ETロボコン2018
デベロッパー部門 プライマリークラス

地区: 東海　　地域: 愛知県豊田市

チームNo. 175　チーム名: チーム八草

所属: 愛知工業大学

チーム紹介、目標、意気込み

チーム八草は、新たに加入した一年生メンバーと、二年生メンバーで構成されたチームです。

目標は全国大会に出場することです。昨年は残念な結果に終わってしまいました。とても悔しかったです。だから、今年は去年の無念を晴らすためにも、全国大会に出場する心づもりです。

頑張りますので、応援よろしくお願いします！

ETロボソナーセンサァー（◉＝◉）

モデルの概要

モデル図全体を読んで得られる分析として、適切な機能の分割により、高次の課題解決に際して、内部の細かい動作を気にしない抽象的な命令で問題解決に当たれることがわかりました。

機能モデルの項では、ユースケース図を記述しました。ユースケースを機能要求と非機能要求に分類することによって、機能の明確化を図っています。

構造モデルの項では、パッケージ図によるクラスの大別を行うことで全体を俯瞰できるようにし、クラス間の関係をよりわかりやすく表現しました。

振る舞いモデルの項では、シーケンス図の「相互作用の利用」を使い、全体、詳細の順で理解できるようにしています。

工夫点の項では、その要素技術がどう作用したのか、わかりやすくグラフを用いているところがアピールポイントです。

モデルの構成

1. 機能モデル

まず、マインドマップにより機能候補を抽出しました。そして、それらを整理するためにユースケース図を書き、完走するために必要な機能が、走行用データを管理する機能、スタートする機能、ラインに沿って走行する機能に決定しました。また、ユースケース記述、アクティビティ図を記述することによって機能の詳細化を図りました。

2. 構造モデル

項目は、『2.1依存関係を俯瞰するためのパッケージ図』、『2.2ユースケースを実現するためのクラス構造』があります。

設計要素の大まかな関係を表すパッケージ図、パッケージの役割をまとめた表を2.1で、設計の詳細を表すクラス図を2.2で取り扱っています。

パッケージ及びパッケージが内包するクラスは下表のようになっています。

パッケージ名	内包するクラス
シナリオ走行部	シナリオ走行クラス
計算部	旋回量計算クラス、倒立演算クラス
スタート部	スタータークラス
データ管理部	走行用データクラス、キャリブレーションクラス
走行部	両輪制御クラス、倒立制御計算クラス
デバイス部	提供されているクラスと、その拡張クラス

3．振る舞いモデル

まず、高次のタスク、サイクルについての定義を表でまとめています。そして、『相互作用の利用』を用いて全体の流れを表現し、それらの詳細についてのシーケンス図を記述しました。

1 機能モデル

1.1 機能の抽出

私たちは課題として『コースを完走する』を選択した。『コースを完走する』で定義していかなければならない範囲を、走行体の起動から、ゴールするまでの間とする。

それから、チーム全員で実装する機能を考えるためにマインドマップを書いた。話し合いにより、要求を選定した(図1)。

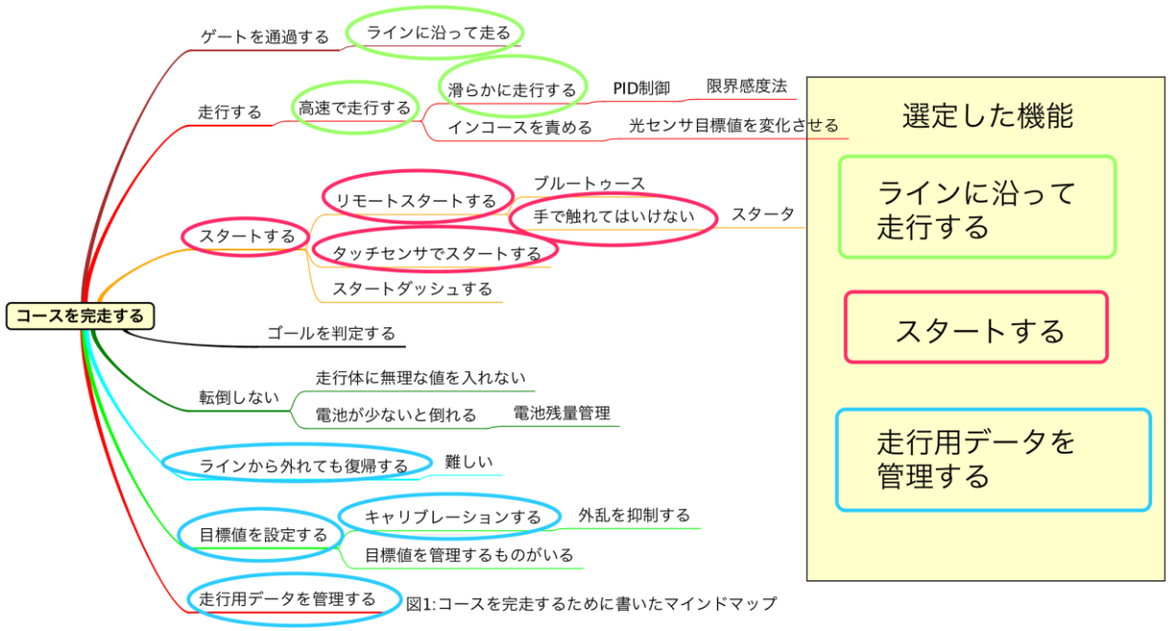


図1:コースを完走するために書いたマインドマップ

1.2 機能の明確化

1.2.1ユースケース図によるシステム機能の明確化

選定した要素を元にユースケース図を作成し、実現する機能を整理した(図2)。機能要求は黄色、非機能要求は青色でマークすることで、機能の明確化を図っている。

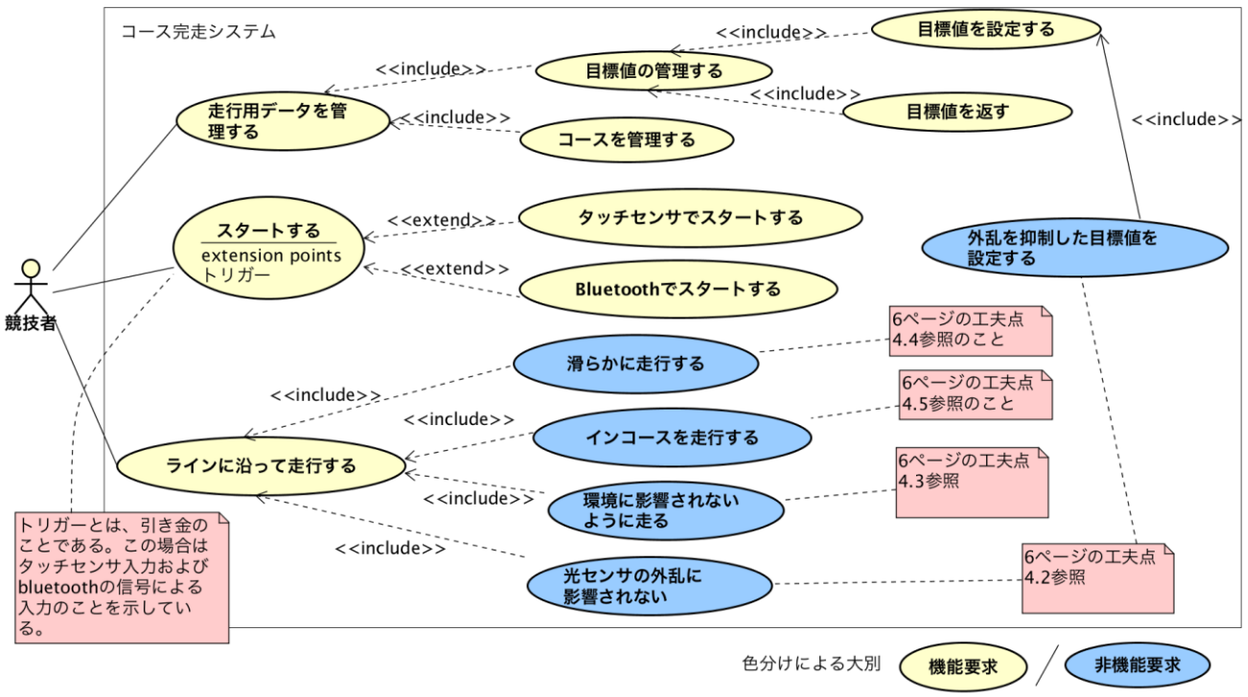


図2：コースを完走するためのユースケース図

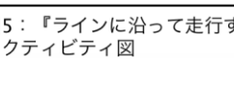
1.2.2イベントフローによる機能の仕様定義

イベントフローを記述することで機能を実現する流れを文章化した(表1)。実際の機能提供の順序は下図のように、「目標値を設定する」から「スタートする」、「ラインに沿って走行する」となる。

表1:コースを完走するためのイベントフロー

ユースケース名	走行用データを管理する	スタートする	ラインに沿って走行する
概要	走行に必要な情報を保存、編集、提供する。	走行体をスタートさせる	各ゲートを通過し、ゴールまで走りきる。
アクター	競技者		
事前条件	走行体が起動している。	走行に必要なデータが設定され終わっている	・目標値が設定されている。 ・走行体が倒立走行するのに適した姿勢になっている。
事後条件	目標値など、走行に必要なデータが設定され終わっている。	走行体が倒立走行に適したものになっている。	ゴールしている。
基本フロー	1.競技者が走行体を起動する。 2.コースの設定を行う。 3.競技者が値取得に適した姿勢を走行体に取らせ目標値を設定していく。 4.ユースケースを終了する。	1．テールを下ろし、三点倒立させる。 2.bluetooth信号待ち状態。 3.bluetooth信号を受信する。 4.走行体を倒立走行に適した姿勢に起こす。 4.テールが倒立走行の邪魔にならないようにする。 5.ユースケースを終了する。	1.ラインを検知する。 2.ラインとの誤差を埋める旋回量を計算する。 3.倒立を維持する操作量を、指定した旋回量、前進量を加味して求める。 3．求めた操作量で走行する。 4.1番に戻る。
代替フロー	なし	2a.bluetoothが受信できない。 2a1.タッチセンサ入力待ち状態。 2a2.競技者がタッチセンサ入力を行う。 2a3.基本フロー4に戻る。	なし
例外フロー	なし	なし	なし

「倒立走行に適した姿勢」とは、走行体が倒立走行する際にとる姿勢にできる限り近づけた状態のことを指す。



1.2.3アクティビティ図による機能仕様の可視化

機能仕様のアクティビティ図を書くことによって処理の流れを可視化した(図3、図4、図5)。

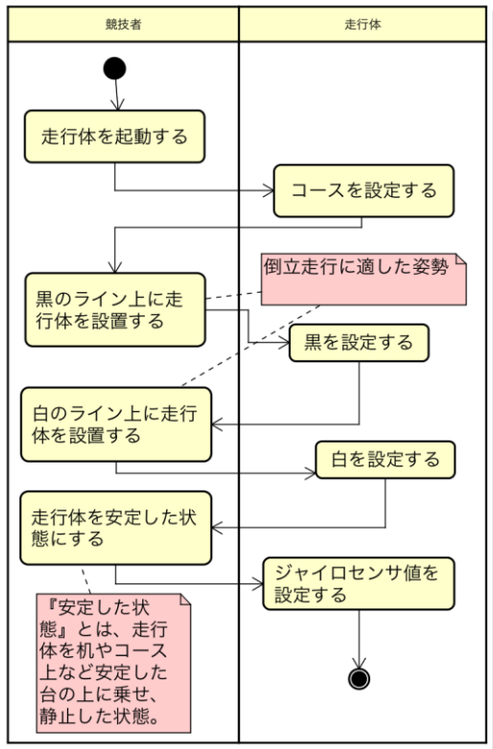


図3：『走行用データを管理する』のアクティビティ図

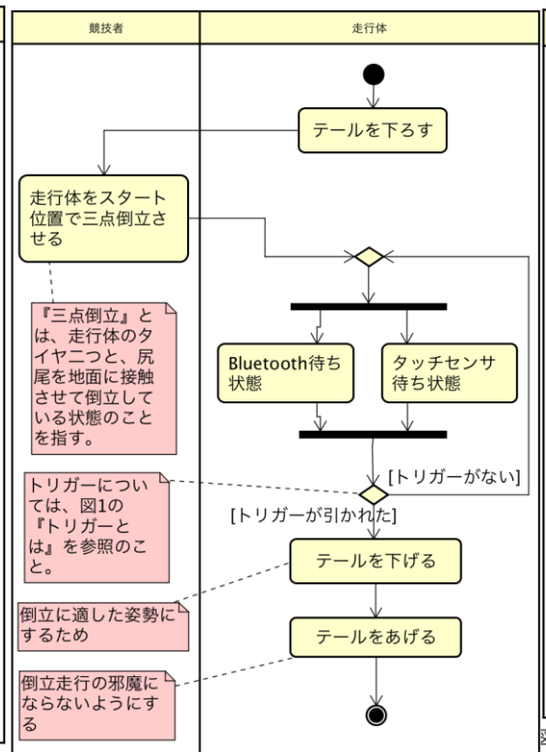


図4：『スタートする』アクティビティ図

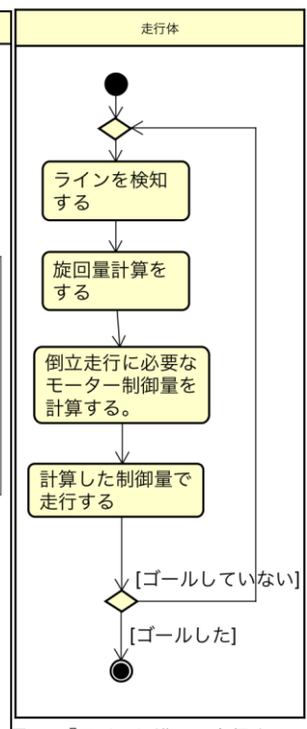


図5：『ラインに沿って走行する』アクティビティ図

2.1依存関係を俯瞰するためのパッケージ図

ここでは後に図示するクラス図の依存関係を示したパッケージ図を下に示す(図6)。また、それぞれの役割を右表(表2)にまとめた。なお、司令部はパッケージではない。高次の目標、『コースを完走する』を達成するためにシナリオ走行部、データ管理部、スタート部を利用するものとしてイメージしやすいように配置している。

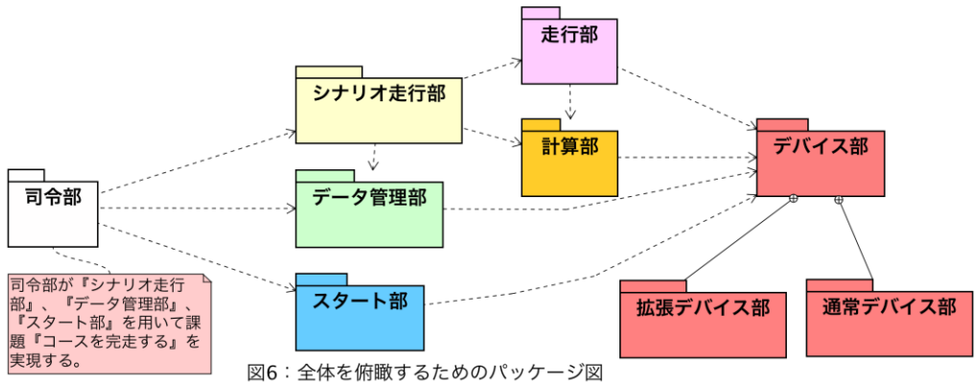


図6：全体を俯瞰するためのパッケージ図

表2：パッケージ図の役割

パッケージ名	役割
シナリオ走行部	データ管理部、走行部、計算部を用いてライントレースを実現する。
データ管理部	光センサ目標値や、ジャイロセンサの初期値など、走行に必要なデータを管理する。
スタート部	タッチセンサ入力信号か、bluetooth信号があった場合、走行体の姿勢を倒立走行に適した姿勢にする。
走行部	モータを制御して走行する。
計算部	様々な計算を行う。
デバイス部	モータを回すなど、基本的な動作を行う。
通常デバイス部	公式の機能を提供する。
拡張デバイス部	通常デバイスの機能を向上させたものや、機能を付加したもの

2.2ユースケースを実現するためのクラス構造

システム静的な構造を可視化ために、以下のようなクラス図を作成した(図6)。

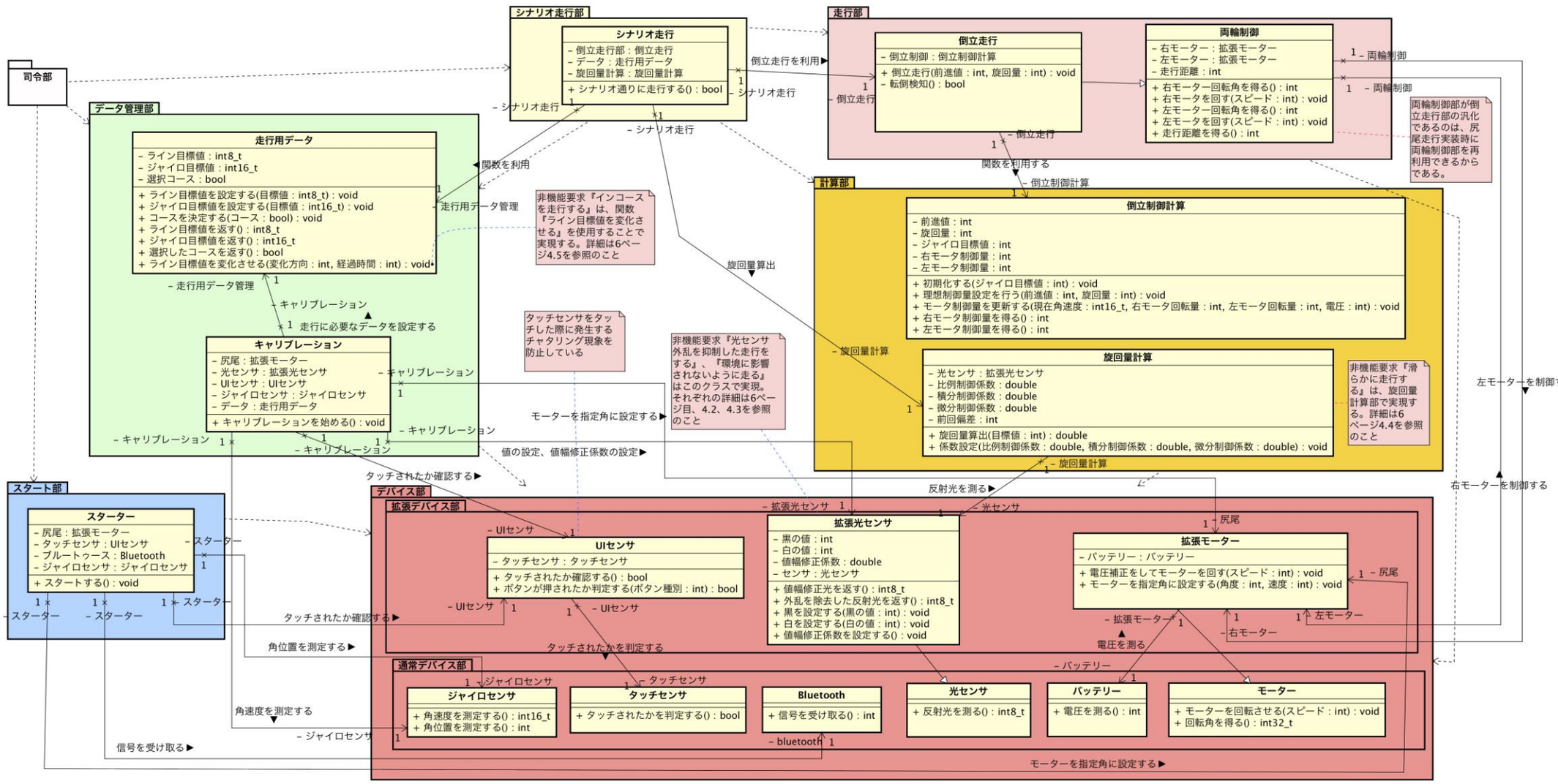


図6：『コースを完走する』を実現するためのクラス図

3.1タスクとサイクルについて

課題『コースを完走する』を達成する上で、作成したクラスのメソッドを実際に実行する関数が必要である。本項目で使用されるタスク及びサイクルについて右表(表3)にまとめた。なお、全体の動作を俯瞰して見ることが可能になる図7の「処理の流れを表すシーケンス図(1段目)」にあるメインタスクとは、課題「コースを完走する」だけでなく、その他難所攻略などを含めた、全体を統括する役割を持つタスクのことである。

表3：タスクとサイクルについて

言葉	定義
タスク	実行処理内容を書く部分。上から順に処理が開始される。ライフライン上では白色に染色した。
サイクル	4msごとに実行する。ライフライン上では灰色で染色した。

3.2全体の振る舞い

課題「コースを完走する」を実現する上で大まかな流れを相互作用の利用を用いて記述した(図7、8、9)。横長になったため、三段に分けて記述した。適切な機能の分割により、高次の課題解決に際して、内部の細かい動作を気にしない抽象的な命令で問題解決に当たれることがわかった。

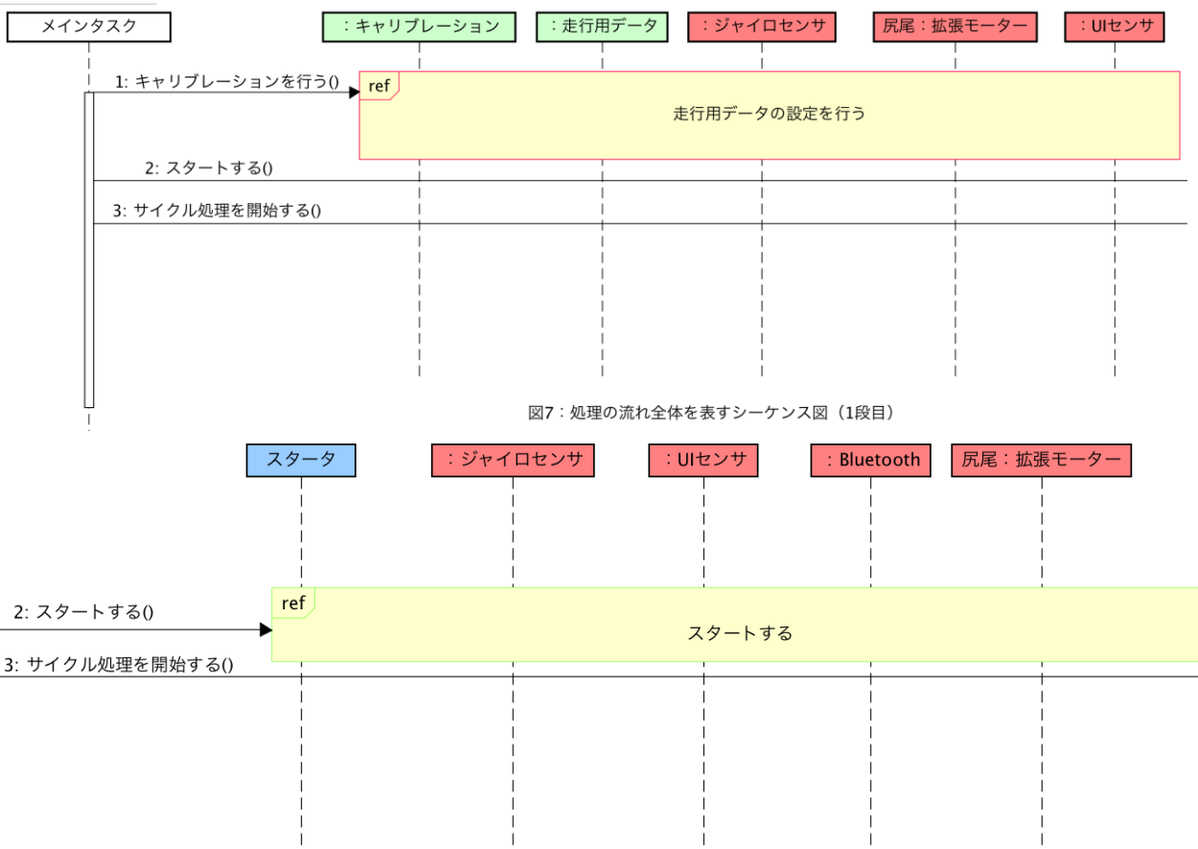


図7：処理の流れ全体を表すシーケンス図（1段目）

図8：処理の流れ全体を表すシーケンス図（2段目）

図9：処理の流れ全体を表すシーケンス図（3段目）

3.3相互作用の利用の詳細

3.2.1走行用データの設定

図7で示した相互作用の利用、「走行用データの設定を行う」を示した（図10）。

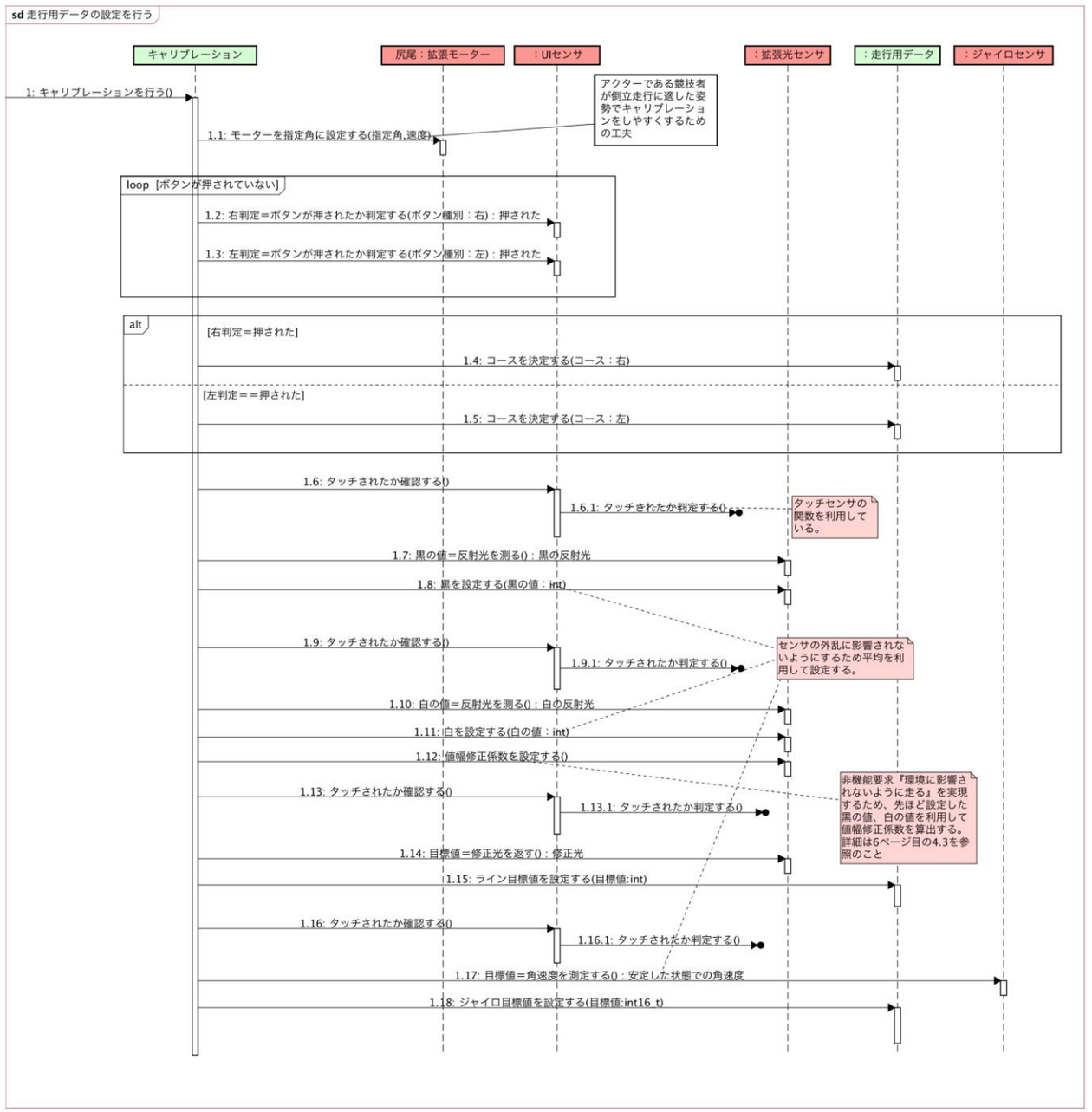


図10:『走行用データの設定を行う』シーケンス図

3.2.3スタートする

図8で示した相互作用の利用、「スタートする」を下図(図11)に示す。

また、機能「スタートする」は状態により動作が変化することから、ステートマシン図を用いることでより理解が深まるため、以下に図示した(図12)。

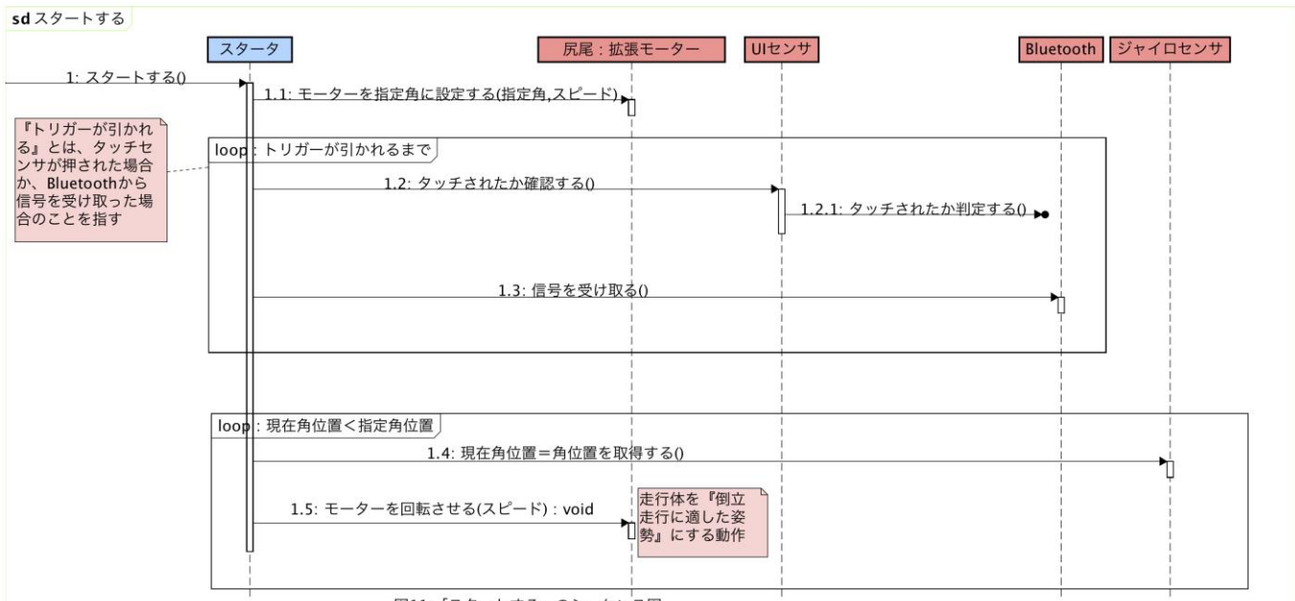


図11:「スタートする」のシーケンス図

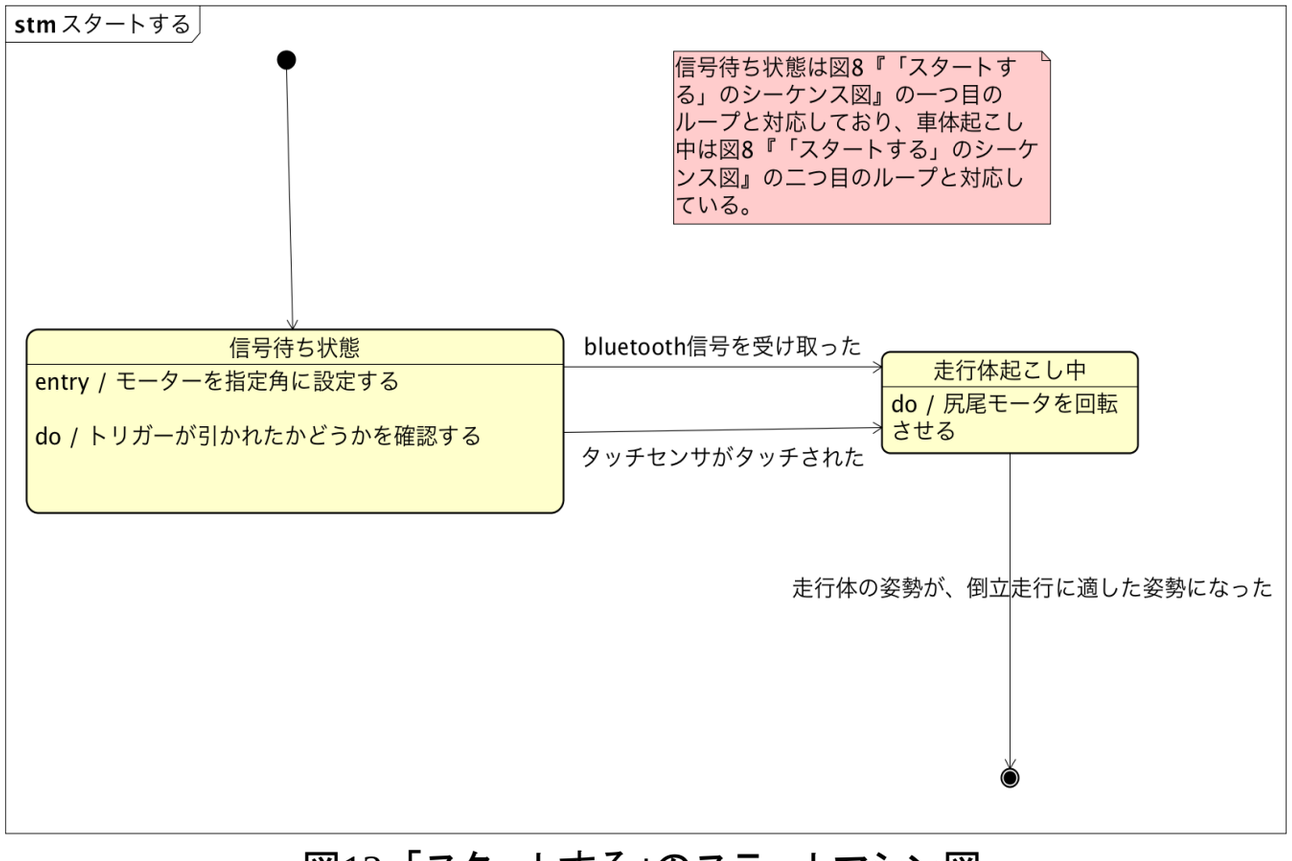


図12:「スタートする」のステートマシン図

3.2.4シナリオ通りに走行する

図9で示した相互作用の利用「シナリオ通りに走行する」を下図(図12)に示した。

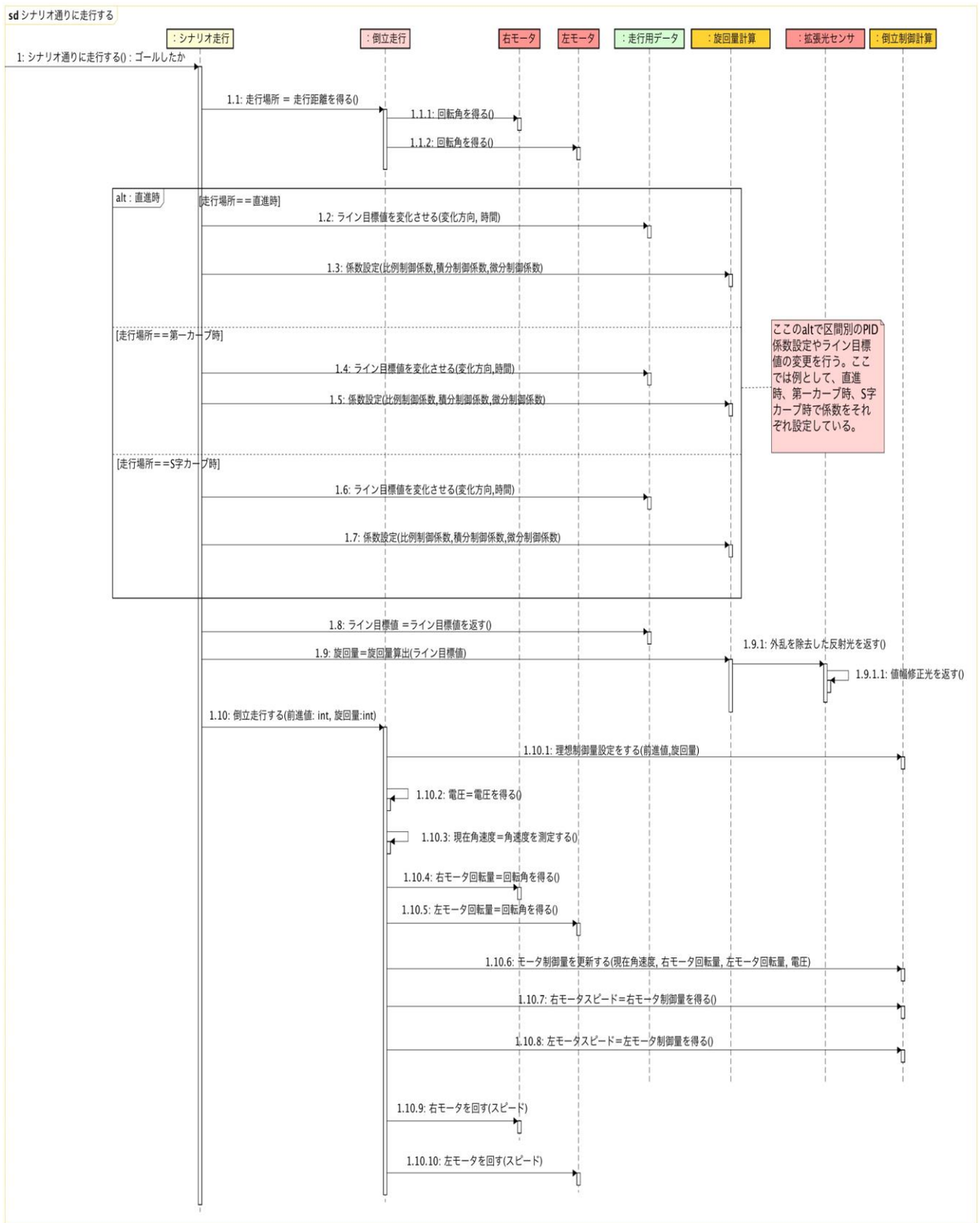


図13:「シナリオ通りに走行する」のシーケンス図

4.1工夫点概要

本項目では、機能モデルの項目で明らかになった非機能要求を実現するために、どのような方法を用いたのかを記述する。

4.2光センサの外乱に影響されないように走る

走行体に使用されている光センサの取得する値には高周波ノイズが混じることがある。光センサで取得できる値をそのまま走行に使用した場合、こうしたノイズは無駄な走行を発生させたり、走行体がコースアウトしたりする原因になる。

対策として、値設定時の補正には平均、走行時の補正には移動加重平均を用いた（下式）。

nは要素数、iは何個目の要素か、wは重み、xは実際の値を表す。

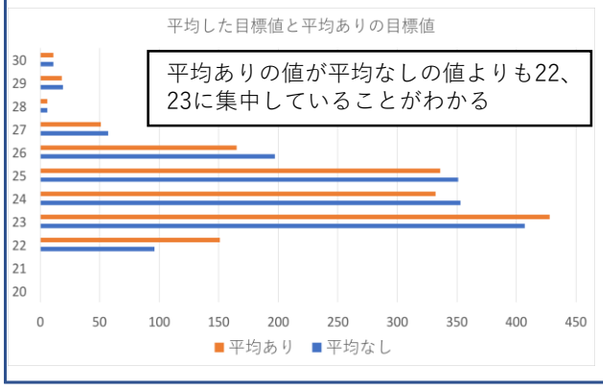
平均の式： $y = \frac{1}{n}\sum_{i=1}^{n} xi$　　移動加重平均の式：$y = \frac{\sum_{i=1}^{n} xi * wi}{\sum_{i=1}^{n} wi}$

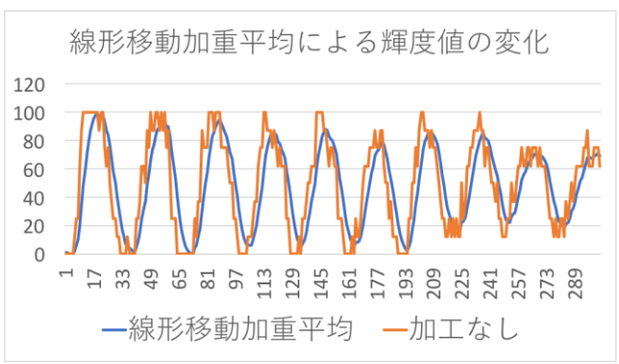
4.2.1目標値設定時に発生する光センサノイズへの対策

目標値設定時には、平均を用いることでノイズ除去を行った。平均yは、１番目の要素x1からxnの総和を要素数nで割った値である。これを使用することでノイズを除去した目標値を設定することができる。左下のグラフを見ていただければその違いがわかる。

4.2.2走行時に発生する光センサノイズへの対策

走行時に発生する光センサノイズ除去には、線形移動加重平均を用いた。移動加重平均は、xiとwiをかけたものの総和を、重みの総和で割ったものである。線形移動加重平均と加工なしのデータを比較した右下のグラフの通り、ノイズを除去することに成功した。





4.3環境に影響されないように走る

照明の種類や強さによって光センサが取得する値は変化する。ラインをトレースする際に算出される旋回量はこの光センサの値に依存している。つまり、開発中の照明でうまくライントレースしていたとしても、大会の照明が場合によっては走行に悪い影響を及ぼし、コースアウトしてしまう可能性がある。

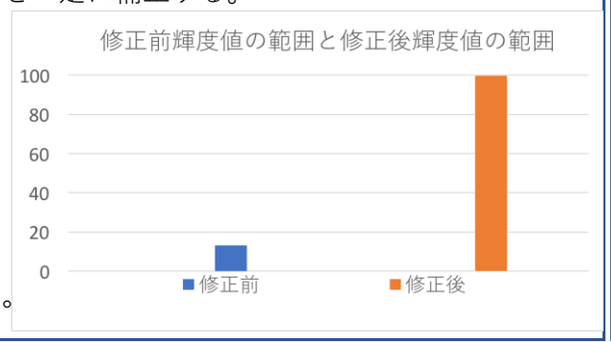
こうしたリスクを除去するために、光センサの値幅を一定に補正する。

補正には以下の式を用いた。

$$\frac{Dmax - Dmin}{Vmax - Vmin}(f - Vmin) + Dmin$$

※Dmaxは補正後の最大値、Dminは補正後の最小値、Vmaxは実際の最大値、Vminは実際の最小値であり、fは現在の輝度値である。

修正後の値の値幅が修正されていることを確認した。



4.4滑らかに走行する

『ラインが白になったら右に旋回し、ラインが黒になったら左に走行する』、というような俗に言うONOFF走行では、無駄な走行距離が発生する上に、安定感も損なわれる。

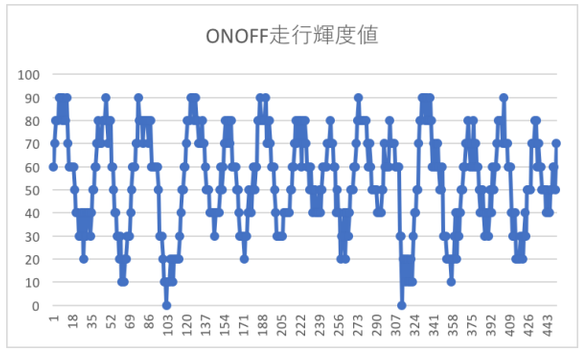
これを解決するために、PID制御を用いて滑らかな走行を実現する。

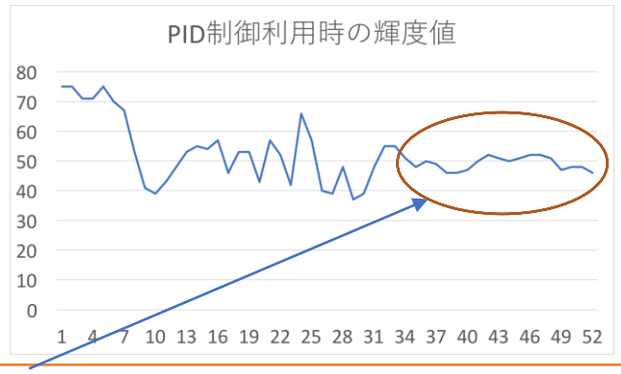
PID制御の式を以下に示す。

$$Y = Kp * e + Ki * \int e\,dt + Kd\frac{de}{dt}$$

※Yは旋回量、Kpは比例ゲイン、Kiは積分ゲイン、Kdは微分ゲイン、eは目標値との偏差を表している。

ONOFF走行では光センサの値が振動していることが見て取れるが、PID制御実装後には振動が収束している。





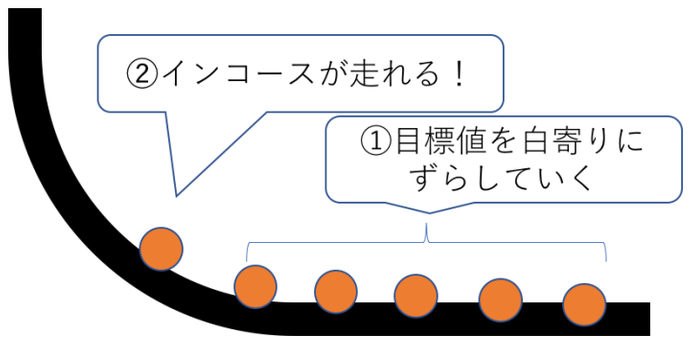
光センサの取得輝度値が目標値付近で収束していることがわかる

4.5インコースを走行する

走行する際にどの程度の輝度値を目標値とするかによって、走行体が走行する場所をある程度指定することができる。カーブの内側を走行するように目標値を変化させることによって、走行タイムの短縮、コースアウトの危険性の低減が見込める。

目標値の管理を担当する走行用データクラスに実装した。クラス図の関数では『ライン目標値を変化させる』に該当する。実装にあたっての処理の流れをアクティビティ図で示す（右下の図）。この処理はサイクルで回すことを想定している。

急激な目標値の変化は走行体が振動してしまう原因になるため、徐々に値を変更している。



※オレンジの円は光センサのライトを示す。