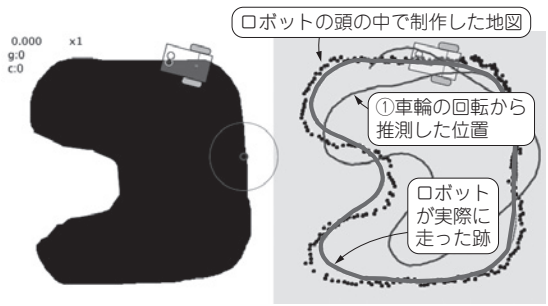
第9章

正確な位置&地図推定「SLAM」のプログラム

牧野 浩二



（a）ロボットの今の位置　（b）ロボットの過去の動き

図1　ライン・トレース・ロボットのシミュレーションでSLAMの動作をみる

紹介するSLAMプログラム

● ダウンロード方法

動作プログラムは本書サポート・ページからダウンロードできます．

```
https://interface.cqpub.co.jp/2023ai45/
```

● 中身

ダウンロードしたプログラムを解凍してProcessingで実行すると，5つのタブが表示されます．

▶ LineTrace タブ（重要）

パーティクルを描いたり，次の動作を決めたりする部分です．

▶ IfThen タブ

ライン・トレース・ロボットをルールに沿って動かすための部分です．

▶ ImageProc タブ

実際に移動するマップを表示したり，白黒情報を取り出したりするための部分です．

▶ ParticleFilter タブ（重要）

パーティクルを作成したり，移動したり，重みを決めたりする部分です．

リスト1　位置情報を修正せずに動いた場合の軌跡を計算する

```
1   // Raw Odometry
2   w_noize = 1.1+ random(0.0, 0.1);
3
4   d_src = t_vel + d_noize;
5   w_src = t_sq*w_noize;
6
7   sq_src += w_src;
8   sx_src += d_src*sin(sq_src);
9   sy_src -= d_src*cos(sq_src);
10
11  cPF.srcTrack.append(sx_src, sy_src, sq_src, 0);
```

▶ RingBuffer タブ

パーティクルの軌跡をある程度の長さだけ覚えておくための部分です．

プログラムの全体像

● ロボットの動作を定める

ライン・トレース・ロボットの動作を決める部分を見ていきます．これは，LineTraceタブの中に書かれているUpdateState関数の中にあります．その中のロボットが位置情報を修正せずに動いた場合の軌跡を計算している部分を**リスト1**に示します．

この場合の位置と方向はそれぞれ，sx_src，sy_src，sq_srcとしています．そして，この計算で得られた結果が**図1**の細い実線①になります．

今回のシミュレーションでは「わざと」誤差が生じるようにしています．**リスト1**の2行目で1.1 ～ 1.2のランダムな数を生成し，4行目で方向に誤差を加えています．その誤差を加えたデータをもとにして，ロボットの位置を7 ～ 9行目で計算しています．

このシミュレーションでは，角度の誤差だけ入れて，速度には誤差を加えていません．ロボットが移動するたびに左に曲がっていくように，1.1 ～ 1.2の値としています．左右同じ程度に曲がる誤差にすることもできますが，その場合はあまり誤差がたまらずにシミュレーション結果を見ても効果が分かりにくいため，わざと左だけに曲がる誤差を加えました．