第三章 三次元回転

作成：Weiwei Wan
修正：Weiwei Wan, Takuya Kiyokawa

2024 年 春～夏学期

本章では，三次元空間における回転変換について説明します．三次元空間における位置変換は直感的に理解しやすいものです．三次元の位置情報は三つの要素を持つベクトル $\boldsymbol{r}_i$=$(r_{ix}, r_{iy}, r_{iz})^\top$ で表現されます．三次元位置の変換は，実質的にはベクトルの加減算に相当します．本章では位置変換については詳述しません．

三次元位置の計算に対して，三次元空間における回転変換は，ベクトルと行列の乗算などの計算に等価になります．三次元空間における回転は，行列だけでなく，オイラー角や回転ベクトル，四元数などの形式でも表現できます．これらの表現形式は具体的な実装においてしばしば組み合わせて使用されます．本章では，それらの表現形式の意味，計算方法，および WRS の実装について詳しく紹介します．前提として，皆様に理解してもらいたいのは，コンピュータの処理のおいて，行列の方が効率的に計算しやすいため，WRS ではどの表現でも最後は必ず行列の形式に変換されます．また，本章で議論した回転表現やその間の変換などを自力で実装する必要はありません．WRS では，それらの計算を簡単に実行できる関数を用意しています（basis/robot_math.py）．

1　回転行列での表現

三次元の回転は 3×3 の行列式が 1 の正交行列 $\boldsymbol{R}$ で表現されます．つまり，任意の回転行列 $\boldsymbol{R}$ が三次元の回転を表現するためには，次の条件を満たす必要があります：①正交性：$\boldsymbol{R}^\top\boldsymbol{R} = \boldsymbol{I}$ 或いは $\boldsymbol{R}^\top = \boldsymbol{R}^{-1}$；②行列式が 1：$\det(\boldsymbol{R}) = 1$．3×3 の行列式が 1 の正交行列を使って三次元の回転を表す理由は，行列の各列が互いに直交する単位ベクトルであり，これによって三つの軸を表現できるためです．図 1 に示すように，行列式が 1 の正交行列の三つの列ベクトルは互いに直交しており，ちょうど座標系の $\boldsymbol{x}$ 軸, $\boldsymbol{y}$ 軸, $\boldsymbol{z}$ 軸に対応しています．これにより，空間における回転姿勢を表現することができます．なお，この場合，原点は $(0, 0, 0)$ であることに注意してください．正交行列には位置信息が含まれておらず，各列ベクトルは原点 $[0, 0, 0]$ から始まると仮定されます．また，座標系を図示する際には，各軸の色も $\boldsymbol{x}$-赤, $\boldsymbol{y}$-緑, $\boldsymbol{z}$-青（$\boldsymbol{xyz}$-RGB）という暗黙のルールがあることに気を付けてください．

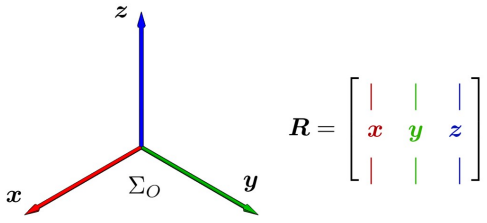
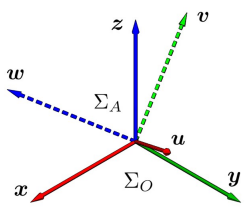


図 1: 3×3 の行列式が 1 の正交行列の各列は座標系の $\boldsymbol{x}$ 軸, $\boldsymbol{y}$ 軸, $\boldsymbol{z}$ 軸として解釈することができます．

空間における回転変換には基準となる座標系があります．通常，この基準座標系を記号 Σ_O で表し，その回転行列を $\boldsymbol{R}_O$ とします．任意の回転行列 $\boldsymbol{R}_A$ が表す座標系 Σ_A は，必ずある基準座標系 $\boldsymbol{R}_O$ の下で記述するものとなります．すなわち，$\boldsymbol{R}_A = \boldsymbol{R}_O{}^O\boldsymbol{R}_A$ のように書くべきです．ここで，${}^O\boldsymbol{R}_A$ は Σ_A の Σ_O に対する回転行列です．${}^O\boldsymbol{R}_A$ のような書き方も慣習的なルールであり，右下の A の座標系から左上の O の座標系までの相対的な値と表します．普段，$\boldsymbol{R}_A$ だけと書かれまして，$\boldsymbol{R}_O$ や ${}^O\boldsymbol{R}_A$ の左上の O が書かれない理由は，基準座標系 $\boldsymbol{R}_O$ を回転しない座標系と想定するためです．その場合，$\boldsymbol{R}_A = \boldsymbol{R}_O{}^O\boldsymbol{R}_A = \boldsymbol{I}{}^O\boldsymbol{R}_A = {}^O\boldsymbol{R}_A$ となります．ここで，$\boldsymbol{I}$ は 3×3 の単位行列です．この行列は，$[1, 0, 0]$，$[0, 1, 0]$，$[0, 0, 1]$ の三つの列ベクトルで構成されています．これらの列ベクトルが対応する $\boldsymbol{x}$，$\boldsymbol{y}$，$\boldsymbol{z}$ 軸は座標系を形成し，この座標系が表す基準座標系は回転していません．

図 2 では単位行列 $\boldsymbol{I}$ が表す基準座標系 Σ_O と原点が Σ_O と重なる別の座標系 Σ_A（$\boldsymbol{R}_A = [\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}]$）を示しています．$\boldsymbol{R}_O = \boldsymbol{I}$ ですので，$\boldsymbol{R}_A$ は ${}^O\boldsymbol{R}_A$ と同じです．$\Sigma_A = {}^O\boldsymbol{R}_A$ は互いに直交する三つの単位ベクトル $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{w}$ に

よって構成され，それぞれの単位ベクトルの要素値は Σ_O の下に記述されています．図 2 の下半分は，$\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{w}$ が Σ_O 座標系における座標を示しています．これらの座標値は，${}^O\boldsymbol{R}_A$ の各列ベクトルの 3 つの要素と同じです．



$$\boldsymbol{R}_O = \begin{bmatrix} | & | & | \\ \boldsymbol{x} & \boldsymbol{y} & \boldsymbol{z} \\ | & | & | \end{bmatrix} = \boldsymbol{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$${}^O\boldsymbol{R}_A = \begin{bmatrix} | & | & | \\ \boldsymbol{u} & \boldsymbol{v} & \boldsymbol{w} \\ | & | & | \end{bmatrix} = \begin{bmatrix} 0.4330127 & -0.64951905 & 0.625 \\ 0.75 & -0.125 & -0.64951905 \\ 0.5 & 0.75 & 0.4330127 \end{bmatrix}$$

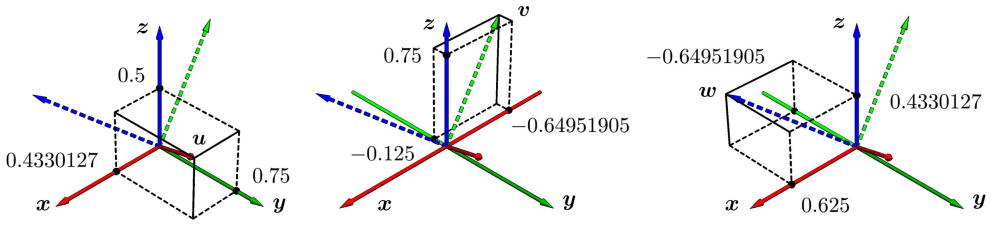
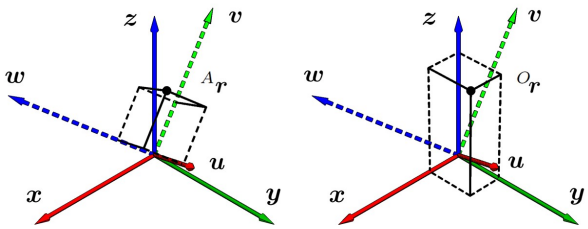


図 2: 破線で示された座標系 Σ_A は三次元の姿勢を表しています．Σ_A に対応する回転行列 ${}^O\boldsymbol{R}_A$ は，三つの軸 $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{w}$ によって構成されます．

図 3 では，座標系 Σ_A におけるベクトル ${}^A\boldsymbol{r}$ を，座標系 Σ_O におけるベクトルに変換した様子を示します．${}^A\boldsymbol{r}$ の左上の添字は $\boldsymbol{r}$ が座標系 Σ_A の下に記述されることを指します．すなわち，Σ_A は ${}^A\boldsymbol{r}$ の基準座標系を意味します．${}^A\boldsymbol{r}$ の基準座標系を座標系 Σ_O に変換する式は ${}^O\boldsymbol{r}={}^O\boldsymbol{R}_A{}^A\boldsymbol{r}$ と表現できます．Σ_A における ${}^A\boldsymbol{r}$ に対して ${}^O\boldsymbol{R}_A$ を掛けると Σ_O における ${}^O\boldsymbol{r}$ となります．${}^O\boldsymbol{r}$ の各要素は ${}^A\boldsymbol{r}$ の各要素が座標系 Σ_A の各軸（$\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{w}$）に対して射影したものの合計を示しています．図 3（右）の式に記載された r_u, r_v, r_w に関する掛け算と足し算はその射影と合計関係を示しています．



$${}^O\boldsymbol{R}_A{}^A\boldsymbol{r} = \begin{bmatrix} | & | & | \\ \boldsymbol{u} & \boldsymbol{v} & \boldsymbol{w} \\ | & | & | \end{bmatrix} {}^A\boldsymbol{r} = r_u \cdot \boldsymbol{u} + r_v \cdot \boldsymbol{v} + r_w \cdot \boldsymbol{w}$$

$$= \begin{bmatrix} r_u u_x + r_v v_x + r_w w_x \\ r_u u_y + r_v v_y + r_w w_y \\ r_u u_z + r_v v_z + r_w w_z \end{bmatrix} = {}^O\boldsymbol{r}$$

図 3: 座標系 Σ_A における点 ${}^A\boldsymbol{r} = [0.15, 0.07, 0.05]$ をワールド座標系 Σ_O における座標点 ${}^O\boldsymbol{r} = [0.05, 0.07, 0.15]$ に変換します．

2　オイラー角での表現

回転行列を使用することで，回転変換の計算をコンピュータに実装しやすい行列計算に変換できます．コンピュータを使った行列計算のアルゴリズムは長年に渡って改善されてきており，，効率的な行列計算を可能にするライブラリが数多く存在しています．そのため，回転行列は各種ソフトウェアに幅広く利用されます．しかし，回転行列の計算がコンピュータに実装しやすい一方で，人間にとっては直感的に理解しにくいものです．回転行列以外にも，三次元回転を直感的に理解しやすい表現方法がいくつかあります．代表的な方法としては，オイラー角や軸－角度形式での回転表現などが挙げられます．本節では，オイラー角による表現について述べます．

三次元空間での回転は三つの自由度を持っています．オイラー角の表現では，直感的にその三つの自由度の回転を順次実行し，三つの回転角度を使って最終的な回転変換を表します．一般的に使用される，回転させる軸の順番は，基準座標系の $\boldsymbol{x}$ 軸周り，$\boldsymbol{y}$ 軸周り，そして $\boldsymbol{z}$ 軸周りのものです．この順番は，ロール・ピッチ・ヨーと呼ばれ

ます．ロール，ピッチ，ヨーという呼び方は,，飛行機の動きに基づいています．飛行機が $\boldsymbol{x}$ 軸（前後方向の軸）周りに回転すると，機体が左右に傾ける動きが出ます．この動きは，外部から見ると，左右ローリングする様子と似ていますから，ロールと呼ばれます．飛行機が $\boldsymbol{y}$ 軸（左右方向の軸）周りに回転すると，機首を上下に動かします．この上下動きがピッチと呼ばれます．残りのヨーは，飛行機が $\boldsymbol{z}$ 軸（上下方向の軸）周りに回転して水平面上で左右に旋回する動きを表します．ロール，ピッチ，ヨーの表現方法を使うと，それぞれの軸周りの回転を直感的に理解しやすくなります．

前節で説明した Σ_O から Σ_A までの回転行列 ${}^O\boldsymbol{R}_A$ を例として，ロール・ピッチ・ヨーの表現を詳しく見てみます．この回転変換行列は，基準座標系 Σ_O の $\boldsymbol{x}$ 軸を 60 度周り，$\boldsymbol{y}$ 軸を -30 度周り，$\boldsymbol{z}$ 軸を 60 度周りの順番で回転させるロール，ピッチ，ヨー形式のオイラー角で表すことができます．図 4 は，Σ_O がこのロール，ピッチ，ヨーの角度順序で回転する過程を示しています．図 4(a) は $\boldsymbol{x}$ 軸を 60 度回す様子を示します．回した後の座標系は $[\boldsymbol{x}', \boldsymbol{y}', \boldsymbol{z}']$ と表します．図 4(b) は $\boldsymbol{y}$ 軸を -30 度回す様子を示します．回した後の座標系は $[\boldsymbol{x}'', \boldsymbol{y}'', \boldsymbol{z}'']$ と表します．図 4(c) は $\boldsymbol{z}$ 軸を 60 度回す様子を示します．回転後の座標系は $[\boldsymbol{x}''', \boldsymbol{y}''', \boldsymbol{z}''']$ で表します．$[\boldsymbol{x}''', \boldsymbol{y}''', \boldsymbol{z}''']$ は $[\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}]$ と同じです．図 4(d) は最終結果です．ここで注意してもらいたいのは，$\boldsymbol{x}$，$\boldsymbol{y}$，$\boldsymbol{z}$ 軸は Σ_O の初期状態における三つの軸を指しており，ロール，ピッチ，ヨーの回転と共に変化しないということです．つまり，ここでの $\boldsymbol{x}$，$\boldsymbol{y}$，$\boldsymbol{z}$ 軸は固定的なものであり，回転過程における軸の変化とは無関係です．したがって，ロール，ピッチ，ヨーのようなオイラー角による表現は，固定軸オイラー角と呼ばれます．

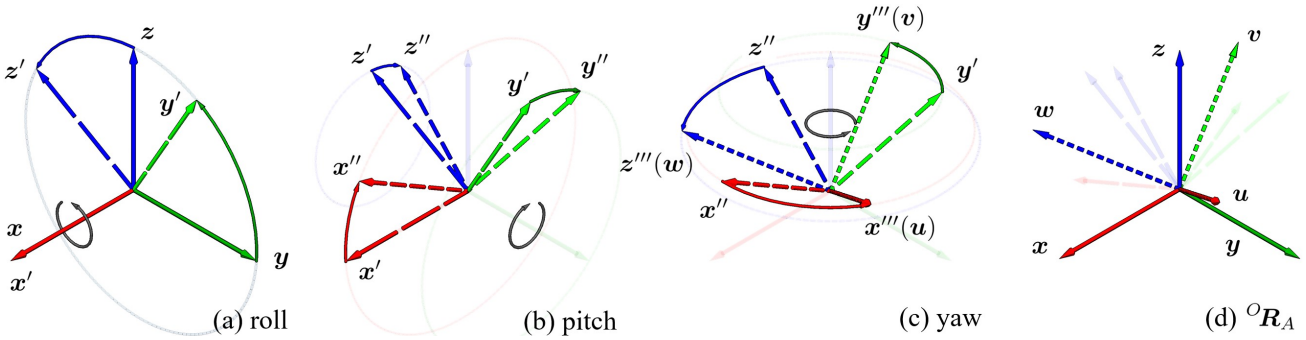


図 4: ロール・ピッチ・ヨーの回転によって座標系 Σ_O を Σ_A に変換する様子．(a) $\boldsymbol{x}$ 軸を 60 度回した結果．(b) 続いて，$\boldsymbol{y}$ 軸を -30 度回した結果．(c) さらに，$\boldsymbol{z}$ 軸を 60 度回した結果．(d) 最終的に行列 ${}^O\boldsymbol{R}_A$ が表す回転と同じです

以上の説明から，ロール・ピッチ・ヨーは $\boldsymbol{x}$-$\boldsymbol{y}$-$\boldsymbol{z}$ の順番で記述した固定軸オイラー角の表現と分かります．ロール・ピッチ・ヨーの他にも，座標系 Σ_O を座標系 Σ_A へ回転するための回転軸の順番の選び方がいくつかあります．例えば，図 5 には $\boldsymbol{x}$-$\boldsymbol{y}$-$\boldsymbol{x}$ 順の回転を示します．Σ_O は，この順番にしたがっても，Σ_A まで回転できます．$\boldsymbol{x}$-$\boldsymbol{y}$-$\boldsymbol{x}$ 順はロール・ピッチ・ヨーと同様に，すべての回転軸が固定であり，固定軸オイラー角の表現となります．

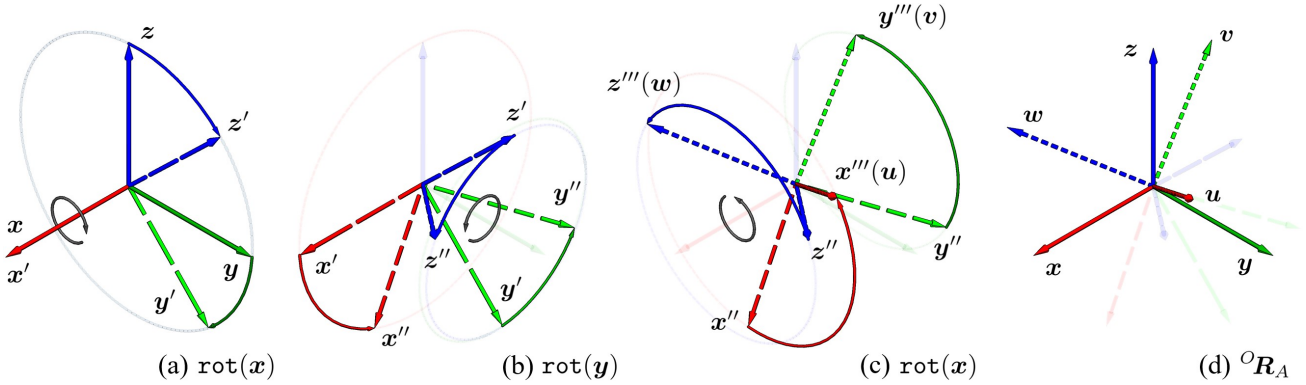
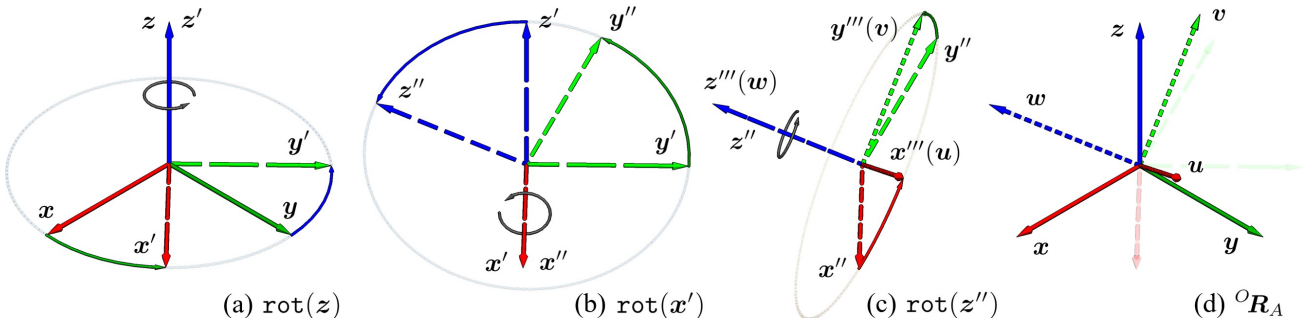


図 5: もう一つの外挿オイラー角の表現．$\boldsymbol{x}$-$\boldsymbol{y}$-$\boldsymbol{x}$ 順の回転で座標系 Σ_O を Σ_A に変換します．(a) $\boldsymbol{x}$ 軸を -46.10211375 度回した結果．(b) 続いて，$\boldsymbol{y}$ 軸を 64.34109373 度回した結果．(c) さらに，$\boldsymbol{x}$ 軸を 123.69006753 度回した結果．(d) 最終的に行列 ${}^O\boldsymbol{R}_A$ が表す回転と同じです．

固定軸オイラー角の表現を整理すると，ロール・ピッチ・ヨー（$\boldsymbol{x}$-$\boldsymbol{y}$-$\boldsymbol{z}$）と $\boldsymbol{x}$-$\boldsymbol{y}$-$\boldsymbol{x}$ の他にも，$\boldsymbol{x}$-$\boldsymbol{z}$-$\boldsymbol{y}$，$\boldsymbol{x}$-$\boldsymbol{z}$-$\boldsymbol{x}$，$\boldsymbol{y}$-$\boldsymbol{z}$-$\boldsymbol{x}$，$\boldsymbol{y}$-$\boldsymbol{z}$-$\boldsymbol{y}$，$\boldsymbol{y}$-$\boldsymbol{x}$-$\boldsymbol{z}$，$\boldsymbol{y}$-$\boldsymbol{x}$-$\boldsymbol{y}$，$\boldsymbol{z}$-$\boldsymbol{x}$-$\boldsymbol{y}$，$\boldsymbol{z}$-$\boldsymbol{x}$-$\boldsymbol{z}$，$\boldsymbol{z}$-$\boldsymbol{y}$-$\boldsymbol{x}$，$\boldsymbol{z}$-$\boldsymbol{y}$-$\boldsymbol{z}$ など計 12 種類が存在しています．よく使用されるのは最初に説明したロール・ピッチ・ヨーです．ロール，ピッチ，ヨーの角度をそれぞれ θ, ϕ, ψ にすると，対応する回転行列は式 (1) のようになります．

$$
\begin{aligned}
{}^{O}\boldsymbol{R}_{A} = \boldsymbol{R}_{r}(\theta)\boldsymbol{R}_{p}(\phi)\boldsymbol{R}_{y}(\psi) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \cos\phi\cos\psi & -\cos\phi\sin\psi & \sin\phi \\ \cos(\theta)\sin\psi + \sin(\theta)\sin\phi\cos\psi & \cos(\theta)\cos\psi - \sin(\theta)\sin\phi\sin\psi & -\sin(\theta)\cos\phi \\ \sin(\theta)\sin\psi - \cos(\theta)\sin\phi\cos\psi & \sin(\theta)\cos\psi + \cos(\theta)\sin\phi\sin\psi & \cos(\theta)\cos\phi \end{bmatrix} \quad (1)
\end{aligned}
$$

固定軸オイラー角に対して，第 2 および第 3 の回転は，回転後のローカルな各軸周りで行うモバイル軸オイラー角の表現もあります．図 6 は，グローバル $\boldsymbol{z}$，ローカル $\boldsymbol{x}'$，ローカル $\boldsymbol{z}''$ の軸周りの順序での回転を示しています．固定軸オイラー角の表現と同様に，計 12 個のモバイル軸オイラー角の表現が存在しています：$\boldsymbol{z}$-$\boldsymbol{y}'$-$\boldsymbol{x}''$，$\boldsymbol{x}$-$\boldsymbol{y}'$-$\boldsymbol{x}''$，$\boldsymbol{y}$-$\boldsymbol{z}'$-$\boldsymbol{x}''$，$\boldsymbol{x}$-$\boldsymbol{z}'$-$\boldsymbol{x}''$，$\boldsymbol{x}$-$\boldsymbol{z}'$-$\boldsymbol{y}''$，$\boldsymbol{y}$-$\boldsymbol{z}'$-$\boldsymbol{y}''$，$\boldsymbol{z}$-$\boldsymbol{x}'$-$\boldsymbol{y}''$，$\boldsymbol{y}$-$\boldsymbol{x}'$-$\boldsymbol{y}''$，$\boldsymbol{y}$-$\boldsymbol{x}'$-$\boldsymbol{z}''$，$\boldsymbol{z}$-$\boldsymbol{x}'$-$\boldsymbol{z}''$，$\boldsymbol{x}$-$\boldsymbol{y}'$-$\boldsymbol{z}''$ と $\boldsymbol{z}$-$\boldsymbol{y}'$-$\boldsymbol{z}''$．特に，$\boldsymbol{z}$-$\boldsymbol{x}'$-$\boldsymbol{z}''$ 順は，機械システムの回転，例えばユニバーサルジョイントやジンバルなどを表現するために広く使用されます．$\boldsymbol{z}$-$\boldsymbol{y}'$-$\boldsymbol{x}''$ 順は，人の頭の方向変化に対する直感に一致するため[1]，実際のアプリケーションで広く使用されています．



(a) rot($\boldsymbol{z}$)　(b) rot($\boldsymbol{x}'$)　(c) rot($\boldsymbol{z}''$)　(d) ${}^{O}\boldsymbol{R}_{A}$

図 6: 内挿オイラー角の表現の例．$\boldsymbol{z}$-$\boldsymbol{x}'$-$\boldsymbol{z}''$ の回転順序で座標系 Σ_O を Σ_A に変換します．(a) $\boldsymbol{z}$ 軸を 43.89788625 度回した結果．(b) $\boldsymbol{x}'$ 軸を 64.34109373 度回した結果．(c) $\boldsymbol{z}''$ 軸を 33.69006753 度回した結果．(d) 最終的に行列 ${}^{O}\boldsymbol{R}_{A}$ が表す回転と同じです．

以上は回転のオイラー角の表現の説明です．上記の例で分かるのは，一つの回転行列が複数のオイラー角による回転順序に対応しているということです．オイラー角形式の最終的な回転結果は回転行列と一致しますが，1 つの回転行列には複数のオイラー角の回転方法が対応しており，これが多くの不便を引き起こします．大多数のシステムは便利なように，$\boldsymbol{r}$-$\boldsymbol{p}$-$\boldsymbol{y}$ 順をデフォルトで使用して回転行列とオイラー角の変換を行います．WRS システムの basis.robot_math 内の関数 rotmat_from_euler と rotmat_to_euler は，回転行列とオイラー角の変換機能を提供しています．これらの関数の第二引数 axes は，回転方式を指定するために使用されます．デフォルトではこの値は'sxyz' であり，固定軸オイラー角の $\boldsymbol{r}$-$\boldsymbol{p}$-$\boldsymbol{y}$ 順の解を求めることに指します．この値を'rzxz' に設定すると，モバイル軸オイラー角の $\boldsymbol{z}$-$\boldsymbol{x}'$-$\boldsymbol{z}''$ 順の解を求めることができます．以下のコードでは，回転行列 ${}^{O}\boldsymbol{R}_{A}$ の $\boldsymbol{r}$-$\boldsymbol{p}$-$\boldsymbol{y}$ 順と $\boldsymbol{z}$-$\boldsymbol{x}'$-$\boldsymbol{z}''$ 順のオイラー角の表現を求めています．rotmat_to_euler 関数の返り値はラジアンですので，numpy の degrees 関数を度数値に変更しています．このコードの出力結果は [60.00000011 -30.00000003 60.00000011] と [43.89788637 64.34109382 33.69006753] です．この結果は図 4 と図 6 に示されているものと一致します．

```
1    import numpy as np
2    import basis.robot_math as rm
3
4    o_r_a = np.array([[0.4330127, -0.64951905, 0.625],
```

[1]人の頭部の回転の難易度は，左右，上下，首の傾きの順です．これはちょうど $\boldsymbol{z}$ 軸，$\boldsymbol{y}'$，$\boldsymbol{x}''$ 軸周りの回転順序に対応します．

```
                        [0.75, -0.125, -0.64951905],
                        [0.5, 0.75, 0.4330127]])
    print(np.degrees(rm.rotmat_to_euler(o_r_a, order='sxyz')))
    print(np.degrees(rm.rotmat_to_euler(o_r_a, order='rzxz')))
```

オイラー角の表現は直感的で，人間にとって最も理解しやすい表現形式のため，実際の実装に幅広く使用されています．一方，オイラー角の表現は回転順序と回転角度に縛られます．同じ回転目標に到達するためでも，オイラー角で表現された回転は普通に冗長です．したがって，オイラー角は最適化の計算には適していません．最適化や補完などの計算では，最小回転が必要ですので，軸-角度の表現（回転ベクトルや四元数）は一般的に使用されます．これらの形式については次の節で詳しく説明します．

3　軸－角度での表現

3.1　オイラーの回転定理

人間が直感的に理解できるもう一つの回転の表現方法は軸－角度法です．この表現方法は，どのような回転でも，一つの軸とその軸周りの回転角度だけで表すことができるという考え方に基づいています．これは，以下のように記されているオイラーの回転定理によって明らかにされています．

「球がその中心の周りを移動するとき，移動しない直径がある．」

球の中心周りの移動は回転であるため，この定理の本質的な意味は，回転中に方向の変わらない軸が存在することを示しています．要するに，回転は方向の変わらない軸とその軸周りの回転角度で表せるという前述の考え方に繋がります．本定理の簡単な証明は図 7 で説明しています．

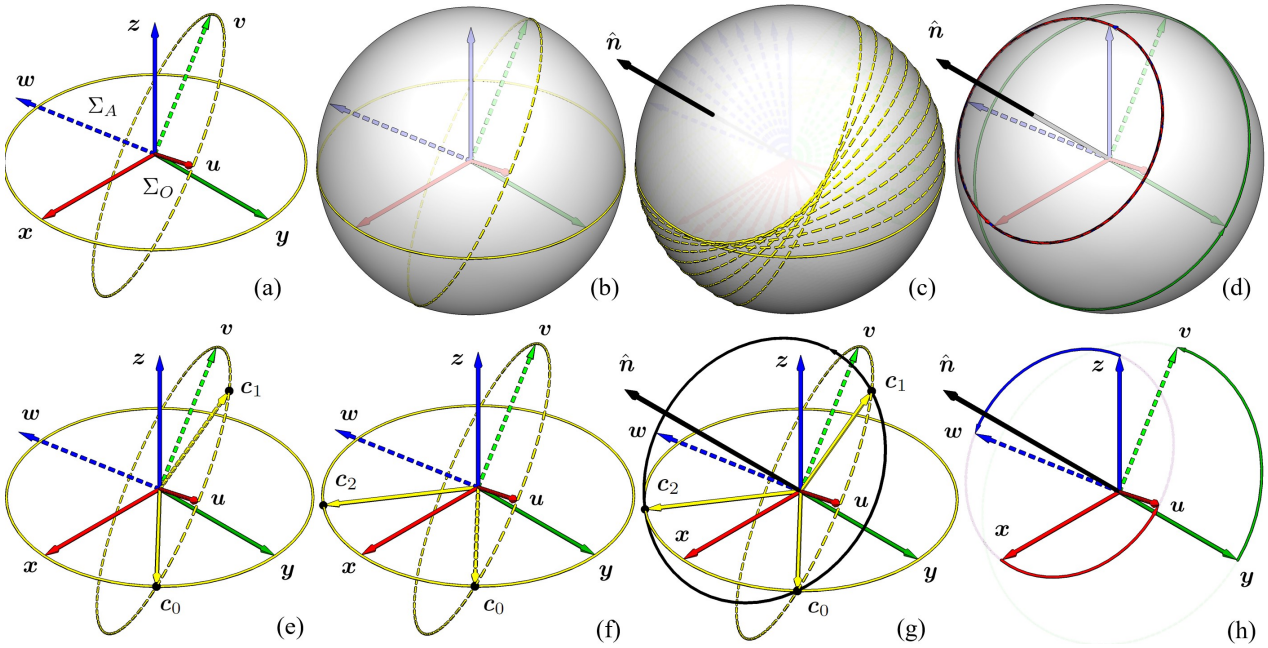


図 7: (a,b) 座標系 Σ_O から座標系 Σ_A までの回転は二つの座標系に示された二つの単位球で表現できます．(c,d) この回転は本質的に一つの軸周りを回転しています．(c) 各座標系の $\boldsymbol{x}$ と $\boldsymbol{y}$ との交点を持つ単位円の移動軌跡．(d) 各座標系の座標軸と単位球表面の交点の移動軌跡が単位球表面上に表示されています．(e,f) この回転が軸 $\hat{\boldsymbol{n}}$ 周りの回転と等価であることを証明しています．

まず，図 7(a) と図 7(b) に示す内容を説明します．図 7(a) では，実線の矢印に示された座標系 Σ_O から点線の矢印に示された座標系 Σ_A までの回転を示しています．この二つの座標系を二つの単位球で表現してみましょう．それぞれの単位球は原点を中心にもち，$\boldsymbol{x}$ と $\boldsymbol{y}$ の軸と $\boldsymbol{u}$ と $\boldsymbol{v}$ の軸上の単位ベクトルの先端を通ります．二つの単位球

は重なっていて，描画しにくいです．図 7(a) では皆さんが識別しやすいように，二つの円で単位球の枠円を示しています．実線の黄色の枠円は座標系 Σ_O の単位球に対応し，破線の黄色の枠円は座標系 Σ_A の単位球に対応します．図 7(b) では透過した単位球を可視化しています．二つの球が重なっているため，一つしかないように見えていることに注意してください．図 7(c) は座標系 Σ_O から座標系 Σ_A までの回転において，実線の円から破線の円まで移動していく様子を示しています．図 7(d) は座標系 Σ_O から座標系 Σ_A までの回転における，座標軸と単位球表面の交点の移動軌跡を示しています．図 7(c) と図 7(d) の結果を見ると，座標系 Σ_O から座標系 Σ_A までの回転は，回転中に位置が不変の固定軸 $\hat{\boldsymbol{n}}$ 周りの回転と等価となっていることが分かります．

次に，図的に説明してきたオイラーの回転定理について，ここでは簡単に証明します．実線の枠円と破線の枠円の交点は同時に両方の円に属しています．回転中にこの交点の移動の様子を観察してみましょう．まず，図 7(e) に示すように，$\boldsymbol{c}_0$ を実線の枠円上の点，すなわち，座標系 Σ_O に固定された一点として考えてみます．座標系 Σ_O を座標系 Σ_A まで回転しても，$\boldsymbol{c}_0$ のローカル座標系での位置は変わってはならないので，$\boldsymbol{c}_0$ は座標系 Σ_A における対応する位置，つまり破線の枠円上の点 $\boldsymbol{c}_1$ に移動すべきです．一方，図 7(f) に示すように，$\boldsymbol{c}_0$ を破線の枠円上の点，すなわち，座標系 Σ_A に固定された一点として考えることもできます．その場合，$\boldsymbol{c}_0$ が回転前後のローカル座標系においた位置として変わってはならないので，$\boldsymbol{c}_0$ は座標系 Σ_O における対応する位置，つまり実線の円上の点 $\boldsymbol{c}_2$ から移動してきたことにしなければ整合性をとれません．図 7(g) に示すように，$\boldsymbol{c}_0$，$\boldsymbol{c}_1$ と $\boldsymbol{c}_2$ の 3 点は円周を決まります．$\boldsymbol{c}_0$ は座標系の原点とその円周の中心を結ぶ中心軸 $\hat{\boldsymbol{n}}$ 周りを回転することで $\boldsymbol{c}_1$ に移動します．同様に，$\boldsymbol{c}_2$ も $\hat{\boldsymbol{n}}$ 周りを回転することで $\boldsymbol{c}_0$ に移動します．ここから広げると，Σ_O の他のすべての軸も Σ_O が示す空間内のすべての点も，$\hat{\boldsymbol{n}}$ 周りに回転して，Σ_A の対応する軸と対応する点まで移動します．詳しい様子は図 7(d) および (h) に示されます．最終的に，図 7(c) に示された円周の移動軌跡のようなイメージで，座標系が回転しています．$\hat{\boldsymbol{n}}$ は回転時に変わらない固定軸です．回転はその軸と軸周りの角度，つまり軸-角度の形で表現できることになります．

3.2 回転行列と軸 − 角度形式の変換

任意の回転行列 ${}^O\boldsymbol{R}_A$ に対応する軸 − 角度表現の値は，以下の式で計算します．

$$\boldsymbol{n} = \begin{pmatrix} {}^O\boldsymbol{R}_A(2,3) - {}^O\boldsymbol{R}_A(3,2) \\ {}^O\boldsymbol{R}_A(3,1) - {}^O\boldsymbol{R}_A(1,3) \\ {}^O\boldsymbol{R}_A(1,2) - {}^O\boldsymbol{R}_A(2,1) \end{pmatrix}, \ \cos(\theta) = \frac{\mathtt{tr}({}^O\boldsymbol{R}_A) - 1}{2}. \tag{2}$$

ここで，${}^O\boldsymbol{R}_A(i,j)$ は，${}^O\boldsymbol{R}_A$ の i 行 j 列目の要素を示し，$\mathtt{tr}({}^O\boldsymbol{R}_A)$ は行列 ${}^O\boldsymbol{R}_A$ の対角要素（対角線上の要素）の総和を指します．トレースと呼ばれます．

以下では上式の導出について述べます．まず，回転軸 $\boldsymbol{n}$ に注目しましょう．オイラーの回転定理によって回転軸 $\boldsymbol{n}$ は式 (3) を満たすことが分ります．回転軸に回転軸周りの回転行列を掛けると回転軸は変らないと意味します．

$${}^O\boldsymbol{R}_A\boldsymbol{n} = \boldsymbol{n}. \tag{3}$$

同時に，$\boldsymbol{n}$ は逆回転を示す回転行列 ${}^O\boldsymbol{R}_A^\top$ で変換しても変わらないので，式 (4) も成立することが分かります．

$${}^O\boldsymbol{R}_A^\top\boldsymbol{n} = \boldsymbol{n}. \tag{4}$$

式 (3) から式 (4) を引き

$$({}^O\boldsymbol{R}_A - {}^O\boldsymbol{R}_A^\top)\boldsymbol{n} = 0, \tag{5}$$

前の項 ${}^O\boldsymbol{R}_A - {}^O\boldsymbol{R}_A^\top$ を展開して

$${}^O\boldsymbol{R}_A - {}^O\boldsymbol{R}_A^\top = \begin{pmatrix} 0 & {}^O\boldsymbol{R}_A(1,2) - {}^O\boldsymbol{R}_A(2,1) & {}^O\boldsymbol{R}_A(1,3) - {}^O\boldsymbol{R}_A(3,1) \\ {}^O\boldsymbol{R}_A(2,1) - {}^O\boldsymbol{R}_A(1,2) & 0 & {}^O\boldsymbol{R}_A(2,3) - {}^O\boldsymbol{R}_A(3,2) \\ {}^O\boldsymbol{R}_A(3,1) - {}^O\boldsymbol{R}_A(1,3) & {}^O\boldsymbol{R}_A(3,2) - {}^O\boldsymbol{R}_A(2,3) & 0 \end{pmatrix}, \tag{6}$$

式 (6) を式 (5) に代入することで，式 (7) が得られます.

$$\boldsymbol{n} = \begin{pmatrix} {}^{O}\boldsymbol{R}_A(2,3) - {}^{O}\boldsymbol{R}_A(3,2) \\ {}^{O}\boldsymbol{R}_A(3,1) - {}^{O}\boldsymbol{R}_A(1,3) \\ {}^{O}\boldsymbol{R}_A(1,2) - {}^{O}\boldsymbol{R}_A(2,1) \end{pmatrix}. \tag{7}$$

必要に応じて，式 (7) の $\boldsymbol{n}$ を下式により正規化することで，回転軸の単位ベクトル $\hat{\boldsymbol{n}}$ も得ることができます.

$$\hat{\boldsymbol{n}} = \begin{pmatrix} \dfrac{{}^{O}\boldsymbol{R}_A(2,3) - {}^{O}\boldsymbol{R}_A(3,2)}{\sqrt{({}^{O}\boldsymbol{R}_A(2,3) - {}^{O}\boldsymbol{R}_A(3,2))^2 + ({}^{O}\boldsymbol{R}_A(3,1) - {}^{O}\boldsymbol{R}_A(1,3))^2 + ({}^{O}\boldsymbol{R}_A(1,2) - {}^{O}\boldsymbol{R}_A(2,1))^2}} \\ \dfrac{{}^{O}\boldsymbol{R}_A(3,1) - {}^{O}\boldsymbol{R}_A(1,3)}{\sqrt{({}^{O}\boldsymbol{R}_A(2,3) - {}^{O}\boldsymbol{R}_A(3,2))^2 + ({}^{O}\boldsymbol{R}_A(3,1) - {}^{O}\boldsymbol{R}_A(1,3))^2 + ({}^{O}\boldsymbol{R}_A(1,2) - {}^{O}\boldsymbol{R}_{A(2,1)})^2}} \\ \dfrac{{}^{O}\boldsymbol{R}_A(1,2) - {}^{O}\boldsymbol{R}_A(2,1)}{\sqrt{({}^{O}\boldsymbol{R}_A(2,3) - {}^{O}\boldsymbol{R}_A(3,2))^2 + ({}^{O}\boldsymbol{R}_A(3,1) - {}^{O}\boldsymbol{R}_A(1,3))^2 + ({}^{O}\boldsymbol{R}_A(1,2) - {}^{O}\boldsymbol{R}_A(2,1))^2}} \end{pmatrix}. \tag{8}$$

次に，回転軸 $\boldsymbol{n}$ 周りの回転角 θ の導出方法について述べます．式 (2) の右側の式を変形すれば，回転角 θ は式 (**??**) で計算できることがわかります.

$$\theta = \cos^{-1}\frac{\mathtt{tr}({}^{O}\boldsymbol{R}_A) - 1}{2}. \tag{9}$$

式 (**??**) を導出するためには，ロドリゲスの回転公式を用いることができます．ここで，まだロドリゲスの回転公式を紹介していませんので，別法で導きます．回転軸 $\boldsymbol{n}$ を Σ_O の $\boldsymbol{z}$ 軸まで回転する回転行列 $\boldsymbol{Q}$ について考察してみましょう．$\boldsymbol{z}$ 軸を中間軸として使用する場合，$\boldsymbol{n}$ 軸周りの回転は，$\boldsymbol{n}$ 軸を $\boldsymbol{z}$ 軸に回転させ，次に $\boldsymbol{z}$ 軸を回して回転させ，最後に回転した姿勢を $\boldsymbol{n}$ 軸に戻すという 3 つの回転の組み合わせに等しい．従って，式 (10) が成立することが分かります.

$${}^{O}\boldsymbol{R}_A = \boldsymbol{Q}^{\top}\mathtt{rot}(\boldsymbol{z},\theta)\boldsymbol{Q}. \tag{10}$$

左側の $\boldsymbol{Q}^{\top}\mathtt{rot}(\boldsymbol{z},\theta)\boldsymbol{Q}$ は上記の三つのステップに対応していて，左側の ${}^{O}\boldsymbol{R}_A$ と等価です.

続いて，式 (10) の両側の行列のトレースが同値であるため

$$\mathtt{tr}({}^{O}\boldsymbol{R}_A) = \mathtt{tr}(\boldsymbol{Q}^{\top}\mathtt{rot}(\boldsymbol{z},\theta)\boldsymbol{Q}), \tag{11}$$

行列のトレースの循環性より，等式 $\mathtt{tr}(\boldsymbol{R}_i\boldsymbol{R}_j) = \mathtt{tr}(\boldsymbol{R}_j\boldsymbol{R}_i)$ を用いることで

$$\begin{aligned} \mathtt{tr}({}^{O}\boldsymbol{R}_A) &= \mathtt{tr}((\boldsymbol{Q}^{\top}\mathtt{rot}(\boldsymbol{z},\theta))\boldsymbol{Q}) = \mathtt{tr}(\boldsymbol{Q}(\boldsymbol{Q}^{\top}\mathtt{rot}(\boldsymbol{z},\theta))) = \mathtt{tr}(\boldsymbol{Q}\boldsymbol{Q}^{\top}\mathtt{rot}(\boldsymbol{z},\theta)) \\ &= \mathtt{tr}(\boldsymbol{I}\mathtt{rot}(\boldsymbol{z},\theta)) = \mathtt{tr}(\mathtt{rot}(\boldsymbol{z},\theta)), \end{aligned} \tag{12}$$

と変形できます．ここで $\mathtt{rot}(\boldsymbol{z},\theta)$ は式 (13) を用いてさらに展開できるため，式 (14) のように表現できます．ここまでで，式 (2) の右半分の証明ができました.

$$\mathtt{rot}(\boldsymbol{z},\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{13}$$

$$\mathtt{tr}({}^{O}\boldsymbol{R}_A) = \mathtt{tr}(\mathtt{rot}(\boldsymbol{z},\theta)) = 2\cos(\theta) + 1 \longrightarrow \cos(\theta) = \frac{\mathtt{tr}({}^{O}\boldsymbol{R}_A) - 1}{2}. \tag{14}$$

3.3 ロドリゲスの回転公式

前の小節では回転行列から対応する軸-角度の表現を求める数式を紹介しました．逆に，軸-角度形式の表現の回転角 θ と単位回転軸 $\hat{\boldsymbol{n}}$ が分かれば，式 (15) を用いて，その軸-角度形式の表現に対応する回転行列を求めることができます．式 (15) はロドリゲスの回転公式と呼ばれます.

$$\boldsymbol{R} = \boldsymbol{I} + \sin(\theta)[\hat{\boldsymbol{n}}\times] + (1 - \cos(\theta))[\hat{\boldsymbol{n}}\times]^2. \tag{15}$$

式中の $[\hat{\boldsymbol{n}}\times]$ は歪対称行列とよばれ，式 (16) によって定義されます．歪対称行列を使うと，外積 $\boldsymbol{n}_i \times \boldsymbol{n}_j$ の計算を歪対象行列 $[\boldsymbol{n}_i\times]$ と $\boldsymbol{n}_j$ の内積の計算に変換計算に変換できます．外積をなくすことで，計算を行列計算に集約できるため，よく利用されます．

$$[\hat{\boldsymbol{n}}\times] = \begin{pmatrix} 0 & -\hat{n_3} & \hat{n_2} \\ \hat{n_3} & 0 & -\hat{n_1} \\ -\hat{n_2} & \hat{n_1} & 0 \end{pmatrix}. \tag{16}$$

図 8 は式 (15) に示されたロドリゲスの回転公式の導出方法を説明しています．ここで，座標系 Σ_O における黄色のベクトル $\boldsymbol{r}$ が軸 $\hat{\boldsymbol{n}}$ 周りを角度 θ で回転するとします．回転後，$\boldsymbol{r}$ は図中の黄色の破線のベクトル $\boldsymbol{r}'$ に移動されます．ロドリゲスの回転公式の導出方法を導出するために，$\boldsymbol{r}$ を回転軸 $\hat{\boldsymbol{n}}$ に平行な部分 $\boldsymbol{r}_\parallel$ と垂直な部分 $\boldsymbol{r}_\perp$ に分解します．

$$\boldsymbol{r} = \boldsymbol{r}_\parallel + \boldsymbol{r}_\perp. \tag{17}$$

ここで，

$$\begin{aligned} \boldsymbol{r}_\parallel &= (\boldsymbol{r}\cdot\hat{\boldsymbol{n}})\hat{\boldsymbol{n}}, \\ \boldsymbol{r}_\perp &= \boldsymbol{r} - (\boldsymbol{r}\cdot\hat{\boldsymbol{n}})\hat{\boldsymbol{n}}. \end{aligned} \tag{18}$$

図 8(a) の黒色の実線のベクトルとオレンジ色の実線のベクトルはそれぞれ $\boldsymbol{r}_\parallel$ と $\boldsymbol{r}_\perp$ を示す．

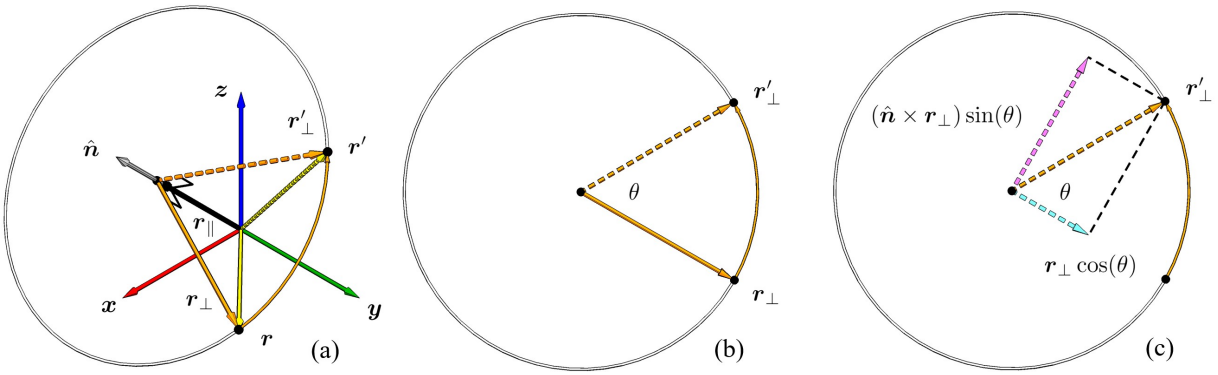


図 8: ロドリゲスの回転公式の導出．(a) ベクトル $\boldsymbol{r}$ が軸 $\hat{\boldsymbol{n}}$ 周りを回転して $\boldsymbol{r}'$ に移動しています．(b)$\hat{\boldsymbol{n}}$ の示す方向の逆方向からの視点．(c) ベクトル $\boldsymbol{r}'_\perp$ の成分分解．

回転後のベクトルとその平行および垂直の部分をそれぞれ $\boldsymbol{r}'$，$\boldsymbol{r}'_\parallel$，$\boldsymbol{r}'_\perp$ と記します．回転後，回転軸 $\hat{\boldsymbol{n}}$ に平行な部分は変化しませんので，

$$\boldsymbol{r}'_\parallel = \boldsymbol{r}_\parallel. \tag{19}$$

垂直な部分は回転軸を中心に角度 θ 回転します．この回転は図 8(b) に示す二次元平面上のみの変化となり，回転後の垂直な部分は以下の式で計算できます．

$$\boldsymbol{r}'_\perp = \boldsymbol{r}_\perp\cos(\theta) + (\hat{\boldsymbol{n}}\times\boldsymbol{r}_\perp)\sin(\theta). \tag{20}$$

回転後の平行部分と垂直部分を合成すると $\boldsymbol{r}'$ を得ます．

$$\begin{aligned} \boldsymbol{r}' &= \boldsymbol{r}'_\parallel + \boldsymbol{r}'_\perp \\ &= \boldsymbol{r}_\parallel + \boldsymbol{r}_\perp\cos(\theta) + (\hat{\boldsymbol{n}}\times\boldsymbol{r}_\perp)\sin(\theta). \end{aligned} \tag{21}$$

式 (18) を上式に代入します．

$$\begin{aligned} \boldsymbol{r}' &= (\boldsymbol{r}\cdot\hat{\boldsymbol{n}})\hat{\boldsymbol{n}} + (\boldsymbol{r} - (\boldsymbol{r}\cdot\hat{\boldsymbol{n}})\hat{\boldsymbol{n}})\cos(\theta) + (\hat{\boldsymbol{n}}\times(\boldsymbol{r} - (\boldsymbol{r}\cdot\hat{\boldsymbol{n}})\hat{\boldsymbol{n}}))\sin(\theta) \\ &= (\boldsymbol{r}\cdot\hat{\boldsymbol{n}})\hat{\boldsymbol{n}} + (\boldsymbol{r} - (\boldsymbol{r}\cdot\hat{\boldsymbol{n}})\hat{\boldsymbol{n}})\cos(\theta) + (\hat{\boldsymbol{n}}\times\boldsymbol{r})\sin(\theta) - (\hat{\boldsymbol{n}}\times(\boldsymbol{r}\cdot\hat{\boldsymbol{n}})\hat{\boldsymbol{n}})\sin(\theta). \end{aligned} \tag{22}$$

最後の項 $(\hat{\boldsymbol{n}} \times (\boldsymbol{r} \cdot \hat{\boldsymbol{n}})\hat{\boldsymbol{n}}) \sin(\theta)$ は自分自身の外積ですので，0 となります．

$$
\begin{aligned}
\boldsymbol{r}' &= (\boldsymbol{r} \cdot \hat{\boldsymbol{n}})\hat{\boldsymbol{n}} + (\boldsymbol{r} - (\boldsymbol{r} \cdot \hat{\boldsymbol{n}})\hat{\boldsymbol{n}}) \cos(\theta) + (\hat{\boldsymbol{n}} \times \boldsymbol{r}) \sin(\theta) \\
&= \boldsymbol{r} \cos(\theta) + (\boldsymbol{r} \cdot \hat{\boldsymbol{n}})\hat{\boldsymbol{n}}(1 - \cos(\theta)) + (\hat{\boldsymbol{n}} \times \boldsymbol{r}) \sin(\theta) \\
&= (\cos(\theta))\boldsymbol{r} + (\boldsymbol{r} + \hat{\boldsymbol{n}} \times (\hat{\boldsymbol{n}} \times \boldsymbol{r}))(1 - \cos(\theta)) + ([\hat{\boldsymbol{n}}\times] \sin(\theta))\boldsymbol{r} \\
&= \boldsymbol{r} + ([\hat{\boldsymbol{n}}\times] \sin(\theta))\boldsymbol{r} + ([\hat{\boldsymbol{n}}\times][\hat{\boldsymbol{n}}\times](1 - \cos(\theta)))\boldsymbol{r} \\
&= (\boldsymbol{I} + \sin(\theta)[\hat{\boldsymbol{n}}\times] + (1 - \cos(\theta))[\hat{\boldsymbol{n}}\times]^2)\boldsymbol{r}
\end{aligned} \tag{23}
$$

この式の最後の右側の係数の部分は回転行列となり，これによって，式 (15) のロドリゲスの回転公式が成立することを証明できました．

WRS システムの basis.robot_math 内の関数 rotmat_from_axangle は，回転行列と軸-角度表現の変換機能を提供しています．例えば，$\boldsymbol{x}$ 軸周りに 30 度だけ回転する回転行列を求めたければ，リスト 3.3-1 のプログラムを作成してください．ここで関数 rotmat_from_axangle には式 (15) が実装されており，ロドリゲスの回転公式を利用して，軸と軸周りの回転角度に相当する回転行列を計算できます．

リスト 3.3-1: 回転行列の計算

```
import math
import numpy as np
import basis.robot_math as rm

x_ax = np.array([1,0,0])
x_30_rotmat = rm.rotmat_from_axangle(x_ax, math.radians(30))
print(x_30_rotmat)
```

また，本章に示した視覚的に理解しやすい図も，basis/robot_math.py に用意した回転変換の関数と，modeling/-geomeric_model.py に用意された幾何形体を書き出すための関数を活用して作成されました．例えば，図 7(c) の生成は，リスト 3.3-2 のプログラムを実行すれば実現できます．

リスト 3.3-2: 図 7(c) の生成

```
import visualization.panda.world as wd
import modeling.geometric_model as mgm
import basis.robot_math as rm
import numpy as np

base = wd.World(cam_pos=[1, 1, 1], lookat_pos=[0, 0, 0], toggle_debug=True)
# 座標系0
rotmat_o = np.eye(3)
mgm.gen_frame(rotmat=rotmat_o, ax_length=.2).attach_to(base)
# 座標系A
## オイラーの表現で回転行列を求めます
rotmat_a = rm.rotmat_from_euler(np.pi / 3, -np.pi / 6, np.pi / 3)
mgm.gen_dashed_frame(rotmat=rotmat_a, ax_length=.2).attach_to(base)
# 座標系0のx,y軸を囲む円を表示します
mgm.gen_torus(axis=rotmat_o[:,2], major_radius=.2, minor_radius=.0015, rgb=np.array([1, 1,
    0]), n_sec_major=64).attach_to(base)
# 座標系Aのx,yを囲む円を表示します
mgm.gen_dashed_torus(axis=rotmat_a[:, 2], major_radius=.2, minor_radius=.0015, rgb=np.
  array([1, 1, 0]), len_interval=.007, len_solid=.01, n_sec_major=64).attach_to(base)
# オイラーで求めた回転行列を軸-角度形式に変換する
ax, angle = rm.axangle_between_rotmat(np.eye(3), rotmat_a)
# 回転軸を書き出します
mgm.gen_arrow(epos=ax * .4, rgb=np.array([0, 0, 0])).attach_to(base)
# 回転の中間状態10個に離散化して画面に表示します
for step_angle in np.linspace(0, angle, 10).tolist():
    rotmat_ = rm.rotmat_from_axangle(ax, step_angle)
    mgm.gen_dashed_frame(ax_length=.2, rotmat=rotmat_).attach_to(base)
    mgm.gen_dashed_torus(axis=rotmat_[:, 2], major_radius=.2, minor_radius=.0015, rgb=np.
      array([1, 1, 0]), len_interval=.007, len_solid=.01, n_sec_major=64).attach_to(base)
# 大きい包囲球を描きます
mgm.gen_sphere(radius=.2, rgb=rm.bc.gray, alpha=.8, ico_level=5).attach_to(base)
base.run()
```

3.4 二つの回転の差

ロドリゲスの回転公式を用いることで，第 3.2 節の式 (9) をより直接的に導出することができます．式 (15) を展開すると，次式のような行列が得られます．

$$\boldsymbol{R} = \begin{pmatrix} \hat{n_1}^2(1-\cos(\theta)) + \cos(\theta) & \hat{n_1}\hat{n_2}(1-\cos(\theta)) - n_3\sin(\theta) & \hat{n_1}\hat{n_3}(1-\cos(\theta)) + n_2\sin(\theta) \\ \hat{n_1}\hat{n_2}(1-\cos(\theta)) + n_3\sin(\theta) & \hat{n_2}^2(1-\cos(\theta)) + \cos(\theta) & \hat{n_2}\hat{n_3}(1-\cos(\theta)) - n_1\sin(\theta) \\ \hat{n_1}\hat{n_3}(1-\cos(\theta)) - n_2\sin(\theta) & \hat{n_2}\hat{n_3}(1-\cos(\theta)) + n_1\sin(\theta) & \hat{n_3}^2(1-\cos(\theta)) + \cos(\theta) \end{pmatrix} \tag{24}$$

この行列のトレースは $1 + 2\cos(\theta)$ であり，式 (9) も同時に成立することが分かります．

回転の軸-角形式で記述される角度は，回転の最小角度です．第 2 節でオイラー角表示について紹介した際，オイラー角表示は一般的に冗長であることを述べました．これは，オイラー角表示が一連の回転軸と回転角度の組み合わせを含んでおり，そのような組み合わせは軸角形式で記述される移動量よりも大きくなるためです．軸-角形式で表される回転角が最小であるという特性を利用することで，二つの座標系間の最小回転距離を計算することができます．座標系 Σ_A を Σ_B 考え，それぞれの回転行列 $\boldsymbol{R}_A$ を $\boldsymbol{R}_B$ とします．Σ_A から Σ_B への回転は $\boldsymbol{R}_B\boldsymbol{R}_A^\top$ で計算できます．最小回転角度は，行列 $\boldsymbol{R}_B\boldsymbol{R}_A^\top$ の軸－角形式の回転角度を計算することで求められます．つまり，

$$\theta = \cos^{-1}\frac{\mathtt{tr}(\boldsymbol{R}_B\boldsymbol{R}_A^\top) - 1}{2}. \tag{25}$$

を用いて，二つの回転行列の差を計算します．

WRS システムの basis.robot_math の axangle_between_rotmat 関数では，上記の差の計算を実装しています．この関数の返り値は回転角度と回転軸の二つの量です．

3.5 回転ベクトルと四元数

回転の軸-角度の表現は軸と角度の二つの量を持ち，計算には適していません．計算を容易にするために，回転ベクトルと四元数の 2 つの具体的な形式がよく利用されます．本節は，回転ベクトルと四元数をそれぞれ詳しく説明します．

3.5.1 回転ベクトル

回転ベクトルは，軸-角度の表現を基にした回転の形式で，回転軸と回転角度を一つの三次元ベクトルにまとめたものです．式 (26) は回転ベクトルの定義を示しています．θ と $\hat{\boldsymbol{n}}$ はそれぞれ回転角度と回転軸を指します．

$$\boldsymbol{\omega} = \theta\hat{\boldsymbol{n}} \tag{26}$$

回転ベクトルはただの回転角度と単位回転軸の積であるため，直感的に理解しやすいです．計算上，回転角度が小さい場合，二つの回転ベクトルの線形和を用いて回転を合成することができます．つまり，2 つの小角度回転 $\boldsymbol{\omega}_1 = \theta_1\hat{\boldsymbol{n}}_1$ と $\boldsymbol{\omega}_2 = \theta_2\hat{\boldsymbol{n}}_2$ を考えます．これらを合成した回転ベクトル $\boldsymbol{\omega}_{12}$ は以下のように近似できます．

$$\boldsymbol{\omega}_{12} \approx \boldsymbol{\omega}_1 + \boldsymbol{\omega}_2 \tag{27}$$

小角度回転の合成がベクトルの線形和として近似できることを証明するために，ロドリゲスの回転公式を使って説明します．小さな角度 θ に対して，以下の近似が成り立ちます．

$$\cos(\theta)) \approx 1,\ \sin(\theta) \approx \theta \tag{28}$$

したがって，回転行列 $\boldsymbol{R}$ はロドリゲス回転公式を利用して次のように近似できます．

$$\boldsymbol{R} = \boldsymbol{I} + \sin(\theta)[\hat{\boldsymbol{n}}\times] + (1-\cos(\theta))[\hat{\boldsymbol{n}}\times]^2 \approx \boldsymbol{I} + \theta[\hat{\boldsymbol{n}}\times]. \tag{29}$$

2 つの小角度回転 $\boldsymbol{\omega}_1 = \theta_1 \hat{\boldsymbol{n}}_1$ と $\boldsymbol{\omega}_2 = \theta_2 \hat{\boldsymbol{n}}_2$ の場合，これらに対応する回転行列はそれぞれ次のようになります.

$$\begin{aligned} \boldsymbol{R}_1 &= \boldsymbol{I} + \theta_1 [\hat{\boldsymbol{n}}_1 \times] \\ \boldsymbol{R}_2 &= \boldsymbol{I} + \theta_2 [\hat{\boldsymbol{n}}_2 \times] \end{aligned} \tag{30}$$

2 つの回転の合成は，行列の積 $\boldsymbol{R} = \boldsymbol{R}_1 \boldsymbol{R}_2$ として表します．すなわち，

$$\boldsymbol{R} \approx (\boldsymbol{I} + \theta_1 [\hat{\boldsymbol{n}}_1 \times])(\boldsymbol{I} + \theta_2 [\hat{\boldsymbol{n}}_2 \times]) = \boldsymbol{I} + \theta_1 [\hat{\boldsymbol{n}}_1 \times] + \theta_2 [\hat{\boldsymbol{n}}_2 \times] + \theta_1 \theta_2 [\hat{\boldsymbol{n}}_1 \times][\hat{\boldsymbol{n}}_2 \times] \tag{31}$$

です．角度は小さいので $\theta_1 \theta_2 [\hat{\boldsymbol{n}}_1 \times][\hat{\boldsymbol{n}}_2 \times]$ のような高次項を無視すると，以下の式が貰えます.

$$\boldsymbol{R} \approx \boldsymbol{I} + \theta_1 [\hat{\boldsymbol{n}}_1 \times] + \theta_2 [\hat{\boldsymbol{n}}_2 \times] = \boldsymbol{I} + [(\theta_1 \hat{\boldsymbol{n}}_1 + \theta_2 \hat{\boldsymbol{n}}_2) \times] \tag{32}$$

この式の右側の項は，歪対称行列の定義に基づいて導き出されています．これは，ちょうど二つの回転ベクトルの線形和の歪対称行列形式です．したがって，回転角度が小さい場合，回転の合成は回転ベクトルの線形和に相当します．この回転ベクトルの特性は，回転の微分計算においてよく使用されます．次の章の直列リンクの逆運動学の計算もこの特性を利用します.

3.5.2 四元数

四元数は，回転軸と角度の表現のもう一つの具体的な形態として見ることができます．これは次のように書かれます.

$$q = \cos(\theta/2) + (n_x i + n_y j + n_z k) \sin(\theta/2) \tag{33}$$

三次元ベクトル $\boldsymbol{r} = (r_x, r_y, r_z)$ を回転するには，このベクトルを純虚四元数として表現します.

$$\boldsymbol{r} = 0 + r_x i + r_y j + r_z k \tag{34}$$

$\boldsymbol{r}$ を q に表れた回転で回転する場合，以下の式を使用します.

$$\boldsymbol{r}' = q \boldsymbol{r} q^* \tag{35}$$

q^* は四元数 q の共役で，次のように定義されます.

$$q = \cos(\theta/2) - (n_x i + n_y j + n_z k) \sin(\theta/2) \tag{36}$$

式 (38) 右側の計算は四元数の積算です．四元数の積算はただの虚数の演算となります．例えば，二つの四元数 q と p の積算は以下のように行われます.

$$\begin{aligned} qp &= (q_0 + q_1 i + q_2 j + q_3 k)(p_0 + p_1 i + p_2 j + p_3 k) \\ &= (q_0 p_0 - q_1 p_1 - q_2 p_2 - q_3 p_3) + (q_0 p_1 + q_1 p_0 + q_2 p_3 - q_3 p_2) i \\ &\quad + (q_0 p_2 - q_1 p_3 + q_2 p_0 + q_3 p_1) j + (q_0 p_3 + q_1 p_2 - q_2 p_1 + q_3 p_0) k \end{aligned} \tag{37}$$

式 (38) の右側の積算を完全に展開すると，回転行列の行列演算のと完全に同じ形式の係数を貰えます．そのため，四元数と回転行列は等価することが分かります.

複数の四元数で表す回転の合成は，対応する四元数の積の計算となります．例えば，二つの回転 q_1 と q_2 を連続して適用する場合，合成された回転は $q_2 q_1$ で表されます．合成後の回転は

$$\boldsymbol{r}' = (q_2 q_1) \boldsymbol{r} (q_2 q_1)^* = q_1 (q_2 \boldsymbol{r} q_1^*) q_2^* \tag{38}$$

となります．積の計算は結果的に，式 (38) に示された四元数回転公式と一致することが分かります．ここで，四元数の積 $q_2 q_1$ の共役 $(q_2 q_1)^*$ は $q_1^* q_2^*$ であることを利用しています．証明は各自にお任せします.

回転の四元数形式の最も重要な応用は回転の補間です．二つの四元数 q_1 と q_2 の間の補間を $t, 0 \leq t \leq 1$ に依存して行う場合，補間された四元数 $q(t)$ は次のように表されます．

$$q(t) = \frac{\sin((1-t)\phi)}{\sin(\phi)} q_1 + \frac{\sin(t\phi)}{\sin(\phi)} q_2. \tag{39}$$

ここで，ϕ は四元数 q_1 と q_2 の間の角度で，内積を用いて次のように計算されます．

$$\phi = \arccos(q_1 q_2) \tag{40}$$

式 (39) に示された補完は球面線形補間 (Spherical Linear Interpolation, SLERP) と呼ばれ，sin() 関数を持つことで，球面上の大円を使って補間を行います．これにより，補間が回転の角度に対して単純な線形補間 $((1-t)q_1 + tq_2)$ になるのではなく，補間の際に回転の角度が均等に分布され，回転の方向が滑らかに変化するようになります．

WRS システムの basis/robot_math には，回転ベクトルや四元数に関する多くの操作関数が提供されています．例えば，delta_w_between_rotomat 関数の計算結果は，二つの回転行列の差を回転ベクトル形式で表したものです．quaternion_で始まる関数は，四元数に関するいくつかの操作を提供します．quaternion_slerp 関数は，二つの四元数の補間を行うことができます．以下のコードは、平行移動と回転を同時に補間する例を示しています。 modeling.geometric_model の中にある GeometricModel を使用して bunny_sim.stl ファイルを読み込み、start_pos、start_rotmat から goal_pos、goal_rotmat まで徐々に補間して移動する各姿勢を表示します。このコードの結果は図 9 に示します．

リスト 3.3-3: 四元数形式を利用した補間

```
import numpy as np
import basis.robot_math as rm
import modeling.geometric_model as mgm
import visualization.panda.world as wd

base = wd.World(cam_pos=np.array([2, .0,.5]), lookat_pos=np.zeros(3))
bunny = mgm.GeometricModel(initor="objects/bunnysim.stl")
bunny.alpha=.3
start_pos = np.array([.0, 0.5, .0])
start_rotmat = np.eye(3)
goal_pos = np.array([.5, -.5, .0])
goal_rotmat = rm.rotmat_from_euler(np.pi , np.pi/2, 0)
start_quaternion = rm.rotmat_to_quaternion(start_rotmat)
goal_quaternion = rm.rotmat_to_quaternion(goal_rotmat)
for t in np.linspace(0, 1, 20):
    pos = start_pos * (1 - t) + goal_pos * t
    quat = rm.quaternion_slerp(start_quaternion, goal_quaternion, fraction=t)
    bunny_copy = bunny.copy()
    bunny_copy.rgb=rm.bc.cool_map(t)
    mgm.gen_frame(ax_radius=.0015).attach_to(bunny_copy)
    bunny_copy.pose = (pos, rm.quaternion_to_rotmat(quat))
    bunny_copy.attach_to(base)
base.run()
```

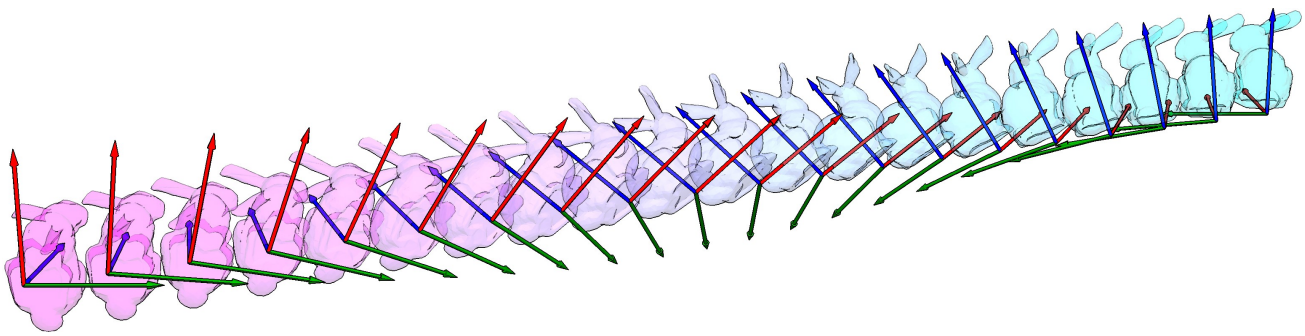


図 9: リスト 3.3-3 の実行結果.