ETロボコン2018

デベロッパー部門 プライマリークラス

地区：南関東　地域：神奈川県平塚市

チームNo. 048　チーム名：コマツロボコン同好会　所属：コマツ

チーム紹介、目標、意気込み

私たち「コマツロボコン同好会」 は、株式会社小松製作所の社員チームです。今年度のチームはICT技術の開発を行っている若手社員４名で構成されています。弊社では一昨年から若手社員がETロボコンに参加し、昨年は走行１位でありながら、モデル評価で順位を落とし、全国大会出場が叶いませんでした。そのため、今年は特にモデル作成に力を注ぎ、汎用性の高く、機能、構造、振る舞いに一貫性のあるモデルを目指して取り組んできました。今年こそ、全国大会出場を果たし、今回培ってきたUMLの知識を実務の開発でも活かして生きたいと思っております。

モデルの概要

選択課題を「シーソーを通過する」とした。
モデルの概要に関して下記表にまとめた。

ページ	タイトル	概要
P2	機能モデル	基本戦略を基にユースケース図を作成し、ミスユースケース図を用いて脅威と緩和策を示した。そこから具体的な走行戦術を表にまとめた。
P3,P4	構造モデル	走行戦術を基にクラス図を導出する過程を記載し(P3)、クラス図とパッケージ図を記載した(P4)。
P5	振る舞いモデル	オブジェクト間の振る舞いをシーケンス図に記載した。
P6	工夫点	「直進性向上」及び「灰色区間検知」に関して記載した。

モデルの構成

1．機能モデル

1-1 基本戦略

シーソーをダブルで攻略するための戦略を記載した。

1-2 要求分析

基本戦略を基にユースケース図を用いて必要な機能を抽出した。
また、ミスユースケース図を用いて脅威と緩和策をまとめた。

1-3 走行戦術

要求分析の結果から具体的な走行戦術を表にまとめた。
走行(走行方法、速度、尻尾状態）が変更するタイミングで区間を区切り、一つの区間を「攻略フェーズ」と名付け、攻略フェーズ毎の走行と攻略フェーズ切り替え条件をまとめた。

2．構造モデル

2-1 導出過程

走行戦術からクラス図を導出する過程を記載した。
導出手順はシステムの「核となるクラスの導出」→「細部のクラスの導出」の手順で行った。

2-2 クラス図：全体のクラス図を記載した。

2-3 パッケージ図

導出したクラスから共通のまとまり毎にパッケージに分類し、パッケージ間の関係をパッケージ図にまとめた。

3．振る舞いモデル

シーソーを通過する全体のシーケンス図を示し、共通部分は別途抽出して記載した。

4．工夫点

4-1 直進性向上：左右のモータの個体差を埋めるロジックを記載した。

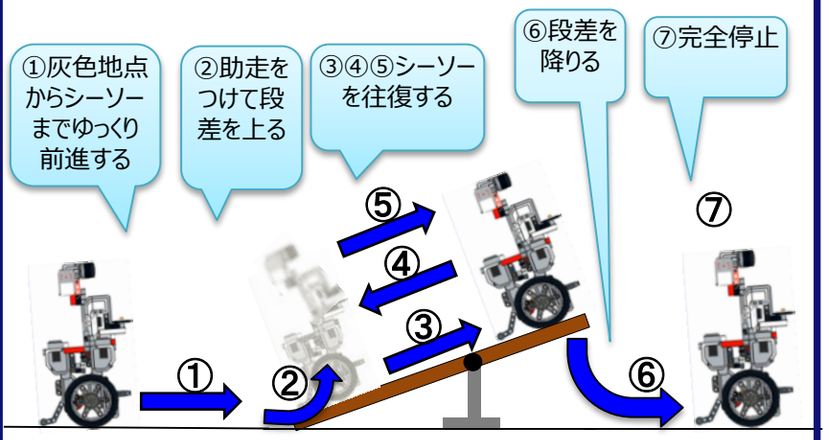
4-2 灰色区間検知：黒ライン走行時と灰色ライン走行時の輝度値の変化量から灰色を検知するロジックを記載した。

1. 機能モデル

コマツロボコン同好会

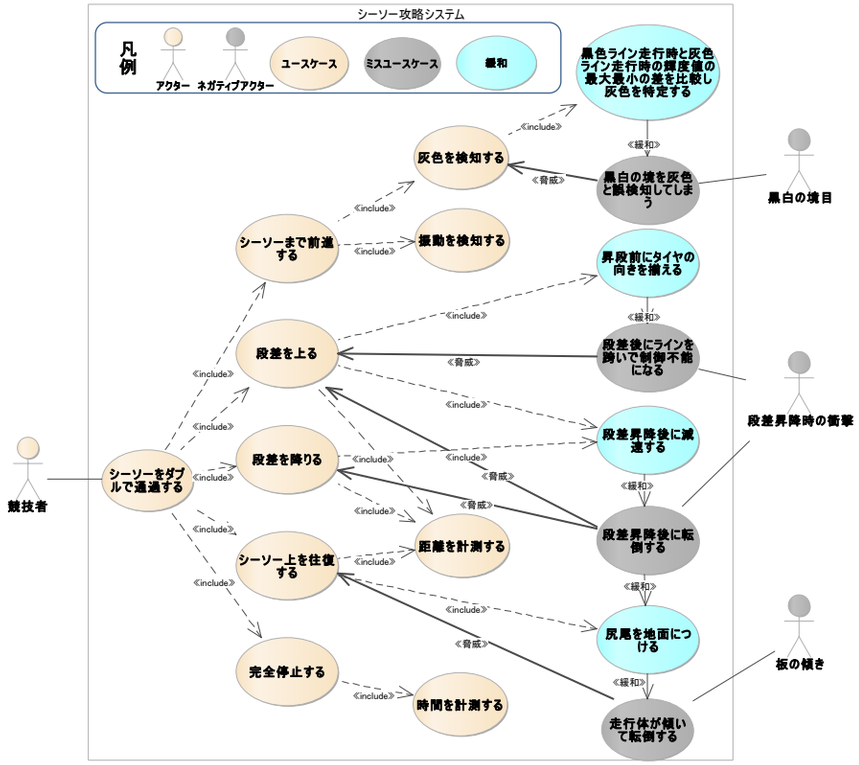
1 基本戦略

シーソーを通過する基本戦略を示す。
ダブルで通過することを想定し①～⑦の
ステップの通り走行する。
振動検知によってシーソーの段差を検知し、
段差昇段の判断やシーソー上の前進後進の
切り替えは距離計測結果を基にして行う。



2 要求分析

基本戦略を基に必要な機能を抽出した。機能に対してミスユースケース図を記載し、脅威と緩和策を下表にまとめた。

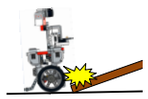
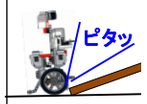
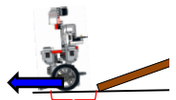
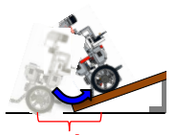
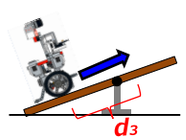
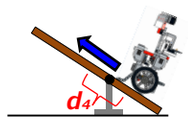
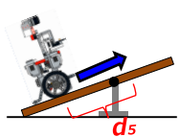
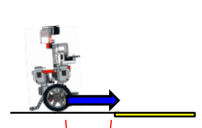


ミスユースケース　補足説明

	脅威	緩和策
黒白の境を灰色と誤検知してしまう	通常の黒色ラインの黒白の境を灰色と判定し、シーソー開始位置に達したと判断し、ゴール手前で失速してしまう。	黒色ライン走行時と灰色ライン走行時の輝度値の最大最小の差を比較し、灰色を特定する。**（工夫点2参照）**
段差後ラインを跨いで制御不能になる	段差に登る時に、左右方向にバランスを崩してシーソー上のラインを跨いでしまい、ライントレースできずに制御不能になる	段差に登る前に、シーソー端にタイヤを押し付けることで、両タイヤの向きをシーソーと平行になるように揃え昇段時にバランスが崩れる事を防ぐ。
段差昇降後に転倒する	段差昇降時の衝撃によってバランスを崩し、転倒する。	①段差昇降段後に尻尾を地面につけてバランスを保つ。 ②段差昇段時は段差をあがるため速度をつけているが、昇段後は減速する。
走行体が傾いて転倒する	シーソーの板の傾きによって走行体が傾いてバランスを崩し転倒する。	尻尾を地面につけてバランスを保つ

3 走行戦術

具体的な走行戦術を示す。ミスユースケースを織り込んだ戦略を基に走行（方法、速度、尻尾状態）を決め、走行が変更するタイミングで区間を区切り、区間を”攻略フェーズ”と名付けた。下記表に攻略フェーズ毎の走行及びフェーズの切り替え条件を示す。

		a.シーソー開始位置検知	b.段差検知	c.車輪方向調整	d.助走のため後進	e.シーソー昇段	f.シーソー上前進（1回目）	g.シーソー上後進	h.シーソー上前進（2回目）	i.シーソー降段	j.降段後前進	k.停止
攻略フェーズ												
		コース上の灰色点を見つけるまでライントレース	車輪がシーソー端に衝突するまでライントレース	両車輪がシーソー端に設置するまで低速で前進	車輪方向を保持し、助走のため後進	助走をつけて段差(シーソー端)を超える	一定距離を前進するまでシーソー上をライントレース	一定距離を後進するまでシーソー上をライントレース	一定距離を前進するまでシーソー上をライントレース	一定距離を前進するまでするまでライントレース	ラインのないコースを一定距離を前進	尻尾モータを下し、全モータを停止
走行	走行方法	ライントレース（前進）	ライントレース（前進）	前進	後進	前進	ライントレース（前進）	ライントレース（後進）	ライントレース（前進）	ライントレース（前進）	前進	停止
	速度	高速	低速	低速	高速	高速	低速	低速	低速	低速	低速	停止
	尻尾状態	上げる	上げる	上げる	上げる	下げる	下げる	下げる	下げる	下げる	下げる	下げる
攻略フェーズ切替判定		光センサの明度が指定値以内となる 灰色検知	加速度センサで検知した振動が指定値を超える 振動検知	左右車輪のモータ角度が一定時間内に変化しない 速度検知	モータ回転から計算した距離が助走に必要な距離***d1***を超える 距離計測	モータ回転から計算した距離がシーソー昇段に必要な距離***d2***を超える 距離計測	モータ回転から計算した距離がシーソー中心を超えるまでの距離***d3***を超える 距離計測	モータ回転から計算した距離がシーソー中心を超えるまでの距離***d4***を超える 距離計測	モータ回転から計算した距離がシーソー中心を超えるまでの距離***d5***を超える 距離計測	モータ回転から計算した距離がシーソー昇段に必要な距離***d6***を超える 距離計測	モータ回転から計算した距離がシーソー終端から停止エリア端までの距離***d7***を超える 距離計測	停止した時間が指定時間を超える 時間計測

2. 構造モデル　~クラス図導出過程~

コマツロボコン同好会

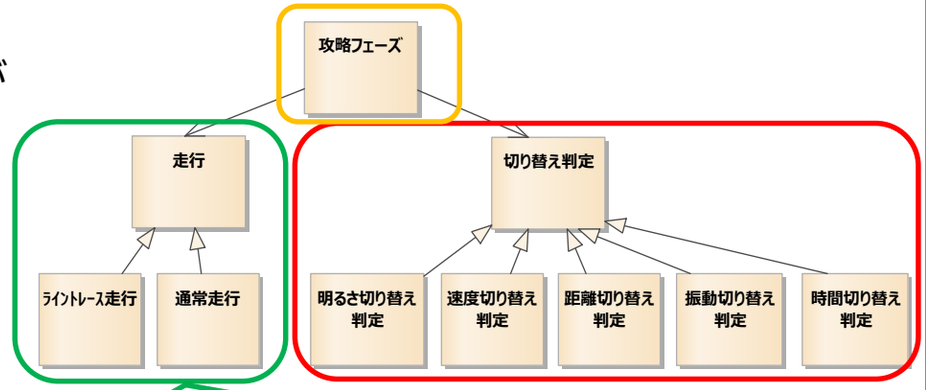
1 クラス図の導出過程

走行戦術からクラス図を導出した過程を示す。下記【１】【２】【３】の順序で導出した。

【１】核となるクラスの導出

機能モデルに記載の走行戦術を簡略化すると以下の表のように、「走行」と「攻略フェーズ切替判定」が「攻略フェーズ」に紐づいており、この３つを中心にクラス図を構成することを考えた。
走行と攻略フェーズ切替判定は下表の種類によって処理が異なる事から別クラスを作ることを考え、さらに「攻略フェーズ」クラスが走行や切り替え判定の種類を意識しなくてもいいように「走行」と「切り替え判定」の抽象クラスを作成した。

攻略フェーズ	a	b	c	d	e	f	g	h	i	j	k
走行	ﾗｲﾝﾄﾚｰｽ前進	ﾗｲﾝﾄﾚｰｽ前進	前進	後進	前進	ﾗｲﾝﾄﾚｰｽ前進	ﾗｲﾝﾄﾚｰｽ後進	ﾗｲﾝﾄﾚｰｽ前進	ﾗｲﾝﾄﾚｰｽ前進	前進	停止
攻略フェーズ切り替え判定	明るさ	速度	速度	距離	距離	距離	距離	距離	距離	距離	時間



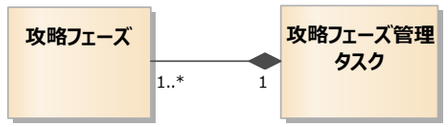
前進、後進は速度の±で表現できるのでクラスを分けない
ライントレース有りの場合は旋回量の計算に輝度値を考慮しなければならないのでライントレース有り/無しによってクラスを分ける

【２】細部のクラスの導出

【１】で大きく３つに分類した「攻略フェーズ」「走行」「切り替え判定」に関して詳細検討を行った。

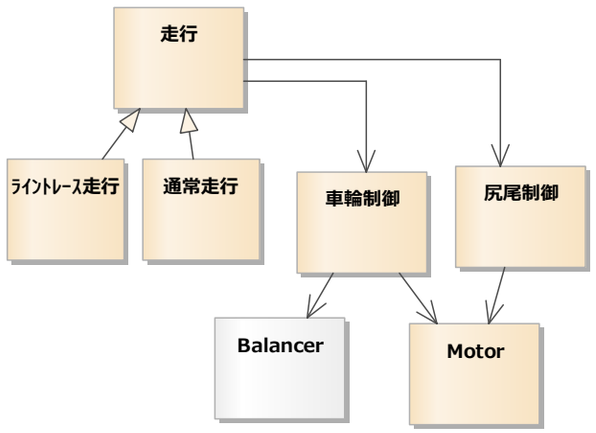
【2-1】攻略フェーズ

戦略図通りに実行されるのは、攻略フェーズを切り替える役割が必要であるため「攻略フェーズ管理タスク」クラスを追加する。



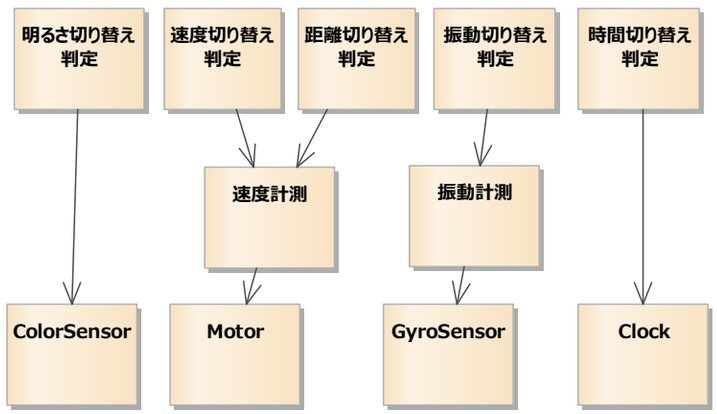
【2-2】走行

ﾗｲﾝﾄﾚｰｽ走行と通常走行ではどちらも走行速度と旋回速度を基にして車輪モータの出力値を設定するが、実際の出力値は倒立振子関数を通して決まる。また、尻尾モータの出力は尻尾角度のみを基に決まるので、「車輪制御」クラスと「尻尾制御」クラスに分けてモータの制御を行う。



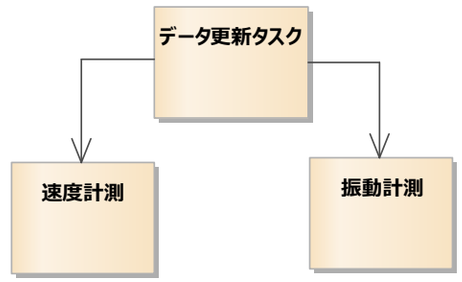
【2-3】切り替え判定

切り替え判定に使うパラメータを取得するためのAPIのクラスやデータの加工が必要な場合にAPIで取得したデータを計算する計測クラスを設ける。



【３】その他（データ更新）

速度・振動は、現在のデータ値と１制御周期前のデータ値を用いて計算する。したがって、攻略フェーズが切り替わった直後は、１制御周期前のモータ角位置やジャイロ角速度を計測できないため速度・振動を計算できない。
そのため、常時制御周期でデータの計測結果を更新するための「データ更新タスク」を設ける。

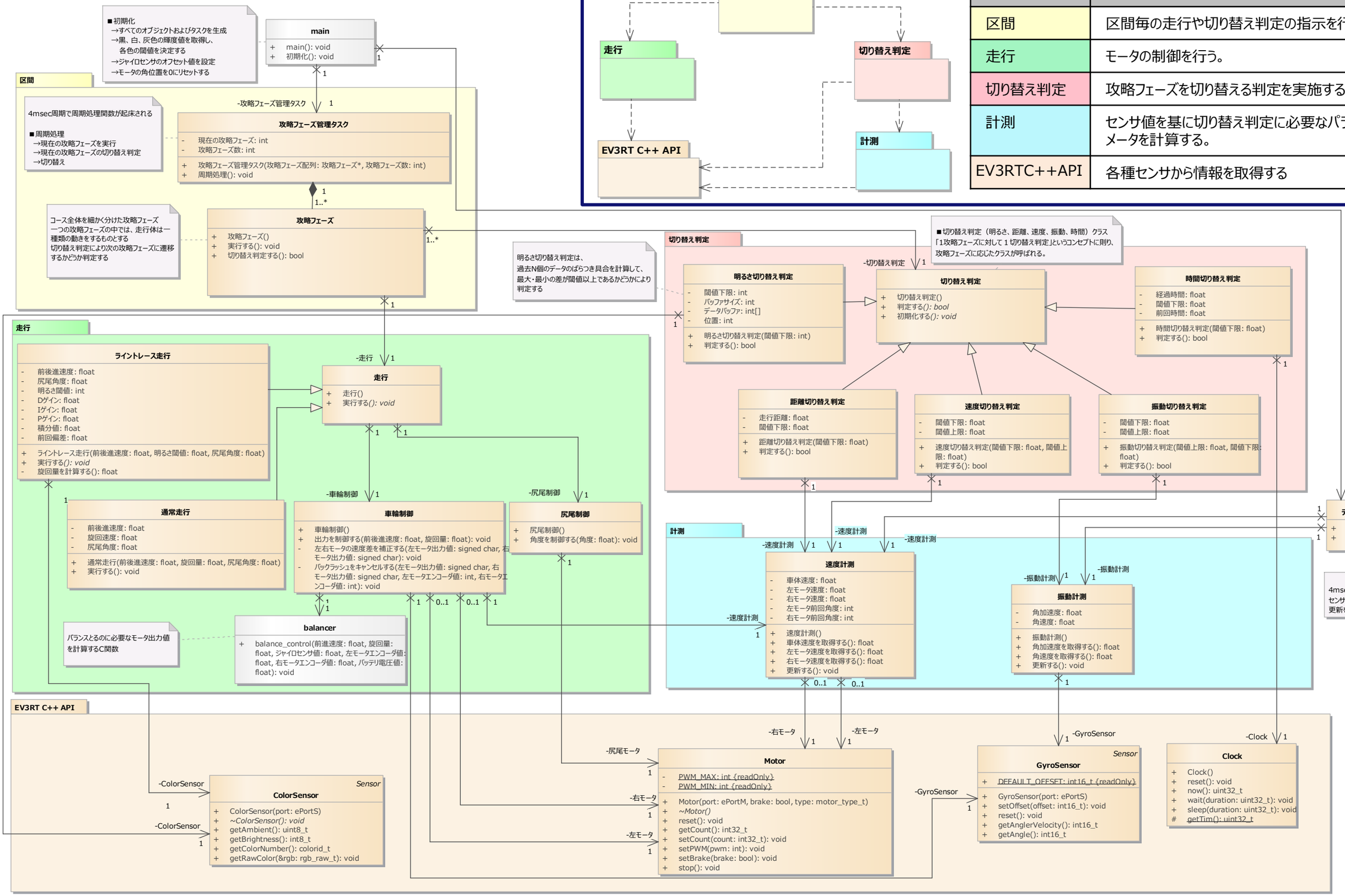


2. 構造モデル

コマツロボコン同好会

2 クラス図

前頁の導出過程の通り作成したクラス図を示す。



3 パッケージ図

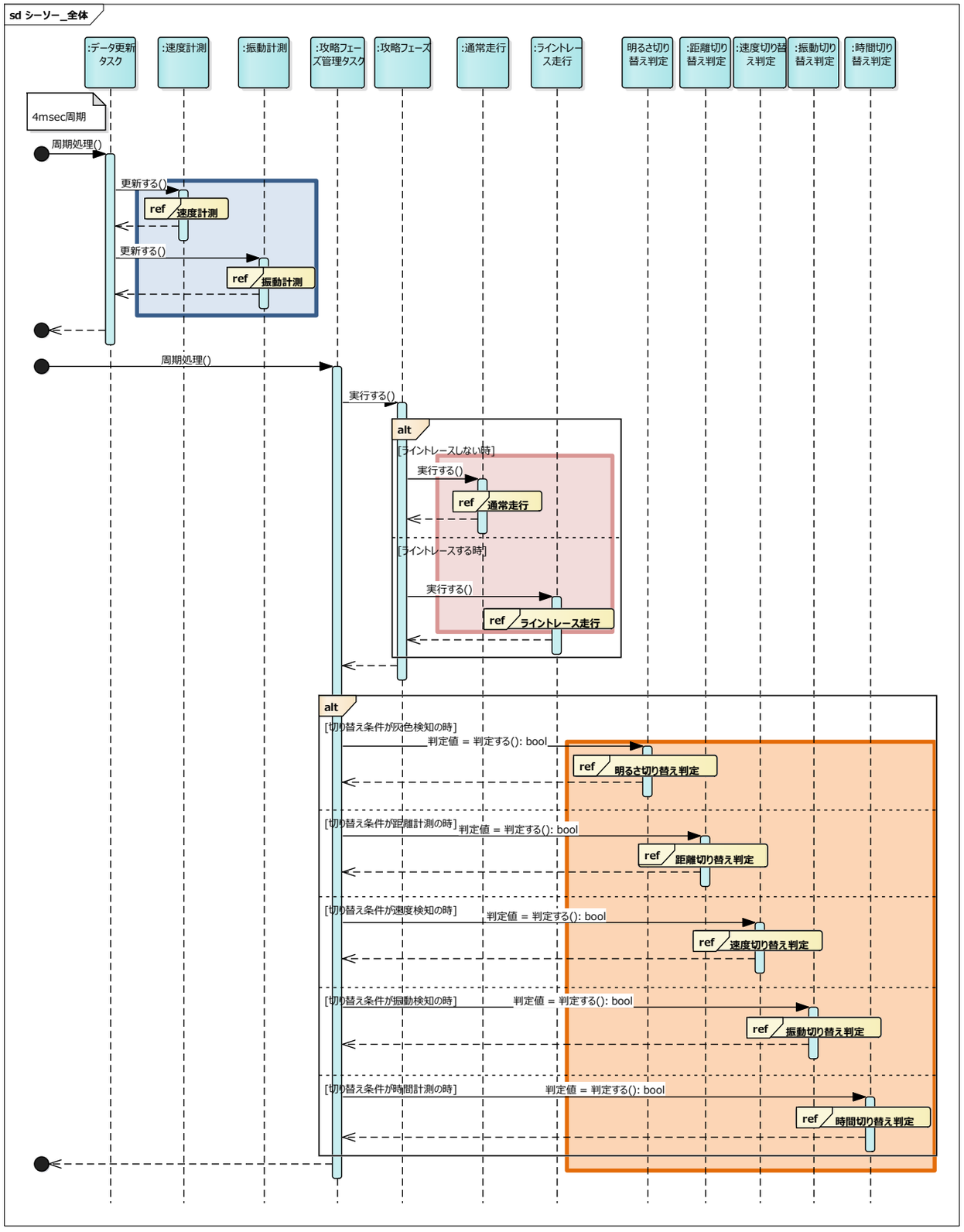
クラス図に記載された要素をパッケージを用いてカテゴリ毎にまとめた。
５つのパッケージ間の関係性をパッケージ図を用いて示す。

パッケージ	概要
区間	区間毎の走行や切り替え判定の指示を行う。
走行	モータの制御を行う。
切り替え判定	攻略フェーズを切り替える判定を実施する。
計測	センサ値を基に切り替え判定に必要なパラメータを計算する。
EV3RTC++API	各種センサから情報を取得する

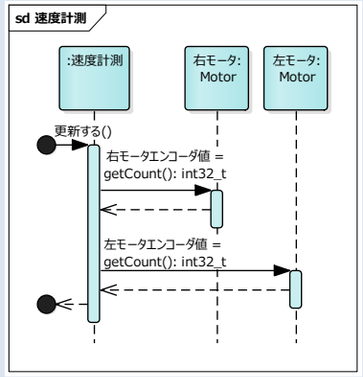
3. 振る舞いモデル

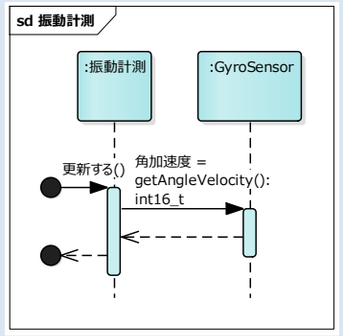
コマツロボコン同好会

シーケンス図を用いてオブジェクト間の振る舞いを示す。最初に全体の振る舞いを示し、「計測」「走行」「切り替え判定」に関して詳細な振る舞いを示した。

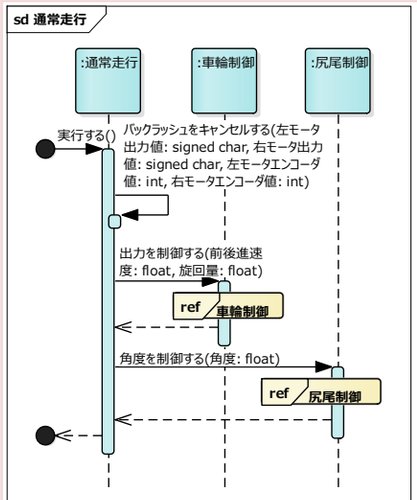


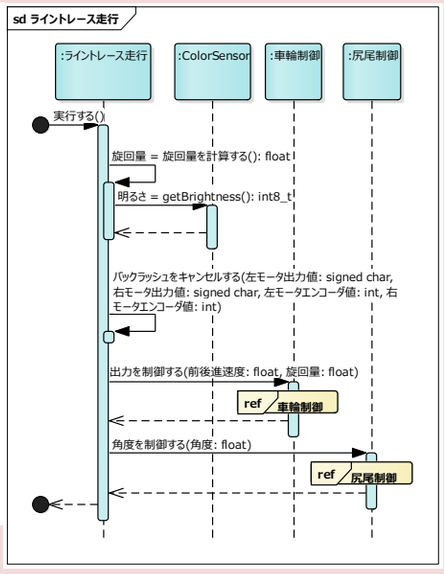
計測

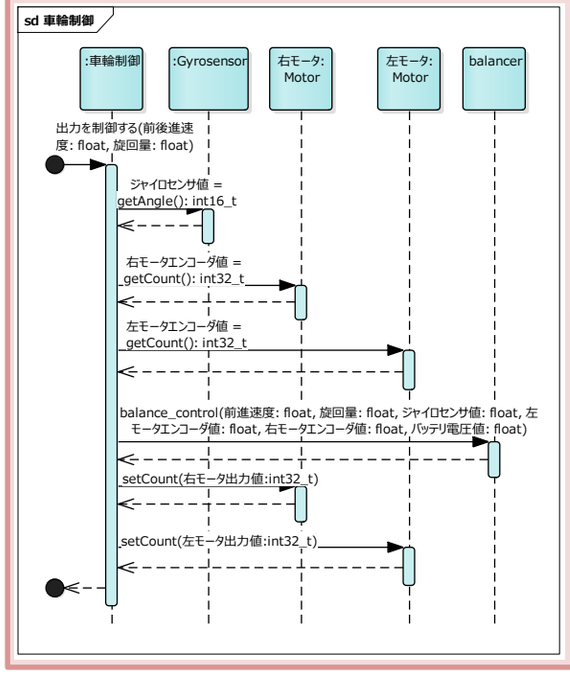


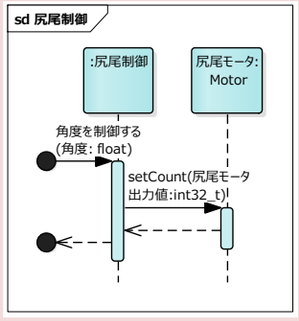


走行

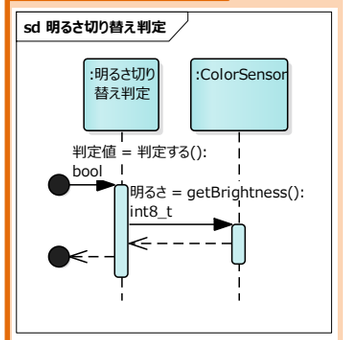


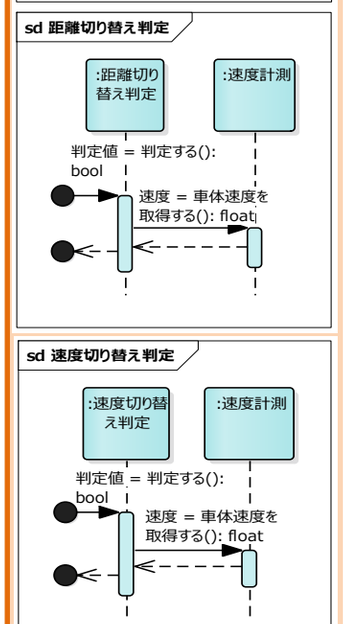


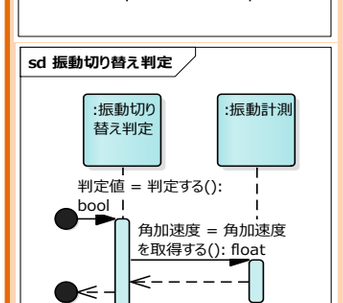


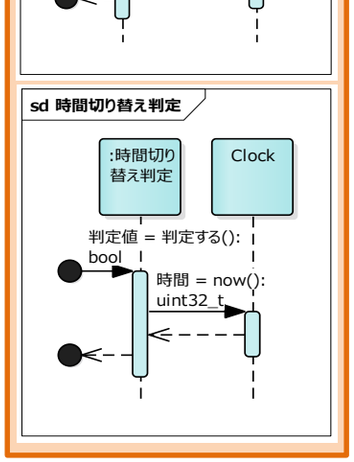


切り替え判定









4. 工夫点

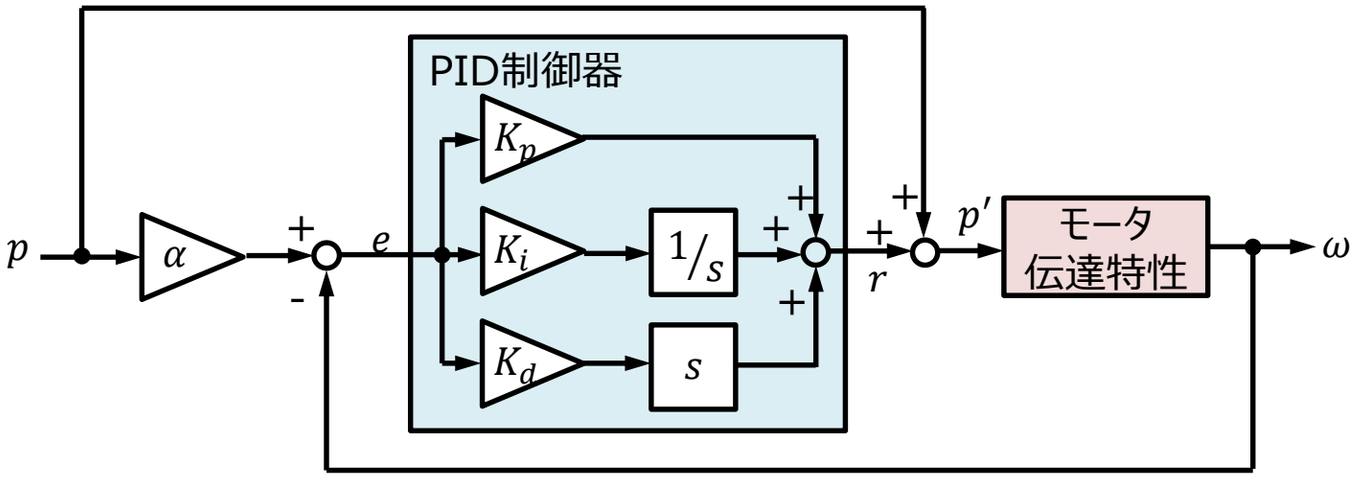
1 直進性の向上

【1-1】課題

モータ伝達特性の個体差により、同じ指令値を与えても、回転速度に差が生じる。
従って、ライントレース無しの前進時に真っ直ぐ進まない。

【1-2】対策

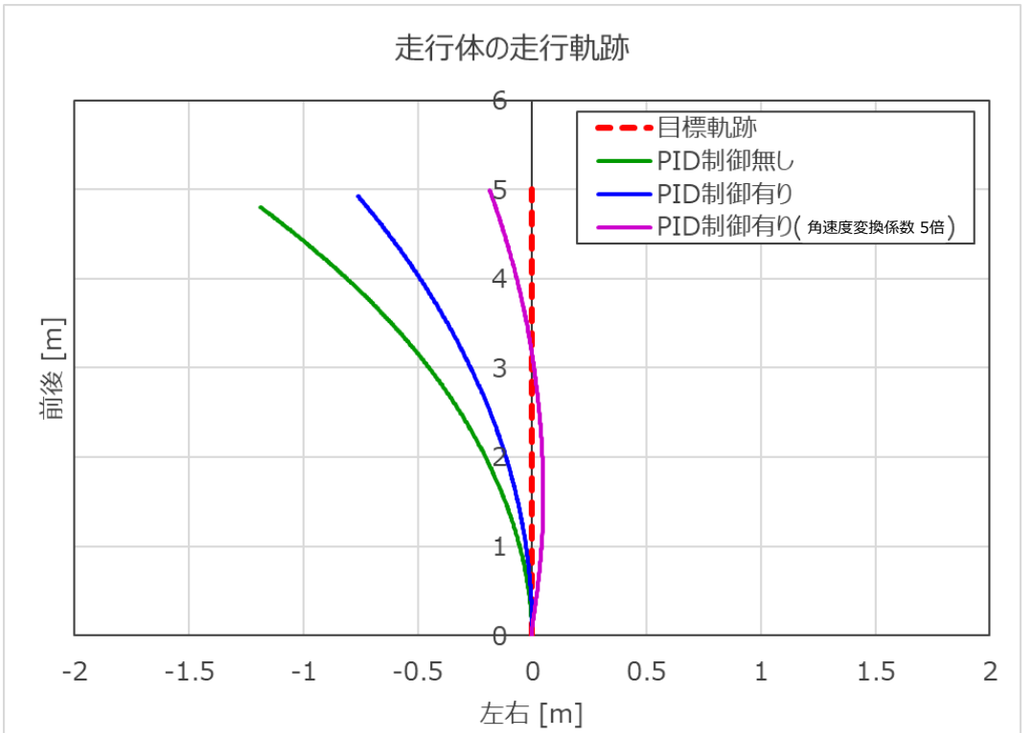
下記のPID制御を行い、左右モータ毎の伝達特性差の違いによる回転速度の差を補償する。



p: 出力値	K_p: 比例ゲイン	r: 補正量
α: 角速度変換係数	K_i: 積分ゲイン	p': 補正した出力値
e: 偏差	K_d: 微分ゲイン	ω: モータ角速度

【1-3】効果確認

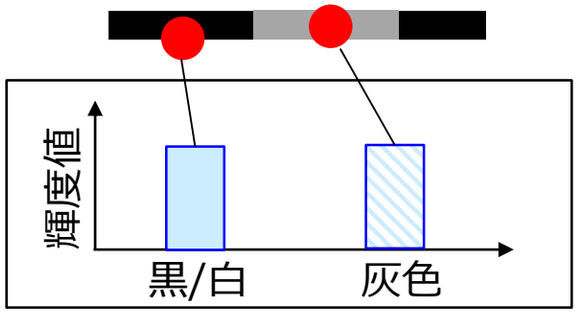
同じ出力値を与えてタイヤを空転させた際の左右モータの回転角度を計測し、数値計算により走行体の走行軌跡を算出した結果を下図に示す。
PID制御により、走行体の直進性を向上した。
また、角速度変換係数を増大させることにより、さらに直進性を向上できることを確認した。



2 灰色区間検知

【2-1】課題

ライン上の黒/白境界と灰色の輝度値が同程度であり黒白境界を灰色と誤検知してしまう。



【2-2】対策

PID制御を用いたライントレースの蛇行により、黒ライン走行時の輝度値の変化量は灰色ライン走行時の輝度値の変化量よりも大きくなる。そこで、過去の輝度値を記録し、その変化から灰色を検知する。

200msec間の輝度値の最大値と最小値を記録して、最大最小の差がある閾値以下になった時に灰色を検知する。