第2部　ロボット機構学 (Robot Kinematics)

宇都宮大学工学研究科 吉田勝俊

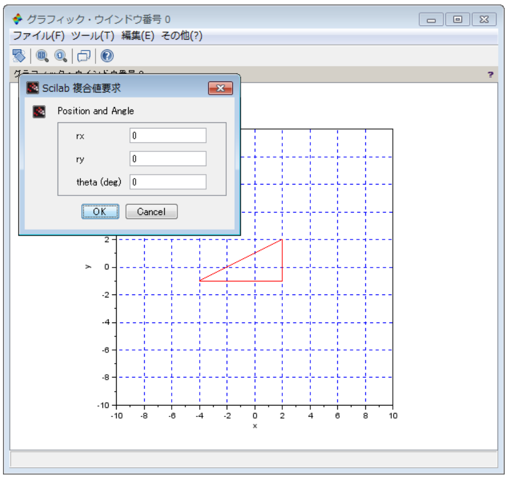
2015.10.1 版

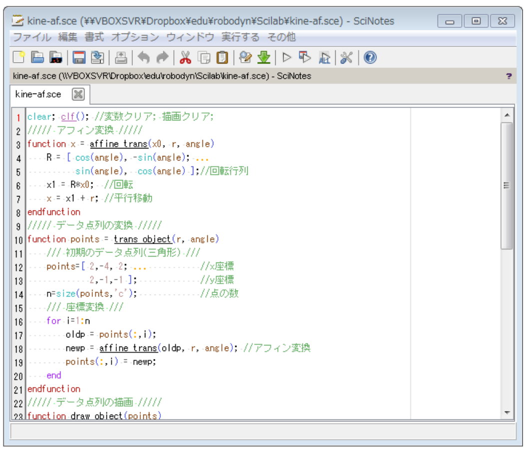
目 次

▶▶ **(Scilab について)**　これ以降に提示するプログラム例は，Scilab というフリーの数値解析ソフトで書かれている．Scilab は市販の Matlab とほぼ同等の機能を持ち，Windows, Linux, Mac OS で安定に動作する．使い方の概要を「Scilab 超入門 - 吉田の教材文庫」

`http://edu.katzlab.jp/lec/scilab`

にまとめておいた．空き時間に自習しておくこと．**自習を前提にレポートを課す．**





5　位置ベクトルと座標系

5.1　ベクトルとその成分

平面上の幾何ベクトル $\boldsymbol{x}$ ↗ (矢印) を考える．矢印は，算数を知らない幼児にも描ける図形であり，成分 (数値) とは無関係に存在する．

こうした図形を数値化するため，図 1 のように，幾何ベクトル $\boldsymbol{x}$ の縦横の寸法 x_1, x_2 を測る．得られた寸法からなる数ベクトル $\tilde{\boldsymbol{x}} = \left[\begin{smallmatrix} x_1 \\ x_2 \end{smallmatrix}\right] = \left[\begin{smallmatrix} 2 \\ 1 \end{smallmatrix}\right]$ を，$\boldsymbol{x}$ の**直交成分** (orthogonal component) という．しかし，寸法の測り方は縦横だけではない．図 2 のように斜めに測ると，同じベクトル $\boldsymbol{x}$ でも寸法の値は変化する．このような，$\boldsymbol{x}$ を対角線とする平行四辺形の 2 辺の寸法 $\tilde{\boldsymbol{x}}' = \left[\begin{smallmatrix} x'_1 \\ x'_2 \end{smallmatrix}\right] = \left[\begin{smallmatrix} 1.43 \\ 1.15 \end{smallmatrix}\right]$ を，$\boldsymbol{x}$ の**斜交成分** (oblique component) という．

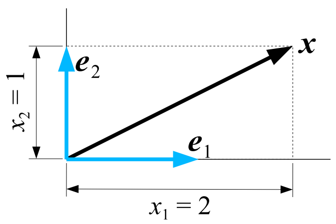


図 1: $\boldsymbol{x}$ の直交成分

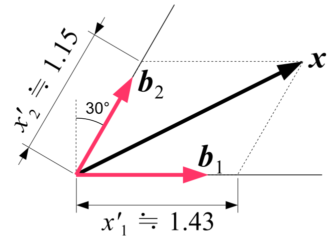


図 2: 同じ $\boldsymbol{x}$ の斜交成分 (図の寸法は実測値で誤差を含む)

このように，幾何ベクトル $\boldsymbol{x}$ の成分とは寸法のことであり，寸法の測り方しだいで成分は変化する．

以上を代数化する．直方体の 2 辺の方向を，単位ベクトル[1] $\boldsymbol{e}_1, \boldsymbol{e}_2$ で表すと，図 1 の測量操作は，次のように数式表現できる．

$$\text{ベクトル } \boldsymbol{x} \quad \overset{\text{展開}}{\Longrightarrow} \quad \boldsymbol{x} = x_1\boldsymbol{e}_1 + x_2\boldsymbol{e}_2 \quad \underset{\text{係数}}{\overset{\text{展開}}{\Longrightarrow}} \quad \text{成分} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \equiv \tilde{\boldsymbol{x}} \tag{5.1}$$

同様に，図 2 については，

$$\text{ベクトル } \boldsymbol{x} \quad \overset{\text{展開}}{\Longrightarrow} \quad \boldsymbol{x} = x'_1\boldsymbol{b}_1 + x'_2\boldsymbol{b}_2 \quad \underset{\text{係数}}{\overset{\text{展開}}{\Longrightarrow}} \quad \text{成分} \begin{bmatrix} x'_1 \\ x'_2 \end{bmatrix} \equiv \tilde{\boldsymbol{x}}' \tag{5.2}$$

である．空間の 3 次元ベクトルについても同様に成分が定義できる．

算法 2.1 (ベクトルの成分表示)　成分測定用のベクトルの組 $\mathcal{E} = \langle \boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3 \rangle$ を，**基底** (basis) という．ベクトル $\boldsymbol{x}$ を基底 $\mathcal{E}$ で展開する．

$$\boldsymbol{x} = x_1\boldsymbol{e}_1 + x_2\boldsymbol{e}_2 + x_3\boldsymbol{e}_3 \quad \underset{\text{係数}}{\overset{\text{展開}}{\Longrightarrow}} \quad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \equiv \tilde{\boldsymbol{x}} \tag{5.3}$$

このときの展開係数 $\tilde{\boldsymbol{x}}$ を，ベクトル $\boldsymbol{x}$ の**成分** (component) という．本書では，$\mathcal{E}$ で測った $\boldsymbol{x}$ の成分を，$[\boldsymbol{x}]_{\mathcal{E}}$ または $\tilde{\boldsymbol{x}}_{\mathcal{E}}$ と書く．

[1] 長さが 1 のベクトル．

実習 2.1

1. 基底 $\mathcal{E} = \langle \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k} \rangle$ で測った，ベクトル $\boldsymbol{v} = 1\boldsymbol{i} + 3\boldsymbol{j} + 2\boldsymbol{k}$ の成分 $\widetilde{\boldsymbol{v}}_{\mathcal{E}}$ を求めよ．
2. 基底 $\mathcal{E} = \langle \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k} \rangle$ で測った，ベクトル $\boldsymbol{w} = 3\boldsymbol{j} + 2\boldsymbol{k}$ の成分 $\widetilde{\boldsymbol{w}}_{\mathcal{E}}$ を求めよ．
3. 基底 $\mathcal{B} = \langle \boldsymbol{i}+\boldsymbol{j}, \boldsymbol{i}-\boldsymbol{j}, \boldsymbol{k} \rangle$ で測った，ベクトル $\boldsymbol{w} = 3\boldsymbol{j} + 2\boldsymbol{k}$ の成分 $\widetilde{\boldsymbol{w}}_{\mathcal{B}}$ を求めよ．

5.2 基底の回転

平面上に正規直交基底 $\mathcal{E} = \langle \boldsymbol{i}, \boldsymbol{j} \rangle$ をとる [2]．これを右ねじ方向（$\boldsymbol{i}$ を $\boldsymbol{j}$ に回す向き）に角度 θ だけ回したものを

$$R_\theta(\mathcal{E}) := \langle R_\theta(\boldsymbol{i}), R_\theta(\boldsymbol{j}) \rangle \tag{5.4}$$

と書く．このとき，作図より，

$$R_\theta(\boldsymbol{i}) = \cos\theta \boldsymbol{i} + \sin\theta \boldsymbol{j}, \quad R_\theta(\boldsymbol{j}) = -\sin\theta \boldsymbol{i} + \cos\theta \boldsymbol{j} \tag{5.5}$$

となる．

実習 2.2 ノートに図示せよ．

ここで，$R_\theta(\mathcal{E})$ で測った $\boldsymbol{x}$ の成分を $\widetilde{\boldsymbol{x}}_{R_\theta(\mathcal{E})} = (x', y')^T$ とすると，

$$\begin{aligned} \boldsymbol{x} &= x' R_\theta(\boldsymbol{i}) + y' R_\theta(\boldsymbol{j}) = x'(\cos\theta \boldsymbol{i} + \sin\theta \boldsymbol{j}) + y'(-\sin\theta \boldsymbol{i} + \cos\theta \boldsymbol{j}) \\ &= (x'\cos\theta - y'\sin\theta)\boldsymbol{i} + (x'\sin\theta + y'\cos\theta)\boldsymbol{j} \end{aligned} \tag{5.6}$$

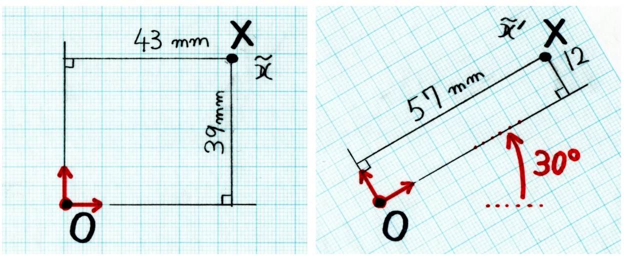
となる．ゆえに，同じベクトル $\boldsymbol{x}$ を元の基底 $\mathcal{E} = \langle \boldsymbol{i}, \boldsymbol{j} \rangle$ で測った成分 $\widetilde{\boldsymbol{x}}_{\mathcal{E}} = (x, y)^T$ は，

$$\widetilde{\boldsymbol{x}}_{\mathcal{E}} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x'\cos\theta - y'\sin\theta \\ x'\sin\theta + y'\cos\theta \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}}_{[R_\theta]} \begin{bmatrix} x' \\ y' \end{bmatrix} = [R_\theta]\, \widetilde{\boldsymbol{x}}_{R_\theta(\mathcal{E})} \tag{5.7}$$

のように表すことができる．行列 $[R_\theta]$ を（2 次元の）**回転行列**という．逆変換は逆回転であるから，次の関係が成立する．

$$[R_\theta]^{-1} = [R_{-\theta}] = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \tag{5.8}$$

実習 2.3 正規直交基底 $\mathcal{E} = \langle \boldsymbol{i}, \boldsymbol{j} \rangle$ と，これを 30° だけ回した基底 $R_{30^\circ}(\mathcal{E}) = \langle R_{30^\circ}(\boldsymbol{i}), R_{30^\circ}(\boldsymbol{j}) \rangle$ をとる．あるベクトル $\boldsymbol{x}$ について，$\widetilde{\boldsymbol{x}}_{R_{30^\circ}(\mathcal{E})} = (57, 12)^T$ であるとき，$\widetilde{\boldsymbol{x}}_{\mathcal{E}}$ を求めよ．



同様の関係式が 3 次元でも成立する．

[2]「正規」とは，基底ベクトルの長さが $|\boldsymbol{i}| = |\boldsymbol{j}| = 1$ であること．

算法 2.2 (基底の回転)　正規直交基底 $\mathcal{E} = \langle \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k} \rangle$ と，これを 3 次元回転させた基底 $R(\mathcal{E}) = \langle R(\boldsymbol{i}), R(\boldsymbol{j}), R(\boldsymbol{k}) \rangle$ をとる．任意の 3 次元ベクトル $\boldsymbol{x}$ について，

$$\widetilde{\boldsymbol{x}}_{\mathcal{E}} = [R]\,\widetilde{\boldsymbol{x}}_{R(\mathcal{E})} \quad \text{または} \quad \widetilde{\boldsymbol{x}}_{R(\mathcal{E})} = [R]^{-1}\,\widetilde{\boldsymbol{x}}_{\mathcal{E}} \tag{5.9}$$

が成立する．$[R]$ は 3 次元の回転を表す 3×3 の**回転行列**である（成分は後述する）.

5.3　3 次元の回転行列

3 次元の回転行列は，9 つの成分：

$$[R] := \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \tag{5.10}$$

を持つが，これらは互いに独立ではない．仮に 9 成分を任意に設定できるなら，ベクトルの長さを変える変換が含まれるが，これはベクトルの長さを保つ回転変換に反する.

5.3.1　回転軸 $\boldsymbol{a}$ まわりの θ 回転

ベクトル $\boldsymbol{x}$ と同じ始点をもつ単位ベクトル $\boldsymbol{a}$ をとる．$\boldsymbol{a}$ を回転軸として，$\boldsymbol{x}$ を右ねじ方向に角度 θ だけ回転させる変換は，若干手の込んだ作図より，

$$\boldsymbol{y} = R(\boldsymbol{x}) := (\boldsymbol{a}\cdot\boldsymbol{x})\boldsymbol{a} + \cos\theta\{\boldsymbol{x} - (\boldsymbol{a}\cdot\boldsymbol{x})\boldsymbol{a}\} + \sin\theta(\boldsymbol{a}\times\boldsymbol{x}) \tag{5.11}$$

と表せる．$\cdot$ は内積，$\times$ はクロス積である．正規直交基底 $\mathcal{E} = \langle \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k} \rangle$ に対して，

$$\boldsymbol{x} = x_1\boldsymbol{i} + x_2\boldsymbol{j} + x_3\boldsymbol{k}, \quad \boldsymbol{a} = a_1\boldsymbol{i} + a_2\boldsymbol{j} + a_3\boldsymbol{k} \tag{5.12}$$

と展開したものを (5.11) に代入し，内積とクロス積の性質，

$$\boldsymbol{i}\cdot\boldsymbol{j} = \boldsymbol{j}\cdot\boldsymbol{k} = \boldsymbol{k}\cdot\boldsymbol{i} = 0, \quad \boldsymbol{i}\cdot\boldsymbol{i} = \boldsymbol{j}\cdot\boldsymbol{j} = \boldsymbol{k}\cdot\boldsymbol{k} = 1,$$

$$\boldsymbol{i}\cdot\boldsymbol{j} = \boldsymbol{j}\times\boldsymbol{k} = \boldsymbol{k}\times\boldsymbol{i} = 0, \quad \boldsymbol{i}\cdot\boldsymbol{j} = \boldsymbol{k}, \quad \boldsymbol{j}\times\boldsymbol{k} = \boldsymbol{i}, \quad \boldsymbol{k}\times\boldsymbol{i} = \boldsymbol{j}, \quad \boldsymbol{u}\times\boldsymbol{v} = -\boldsymbol{v}\times\boldsymbol{u}$$

で整理する．得られた結果を，$\boldsymbol{y} = y_1\boldsymbol{i} + y_2\boldsymbol{j} + y_3\boldsymbol{k}$ と等値すると，

$$\widetilde{\boldsymbol{y}}_{\mathcal{E}} = [R(a_1, a_2, a_3, \theta)]\,[\boldsymbol{x}]_{\mathcal{E}}：$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} a_1^2 + (1-a_1^2)C_\theta & a_1a_2(1-C_\theta) - a_3S_\theta & a_1a_3(1-C_\theta) + a_2S_\theta \\ a_1a_2(1-C_\theta) + a_3S_\theta & a_2^2 + (1-a_2^2)C_\theta & a_2a_3(1-C_\theta) - a_1S_\theta \\ a_1a_3(1-C_\theta) - a_2S_\theta & a_2a_3(1-C_\theta) + a_1S_\theta & a_3^2 + (1-a_3^2)C_\theta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{5.13}$$

という成分を得る $(\cos\theta = C_\theta, \sin\theta = S_\theta)$．この表現では，回転行列の 9 成分 $R_{11}, R_{12}, \cdots, R_{33}$ は各々4 つのパラメータ a_1, a_2, a_3, θ で表されるが，単位ベクトルの制約条件 $a_1^2 + a_2^2 + a_3^2 = 1$ があるので，代数的には 3 つ a_i, a_j, θ を指定すれば残りが決まる.

特にロボット・マニピュレータの議論では，回転軸を x 軸：$\widetilde{\boldsymbol{a}} = (1,0,0)^T$，$y$ 軸：$\widetilde{\boldsymbol{a}} = (0,1,0)^T$，$z$ 軸：$\widetilde{\boldsymbol{a}} = (0,0,1)^T$ にとった次の回転行列がよく登場する.

$$[R_x(\theta)] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C_\theta & -S_\theta \\ 0 & S_\theta & C_\theta \end{bmatrix}, \ [R_y(\theta)] = \begin{bmatrix} C_\theta & 0 & S_\theta \\ 0 & 1 & 0 \\ -S_\theta & 0 & C_\theta \end{bmatrix}, \ [R_z(\theta)] = \begin{bmatrix} C_\theta & -S_\theta & 0 \\ S_\theta & C_\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.14}$$

5.3.2 オイラー角

(5.14) の行列の掛け合わせで表した回転行列を，**オイラー角**という．

- **Z-Y-Z 型（狭義のオイラー角）** $[\mathrm{Euler}(\phi,\theta,\psi)] := [R_z(\phi)]\,[R_y(\theta)]\,[R_z(\psi)]$
- **X-Y-Z 型（ロール・ピッチ・ヨー角）** $[\mathrm{RPY}(\phi,\theta,\psi)] := [R_x(\phi)]\,[R_y(\theta)]\,[R_z(\psi)]$

いずれの方式も，回転行列の成分 $R_{11}, R_{12}, \cdots, R_{33}$ が取りうる全ての値を ϕ, θ, ψ の組み合わせで表せる．用途に応じて，X,Y,Z の他の組み合わせも使用される．

Z-Y-Z 型は，コマの姿勢を表すのに適している．z 軸まわりに ψ だけスピンさせたコマを θ だけ傾け，これを z 軸まわりに ϕ だけ公転させた姿勢を表す．X-Y-Z 型は，車両（または航空機）の姿勢表現によく使われる．x 軸を進行方向，z を鉛直上向きとして，操舵による回頭 ψ をヨー角，左右への傾き θ をロール角，前後への傾き ϕ をピッチ角という．

このように，オイラー角は 3 つのパラメータ（必要最小限）で回転行列を表せるが，**ジンバルロック**という不具合を引き起こす欠点がある．

> ▶▶ **(ジンバルロック)** ある操作盤に，ϕ 用のツマミ，θ 用のツマミ，ψ 用のツマミがあり，これらを操作すると，X-Y-Z 型のオイラー角にしたがって物体の姿勢を操作できるとする．ツマミが 1 つ壊れても，残りの 2 方向は操作できるはずだが，そうならない場合がある．例えば，θ のツマミが $\pi/2$ の位置で壊れたとする．このとき，残りの ψ と ϕ の回転軸が重なるので，ψ と ϕ をどのように操作しても，1 軸まわりの回転しか起こせない．すなわち，実質的なツマミが 1 つ減る．このような不具合をジンバルロックという．シンバルロックを引き起こす $\theta = \pi/2$ のような姿勢を**特異点**という．

5.3.3 オイラーパラメータ（クォータニオン）

人工衛星の姿勢制御だとか，体操選手の「前方かかえ込み 2 回宙返り 1/2 ひねり」など，ものすごいグルグル回る対象を計算したいとき，オイラー角のジンバルロックは致命的である．ジンバルロックを解消するには，(5.13) の回転軸 $\widetilde{\boldsymbol{a}} = (a_1, a_2, a_3)$ と回転角 θ に戻ればよい．ただし，そのまま使うことは稀で，面倒な三角関数 $\sin, \cos$ を嫌って，

$$\boldsymbol{q} = (q_0, q_1, q_2, q_3)^T \ := \ \left(\cos\frac{\theta}{2}, a_1 \sin\frac{\theta}{2}, a_2 \sin\frac{\theta}{2}, a_3 \sin\frac{\theta}{2}\right)^T \tag{5.15}$$

という書き直しが行われる．書き直した $\boldsymbol{q}$ を，**オイラーパラメータ** (Euler parameters) もしくは**単位クォータニオン**という．

オイラーパラメータ $\boldsymbol{q}$ を使うと (5.13) の回転行列は，

$$[R(a_1,a_2,a_3,\theta)] = \begin{bmatrix} 1-2q_2^2-2q_3^2 & 2q_1q_2-2q_0q_3 & 2q_0q_2+2q_1q_3 \\ 2q_1q_2+2q_0q_3 & 1-2q_1^2-2q_3^2 & -2q_0q_1+2q_2q_3 \\ -2q_0q_2+2q_1q_3 & 2q_0q_1+2q_2q_3 & 1-2q_1^2-2q_2^2 \end{bmatrix} =: [R(\boldsymbol{q})], \tag{5.16}$$

$$\text{ただし，} q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1. \tag{5.17}$$

に書きかわる ($\because \sin^{\theta} + \cos^2\theta = 1,\ a_1^2 + a_2^2 + a_3^2 = 1$)．成分から三角関数が消えてすっきりしたが，見た目だけではない．(a_1, a_2, a_3, θ) において，$\sin\theta,\ \cos\theta,\ a_1^2 + a_2^2 + a_3^2 = 1$ の果した役割が，1 本の拘束条件 (5.17) に集約されてしまった．というわけで，オイラーパラメータ $\boldsymbol{q}$ を計算するときは，三角関数のことは忘れて，単なる 2 次式であるところの拘束条件 (5.17) だけ念頭に置けばよい．この新しい表現において，ジンバルロックは起らない．

6　多体系の運動学 (Multibody Kinematics)

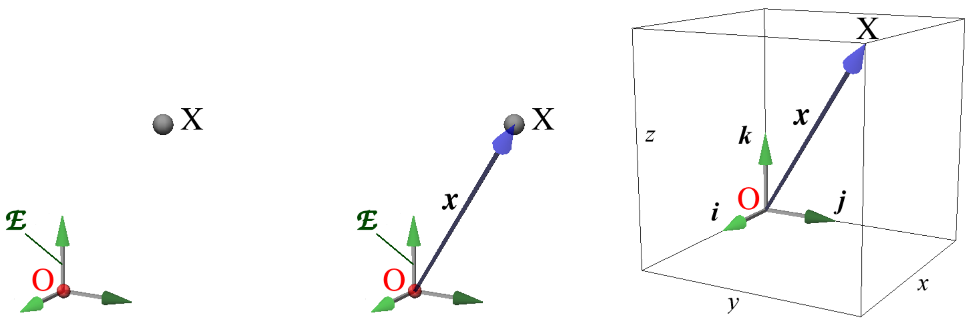
ロボットの各関節が時々刻々と角度を変え，手先がある運動経路を実現する．このような複数の部品からなる構造を，**多体系** (multibody system) という．ここでは，このような多体系の姿勢を数式表現する．最後に，仮想マニピュレータをキーボードで操作する．

6.1　座標系と空間座標

点 X の空間座標 $\widetilde{\mathrm{X}}_{(\mathrm{O},\mathcal{E})}$

1. 測定基準点 O と基底 $\mathcal{E}$ を選ぶ．その組 $(\mathrm{O},\mathcal{E})$ を**座標系**という．
2. O〜X に矢印 $\boldsymbol{x}$ を引く．この $\boldsymbol{x}$ を**位置ベクトル**という．
3. $\mathcal{E}$ による $\boldsymbol{x}$ の成分 $\widetilde{\boldsymbol{x}}_{\mathcal{E}} = (x,y,z)^T$ を $(\mathrm{O},\mathcal{E})$ で測った X の**空間座標**という [3]．

$$\text{本書の表記：} \widetilde{\mathrm{X}}_{(\mathrm{O},\mathcal{E})} \equiv \widetilde{\boldsymbol{x}}_{\mathcal{E}} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{6.1}$$



6.2　部品図と組立図

パソコンの階層的なフォルダのように，多体系の姿勢を部品図と組立図で整理する．

6.2.1　部品図 — 局所座標系

まず，多体系を構成する各部品の形状をデータ化する．

部品形状の表現

1. i 番目の部品用の部品図を用意する．
2. その部品図に直交座標系 $\mathcal{E}_i = (\mathrm{O}_i, \mathcal{E}_i)$ を設定する．
   ※部品図に原点 O_i を定め，直交基底 $\mathcal{E}_i$ の方向を決める．
3. 部品の代表点 $\mathrm{P}_1,\cdots,\mathrm{P}_k$ を，$\mathcal{E}_i$ による座標 $\boldsymbol{\xi}^i_1,\cdots,\boldsymbol{\xi}^i_k$ として記録・保持する．
   これらの座標データの集合を $\mathcal{X}_i := \{\boldsymbol{\xi}^i_1,\cdots,\boldsymbol{\xi}^i_k\}$ で表す．

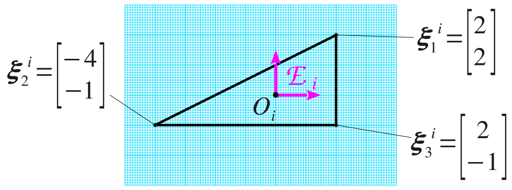
[3] $(\cdots)^T$ は転置を表す．

部品ごとに設定する直交座標系 $\mathcal{E}_i$ のことを，**局所座標系** (local coordinate system; local frame) という．この $\mathcal{E}_i$ で測った部品図上の座標 $\boldsymbol{\xi}_1^i, \cdots, \boldsymbol{\xi}_k^i$ を**局所座標** (local coordinate) という．一つ気付く点として，

- 部品が剛体であれば，その形状を表す局所座標は，永遠不変の定ベクトルとなる．

ということに注意しよう．

方眼紙に部品図を作成した例を以下に示す．一応，i 番目の部品という想定である．



この部品の代表点を 3 頂点として，図の局所座標系で測ると，形状データは，

$$\boldsymbol{\mathcal{X}}_i = \{\boldsymbol{\xi}_1^i, \boldsymbol{\xi}_2^i, \boldsymbol{\xi}_3^i\} = \left\{ \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \begin{bmatrix} -4 \\ -1 \end{bmatrix}, \begin{bmatrix} 2 \\ -1 \end{bmatrix} \right\}$$

となる．3 角形なので 3 頂点としたが，もちろん，別の選び方をしてもよい．

6.2.2　組立図 — 基準座標系

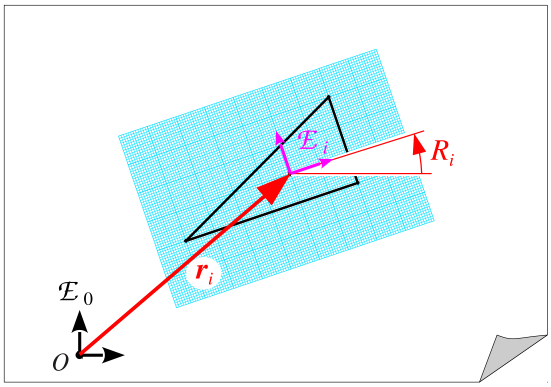


図 3: 組立図上の部品図

部品図が完成したら，図 3 のように，組立図[4] に配置する．トレース紙に書いた部品図を組立図上を滑らせて配置するイメージだ．この状況を次の手順で数式表現する．

部品図の位置と姿勢

1. 組立図用の直交座標系 $\mathcal{E}_0 = (\mathrm{O}, \mathcal{E})$ をとる．
2. 部品図 $\mathcal{E}_i = (\mathrm{O}_i, \mathcal{E}_i)$ の位置を，$\mathcal{E}_0$ 上の位置ベクトル $\boldsymbol{r}_i = \overrightarrow{\mathrm{OO}_i}$ で表す．
3. 部品図 $\mathcal{E}_i = (\mathrm{O}_i, \mathcal{E}_i)$ の姿勢を，回転変換 R_i で表す．

※ R_i は，基準基底 $\mathcal{E}$ を，$\mathcal{E}_i$ になるまで回す変換．$\mathcal{E}_i = R_i(\mathcal{E})$ と表記する．

4. 組立図 $\mathcal{E}_0$ に対する部品図 $\mathcal{E}_i$ の配置を，ペア $(\boldsymbol{r}_i, R_i)$ で表す．

[4] 部品を本来の位置に配置した全体像の図面を「組立図」という．

組立図にとる座標系 $\mathcal{E}_0$ のことを，**基準座標系** (reference coordinate system; reference frame) という．$\mathcal{E}_0$ で測った座標を，**基準座標** (reference coordinate) という．

6.3　部品図から組立図への座標変換

部品図上の局所座標と，組立図上の基準座標は，次の算法で関係づけられる．

算法 2.3　基準座標系 (組立図) に対する局所座標系 (部品図) の配置を $(\boldsymbol{r}, R)$ とする．このとき，局所座標 $\boldsymbol{\xi}$ が指す点は，基準座標，

$$\boldsymbol{x} = R\boldsymbol{\xi} + \boldsymbol{r} \tag{6.2}$$

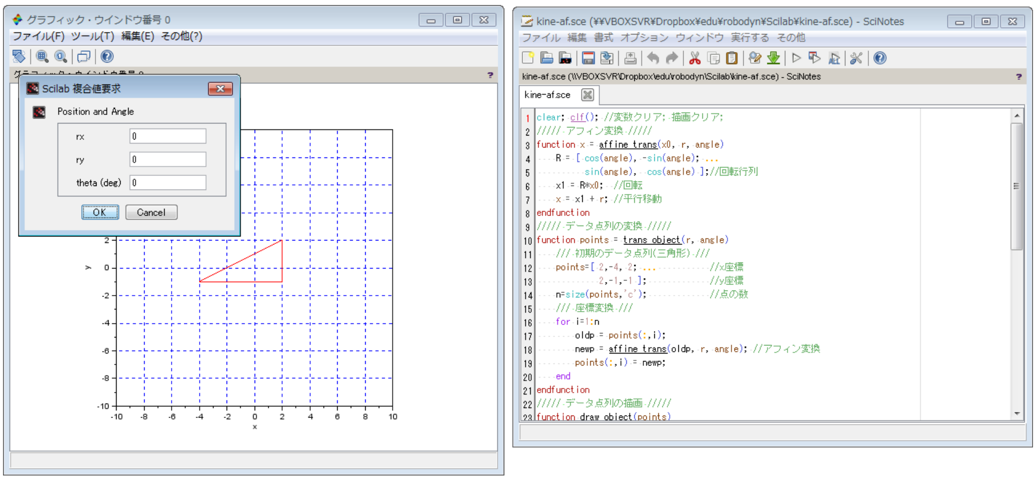
の位置にくる．このような，回してから平行移動する変換のことを，**アフィン変換** (affine trasformation) という [5]．(証明は図 3 p7 より明らか)

算法 2.3 を使うと，複数の部品間の結合や接触が議論できる．2 つの部品 i, j が，互いに結合/接触するということは，両者が空間の同一点を占めていることに他ならないが，局所座標 $\boldsymbol{\xi}_i, \boldsymbol{\xi}_j$ で比較するのは無意味である．なぜなら，部品図を組立図にどう置くかで，結果が変ってしまう．そこで，組立図に対する部品図の配置 $(\boldsymbol{r}_i, R_i), (\boldsymbol{r}_j, R_j)$ の情報を使って，算法 2.3 により，両者の基準座標，

$$\boldsymbol{x}^i = R_i\boldsymbol{\xi}^i + \boldsymbol{r}_i, \quad \boldsymbol{x}^j = R_j\boldsymbol{\xi}^j + \boldsymbol{r}_j \tag{6.3}$$

をとる．このとき，$\boldsymbol{x}^i = \boldsymbol{x}^j$ は組立図上の同じ点を表す．このような共通の基準座標をもつ 2 つの部品は，結合/接触の状態にあるといえる．

実習 2.4　**Code 1** p14 を実行せよ．ダイアログボックスに $\boldsymbol{r}$ の成分と回転角を入力すると，**6.2.2 節** p7 の組立図上の三角形が実際に平行移動・回転する．



[5] 同じ affine を「アファイン」と読む本も多い．

7 ロボット運動学

7.1 アフィン変換の行列表示

アフィン変換 (6.2) p8 を，成分が見えるように書くと，

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} + \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \quad ※ 2 次元 \tag{7.1a}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix} + \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} \quad ※ 3 次元 \tag{7.1b}$$

のような格好をしている．ここで先人のアイデアだが，以上の変換則は，

$$\begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} R_{11} & R_{12} & r_1 \\ R_{21} & R_{22} & r_2 \\ 0 & 0 & 1 \end{bmatrix}}_{H} \begin{bmatrix} \xi_1 \\ \xi_2 \\ 1 \end{bmatrix} \quad ※ 2 次元 \tag{7.2a}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} R_{11} & R_{12} & R_{13} & r_1 \\ R_{21} & R_{22} & R_{23} & r_2 \\ R_{31} & R_{32} & R_{33} & r_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{H} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ 1 \end{bmatrix} \quad ※ 3 次元 \tag{7.2b}$$

のようにすると，単独の行列 H で表せる．この変換式 (7.2) を**同次変換** (homogeneous transformation) といい，行列 H を**同次変換行列**という [6]．ちなみに，ベクトルの末尾に追加した “1” は，こうした算法上のトリックを実現するためのダミー成分であり，物理的な意味はない．実用上は，注目するベクトル $\boldsymbol{x}$ の末尾に “1” を追加して，同次変換を行い，同次変換が済んだら，末尾の “1” を取り除く．

実習 2.5 行列とベクトルの積を実行し，(7.2a) の上 2 行が (7.1a) に一致すること，(7.2b) の上 3 行が (7.1b) に一致することを示せ．

ロボット工学や 3 次元 CG(Computer Graphics) の分野では，表 1〜表 2 p10 のような同次変換行列がよく使われる．

実習 2.6 実習 2.4 のアフィン変換を，同次変換に書き換えよ．これを実行せよ．

▶ **解答例** 書き変えたプログラム例を **Code 2** p14 に示す．動作は実習 2.4 と同じである．

表 1: 同次変換行列 H の例 (2 次元) ※ $C_\theta := \cos\theta$, $S_\theta := \sin\theta$

作用	表記	成分
平行移動	$\mathrm{Trans}([r_i])$ または $\mathrm{Trans}(r_1, r_2)$	$\begin{bmatrix} 1 & 0 & r_1 \\ 0 & 1 & r_2 \\ 0 & 0 & 1 \end{bmatrix}$
回転	$\mathrm{Rot}(\theta)$	$\begin{bmatrix} C_\theta & -S_\theta & 0 \\ S_\theta & C_\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$
拡大縮小	$\mathrm{Scal}(\alpha, \beta)$	$\begin{bmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & 1 \end{bmatrix}$

[6] または，Denavit-Hartenberg (デナビット・ハーテンベルグ) の変換行列ともいう．詳細は吉川 [1] の 1 章など．

表 2: 同次変換行列 H の例 (3 次元)　※ $C_\theta := \cos\theta,\ S_\theta := \sin\theta$

作用	表記	成分
平行移動	$\mathrm{Trans}([r_i])$ または $\mathrm{Trans}(r_1, r_2, r_3)$	$\begin{bmatrix} 1 & 0 & 0 & r_1 \\ 0 & 1 & 0 & r_2 \\ 0 & 0 & 1 & r_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
x 軸回転	$\mathrm{Rot}(x, \theta)$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C_\theta & -S_\theta & 0 \\ 0 & S_\theta & C_\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
y 軸回転	$\mathrm{Rot}(y, \theta)$	$\begin{bmatrix} S_\theta & 0 & C_\theta & 0 \\ 0 & 1 & 0 & 0 \\ C_\theta & 0 & -S_\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
z 軸回転	$\mathrm{Rot}(z, \theta)$	$\begin{bmatrix} C_\theta & -S_\theta & 0 & 0 \\ S_\theta & C_\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
$\boldsymbol{a}$ 軸回転	$\mathrm{Rot}(\boldsymbol{a}, \theta)$	$\begin{bmatrix} & & & 0 \\ & \text{3 次元の回転行列} & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
拡大縮小	$\mathrm{Scal}(\alpha, \beta, \gamma)$	$\begin{bmatrix} \alpha & 0 & 0 & 0 \\ 0 & \beta & 0 & 0 \\ 0 & 0 & \gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

7.2　ロボット・マニピュレータ

図 4 の 4 自由度マニピュレータを例にとる．このマニピュレータは，手首が旋回し，肘

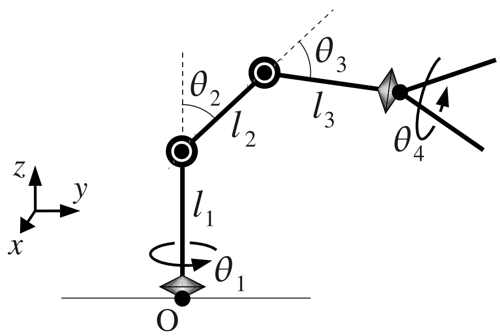


図 4: 4 自由度マニピュレータ

と肩が屈曲し，それ全体が旋回できる．基準座標系 $\mathcal{E}_0 = (O, \mathcal{E})$ の原点 O をマニピュレータの根本にとり，基底 $\mathcal{E}$ は図の向きに取る (x 軸は紙面垂直こちら向き)．

表 1〜表 2 の同次変換を組み合せて，図 4 のマニピュレータの姿勢を表現してみる．手先からたどっていくのがコツである．

【Step 1】　まず，図 5 左上の局所座標系 $\mathcal{E}_1$ に，手先の部品図を作成する．図の 3 点を代表点としよう．手首 $\boldsymbol{\xi}_1^1$ に $\mathcal{E}_1$ の原点 O_1 をとる．

$$\mathcal{X}_1 = \{\boldsymbol{\xi}_1^1, \boldsymbol{\xi}_2^1, \boldsymbol{\xi}_3^1\} \tag{7.3}$$

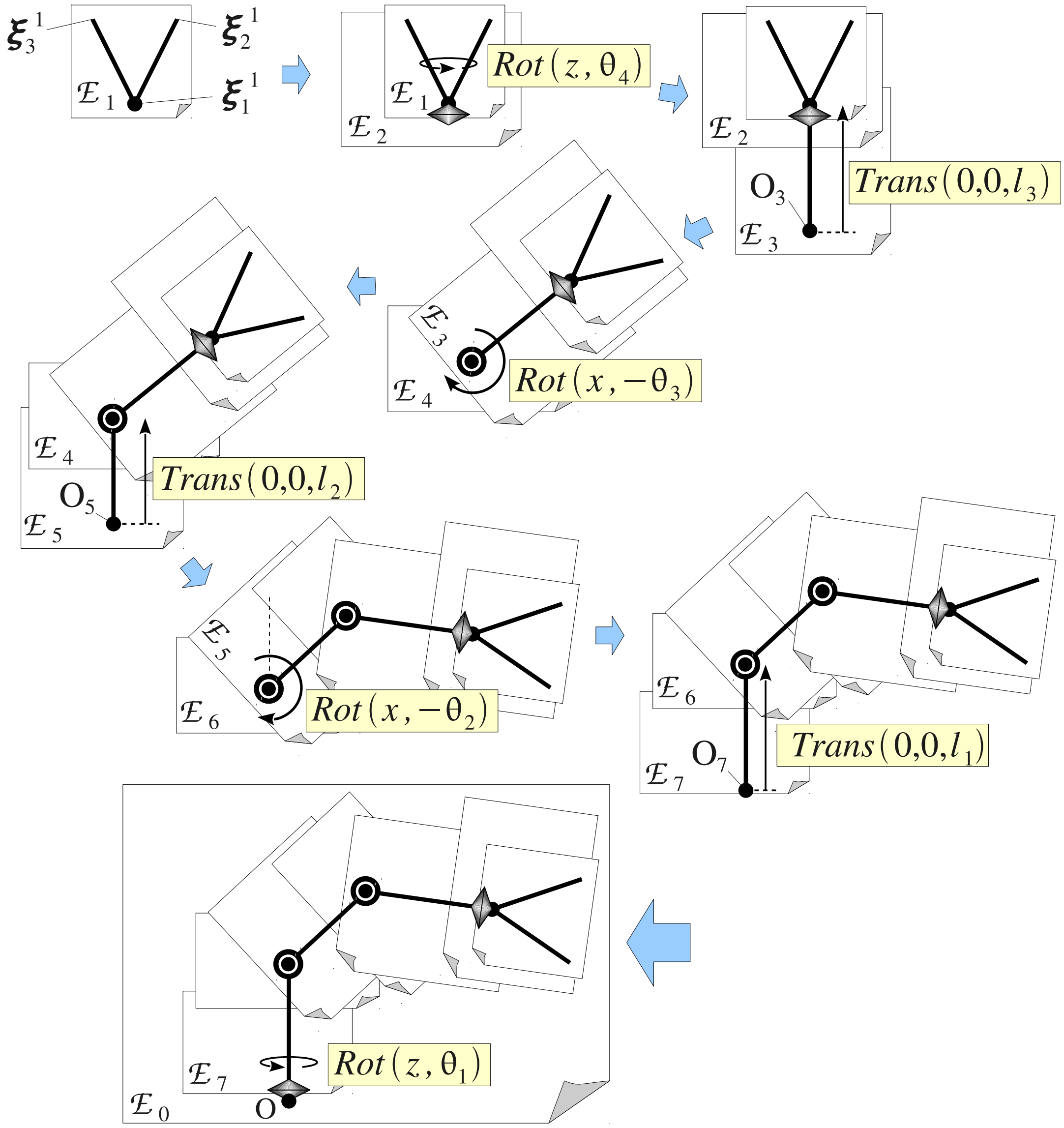


図 5: 4 自由度マニピュレータの運動学

【Step 2】 $\mathcal{E}_1$ の原点 O_1 (手首) を，次の局所座標系 $\mathcal{E}_2$ の原点 O_2 に重ね，$\mathcal{X}_1$ を $\mathcal{E}_2$ に対して θ_4 だけ z 軸回転させる．この変換を,

$$\mathrm{Rot}(z,\theta_4)\mathcal{X}_1 := \{\,\mathrm{Rot}(z,\theta_4)\boldsymbol{\xi}_1^1,\ \mathrm{Rot}(z,\theta_4)\boldsymbol{\xi}_2^1,\ \mathrm{Rot}(z,\theta_4)\boldsymbol{\xi}_3^1\}$$

と表記しよう．このとき，$\mathcal{E}_2$ 上の代表点 $\mathcal{X}_2$ は,

$$\mathcal{X}_2 = R_2\mathcal{X}_1 := \{R_2\boldsymbol{\xi}_1^1,\ R_2\boldsymbol{\xi}_2^1,\ R_2\boldsymbol{\xi}_3^1\},\quad R_2 := \mathrm{Rot}(z,\theta_4) \tag{7.4}$$

となる．

【Step 3】 $\mathcal{E}_2$ の原点 O_2 を，次の局所座標系 $\mathcal{E}_3$ の原点 O_3 に重ね，$\mathcal{X}_2$ を $\mathcal{E}_3$ に対してリンク長 l_3 分だけ z 軸方向に平行移動する．

$$T_3\mathcal{X}_2 = T_3R_2\mathcal{X}_1 = \{T_3R_2\boldsymbol{\xi}_1^1,\ T_3R_2\boldsymbol{\xi}_2^1,\ T_3R_2\boldsymbol{\xi}_3^1\},\quad T_3 := \mathrm{Trans}(0,0,l_3)$$

これに，$\mathcal{E}_3$ の原点 $\mathrm{O}_3 = \mathbb{O}$ を付加した集合,

$$\mathcal{X}_3 = \{T_3R_2\boldsymbol{\xi}_1^1,\ T_3R_2\boldsymbol{\xi}_2^1,\ T_3R_2\boldsymbol{\xi}_3^1,\ \mathbb{O}\} \tag{7.5}$$

が，部品図 $\mathcal{E}_3$ 上の代表点となる．

【Step 4】 $\mathcal{E}_3$ の原点 O_3 を，次の局所座標系 $\mathcal{E}_4$ の原点 O_4 に重ね，$\mathcal{X}_3$ を $\mathcal{E}_4$ に対して肘角度 θ_3 だけ x 軸回転させる．

$$\begin{aligned}\mathcal{X}_4 = R_4\mathcal{X}_3 &= \{R_4T_3R_2\boldsymbol{\xi}_1^1,\ R_4T_3R_2\boldsymbol{\xi}_2^1,\ R_4T_3R_2\boldsymbol{\xi}_3^1,\ R_4\mathbb{O}\}\\ &= \{R_4T_3R_2\boldsymbol{\xi}_1^1,\ R_4T_3R_2\boldsymbol{\xi}_2^1,\ R_4T_3R_2\boldsymbol{\xi}_3^1,\ \mathbb{O}\}, \qquad R_4 := \mathrm{Rot}(x, -\theta_3)\end{aligned} \tag{7.6}$$

角度の正は，紙面手前に進む右ねじの回転方向なので，図 4 のとり方に合せて $-\theta_3$ とした．零ベクトル $\mathbb{O}$ を回転しても $\mathbb{O}$ なので，$R_4\mathbb{O} = \mathbb{O}$ とした．

【Step 5】 $\mathcal{E}_4$ の原点 O_4 を，次の局所座標系 $\mathcal{E}_5$ の原点 O_5 に重ね，$\mathcal{X}_4$ を $\mathcal{E}_5$ に対してリンク長 l_2 分だけ z 軸方向に平行移動する．これに，$\mathcal{E}_5$ の原点 O_5 を付加した集合,

$$\begin{aligned}&\mathcal{X}_5 = T_5\mathcal{X}_4 \cup \{\mathbb{O}\} = \{T_5R_4T_3R_2\boldsymbol{\xi}_1^1,\ T_5R_4T_3R_2\boldsymbol{\xi}_2^1,\ T_5R_4T_3R_2\boldsymbol{\xi}_3^1,\ T_5\mathbb{O},\ \mathbb{O}\},\\ &\qquad\qquad T_5 := \mathrm{Trans}(0, 0, l_2)\end{aligned} \tag{7.7}$$

を $\mathcal{E}_5$ 上の代表点とする．

【Step 6】 $\mathcal{E}_5$ の原点 O_5 を，次の局所座標系 $\mathcal{E}_6$ の原点 O_6 に重ね，$\mathcal{X}_5$ を $\mathcal{E}_6$ に対して肩角度 θ_2 だけ x 軸回転させた集合,

$$\begin{aligned}\mathcal{X}_6 = R_6\mathcal{X}_5 &= \{R_6T_5R_4T_3R_2\boldsymbol{\xi}_1^1,\ R_6T_5R_4T_3R_2\boldsymbol{\xi}_2^1,\ R_6T_5R_4T_3R_2\boldsymbol{\xi}_3^1,\ R_6T_5\mathbb{O},\ R_6\mathbb{O}\}\\ &= \{R_6T_5R_4T_3R_2\boldsymbol{\xi}_1^1,\ R_6T_5R_4T_3R_2\boldsymbol{\xi}_2^1,\ R_6T_5R_4T_3R_2\boldsymbol{\xi}_3^1,\ R_6T_5\mathbb{O},\ \mathbb{O}\},\\ &\qquad R_6 := \mathrm{Rot}(x, \theta_2)\end{aligned} \tag{7.8}$$

を $\mathcal{E}_6$ 上の代表点とする．回転の場合は代表点は追加しない．

【Step 7】 $\mathcal{E}_6$ の原点 O_6 を，次の局所座標系 $\mathcal{E}_7$ の原点 O_7 に重ね，$\mathcal{E}_6$ をリンク長 l_1 分だけ z 軸方向に平行移動し，これに原点 O_7 を付加した集合,

$$\begin{aligned}\mathcal{X}_7 = T_7\mathcal{X}_6 \cup \{\mathbb{O}\} = \{&T_7R_6T_5R_4T_3R_2\boldsymbol{\xi}_1^1,\ T_7R_6T_5R_4T_3R_2\boldsymbol{\xi}_2^1,\ T_7R_6T_5R_4T_3R_2\boldsymbol{\xi}_3^1,\\ &T_7R_6T_5\mathbb{O},\ T_7\mathbb{O}, \mathbb{O}\}, \qquad T_7 := \mathrm{Trans}(0, 0, l_1)\end{aligned} \tag{7.9}$$

を $\mathcal{E}_7$ 上の代表点とする．

【Step 8】 最後に，$\mathcal{E}_7$ の原点 O_7 を，基準座標系 $\mathcal{E}_0 = (\mathrm{O}, \mathcal{E})$ の原点に重ね，θ_1 だけ z 軸回転させ，これに基準座標の原点 $O = \mathbb{O}$ を付加した集合,

$$\begin{aligned}\mathcal{X}_0 = \{&R_8T_7R_6T_5R_4T_3R_2\boldsymbol{\xi}_1^1,\ R_8T_7R_6T_5R_4T_3R_2\boldsymbol{\xi}_2^1,\ R_8T_7R_6T_5R_4T_3R_2\boldsymbol{\xi}_3^1,\\ &R_8T_7R_6T_5\mathbb{O},\ R_8T_7\mathbb{O},\ \mathbb{O}\}, \qquad R_8 := \mathrm{Rot}(z, \theta_1)\end{aligned} \tag{7.10}$$

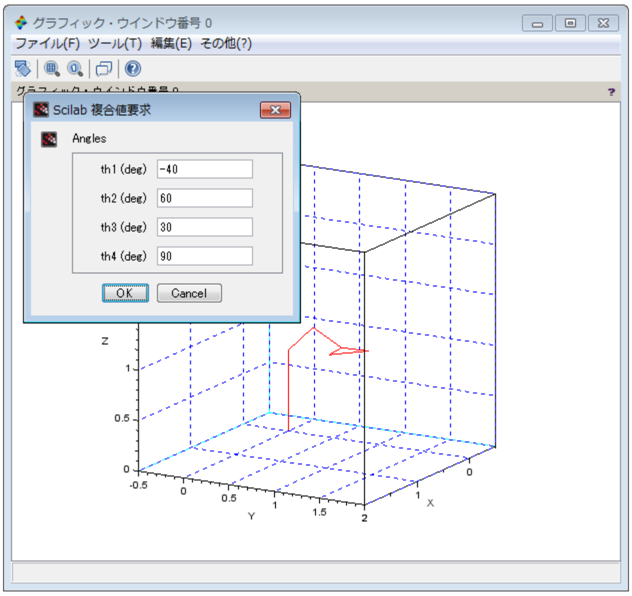
を作る．($R_8\mathbb{O} = \mathbb{O}$)

この $\mathcal{X}_0$ が，基準座標系 $\mathcal{E}_0$ における図 4 のマニピュレータの形状データとなる．改めて列挙すると，次のようになる．

$$
\begin{cases}
\boldsymbol{\xi}_1^0 = R_8(\theta_1)\,T_7\;R_6(\theta_2)\,T_5\;R_4(\theta_3)\,T_3\,R_2(\theta_4)\,\boldsymbol{\xi}_1^1 & \text{※手首} \\
\boldsymbol{\xi}_2^0 = R_8(\theta_1)\,T_7\;R_6(\theta_2)\,T_5\;R_4(\theta_3)\,T_3\,R_2(\theta_4)\,\boldsymbol{\xi}_2^1 & \text{※手先 1} \\
\boldsymbol{\xi}_3^0 = R_8(\theta_1)\,T_7\;R_6(\theta_2)\,T_5\;R_4(\theta_3)\,T_3\,R_2(\theta_4)\,\boldsymbol{\xi}_3^1 & \text{※手先 2} \\
\boldsymbol{\xi}_4^0 = R_8(\theta_1)\,T_7\;R_6(\theta_2)\,T_5\;\mathbb{O} & \text{※肘} \\
\boldsymbol{\xi}_5^0 = R_8(\theta_1)\,T_7\;\mathbb{O} & \text{※肩} \\
\boldsymbol{\xi}_6^0 = \mathbb{O} & \text{※根本}
\end{cases}
\tag{7.11}
$$

各回転変換 R_i は角度に依存するので，R_i(角度) の形式で書いた．他方，リンク長 l_i を定数と仮定すれば平行移動 T_i は定数行列となる．

実習 2.7　Code 3 p15 を実行せよ．実習 2.4 などと同様の操作でリンク機構が動作する．



参考文献

[1] P.R. ポール著・吉川恒夫訳：「ロボット・マニピュレータ」（コロナ社，2002 年）

A　プログラム例

Code 1: “kine-af.sce” (Scilab)

```
clear; clf(); //変数クリア; 描画クリア;
///// アフィン変換 /////
function x = affine_trans(x0, r, angle)
    R = [ cos(angle), -sin(angle); ...
          sin(angle),  cos(angle) ];//回転行列
    x1 = R*x0;  //回転
    x = x1 + r; //平行移動
endfunction
///// データ点列の変換 /////
function points = trans_object(r, angle)
    /// 初期のデータ点列(三角形) ///
    points=[ 2,-4, 2; ...            //x座標
             2,-1,-1 ];              //y座標
    n=size(points,'c');              //点の数
    /// 座標変換 ///
    for i=1:n
        oldp = points(:,i);
        newp = affine_trans(oldp, r, angle); //アフィン変換
        points(:,i) = newp;
    end
endfunction
///// データ点列の描画 /////
function draw_object(points)
    drawlater; clf();                //描画延期; 描画クリア;
    points=[points, points(:,1)];    //描画用に図形を閉じる
    plot(points(1,:),points(2,:),"r-"); //折れ線グラフ
    g=gca();  g.isoview="on";        //座標軸の取得; 縦横比1;
    g.data_bounds=[-10,-10;10,10];   //座標軸の範囲
    p=gce(); p.children.thickness=3;//描画線の太さ
    xlabel("x"); ylabel("y");        //軸ラベル
    xgrid(2); drawnow;               //グリッドon; 描画更新;
endfunction
///// 処理の実行 /////
rr=[0;0]; theta=0;   //初期姿勢
points=trans_object(rr, theta*%pi/180);
draw_object(points);
while(1) //意図的な無限ループ
    disp([rr',theta]); disp(points);    //コンソールへ数値を出力
    txt = ['rx','ry','theta (deg)'];
    sig0 = string([rr;theta])';
    sig = x_mdialog("Position and Angle", txt, sig0);
    if ( size(sig) == 0 )        //もし入力が空なら
        break;            //while脱出
    end
    rr(1) = evstr(sig(1));       //x方向変位
    rr(2) = evstr(sig(2));       //y方向変位
    theta = evstr(sig(3));       //姿勢角
    points=trans_object(rr, theta*%pi/180);
    draw_object(points);         //データ点列の描画
end
```

Code 2: “kine-H.sce” (Scilab)

```
clear; clf(); //変数クリア; 描画クリア;
///// 同次変換 /////
function x = H(x0, r, angle)
    A = [ cos(angle), -sin(angle), r(1); ...
          sin(angle),  cos(angle), r(2); ...
             0,           0,        1 ];//同次変換行列
    x1 = A*[x0;1];  //ダミー成分1を付けて同次変換
    x = x1(1:2);    //ダミー成分の除去
endfunction
///// データ点列の変換 /////
function points = trans_object(r, angle)
    /// 初期のデータ点列(三角形) ///
    points=[ 2,-4, 2; ...            //x座標
             2,-1,-1 ];              //y座標
    n=size(points,'c');              //点の数
    /// 座標変換 ///
    for i=1:n
        oldp = points(:,i);
        newp = H(oldp, r, angle); //同次変換
        points(:,i) = newp;
    end
```

```
endfunction
///// データ点列の描画 /////
function draw_object(points)
    drawlater; clf();                   //描画延期; 描画クリア;
    points=[points, points(:,1)];    //描画用に図形を閉じる
    plot(points(1,:),points(2,:),"r-"); //折れ線グラフ
    g=gca();  g.isoview="on";         //座標軸の取得; 縦横比1;
    g.data_bounds=[-10,-10;10,10];  //座標軸の範囲
    p=gce(); p.children.thickness=3;//直前の描画の線の太さ
    xlabel("x"); ylabel("y");         //軸ラベル
    xgrid(2); drawnow;                //グリッドon; 描画更新;
endfunction
///// 処理の実行 /////
rr=[0;0]; theta=0;   //初期姿勢
points=trans_object(rr, theta*%pi/180);
draw_object(points);
while(1) //意図的な無限ループ
    disp([rr',theta]); disp(points);    //コンソールへ数値を出力
    txt = ['rx','ry','theta (deg)'];
    sig0 = string([rr;theta])';
    sig = x_mdialog("Position and Angle", txt, sig0);
    if ( size(sig) == 0 )       //もし入力が空なら
        break;           //while脱出
    end
    rr(1) = evstr(sig(1));      //x方向変位
    rr(2) = evstr(sig(2));      //y方向変位
    theta = evstr(sig(3));      //姿勢角
    points=trans_object(rr, theta*%pi/180);
    draw_object(points);        //データ点列の描画
end
```

Code 3: “kine-robo.sce” (Scilab)

```
clear; clf(); //変数クリア; 描画クリア;
///// 同次変換(3次元) /////
function A = Rotz(q)
  A = [cos(q), -sin(q), 0, 0; ...
       sin(q),  cos(q), 0, 0; ...
            0,       0, 1, 0; ...
            0,       0, 0, 1];
endfunction
function A = Rotx(q)
  A = [1,      0,       0, 0; ...
       0, cos(q), -sin(q), 0; ...
       0, sin(q),  cos(q), 0; ...
       0,      0,       0, 1];
endfunction
function A = Trans(r1,r2,r3)
  A = [1, 0, 0, r1; ...
       0, 1, 0, r2; ...
       0, 0, 1, r3; ...
       0, 0, 0, 1];
endfunction
///// データ点列 /////
global HandData;
///// マニピュレータ形状データの生成 /////
function points = trans_object( th ) // th=[th1,th2,th3,th4]
    /// 手首〜手先の形状データ列(4点) ///
    l1 = 0.8; l2 = 0.8; l3 = 0.8;
    hand = [ 0,    0,   0, 0; ...   //x成分
             0, -0.2, 0.2, 0; ...   //y成分
             0,  0.2, 0.2, 0; ...   //z成分
             1,    1,   1, 1 ];     //ダミー成分1
    /// 同次変換行列 ///
    R2 = Rotz( th(4) );
    T3 = Trans(0,0,l3);
    R4 = Rotx( -th(3) );
    T5 = Trans(0,0,l2);
    R6 = Rotx( -th(2) );
    T7 = Trans(0,0,l1);
    R8 = Rotz( th(1) );
    /// 形状データの生成 ///
    for i=1:4                         //手首〜手先(4点)
        oldp = hand(:,i);
        newp = R8*T7*R6*T5*R4*T3*R2*oldp;
        xi1s(:,i) = newp;
    end
    xi4 = R8*T7*R6*T5*[0;0;0;1];     //肘
    xi5 = R8*T7*[0;0;0;1];           //肩
    xi6 = [0;0;0;1];                 //根本
    xx = [ xi1s, xi4, xi5, xi6 ];    //一筆書きの点列
```

```
    points = xx(1:3,:);               //ダミー成分(4行目)除去
endfunction
///// データ点列の描画 /////
function draw_object(pts)
    drawlater; clf();                 //描画延期; 描画クリア;
    param3d(pts(1,:),pts(2,:),pts(3,:));   //3次元折れ線グラフ
    g=gca(); g.isoview="on";          //座標軸の取得; 縦横比1;
    g.data_bounds=[-0.5,-0.5,0;2,2,2.5];   //座標軸の範囲
    g.rotation_angles=[75,30];        //3次元透視角度
    p=gce(); p.thickness=3;           //描画線の太さ
    p.foreground=5;                   //描画線の色番号
    xlabel("X"); ylabel("Y"); zlabel("Z"); //軸ラベル
    xgrid(2); drawnow;                //グリッドon; 描画更新;
endfunction
///// 処理の実行 /////
thdeg = [-40, 60, 30, 90];        //初期姿勢 deg
points=trans_object( thdeg*%pi/180 );
draw_object(points);
while(1) //意図的な無限ループ
    disp(thdeg); disp(points);  //コンソールへ数値を出力
    txt = ['th1 (deg)','th2 (deg)','th3 (deg)','th4 (deg)'];
    sig0 = string(thdeg);
    sig = x_mdialog("Angles", txt, sig0);
    if ( size(sig) == 0 )        //もし入力が空なら
        break;           //while脱出
    end
    thdeg = evstr(sig)';         //姿勢角 deg
    points=trans_object( thdeg*%pi/180 );
    draw_object(points);         //データ点列の描画
end
```