令和 7 年度

実 験 レ ポ ー ト

題目	～2 軸ロボットの運動制御実験～ Quanser Consulting 社製　2軸ロボット

学　　科	電気電子システム工学コース
学籍番号	a0527
氏　　名	野口 史遠
提 出 日	令和 7 年 5 月 12 日

舞鶴工業高等専門学校

目 次

第 1 章　概要

図 1.1, 1.2 に示す 4 リンク 2 軸ロボット実験装置は，ギアを介して軸 P_1, P_5 に取り付けられた 2 つのモータにより手先の位置 P_3 を目標位置に追従させることを目的とした実験装置である．このことを実現するためには，手先位置からモータの回転角度を求める逆動力学とモータの回転角度から手先位置を求める順動力学を理解する必要がある．

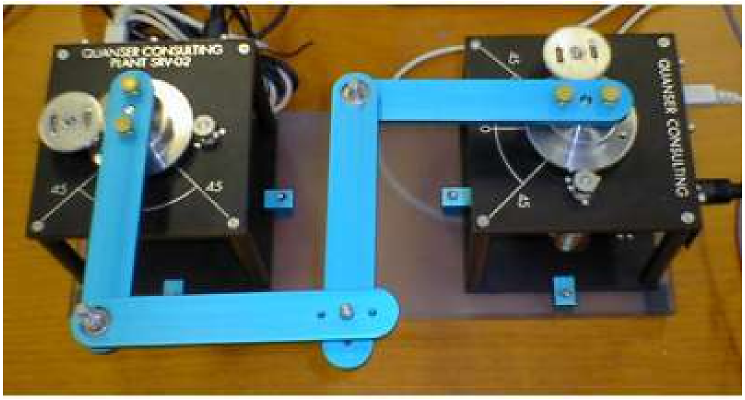


図 1.1 ： 4 リンク 2 軸ロボット実験装置（型番：55023）

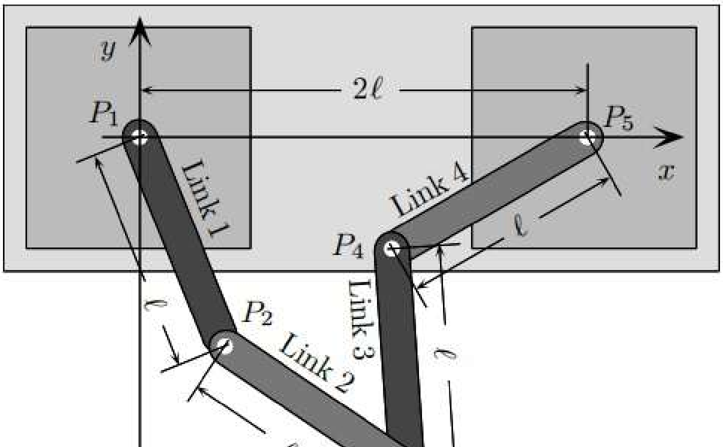


図 1.2 ： 4 つリンクの関係

第 2 章　順運動学と逆運動学

2.1　運動方程式

リンク 1 を手で基準位置から回転させてリンク 1 の基準位置からの回転角 $\theta_x(t)$ がさほど大きくなければ、リンク 4 は静止したままである．したがって、P_1, P_5 の角度 $\theta_x(t)$, $\theta_y(t)$ がその基準角度の近傍で動作するとすると、リンク 1 とリンク 4 との干渉は無視できる（実際、リンク 1 あるいはリンク 4 を手で微小回転させてもリンク 4 あるいはリンク 1 は静止したままである）．このとき、リンク 1、リンク 4 の運動方程式は

$$
\begin{aligned}
J_x\ddot{\theta}_x(t) &= -c_x\dot{\theta}_x(t) + \tau_x(t) \\
J_y\ddot{\theta}_y(t) &= -c_y\dot{\theta}_y(t) + \tau_y(t)
\end{aligned}
\tag{2.1}
$$

となる．ただし、$\tau_x(t)$, $\tau_y(t)$ は軸 P_1, P_5 に加わるトルク、J_x, J_y はリンク 1、リンク 4 の慣性モーメントであり、c_x, c_y は軸 P_1, P_5 の粘性摩擦係数である．また、モータドライバに加える電圧 $v_x(t)$, $v_y(t)$ とトルク $\tau_x(t)$, $\tau_y(t)$ が比例関係

$$
\begin{cases}
\tau_x(t) = k_{fx}v_x(t) \\
\tau_y(t) = k_{fy}v_y(t)
\end{cases}
\tag{2.2}
$$

にあると仮定すると、

$$
\begin{aligned}
J_x\ddot{\theta}_x(t) &= -c_x\dot{\theta}_x(t) + k_{fx}v_x(t) \\
J_y\ddot{\theta}_y(t) &= -c_y\dot{\theta}_y(t) + k_{fy}v_y(t)
\end{aligned}
\tag{2.3}
$$

であるから、$v_x(t)$, $v_y(t)$ から $\theta_x(t)$, $\theta_y(t)$ への伝達関数 $G_x(s)$, $G_y(s)$ は

$$
\begin{cases}
G_x(s) = \dfrac{b_x}{s(s+a_x)}, \quad a_x = \dfrac{c_x}{J_x}, \quad b_x = \dfrac{k_{fx}}{J_x} \\
G_y(s) = \dfrac{b_y}{s(s+a_y)}, \quad a_y = \dfrac{c_y}{J_y}, \quad b_y = \dfrac{k_{fy}}{J_y}
\end{cases}
\tag{2.4}
$$

となる.

2.2　逆運動学

ここでは，手先にあたる軸 P_3 の座標 (x, y) からモータの回転軸 P_1, P_5 の角度 θ_x, θ_y を求める．このように，手先の座標から回転角を求めることを逆運動学という．

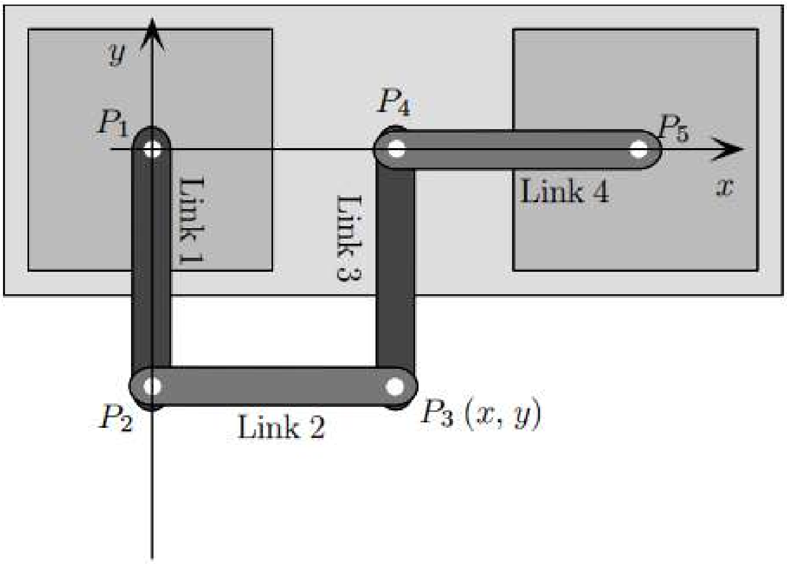


図 2.1 ：初期位置（基準位置）

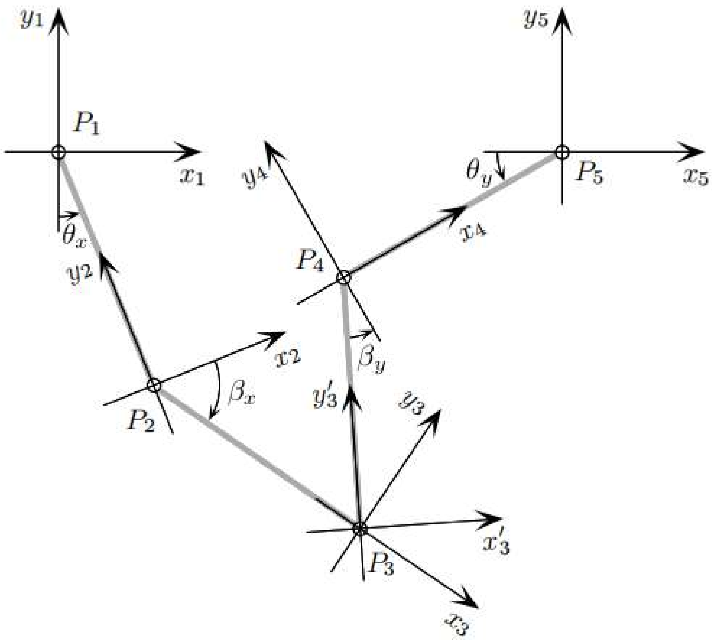


図 2.2 ：逆運動学のための各座標系

図 2.2 に示すように，リンク 1，リンク 2 に関しては x_1y_1 座標系と x_2y_2 座標系，x_2y_2 座標系と x_3y_3 座標系は

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \cos\theta_x & -\sin\theta_x \\ \sin\theta_x & \cos\theta_x \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} + \begin{bmatrix} \ell\sin\theta_x \\ -\ell\cos\theta_x \end{bmatrix} \tag{2.5}$$

$$\begin{aligned} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} &= \begin{bmatrix} \cos(-\beta_x) & -\sin(-\beta_x) \\ \sin(-\beta_x) & \cos(-\beta_x) \end{bmatrix} \begin{bmatrix} x_3 \\ y_3 \end{bmatrix} + \begin{bmatrix} \ell\cos\beta_x \\ -\ell\sin\beta_x \end{bmatrix} \\ &= \begin{bmatrix} \cos\beta_x & \sin\beta_x \\ -\sin\beta_x & \cos\beta_x \end{bmatrix} \begin{bmatrix} x_3 \\ y_3 \end{bmatrix} + \begin{bmatrix} \ell\cos\beta_x \\ -\ell\sin\beta_x \end{bmatrix} \end{aligned} \tag{2.6}$$

という関係にある．(2.5)，(2.6) 式を書き換えると，

$$\begin{bmatrix} x_1 & y_1 & 1 \end{bmatrix}^T = T_{12} \begin{bmatrix} x_2 & y_2 & 1 \end{bmatrix}^T \tag{2.7}$$

$$\begin{bmatrix} x_2 & y_2 & 1 \end{bmatrix}^T = T_{23} \begin{bmatrix} x_3 & y_3 & 1 \end{bmatrix}^T \tag{2.8}$$

$$T_{12} = \begin{bmatrix} \cos\theta_x & -\sin\theta_x & \ell\sin\theta_x \\ \sin\theta_x & \cos\theta_x & -\ell\cos\theta_x \\ 0 & 0 & 1 \end{bmatrix}, \quad T_{23} = \begin{bmatrix} \cos\beta_x & \sin\beta_x & -\ell\sin\beta_x \\ -\sin\beta_x & \cos\beta_x & \ell\cos\beta_x \\ 0 & 0 & 1 \end{bmatrix}$$

となるから，x_1y_1 座標系と x_3y_3 座標系との間に以下の関係式が成立する．

$$\begin{bmatrix} x_1 & y_1 & 1 \end{bmatrix}^T = T_{13} \begin{bmatrix} x_3 & y_3 & 1 \end{bmatrix}^T, \quad T_{13} = T_{12}T_{23} \tag{2.9}$$

(2.9) 式を利用して，x_3y_3 座標系の原点 P_3 を x_1y_1 座標系で表すと，

$$\begin{aligned} \begin{bmatrix} x & y & 1 \end{bmatrix}^T &= T_{13} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \\ &= [\ell\cos\theta_x\cos\beta_x + \ell\sin\theta_x\sin\beta_x + \ell\sin\theta_x, \quad -\ell\cos\theta_x\sin\beta_x + \ell\sin\theta_x\cos\beta_x - \ell\cos\theta_x, \quad 1]^T \\ &= [\ell\cos(\theta_x - \beta_x) + \ell\sin\theta_x, \quad \ell\sin(\theta_x - \beta_x) - \ell\cos\theta_x, \quad 1]^T \end{aligned} \tag{2.10}$$

となる．(2.10) 式より

$$\begin{cases} x = \ell\cos(\theta_x - \beta_x) + \ell\sin\theta_x \\ y = \ell\sin(\theta_x - \beta_x) - \ell\cos\theta_x \end{cases} \tag{2.11}$$

$$\begin{aligned} &\Longrightarrow \begin{cases} x - \ell\sin\theta_x = \ell\cos(\theta_x - \beta_x) \\ y + \ell\cos\theta_x = \ell\sin(\theta_x - \beta_x) \end{cases} \\ &\Longrightarrow (x - \ell\sin\theta_x)^2 + (y + \ell\cos\theta_x)^2 = \ell^2 \end{aligned} \tag{2.12}$$

であるから，P_3 は中心 $(\ell\sin\theta_x, -\ell\cos\theta_x)$，半径 ℓ の円上を移動することがわかる．(2.12) 式を書き換えると

$$x^2+y^2-2\ell(\sin\theta_x x-\cos\theta_x y)=0$$

$$\Longrightarrow \quad x^2+y^2-2\ell\sqrt{x^2+y^2}\sin(\theta_x+\phi_x)=0 \quad , \quad \begin{cases}\cos\phi_x = \dfrac{x}{\sqrt{x^2+y^2}} \\ \sin\phi_x = \dfrac{y}{\sqrt{x^2+y^2}}\end{cases}$$

$$\Longrightarrow \quad \sin(\theta_x+\phi_x)=\frac{\sqrt{x^2+y^2}}{2\ell} \quad , \quad \theta_x=\sin^{-1}\left(\frac{\sqrt{x^2+y^2}}{2\ell}\right)-\phi_x \tag{2.13}$$

となる．したがって，P_3 の座標 (x,y) からモータの回転軸 P_1 の角度 θ_x が次式により求まる．

$$\theta_x=\sin^{-1}\left(\frac{\sqrt{x^2+y^2}}{2\ell}\right)+\tan^{-1}\left(\frac{y}{x}\right) \tag{2.14}$$

同様に，リンク 3，リンク 4 に関して考えると，x_1y_1 座標系と x_5y_5 座標系，x_5y_5 座標系と x_4y_4 座標系，x_4y_4 座標系と $x_3'y_3'$ 座標系は

$$\begin{bmatrix}x_1\\y_1\end{bmatrix}=\begin{bmatrix}1&0\\0&1\end{bmatrix}\begin{bmatrix}x_5\\y_5\end{bmatrix}+\begin{bmatrix}2\ell\\0\end{bmatrix} \tag{2.15}$$

$$\begin{bmatrix}x_5\\y_5\end{bmatrix}=\begin{bmatrix}\cos\theta_y&-\sin\theta_y\\\sin\theta_y&\cos\theta_y\end{bmatrix}\begin{bmatrix}x_4\\y_4\end{bmatrix}+\begin{bmatrix}-\ell\cos\theta_y\\-\ell\sin\theta_y\end{bmatrix} \tag{2.16}$$

$$\begin{aligned}\begin{bmatrix}x_4\\y_4\end{bmatrix}&=\begin{bmatrix}\cos(-\beta_y)&-\sin(-\beta_y)\\\sin(-\beta_y)&\cos(-\beta_y)\end{bmatrix}\begin{bmatrix}x_3'\\y_3'\end{bmatrix}+\begin{bmatrix}-\ell\sin\beta_y\\-\ell\cos\beta_y\end{bmatrix}\\&=\begin{bmatrix}\cos\beta_y&\sin\beta_y\\-\sin\beta_y&\cos\beta_y\end{bmatrix}\begin{bmatrix}x_3'\\y_3'\end{bmatrix}+\begin{bmatrix}-\ell\sin\beta_y\\-\ell\cos\beta_y\end{bmatrix}\end{aligned} \tag{2.17}$$

という関係にある．(2.15)，(2.16)，(2.17) 式を書き換えると，

$$\begin{bmatrix}x_1&y_1&1\end{bmatrix}^T=T_{15}\begin{bmatrix}x_5&y_5&1\end{bmatrix}^T \tag{2.18}$$

$$\begin{bmatrix}x_5&y_5&1\end{bmatrix}^T=T_{54}\begin{bmatrix}x_4&y_4&1\end{bmatrix}^T \tag{2.19}$$

$$\begin{bmatrix}x_4&y_4&1\end{bmatrix}^T=T_{43}'\begin{bmatrix}x_3'&y_3'&1\end{bmatrix}^T \tag{2.20}$$

ここで各変換行列は

$$T_{15}=\begin{bmatrix}1&0&2\ell\\0&1&0\\0&0&1\end{bmatrix}\quad T_{54}=\begin{bmatrix}\cos\theta_y&-\sin\theta_y&-\ell\cos\theta_y\\\sin\theta_y&\cos\theta_y&-\ell\sin\theta_y\\0&0&1\end{bmatrix}\quad T'_{43}=\begin{bmatrix}\cos\beta_y&\sin\beta_y&-\ell\sin\beta_y\\-\sin\beta_y&\cos\beta_y&-\ell\cos\beta_y\\0&0&1\end{bmatrix}$$

となるから，x_1y_1 座標系と $x'_3y'_3$ 座標系との間に以下の関係式が成立する．

$$\begin{bmatrix}x_1&y_1&1\end{bmatrix}^T=T'_{13}\begin{bmatrix}x'_3&y'_3&1\end{bmatrix}^T,\quad T'_{13}=T_{15}T_{54}T'_{43}\tag{2.21}$$

(2.21) 式を利用して，$x'_3y'_3$ 座標系の原点 P_3 を x_1y_1 座標系で表すと，

$$\begin{aligned}\begin{bmatrix}x&y&1\end{bmatrix}^T&=T'_{13}\begin{bmatrix}0\\0\\1\end{bmatrix}\\&=\begin{bmatrix}-\ell\cos\theta_y\sin\beta_y+\ell\sin\theta_y\cos\beta_y-\ell\cos\theta_y+2\ell\\-\ell\cos\theta_y\cos\beta_y+\ell\sin\theta_y\sin\beta_y-\ell\sin\theta_y\\1\end{bmatrix}\\&=\begin{bmatrix}-\ell\cos(\theta_y+\beta_y)-\ell\cos\theta_y+2\ell\\-\ell\sin(\theta_y+\beta_y)-\ell\sin\theta_y\\1\end{bmatrix}\end{aligned}\tag{2.22}$$

となる．(2.22) 式より

$$\begin{cases}x'=x-2\ell=-\ell\cos(\theta_y+\beta_y)-\ell\cos\theta_y\\y=-\ell\sin(\theta_y+\beta_y)-\ell\sin\theta_y\end{cases}\tag{2.23}$$

$$\Longrightarrow\begin{cases}x'+\ell\cos\theta_y=-\ell\cos(\theta_y+\beta_y)\\y+\ell\sin\theta_y=-\ell\sin(\theta_y+\beta_y)\end{cases}\Longrightarrow(x'+\ell\cos\theta_y)^2+(y+\ell\sin\theta_y)^2=\ell^2\tag{2.24}$$

であるから，P_3 は中心 $(2\ell-\ell\cos\theta_y,\ -\ell\sin\theta_y)$，半径 ℓ の円上を移動することがわかる．(2.24) 式を書き換えると

$$\begin{aligned}&x'^2+y^2+2\ell(x'\cos\theta_y+y\sin\theta_y)=0\\\Longrightarrow\quad&x'^2+y^2-2\ell\sqrt{x'^2+y^2}\sin(\theta_y+\phi_y)=0,\quad\begin{cases}\cos\phi_y=-\dfrac{y}{\sqrt{x'^2+y^2}}\\\sin\phi_y=-\dfrac{x'}{\sqrt{x'^2+y^2}}\end{cases}\\\Longrightarrow\quad&\sin(\theta_y+\phi_y)=\frac{\sqrt{x'^2+y^2}}{2\ell}\\\Longrightarrow\quad&\theta_y=\sin^{-1}\left(\frac{\sqrt{x'^2+y^2}}{2\ell}\right)-\phi_y\end{aligned}\tag{2.25}$$

したがって，P_3 の座標 (x, y) からモータの回転軸 P_5 の角度 θ_y が次式により求まる．

$$\begin{aligned}\theta_y &= \sin^{-1}\left(\frac{\sqrt{x'^2 + y^2}}{2\ell}\right) - \tan^{-1}\left(\frac{x'}{y}\right) \\ &= \sin^{-1}\left(\frac{\sqrt{(x-2\ell)^2 + y^2}}{2\ell}\right) - \tan^{-1}\left(\frac{x-2\ell}{y}\right) \end{aligned} \tag{2.26}$$

2.3 順運動学

ここでは、モータの回転軸 P_1，P_5 の角度 θ_x，θ_y から手先 P_3 の座標 (x, y) を求める．このように、回転軸から手先の座標を求めることを順運動学という．

軸 P_2，P_4 の回転角 β_x，β_y が検出可能であれば (2.11) 式や (2.23) 式により手先 P_3 の座標 (x, y) を求めることができるが、本実験装置は軸 P_2，P_4 にはセンサが取り付けられていない．そこで、x_1y_1 座標系、x_2y_2 座標系、x_3y_3 座標系を考える代わりに、図 2.3 に示す x_1y_1 座標系、x_2y_2 座標系、x_my_m 座標系、$x'_3y'_3$ 座標系を考える．

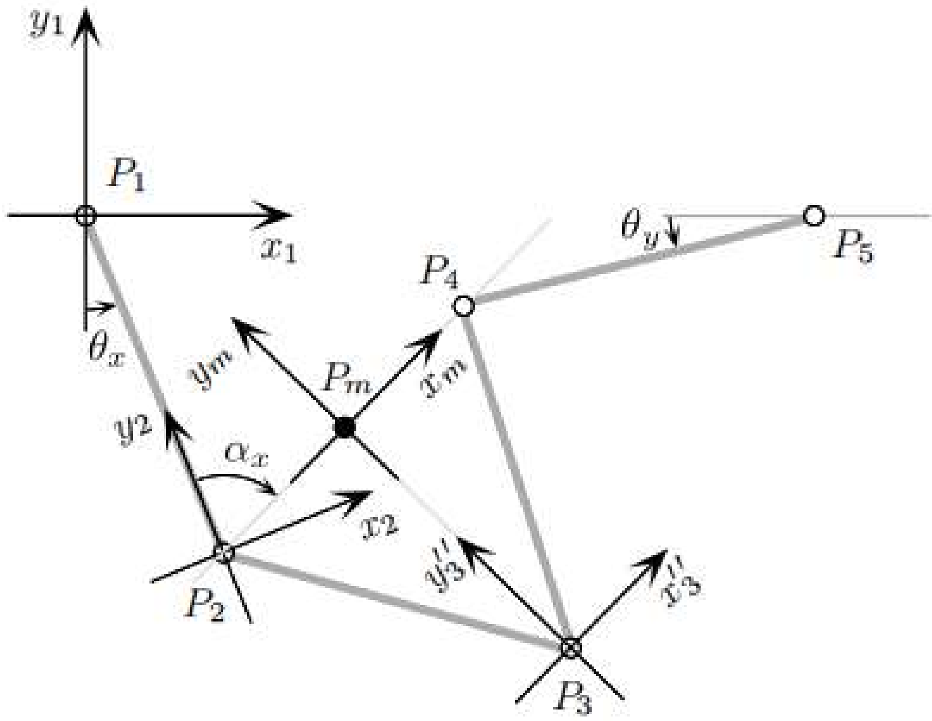


図 **2.3** : 順運動学のための各座標系 1

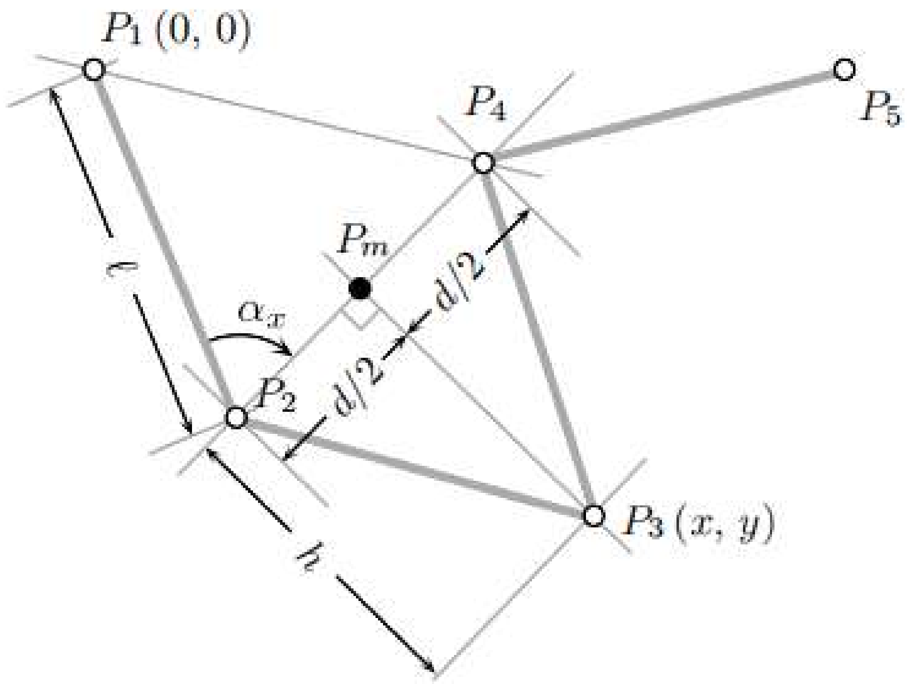


図 2.4 ：順運動学のための各座標系 2

図 2.3，2.4 より x_2y_2 座標系と x_my_m 座標系，x_my_m 座標系と $x'_3y'_3$ 座標系は

$$\begin{aligned}\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} &= \begin{bmatrix} \cos\left(\frac{\pi}{2}-\alpha_x\right) & -\sin\left(\frac{\pi}{2}-\alpha_x\right) \\ \sin\left(\frac{\pi}{2}-\alpha_x\right) & \cos\left(\frac{\pi}{2}-\alpha_x\right) \end{bmatrix} \begin{bmatrix} x_m \\ y_m \end{bmatrix} + \begin{bmatrix} \frac{d}{2}\cos\left(\frac{\pi}{2}-\alpha_x\right) \\ \frac{d}{2}\sin\left(\frac{\pi}{2}-\alpha_x\right) \end{bmatrix} \\ &= \begin{bmatrix} \sin\alpha_x & -\cos\alpha_x \\ \cos\alpha_x & \sin\alpha_x \end{bmatrix} \begin{bmatrix} x_m \\ y_m \end{bmatrix} + \begin{bmatrix} \frac{d}{2}\sin\alpha_x \\ \frac{d}{2}\cos\alpha_x \end{bmatrix} \end{aligned} \tag{2.27}$$

$$\begin{bmatrix} x_m \\ y_m \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x''_3 \\ y''_3 \end{bmatrix} + \begin{bmatrix} 0 \\ -h \end{bmatrix} \tag{2.28}$$

という関係にある．(2.27)，(2.28) 式を書き換えると，

$$\begin{bmatrix} x_2 & y_2 & 1 \end{bmatrix}^T = T_{2m} \begin{bmatrix} x_m & y_m & 1 \end{bmatrix}^T \tag{2.29}$$

$$\begin{bmatrix} x_m & y_m & 1 \end{bmatrix}^T = T''_{m3} \begin{bmatrix} x''_3 & y''_3 & 1 \end{bmatrix}^T \tag{2.30}$$

$$T_{2m} = \begin{bmatrix} \sin\alpha_x & -\cos\alpha_x & \frac{d}{2}\sin\alpha_x \\ \cos\alpha_x & \sin\alpha_x & \frac{d}{2}\cos\alpha_x \\ 0 & 0 & 1 \end{bmatrix}, \quad T''_{m3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -h \\ 0 & 0 & 1 \end{bmatrix}$$

となるから，x_1y_1 座標系と $x_3''y_3''$ 座標系との間に以下の関係式が成立する．

$$\begin{bmatrix} x_1 & y_1 & 1 \end{bmatrix}^T = T_{13}'' \begin{bmatrix} x_3'' & y_3'' & 1 \end{bmatrix}^T, \quad T_{13}'' = T_{12}T_{2m}T_{m3}'' \tag{2.31}$$

(2.31) 式を利用して，$x_3''y_3''$ 座標系の原点 P_3 を x_1y_1 座標系で表すと，

$$\begin{aligned}
\begin{bmatrix} x & y & 1 \end{bmatrix}^T &= T_{13}'' \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \\
&= \begin{bmatrix} h(\cos\theta_x \cos\alpha_x + \sin\theta_x \sin\alpha_x) - \dfrac{d}{2}(\sin\theta_x \cos\alpha_x - \cos\theta_x \sin\alpha_x) + \ell \sin\theta_x \\ h(\sin\theta_x \cos\alpha_x - \cos\theta_x \sin\alpha_x) + \dfrac{d}{2}(\cos\theta_x \cos\alpha_x + \sin\theta_x \sin\alpha_x) - \ell \cos\theta_x \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} h\cos(\theta_x - \alpha_x) - \dfrac{d}{2}\sin(\theta_x - \alpha_x) + \ell \sin\theta_x \\ h\sin(\theta_x - \alpha_x) + \dfrac{d}{2}\cos(\theta_x - \alpha_x) - \ell \cos\theta_x \\ 1 \end{bmatrix}
\end{aligned} \tag{2.32}$$

となる．したがって，d，h，α_x が求まれば手先 P_3 の座標が

$$\begin{cases} x = h\cos(\theta_x - \alpha_x) - \dfrac{d}{2}\sin(\theta_x - \alpha_x) + \ell \sin\theta_x \\ y = h\sin(\theta_x - \alpha_x) + \dfrac{d}{2}\cos(\theta_x - \alpha_x) - \ell \cos\theta_x \end{cases} \tag{2.33}$$

により求まる．

x_1y_1 座標系における P_2，P_4 の座標をそれぞれ (p_{2x}, p_{2y})，(p_{4x}, p_{4y}) とすると，

$$\begin{cases} p_{2x} = \ell \sin\theta_x \\ p_{2y} = -\ell \cos\theta_x \end{cases}, \quad \begin{cases} p_{4x} = 2\ell - \ell \cos\theta_y \\ p_{4y} = -\ell \sin\theta_y \end{cases} \tag{2.34}$$

であるから，d は

$$d = P_2P_4 = \sqrt{(p_{2x} - p_{4x})^2 + (p_{2y} - p_{4y})^2} \tag{2.35}$$

と定まる．

また，h は三角形 $P_mP_2P_3$ における三平方の定理より

$$h = \sqrt{\ell^2 - \left(\frac{d}{2}\right)^2} \tag{2.36}$$

と定まり，α_x は三角形 $P_1P_2P_4$ における余弦定理より

$$(P_1P_4)^2 = \ell^2 + d^2 - 2d\ell \cos\alpha_x$$

$$
\begin{aligned}
\Rightarrow \quad & p_{4x}^2 + p_{4y}^2 = \ell^2 + d^2 - 2d\ell\cos\alpha_x \\
\Rightarrow \quad & \alpha_x = \cos^{-1}\left(\frac{\ell^2 + d^2 - (p_{4x}^2 + p_{4y}^2)}{2d\ell}\right) \qquad (2.37)
\end{aligned}
$$

と定まる.

第 3 章　アクチュエータ，センサの動作確認

3.1　実験内容

3.1.1　2 軸ロボット実験装置におけるアクチュエータとセンサ

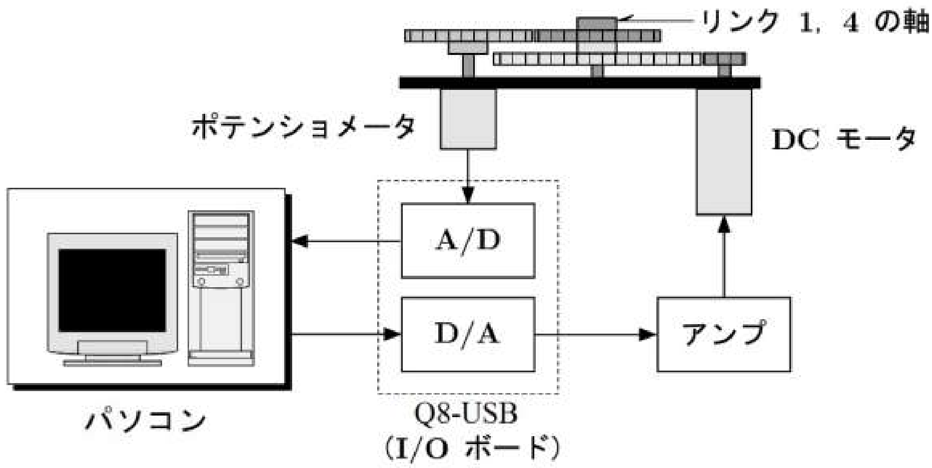


図 3.1 ：アクチュエータ（DC モータ）とセンサ（ポテンショメータ）

図 3.1 に示すように，2 軸ロボット実験装置はリンク 1，リンク 4 がギアを介して DC モータにより回転するようになっている．DC モータを駆動させるためにパソコンにより計算された指令電圧は，I/O ボード Q8-USB で D/A 変換された後，アンプを介して DC モータに入力される．また，リンク 1，リンク 4 の回転角は角度センサであるポテンショメータにより電圧値として検出され，I/O ボード Q8-USB で A/D 変換された後，パソコンに取り込まれている．

3.1.2　D/A 変換とアクチュエータの動作確認

3.1.3　実験装置のセッティングと MATLAB/Simulink の起動

図 3.2 に示すように，実験装置の 2 つの軸にそれぞれリンクを取り付ける．また，図 3.3 のようにターミナル，Universal Power Module，2 軸ロボットのケーブルが接続されていることを確認する．ただし，左側の DC モータおよびポテンショメータに対するチャネルを **CH0**（チャネル 0），右側の DC モータおよびポテンショメータに対するチャネルを **CH1**（チャネル 1）とする．

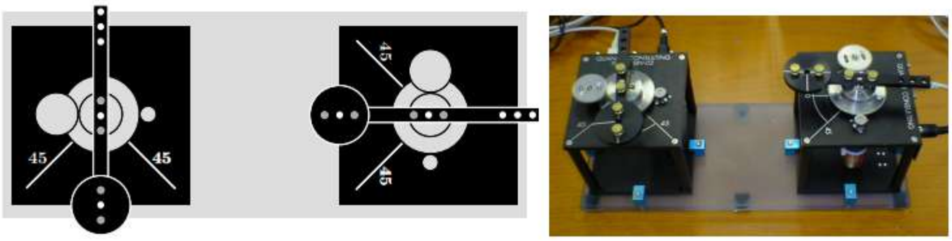


図 3.2 : 実験装置

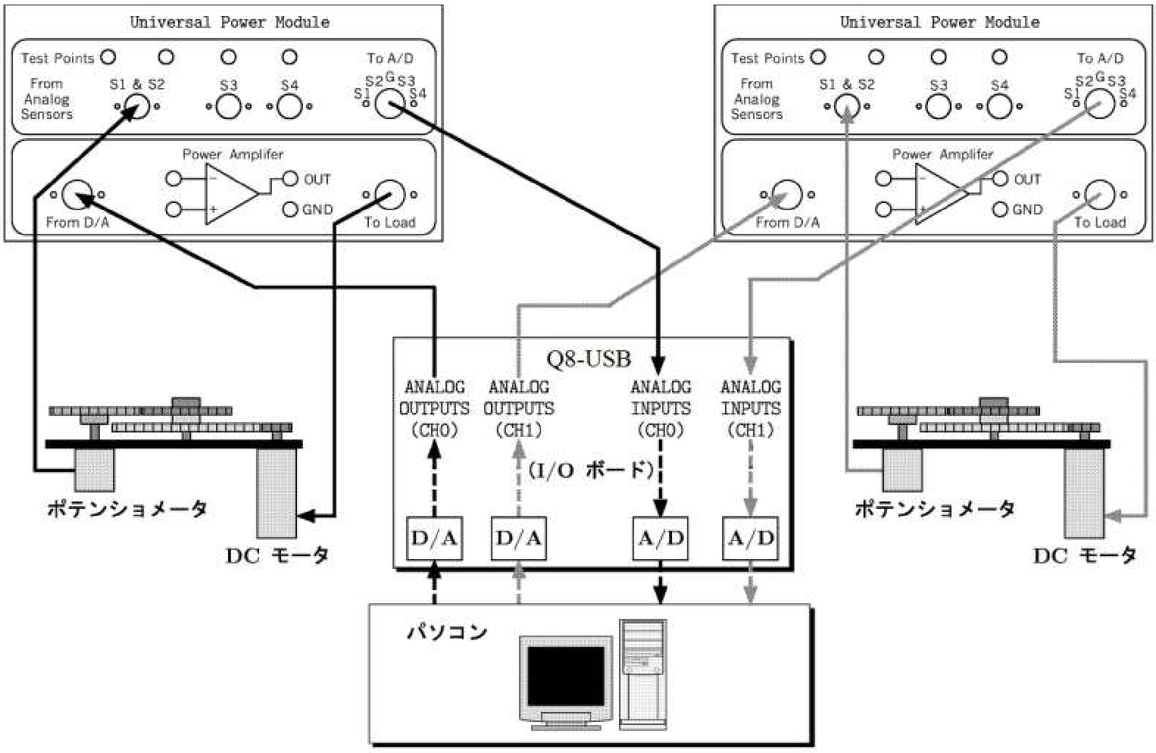


図 3.3 : 実験装置の接続

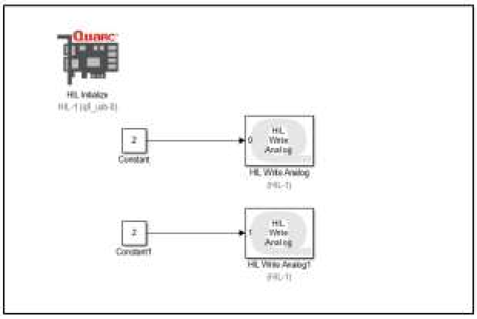
つぎに，Windows スタートメニューより MATLAB R2013a を起動する．MATLAB のコマンドウィンドウ上で以下のように入力し，カレントディレクトリ（作業するフォルダ）を変更する．（X は自分の班名に置き換える．）

```
cd D:\student_senkouka\group_X
```

3.1.4　モデルの移動と結線

左右の DC モータに 2 [V] の電圧を加えたときのリンクの動きを調べてみよう。MATLAB の“現在のフォルダー”にある，“da”フォルダーを選択し，“da_conv.slx”をダブルクリックして Simulink モデルを起動する。起動したモデルを図 3.4 のように結線する。また，各ブロックをダブルクリッ

クして，パラメータを変更する。



Constant：	定数値 を “2” に変更
Constant1：	定数値 を “2” に変更
HIL Write Analog：	Channels を “0” に変更
HIL Write Analog1：	Channels を “1” に変更

図 3.4 ： da_conv.slx

3.1.5　Simulation Parameters のパラメータ設定

Simulink モデルウィンドウのツールバーから “モデルコンフィグレーションパラメーター” を選択し，“ソルバー” を選択，図 5.1 のようにパラメータを設定する．

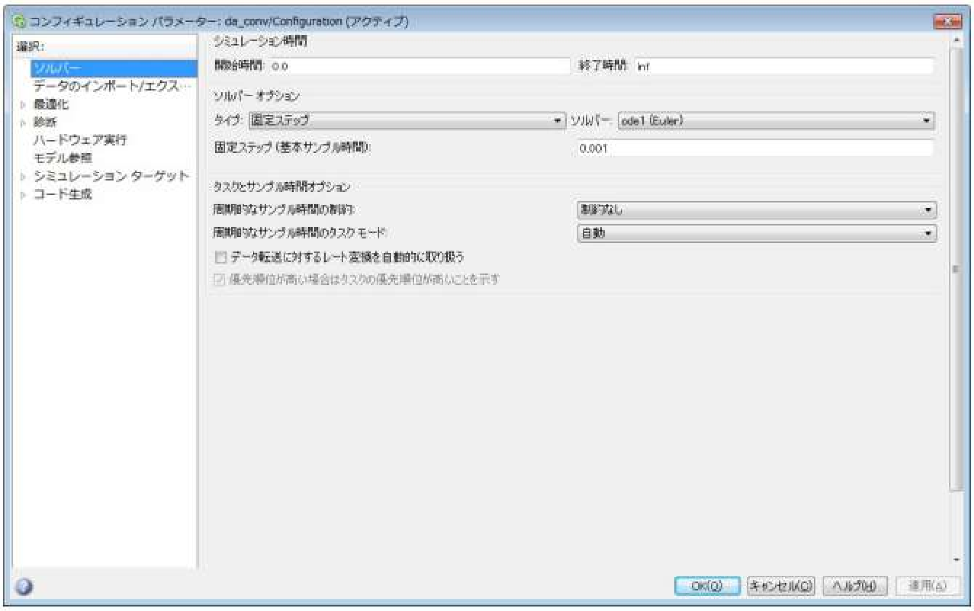


図 3.5 ： Simulation Parameters のパラメータ設定

3.1.6　コンパイル

D ドライブのディレクトリ（フォルダ）D:\student_senkouka\group_X\da（X は自分の班名）がカレントディレクトリとなっているかどうか確かめる．カレントディレクトリとなっていない場合は現在のフォルダーを操作してカレントディレクトリを変更する．

図 3.6 ： コンパイル

つぎに，“da_conv.slx” をコンパイルするためには，図 3.6 に示すように Simulink モデルのツールバーの右端にあるアイコンをクリック，モデルのビルドを選択すればよい．別の方法としては Simulink モデルのメニューから「QUARC/Build」を選択しても良い．このとき，エラーがなければ MATLAB Command Window に以下のようなメッセージが表示され，コンパイルが終了する．

```
### コードをビルドフォルダーに生成しています: D:\student_senkouka\group_xx\da\da_conv_quarc_win64
### Invoking Target Language Compiler on da_conv.rtw
### Using System Target File: C:\Program Files\Quanser\QUARC\quarc_win64.tlc
### Loading TLC function libraries
.....................《略》.....................
### Created executable da_conv.rtw-win64
### Downloading da_conv to target ’shmem://quarc-target:1’ ...
### Model da_conv has been downloaded to target ’shmem://quarc-target:1’
```

3.1.7　実験

コンパイル終了後，図 3.7 に示すように Simulink モデルのツールバーで “ターゲットに接続” アイコンをクリックし，その右のアイコンの “実行” をクリックすると，モータドライバに一定の電圧 2 [V] が加わり，リンクが一定速度で時計回りに回転することが確認できる．停止させる場合には “実行” アイコンの右にある “停止” アイコンをクリックすればよい．実行の方法として，Simulink モデルのメニューから「QUARC/Start」を選択しても実行ができる．また停止方法として Simulink モデルのメニューから「QUARC/Stop」を選択してもよい．

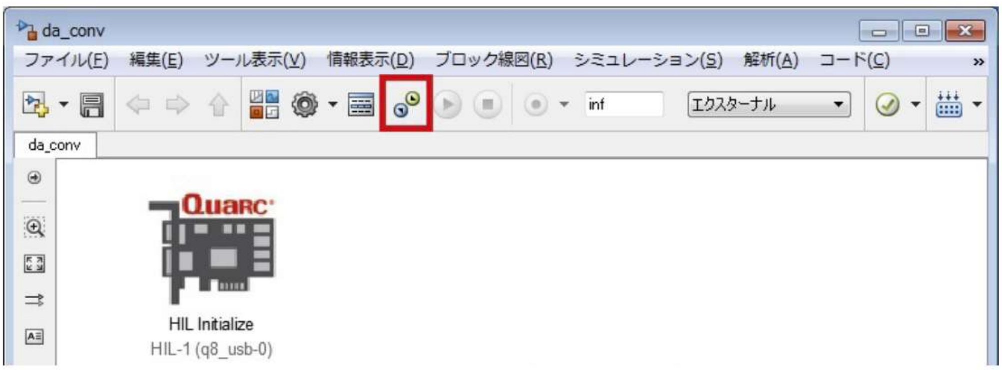
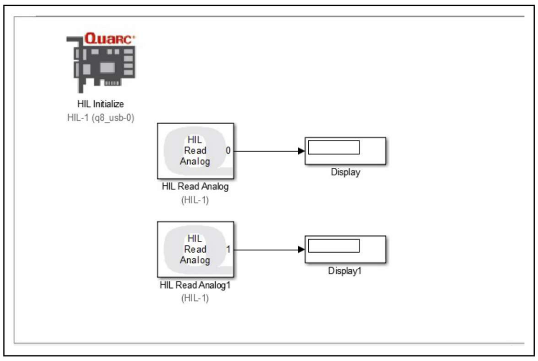


図 3.7 ： 実機実験の開始

3.1.8　A/D 変換とセンサの動作確認

本実験装置では，角度センサにポテンショメータを用いている．このポテンショメータは軸の回転角に比例した電圧 $-5 \sim +5$ [V] をアナログ信号として出力し，また，この電圧と軸の回転角との関係は，1 [V] あたり -35 [deg] である．

ポテンショメータの動作確認をするために，図 3.8 の実機実験モデル “`ad_conv.slx`” を作成する．



(`HIL Read Analog :`　`Channels` を “`0`” に変更
`HIL Read Analog1 :`　`Channels` を “`1`” に変更)

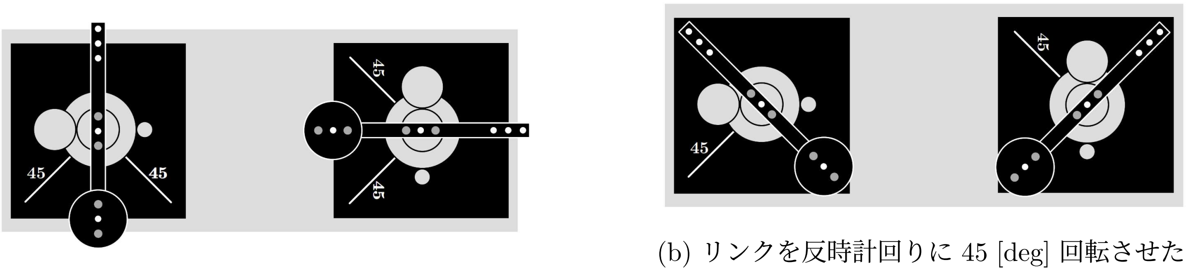
図 3.8 ： “`ad_conv.slx`”

つぎに，図 **??** で示したようにパラメータ設定を行い，図 **??** 節で示したように “`ad_conv.slx`” をコンパイルする．

以上の準備の下，リンクを図 3.9 (a) の状態にして，“Start” アイコンをクリックすると，センサからの電圧が図 3.10a (a) のように 0 [V] 程度であることが確認できる．

先に述べたように，本実験装置で用いているポテンショメータは，1 [V] あたり -35 [deg]（$= -0.6109$ [rad]）であるから，リンクを反時計回りに 45 [deg] $= \pi/4$ [rad] 回転させると -1.2857 [V] の電圧が発生するはずである．

実際，リンクを図 3.9 (b) のように反時計回りに 45 [deg] 回転させると，センサからの電圧が図 3.10b (b) のように -1.2857 [V] 程度であることが確認できる．



(a) 初期状態

(b) リンクを反時計回りに 45 [deg] 回転させた状態

図 3.9 ： リンクを手で反時計回りに 45 [deg] 回転させた状態

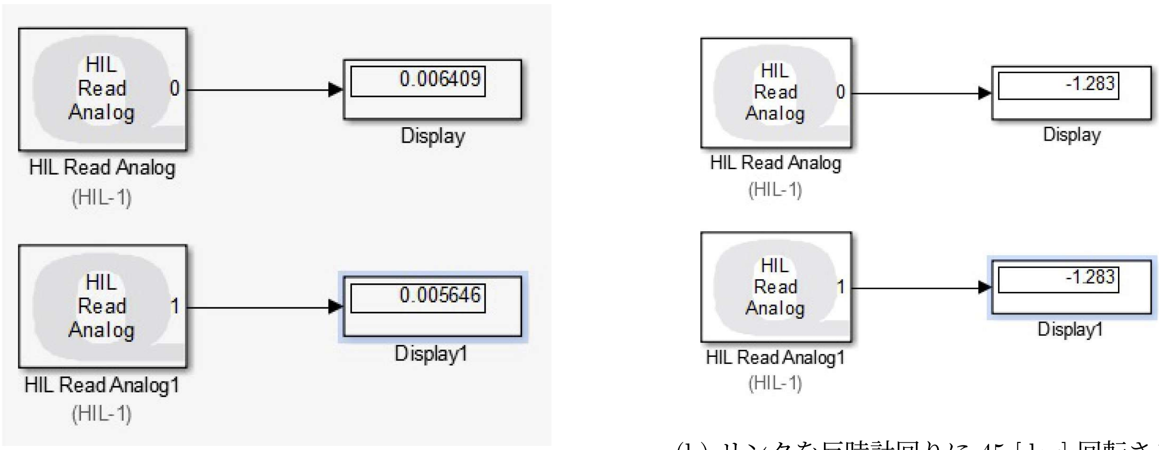
3.2　実験結果

3.2.1　D/A 変換とアクチュエータの動作確認

図 **??** に示すように，DC モータに 2 [V] の電圧を加えると，リンクが時計回りに回転し，リンクの角度 $\theta_x(t)$，$\theta_y(t)$ が弧度法の単位 [rad] で得られることが確認できた．

3.2.2　A/D 変換とセンサの動作確認

図 3.10 に示すように，リンクを反時計回りに 45 [deg] 回転させたとき，センサからの電圧が -1.2857 [V] 程度であることが確認できた．



(a) 初期状態

(b) リンクを反時計回りに 45 [deg] 回転させた状態

図 3.10 : リンクを手で反時計回りに 45 [deg] 回転させたときの出力電圧

3.3 考察

アクチュエータの動作確認では，正の電圧を DC モータに加えるとリンクが時計回りに回転することが確認できた．また，センサの動作確認では，ポテンショメータが角度に比例した電圧を出力することが確認できた．これにより，2 軸ロボットの動作が正確に制御可能であることが示された．

第 4 章　時間応答に基づくパラメータ同定

4.1　実験内容

4.1.1　パラメータ同定の手順

P コントローラ

$$v_x(t) = k_{Px}(\theta_x^{\mathrm{ref}}(t) - \theta_x(t)) \tag{4.2}$$

によりリンク 1 の関節角 $\theta_x(t)$ を制御したとき，2 軸ロボットの数学モデル（式 (2.4)）と仮定すると，$\theta_x^{\mathrm{ref}}(s)$ から $\theta_x(s)$ への伝達関数は 2 次遅れ系

$$T_x(s) = \frac{G_x(s)k_{Px}}{1 + G_x(s)k_{Px}} = \frac{b_x k_{Px}}{s^2 + a_x s + b_x k_{Px}} = \frac{\omega_{nx}^2}{s^2 + 2\zeta_x\omega_{nx}s + \omega_{nx}^2} \tag{4.3}$$

$$\begin{cases} \text{固有角周波数:} & \omega_{nx} = \sqrt{b_x k_{Px}} \\ \text{減衰係数:} & \zeta_x = \dfrac{a_x}{2\omega_{nx}} = \dfrac{a_x}{2\sqrt{b_x k_{Px}}} \end{cases} \tag{4.4}$$

となる．

(4.4) 式から明らかなように，比例ゲイン k_{Px} を大きくすると ζ_x は零に近づくため，ステップ応答はオーバーシュートが生じ，振動的になっていく．そこで，ある程度大きな比例ゲイン k_{Px} を設定し，目標値を

$$\theta_x^{\mathrm{ref}}(t) = \begin{cases} 0 & (t < 0) \\ \theta_{dx} & (t \geq 0) \end{cases} \tag{4.5}$$

としたステップ応答にオーバシュートを生じさせると，最大ピーク値 $\max\theta_x$ と行き過ぎ時間 t_{px} は以下のようになる．

$$\begin{cases} \max\theta_x = \theta_{dx}\left(1 + \exp\left(-\dfrac{\gamma_x\pi}{\delta_x}\right)\right) \\ \quad t_{px} = \dfrac{\pi}{\delta_x} \end{cases} \tag{4.6}$$

ただし，

$$\delta_x = \zeta_x\omega_{nx}, \quad \gamma_x = \omega_{nx}\sqrt{1 - \zeta_x^2} \tag{4.7}$$

以上のことから，未知パラメータ a_x，b_x の同定手順は以下のようになる．

パラメータ同定の手順

1. ある程度大きい比例ゲイン k_{Px} を設定し，P コントローラによるリンク 1 の角度制御を行う．ただし，θ_{dx} はあまり大きく設定しないようにする．たとえば，$k_{Px} = 20$，$\theta_{dx} = 0.2$ とする．
2. 1) で設定した k_{Px}，θ_{dx} を用いて実機実験を行い，$\theta_x(t)$ をデータ列 θ_x^{data} として取得する．
3. 2) で取得したデータ θ_x^{data} の最大値 $\max\theta_x^{\mathrm{data}}$ とそのときの時間 t_{px}^{data} を求める．
4. γ_x，δ_x の同定値を以下の式により導出する：

$$\gamma_x^{\mathrm{data}} = -\frac{1}{t_{px}^{\mathrm{data}}}\log_e\left(\frac{\max\theta_x^{\mathrm{data}}}{\theta_{dx}} - 1\right) \tag{4.8}$$

$$\delta_x^{\mathrm{data}} = \frac{\pi}{t_{px}^{\mathrm{data}}} \tag{4.9}$$

5. 固有角周波数 ω_{nx}，減衰係数 ζ_x の同定値を以下で定める：

$$\omega_{nx}^{\mathrm{data}} = \sqrt{(\gamma_x^{\mathrm{data}})^2 + (\delta_x^{\mathrm{data}})^2} \tag{4.10}$$

$$\zeta_x^{\mathrm{data}} = \frac{\gamma_x^{\mathrm{data}}}{\omega_{nx}^{\mathrm{data}}} \tag{4.11}$$

6. a_x，b_x の同定値 $a_x^{\rm data}$，$b_x^{\rm data}$ を次式により定める：

$$a_x^{\rm data} = 2\zeta_x^{\rm data}\omega_{nx}^{\rm data} \tag{4.12}$$

$$b_x^{\rm data} = \frac{(\omega_{nx}^{\rm data})^2}{k_{Px}} \tag{4.13}$$

なお，未知パラメータ a_y，b_y の同定手順は a_x，b_x の同定手順と同様なので，ここでは省略する．

ステップ 1：まず，MATLAB/Simulink を起動し，以下の M ファイル "`idarm.m`" と実機実験モデル "`ex_P.slx`" を作成し，フォルダ `¥ID` に保存する．

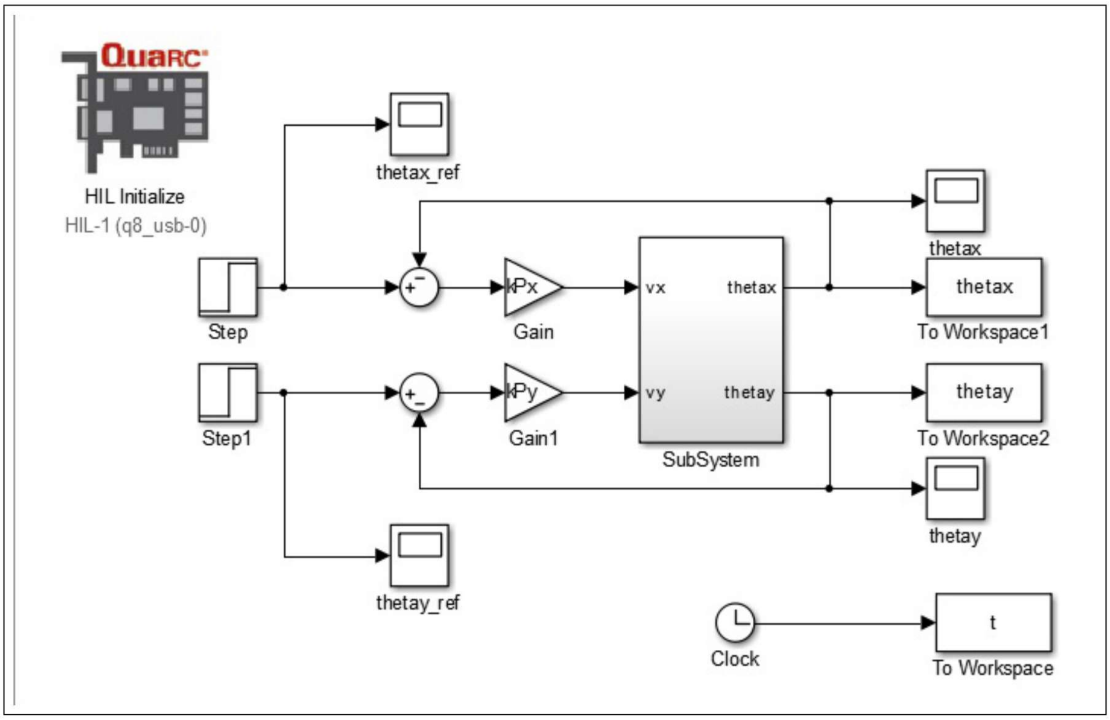


図 4.1 ：P 制御の Simulink モデル（実機実験モデル "`ex_P.slx`"）

ステップ 2：ステップ 1 の `¥ID` をカレントフォルダとして，実機実験モデル "`ex_P.slx`" をコンパイルする．

ステップ 2：ステップ 1 の `¥ID` をカレントフォルダとして，実機実験モデル "`ex_P.slx`" をコンパイルする．

ステップ 3：モータ軸の角度 $\theta_x(t)$ と $\theta_y(t)$ が 0 [rad] となるように P 制御する．実機実験モデル "`ex_P.slx`" のブロック "Step"，"Step1" をダブルクリックし，

$$\text{Step } (\theta_x^{\rm ref}(t)) \colon \begin{cases} \text{ステップ時間: } 0 \\ \text{初期値: } 0 \\ \text{最終値: } 0 \\ \text{サンプル時間: } 0 \end{cases}, \quad \text{Step1 } (\theta_y^{\rm ref}(t)) \colon \begin{cases} \text{ステップ時間: } 0 \\ \text{初期値: } 0 \\ \text{最終値: } 0 \\ \text{サンプル時間: } 0 \end{cases}$$

のように変更して，モータ軸の目標角度 $\theta_x^{\rm ref}(t)$，$\theta_y^{\rm ref}(t)$ を設定する．ターゲットに接続し，実行してアーム角度が 0 になっていることを確認する．

ステップ 4：実機実験モデル "`ex_P.slx`" のブロック "Step"，"Step1" をダブルクリックし，

$$\text{Step } (\theta_x^{\rm ref}(t)) \colon \begin{cases} \text{ステップ時間: } 0 \\ \text{初期値: } 0 \\ \text{最終値: } 0.2 \\ \text{サンプル時間: } 0 \end{cases}, \quad \text{Step1 } (\theta_y^{\rm ref}(t)) \colon \begin{cases} \text{ステップ時間: } 0 \\ \text{初期値: } 0 \\ \text{最終値: } 0.2 \\ \text{サンプル時間: } 0 \end{cases}$$

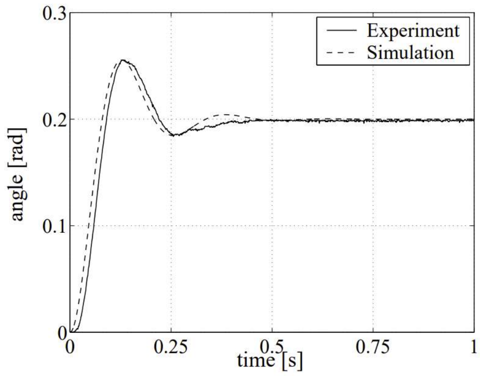
のように変更して，モータ軸の目標角度 $\theta_x^{\rm ref}(t)$，$\theta_y^{\rm ref}(t)$ を設定する．
ステップ 5： MATLAB Command Window で M ファイル "`idarm.m`" を実行する．

4.2 実験結果

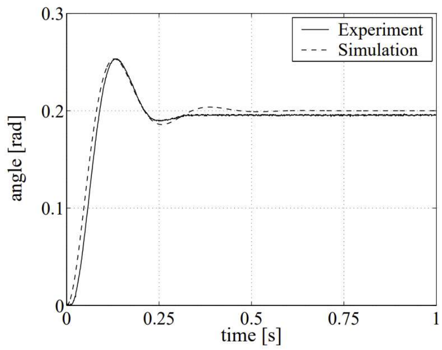
```
>> idarm
ax = 40.9262
bx = 76.7858
ay = 42.1695
by = 75.7744
```

```
a_x = 40.9262
b_x = 76.7858
a_y = 42.1695
b_y = 75.7744
```

図 4.2 に示すように，実験データと同定された値を用いたシミュレーションデータが一致していることが確認できた．



(a) $\theta_x(t)$ の応答 ($\theta_x^{\rm ref}(t) = 0.2$, $\theta_y^{\rm ref}(t) = 0$)



(b) $\theta_y(t)$ の応答 ($\theta_x^{\rm ref}(t) = 0$, $\theta_y^{\rm ref}(t) = 0.2$)

図 4.2 ：P コントローラを用いたときのステップ応答 ($k_{Px} = 20$, $k_{Py} = 20$)

4.3 考察

実験結果より，未知パラメータ a_x，b_x，a_y，b_y を正確に同定できたことが確認できた．しかし，定常偏差が残っていることが確認できた．これは，P コントローラの特性に起因するものである．

第 5 章　角度制御

5.1　実験内容

5.1.1　P 制御

5.1.2　パラメータ設定

4 章で述べたように，P コントローラ

$$\begin{aligned} v_x(t) &= k_{Px}e_x(t), \quad e_x(t) = \theta_x^{\mathrm{ref}}(t) - \theta_x(t) \\ v_y(t) &= k_{Py}e_y(t), \quad e_y(t) = \theta_y^{\mathrm{ref}}(t) - \theta_y(t) \end{aligned} \tag{5.1}$$

を用いると，$\theta_x^{\mathrm{ref}}(s)$ から $\theta_x(s)$ への伝達関数は 2 次遅れ要素

$$\begin{cases} T_x(s) = \dfrac{\omega_{nx}^2}{s^2 + 2\zeta_x\omega_{nx}s + \omega_{nx}^2} \\ T_y(s) = \dfrac{\omega_{ny}^2}{s^2 + 2\zeta_y\omega_{ny}s + \omega_{ny}^2} \end{cases} \tag{5.2}$$

となる．ただし，

固有角周波数: $\omega_{nx} = \sqrt{b_x k_{Px}}, \quad \omega_{ny} = \sqrt{b_y k_{Py}}$
減衰係数: $\zeta_x = \dfrac{a_x}{2\omega_{nx}} = \dfrac{a_x}{2\sqrt{b_x k_{Px}}}, \quad \zeta_y = \dfrac{a_y}{2\omega_{ny}} = \dfrac{a_y}{2\sqrt{b_y k_{Py}}}$

である．したがって，P コントローラで指定できるのは，固有角周波数 ω_{nx}，ω_{ny} か減衰係数 ζ_x，ζ_y のいずれかである．

たとえば，固有角周波数 ω_{nx}，ω_{ny} を指定した値 ω_{Mx}，ω_{My} とするとには，比例ゲイン k_{Px}，k_{Py} を

$$\begin{cases} k_{Px} = \dfrac{\omega_{Mx}^2}{b_x} \\ k_{Py} = \dfrac{\omega_{My}^2}{b_y} \end{cases} \tag{5.4}$$

とすればよいが，ω_{Mx}, ω_{My} を大きくするにつれて比例ゲイン k_{Px}，k_{Py} が大きくなり，減衰係数 ζ_x, ζ_y を零に近づけてしまうため，振動的な応答となってしまう．

また，減衰係数 ζ_x, ζ_y を指定した値（たとえば ζ_{Mx}, ζ_{My}）とするとには，比例ゲイン k_{Px}，k_{Py} を

$$\begin{cases} k_{Px} = \dfrac{a_x^2}{4\zeta_{Mx}^2 b_x} \\ k_{Py} = \dfrac{a_y^2}{4\zeta_{My}^2 b_y} \end{cases} \tag{5.5}$$

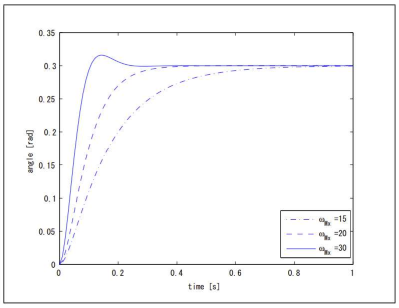
とすればよいが，固有角周波数 ω_{nx}, ω_{ny} が決まってしまい，速応性を考慮することはできない．

5.1.3　シミュレーションと実験

ステップ 1：MATLAB/Simulink を起動し，カレントディレクトリを `¥Pcont` に移動する．つぎに，4.3 節で定めたパラメータを保存した MAT ファイル “`armpara.mat`”，(5.4) 式にしたがって P コントローラを設計する M ファイル “`armP.m`”，シミュレーションモデル “`sim_P.slx`”（図??），実機実験モデル “`ex_P.slx`”（図 5.1）を作成し，`C:\robot\Pcont` に保存する．ただし，目標値は

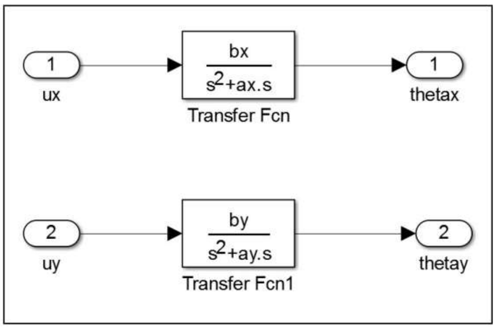
$$\mathrm{Step}\left(\theta_x^{\mathrm{ref}}(t)\right):\begin{cases}\text{ステップ時間}:0\\\text{初期値}:0\\\text{最終値}:0.3\\\text{サンプル時間}:0\end{cases},\quad \mathrm{Step1}\left(\theta_y^{\mathrm{ref}}(t)\right):\begin{cases}\text{ステップ時間}:0\\\text{初期値}:0\\\text{最終値}:0\\\text{サンプル時間}:0\end{cases}\tag{5.6}$$

と設定する．




```
モデルウィンドウのメニューで
     “シミュレーション/モデル コンフィグ...”
 を選択し，
     開始時間:  0
     終了時間:  1
     ソルバー, ode4 (Runge-Kutta)
     固定ステップ:  0.01
 に変更
ブロック “To Workspace (2,3,4,5,6)”
 をダブルクリックし，
     保存形式:  配列
 に変更
```

(a) シミュレーションモデル “sim_P.slx”



(b) Subsystem “2D Robot Model” の内容

図 5.1 ：P 制御の Simulink モデル

モデルウィンドウのメニューで “シミュレーション／モデル コンフィグ...” を選択し，

- 開始時間：0
- 終了時間：1
- ソルバー：ode4 (Runge-Kutta)
- 固定ステップ：0.01

に変更．ブロック “To Workspace (2,3,4,5,6)” をダブルクリックし，保存形式を **配列** に変更する．

ステップ 2 ：“`armP_sim.m`” を実行し，図 5.1 (a) のシミュレーション結果を取得する．

ステップ 3 ：$\omega_{Mx}=15$，$\omega_{My}=15$ としたときの P コントローラを設計するため，M ファイル “`armP.m`” を実行する．その結果，以下の実行結果が得られる．

armP.m の実行結果

```
>> armP
各パラメータを設定して下さい
omegaMx = 15
omegaMy = 15
kPx =
    2.9302
kPy =
    2.9693
```

つぎに，設計された P コントローラを用いて，“ex_p.slx” を実行し，角度制御の実機実験を行う．得られた MAT ファイル “thetax.mat” の名前を “thetax15.mat” に変更する．

ステップ 4：アームの角度を 0 にし，ステップ 3 において $\omega_{Mx} = 20$，$\omega_{My} = 20$ と指定し，同様の作業を行う．ただし，得られた MAT ファイル “thetax.mat” の名前を “thetax20.mat” に変更する．

ステップ 5：アームの角度を 0 にし，ステップ 3 において $\omega_{Mx} = 30$，$\omega_{My} = 30$ と指定し，同様の作業を行う．ただし，得られた MAT ファイル “thetax.mat” の名前を “thetax30.mat” に変更する．

ステップ 6：MATLAB Command Window で M ファイル “plot_figure.m” を実行する．

5.1.4 P–D 制御

5.1.5 パラメータ設定

P–D コントローラ（微分先行型 PD コントローラ）

$$\begin{cases} v_x(t) = k_{Px}e_x(t) - k_{Dx}\dfrac{d\theta_x(t)}{dt} \\ v_y(t) = k_{Py}e_y(t) - k_{Dy}\dfrac{d\theta_y(t)}{dt} \end{cases} \iff \begin{cases} v_x(s) = k_{Px}e_x(s) - k_{Dx}s\theta_x(s) \\ v_y(s) = k_{Py}e_y(s) - k_{Dy}s\theta_y(s) \end{cases} \tag{5.7}$$

を用いると，$\theta_x^{\mathrm{ref}}(s)$ から $\theta_x(s)$ への伝達関数は 2 次遅れ要素

$$\begin{cases} T_x(s) = \dfrac{\omega_{nx}^2}{s^2 + 2\zeta_x\omega_{nx}s + \omega_{nx}^2} \\ T_y(s) = \dfrac{\omega_{ny}^2}{s^2 + 2\zeta_y\omega_{ny}s + \omega_{ny}^2} \end{cases} \tag{5.8}$$

となる．ただし，

$$\begin{cases} \text{固有角周波数:} \quad \omega_{nx} = \sqrt{b_x k_{Px}}, \quad \omega_{ny} = \sqrt{b_y k_{Py}} \\ \text{減衰係数:} \quad \zeta_x = \dfrac{a_x + b_x k_{Dx}}{2\omega_{nx}}, \quad \zeta_y = \dfrac{a_y + b_y k_{Dy}}{2\omega_{ny}} \end{cases} \tag{5.9}$$

である．したがって，P–D 制御では比例ゲイン k_{Px}, k_{Py} により速応性に関するパラメータ ω_{nx}, ω_{ny} を指定し，微分ゲイン k_{Dx}, k_{Dy} により減衰性に関するパラメータ ζ_x, ζ_y を指定することができる．つまり，固有角周波数 ω_{nx}, ω_{ny} および減衰係数 ζ_x, ζ_y を指定した値 $\omega_{Mx}, \omega_{My}, \zeta_{Mx}, \zeta_{My}$ とするには，

$$\begin{cases} k_{Px} = \dfrac{\omega_{Mx}^2}{b_x}, \quad k_{Dx} = \dfrac{2\zeta_{Mx}\omega_{Mx} - a_x}{b_x} \\ k_{Py} = \dfrac{\omega_{My}^2}{b_y}, \quad k_{Dy} = \dfrac{2\zeta_{My}\omega_{My} - a_y}{b_y} \end{cases} \tag{5.10}$$

なお，ポテンショメータによって検出された角度には高周波成分の観測雑音（ノイズ）が含まれているため，検出された角度をもとに角速度を算出すると，インパルス状の成分を含んでしまう．そこで，実際には，検出された角度を 1 次のローパスフィルタ

$$G_{fx}(s) = \frac{1}{1 + T_{dx}s}, \quad G_{fy}(s) = \frac{1}{1 + T_{dy}s} \tag{5.1}$$

に通して高周波成分の観測雑音を除去した後，角度信号を微分する必要がある．以上のことを考慮すると，P–D コントローラは次式のようになる．

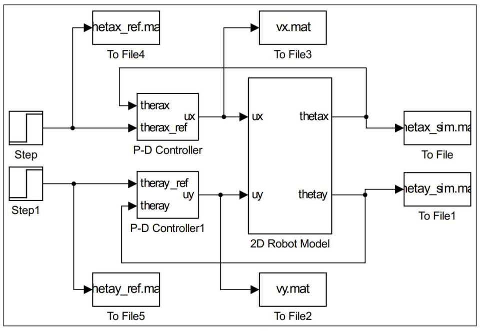
$$\begin{cases} v_x(s) = k_{Px}e_x(s) - \dfrac{k_{Dx}s}{1+T_{dx}s}\theta_x(s) \\ v_y(s) = k_{Py}e_y(s) - \dfrac{k_{Dy}s}{1+T_{dy}s}\theta_y(s) \end{cases} \tag{5.11}$$

5.1.6　シミュレーションと実験

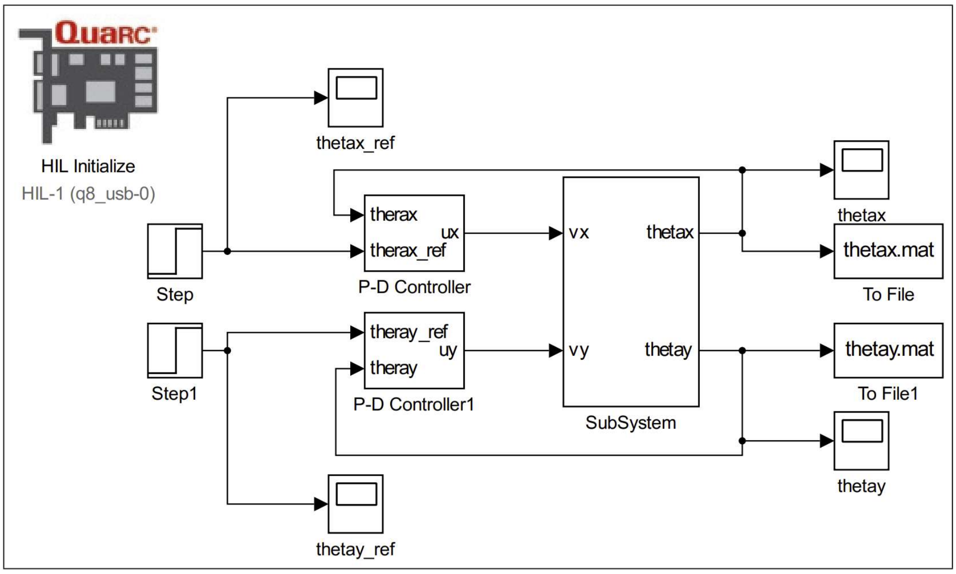
ステップ 1：パラメータを記述した M ファイル “armpara.m” および (5.11) 式にしたがって P–D コントローラのパラメータを設計する M ファイル “armPD.m”，シミュレーションモデル “sim_PD.slx”（図 5.3 (a)），実機実験モデル “ex_PD.slx”（図 5.3 (b)）を作成し，C:\robot\PDcont に保存する．

ステップ 2：アームの角度を 0 にし，$\omega_{Mx} = 30$, $\zeta_{Mx} = 0.7$ と指定し，実験する．ただし，得られた MAT ファイル “`thetax.mat`” の名前を “`thetax7.mat`” に変更する．

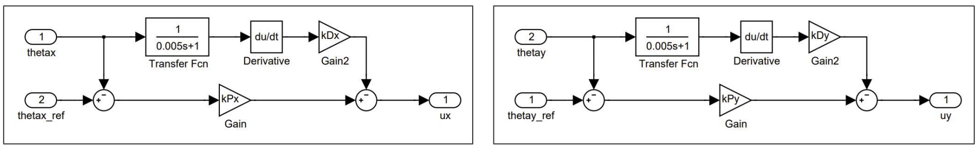
ステップ 3：アームの角度を 0 にし，$\omega_{Mx} = 30$, $\zeta_{Mx} = 1.0$ と指定し，実験する．ただし，得られた MAT ファイル “`thetax.mat`” の名前を “`thetax1.mat`” に変更する．



(a) シミュレーションモデル “sim_PD.slx”



(b) 実機実験モデル “ex_PD.slx”



(c) Subsystem “PD Controller”, “PD Controller1” の内容

図 5.2 ：P–D 制御の Simulink モデル

5.1.7 I–PD 制御

5.1.8 パラメータ設定

P 制御や P–D 制御はコントローラに積分器 $1/s$ を含んでいないため，クローン摩擦の影響でステップ応答に定常偏差が生じた．そこで，コントローラに積分器 $1/s$ を含ませることによって，ステップ状の目標値に対する定常偏差を解消することを考える．

I–PD コントローラ（比例・微分先行型 PID コントローラ）

$$
\begin{cases} v_x(t) = -k_{Px}\theta_x(t) + k_{Ix}\displaystyle\int_0^t e_x(\tau)d\tau - k_{Dx}\frac{d\theta_x(t)}{dt} \\ v_y(t) = -k_{Py}\theta_y(t) + k_{Iy}\displaystyle\int_0^t e_y(\tau)d\tau - k_{Dy}\frac{d\theta_y(t)}{dt} \end{cases} \Longleftrightarrow \begin{cases} v_x(s) = -k_{Px}\theta_x(s) + \dfrac{k_{Ix}}{s}e_x(s) - k_{Dx}s\theta_x(s) \\ v_y(s) = -k_{Py}\theta_y(s) + \dfrac{k_{Iy}}{s}e_y(s) - k_{Dx}s\theta_y(s) \end{cases} \tag{5.12}
$$

を用いると，$\theta_x^{ref}(s)$ から $\theta_x(s)$ への伝達関数は 3 次遅れ要素

$$
\begin{cases} T_x(s) = \dfrac{b_x k_{Ix}}{s^3 + (a_x + b_x k_{Dx})s^2 + b_x k_{Px} s + b_x k_{Ix}} \\ T_y(s) = \dfrac{b_y k_{Iy}}{s^3 + (a_y + b_y k_{Dx})s^2 + b_y k_{Py} s + b_y k_{Iy}} \end{cases} \tag{5.13}
$$

となる．したがって，(5.13) 式を規範モデル

$$
\begin{cases} T_{Mx}(s) = \dfrac{\omega_{Mx}^3}{s^3 + \alpha_{Mx2}\omega_{Mx}^2 s^2 + \alpha_{Mx1}\omega_{Mx} s + \omega_{Mx}^3} \\ T_{My}(s) = \dfrac{\omega_{My}^3}{s^3 + \alpha_{My2}\omega_{My}^2 s^2 + \alpha_{My1}\omega_{My} s + \omega_{My}^3} \end{cases} \tag{5.14}
$$

と完全に一致させるには

$$
\begin{cases} k_{Ix} = \dfrac{\omega_{Mx}^3}{b_x}, \quad k_{Px} = \dfrac{\alpha_{Mx1}\omega_{Mx}^2}{b_x}, \quad k_{Dx} = \dfrac{\alpha_{Mx2}\omega_{Mx} - a_x}{b_x} \\ k_{Iy} = \dfrac{\omega_{My}^3}{b_y}, \quad k_{Py} = \dfrac{\alpha_{My1}\omega_{My}^2}{b_y}, \quad k_{Dx} = \dfrac{\alpha_{My2}\omega_{My} - a_y}{b_y} \end{cases} \tag{5.15}
$$

と選べばよい．ただし，ω_{Mx}, ω_{My} は速度応答に関するパラメータ，$\alpha_{M1x}, \alpha_{M2x}, \alpha_{M1y}, \alpha_{M2y}$ は減衰性に関するパラメータであり，

規範モデルの標準形

- **バターワース標準形**：

$$
\begin{cases} \alpha_{M1x} = 2, \quad \alpha_{M2x} = 2 \\ \alpha_{M1y} = 2, \quad \alpha_{M2y} = 2 \end{cases}
$$

- **二項標準形**：

$$
\begin{cases} \alpha_{M1x} = 3, \quad \alpha_{M2x} = 3 \\ \alpha_{M1y} = 3, \quad \alpha_{M2y} = 3 \end{cases}
$$

- **ITAE 最小標準形**：

$$
\begin{cases} \alpha_{M1x} = 2.15, \quad \alpha_{M2x} = 1.75 \\ \alpha_{M1y} = 2.15, \quad \alpha_{M2y} = 1.75 \end{cases}
$$

が用いられることが多い．

なお，実際には高周波成分の観測雑音を除去するため，次式の I–PD コントローラを用いることになる：

$$
\begin{cases} v_x(s) = -k_{Px}\theta_x(s) + \dfrac{k_{Ix}}{s}e_x(s) - \dfrac{k_{Dx}s}{1 + T_{dx}s}\theta_x(s) \\ v_y(s) = -k_{Py}\theta_y(s) + \dfrac{k_{Iy}}{s}e_y(s) - \dfrac{k_{Dy}s}{1 + T_{dy}s}\theta_y(s) \end{cases} \tag{5.16}
$$

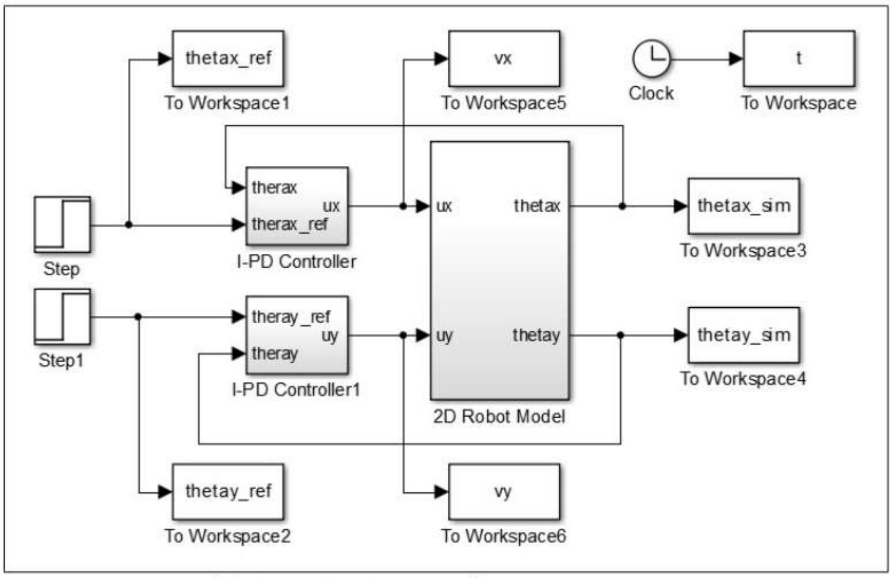
5.1.9　シミュレーションと実験

ステップ 1：パラメータを記述した M ファイル “armpara.m” および (5.15) 式により I–PD コントローラのパラメータを設計する M ファイル “armIPD.m”，シミュレーションモデル “sim_IPD.slx”（図 5.5 (a)），実機実験モデル “ex_IPD.slx”（図 5.5 (b)）を作成し，C:\robot\IPDcont に保存する．

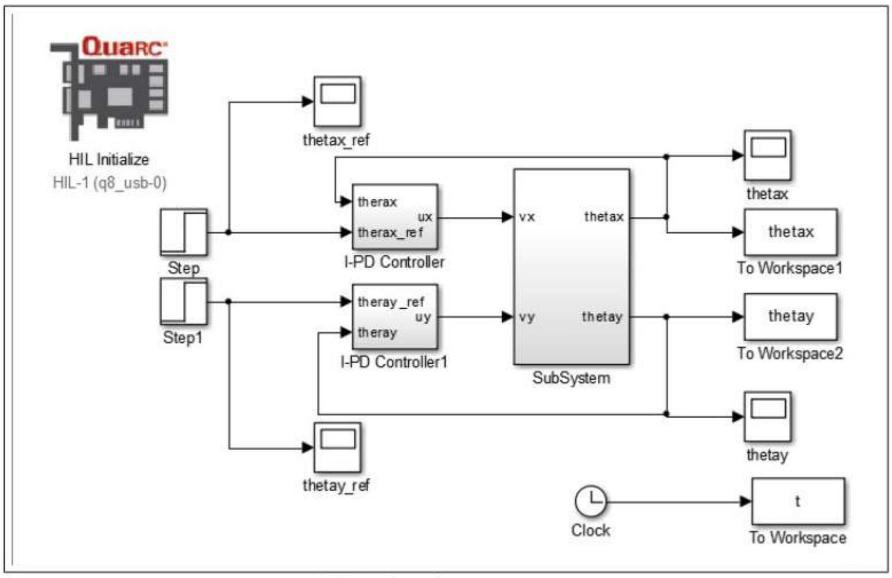
ステップ 2：アームの角度を 0 にし，$\omega_{Mx} = 30$, $\alpha_{M1x} = 2$, $\alpha_{M2x} = 2$ と指定し，実験する．ただし，得られた MAT ファイル “thetax.mat” の名前を “thetaxb.mat” に変更する．

ステップ 3 ：アームの角度を 0 にし，$\omega_{Mx} = 30$, $\alpha_{M1x} = 3$, $\alpha_{M2x} = 3$ と指定し，実験する．ただし，得られた MAT ファイル “thetax.mat” の名前を “thetax2.mat” に変更する．

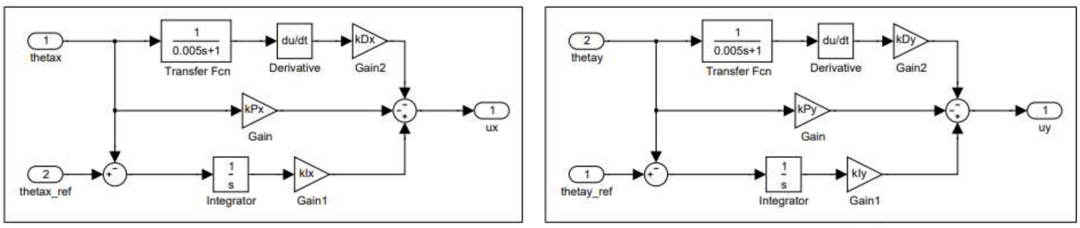
ステップ 4 ：アームの角度を 0 にし，$\omega_{Mx} = 30$, $\alpha_{M1x} = 2.15$, $\alpha_{M2x} = 1.75$ と指定し，実験する．ただし，得られた MAT ファイル “thetax.mat” の名前を “thetaxi.mat” に変更する．



(a) シミュレーションモデル “`sim_IPD.slx`”



(b) 実機実験モデル “`ex_IPD.slx`”



(c) `Subsystem` “`I-PD Controller`”, “`I-PD Controller1`” の内容

図 5.3 ：I–PD 制御の Simulink モデル

5.2　実験結果

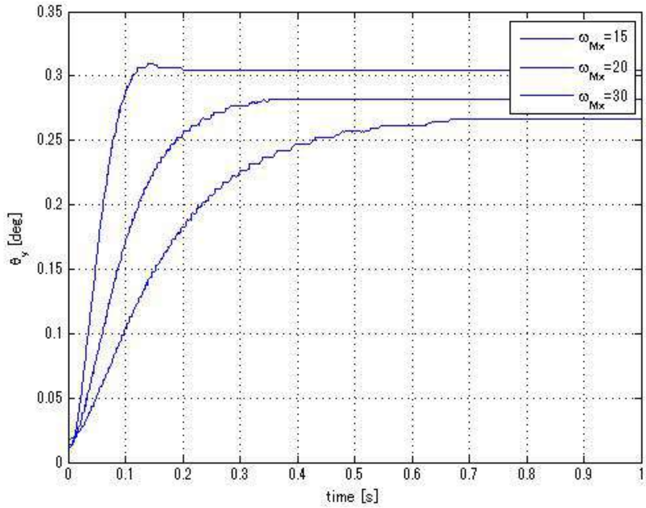


図 5.4 : P 制御の応答

図 5.4 に示すように，P 制御では速応性が改善される一方でオーバーシュートが大きくなることが確認された.

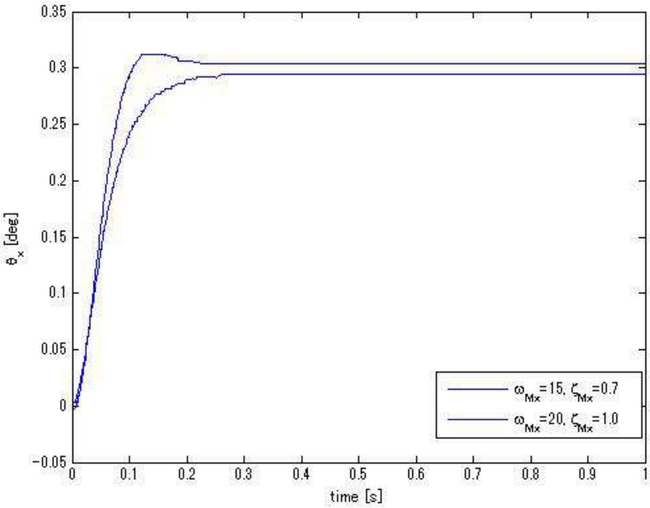


図 5.5 : P–D 制御の応答

また，図 5.5 に示すように，P–D 制御では安定度が改善され，たが定常偏差が残るようになった.

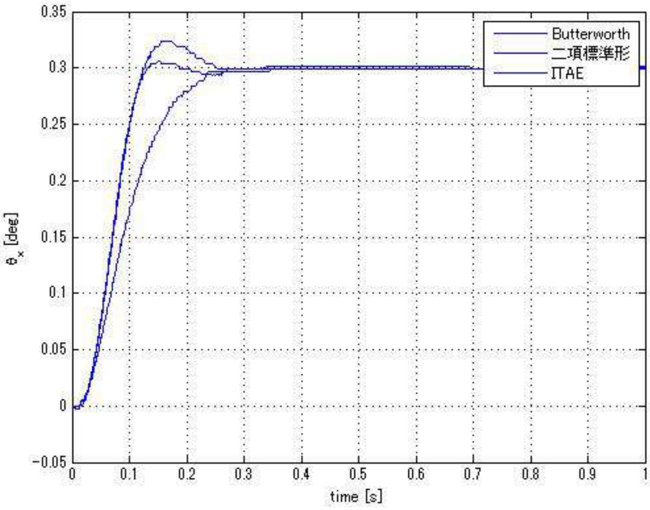


図 5.6 : I–PD 制御の応答

図 5.6 に示すように，I–PD 制御では定常偏差が解消された．

5.3 考察

P 制御では速応性が向上するが，オーバーシュートが大きくなるため，減衰係数の調整が必要である．P–D 制御では安定度が向上し，振動が抑制された．さらに，I–PD 制御では定常偏差が解消され，目標値への追従性が向上した．

第 6 章　手先位置制御

6.1　順運動学と逆運動学

ここでは，手先位置 $(x(t), y(t))$ をその目標値 $(x^{\rm ref}(t), y^{\rm ref}(t))$ に追従させる制御を考える．

手先 P_3 の座標 $(x(t), y(t))$ を目標値 $(x^{\rm ref}(t), y^{\rm ref}(t))$ に追従させるためには，2.3 節で述べたように，逆運動学により角度の目標値 $\theta_x^{\rm ref}(t), \theta_y^{\rm ref}(t)$ を

$$\begin{cases}\theta_x^{\rm ref}(t) = \sin^{-1}\left(\frac{\sqrt{x^{\rm ref}(t)^2 + y^{\rm ref}(t)^2}}{2\ell}\right) + \tan^{-1}\left(\frac{y^{\rm ref}(t)}{x^{\rm ref}(t)}\right) \\ \theta_y^{\rm ref}(t) = \sin^{-1}\left(\frac{\sqrt{(x^{\rm ref}(t)-2\ell)^2 + y^{\rm ref}(t)^2}}{2\ell}\right) - \tan^{-1}\left(\frac{x^{\rm ref}(t)-2\ell}{y^{\rm ref}(t)}\right)\end{cases} \tag{6.1}$$

と設定すればよい．

たとえば I–PD コントローラ（比例・微分先行型 PID コントローラ）

$$\begin{cases}v_x(t) = -k_{Px}\theta_x(t) + \frac{k_{Ix}}{s}e_x(t) - k_{Dx}\frac{d\theta_x(t)}{dt} \\ v_y(t) = -k_{Py}\theta_y(t) + \frac{k_{Iy}}{s}e_y(t) - k_{Dy}\frac{d\theta_y(t)}{dt}\end{cases} \tag{6.2}$$

を用いると，$\theta_x(t), \theta_y(t)$ をその目標値に追従させることができ，その結果，2.2 節で述べたように，順運動学により手先位置 $(x(t), y(t))$ は次式のようになる：

$$\begin{cases}x(t) = h(t)\cos(\theta_x(t) - \alpha_x(t)) - \frac{d(t)}{2}\sin(\theta_x(t) - \alpha_x(t)) + \ell\sin\theta_x(t) \\ y(t) = h(t)\sin(\theta_x(t) - \alpha_x(t)) + \frac{d(t)}{2}\cos(\theta_x(t) - \alpha_x(t)) - \ell\cos\theta_x(t)\end{cases} \tag{6.3}$$

ただし，

$$\begin{cases}d(t) = \sqrt{(p_{2x}(t) - p_{4x}(t))^2 + (p_{2y}(t) - p_{4y}(t))^2} \\ h(t) = \sqrt{\ell^2 - \left(\frac{d(t)}{2}\right)^2} \\ \alpha_x(t) = \cos^{-1}\left(\frac{\ell^2 + d(t)^2 - (p_{4x}(t)^2 + p_{4y}(t)^2)}{2\ell d}\right) \\ p_{2x}(t) = \ell\sin\theta_x(t), \quad p_{2y}(t) = -\ell\cos\theta_x(t) \\ p_{4x}(t) = 2\ell - \ell\cos\theta_y(t), \quad p_{4y}(t) = -\ell\sin\theta_y(t)\end{cases} \tag{6.4}$$

である．

図 6.1 に手先位置 $(x(t), y(t))$ をその目標値 $(x^{\rm ref}(t), y^{\rm ref}(t))$ に追従させるフィードバック制御系のブロック線図を示す．

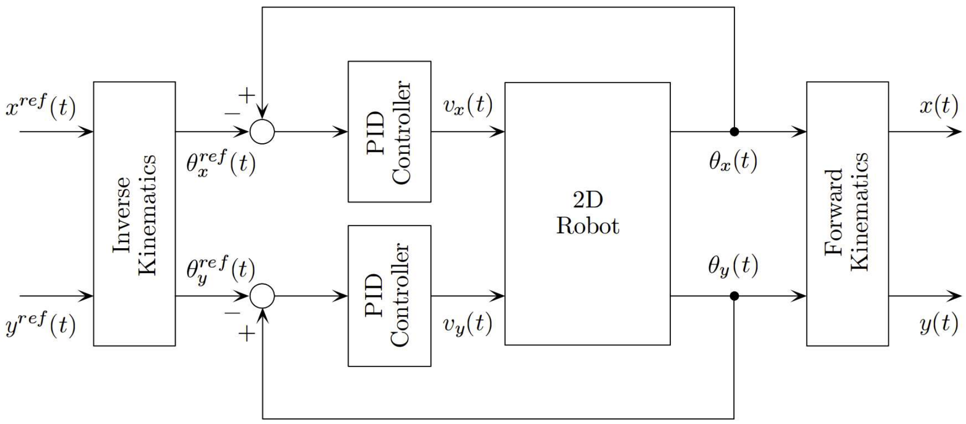
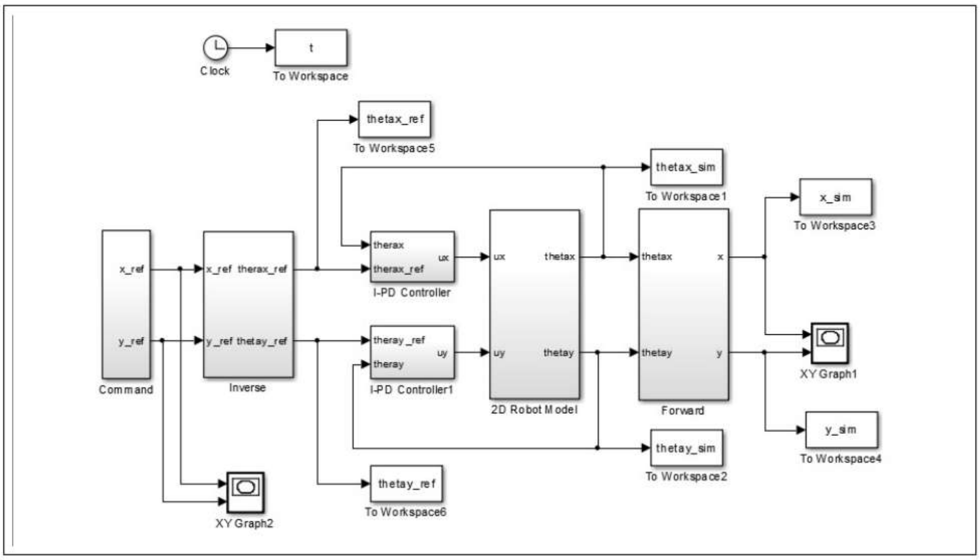


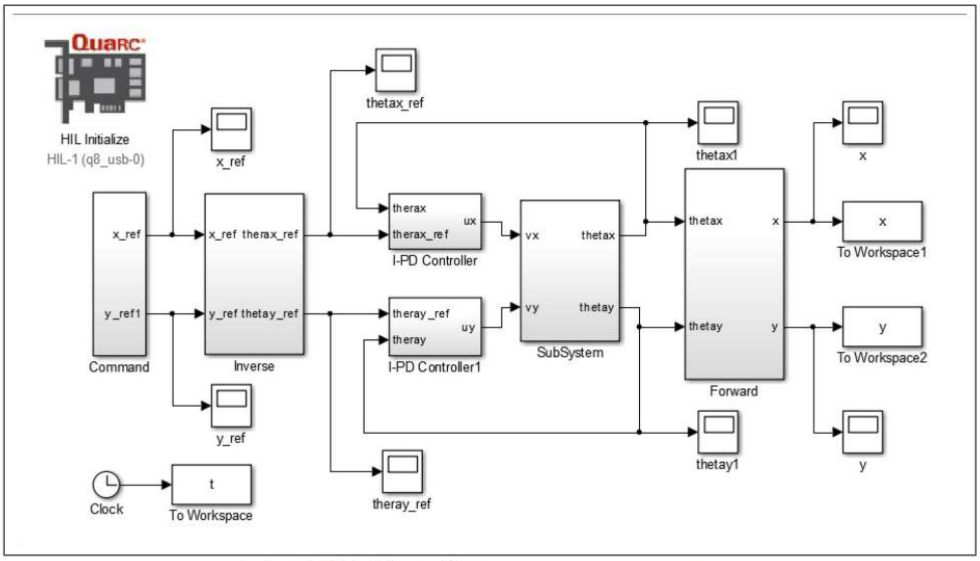
図 6.1 ：フィードバック制御系

6.2　シミュレーションと実験（目標値が一定値の場合）

ステップ 1 ：5.1.2 節で示した “armpara.m”，5.3.2 節で示した “armIPD.m” およびシミュレーション結果をアニメーション表示する以下の M ファイル “sim_anime.m” を作成し，\xy に保存する．また，シミュレーションモデル “sim_robot_xy.slx”（図 6.2(a)），実機実験モデル “ex_robot_xy.slx”（図 6.2(b)）を作成し，\xy に保存する．ただし，各 `Subsystem` の内容は以下に示すとおりである．

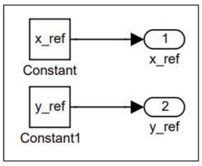


(a) シミュレーションモデル“`sim_robot_xy.slx`”

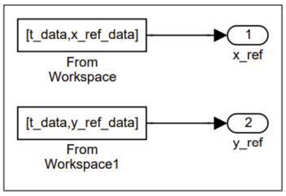


(b) 実機実験モデル“`ex_robot_xy.slx`”

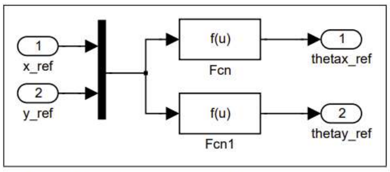
図 6.2：手先位置制御（目標値が一定値の場合）



(a1)　Subsystem “Command” の内容
(‘sim_robot_xy.slx”, “ex_robot_xy.slx”)

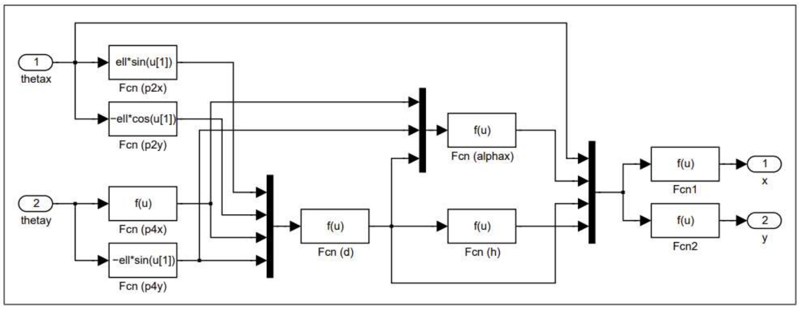


(a2)　Subsystem “Command” の内容
(“sim_robot_xy2.slx”, “ex_robot_xy2.slx”)



(Fcn :　Expression を “asin(sqrt(u[1]^2+u[2]^2)/(2*ell)) + atan(u[2]/u[1])” に変更
Fcn1 :　Expression を “asin(sqrt((u[1]-2*ell)^2+u[2]^2)/(2*ell)) - atan((u[1]-2*ell)/u[2])” に変更)

(b)　Subsystem “Inverse” の内容



(Fcn (p2x) :　Expression を “ell*sin(u[1])” に変更
Fcn (p2y) :　Expression を “-ell*cos(u[1])” に変更
Fcn (p4x) :　Expression を “2*ell-ell*cos(u[1])” に変更
Fcn (p4y) :　Expression を “-ell*sin(u[1])” に変更
Fcn (alphax) :　Expression を “acos((ell^2 + u[3]^2 - (u[1]^2 + u[2]^2))/(2*ell*u[3]))” に変更
Fcn (d) :　Expression を “sqrt((u[1]-u[3])^2+(u[2]-u[4])^2)” に変更
Fcn (h) :　Expression を “sqrt(ell^2 - (u[1]/2)^2)” に変更
Fcn1 :　Expression を “u[4]*cos(u[1]-u[2])-(u[3]/2)*sin(u[1]-u[2])+ell*sin(u[1])” に変更
Fcn2 :　Expression を “u[4]*sin(u[1]-u[2])+(u[3]/2)*cos(u[1]-u[2])-ell*cos(u[1])” に変更)

(c)　Subsystem “Forward” の内容

図 6.3 ：Subsystem “Command”, “Inverse”, “Forward” の構成（図 6.3）

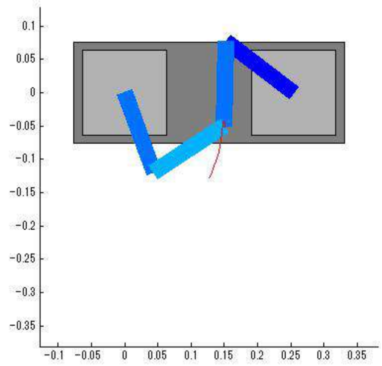
ステップ 2 ：まず，MATLAB Command Window で

```
>> armIPD
各パラメータを設定して下さい
omegaMx = 20
omegaMy = 20
  …（略）…
alphaM1x = 2
alphaM2x = 2
alphaM1y = 2
alphaM2y = 2
  …（略）…
x_ref = 0.15; y_ref = -0.05;
```

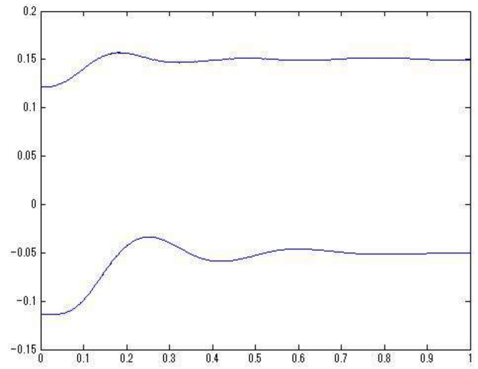
と入力し，パラメータを設定する．つぎに，シミュレーションを開始し，シミュレーション終了後に以下のように入力して結果をアニメーション表示する．

```
>> sim_anime;
```

アニメーション（図 6.4 (a)）によって各リンクに無理な動きがないことを確認した後，“ex_robot_xy.slx” をコンパイルしてから実機実験を行うと，図 6.4 (b) の実験結果が得られる．



(a) シミュレーション結果のアニメーション表示



(b) 実験結果

図 6.4 ：手先位置制御（$x^{\mathrm{ref}} = 0.15$ [m], $y^{\mathrm{ref}} = -0.05$ [m]）

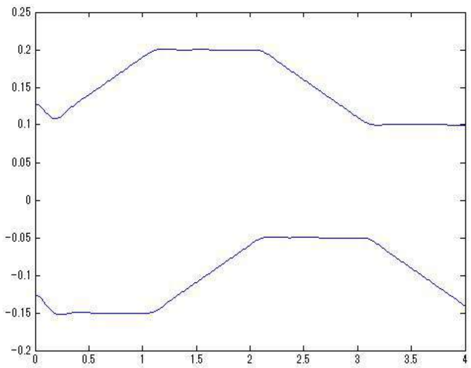
6.3　シミュレーションと実験 (目標値をデータで与える場合)

6.3.1　目標値の与え方

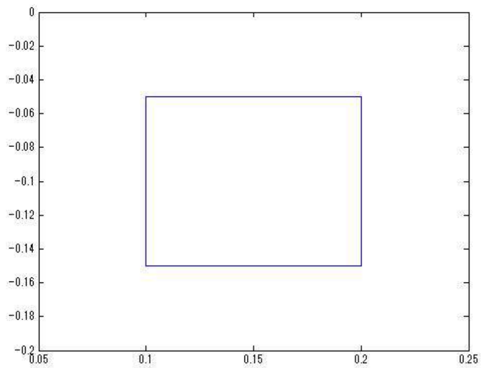
ここでは，目標値を以下のようにデータ列で与え，手先位置制御を行うことを考える．

```
t_data = [0; 1; 2; 3; 4];
x_ref_data = [0.1; 0.2; 0.2; 0.1; 0.1];
y_ref_data = [-0.15; -0.15; -0.05; -0.05; -0.15];
```

なお，このようにデータ列で目標値を与えたとき，図 6.5 のようになる．



(a) $x^{\mathrm{ref}}(t)$ と $y^{\mathrm{ref}}(t)$



(b) y^{ref} vs x^{ref}

図 6.5 ：データ列で与えられた手先の目標位置

6.3.2　デモンストレーション

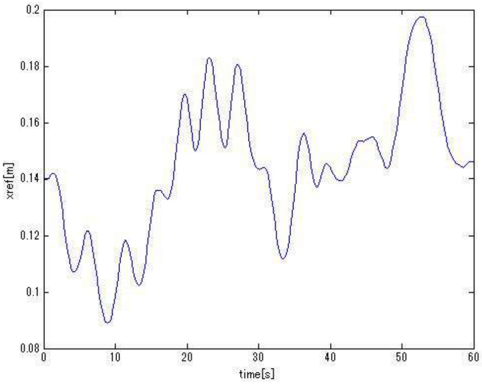
ここでは，花の絵を描くような目標位置データを用いた手先位置制御のデモンストレーションを行う．このデータは M ファイル “flower_data.m” に含まれており，$(x^{\rm ref}(t), y^{\rm ref}(t))$ の時系列データとして定義されている．この目標位置に追従する様子をアニメーション表示するために，MATLAB Command Window で以下のように入力する．

```
flower_data % flower_data.m の実行（手先位置の目標値設定）
figure(1); plot(t_data,x_ref_data);
xlabel('time [s]'); ylabel('x^{ref} [m]'); % (t, x_ref(t)) のプロット
figure(2); plot(t_data,y_ref_data);
xlabel('time [s]'); ylabel('y^{ref} [m]'); % (t, y_ref(t)) のプロット
figure(3); plot(x_ref_data,y_ref_data); % (x_ref(t), y_ref(t)) のプロット
xlabel('x^{ref} [m]'); ylabel('y^{ref} [m]');
t = t_data; x = x_ref_data; y = y_ref_data;
figure(4); sim_anime % flower_anime.m の実行（アニメーション）
```

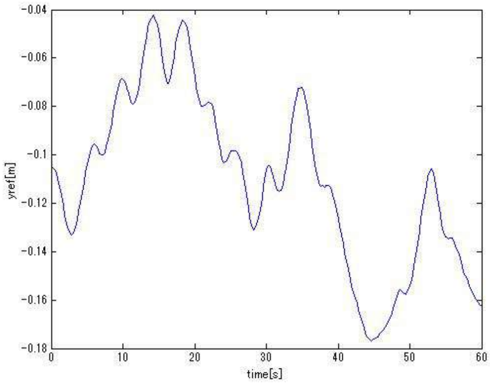
ここでは，M ファイル “flower_data.m” に含まれている花の絵のような目標位置データを用いて，手先位置制御の動作をデモンストレーションする．このファイルには，時系列で与えられた手先の目標位置 $(x^{\rm ref}(t), y^{\rm ref}(t))$ が含まれており，それを基にアニメーション表示およびグラフ出力を行う．MATLAB Command Window で以下のように入力すればよい．


```
flower_data % flower_data.m の実行（手先位置の目標値設定）
figure(1); plot(t_data,x_ref_data);
xlabel('time [s]'); ylabel('x^{ref} [m]'); % (t, x_ref(t)) のプロット
figure(2); plot(t_data,y_ref_data);
xlabel('time [s]'); ylabel('y^{ref} [m]'); % (t, y_ref(t)) のプロット
figure(3); plot(x_ref_data,y_ref_data); % (x_ref(t), y_ref(t)) のプロット
xlabel('x^{ref} [m]'); ylabel('y^{ref} [m]');
t = t_data; x = x_ref_data; y = y_ref_data;
figure(4); sim_anime % flower_anime.m の実行（アニメーション）
```

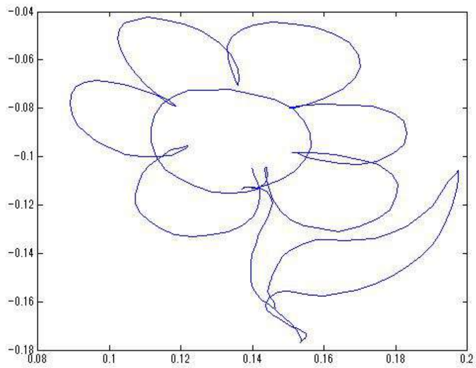

その結果，図 6.6 に示すような時系列グラフおよびアニメーション表示が得られる．



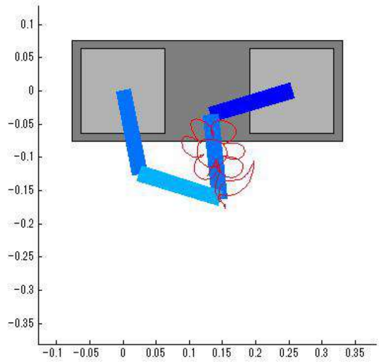
(a) $x^{\mathrm{ref}}(t)$



(b) $y^{\mathrm{ref}}(t)$



(c) 目標軌道



(d) アニメーション表示

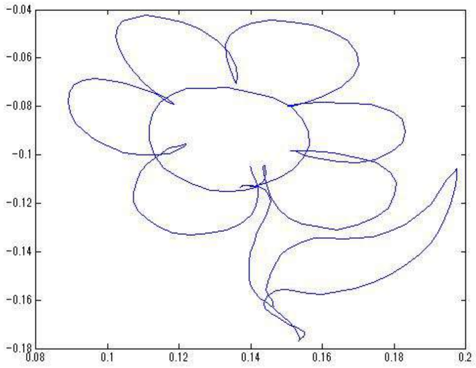
図 6.6 ：`flower_data.m` により設定された手先の目標位置

この花の絵の目標値データを利用して実験を行うために，MATLAB Command Window で以下のように入力し，パラメータ設定を行う．

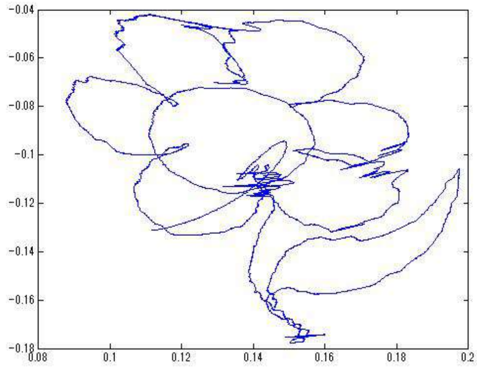
```
armIPD各パラメータを設定して下さい

omegaMx = 20
omegaMy = 20
............................................................ 《略》
alphaM1x = 2
alphaM2x = 2
alphaM1y = 2
alphaM2y = 2
............................................................ 《略》
flower_data % flower_data.m の実行（手先位置の目標値設定）
```

つぎに，“sim_robot_xy2.slx” によりシミュレーションを実行する．M ファイル `sim_anime` を実行してアニメーションで各リンクに無理な動きがないことを確認した後，“ex_robot_xy2.slx” をコンパイルしてから実機実験を行うと，図 6.7 の実験結果が得られる．



(a) 目標軌道



(b) 実験結果

図 6.7 : 手先位置制御

第 7 章　ダイレクトティーチング

7.1　実験内容

7.1.1　手順

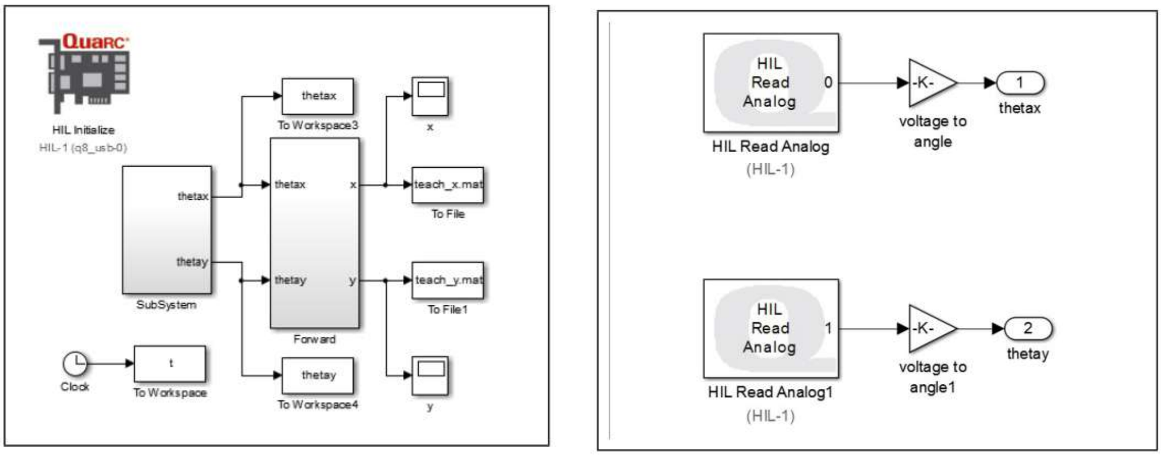
ダイレクトティーチングとは，手動で手先を動かし，2 軸ロボットにその動きを再現させることである．ダイレクトティーチングを行うためには，以下のステップを実行すればよい．

<u>ステップ 1</u>　手先位置 P_3 が所望の軌道に沿うように手先を手で動かし，関節角 $\theta_x(t)$，$\theta_y(t)$（あるいは手先位置 $x(t)$，$y(t)$）をデータとして取得する．

<u>ステップ 2</u>　ステップ 1 で取得した教師データを目標値 $\theta_x^{\mathrm{ref}}(t)$，$\theta_y^{\mathrm{ref}}(t)$（あるいは $x^{\mathrm{ref}}(t)$，$y^{\mathrm{ref}}(t)$）として手先位置（$x(t)$，$y(t)$）の動きを再現する．

7.1.2　教師データの取得

教師データを取得するため，図 7.1 に示す実機実験モデル “ex_robot_teach.slx” を作成し，¥ `teach` に保存する．M ファイル “armpara.m”，?? 節で示した M ファイル “armIPD.m”， “sim_anime.m” および “sim_robot_xy2.slx”，“ex_robot_xy2.slx” を ¥ `teach` に保存する．



(a) 実機実験モデル “ex_robot_teach.slx”　　(b) Subsystem “2D Robot” の内容

図 7.1 ： 教師データ取得のための実機実験モデル “ex_robot_teach.slx”

つぎに，“`ex_robot_teach.slx`” はデータ取得は 30 秒間であることに注意し，“Start” アイコンをクリックした後，所望の軌道に沿うように手先を動かす．実験終了後，プロットされたデータが “`teach_x.mat`”， “`teach_y.mat`” という名前で \\`teach` に保存されていることを確認する．

以上の作業が終了した後，**MATLAB Command Window** で

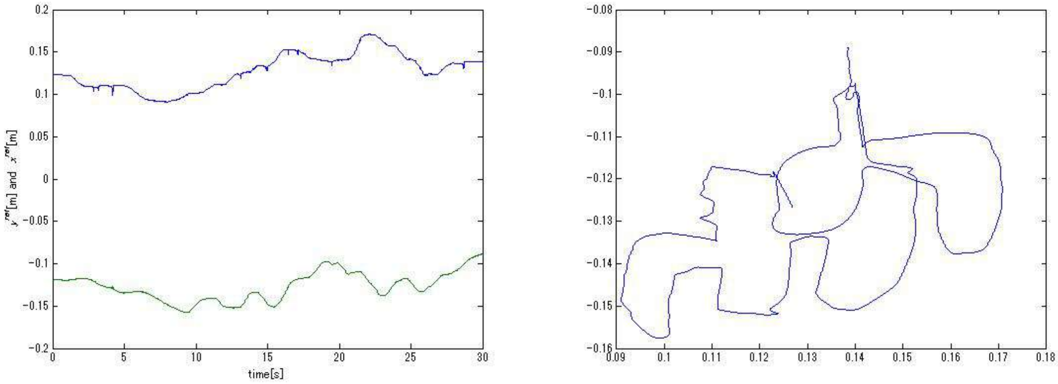
```
load teach_x;
load teach_y;
x_ref_data = teach_x(2,:); % x_ref の取得
y_ref_data = teach_y(2,:); % y_ref の取得
t_data = teach_x(1,:);

figure(1); plot(t_data,x_ref_data,t_data,y_ref_data);
xlabel('time [s]'); ylabel('¥ {it y}^{ref} [m] and ¥ {it x}^{ref} [m]');

```

```
figure(2); plot(x_ref_data,y_ref_data);
xlabel('¥ {it x}^{ref} [m]'); ylabel('¥ {it y}^{ref} [m]');
```

のように入力すると，取得された教師データが図 7.2 のように表示される．



(a) $x^{\mathrm{ref}}(t)$, $y^{\mathrm{ref}}(t)$　　(b) x–y 平面での表示

図 7.2 ：取得された教師データ

7.1.3　シミュレーションと実験

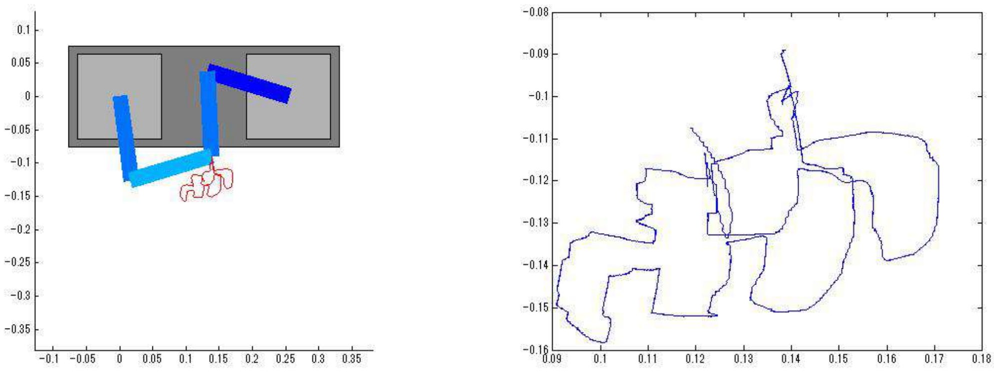
ステップ 1 で取得した教師データを目標値（$x^{\mathrm{ref}}, y^{\mathrm{ref}}$）として "sim_robot_xy2.slx" によりシミュレーションを行い，以下のように入力してシミュレーション結果をアニメーション表示する．

```
sim_anime;
```

アニメーションにより各リンクに無理な動きがないことを確認した後，"ex_robot_xy2.slx" をコンパイルして実機実験を行う．

7.2　実験結果

図 7.3a に示すように，シミュレーション結果のアニメーション表示では，手先が所望の軌道に沿って動いていることが確認できた．また，図 7.3b に示すように，実機実験でも同様の結果が得られた．



(a) シミュレーション結果のアニメーション表示　　(b) ダイレクトティーチングの実験結果

図 7.3 ：ダイレクトティーチングにおけるアニメーションと実験結果

7.3　考察

ダイレクトティーチングでは，手動で動かした手先の軌道を正確に再現できることが確認された．シミュレーションと実機実験の結果が一致しており，制御モデルが正確であることが示された．

ただし，実機実験ではセンサの観測データに若干のノイズが含まれていることが確認された．このノイズはモーターの不感帯や外部環境の影響によるものであると考えられる．