

手先空間におけるマスタスレーブ制御

2017SC019 神谷拓海 2017SC057 小川司 2017SC087 渡邊隼也

指導教員：中島明

1 はじめに

近年、マスタスレーブシステムは産業用ロボットや医療用ロボットなどに用いられており、どのような環境下でも安全性が保障されることや複雑な作業に対する操作性を向上させることが課題となっている。

マスタスレーブ制御とはロボットやマニピュレータを遠隔操作するための手段である。先行研究 [1] ではマスタ側から取得した関節角をスレーブに目標値として入力するマスタスレーブを行った。

本研究ではハプティクスデバイスをマスタ機とし、ロボットアームをスレーブ機としたマスタスレーブシステムを用いて、マスタから取得した手先位置をスレーブの目標値として入力するマスタスレーブ制御を行った。また、3次元6軸のロボットアームのシミュレーションを作成し、シミュレーションを用いたマスタスレーブシステムを作成した。スレーブ側の目標値を手先位置にする事で、マスタ側とスレーブ側の機構が違っていてもマスタスレーブ制御が可能である。今回の場合では4軸の部分のマスタ側とスレーブ側で違っている。

2 実験環境

2.1 ロボットアームと座標系の設定

シミュレーションでのマニピュレータはMOTOMAN-HP3Jのロボットアームを参考にした。参考にしたロボットアームの座標系を図1の $\Sigma_0 \sim \Sigma_7$ のように定義する。図1のように各座標系間の距離を d_1, d_4, a_2, a_3, a_7 とおく。具体的な値は表1に示す。

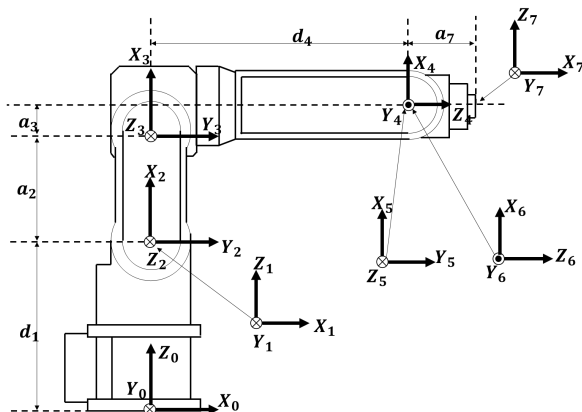


図1 アームのフレーム配置

表1 ロボットアームの各座標間のパラメータ

パラメータ	座標間距離 [m]
d_1	0.2900
d_4	0.2700
a_2	0.2600
a_3	0.0300
a_7	0.1315

2.2 Geomagic Touch

本研究ではマスタ機としてハプティクスデバイスのGeomagic Touch (以下、Touch とする) を使用し、手先位置を取得する。Touch のフレーム配置 $\Sigma_{t0} \sim \Sigma_{t7}$ を図2に示す。図2のように各座標間の距離を l_1, l_2, l_4, l_5 , フレーム Σ_{t1} から見たフレーム Σ_{t2} のy軸正方向の長さを l_3 と置き、具体的な数値を表2に示す。

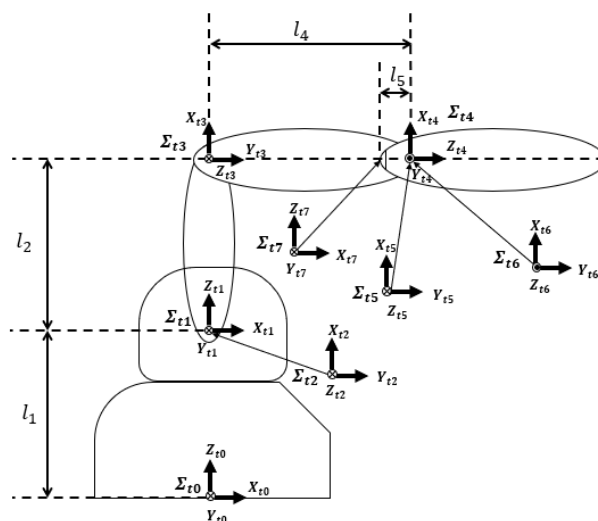


図2 Touch の機構

表2 Touch の各座標間のパラメータ

パラメータ	座標間距離 [m]
l_1	0.16535
l_2	0.13317
l_3	0.01735
l_4	0.13142
l_5	0.03604

3 マスタスレーブシステムの構築

3.1 システムの構築

実験環境は図3に示されるように Touch, HostPC, RealTimePC, ロボットアームの4つの機器によって構成されている。HostPCではTouchの制御用のプログラムをVisualStudioを用いて実行しており、TouchとHostPCを合わせてマスタ側としている。RealTimePCではロボットアームの制御用プログラムをLabVIEWを用いて実行しており、ロボットアームとRealTimePCを合わせてスレーブ側としている。

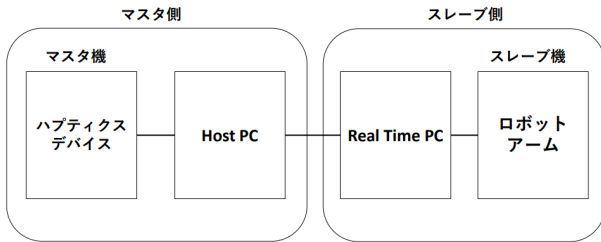


図3 実験環境の構成

3.2 OpenHaptics Toolkit によるデータの送信

OpenHaptics Toolkit とはマスタ機として使用している Touch の販売元である 3D Systems 社が提供しているハプティクスデバイス用のツールである。このツールを導入することでハプティクスデバイス専用のコマンドを使用できる。ただし、Touch に関してのプログラムは Microsoft 社が提供している統合開発環境である Visual Studio を使用し、開発している。

しかし、ロボットアームを制御するプログラムは制御システム開発ソフトウェアの一つである LabVIEW を用いて作成しているため、Visual Studio 側で取得した値を LabVIEW へ受け渡す必要がある。そこで、UDP 通信という通信方法を用いた。

UDP 送信をするにあたり double 値を送信する方法として、x, y, z 座標の3つのデータを1つの文字列として送信する方法で行っている。

3.3 MATLAB/simulink への UDP 通信

MATLAB/simulink への UDP 通信を実現するために、DSP System Toolbox の UDP Receive ブロックを用いてシミュレーションを作成した。受信したデータの形式は文字列であるため、これを Double 型に変換し、作成したシミュレーションの入力としている。

3.4 フローチャート

Touch から手先位置を取得し、UDP 送信するプログラムのフローチャートを図4に示す。ただし図4において j は int 型変数であり、2つのボタンの状態と j に代入される値の組み合わせを表3に示す。Visual Studio 側の送信

と LabVIEW 側で受信処理を完了するまでにラグが発生してしまうので、タイミングを合わせるために待機させている。

プログラムの流れとしては、実行開始を行うと待機状態になり、黒いボタンのみが押されている状態 ($j=1$) のとき値の送信を行い、j に 1 以外の値が代入されたタイミングで送信を一時停止させる。停止中に再び黒いボタンを押すと送信を再開させ、停止中に2つのボタンを同時押しするとプログラムの実行を終了させる。

表3 2つのボタンの状態と代入される値の対応表

黒ボタン	白ボタン	代入される値
OFF	OFF	0
ON	OFF	1
OFF	ON	2
ON	ON	3

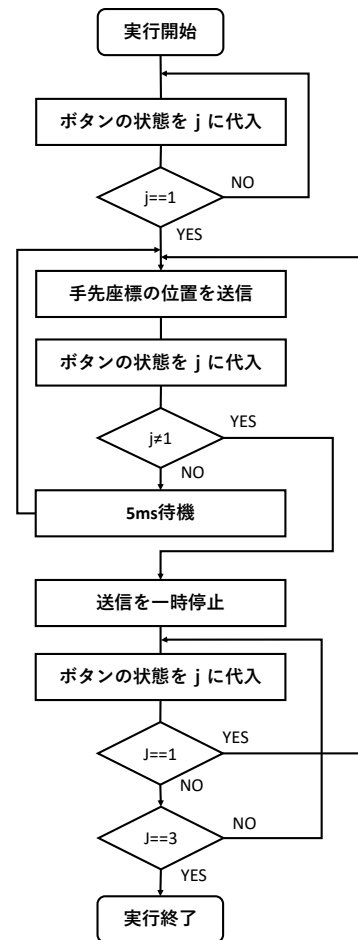


図4 プログラムのフローチャート

4 順運動学 [2][3][4]

4.1 マスタの順運動学

Touch のフレーム配置 $\Sigma_{t0} \sim \Sigma_{t7}$ の同時変換行列を (1) 式に示す.

$${}^{t0}T_{t7} = \begin{bmatrix} {}^{t0}R_{t7} & {}^{t0}p_{t7} \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (1)$$

このとき, Touch のペン先の位置は ${}^{t0}p_{t7}$ と表される. また, ${}^{t0}T_{t7}$ は以下の (2) 式で求まる.

$${}^{t0}T_{t7} = {}^{t0}T_{t1} {}^{t1}T_{t2} {}^{t2}T_{t3} {}^{t3}T_{t4} {}^{t4}T_{t5} {}^{t5}T_{t6} {}^{t6}T_{t7} \quad (2)$$

各同時変換行列は (3) 式のように示される.

$${}^{ti}T_{t(i+1)} = \begin{bmatrix} {}^{ti}R_{t(i+1)} & {}^{ti}p_{t(i+1)} \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (i = 0, 1, 2, 3, 4, 5, 6) \quad (3)$$

隣り合う座標系ごとの位置と姿勢の関係を以下に示す. ただし, Touch の第 i 軸の関節角を θ_i とし, $S_{ti} := \sin(\theta_i)$, $C_{ti} := \cos(\theta_i)$ と表記する.

$${}^{t0}R_{t1} = \begin{bmatrix} C_{t1} & -S_{t1} & 0 \\ S_{t1} & C_{t1} & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^{t0}p_{t1} = \begin{bmatrix} 0 \\ 0 \\ l_1 \end{bmatrix}$$

$${}^{t1}R_{t2} = \begin{bmatrix} S_{t2} & C_{t2} & 0 \\ 0 & 0 & 1 \\ C_{t2} & -S_{t2} & 0 \end{bmatrix} {}^{t1}p_{t2} = \begin{bmatrix} 0 \\ l_3 \\ 0 \end{bmatrix}$$

$${}^{t2}R_{t3} = \begin{bmatrix} C_{t3} & -S_{t3} & 0 \\ S_{t3} & C_{t3} & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^{t2}p_{t3} = \begin{bmatrix} l_2 \\ 0 \\ -l_3 \end{bmatrix}$$

$${}^{t3}R_{t4} = \begin{bmatrix} C_{t4} & -S_{t4} & 0 \\ 0 & 0 & 1 \\ -S_{t4} & -C_{t4} & 0 \end{bmatrix} {}^{t3}p_{t4} = \begin{bmatrix} 0 \\ l_4 \\ 0 \end{bmatrix}$$

$${}^{t4}R_{t5} = \begin{bmatrix} C_{t5} & -S_{t5} & 0 \\ 0 & 0 & -1 \\ S_{t5} & C_{t5} & 0 \end{bmatrix} {}^{t4}p_{t5} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$${}^{t5}R_{t6} = \begin{bmatrix} C_{t6} & -S_{t6} & 0 \\ 0 & 0 & 1 \\ -S_{t6} & -C_{t6} & 0 \end{bmatrix} {}^{t5}p_{t6} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$${}^{t6}R_{t7} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix} {}^{t6}p_{t7} = \begin{bmatrix} 0 \\ 0 \\ -l_5 \end{bmatrix}$$

4.2 スレーブの順運動学

ロボットアームの手先位置と同時変換行列も 4.1 節と同様に求まる. ロボットアームの隣り合う座標系ごとの位置と姿勢の関係を以下に示す. ただし, ロボットアームの第 i 軸の関節角を q_i とし, $S_i := \sin(q_i)$, $C_i := \cos(q_i)$ と表記する.

$${}^0R_1 = \begin{bmatrix} C_1 & -S_1 & 0 \\ S_1 & C_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^0p_1 = \begin{bmatrix} 0 \\ 0 \\ d_1 \end{bmatrix}$$

$${}^1R_2 = \begin{bmatrix} S_2 & C_2 & 0 \\ 0 & 0 & 1 \\ C_2 & -S_2 & 0 \end{bmatrix} {}^1p_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$${}^2R_3 = \begin{bmatrix} C_3 & -S_3 & 0 \\ S_3 & C_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^2p_3 = \begin{bmatrix} a_2 \\ 0 \\ 0 \end{bmatrix}$$

$${}^3R_4 = \begin{bmatrix} C_4 & -S_4 & 0 \\ 0 & 0 & 1 \\ -S_4 & -C_4 & 0 \end{bmatrix} {}^3p_4 = \begin{bmatrix} a_3 \\ d_4 \\ 0 \end{bmatrix}$$

$${}^4R_5 = \begin{bmatrix} C_5 & -S_5 & 0 \\ 0 & 0 & -1 \\ S_5 & C_5 & 0 \end{bmatrix} {}^4p_5 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$${}^5R_6 = \begin{bmatrix} C_6 & -S_6 & 0 \\ 0 & 0 & 1 \\ -S_6 & -C_6 & 0 \end{bmatrix} {}^5p_6 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$${}^6R_7 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix} {}^6p_7 = \begin{bmatrix} 0 \\ 0 \\ a_7 \end{bmatrix}$$

5 運動方程式の導出 [2][3][4]

6 自由度のロボットアームの運動方程式の導出を行う. 各リンクに, リンク座標系と平行で重心を原点とするリンク重心座標系 $\Sigma_{c1} \sim \Sigma_{c6}$ を設置する. このリンク座標系での慣性テンソル I_{ci} の各成分は, 次のように示される.

$$I_{ci} = \begin{bmatrix} I_{cxxi} & I_{cxyi} & I_{cxzi} \\ I_{cyxi} & I_{cyyi} & I_{cyzi} \\ I_{czxi} & I_{czyi} & I_{czzi} \end{bmatrix}$$

並進運動に関するヤコビ行列を $J_{cvi}(q)$ とおき, 回転速度に関するヤコビ行列を $J_{cwi}(q)$ とおくと, 並進と回転のヤコビアンは以下ようになる.

$$\begin{cases} {}^0v_{ci} = J_{cvi}(q)\dot{q} & (i = 1, 2, 3, 4, 5, 6) \\ {}^0\omega_{ci} = J_{cwi}(q)\dot{q} & (i = 1, 2, 3, 4, 5, 6) \end{cases}$$

Σ_0 から見た各リンクの重心の角速度ベクトル ${}^0\omega_{ci}$ を回転行列 0R_i を用いて各リンクから見た各リンクの重心の角速度ベクトル ${}^i\omega_{ci}$ に変換する. 各リンクの慣性テンソルとヤコビアンより, 運動エネルギーは次のようになる. ただし, $(i = 1, 2, 3, 4, 5, 6)$ とする.

$$T_i = \frac{1}{2} m_i {}^0v_{ci}^T {}^0v_{ci} + \frac{1}{2} {}^i\omega_{ci}^T I_{ci} {}^i\omega_{ci}$$

第 i リンクの重心位置の z 成分を p_{czi} とおくと, ポテンシャルエネルギーは以下ようになる. そのとき, $(i = 1, 2, 3, 4, 5, 6)$ とする.

$$U_i = m_i g {}^0p_{czi}$$

運動エネルギー T とポテンシャルエネルギー U より, ラグランジアンは次のようになる.

$$L = T - U \quad (4)$$

この式 (4) をラグランジュの運動方程式 (式 (5))

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right)^T = \left(\frac{\partial L}{\partial q} \right)^T + F \quad (5)$$

に適用することで、マニピュレータの運動方程式

$$M(q)\ddot{q} + h(q, \dot{q}) + g(q) = \tau \quad (6)$$

が得られる。なお、式(5)の F は非保存外力であり、この場合は関節トルク τ である。ここで、 $M(q)\ddot{q}$ は慣性力項、 $h(q, \dot{q})$ は遠心力コリオリ力項、 $g(q)$ はマニピュレータに加わる重力項である。

6 スレーブの制御則 [5]

6.1 目標軌道の生成

スレーブ機の目標軌道を滑らかなものとするために、目標値 p に(7)式で表せられる2次遅れ系の伝達関数 $G_S(s)$ を掛ける。

$$G_S(s) = \left(\frac{1}{Ts+1} \right)^2 \quad (7)$$

ここで、ステップ入力で与えられる目標値を \bar{p} とすると、 $Y(s) = \bar{p}G_S(s)/s$ より目標軌道は

$$\begin{aligned} y(t) &= \mathcal{L}^{-1} \left[\frac{\bar{p}G_S(s)}{s} \right] \\ &= \mathcal{L}^{-1} \left[\frac{\bar{p}}{s} - \frac{\bar{p}T}{Ts+1} - \frac{\bar{p}T}{(Ts+1)^2} \right] \\ &= \bar{p}1 - (1 + \omega_n t)e^{-\omega_n t} \end{aligned}$$

と表される。両辺を t で微分すると、

$$y'(t) = -\bar{p}\omega_n^2 t e^{-\omega_n t}$$

であり、 $t=0$ のとき、

$$y'(0) = 0$$

となる。また、 $\omega_n = \frac{1}{T}$ より、

$$y(t) = \bar{p} \left\{ 1 - \left(1 + \frac{t}{T} \right) e^{-\frac{t}{T}} \right\}$$

と与えられる。ここで、 $t=T$ のとき、

$$y(T) = \bar{p}(1 - 2e^{-1}) \approx 0.264\bar{p}$$

となり、 $t=6T$ のとき、

$$y(6T) = \bar{p}(1 - 7e^{-6}) \approx 0.982\bar{p}$$

となる。

したがって、時刻 $t=T$ のとき目標値を約 $0.264\bar{p}$ に抑え、時刻 $t=6T$ のとき目標値を約 $0.982\bar{p}$ にすることができる。よって、ステップ入力で与えられる目標値に対し、2次遅れ系の伝達関数 $G_S(s)$ を掛けることで、目標軌道を滑らかにすることができる。また、本研究では時定数を $T=0.34$ とし、整定時間は約2秒となっている。

6.2 手先位置制御

手先位置を ${}^B P_R$ 、手先の姿勢を ${}^B \theta_R$ とする。

$$\begin{aligned} {}^B P_R &= h_1(q), \theta_R = h_2(q) \\ x_R &= \begin{bmatrix} {}^B P_R \\ {}^B \theta_R \end{bmatrix} = \begin{bmatrix} h_1(q) \\ h_2(q) \end{bmatrix} \end{aligned}$$

手先の位置と姿勢をまとめて、ヤコビアンは以下のように求めることができる。

$$J_R = \begin{bmatrix} \left(\frac{\partial h_1}{\partial q} \right)^T \\ \left(\frac{\partial h_2}{\partial q} \right)^T \end{bmatrix} = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix}$$

この J_R を使うことによって手先位置と姿勢の時間微分 \dot{x}_R を関節角の速度からヤコビアンにかけることで求めることができる。

$$\dot{x}_R = J_R(q)\dot{q} \quad (8)$$

また上記の(8)式を時間微分すれば以下の式が得られる。

$$\ddot{x}_R = \dot{J}_R(q)\dot{q} + J_R(q)\ddot{q} \quad (9)$$

(8)式より

$$\dot{q} = J_R(q)^{-1}\dot{x}_R \quad (10)$$

(9)式を \dot{q} について解き、(10)式を代入すると以下のようになる。

$$\ddot{q} = J_R(q)^{-1}(\ddot{x}_R - \dot{J}_R(q)J_R(q)^{-1}\dot{x}_R) \quad (11)$$

(6)式を上記の(10)、(11)式を用いて式変形をすると以下のようになる。

$$M(q)J_R^{-1}(\ddot{x}_R - \dot{J}_R J_R(q)^{-1}\dot{x}_R) + h(q, \dot{q})J_R^{-1}\dot{x}_R + g(q) = \tau$$

上記の式を x_R に関してまとめると以下のようになる。

$$M(q)J_R^{-1}\ddot{x}_R + (h(q, \dot{q}) - M J_R^{-1} \dot{J}_R) J_R^{-1} \dot{x}_R + g(q) = \tau \quad (12)$$

また、下記の(13)式を時間微分すると(14)式のようになる。

$$J_R J_R^{-1} = I \quad (13)$$

$$\dot{J}_R J_R^{-1}(q) + J_R \frac{d}{dt} J_R^{-1}(q) = 0 \quad (14)$$

上記の(14)式を $\frac{d}{dt} J_R^{-1}$ について解くと以下のようになる。

$$\frac{d}{dt} J_R^{-1} = -J_R^{-1} \dot{J}_R J_R^{-1} \quad (15)$$

上記の(15)式を(12)式に代入すると以下のようになる。

$$M(q)J_R^{-1}\ddot{x}_R + (h(q, \dot{q})J_R^{-1} + M \frac{d}{dt} J_R^{-1})\dot{x}_R + g(q) = \tau \quad (16)$$

右辺の τ は関節角ベースでのトルクなのでこれを手先空間ベースでのトルクに直す必要がある。ロボットの手に加わる力 $F \in \mathbb{R}^3$ と関節トルク τ の静力学的関係は

$$\begin{aligned}\tau &= J_R^T F \\ F &= J_R^{-T} \tau\end{aligned}$$

となる。よって、(16) 式の両辺に J_R^{-T} をかけると以下のようなになる。

$$M_R(q)\ddot{x}_R + h_R(q, \dot{x}_R) + g_R(q) = J_R^{-T}(q)\tau$$

ここで、 $M_R(q)\ddot{q}$, $h_R(q, \dot{x}_R)$, $g_R(q)$ は手先空間における慣性力項、遠心コリオリ力項、重力項である。この関係を用いると、手先位置ベースのフィードバック制御則が得られる。

$$\tau = J_R^T \left\{ -K_p(x_R - x_{Rd}) - K_i \int (x_R - x_{Rd}) dt - K_d \dot{x}_R + G \right\}$$

なお、 G は重力補償である。

6.3 ヤコビアン の 導 出

手先の位置と姿勢をまとめヤコビアンは以下のように求めることができる。

$$J_R = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix}$$

4 節の順運動学の式は次のようになる。

$$p = f(q) \quad (17)$$

式 (17) を時間微分すると、関節角の角速度 \dot{q} と手先位置の微分 \dot{p} の関係はヤコビアン $J_v(q)$ を用いて次の式で表すことができる。

$$\dot{p} = J_v(q)\dot{q}$$

ただし、 $J_v(q)$ は以下で与えられる。

$$J_v = \frac{\partial f}{\partial q}$$

関節角の角速度 \dot{q} と手先姿勢の時間微分 $\dot{\phi}$ の関係をヤコビアン J_ϕ を用いて表すことができる。

$$\dot{\phi} = J_\phi(q)\dot{q}$$

参考文献 [6] より姿勢角の時間微分 $\dot{\phi}$ と手先空間での回転速度 ω を結ぶ T_r が存在する。

$$\begin{aligned}\omega &= T_r(\phi)\dot{\phi} \\ \dot{\phi} &= T_r^{-1}(\phi)\omega\end{aligned}$$

$$T_r({}^A\phi_B) = \begin{bmatrix} 0 & -\sin(\alpha) & \cos(\alpha)\cos\beta \\ 0 & \cos(\alpha) & \sin(\alpha)\cos(\beta) \\ 1 & 0 & -\sin(\beta) \end{bmatrix}$$

また関節の角速度 \dot{q} と手先の回転速度 ω を結ぶヤコビアン J_ω が存在する。

$$\omega = J_\omega(q)\dot{q}$$

ヤコビアン J_ω は参考文献 [4] より以下のようなになる。

$$J_\omega = [{}^0z_i] (i = 0, 1, 2, 3, 4, 5, 6)$$

$${}^0z_i = {}^0R_i e_z$$

$$e_z = [0, 0, 1]^T$$

よって以下の式で $\dot{\phi}$ を求めることができる。

$$\dot{\phi} = T_r^{-1}(\phi)J_\omega(q)\dot{q}$$

なので、上記の式 (18) より関節角の角速度 \dot{q} と手先姿勢の時間微分 $\dot{\phi}$ を結ぶヤコビアン J_ϕ は以下の式のようなになる。

$$J_\phi = T_r^{-1}(\phi)J_\omega(q)$$

6.4 シミュレーション

6.2 節の制御則を用いて、MATLAB/simulink で 3 次元 6 軸のロボットアームでの手先位置制御のシミュレーションを作成した。そのシミュレーションの手先位置・姿勢の実験結果を図 5, 図 6 に示す。各関節の初期値を 0° とし、手先の目標値を $[0.2, 0.2, 0.2]$ とし、初期の姿勢を維持したまま移動する。

青線が目標軌道で赤線がシミュレーションの結果である。

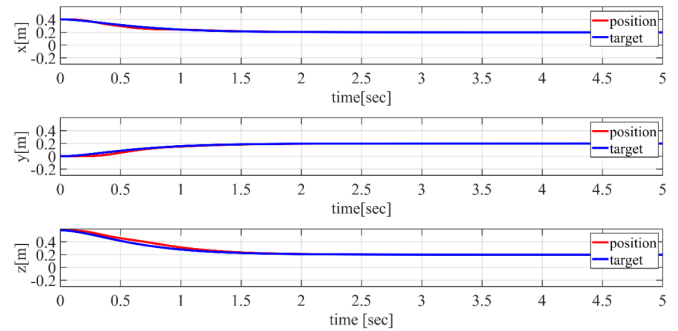


図 5 目標軌道とシミュレーションの比較 (手先位置)

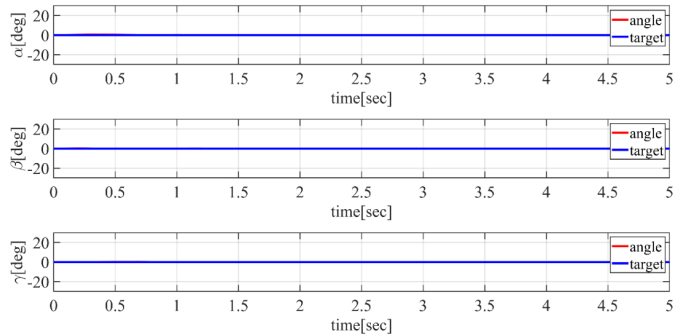


図 6 目標軌道とシミュレーションの比較 (姿勢)

シミュレーションの結果が目標軌道とほぼ同じ動きをしていることが確認できた。

7 マスタスレーブ制御

7.1 マスタとスレーブの初期姿勢

マスタスレーブシステムについて、マスタとスレーブの初期姿勢を合わせることは必須になる。本研究ではマスタとスレーブの初期姿勢をスレーブの各軸の関節角が全て $0[\text{deg}]$ のときの手先座標 $[0.4, 0, 0.58][\text{m}]$ としている。その時の姿勢を図 7 に示す。

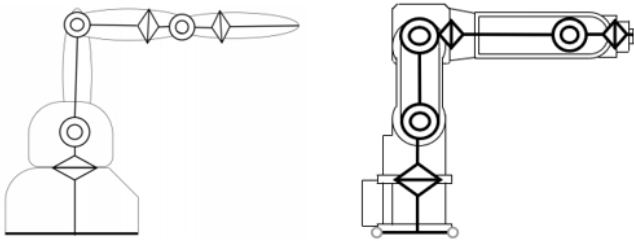


図 7 マスタ機 (左) とスレーブ機 (右) の初期姿勢

7.2 スケール合わせ

目標の座標をマスタ側からスレーブ側へ送る際に、マスタの座標をそのまま送るだけではスレーブのロボットアームの動く範囲が狭くなってしまいます。そこで、目標座標を大きくすることでスレーブ側の手先の動く範囲を拡大することができる。その際のゲインを以下のように設定した。

$$K_s = \frac{\text{スレーブ側の手先位置}}{\text{マスタ側の手先位置}}$$

図 8 の赤線はマスタ側での手先位置の座標である。その値を $K_s = [2.5001, 2.5001, 2.2131]$ 倍し、スレーブへの入力としたものが青線である。

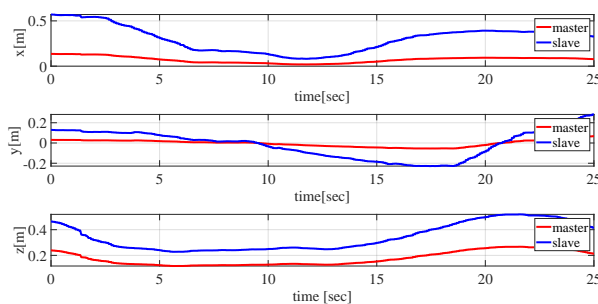


図 8 スケール比較

7.3 マスタスレーブ制御のシミュレーション

6.4 節のシミュレーションを用いて、Touch から受け取った値を目標値とするマスタスレーブシステムを作成した。そのときの実験結果のうち、手先位置の結果を図 9,

姿勢角の結果を図 10 に示す。

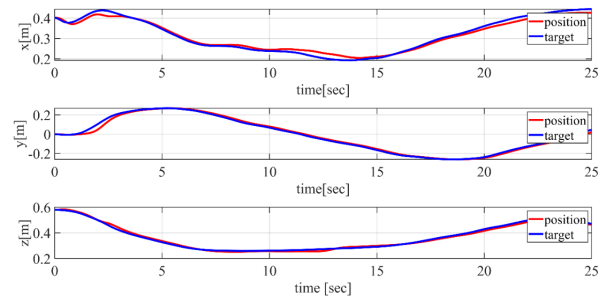


図 9 Touch とシミュレーションの比較 (手先位置)

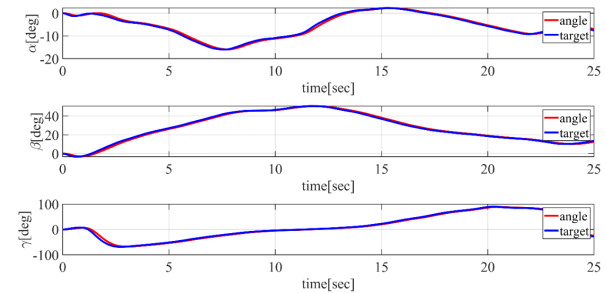


図 10 Touch とシミュレーションの比較 (姿勢)

青線がマスタ機の動きで、赤線がシミュレーションの動きである。追従できていることが分かる。

8 おわりに

これまでに、MATLAB/simulink で 3 次元 6 軸のロボットアームでの手先位置制御のシミュレーションを作成した。さらにハプティクスデバイスを導入し、マスタ機から送信した手先位置・姿勢を目標値として入力し、シミュレーション上での 6 自由度のロボットアームをマスタスレーブ系として制御することができた。今後の展望として、スレーブ機を実機のロボットアームとしたマスタスレーブ系の制御があげられる。

参考文献

- [1] 井後圭人・山口正成：ハプティクスデバイスを用いたマスタスレーブ制御。南山大学,2019.
- [2] 永井 清・土橋 宏規：『ロボティクスシリーズ 8 ロボット機構学』。コロナ社，東京，2015.
- [3] John J.Craig：『ロボティクス-機構・力学・制御』。共立出版社，東京，2016.
- [4] 吉川 恒夫：『コンピュータ制御 機械システムシリーズ ロボット制御基礎論』。コロナ社，東京，1988.
- [5] 石黒太雅・松永純弥・吉田樹：視覚情報を用いたマニピュレータによるボールの打ち上げ。南山大学，2018
- [6] 中島 明：『メカトロニクス研究 講義ノート』。南山大学