

## Robotics Challenge 考察レポート

### ・作成したプログラムとその説明

Wall following のコードを利用し、カラーセンサで用いることができるように拡張した。Wall following のプログラムは2つの感圧センサの ON/OFF 状態に対する動作を学習し、壁沿いを走行するようにするというものであったが、今回は2つのカラーセンサで路面の色を検知し、路面の色に対する行動を学習し、ライントレースができるようにすることを目指した。

処理の中心となるプログラムは LineTrace.java。

```
while(Button.ESCAPE.isUp()){  
  
    states = getEnvironment(states, color1, color2);  
  
    int a = Q.getAction(states);  
  
    int command = q.getCommand(a);  
  
    motorCommand(pilot, command);  
    Delay.msDelay(500);  
  
} exit(sensor1, sensor2);
```

このプログラムの主要部を示す。まず getEnvironment メソッドにより左右センサで感知された路面の色から状態を割り出す。color1 = 左カラーセンサの認識した色、color2 = 右カラーセンサの認識した色であり、各センサの検知した色に対する状態は以下の通り。

表1 カラーセンサの検知した色と状態の対応表

状態	左カラーセンサ	右カラーセンサ
0	白	白
1	白	黒
2	黒	白
3	黒	黒

次に Q.getAction メソッドで Q 値を更新、行動を選択する。

```
public int getAction(int[] states){
    int st = qT.getState(states); //Environment -> ID state (New state)
    if(ls >= 0){
        ++ct;
        r = qT.reward(ls, la);

        if(r == 1){ //In case that the reward is greater than zero
            if(LEARN_RATE > 0 && ct > 30){
                LEARN_RATE = LEARN_RATE - 0.001f;
            }
            if(ct <= 30) {
                LEARN_RATE = LEARN_RATE + 0.001f;
            }
            if(EXPLORE_RATE > 0){
                EXPLORE_RATE = EXPLORE_RATE -0.001f;
            }
        }

        MaxA = getMaxAction(st);
        Q[la][ls] = Q[la][ls] + LEARN_RATE*(r + Q[MaxA][st] - Q[la][ls]);
    }

    ls = st;

    float rand = (float) Math.random();
    if(rand > EXPLORE_RATE){
        la = MaxA;
    }else{
        la = (byte)(Math.random()*11); // 6 is the number of actions
    }

    writeMess("State & Action", "Reward = ", ls, la, r);
}
```

Wall following で用いていたコードと基本的に同一のもの。定められた確率  $\epsilon$  でランダムな行動を選択し、 $1 - \epsilon$  の確率で Q 値の最も高い行動を選択する  $\epsilon$ -greedy 法を用いている。報酬が正のときは  $\epsilon$  を減少させ、最終的には Q 値の最も高い行動だけを選択するようになる。その後モータの回転を制御する指示を送る。以後これを繰り返す。

それぞれの状態と選択する行動による報酬は以下の通り。

表 2 状態と行動に対する報酬の一覧

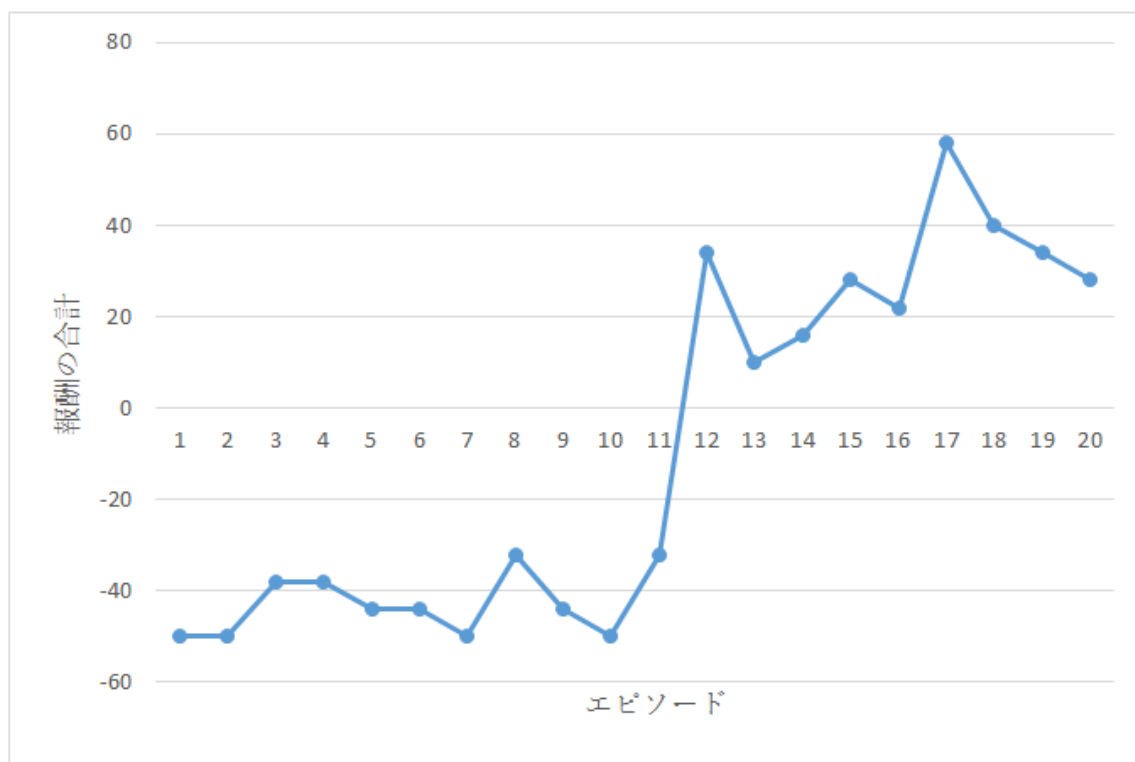
	0(白,白)	1(白, 黒)	2(黒, 白)	3(黒, 黒)
前進	1	1	1	-1
左折 (弱)	-0.3	0	0.2	0.2
左折 (強)	-1	-1	-1	0.8
右折 (弱)	-0.3	0.3	0.3	0.2
右折 (強)	-1	-0.2	-0.2	0.8
後退	-1	-0.2	-0.2	0
左後退 (弱)	-1	-0.2	-0.3	-0.5
左後退 (強)	-1	-1	-0.3	-0.5
右後退 (弱)	-1	-0.3	-1	-0.5
右後退 (強)	-1	-0.3	-1	-0.5
停止	-0.2	-0.2	-0.2	-0.2

・結果の表示  
動画を参照。

・結果に対する考察と評価

最初はランダムな動きを選択するが、次第に白線に沿って前進することを学習し、最終的にライントレーサとして機能するようになる結果を得られたが、左右 2 つのカラーセンサに対して、(左, 右) = (白, 白), (白, 黒), (黒, 白), (黒, 黒) という状態に対して次の行動を選択する際に、(左, 右) = (黒, 黒) という状態に対しては「左または右の一方にしか曲がらない」ように学習してしまい、ジグザグ状のコースでは必ずコースアウトするようになった。コースに戻るためには、左右のどちらにも旋回してコースを探せるようにしておく必要がある。実験の際に、学習用に用いたコースが楕円状のもので、しかも一方向にしか走行させていなかったの一方に曲がることしか学習しなかったということが原因のひとつではないかと思われる。また、行動選択において「Q 値がほぼ同じであっても、少しでも大きい行動が必ず選択される」という  $\epsilon$ -greedy 法を用いていることにより、学習の最終段階では必ずある状態に対しては必ずひとつの行動しか選択できないことになっているので、確率的

に行動を選択するボルツマン選択のような方法を用いることで、この問題を解決できる可能性がある。



エピソードごとの報酬の合計