

## 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
1年	松井 悠馬
1年	池之内 俊弥
1年	鷹島 大拓
1年	土田 郁哉
1年	田中 晴隆
システム科学技術学部	電子情報システム学科
1年	坂本 圭佑
1年	越後 大樹
システム科学技術学部	経営システム工学科
1年	中川 真輝
指導教員	システム科学技術学部 機械知能システム学科
	准教授 間所 洋和
	准教授 佐藤 和人
システム科学技術学部	電子情報システム学科
	助教 寺田 祐樹

### 1. 背景

2012年1月、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」が開催された。数多の選択肢から最良手を探索するコンピュータに対し、プロ棋士は、定跡を読まれないように奇策を採用したものの、コンピュータが勝利するという結果となった。なお「将棋電王戦」は、今年度を含め3回開催されている。対戦相手となるプロ棋士は、6割5分以上の勝率をもつ実力者が集められているが、2013年度第2回の五番勝負では3勝1敗1分、2014年度第3回の五番勝負では4勝1敗となり、いずれも人工知能を搭載したコンピュータ側が勝ち越している。

### 2. 目的

「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]。「電王手くん」が導入された理由は、「人間対コンピュータの構図が分かりにくく、絵にならない」との要望が主催者側から出たためである。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。

### 3. 使用機材

#### 3.1. LEGO MINDSTORMS Education EV3

LEGO MINDSTORMS Education EV3 (以下、EV3と略記する) は、教育用のレゴブロックである。マイクロプロセッサが搭載されているEV3ブロックにプログラミングのデー

タを取り込ませることで、各種センサからの情報取得や、モータ制御を行うことが可能である。本研究では、ロボットアームのフレーム、アームのモータ制御に使用した。

### 3.2. Logicool® HD Webcam c270

Logicool® HD Webcam c270 (以下、Webcamと略記する) は、最大画素1280×720ピクセル、USB2.0接続のWEBカメラである。本研究では、将棋駒の検出に用いた。

## 4. 製作

### 4.1. ロボットアームの製作

#### 4.1.1. ロボットアームの構成

ロボットアームは、図1～3に示すボディ、アーム、ハンドから構成される。図1に示すボディには、EV3とロボットアームを左右に回転させるDCモータが1個内蔵されている。図2に示すアームは、モータの回転運動を組み合わせたギアにより上下運動に変換する。図3に示すハンドは、駒を把持するために使用する。

#### 4.1.2. 製作したロボットアーム

製作したロボットアームを図4～6に示す。図4に初号機を示す。問題点は、手首が前後に動かない、アーム本体の重さに耐えられない、コードが多いため絡まってしまう、前後の運動ができない(手前に届かない)、アームの長さが足りない、全体的に強度が足りない、重心のバランスからモータの付ける位置を考えるという点があった。

図5に初号機を改善した弐号機を示す。弐号機では、アームの関節を2個から3個に増やし、盤面の奥にも届くようにした。また、ボディを16cm高くすることでアームを手前の位置に届くことが可能になった。ただし、将棋盤の両斜め奥が届かない、コードの長さが短く、EV3とモータの接続ができないという問題点があった。

図6に弐号機を改善した参号機を示す。参号機では、50cmコードから90cmのコードに変更し、モータとの接続を可能にした。またボディにタイヤを取り付け、横方向へ並行移動でさせることで、盤面両斜め奥にもアームが届くようにした。また、強度を確保するためにTETRIXのアルミフレームを使用し、アーム本体の重さにも耐えられるようになった。

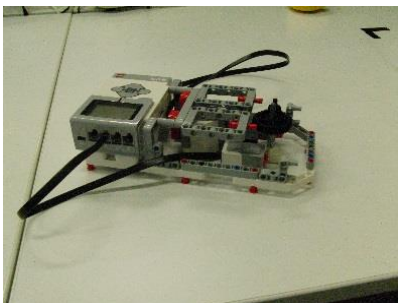


図1 ボディ



図2 アーム



図3 ハンド

#### 4.2. 環境構築

環境構築は、開発ホストであるPCにUbuntu Linux環境を構築し、EV3ファームウェアおよびプログラムをビルドした。また、MicroSDHCカードに格納し、C言語でプログラミングしたソフトウェアをEV3で動作させた。

#### 4.3. 将棋駒の検出

対局状況を思考エンジンに伝えるため、盤面上の駒の種類を認識し、存在する位置を把握させる必要がある。駒の検出には、USBカメラから取得した画像をグレースケールに変換し、文字・盤面のマスを二値化する処理を用いた。ただし、部屋の照明条件により、二値化の閾値を変更しなければ、検出が困難であるという問題点があった。本研究では、照明変化に頑強なHSV色空間を用いて文字輪郭を抽出した。図7に、Webcamで撮影した駒と将棋の駒を文字検出結果を示す。

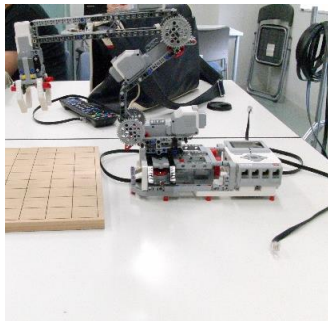


図4 初号機

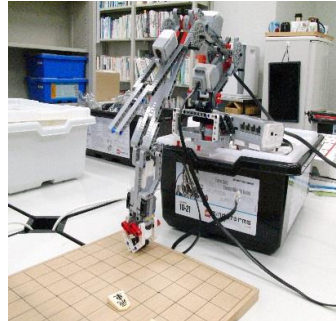


図5 弐号機

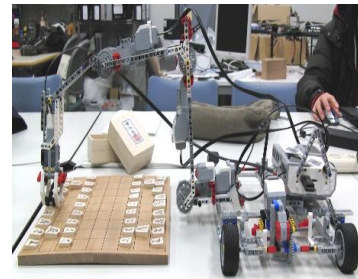


図6 参考機

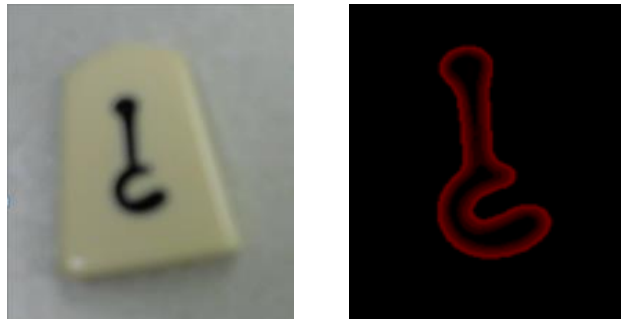


図7 文字検出(左：原画像，右：文字検出画像)

#### 4.4. 将棋対戦ソフトの製作と思考エンジンの調査

「うさびよん Official Home Page」を参考に、プログラミング言語にC++、開発環境にMicrosoft Visual C++を用いて人対人で将棋対戦可能なプログラムを製作した[2]。

まず、盤上の表示、駒の移動、そして将棋ルールなどのプログラムの作製をした。盤上の表示については配列などを用い、その座標に対応する駒画像を割り当てた。図8は盤上の表示の画像である。駒の移動については、ポインタを用いて配列の中身を書き換えるようにした。図9は駒を移動させた盤上の表示である。将棋ルールについては、条件文を用いて駒の打てる範囲を制限することで再現した。

思考エンジンが将棋を打つためのプログラムの作製として、手の探索や打ち手の評価の

方法について調査した。手の探索手法は、「枝刈り法」と「全幅探索法」がある。「枝刈り法」は、探索価値の低い手を切り捨て、探索価値の高い手だけをより深く探索する手法である。「全幅探索法」は、すべての手を探索する方式である。二つの違いとしては、「枝刈り法」は「全幅探索法」に比べて一手に対してより深い読みができるが、「全幅探索法」では浅い読みでは価値の見出せない手の中からも良い手を見つけ出すことができる。ただし「全幅探索法」のほうが処理に時間がかかる。なお、将棋の探索空間は、宇宙の粒子数よりも多い10の220乗の指し手があり、計算機の性能が向上しても、全幅探索法では、最善手を探索しきれない。



図8 盤上の表示



図9 駒の移動後の盤上の表示

## 5. 考察

参号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。

思考エンジンの指し手の探索法は「全幅探索法」が良いと考えられる。これは、人間では価値の見出せない手にも評価をつけ、対戦相手が予測できない打ち筋が生成できるためである。ただし処理する計算量が多いため、探索する局面の分岐数を人間の基準で評価を設定すべきか、あるいは機械学習によって設定すべきかを実験によって判断するべきであると考えられる。

## 6. 結論

- ・モータの回転運動を前後へ移動できる四輪ロボットアームを製作することができた。
- ・EV3のプログラミング環境を構築し、カメラ画像から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/>