3. 基本的な制御系の解析および設計

1　目的

本実験の目的は基本的なシステムの過渡特性および制御システム設計の基礎を理解することである。ここで、2次システムの過渡特性や設計について例題を解きながら説明する。最後にコンピュータシミュレーションをしながら実験の課題を解くことになる。

2　2次サーボシステム

本実験において、2次システムを対象とする。以下は、具体的なサーボシステムを例として、そのモデル化、応答特性および設計を行う。

2.1　2次サーボシステムのモデル

サーボシステムの構成は図1に示されている。このシステムでは、可変抵抗からなるブリッジ回路は

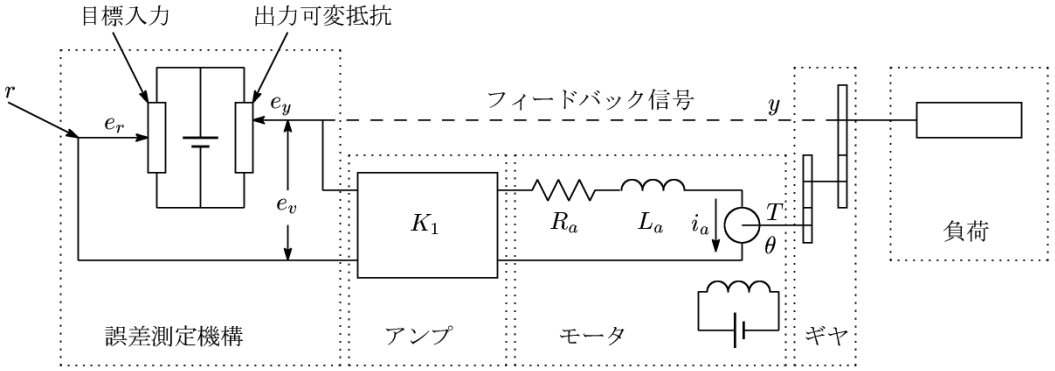


図 1: サーボシステムの構成図

は誤差測定回路の役割を果たしている。入力側可変抵抗のワイパーの角度 r は電圧 e_r と比例し、出力シャフトの角度 y は電圧 e_y と比例している。入力側と出力側の角度の差は

$$e = r - y \tag{1}$$

であり、ポテンシャルの差

$$e_r - e_y = e_v$$

は誤差電圧である。ここで、$e_r = K_0 r$ と $e_y = K_0 y$（K_0 は比例定数である）。e_v の誤差電圧が、ゲイン K_1 のアンプによって増幅される。誤差が生じているときに、モータが負荷を誤差が零になるまでに回転させる。一定の電磁界の場合、モータの出力トルク T は

$$T = K_2 i_a \tag{2}$$

である。ここで K_2 はモータのトルク係数で、i_a 回転子電流である。

$$e_b = K_3 \frac{d\theta}{dt} \tag{3}$$

ここで e_b 逆起電力で、K_3 はモータの逆起電力係数である。θ は回転子の回転角である。回転子の角速度は電圧 $e_a = K_1 e_v$ と直接比例している。回路の動作を表している微分方程式は

$$L_a \frac{di_a}{dt} + R_a i_a + e_b = e_a$$

、

$$L_a \frac{di_a}{dt} + R_a i_a + K_3 \frac{d\theta}{dt} = K_2 i_a \tag{4}$$

である。また、トルクの方程式は

$$J_0 \frac{d^2\theta}{dt^2} + b_0 \frac{d\theta}{dt} = T = K_2 i_a \tag{5}$$

である。ここで J_0 と b_0 は、それぞれモータ回転軸でのモータ・負荷・ギヤ - の慣性モーメントと粘性摩擦係数である。式 (4)、(5) から、モータの回転子の回転角度と誤差電圧関係を現す伝達関数は次の式である。

$$\frac{\Theta(s)}{E_v(s)} = \frac{K_1 K_2}{s(L_a s + R_a)(J_0 s + b_0) + K_2 K_3 s}. \tag{6}$$

ここで、$\Theta(s) = \mathcal{L}[\theta(t)]$、$E_v(s) = \mathcal{L}[e_v(t)]$ である。ギヤ比を n にすると出力のシャフトの角変異は

$$Y(s) = n\Theta(s) \tag{7}$$

となる。$E_v(s)$、$R(s)$ と $Y(s)$ の関係は次のようなる。

$$E_v(s) = K_0[R(s) - Y(s)] = K_0 E(s) \tag{8}$$

式 (6)、(7)、(8) によってシステムブロック線図は図 2 に示されている。開ループの伝達関数は次の通り

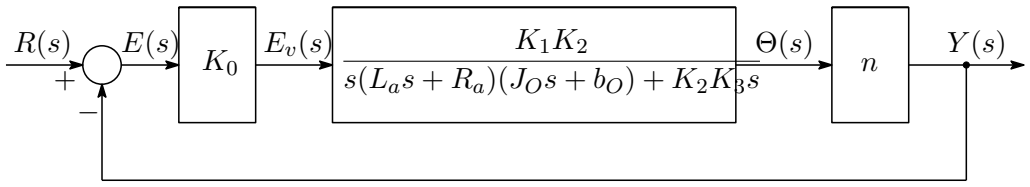


図 2: システムのブロック線図

である。

$$G(s) = \frac{Y(s)}{\Theta(s)} \frac{\Theta(s)}{E_v(s)} \frac{E_v(s)}{E(s)} = \frac{K_0 K_1 K_2 n}{s[(L_a s + R_a)(J_0 s + b_0) + K_2 K_3]} \tag{9}$$

多くのモータでは L_a は小さいので、無視することができる。

$$G(s) = \frac{K_0 K_1 K_2 n}{s[R_a(J_0 s + b_0) + K_2 K_3]} \tag{10}$$

$$= \frac{K_0 K_1 K_2 n / R_a}{J_O s^2 + \left(b_0 + \frac{K_2 K_3}{R_a}\right) s} \tag{11}$$

次の新たな係数を定義する。

$$J = J_O/n^2 = \text{出力シャフトでの慣性モーメント} \tag{12}$$

$$B = [b_0 + (K_2 K_3 / R_a)]s = \text{出力シャフトでの粘性摩擦定数} \tag{13}$$

$$K = K_0 K_1 K_2 / n R_a \tag{14}$$

このことにより、式 (10) での伝達関数は

$$G(s) = \frac{K}{Js^2 + Bs} \tag{15}$$

となる。ここでさらに、

$$K_m = \frac{K}{B}, \quad T_m = \frac{J}{B} = \frac{R_a J_O}{R_a b_O + K_2 K_3}$$

とおくと

$$G(s) = \frac{K_m}{s(T_m s + 1)} \tag{16}$$

となり、ブロック線図は図（3）に示されているように簡単になる。

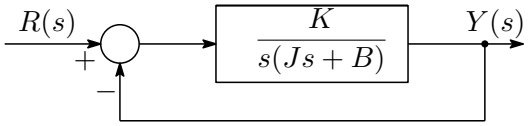


図 3: 閉ループのブロック線図

2.2　2 次元システムのステップ応答

図 3 での閉ループサーボシステムの伝達関数は

$$\frac{Y(s)}{R(s)} = \frac{K}{Js^2 + Bs + K} \tag{17}$$

である。この伝達関数を次のように書くことができる。

$$\frac{Y(s)}{R(s)} = \frac{\frac{K}{J}}{\left[s + \frac{B}{2J} + \sqrt{\left(\frac{B}{2J}\right)^2 - \frac{K}{J}}\right]\left[s + \frac{B}{2J} - \sqrt{\left(\frac{B}{2J}\right)^2 - \frac{K}{J}}\right]} \tag{18}$$

$B^2 - 4JK < 0$ のときに閉ループの極は複素極で、$B^2 - 4JK \geq 0$ の場合、実数である。

過渡特性解析を行うときに次の記述が便利である。

$$\frac{K}{J} = \omega_n^2, \qquad \frac{B}{J} = 2\zeta\omega_n = 2\sigma$$

ここで σ は減衰係数といい、ω_n 固有周波数という。σ は減衰率という。減衰率 ζ は、実際の減衰 B と臨界減衰 $B_c = 2\sqrt{JK}$ との比を表すので減衰比とも呼ばれる。

$$\zeta = \frac{B}{2\sqrt{JK}}$$

上記の記号によって図 3 のシステムを図 4 のように現され 閉ループの伝達関数（式 (18)）は

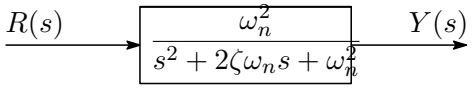


図 4: 2 次システム

$$\frac{Y(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{19}$$

となる。このことにより、2 次システムの動的特性が ζ と ω_n によって現すことができる。もし $0 < \zeta < 1$ であれば、閉ループの極が共役で、s 左半面に配置されている。システムの過渡特性は振動的である。$\zeta = 1$ のときシステムは臨界減衰システムという。また $\zeta > 1$ のときの状態をオバーダンピングといい、応答波形は非振動的である。$\zeta = 0$ のときの応答は振動的で、減衰されず永遠に発振する。

2.2.1 $0 < \zeta < 1$**:**

このとき伝達関数は次のように書くことができる。

$$\frac{Y(s)}{R(s)} = \frac{\omega_n^2}{(s + \zeta\omega_n + j\omega_d)(s + \zeta\omega_n - j\omega_d)} \tag{20}$$

ここで

$$\omega_d = \omega_n\sqrt{1 - \zeta^2} \tag{21}$$

である。ω_d は減衰固有周波数という。

ステップ入力が与えられたときに、出力は

$$Y(s) = \frac{\omega_n^2}{(s^2 + 2\zeta\omega_n s + \omega_n^2)s} \tag{22}$$

である。式 (22) の逆ラプラス変換を求めるために部分分数展開を使う。

$$\begin{aligned} Y(s) &= \frac{1}{s} - \frac{s + 2\zeta\omega_n}{s^2 + 2\zeta\omega_n s + \omega_n^2} \\ &= \frac{1}{s} - \frac{s + \zeta\omega_n}{(s + \zeta\omega_n)^2 + \omega_d^2} - \frac{\zeta\omega_n}{(s + \zeta\omega_n)^2 + \omega_d^2} \end{aligned} \tag{23}$$

逆ラプラス変換票によって

$$\begin{aligned} \mathcal{L}^{-1}\left[\frac{s + \zeta\omega_n}{(s + \zeta\omega_n)^2 + \omega_d^2}\right] &= e^{-\zeta\omega_n t}\cos\omega_d t \\ \mathcal{L}^{-1}\left[\frac{\omega_n}{(s + \zeta\omega_n)^2 + \omega_d^2}\right] &= e^{-\zeta\omega_n t}\sin\omega_d t \end{aligned}$$

であるので、式（22）の逆ラプラス変換は

$$\begin{aligned} \mathcal{L}^{-1}[Y(s)] &= y(t) \\ &= 1 - e^{-\zeta\omega_n t}\left(\cos\omega_d t + \frac{\zeta}{\sqrt{1 - \zeta^2}}\sin\omega_d t\right) \\ &= 1 - \frac{e^{-\zeta\omega_n t}}{\sqrt{1 - \zeta^2}}\sin\left(\omega_d t + \tan^{-1}\frac{\sqrt{1 - \zeta^2}}{\zeta}\right), \quad \text{for } t \geq 0 \end{aligned} \tag{24}$$

入力と出力の差（誤差）は

$$\begin{aligned} e(t) &= r(t) - y(t) \\ &= e^{-\zeta\omega_n t}\left(\cos\omega_d t + \frac{\zeta}{\sqrt{1 - \zeta^2}}\sin\omega_d t\right), \quad \text{for } t \geq 0 \end{aligned} \tag{25}$$

この誤差は減衰正弦波形であるが、定常状態で（$t = \infty$）$e(t) = 0$ である。$\zeta = 0$ であれば、非減衰振動的な応答になる。$\zeta = 0$ を式（24）に代入すれば

$$y(t) = 1 - \cos\omega_n t, \quad \text{for } t \geq 0 \tag{26}$$

となる。式（26）から ω_n は固有周波数であることが分かる。すなわち、減衰が零にしたときに、システムの出力が永久に振動することになる。もし、線形系が少しでも減衰があれば、固有周波数を見ることはできなくなる。このときに見られる周波数は減衰固有周波数（$\omega_d = \omega_n\sqrt{1 - \zeta^2}$）である。$\omega_d$ は ω_n より小さい。

2.2.2 $\zeta = 1$:

$Y(s)/R(s)$ の極がほぼ等しければ、システムは臨界減衰システムに近似する。ステップ入力（$R(s) = 1/s$）に対する応答 $Y(s)$ は

$$Y(s) = \frac{\omega_n}{(s+\omega_n)^2 s} \tag{27}$$

である。式（27）の逆ラプラス変換は

$$y(t) = 1 - e^{-\omega_n t}(1+\omega_n t), \quad \text{for } t \geq 0 \tag{28}$$

となる。この結果は、次のリミットにより得ることができる（式（24）参照）。

$$\lim_{\zeta \to 1} \frac{\sin \omega_d t}{\sqrt{1-\zeta^2}} = \lim_{\zeta \to 1} \frac{\sin \omega_n \sqrt{1-\zeta^2} t}{\sqrt{1-\zeta^2}} = \omega_n t$$

2.2.3 $\zeta > 1$:

このときシステム極は不の実数である。ステップ入力（$R(s) = 1/s$）に対する応答 $Y(s)$ は

$$Y(s) = \frac{\omega_n^2}{(s+\zeta\omega_n+\omega_n\sqrt{\zeta^2-1})(s+\zeta\omega_n-\omega_n\sqrt{\zeta^2-1})s} \tag{29}$$

これの逆ラプラス変換は

$$\begin{aligned} y(t) &= 1 + \frac{1}{2\sqrt{\zeta^2-1}(\zeta+\sqrt{\zeta^2-1})} e^{-(\zeta+\sqrt{\zeta^2-1})\omega_n t} \\ &- \frac{1}{2\sqrt{\zeta^2-1}(\zeta-\sqrt{\zeta^2-1})} e^{-(\zeta-\sqrt{\zeta^2-1})\omega_n t} \\ &= 1 + \frac{\omega_n}{2\sqrt{\zeta^2-1}} \left(\frac{e^{-s_1 t}}{s_1} - \frac{e^{-s_2 t}}{s_2} \right), \quad \text{for } t \geq 0 \end{aligned} \tag{30}$$

ここで $s_1 = (\zeta + \sqrt{\zeta^2-1})\omega_n$、$s_2 = (\zeta - \sqrt{\zeta^2-1})\omega_n$ である。

$|s_2| \ll |s_1|$ のとき、s_1 を無視することができ、システム応答は一次システムの応答と同じものになる。

$$\frac{Y(s)}{R(s)} = \frac{\zeta\omega_n - \omega_n\sqrt{\zeta^2-1}}{s+\zeta\omega_n-\omega_n\sqrt{\zeta^2-1}} = \frac{s_2}{s+s_2}$$

この近似関数でのステップ応答は

$$Y(s) = \frac{\zeta\omega_n - \omega_n\sqrt{\zeta^2-1}}{(s+\zeta\omega_n-\omega_n\sqrt{\zeta^2-1})s}$$

であり、時間領域では

$$y(t) = 1 - e^{-(\zeta-\sqrt{\zeta^2-1})\omega_n t}, \quad \text{for } t \geq 0$$

である。

式（24）（28）と（30）を用いてシステムの応答を図 5 に示す。

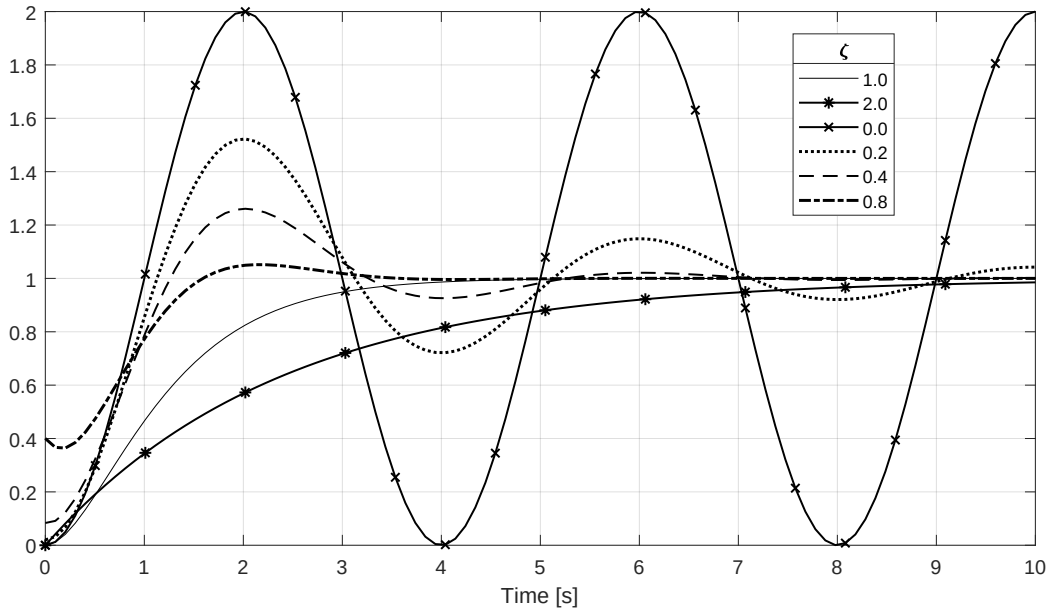


図 5: ２次システムのステップ応答

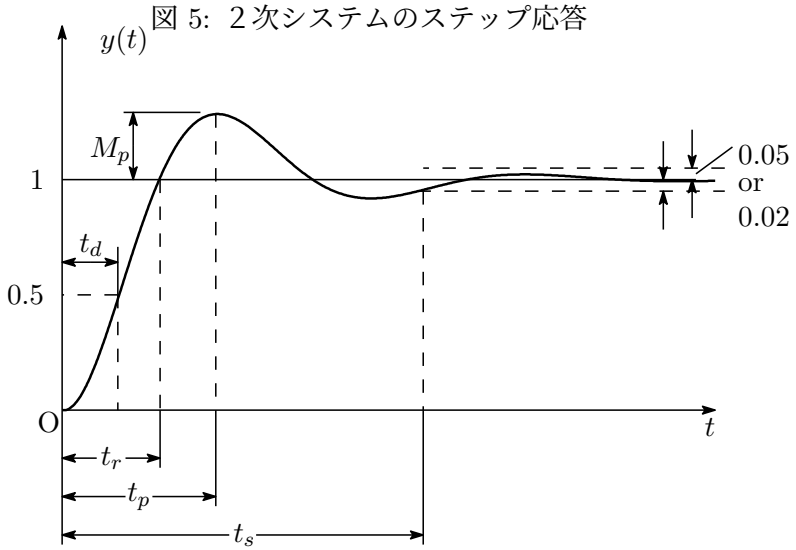


図 6: ステップ応答

2.3　過渡特性に関する基本定義

ステップ入力に対するシステムの応答は変数の初期値に影響される。様々なシステムの過渡特性が比較されるためにシステムの変数の初期値を零にすることが多い。システムの応答特性に対して次の量が定義され、図 6 に示されている。

1. 遅れ時間（t_d）。

   遅れ時間は、応答が定常状態の 50%の所に達するまでの時間である。

2. 立ち上がり時間（t_r）

   立ち上がり時間は、応答が最終値の 10%から 90%までに達する時間である。減衰振動的な応答のシステムでは、普通、0%〜100%が使われる。5%〜95% も使われる。

3. ピーク時間（t_p）

   行き過ぎの第 1 ピークまでの時間である。

4. 最大行き過ぎ量（M_p）

   応答の定常状態に対する最大ピープの値である。パーセント（%）で現されることもある。

$$\text{最大行き過ぎ量} = \frac{y(c_p) - c(\infty)}{\infty} \times 100\%$$

   最大行き過ぎ量はシステムの相対安定性と直接比例していてる。

5. 設定時間（t_s）

   応答が定常状態の 2%または 5%の所に達するまでの時間である。

ここで式 (19) で現されているシステムの立ち上がり時間、ピーク時間、最大行き過ぎ量および設定時間を ζ と ω_n により現す。

立上がり時間 t_r

式（24）で $y(t_r) = 1$ にすると

$$y(t_r) = 1 = 1 - e^{-\zeta\omega_n t_r}\left(\cos\omega_d t_r + \frac{\zeta}{\sqrt{1-\zeta^2}}\sin\omega_d t_r\right) \tag{31}$$

上記の式で $e^{-\zeta\omega_n t_r} \neq 0$ であるので

$$\cos\omega_d t_r + \frac{\zeta}{\sqrt{1-\zeta^2}}\sin\omega_d t_r = 0$$

である。または

$$\tan\omega_d t_r = -\frac{\sqrt{1-\zeta^2}}{\zeta} = -\frac{\omega_d}{\sigma}$$

立上がり時間は

$$t_r = \frac{1}{\omega_d}\tan^{-1}\left(\frac{\omega_d}{-\sigma}\right) = \frac{\pi - \beta}{\omega_d} \tag{32}$$

ここでの角 β は図 7 に定義されている。上記のことから、小さい t_r のために大きい ω_d が必要であることがわかる。

ピーク時間 t_p

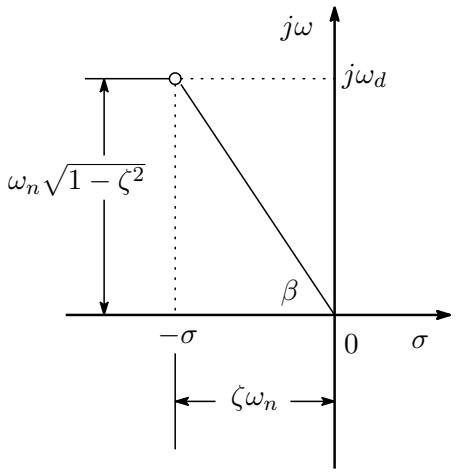


図 7: 角 β の定義

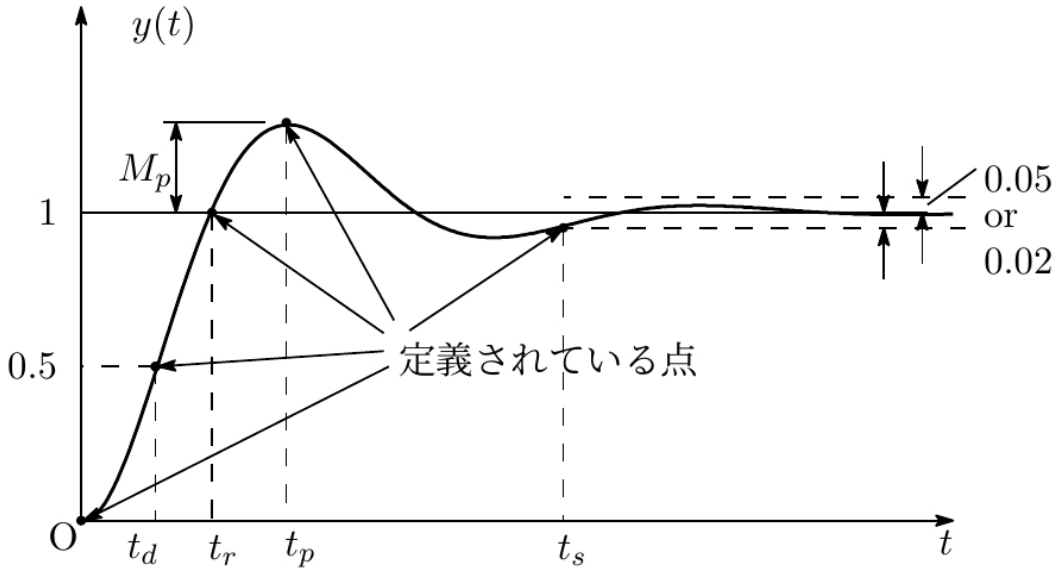


図 8: 過渡特性に関する定義

式（24）を時間に対して微分し、t_p を得る。

$$\begin{aligned}\frac{dy}{dt} &= \zeta\omega_n e^{-\zeta\omega_n t}\left(\cos\omega_d t + \frac{\zeta}{\sqrt{1-\zeta^2}}\sin\omega_d t\right) \\ &\quad + e^{-\zeta\omega_n t}\left(\omega_d \sin\omega_d t - \frac{\zeta\omega_d}{\sqrt{1-\zeta^2}}\cos\omega_d t\right) \end{aligned} \tag{33}$$

ここから

$$\left.\frac{dy}{dt}\right|_{t=t_p} = \sin\omega_d t_p \frac{\omega_n}{\sqrt{1-\omega^2}} e^{-\zeta\omega_n t_p} = 0$$

となり、次の式が得られる。

$$\sin\omega_d t_p = 0$$

そして、

$$\omega_d t_p = 0,\ \pi,\ 2\pi,\ 3\pi,\ \ldots。$$

t_p は第 1 ピークの時刻（$\omega_d t_p = 0$）であるので、

$$t_p = \frac{\pi}{\omega_d} \tag{34}$$

最大行きすぎ量 M_p

最大行きすぎ量は $t = t_p = \pi/\omega_d$ のときの行き過ぎ量であるので、式（24）から

$$\begin{aligned} M_p &= y(t) - 1 \\ &= -e^{-\zeta\omega_n(\pi/\omega_d)}\left(\cos\pi + \frac{\zeta}{\sqrt{1-\zeta^2}}\sin\pi\right) \\ &= e^{-(\sigma/\omega_d)\pi} = e^{-(\zeta/\sqrt{1-\zeta^2})\pi} \end{aligned} \tag{35}$$

最大行き過ぎ量は $e^{-(\zeta/\sqrt{1-\zeta^2})\pi} \times 100\%$ である。

設定時間 t_s

式（24）によりシステムの応答は

$$y(t) = 1 - \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}}\sin\left(\omega_d t + \tan^{-1}\frac{\sqrt{1-\zeta^2}}{\zeta}\right),\quad \text{for } t \geq 0$$

$e^{-\zeta\omega_n t}/\sqrt{1-\zeta^2}$ は過渡特性を囲むグラフ線である。このグラフ線の時間定数は $1/\zeta\omega_n$ である。

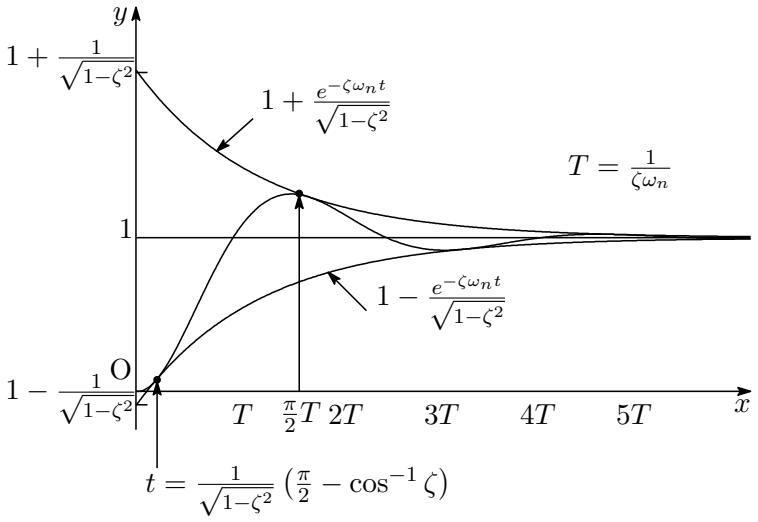
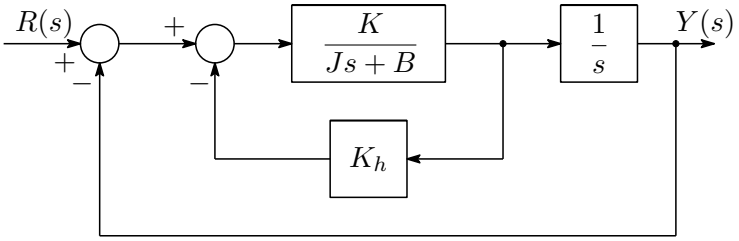
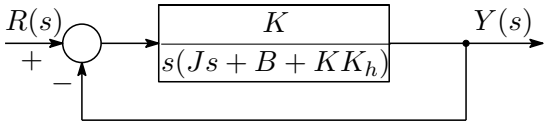


図 9: 過渡特性



(a) サーボシステムのブロック線図



(b) 簡単化されたブロック線図

図 10: 課題のブロック線図

2.4 速度フィードバックを持つシステムの設計

システムの過渡特性を改善するために出力の微分値のフィードバックが使われることがある。出力の微分を得るのに微分回路を使用するよりタコメータを使うことが望ましい。図 10(a) のシステムでは、出力の位置と速度のフィードバックがかかっている。このブロック線図を図 10(b) のように簡単化できる。図 10(b) により伝達関数は次のようになる。

$$\frac{Y(s)}{R(s)} = \frac{K}{Js^2 + (B + KK_h)s + K} \tag{36}$$

式（36）と式（17）を比較すれば、速度フィードバックは減衰率を増加するような役割していることがわかる。減衰比と固有周波数は

$$\begin{aligned} \zeta &= \frac{B + KK_h}{2\sqrt{KJ}} \\ \omega_n &= \sqrt{K/J} \end{aligned} \tag{37}$$

となる。ただし、フィードバックは ω_n に対して影響をしない。最大行き過ぎ量は ζ に依存しているので、K_h を調節することによって M_p を減少することができる。

3 課題

1. 図 1 のシステムでは、$\zeta = 0.6$、$\omega_n = 5$rad/sec とする。ステップ入力の場合、立ち上がり時間 t_r、ピーク時刻 t_p、行き過ぎ量 M_p、設定時間 t_s を求めよ。

   (a) ζ と ω_n から ω_d および σ を得る。

   (b) 立上がり時間 t_r を求める前に β を計算する。

   (c) t_r を求める。

   (d) ピーク時刻 t_p を求める。

   (e) 最大行き過ぎ量 $M_p[\%\ ]$ を計算する。

   (f) 2%領域と 5%領域での設定時間 t_s を求める。

   (g) シミュレーションによって、システムのステップ応答を得、上記の計算結果と比較せよ。

2. 図 10(a) のシステムに対して、ステップ入力が与えられたときに最大行き過ぎ量が 0.2、ピーク時間が 1 sec になるように、ゲイン K およびフィードバック係数 K_h を求めよ。なお、$J = 1$kg-m^2、$B = 1$N-m/rad/sec とする。

   (a) 最大行き過ぎ量の式 (式 (35)) より ζ を求めよ。

   (b) 式（34）より ω_d を求めよ。

   (c) ω_n および K を求めよ。

   (d) 式（37）より K_h を計算せよ。

   (e) 立上がり時間 t_r は式（32）より求まる。

   (f) 設定時間 t_s を 2%領域と 5%領域で計算せよ。

   (g) シミュレーションによって、システムのステップ応答を得、上記の計算結果と比較せよ。

参考文献

[1] Brogan, W. L., *Modern Control Theory.* Upper Saddle River, NJ, Prentice-Hall, Inc., 1985.

[2] Kailath, T., *Linear Systems*, Upper Saddle River, NJ, Prentice-Hall, Inc., 1980.

[3] 小郷　寛、美多　勉：システム制御理論入門、実教出版、１９９５。

[4] Ogata, K., *Modern Control Engineering*, 3-rd ed., Prentice-Hall International, Inc., 1997.

4．多関節ロボット

1　目　的

ロボット工学は、機械、電子、制御、情報、計算機、材料と幅広い分野に多岐にわたり関係している。本実験は、その中で電子工学のカリキュラムに沿った内容しており、産業用ロボットに関する基礎知識を習得することを目的している。

2　ロボットマニピュレータの構成

2.1　ロボットの定義

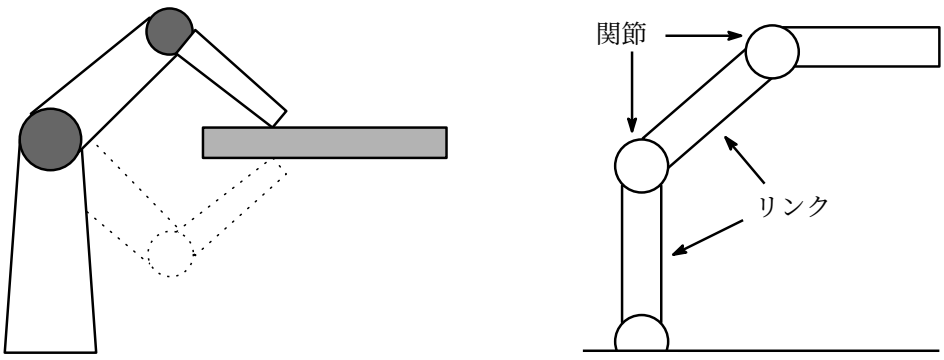
ISO（国際標準化機構）／TC184（産業オートメーションシステム）／SC2（工業用ロボット）では、マニピュレーティング・インダストリアル・ロボット（Manipulating Industrial Robot）とは『自動制御された再プログラム可能な、多用途で、いくつかの自由度を有するマニピュレーション機能を持つ機械』と定義している。産業用ロボットと他の専用自動機械とは、次の点において相違があるといえる。専用自動機械は大量生産に有効な手段として単一もしくは固定した作業をするものである。一方、産業用ロボットは中種中量生産や多品種少量生産に有効な手段として、対象物の種類の変化、設計変更、作業変更に対して再プログラムによって、または自律的に他用途の作業をするものである [1]。

2.2　構成

ここからは、本実験で使用するロボットの構成を中心にし述べる。

ロボットは多数自由度を持つ機械であり、主構造部（腕または arm、アームも呼ばれている）と手首（hand、ハンド）からなっている。通常は、マニピュレータは 3 次元空間で任意の位置と姿勢をとれるように 6 自由度を用意する [2]。

主構造部は、通常、図 1 に示すように関節で結合された一連のリンクによって構成されている。ロボットの腕は回転運動と直線運動の両方を行える。リンク間の動作を制御するのが関節の機能である。回転軸（関節）は DC サーボモータによって駆動され、全体の制御はコンピュータを使用して PID 制御則の下に運転する。ロボットは専用のプログラムによりさまざまな方向に動くことができる。ロボットアーム（ 図 2 、図 4）は 6 つの関節から構成



(a)　　(b)

図 1: (a)：方向が異なる場合の腕の配置　(b)：腕の概念図

される。WAIST（ねじり）、SHOULDER（上下方向の曲げ）、 ELBOW（左右方向の曲げ）の 3 自由度を持つ。

図 1 右のように梁（はり）の上面を溶接する場合と、同じ点を下面から溶接する場合とでは、ロボットの姿勢は変わる。基本的には、ロボットが空間の任意の位置に所定の方向で達するには、６つの可動軸 (６自由度) を必要とする。達する位置が同じでも方向が異なるとロボットの姿勢は完全に変わってしまう。作業ツールの細かい動作

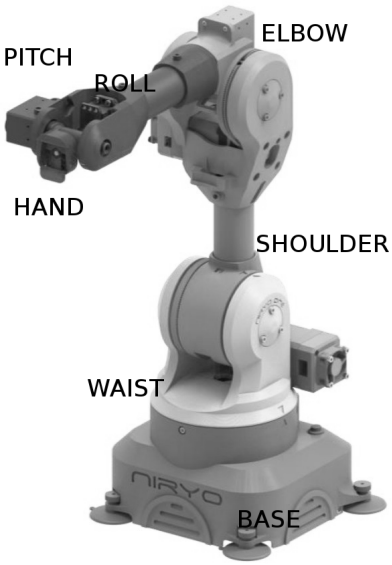


図 2: 主構造部

	Min	Max	位置データ
J1	-175°	175°	0〜9700
J2	-90°	36.7°	0〜13300
J3	-80°	90°	0〜12600
J4	-175°	175°	0〜4096
J5	-100°	110°	0〜4096
J6	-147.5°	147.5°	0〜1024

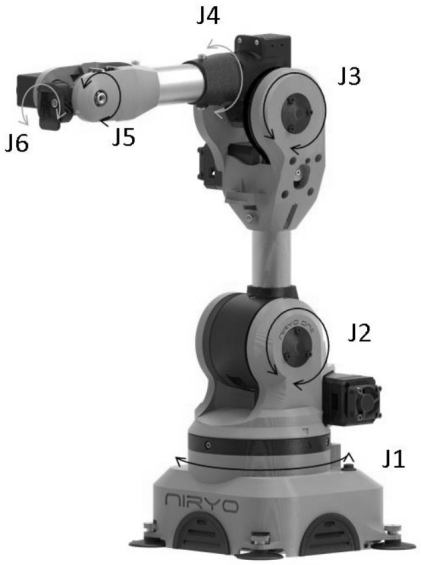


図 3: 各関節の回転角

に影響を与える最先端部を手首である。手首も 3 つの関節から構成され、図 2 に示す。ROLL（ねじり）、PITCH（上下方向の曲げ）、HAND（左右方向の曲げ）の 3 自由度を持つ。各関節の動作範囲は図 3 及び図 4 に示す。作業ツールはグリッパ、真空ポンプ、電磁石などロボットの使用目的に応じて変えられる（付録 B）。

本ロボットのインターフェイスについて付録 A に説明している。

エンド・エフェクタの動作は、ロボットの可動軸の位置と運動速度を制御することにより得られる。ロボットシステムの基本構成を図 5 に示す。次に図 5 の各要素について説明する。

2.3 制御用コンピュータ（コントローラ）について

制御用コンピュータはロボット本体に搭載されており、モータ制御やホストコンピュータ用インターフェイスである。ロボットのローカルコントロールにより動作テストあるいは 6 軸全てを動作範囲内に移動させるために使用する。コントローラは Raspberry Pi[4] で実現されており、ロボット制御に ROS（Robot Operating System）が使われている [3]。

Raspberry Pi とホストコンピュータは WiFi にて接続されている。

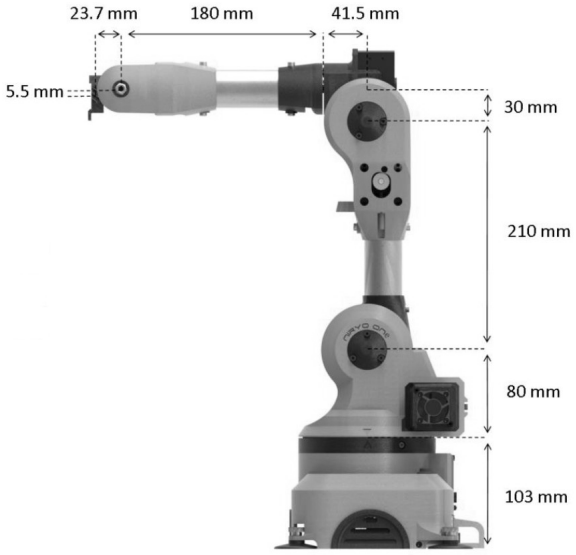


図 4: ロボットーの寸法

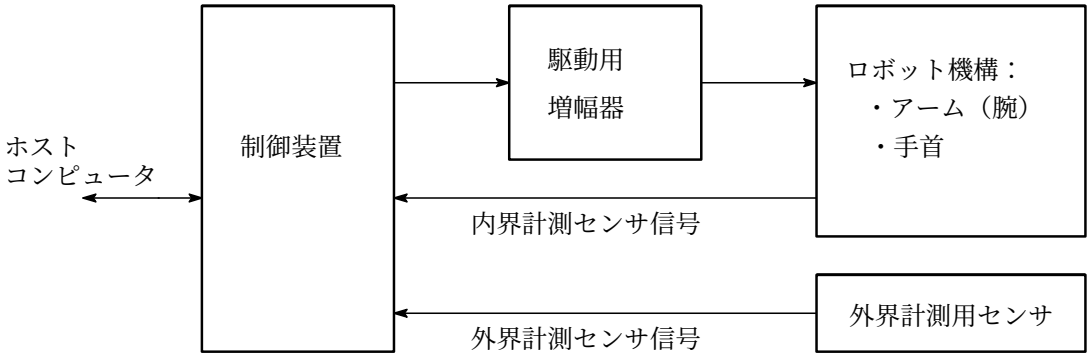


図 5: ロボットシステムの基本構成

3　実験を行うための準備

3.1　制御用ソフトウェアの実行

「NiryoOneStudio」を用いてロボットを操作･プログラミングすることができる。実行をするため「NiryoOneStudio」をダブルクする（図 6 参照)。

本ソフトウエアの左側にあるメニューとその説明文を図 7 に示す。

3.2　電源の操作

3.2.1　電源を入れる

電源を入れる前に次のことを確認する

- 電源スイッチが「OFF」である。
- 物や人との接触の危険性がない。
- ホストコンピュータが起動されている。

電源スイッチを「ON」にすると LED がまず赤色で点灯し、およそ 2 分後に緑色に代わる。**注意：**LED が点滅する場合、モータの接続問題を示し、ロボットを利用できない。

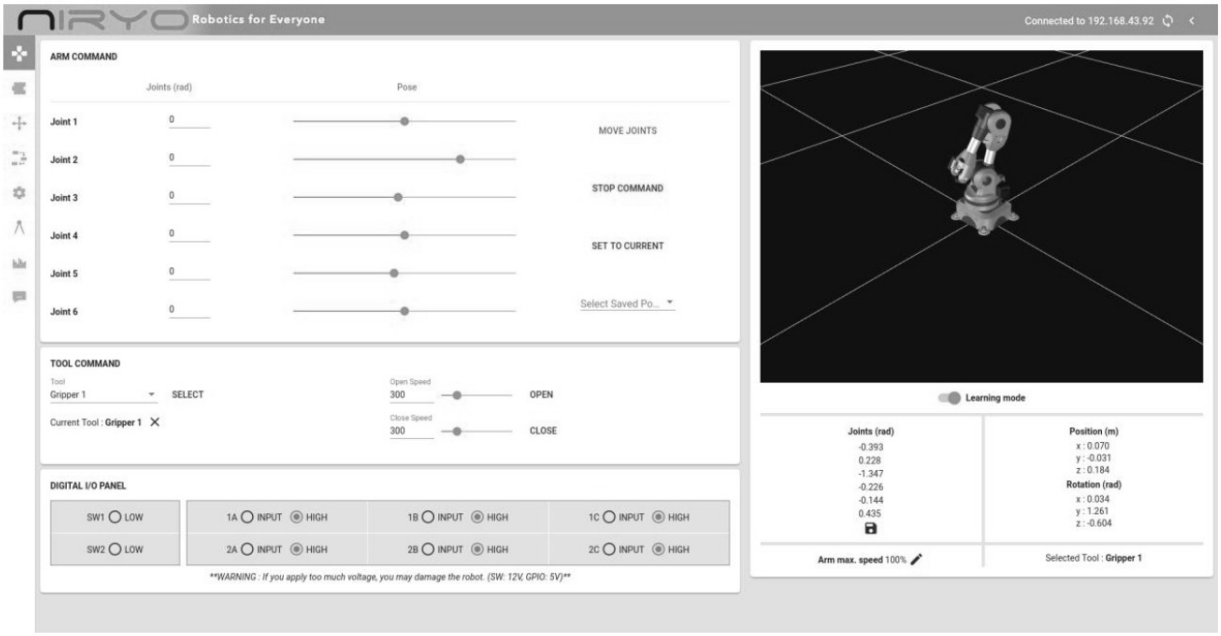


図 6: NiryoOneStudio のメイン画面

3.2.2 電源を切る

注意： Do not power off the robot directly with the power switch or by removing the power adapter. This is the same as if you directly unplug your own computer without shutting it down before.

電源を次のいずれの方法で切ることができる。

1. LED が紫色に切り替わるまでに（およそ 3 秒）トップボタンを押す。LED が赤色に変わったら、電源スイッチを切る。
2. Niryo One Studio の settings パネルで、Raspberry Pi settings において、「Shutdown Raspberry Pi」をクリックする。LED が紫色から赤色に変わったら、電源スイッチを切る。

3.3 トップボタン

トップボタンは押される時間により次の機能を持つ。

1. 1 回押し：保存されたプログラムシーケンスが実行される。もう一度ボタンを押すと動作が停止される。
2. 3 秒押し：LED が紫色に変わり、コンピュータがシャットダウンされる。

3.4 Wi-Fi 接続

ホストコンピュータとロボットとの接続は Wi-Fi ネットワークにりされる。NiryoOneStudio 実行後に右上に「Not connected」が表示される。ロボットが実行されたら（LED が緑色）右上のボックスの矢印をクリックし、ロボットと接続をする（図 8 参照）。

- ロボット操作
- Niryo ブロック
- 保存された姿勢
- 保存されたロボットプログラム呼び込み・実行
- 設定（WiFi、Raspberry Pi、ソフトウエア）
- キャリブレーション（較正）
- ハードウェア状態確認
- 動作ログ閲覧

図 7: NiryoOneStudio のメニューバー

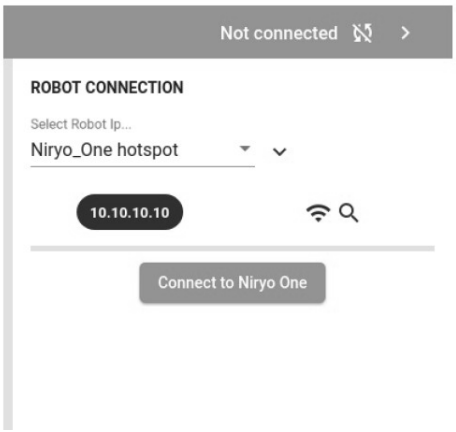


図 8: キャリブレーションメッセージ

3.5 ロボットのキャリブレーション

ボットとの接続ができたら「Niryo One Studio」上に図 9 のメッセージが表示される。「Calibration needed」をクリックして、ロボットの関節を図 10 のように合わせてから図 11 の「Manual calibration」を選ぶとキャリブ

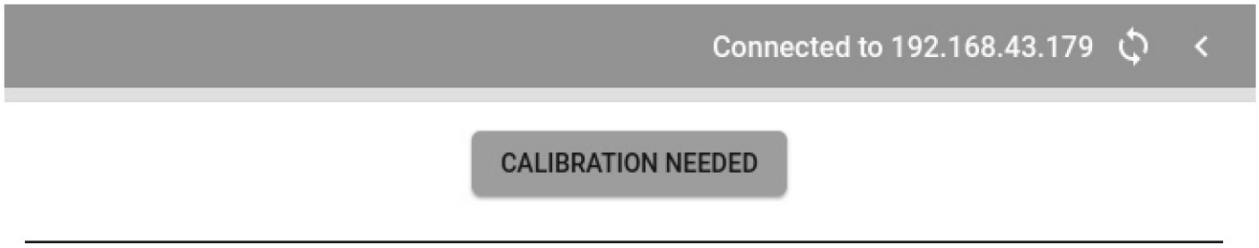


図 9: キャリブレーションメッセージ

レーションが完了する。

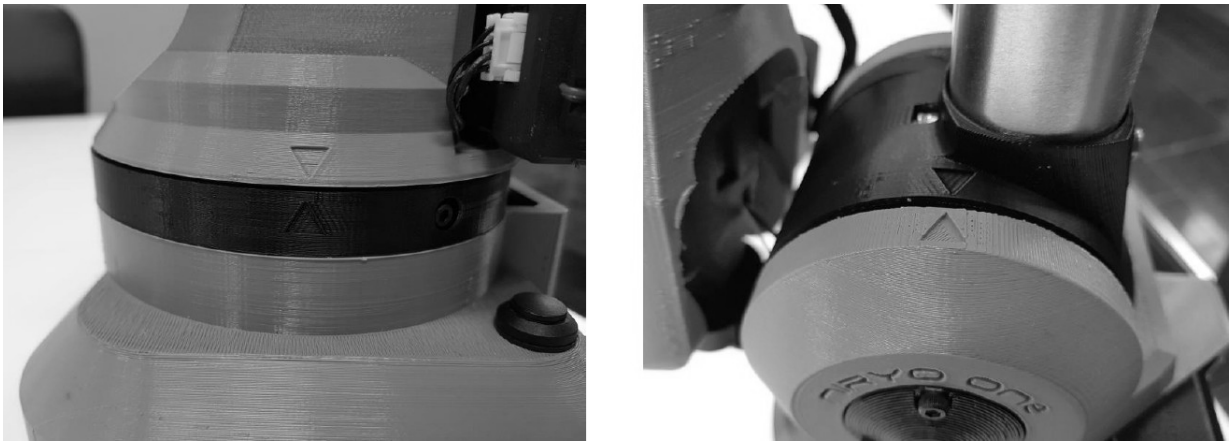


図 10: キャリブレーションメッセージのため調整

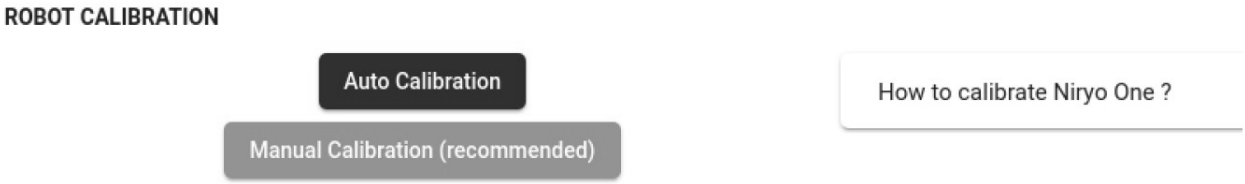


図 11: Manual Calibration をクリック

4　ロボットの操作

ロボットの操作ができる前に次の条件が必要である。

- ロボットが起動している。
- ホストコンピュータとの接続が行われている。
- キャリブレーションが行われている。

4.1　State section 及び Learninn mode

Niryo One Studio の右側パネルは常に表示されている（図 12）。このパネルにはロボットの状態表示（state section）と Learning Mode 機能が含まれている。図 12 中の各要素は次の通りである。

1. ロボットの姿勢
2. Learning mode スイッチが ON のとき、各モータのトルクが零となり、ロボット先端を目的位置へ移動させ、同スイッチが OFF 状態では、モータにトルクがかかって、姿勢が維持される。**注意：**Learning modeOFF のときに、力を加えてロボットアームを動かさないこと。なお、ロボットを操作しないときに、Learning modeON にするのが望ましい。
3. 各関節の現在位置（角度）
4. 作業ツールの現在座標と角度
5. “Save current position” ボタンで、名前をつけて現在の姿勢を保存できる。
6. アーム先端の現在速度（0〜100%）
7. 現在の作業ツール

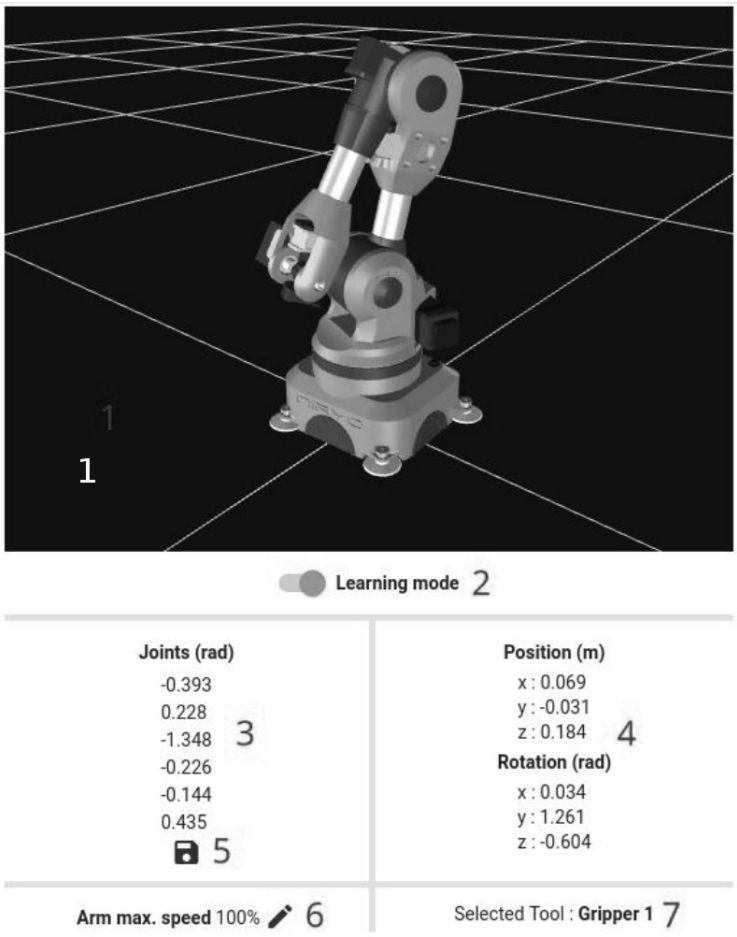


図 12: State section

4.2 ロボットを動かす

コマンドパネルにおいて、「Joints」タブ（図 13 参照）で各関節の角度を設定し、ロボットの姿勢を調整することができる。同パネルで、「Pose」タブ（図 14 参照）上では、ロボットの作業領域座標系上でロボットの先端の位置及び傾きの設定により、姿勢を決定できる。 設定した姿勢へ動かすため、次の順で操作する。

図 13: 「Joints」タブ

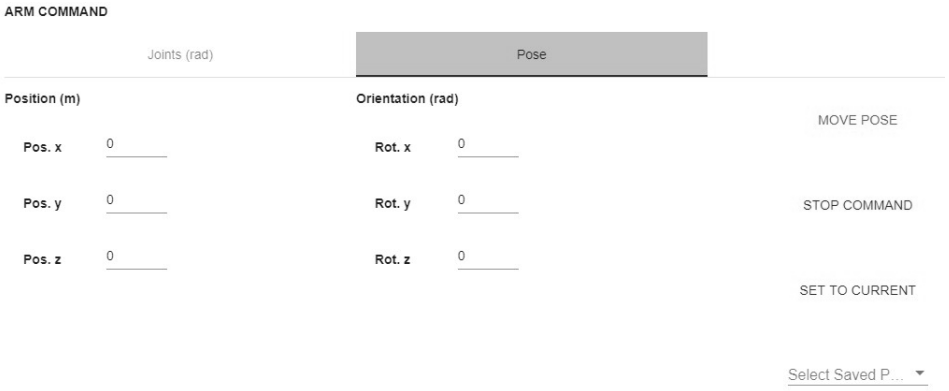


図 14: 「Pose」タブ

1. （オプション）「Set to current」をクリックし、各軸の角度、座標及び傾きを現在のロボット状態にする。これは、現在の状態をもとにして、（たとえば、3 番目の間接のみを調整したい、または、作業ツールを y 軸に沿ってスライドさせたい）姿勢の細かい調整を行う場合に便利である。
2. 各関節の角度または pose を調整する。
3. 「Move joints」（なたは「Move Pose」をクリックする。
4. 動作が完了したら、画面の下側に動作終了結果（図 15 が表示される。

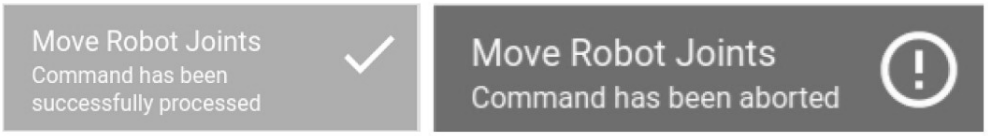


図 15: 動作完了結果（どちらかが表示される）

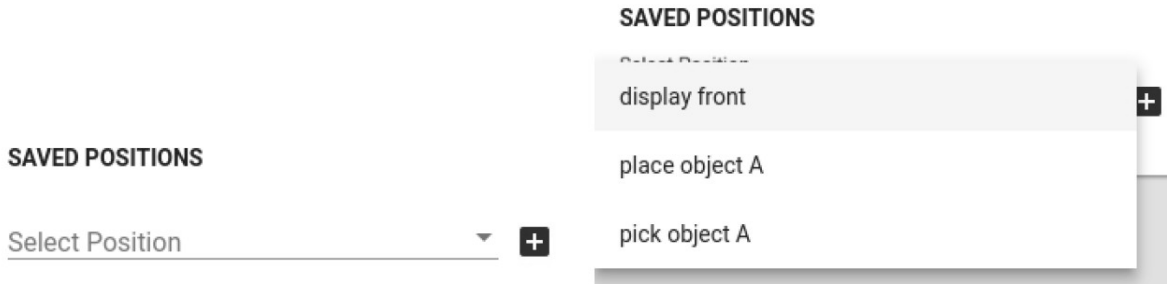
「Select saved position」より、以前に保存した姿勢を上述のステップ 1 とステップ 2 の代わりに実行できる。

「Tool command」により、ロボットに装着されているグリッパを設定できる。その開閉速度も個別に調整することが可能である。

4.3 保存された姿勢について

4.3.1 概要

「saved positions」パネル（図 16 参照）により保存された姿勢のリストを閲覧できる。



(a)「Saved positions」タブ　　(b) 保存された姿勢リスト

図 16: 保存されたロボットの姿勢

保存姿勢を選択するとその特性が図 17 のように表示される。そこで次のことができる。

1. 各関節の状態を調べる。
2. 内容を編集
3. 削除

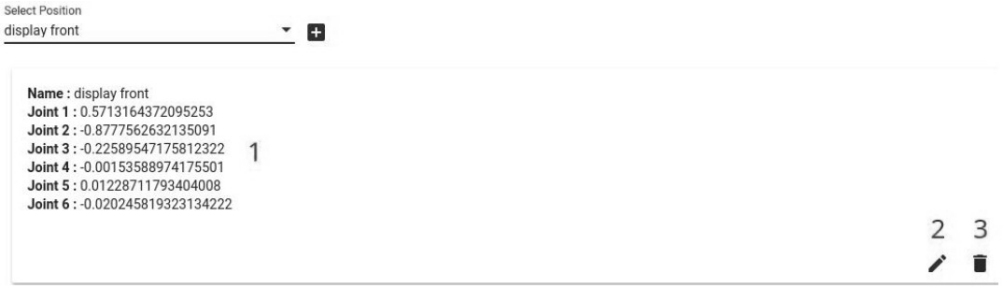


図 17: 以前に保存された姿勢の特性画面

4.3.2 ロボット姿勢の保存

現在のロボットの姿勢を保存するのに「State section」の「5」（図 12）のマークをクリックして、その姿勢の名前を入力し、保存する。保存した姿勢を「Arm command」の「Joints」または「Pose」タブで再利用される。複数の姿勢をつないで、ロボットの作業シーケンスを構成することもできる。

4.4 Niryo blocks ロボットプログラミングインターフェイス

本ロボットプログラミングインターフェイスは、MIT の Scratch プロジェクト [5] も利用している Google の Blockly[6] をもとにしたものである。インターフェイス環境画面を図 18 に示す。プログラミング環境の各要素は次の通りである。

1. プログラム用ワークスペース
2. 作成したプログラム内容はワークスペース内にセンタリングされる

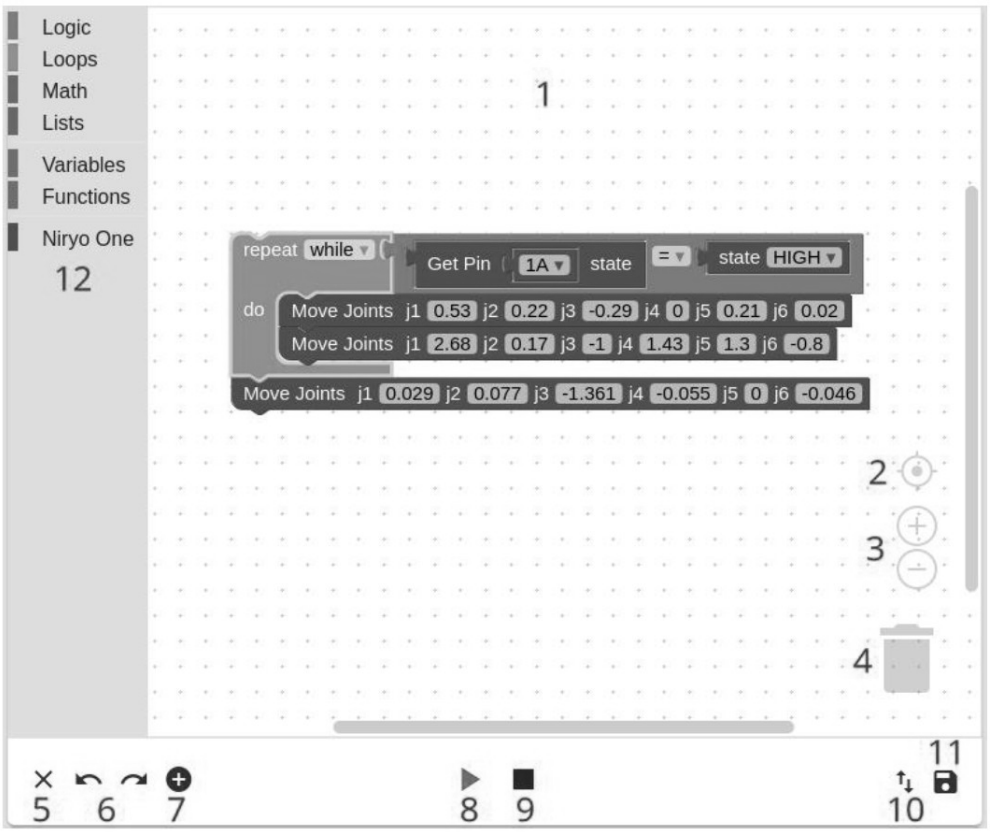


図 18: Niryo blocks 環境

3. ズーム
4. ブロックをドラッグして削除する。「Delete」ボタンも利用できる。
5. 内容全体削除
6. Undo (Ctrl+z)/Redo (Ctrl + Shift + z)
7. あらかじめ保存した姿勢を挿入するまたは現在のロボットの姿勢を採用する。
8. 作成したプログラムを実行する。
9. 実行中プログラムシーケンス停止
10. XML ファイルからブロックシーケンス挿入またはファイルへ保存する。Python プログラムとして保存も可能である。
11. プログラムをロボットコンピュータへ保存
12. 本ロボット特有の追加機能

5 実 験

5.1 実験 I：プログラミング基礎

ここで、ラーニングモードを用いて、簡単なプログラムを作成する。

1. 3 節の内容にしたがって次ぎの手順で実験開始の準備を行う。

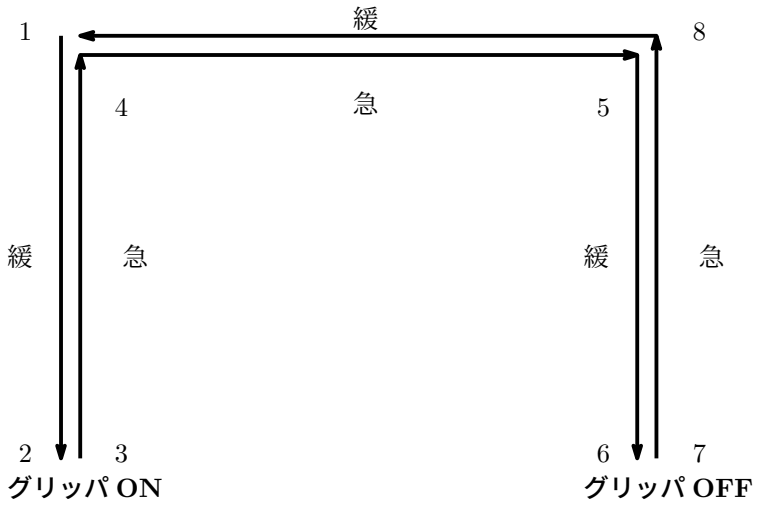


図 19: 実験 II の経路

(a) ホストコンピュータで「NiryoOneStudio」を実行する。

(b) ロボットを起動させる。

(c) ホストコンピュータとロボットとの Wi-Fi 接続を開始する。

(d) ロボットのキャリブレーションを行う。

2. 保存されたプログラム用画面 に切り替え「Test sequence」を実行する。エラーが表示されなかったら、次へ進む。

3. プログラム作成画面 を選び、Learning Mode を ON にする（図 12 参照）。

(a) ロボットアームを図 12 のような姿勢にし、Learning Mode を OFF 状態にする。

(b) プログラム作成画面（図 18 参照）の左下 をクリックし、プログラムステップを作成する。

(c) Learning Mode を ON にし、Shoulder を垂直位置と Roll を水平位置へ動かし、Learning Mode を OFF 状態にする。

(d) プログラムの次のステップを作成する。そのステップを前のステップの後に配置させる。

(e) Waist をおよそ 90° 回転させ、プログラムの次のステップを作成し、前のステップの後に配置させる。

(f) プログラムを実行する（図 18 中の「8」をクリック）。

4. ステップ 3 同様に、他のプログラムを作成する（任意）。

5. 「while」ループでステップ 4 のプログラムを囲み、「while」の実行回数を 3 回にしてからプログラムを実行する。

5.2 実験 II：プログラムの作成

図 19 示すような経路に沿った動作を実現する。そのときにプラスチック瓶を搬送させる。

6 報告事項

1. 実験の目的について述べ、ロボットの構成図を作成し、簡単な動作の説明をする。

2. 各関節の分解能を計算し、結果を表にまとめる。

3. 図 20 での姿 う先端までの長さを
   図 4 を参考

4. 考察

図 20: 課題 3 の画像

A　エンドエフェクター

本ロボットに装着できるエンドエフェクタは、グリッパ（図 21、真空ポンプ（図 22)、電磁石（図 23）などである。

図 21: グリッパ

図 22: 真空ポンプ

図 23: 電磁石

図 24: ロボットの各インターフェイス

B　ロボットのインターフェイス

本ロボットの裏側にいくつかのインターフェイスコネクタがあり、ここでその概要と注意事項について説明する（図 24 参照）。

1. トップボタン
2. Ethernet connector (Raspberry Pi 3B)
3. USB コネクタ ×4
4. LED
5. CAN バス用コネクタ ×2
6. バキュームポンプ（未使用）用コネクタ
7. オプションコネクタ（未使用）
8. 12V スィッチ出力（ソフトウエアにより操作可）
9. GPIO コネクタ ×2。6 つのデジタル入出力（ソフトウエアにより操作可）
10. 電源スイッチ
11. アダプタ接続用コネクタ

入出力インターフェイスには他の機器との接続用入出力ポートがあり、外部信号によって同期させることが出来る。

参考文献

[1] Yoram Koren (藤田 辰二 訳): 技術者のためのロボット工学, 日経ＢＰ社（1986)。

[2] John J.Craig (三浦 宏文・下山 勲 訳) : ロボティクス（機構、力学、制御）, 共立出　版株式会社（1991)。

[3] ROS: Powering the world's robots. [Online]. Available: https://www.ros.org/.

[4] Paspberry Pi. [Online]. Available: https://ja.wikipedia.org/wiki/Raspberry_Pi.

[5] Scratch について. [Online]. Available: https://scratch.mit.edu/.

[6] Blockly: A JavaScript library for building visual programming editors. [Online]. Available: https://developers.google.com/blockly/.

Memo