

チーム紹介、目標、意気込み

私たちSojoRoboConsは同じ研究室の4年生4名で構成されたチームです。チーム名とロゴは今までの先輩方から受け継がれてきた大切なものです。研究とETロボコンを両立しながら活動してきました。

私たちの目標は「チャンピオンシップ出場」です。分からないながら、チームで協力して活動を続けてきました。

夏休みを返上し、このETロボコンに力を注いできたので、私たちの全力のロボット捌きを是非堪能してもらいたいと思います。

モデルの概要

- ・ メイントレーサーで状態に応じて処理を行うことで、プログラムの記述に柔軟性ができました。
- ・ 安定した走行を実現するために、ローパスフィルタでノイズを削減し、滑らかな走行を実現しました。また、PID制御を使用することで、カーブでの脱線を防ぐことができました。
- ・ Bluetoothでリモートスタートを行うことにより、スタート時の振動を避けることができ、スタートの安定性を維持することができています。
- ・ キャリブレーションの成功率を高めるために、尻尾の角度を1度ずつ上下に設定できるようにしました。その結果、余裕を持って準備できるようになり、安定した角度に調節できることでロケットスタートの成功率もあげることができました。

モデルの構成

1. 要求分析

- I. 「コースを完走する」ためにマインドマップで、要求・実現方法・要求技術を洗い出し、このモデリング図では、ライントレース走行、キャリブレーション、リモートスタート、ロケットスタートをそれぞれ、赤、青、緑、黄色の図で示し、このモデリング図ではこれら4つに着目します。
- II. マインドマップで洗い出した実現方法で考えられる脅威をミスユースケース図で示し、要求技術に結びつけました。
- III. I.とII.を元にev3-rtの全体の流れをキャリブレーション、スタート指示後からロケットスタート、走行中の3段階に分け、アクティビティ図で示しました。

2. 機能モデル

- I. 似たような働きを持つクラスをパッケージ化し、5つに分類しました。それらの概要と関連をまとめました。
- II. ライントレース走行、尻尾調整、リモートスタート、ロケットスタートの機能を実現するための必要な要素をクラス図で示しました。

3. 振る舞いモデル

- I. 定義された要素を元に、ステートマシン図で全体の流れを示しました。
- II.とIII. キャリブレーションの振る舞いとロケットスタートから走行中の振る舞いをそれぞれシーケンス図で、部品間の細かいメッセージのやりとりを記述しました。

4. 工夫点

- ・ 急な旋回や誤作動を減らし、安定走行を実現するためにローパスフィルタを使用しました。
- ・ ローパスフィルタをより効果的に使用するにはローパスフィルタ適用段階“n”を求める必要がありました。
- ・ フィルタの適用段階に応じて、完走率と平均タイムで表でまとめ、比較を行い選定をしました。



課題選択

「コースを完走する」

1.1 マインドマップ

チームの「チャンピオンシップに出場する」という目標果たすためには避けては通れない「コースを完走する」という課題を攻略すべく、まずはマインドマップで必要な技術要素をあげた。このモデル図では、上の4つの実現方法に着目する。

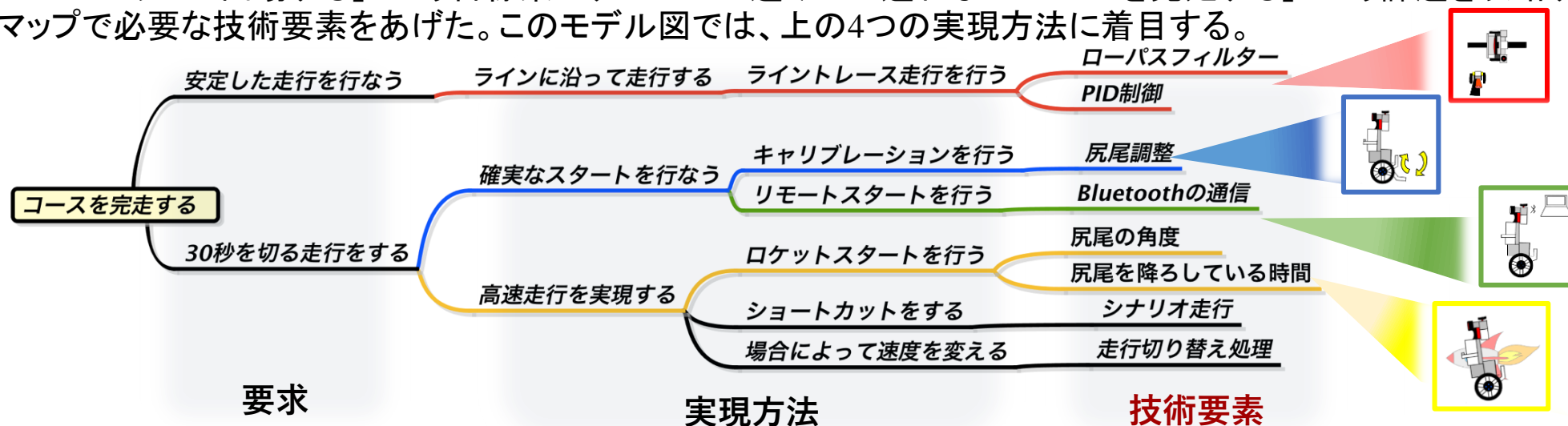


図1.1 マインドマップ

1.2 ミスユースケース図

1.1で挙げた実現方法の脅威を以下の図で示す。この脅威を1.1の技術要素で解決する。

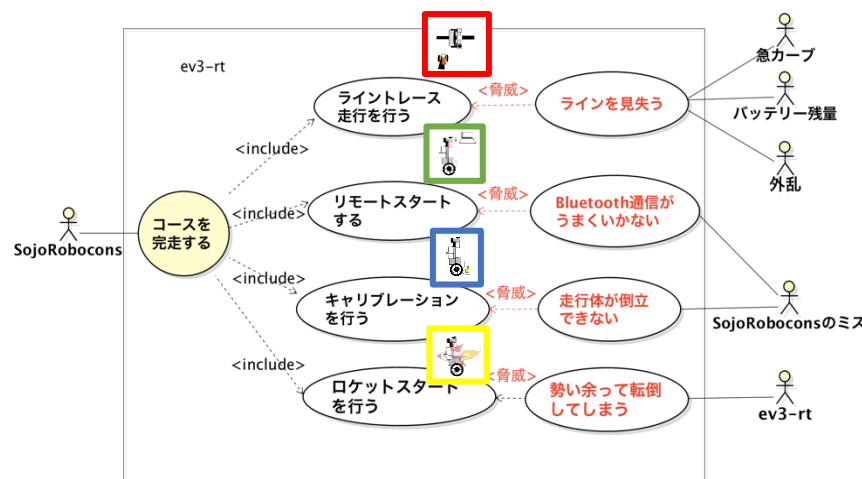


図1.2 ミスユースケース図

1.3 アクティビティ図

1.2で挙げた技術要素を使用し、ev3-rtが行う流れを段階毎に示す。

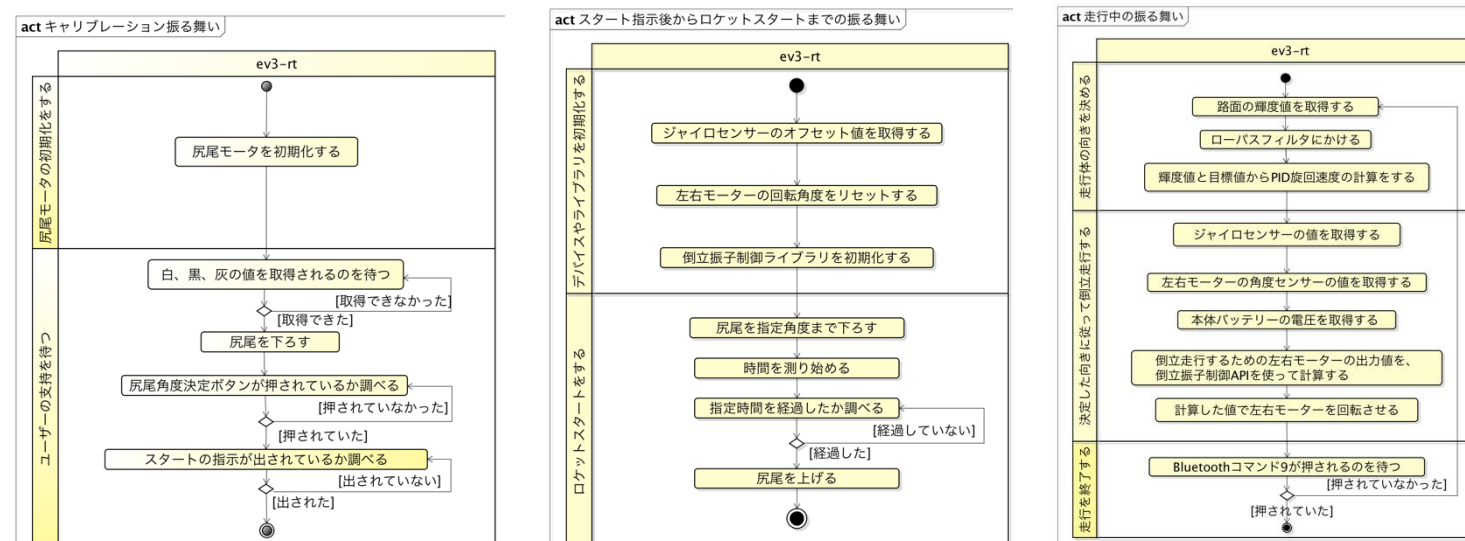


図1.3 アクティビティ図

表2.1 パッケージの概要

```
graph TD; Management[管理] -.-> Function[機能]; Management -.-> UI[UI]; Function -.-> Device[デバイス]; UI -.-> Device; UI -.-> Library[ライブラリ]; Device -.-> Library;
```

図2.1 パッケージ同士の関係

機能を実現するために、以下のクラス図で示す。

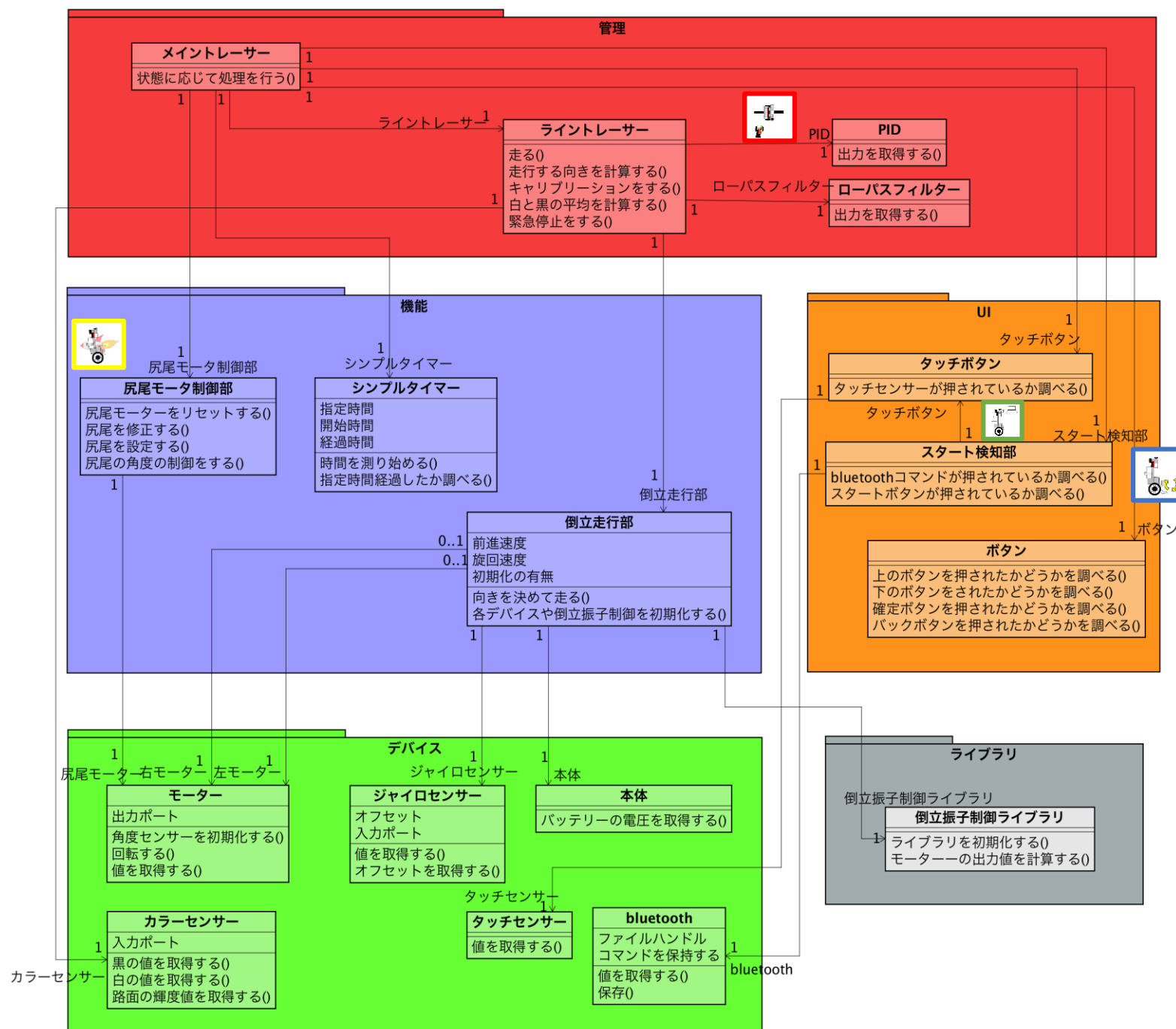


図2.2 クラス図



3.1 ステートマシン図

下記に機能を実現するための状態遷移図を記載する。

キャリブレーションの
詳しい流れを3.2に示す

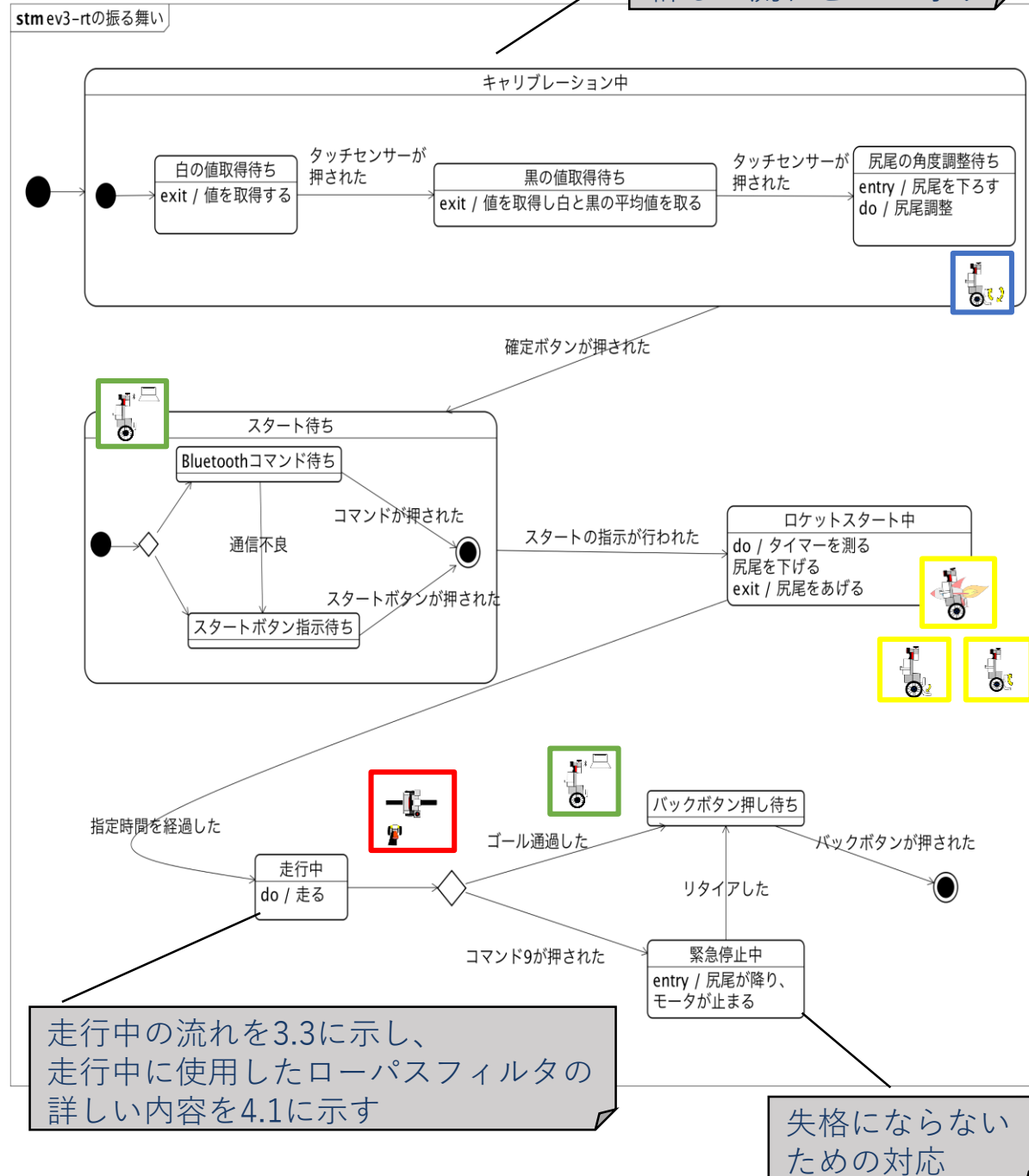


図3.1 ステートマシン図

3.2 キャリブレーションのシーケンス図

キャリブレーションのみの細かい流れを下記に記載する。

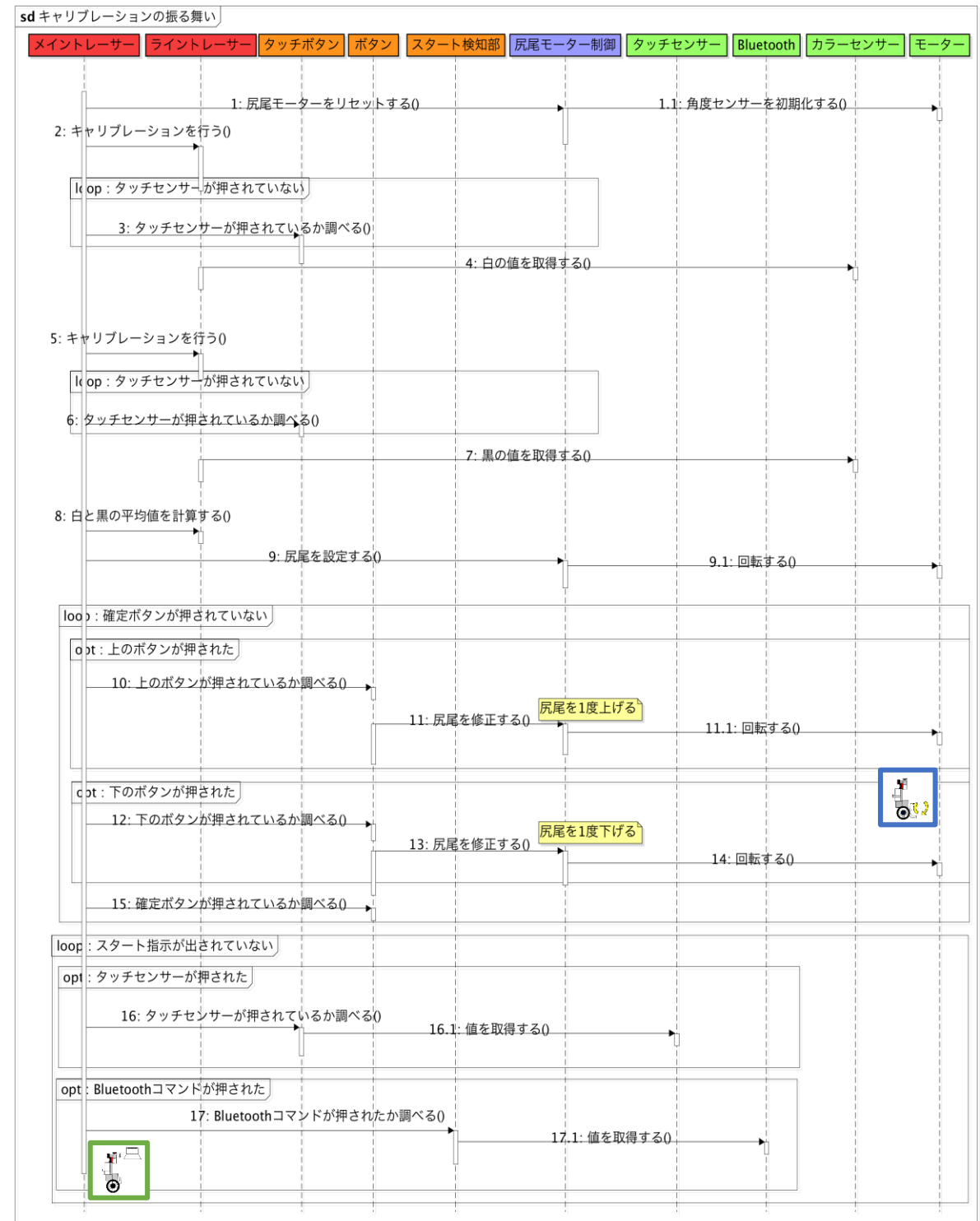


図3.2 キャリブレーションのシーケンス図



3.3 ロケットスタートと走行中のシーケンス図

ロケットスタートから走行中の細かい流れを下記に示す。

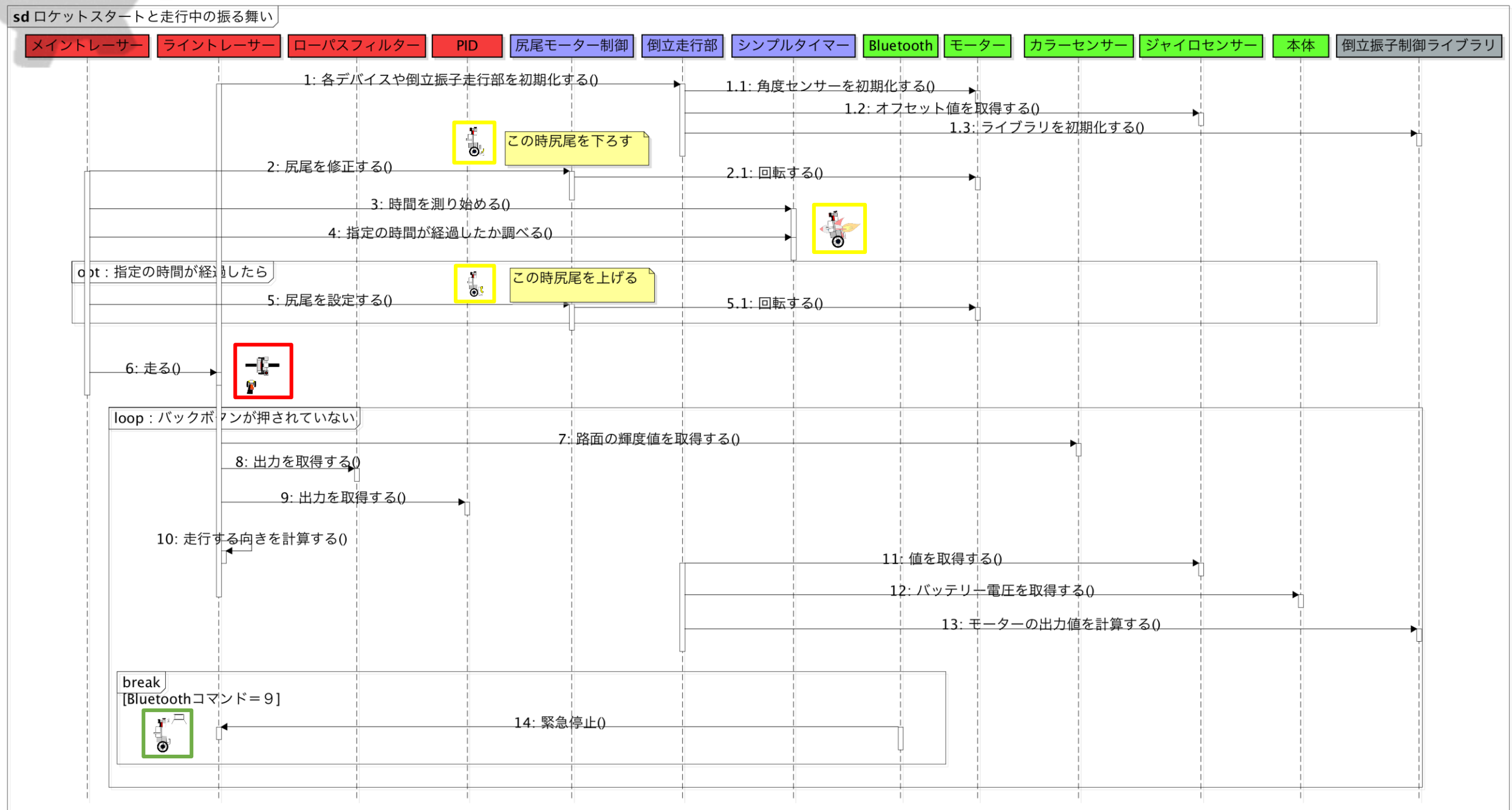


図3.3 ロケットスタートと走行中のシーケンス図



4.1 ローパスフィルタ

1. 目的

急な旋回や誤作動を減らし、安定した走行を実現する。

2. 課題

- I. 外乱の影響で光センサーの値にノイズが発生する。
- II. 急な旋回や誤作動が起きてしまうことがある。

3. 改善案

カラーセンサーとジャイロセンサーにローパスフィルタを用いて値の急激な変化を抑止する。
以下の式より輝度値を算出する。

$$x[s]: \text{光のセンサの値}, s: \text{現在のサンプリング}, n: \text{ローパスフィルタの適用段階}$$

$$\text{輝度値} = (x[s] + x[s-1] + x[s-2] + x[s-3] + \dots + x[s-(n-1)]) / n$$

図4.1に示すように、ローパスフィルタを用いることで値が滑らかになる。
しかし、より効果的にローパスフィルタを使用するには
ローパスフィルタ適用段階“n”を求める必要があると考えた。

4. 検証方法

- I. 表4.1に示すようにフィルタ数を8パターン10回ずつLコースを走行させ、
図4.2に示すGate1~Gate2のタイムを計測する。
(タイム測定時コース範囲設定理由：ロケットスタートの影響をさけ、
ローパスフィルタの影響が分かりやすいカーブが多いため。)
- II. 完走率と平均タイムで比較する。
- III. 差があまり無い場合、さらに10回測定し、合計20回で比較する。

5. 結果

- I. 表の4.1の □ で囲んでるように、
フィルタの7と8が完走率が高く、タイムも良い。
- II. 表4.2より、フィルタ7は平均タイムでは8より高いが、
フィルタ8の方が標準偏差が低いことから走行タイムに
バラつきがない。また、完走率が高いため、安定性が優れている。

よって、ローパスフィルタの適用段階は8に定めた

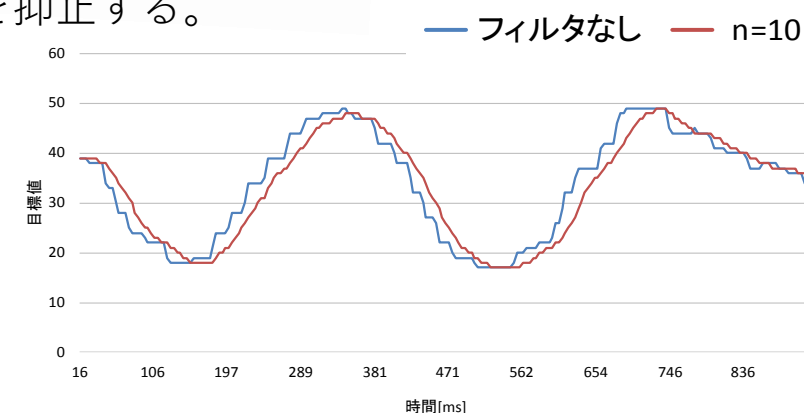
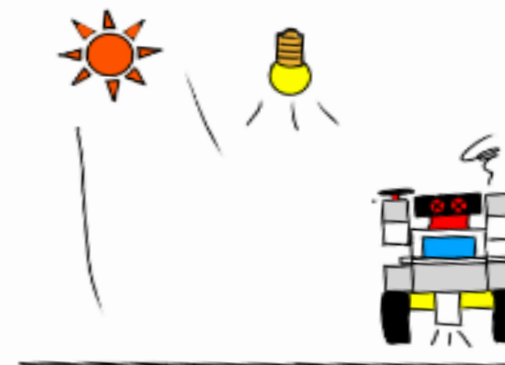


図4.1 ローパスフィルタ有無比較

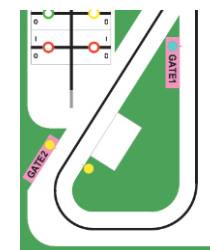


図4.2 検証コース

表4.1 フィルタ比較

フィルタ	完走率 (%)	平均タイム	タイムの順位
10	100	11.092	7
9	90	10.760	4
8	100	10.645	2
7	90	10.561	1
6	90	10.801	5
5	80	10.818	6
4	70	10.738	3
2	50	11.136	8

表4.2 フィルタ最終比較

フィルタ	完走率 (%)	平均タイム	標準偏差
8	95	10.819	2.51
7	90	10.656	3.57