

遠隔操作ロボットシステムの構築に関する研究

2020SC002 安立周生 2020SC054 永田裕武

2020SC060 新田陽斗

指導教員：中島明

1 はじめに

近年、日本では少子高齢化により労働人口が年々減少している。それに伴い多くの仕事に様々な機械が導入され仕事の変化が起こっている。機械ができることは機械に任せるという考え方により産業用ロボットへの需要がさらに高まっている。しかし、現在多くを人の手によって支えている介護現場での人手不足が深刻化してきている。また、コロナウイルスの流行時や東日本大震災での放射能処理など人の立ち入り自体が制限、禁止される中での作業が必要となっている。

そのため、可能な限り無駄な工程を減らし、人員を減らすことができる遠隔操作ロボットへの期待が高まっている。また、工業用ロボットだけでなく、一般家庭での機械と人との共存が当たり前になりつつある環境の中で、工場などにある産業用ロボットとは違い、イレギュラーが起こりうる一般家庭においても安全に正確な動作を実現できる機械というのが重要視されている。

本研究では力覚センサを搭載した6自由度マニピュレータとハプティクスデバイスタッチを用いてマスタースレーブ制御を行い、遠く離れた場所からマニピュレータを操作すること、スレーブ側が障害物に接触した際にマスタ側でも反力を受けているようにふるまうことを最終的な研究目標としている。

2 マスタースレーブについて

マスタースレーブとは複数の機器が連携して動作する際に制御する側(マスタ)の情報から一方的に制御される側(スレーブ)にコマンドを送信しマスタ側とスレーブ側が同一の動きを行うことができるような制御のことである。マスタースレーブを使用することで離れた位置からスレーブ機を操作することができるメリットがある一方、マスタ機とスレーブ機、PCを連携させて通信を行うため動作の遅延が起こってしまうというデメリットも存在する。

3 スレーブアームの概要

本研究では、スレーブ機に6自由度マニピュレータであるHEBI製のX-Seriesアームキットを使用している。X-Seriesアームキットはアクチュエータ、減速機、エンコーダに加え、トルク、位置そして速度制御を可能にするセンサ、3軸の慣性センサを内蔵しているロボットモジュールである[1]。PCから受け取った各関節の角度の情報をそれぞれのアクチュエータが受け取りマスタ機と同様の姿勢をとることができるように制御を行うことができる。また、スレーブ機側で障害物に接触することで発生す

る反力を各種センサで検出しその情報をPCに送るという役割もある。これらの制御はMATLABを用いて行う。実際の6自由度マニピュレータの図を図1にマニピュレータの全体モデルを以下の図2に示す。また実験をする上で各モーターの名称を図の様に設定した。各関節の動作範囲は $-180[\text{deg}] \sim 180[\text{deg}]$ とする。各軸のモータと2軸から3軸、3軸から4軸を繋ぐリンクの重量は以下の表1に示す。



図1 実際の6自由度マニピュレータ

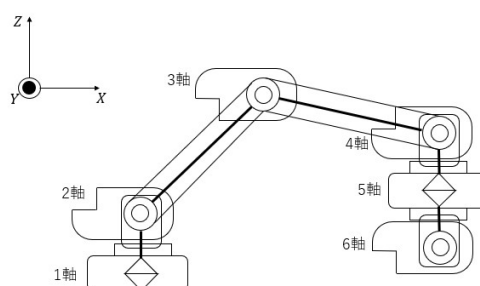


図2 マニピュレータの全体モデル

表1 モータとリンクの重量

軸, リンク	重量
1 軸	480[g]
2 軸	500[g]
3 軸	480[g]
4 軸	315[g]
5 軸	315[g]
6 軸	315[g]
リンク	250[g]

次にスレーブ機の重心周りの慣性モーメントを計算する。今回の実験においては6関節全ての制御を行うのが難しい進捗状況であった為、第2関節、第3関節の2関節の

みの制御し実験を行った。また重心は図2に示すスレーブ機全体の基準軸の内、X軸、Z軸方面における平面上で定義していく。以下の図3に第1リンクと3軸の重心の関係、図4に第2リンクと4~6軸の重心の関係を示す。

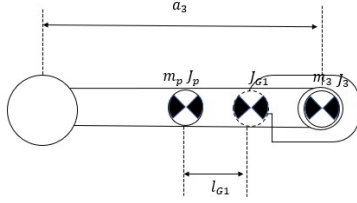


図3 第1リンクと3軸の重心の関係

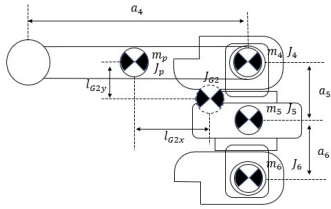


図4 第2リンクと4~6軸の重心の関係

各軸のモータはそれぞれ図5の様に直方体に近似して計算して行く。

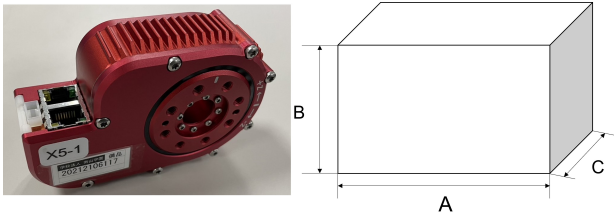


図5 各軸のモータの直方体近似

まず第1リンクと3軸の重心周りの慣性モーメントを求める。リンクを一樣な棒と近似して慣性モーメント J_p を導出すると、

$$J_p = \frac{1}{12} m_p a_3^2 \quad (1)$$

となる。モータは直方体を近似して、慣性モーメント J_3 を導出すると、

$$J_3 = \frac{1}{3} m_3 (A^2 + B^2) \quad (2)$$

となる。第1リンクと3軸の重心を求めると、

$$l_G = \frac{m_2 a_3}{m_p + m_3} \quad (3)$$

となる。平行軸の定理より、重心周りの慣性モーメント J'_p, J'_3 はそれぞれ

$$J'_p = J_p + m_p l_{G1}^2 \quad (4)$$

$$J'_3 = J_3 + m_3 (a_3 - l_{G1})^2 \quad (5)$$

となる。よって第1リンクと3軸の重心周りの慣性モーメント J_{G1} は、

$$J_{G1} = J'_p + J'_3 \quad (6)$$

となる。

次に第2リンクと4~6軸の重心周りの慣性モーメントを求める。同様に第2リンク周りの慣性モーメントを求めると、

$$J_p = \frac{1}{12} m_p a_4^2 \quad (7)$$

となる。 J_5 のみモータの向きが異なる事に注意して慣性モーメントを導出すると、

$$J_4 = \frac{1}{3} m_4 (A^2 + B^2) \quad (8)$$

$$J_5 = \frac{1}{3} m_5 (A^2 + C^2) \quad (9)$$

$$J_6 = \frac{1}{3} m_6 (A^2 + B^2) \quad (10)$$

となる。第2リンクと4~6軸の重心を求めると、

$$l_{G2x} = \frac{m_4 a_4 + m_5 a_4 + m_6 a_4}{m_p + m_4 + m_5 + m_6} \quad (11)$$

$$l_{G2y} = \frac{m_5 a_5 + m_6 (a_5 + a_6)}{m_p + m_4 + m_5 + m_6} \quad (12)$$

となる。平行軸の定理より、重心周りの慣性モーメントは J'_p, J'_4, J'_5, J'_6 はそれぞれ、

$$J'_p = J_p + m_p l_{G2x}^2 \quad (13)$$

$$J'_4 = J_4 + m_4 (a_4 - l_{G2x})^2 \quad (14)$$

$$J'_5 = J_5 + m_5 \{ (a_4 - l_{G2x})^2 + (a_5 - l_{G2y})^2 \} \quad (15)$$

$$J'_6 = J_6 + m_6 \{ (a_4 - l_{G2x})^2 + (a_5 + a_6 - l_{G2y})^2 \} \quad (16)$$

となる。よって第2リンクと4~6軸の重心周りの慣性モーメント J_{G2} は、

$$J_{G2} = J'_p + J'_4 + J'_5 + J'_6 \quad (17)$$

となる。

4 マスターアームの概要

本研究ではマスター機として、3D Systems 製のハプティクスデバイスである GeomagicTouch™(以下、Touch と呼ぶ。)を使用している [2]。図 6 に示すように Touch は 6 自由度の機構をしている。台座部分にはインクウェルがあり、そこにペン先を格納した時の姿勢が Touch の基本姿勢となる。各関節には絶対値エンコーダが搭載されているため Touch の関節角などが取得可能となっている。6 軸が搭載されている部分が持ち手となるが、持ち手部分にはボタンが 2 つ付いているため、持ち手を操作している最中にボタン入力をする事も可能である。Touch の各軸の動作範囲を表 2 に示す。

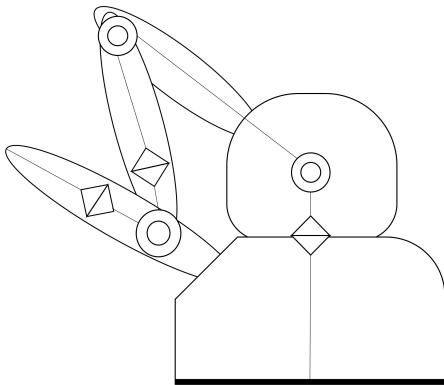


図 6 Touch の機構

表 2 Touch の動作範囲

軸	動作範囲
1 軸	約 $-57 \sim$ 約 $+55$ [deg]
2 軸	約 $-12 \sim$ 約 $+90$ [deg]
3 軸	約 $-67 \sim$ 約 $+48$ [deg]
4 軸	約 $-142 \sim$ 約 $+145$ [deg]
5 軸	約 $-150 \sim$ 約 -7 [deg]
6 軸	約 $-149 \sim$ 約 $+149$ [deg]

Touch に関するプログラムは OpenHaptics Toolkit というハプティクスデバイス専用のコマンドを利用し、Microsoft 社が提供している統合開発環境である Visual Studio で作成した。

5 実験環境

実験環境としては、以下の図 7、図 8 のようにハプティクスデバイスタッチと PC、HEBI Robotics アクチュエータを連携させる。マスター機であるハプティクスデバイスタッチからマスター機の手先位置の座標を PC を通じてスレーブ機である HEBI Robotics に送信する。UDP 通信を用いてマスター機から送信された座標位置をもとに PC 内の MATLAB で逆運動学を用いて各関節の角度を導出する。また、PID 制御と重力補償、加速度補償により制御対象であるスレーブ機をフィードバック制御する。

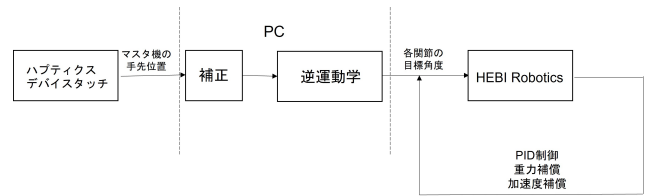


図 7 マスタースレーブの構造

スレーブ機が障害物に接触するとその反力を静力学関係式からヤコビ行列を用いて関節駆動力に変換し PC を通じてマスター側に送信する。マスター機とスレーブ機の大きさに差異があるため関節駆動力を調整するように制御を行う。

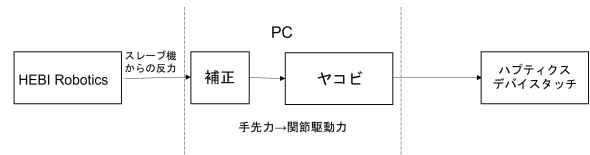


図 8 反力を受け取った際の構造

6 モデリング

スレーブアームからマスターアームを通信する上で順運動学が必要となる。マニピュレータのフレーム配置を図 9 の様に定義する。また図 10 にマニピュレータの各関節を 0 [deg] にした状況でのマニピュレータの全体のモデルと基準座標を示す。更に各座標系間の距離を $a_1, a_2, a_3, a_4, a_5, a_6$ と置き図 10 に示し、具体的な値は表 3 に示す。

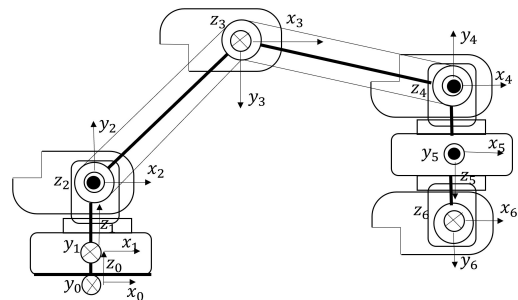


図 9 マニピュレータのフレーム配置

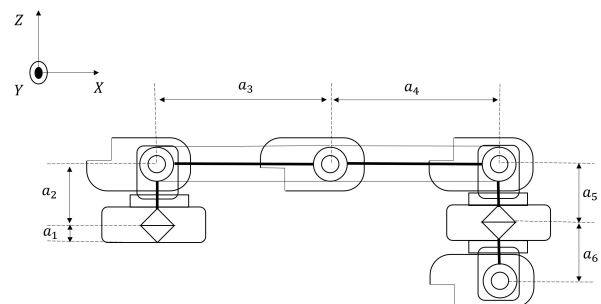


図 10 マニピュレータの全体のモデルと基準座標

表3 マニピュレータの各座標間距離

文字	座標間距離
a_1	5[cm]
a_2	9[cm]
a_3	33[cm]
a_4	33[cm]
a_5	9[cm]
a_6	9[cm]

図9に示す6自由度マニピュレータを表す座標系を用いて順運動学 [3] についての説明を行う。まず初めに関節構造に着目しリンク座標系 Σ_1, Σ_2 を定義する。今回の卒業研究においては、6自由度マニピュレータの運動学を計算する事が厳しい進捗状況であったため、本来の実機である6自由度マニピュレータを図11に示す様に2自由度マニピュレータに近似し、2軸、3軸のみの運動学を計算する。

6.1 順運動学

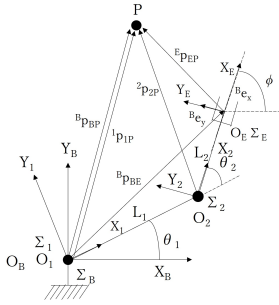


図11 2DOF マニピュレータの順運動学

次に表3のリンクの長さ $a_i (i = 3, 4)$ を図10の $L_i (i = 1, 2)$ と関節変数 $\theta_i (i = 1, 2)$ を用いて隣り合う座標系ごとの位置と姿勢の関係を求める。図11に示すように Σ_1, Σ_2 を定義すると、

$$\begin{cases} {}^B p_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ {}^B R_1 = \begin{bmatrix} C_1 & -S_1 \\ S_1 & C_1 \end{bmatrix} \end{cases}$$

$$\begin{cases} {}^1 p_{12} = \begin{bmatrix} L_1 \\ 0 \end{bmatrix} \\ {}^1 R_2 = \begin{bmatrix} C_2 & -S_2 \\ S_2 & C_2 \end{bmatrix} \end{cases}$$

$$\begin{cases} {}^2 p_{2E} = \begin{bmatrix} L_2 \\ 0 \end{bmatrix} \\ {}^2 R_E = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{cases}$$

となる。ただし C_i, S_i はそれぞれ $\sin\theta_i, \cos\theta_i$ のことを示す。これらを用いてリンク間の同次変換行列は次のように得られる。

$${}^B T_1 = \begin{bmatrix} C_1 & -S_1 & 0 \\ S_1 & C_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (18)$$

$${}^1 T_2 = \begin{bmatrix} C_2 & -S_2 & L_1 \\ S_2 & C_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (19)$$

$${}^2 T_E = \begin{bmatrix} 1 & 0 & L_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (20)$$

${}^B T_E$ を求めるために隣り合うリンクに対して同次変換を行うと、

$$\begin{aligned} {}^B T_E &= {}^B T_1 {}^1 T_2 {}^2 T_E \\ &= \begin{bmatrix} C_1 & -S_1 & 0 \\ S_1 & C_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_2 & -S_2 & L_1 \\ S_2 & C_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & L_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} C_1 C_2 - S_1 S_2 & -C_1 S_2 - S_1 C_2 & L_2(C_1 C_2 - S_1 S_2) + L_1 C_1 \\ S_1 C_2 + C_1 S_2 & -S_1 S_2 + C_1 C_2 & L_2(S_1 C_2 + C_1 S_2) + L_1 S_1 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} C_{12} & -S_{12} & L_1 C_1 + L_2 C_{12} \\ S_{12} & C_{12} & L_1 S_1 + L_2 S_{12} \\ 0 & 0 & 1 \end{bmatrix} \quad (21) \end{aligned}$$

を得る。ただし、 $C_{12} = \cos(\theta_1 + \theta_2)$ 、 $S_{12} = \sin(\theta_1 + \theta_2)$ のことを示す。

順運動学によってリンクの長さや関節角から手先位置を一つに定めることができる。

6.2 逆運動学

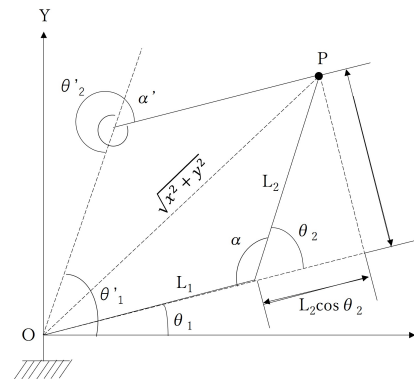


図12 逆運動学

順運動学と同様に2自由度マニピュレータの逆運動学の説明を行う。逆運動学 [3] では座標の目標値が設定されており、リンクの長さや目標値を用いて関節角を求めるものである。2自由度マニピュレータの手先位置 ${}^0 p^3$ を以下のように定義する。

$${}^0p^3 = \begin{bmatrix} x \\ y \end{bmatrix} \quad (22)$$

このとき θ_1 と θ_2 は図 12 より以下ようになる。

$$\theta_2 = \pi - \alpha \quad (23)$$

$$\theta_1 = \text{atan2}(y, x) - \text{atan2}(L_2 \sin \theta_2, L_1 + L_2 \cos \theta_2) \quad (24)$$

ただし, α は以下の式より求めることができる。

$$\alpha = \pm \text{atan2}(\sqrt{1 - \cos^2 \alpha}, \cos \alpha) \quad (25)$$

$$\cos \alpha = \frac{-(x^2 + y^2) + L_1^2 + L_2^2}{2L_1L_2} \quad (26)$$

(25) 式のように α の解は 2 つあるため逆運動学の解は必ずしも一意に定まらず複数解存在する。

6.3 ヤコビ行列

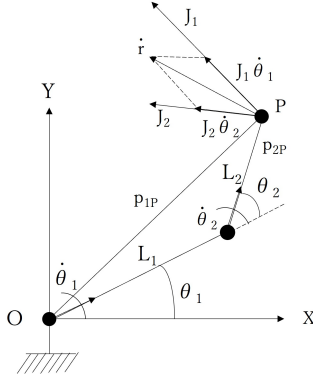


図 13 ヤコビ行列

ヤコビ行列 [3] は HEBI Robotics が障害物に接触した際に発生した反力を関節駆動力に変換する際に必要となるものであり, 図 13 において運動方程式を導出すると,

$$x = L_1 C_1 + L_2 C_{12} \quad (27)$$

$$y = L_1 S_1 + L_2 S_{12} \quad (28)$$

となる。ヤコビ行列より,

$$\frac{\partial x}{\partial \theta_1} = -L_1 S_1 - L_2 S_{12} \quad (29)$$

$$\frac{\partial x}{\partial \theta_2} = -L_2 S_{12} \quad (30)$$

$$\frac{\partial y}{\partial \theta_1} = L_1 C_1 + L_2 C_{12} \quad (31)$$

$$\frac{\partial y}{\partial \theta_2} = L_2 C_{12} \quad (32)$$

となる。よって,

$$J = \begin{bmatrix} -L_1 S_1 - L_2 S_{12} & -L_2 S_{12} \\ L_1 C_1 + L_2 C_{12} & L_2 C_{12} \end{bmatrix} \quad (33)$$

を得る。

7 Touch の関節角を用いた PID 制御 + 重力補償のシミュレーション

実験を行っていく上で, 必要となる第 1 リンクと 3 軸の重心周りの慣性モーメント J_{G1} と, 第 2 リンクと 4~6 軸の重心周りの慣性モーメント J_{G2} はスレーブアームの概要にて導出した式に各物理パラメータを代入する事で以下の表 4 のようになる。

表 4 重心周りの慣性モーメント

文字	慣性モーメント
J_{G1}	$7.44 \times 10^{-3} [kg \cdot m^2]$
J_{G2}	$14.93 \times 10^{-3} [kg \cdot m^2]$

以降は, 近似した 2 自由度マニピュレータの物理パラメータを用いて, MATLAB 上でモデルを作成し, シミュレーションした結果を示す。この際, UDP 通信で Touch の 2 軸と 3 軸の関節角を受け取り, その関節角を目標値として PID 制御と重力補償を行った [4]。

まず, 2 自由度マニピュレータの運動方程式を解くことで関節角の加速度を計算する関数を作成し, その関数を用いて数値積分により関節角 q と関節角の速度 \dot{q} を求めるシミュレータを作成するとブロック線図は次のようになる。

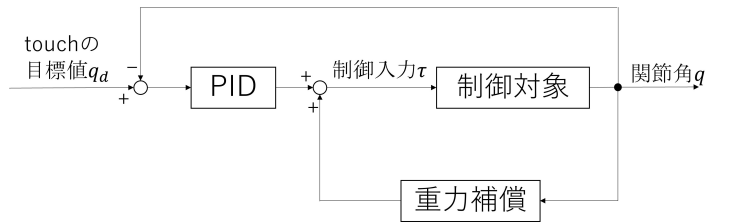


図 14 PID 制御 + 重力補償を行うブロック線図

PID 制御 + 重力補償の制御入力を τ とし, UDP 通信で受け取る Touch の関節角を q_d , 関節角のシミュレーション結果を q , 関節角の速度を \dot{q} , P 制御のゲインを K_P , D 制御のゲインを K_D , 重力補償をする際に用いる重力項を $g(\theta)$ とすると, τ は以下の式となる。

$$\tau = -K_P(q - q_d) - K_I \times \int_0^t (q - q_d) dx - K_D \times \dot{q} + g(\theta) \quad (34)$$

ただし、 $g(\theta)$ は以下ようになる。

$$g(\theta) = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}$$

$$g_1 = m_1 g L_{C1} C_1 + m_2 g (L_1 C_1 + L_{C2} C_{12})$$

$$g_2 = m_2 g L_{C2} C_{12}$$

この $g(\theta)$ はラグランジュの運動方程式を解くことで導出した。2 自由度マニピュレータの 2 つの関節角をそれぞれ q_1 , q_2 とし、Touch の 2 軸と 3 軸は図 15 に示す初期姿勢をとった。また、 K_P ゲインを $8[\text{Nm/rad}]$, K_I ゲインを $8[\text{Nm/rad}]$, K_D ゲインを $3[\text{Nm/rad}]$ とした時、関節角に対する PID 制御と重力補償を加えたシミュレーションの実行結果は図 16 となった。

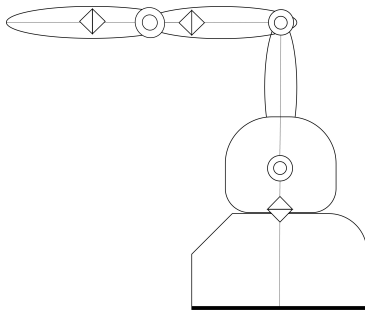


図 15 Touch の初期姿勢

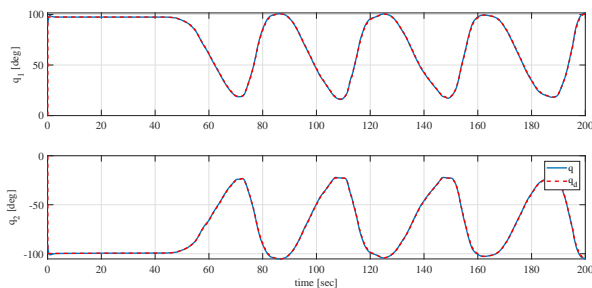


図 16 関節角のシミュレーション

図 16 において青の実線が 2 自由度マニピュレータ（スレーブアーム）の関節角，赤の破線が Touch(マスターアーム) の 2 軸と 3 軸の関節角を表している。シミュレーション上の 2 自由度マニピュレータが Touch の関節角に追従しているように見えるが、シミュレーションを開始した 0 秒の時点で、Touch の 2 軸は 0 度から 100 度、3 軸は 0 度から -100 度に急に立ち上がっている。なぜこのように急激に立ち上がるのかは現時点では解決できなかったため、今後の課題として研究を進める。

8 最後に

本研究では実機である 6 自由度マニピュレータを 2 自由度マニピュレータとして近似し、物理パラメータの導出を

行い、そのパラメータを用いてシミュレーションをした。今後の目標は大きく分けて 2 つ挙げられる。

1 つ目の目標は、Touch の関節角の情報を Visual Studio で読み取り、UDP 通信を用いて Simulink を使用せず、Matlab 上で情報を受け取ることである。第 7 章で行ったシミュレーションのように、Simulink を用いた場合は Touch の関節角の情報をリアルタイムで Simulink 上に送り、シミュレーションを行うことができる。しかし、Simulink を用いずに Matlab だけで UDP 通信を行う方法は現時点では解決できていない。Simulink を用いずに UDP 通信を行い、リアルタイムで Touch の関節角の情報を受け取ることが 1 つ目の目標である。

2 つ目の目標は、UDP 通信を用いて Matlab で受け取った Touch の関節角度を実機の 6 自由度マニピュレータにリアルタイムで送信し、追従させることである。本研究では Touch の 2 軸と 3 軸の関節角のみを使用していたが、今後は他の軸も追加して操作する。その際に実機のマニピュレータ（スレーブアーム）と Touch(マスターアーム) でリンクの長さや各関節の初期位置などが完全に異なるため、それらを補正する必要がある。

参考文献

- [1] HEBI Robotics <https://docs.hebi.us/tools.html#scope-gui>
- [2] HapticsDeviceGuide https://support.3dsystems.com/s/article/Haptic-Device-Guides?language=en_US
- [3] 永井清・土橋宏規:『ロボット機構学』。コロナ社, 2015.
- [4] 岡本颯太・大橋美咲:『仮想空間において接触を伴うロボットに対する反力を考慮したマスタースレーブ制御』。南山大学, 2022.