

将棋AIロボを創ろう

システム科学技術学部 機械知能システム学科
2年 松井 悠馬 田中 晴隆 鷹島 大拓
土田 郁哉 池之内 俊弥
電子情報システム学科 2年 越後 大樹
経営システム工学科 2年 中川 真輝
指導教員 機械知能システム学科
准教授 間所 洋和 教授 佐藤 和人
電子情報システム学科 助教 寺田 祐樹

1. 背景・目的

コンピュータソフトとプロ棋士が将棋の対決をする「将棋電王戦」に用いられた「電王手くん」は、コンピュータソフトの代指しを行う6軸垂直多関節アームロボットである[1]。ただし、電王手くんは、一説には開発費用3000万円ともいわれ、コストの高いロボットである。本研究では、LEGO MINDSTORMSを用いて代指しロボットのボディを製作し、指し手生成ソフトウェアを自作することで、低コストで臨場感の高い将棋対戦ロボットの開発を目的とする。

2. 機体の製作と制御

製作した機構は、駒を横方向に移動する土台部と駒を縦方向に移動するアーム部から構成される。機構は、LEGO MINDSTORMS の教育版 EV3 を用いて制御した。

2.1. 駒の横方向移動：土台部

図1に駒の横方移動の機構を示す。機体は、大小2枚の歯車を組み合わせ、タイヤの回転運動に変換し、横移動する。2枚の歯車の直径の比は21:5でタイヤは前者の歯車につながっているため、タイヤの回転数はモータの回転数の0.238倍となる。

モータを350°回転させたところ、タイヤは83.3°回転した。タイヤの半径は28mmであるため、1マス分の移動距離の理論値は、40.6mmとなる。しかし、移動距離の実測値は31.7mmであったため、理論値よりも8.9mm不足した。これは、機構の重さによってモータが入力した値よりも回転していないことが考えられる。解決策としては、部品点数を減らし、アームを軽量化することが考えられる。また、土台部を改良し、アームの安定化させたが、その分重さが増した。この誤差を埋めるための解決策として誤差の分を含めてモータの回転数を増やす、ギア比を変更する等が考えられる。

2.2. 駒の縦方向移動：アーム

図2は初期位置の状態を示している。この状態は、車体の梁が将棋盤の端から140mmの地点としており、モータBで90.0°、モータCで126.0°、モータDで15.0°となる。アームは、駒を手前と奥、上下に移動させるための部位である。所定の場所から動作を始め、その場所に戻ってくる位置を初期位置と定めた。

図3に示すアームにおいて、ハンドから順にモータA、モータB、モータC、モータDと定めた。なお、全ての節は、21:5の直径を持つ2つの歯車から構成される。

ハンドを将棋盤の縦5マス目に移動させるプログラムを作成したところ、所定の位置まで移動し、把持できた。このプログラムは、まず、モータDを常に回転させ、段階的に重心を後方に保つ必要がある。1秒後モータBを380°回転させ、その節の回転角は90.5°になる。その1秒後モータCを130°回転させ、その節が31.0°回転すると、把持したい駒に到達する。このとき、モータDからの水平到達距離の理論値は、294mmになった。実測値では、265mmとなり、理論値よりも29mm不足した。これは、アームにかかる重力や慣性力によって誤差が生じたためと考えられる。このときの到達地点を図3に示す。

同様に初期位置にハンドを戻す場合の角度の理論値、実測値と誤差を表1に示す。表1より、モータBとCは、モータDと比較して、理論値と実測値の差が大きいことから、重力や慣性力が働きやすいため、誤差が生じたと考えられる。解決策として、伸縮時のモータのパワーを下げる事が考えられる。

表1 モータの理論値、実測値と誤差

モータ	理論値	実測値	誤差
B	90.0°	67.1°	22.9°
C	126.0°	153.1°	27.1°
D	15.0°	19.0°	0.04°

2.3. 駒の把持：ハンド

改良前後のモータを図4、改良後のハンドを図5に示す。ハンドは駒を把持する部位で、回転運動をリンクと二つの軸を介して運動をする機構である。当初、作成したハンドでは、横回転を縦回転に変換する際に歯車同士の摩擦が大きく、歯車が擦り減ってしまった。このため、図4に示すようにMモータをLモータに変えて縦回転で制御させ、図5に示すハンドに改良した。改良後は歯車の回転が滑らかになり、制御の安定化に成功した。

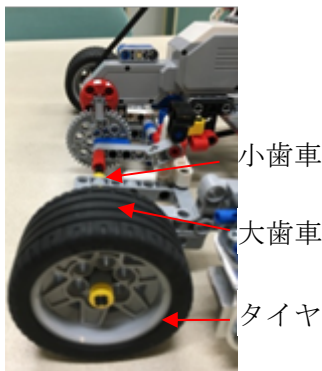


図1 横移動の機構

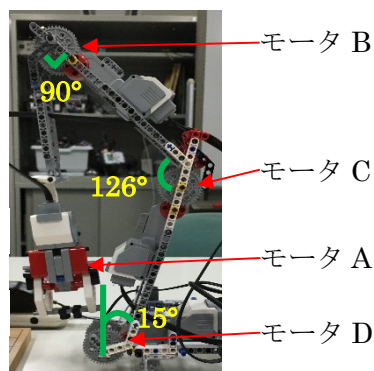


図2 初期位置

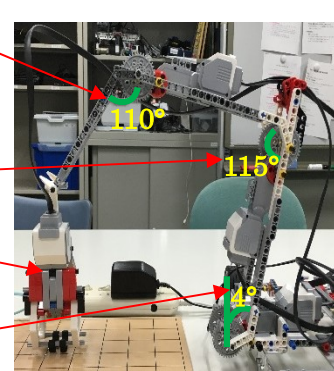


図3 到達地点

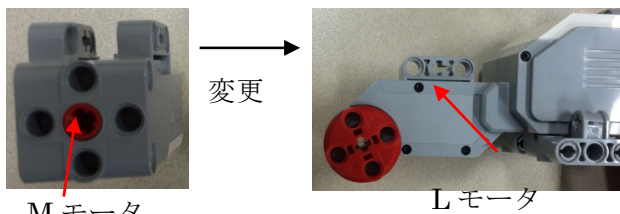


図4 改良前後のモータ

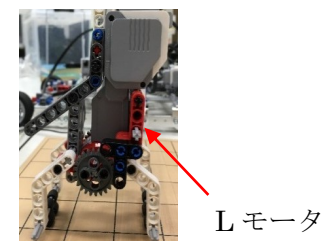


図5 改良後のハンド

3. 駒の検出

本研究では、各駒が盤面のどの位置にあるのかを検出するためにテンプレートマッチング法を利用した。入力画像を将棋の盤面、テンプレート画像を各駒とし、入力画像内でテンプレート画像を移動させて相関値を算出することで駒を検出した。プログラムについては、テンプレート画像に番号を付け、0番から順番に探索させる繰り返し文を作成した。設定した相関値のしきい値以上の画像を最大で9枚探し、設定した相関値のしきい値を下回る場合は処理を終了させた。画像の撮影には将棋盤の真上に設置したUSBカメラを用いた。

図6に示す盤面を入力画像とし、テンプレート画像である各駒を見つけた場合は図7、8のように赤で塗りつぶし、対応する色で枠を付け表示させた。相関値のしきい値を0.7として探索したところ、図7に示すように、金将と香車が検出されなかった。金将が検出できなかった理由として、金将のテンプレート画像が入力画像中の金将の傾きに対応できていなかったことが考えられたため、傾きが3°と5°のテンプレート画像を作成した。また、香車が検出されなかった理由としては、将棋盤の端に位置するためカメラのピントのズレや光の影響を受けたことが考えられたので、香車を将棋盤の端に置いてテンプレート画像を作成した。以上のテンプレート画像を追加したところ、図8に示すように、入力画像中の駒をすべて検出できた。このようにテンプレート画像を追加していくことで、用意した17枚の入力画像において駒をすべて検出できた。なお、相関値は0.7より高くするとテンプレート画像の数を増やす必要があり、処理速度も遅くなってしまうため0.7に設定した。



図6 対局面探索前

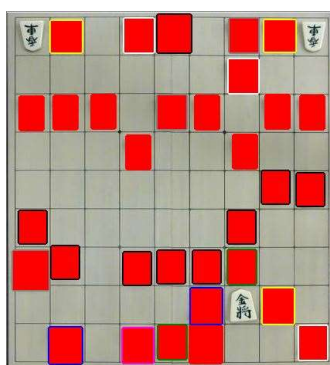


図7 対局面探索後(追加前)

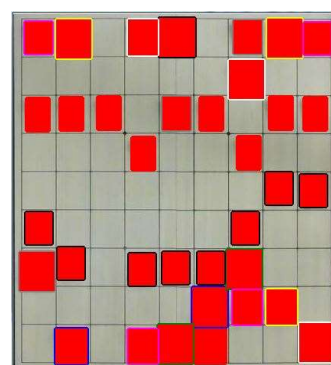


図8 対局面探索後(追加後)

4. 将棋対戦ソフトの製作と思考エンジンの調査

人と人工知能が対戦出来るように思考エンジンをC++言語で制作した。

表2のように将棋の駒を点数化し、点数が20000点を超えた方が勝ちという点数制のプログラムを作成した。「二歩」「千日手」などの細かいルールに関してはif文、for文を用いて条件を指定し、駒の打てる範囲を制限することで再現した。

手の探索や打ち手の評価の方法として、「全幅探索法」と「枝刈り法」を検証した。全幅探索法は、全範囲を検索するため、人間の思考以上の手を見つけることができるが、将棋の盤面では10の220乗の指し手があるため、検索しきれなかった。そこで、「枝刈り法」を用いた。枝刈り法は、探索価値の低い手を切り捨て高い手だけを探索することで、計算負荷

を減らすことができるため、枝刈り法を採用した。

なお、序盤においては相手の駒が取れずに点数の変動はないため、プログラムとして点数による評価ができない。このような時に、予め設定された陣形（「振り飛車」、「居飛車」などの序盤における定跡）に持ち込む評価点数に係わらないプログラムを作成した。

表2 駒の点数

駒	点数	点数(成り後)
歩	100	900
香	501	1002
桂	601	1003
銀	803	904
金	904	0
角	1305	1556
飛	1506	1707
王	19999	0

5. 考察

参照機の製作により現実的に将棋を指すロボットにするためには、ロボットアームの機構は、モータがアームの重さに負けないようなモータ制御のできるプログラムを作製する必要があると考えられる。

テンプレートマッチングでは、探索する駒の傾きに応じたテンプレート画像をその都度作成する必要があった。これを改善するために、探索画像の角度が変化しても読み取ることができるように特徴点を検出し、トラッキングするようにしたい。

思考エンジンにおいて枝刈り法では三手先までしか読めなく、価値のある指し手が見落とされている可能性があるため、四手、五手先まで思考できるようにしたい。

6. 結論

- ・モータの回転運動を前後左右へ移動できる四輪ロボットアームを製作し、制御した。
- ・カメラ画像からテンプレートマッチングを用いて、盤上の将棋駒の文字を検出した。
- ・プログラミング言語を理解し、将棋対戦可能なソフトウェアを作成した。

7. 参考文献

- [1]"デンソー、「将棋電王戦 FINAL」に協賛～コンピュータ代指しロボットを提供～",
株式会社デンソー,2015/02/10閲覧
<http://www.denso.co.jp/ja/news/topics/2014/141126-01.html>
- [2]うさびよん OfficialHomePage,2015/02/27 閲覧
<http://homepage1.nifty.com/Ike/usapyon/>