

自律制御型ロボットの研究

2519 鈴木 拓矢 2631 廣瀬 修也 2632 古川 泰地

要旨

ロボカップジュニア・サッカーの試合を見て、そのロボットがどうして正確にボールを追い、ゴールに向かうのか不思議に思った。この疑問を解決するために、自律制御型サッカーロボットを製作、プログラミングして、実際に試合を行った。その結果、ハード面は整備性が良く、同時にロボット同士の衝突に耐えられる構造、ソフト面ではセンサ値を基とした簡単な動作プログラムに変数などを用いて、状況に応じたプログラムの作成が必要だと分かった。

1. 目的

自律制御型ロボットが正確に動く仕組みをロボカップジュニア・サッカーの競技を通し、ハードとソフトの両面から探究する。

2. 使用した器具・装置など

(1) 工具

- ・ペンチ
- ・ニッパー
- ・ドライバー（精密含む）
- ・はんだごて
- ・やすり
- ・ワイヤーストリッパー
- ・圧着ペンチ
- ・グルーガン
- ・ボール盤 ※板に穴をあける工具

(2) 消耗品

- ・電池（単三，単四）
- ・電池ボックス
- ・2PIN 端子※
- ・3PIN 端子※
- ※それぞれセンサと基盤を繋ぐための端子
- ・螺子
- ・ナット
- ・スペーサー
- ・はんだ

- ・ユニバーサルプレート
- ・ギヤ付きモーター
- ・スポーツタイヤ
- ・結束バンド
- ・超強力両面テープ
- ・ビニールテープ
- ・偏光板

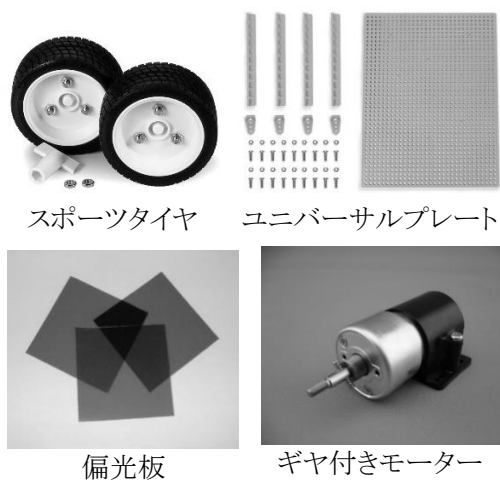
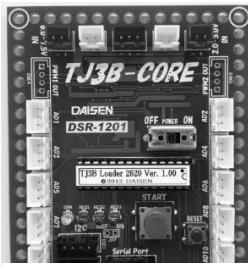


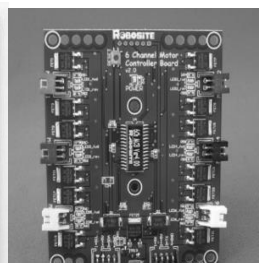
図1 消耗品の例

(3) 基盤・センサなど

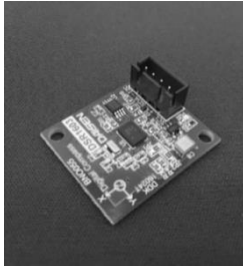
- ・TJ3b CORE × 2
- ・6CH モーターアンプ × 2
- ・ボール
- ・ボールセンサ × 10
- ・ラインセンサ × 2
- ・9D コンパス × 2
- ・超音波センサ × 6



TJ3b core



6CH モーターアンプ



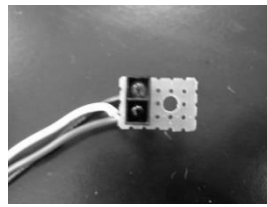
9D コンパス



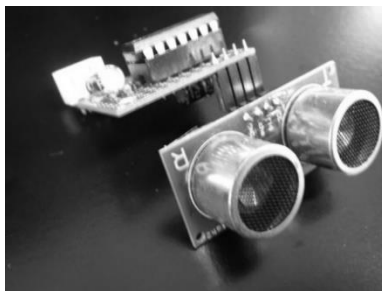
超音波センサ



赤外線発光ボール



ラインセンサ



連続通信回路

図2 各種基盤・センサの例

(4) その他

- ・ PC
- ・ コード
- ・ C-style for TJ3b
- ・ 5mm方眼用紙（設計図用）
- ・ 二輪駆動機体（図3）

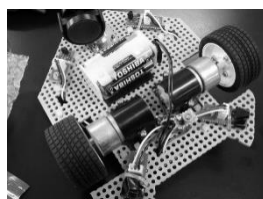


図3 二輪駆動機体（表面と裏面）

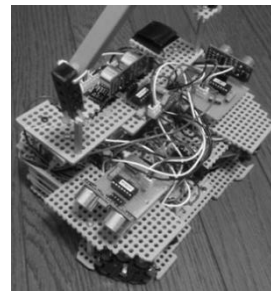


図4 三輪駆動
オムニ型ロボット

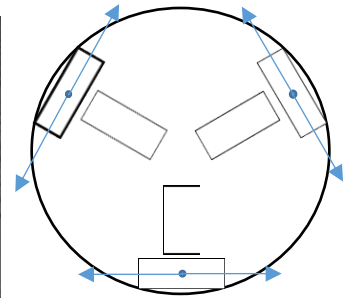


図5 モーター配置とそれ
に対応したベクトル
に関する設計図

3. 研究の手順

目的にあるような技術の習得を確認する方法として、ロボカップジュニア・サッカー部門に出場した。他チームとの試合や交流を通じて自律制御型ロボットのよりよい構造やプログラムの存在を探究した。

(1) 設計

当初、四輪駆動制御のオムニ型ロボットを二台設計しようと考えたが、製作期間が短かったため、二輪制御ロボット、三輪オムニ制御ロボットそれぞれ1台として設計図を作成した。（図5）

(2) プログラムの学習

プログラムは現在主流である「C言語」を使用した。（中学校などで基本的なプログラムの構造は学んだが、目指すのは実際のロボット工学における制御であるため）

「C言語」はプログラムの構造が今までとは格段に違い、難解なものとなったので、その学習期間だけ、製作と平行して二か月を使って機体に対応したロボット動作プログラムの書き方を習得した。

(3) 製作

センサをそれぞれはんだ付けし、配線を作製後接続した。モーターも同様に TJ3b core 基盤にはんだ付けした。

メインの板には、市販のユニバーサルプレート（直径3mmの穴が2mm間隔で配置されているもの）を使用した。しかし、設計の

段階でプレートの大きさに合わなかったため、最下段の二枚を、グルーガンで繋いだ。その部分の補強は L 字のプラスチック棒を使用した。

モーターは三輪につける場合それぞれの軸間の角度を合わせて付けなければならなかった。よって板にボール盤で穴を開け、螺子と結束バンドで止められるよう工夫した。

製作の段階で困難に感じたのはボールセンサやラインセンサ，方位センサなどをはんだ付けする必要があったことである。

(4) プログラムの試行

機体の完成後，作成してきたプログラムに実際の計測値を代入し，試験的に動作させた。しかし，実行する場所の明るさなどによって実際の動きの中では計測値がそのまま使うことができなかった。これは，プログラム内の反応値を上げることでうまく対応できることが分かった。

また，過去に考えたプログラムも周囲の明るさや電池の消耗による電圧の変化によってうまく動作しなくなってきたため，条件に合わせた修正プログラムを考えた。

(図 6，図 7 のプログラムの一例を参照)

```
001 //-----
002 #include "D_Main.h"
003 #include "D_I2C.h"
004 #include "D_SIO.h"
005 #include "D_EIO.h"
006 //-----
007 // Program Name : NewF... main170908[1].C
008 //-----
009 void user main(void)
010 {
011     gV[VAR_A] = 0;
012     while (TRUE) {
013         gV[VAR_B] = 0;
014         gV[VAR_C] = 0;
015         if (gAD[CN10] > 818) {
016             user_sub_1();
017             if (gAD[CN8] < 255) {
018                 if (gAD[CN9] < 255) {
019                     gPwm[0] = 0x00;
020                     gPwm[1] = 0x00;
021                     gPwm[2] = 0x00;
022                     gPwm[3] = 0x00;
023                     pwm_out();
024                     wait_ms(500);
025                 } else {
026                     gPwm[0] = 0x1E; 0x80;
027                     gPwm[1] = 0x1E; 0x80;
028                     gPwm[2] = 0x1E; 0x80;
029                     gPwm[3] = 0x1E; 0x80;
030                     pwm_out();
031                     wait_ms(500);
032                 }
033             } else if (gAD[CN9] < 255) {
034                 gPwm[0] = 0x1E;
035                 gPwm[1] = 0x1E;
036                 gPwm[2] = 0x1E;
037                 gPwm[3] = 0x1E;
038                 pwm_out();
039                 wait_ms(500);
040             }
041             gV[VAR_A] = 0;
042         } else if (gV[VAR_A] == 1) {
043             user_sub_1();
044             if (gAD[CN8] < 255) {
045                 if (gAD[CN9] < 255) {
046                     gPwm[0] = 0x00;
047                     gPwm[1] = 0x00;
048                     gPwm[2] = 0x00;
049                     gPwm[3] = 0x00;
050                     pwm_out();
051                     wait_ms(500);
052                 } else {
053                     gPwm[0] = 0x1E; 0x80;
054                     gPwm[1] = 0x1E; 0x80;
055                     gPwm[2] = 0x1E; 0x80;
056                     gPwm[3] = 0x1E; 0x80;
057                     pwm_out();
058                     wait_ms(500);
059                 }
060             } else if (gAD[CN9] < 255) {
061                 gPwm[0] = 0x1E;
062                 gPwm[1] = 0x1E;
063                 gPwm[2] = 0x1E;
064                 gPwm[3] = 0x1E;
065                 pwm_out();
066                 wait_ms(500);
067             }
068             gV[VAR_A] = 0;
069         } else {
070             if (gAD[CN1] > 864) {
071                 if (gAD[CN8] < 511) {
072                     gPwm[0] = 0x50;
073                     gPwm[1] = 0x46;
074                     gPwm[2] = 0x50;
075                     gPwm[3] = 0x46;
076                     pwm_out();
077                 } else if (gAD[CN9] < 511) {
078                     gPwm[0] = 0x46;
079                     gPwm[1] = 0x50;
080                     gPwm[2] = 0x50;
081                     gPwm[3] = 0x46;
082                 }
083             }
084         }
085     }
086 }
```

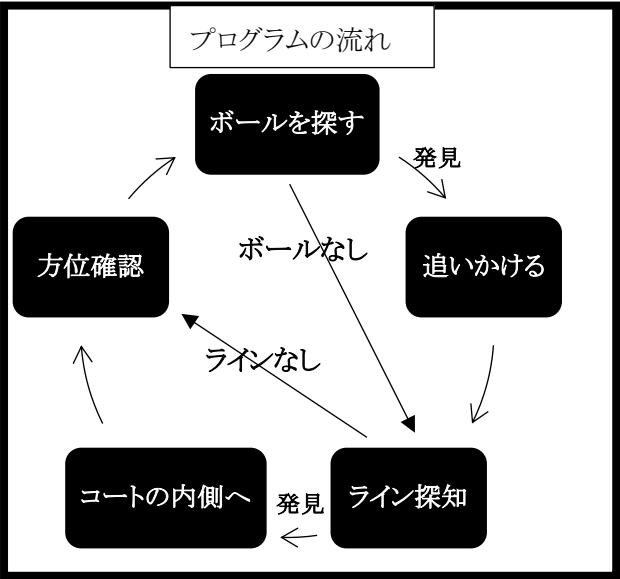


図6 プログラムの概略図

```

082 |         gPwm[3] = 0x50;
083 |         pwm_out();
084 |     } else {
085 |         gPwm[0] = 0x50;
086 |         gPwm[1] = 0x50;
087 |         gPwm[2] = 0x50;
088 |         gPwm[3] = 0x50;
089 |         pwm_out();
090 |     }
091 | } else {
092 |     if (gAD[CN2] > 716) {
093 |         user_sub_2();
094 |         continue;
095 |     } else if (gAD[CN3] > 716) {
096 |         user_sub_3();
097 |         continue;
098 |     }
099 | }
100 | } else if (gAD[CN2] > 613) {
101 |     user_sub_2();
102 |     if (gV[VAR_A] == 1) {
103 |         continue;
104 |     }
105 |     if (gAD[CN1] > 869) {
106 |         user_sub_4();
107 |     } else {
108 |         continue;
109 |     }
110 | } else if (gAD[CN3] > 613) {
111 |     user_sub_3();
112 |     if (gV[VAR_A] == 1) {
113 |         continue;
114 |     }
115 |     if (gAD[CN1] > 869) {
116 |         user_sub_4();
117 |     } else {
118 |         continue;
119 |     }
120 | } else {
121 |     if (gAD[CN4] > 613) {
122 |         user_sub_5();
123 |         if (gV[VAR_A] == 1) {
124 |             continue;
125 |         }
126 |     } else if (gAD[CN5] > 613) {
127 |         user_sub_6();
128 |         if (gV[VAR_A] == 1) {
129 |             continue;
130 |         }
131 |     }
132 | }
133 | }
134 | gV[VAR_B] = gV[VAR_B] + 1;
135 | if (gV[VAR_B] == 600) {
136 |     while (gAD[CN9] > 102)
137 |         user_sub_1();
138 |     if (gAD[CN7] > 613) {
139 |         if (gAD[CN8] > 613) {
140 |             gPwm[0] = 0x28; | 0x80;
141 |             gPwm[1] = 0x28; | 0x80;
142 |             gPwm[2] = 0x28; | 0x80;
143 |             gPwm[3] = 0x28; | 0x80;
144 |             pwm_out();
145 |         } else {
146 |             gPwm[0] = 0x28; | 0x80;
147 |             gPwm[1] = 0x28; | 0x80;
148 |             gPwm[2] = 0x28; | 0x80;
149 |             gPwm[3] = 0x28; | 0x80;
150 |             pwm_out();
151 |         }
152 |     } else {
153 |         gPwm[0] = 0x28; | 0x80;
154 |         gPwm[1] = 0x28; | 0x80;
155 |         gPwm[2] = 0x28; | 0x80;
156 |         gPwm[3] = 0x28; | 0x80;
157 |         pwm_out();
158 |     }
159 | }
160 | }
161 | }
162 | }
163 | //

```

変数でプログラムの周回数を計算

If文以降、ボールがなかった場合に方向を確認

ループ終了

プログラム終了

図7 使用したプログラム例

(5) 競技会への参加

ロボカップジュニア・サッカーの試合を通してロボットの課題点を探った。

参加した試合は8月に行われた交流試合と12月に行われたロボカップ本戦へとつながる中津川ノード大会である。

交流試合では、他のロボカップ選手と試合、交流することでロボットの構造、プログラムの新たな知見を得た。実際の試合でロボットが使用するセンサ値は常に変動し、値に合わせてプログラムを書く必要があることも理解した。

中津川ノード大会では本戦と同じ試合方式だったため、試合中に発生した不具合の即時修正の必要性を知った。設計の部分では予期していなかった基盤の整備性が悪かったことを、試合を通じて発見し、試合の空いた時間に修正を施すことができた。

試合結果は総当たりで同率1位となったため延長戦が実施され、最終順位は2位だった。

表1 中津川ノード大会結果

ライトウェイト			
ESSH	RN	B&W	
ESSH	0:2	3:0	1
RN	2:0	0:1	1
B&W	0:3	1:0	3

この結果を踏まえ、県のロボカップジュニア、ブロック大会へ出場し、さらに高度な技術の交流や習得をしようと考えている。

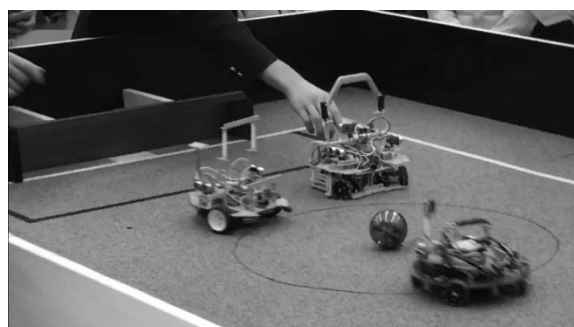


図8 中津川ノード大会の様子

(6) 競技会の観戦

7月に愛知県で開催された「ロボカップジュニア世界大会 in 名古屋」、12月9日に愛知工業大学付属高校で行われたロボカップジュニア東海ブロック大会(図9)の観戦、競技者との交流を行った。

世界大会には、さまざまな国から競技者が訪れており、それらのチームのポスターを見て、ロボットの構造の改良した点などから自分たちにも使える技術に関する知識を得た。

東海ブロック大会では強豪チームの試合を録画し、動作の良い部分をそのチームの人と交流してなぜ良かったのか、使用した部品や工具、プログラムの方法などの観点から話し合った。



図9 東海ブロック大会会場
(愛知工業大学付属高校)

(7) 改善予定の課題

①二輪型自律制御ロボット

- ・ラインセンサと左前のボールセンサの確認し、改良を加える。
- ・超音波センサと方位センサを有効利用したプログラムの作成をする。

②三輪オムニ型自律制御型ロボット

- ・後方ボールセンサの不具合の改善。
- ・ラインセンサの使用する光周波数を可視光のものにする。

③全体として

- ・プログラムの精度を上げる。
- ・機体の構造をさらに簡易化し、分解を楽にすることで各部品の故障時に手早く取り換えできるようにする。

(8) 実際行った細かな仕様の改善など

①偏光板

偏光板とは、ある一定の光を通す構造をもつ薄い板の総称であり、今回私たちは板に対して垂直の光のみを通過させるタイプの偏光板を用いた。

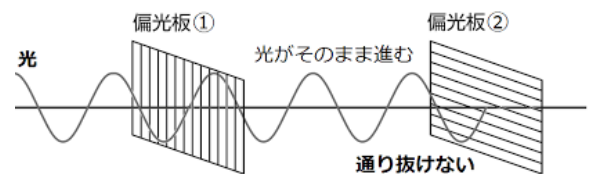


図10 偏光板の仕様

この偏光板を、超強力両面テープでボールセンサに付けることで、その方向だけの赤外線を読み取れるようにした。

②ボールセンサについて

初め、受光部、コンデンサ、抵抗だけを積んだ単純な回路のものを使用したが、後にノイズが発生する欠陥が見つかり改良した。

この欠陥は、回路内に電流のノイズが発生し、それによって赤外線に対する反応が鈍くなってしまうものであった。そのため回路内に、受光部に入る電流を30m秒毎に0Aにする回路を組み込んだ。この改良により赤外線ボールに対する反応が上がった。

表2 ボールセンサ値と距離の関係

距離	旧型センサ値	新型センサ値
0.20m	62	50
0.40m	65	54
0.60m	71	58
0.80m	72	62
1.00m	72	66

※すべてセンサ正面から測定

表3 ボールセンサ値と角度の関係

角度	左側	右側
10°	48	48
20°	48	48
30°	48	48
40°	48	48
50°	55	55

※すべてセンサから 10cm 離れた位置で測定

③超音波センサについて

超音波センサの動作方式は「I/O」通信方式となっており，これは基盤側からデータの取得を指示された時だけに基盤へのデータ出力を行うものとなっている。

基本的に，この方式をとることで電力の消耗を抑え，長時間の運用に適した形となっていた。だが，今回参加するロボカップ・サッカーでは瞬時の自己判断が求められているため，基盤側から指示するよりも早くセンサ値を受け取る必要がある。よって，超音波センサの値を永続的に出力し続ける回路を作成し，これを機体に組み込んだ。仕様は以下に示す。

「超音波センサ HC-SR04 駆動基板」

- ・ 4 端子超音波距離センサ
- ・ Trg 端子に $10\mu s$ のパルス入力が入ると測定開始。
- ・ 超音波パルスの戻りを検知し Echo 端子に $150\mu s \sim 25ms$ のパルスを出力。
- ・ この動作を R8Cm11 マイコンで 100ms 毎に繰り返し，測定結果を 2 回平均して AD ポートに出力する。
- ・ C-style のセンサーモニターで物体までの距離が 3cm の時約 4%，1 m の時約 22% が出る。
- ・ ほぼ 5cm で 1% 変化する値が得られる。

4. 結果

試合に 2 回参加し以下のことが分かった。

- ・ 自律制御型ロボットにはセンサ同士の干渉があるため，干渉しない機械構造に改良するか，プログラム内で干渉している条件を満たす場合に動作を変更する一連の自律制御が必要である。
- ・ プログラムの試行をできるだけ増やすことで動作環境でのセンサ値を発見し，プログラムに組み込むことが必要である。

5. 考察

正確な自律制御を確立させるには，結果で述べた方法の他にも，あらゆる状況を想定し，プログラムを組み込むことが必要だと考えられる。

6. 参考文献，引用資料

- ・ TJ3b C-Style 操作編

www.daisendenshi.com/download/TJ3B_C-StyleManual_2-161015.pdf

- ・ TJ3b Core マニュアル

www.daisendenshi.com/download/TJ3BCore_Manual-130225.pdf

- ・ 制御の基礎 - 東北学院大学

http://www.mech.tohoku-gakuin.ac.jp/rde/contents/sendai/mechatro/archive/RMSeminar_No09_s8.pdf