

ロボットの基礎

1610581 堀田 大地

2018 年 10 月 18 日

1 目的

バレタイズ作業のプログラムを行い，それを動作させることを通してロボットの操作方法，メカニズム，運動学，制御法などに関する理解を深める．

2 システム

2.1 ロボットアーム

本実験では三菱電機のロボット RV-2SD を用いた．RV-2SD は 6 軸多関節ロボットであり，各関節はベースから順にウエスト，ショルダ，エルボ，リストツイスト，リストピッチ，リストロールとである．各関節はハーモニックドライブ，タイミングベルトを介して AC モータにより鼓動される．

2.2 電動ハンド

ロボットアームに TAIYO 製の電動ハンド ESG1-FT-2840 を装着して対象物の把持を行う．このハンドは 2 爪の平行グリップによって把持を行い，グリップはモータとボールねじを利用して駆動している．また，ハンドには絶対位置エンコーダーではなく相対位置エンコーダが搭載されているため，ハンドはアームと異なり，電源を入れるたびに原点出しの準備動作が必要となる．

2.3 コントローラ

コントローラはロボットの制御に使用するプログラムを実行し，制御時に必要な各種計算（運動学計算や逆運動学計算）を行う装置である．本実験で使用するコントローラは TCP/IP 通信でパーソナルコンピュータに接続されており，PC との間でプログラムの送受信を行い，プログラムの選択・実行などを PC 側から行うことができる．

2.4 ティーチングボックス

ティーチングボックスは，ロボットアームに座標を教示するための装置である．

3 手順

3.1 教示

ロボットにバレタイズ作業を行わせるために，ティーチングボックス (TB) を用いて各ポジションの教示を行った．バレタイズとは，卵のパックのような容器に物体を指定した順番で，適当な位置に置く作業のことである．

3.2 教示のための準備

1. コントローラの電源を入れ，MODE 切替スイッチ”MANUAL”にした．
2. TB の”[TB ENABLE] スイッチ”を押し点灯する状態にした．
3. ”[EXE] キー”を 2 回押し，メインメニューが表示されたら，カーソルを”1. 管理・編集”に合わせ，”[EXE] キー”を押して選択した．
4. ”SAMPLE2”にカーソルを合わせ，”編集”を選択した．
5. プログラムを開いたら，”FUNCTION キー”を 2 回押してファンクションメニューを切り替え，”切替”を選択した．
6. ”<位置>モード”が開かれるので，教示したいポイントまでアームを移動させ，ファンクションメニューを 1 回切り替えて，”教示”を選択して教示を行った．

3.3 ロボットアームの動作

TB 背面の”イネーブルスイッチ”を押しながら，”[SERVO] キー”を押しサーボモータの電源を入れた．次に”[JOG] キー”を押してジョグモードに切り替え，この状態で”[ジョグ操作] キー”を操作することでロボットアームを動かした．

3.4 ハンドの動作

”[HAND] キー”を押してハンド操作モードに変更した．”原点”を原点チェックボックスにチェックが入るまで長押しして初期化を行った後，ハンド操作を行った．

3.5 教示の位置

1. ボルトを掴む位置
 - (a) TB でボルトをハンドの中心でつかめる位置までロボットハンドを動かした．
 - (b) ハンドを開いたままポジションを教示した．
 - (c) ハンドを閉じ，ボルトをつかみ，持ち上げた．
2. スイッチを押す位置
 - (a) ボルトを掴んだ状態で，ボタンスイッチを押せるような位置を決めた．

(b) ポジションを記憶させ、ボルトを上回避させた。

3. ボルトを落とす位置

(a) パレットにある 10 個の穴の位置をロボットに教えるために、4 隅の上でボルトを落とせる位置を教示した。

3.6 実行

1. 制御用プログラム”RT ToolBox2”を起動した。
2. ワークスペース”test3”を開く。
3. コントローラのモードを”AUTOMATIC”にし、オンラインモードに切り替えた。
4. オンラインのディレクトリを開いた。
5. 実行するプログラム上で右クリックし、”デバッグ状態で開く”を選択した。
6. サーボの電源を入れ、プログラム実行ボタンを選択する。

4 結果

4.1 パターン 4

図 1 のパターンのバレンタイズを実行するプログラムを作成した。作成したプログラムを示す。

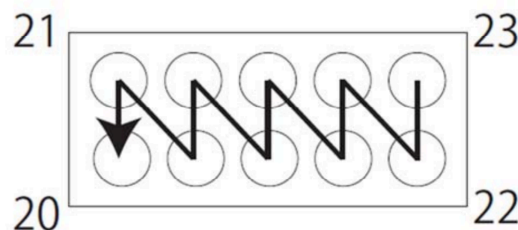


図 1 パターン 4 のバレンタイズ

ソースコード 1 パターン 4 のソースコード

- ```
1 1 Spd 20 速度を設定.
2 2 If M_EH0rg=0 Then 原点復帰未完了のとき.
```

```

3 3 EHOrg 1 ハンド 1 復帰.
4 4 Wait M_EHOrg=1 ハンド 1 原点復帰完了待ち.
5 5 EndIf If 終了.
6 6 For MJ=0 To 4 変数MJ を 0~4 まで +1 ずつ増加させながら反復.
7 7 For MI=0 To 1 変数MI を 0~4 まで +1 ずつ増加させながら反復.
8 8 PTemp=P23 変数PTemp に P23 を代入.
9 9 PTemp=PTemp - PV*MJ - PW*MI PTemp に右辺の計算結果を代入.
10 10 *LBL1 Mov P1, -30 ラベル 1:ワークをつかむ位置P1 の上空 30mm へ関節補間移動.
11 11 EHOpen 1, 50, 30 ハンド 1 を 50%の速度と 30%の力で開く.
12 12 Wait M_EHBusy=0 ハンドの完了動作を待つ.
13 13 Mvs P1 ワークをつかむ位置P1 へ直線補間移動.
14 14 EHCclose 1, 50, 30 ハンド 1 を 50%の速度と 30%の力で閉じる.
15 15 Wait M_EHBusy=0 ハンドの完了動作を待つ.
16 16 Mvs P1, -15 ワークをつかむ位置P1 の上空 15mm へ直線補間移動.
17 17 Mov P1, -70 ワークをつかむ位置P1 の上空 70mm へ関節補間移動.
18 18 Mov P3, -20 スイッチを押す位置P3 の上空 20mm へ関節補間移動.
19 19 Def Act 1, M_In(31)=1 GoTo *LBL2 優先番号 1 の割り込み処理を定義.入力信号 31 の信号
 が 1 になったら, ラベル 2 の行にジャンプする.
20 20 Act 1=1 優先番号 1 の割り込み処理.
21 21 Mvs P3 スイッチを押す位置P3 へ直線補間移動.
22 22 Act 1=0 優先番号 1 の割り込み処理を禁止.
23 23 Mvs P3, -20 スイッチを押す位置P3 の上空 20mm へ直線補間移動.
24 24 Mov PErr スイッチを押せなかったときに行く位置PErr へ関節補間移動.
25 25 EHOpen 1, 50, 30 ハンド 1 を 50%の速度と 30%の力で開く.
26 26 Wait M_EHBusy=0 ハンドの動作完了を待つ.
27 27 GoTo *LBL1 LBL1 へ GoTo.
28 28 *LBL2 Act 1 = 0 ラベル 2, 優先番号 1 の割り込み処理を禁止.
29 29 Mvs P3, -20 スイッチを押す位置P3 の上空 20mm へ直線補間移動.
30 30 Mov PTemp, -90 PTemp の上空 90mm へ関節補間移動.
31 31 Mov PTemp, -10 PTemp の上空 10mm へ関節補間移動.
32 32 EHOpen 1, 30, 30 ハンド 1 を速度 50%, 力 30%で開く.
33 33 Wait M_EHBusy=0 ハンドの動作完了を待つ.
34 34 Mov PTemp, -50 PTemp の上空 50mm へ関節補間移動.
35 35 Next MI 7行目へ戻る.
36 36 Next MJ 6行目へ戻る.
37 37 End プログラム終了.
38 PTemp=(+424.52,-31.92,+172.64,-171.43,-12.49,-94.16,+0.00,+0.00)(7,0)
39 P20=(+403.76,+138.90,+166.76,+175.18,-1.73,-96.51,+0.00,+0.00)(7,0)
40 PV=(+0.00,-48.26,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
41 PW=(+46.12,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
42 P1=(+365.96,-226.66,+156.70,-165.67,+4.82,-167.50,+0.00,+0.00)(7,0)
43 P3=(+260.04,+25.14,+248.15,+179.30,-1.25,-96.72,+0.00,+0.00)(7,0)
44 PErr=(+460.54,-133.35,+184.36,+177.63,-4.42,-134.98)(7,0)
45 P0=(+375.14,-11.15,+299.92,-179.07,-4.90,-166.26)(7,0)
46 P21=(+450.71,+141.59,+169.03,+174.36,-0.79,-84.77,+0.00,+0.00)(7,0)
47 P22=(+408.24,-45.75,+166.76,+175.19,-1.72,-96.51,+0.00,+0.00)(7,0)
48 P23=(+450.46,-47.81,+169.03,+174.36,-0.80,-84.82,+0.00,+0.00)(7,0)

```

## 4.2 パターン 6

図 2 のパターンのバレンタイズを実行するプログラムを作成した。作成したプログラムを示す。

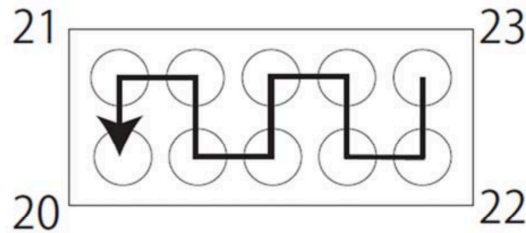


図 2 パターン 6 のバレンタイズ

### ソースコード 2 パターン 4 のソースコード

```
1 1 Spd 20 速度を設定.
2 2 If M_EHOrg=0 Then 原点復帰未完了のとき.
3 3 EHOrg 1 ハンド 1 復帰.
4 4 Wait M_EHOrg=1 ハンド 1 原点復帰完了待ち.
5 5 EndIf If 終了.
6 6 MI = 0 MI = 0 を代入.
7 7 MJ = 0 MJ = 0 を代入.
8 8 For a=0 To 9 変数 a を 0~9 まで +1 ずつ増加.
9 9 PTemp=P23 変数 PTemp に P23 を代入.
10 10 PTemp=PTemp - PV*MJ - PW*MI PTemp に右辺の計算結果を代入.
11 11 *LBL1 Mov P1, -30 ラベル 1:ワークをつかむ位置 P1 の上空 30mm へ関節補間移動.
12 12 EHOpen 1, 50, 30 ハンド 1 を 50% の速度と 30% の力で開く.
13 13 Wait M_EHBusy=0 ハンドの完了動作を待つ.
14 14 Mvs P1 ワークをつかむ位置 P1 へ直線補間移動.
15 15 EHCclose 1, 50, 30 ハンド 1 を 50% の速度と 30% の力で閉じる.
16 16 Wait M_EHBusy=0 ハンドの完了動作を待つ.
17 17 Mvs P1, -15 ワークをつかむ位置 P1 の上空 15mm へ直線補間移動.
18 18 Mov P1, -70 ワークをつかむ位置 P1 の上空 70mm へ関節補間移動.
19 19 Mov P3, -20 スイッチを押す位置 P3 の上空 20mm へ関節補間移動.
20 20 Def Act 1, M_In(31)=1 GoTo *LBL2 優先番号 1 の割り込み処理を定義. 入力信号 31 の信号
 が 1 になったら, ラベル 2 の行にジャンプする.
21 21 Act 1=1 優先番号 1 の割り込み処理.
22 22 Mvs P3 スイッチを押す位置 P3 へ直線補間移動.
23 23 Act 1=0 優先番号 1 の割り込み処理を禁止.
24 24 Mvs P3, -20 スイッチを押す位置 P3 の上空 20mm へ直線補間移動.
25 25 Mov PErr スイッチを押せなかったときに行く位置 PErr へ関節補間移動.
26 26 EHOpen 1, 50, 30 ハンド 1 を 50% の速度と 30% の力で開く.
27 27 Wait M_EHBusy=0 ハンドの動作完了を待つ.
28 28 GoTo *LBL1 LBL1 へ GoTo.
```

```

29 29 *LBL2 Act 1 = 0 ラベル 2, 優先番号 1 の割り込み処理を禁止.
30 30 Mvs P3, -20 スイッチを押す位置 P3 の上空 20mm へ直線補間移動.
31 31 Mov PTemp, -90 PTemp の上空 90mm へ関節補間移動.
32 32 Mov PTemp, -10 PTemp の上空 10mm へ関節補間移動.
33 33 EHOpen 1, 30, 30 ハンド 1 を速度 50%, 力 30% で開く.
34 34 Wait M_EHBusy=0 ハンドの動作完了を待つ.
35 35 Mov PTemp, -50 PTemp の上空 50mm へ関節補間移動.
36 36 MK = MI + MJ MJ に MI+MJ を代入.
37 37 M = MK Mod 2 M に MK を 2 で割った余りを代入.
38 38 IF M == 0 Then M=0 のとき.
39 39 IF MI == 1 Then MI=1 のとき.
40 40 MI = MI - 1 MI から 1 を引く.
41 41 Else MI=1 じゃないとき.
42 42 MI = MI + 1 MI に 1 を足す.
43 43 Endif If 終了.
44 44 Else M=0 じゃないとき.
45 45 MJ = MJ + 1 MJ に 1 を足す.
46 46 Endif if 終了.
47 47 Next a 8 に戻って a に 1 を足す.
48 48 End
49 PTemp=(+424.52,-31.92,+172.64,-171.43,-12.49,-94.16,+0.00,+0.00)(7,0)
50 P20=(+403.76,+138.90,+166.76,+175.18,-1.73,-96.51,+0.00,+0.00)(7,0)
51 PV=(+0.00,-48.26,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
52 PW=(+46.12,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00,+0.00)(0,0)
53 P1=(+365.96,-226.66,+156.70,-165.67,+4.82,-167.50,+0.00,+0.00)(7,0)
54 P3=(+260.04,+25.14,+248.15,+179.30,-1.25,-96.72,+0.00,+0.00)(7,0)
55 PErr=(+460.54,-133.35,+184.36,+177.63,-4.42,-134.98)(7,0)
56 P0=(+375.14,-11.15,+299.92,-179.07,-4.90,-166.26)(7,0)
57 P21=(+450.71,+141.59,+169.03,+174.36,-0.79,-84.77,+0.00,+0.00)(7,0)
58 P22=(+408.24,-45.75,+166.76,+175.19,-1.72,-96.51,+0.00,+0.00)(7,0)
59 P23=(+450.46,-47.81,+169.03,+174.36,-0.80,-84.82,+0.00,+0.00)(7,0)

```

---

## 5 課題

### 5.1 必須課題 1

順運動学では各関節状態からロボットの手先の位置・姿勢が一意な解として求まるのに対し、逆運動学では目的とするロボットの手先位置・姿勢から各関節の状態が一意に求まらないことを説明する。

まず、順運動学について図 3 の解を求める。

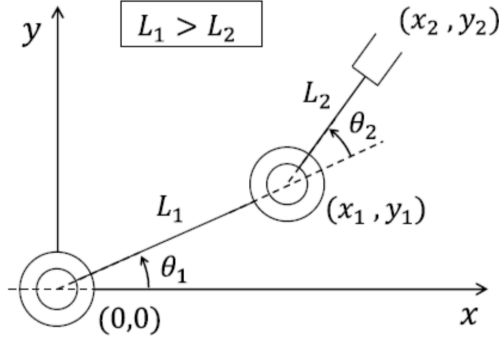


図3 パターン4のバレンタイズ

Eq.1, 2 より  $\theta_1$  と  $\theta_2$  が求まれば解が一つに定まる. なので, 順運動学では各関節の状態からロボットの手先の位置・姿勢が一意な解として求まる.

$$(x_1, y_1) = (L_1 \cos \theta_1, L_1 \sin \theta_1). \quad (1)$$

$$(x_2, y_2) = (L_1 \cos \theta_1 + L_2 \cos (\theta_1 + \theta_2), L_1 \sin \theta_1 + L_2 \sin (\theta_1 + \theta_2)). \quad (2)$$

次に, 逆運動学についての解を求める.Eq.3, 4 として, Eq.2 の各成分の二乗の和を取ると Eq.5 になる. よって, Eq.6 となる.

$$x_2 = x. \quad (3)$$

$$y_2 = y. \quad (4)$$

$$x^2 + y^2 = L_1^2 + L_2^2 + 2L_1L_2 \cos \theta_2. \quad (5)$$

$$\theta_2 = \cos^{-1} \left[ \frac{(x^2 + y^2) - (L_1^2 + L_2^2)}{2L_1L_2} \right]. \quad (6)$$

さらに, Eq.2 について, Eq.7 のように変形した. また, Eq.7 の両辺に

$$\begin{bmatrix} L_1 + L_2 \cos \theta_2 & -L_2 \sin \theta_2 \\ L_2 \sin \theta_2 & L_1 + L_2 \cos \theta_2 \end{bmatrix}^{-1}$$

を左からかけると, Eq.8 となった. これより, Eq.9, 10 が導出される. つまり, Eq.11 より,  $x, y$  の値が求まっても解が二つ出てくることより, 解が一意に定まらない. よって, 逆運動学では目的とするロボットの手先位置・姿勢から各関節の状態が一意に求まらない.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} L_1 + L_2 \cos \theta_2 & -L_2 \sin \theta_2 \\ L_2 \sin \theta_2 & L_1 + L_2 \cos \theta_2 \end{bmatrix} \begin{bmatrix} \cos \theta_1 \\ \sin \theta_1 \end{bmatrix}. \quad (7)$$

$$\begin{bmatrix} \cos \theta_1 \\ \sin \theta_1 \end{bmatrix} = \begin{bmatrix} L_1 + L_2 \cos \theta_2 & -L_2 \sin \theta_2 \\ L_2 \sin \theta_2 & L_1 + L_2 \cos \theta_2 \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \end{bmatrix}. \quad (8)$$

$$\begin{aligned} \tan \theta_1 &= \frac{L_1 + L_2 \cos \theta_2 y - (L_2 \sin \theta_2 x)}{L_1 + L_2 \cos \theta_2 y + (L_2 \sin \theta_2 x)} \\ &= \frac{\frac{y}{x} - \frac{L_2 \sin \theta_2}{L_1 + L_2 \cos \theta_2}}{1 + \frac{y}{x} \cdot \frac{L_2 \sin \theta_2}{L_1 + L_2 \cos \theta_2}}. \end{aligned} \quad (9)$$

$$\theta_1 = \tan^{-1} \left( \frac{y}{x} \right) - \tan^{-1} \left( \frac{\sqrt{4L_1^2L_2^2 - (x^2 + y^2) - (L_1^2 + L_2^2)^2}}{2L_1^2 + (x^2 + y^2) - (L_1^2 + K_2^2)} \right). \quad (10)$$

## 5.2 必須課題 2

直線補間動作の時，ロボットが実行可能な経路を生成できなく例を考える．図 7 のロボットアームにおいて，原点を中心とした半径  $L_1 - L_2$  の円の領域には長さ  $L_2$  の腕は入ることはできない．なぜなら，本体にぶつかるからだ．よって，実行不可能な経路の例として，図の座標  $(x_2, y_2)$  を始点にして，終点を  $(-x_2, y_2)$  にした場合の経路である．

## 参考文献

- [1] ロボットの基礎 授業テキスト