平 16－1　八戸工業高等専門学校　電気情報工学科
釜谷博行,　Tel.: 0178-27-7283,　E-mail: kamaya-e@hachinohe-ct.ac.jp

1. はじめに

災害救助や宇宙開発の分野での利用を目的とで、移動ロボットの自律移動を制御する手法に関する研究が盛んに行われている。特に、柔軟性や自律性に優れている強化学習などの人工知能を用いた手法が注目されてきた[1]。しかし、近年ではロボットに要求される行動が複雑化してきており、実問題に完全に対応できる人工知能の実装は非常に難しく、また、柔軟な判断で要求に答えることは難しい。

一方で、人間がロボットを直接操作する場合、柔軟な判断が可能であるという長所があり、災害救助や宇宙開発の分野においては人工知能による自律移動制御よりも、人間が直接操作する方が適していると考えられる。しかし、人間の操作ミスによってロボットや外部環境を損傷させてしまうなどの問題がある。

そのような問題に対応するため、与えられた入出力関係から適切な非線形システムを生成することのできるニューラルネットワーク[2]を用いる手法に着目した。本研究ではニューラルネットワークを用いて、人間のロボット操縦を支援し、操作ミスによるロボットの損傷を最小限に抑えるための方法を検討する。また、Android 端末に搭載されている各種センサを用いた、より直観的なインターフェースによるロボットの操縦システムについて検討する。

2. ニューラルネットワーク

生体の神経細胞をモデル化した複数のユニットを接続して構成されるネットワークをニューラルネットワークという[3]。ニューラルネットワークは、入力-出力例を教師信号として提示することで、内部システムのパラメータを変更し、目的の機能を学習して非線形システムを自動生成することができる。また、Fig.1 に示すような、入力層、中間層、出力層からなるニューラルネットワークを階層型ニューラル

ネットワークという。

教師パターンの入力信号をニューラルネットワークに入力し、出力された信号と教師パターンの出力信号を比較して誤差を求め、その誤差が小さくなるように最急降下法を用いて各層の結合荷重を調整していく方法を誤差逆伝搬法（バックプロパゲーション）という[4]。これは、3 層以上の階層型ニューラルネットワークに用いられる学習法である。

k 番目の出力層ユニットと j 番目の中間層ユニットをつなぐ結合荷重 V_{kj} の修正量は以下のようになる。

$$\Delta V_{kj} = \eta \delta_k H_j$$

$$\delta_k = (t_k - O_k) O_k (1 - O_k)$$

ただし、η は学習率、H_j は j 番目の中間層ユニットの出力、t_k を k 番目の出力層ユニットの教師信号、O_k を k 番目の出力層ユニットの出力信号とする。

同様に j 番目の中間層ユニットと i 番目の入力層ユニットをつなぐ結合荷重 W_{ji} の修正量は以下のようになる。

$$\Delta W_{ji} = \eta \delta_j X_i$$

$$\delta_j = H_j (1 - H_j) \sum_{k=0}^{n} V_{kj} (t_k - O_k) O_k (1 - O_k)$$

ただし、Xi は i 番目の入力層ユニットの出力である。

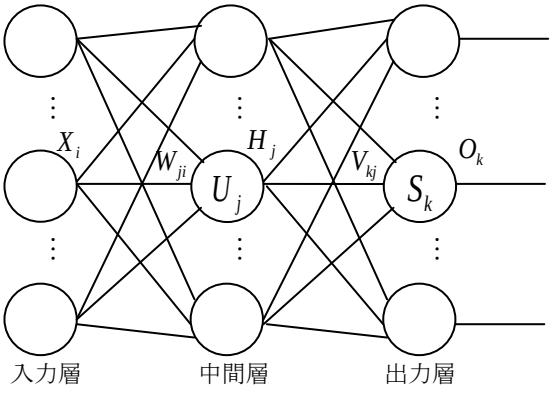


Fig.1　3 層ニューラルネットワーク

3. システム構成

本システムはニューラルネットワークによるロボットの移動制御システムと、Android 端末によるロボットの操縦支援システムで構成されている（現段階ではこれらは統合されていない）。

使用するロボットは超音波センサを搭載した MobileRobots 社製の移動ロボット pioneer 3dx[5]で、シミュレーション実験は MobileSim 上で行う。

3.1　移動制御システム

本システムには学習モードと支援モードがある。学習モードでは人間が障害物を回避するように指示した操作を教師データとして保存・学習する。支援モードでは障害物との距離によってニューラルネットワークの出力する行動と人間の指示する行動を合成して移動する。

ロボットの行動は左右の車輪速度のバランスによって決定される。車輪速度の比が[1 : 1]であれば前進、[1 : -1]であれば右回転、というように前進、停止、左小旋回、左大旋回、右小旋回、右大旋回、右回転、左回転の 8 つの行動にそれぞれ[1 : 1]、[0 : 0]、[0.6 : 1]、[0.2 : 1]、[1 : 0.6]、[1 : 0.2]、[1 : -1]、[-1 : 1]という車輪速度の比を割り当てる。それに 0~7 までの番号を付け、3 ビット符号化したものをニューラルネットワークの出力とする(Fig.2)。

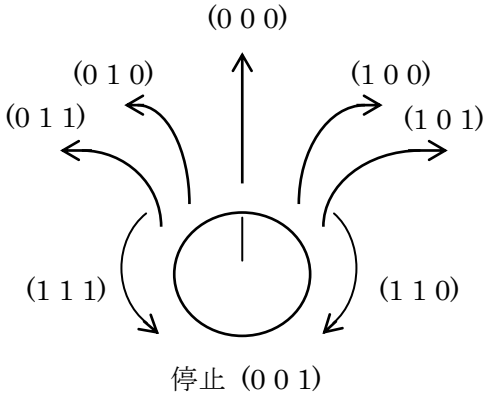


Fig.2　ロボットの行動

超音波センサの値は 0~5000mm の整数値で与えられる。障害物が近くにあるほど値は小さくなる。この値を以下の式で正規化し、壁に近くづくと 1 になり、障害物から遠ざかると 0 になるように設定した。

$$L = 1 - \frac{sensor\_value}{5000}$$

これを本システム内のセンサ値と定義し、ニューラルネットワークの入力とする(Fig.3)。

学習モード時には、人間が操作した際の入出力関係、すなわちセンサ値と行動の組み合わせを教師データとして保存する。熟練した操縦者による操作を教師データとして保存することができれば、初心者が操縦した場合でも、壁や障害物に衝突することなく操縦することができる。

学習モードから支援モードに切り替える際に、学習モードで獲得した教師データを用いて学習する。学習には、入力層ユニット数 6、中間層ユニット数 4、出力層ユニット数 3 の 3 層ニューラルネットワークを使用する。また、学習には誤差逆伝搬法を使用し、学習の高速化のため修正モーメント法を導入する。

支援モードでは、人間の操作命令を尊重しつつ、障害物との衝突を避けるような行動をとることが要求される。そのため、衝突の危険がないときは人間の命令を優先し、近くに障害物があるときはニューラルネットワークの出力を優先するように設計する。ロボットが実行する行動の車輪速度 v を以下のように設定する。

$$v = (1 - L)v_m + Lv_n$$

ただし、v_m を人間の命令した行動の車輪速度、v_n をニューラルネットワークの出力した行動の車輪速度、L を最も障害物と近いセンサの値（0~1）とする。

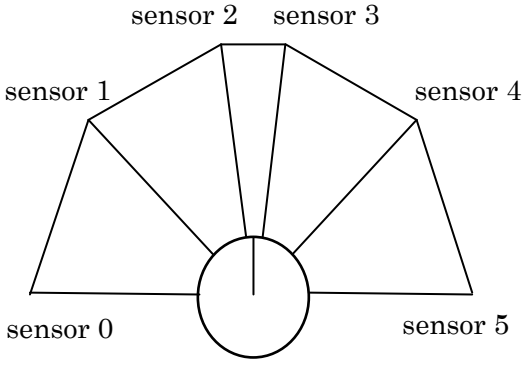


Fig.3 ロボットのセンサ

3.2 操縦支援システム

ロボットの操縦はジョイパッド等ではなく、加速度センサなどを搭載した Android 端末を使用して、より直観的に操縦可能なシステムの設計を検討した。

Android 端末の加速度センサを使用して端末の傾きを検出し、傾き方向と傾き量から移動方向と移動速度を決定する。そのデータを Android 端末からロボットを制御する PC へ無線転送し、ロボットの移動に反映させる。

また、センサ情報や速度の情報などのロボットの内部情報をロボットから Android 端末に送信し、グラフィカルに表示することで、よりユーザーに親切なインターフェースを提案する。システム構成は以下の Fig.4 に示すとおりである。

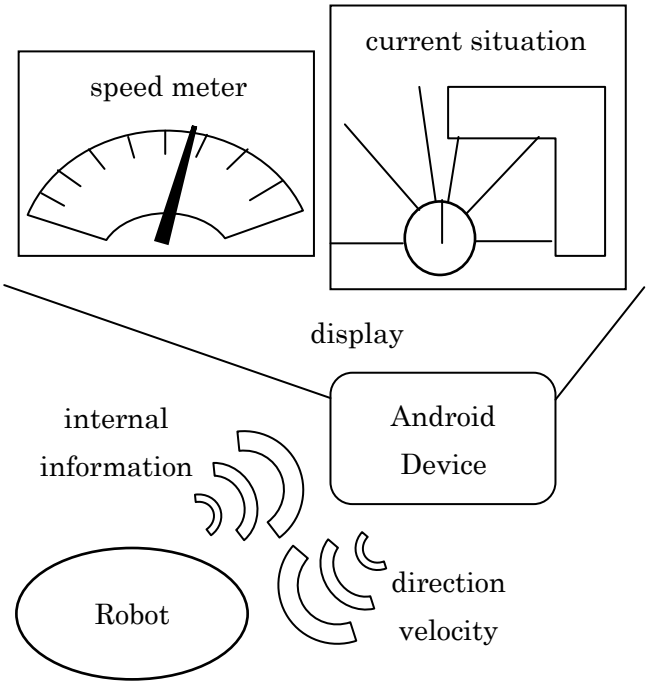


Fig.4 操縦支援システムの構成

4. 実験結果

4.1 修正法による学習速度の確認

誤差逆伝搬法による結合荷重の修正方法には、教師データ 1 パターンごとに修正していく逐次修正法と、一通りパターンを実行し終わった後にまとめて修正する一括修正法の 2 種類がある。

そこで、あらかじめ Fig.5 に示すようなルールに則って作成した教師データ 36 パターンを様々な学習法で学習させ、どの学習法が適切か調べた。結果(Fig.6)に示すように、逐次修正法と一括修正法には大きな差は見られないが、修正モーメント法を導入すると、逐次修正法の誤差収束速度は向上し、一括修正法では低下した。このことから、逐次修正法に修正モーメント法を適用したものが最も適切であることが確認できた。

また、この学習で得られたパラメータを用いて、シミュレータ上でロボットの移動を支援した。人間の命令を前進のみに固定したとき、障害物を回避して移動できるかどうかを確認する。Fig.7 に示すように、障害物を回避して進むことができた。

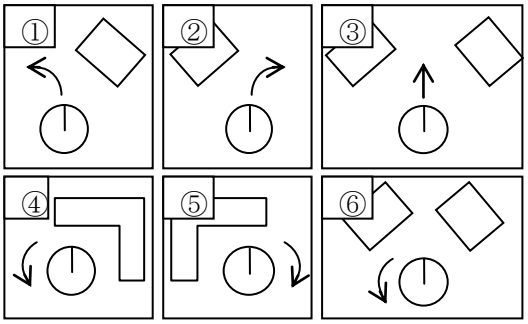
4.2 教師データの取得と学習

あらかじめ用意された教師データの学習および学習した機能による操作支援が可能であることが確認できた。次に、あらかじめ用意された教師データではなく、人間の操作から取得した動作を教師データとする方法について実験する。

まず、どのような状態を教師データとして保存するかを設定する。

壁から遠い場合、ニューラルネットワークの出力する行動よりも、人間の命令する行動を優先するため、壁から遠い状態、すなわち最も障害物に近いセンサ値が 0.3 未満の状態は教師データとして保存しない。また、データ量の爆発的な増加や、教師データの矛盾を防止するため、過去に取得した教師データと類似した状態、すなわち各センサ値の差異が 0.1 未満の状態は、取得済みの教師データと重複しているとみなし保存しない。

センサの値が 0.3 以上で、かつ、取得済みの教師データと重複していない状態で、その時のセンサ値と実行された行動を教師データとして保存する。



	I_1	I_2	I_3	I_4	I_5	I_6	O_1	O_2	O_3
①	0	0	0	0.5	0.5	0	0	1	0
②	0	0.5	0.5	0	0	0	1	0	0
③	0	0.5	0	0	0.5	0	0	0	0
④	0	0	0.8	0.8	0.5	0.5	1	1	1
⑤	0.5	0.5	0.8	0.8	0	0	1	1	0
⑥	0	0.5	0.5	0.5	0.5	0	1	1	1

Fig.5 教師データ例

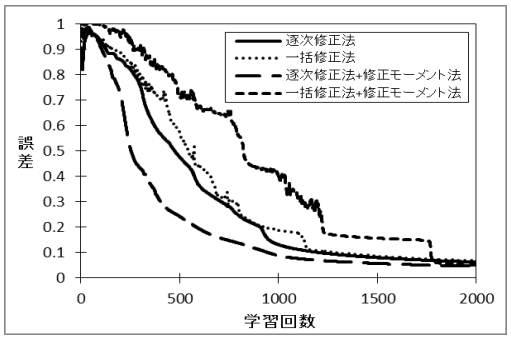


Fig.6 各修正法による学習速度

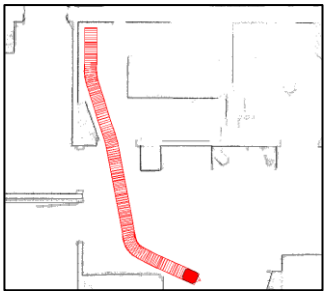


Fig.7 実行結果

上記の条件のもと、Fig.8 に示す環境における操作を教師データとして学習させ Fig.9、Fig.10 に示す環境で人間の命令を前進のみに固定したとき、障害物を回避して移動することができるかどうか確認する。Fig.9 に示すように、教師データを取得した環境と同じ環境においては障害物を回避して移動することができた。しかし、Fig.10 に示すように、他の環境においては障害物を回避することができなかった。これは、単純な環境で教師データを取得した場合、教師データの多様性を確保できず、汎用性のあるルールを生成できなかったためだと考えられる。

次に、Fig.11 に示す環境における操作を教師データとして学習させ、Fig.12、Fig.13 に示す環境で人間の命令を前進のみに固定したとき、障害物を回避して移動できるかどうかを確認する。Fig.12 に示す環境においては、障害物を回避し、開けている空間へ進むことが確認できた。しかし、Fig.13 に示す環境においては、自ら壁に向かうことや、迂回するための十分な空間が確保できず壁に衝突してしまうことがあった。これは、まだ、教師データの数が少ないため完全な学習ができず、また、取得した教師データの中で矛盾が生じているからだと考えられる。

5. まとめ

人間が操作した際のセンサ値と行動を教師データとして保存し、ニューラルネットワークを用いて障害物を回避する方法の有効性をシミュレータ上で確認することができた。大部分の環境では障害物を回避することができたが、一部の環境では教師データに矛盾が生じてしまうことがあった。

今後の展望として、Android 端末でロボットを遠隔操作するシステムを本システムに組み込むこと、教師データの矛盾などを解消するために Android 端末上で教師データをグラフィカルに表示し、教師データの矛盾を人間の手で修正できる機能を実装すること、移動障害物を回避することができるかどうかを確認することなどが考えられる。

参考文献

1) 柴田 克成、岡部 洋一、伊藤 宏司 : ニューラルネットワークを用いた Direct-Vision-Based 強化学習－センサからモータまで－, 計測自動制御学会論文集, **37**, 2（2001）

2)熊沢 逸夫 : "学習とニューラルネットワーク", 森北出版(1998)

3) 中野 馨, 飯沼 一元,ニューロンネットグループ, 桐谷 滋: "ニューロコンピュータ", 技術評論社(1989)

4) 平野 廣美 : “C++と Java でつくるニューラルネットワーク”, パーソナルメディア株式会社(2008)

5) http://www.mobilerobots.com/

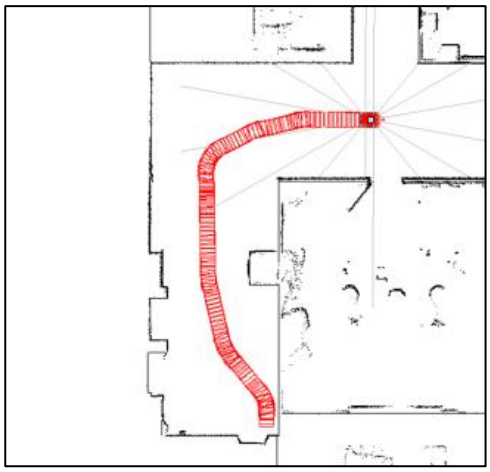


Fig.8 単純な環境で取得した教師データ

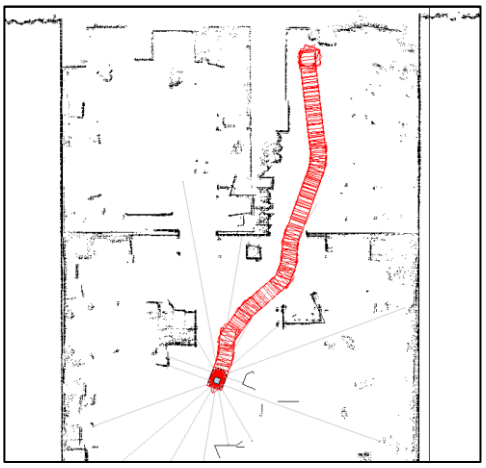


Fig.11 複雑な環境で取得した教師データ

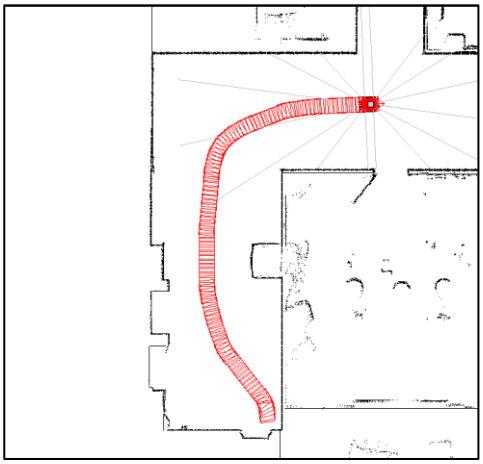
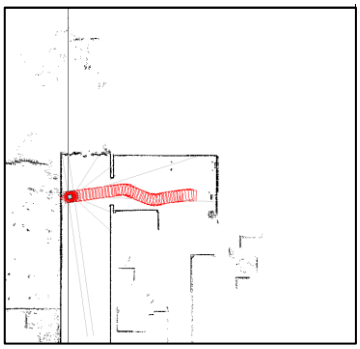


Fig.9 成功例



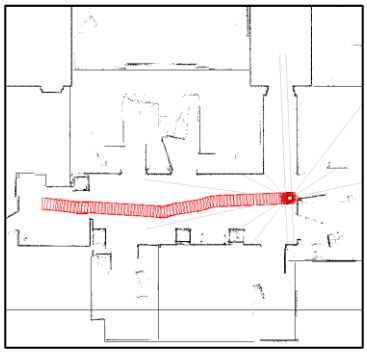


Fig.10 失敗例

Fig.12 成功例

Fig.13 失敗例