

1. 走行戦略(要求分析)

芝浦雑伎団

オービス総研

走行戦略

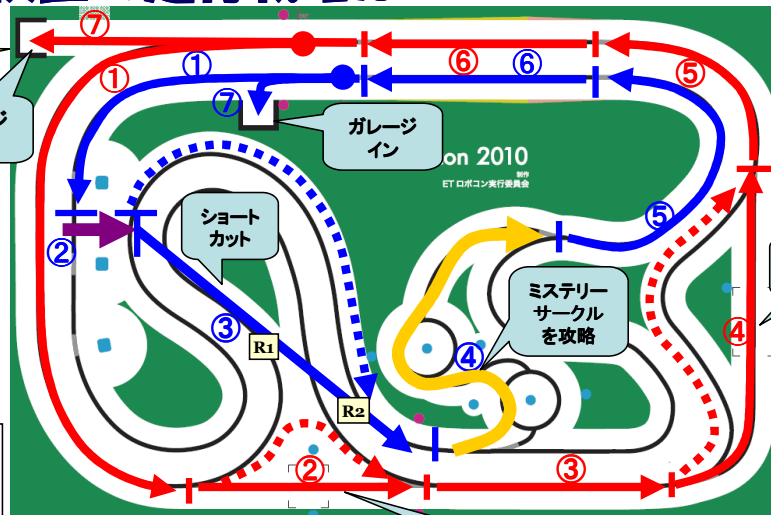
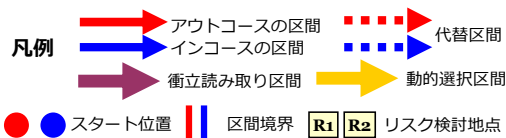
今回の芝浦雑伎団の走行戦略は！

全難所クリア
+早くゴールする！

よって今回は
右のようなコース(*)で走行！

※コースは区間を順番に組み合わせたものを指す。区間の組み合わせが違えば(代替区間を走る場合)別のコースとして扱う。

- 衝突を読み取る部分は衝突読み取り区間とした。
(②エニグマデコーディング部分)
- 走行開始後、走行する区間が変わる部分を動的選択区間とした。(④ミステリーサークル部分)



走行戦略上必要になる
機能を抽出

シーソーを
攻略

走行戦略を実現する
振る舞いを抽出

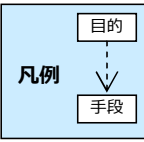
ゴールモデルによる機能抽出

Point ゴール(目標)から具体的にシステム
が実現すべき機能を抽出。

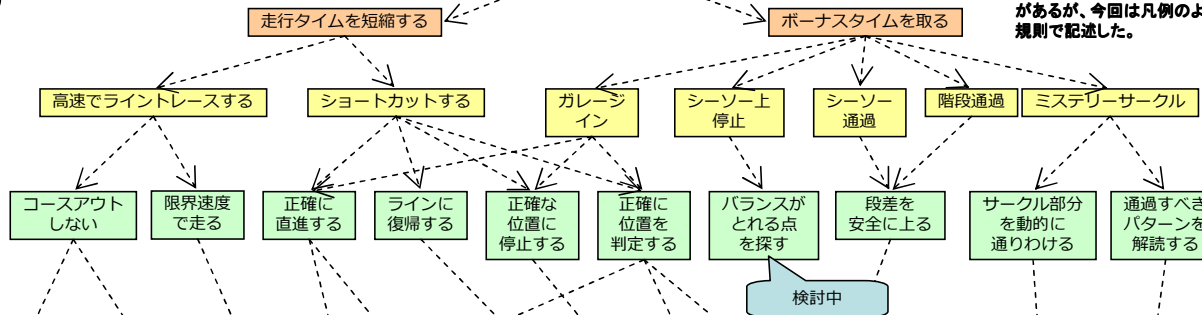
機能

競技部門優勝

チャンピオンシップ
大会の目標



※ゴールモデルは様々な記法
があるが、今回は凡例のような
規則で記述した。



高性能を発揮するための機能

走りの切り替えに必要な判定群
右UC・基本フローの8に関連

動的選択区間の実現
右UC・代替フローに関連

実現すべき機能

リスク

インコースの③走行時、衝突の可能性はある

衝突の可能性を以下のようにリスク分析した結果、
このリスクは保有しても十分に安全だと判断した。

1. 倒立制御の限界まで速度を上げて走行
するので、接触することは少ない。

実測では、アウトコースの走行体が衝突可能性があるR1に到達するまで
16秒なのに対して、インコースでショートカットする場合16秒でR2に
到達する。

よって、今回は衝突する可能性は少ない。

2. チャンピオンシップでは高レベルな戦いになるためシー
ソーを敬遠するチームは少ない。

(東海大会、南関東大会からチャンピオンシップに出場するチームでは、
11チーム中10チームがシーソーを成功させている。)

よって、インコース③のショートカットを採用する(※1)

(※1) 当日危険と判断した場合の代替区間は左図の青破線区間となる

リスク

当日のシーソー、階段の成功率が低い

当日試走時に3Dの難所部分での成功率が低い場合、代替区間を走行さ
せる(※2)

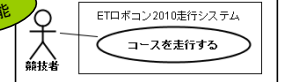
(※2) 当日危険と判断した場合の代替区間は左図のシーソー、階段脇の赤破線区間となる

ユースケース

Point

スタートから走行終了までの、
外から見た振る舞いを記述。

機能



振る舞い

ユースケース名	コースを走行する
基本フロー	1. アクタはコースを選択する。 2. システムは選択されたコースを設定する。 3. アクタはキャリブレーション(※1)を開始する。 4. システムはキャリブレーションする。 5. アクタは走行体を立てて置き、走行開始を指示する(※2)。 6. システムは倒立を開始する。 7. システムは設定されたコースに従って区間を走行する。 8. システムは区間の終了を検知し、次の区間に移る。 9. システムは次の区間がなくなるまで7と8を繰り返す。 10. システムは走行体を停止させる。
代替フロー	基本フローの7で、走行する区間が衝突読み取り区間の場合： 1. システムは走行しながら衝突の有無を確認する。 2. 基本フローの8に戻る。 基本フローの8で、次の区間が動的選択区間の場合： 1. システムは走行すべき区間を選択する。 2. 基本フローの7に戻る。
例外フロー	走行体が走行中に転倒した場合： 1. システムは走行を中止する。
備考	(※1) 白・黒・灰の各色の基準値を光センサで取得し、保存する。 (※2) タッチセンサの押下による。

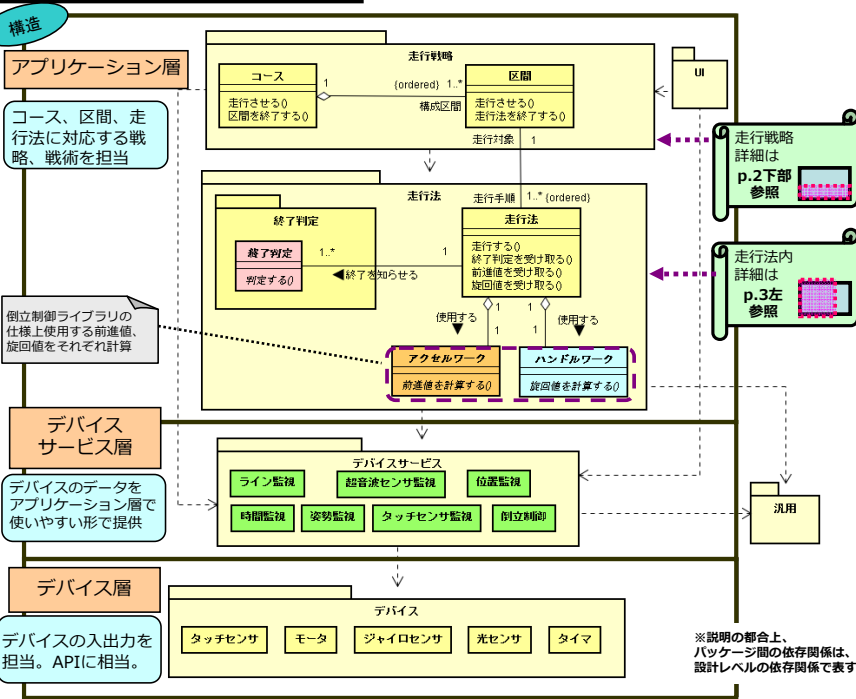
2. 全体構造と走行戦略(分析)

芝浦雑伎団

オーグス総研

全体構造と基本パターン

ゴールモデル、ユースケースで抽出された機能に対する実現を構造と振る舞いの面から検討。



Point 1 層、パッケージごとに責務を分割。責務が明確化！！

部品化が容易に

Point 2 依存関係を単方向に徹底！！(設計レベルでは)

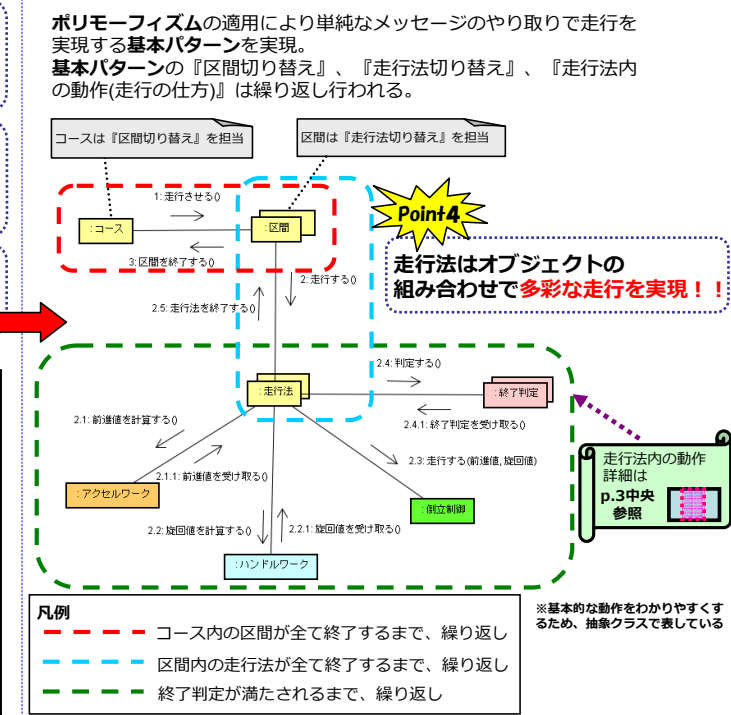
変更時の影響範囲が限定的に

Point 3 ポリモーフィズムの適用を可能にし、単純なメッセージのやり取りで走行を実現！！

パッケージの概要

パッケージ	主責務
走行戦略	区間切り替え、走行法切り替え、走行パターンの動的選択を担当
走行法	走行の仕方を担当
終了判定	走行法を終了させる条件の判定を担当
UI	コース選択、キャリブレーション、スタート指示を担当
デバイスサービス	デバイスからのデータを上の層で使いやすい加工を担当
デバイス	実行環境が提供する入出力機能を担当
汎用	基本的なデータ構造やアルゴリズム

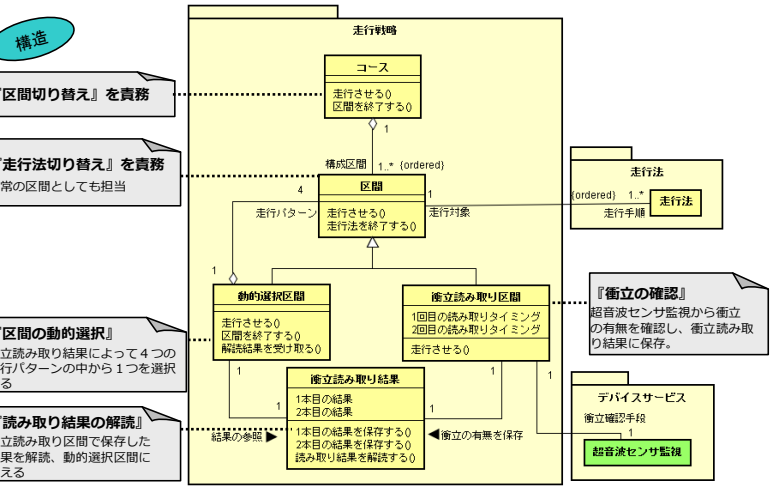
基本パターン



走行戦略の構造と振る舞い

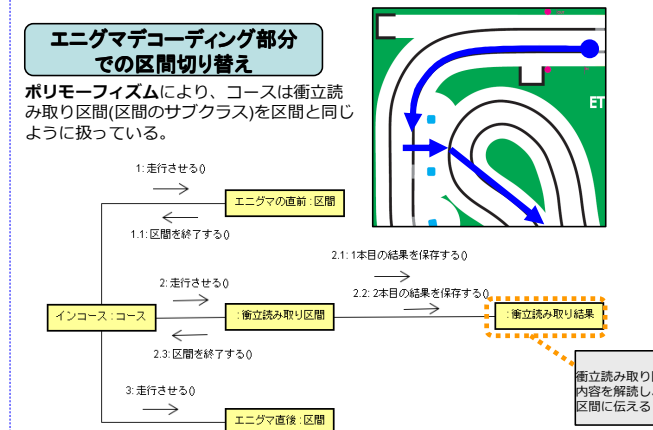
走行戦略の責務は『区間切り替え』、『走行パターンの動的選択』、『走行法切り替え』。エニグマデコーディング、ミステリーサークルの対策はこのパッケージで実現。

p.1の走行戦略で決めたコースと区間の関係を構造として抽出。区間にポリモーフィズムを適用することで特殊な走行区間も通常の区間と同じように扱える。



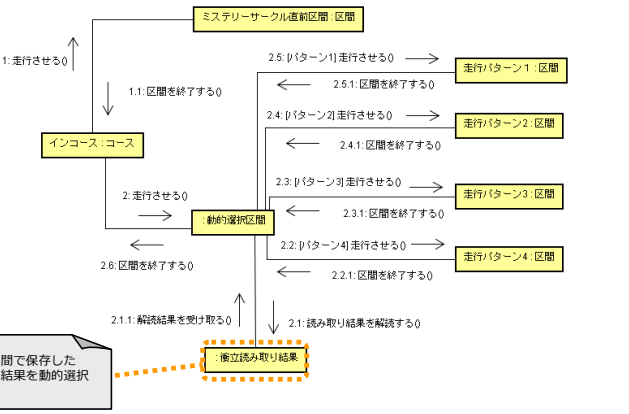
振る舞い 走行戦略の振る舞い例

振る舞いの例として『エニグマデコーディング部分での区間切り替え』、『ミステリーサークルでの走行パターンの動的選択』を示す。



ミステリーサークルでの走行パターンの動的選択

ミステリーサークルでは動的選択区間が衛立読み取り結果を元に、走行パターンを動的に変更。その後動的選択区間は走行パターンの代理として振る舞う。



3. 走行法と並行性検討

芝浦雑伎団

オービス総研

走行法の構造と振る舞い

走行法は『走行法内の動作』を責務とする。

走行法に終了判定を加えることにより様々な条件で、切り替えが可能。またそれらを部品化、ポリモーフィズムの適用により、組み合わせによって多彩な走行を実現。

Point 走行をアクセルワーク、ハンドルワークと終了判定に分解して部品化。

具体例は
p.4上部
参照

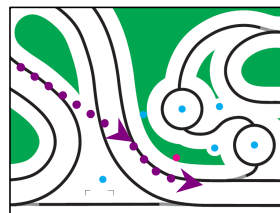
振る舞い

走行法切り替え

Point

走行法の終了判定が成立すると、
区間は次の走行法に切り替える。
終了判定は複数指定できる。

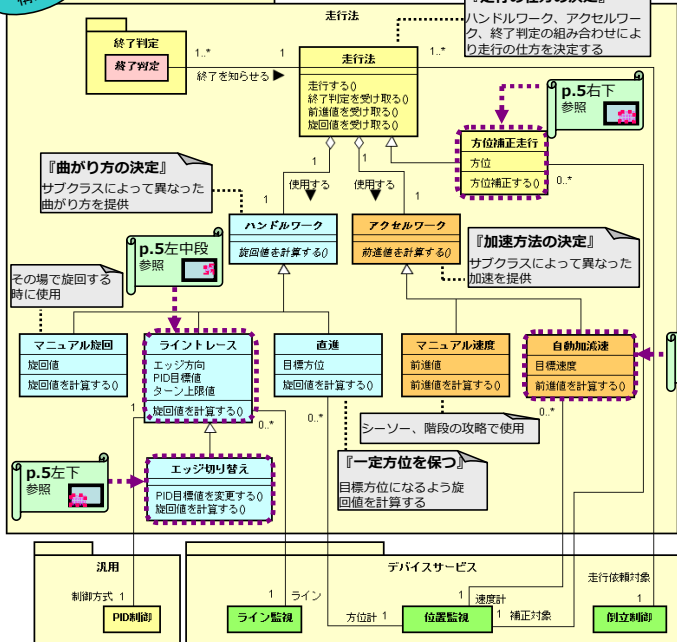
例) ショートカットからラインに復帰するとき(右図)の相互作用



凡例

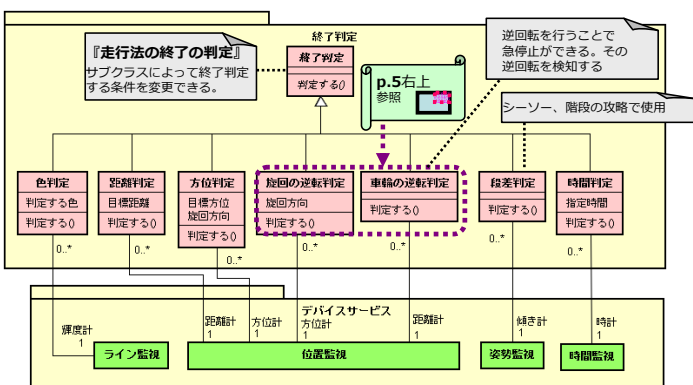
- アクセルワークとハンドルワークのサブクラスをポリモーフィズムで利用。これらのクラスから前進値、旋回値を計算している
- 終了判定のサブクラスをポリモーフィズムで利用。複数の終了判定を走行法が持っている場合、全ての終了判定が満たされた場合のみ(AND条件)走行法は終了する
- 終了判定が満たされるまで、繰り返し

構造



終了判定の構造

終了判定は走行法が終了する条件を判定することを責務とする。



並行性検討

Point

実測処理時間から、最低限のタスク分割を行う。

超音波を使う上で、超音波が返ってくるまでの時間があるので、タスク分割の必要性があると考えた。

タスク分割を検討する上で、検討対象を以下のようにした。

1. 緊急性の高い処理 → **優先度高**
2. 処理時間が長い処理 → **優先度低**
3. 処理周期の頻度が少なくてもよい処理 → **優先度低**

検討する処理として、以下の4つの処理が挙がった。

倒立制御: 最悪間に合わない恐れがあり、最優先処理となる。また倒立制御のライブラリが4msで制御されること前提で設計されている為。
→1に該当、**優先度高**

アプリケーション層からの制御指示: センサ系の値を元にシステムに対して適切な指示を素早く出した為。
→1に該当、**優先度高**

超音波の障害物検知: 超音波センサのドライバが照射指示から、結果取得まで5msかかる。これは倒立制御の前提周期より長い為、倒立制御と別タスクとなる。また、波の干渉を抑えるため40ms程度に1回の処理でよい。
→2、3に該当、**優先度低**

タッチセンサ、ボタンの監視: 人間がボタンを押してから、違和感なく動作すればよい。
→3に該当、**優先度低**

以上より処理を割り当てた結果を以下に示す

タスク名	走行タスク	補助タスク
タスク優先度	優先度高	優先度低
周期	4ms	40ms
主な役割	倒立制御 アプリケーション層からの制御指示	超音波の障害物検知 タッチセンサの監視、ボタンの監視
タスクの平均処理時間(※1)	1ms以下	5ms(※2)
タスクの最長処理時間(※1)	2ms	5ms(※2)

(※1)インコース一周を行ったログ5本分の結果
(※2)優先度高のタスク処理と合計の処理時間

上記割り当て時、優先度低のタスクの最長待ち時間は、

補助タスクの最長処理時間 + 走行タスクの最長処理時間
× (補助タスクの周期 / 走行タスクの周期)
= 5 + 2 × (40 / 4) = 25 < 40(ms)

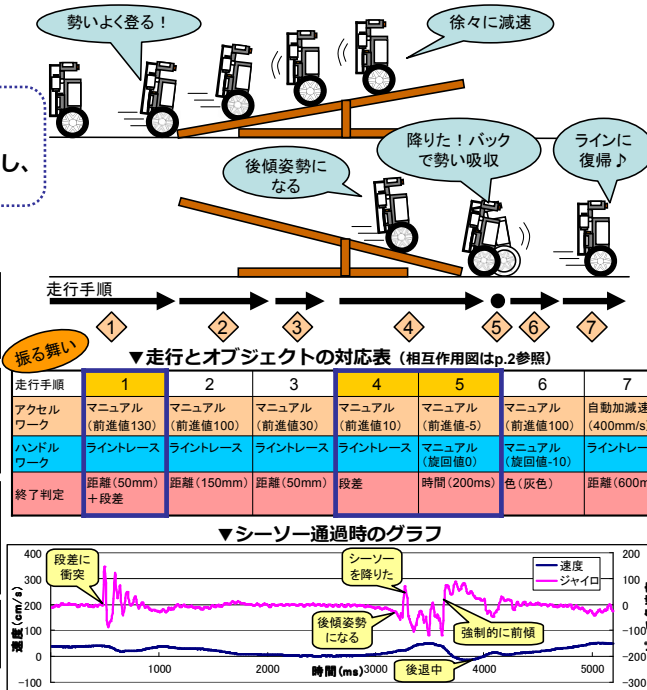
十分に2タスクとも処理時間が間に合うので**問題ない**。

シーソー安定通過

Point 速すぎず、遅すぎず1回で登る速度で進入。
毎回後傾姿勢になるようにし、降りてから姿勢を整える。

走行手順の解説

- 1 何度も段差にぶつかるとうまく登れないので、1回で登れるような速度で進入する。
- 2 徐々に減速して、シーソーが傾く地点まで進む。このとき、後傾姿勢にするため、止まらずに進み続ける。
- 3 強制的に前傾させて姿勢が安定させるため、シーソーを降りたときの衝撃を検知して、0.2秒間後退指示(負の前進値)を出す。
- 4 ラインを探索し、ライントレースを再開する。



シーソー上停止はできる?

Point ジャイロセンサの反応は遅く、フィードバックには使えない。微妙な速度の変化を捉えれば、可能性はあるかも?

慣性により、シーソーが傾いても車体はしばらく傾かないので、ジャイロセンサに変化があったときには、すでにシーソーのバランスが取れる点を越えてしまっている。

一定周期で前後運動をしながら、傾斜で微妙に速度が変化する点を探せば、シーソー上停止に成功する可能性は残っているが...



確実な衝突読み取り

Point ショートカットを兼ねて、2つの衝突の間から読み取り。
衝突の正面で一旦旋回を止めることで、確実な読み取りが可能。

ミステリーサークルの通過パターンは、衝突2本の有無を読み取れば解説することができるので、ショートカットしながら読み取る。

衝突の方向を向いたときに一旦旋回を止めることで、確実な衝突読み取りを実現している。

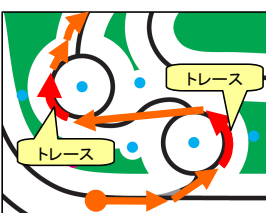


ミステリーサークルは安全重視

Point ゲートへの衝突を避けるため、あえて愚直にラインをトレース。

リスク

ミステリーサークルの通過方法には、ラインをトレースせずに、直角に曲がったり曲率を一定にする方法もあるが、モータエンコーダの誤差などで微妙なずれが発生しやすい。

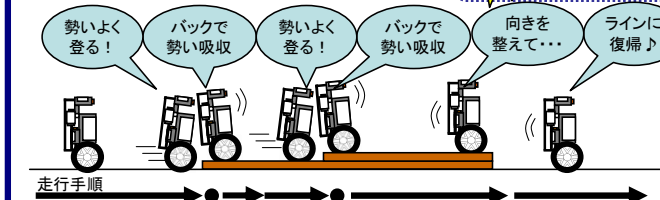


そのため、ミステリーサークルに置かれたゲートに衝突し、リタイアする危険がある。

ライントレースを使う方法では、ラインを探索するためスピードを上げにくい。しかし、モータエンコーダの誤差によるゲートへの衝突を避けることができ安全性は高い。

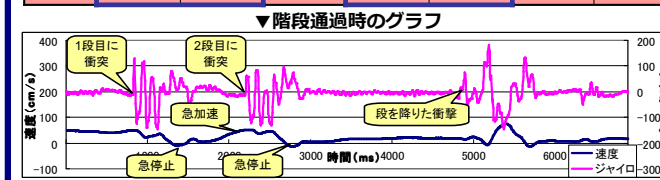
階段は1段ずつ確実に登る

Point シーソーと同様に、1回で登る速度で進入。段を登ってすぐに急停止して、姿勢を整える。



▼走行とオブジェクトの対応表 (相互作用図はp.2参照)

走行手順	1	2	3	4	5	6	7
アクセルワーク	マニュアル(前進値125)	マニュアル(前進値-250)	マニュアル(前進値30)	マニュアル(前進値140)	マニュアル(前進値-300)	マニュアル(前進値50)	マニュアル(前進値50)
ハンドルワーク	ライントレース	ライントレース	ライントレース	ライントレース	ライントレース	ライントレース	ライントレース
終了判定	段差	時間(200ms)	時間(500ms)	距離(50mm)+段差	時間(200ms)	段差	距離(200mm)



- 走行手順の解説**
- 1 まっすぐ登るため、一度で登れるような速度で段差に進入。
 - 2 姿勢を安定させるため、1段目を登り終えた直後に急停止。
 - 3 ラインをトレースし、段差に対する向きを整えてから2段目を登る。
 - 4 2段目を登る際も、1段目を登る際と同様の手順を使う。
 - 5 段差を降りる際は、ライントレースをしたままにすることで、必ずラインの右側に降りるようにする。

◆段差を登った直後に停止する方法
一定時間後退指示(負の前進値)を出す方法を使っているが(手順2, 5)、時間が長すぎると後退してしまう。実験の結果、後退指示を出す時間は0.2秒程度が適していた。

ガレージまでの距離を正確に測る工夫

Point 確実なガレージイン実現のため、距離計測の開始点は確実に定まる地点を選択。

インコース・アウトコースともに、ガレージインを成功させるには距離を正確に測る必要がある。

そのために距離計測の開始点を慎重に選ばなければならない。

ガレージまでの距離を正確に測るために、右図のように、手前の①直線で方位を補正した後、最終コーナーで②特定の方位になったときを距離計測の開始点としている。



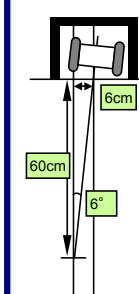
アウトコースのガレージインを支える迅速調整

Point アウトコースのガレージは当日の微調整が必須。簡単に調整できるよう、1cm単位での調整が可能。

アウトコースのガレージは布の張り方で微妙に動くため、ゴール後まっすぐ走ってもたどり着けないことが多い。そこで、目標方位を当日に変更することで、左右のスレに対応する。

$60\sin 1^\circ = 1.05(\text{cm})$ より、60cm手前で方向転換すると約1cm単位の調節ができる。

(極端な例) 6cm左にずれている場合: 60cm手前で右に6度方向転換する。



インコースのガレージは片方の車輪を軸に旋回

Point 片方の車輪を軸に旋回することで、正確な旋回が要求されるインコースのガレージを実現。

インコースのガレージはラインと直角に位置しているので、正確な旋回が要求される。

一旦停止して走行体の中心を軸に旋回すると、そのときの微妙なバランスで旋回の軸がぶれ、位置がずれやすい。

前進値と旋回値を『マニュアル速度』、『マニュアル旋回』を用いて1:1にすると、片方の車輪を軸に旋回する。これを利用して正確な位置に旋回することができる。

限界速度を引き出す自動加減速

Point P制御 + 速度リミッタ + 転倒防止制御で
倒れず限界速度まで加速可能。

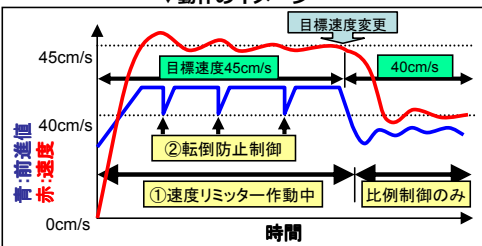
難し以外の走行を高速化し、かつ細かいパラメータ調整が不要になるように、
自動で目標速度までの加減速を行う機能を開発した。
目標とする限界速度は昨年のサヌクのモデルシートより、45cm/sとする。

1. 理論の検討

倒立制御では、モータに倒立姿勢を保つための余力を残しておく必要がある。
そのため、P制御に加えて速度リミッタ・転倒防止制御を追加する。

▼動作のイメージ

- ①速度リミッタは、加減速時に現在速度に
適した前進値に補
正する。
- ②転倒防止制御は、
左右のモータ出力が
どちらか一方でも
90%を超えると危険
と判断して前進値を
下げる。

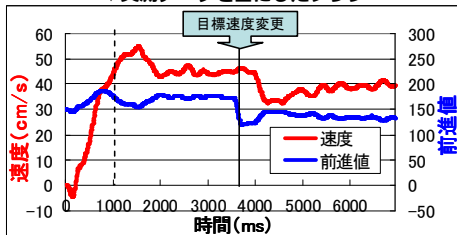


3. 実装の結果

スタートから1秒強で限界速度まで
加速し、その速度を維持している。

結果、コース1周が東京大会より
約10秒短縮！！

▼実測データを基にしたグラフ



要求との対応

自動
加減速

構造との対応

アクセルワーク
↑
自動加減速

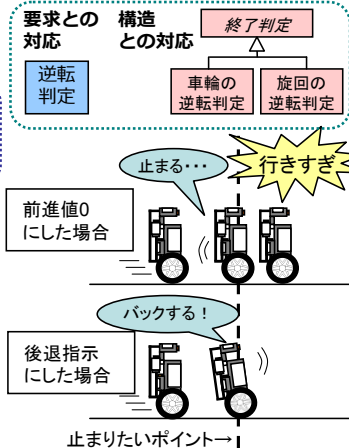
逆転判定を使った正確停止

Point 狙った位置でピタリと止めるために、後退
指示と逆転判定を使用。車輪の逆転を検知
して、後退指示をやめる。

走行中に停止する場合、単純に前進値を0にしても、倒立
制御ライブラリ内で入力値のフィルタが働き、急に止まる
ことはできない。しかし、後退指示（負の前進値）を与え
た場合には続けて後退しようとするために、速度が早く落
ちる。

そのままにしておくと後退を始めてしまうが、モータエン
コーダを使って車輪が逆転を始めたときを検知して後退指
示をやめると、そのまま止まることができる。

また、同じ方法が旋回にも適用でき、衝突の読み取りで正
確な方位で止めるのに使用している。

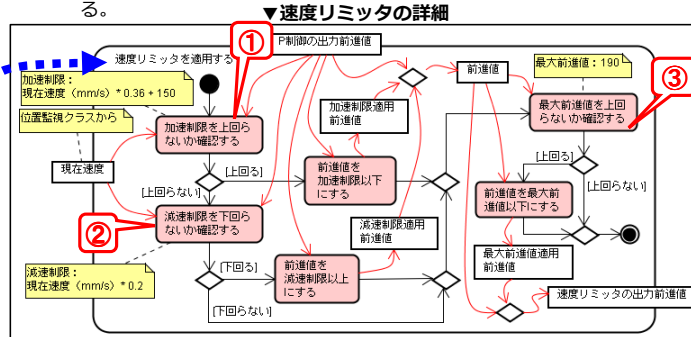
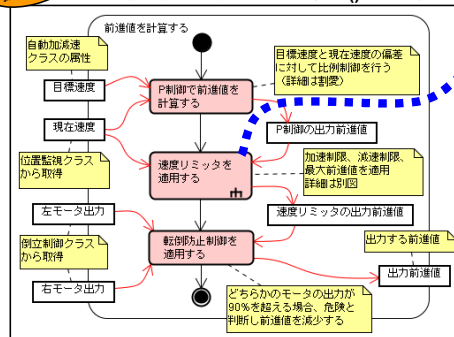


2. クラス内部の振る舞い

P制御、速度リミッタ、
転倒防止制御の順で行う。

速度リミッタでは、加速制限・減速制限・最大前進値の3つで前進値を
制限している。現在の速度を元に①加速制限・②減速制限を計算し、制
限に抵触する場合は補正する。その後、③最大前進値による制限をかけ
る。

▼自動加減速::前進値を計算する()の振る舞い



高速走行時の改良ライトレース

Point 偏差の微分を平滑化し、急激な変化をやわらげる。
さらに、コースアウトするような旋回値は全てカット。

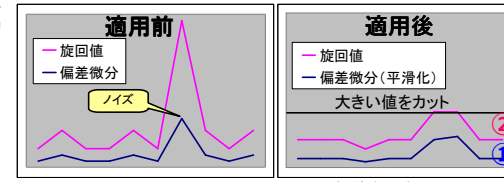
ライトレースには、光センサによるPD制御を採用する。高速走行時の追従性
を上げるには、D制御の働きを強くする（微分ゲインを上げる）必要がある。

ところが、微分ゲインを上げると小さな変
化にも強く反応できる反面、ノイズには弱
くなり、コースアウトの原因となる。

- 対策**
- ①偏差の微分を平滑化
 - ②コースアウトの原因となる
大きい旋回値は全てカット

→ 微分ゲインを上げることが可能に

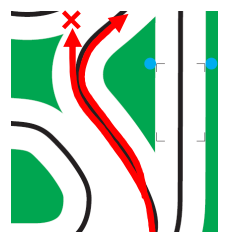
▲ノイズ対策のイメージ



エッジ切り替え

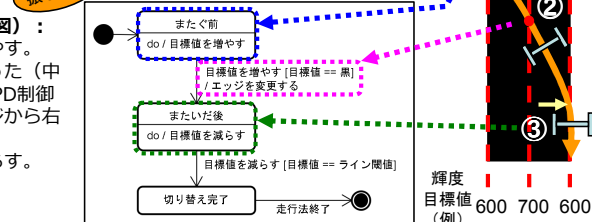
Point ヘアピンカーブでコースアウト
しにくい側に移動することで、
速度を上げることができる。

ライトレース（PD制御）の輝度目標値を徐々に変化させ、
途中でエッジ方向を切り替える。



- エッジ切り替えの手順（右図）：**
- ①徐々に輝度目標値を増やす。
 - ②輝度目標値が黒色になった（中
央をまたいだ）とき、PD制御
のエッジ方向を左エッジから右
エッジに切り替える。
 - ③徐々に輝度目標値を減らす。

▼エッジ切り替えクラスの状態遷移



要求との対応 構造との対応

エッジ
切り替え

ハンドルワーク
↑
ライトレース
↑
エッジ切り替え

方位誤差の補正

Point 走行法の切り替えに使われている方位を、
直線部分のライトレースで補正。
走行の再現性に大きな効果。

スタート時の走行体の置き方や、走行中に蓄積するモータエン
コーダの誤差によって、計算される方位にズレが生じてし
まう。

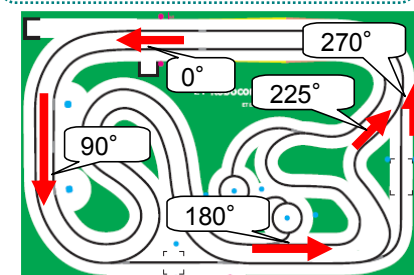
直線をライトレースする箇所、あらかじめ設定した方位
に補正することで、誤差の蓄積を低減。ガレージインの精度
にも貢献している。

実装結果

→ 1周約4度のズレが改善！

要求との対応 方位
修正

構造との対応 走行法
↑
方位補正走行



* 坂道はPD制御中に左右にブレが確認できているので、方位
修正は行わない。