自律走行・演技のために
やったこと。学んだこと。

荒川祐真

2014年6月14日(土)

目次

- チーム・パフォーマンス紹介
- チームについて
- パフォーマンス内容
- スケジュール

- 工夫点・苦労点・反省点

- まとめ

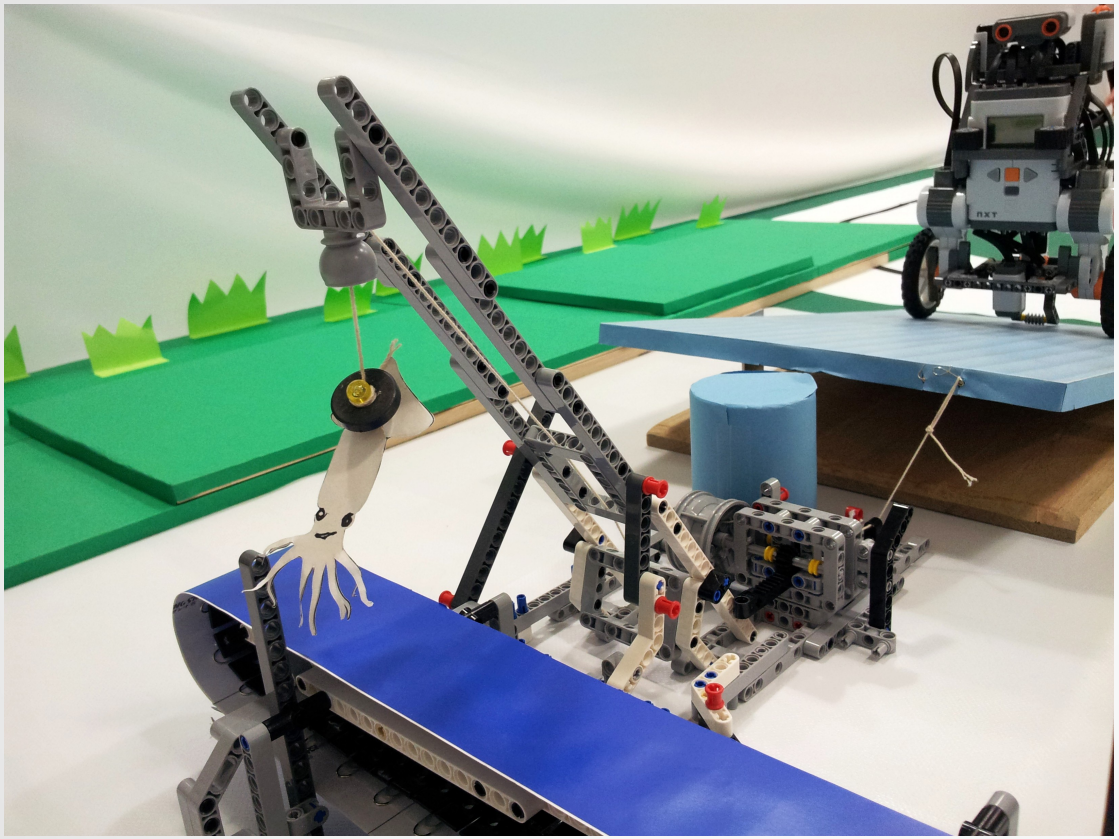
チーム・パフォーマンス紹介

- チームについて
- パフォーマンス内容
- スケジュール

チームについて

参加チーム名	イカポックル
参加部門	アーキテクト
参加資格	大学
参加地区	北海道
所属	公立はこだて未来大学
地域	北海道函館市
人数	4
プラットフォーム	nxtOSEK
開発言語	C++

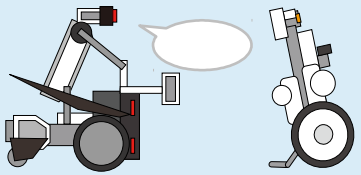
パフォーマンス内容



テーマ

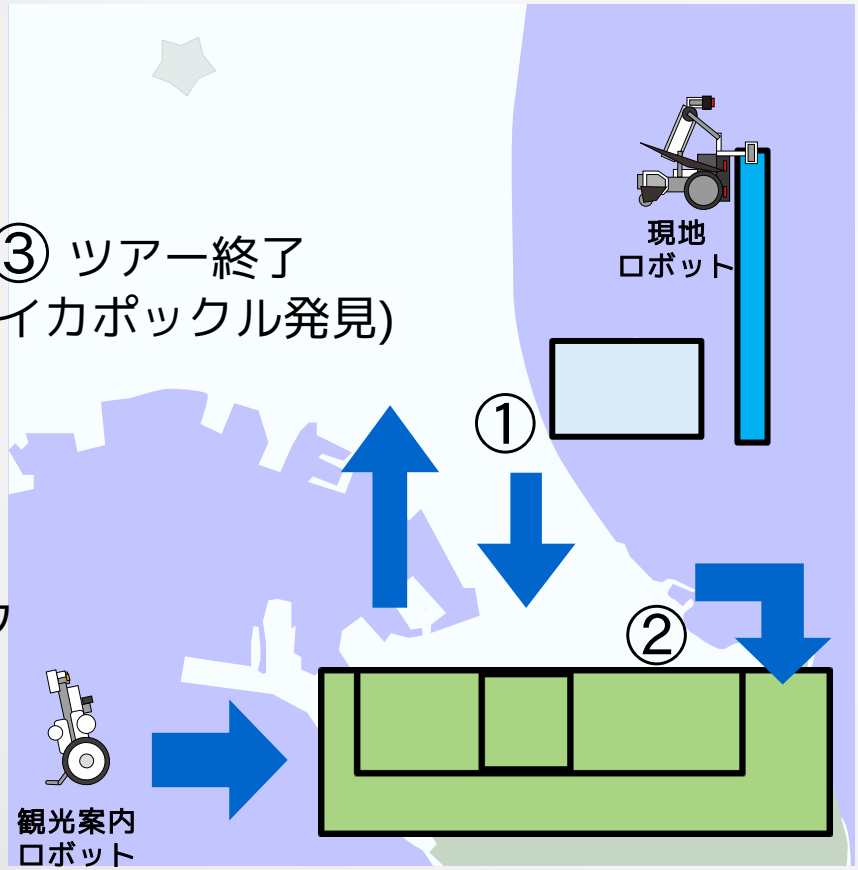
未来の
函館観光案内

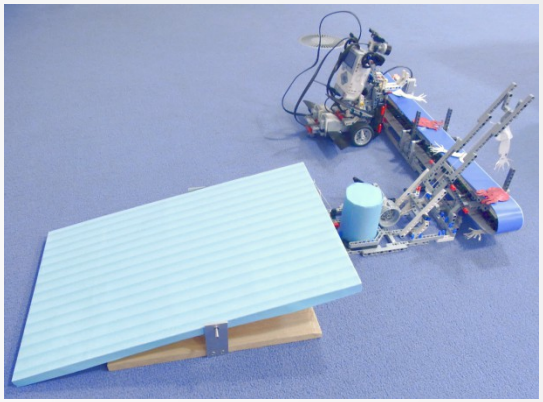
未来の函館
ロボットを用いた
観光案内



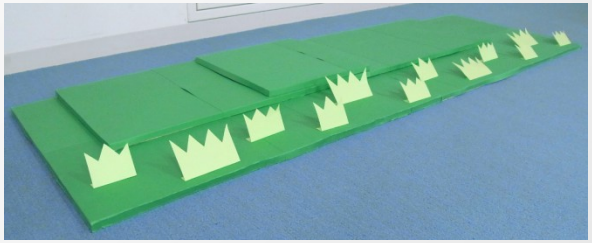
パフォーマンス内容

ツアー内容　函館観光で「イカポックル」を見つけよう
※　「イカポックル探し」はTwitterタイムライン上のストーリー





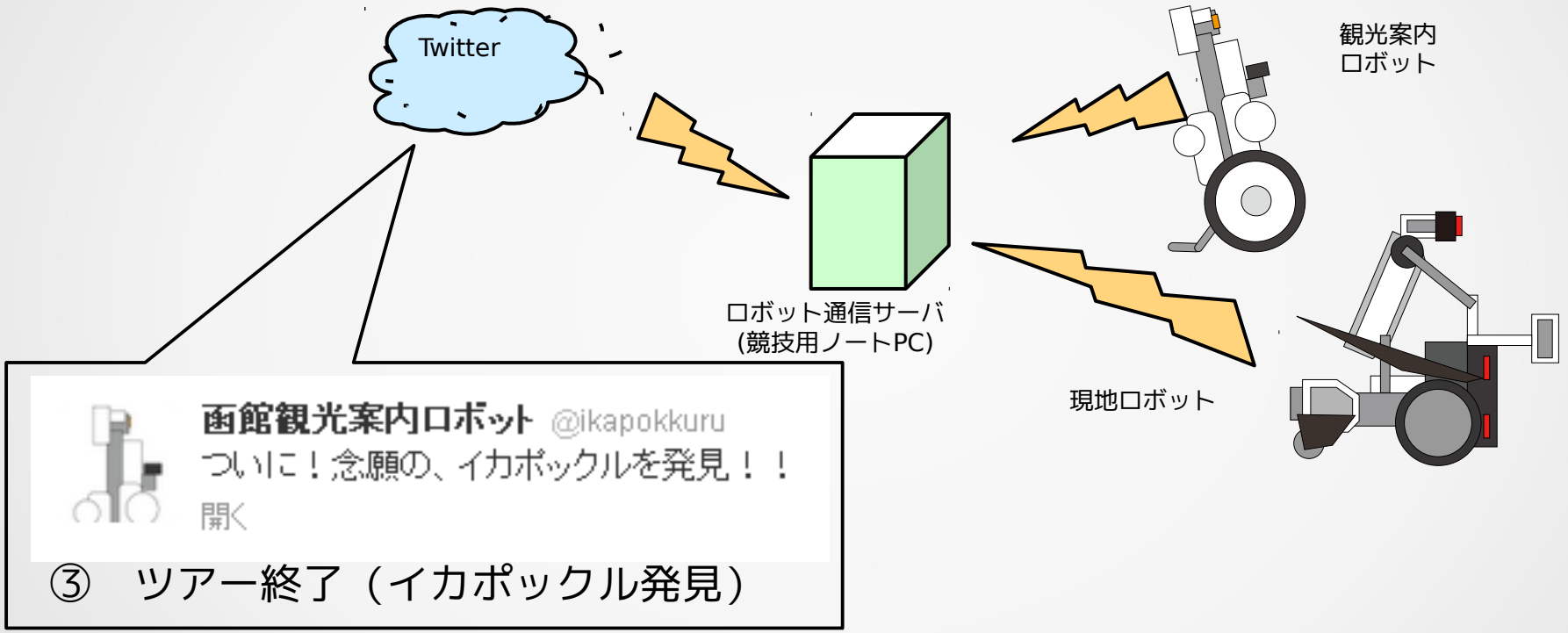
①　イカ釣り体験



②　函館山

パフォーマンス内容

パフォーマンス各所でTweet送信



「イカポックル探し」の演出（ゴール地点）

➡ 発展：ツアーを終えるとオリジナルグッズ（待ち受け、着うた等）DL
アルバム写真DL（ロボットが各所で撮影してくれる）

スケジュール

4〜5月	6〜7月	8〜10月
企画立案 ・企画内容相談 ・ガジェット素材選定 ・センサ特性調査 ・部品化を意識した テストコード作成	走行プログラム 第１弾作成 ・オープンキャンパス (8/4)での走行を目処 ・ライントレースは ほぼ完成 ・ガジェットの素材集め ・ガジェット作成	パフォーマンス作成 ・ガジェットの素材集め ・ガジェット作成 ・パフォーマンス プログラム作成 ・コンセプトシート・ 企画書作成
作業時間 1.5時間を週1回	1.5時間を週2回	3-5時間を週3回

工夫点・苦労点・反省点

- チーム運営
- 企画立案
- 設計
- ベーシック走行
- 要素技術
- パフォーマンス
- プレゼンテーション

チーム運営（工夫点）

- Gitでバージョン管理
  - マージ担当を決めて、最新版を管理
  - ただし、使用を徹底できなかった
- Google Driveでドキュメントやファイルを共有
  - 特にCalcはアイディア出しで使用
    - 作業項目リスト
      - 大分類、中分類、小分類、作業内容、詳細、担当、優先度、状況、期限の表で管理

企画立案（苦労点・反省点）

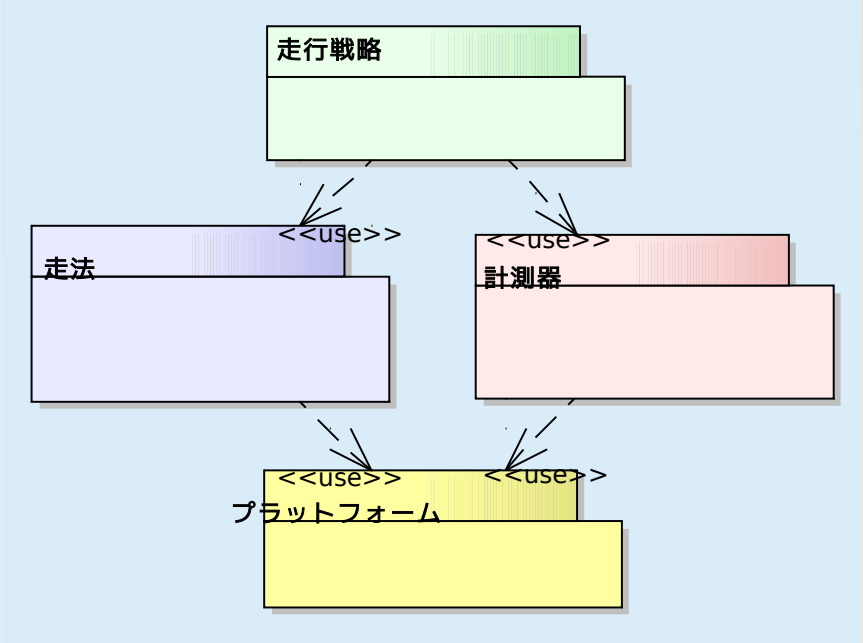
- 光センサはあまり使いたくなかった
  - 開催場所が太陽光の差し込む場所（はこだて未来大学のガラス張り箇所）
  - モーターのロータリエンコーダを使うことを検討
    - 「パフォーマンスステージいっぱいに布を敷く」は検討から外していた
      - シワやたわみがロータリエンコーダを狂わす
      - しっぽ走行によりシワを生み出す恐れ
      - 布を固定しておく方法が思いつかなかった
        - 「四隅をピン留め」はステージ破損により不可

企画立案（苦労点・反省点）

- パフォーマンスステージの破損に注意
  - 函館山は、設置の過程で「手を滑らせて端から落下」ということが無いように
- 人手が足らず、大規模な仕掛けを断念
- 「装飾」にあまり力を入れることができなかった
  - 函館をイメージしづらい結果に
- SFに走り過ぎてしまった
  - 短い時間で、想定している未来社会の背景まで理解してもらうのは困難

設計（工夫点）

- 階層を意識した設計
  - 各階層毎の役割を意識した
- クラス図には早期に取り掛かり、上位層と下位層に分かれて開発



ベーシック走行（工夫点）

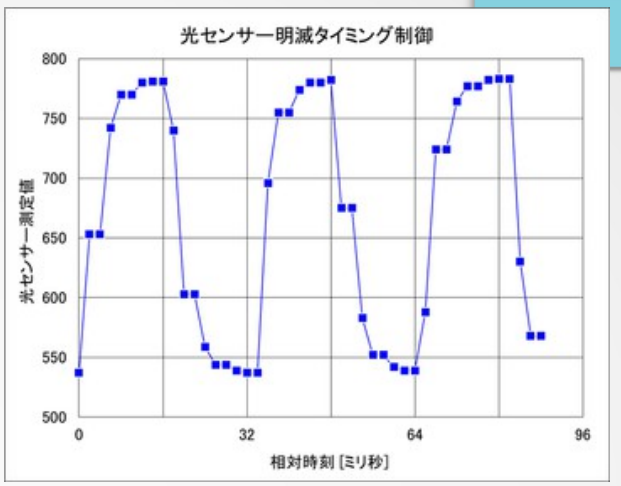
- 光センサに頼らない設計
  - GPS、区間ごとの曲率
  - ハイブリッド走行

- 初期段階で距離計、速度計、曲率計などの要素技術を用意

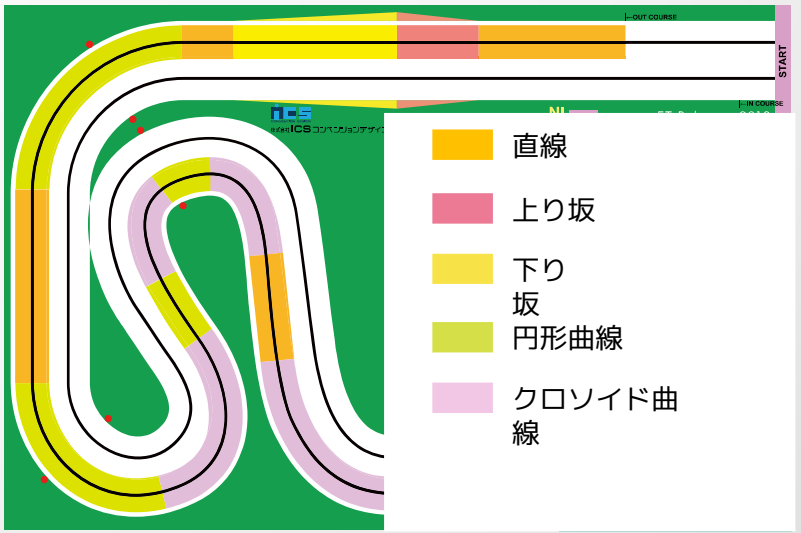
ベーシック走行（工夫点）

- まいまい式での輝度値更新周期は最短でも28ms
  - 高速走行は困難
- コースを区間で分け、区間ごとの曲率によるナビゲーショントレースを実装
  - 通常のライントレースとナビベーションとレースの割合により旋回量を求める

旋回量　＝
　ライントレース(輝度値)による旋回量
　　×ライントレース参照割合
　＋　ナビゲーショントレース(曲率)による旋回量
　　×　(1　－　ライントレース参照割合)



まいまい式改良の狙い目 | すねいる　ＥＴロボコン日記
http://www.chihayafuru.jp/etrobo/?p=1992



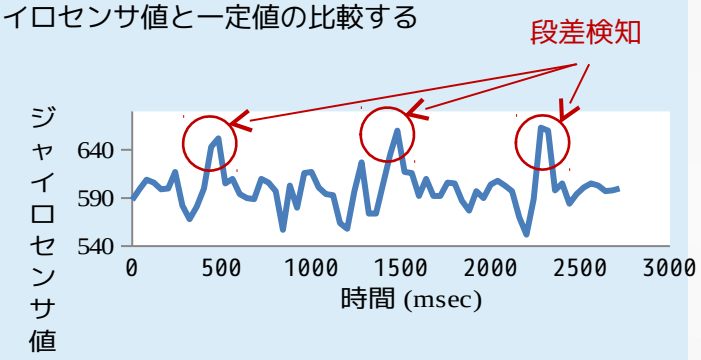
要素技術（工夫点）

段差検知

ジャイロセンサ値の変化とモータ速度の変化を利用して段差を検知する

ジャイロセンサによる段差検知

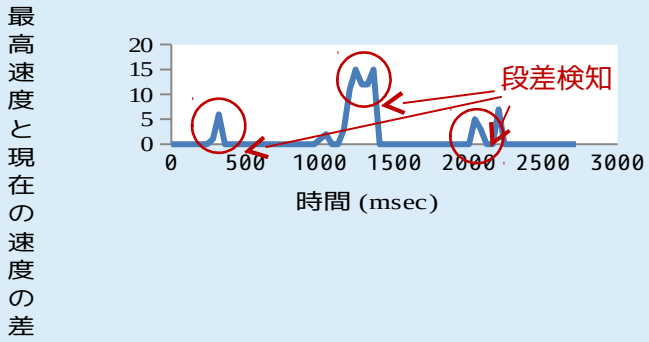
ジャイロセンサ値と一定値の比較する



強み：正面から段差に衝突した時の検知精度が高い
リスク：段差に斜めから進入した際に検知できない

モータ速度による段差検知

最高速度と現在の速度の差と一定値を比較する



強み：段差への進入角度によらず検知できる
リスク：検知が遅く、段差を上る可能性がある

対策：2つの検知を組み合わせることで、互いのリスクを互いの利点で補う→高精度な段差検知

過去の活用できる資産は、有効に活用

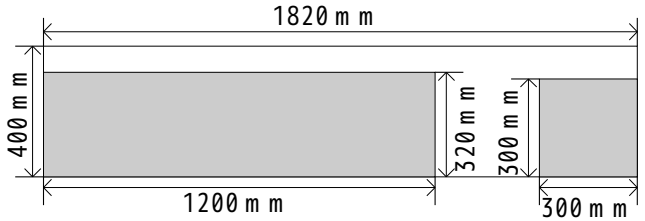
パフォーマンス（工夫点）

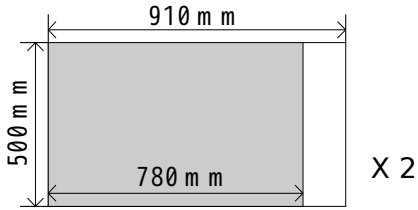
函館山について

函館山は、牛が寝そべっている形状
（別名：臥牛山[がぎゅうざん]）

・パフォーマンスステージで再現可能な
縮尺を求め、3枚の板で再現

・リスク：
必然的に最下層の板は一番大きくなる
重さによりシステム設置時に角から
落下の恐れ

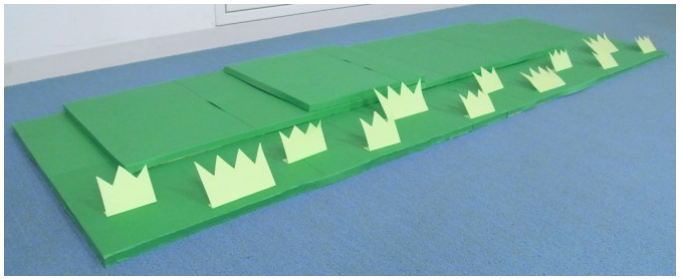




X２

・軽く加工のし易い「桐集成材」を使用
最下層の板は半分に切断

・比重は0.29で、最下層の2枚はそれぞれ
1.5kgほど多少不安定な姿勢であっても、
持つことに支障の無い重さであることを
確認した。



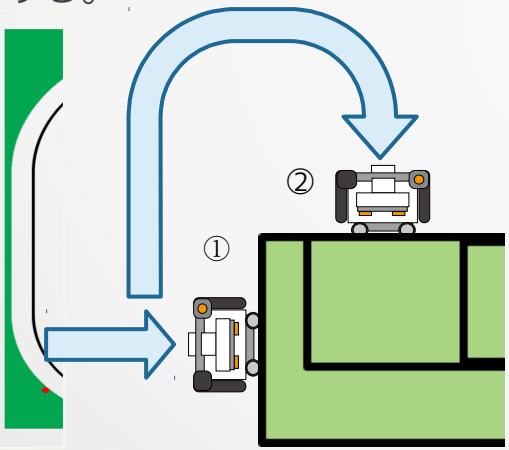
リアリティをこだわる

パフォーマンス（工夫点）

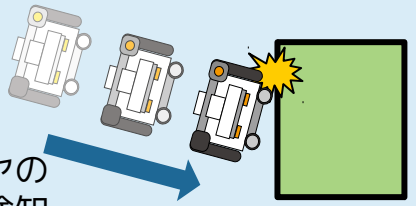
オープニング　（金森倉庫の観光）

パフォーマンス・ステージ侵入後の位置補正のための動作。

①, ②の段差へX,Y方向からそれぞれ衝突することにより、自己位置と向きを補正する。



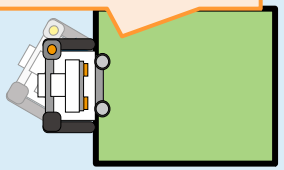
向き修正の振る舞い



(1) 段差に対し斜めに侵入した場合、段差に衝突したタイヤの回転速度が落ちる。これを検知し、現在の向きを判断する。

(2) 衝突していない側のモータのみを回転させ、段差に対して垂直に衝突する状態にする。

衝突による転倒を避けるため、低速で行う。



(3) 予め指定された方位と自己位置のパラメータで初期化する。

向き修正完了！

➡ 距離の初期化

パフォーマンス（工夫点）

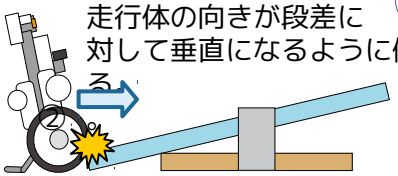
釣り場　（パフォーマンスポイント１）

達成条件

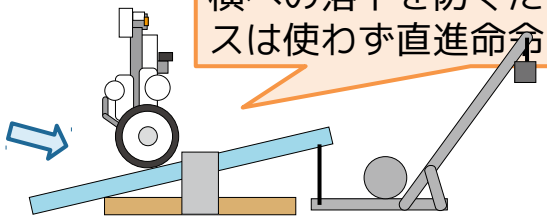
- 観光案内ロボットがシーソー上で前後し、転倒せずに釣りを行う。
- 「イカ」を釣り上げる。

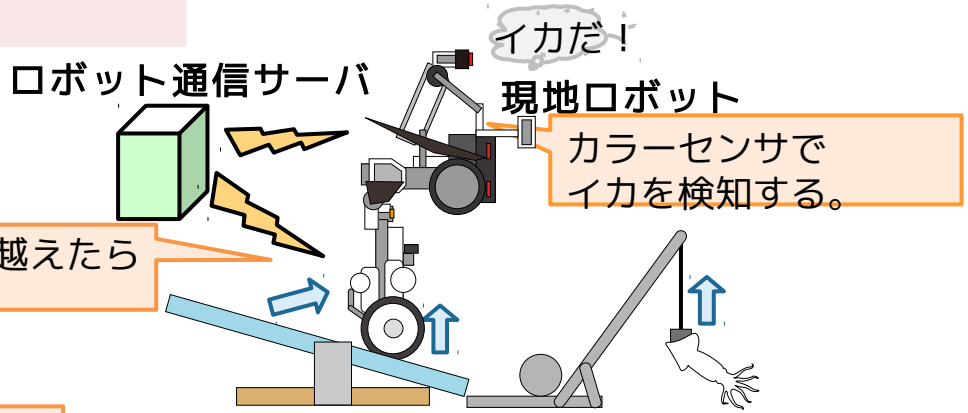
「NXT-EV3ネットワーク機能拡張ソフト」を用いて、
NXTはベルトコンベア上の魚を検知する。

① タイヤを段差に衝突させ走行体の向きが段差に対して垂直になるように修正する。

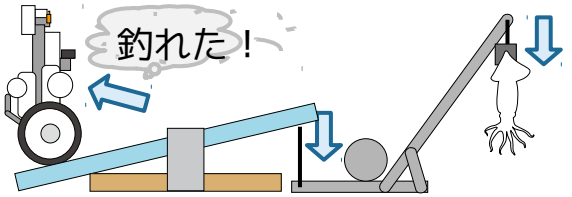


② 倒立振子走行で段差に上る
③ 現地ロボットがカラーセンサでイカを検知するまで待機する。





④ 現地ロボットがイカを検知したらシーソー上を前進し傾けることで、磁石をベルトコンベア上のイカに取り付ける。
⑤ シーソー上を後退し、イカを釣り上げる。



ロボット通信サーバの画面。現地ロボットのイカ検出通知を観光案内ロボットに伝える。

パフォーマンス（工夫点）

- EV3の色センサ
  - 対象物との距離が10cm程度を超えると誤検出
- EV3側で検知してからNXTがシーソーを始めるタイミングは時間で制御
  - ベルトコンベアの速度を一定に保つことが重要
  - 使用出来る電池残量の区間を決めた
  - 電池残量に応じた処理をしても良かった

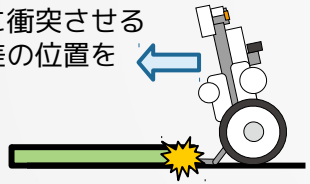
パフォーマンス（工夫点）

函館山　（パフォーマンスポイント２）

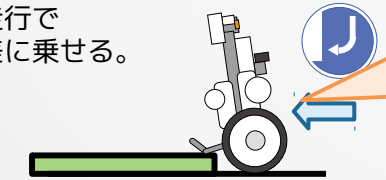
達成条件
・図示する規定方向で山を登り、　下山する。
・最後の2段分の落下では、転倒せずに尻尾走行に移行する。

① タイヤを段差に衝突させ、走行体の向きが段差に対して垂直になるように修正する。

② 尻尾を段差に衝突させることで、段差の位置を把握する。

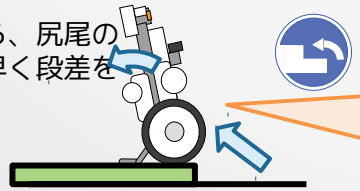


③ 倒立振子走行で尻尾を段差に乗せる。



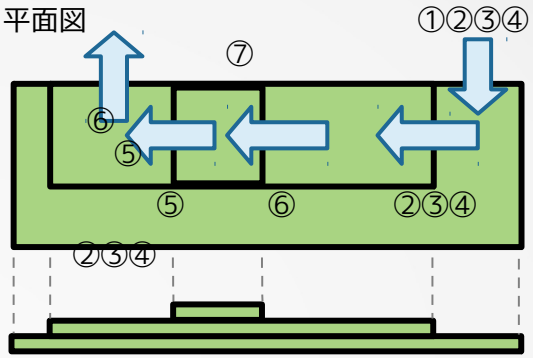
前に倒れないよう姿勢を後ろに傾けておく

④ 後退しながら、尻尾の力を使い素早く段差を上る。



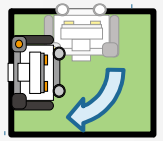
尻尾を常に接地させて上ることで振動を抑える

平面図

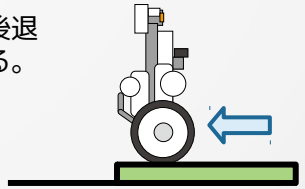


正面図

⑤ 右向きに90度旋回する。



⑥ 倒立振子走行で後退し、段差を降りる。



⑦ 尻尾を下ろし停止する。

➡ 尻尾の制御タイミングに注意

パフォーマンス（工夫点）

- 見えない部分を見える化することを考える
- 尻尾の制御タイミング
  - 目で追い切れない

  - デジカメの連射機能を活用

パフォーマンス（工夫点）

- Twitterのツイート数制限
  - ホームTL・リプライの取得上限回数：15回／15分
  - 通常は開発用アカウント、本番に本番用アカウントへ切り替え

プレゼンテーション（苦労点・反省点）

- プレゼンの練習をしていなかった
  - ほとんどしていなかった
  - 練習時間を確保し、開発作業とは分けるべき

まとめ

まとめ

- 昨年、「イカポックル」というチームで参加した経験をお話しました

- 「時間をかけるべき作業」と「時間をかけるべきでない作業」を意識すべき
  - 「効率化」や「知識を深める」点には手を抜かず
  - 「単純作業」や「必要のない開発」はしないように