

# ルービックキューブを完成させるロボットの性能改善

2016SC036 粥川優樹 2016SC040 近藤真紀

指導教員：大石泰章

## 1 はじめに

製造の現場では、コストの制約から高価な機材を用いることができない場合がたびたび存在する．本研究では安価な機材を用いて製作したロボットを様々な機能の改善によって性能を向上させることを目標とする．具体的には LEGO Mindstorms を使用してルービックキューブを6面揃えるロボットを製作し、ハードウェアの改善や、ソフトウェアの工夫、追加によって機構上の問題や、センサ性能の限界、ソフトウェアの不備などの問題を克服し、ロボットの性能を向上させる．

## 2 使用機材の概要

### 2.1 ルービックキューブ

本研究で使用するルービックキューブは26個の立方体を表面に貼りつけた形状をしており、この立方体をブロックという．ブロックの各面をブロック面といい、各ブロック面は単色で塗られている．ルービックキューブの1面はブロック面9面からなる．ブロック面は6色を使用して塗り分けられており、ブロックを回転させることでブロックを移動させ、ルービックキューブの各面を同じ色で揃え、6面それぞれが同じ色で揃ったとき、ルービックキューブが完成したと言う．本研究では、2つのルービックキューブを使用した．配色の違いから、以後、日本配色キューブ、世界配色キューブと呼ぶ [1]．図1と図2にそれぞれの写真を示す．

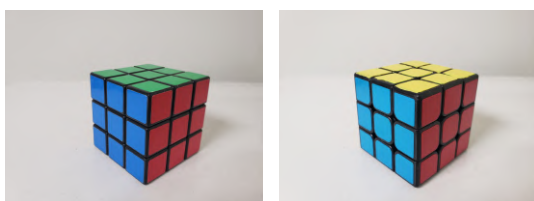


図1: 日本配色キューブ 図2: 世界配色キューブ

### 2.2 ロボットの基本構造

ロボットの製作では、材料として LEGO Mindstorms を使用し、基本構造として Gilday が提供する組立図を参考にした [2]．図3が、組立図通りに製作したロボットである．このロボットはブロックの存在を判定する超音波センサ (図4)、ブロックの色を識別するカラーセンサ (図5)、キューブそのものを動かすアーム (図6)、キューブを保持する台座 (図8) からなる．

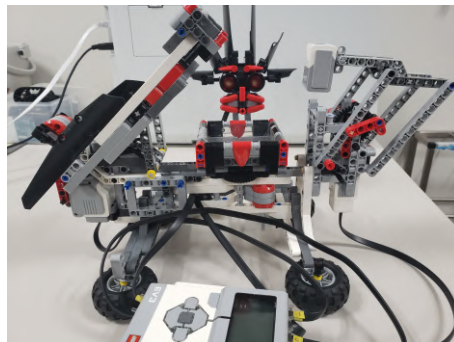
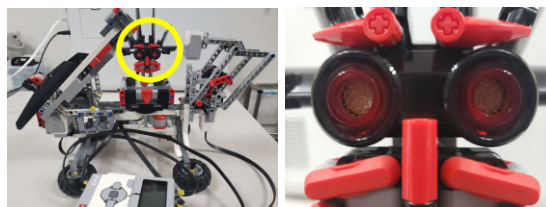


図3: 基本構造

### 2.3 超音波センサ

超音波センサは、正面にある物体までの距離を測定することができる．これは、音波を送信し、その音が反射してセンサに戻ってくるのにかかる時間を測定することで実現される．本研究では、距離を測定するためではなくキューブの存在の有無を判定するために使用した．超音波センサは音をよく反射する硬い表面の物体を最も良く検知するため、キューブの存在の有無を判別するのに適したセンサといえる．



(a) 全体図

(b) 拡大図

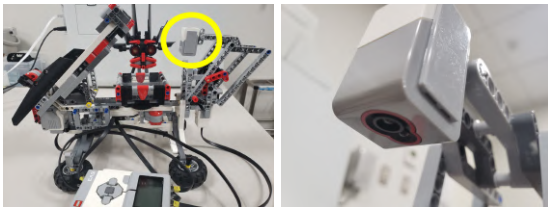
図4: 超音波センサ

### 2.4 カラーセンサ

カラーセンサは、色の判別、反射光の強さ、周辺の光の強さを測定することができる．本研究では、主に色の判別をするために使用した．このカラーセンサは黒、青、緑、黄、赤、白、茶の異なる7色の色を検知することができる．

### 2.5 アーム

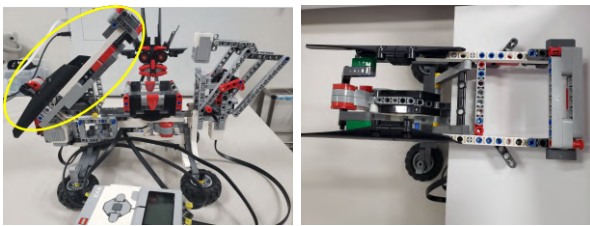
図6(a)のアームは1つのモータを使用して動作を行う．図6(b)の右側の四角い穴の開いた部分が、モータの回転によって、台座に置かれたキューブにかぶさり、キューブを固定する (図7(a))．かぶさった状態でアームが後ろに下がると、キューブの上部がアームに引っ張られ、下部が台座に引っかかる．それにより図7(b)のようにキューブ



(a) 全体図 (b) 拡大図

図 5: カラーセンサ

が傾く．その状態から，アームを前に進めることで，傾いていたキューブがアームに押され，キューブが縦方向に回転する (図 7(c))．



(a) 全体図 (b) 拡大図

図 6: アーム

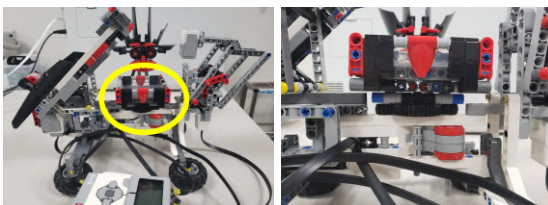


(a) (b) (c)

図 7: 縦方向の回転

## 2.6 台座

キューブを設置するため台座を用意する．台座にはモータが取り付けられており，台座が回転することでキューブを横方向に回転させることができる．この動作を以後，キューブの横方向の回転と呼ぶ．

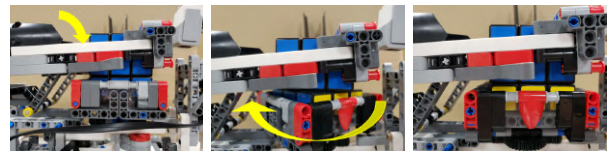


(a) 全体図 (b) 拡大図

図 8: 台座

図 9(a) のようにブロックの上段 2 段をアームで固定する．その状態で図 9(b) にあるように台座を動かすと，最

下段だけが回転し，図 9(c) のようにブロックの色の配列を変えることができる．このような動作を以後，ブロックの横回転と呼ぶ．



(a) (b) (c)

図 9: ブロックの横回転

## 2.7 ロボットの動作

ブロックを台座に設置し，ロボットがキューブの存在を認識すると，カラーセンサがキューブ上面のブロックの色をスキャンする．台座を回転させながらスキャンを行い，各ブロック面の位置と色を記録する．上面 1 面のスキャンが終了すると，アームを使用してキューブを縦方向に 90 度回転させて上面を変える．これを繰り返して 6 面全てのスキャンが終了したところで，6 面の色を揃えるための手順を算出する．手順を算出後，アームを用いたキューブの縦方向の回転，台座を用いたキューブの横方向の回転，アームと台座を両方用いたブロックの横方向の回転，これらの動作を適切な順序で実行することにより，6 面全てのブロックの色を揃えることができる．

## 3 ロボットの性能改善

### 3.1 実験結果

製作したロボットを世界配色のキューブを用いて動作させたところ，カラーセンサによる色のスキャン以降の動作に進むことはなかった．そのため，キューブを完成させることは一度もなかった．また，キューブを縦方向に回転させる動作だけの成功率は 85% であった．日本配色のキューブを用いた動作では，縦方向に回転させる動作の成功率は 85% で世界配色と変わらなかったが，カラーセンサが色を認識する動作の成功率は 80% であった．しかし，日本配色のキューブは回転がしにくらしく，世界配色のキューブでは問題にならなかったブロックの横回転の成功率が 53% と低かった．これらの成功率は単体では高いように見えるが，キューブが完成するまでにこれらの動作はそれぞれ複数回繰り返されるので，実際に日本配色のキューブが完成する比率は 10% 程度になってしまう．

### 3.2 問題点と改善

問題は大きく分けて，機械的な問題，カラーセンサの問題，ソフトウェアの問題の 3 つである．以下これらについて詳述し，改善策を考える．

機械的な問題とは，キューブが転がらない，台座から落ちる，アームでキューブを固定できない，のような失敗が生じるというものである．このような失敗が生じるとキューブ

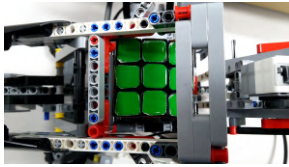


図 10: ぶつかるアーム



図 11: 台座の内側の厚み

ブを縦方向に回転させることができない．このロボットには、失敗を検知する機能が搭載されていないため、こうした失敗が手順の間に一度でも発生すると、キューブが完成することはない．この問題を解決するため2つのことを考えた．まず、台座でのキューブの位置が端に寄りすぎると、図 10 のようにアームが下がってきた際にキューブの端にアームが当たり、弾いてしまうことがある．これを防ぐため、図 11 のように黄色いテープで台座の内側の四方に厚みを作り、キューブが端に寄らないようにした．このとき、四方の厚みが大きすぎてルービックキューブが台座の中で固定されると、キューブを縦方向に回転させる際に引っかかり、台座の中にキューブが戻らないことがあった．そのため、キューブが動ける余裕を確保しつつ、厚みを作成した．次に、アームのパワーが低すぎるようで、キューブが回転しないことがあった．その改善のため、プログラム内でアームのモータの出力値の数値を変更し、アームのパワーを上げた．

カラーセンサの問題とは、カラーセンサが色を正しく認識しないというものである．これには2つの原因がある．



(a) 白，黄，橙



(b) 青，赤，緑

図 12: 日本配色のキューブ

1つめはカラーセンサの特性上の問題である．本研究で使用したカラーセンサは、正確に検知するために、センサと物体を接触しない程度でかなり近づけなければならない．そこで、先に反射光の強さを測定することで、正確に色が検知できる距離までセンサが近づいていることを確認してから色の識別を行うようにプログラムを作製した．2つめは、キューブの日本配色と世界配色が異なることである．図 12(a)，(b) の日本配色キューブの黄、緑、橙と図 13(a) の世界配色キューブを比較すると、ブロックの色が異なる



(a) 世界配色キューブ (緑，橙，黄) (b) シールを貼ったキューブ (緑，橙，黄)

図 13: 世界配色を日本配色に近づけた

ことがわかる．用いたカラーセンサは日本配色キューブの色を認識することはできたが、世界配色キューブの色は認識しなかった．このことの改善のために図 13(b) のようにテープを用いて、世界配色キューブの色を日本配色キューブの色に近づけた．

ソフトウェアに関する問題とは、手順を算出した後にキューブを動かすことに失敗しても、手順を最後まで実行したら、キューブは完成したとみなされてしまうということである．これを防ぐため、最後にカラーセンサでキューブ全面のブロックの色を読み取り、完成したか確認するプログラムを追加した．このプログラムはキューブの色を読み取り、完成していれば終了する．完成していなければ、もう一度手順を計算しキューブを揃えるというものである．プログラムの追加には LEGO Mindstorms EV3 ヘルプ [3] を参考にした．最後に、大きな問題とはなっていないが、動作中のロボットの動きが壊れやすそうな印象を受けたことと、実際にパーツが緩む場所があったことがある．改善のため、いくつかパーツを加えて補強を行った．

### 3.3 改善結果

前節の改善を行った結果、世界配色キューブでは、縦方向に回転させる際の失敗は減少し、成功する比率は 95% となった．カラーセンサが色を認識する比率も向上し、50% となった．これらの改善に加え、完成を確認するプログラムを追加したことにより、完成する比率は約 43% (60 回動作中 26 回完成) になり、大幅な改善となった．色の認識については、認識する比率が向上したが、認識しないことも多く、より良い結果を得るために、色の調整を行う必要がある．シールを用いているので、シールが削れ、色も削れてしまう問題も確認された．

日本配色のキューブでは、縦方向に回転させる動作の成功率は世界配色と同じく 95% まで向上し、カラーセンサが色を認識する比率は 97% に向上した．その結果、完成率は約 48% (60 回動作中 29 回完成) となった．ロボットの耐久性についても補強を行った結果、パーツが緩むことはなくなり、連続した稼働にも耐えられるようになった．

## 4 失敗の検知と修正機能の付加

### 4.1 ブロックのずれとその影響

前章で行った改善により、日本配色の場合も、世界配色の場合もともにキューブの完成する比率は向上した．しか



図 14: 横回転によって生じたずれ

し、ブロックの横回転の失敗は依然として解決されていない。前章の改善を行う前は、図 14 のようにずれが生じた状態でブロックを横回転させようとすると、キューブが台座から外れる、ブロックが横回転しきれない等の失敗が発生することが問題であった。しかし、改善後はブロックが転んで動かなくなり、ロボットが動作不能に陥り、人の手の介入や、ロボットの再稼働を必要とするようになった。この問題の原因は、ブロックを横回転させたとき、ずれが生じることにある。1 度の回転で生じるずれは小さいが、ずれが積み重なり、ブロックを横回転させづらくなる。加えて、前章での改善による動作の成功率の向上によって、多くの動作を実行するので、改善前よりも多くのずれが積み重なり、大きなずれを生じることとなった。その結果として、動作不能のような大きな問題へとつながった。

#### 4.2 大きなずれの検知と修正

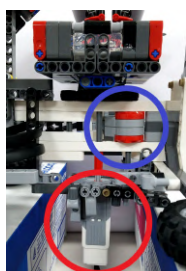


図 15: 2つのモータ

前節の問題に対処するために失敗を検知し、動作を変更する機能を付加することを考えた。この機能は図 15 にある 2 つのモータを用いて実装される。図 15 の中央部の青丸で囲まれたモータは、計算された手順に基づいて台座を指定した角度回転させるためのものである。中央下部の赤丸で囲まれたモータは台座と直接繋がれている。これは台座を回転させるために用いるのではなく、内蔵された回転センサによって、台座が回転した角度を計測するために用いる。指定した角度と計測した角度に大きな差が存在するならば、台座がうまく回転できておらず、ブロックの横回転がうまくできていない、ということを表している。以上のアイデアに基づいてプログラムを作成し、ブロックの横回転の失敗を検知する機能を実現した。加えて、失敗を検知したとき、計測した角度だけ、台座を回した方向と逆方向に回転させ、台座を元の位置に戻し、再び色のスキャン

を行って手順を再スタートするプログラムを付加した。

#### 4.3 結果

前節の機能を付加することによって、大きなずれを生じ、動作不能に陥っても、人の手の介入や、ロボットの再稼働を必要とせずキューブを完成へと進めていくことが可能となった。ここまでの改善によって、日本配色では、完成率は 90%(30 回動作中 27 回完成) となった。世界配色では、完成率は約 93%(30 回動作中 28 回完成) となった。しかし、キューブの縦方向の回転等、それぞれの動作で失敗が発生するたびに色のスキャンを行って再スタートするので、キューブの完成までにスキャンと再スタートを 5 回以上繰り返すなど、長い時間を必要とする場合がある。

#### 5 おわりに

本研究では、ロボットの性能を改善するとともに、失敗を検知する機能の付加を行い、6 面揃えることの成功確率をあげることができた。今後の課題としては、3 つのことが考えられる。1 つめが、失敗を検知した後の修正機能の改善である。今回は、失敗を検知した後、再スキャンを行い手順を再スタートすることにした。もし、再スキャンと再スタートを行わず、失敗の前の手順に戻り、そこからの動作を行うプログラムにできれば、より少ない動作で完成させることができる。2 つめが、より高精度でのずれの検知と修正である。ずれが生じない、あるいは小さいずれのまま動作を進めることができれば、カラーセンサの問題、キューブの縦方向の回転の問題、ブロックの横回転の問題、これらすべての問題での失敗を減らすことが可能となる。3 つめが、より少ない手数でキューブを完成させるプログラムに変更することである。ルービックキューブの最短手は 20 手以内と証明されており [4, 5]、神の数字 (God's Number) と呼ばれている。この数字に近い少ない手数でキューブを完成させるようにすれば、失敗の発生回数を減らすことが可能となり、より少ない時間でキューブを完成させることが可能になると期待できる。

#### 参考文献

- [1] Takafumi Haseda : 全商品を世界基準配色へ移行します, <https://tribox.com/2013/05/world-standard-color-scheme/>, 2019 年 12 月
- [2] Daivid Gilday : MindCub3r for EV3 and NXT, <http://mindcuber.com>, 2019 年 12 月
- [3] LEGO MINDSTORMS EV3 ヘルプへようこそ, <https://ev3-help-online.api.education.lego.com/Retail/ja-jp/>, 2019 年 8 月
- [4] God's Number is 20, <http://cube20.org/>, 2019 年 10 月
- [5] 大石弥幸 : 「ルービック・キューブとその仲間たち」. 大同大学紀要, 54 巻 (2019), pp. 33-38 .