

# 1. 要求

# 2. 分析

# 3. 制御

# 4. 設計(1)

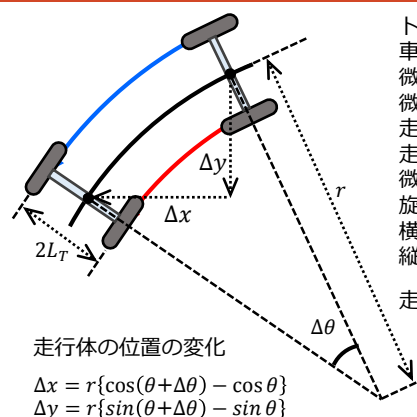
# 4. 設計(2)

しんよこ  
(° D°)

Rコースをクリアするための要素技術を検討し、制御戦略を立てた

## 3-1. 自己位置推定機能

### 要素技術



トレッド長(mm):  $2L_T$   
車輪半径(mm):  $R_W$   
微小時間での右サーボモーターの回転角の変化量(deg.):  $\Delta\phi_R$   
微小時間での左サーボモーターの回転角の変化量(deg.):  $\Delta\phi_L$   
走行体の座標(mm):  $x, y$   
走行体の方向(rad.):  $\theta$   
微小時間での機体の進行方向の変化量(rad.):  $\Delta\theta$   
旋回半径:  $r$   
横軸方向の位置変化(mm):  $\Delta x$   
縦軸方向の位置変化(mm):  $\Delta y$   
走行体の旋回半径と方向変化  
$$r = \frac{L_T(\Delta\phi_R + \Delta\phi_L)}{\Delta\phi_R - \Delta\phi_L}$$
$$\Delta\theta = \frac{\pi R_W(\Delta\phi_R - \Delta\phi_L)}{360L_T}$$
  
(直進時は無限大となる)  
走行体の位置の変化  
$$\Delta x = r\{\cos(\theta + \Delta\theta) - \cos\theta\}$$
$$\Delta y = r\{\sin(\theta + \Delta\theta) - \sin\theta\}$$

## 3-3. 座標指定移動

### 要素技術

走行体の現在位置  $\vec{P}_0 = \begin{pmatrix} P_{0x} \\ P_{0y} \end{pmatrix}$  での速度および向き  $\vec{v}_0 = \begin{pmatrix} v_{0x} \\ v_{0y} \end{pmatrix}$

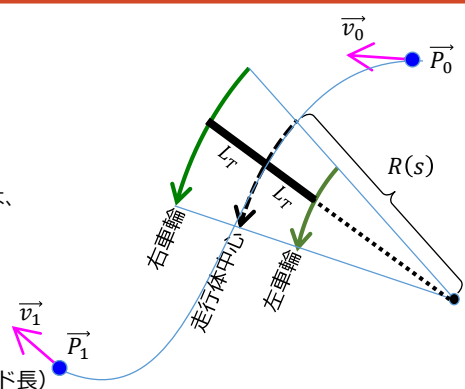
目標点  $\vec{P}_1 = \begin{pmatrix} P_{1x} \\ P_{1y} \end{pmatrix}$  での速度および向き  $\vec{v}_1 = \begin{pmatrix} v_{1x} \\ v_{1y} \end{pmatrix}$

3次スプライン曲線  $\vec{P}(s) = \vec{a}_0 + \vec{a}_1s + \vec{a}_2s^2 + \vec{a}_3s^3$  ( $0 \leq s \leq 1$ ) の係数は、

$$\begin{aligned} \vec{a}_0 &= \vec{P}_0 \\ \vec{a}_1 &= \vec{P}_1 \\ \vec{a}_2 &= 3\vec{P}_1 - 3\vec{P}_0 - 2\vec{v}_0 - \vec{v}_1 \\ \vec{a}_3 &= -2\vec{P}_1 + 2\vec{P}_0 + \vec{v}_0 + \vec{v}_1 \end{aligned}$$

$$\text{曲率半径 } R(s) = \frac{|\vec{P}(s)|^3}{|\vec{P}(s)' \times \vec{P}(s)'|}$$

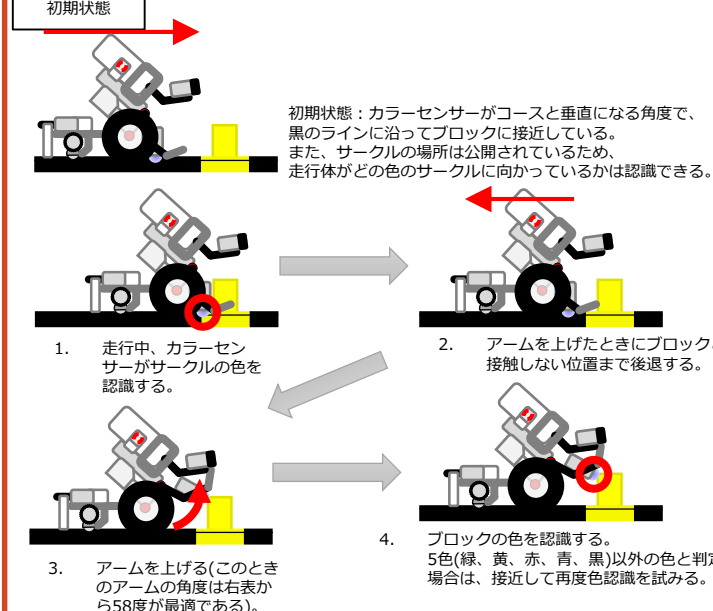
左右モーターの回転速度の比 =  $(R(s) - L_T) : (R(s) + L_T)$  ( $L_T$ : トレッド長)  
現在の右車輪の回転速度から、左車輪の回転速度の目標値を決定し、負のフィードバック制御を行う。



## 3-5. 色認識・アーム制御

### 要素技術

色認識、アーム制御、車輪制御を用いて、次の手順によってブロックをアームに収めることを実現する。



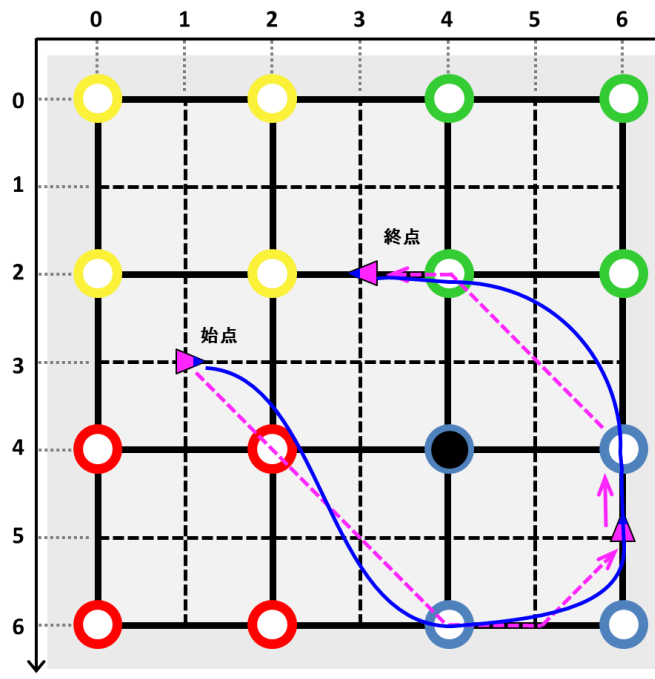
アーム角度ごとのブロックの色判定結果

アーム角度 [度]	対象ブロックの色				
	緑	黄	青	赤	黒
40	BLACK	BLACK	BLACK	BLACK	BLACK
41	BLACK	BLACK	BLACK	BLACK	BLACK
42	BLACK	BLACK	BLACK	BLACK	NONE
43	BLACK	BROWN	BLACK	RED	NONE
44	BLACK	BROWN	BLACK	RED	NONE
45	BLACK	BROWN	BLACK	RED	NONE
46	BLACK	BROWN	BLACK	RED	NONE
47	BLACK	BROWN	BLACK	RED	NONE
48	BLACK	BROWN	BLUE	RED	NONE
49	BLACK	BROWN	BLUE	RED	NONE
50	BLACK	BROWN	BLUE	RED	NONE
51	BLACK	BROWN	BLUE	RED	NONE
52	GREEN	BROWN	BLUE	RED	NONE
53	GREEN	BROWN	BLUE	RED	NONE
54	GREEN	BROWN	BLUE	RED	NONE
55	GREEN	YELLOW	BLUE	RED	NONE
56	GREEN	YELLOW	BLUE	RED	NONE
57	GREEN	YELLOW	BLUE	RED	NONE
58	GREEN	YELLOW	BLUE	RED	NONE
59	GREEN	YELLOW	BLUE	RED	NONE
60	GREEN	YELLOW	BLUE	RED	NONE
61	GREEN	YELLOW	BLUE	RED	NONE
62	GREEN	YELLOW	BLUE	RED	NONE
63	BLACK	YELLOW	BLUE	RED	NONE
64	BLACK	YELLOW	BLUE	RED	NONE
65	BLACK	YELLOW	BLUE	RED	BLACK
66	BLACK	YELLOW	BLUE	RED	BLACK
67	BLACK	YELLOW	BLUE	RED	BLACK
68	BLACK	YELLOW	BLUE	RED	NONE
69	BLACK	YELLOW	BLUE	RED	NONE
70	BLACK	BROWN	BLUE	BLACK	NONE

アーム角度58度周辺で色が確実に区別できることがわかった。  
(このとき、NONEはBLACKとみなす)

## 3-7. ゲーム

### 制御戦略



抽象マップ上で決定された移動経路は直線的な軌跡だが、これを現実の走行体の動きに変換する際には、要素技術3-3.を用いて、経路点を滑らかに結ぶ3次スプライン曲線に変換する。

これによって、滑らかに迅速に移動する。

--- 抽象マップ上の走行体の移動経路  
— 現実での走行体の移動経路