ETロボコン2018

デベロッパー部門 プライマリークラス

チームNo. 202 チーム名: **tadaima G3**

情報技術で未来を創設

地区:東京地域:新宿区





チーム紹介、目標、意気込み

tdi 情報技術開発株式会社の tadaima G3です。

新入社員4人が中心のチームです。

今年の目標は、 モデルの評価B+以上を獲得し、 チャンピオンシップ大会に出場することです! 基本走行も難所も完全攻略します!

どんな試練もチーム一丸となって乗り越え、最高の結果を掴み取ります!

モデルの概要

選択課題…シーソーを通過する

基本方針…完全攻略&安全第一

- ユースケースを実現するために必要な**挙動**は、**走行方法と終了判定** で構成されているものとした。そのうち**走行方法**について『**安全第一**』を 妨げるリスクを分析した。その結果から『完全攻略&安全第一』を実 現する**走行戦略**を定義した。
- 変更の影響範囲を最小限に抑えつつ、機能拡張が容易である構造 モデルを目指し、**挙動、走行方法、終了判定**をそれぞれ別々のクラス としてクラス図を作成した。
- 挙動の実行を制御しているクラスに着目し、その振る舞いをステートマシン図とシーケンス図で表した。また、挙動のひとつを取り上げ、**走行方法と終了判定**の詳細な振る舞いをシーケンス図で示した。
- リスク回避機能の実装前後で効果を比較し、リスク回避に繋がったことを確認できた。

モデルの構成

タイトル	内容
1. 機能モデル	シーソーを『完全攻略』するための要求分析を行った。 ミスユースケース図で、ユースケースと脅威を抽出し、 ユースケースを実現するために必要な挙動を洗い出した。 洗い出した挙動は、走行方法と終了判定で構成されて いる。そのうち走行方法について、『安全第一』を実現 するために、発生しうるリスクを分析し、対策を考案した。 要求分析とリスク分析の結果を総合し、 『完全攻略&安全第一』を実現する走行戦略の定義を 行った。
2. 構造モデル	走行戦略を踏まえた構造モデルの設計を行った。 1つの挙動は走行方法と終了判定で構成されており、 それぞれ別々のクラスとしてクラス図を作成した。 また、責務や役割毎に、6つのパッケージを定義した。 構造モデルは、変更の影響範囲を最小限に抑えつつ、 機能拡張が容易であることを目指した。
3. 振る舞いモデル	クラス図で静的に表現した一連の挙動が、 どのように行われているかを示した。 挙動の実行を制御しているシーソー攻略クラスに着目し、 その振る舞いをステートマシン図とシーケンス図で表した。 各挙動の走行方法及び終了判定の組み合わせを整理 し、その中から具体的な例として「後退する」という挙動に ついて、詳細な振る舞いを示した。
4. 工夫点	機能モデルのリスク分析から導出されたリスク対策機能「ライン復帰」について、実現方法の詳細を示した。 「ライン復帰」機能を実装したことにより、昇段後のライントレースの成功率が向上し、リスク回避に繋がった。

1. 機能モデル

tadaima G3

選択課題シーソーを通過する

基本方針

完全攻略&安全第一

1.1 要求分析

競技規約をもとに、 シーソーを『完全攻略』し、 最大ボーナスタイムを獲得する ユースケースを抽出した。 同時に、『安全第一』を実現 するにあたって妨げとなる脅威 を抽出した。 これらをミスユースケース図

(図1-1) 及びユースケース

記述(表1-1)で示す。

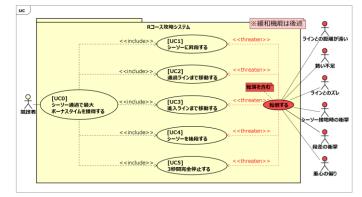


図1-1:「シーソーを通過する」を攻略するミスユースケース図

表1-1・「シーソーを通過する」を攻略するフーフケーフ記述

表1-1:「シーソーを通過する」を攻略するユー人ケー人記述			
ユースケース名	シーソーにおいて最大ボーナスタイムを獲得する		
概要	走行体がシーソーに進入し、通過ライン・進入ライン・通過ライン の順で接地させてから降段し、着地エリアで完全停止する		
アクター	競技者		
事前条件	走行体がシーソーの進入ラインに向かってライントレースしている		
トリガー	走行体が灰色を検知する		
基本フロー	UC1 シーソーに進入するまで前進する UC2 通過ラインが接地するまで前進する UC3 進入ラインが接地するまで後退する UC4 シーソーを通過するまで前進する UC5 着地エリアで3秒間完全停止する		
代替フロー	Alt-1:UC5で、3秒間経過が検知できなかった時 1.システムは完全停止を継続する		
例外フロー	Ex-1:基本フローの途中で転倒した時 1.システムはこのユースケースを中止する		
事後条件	最大ボーナスタイムを獲得している		
備考	完全停止とは、走行体に接続されている 全てのモータが回っていない状態		

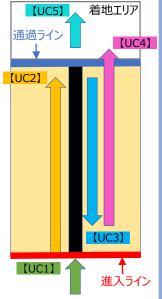


図1-2:基本フローの動作イメージ

Point!!

基本フローを実現するために必要な挙動と、その際の走行方法及び走行姿勢について、 下表1-2のとおり洗い出し、これを走行戦略とした。

表1-2:基本フローに基づく走行戦略

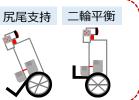
基本フロー	UC1	UC2	UC3	UC4	UC5
挙動	UC1-1.昇段する	UC2-1.前進する	UC3-1.後退する	UC4-1.降段する	UC5-1.停止する
説明図		3			
走行方法	ライントレース(シーソー左右両端から転落しないよう、シーソー中央を走行するため) 停止				
走行姿勢	二輪平衡	尻尾支持	尻尾支持	二輪平衡	尻尾支持

走行体の姿勢を**尻尾支持と二輪平衡**の2つに分けた。(右イラスト) 倒立振子では、姿勢安定のために行う前後動作を 制御できない等の潜在的なリスクがあるため、 シーソー上での姿勢については、基本的に尻尾支持とした。

ただし、尻尾が段差に引っかかるリスクがあるため、

昇降段時は二輪平衡とした。

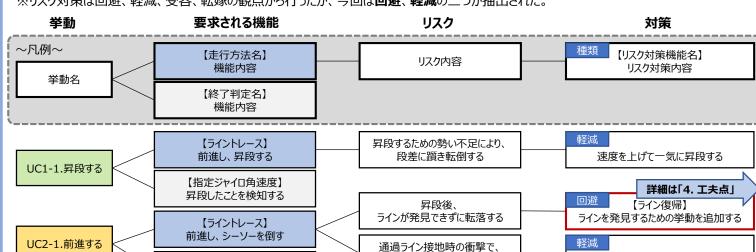




1.2 リスク分析

「シーソーを通過する」を攻略するために、走行戦略(表1-2)で示した挙動に求められる機能を定義した結果 各挙動は**走行方法**に加え、それをいつまで行うのかという**終了判定**の2つで構成されていることが分かった。 その上で、基本方針である『安全第一』を実現するために、ミスユースケース図(図1-1)で挙げた 「転倒する」という脅威に着目したリスク分析と、その対策の検討を行った。(図1-3)

※リスク対策は回避、軽減、受容、転嫁の観点から行ったが、今回は**回避、軽減**の二つが抽出された。



重心を後ろにずらす 【指定ジャイロ角速度】 シーソーが傾いたことを検知する 回避 カラーセンサとラインとの距離が遠く 後退時はライントレースを行わない ラインを認識できずに転落する 【ライントレース】 後退し、シーソーを倒す 軽減 UC3-1.後退する 進入ライン接地時の衝撃で、 後ろに転倒する 重心を前にずらす 【指定ジャイロ角速度】 シーソーが傾いたことを検知する 軽減 【ライントレース】 降段時の衝撃で、 速度を落として、衝撃を減らす 前進し、降段する 転倒する UC4-1.降段する 【指定ジャイロ角速度】 降段したことを検知する

完全停止する UC5-1.停止する

【指定経過時間】 時間が経過したことを検知する

【停止】

尻尾支持に移行する際、

重心が前に偏ると転倒する

前に転倒する

重心を後ろに置いた後、尻尾支持する ※軽減リスクに関しては、

尻尾角度・走行速度を調整する

【直進】

リスク分析結果を踏まえ、『安全第一』を実現するために、「ライン復帰する」という挙動と、「直進」という走行方法を走行戦略(表1-2) に取り入れた。また、各挙動における走行方法及び終了判定も併せ、これを**新走行戦略**(表1-3)として再定義する。

図1-3: リスク分析マップ

「ライン復帰する」は実際には複数の挙動で 構成されるので、詳細は「**4. 工夫点**」で示す。 表1-3: リスク分析に基づく新走行戦略

直前の挙動(UC3-1)で、ラインを見失う 可能性があるため、ライン復帰する必要がある。

軽減

基本フロー	UC1	UC	2	UC3	UC	24	UC5
挙動	UC1-1.昇段する	UC2-1.ライン復帰する	UC2-2.前進する	UC3-1.後退する	UC4-1.ライン復帰する	UC4-2.降段する	UC5-1.停止する
説明図							
走行方法 N	ew ライントレース	ライン復帰	ライントレース	直進 New	ライン復帰	ライントレース	停止
終了判定	指定ジャイロ角速度	指定輝度値	指定ジャイロ角速度	指定ジャイロ角速度	指定輝度値	指定ジャイロ角速度	指定経過時間
走行姿勢	二輪平衡	尻尾支持	尻尾支持	尻尾支持	尻尾支持	二輪平衡	尻尾支持
		New			New		

2. 構造モデル

tadaima G3

選択課題 シーソーを通過する

2.1 設計方針

「1. 機能モデル」で、要求分析とリスク分析を行った結果を用い、 構造の検討を行った。

「1.2 リスク分析」で述べたように、1つの挙動は走行方法と 終了判定で構成されており、それを実現する構成になっている。 変更時の影響範囲を最小限に抑えつつ、機能の追加をしやすい 設計を目指した。

2.2 パッケージ構成

青務や役割毎に6つのパッケージを定義した。

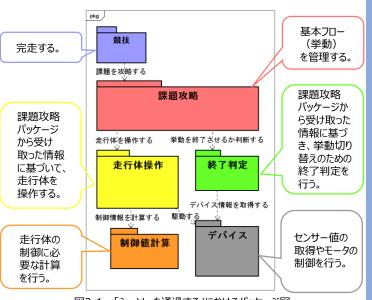


図2-1:「シーソーを通過する」におけるパッケージ図

2.4 インスタンス関係

挙動実行時のインスタンス関係を分かりやすく示すために オブジェクト図を用いる。

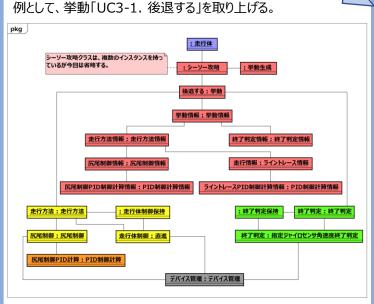


図2-3:「UC3-1. 後退する」実行時におけるオブジェクト図

2.3 クラス構造

基本方針である『完全攻略&安全第一』を実現するためのクラス構造をクラス図(図2-2)で示す。

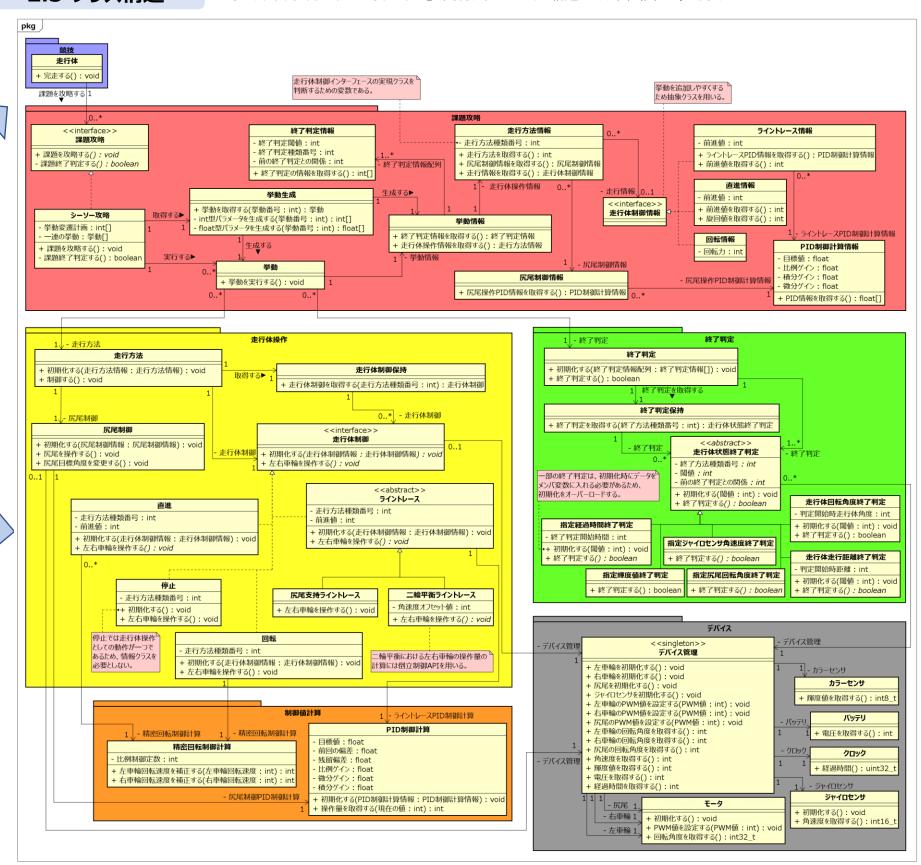


図2-2:「シーソーを通過する」におけるクラス図

3. 振る舞いモデル①

tadaima G3

どうすれば いいの?

選択課題 シーソーを通過する

3.1 シーソー攻略クラスの振る舞い

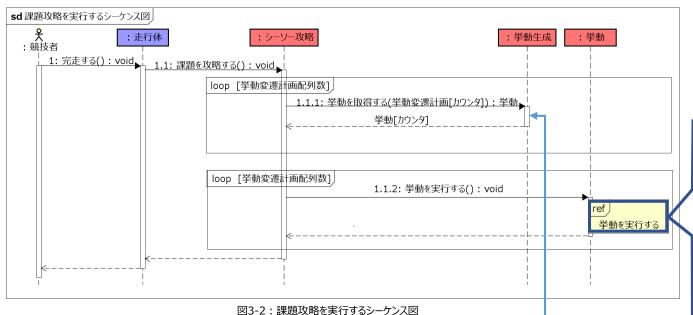
- 「1.機能モデル」で示した新走行戦略(表1-3)は、複数の挙動を順番に実行することによって実現する。
- 「2. 構造モデル」のクラス図(図2-2)において、挙動の実行を制御しているクラスはシーソー攻略クラスである。
- シーソー攻略クラスの属性である挙動の振る舞いを、右記のステートマシン図(図3-1)に示す。
- また、その挙動がどのように実行されているのかをシーケンス図(図3-2)に示す。

表1-3:リスク分析に基づく新走行戦略(再掲載)

基本フロー	UC1	UC2		UC3	UC4		UC5
挙動	UC1-1.昇段する	UC2-1.ライン復帰する	UC2-2.前進する	UC3-1.後退する	UC4-1.ライン復帰する	UC4-2.降段する	UC5-1.停止する
説明図							
走行方法	ライントレース	ライン復帰	ライントレース	直進	ライン復帰	ライントレース	停止
終了判定	指定ジャイロ角速度	指定輝度値	指定ジャイロ角速度	指定ジャイロ角速度	指定輝度値	指定ジャイロ角速度	指定経過時間
走行姿勢	二輪平衡	尻尾支持	尻尾支持	尻尾支持	尻尾支持	二輪平衡	尻尾支持



図3-1:シーソー攻略クラスのステートマシン図



各挙動で実行する走行方法及び終了判定は、

それぞれ走行方法種類番号と終了判定種類番号で管理している。 挙動と種類番号の対応を表3-1に示す。

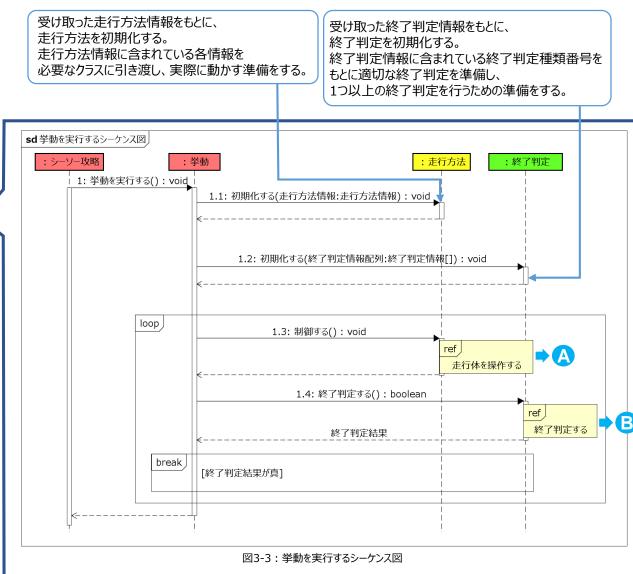
※ライン復帰に関しては「4. 工夫点」で触れるため省略する。

引数で取得した挙動番号と合致する挙動情報を生成し、 それを用いて挙動を生成し、返却する。

「**3. 振る舞いモデル**②」では、UC3-1に着目する

表3-1: 挙動と種類番号の組み合わせ

	種類番号	機能名	UC1-1	UC2-2	UC3-1	UC4-2	UC5-1
+	1	二輪平衡ライントレース	0			0	
走行方法	2	尻尾支持ライントレース		0	0		
法	3	停止					0
終 了	1	指定ジャイロ角速度	0	0	0	0	
判定	2	指定経過時間					0



3.2 挙動「後退する」の振る舞い

 $A \Rightarrow$

「3.1 シーソー攻略クラスの振る舞い」で示した「挙動を実行するシーケンス図」(図3-3)において、refとした「走行体を操作する」と「終了判定を行う」のシーケンス図を以下に示す。 シーソー攻略は複数の挙動で構成しているが、ここでは「1. 機能モデル」の表1-3で示した挙動「UC3-1. 後退する」を取り上げる。

sd 走行体を操作する (尻尾制御+直進) シーケンス図 尻尾:モータ 右車輪:モータ 左車輪:モータ : 挙動 : 走行方法 : 尻尾制御 <<singleton>> : デバイス管理 PID制御計算 1:制御する(): void、 1.1: 尻尾を操作する(): void 1.1.1: 尻尾の回転角度を取得する(): int 1.1.1.1: 回転角度を取得する(): int32_t 1.1.2: 操作量を取得する(回転角度): int 尻尾PWM値 1.1.3: 尻尾のPWM値を設定する(尻尾PWM値): void 1.1.3.1: PWM値を設定する(PWM値:int): void 1.2: 左右車輪を操作する() 1.2.1: 右車輪回転速度を補正する(右車輪回転速度:int): int 右車輪PWM値 1.2.2: 右車輪のPWM値を設定する(PWM値:int): void 1.2.2.1: PWM値を設定する(PWM値:int): void 1.2.3: 左車輪回転速度を補正する(左車輪回転速度:int): int 左車輪PWM値 1.2.4: 左車輪のPWM値を設定する(PWM値:int): void 1.2.4.1: PWM値を設定する(PWM値:int): void

表1-3:リスク分析に基づく新走行戦略(抜粋)

基本フロー	UC3	
挙動	UC3-1.後退する	
説明図		
走行方法	直進	
終了判定	指定ジャイロ角速度	
走行姿勢	尻尾支持	

なんだかできそうだ!

図3-4: 「UC3-1. 後退する」における走行体を操作するシーケンス図

ま2.2. タフ州ウレの門を

: 挙動	: 終了判定 : 指定ジャイロセンサ角速度終了判定 : デバイス管理
1: 終了判定する()	: boolean
	1.1.1: 終了判定する(): boolean 1.1.1: 角速度を取得する(): int 1.1.1.1: 角速度を取得する(): int 1.1.1.1: 角速度を取得する(): int 1.1.1.1: 角速度を取得する(): int16_t 角速度 角速度 角速度
	[前の終了判定との関係 == 2] 終了判定結果 = 終了判定結果 OR 今回の終了判定結果
	[前の終了判定との関係 == 3] 終了判定結果 = 終了判定結果 AND 今回の終了判定結果
終了判定統	課

図3-5:	ΓUC3-1.	後退する」における終了判定を行うシーケ	ンス図

	表3-2 : 終了判定と	の関係
前の終了判定 との関係	説明	使用例
1	使用演算子: なし単一の走行体状態終了判定で終了 判定を行う場合に用いる事前の走行体状態終了判定がない 場合に用いる	指定ジャイロ角速度終了判定のみを使用し、 判定が真だった時に挙動を終了させたい場合。
2	使用演算子: OR複数使用する終了判定のうち、いずれかの終了判定を満たした場合に終了させたい場合※ライン復帰で用いる	指定走行距離終了判定と、指定輝度値終了 判定のうち、少なくとも1つの判定が真だったとき に挙動を終了させたい場合。
3	使用演算子: AND複数使用する終了判定のうち、 全ての終了判定を満たした場合のみ 終了させたい場合※シーソー攻略では用いない	指定走行距離終了判定と、指定輝度値終了 判定のうち、全ての判定も真だったときに挙動を 終了させたい場合。

4. 工夫点

tadaima G3

選択課題

シーソーを通過する

4.1 ラインを見つけるには

<--- 「1. 機能モデル」より

走行中、シーソーの左右両端から転落するというリスクを軽減するためにライントレースを行っているが、 ライントレースを行うためには、シーソー上で**確実にラインを見つける必要がある**。

ラインを見つけるために必要となる一連の動きを「ライン復帰」と定義し、

ライン復帰における挙動を下表4-1に示す。

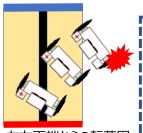


表4-1:ライン復帰における挙動と機能

図4-1:左右両端からの転落図

	公TI. 江口岡州III ウバ和石区			
挙動	1.左回転する 2.前進する 3.後退する 4.右回転する 5.前進する 6.後退する			
説明図				
走行方法	回転 直進 回転 直進			
終了判定1	回転角度 走行距離 回転角度 走行距離 走行距離 走行距離			
終了判定2	指定輝度値			
走行姿勢	尻尾支持			
備考	ライン復帰は、挙動1から6を2セット繋げたもので構成されている。2セット目は1セット目より捜索範囲を 広げるため、前進・後退する距離を延ばす。終了判定1と2の関係はORであるため、指定輝度値終了 判定が真だった場合、ライン復帰の挙動すべての終了判定も真となるため、ライン復帰の動きが終了する。			

走行方法



右車輪の走行距離 R[mm] 左車輪の走行距離 L[mm] 車輪間の距離 W[mm] 車輪の半径 r[mm]

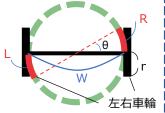


図4-2:参考図

挙動1及び4の左右回転に関して、

回転角度を大きくしすぎると、シーソーから転落してしまうリスクがある。 走行体が回転した**角度**を算出するための計算式は、

回転角度 $\theta_{rad} = \frac{R-L}{W}$

回転角度を指定する際、radよりもdegの方が競技者にとって明瞭なため、 単位変換する。 $2\pi r = 360$ °より

回転角度 $\theta_{deg} = \frac{180}{\pi} \times \theta_{rad} = r \times \frac{R-L}{W}$

終了判定



指定した回転角度と、 モータ角位置を基に 導き出した回転角度が 一致したことを判定する。



指定した<mark>走行距離</mark>と、 モータ角位置を基に 導き出した<mark>走行距離</mark>が 一致したことを判定する。

モータの個体差等により、導き出した値と実際の値の間には誤差が生じる



指定した目標輝度値と、

カラーセンサが取得した値が一致したかどうかを判定する。 指定する目標輝度値に関しては、黒色ライン中央の値とする。

4.2 効果測定

ライン復帰の導入前後で、 UC2の成功率の変化を測定し、 結果を右図(図4-3)にまとめた。 テスト環境は、以下のとおりである。

さあ検証だ!!

テスト環境

2018年度レプリカコース

室内でスタイロフォームを敷いて行う電圧 : 7900~8300[mV]

• 実験回数:20回

時間帯 : 09:00~11:00外乱光 : 蛍光灯、外からの光

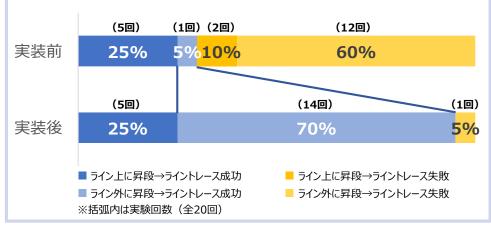


図4-3: ライン復帰導入前後におけるUC2の成功率の比較

~結論~

実験結果より、ライン復帰の導入前後で、ライントレース 成功率が30%から95%へ向上したことが分かる。 昇段後のライン復帰を導入することによって、 走行体はほぼ確実にライントレースを行うようになった。 失敗した1回については、回転角度・走行距離の 誤差によるものが原因と想定しており、 この問題は補正を行うことによって、精度を高めることができる。 従って、走行中にシーソーの左右両端から転落するという リスクを回避することができた。

