

目標 “確実”にリザルトタイム0秒を目指す

【Ⅰ】目標実現するために

- ①目標リザルトタイムを達成するために必要な機能をマインドマップで導出し、機能実現に必要な要件を要求図で整理。
- ②“確実”に目標を達成するために、目標未達になる機能不全モードを分析し、対策を講じることで、目標達成率を高める。

【Ⅱ】目標達成を確実にする施策

ステップ1：目標未達になる機能不全モード/影響を洗い出す

★走行に必要な基本機能の機能不全モード/影響を一部ピックアップ★

機能	機能不全モード	影響
キャリブレーション	意図せず走行中にキャリブレーションが実行される	(走行停止による)リタイヤ
走行開始	意図せず走行が開始される	(フライングによる)ボーナスタイム減点
ライントレース	意図通りにライントレースができない	転倒/コースアウト/走行妨害

ステップ2：目標未達のリスクが高い機能不全モードに対策方針を定義する

※「リスク」は、影響度(1~5)×頻度(1~5)により算出。「重要度」は、低(リスク5未満)、中(リスク5~9)、高(リスク10~14)、最高(リスク15以上)と定義。なお、重要度「低」はリスクが低いため対策を実施しない。

★走行に必要な基本機能の機能不全モードにおける対策方針を一部ピックアップ★

機能不全モード	影響	頻度	リスク	重要度	対策方針
意図せず走行中にキャリブレーションが実行される	4	1	4	低	-
意図せず走行が開始される	3	3	9	中	意図しない走行開始を防止する
意図通りにライントレースができない	5	4	20	最高	意図通りにライントレースできるようにする

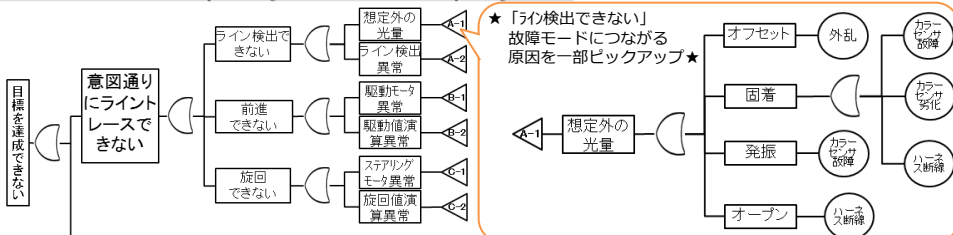
ステップ3：機能不全モードにつながる原因分析(FMEA)と対策

※「リスク」は、故障影響度(1~5)×故障発生頻度(1~5)×故障検出難易度(1~5)により算出。なお、リスクが低いもの(2×2×2=リスク8以下)は対策を実施しない。

★走行に必要な基本機能の機能不全モードにつながる原因と対策方針に従った対策を一部ピックアップ★

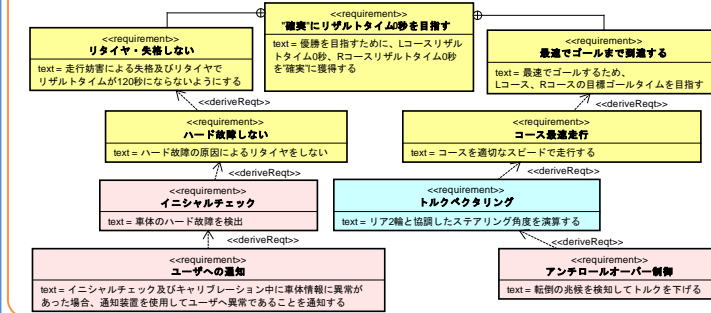
HW/SW	原因	故障モード	影響	故障影響度	故障発生頻度	故障検出難易度	リスク	対策要否	対策
タッチセンサ	故障、劣化	固着	意図しない走行開始	4	2	2	16	○	走行前にインシャルチェックで異常を検出
	ソフトバグ	ON/OFF反転	意図しない走行開始	4	1	1	4	-	-
トルクベクタリング	外乱	演算値の急激な上昇	意図通りにライントレースできない	5	4	2	40	○	転倒でライントレースができない前にはトルクを下げる(アンチロールオーバー制御)
		演算値の急激な低下	意図通りにライントレースできない	5	4	2	40	○	-

ステップ4：原因分析(FMEA)の正しさを別手法(FTA)で検証



【Ⅲ】要件分析 (SysML：要求図)

★走行に必要な機能で【Ⅱ】によって追加された対策を一部ピックアップ★
(見やすくするために配置を変更)



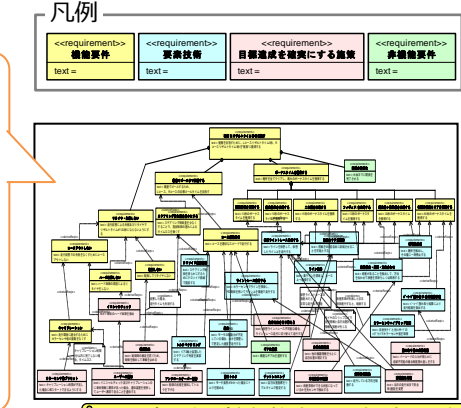
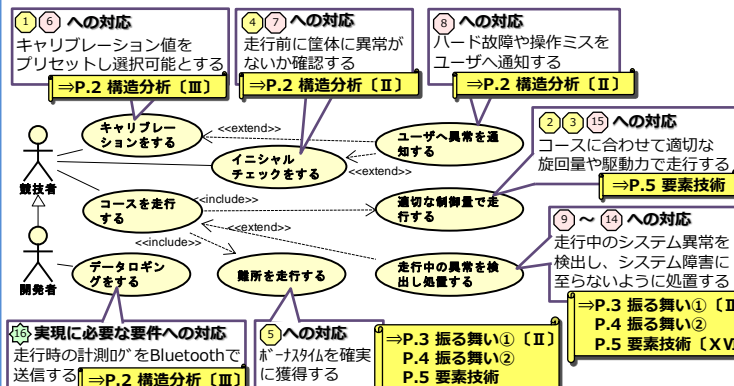
機能要件一覧

要件名	概要
キャリブレーション	会場に合わせてカラーセンサ値を適合
ｽﾌﾟﾘﾝｸﾞ変化量最適化	旋回制御の遅れによるタイムロス無くす
コース最速走行	コースを適切なスピードで走行
リタイヤ/失格しない	コースアウト/ハード故障/転倒しない
ボーナスタイムを獲得	各種難所ボーナスタイムを獲得

目標達成を確実にする施策(要件)一覧

要件名	概要
カラーセンサ値アダプ	試走時のカラーセンサ値をアダプ
インシャルチェック	車体のハード故障を検出
ユーザへの通知	インシャルチェック/キャリブレーションの異常を通知
転倒検出	転倒検出して車輪を止める
アンチロールオーバー制御	転倒の兆候を検出してトルクを下げる
走行方法切り替え	仮想ライントレースとライントレースを切り替える
自己位置補正	左右の車輪スレ補正や車両推定座標補正
段差衝突回避	段差にぶつかりが衝突しないよう進入位置調整
読み取り値可否確認	バーコード読み取り異常の場合再読み取り
環境影響対策	走行状況に応じてPDR起動後に速度調整 会場環境(フラッシュ等)による外乱の影響を抑える

【Ⅳ】機能要件分析 (UML：ユースケース図)



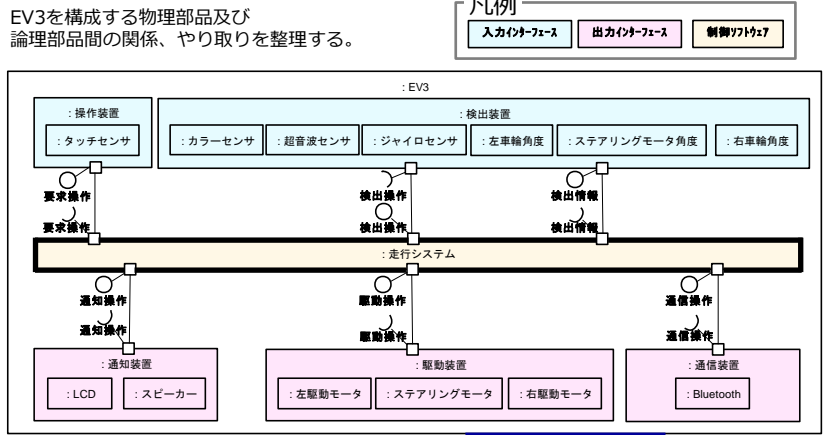
「バーコード」読み取りと走行経路決定 ⇒ P.3 振る舞い①【Ⅱ】
障害物検出 ⇒ P.4 振る舞い②【Ⅲ】
各要素技術 ⇒ P.5 要素技術

要素技術一覧	要素技術名	概要
①	クロソイド曲線旋回	クロソイド曲線で滑らかに旋回
②	トルクベクタリング	リア2輪と協調したステアリング角度を演算
③	速度PID	目標速度で走行するためにモータの駆動力不足を補正
④	ライントレース走行	カラーセンサでラインを探検しPID制御で走行
⑤	仮想ライントレース走行	仮想経路と現在のズレをフィードバックして走行
⑥	走行方向転換	仮想ライントレース走行で走行方向を転換
⑦	ライン復帰	カラーセンサで黒ラインを探検し、ラインへ復帰
⑧	段差検出・昇降・方位合わせ	段差を検出し、方位を合わせ、段差昇降/降段
⑨	障害物検出	超音波センサで障害物(新幹線)を検出
⑩	バーコード読み取りと走行経路決定	バーコード読み取り結果により走行経路を算出する
⑪	カラーセンサ/リアリッド制御	反射光モードとカラーモードを交互に切り替える
⑫	弱めトルク制御	モータ負荷がかかった場合にトルクを弱める
⑬	最適ステア角制御	ステアの移動角度を減らし高速化
⑭	デッドレコニング	自己位置座標をリアルタイムで推定
⑮	非機能要件一覧	
⑯	要件名	概要
⑰	開発の効率化	走行時の計測ログを取得するソフトを開発
⑱	ギア比の選択	最適なギア比を選択

UC名	コースを走行する
概要	競走者に選択されたコースを走行する
アクター	競走者
事前条件	競走者が走行するコースを選択していること
事後条件	コース全てを走行完了していること
基本系列	1.競走者は、タッチセンサを押下する 2.システムは、走行するコースを確認する 3.システムは、走行状態を確認する 4.システムは、走行するコースの走行方法を決定する 5.システムは、モータへの駆動要求値を演算する 6.本ユースケースを終了する
代替系列	-
例外系列	基本系列5で駆動要求値と実駆動値が異なっていた場合は、UC「走行中の異常を検出し処置する」を実施する

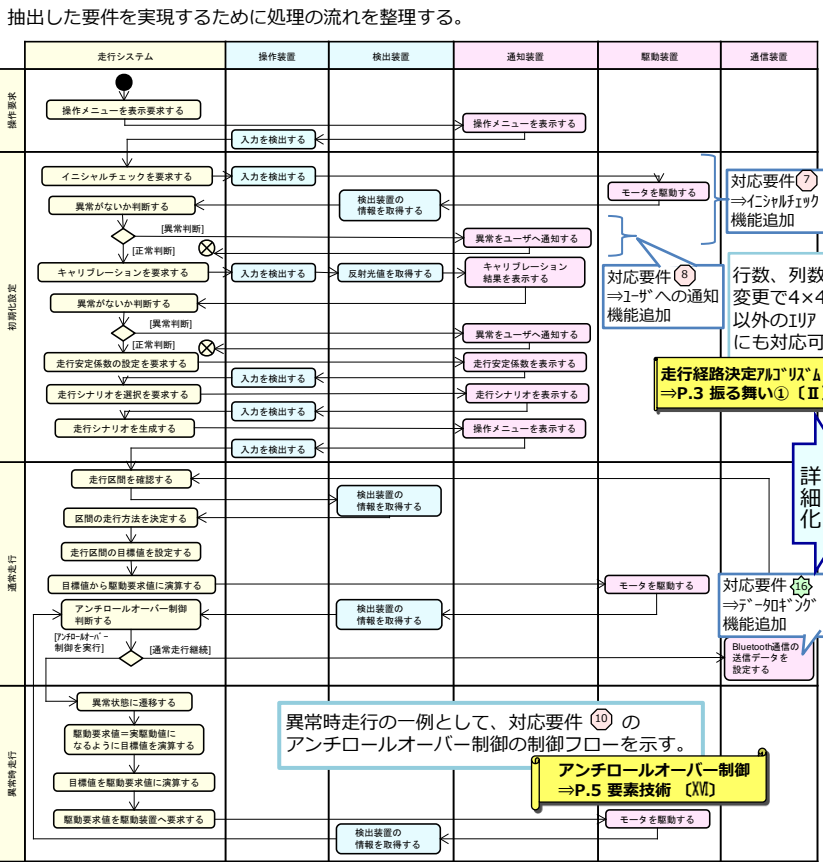
【Ⅰ】物理部品・論理部品間の整理

(SysML：内部ブロック図)



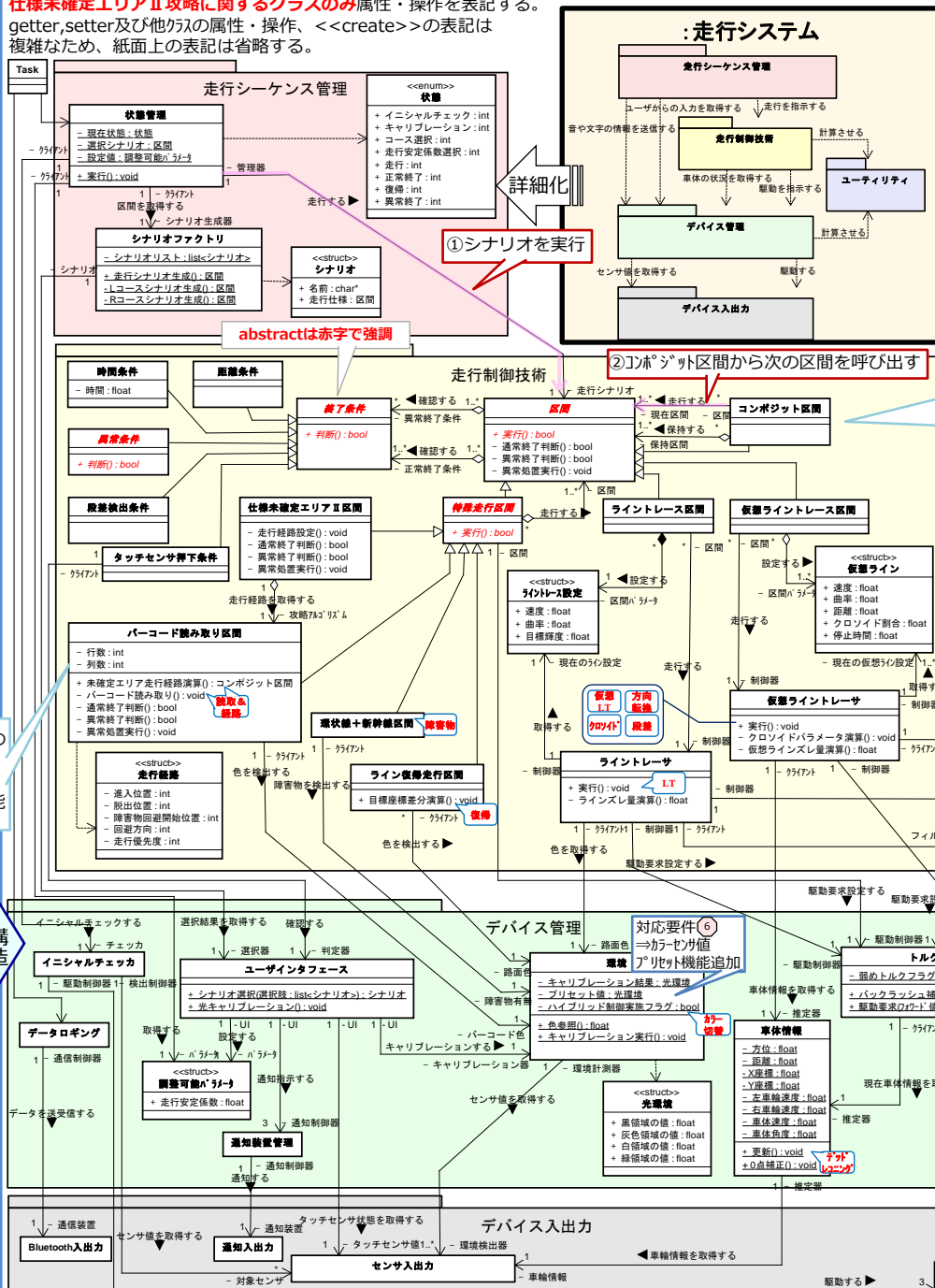
【Ⅱ】全体処理手順の整理

(UML：アクティビティ図)



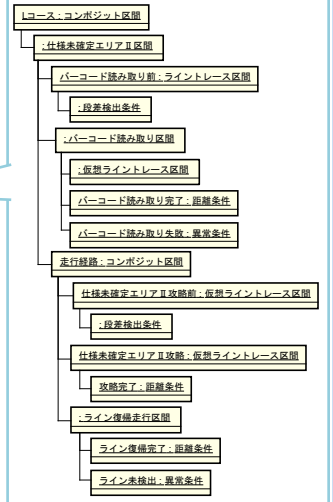
【Ⅲ】論理部品内の構造分析

(UML：クラス図) P.2 【Ⅰ】部品間の関係整理で示した「走行システム」内の構造を分析する。(UML：パッケージ図)



☆ピックアップシナリオを実行し、コンジョイント区間の呼び出しまでの制御フローについて示す。
→ で実行されるフローを示し、① で実行順序と内容を示す。

○コンジョイント区間に各区間のオブジェクトを格納。(紙面の都合上、仕様未確定エリアⅡ区間のみ表記)(UML：オブジェクト図)



P.1 【Ⅲ】要件分析で示した要素技術のアイコン(***)で各クラスの関連する要素技術を示す。
アイコンは全ての走行区間に必要な要素技術を示す。

① B-コード読み取り & 走行経路決定 ⇒ P.3 振る舞い①【Ⅱ】
② 障害物検出 ⇒ P.4 振る舞い②【Ⅲ】
③ 各要素技術 ⇒ P.5 要素技術

※P.3【Ⅱ】でシークス図とスタートマップ図を用いて示した詳細構造と振る舞いの関係は紙面の都合上、以降省略する。

【Ⅰ】二本橋走行戦略 (UML: アクティビティ図)

走行仕様

○: 仮想ライトレース走行

通常フロー

例外

走行の流れ

① 段差検出 ② 段差昇段 ③ 段差降段 ④ ライン復帰

二本橋は上記図①～④の順に走行する。仮想ライトレース走行にて二本橋の「コ」の字の奥まで侵入し、前輪が段差に触れたところで段差検出する。段差検出後、段差昇段し二本橋上を走行する。

目標達成を確実にする施策

ミケース: 前輪が二本橋の間を通過できない
⇒「コ」の字の端で段差検出してしまふ。
対策: 自己位置推定により二本橋付近に到達したら前輪がライン上となるように仮想ライトレース走行させる。手前で段差検出した場合は一旦後退し再度実施する。

右エッジのまま走行すると二本橋の端で段差検出してしまふ
仮想ライトレースにより車両位置を補正することで二本橋の奥まで進入可能

【Ⅱ】フィギュアL走行戦略 (UML: アクティビティ図)

走行仕様

○: ライトレース走行
○: 仮想ライトレース走行

基本フロー

例外

走行の流れ

フィギュアLは以下①～⑤の順に走行する。

仮想ライン走行パラメータ

	距離 [mm]	曲率 [1/mm]	速度 (Fwd)	クロソイド割合 [%]
①	70	0	30	0
②	267	0.1333	30	100
③	242	-0.00667	30	100
④	175	0	30	0

対策: フィギュアL後輪昇段時に「段差方位合わせ」を実施。
⇒段差方位合わせにより得られた角度を使用し仮想ライトレースを実施することで、左右のズレ量を±20mmまで許容できる。

目標達成を確実にする施策

ミケース: 段差を昇る際にスリップする。
⇒車両方位や左右のズレが合わさりフィギュアLを攻略中に脱輪する。

左右のズレ 方位のズレ

段差方位合わせ ⇒ P.5 要素技術【XIV】

【Ⅲ】環状線+新幹線走行戦略 (UML: アクティビティ図)

走行仕様

○: ライトレース走行
○: 仮想ライトレース走行

基本フロー

例外

走行の流れ

① 緑枠内は超音波探索範囲。 ② 手前線路探索位置 = (185-走行距離) ± 75mm ③ 奥側線路探索位置 = (785-走行距離) ± 75mm ④ 探索可能範囲 (赤色範囲) ⑤ 障害物検出範囲

新幹線検出方法

～障害物検出閾値～
障害物を100%検出できる場所を、障害物を前方～1000mm、左右±200mmに設置し測定。(筐体3台で10回計測)
⇒左右±100mm、奥行き100mm～900mmの範囲を探索できる。
～新幹線速度推定～
手前線路or奥側線路検出後、次の新幹線検出までの時間をカウント。半周約1000mmの為下記の計算式により計算する。
⇒新幹線速度(mm/s) = 1000 / 検出時間(s)
～待ち時間の設定～
プラ板が付いていない後部車両が通過するまで待機する。
⇒待ち時間n = (後部車両長さ) / (新幹線速度)

目標達成を確実にする施策

ミケース: 超音波センサで新幹線を検出できない。
⇒その場で待機し続け、難所を攻略できない。
対策: 新幹線を検出できない理由は以下の2つが考えられる。
・新幹線が脱線もしくは電池が切れている
・EV3の超音波センサが故障し、作動していない
超音波センサ故障は走行前にインシャルチェックで検出。(走行中のセンサ故障は発生頻度が低いため、対策しない。)
超音波センサ検出開始から20秒が経過したら、新幹線が走行していないと判断して前進開始する。

【Ⅳ】縦列駐車・直角駐車走行戦略 (UML: アクティビティ図)

走行仕様

○: ライトレース走行
○: 仮想ライトレース走行

通常フロー

例外

走行の流れ

目標達成を確実にする施策

ミケース: 駐車走行までにタイヤスリップ等で距離ズレが生じる。
⇒駐車位置のズレが生じ、正確な位置への駐車ができない
対策: エッジを検出し車両の走行距離をリセットすることで、走行距離の正確性を保持して、走行距離のズレによる失敗の可能性を低くする。

エッジを検出し、自己位置補正

仮想ライン走行パラメータ

直角駐車

	距離 [mm]	曲率 [1/mm]	速度 (Fwd)	クロソイド割合 [%]	停止時間 [ms]
①					
②					
③	215.1	-0.0073	-30	100	300
④	346	0	-30	0	

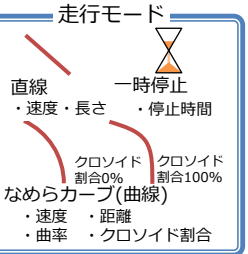
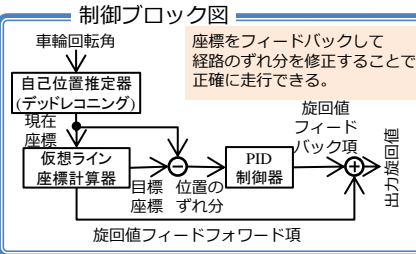
縦列駐車

	距離 [mm]	曲率 [1/mm]	速度 (Fwd)	クロソイド割合 [%]	停止時間 [ms]
①'					
②'					
③'					
④'	362	0.019231	-30	100	300
⑤'	355	-0.02	-30	100	
⑥'	152	0	-30	0	

【I】仮想ライントレース走行

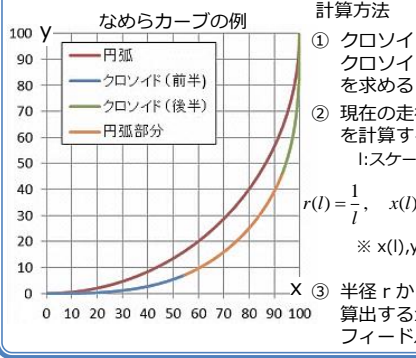
仮想 LT

要件②③④⑤への対応



なめらかカーブ

円弧をクロソイド曲線で挟んだカーブを定義。クロソイド部分はステアリングの操舵速度が一定になり、なめらかに走行できる。



クロソイド

計算方法

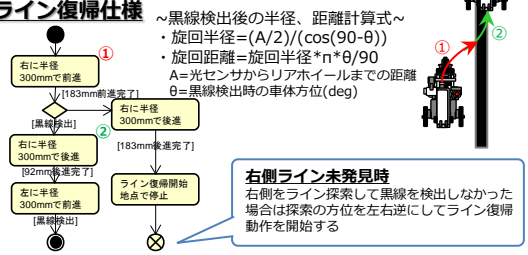
- ① クロソイド部分の終点座標と旋回角度からクロソイドパラメータ(スケーリング因子)を求める
- ② 現在の走行距離から、半径rとx,y座標を計算する
- ③ 半径rからフィードフォワード項を算出するx,y座標は走行位置のフィードバック制御に用いる

【VIII】ライン復帰

要件④⑤への対応

復帰

ライン復帰は、黒線を検出した後、ライン上に乗るように仮想ライントレースを行う。その結果、ラインを探索完了後、スムーズにライントレースを開始することが出来る。



【IX】走行方向転換

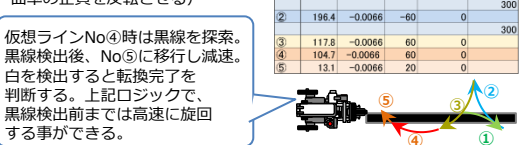
要件③⑤への対応

方向転換

仮想ライントレースにて転換を行う。(転換後に左エッジトレスを行う場合は、右図仮想ライントレースパラメータで指示する曲率の正負を反転させる)

右エッジトレス開始用仮想ライントレースパラメータ

距離 [mm]	曲率 [1/mm]	速度 [Fwd]	クロソイド割合 [%]	停止時間 [ms]
1	39.3	-0.0066	60	0
2	196.4	-0.0066	-60	300
3	117.8	-0.0066	60	0
4	104.7	-0.0066	60	0
5	13.1	-0.0066	20	0

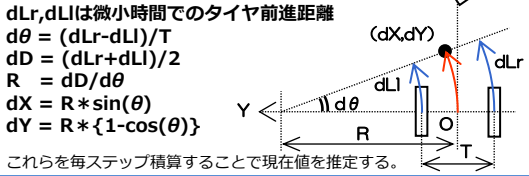


【II】デッドレコニング

デッドレコニング

要件③⑤への対応

リア2輪の回転角から、車体方位、走行距離、自己位置座標をリアルタイムで推定する。

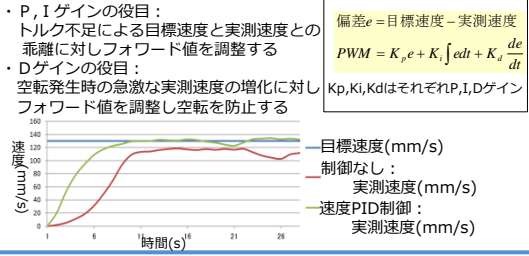


【III】速度PID

速度PID

要件③⑤への対応

目標速度に対して、実測速度がどの程度出ているかを測定。実測速度に対して速度PID制御を行う。出力PWM値の計算方法



【IV】ギア比選択理由

ギア比

要件④への対応

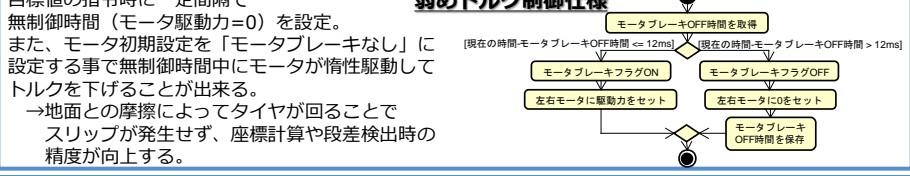
観点	ギア比	1	3	5
二橋昇降	◎	△	×	(不可)
速度	○	◎	◎	◎

【V】弱めトルク制御

弱めトルク

要件③⑤への対応

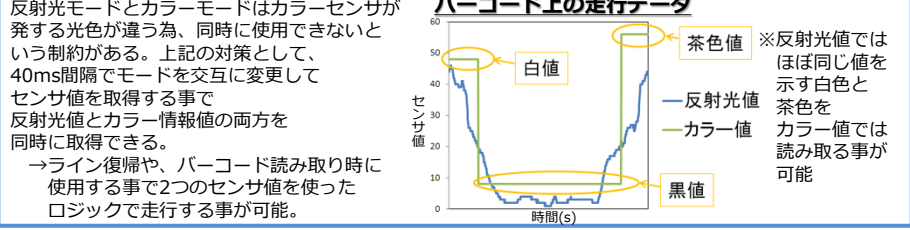
段差衝突時や、仮想ライントレース時に、リアのトルクを弱める事でリアタイヤのスリップ防止が可能。



【VI】反射光&カラーモード ハイブリッド制御

カラー切替

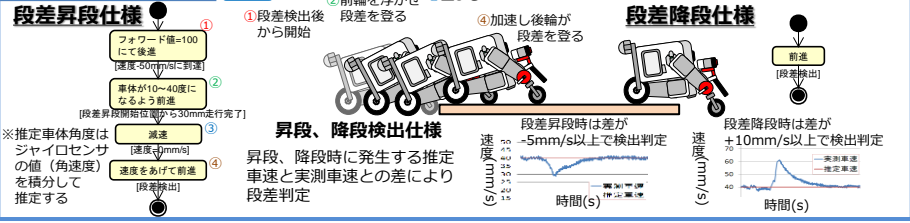
要件④⑤への対応



【VII】段差昇降

段差

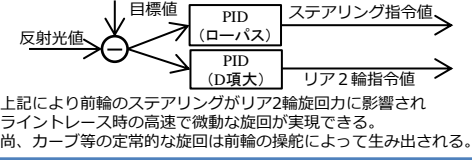
要件⑤への対応



【X】高応答ライントレース

LT

ステアリング操舵速度はリア2輪の旋回力と比べて応答性が悪い。そこで、ステアリング指令値と、リア2輪指令値を別々にチューニングし高応答なレースを実現。

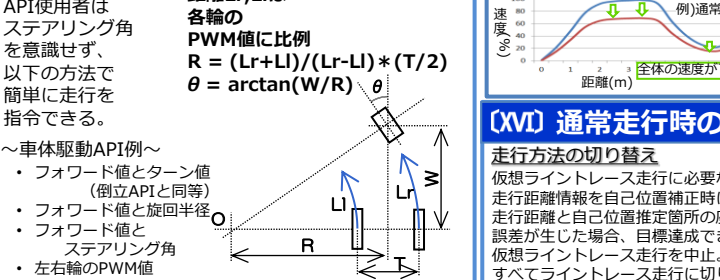


【XI】トルクベクタリング

トルク配分

要件③⑤への対応

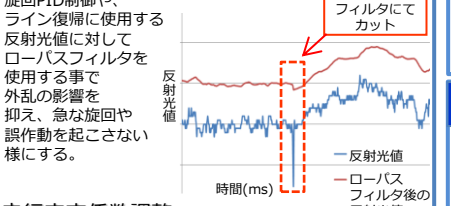
リア2輪出力と協調した最適舵角を自動決定。API使用者はステアリング角を意識せず、以下の方で簡単に走行を指令できる。



【XII】環境影響対策

要件④⑤への対応

カラーセンサのローパスフィルタ



走行安定係数調整

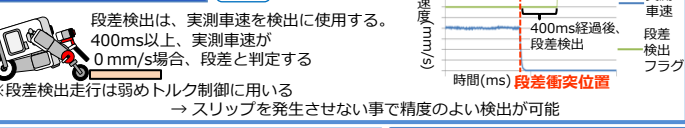
当日の周囲環境(光、路面)や状況(自、他チームタイム)に応じてプログラム起動後に走行安定係数(速度、車体安定待ち時間等)に影響を調整する事が可能。



【XIII】段差検出

段差

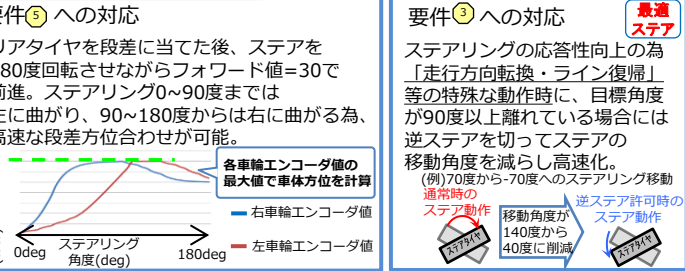
要件⑤への対応



【XIV】段差方位合わせ

段差

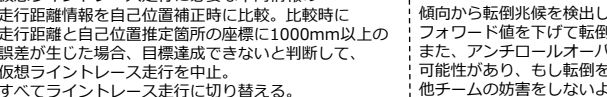
要件⑤への対応



ステアリングの応答性向上の為「走行方向転換・ライン復帰」等の特殊な動作時に、目標角度が90度以上離れている場合には逆ステアを切ってステアの移動角度を減らし高速化。(例)70度から-70度へのステアリング移動通常動作のステア動作逆ステア許可時のステア動作

【XVI】通常走行時の目標達成を確実にする施策

走行方法の切り替え



アンチロールオーバー制御・転倒検出

