

森のくまさん

選択課題

: コースを完走する。

目標

: どんなコースでもゴールできる走行を実現する。

要件定義と開発目標

要件を分析・定義し、開発目標を設定した。

要件

A.
安定したスタートをする。

B.
コースを走行してゴールまで完走する。

B'.
高速かつ安定した走行をする。

コースを完走するためには、まずは「スタート」しなければならない。言わずがな、競技最初の重要ポイントはスタートである。しかも、ただ倒立走行&ライントレースをすればいいのではない。成功確率とロス少なさを合わせた「安定した」スタートが必要である。

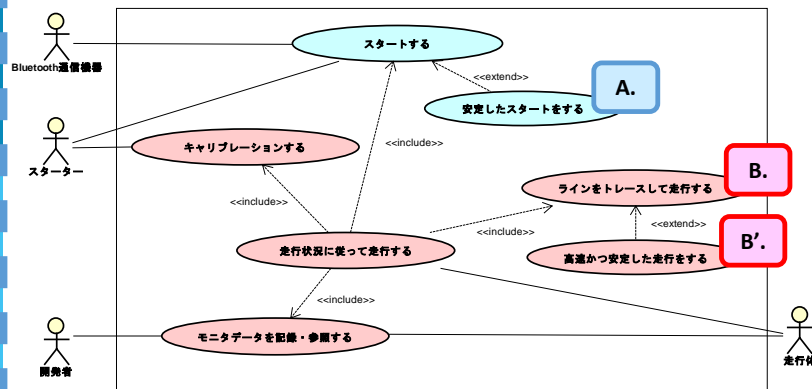
また、設計思想である「どんなコースでも変わらない走り」に掲げられた「コース」とは「ライン」そのものである。よって必然的にライン上をトレースしてゴール（完走）を目指す。

実際のところ、設計思想そのものの実現については「低速かつライン追従性」を求めれば可能であるが、競技という性質上、スピードが求められる。「スピードと安定性」という、相反する要素をバランスさせての実現が理想であり、これも重要な要件である。

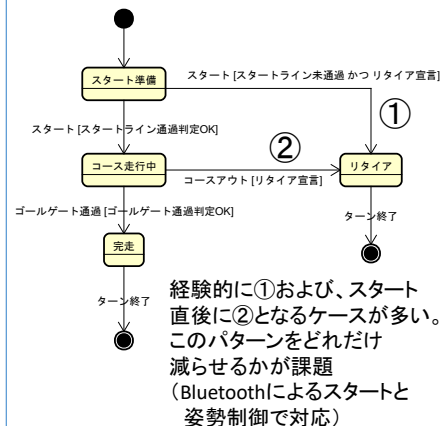
走行戦略



要件実現のためのユースケース



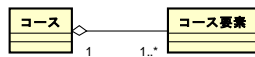
A. 安定したスタートをする。



経験的に①および、スタート直後に②となるケースが多い。このパターンをどれだけ減らせるかが課題（Bluetoothによるスタートと姿勢制御で対応）

B. コースを走行してゴールまで完走する。 （「ラインをトレースして走行する」と同義）

コースの構造



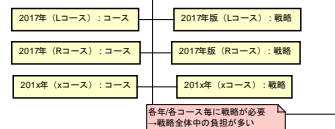
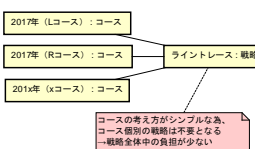
どんなコースでも変わらない走りを実現する為、個別対応を行わない。
（前提：コースの定義は変わらない事）

コースの捉え方による戦略の違い

「私たちの考えるコース」
・ライン以外はコースと見ない
・よってラインを外れるショートカットも存在しない
・各ラウンドのターン毎に走るラインは1本に定まるのでL/Rの区別は持たない
（ライントレース部分はL/Rコースで共通）

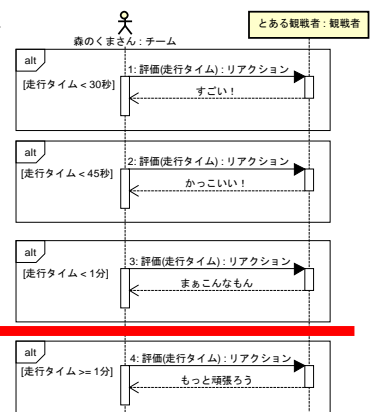
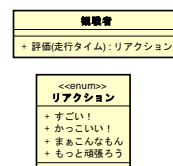
L/R区別無し（ラインのみ）：コース
ライン：コース要素

（参考）別の考え方のコース
Lコースライン：コース要素
Rコースライン：コース要素
ライン外：コース要素



B'. 高速かつ安定した走行をする。

観戦者のリアクション



- 観戦者からのリアクションを妥当なものにするべく、目標タイム（ボーダーライン）を設定（タイムはレプリカコース到着初期の 試走タイム（1分8秒）を参考に設定）
- 実現については走行スピードとライントレースのPIDパラメータ動的制御で対応

森のくまさん

機能

走行体が備える機能を、以下のように整理した。

安定スタート

Bluetooth通信

姿勢制御

キャリブレーション

キャリブレーション

ナビゲーション

モニタデータの集積

走行状況の管理

ライントレース走行制御

PID制御

スピード制御

機能



安定したスタートに必要な機能

要件 A.

Bluetooth通信

Bluetoothの通信を行う。

①スタートの合図を、走行体で受信する。

タッチセンサでのスタートは、スターターが外乱となってしまう、安定性に欠く

尻尾の制御・モーターの駆動

起立した状態から、

②尻尾を回し、走行体を前方に傾ける

③モーターを駆動し、前方へ直進する

④後方へ傾かない程度に加速する

※ 大きなパワーをかけて発進しようとする、反動で頭頂部が大きく振れる。
→ さらに反対側に傾けてバランスを取ろうとする。
→ かえってスピードが出ない。
→ 小さすぎるパワーでは、その場で静止してしまう。

発進に1回のモーターの駆動だけでなく、細かい動作単位に分け、状態遷移による制御を行う

Bluetooth通信



姿勢制御

安定スタート

ライントレース走行に必要な機能

要件 B.

要件 B'.

ライン判定／走行体状態監視

どんなコースでも環境光の影響は無視できない（白：100、黒：0の設定では走れないケースの方が多い）
よって、スタート前に**キャリブレーション**を行い、「しきい値」を適切なものに調整する。

キャリブレーション

キャリブレーション

走るコースの走行情報を持たない分、走行中の各種センサー／モーターの状態から走行体の走行状況を的確につかみ、安定した走行へ**ナビゲーション**する必要がある。

モニタデータの集積

走行状況の管理

ナビゲーション

安定した走行の継続

事故（転倒／コースアウト）の原因となるもの

急ハンドル

急アクセル

急ブレーキ

これらの挙動は厳禁！

ライントレースの際の旋回角度（ハンドルの切れ角）を求めるために、フィードバック制御（PID制御）を行う。

PIDゲインは、走行中情報から求める。また、走行体のスピードにより、PIDそれぞれのパラメータを切り替える。

走行中にスピードを制御し、急加速、急減速をしない。

走行体の前後のバランスを大きく崩すことなく走行し、転倒のリスクを減らす。

また、前後の振幅を少なくすることでパワーの無駄遣いを減少させる。

PID制御



スピード制御

ライントレース走行制御

森のくまさん

安定スタートのふるまい

安定スタートは、スタート待機から次のステップとして起動される走行クラスである。内部状態は主に経過時間で遷移することで、より細かな動作を定義する。

※走行体個体によるセンサー値やモータの出力パワーの差異は、実機によるテストを行うことにより、パラメータリストとして、設定できるように実装する。

走行およびライトレースのふるまい

1. キャリブレーション

スタート前にキャリブレーションを行い、白地および黒ラインの輝度を測定する。これをもとに、ライトレースするためのラインのエッジを検出するための「しきい値」を計算式によって求める。

2. ナビゲーション

ドライバに対して、ナビゲータは走行状況を管理・提供する。その情報をもとにドライバは走行クラスを選択し、走行を指示する。走行とともに、走行(モータの回転角度)、ジャイロセンサー値を記録する。(ライトレースの場合は光センサーの輝度も記録する。)これらのモニタ情報は集積され、一定期間の最小・最大・平均など統計情報を提供する準備を整える。

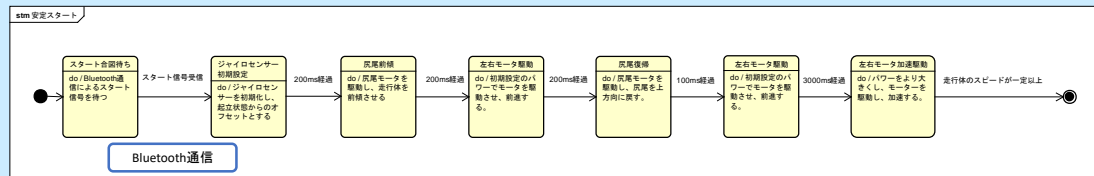
3. ライトレース

現在、直線に近い走行か、または、カーブに沿った走行なのかによって、走行スピードを決定する。ナビゲータ経由で取得した輝度情報をもとに、走行体の左右の向きを決定する。その角度はPID制御によって動的に決定する。また、そのPID制御のゲインは、走行スピードによって変更される。

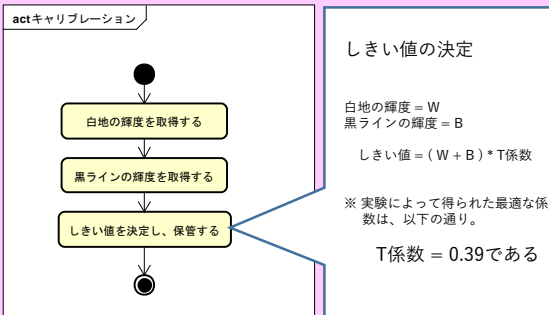
ふるまい



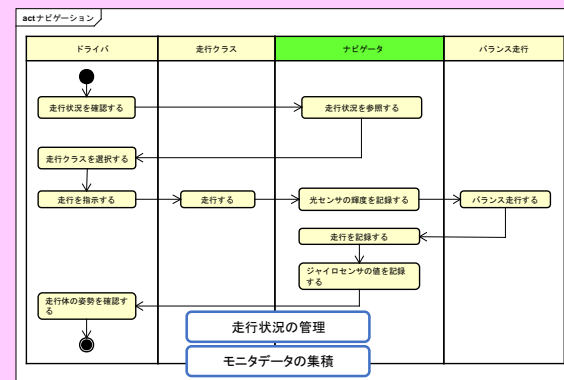
安定スタート



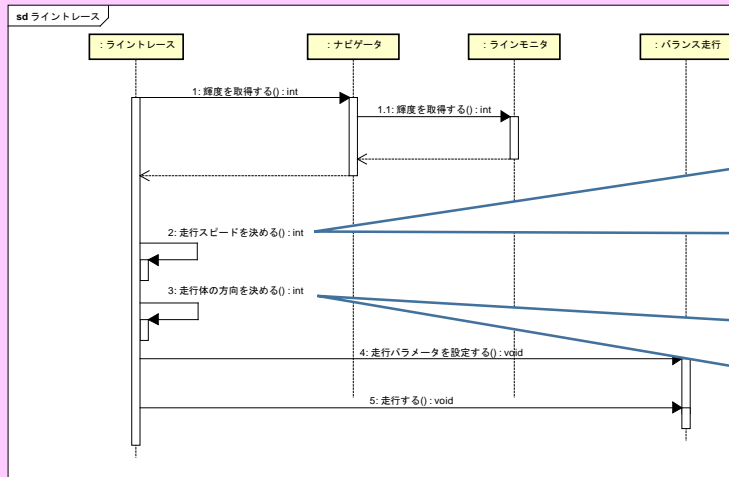
キャリブレーション



ナビゲーション



ライトレース



スピード制御

直線ではスピードを速め、カーブに差し掛かったところで減速する。

前回の周期までのハンドル操作(左右の方向)と、現在の向きとの差分をとる。

$$\text{速度} = (100 - \text{差分}) * S\text{係数} * \text{標準速度}$$

※ S係数 = 0.2, 標準速度 = 50 である。ただし、これはモータ駆動に対する、パワーとして用いられるため、現実の走行速度とはギャップを生じる。
※ 駆動するモータの回転角をモニタすることにより、実速度との差を吸収するものとする。

PID制御

比例動作 = P、積分動作 = I、微分動作 = D とし、

これらのゲインの元になる情報を、走行する速度に対応する形で、配列に格納する。速度に対応したゲインを配列に取り出し、PID制御値を求め走行体のハンドル操作(左右の方向)を決定する。

森のくまさん

スタート時の初速(モータへのパワー)について

スタート時のスピードによって走行体の挙動が大きく変化する。

モータへのパワーを Forward (=10, 20, 30 ...100) とし、スタート時点から、ジャイロセンサー値をモニタする実験と検証を行った。

- 床面は、摩擦を考慮しできるだけ競技と同様の布を敷く。
- ライントレースを行わない。
- 尻尾はスタート直後に跳ね上げ、2輪走行を行う。
- 倒立振りライブラリにより、走行体のバランスをとる。

実験の結果、安定性およびゴールまでの時間を考えると、Forward=30~50の範囲で設定し、尻尾による補助動作により、前方への移動や揺り返しを軽減できることを確認した。

ライン検出のしきい値について

キャリブレーションを行うことによって、ラインのエッジを検知するための「しきい値」を決定する。

光センサーを用い、環境光および白地の輝度、黒ラインの輝度を測定し、しきい値を設定した上で、ライントレースを行う実験を行った。ラインの進行方向に対して左側のエッジを認識する。設計時に算出した T 係数を用いたしきい値では、ラインの中心に近い部分(より暗い部分)をエッジとみなすため、グレーの部分を白色とみなし、ライントレースができなかった。

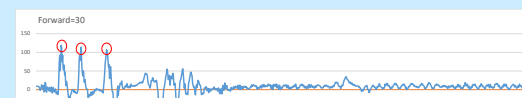
そのため、T 係数にさらに補正値を加え、しきい値を設定することとした。

実験と検証

スタート時の挙動 尻尾による補助動作なし (ジャイロセンサー値の変動 (スタートから4秒後まで))



Forward=10
ジャイロセンサー値の変動は小さいが、周期的に+-を行き来する。安定しているが、モータ駆動に「遊び」が生じている。



Forward=30
スタート直後に前方に傾く動作に続き、約1秒後後方への「揺り返し」が発生している。安定するまで3秒程度要する。



Forward=50
前方への傾きが小さいが、「揺り返し」が比較的大きい。

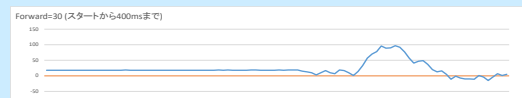


Forward=70
スタート直後から、前後への揺れが大きく、4秒経過しても安定しない。

スタート時の挙動 (ジャイロセンサー値の変動 (スタートから400ms後まで))



Forward=30
尻尾による補助動作なし



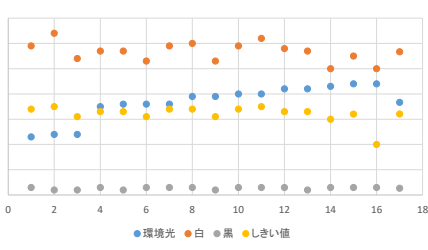
Forward=30
尻尾による補助動作あり

尻尾による補助動作を行うことによって、ジャイロセンサーの値の変化が滑らかになった。 → 走行体の前後の揺れが減少した。

ライン検知のしきい値について

No	環境光	白	黒	灰色(参考)
1	23	59	3	28
2	24	64	2	27
3	24	54	2	27
4	35	57	3	28
5	36	57	2	27
6	36	53	3	28
7	36	59	3	28
8	39	60	3	28
9	39	53	2	27
10	40	59	3	28
11	40	62	3	28
12	42	58	3	28
13	42	57	2	27
14	43	50	3	28
15	44	55	3	28
平均	36.2	57.1	2.7	

キャリブレーションデータ

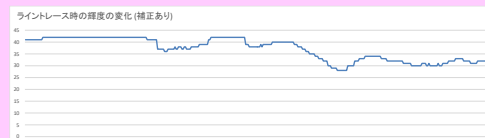
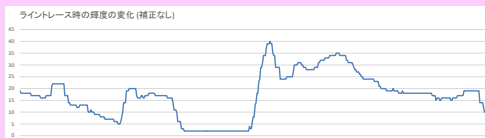


分析

- 環境光が変動しても、黒ラインの輝度はほとんど変化しない。
- 白地の輝度を測定する時、走行体の傾きにより大きく変動する。
- T 係数 (=0.39) は、テスト環境で求めた値であった。これを用いてライントレースを行うと、ラインのより中心に近い部分をエッジとみなす。安定したトレースを行うが、グレーラインに差し掛かると、左エッジを誤認識し、右側に大きく方向を変えることを確認した。

※ しきい値の設定には補正が必要である。

しきい値の補正について



補正値を加えることにより、ライン検出の輝度の値が全体的に上昇、光センサー値の変化が滑らかになった。 → 滑らかな走行になった。

