



1 要求分析



要求分類と要件図

ETロボコンにおいてもっとも要求されることは、**速く確実に走り、より良いリザルトタイムを出すこと**である。また、速く走るだけでなく**ソフトウェアの開発効率向上**も大きな要求と考えた。その二つの要求について分析し、まとめたものが下の要求図です。

速く確実に走りより良いリザルトタイムを出すことを基本機能とし、その実現のためには**コースの環境を把握すること**、その環境に応じて**走る**ことが必要と考えました。

このような色分けで要件図を表しました。

基本機能要件

速く走るために必要なこと

環境機能

コースの環境を把握するための機能

走行機能

走行するための機能

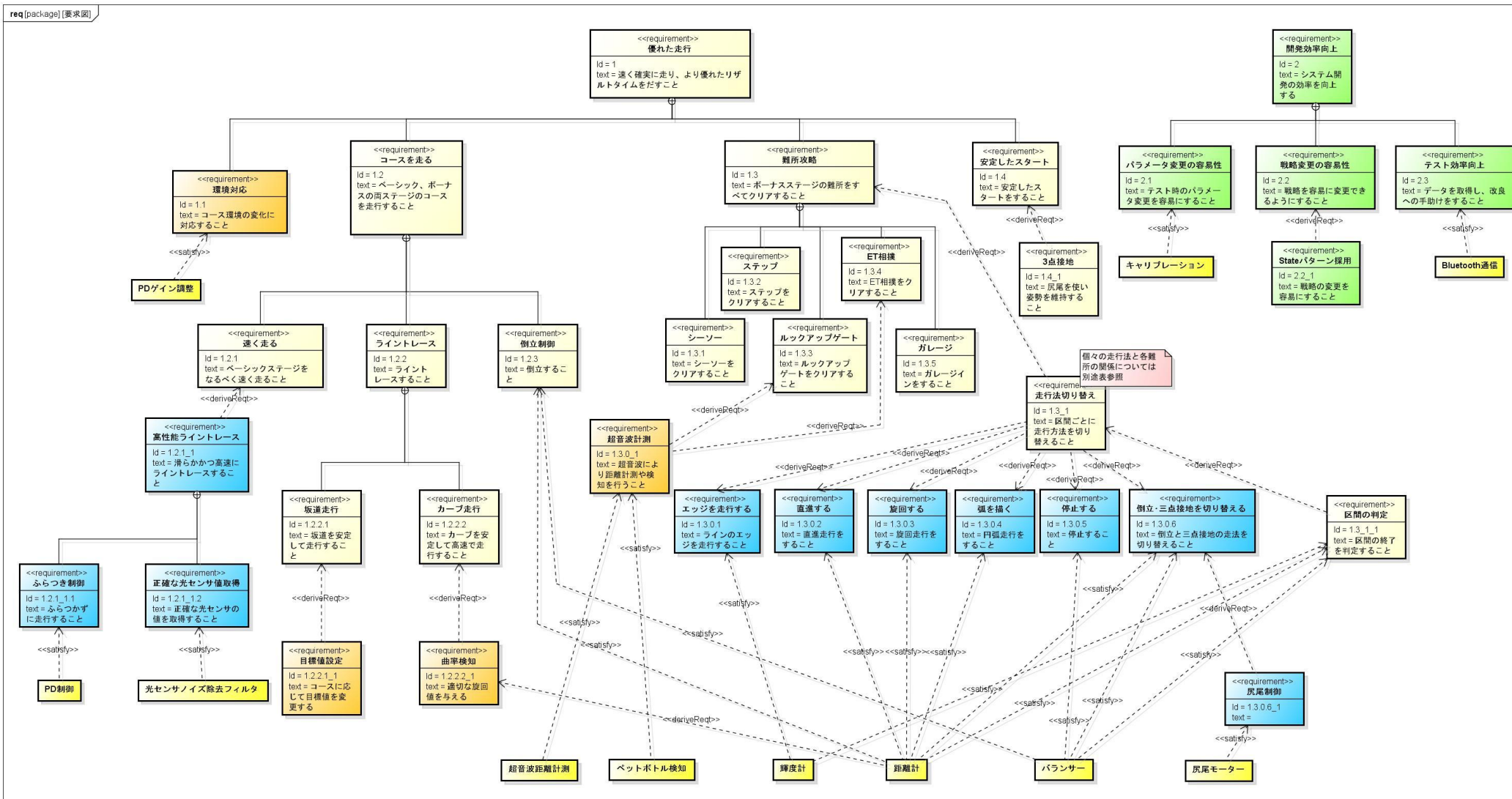
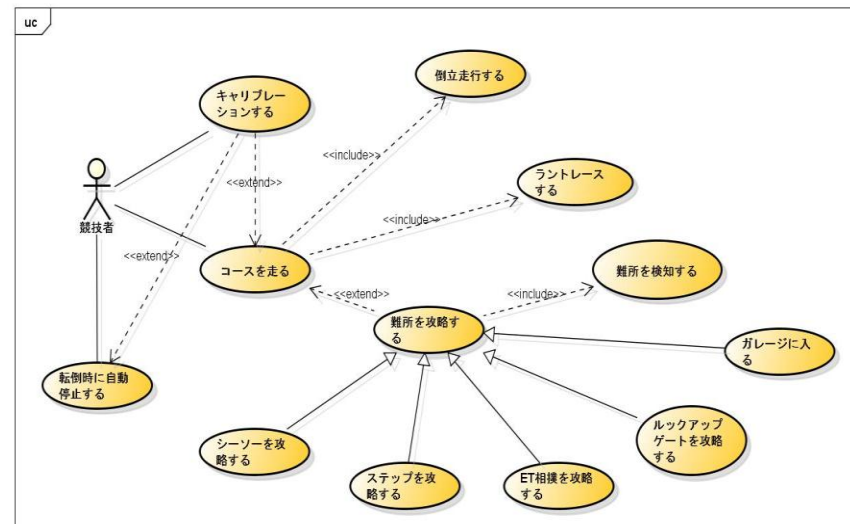
非機能要件

ソフトウェア開発効率向上に必要なこと

技術要素

機能の実現方法

与えるべき機能



難所と走行法の関係

左の要求図で記載することのできなかった、難所攻略と走行法は以下の表のとおりである。

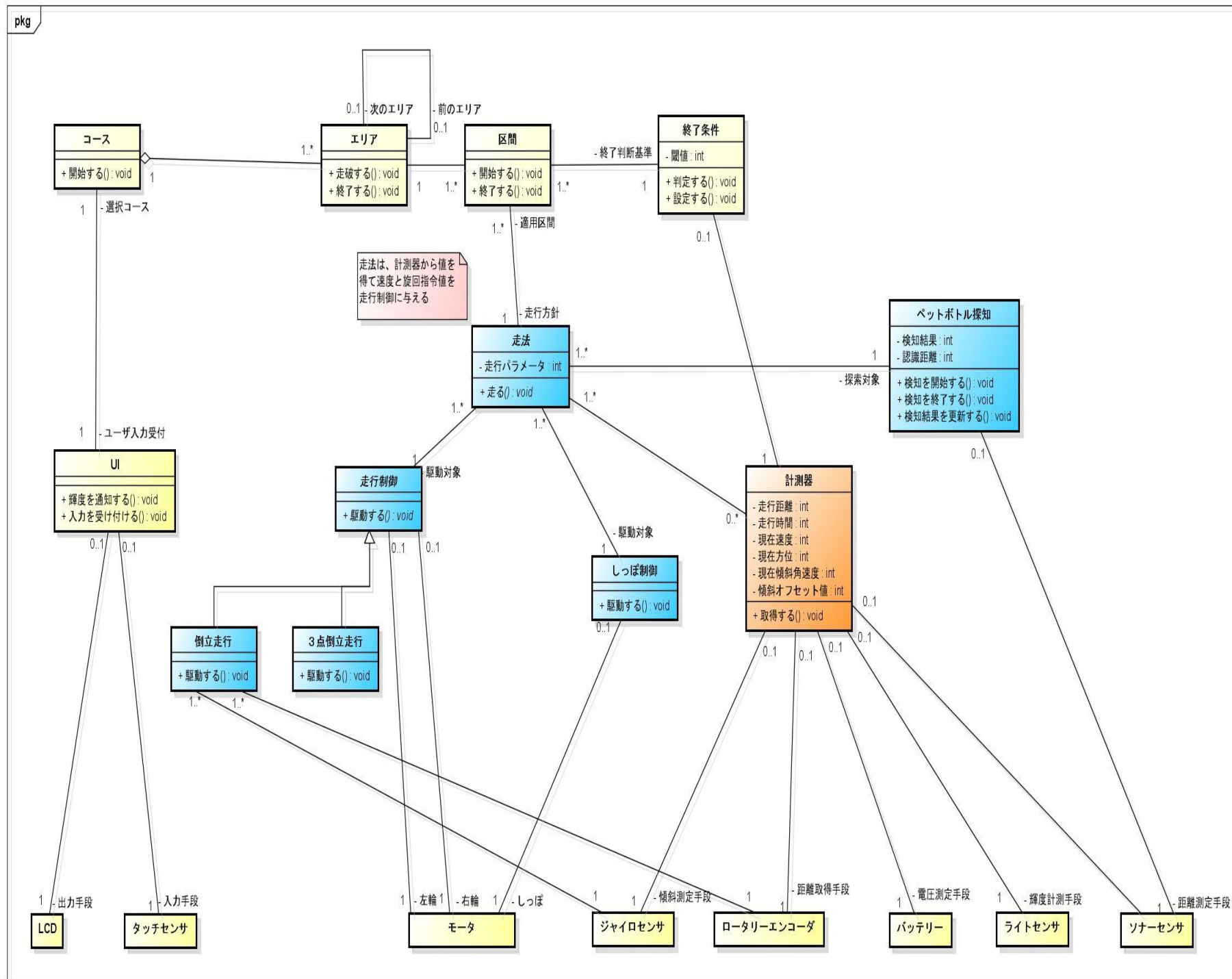
難所	必要な走行法
シーソー	直進、円弧、エッジ、停止
ステップ	直進、円弧、エッジ
ルックアップゲート	エッジ、倒立・3点切り替え、停止
ET相撲	直進、旋回、エッジ
ガレージ	エッジ、倒立・3点切り替え、停止



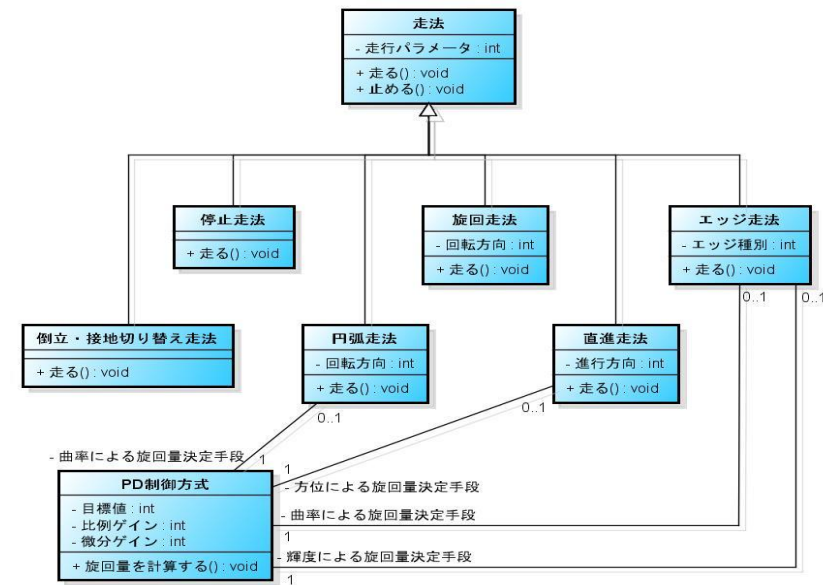
2 要求を実現した構造



全体の設計モデル

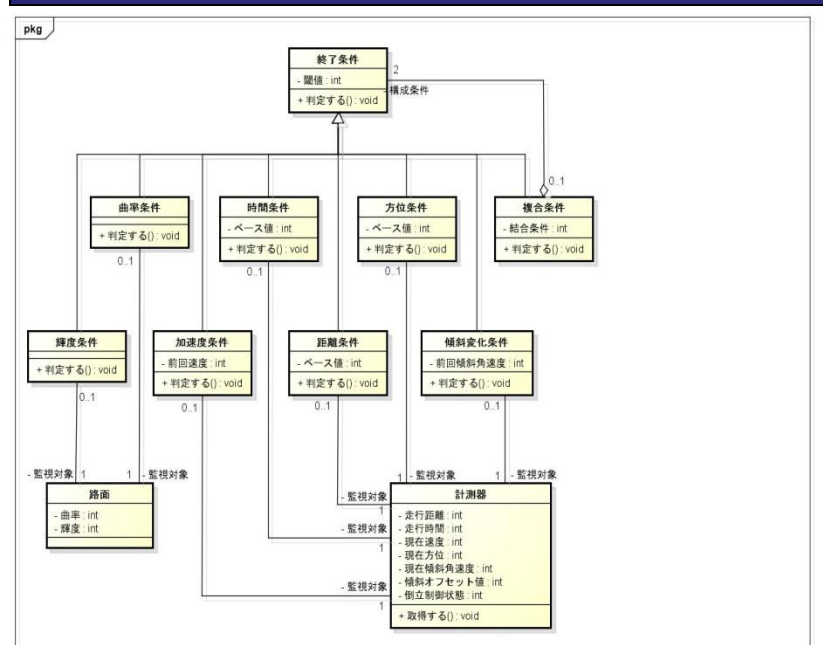


走法詳細



走法クラスには、倒立・接地切り替え走法、停止走法、円弧走法、旋回走法、直進走法、エッジ走法のクラスがある。円弧走法、直進走法、エッジ走法クラスはPD制御方式クラスを用いる。

終了条件詳細

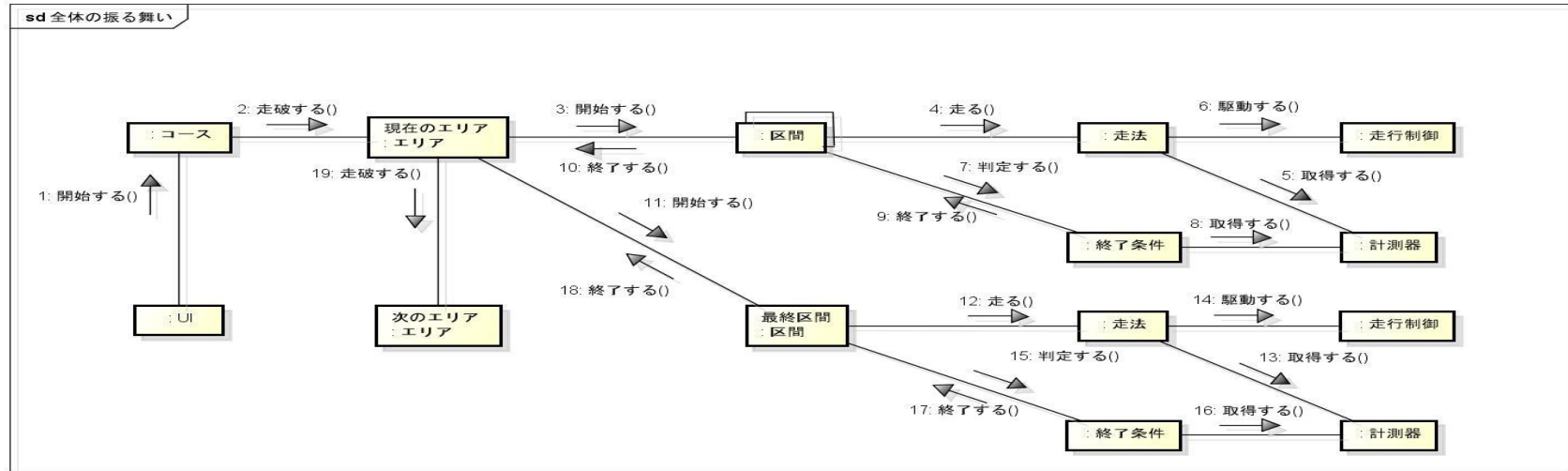


終了条件クラスは、輝度条件、曲率条件、加速度条件、時間条件、距離条件、方位条件、傾斜変化条件、複合条件のクラスから成る。曲率条件、輝度条件のクラスは路面クラスを用いて判定し、そのほかの条件クラスは、計測器クラスから状態を取得する。

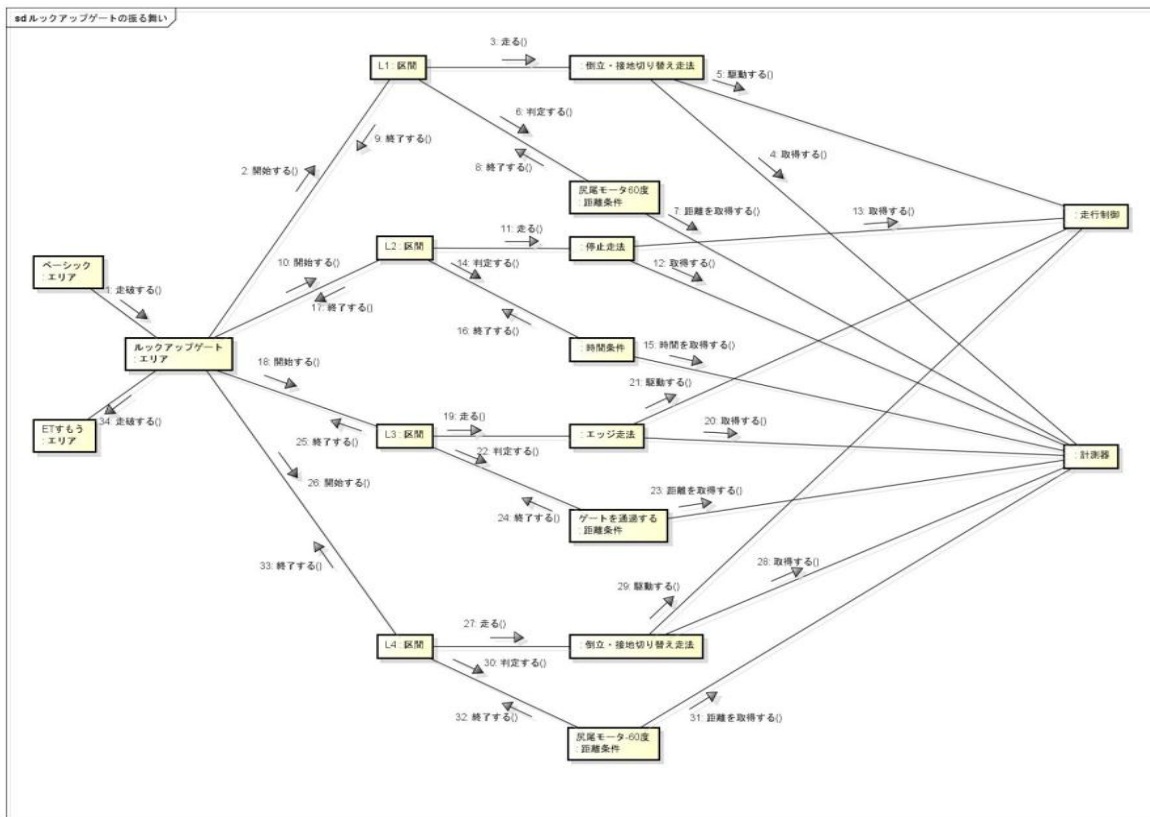
ベーシックステージ・ボーナスステージのそれぞれの難所毎に**エリア**を定義し、エリア内で走行方法の異なる走行部分を区間と定義する。各エリアはエリアクラスのインスタンスとして生成され、エリア内の区間の遷移を管理する。各区間は**区間**クラスのインスタンスとして生成され、区間毎の走法は抽象クラスである走法クラスの子クラスとして定義する。このような構造にすることにより、コース攻略戦術(エリア、区間の分割とその走法)の変更容易性を高めている。また、要求図と対応が取れるものと同じ配色をしている。

全体の振り舞い

ベーシックステージ・ボーナスステージの各難所を**エリア**で区切り、その**エリア**をいくつかの**区間**でわけける。その**区間**を指定した走法と終了条件によって順番にクリアしていく。一つの**区間**をクリアしたら次の区間を攻略し、全ての**区間**をクリアしたら次の**エリア**に切り替える。これを繰り返すことにより、コース全体を攻略する。



ルックアップゲート攻略法



ルックアップゲートは、L1からL4の4つの区間に区切られる。これらの4つの区間をクリアすることにより、ルックアップゲートエリアを攻略する。区間で走法と終了条件を決めている。

L1: 3点倒立走行をするために、倒立・接地走法を用いて尻尾モータを動かし、3点倒立状態にする。尻尾モータが指定量回転したら終了する。

L2: 停止状態でソナーセンサからの値を取得する。一定時間経過したら終了する。

L3: 3点倒立状態でエッジ走法を用いてルックアップゲートを通過する。一定距離走行したら終了する。

L4: 3点倒立状態から、倒立・接地走法を用いて倒立状態にする。尻尾モータが指定量回転したら終了する。

ET相撲攻略法

ET相撲は、S1からS7の7つの区間に区切られる。これらの7つの区間をクリアすることにより、ET相撲を攻略する。

S1: ルックアップゲート終了後、エッジ走法を用いて相撲エリアの真ん中まで移動する。指定した距離を走行したら終了する。

S2: 旋回走法により旋回しながら、ソナーセンサの値を取得することによりペットボルの位置を確認する。指定した角度を回転したら終了する。

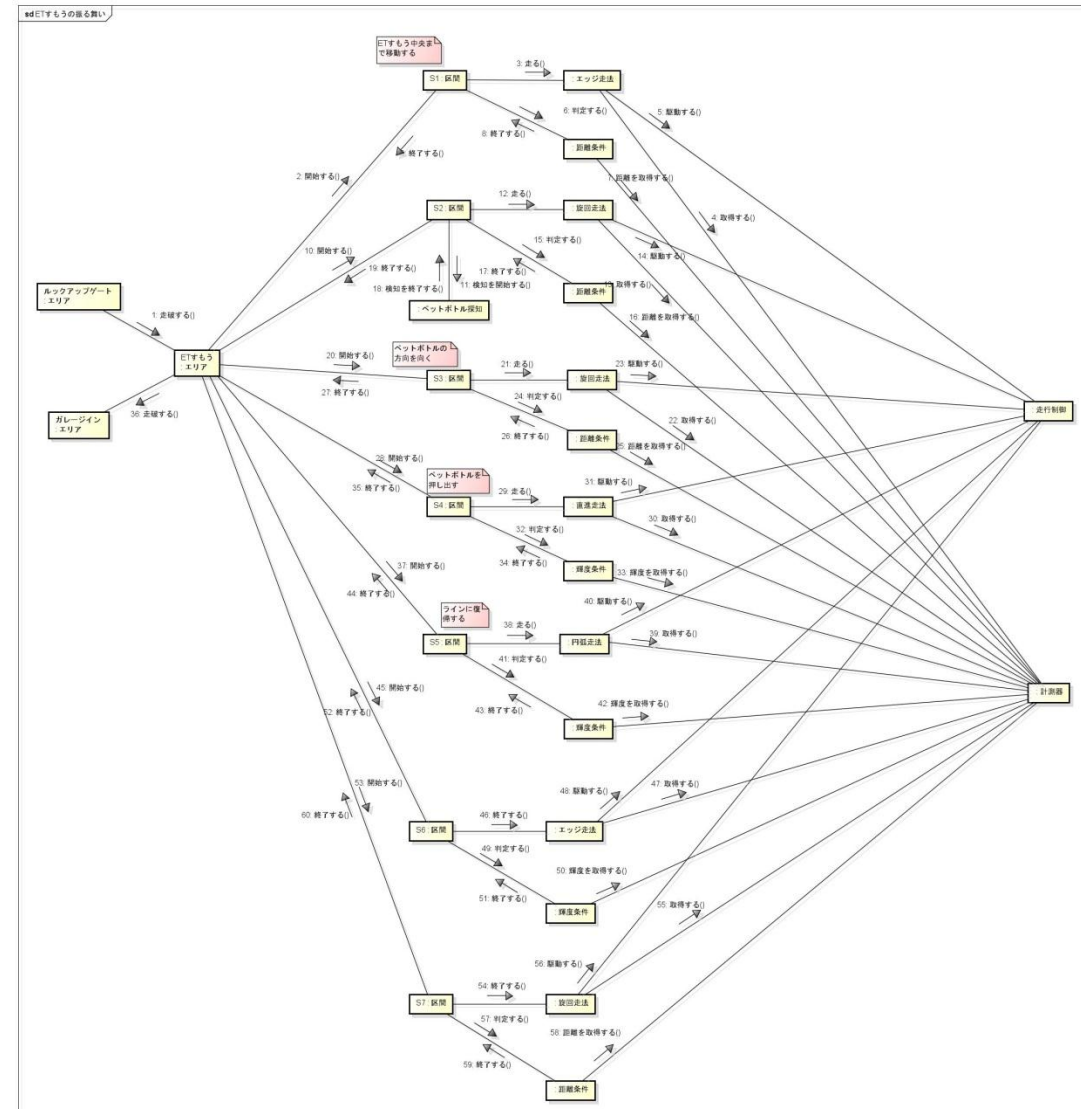
S3: 旋回走法により、ペットボルのある方向を向く。指定した角度を回転したら終了する。

S4: 直進走法により、ペットボルを押し出す。黒のラインを発見したら終了する。

S5: 円弧走法により、ラインレースに復帰する。黒のラインを発見したら終了する。

S6: エッジ走法により、ラインレース走行をする。又(黒ライン)を発見したら終了する。

S7: 旋回走法により、旋回してガレージインの準備をする。指定角度回転したら終了する。



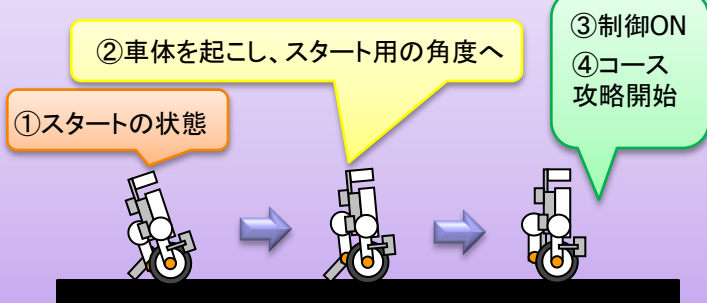
4 難所攻略方法



滑らかなスタートを実現

スタートの手順

- ①尻尾を使って立っている状態。
- ②スタート用の角度まで、尻尾を起こす。
- ③倒立制御、ライトレースをONにする。
- ④走行開始！



ふらつかず、安定したスタートを実現！

ルックアップゲート攻略

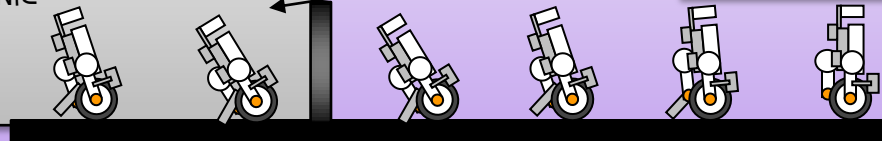
要件ID: 1.3.3
ルックアップゲート

攻略の手順

- ①ゲート直前までライトレースで走行する。
- ②尻尾をおろし、倒立制御を切り、尻尾で立つ。
- ③ゲート通過用の角度に尻尾を倒し、姿勢を変える。
- ④直進し、ゲートを通過する。
- ⑤スタート用の角度まで尻尾を起こす。
- ⑥倒立制御、ライトレースをONにする。
- ⑦ゲートクリア！

安定したゲートの攻略を実現！

OPOINT
ゲート通過用に尻尾の角度を調整！

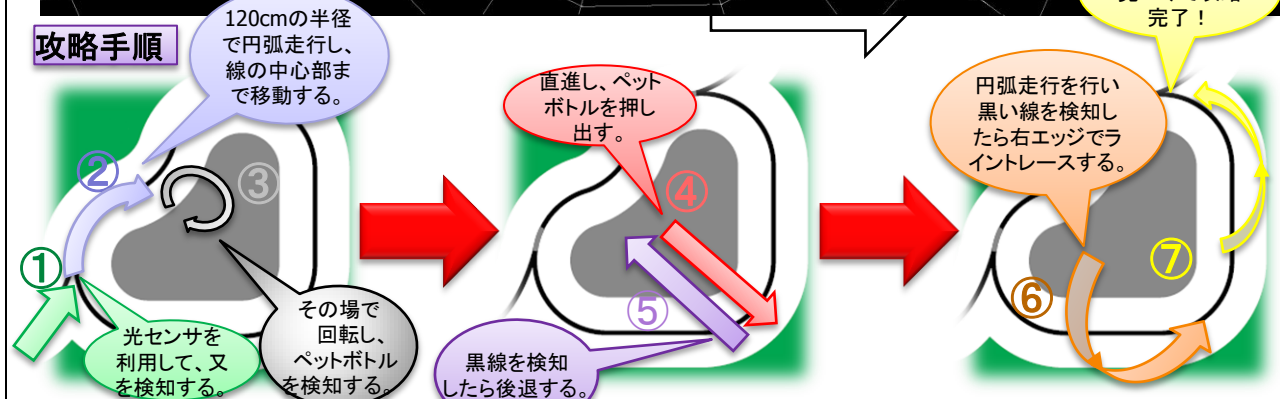


ET相撲攻略

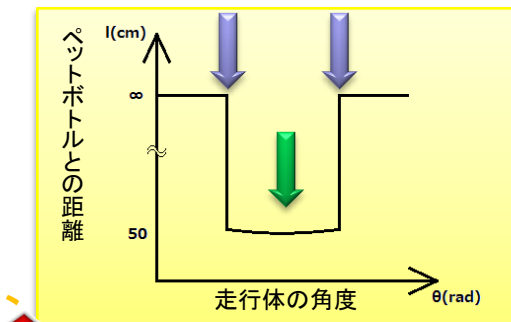
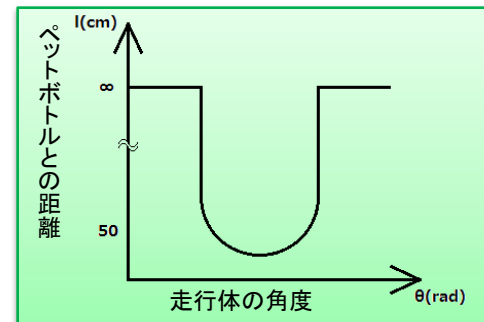
要件ID: 1.3.4
ET相撲

超音波の世界へようこそ

攻略手順



超音波センサ

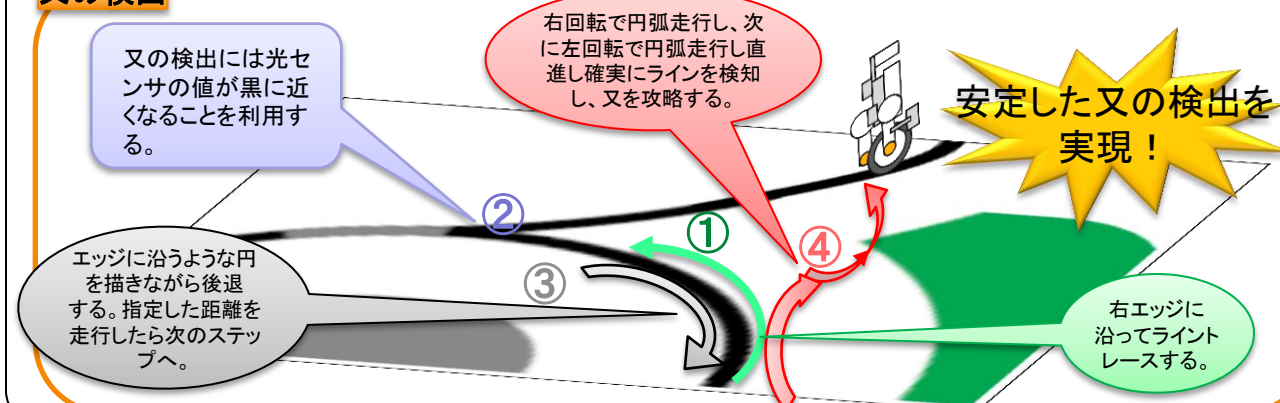


しかし！
実際は、

丸いペットボトルの距離を測定するため、理想的なグラフのようになると、値の最小値を取ればペットボトルに正対できるはずである。

実際は、左図のように矩形になってしまう。そのためペットボトルのエッジを検出しその中心がペットボトルと正対する場所だと判定する。

又の検出



安定した又の検出を実現！

又の検出には光センサの値が黒に近くなることを利用する。

右回転で円弧走行し、次に左回転で円弧走行し直進し確実にラインを検出し、又を攻略する。

エッジに沿うような円を描きながら後退する。指定した距離を走行したら次のステップへ。

右エッジに沿ってライトレースする。



5 技術要素



カーブ走行

要件ID: 1.2.2.2
カーブ走行

カーブを走行するときに、内輪の外輪の速度差を考慮して旋回値を与えることで、スムーズにカーブを走行することができる。

周回速度: V
 車輪間の距離: $2r$
 カーブ半径: R
 とすると...
 内輪の速度 $\Rightarrow V - \frac{r}{R}V$
 外輪の速度 $\Rightarrow V + \frac{r}{R}V$

直進・円弧走行

要件ID: 1.3.0.2, 1.3.0.4
直進・円弧走行

昨年度まで使用していた仮想ライトレース代わりに、より実装が簡単な方向を目標とする制御により、直進と円弧走法を実現する。目標方位からのずれに対して、PIDフィードバック制御で旋回速度値を与える。

直進走行
常に目標方位を0として、走行体の向きがそこからずれたら修正を行うようなフィードバックをかける。

円弧走行
走行距離に対応した方位目標を定めることにより、ずれとの修正が可能になる。

坂道のはしりかた

要件ID: 1.2.2.1
坂道走行

① 垂直に倒立して走るNXT走行体。坂道では光センサの対地高度が低くなる。

② 平坦な道と坂道では、黒白の値が異なってしまう。そのため、坂道走行時は、ライトレースの目標値を、坂道の上下りそれぞれ黒白の中央値に変更する。これにより、安定した坂道走行を実現する。

センサがいつもより明るめの値を示す。

PD制御をなめらかで高速に

要件ID: 1.2.1.1
ふらつき抑制

これがPD制御

ずれ = トレース目標色 - 現在の光センサ値
 ずれの微分 = 今回のずれ - 前回のずれ
 旋回指令値 = Pゲイン * ずれ + Dゲイン * ずれの微分

光センサのノイズ除去

要件ID: 1.2.1.2
正確な光センサ値取得

目的
高速走行するため、PDゲインを大きくしたいが... 光センサに乗っているノイズのせいで微分値が暴れてしまう！そこで、フィルタをかけてノイズを除去することにした！

使うのはローパスフィルタ
今回は小さな計算負荷で高次特性を得られるIIRデジタルフィルタを採用。

出力 = 今回の入力 * (1-A) + 前回の出力 * A

光センサ値とその微分値
ノイズが乗った光センサ値
ノイズの影響で暴れる微分値

フィルタ適用後
なめらかになったセンサ値
必要な微分成分のみが残った！

ローパスフィルタでノイズ除去

Dゲインを大きくすると旋回指令値が暴れてしまう！

Dゲインを大きくしても必要な方向に出力が得られる！

大きなゲインで早めの制御！振れが小さく、なめらかに！

PDゲインの自動最適化

要件ID: 1.1
環境適応

環境に応じて自動ゲイン決定

光センサのキャリブレーション結果から、最適なゲインを自動でスケールリング。環境の変化に強く、コースアウトしにくい走行が可能に！

テスト効率の向上

要件ID: 2.1
パラメータ変更の容易性

NXT本体でパラメータを調整可能に

プログラム開始後に、NXT本体のボタンを使ってPDゲインや速度などを調整できるようにした。これによりテスト効率が大幅にUP！

減らす
決定
増やす

走行位置シフト

要件ID: 1.2.2.1
目標値の変更

エッジチェンジや坂道走行、マーカー走行の際に、ライトレースの目標値をシフトさせ、ライン内を走り分ける。

この機能により、コースアウトの防止や、より正確なラインの乗り越え検知を行うことができるようになった。

なめらかな値変更

要件ID: 1.2.1.1
高性能ライトレース

目標値を変更するときには、値を徐々に変化させることで安定した走行を可能としている。

この仕組みは、加速・減速や走行位置シフトで使われている。

例) なめらかな減速