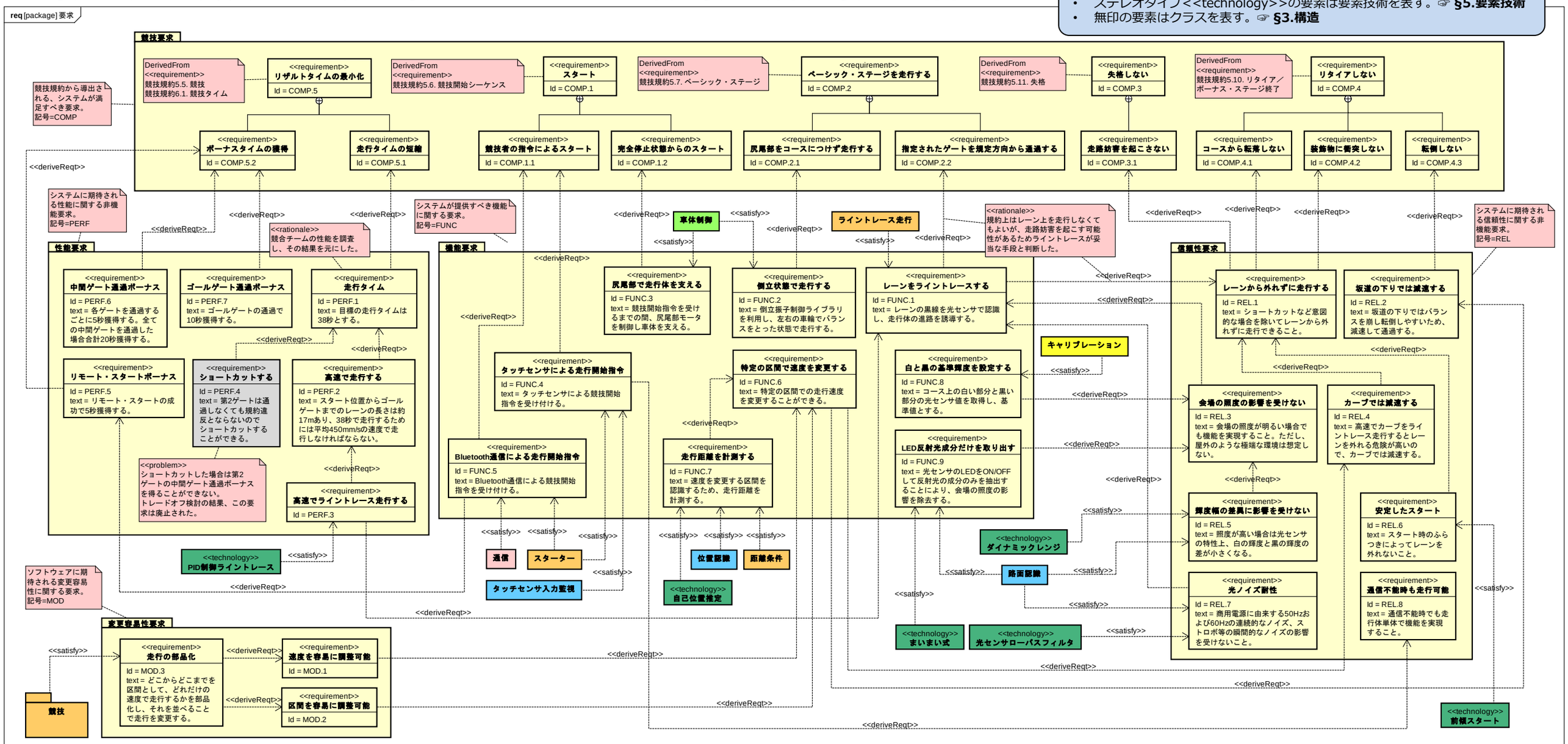
1. 要求

選択課題「ベーシック・ステージを走行する」について派生する要求を導出する
導出された要求と実現方法の対応関係を示す

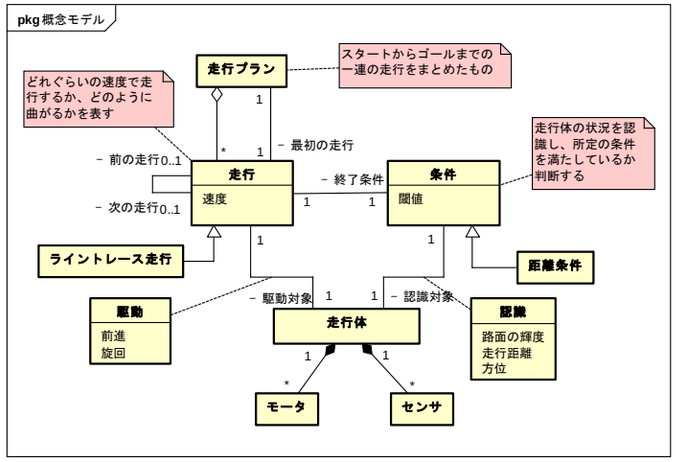
オージス総研 SHINAGAWA RACING

1.1 要求の分析と実現方法：競技上の要求をもとにシステムの機能要求・非機能要求を導出

SysMLの要求図を下記のように拡張して使用した。
- ステレオタイプ<<technology>>の要素は要素技術を表す。☞ **§5.要素技術**
- 無印の要素はクラスを表す。☞ **§3.構造**



1.2 概念モデル：機能に関連する概念を整理



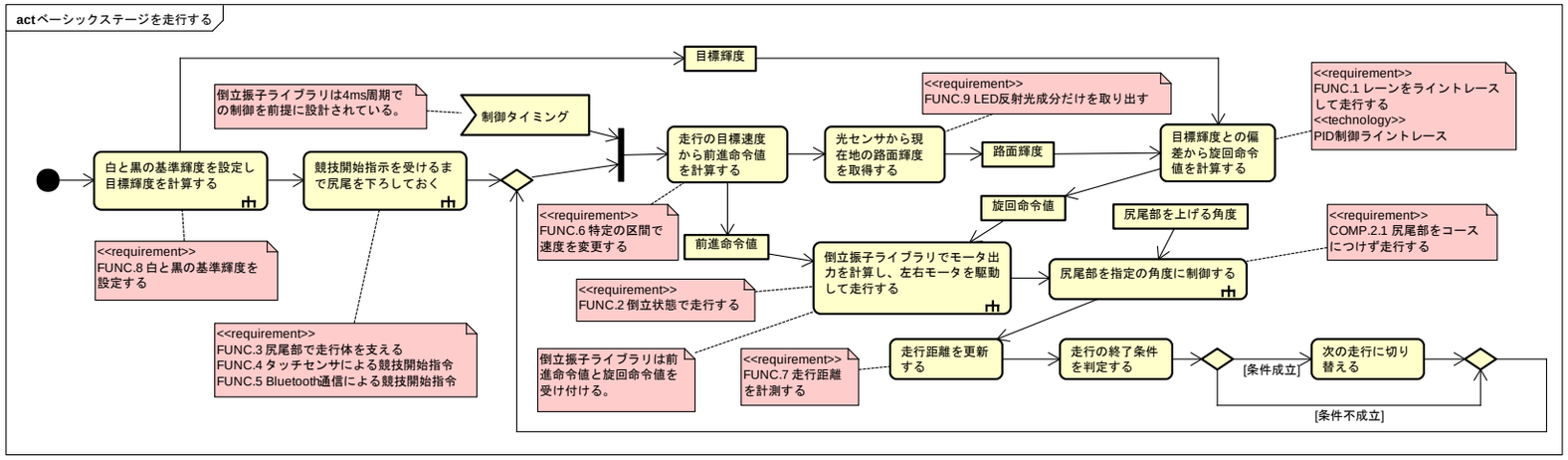
要求「FUNC.6 特定の区間で速度を変更する」を実現するため、速度を属性として持つ「走行」を順番に並べた「走行プラン」という概念で整理した。

各走行は距離などの終了条件を持ち、条件が満たされると次の走行に切り替える。

具体的な走行プランは
☞ **§2.走行戦略** を参照

走行切り替えを実現する振る舞いは
☞ **§4.振る舞い** を参照

1.3 機能を実現する手順：機能要求を満たすことを確認



2. 走行戦略

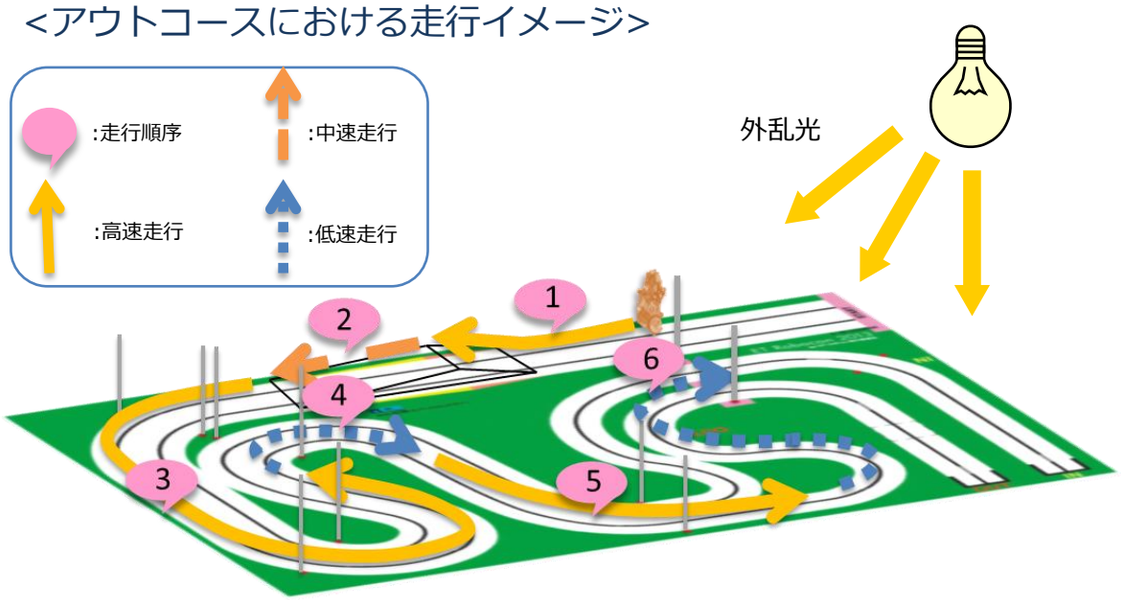
コース図から計測した距離を用いて要求を満たすような走行戦略を設定する
戦略を元に走行プランを作成し、要求走行タイムの充足可能性を検証する

オージス総研 SHINAGAWA RACING

2.1 走行戦略の設定

[走行戦略]
・ラインに沿って走行し、直線は加速し、カーブは減速する
・中間ゲート通過でボーナスタイムを獲得する

<アウトコースにおける走行イメージ>

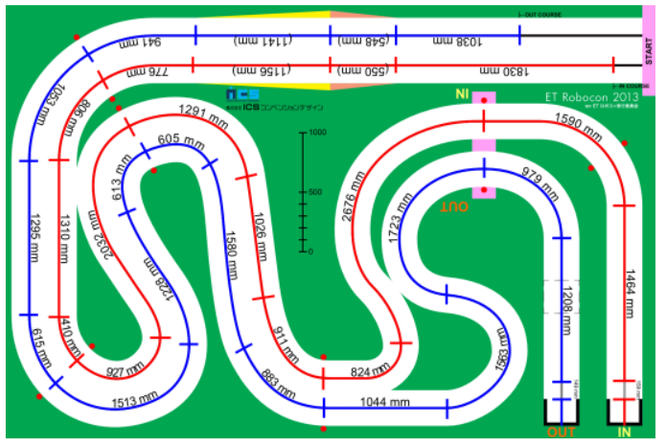


急カーブ・下り坂でコースアウト・転倒の可能性がある。
しかし、可能な限りベーシックステージを早く走行したい。
→走行順序の 2 , 4 , 6 ,について
減速の割合、開始位置、終了位置の検証の必要がある。

2.2 走行プラン作成①:コース図から各部の距離を計測して作成

コース図のデータから特徴(カーブ・坂・直線)の変化する位置とその区間の距離(mm)を計測し、それらの値を基準に 2 , 4 , 6 での速度を検討した。

<コース図から各部の距離を計測>



実験的な方法のみで走行プランを作成すると、走行プランの各ステップで用いられる値が設定した人にしか理解できず、他者による調整が困難。

→**ミリメートル単位で表した距離を基準とすることで、走行プランの理解・調整が容易になる。**

→**大まかな走行プランは事前に作成しておき、試走会では調整に専念できる。**

検討すべき箇所を抽出

下記の区間は走行タイム要求を満たす範囲で減速する

走行順序	区間	開始距離	区間距離
2	坂道の下り	1586mm	1141mm
4	ヘアピンカーブ	9642mm	1218mm
6	S字カーブ	14367mm	3286mm

2.3 走行プラン作成②:走行タイムの要求を満たせることを確認

§2.2で作成した走行プランに基づき、理論値で走行タイム要求を満たせるように速度を設定した。

最高速度を100% = 500mm/secとした場合、理論値では**38秒以下**でベーシックステージを走行でき、要求を満たす。

走行順序	走行内容	走行終了条件	速度 (mm/s)	所要時間 (sec)
1	ライントレース走行::走行速度 = 100%	距離条件::目標距離 = 1538mm	500	3.07
2	ライントレース走行::走行速度 = 90%	距離条件::目標距離 = 1241mm	450	2.76
3	ライントレース走行::走行速度 = 100%	距離条件::目標距離 = 6645mm	500	13.3
4	ライントレース走行::走行速度 = 80%	距離条件::目標距離 = 1218mm	400	3.05
5	ライントレース走行::走行速度 = 100%	距離条件::目標距離 = 3570mm	500	7.01
6	ライントレース走行::走行速度 = 80%	距離条件::目標距離 = 3286mm	400	8.22
	合計			37.41

2.4 走行プランを実現するための要素技術

ライントレース走行

★**PID制御ライントレース** (☞§5.2)
高速ライントレース走行を実現。
<<technology>> PID制御ライントレース

★**光センサローパスフィルタ** (☞§5.5)
照明からのノイズをカットできる。
<<technology>> 光センサローパスフィルタ

会場の照度の影響を受けない

★**ダイナミックレンジ** (☞§5.4)
照明の強い環境での安定した走行を実現。
<<technology>> ダイナミックレンジ

★**まいまい式** (☞§5.6)
外乱光の影響を除去できる。
*ダイナミックレンジで照度の影響を除去しきれない場合、利用を検討する。
<<technology>> まいまい式

走行距離を計測する

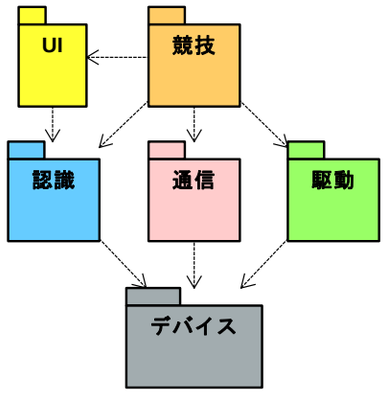
★**自己位置推定** (☞§5.3)
<<technology>> 自己位置推定

3. 構造

要求と走行戦略を実現する為に必要な部品を整理し、設計する上で取り入れた工夫を示す

オージス総研 SHINAGAWA RACING

3.1 機能を役割に応じて分類したパッケージ



パッケージ名	説明
競技	競技の進め方と走行方法を保持
UI	ユーザとシステム間のやりとりを担う
通信	Bluetoothからのコマンドを解釈し、スタート指令や走行プランの設定を行う
駆動	走行体を動かす手段を保持
認識	走行体の状態を認識する手段を保持
デバイス	NXTのデバイスAPIを保持

責務の明確化

各機能を役割に応じて分類することで、まとめた機能群の責務を明確にする

変更の影響範囲の局所化

双方向の依存を無くすことで変更の影響を減らす

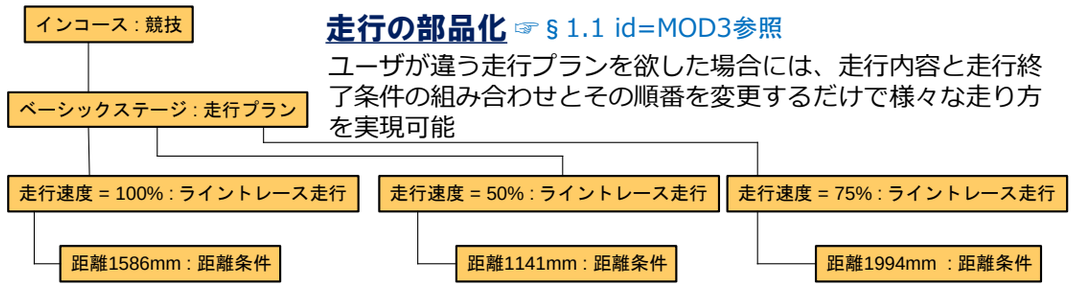
3.2 走行プランを実現したオブジェクト

走行体の動きとオブジェクトの対応

§2.3で示した走行内容、走行終了条件がそのままオブジェクトに置き換わる

走行の部品化 ☞§1.1 id=MOD3参照

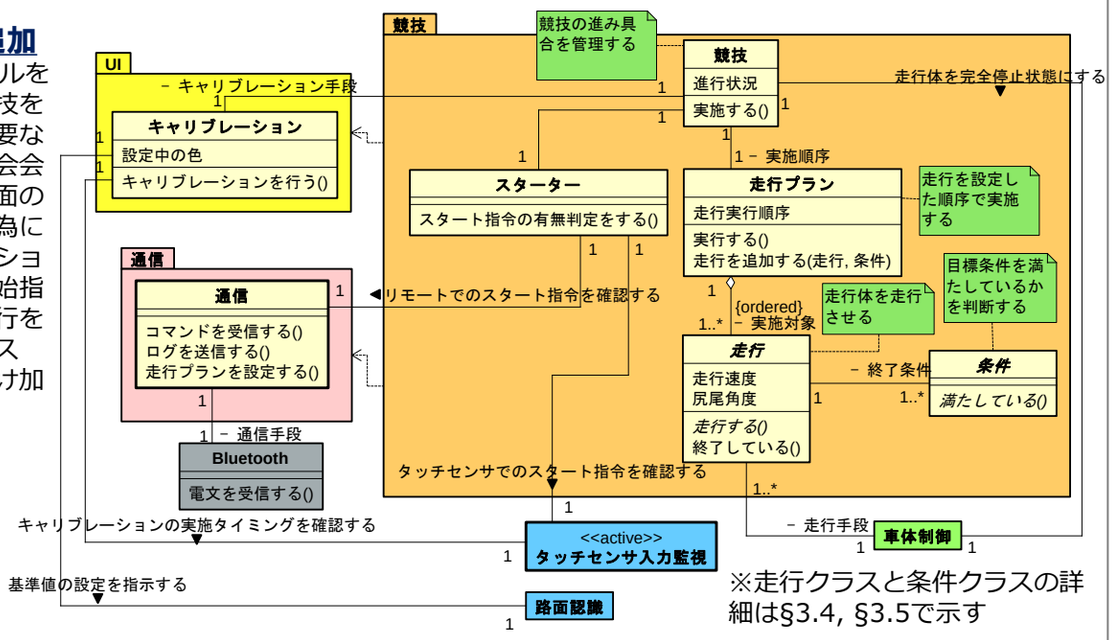
ユーザが違う走行プランを欲した場合には、走行内容と走行終了条件の組み合わせとその順番を変更するだけで様々な走り方を実現可能



3.3 概念モデルに則った構造で競技パッケージを構築

不足部品の追加

§1.2の概念モデルをベースにし、競技を実施する上で必要な部品として、大会会場に合わせて路面の輝度を設定する為に「キャリブレーション」と、競技開始指令に合わせて走行を開始する為に「スターター」を付け加えた



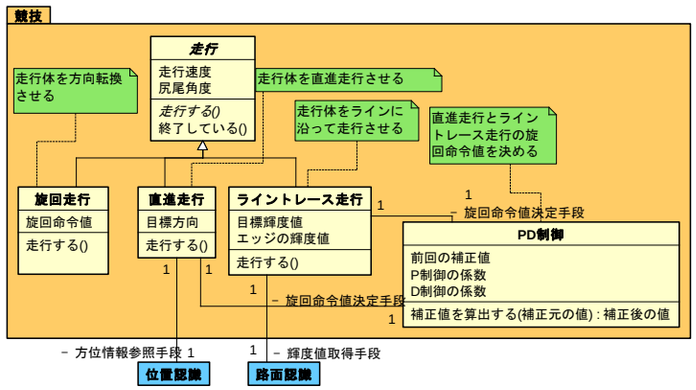
※走行クラスと条件クラスの詳細は§3.4, §3.5で示す

3.4 共通機能を部品化した走行クラス

機能重複の削除

ライントレース走行・直進走行で必要なPID制御を、それぞれの走行サブクラス内に処理を持たせるのではなく、PD制御クラスという共有可能な部品に持たせる。

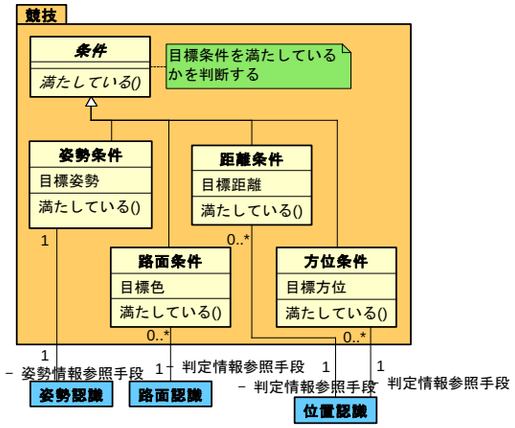
PID制御の詳細は ☞§5.2参照



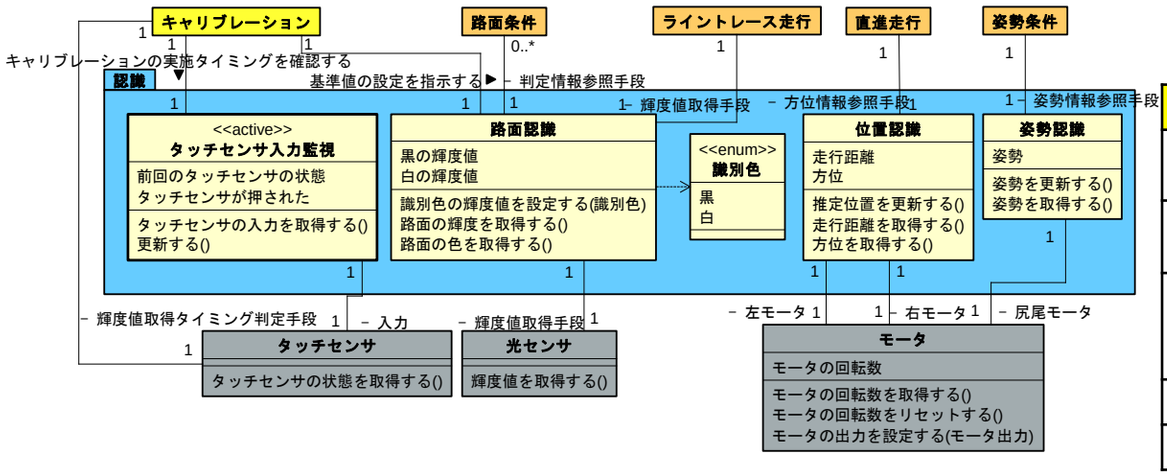
3.5 ポリモーフィズムを用いた条件クラス

変更容易性の担保

増減が予想されるクラスをポリモーフィズムで実現している。新しくサブクラスを追加しても、使用する側のクラスは振る舞いを変える事無く、追加されたクラスを扱える



3.6 デバイス情報を取得し、走行戦略に必要な情報に変換する認識パッケージ



扱いやすい情報に変換

「モータのエンコーダ値」⇒「走行体の走行距離」のように、デバイスから取得出来る値を、走行で必要な情報に変換する

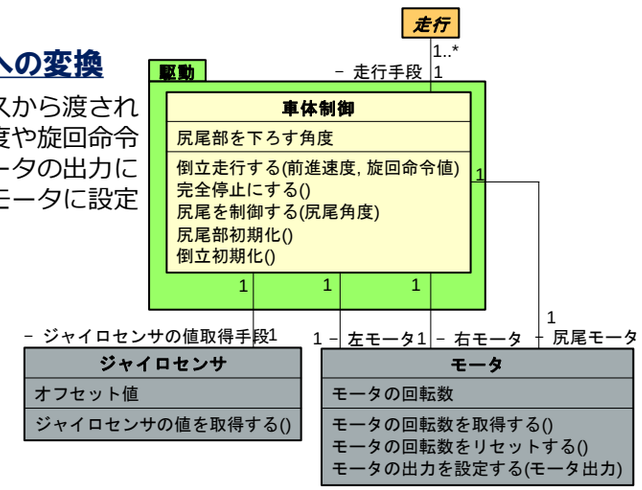
認識パッケージで変換されるデータ

変換前データ	変換後データ
左右モータの回転数	走行距離 変換方法は☞§5.3参照
左右モータの回転数	方位 変換方法は☞§5.3参照
光センサの輝度値	環境光に左右されない路面輝度 変換方法は☞§5.4, §5.5, §5.6参照
尻尾モータの回転数	走行体の姿勢
タッチセンサ入力状態	タッチセンサを押した瞬間

3.7 デバイスへの出力を担い、走行体を動かす駆動パッケージ

出力値への変換

上位クラスから渡される前進速度や旋回命令値を、モータの出力に変換し各モータに設定する



4. 振る舞い

「3.構造」で設計した構造を用いて機能要求の実現ができることを検証する

オージス総研 SHINAGAWA RACING

4.1 本シートで語ること

競技開始から終了までのうち、代表的なシナリオを抽出して、機能実現の検証をする。その中で、**独自記法(凡例参照)を用いて、以下に示す全ての「機能要求」との対応付けを示す。**

☞ **§1.要求「機能要求」**

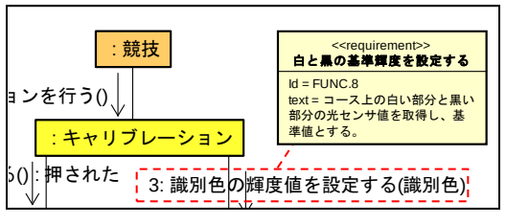
FUNC.8　白と黒の基準値を設定する
FUNC.5　Bluetoothによる走行開始指令
FUNC.4　タッチセンサによる走行開始指令
FUNC.3　尻尾部で走行体を支える
FUNC.2　倒立状態で走行する
FUNC.1　レーンをライントレースする
FUNC.7　走行距離を計測する
FUNC.6　特定の区間で速度を変更する
FUNC.9　LED反射光成分だけを取り出す※

※「LED反射光成分だけを取り出す」の機能実現にあたる「まいまい式」のシナリオは、本番の環境によっては用いない。紙面の都合で割愛する。

☞ **§5.6 「まいまい式」**

凡例（機能要求との対応付け）

以下の例のように、機能要求を満たす振る舞いとの対応関係を赤点線枠で示す。



4.2 クラスの状態遷移

クラスは状態によって振る舞いを変えるものがある。本項ではこれらのクラスの状態遷移について説明する。

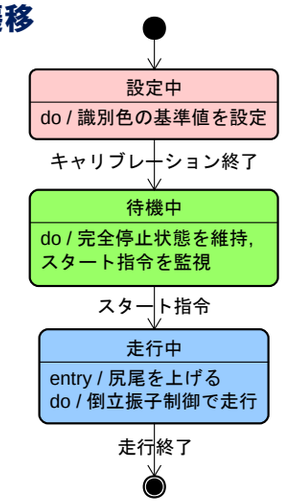
4.2.1「競技」クラスの状態遷移

「競技」クラスは、状態に応じてメッセージを投げる先を変えることで、**競技の進行に合わせて実行する機能を変えている。**

本シートでは、シナリオに以下のアイコンを付すことで、「競技」クラスの状態を示している。

凡例（競技の状態）

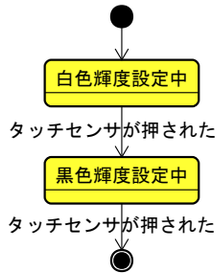
設定中　**走行中**　**待機中**



4.2.2「キャリブレーション」クラスの状態遷移

どの識別色に輝度を設定するかは、「キャリブレーション」クラスの状態によって決定している。

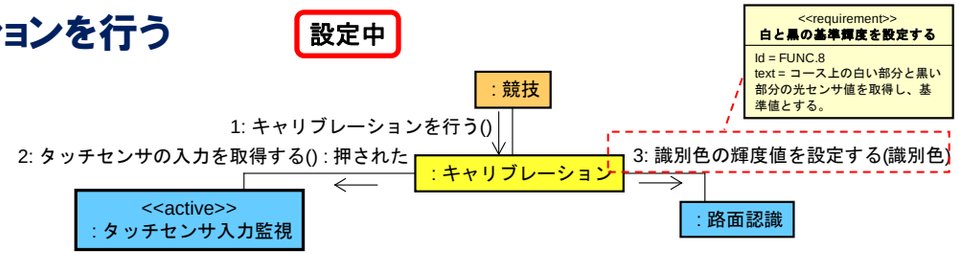
黒色輝度設定を終えた時、状態は終了し、「競技」クラスは待機中に遷移する。



4.3 キャリブレーションを行う

設定中

会場の環境に合わせてレーンをライントレースするために「白」「黒」の識別色別に輝度を設定する。

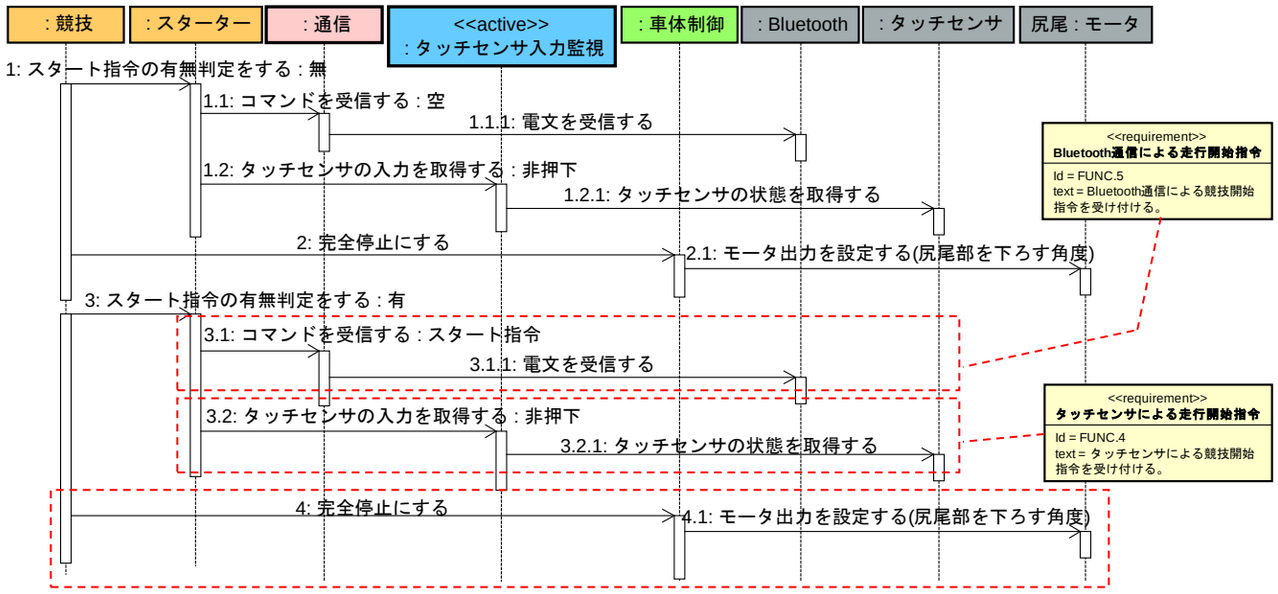


4.4 走行を開始する

待機中 → 走行中

完全停止状態でスタート指令を待ち、指令を受けたら走行を開始する。

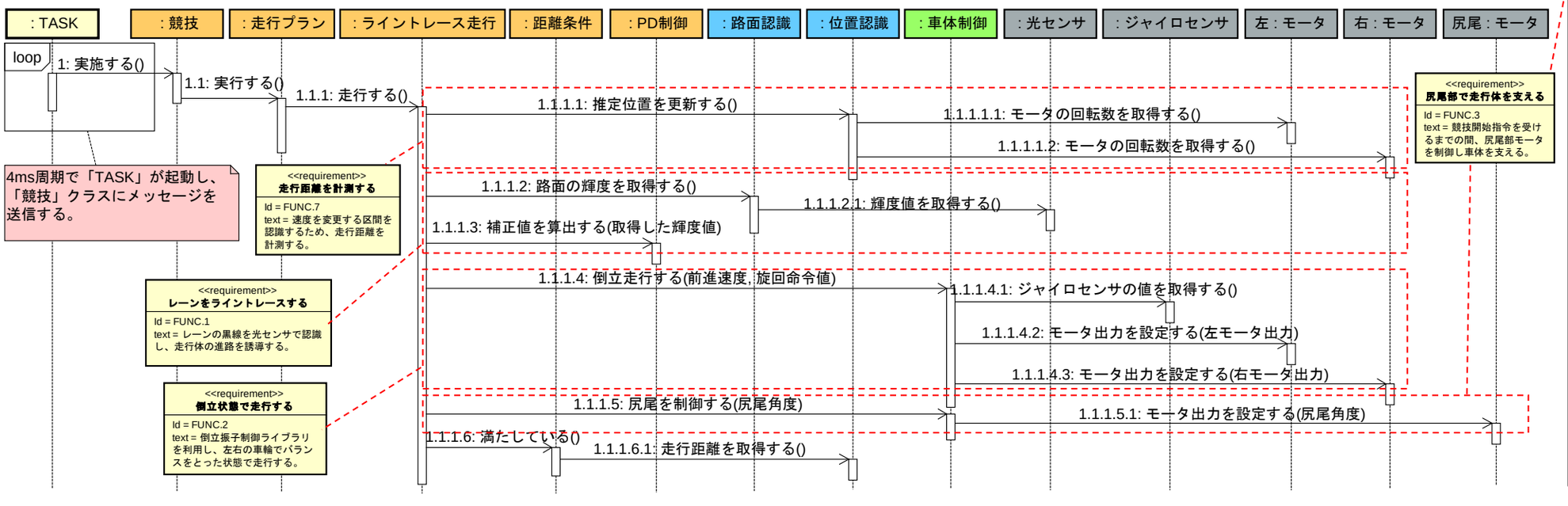
「Bluetooth」「タッチセンサ」のどちらからもスタート指令を受け付けることで冗長性を持たせている。



4.5 走行を行う（基本的な走行の振る舞い）

走行中

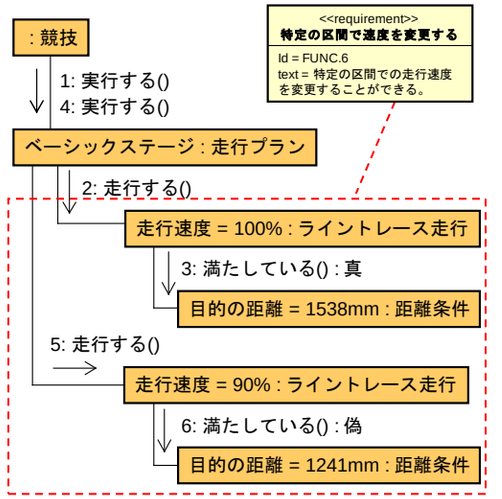
下記の振る舞いの繰り返しでコースを走行できることを検証する。
特に本シナリオは競技の大部分を占めるもの、デバイス層までの全振る舞いを示す。



4.6 走行の切り替え

走行中

「走行プラン」で組み立てた**「条件」に従って「走行」を切り替える**ことで、コースに適した戦略で完走できることを示す。



5. 要素技術

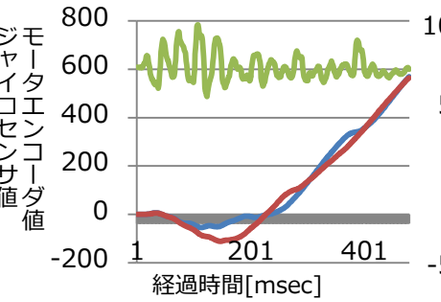
導出された要求を満たす上で障害となる問題、対策と検証結果を示す

オージス総研 SHINAGAWA RACING

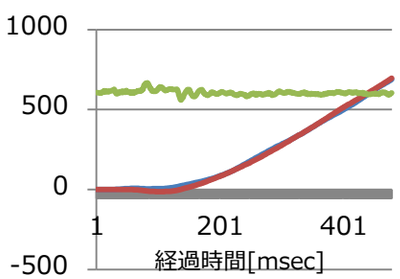
5.1 前傾スタート

要求	REL.6 安定したスタート
問題	倒立走行へ切り替わる際に走行体が安定せずラインから外れる、または前進を開始するまでに時間がかかってしまう場合がある。
対策	しっぽの角度を115度にして前傾状態から倒立走行へ切り替える。
検証	しっぽの角度がほぼ直立である105度でスタートさせた場合と、115度になった瞬間にスタートさせた場合でのスタート精度および加速を調べたところ、精度、加速ともに115度の方が優れていた。

しっぽ角度105度でスタートした場合



しっぽ角度115度でスタートした場合



RightMotorのエンコーダ
LeftMotorのエンコーダ
ジャイロセンサ

しっぽ角度の値が115度の方が安定し、左右モータエンコーダもスタート直後に上昇している

5.2 PID制御ライントレース

関連クラス：PD制御, ライントレース

要求	PERF.3 高速でライントレース走行する
問題	ライントレース時に、路面が「白」か「黒」かの判断のみで固定の旋回量を与えていると、無駄な旋回が生じる。
対策	走行閾値(白と黒の間)を目標値としてPD制御を行うことで、ラインから外れた分に応じた旋回量を補正し、滑らかなライントレースを実現する。※ I制御が無くても安定しており、設定の簡易化のため外している
検証	On/Off制御に比べてPD制御の方がラインを外れずに滑らかにトレースできていることを確認。

以下の計算式を用いる

偏差 = (入力光センサ値 − 走行閾値)

旋回命令値

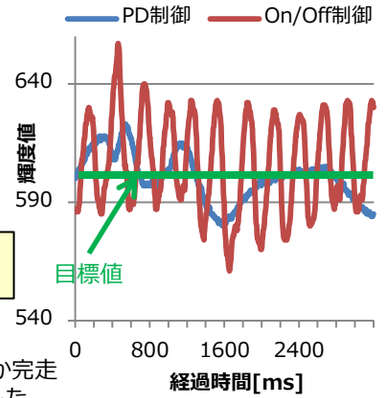
$= (K_p \times 偏差 - K_d(偏差 - 前回の偏差))$

サンプルコースをコースアウトせずに完走するように設定し走行させた結果

走行方法	完走タイム
On/Off制御	**27.31sec**
PID制御	**14.38sec**

約2倍の速さ！

On/Off制御は前進量50,旋回量±50で何とか完走
PID制御は前進量を100に設定しても安定した

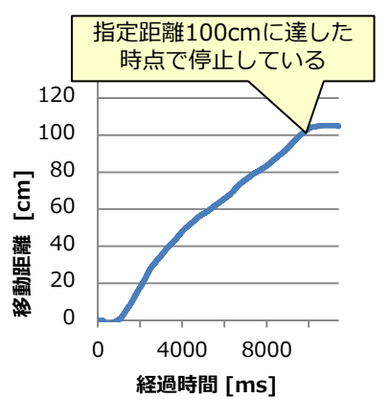


PD制御およびOn/Off制御で走行させた輝度値のグラフ

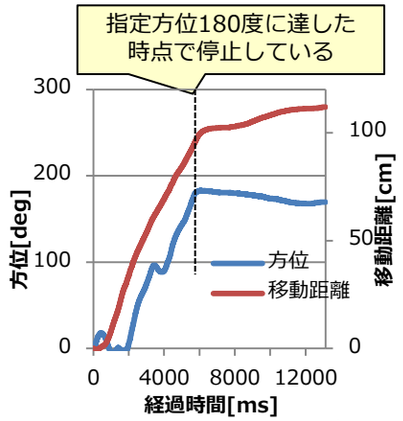
5.3 自己位置推定

関連クラス：位置認識

要求	FUNC.7 走行距離を計測する
問題	走行体が現在どこを走っているか把握できなければ走行プランを切り替えられない。
対策	モータエンコーダ値から距離および走行体の位置を推測する。左右のモータの差から走行体の方位を推測する。
検証	走行距離を認識し停止できているか、および左右のモータエンコーダ値の差から方位を推定し停止できているかを確認。※ 移動距離が増えていなければ停止していると判断(倒立振子のため多少の増減はある)



100cm走らせ停止させたグラフ



半円状に走らせ停止させたグラフ

5.4 ダイナミックレンジ

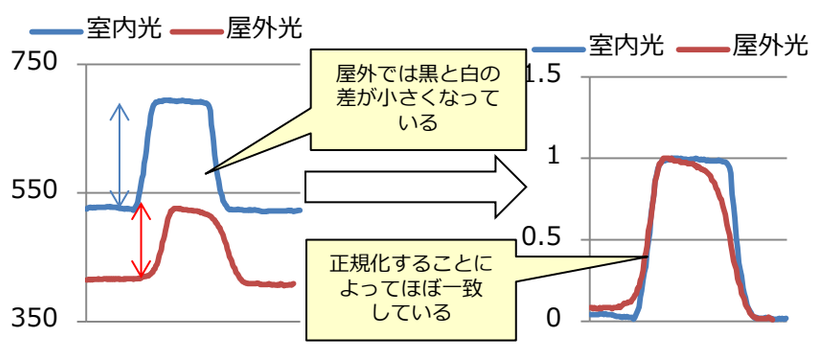
関連クラス：路面認識

要求	REL.5 輝度幅の際に影響を受けない
問題	走行する環境によって明暗が異なり、PD制御における旋回量が少なくなりコースアウトしてしまう場合がある。
対策	基準値に対する相対的な値を環境光から取得し、0から1の間で値を正規化することにより、環境光の影響を受けずにライントレースを実現する。
検証	明度を変化させた環境においても0から1の間に値を正規化できていることを確認。

右記の計算式を用いる

$$正規化輝度 = \frac{入力光センサ値 - 黒基準値}{白基準値 - 黒基準値※}$$

※ 基準値：キャリブレーション時の値



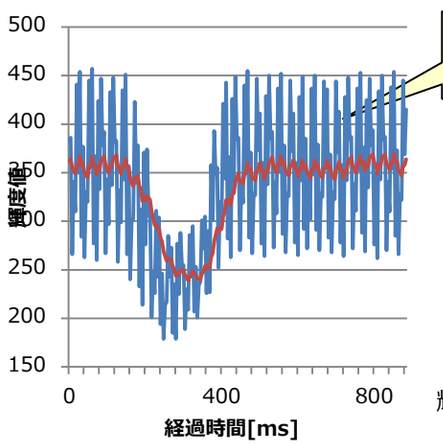
室内、室外において異なる外乱光下において取得した輝度値

正規化した室内、室外の輝度値

5.5 光センサローパスフィルタ

関連クラス：路面認識

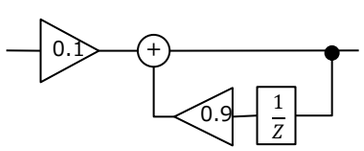
要求	REL.7 光ノイズ耐性
問題	環境光が蛍光灯の場合に周波数の影響で取得する値に一定のノイズが存在しPD制御に影響を及ぼす。
対策	ローパスフィルタを導入し照明からのノイズをカットする。
検証	ローパスフィルタの有無において、ローパスフィルタ有りの光センサ値の推移にノイズが入っていないことを確認。



フィルタ適用前
フィルタ適用後

※ ローパスフィルタの有用性を試すためにLEDを明滅させてログを取得した。

1 次フィルタのブロック線図



※1/Zは1周期前の値

以下の計算式を用いる

輝度値 = 今回の輝度値 * 0.1 + 前回の輝度値 * 0.9

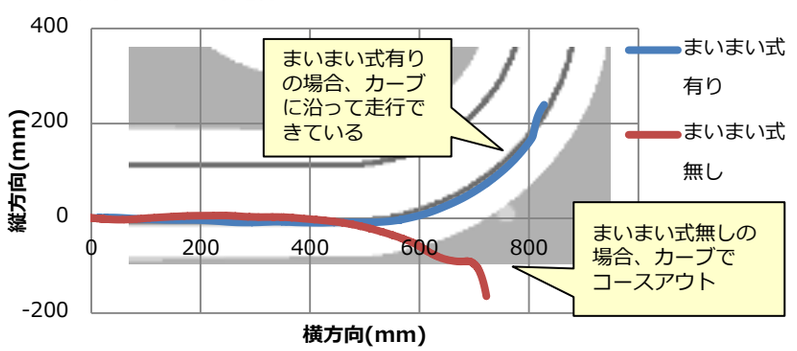
※ 「今回の輝度値」に掛ける係数の値が低い場合、計算後の輝度値に反映する時間が遅れるため、走行速度との調整が必要。

5.6 まいまい式

関連クラス：路面認識, ライントレース

要求	FUNC.9 LED反射光成分だけを取り出す
問題	走行する会場の外乱光が想定以上に明るい場合、外乱光の影響が強く、正確にラインをトレースできない。
対策	光センサーのLEDをONにして取得した輝度値と、OFFにして取得した輝度値を比較し、**外乱光の影響分を除いた値**を用いてラインをトレースする。
検証	走行させる環境を極度に明るくしたうえで、まいまい式有りの場合と無しの場合において走行させ、まいまい式有りの場合ラインを正しくトレースできていることを確認。

直線からカーブにかけて以下のようにまいまい式有り、無しにおいて走行させた。



モータのエンコーダ値からx,y座標を求めた図