

システム実験 実験後期第3回レポート

6119019056 山口力也

2019/10/23 日提出

1 レポート 11.3.5.1

演習 11-3-4-1 のプログラムの実行状況を説明せよ。その際,roll 角の正負と加速度の y 方向分の正負の関係,pitch 角の正負と加速度の x 方向分の正負の関係などを表にまとめよ。

ロボットを左右に傾ける (roll 角が \mp に変化する) と a_Y の値が \mp に変化した。ロボットを前後に傾けると (pitch 角が \mp に変化する) と a_X が \pm 値が変化した。

以下表 1 と表 2 に結果を示す。

表 1: roll 角の変化による a_Y の変化

roll 角	a_Y
\mp	\mp

表 2: pitch 角の変化による a_X の変化

pitch 角	a_X
\mp	\pm

2 レポート 11.3.5.2

課題 11-3-4-1 で最終的に出来上がった zone3beta() 関数における mode_G の各値における動作を日本語または英語で説明せよ。

以下ソースコードにソースコードを示す。

ソースコード 1: 課題 11.3.4-1

```
1 //
2 // Zone 3 Beta
3 //
4
5 float avex = 0;
6
7 float turnTo(float dir) {
8     float heading, diff;
9
10    heading = direction_G;
11    if (heading>180) heading -= 360; //-180 < heading < 180 ,
        E=90,W=-90,N=180,S=-180
12    heading -= dir; //常に 90引いてる
13    if (heading>180) heading -= 360; //200とかなら-160、つまり
        一周したらその分もどして-180<heading<180にする
14    if (heading<-180) heading += 360;
15    diff = -heading*5; // P-制御
16    if (diff>200) diff = 200; // 飽和 -200<diff<200に正規化
17    if (diff<-200) diff = -200; // 飽和
18    return diff;
19 }
20 void zone3beta()
21 {
22     static int countTurn = 0;
23     int done = 0;
24     float speed0, diff;
25
26     avex = 0.9*avex + 0.1*compass.a.x;
27
28     switch ( mode_G ) {
29         case 0: // setup はここ
30             countTurn = 0;
31             mode_G = 1;
32         case 1: // 山探し (ここでは直進)
33             if (avex>4000) { // 登り始めたら
34                 mode_G=2;
35             }
36             speed0 = 150;
37             diff = -0.02*compass.a.y; // P-制御
38             break;
39         case 2: // 山登り中
40             if ((avex<1500)&&(avex>-1500)) { // 登頂したら頂上の
                中心部まで進む
41                 done = waitfor(150); // 200ms は要調整
42             }
```

```

43     if (done == 1) { // 頂上の中心部まで来たら停止
44         speed0=0;
45         diff =0;
46         mode_G = 3;
47     } else {
48         speed0 = 150;
49         diff = -0.02*compass.a.y; // P-制御
50     }
51     break;
52 case 3: // 500ms 停止
53     speed0 = 0;
54     diff = 0;
55     done = waitfor(500);
56     if ( done == 1 )
57         mode_G = 4;
58     break;
59 case 4: // その場回転で東を向く
60     speed0 = 0;
61     diff = turnTo(90); // P-制御
62     if (abs(diff)<=25) { //80-100
63         diff = 0;
64         done = waitfor(500);
65         if ( done == 1)
66             mode_G = 5;
67     }
68 case 5: //南
69     diff = turnTo(0); // P-制御
70     if (abs(diff)<=25) { //80-100
71         diff = 0;
72         done = waitfor(500);
73         if ( done == 1)
74             mode_G = 6;
75     }
76     break;
77 case 6: //にし
78     diff = turnTo(-90); // P-制御
79     if (abs(diff)<=25) { //80-100
80         diff = 0;
81         done = waitfor(500);
82         if ( done == 1)
83             mode_G = 7;
84     }
85     break;
86 case 7: //北
87     diff = turnTo(180); // P-制御
88     if (abs(diff)<=25) { //80-100

```

```

89         diff = 0;
90         done = waitfor(500);
91         if ( done == 1)
92             mode_G = 8;
93     }
94     case 8: // その場回転で東を向く
95         speed0 = 0;
96         diff = turnTo(90); // P-制御
97         if (abs(diff)<=25) { //80-100
98             diff = 0;
99             done = waitfor(500);
100             if ( done == 1)
101                 mode_G = 9;
102         }
103         break;
104     case 9: //下る
105         ;
106         //下るまで直進
107         speed0 = 150;
108         diff = -0.02*compass.a.y; // P-制御
109     case 10:
110         speed0 = 150;
111         diff = 0.02*compass.a.y; // P-制御
112         if((avex<1500)&&(avex>-1500)){
113             mode_G = 99;
114         }
115         break;
116     case 99: // 停止
117         speed0 = 0;
118         diff = 0;
119         break;
120     default:
121         break;
122 }
123 motorL_G = speed0 + diff;
124 motorR_G = speed0 - diff;
125 }
126
127
128 int waitfor( unsigned long period )
129 {
130     static int flagStart = 0; // 0:待ち状態, 1:現在計測中
131     static unsigned long startTime = 0;
132
133     if ( flagStart == 0 ) {
134         startTime = timeNow_G;

```

```

135     flagStart = 1; // 現在計測中にしておく
136 }
137
138 if ( timeNow_G - startTime > period ) { // 計測開始からの
    経過時間が指定時間を越えた
139     flagStart = 0; // 待ち状態に戻しておく
140     startTime = 0; // なくても良いが,形式的に初期化
141     return 1;
142 }
143 else
144     return 0;
145 }

```

mode_G が 1 のとき山を探して直進する. その後山を見つけると mode_G が 2 に移行して P 制御で山を登る. 頂上まできたら mode_G が 3 に移行して 200ms 直進してそこで止まる.mode_G が 4 8 に移行したあとその場で一回転し mode_G が 9,10 で山を下る.

3 レポート 11.3.5.3

課題 11-3-4-2-1 で作成した Arduino スケッチの PI-制御の部分を一覧として報告せよ. 大域変数, また,static 変数を用いた場合は, その宣言部分や初期値も報告すること. また, 設計パラメータの値も報告すること. 課題 11-3-4.2-2 で作成した Arduino スケッチを日本語または英語で説明せよ.

こちらは時間内に実験を終えることができなかった.

4 発展課題レポート 11.3.5.1

発展課題 11-3-4-1 で作成した Processing スケッチの特徴的なスクリーンショットを図として掲載し, ロボットの状態をどのように表示しているか日本語または英語で説明せよ. その工夫点を説明せよ.

こちらは時間内に実験を終えることができなかった.