

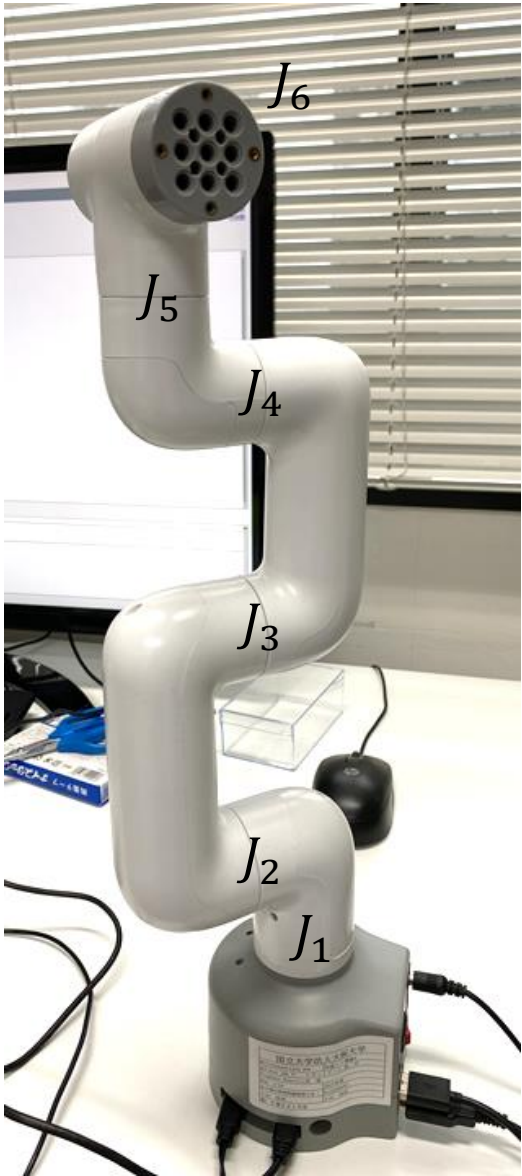
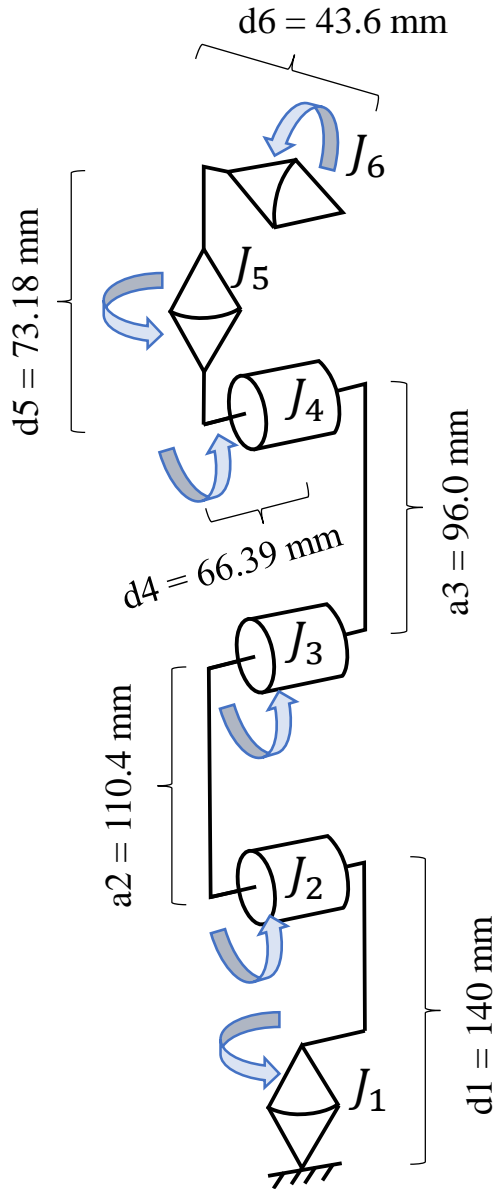
演習一回目
(一日目)

補足情報：ロボットアームの外観と関節配置，リンク長

- アーム名：myCobot 280 Pi
- 6関節6自由度
- 可搬重量：250 g
- 作業半径：280 mm

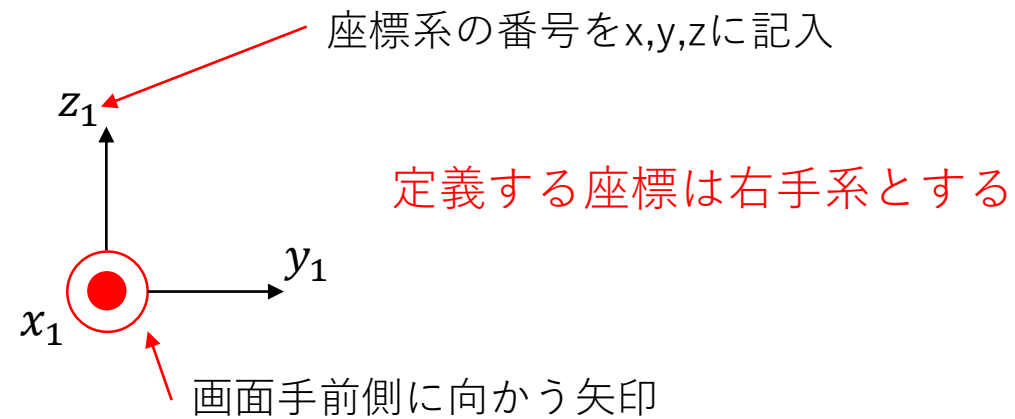
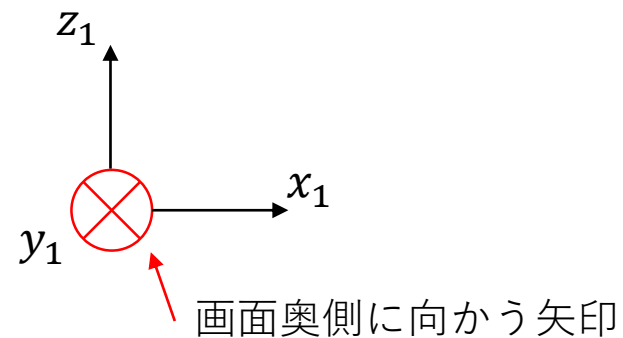
関節名	可動範囲[degree]
J1関節	-90～90
J2関節	-120～120
J3関節	-150～150
J4関節	-120～120
J5関節	-120～120
J6関節	-90～90

※安全のためハードウェアリミットより小さく設定



補足情報：演習解答する際の注意事項

■ ヒントに登場する座標系の表記



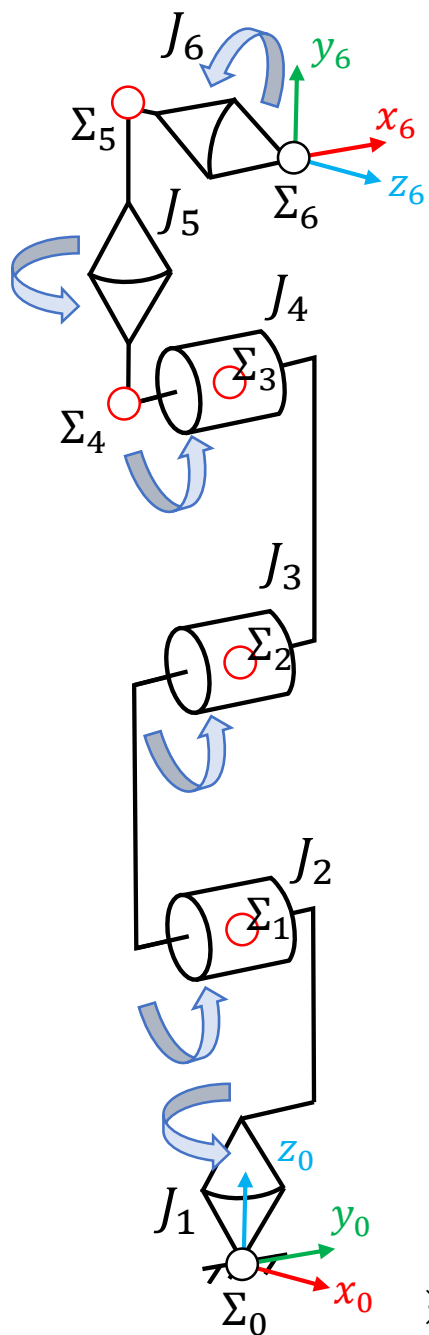
■ 計算過程，最終解答共に三角関数の略記を用いること

$$\sin \theta_1 \rightarrow S_1 \qquad \sin(\theta_1 + \theta_2) \rightarrow S_{12}$$

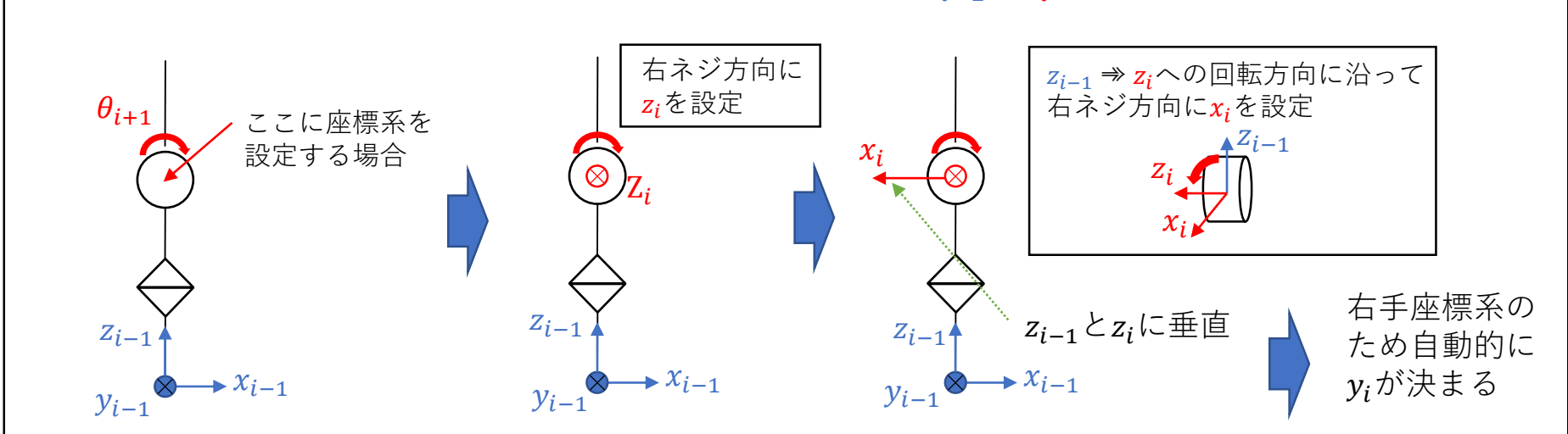
$$\cos \theta_2 \rightarrow C_2 \qquad \cos(\theta_1 + \theta_2) \rightarrow C_{12}$$

※加法定理を用いてまとめられる部分は必ず整理すること

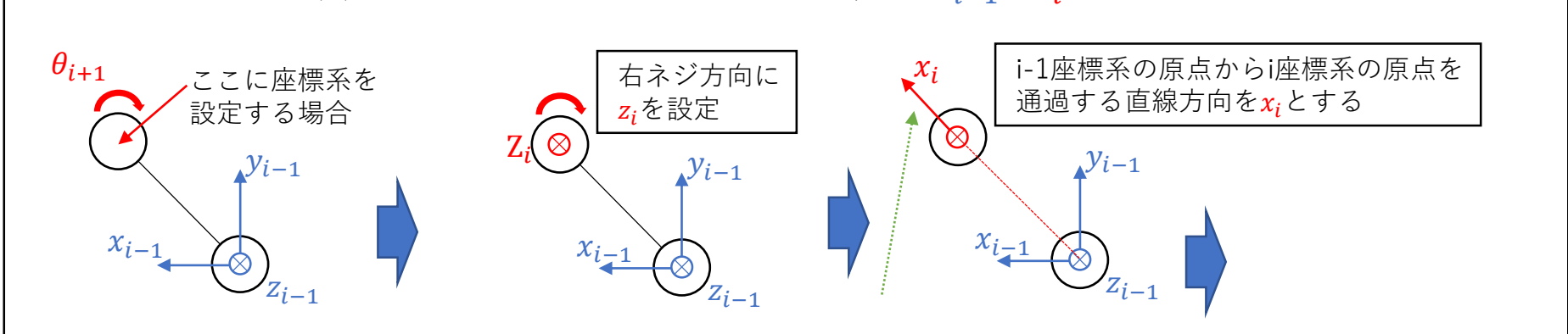
問1：指導書4.2.2のDH法による座標定義を参照し，リンク間の座標 $\Sigma_i(i = 1, \dots, 5)$ を定義せよ



【例】指導書の「(a) 関節 $i+1$ が回転関節の場合」，「 z_{i-1} と z_i が交錯する時」



【例】指導書の「(a) 関節 $i+1$ が回転関節の場合」，「 z_{i-1} と z_i が平行の時」



※ x_0 と z_6 は平行， y_0 と x_6 は平行， z_0 と y_6 は平行

問2： 指導書4.2.3を参照し， リンクパラメータを求め， 同次変換行列 ${}^{i-1}T_i(i = 1, \dots, 6)$ を定義せよ

	d_i	θ_i	l_i	α_i
0T_1		θ_1		
1T_2		θ_2		
2T_3		θ_3		
3T_4		θ_4		
4T_5		θ_5		
5T_6		θ_6		

${}^6p_t = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$

$\left. \vphantom{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}} \right\}$

Σ_6 座標系の原点,
問7で使用する

$${}^0T_1 = \begin{bmatrix} & & & \\ & & & \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1T_2 = \begin{bmatrix} & & & \\ & & & \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2T_3 = \begin{bmatrix} & & & \\ & & & \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3T_4 = \begin{bmatrix} & & & \\ & & & \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

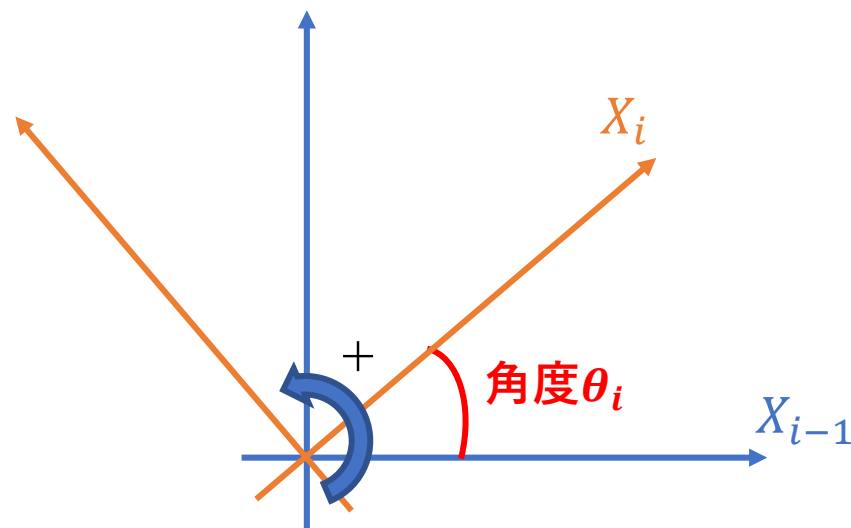
$${}^4T_5 = \begin{bmatrix} & & & \\ & & & \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^5T_6 = \begin{bmatrix} & & & \\ & & & \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

問3：ロボットメーカーが定義した角度 $J_i (i = 1, \dots, 6)$ とDH法で定義した角度 θ_i は一部で角度原点が異なる.
角度 J_i から角度 θ_i に変換するためのオフセット角度（定数）を設定せよ.

$$\left\{ \begin{array}{l} \theta_1 = \frac{J_1}{180} \pi + ? \\ \theta_2 = \frac{J_2}{180} \pi + ? \\ \theta_3 = \frac{J_3}{180} \pi + ? \\ \theta_4 = \frac{J_4}{180} \pi + ? \\ \theta_5 = \frac{J_5}{180} \pi - ? \\ \theta_6 = \frac{J_6}{180} \pi + ? \end{array} \right.$$

DH法の角度 θ_i は、 x_{i-1} 軸と x_i 軸のなす角度



一方、問1の直立姿勢で $J_i = 0 (i = 1, \dots, 6)$

※オフセット角度はゼロの場合もあり.

※プログラム上で J_i は度単位、 θ_i はラジアン単位で扱う

問4：指導書4.3を参考に，座標系 $\Sigma_0 (x_0, y_0, z_0)$ 基準でのアーム手先位置 0p_t を求めよ（順運動学問題）

ヒント：まず ${}^5T_6 {}^6P_t$ を計算し，次に ${}^4T_5 \dots {}^6P_t$ と，式の後部から計算すると単純化される．
なお，次ページのシンボリック計算を用いても良い．

注意：加法定理による簡略化は S_2, c_2, S_3, c_3 に対してのみ行う事

ただし， a_3 を含む項は S_{23}, c_{23} でまとめないように（後の問題を解きやすくするため）

A, B, E, F, H, I には \cos, \sin 関数や d_i の組み合わせ項などが入る．

$${}^0p_t = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} \textcolor{red}{A} c_{\textcolor{blue}{?}} c_{\textcolor{blue}{??}} + \textcolor{blue}{B} c_{\textcolor{blue}{?}} s_{\textcolor{blue}{??}} + (a_2 + a_3 c_3) c_1 c_2 + \textcolor{orange}{E} s_{\textcolor{blue}{?}} - a_3 c_1 s_2 s_3 \\ \textcolor{red}{A} s_{\textcolor{blue}{?}} c_{\textcolor{blue}{??}} + \textcolor{blue}{B} s_{\textcolor{blue}{?}} s_{\textcolor{blue}{??}} + (a_2 + a_3 c_3) s_1 c_2 - \textcolor{orange}{E} c_{\textcolor{blue}{?}} - a_3 s_1 s_2 s_3 \\ \textcolor{red}{A} s_{\textcolor{blue}{??}} - \textcolor{blue}{B} c_{\textcolor{blue}{??}} + (a_2 + a_3 c_3) s_2 + H + I \end{bmatrix}$$

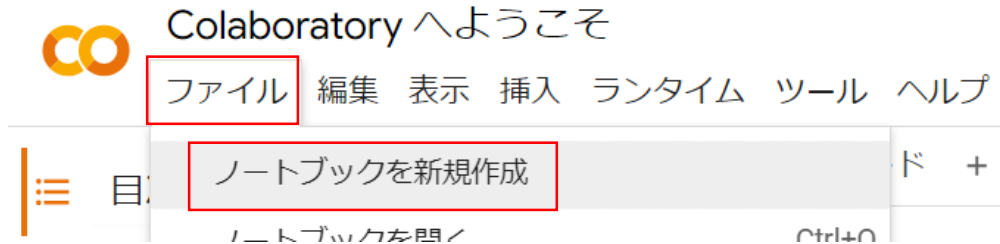
※ $\textcolor{red}{A}, \textcolor{blue}{B}, \textcolor{orange}{E}$ は複数項含む， H, I は各一項

問4補足：pythonなどのシンボリック計算による導出を利用しても良い。

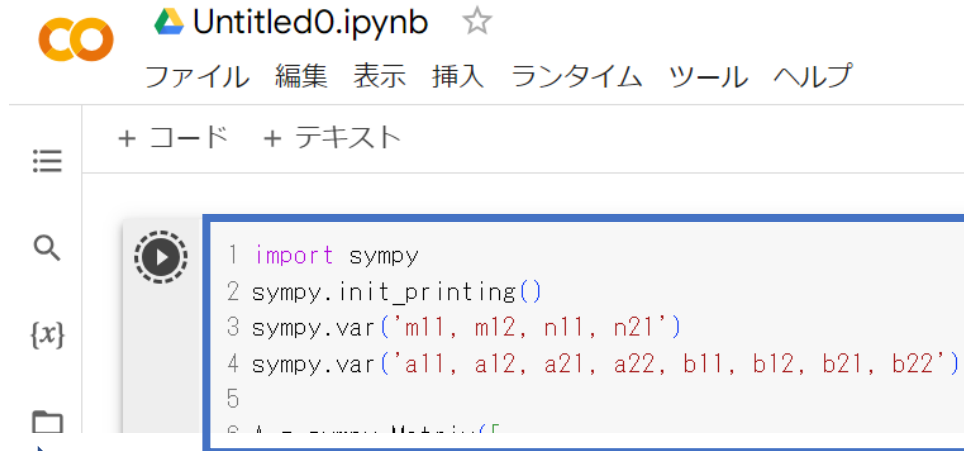
1. Googleアカウントにログイン後，以下にアクセス
<https://colab.research.google.com/?hl=ja>



2. ファイルからノートブックを新規作成



3. セルにコードを入力して再生をクリック



■ プログラム例： 2×2 行列の積

```
import sympy
sympy.init_printing()
sympy.var('a11, a12, a21, a22, b11, b12, b21, b22')

A = sympy.Matrix([
[a11,a12],
[a21,a22]
])

B = sympy.Matrix([
[b11,b12],
[b21,b22]
])

print(A*B)
```

2×2行列の積が表示される。活用して順運動学を求めて良い（ただし加法定理による式の簡略化は手計算する必要あり）。

実験一回目 (二日目)

まず、「開発環境構築__実験B__ロボットアーム.pdf」を参照し、
プログラミング開発環境を構築すること

プログラム1：sample.pyをPCで実行しシミュレーション上で動作を確認せよ。また、故意に角度範囲を超える入力値を与えよ。エラー内容を確認し、AngleErrorの挿入箇所とmain関数内のtryとexcept構文の働きを理解せよ。

※エラー処理はプログラム4で必要になるため必ず理解して進むこと

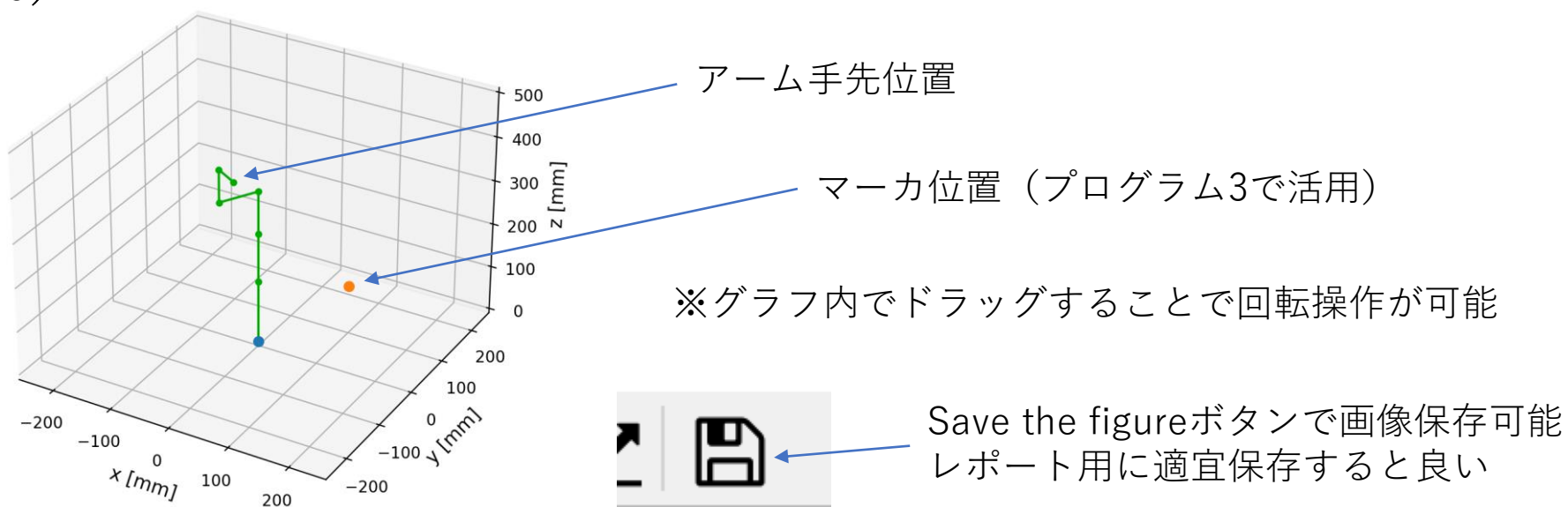
■sample.py補足説明

```
mode select:  
* 0 -> value check  
* 1 -> simulation  
* 2 -> move mode
```

- ・プログラム実行後、0,1,2のいずれかを端末で入力、enterすることでモード選択
- ・0は数値結果のみ表示、1はグラフ上でロボットアーム全体の姿勢を簡易表示
- ・2はロボットアーム本体にプログラムを転送して実行する際の動作モード。
現時点では実行するとエラーが発生するため選択しないこと

■実行例（入力角度-30）

```
input J[0]:  
-30  
J1:  -30.0  
J2:   0.0  
J3:   0.0  
J4:   0.0  
J5:   0.0  
J6:   0.0  
angle_check... OK
```



プログラム2： 6 関節角度を入力し，全関節を同時に動作させるプログラムを作成せよ．
(ヒント：for文を使用して6関節角度を入力させ，moveto関数を一度だけ実行する)

プログラム3：順運動学により，入力した関節角度値から手先位置を計算するプログラムを作成せよ．
計算結果を数値で表示し，シミュレータ上でマーカ位置と手先位置が一致することを確認せよ．
なお，本処理は関数化すること（入力：関節角度，返り値：手先位置）．

■入力例1

$$\left\{ \begin{array}{l} J_1 = -12.08 \\ J_2 = -51.17 \\ J_3 = -123.8, \\ J_4 = 84.96 \\ J_5 = 0.0 \\ J_6 = -12.08 \end{array} \right.$$



$$(p_x, p_y, p_z) \\ = (150, -100, 70)$$

■入力例2

$$\left\{ \begin{array}{l} J_1 = -34.7 \\ J_2 = -63.36 \\ J_3 = -119.18 \\ J_4 = 92.54 \\ J_5 = 0.0 \\ J_6 = -34.7 \end{array} \right.$$



$$(p_x, p_y, p_z) \\ = (100, -150, 50)$$

■入力例3

$$\left\{ \begin{array}{l} J_1 = 26.27 \\ J_2 = -26.46 \\ J_3 = -146.28 \\ J_4 = 82.74 \\ J_5 = 0.0 \\ J_6 = 26.27 \end{array} \right.$$



$$(p_x, p_y, p_z) \\ = (150, 0, 100)$$

■入力例4

$$\left\{ \begin{array}{l} J_1 = 55.30 \\ J_2 = -51.17 \\ J_3 = -123.80 \\ J_4 = 84.96 \\ J_5 = 0.0 \\ J_6 = 55.30 \end{array} \right.$$



$$(p_x, p_y, p_z) \\ = (150, 100, 70)$$

■入力例5

$$\left\{ \begin{array}{l} J_1 = 77.92 \\ J_2 = -63.36 \\ J_3 = -119.18 \\ J_4 = 92.54 \\ J_5 = 0.0 \\ J_6 = 77.92 \end{array} \right.$$



$$(p_x, p_y, p_z) \\ = (100, 150, 50)$$

※PC環境に依存して小数点第3位以下で計算誤差は生じる

プログラム4：入力関節角度から順運動学を計算し、Z方向手先位置が15mm以下($p_z < 15.0$) の場合にエラーが発生するようにせよ．具体的には、Z_ERRORを定義し、条件分岐でエラーをraiseすることで“ $p_z < 15$ ”などのエラーメッセージを文字表示すること．
(以下の各例でのエラーをメモしレポートに記載すること)

■入力例1(OK)

$$\left\{ \begin{array}{l} J_1 = 63.24 \\ J_2 = -41.72 \\ J_3 = -103.45 \\ J_4 = 55.17 \\ J_5 = 0.0 \\ J_6 = 63.24 \end{array} \right.$$



$$(p_x, p_y, p_z) \\ = (150, 150, 100)$$

■入力例2(NG)

$$\left\{ \begin{array}{l} J_1 = 63.24 \\ J_2 = -85.92 \\ J_3 = -83.17 \\ J_4 = 79.09 \\ J_5 = 0.0 \\ J_6 = 63.24 \end{array} \right.$$



$$(p_x, p_y, p_z) \\ = (150, 150, 10)$$

■入力例3(NG)

$$\left\{ \begin{array}{l} J_1 = 145.3 \\ J_2 = -63.36 \\ J_3 = -119.18 \\ J_4 = 92.54 \\ J_5 = 0.0 \\ J_6 = 145.3 \end{array} \right.$$



$$(p_x, p_y, p_z) \\ = (-100, 150, 50)$$

■入力例4(NG)

$$\left\{ \begin{array}{l} J_1 = 41.6 \\ J_2 = -118.04 \\ J_3 = -155.24 \\ J_4 = 183.28 \\ J_5 = 0.0 \\ J_6 = 41.6 \end{array} \right.$$



$$(p_x, p_y, p_z) \\ = (100, 50, 50)$$

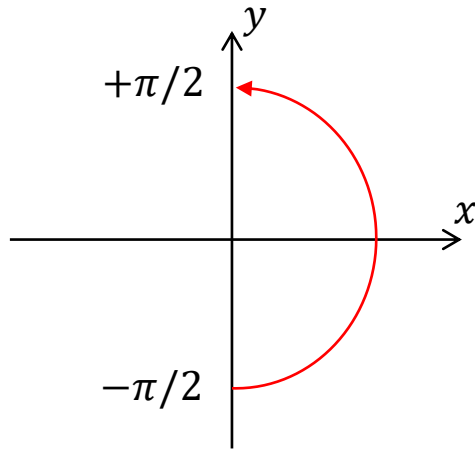
プログラム5：実機でプログラム4の動作を確認せよ.

※実機での動作方法は「実機でのロボット動作手順__実験B__ロボットアーム.pdf」を参照せよ.

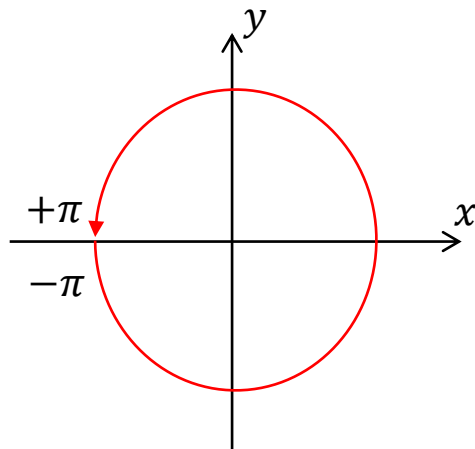
演習 2 回目
(三日目)

補足情報：atan2関数について

■ 逆正接 $\text{atan}(y/x)$ は、 y および x の値に基づいて閉区間 $[-\pi/2, +\pi/2,]$ の値を返す



■ 4象限逆正接 $\text{atan 2}(y, x)$ は、 y および x の値に基づいて閉区間 $[-\pi, +\pi]$ の値を返す

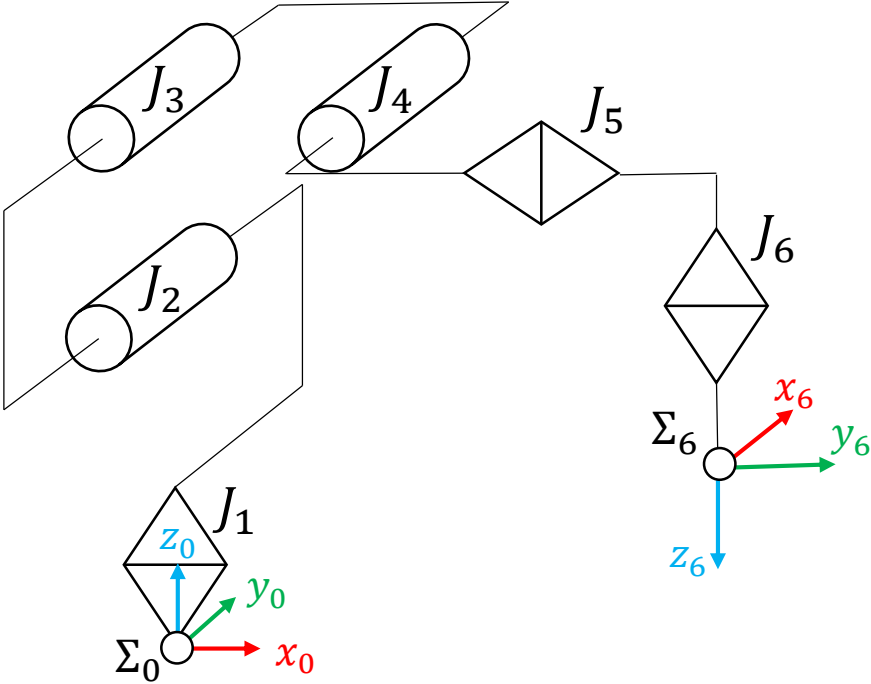


- ・ $\text{atan 2}(y, x)$ の方が値域が広いためこちらを使用することが多い
- ・ 演習の解答では $\text{atan}(y/x)$ 関数部分は $\text{atan 2}(y, x)$ と表記すること
(ただし y, x が定数の場合は asin , acos , atan を使用してOK)

参考： <https://jp.mathworks.com/help/matlab/ref/atan2.html>

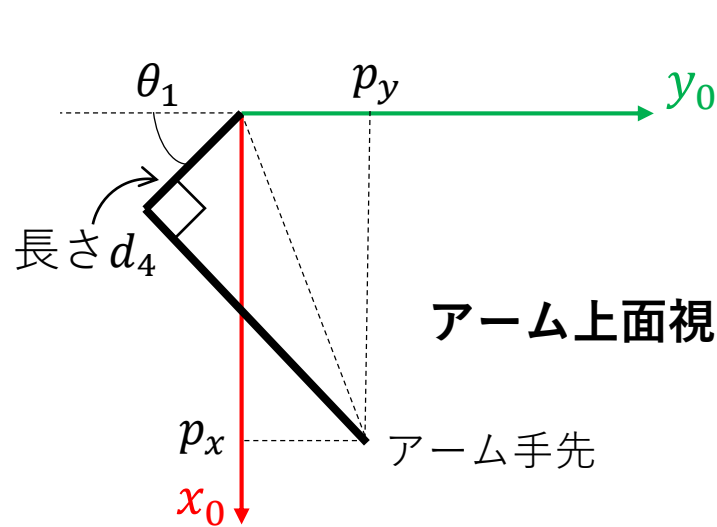
問5：図のアーム手先姿勢に固定した場合の手先位置 0p_t を $\theta_1, \theta_2, \theta_3$ を使って表せ（順運動学の簡略化）．
なお，図の姿勢を維持する拘束条件は， $\theta_4 = \frac{\pi}{2} - \theta_2 - \theta_3$ ， $\theta_5 = -\frac{\pi}{2}$ ， $\theta_6 = \theta_1$ である．

${}^0p_t = \begin{bmatrix} \\ \\ \end{bmatrix}$ $\swarrow \theta_1, \theta_2, \theta_3 \text{ の関数とする}$

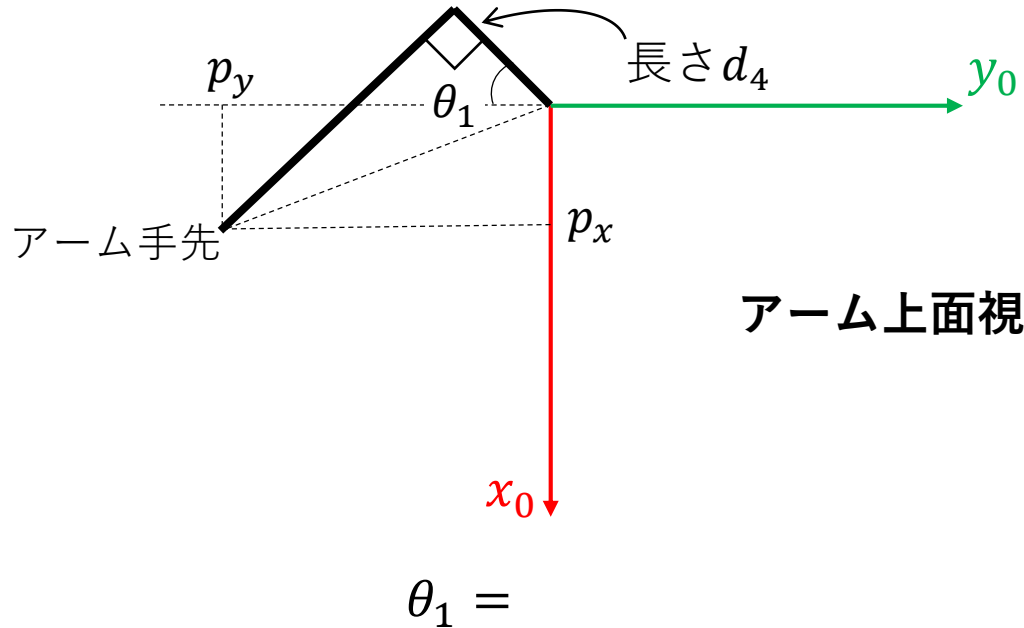


問6：手先位置 $[p_x, p_y, p_z]^T$ が与えられた時の θ_1 を求めよ．なお，解きやすくするために以下の2つのケースに分け，各ケースにおける θ_1 を図の角度関係から求めること．なお p_x は常に正とする．

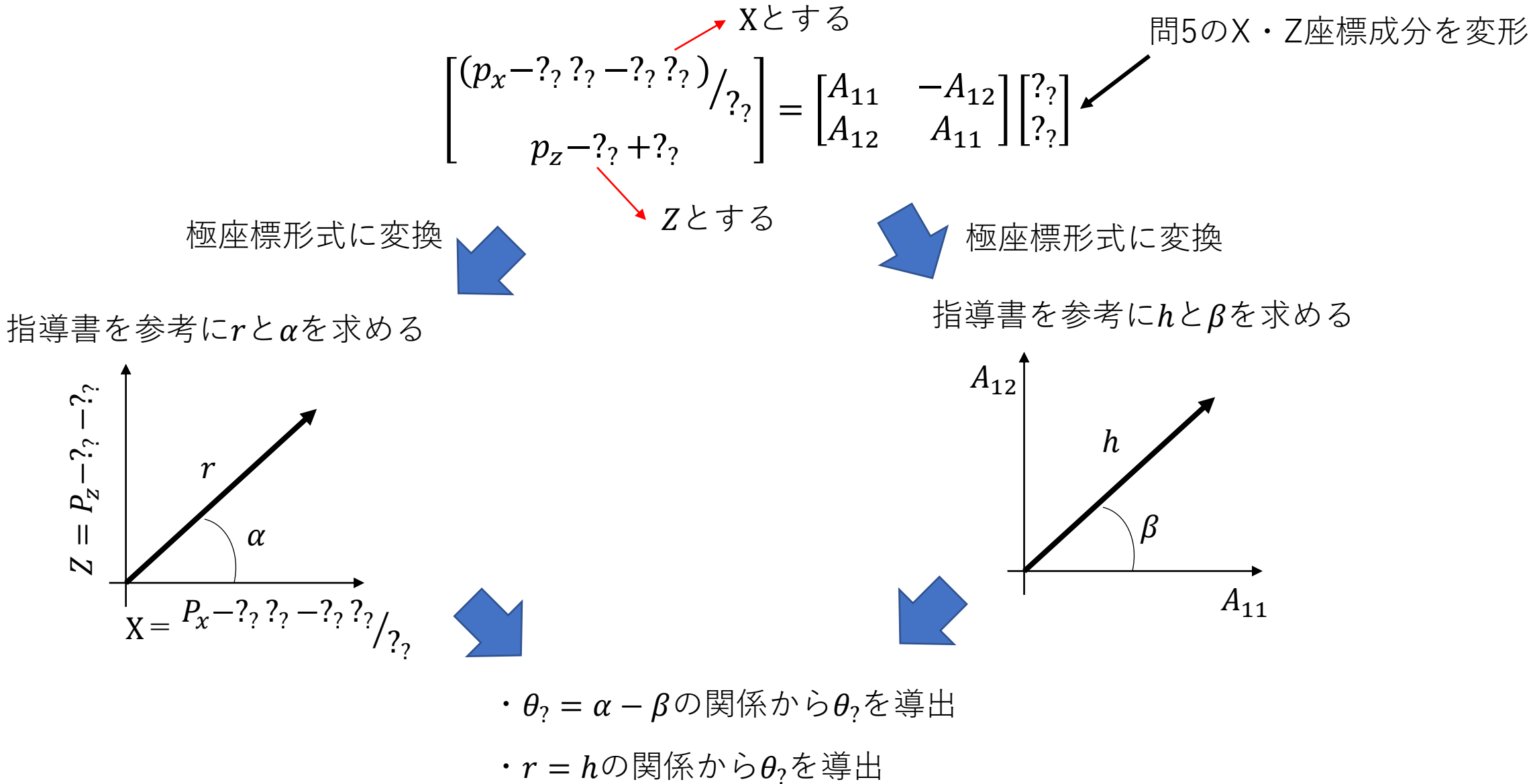
■問6-1： $p_y > 0$ における θ_1 を導出せよ



■問6-2： $p_y \leq 0$ における θ_1 を導出せよ



問7：手先位置 $[p_x, p_y, p_z]^T$ が与えられた時の θ_2, θ_3 を求めよ。
 まず，指導書4.4を熟読し，2リンクの場合での逆運動学の解き方を理解すること。
 その後，以下のヒントを参照して問題を解くように。



実験二回目 (四日目)

プログラム6：逆運動学により，入力した位置に手先を移動させるプログラムを作成し，シミュレータ上で動作を確認せよ（入力値例はプログラム3を参照）．
また，以下の目標位置を入力した場合に発生するエラーを確認し，原因を考察せよ．

※注意： θ_3 の $\pm \text{atan2}$ 関数は「 $-\text{atan2}$ 」で計算すること

■入力例1

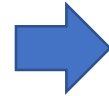
$$(p_x, p_y, p_z) = (0, 0, 100)$$



どういうエラーが出て原因は何か？

■入力例2

$$(p_x, p_y, p_z) = (1000, 1000, 100)$$



“

プログラム7：math関数に関する計算エラーをより分かりやすく表示させたい。
atan2ErrorクラスとsqrtErrorクラスを作成し，atan2とsqrt関数の実行前に
二つの関数が数学的に計算が可能か判定し，計算不可の場合は各エラーをraiseせよ。

※atan2の分母が小さすぎる場合（例えば0.001以下）にatan2Errorを表示，
sqrtの中身が負の場合にsqrtErrorを表示する

プログラム8：初期位置にあるプレートを最終位置に移動させるプログラムを作成せよ.

プログラム9：実機にてプログラム8の動作を確認せよ

プログラム10：（自由課題）LED_control.pyを参考に動作状態に応じてアーム先端のLED色を変化させよ．例えば通常待機時：緑，プログラム実行中：青，エラー発生時：赤など．