

1 実行方法

1.1 動作環境

Linux のディストリビューションのひとつである Ubuntu で動作を確認している。Ubuntu で動作させるための必要なアプリケーションは以下の通りである。

```
1 sudo apt-get install build-essential
2 sudo apt-get install cmake
3 sudo apt-get install gnuplot
4 sudo apt-get install meshlab
```

コンパイル時に必要なビルドシステムは build-essential と cmake が必要である。そして各関節の座標系とアームの姿勢を確認するために gnuplot を用いており、またアームの stl のデータ確認のため meshlab というアプリケーションを用いている。

Visual Studio でのコンパイルも可能であるが、その場合 Struct.h の最初の記述を変更する。

```
4 #define _CRT_SECURE_NO_DEPRECATED
5 #include <stdio.h>
6 #include <stdlib.h>
7 #define _USE_MATH_DEFINES
8 #include <math.h>
```

1.2 ファイル構造

この zip 中のファイル構造を示す。

```
1 .
2 |-- CMakeLists.txt
3 |-- Struct.h
4 |-- document
5 |   |-- document.pdf
6 |-- kadai.h
7 |-- kadai1A.cpp
8 |-- kadai1A.h
9 |-- kadai2A.cpp
10 |-- kadai2A.h
11 |-- kadai2B.cpp
12 |-- kadai2B.h
13 |-- main.cpp
14 |-- plot
15 |   |-- Arm.stl
16 |   |-- arm.dat
17 |   |-- plot.plt
18 |   |-- tetra_point.dat
19 |   |-- x.dat
20 |   |-- y.dat
21 |   |-- z.dat
```

tetra_point.dat が三角柱を作成するための初期位置が記されたデータであり、Arm.stl が課題 3 の結果である。その他のデータは gnuplot で姿勢を確認するために生成されたファイルであり、プログラムを実行すると生成される。このデータの可視化方法を下に示す。

```

1 cd plot
2 gnuplot
3 load "plot.plt"

```

1.3 コンパイル方法と実行方法

```

1 mkdir build
2 cd build
3 cmake ..
4 make
5 ./実行ファイル名

```

2 実装した関数の説明

```
void ForwardKinematics(double joint[], double link[]);
```

- ロボットアームの順運動学計算 (各関節角度とリンクのパラメータ指定)
- 具体的な計算は次の節で示す

```
void ShowTfAxis(double joint[], double link[])
```

- ロボットの各軸の座標を求める
- 各軸での座標系を作成
- 外部ファイルに保存

```
void ArmOutputdat(double P[][VEC_SIZE]);
```

- 求めたロボットの各軸の座標を用いてリンク自体の骨格モデルを出力

```
void TfAxisOutputdat(double P[][VEC_SIZE], FILE *fpx, FILE *fpy, FILE *fpz);
```

- 各軸での座標系を可視化するためのファイル作成補助

```
void OutputPlt();
```

- リンクのモデルを出力
- Gnuplot で可視化

```
void OutputSTL(double part1[][VEC_SIZE], double part2[][VEC_SIZE], double part3[][VEC_SIZE]);
```

- 各リンクのパーツを STL 形式で出力

```
void LinkCreator(double joint[], double link[]);
```

- 与えられた関節角度とリンク長からアームの姿勢を求め、STL で出力

```
void set_arm(double Origin[][VEC_SIZE], double P[][VEC_SIZE], double h);
```

- リンクの初期姿勢を定義
- STL を作成するための姿勢を準備

3 計算方法

基準座標系 (X_0, Y_0, Z_0) におけるハンド座標系 (X_5, Y_5, Z_5) を求める式を示す．ただし以下に示す行列は並進移動と回転移動を同時に行っている．1 リンク目から順に変換行列を示すと

$$T_0 = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & a \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$T_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta_2 & -\sin \theta_2 & 0 \\ 0 & \sin \theta_2 & \cos \theta_2 & b \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$T_2 = \begin{bmatrix} \cos \theta_3 & 0 & \sin \theta_3 & 0 \\ 0 & 1 & 0 & c \\ \sin \theta_3 & 0 & \cos \theta_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$T_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta_4 & -\sin \theta_4 & 0 \\ 0 & \sin \theta_4 & \cos \theta_4 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$T_4 = \begin{bmatrix} \cos \theta_5 & 0 & \sin \theta_5 & 0 \\ 0 & 1 & 0 & e \\ \sin \theta_5 & 0 & \cos \theta_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$T_5 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & f \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

変換行列中に含まれている a, b, c, d, e, f は $L_1 = a + b, L_3 = c + d, L_5 = e + f$ が成り立ち、リンクと同じ方向軸で回転するリンクの位置を任意の場所にできるようプログラム上ではそれぞれの別のリンクと考え変数としている．式 (1) によって基準座標系 (X_0, Y_0, Z_0) における点 P_0 からハンドの先端の座標 P_5 を求めると以下の式で表される．

$$P_5 = T_0 T_1 T_2 T_3 T_4 T_5 P_0 \quad (7)$$

4 結果

それぞれの課題で結果を示す.

4.1 課題 1

課題 1 は上の節で式を示した.

4.2 課題 2

課題 2 は gnuplot を用いて各関節の角度を指定したときのロボットアームの姿勢と各軸での座標系を表示することで確認を行った. この結果を Fig. 1 に示す. Fig. 1 は課題 3 で指定された関節角度を指定した.

4.3 課題 3

指定された角度にしたときのロボットアームの姿勢を STL に出力したときの図を Fig. 2 から fig:ロボットアームの姿勢 4 に示す.

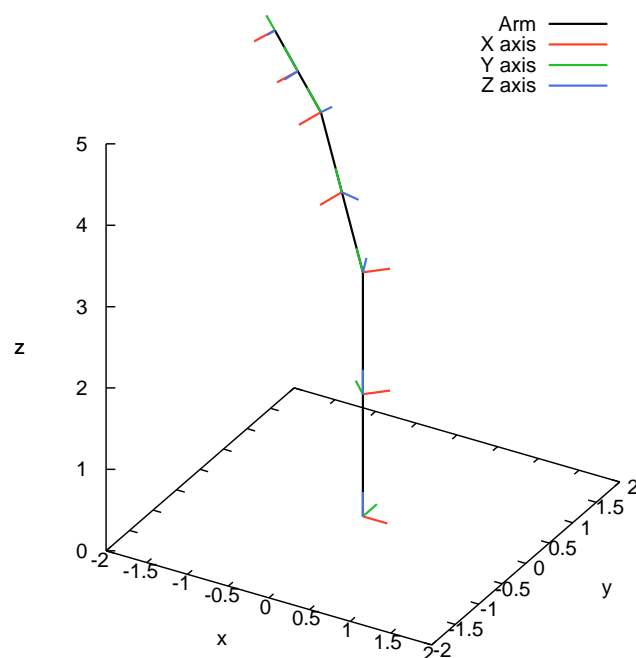


Fig.1 ロボットの姿勢と途中の関節の動作を表す座標系

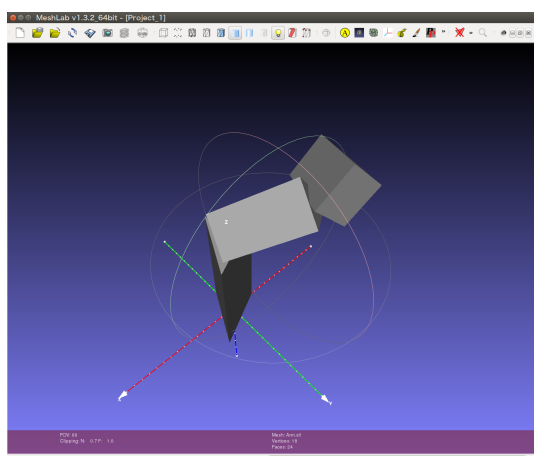


Fig.2 ロボットアームの姿勢 1

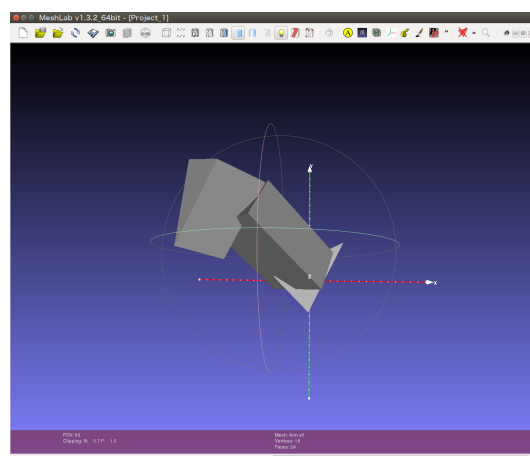


Fig.3 ロボットアームの姿勢 2

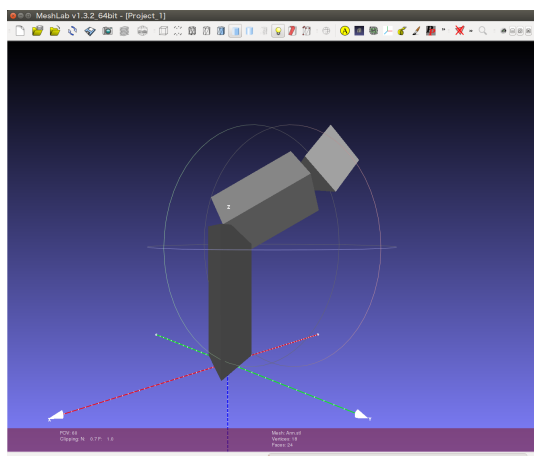


Fig.4 ロボットアームの姿勢 3

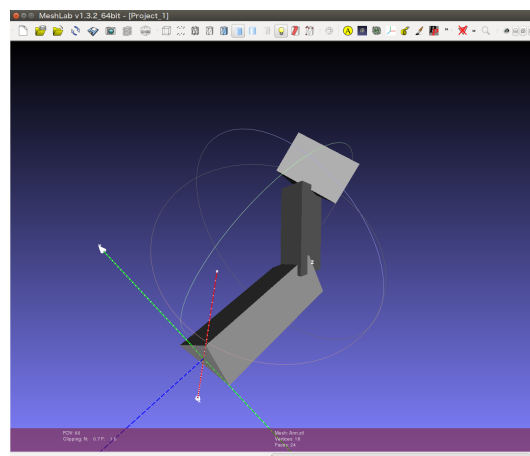


Fig.5 ロボットアームの姿勢 4