

---

# CPS-lecture 説明資料

次世代半導体・センサ科学研究所

Information Robotic System Laboratory

(IRSL, 垣内研究室)

- Cyber-Physical-System (サイバー フィジカル システム)
  - ロボットの全体システムの構成方法を学びながらCPSについての理解を深める
- <https://www.jeita.or.jp/cps/about/> (電子情報技術産業協会の説明)
  - CPSとは、実世界(フィジカル空間)にある多様なデータをセンサーネットワーク等で収集し、サイバー空間で大規模データ処理技術等を駆使して分析／知識化を行い、そこで創出した情報／価値によって、産業の活性化や社会問題の解決を図っていくものです。
- ロボット利用システム (CPSと考えられる)
  - ロボットシステム(CPS要素)を利用したシステム(CPSの社会実装)
  - ロボットシステムがCPSのPhysicalに相当する
  - 利用システム内で、ロボットシステムでがデータ収集をしたり、仕事をしたりする

# CPS講義

---

- CPS講義のページ
  - <https://irsl-tut.github.io/CPS-lecture/index.ja.html>
- 垣内研のホームページ
  - <https://irsl.eiiris.tut.ac.jp/>
- 連絡先
  - [faculty@irsl.eiiris.tut.ac.jp](mailto:faculty@irsl.eiiris.tut.ac.jp)
  - CPS-lectureや他のgithub discussion/issue

# はじめに

---

- Windowsで行うことが前提になっています
  - 開発はLinux(Ubuntu)で行っているので、Ubuntuでの動作は問題ありません
  - MacOSは未確認ですが、動くのではないかと思います。
- 推奨するエディタ
  - VSCode (visual studio code)
  - Microsoft Store からインストールできる
  - 他のエディタを常に使っているならそれで構いません
- 推奨するブラウザ
  - Google Chromeでチェックしています。
  - Microsoft Edge, Firefox, Safari などでも問題ないと思います。

# Jupyterの使い方

- Jupyterへ繋いでみよう
  - IDとパスワードを配ります

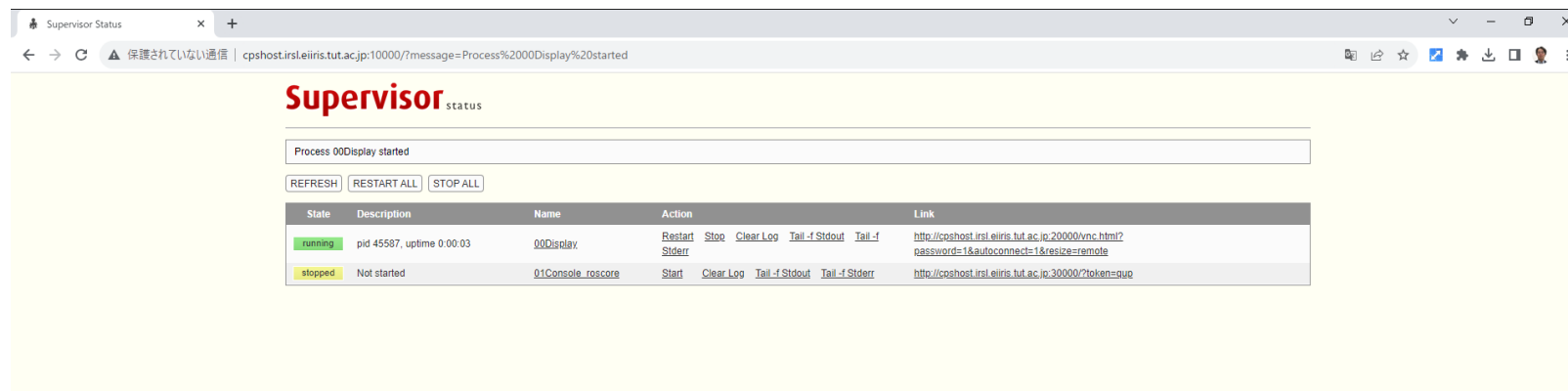
- 画面を出してみよう

- カーネルの実行

- Jupyterへのファイルのアップロード

- ディレクトリ <https://github.com/IRSL-tut/CPS-lecture/tree/main/notebooks>
- [https://github.com/IRSL-tut/CPS-lecture/raw/main/notebooks/cps\\_lecture\\_python.ipynb](https://github.com/IRSL-tut/CPS-lecture/raw/main/notebooks/cps_lecture_python.ipynb)

- Jupyterからのファイルのダウンロード



# Jupyterの使い方

---

- ノートブック(ipynb)の使い方

- Restart kernel
- Shutdown kernel

- モードの違い

- エディットモード
- コマンドモード

- タイプの違い

- Codeタイプ
- Markdownタイプ

- 資料

- <https://github.com/IRSL-tut/CPS-lecture/wiki/Jupyter%E3%81%AE%E4%BD%BF%E3%81%84%E6%96%B9>

# ロボットプログラミング / プログラミング基礎 (Python)

---

- プログラミング基礎 (Python)

- <https://github.com/IRSL-tut/CPS-lecture/wiki/Python%E3%83%97%E3%83%AD%E3%82%B0%E3%83%A9%E3%83%9F%E3%83%B3%E3%82%B0>
- ノートブック
  - [https://github.com/IRSL-tut/CPS-lecture/blob/main/notebooks/cps\\_lecture\\_python.ipynb](https://github.com/IRSL-tut/CPS-lecture/blob/main/notebooks/cps_lecture_python.ipynb)

# ロボットプログラミング / 3次元座標系のプログラミング

---

- 3次元座標系のプログラミング

- ノートブック

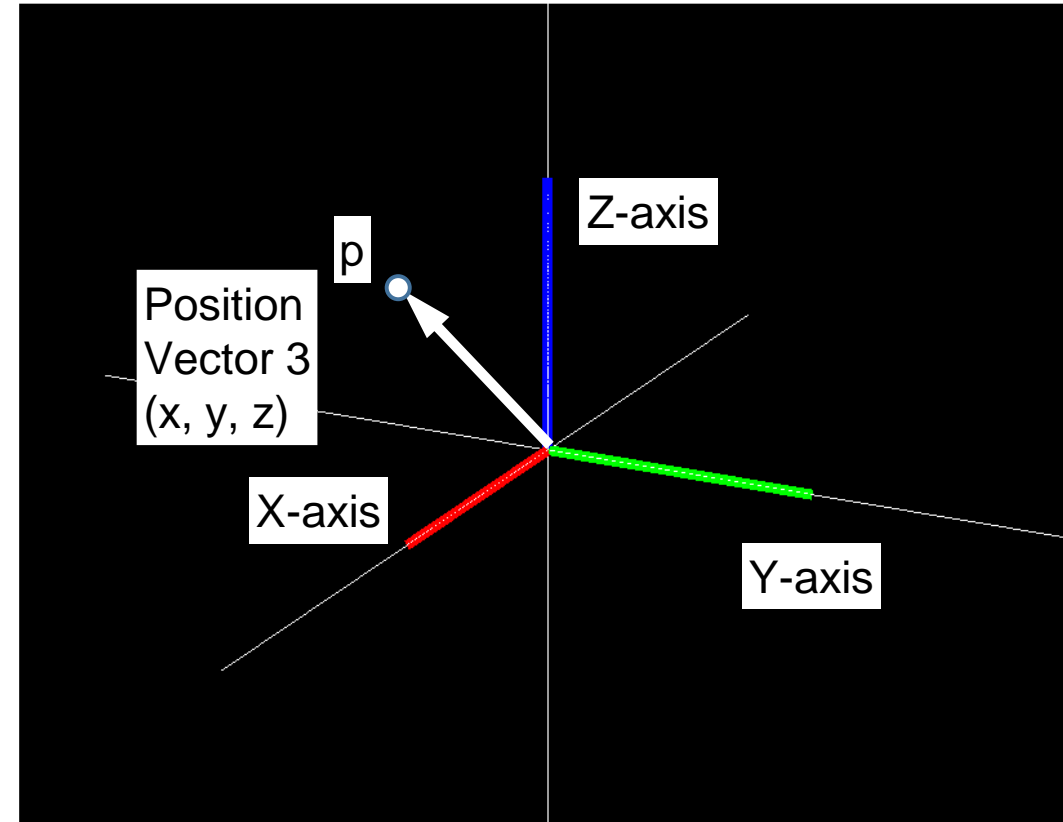
- [https://github.com/IRSL-tut/CPS-lecture/blob/main/notebooks/cps\\_lecture\\_coords00.ipynb](https://github.com/IRSL-tut/CPS-lecture/blob/main/notebooks/cps_lecture_coords00.ipynb)
    - [https://github.com/IRSL-tut/CPS-lecture/blob/main/notebooks/cps\\_lecture\\_coords01.ipynb](https://github.com/IRSL-tut/CPS-lecture/blob/main/notebooks/cps_lecture_coords01.ipynb)
    - [https://github.com/IRSL-tut/CPS-lecture/blob/main/notebooks/cps\\_lecture\\_coords02.ipynb](https://github.com/IRSL-tut/CPS-lecture/blob/main/notebooks/cps_lecture_coords02.ipynb)



# ロボットプログラミング 3次元座標系解説1

- 3次元位置

- 3次元の位置は3要素の実数ベクトルで表される
- 要素はそれぞれ、x軸位置、y軸位置、z軸位置である
- 位置pとして表す



# ロボットプログラミング 3次元座標系解説2

## • 3次元の回転

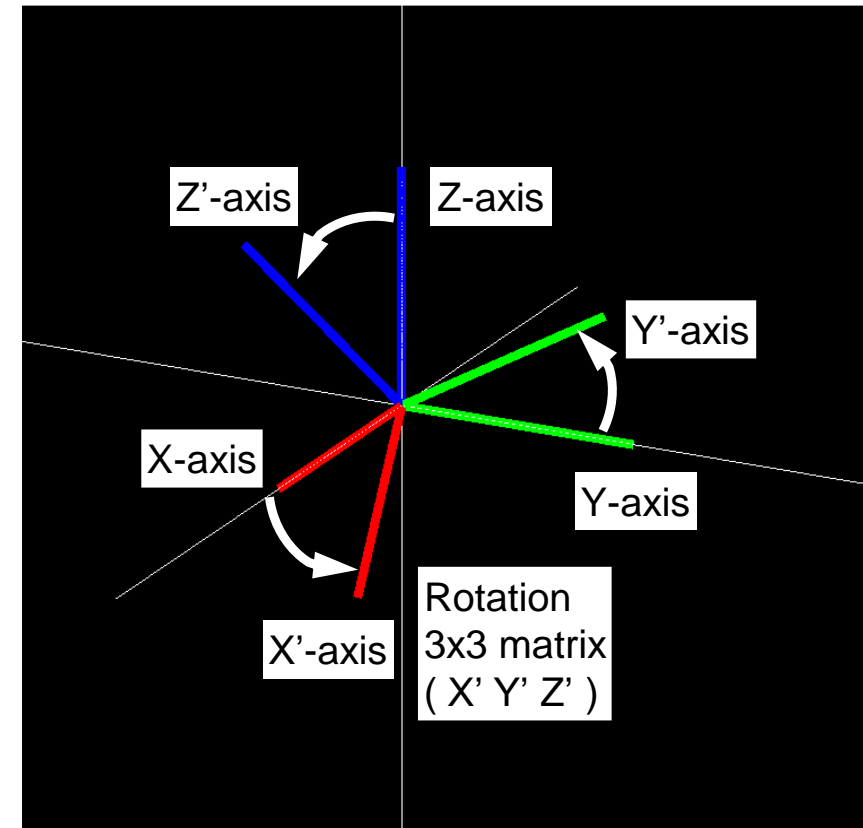
- 3次元の回転は回転行列を用いて表すことができる
- 回転行列は3x3の行列である(行列Rとして表す)
- 行列の各列が回転したxyz軸になっている

$$\mathbf{R} = \begin{bmatrix} \mathbf{x} & \mathbf{y} & \mathbf{z} \end{bmatrix}$$

- 直交行列であり、逆行列と転置行列が同じである

$$\mathbf{R}^{-1} = \mathbf{R}^T$$

- 回転行列の他に以下の表し方がある
  - RPY(Roll Pitch Yaw)
  - AngleAxis
  - Quaternion



# ロボットプログラミング 3次元座標系解説3

## • 3次元の座標系

- 3次元の座標系は3次元の位置と回転によって表される
- 計算のために4x4の同時変換行列を用いる
- 行列Rと位置pを用いて以下のように定義する

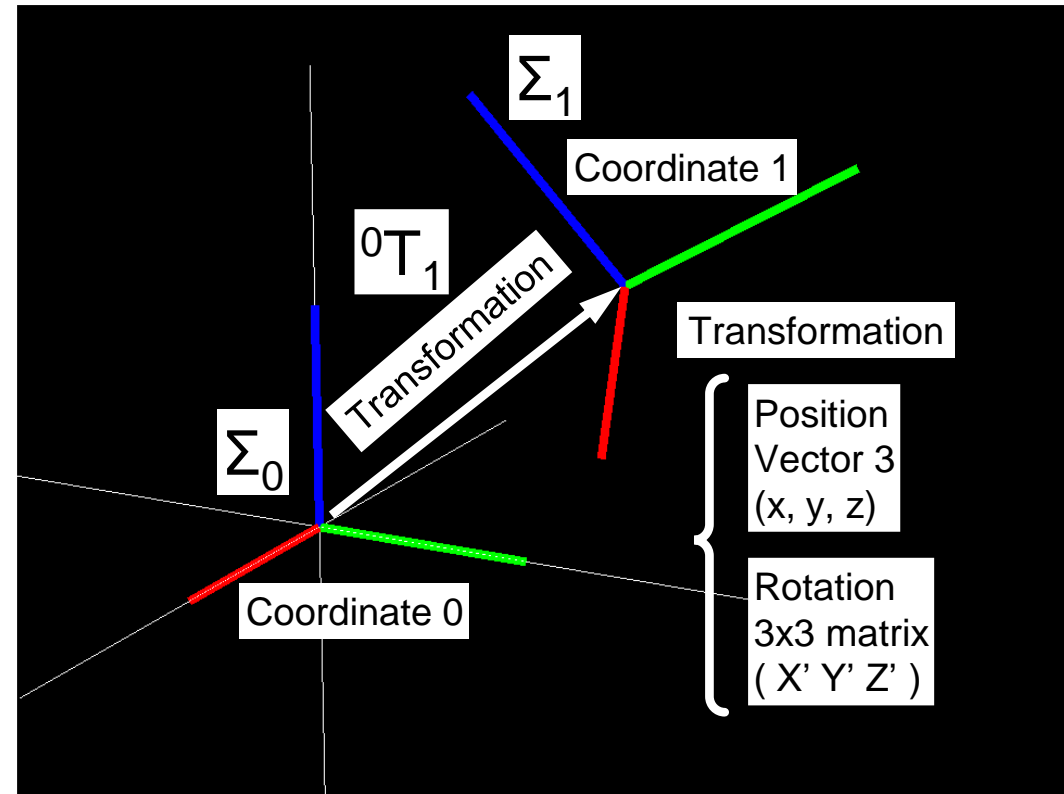
$$T = \begin{pmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0} & 1 \end{pmatrix}$$

- 逆行列

$$T^{-1} = \begin{pmatrix} \mathbf{R}^{-1} & -\mathbf{R}^{-1}\mathbf{p} \\ \mathbf{0} & 1 \end{pmatrix}$$

- 掛け算

$$T_a \times T_b = \begin{pmatrix} \mathbf{R}_a \mathbf{R}_b & \mathbf{R}_a \mathbf{p}_b + \mathbf{p}_a \\ \mathbf{0} & 1 \end{pmatrix}$$



# ロボットプログラミング 3次元座標系解説4

## • 3次元の座標系

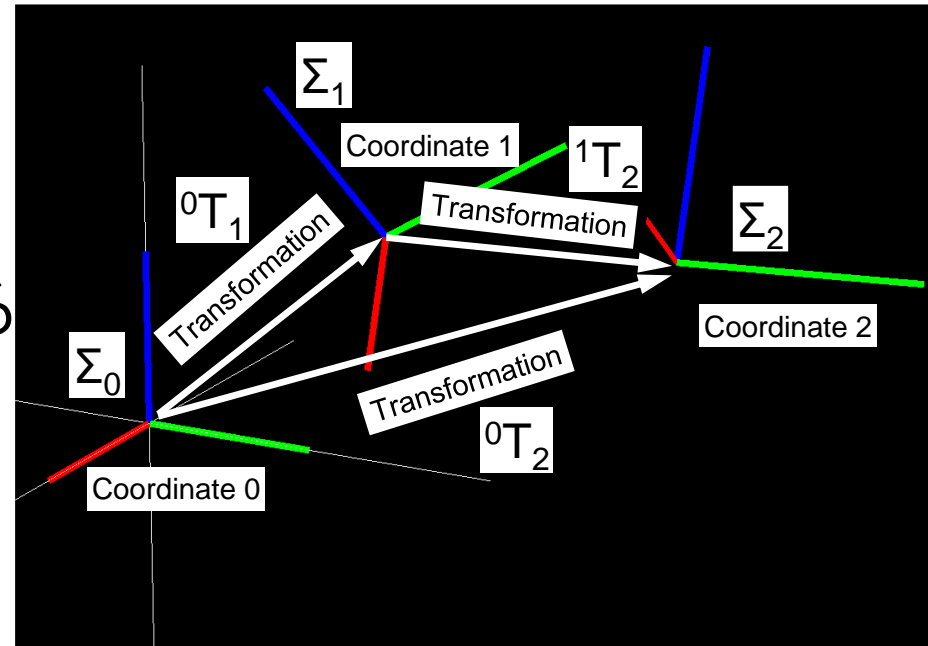
- 3次元の座標系を $\Sigma_{\langle\text{添え字}\rangle}$ にて表す
- 座標変換を ${}^{\langle\text{基準座標系}\rangle}T_{\langle\text{添え字}\rangle}$ にて表す
- ある座標系で表した位置を ${}^{\langle\text{基準座標系}\rangle}p$ にて表す
- 以下のルールを用いて考える
  - ワールド原点は $\Sigma_0$
  - 基準座標系を座標系の添え字にする
  - 座標変換の添え字を  
変換先の座標系添え字とする

$\Sigma_N$  のワールド基準の座標変換は  ${}^0T_N$  になる

$${}^0T_N {}^NT_M = {}^0T_M \text{ が成り立つ}$$

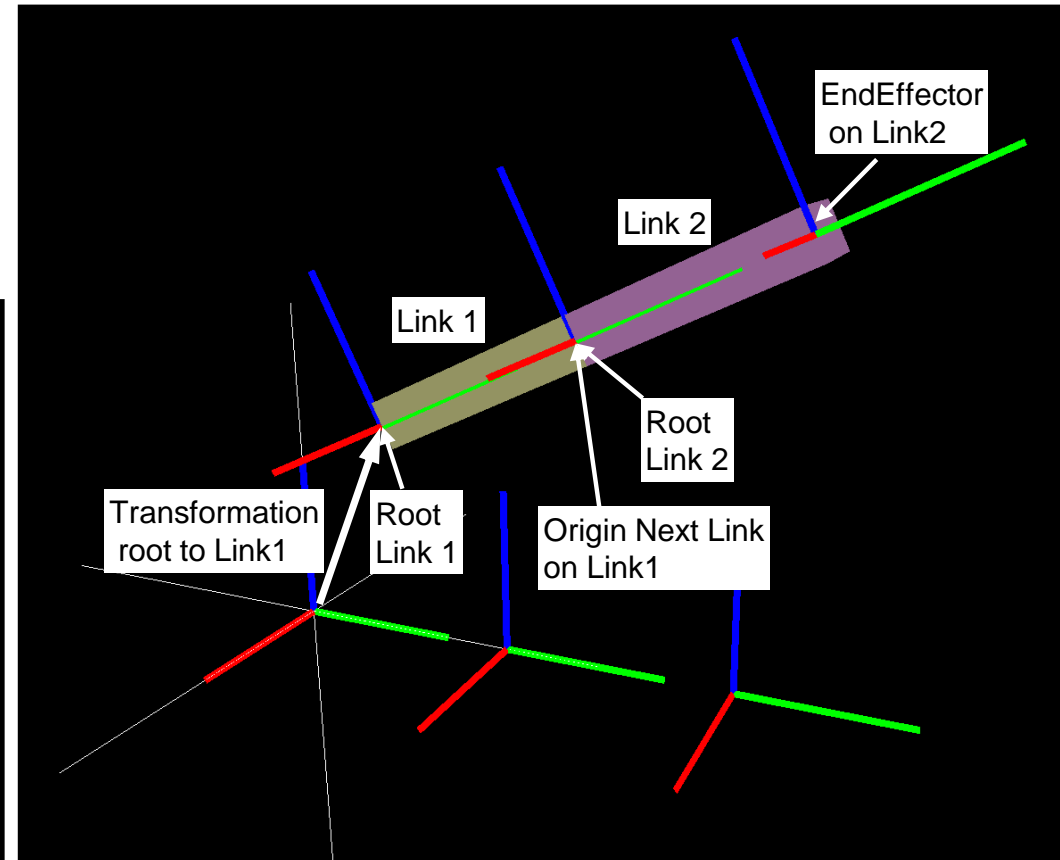
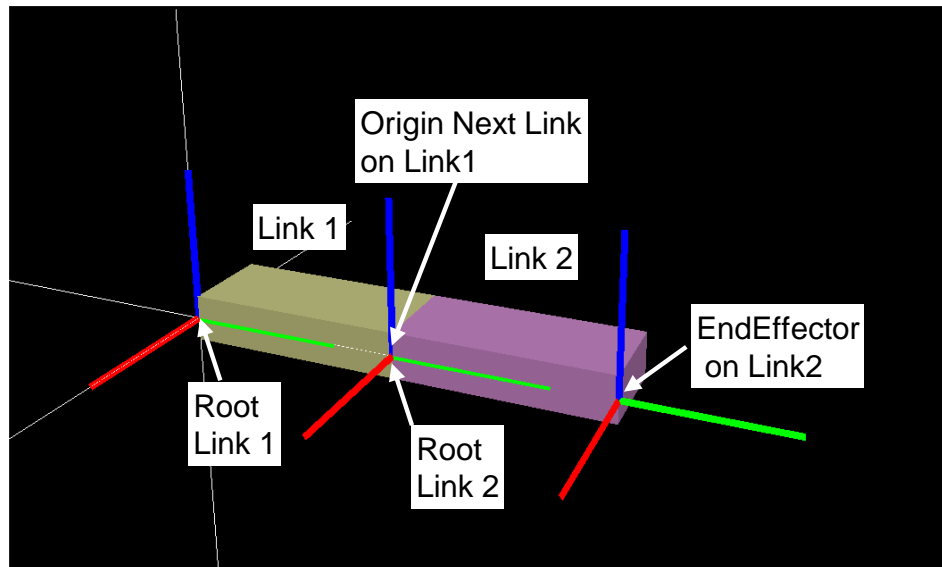
$${}^0T_N {}^Np = {}^0p \text{ が成り立つ}$$

$$({}^NT_M)^{-1} = {}^MT_N \text{ が成り立つ}$$



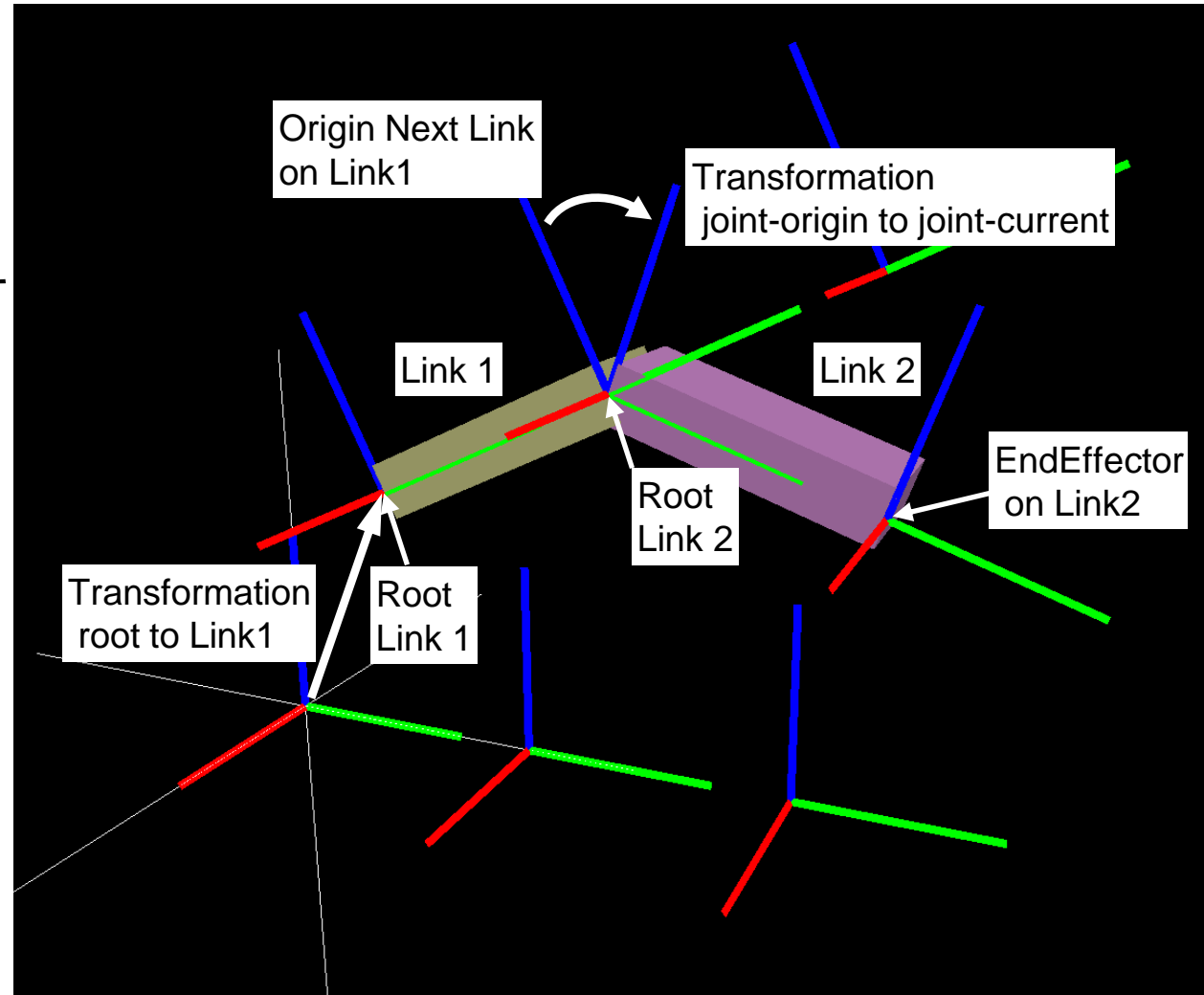
# ロボットプログラミング 3次元座標系解説5

- リンクの座標系
  - ルートリンク原点の移動



# ロボットプログラミング 3次元座標系解説6

- リンクの座標系
  - リンク原点の移動(中間リンク)
- ロボット等はリンク構造
  - 座標系がロボットの構造と一致させているのがロボットモデル
  - ルートリンク座標、途中のリンク間の移動(ロボットにおける関節駆動)があると、リンク全体の座標系を更新する必要があることが分かる
  - ロボットモデルはこの処理をする必要がある



# ロボットプログラミング / ロボットモデルのプログラミング

---

- ロボットモデルのプログラミング

- [https://github.com/IRSL-tut/CPS-lecture/blob/main/notebooks/cps\\_lecture\\_robot\\_model.ipynb](https://github.com/IRSL-tut/CPS-lecture/blob/main/notebooks/cps_lecture_robot_model.ipynb)

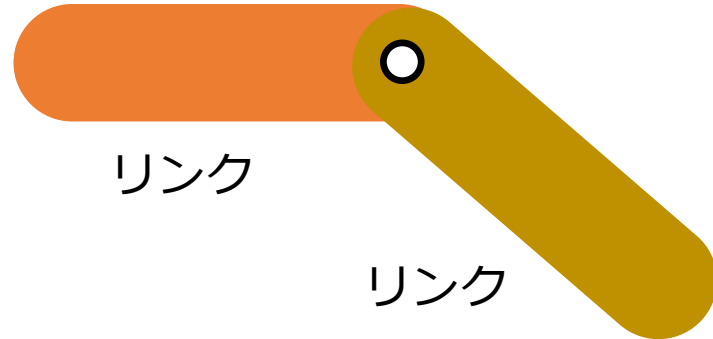
- 参考: HRP4C

- [https://www.aist.go.jp/aist\\_j/press\\_release/pr2009/pr20090316/pr20090316.html](https://www.aist.go.jp/aist_j/press_release/pr2009/pr20090316/pr20090316.html)

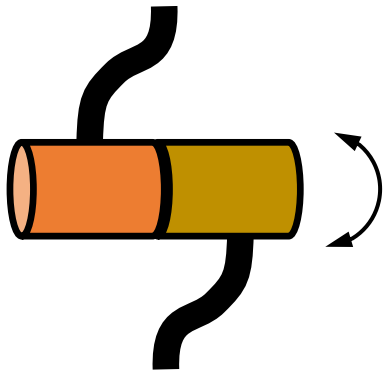
# ロボットプログラミング リンク連結構造

## リンク連結構造

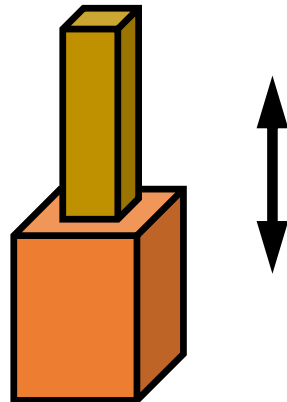
### 回転関節(ジョイント)



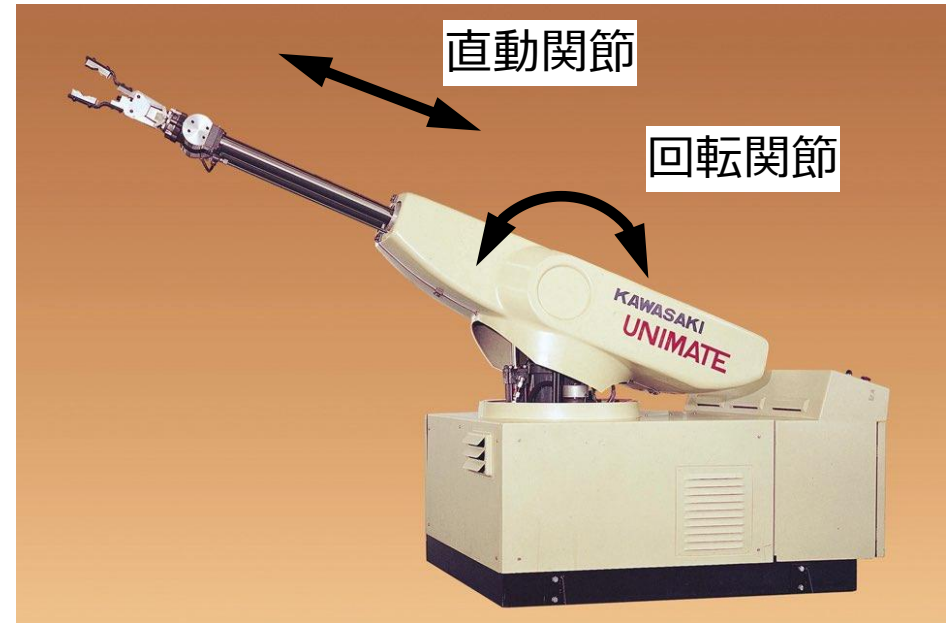
### 回転関節



### 直動関節



## 実ロボットでの関節の例



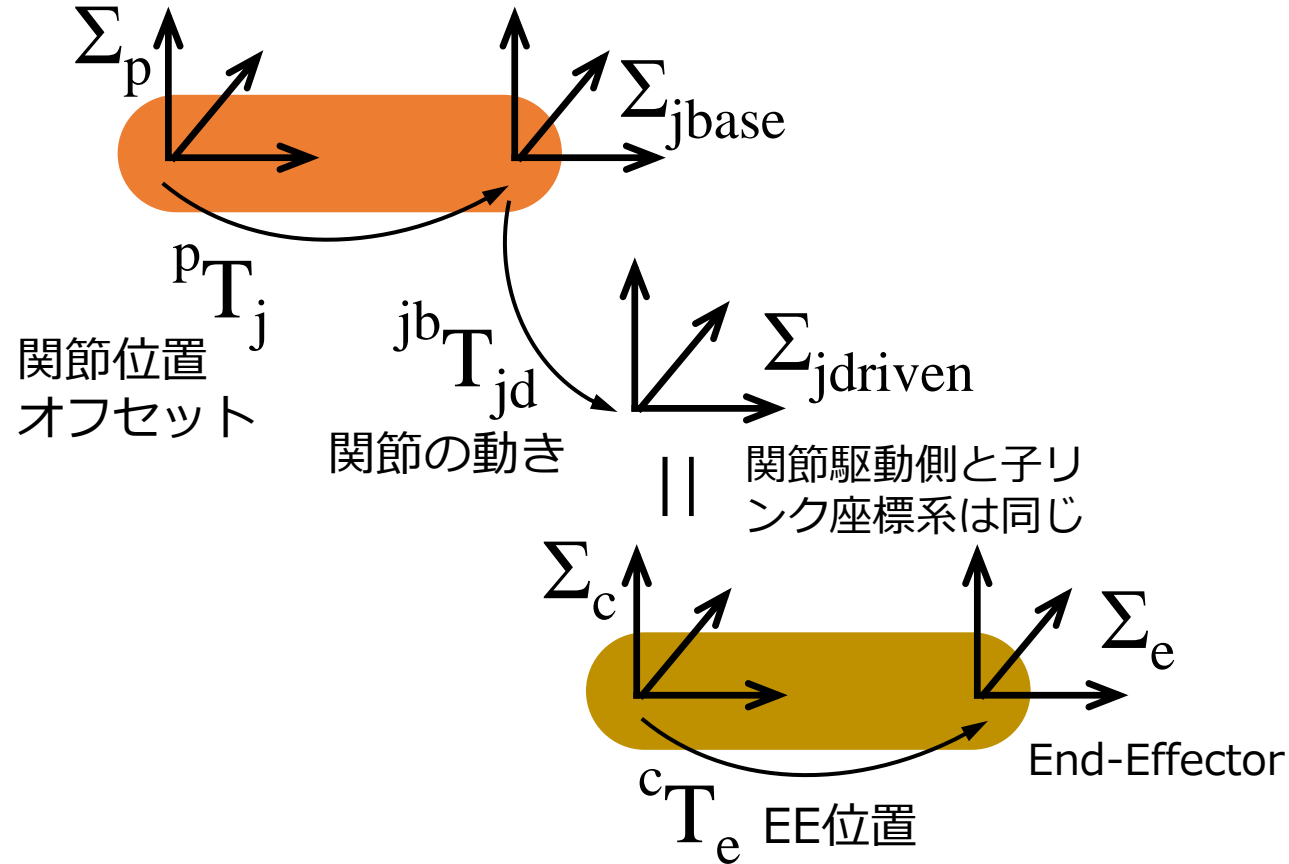
川崎重工業 ユニメイト2000型

[https://robotics.kawasaki.com/ja1/anniversary/history/history\\_02.html](https://robotics.kawasaki.com/ja1/anniversary/history/history_02.html)



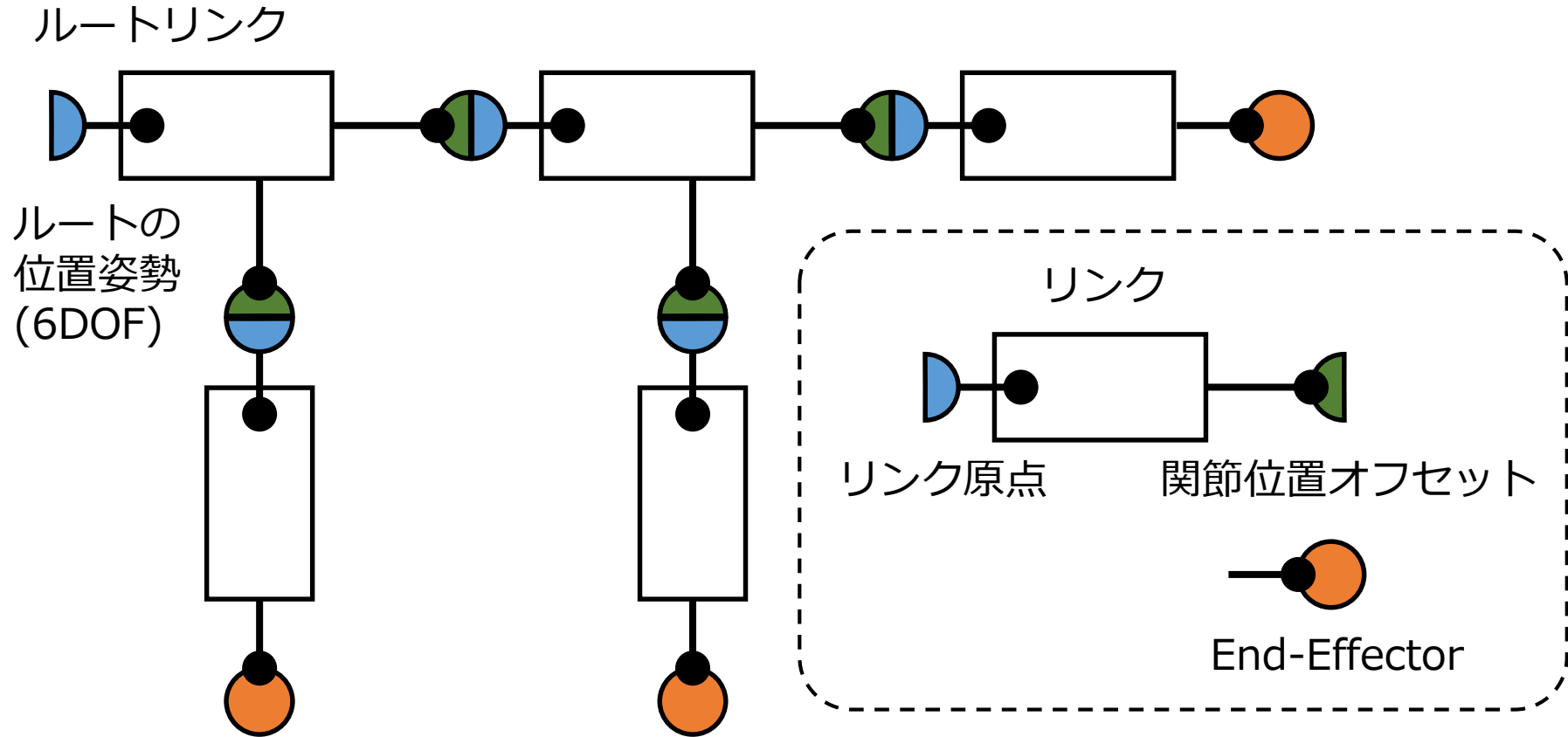
# ロボットプログラミング リンク連結構造の座標系の設定

リンク構造を表す座標系の設定

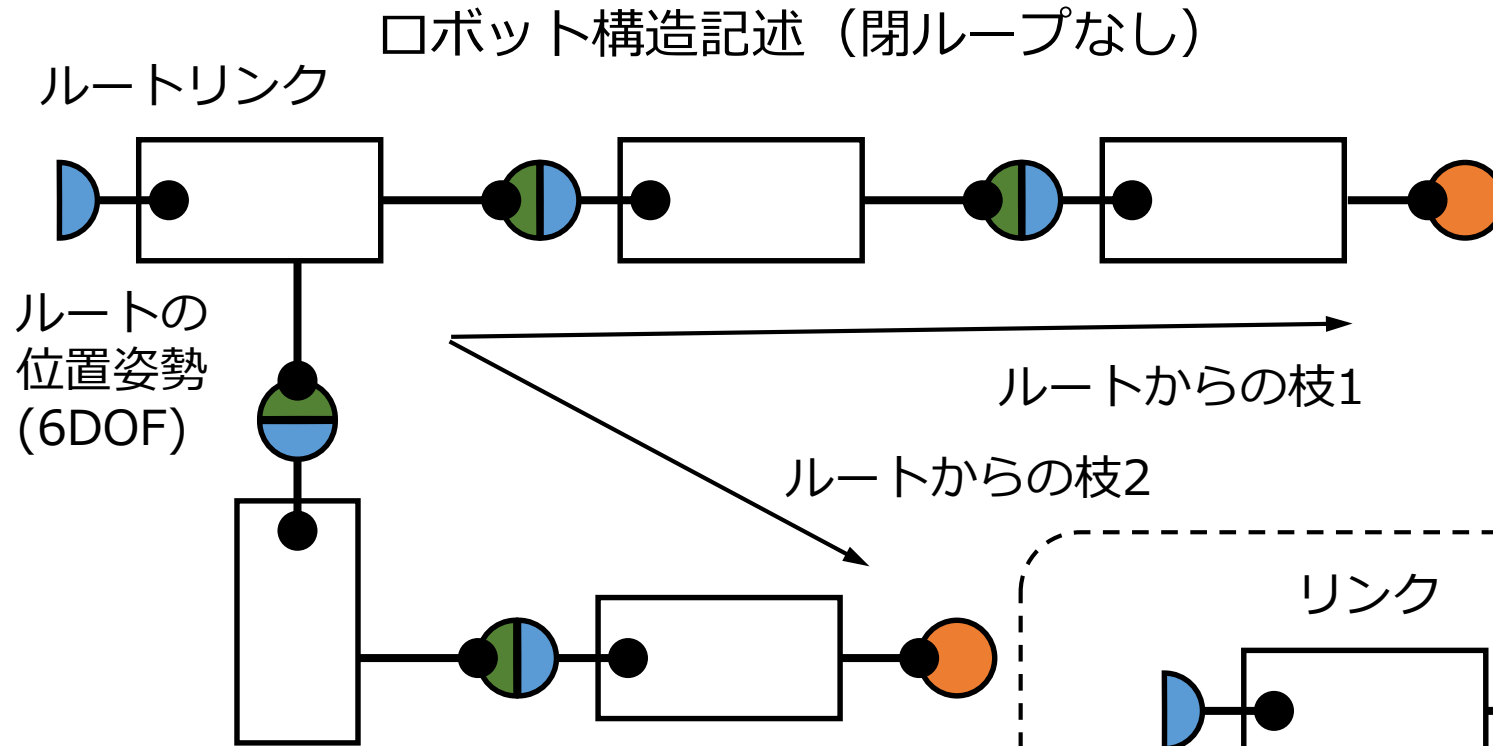


# ロボットプログラミング リンク連結構造によるロボット構造の記述

ロボット構造記述（閉ループなし）



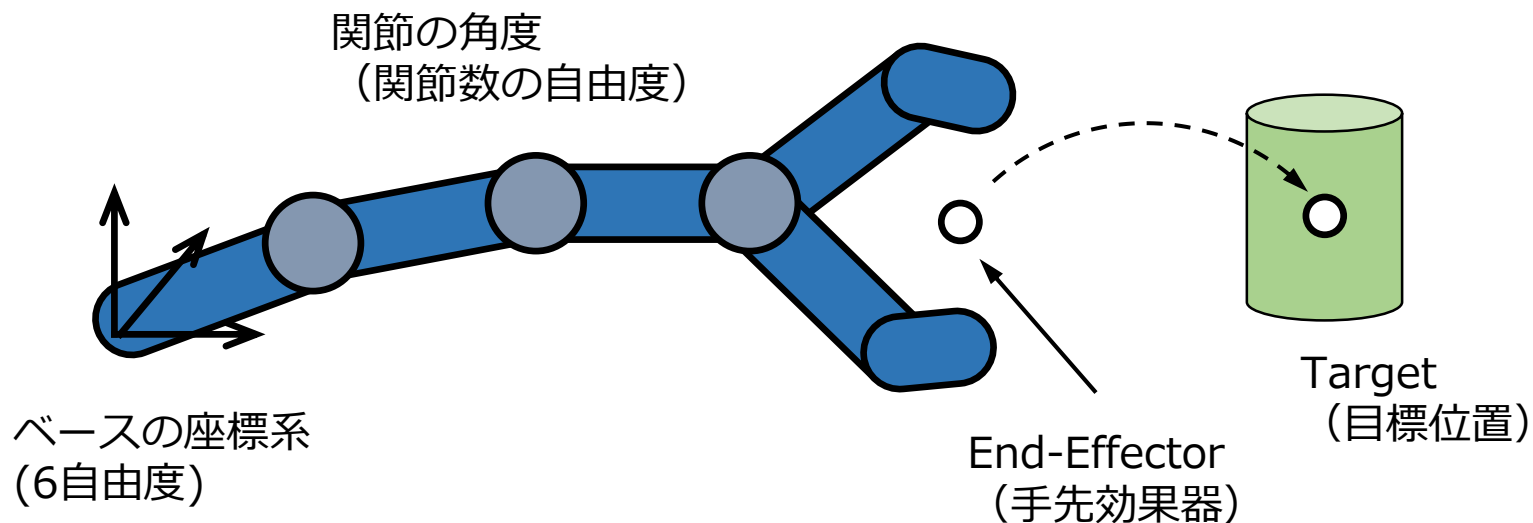
# ロボットプログラミング ロボットはツリー構造



あまり本質的ではないのですが、  
ルートを固定して、ルートから伸びている  
枝が動くようになっています。

ロボットを作る際は、ルートの位置に注意しましょう

# ロボットプログラミング 手先効果器の使い方



手先効果器の自由度は6  
位置の自由度 3  
回転の自由度 3

自由度が6未満の場合、任意  
の目標への解が存在しない

位置だけ、回転だけを合わせることできる