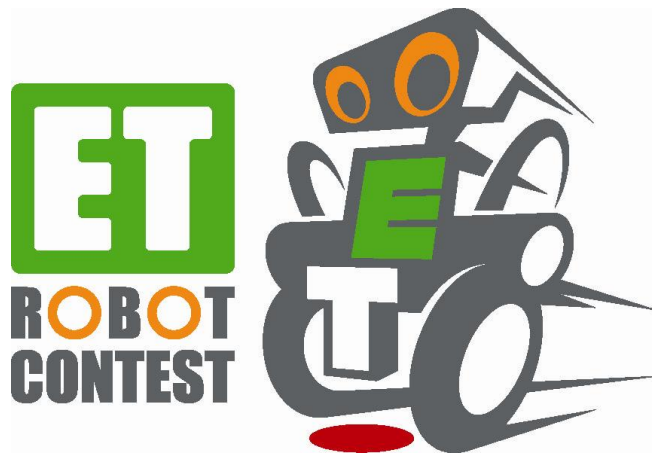


2011年モデル審査基準 補足資料

2011年5月9日
本部審査委員会

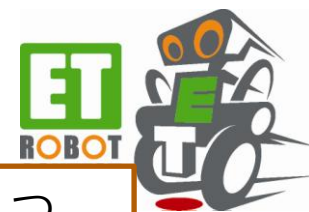


本資料の目的



- 本資料は、ETロボコン2011大会モデル審査基準の理解向上を目的に作成された補足資料です
- 審査基準の各項目ごとに、審査基準を満たしており、サンプルとして適切と思われるモデルを選び、審査基準に該当する部分について解説を行っています
 - モデルは、2010年チャンピオンシップ大会のモデルを使っています
- なお、本資料で紹介しているモデルは、審査基準を満たしている一例として取り上げており、このように記述することを強制するものではありませんので、ご注意ください

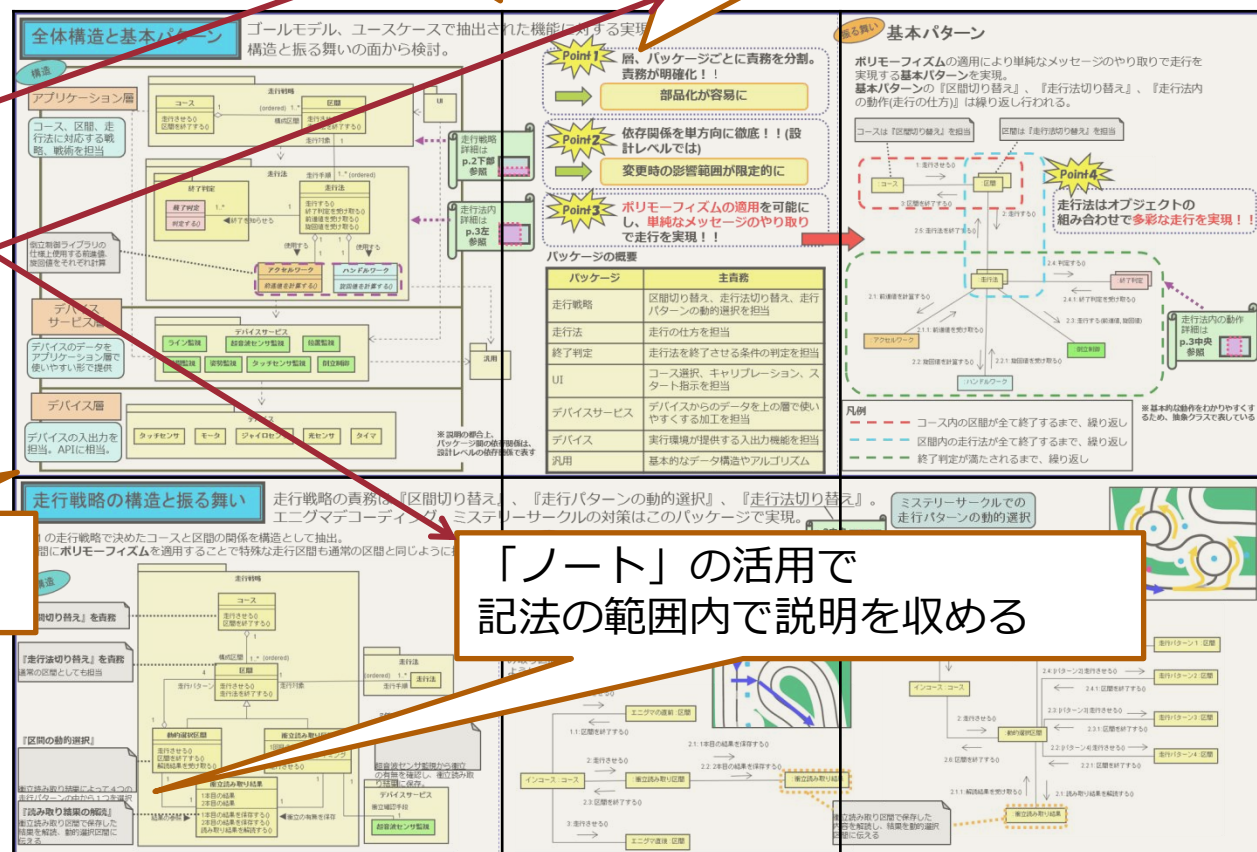
図の見やすさ・図の補足 (1)



- レイアウト・要素数は適切か
- 色分け・仕切り等による可読性の向上
- 印刷された段階で適切な大きさになっている
- 過度に書き込まれていない

縦2横3のグリッド配置、各グリッドに話題1つ
(この図は、説明のためグリッドを重ね表示してある)

ポイントにまとめ、説明をコンパクトに



全体構造と走行戦略で枠をつけて上下に2分割

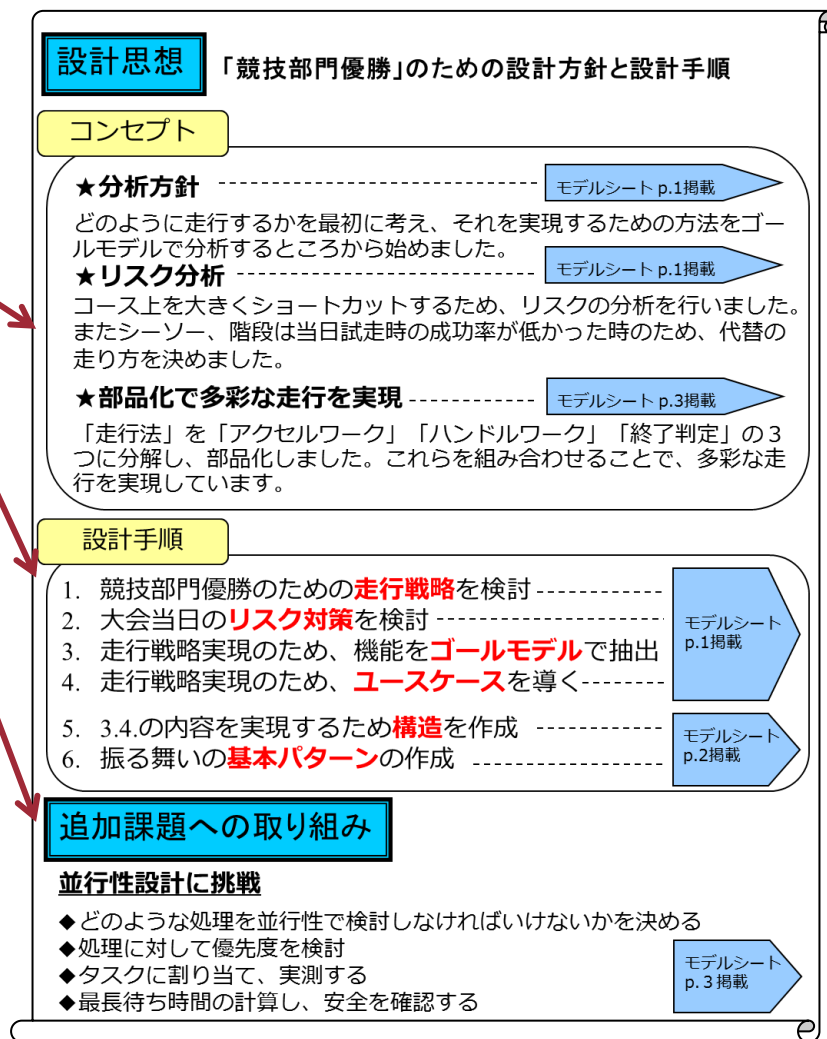
「ノート」の活用で記法の範囲内で説明を収める

図の見やすさ・図の補足 (2)



■ コンセプトシート

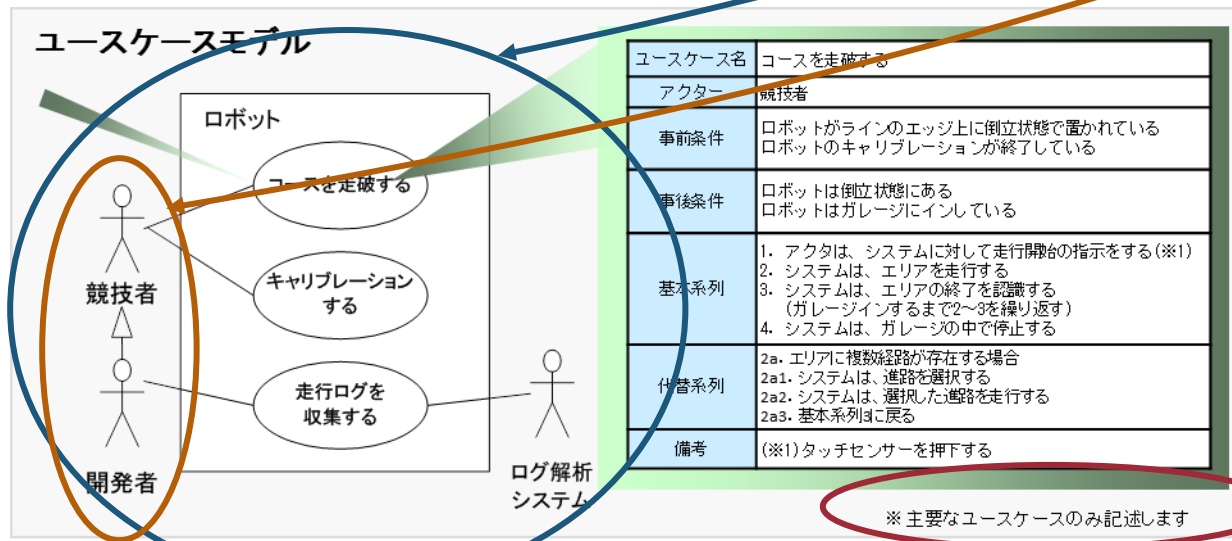
- モデル全体の説明
- 設計上の訴求ポイント
- 追加課題への取り組み



機能モデル



- すべての機能が網羅されている
- 利用者から見て意味・価値のあるものを機能とみなしている
- 機能が多くなる場合は、グルーピングや階層化が実施されている
- モデルの提出枚数に制限があるため、要素の詳細内容の記述は必要ない

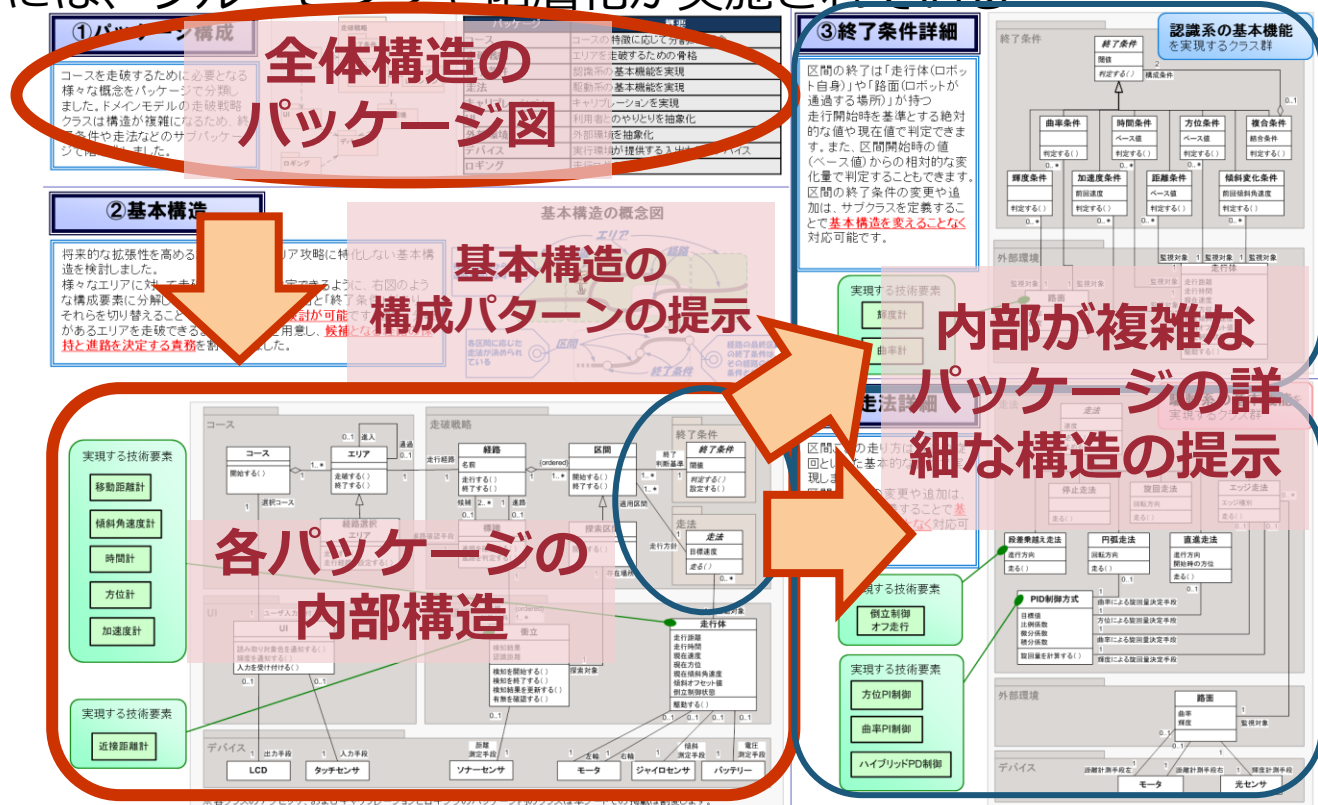


田町レーシング (2010)

構造モデル



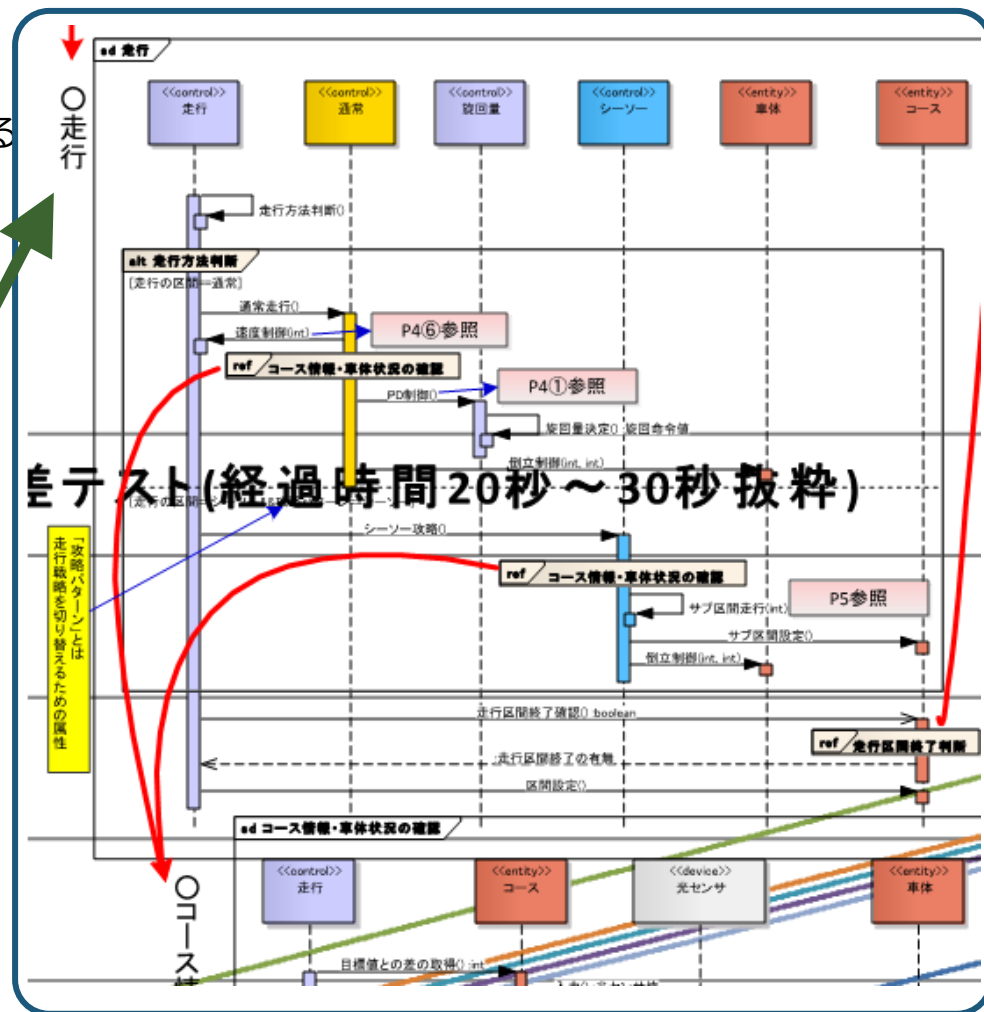
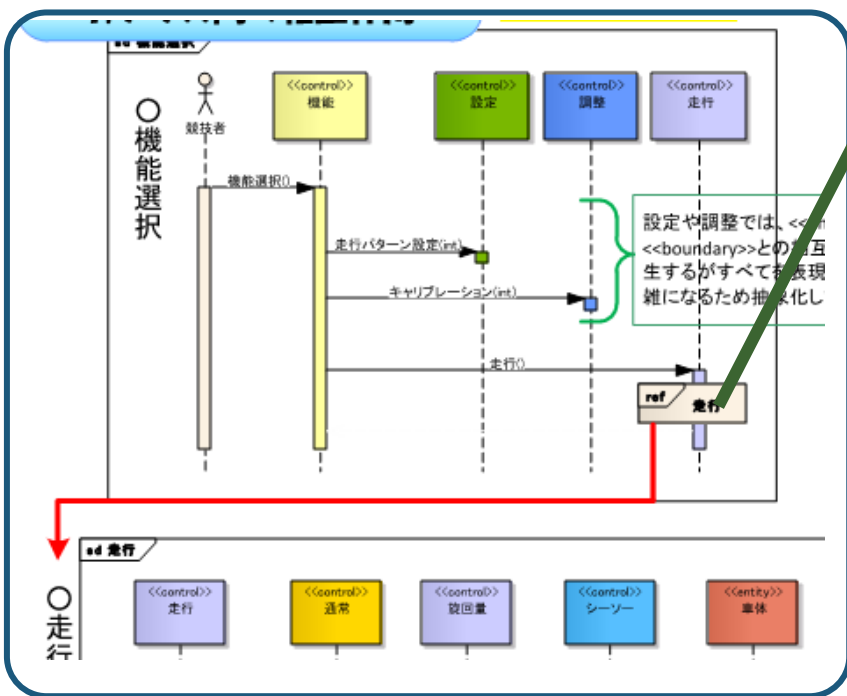
- すべての要素および要素間の関係が網羅されている
- 要素および要素間の関係が妥当である
- 要素の「凝集度の高さ」、要素間の「結合度の低さ」
- 要素が多くなる場合には、グルーピングや階層化が実施されている



振舞いモデル (2)

■ 要素間の相互作用

- 特定の機能に対する要素間の相互作用
- 対象となる機能の実現可能性が判断できる



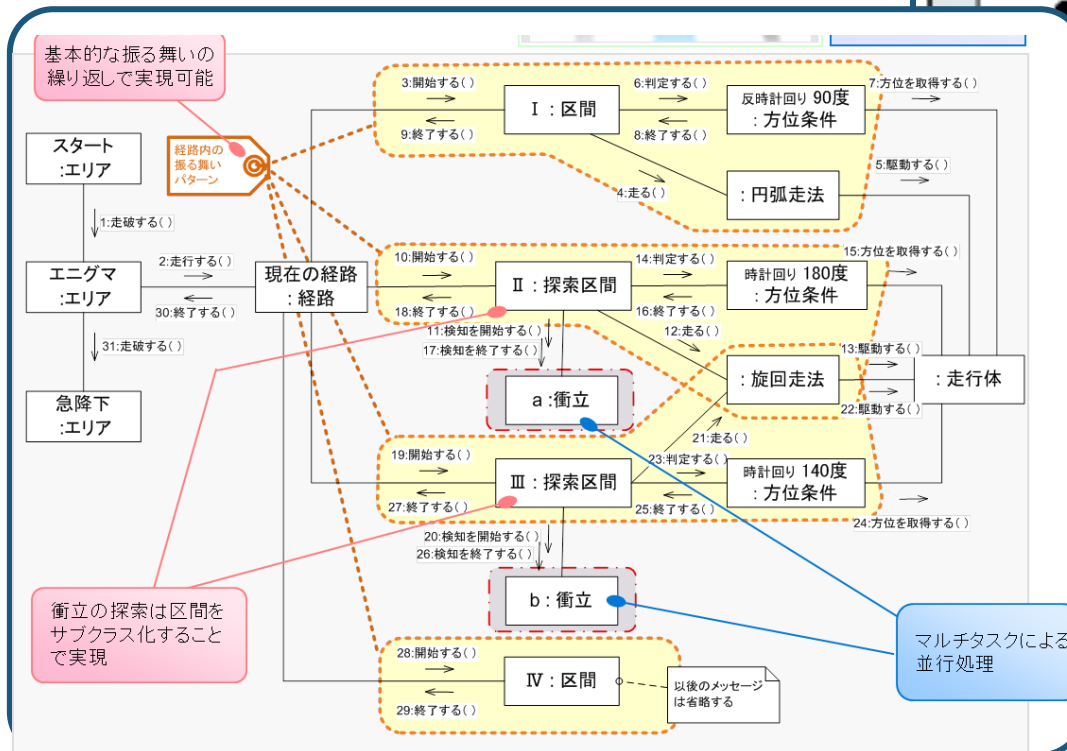
皇テスト(経過時間20秒～30秒抜粋)

走行パターンは「改定パターン」は「改定パターン」を切り替えるための属性

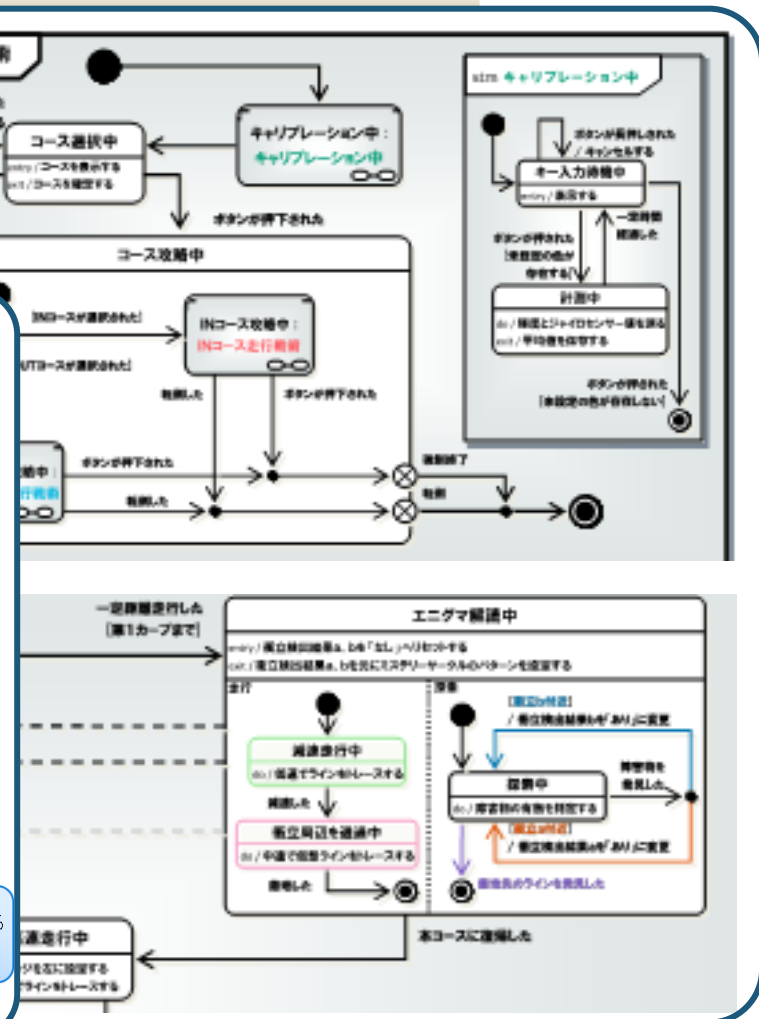
振舞いモデル (3)

■ 要素内部の振る舞い

- 要素内部の振る舞いが記述されている
- 振舞い（状態、イベント、アクションの捉え方）が妥当である



田町レーシング (2010)



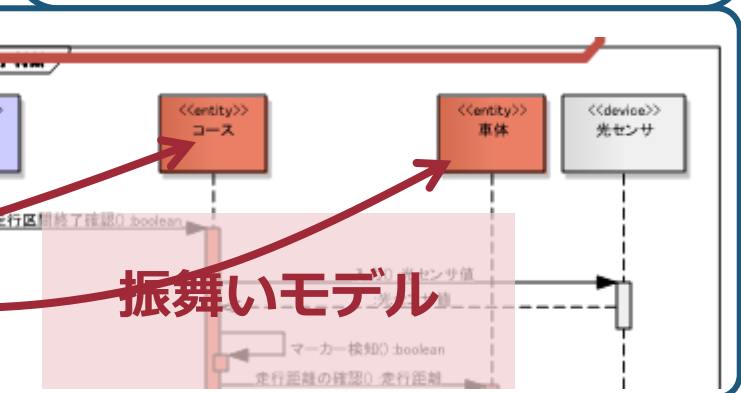
サヌック (2010)

-
- The diagram illustrates a Use Case Model for a system labeled "NXT 次行システム". It features two actors on the left: "競技者" (Competitor) and "観衆者" (Audience). The use cases are represented by ovals and are organized into two groups. A red dashed line encloses the first group, which includes use cases <<UO-01>> (機能を選択する), <<UO-02>> (実行パターンを設定する), <<UO-03>> (キャリブレーションする), and <<UO-04>> (コースを実行する). A text box points to this group, stating it contains the elements developed first, with the exception of those expanded later. A red arrow points to the top of this group. The second group, outside the red dashed line, includes use cases <<UO-05>> (ログを取得する), <<UO-06>> (ライトレースで走行する), <<UO-07>> (場所を記録する), and <<UO-08>> (安全性を保持する). An arrow labeled "<<extend>>" points from <<UO-08>> to <<UO-04>>. The diagram also includes a title "UML ユースケースモデル" and a small icon of a person and a document.
- ```

graph LR
 subgraph "NXT 次行システム"
 direction TB
 UO01((<<UO-01>>
機能を選択する))
 UO02((<<UO-02>>
実行パターンを設定する))
 UO03((<<UO-03>>
キャリブレーションする))
 UO04((<<UO-04>>
コースを実行する
拡張点:
転倒時))
 UO05((<<UO-05>>
ログを取得する))
 UO06((<<UO-06>>
ライトレースで走行する))
 UO07((<<UO-07>>
場所を記録する))
 UO08((<<UO-08>>
安全性を保持する))
 end

 Actor[競技者] --- UO01
 Actor[競技者] --- UO02
 Actor[競技者] --- UO03
 Actor[観衆者] --- UO05
 UO08 -- "<<extend>>" --> UO04

```



# トレーサビリティ (2)



- 走行方法と要素技術に関連性がある
- 要素技術の成果が、設計モデルに紐づけられている

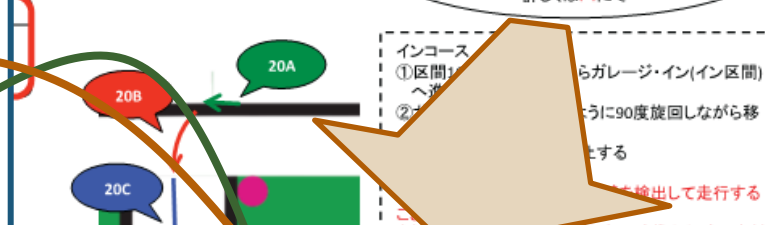
## PID制御

走行体の旋回量(turn)をPID制御で以下のように求める。 $e_n$ : 偏差=(測定値-目標値)  
 $K_P, K_I, K_D$ : 比例、積分、微分パラメータ  
 $\Delta t$ : 時間ステップ(4[ms])  

$$\text{旋回量} = K_P e_n + K_I \sum e_n + K_D \frac{(e_n - e_{n-1})}{\Delta t}$$
 エッジ走行・直進・旋回の各走行の旋回量の目標値と測定値から計算する

| 輝度PID制御(エッジ走行)                                                                  | 角度PID制御(直進)                                                                                            | 曲率PID制御(旋回)                                                                                             |
|---------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|
| 輝度を測定し、ラインレースする<br>・目標値: エッジ走行の輝度しきい値<br>・測定値: 現在の輝度                            | 走行体を指定方向(0~360°)へ直進させる<br>・目標値: 指定した走行体の進行方向<br>・測定値: 現在の走行体の進行方向                                      | 走行体を指定した曲率半径で旋回させる<br>・目標値: 指定した旋回半径<br>・測定値: 現在の曲率半径                                                   |
| <b>テストコース走行結果</b><br>ON/OFF制御と輝度PID制御を比較し、輝度PID制御によるエッジ走行で、ラインからのぶれが少ないことを確認した。 | <b>1m直進 走行結果</b><br>走行体を0°方向に1m直進する動作を、角度PID制御を使用/不使用で比較した。その結果、角度PID制御による直進で左右の誤差を1(cm)以下に出来ることを確認した。 | <b>半径50cm 走行結果</b><br>半径50cmの旋回動作を曲率PID制御で使用する/不使用で比較した。その結果、曲率PID制御による旋回が、より正確でぶれの少ない円軌道で旋回できることを確認した。 |
|                                                                                 |                                                                                                        |                                                                                                         |

## ガレージ・イン走行(区間イン:20,アウト:21)



旋回には曲率PID  
 直進には角度PID  
 を使用して走行を補正している  
 詳しくはP4にて

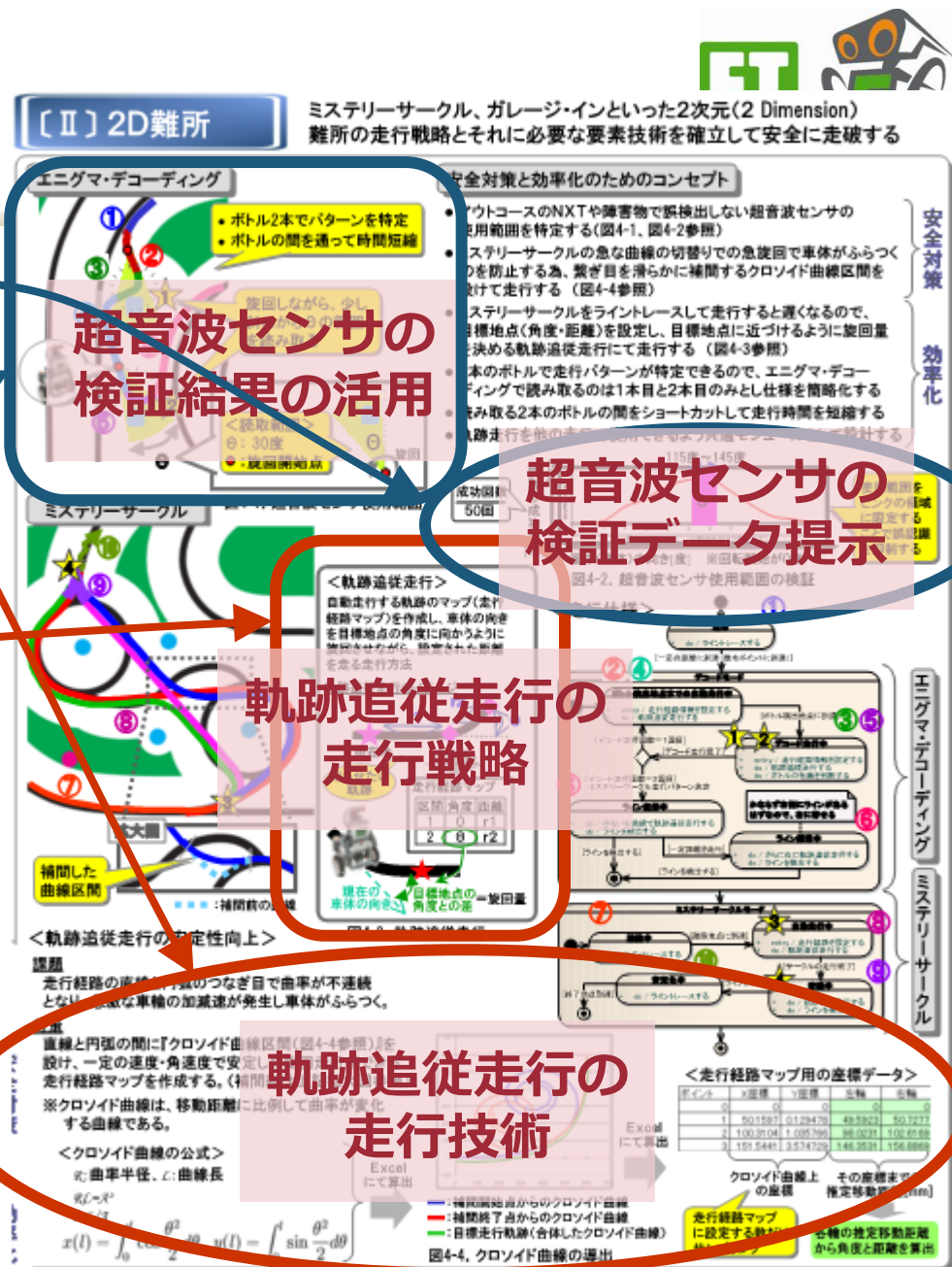
# 性能モデル (1)

## ■ 要素技術

- デバイス要素技術
- 基本走行技術
- 自律性

## ■ 走行戦略

- 走行戦略性能
- 戦略技術の多様性/冗長性/リカバリー性





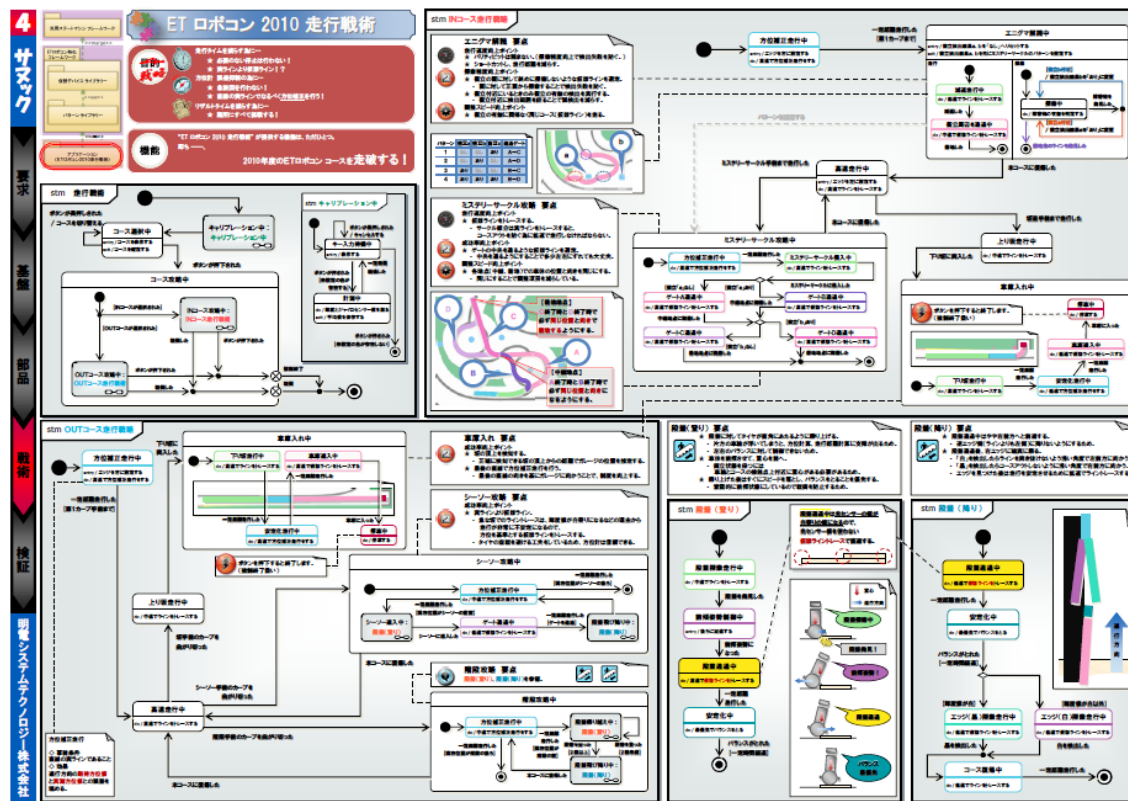


# 性能モデル (3)



## ■ 走行戦略

- コース戦略が万遍なく検討されている
- どの難所を攻略するためにどんな情報が必要かが明示されており、必要な情報を取得してコース攻略に活用している



サヌック (2010)



# 並行性（追加課題）

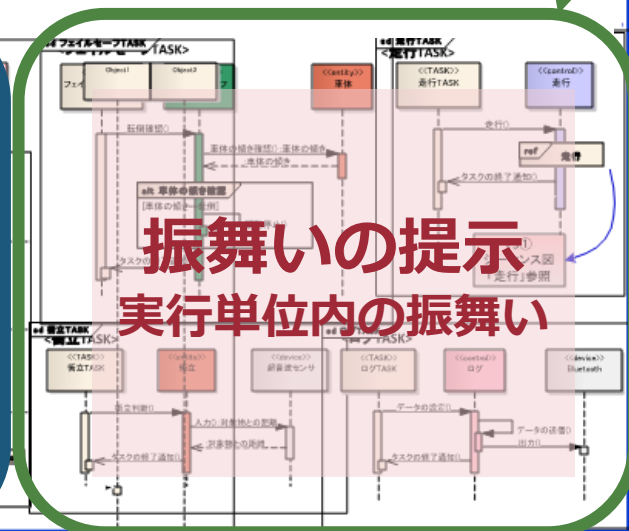
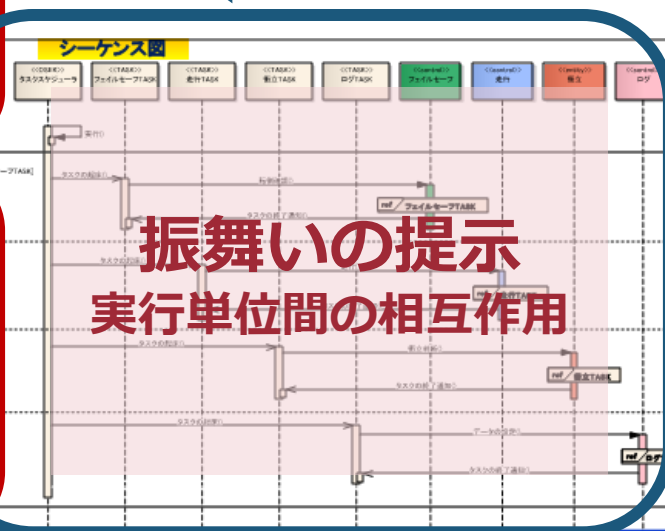
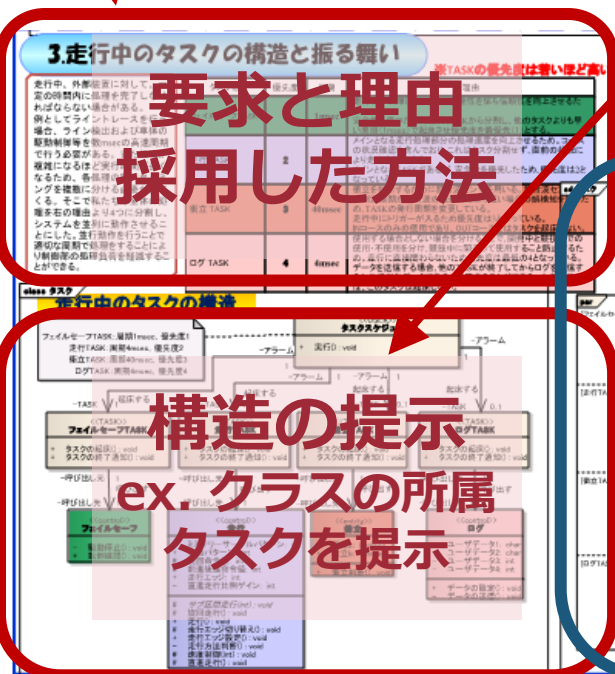


## ■ 設計指針

- 並行性に対する要求と理由
- 要求を実現するために採用した方法

## ■ 設計内容

- 構造面
  - 実行単位を定義している
- 振る舞い面
  - 実行単位間の相互作用
  - 実行単位内部の振る舞い



# 要求モデル（追加課題）（1）

1. 機能モデルを導出するに至った経緯
2. 非機能的な要求の記述

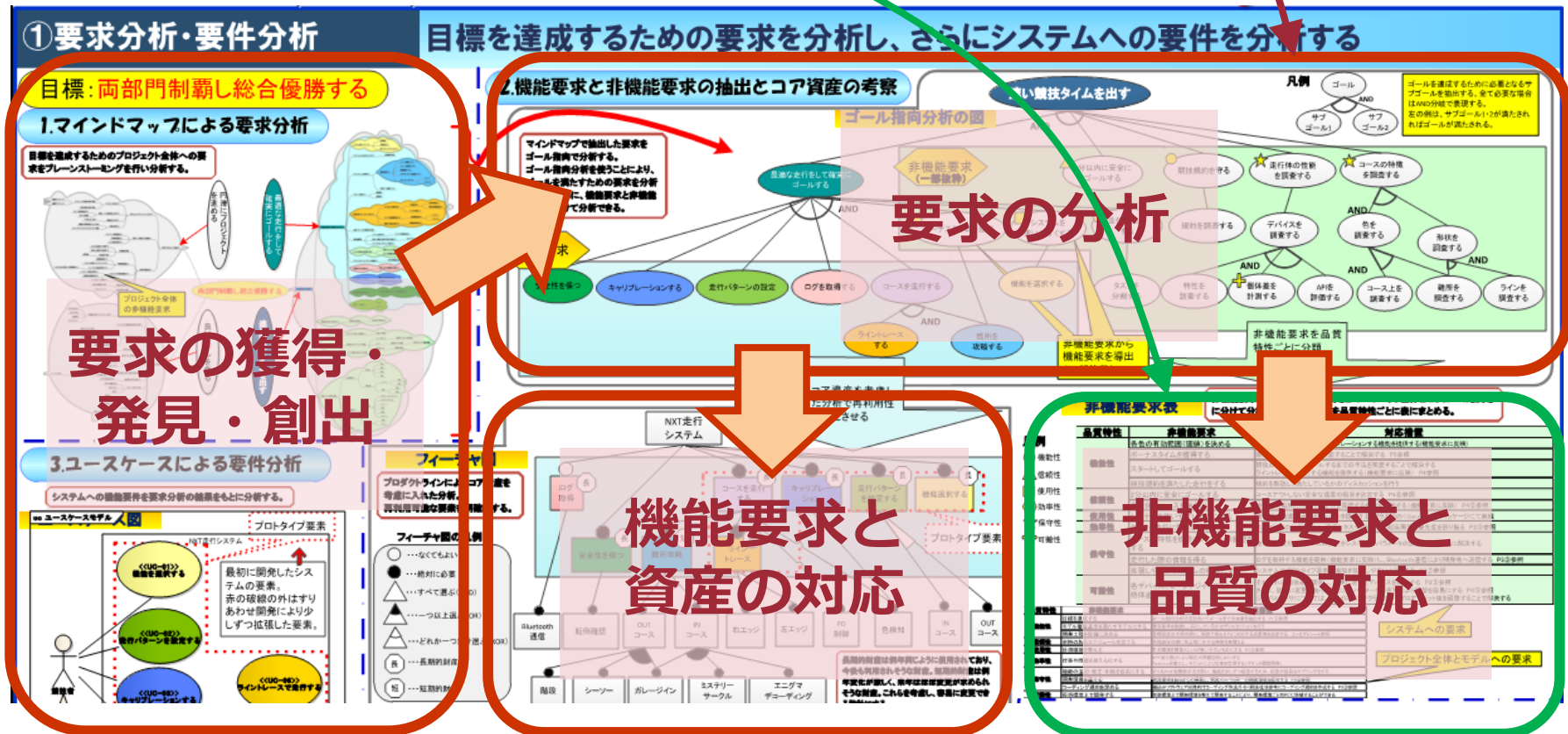


青大ロボコン研MAX2010 (2010)

# 要求モデル（追加課題）（2）



1. 機能モデルを導出するに至った経緯
2. 非機能的な要求の記述





# 開発支援（追加課題）（1）



- 開発効率や品質の向上など、開発環境によって対策する課題・目的を明示している
- 支援手段を提示し、課題との対応づけができています
- 課題に対する効果が客観的に提示できている
- 開発支援は、開発「環境」やツールとは限らない  
変換ルールによる手動実装、プロジェクター投影による情報共有など、支援手法であっても、課題、支援手段、対応付け、効果測定が提示されれば開発支援として評価しうる

**ET ロボコン走行体 監視制御システム**

**目的**  
デバッグや走行教習の妥当性検証に必要なものは、情報は情報です。／また、走行体へのプログラムアップロードは手間のかかる作業です。私たちは「これで良い」と確認できるだけの情報を収集する仕組みが必要だ」「数多い制御パラメータを調整する毎に再コンパイルしたくない」「試走中に発生した事象をすぐに観覧したい」と考え、これらを実現する監視制御システムを構築しました。

ETロボコン走行体 監視制御システムは、次に挙げる7つの機能を提供します。

- ① 走行中の走行体から、内部状態に関するログデータを受信／記録する
- ② 記録しているログデータをリアルタイムにグラフやアニメーションで表示する
- ③ 記録されているログをさまざまな計算やグラフを用いて解析する
- ④ ログデータと解析結果を注釈つきで履歴に残す
- ⑤ 履歴に残された情報を多数のPCで共有する
- ⑥ 制御パラメータを走行体に送信／変更する
- ⑦ 走行体の走行をコントローラーで制御する

**開発 PC**  
監視制御システム  
↓  
分析用ログデータ (Excel 形式)

**クラウド環境に**  
全てのデータはサーバー側で管理します

**クラウド環境に**  
様々な分析結果が  
たくさんグラフやデータとして  
Excel 上に集まる！  
その集まるに新しい分析を！

**いつでも**  
パラメータ変更や走行制御が  
可能！

**注釈とともに 皆で共有！**  
振り返りの記録にも役立つ！

**ログ**  
とったよ

**OK, モデル**  
に載せるよ

**ログは即座に全員で共有できる！**  
モバイルルーターで「観覧」も可能！

**ETロボコン ログデータ**

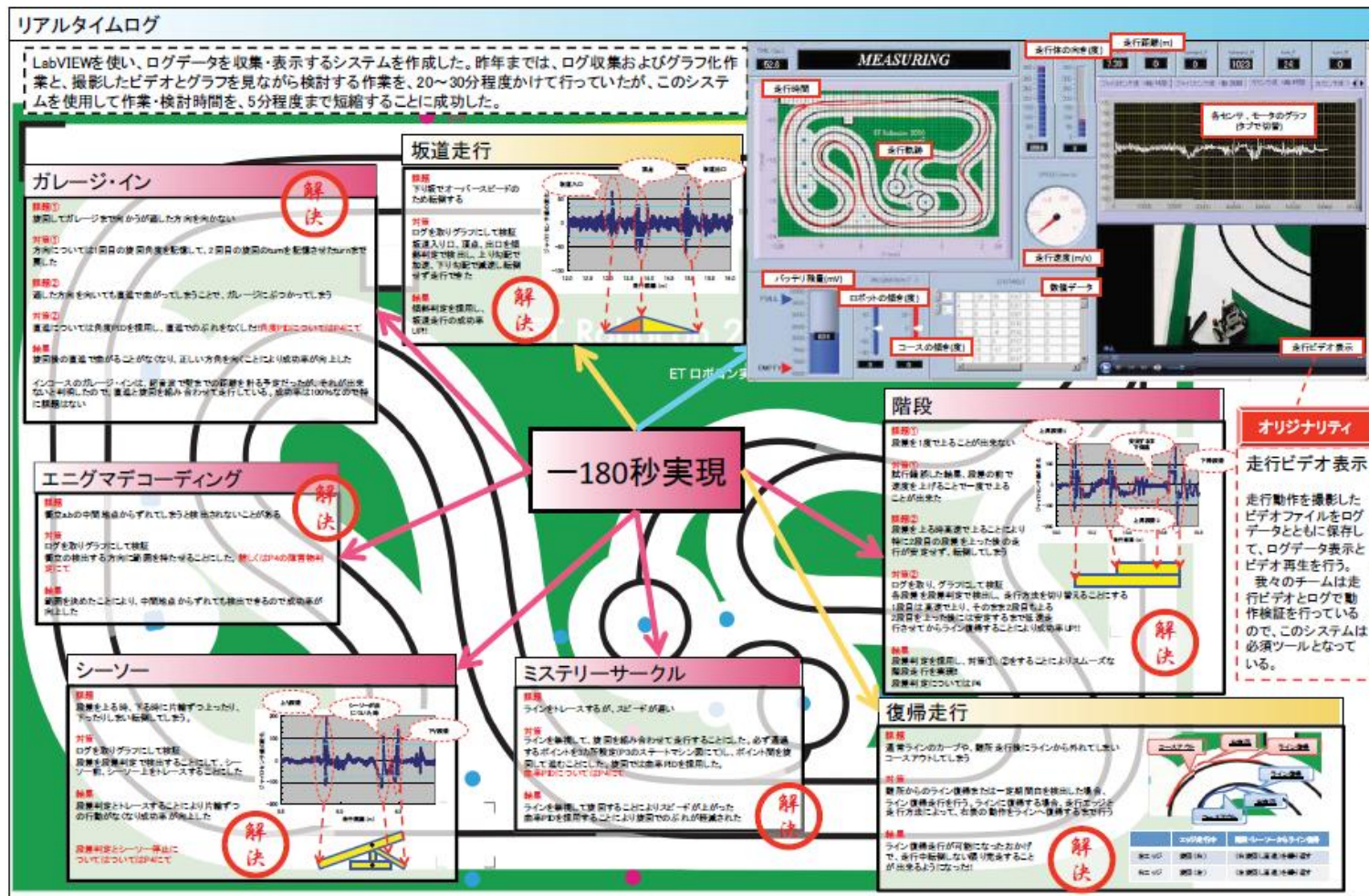
| 日時                  | ログデータ | 種類 | 結果 | コメント | ファイル名               | ダウンロード | 削除 |
|---------------------|-------|----|----|------|---------------------|--------|----|
| 2010/03/10 10:00:00 | 走行開始  | 走行 | 成功 | 走行開始 | 20100310_100000.csv | ダウンロード | 削除 |
| 2010/03/10 10:00:01 | 走行開始  | 走行 | 成功 | 走行開始 | 20100310_100001.csv | ダウンロード | 削除 |
| 2010/03/10 10:00:02 | 走行開始  | 走行 | 成功 | 走行開始 | 20100310_100002.csv | ダウンロード | 削除 |
| 2010/03/10 10:00:03 | 走行開始  | 走行 | 成功 | 走行開始 | 20100310_100003.csv | ダウンロード | 削除 |
| 2010/03/10 10:00:04 | 走行開始  | 走行 | 成功 | 走行開始 | 20100310_100004.csv | ダウンロード | 削除 |
| 2010/03/10 10:00:05 | 走行開始  | 走行 | 成功 | 走行開始 | 20100310_100005.csv | ダウンロード | 削除 |
| 2010/03/10 10:00:06 | 走行開始  | 走行 | 成功 | 走行開始 | 20100310_100006.csv | ダウンロード | 削除 |
| 2010/03/10 10:00:07 | 走行開始  | 走行 | 成功 | 走行開始 | 20100310_100007.csv | ダウンロード | 削除 |
| 2010/03/10 10:00:08 | 走行開始  | 走行 | 成功 | 走行開始 | 20100310_100008.csv | ダウンロード | 削除 |
| 2010/03/10 10:00:09 | 走行開始  | 走行 | 成功 | 走行開始 | 20100310_100009.csv | ダウンロード | 削除 |
| 2010/03/10 10:00:10 | 走行開始  | 走行 | 成功 | 走行開始 | 20100310_100010.csv | ダウンロード | 削除 |

サヌック (2010)

# 開発支援（追加課題）（2）



- 計測ログと動画を組み合わせて記録できるシステムが効果を上げている





# 開発支援（追加課題）（3）



- シンプルに表現されたモータ・走行体のモデルで実機の動作を表現できている
- 何ととっても強力なシミュレーション環境は開発効率の向上に貢献している

## 5-2-3 シミュレーション環境

本機能の制御器設計は主にシミュレーションを用いて設計することを考えた。シミュレーションするため、制御器を表現しなければならない。まず、制御対象のモデル化を考えると、モータ、タイヤ、ロボット本体、タイヤはスリップせず、モータトルクは全てロボット筐体に伝わると仮定することで、タイヤ慣性モーメントを無視する。最終的に制御対象は、モータとロボット筐体を制御対象としてモデル化した。次に、制御器についてモデル化した。

### 5-2-3-1 モータモデル

モータをモデル化するため、システム同定による手法ではなく運動方程式を用いて表現した。その際、シミュレーションにより決定した。

NXT モータは DC モータのため、モータの運動方程式は次のような以下の式で表現できる。

【機械系】

$$J \frac{d^2 \theta}{dt^2} = K_T \cdot i_a - D \cdot \frac{d\theta}{dt} - T_{load}$$

【電気系】

$$L \frac{di_a}{dt} = E_a - K_e \cdot \frac{d\theta}{dt} - R \cdot i_a$$

$$E_a = \frac{PWM}{100} \cdot E_{battery}$$

表2 NXT モータの記号とパラメータ

|            | 記号              | 値                    |          |
|------------|-----------------|----------------------|----------|
| トルク定数      | $K_T$           | 0.31739              |          |
| 回転子逆起電力定数  | $K_e$           | 0.46839              |          |
| 回転子抵抗      | $R$             | 6.8562               | $\Omega$ |
| 回転子インダクタンス | $L^*$           | 0.1                  | H        |
| 慣性モーメント    | $J^*$           | $1/(1e-5)$           | $Kgm^2$  |
| バッテリー電圧    | $E_{battery}^*$ | $9.5 (7.6 \sim 9.5)$ | V        |

## 5-2-3-2 制御器

制御器は、モータロータリーエンコーダからの値から、地上固定座標系に対するロボット筐体の座標及び角度を算出し、コース情報の座標値により、ロボット本体の座標系に座標変換し、ロボット本体の進行方向と、目標方向とのずれを偏差として入力する PID 制御器を用いたフィードバック制御とした。また、PID 制御器は離散化されたアンチwindアップ機能を含む非干渉形 PID 制御器として構成している（詳細は省く）。

## 5-2-4 シミュレーションによる制御パラメータの自動計算

制御器の各ゲインは、制御出力と偏差が最小になるよう、シミュレーションと数値イテレーションによる最適化演算により算出した（詳細な記載は省く）。

## 5-2-5 シミュレーション結果及び実機による動作結果

シミュレーション結果及び実機による動作結果を以下に示す。

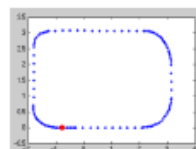


図9 オープルコース設定

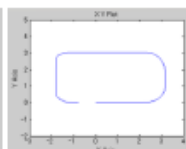


図10 シミュレーション結果1

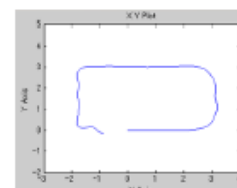


図11 シミュレーション結果2

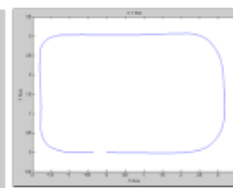


図12 実機による動作結果

・図9 コース情報から取得した座標値（オープンコース教習コース）

・図10 シミュレーション結果1（コース情報に基づく座標値の更新軌跡）

・図11 シミュレーション結果2（コース情報の座標値をターゲットとしたロボット本体の軌跡）

・図12 実機による動作結果（コース情報の座標値をターゲットとしたロボット本体の軌跡）

図12 より、シミュレーションが概ね実機動作と一致していることを確認できた。よって、PID 制御器のゲインパラメータをシミュレーション上である程度決定可能であることがわかった。更に、シミュレーションを利用し、コース情報から取得する座標値をショートカット走行用に成形し検対することができ、また、旋回における PWM の最大値についても確認可能であることの裏付けとなった。

M a s h i - M a s h i (2010)