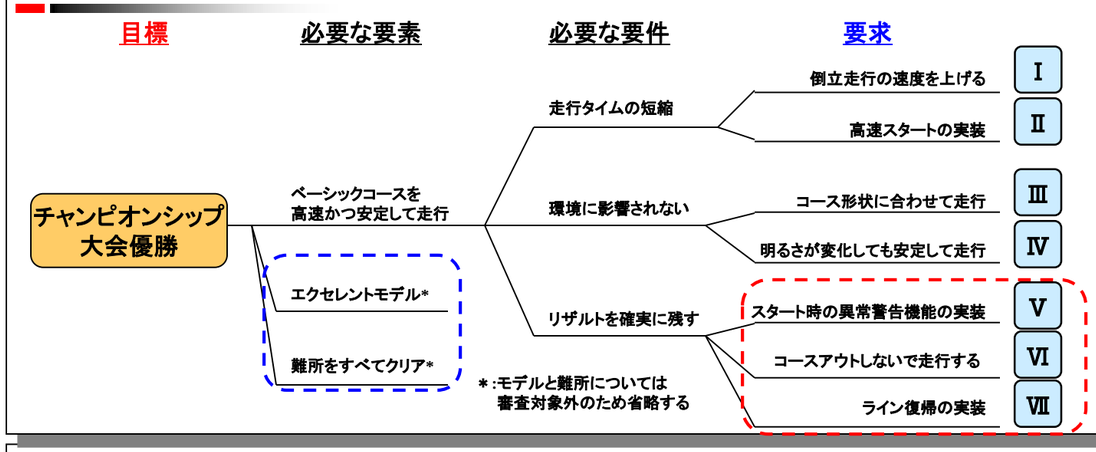
1．要件分析 / 品質要件

じぇっとあーる 株式会社ジェイテクト JTEKT

目標　チャンピオンシップ大会優勝！！

要件分析 マインドマップを用いて目標達成のための要求を抽出した。



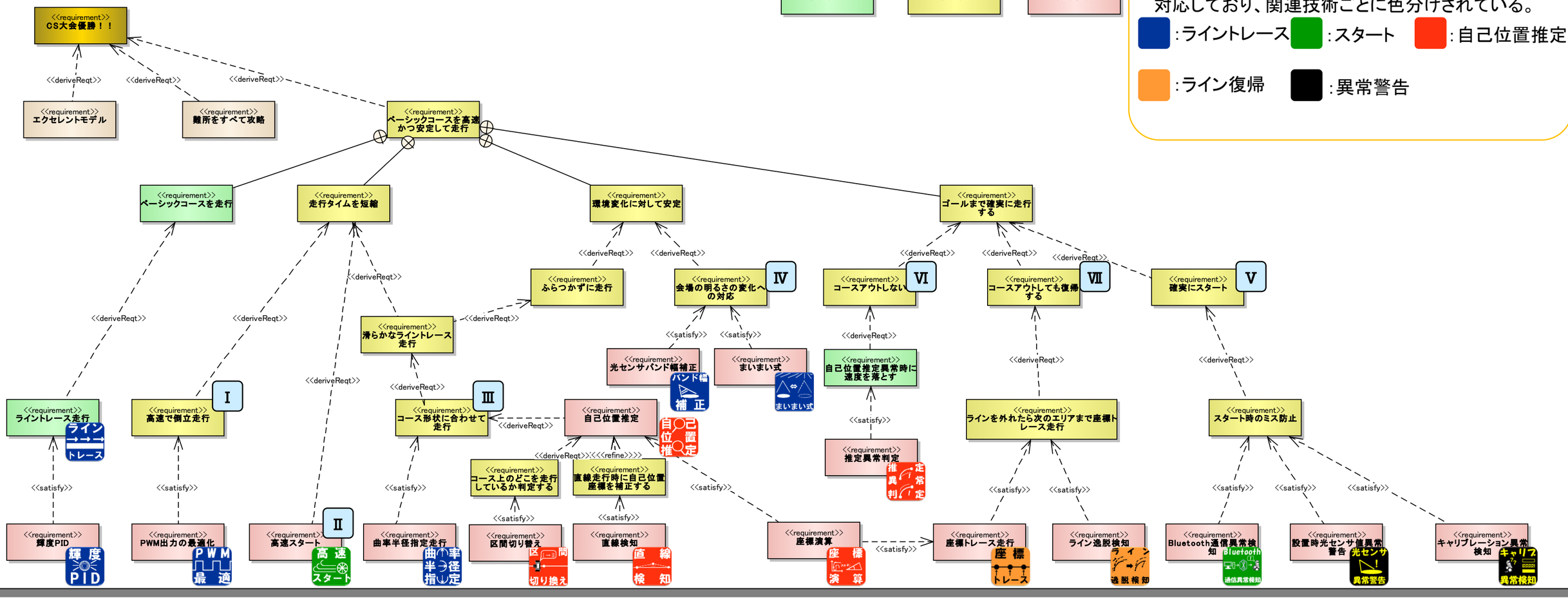
品質要件

抽出された要求から期待された機能を安定して実現するための要素を整理した。

関連要素	要素名	概要	アイコン
V	Bluetooth通信異常検知	スタート時にBluetooth通信が正常に行われなかった場合には、通信異常を検知する。	
	設置時光センサ異常警告	スタート位置に走行体を置く際に、適切な場所に置かれていない場合には警告音を出す。	
	キャリブレーション異常検知	光センサのキャリブレーション時に順番を誤るなど適切に行えていない場合には警告音を出す。	
VI	推定異常判定	走行体が推定している位置と実際の位置にずれが生じた場合には、異常と判定する。	
VII	ライン逸脱検知	走行体がラインを外れた場合には、正確に検知する。	

P.4制御戦略、P.5要素技術に詳細を示す。

sysML要求図 マインドマップで抽出した要件をsysML要求図を使用して詳細に分析を行い、必要な要素技術の洗い出しを行った。

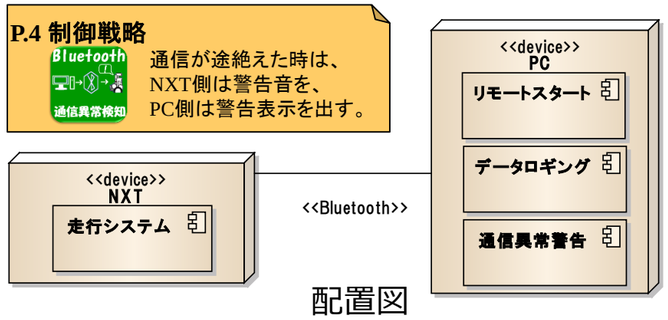


2．構造設計

じぇっとあーる 株式会社ジェイテクト JTEKT

システム構造

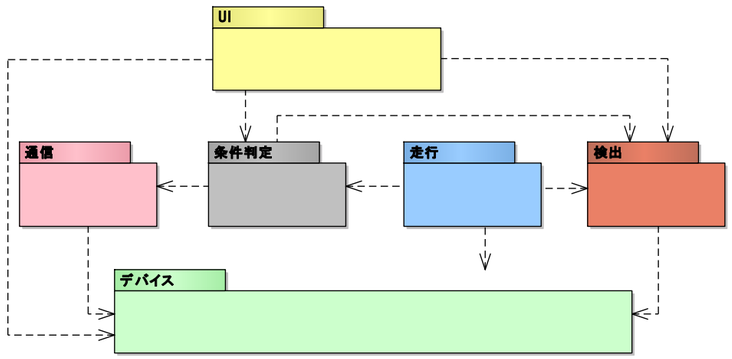
システムはBluetoothで接続された
NXT（走行体）とPCで構成される。



配置図

パッケージ構造

構造的な繋がりを明示的にし、ソフトウェアの変更性等の保守性向上のため、
実現する機能毎にクラス群をパッケージとして分別した。



パッケージ構成図

パッケージ一覧

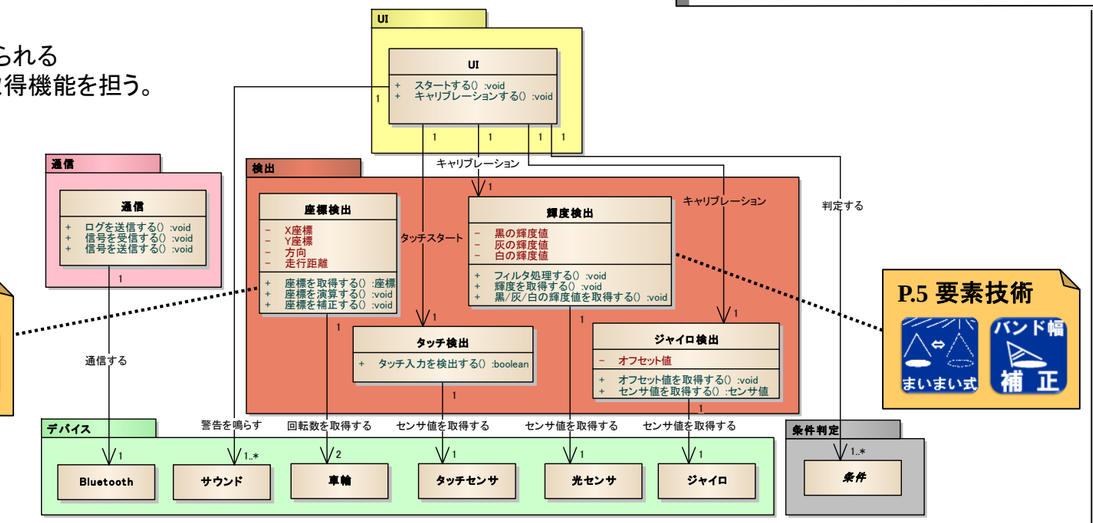
UI	ユーザによる操作を実現する。
走行	走行制御を実現する。
通信	Bluetoothによる通信機能を実現する。
条件判定	検知・警告等の判定機能を実現する。
検出	センサ値の取得・加工処理を実現する。
デバイス	各API呼び出しを実現する（クラス図は省略）。

クラス構造

ベーシックステージ走行を実現する
ソフト構造をクラス図で示す。
（『制御戦略』『要素技術』との対応をアイコンで示す。）

パッケージ：検出

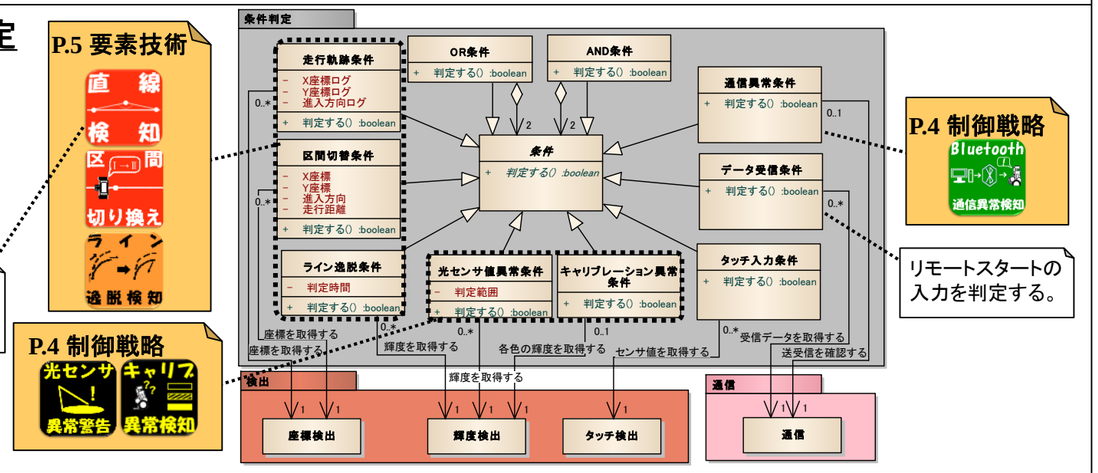
デバイスパッケージから得られる
各センサ値の加工処理、取得機能を担う。



パッケージ：条件判定

走行時の判定や異常の
判定を担う。
条件クラスを派生させる
ことで拡張・追加が容易な
構造となっている。

走行軌跡条件で
『直線を走行している』等を
判定する。

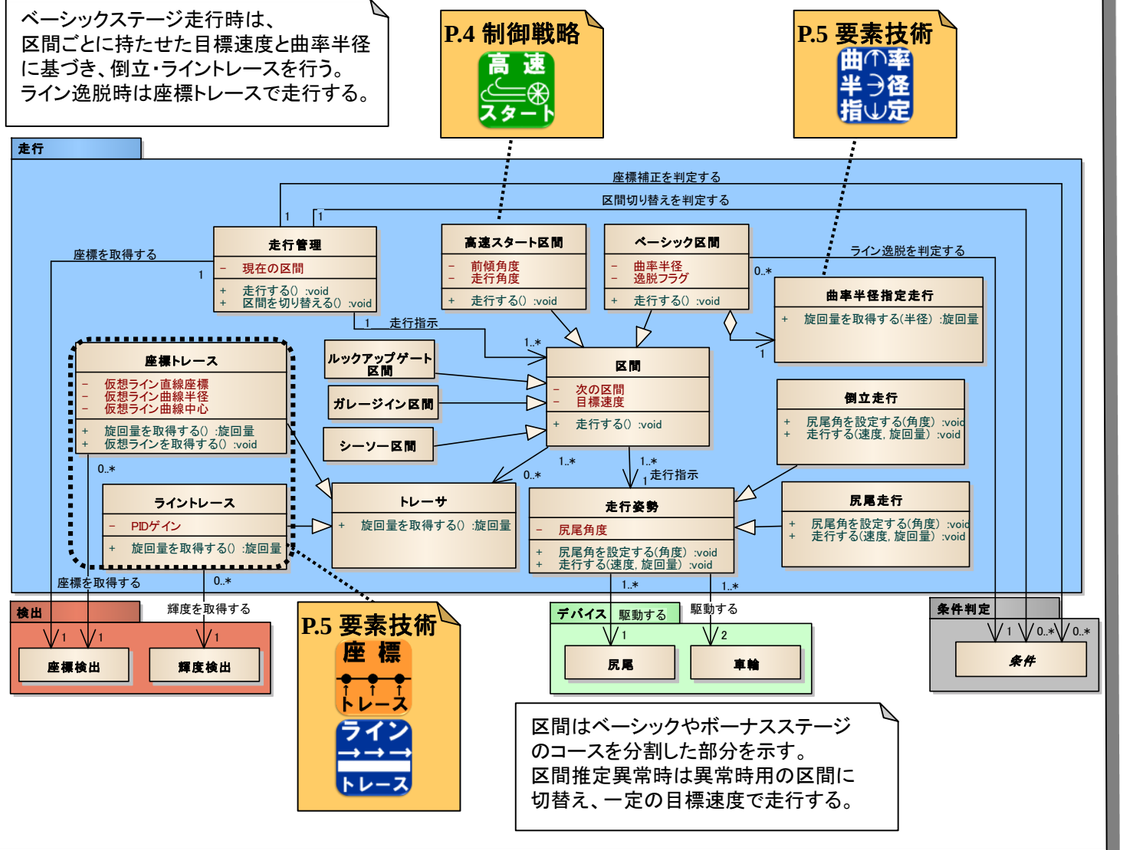


リモートスタートの
入力を判定する。

パッケージ：走行

走行の制御機能を担う。区間ごとにトレーサ、
走行姿勢を選択することで、様々な走行方法を
実現する（ボーナスは省略）。

ベーシックステージ走行時は、
区間ごとに持たせた目標速度と曲率半径
に基づき、倒立・ライントレースを行う。
ライン逸脱時は座標トレースで走行する。

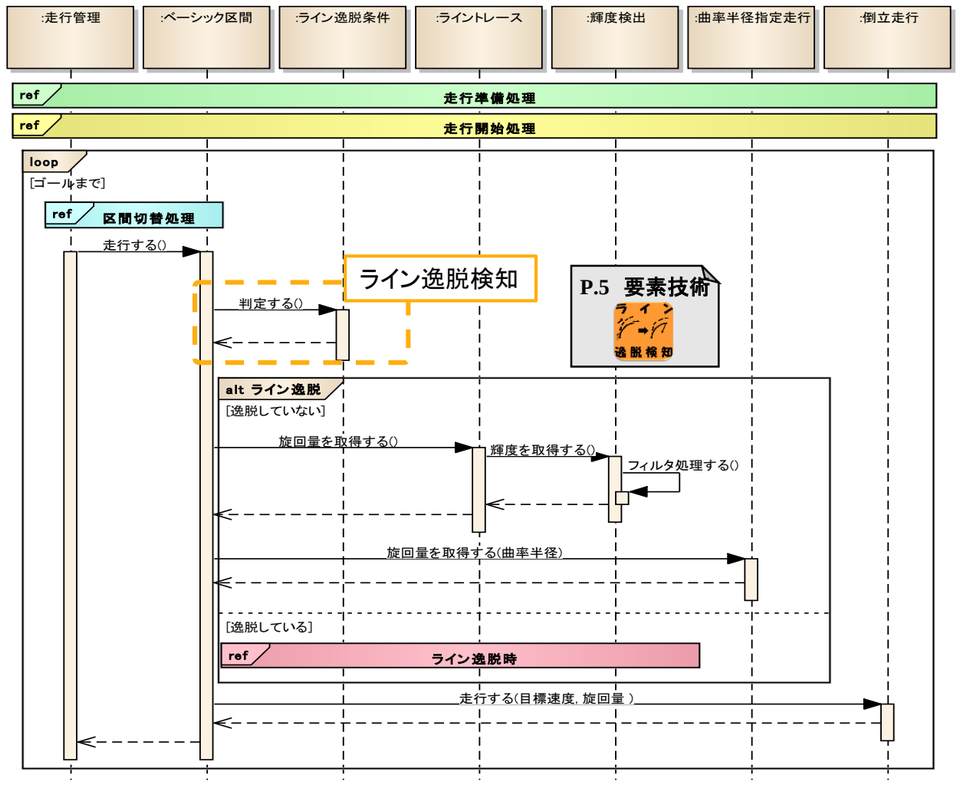


区間はベーシックやボーナスステージ
のコースを分割した部分を示す。
区間推定異常時は異常時用の区間に
切替え、一定の目標速度で走行する。

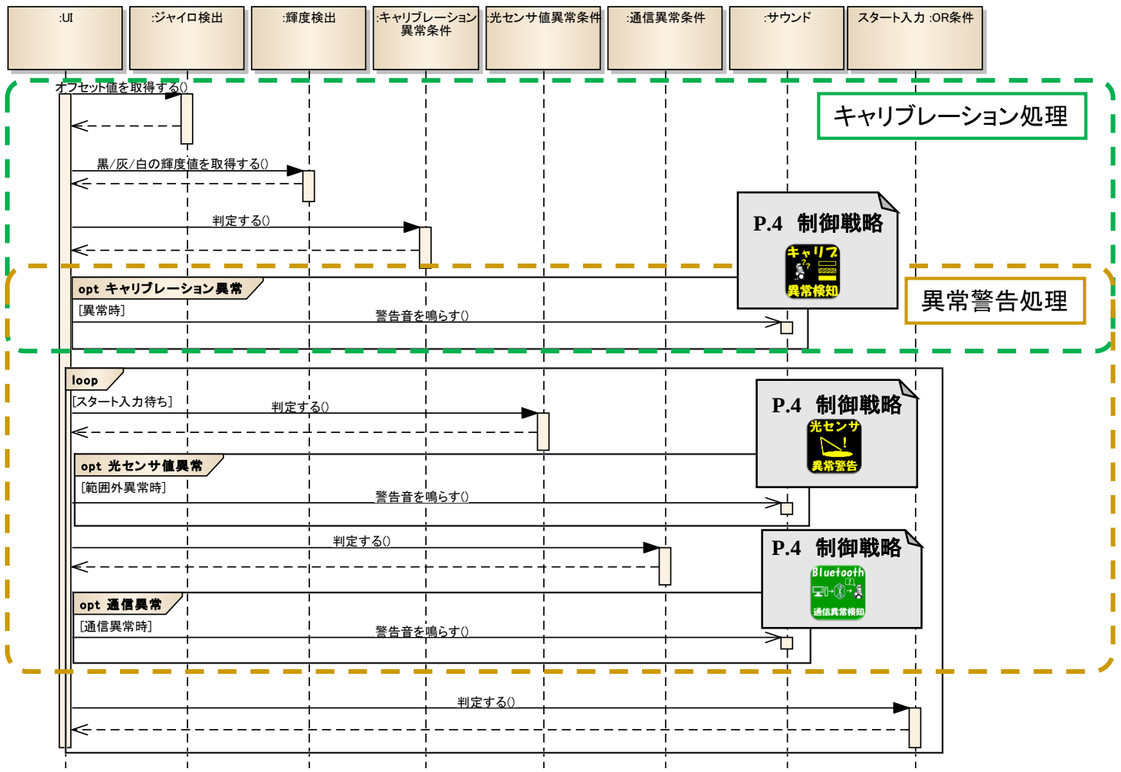
3．振る舞い設計

じぇっとあーる 株式会社ジェイテクト JTEKT

全体処理



走行準備処理



ベーシックステージ走行の振る舞いを シーケンス図で示す。

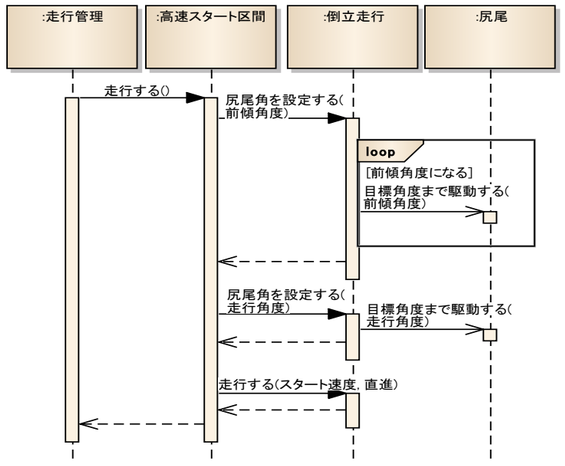
走行準備処理
（ キャリブレーション、走行体設置、スタート入力待ち）

走行開始処理
（ 高速スタートのための重心移動 ）

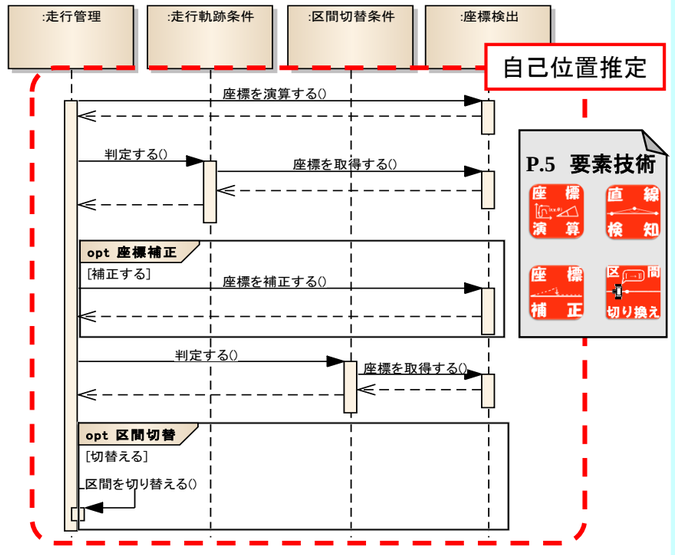
走行処理
（ 区間切り換え処理、走行 ）

という処理の流れとなる。

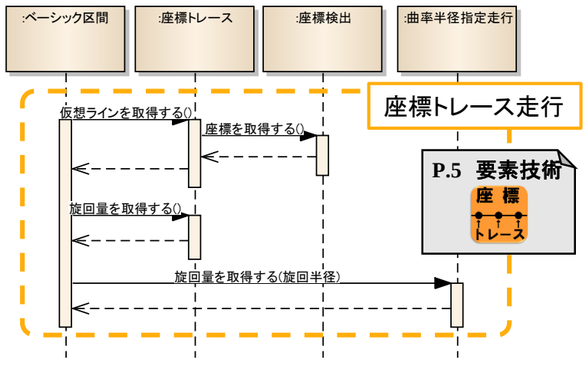
走行開始処理



区間切り替え処理



ライン逸脱時



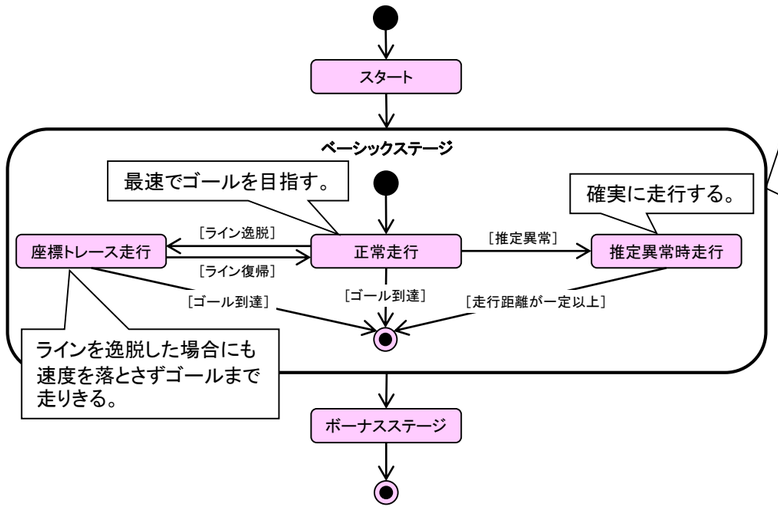
ライン逸脱判定時は座標トレース走行に移行する。

4．制御戦略

じぇっとあーる 株式会社ジェイテクト JTEKT

全体の流れ

様々な異常状態に対応しながら最速でゴールすることを目指す。



光センサ輝度値取得方法の選択

会場の外乱光の状況に応じてまいまい式（→P.5 要素技術）の有り無しを選択する。

外乱光弱
→まいまい式無し

外乱光強
→まいまい式有り

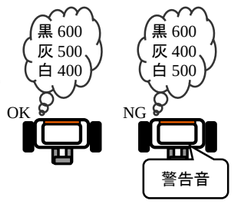
まいまい式有りの場合、輝度値取得回数が減り、応答が遅れ、ライン逸脱の恐れがあるため速度を落として走行する。

スタート

ミスなく正確にキャリブレーションを行い、確実なスタートを行う。

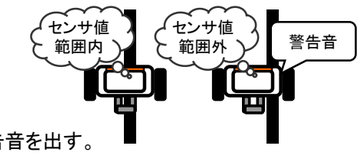
キャリブレーション異常検知

【課題】
・キャリブレーションを失敗したままスタートさせる恐れがある。
【対策】
・キャリブレーション終了後に結果を比較し、各色の輝度値が「黒＞白＞灰」など失敗が疑われる場合、警告音を出す。
・警告発生時はユーザが判断し、必要に応じてリセットする。



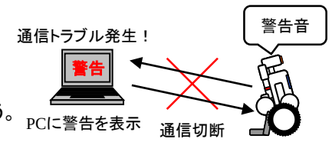
設置時光センサ異常警告

【課題】
・環境や走行体の置き方次第で、スタート時のセンサ値が目標値から外れている場合があり、スタート直後にふらつく恐れがある。
【対策】
・光センサの値が目標値から外れていたら警告音を出す。



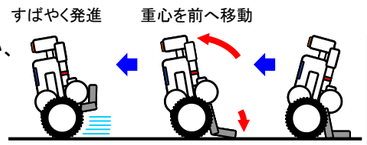
Bluetooth通信異常検知

【課題】
・通信が上手くいかず、動作しない恐れがある。
【対策】
・キープアライブ機能によって、通信異常検知を行う。



高速スタート

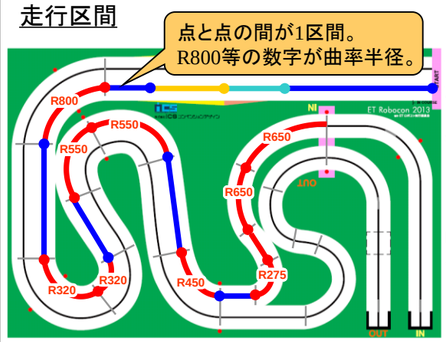
【課題】
・通常スタートでは倒立状態になる際、走行体のバランスをとるために後退してしまい、タイムロスとなる。
【対策】
・スタートする際、尻尾で地面を押す事で重心を前に移動させる。地面を押す尻尾角は、安定と速度が両立可能な128度を採用した。



正常走行

高速走行を行うために、走行区間毎に最適な曲率半径、区間速度を指定し、輝度PIDによりラインレースを行う。

走行区間

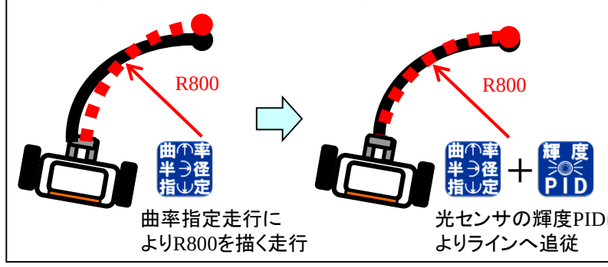


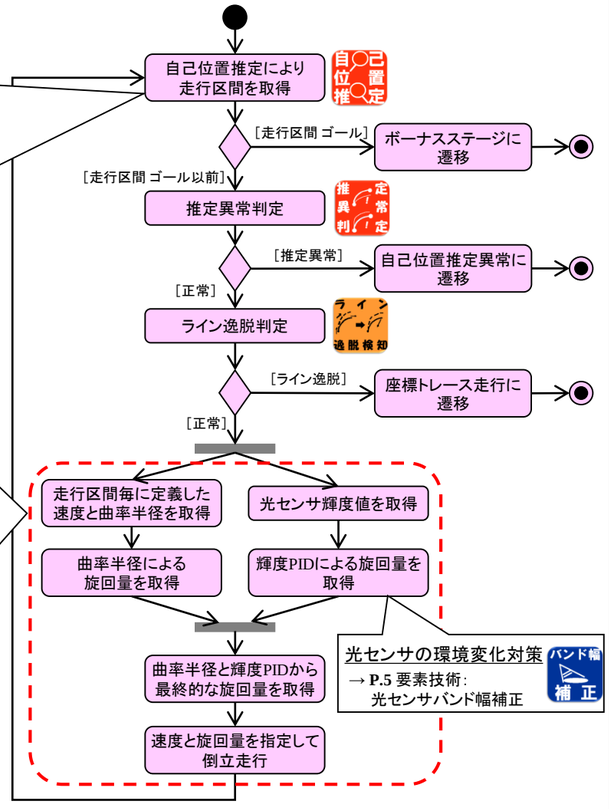
コースを直線とコーナーで走行区間として分割する。

各区間に、以下を定義
『曲率半径』
『区間距離』
及び、
PWM値が100近くになるよう、実験から求めた
『区間速度』

曲率半径と輝度PIDによるライントレース

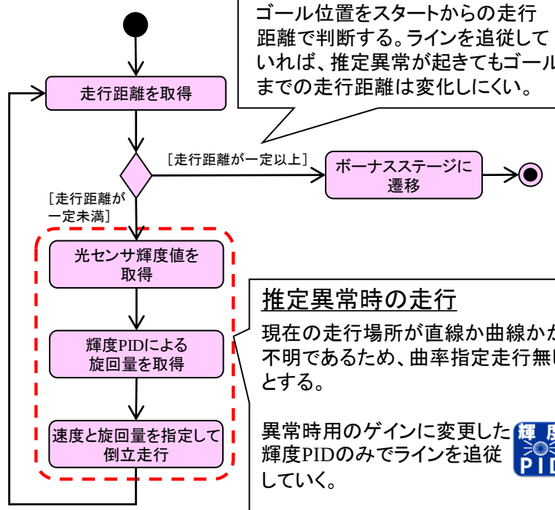
区間毎に定義された曲率半径と走行体の旋回半径が同じになるよう制御。（直線は半径0として直進。）
曲率半径のみの旋回によるラインずれを輝度PIDによって補正する。





推定異常時走行

推定異常と判断された場合は、速さよりも確実に走行することを優先する。



ゴール条件

ゴール位置をスタートからの走行距離で判断する。ラインを追従していれば、推定異常が起きてもゴールまでの走行距離は変化しにくい。

推定異常時の走行

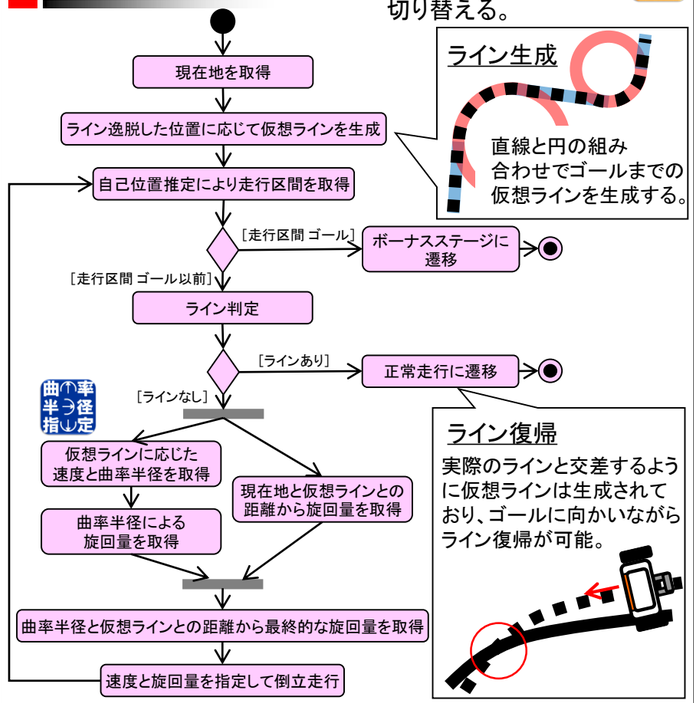
現在の走行場所が直線か曲線かが不明であるため、曲率指定走行無しとする。

異常時用のゲインに変更した輝度PIDのみでラインを追従していく。

速度は、コースから逸脱しない値で一定値固定とする。

座標トレース走行

ライン逸脱と判定された場合は座標トレースに切り替える。



ライン生成

直線と円の組み合わせでゴールまでの仮想ラインを生成する。

ライン復帰

実際のラインと交差するように仮想ラインは生成されており、ゴールに向かいながらライン復帰が可能。

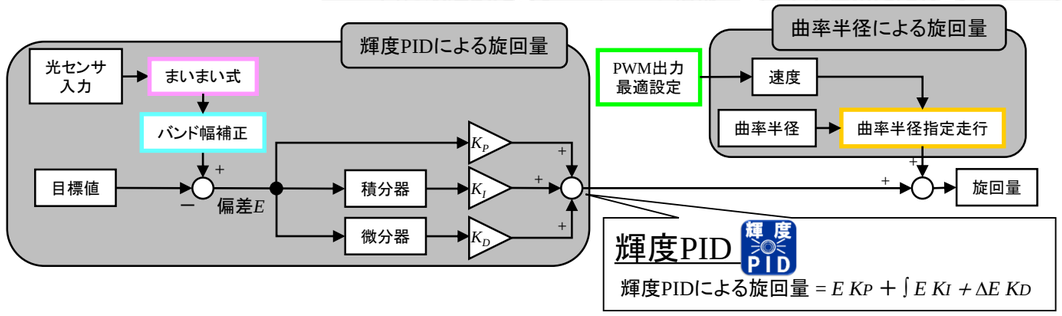
5．要素技術

じぇっとあーる 株式会社ジェイテクト JTEKT

ライントレース

【課題】
・高速な倒立走行をするために、コースの区間毎に転倒しない範囲の最速設定を見極める。
・環境変化や外乱の影響を受けない光センサによる制御を確立する。

【対策】
下記四項目の技術を採用する。
PWM出力最適設定、**光センサバンド幅補正**、**曲率半径指定走行**、**まいまい式**



輝度PID

輝度PIDによる旋回量 $= E\,K_P + \int E\,K_I + \Delta E\,K_D$

曲率半径指定走行

・速度V、曲率半径Rをコースの区間毎に指定する。
・左輪の速度がV-$Turn$、右輪の速度がV+$Turn$となるとき、図の速度と半径の比から旋回速度$Turn$を導出可能。

$R : V = R - W/2 : V - Turn$

$Turn = VW / 2R$

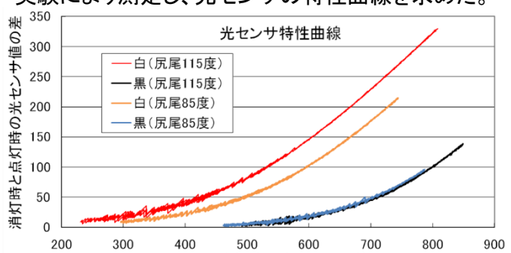
曲率(旋回)半径：R
速度：V
車輪間幅：W
旋回速度：$Turn$

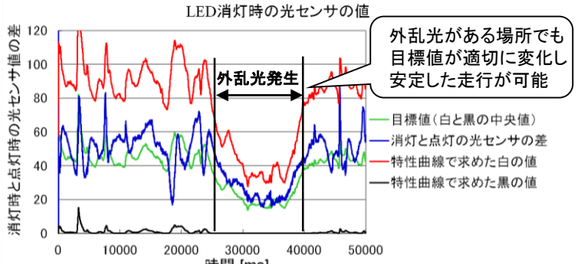
PWM出力最適設定

・倒立走行中、モータのPWM値が100を越え続けるとバランスがとれず転倒、もしくはカーブが曲がりきれないといった問題が生じる。
・走行中の倒立振子APIの出力PWM値を計測し、PWM値が100を超えない最適な速度を設定した。

まいまい式

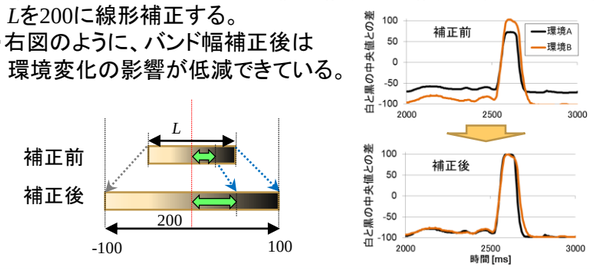
・外乱光の影響を取り除くために、LED消灯時・点灯時の光センサ値の差を用いる。
・LED消灯時・点灯時の光センサ値と、路面の色の対応関係を実験により測定し、光センサの特性曲線を求めた。





光センサバンド幅補正

・環境により黒と白の光センサ値の幅（バンド幅）Lが異なるため、Lを200に線形補正する。
・右図のように、バンド幅補正後は環境変化の影響が低減できている。



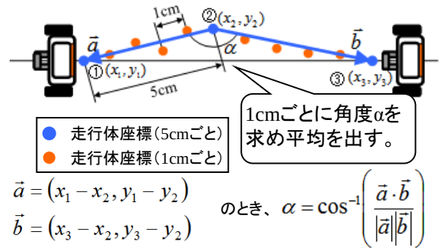
自己位置推定

【課題】
・曲率半径指定旋回のため、コースのどの区間にいるのか推定する。
・タイヤのエンコーダ値からコース上の座標を演算し推定する。
・単純な座標演算のみでは、ずれが累積していき、推定誤りが生じる。
・推定区間が実際の位置とずれている場合、ライン逸脱の懸念がある。

【対策】
・走行体の座標を算出し、区間毎に設定した座標と比較し推定する。
→ **座標演算** **区間切り換え**
・ふらつき無く、走行が安定する直線走行時に座標のずれを補正する。
→ **直線検知** **座標補正**
・走行距離と実際の区間距離の差から推定異常を判定する。
→ **推定異常判定**

直線検知

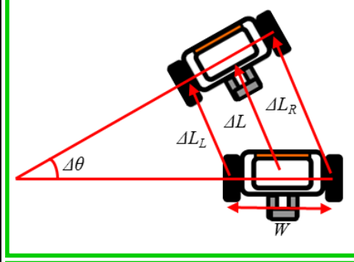
・現在点と5cm前、10cm前の点の3点の座標の成す角αを求める。
・1cmごとにαを求め、過去10cm分の移動平均$\bar{\alpha}$を算出する。
・$\bar{\alpha}$が180±5°の時、直線走行と判断。



$\vec{a} = (x_1 - x_2, y_1 - y_2)$
$\vec{b} = (x_3 - x_2, y_3 - y_2)$
のとき、$\alpha = \cos^{-1}\left(\dfrac{\vec{a}\cdot\vec{b}}{|\vec{a}||\vec{b}|}\right)$

座標演算

・車両の運動で一般的なオドメトリ手法によって、走行体の座標(X,Y)、方向θ、移動距離Lを算出する。
・スタートを原点とし、前後方向をX座標、左右方向をY座標としている。



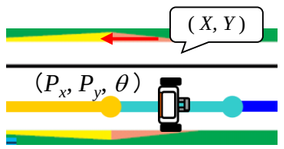
$\Delta L = (\Delta L_R + \Delta L_L)/2$
$\Delta\theta = (\Delta L_R - \Delta L_L)/W$
$X_{i+1} = X_i + \Delta L\cos(\theta_i + \Delta\theta/2)$
$Y_{i+1} = Y_i + \Delta L\sin(\theta_i + \Delta\theta/2)$
$\theta_{i+1} = \theta_i + \Delta\theta$

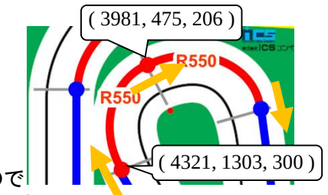
移動前	移動後
座標：(X_i, Y_i)	座標：(X_{i+1}, Y_{i+1})
方向：θ_i	方向：θ_{i+1}

区間切り替え

・区間の切り換え点にX, Y座標と進入方向θを定義する。
・次の区間の進入方向に応じて、走行体の座標と切り換え点の座標を下記条件式で比較し、次区間へ進入したかどうかを推定する。

$(X - P_x)\cos\theta + (Y - P_y)\sin\theta \geq 0$

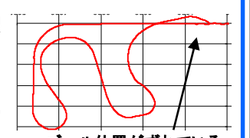


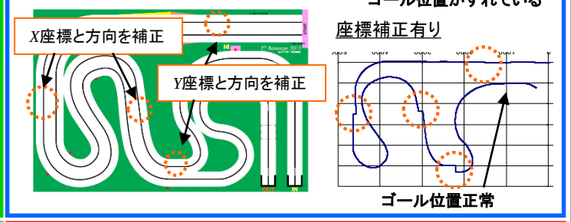


上図の場合次の進入方向は0度なので条件式より[$X \geqq P_x$]の時に次区間と判定。
切換点定義例

座標補正

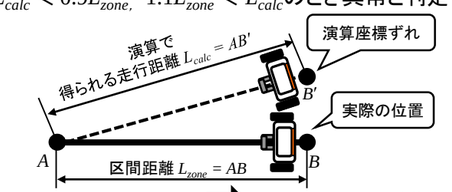
・走行体の状態が安定する直線コース上で、直線検知をしてふらつきがないことを確認し、X or Y 座標と方向を補正する。





推定異常判定

・異常発生時は、認識している座標にずれが生じるため、実際のコースの長さである区間距離L_{zone}と、座標演算より求めた区間の走行距離L_{calc}との差が大きくなる。
・実際の走行データを確認したところ、L_{zone}とL_{calc}の差は、正常の場合は、±10%以内に収まる。
・$L_{calc} < 0.9L_{zone}$, $1.1L_{zone} < L_{calc}$のとき異常と判定する。

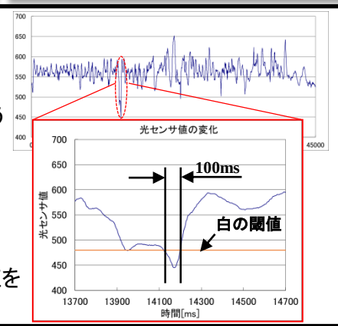


区間ごとにL_{zone}とL_{calc}を比較 ⇒ $L_{calc} > 1.1L_{zone}$のとき推定異常

ライン逸脱検知

【課題】
・正常走行と座標トレース走行（→ P.4 制御戦略）の切り替えを行うために、ライン逸脱を検知する必要がある。

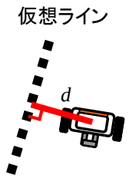
【対策】
・光センサ値より白と判定した時間が一定時間（閾値）を超過した場合にラインから逸脱したと判断する。
・スタートからゴールまでの走行を10回計測した際、白と判定した時間は最長100msであった。以上より、2倍のマージンを見て閾値を200msとし、超過したらライン逸脱状態とみなす。



座標トレース

【課題】
・ライン逸脱した場合も、速度を落とすことなくライン復帰し、ゴールまで走行する。

【対策】
・ライン逸脱を検知した後、進行方向に実際のラインと交差する仮想ラインの座標を生成する。
・輝度PIDの代わりに現在地と仮想ラインとの距離dから旋回量を取得して走行する。



距離dによる旋回量
$= d\,K_P + \int d\,K_I + \Delta d\,K_D$