

# ロボットソフトウェア特論 (5-2)

2017.5.17

電気通信大学

大学院情報理工学研究科

末廣尚士

## 6. 座標系の連鎖

### ◆ 何をしたいか

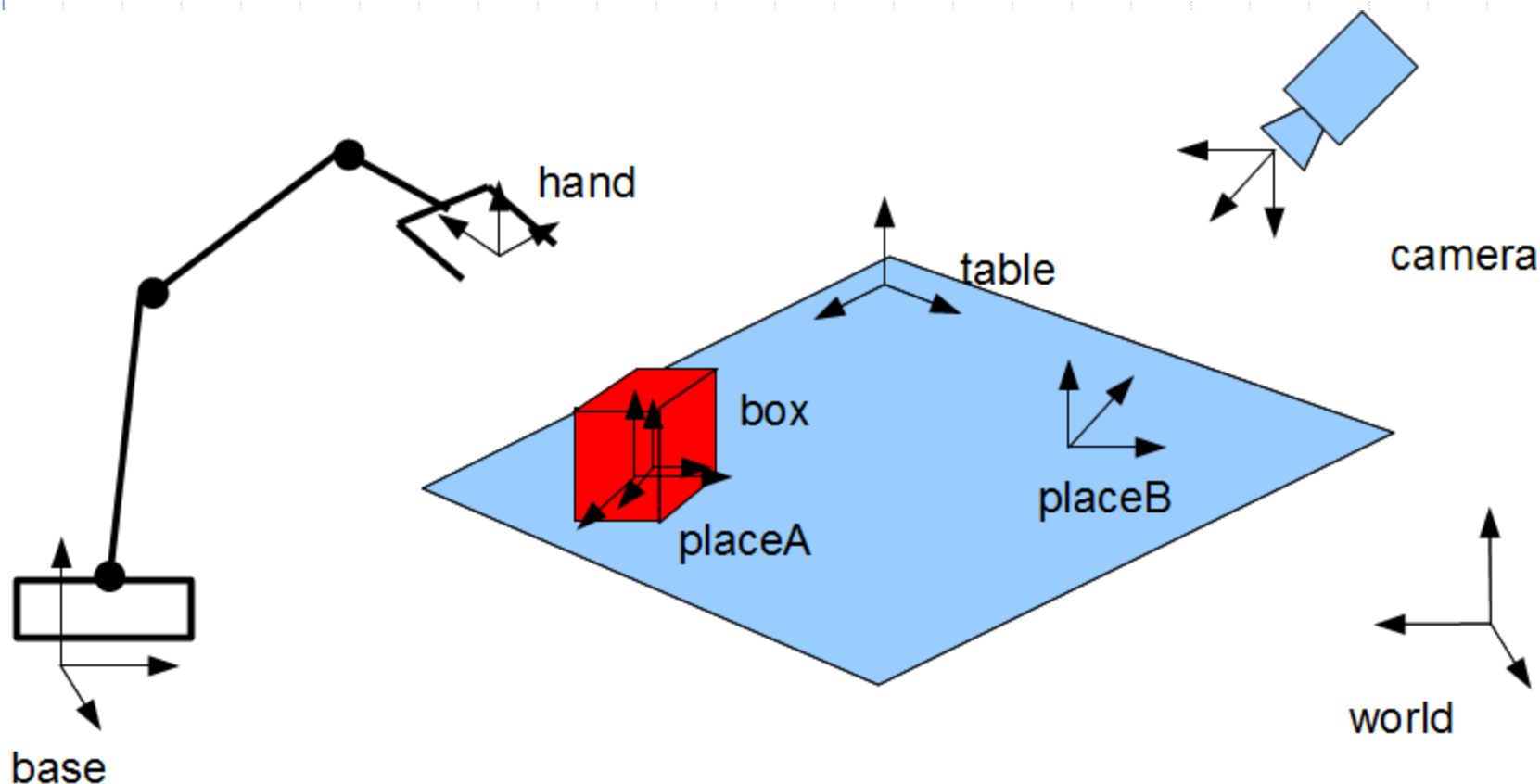
- 座標系を使って対象物の位置、姿勢を管理する。
- 対象物の属性
  - ◆ アプローチ点、把持点など
  - ◆ 形状、質量、慣性モーメント
- 物と物との関係

### ◆ 何のために

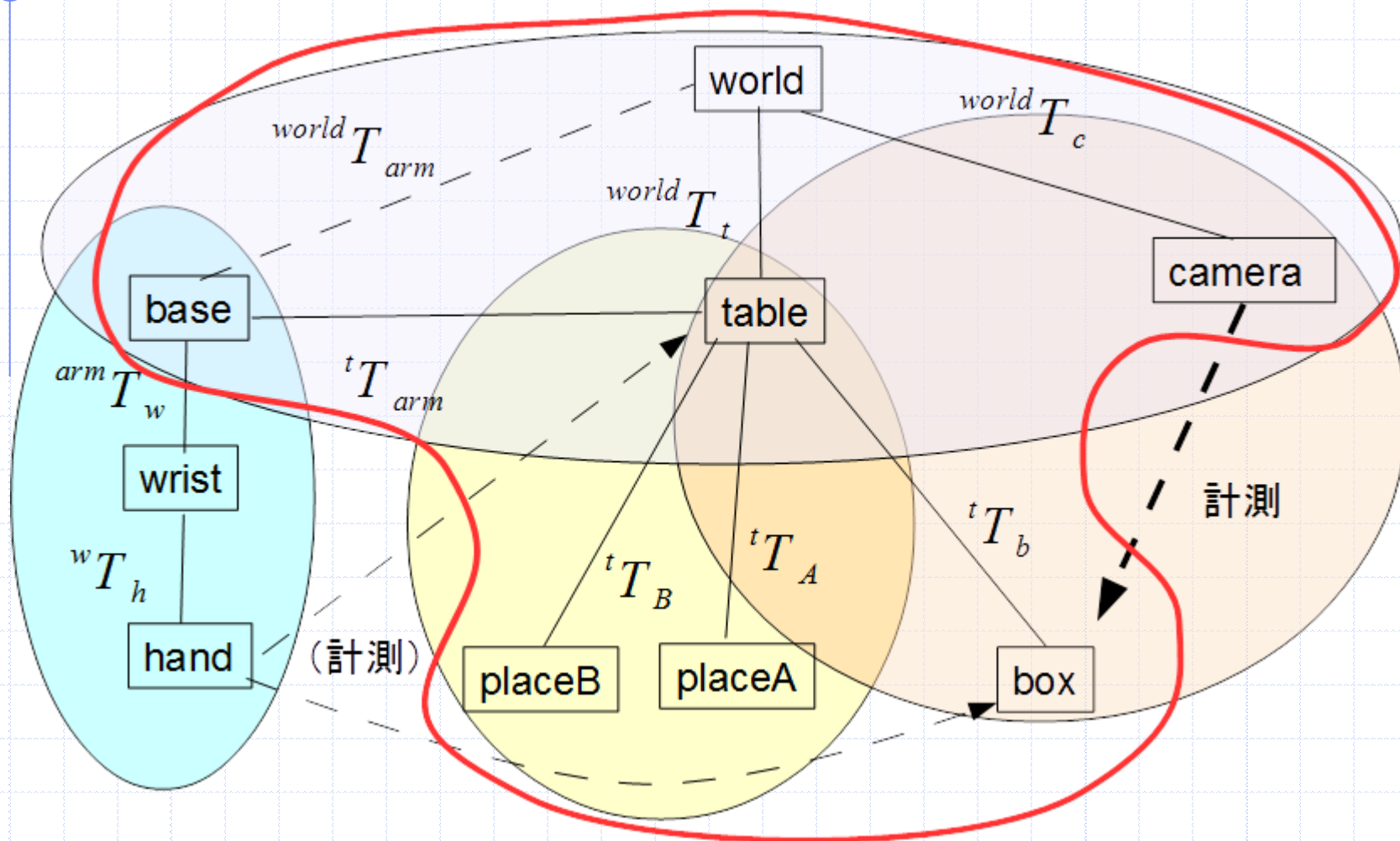
- 作業プログラムの記述
- オフラインプログラミング
- 複数腕での作業
- カメラ、移動台車などとの連携

# - 座標系を用いた表現

- ◆ table上のplace\_Aにあるboxをplace\_Bに動かす。



# - 座標系の関係の決めり方

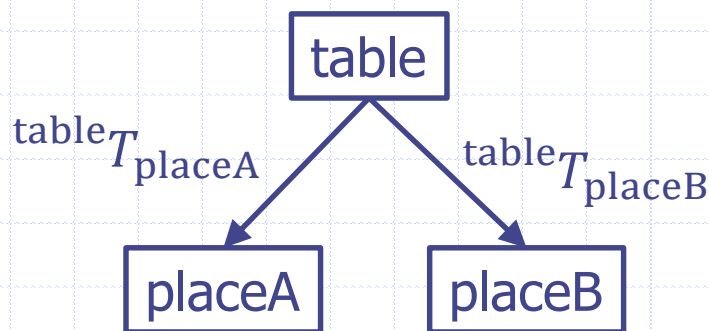


# – 座標系の連鎖の考え方

- ◆ 対象(に付けられた座標系)の位置・姿勢を「ある座標系」から見た座標変換(相対座標)で表す。
- ◆ たとえば、カップの取っ手：
  - その位置はカップにくっついていてカップを動かすとカップとともに動く。
- ◆ 部屋の中の椅子の位置：
  - 緯度、経度などで記述すると結局部屋のどこにあるか分かりにくい。
- ◆ そういうものはカップ座標系や部屋座標系の中でローカルに記述した方が良い。

# – 座標系の連鎖の表現

- ◆ たとえばplaceA, placeBはtableの上であり, tableが移動すると相対関係を保ったまま同時に移動する.
- ◆ これを座標系の連鎖で表現し, 座標の親子関係などと呼ぶ(この場合tableが親, placeA,placeBが子)
- ◆ tableからみたplaceAの変換 (placeA wrt (with respect to) table) は  ${}^{\text{table}}T_{\text{placeA}}$  で表現される.
- ◆ 図示すると,



## - 座標系の根っ子: ワールド座標系(1)

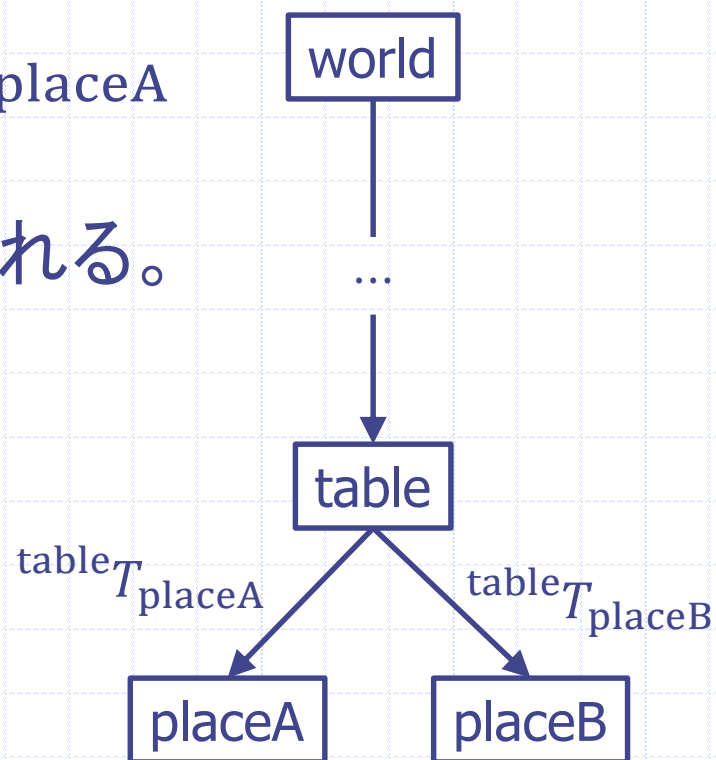
- ◆ 親の座標系も別な座標系から座標変換(相対座標)で表される。
- ◆ その連鎖は、最終的にはワールド座標系(\*)と呼ばれる座標系までつながっている。
  - \*絶対座標系、基準座標系などとも呼ばれる。
- ◆ 実際には「絶対座標系」はありえないので、適当なところで妥協して仮置きする。
- ◆ その心は、アーム、カメラ、ロボット本体といった特定のデバイスに依存しない座標系表現にしたいということ。

# – ワールド座標系からの位置姿勢

◆ ワールドからの位置・姿勢は,

$$\text{world} T_{xx} \cdots \text{xx} T_{yy} \text{yy} T_{\text{table}} \text{table} T_{\text{placeA}}$$

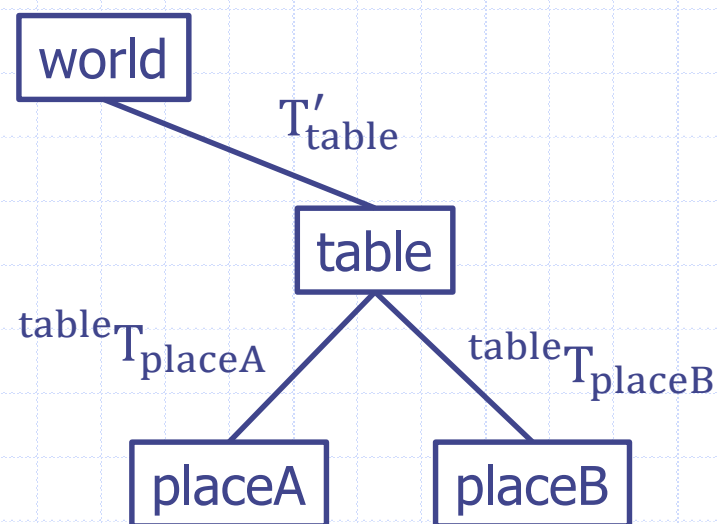
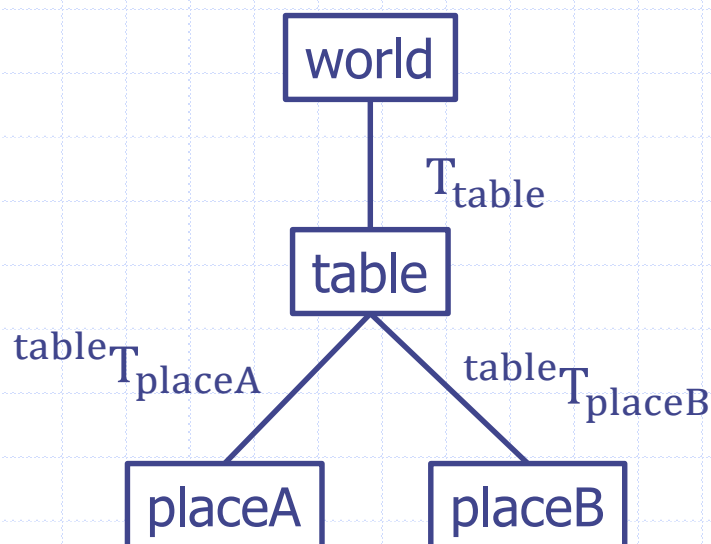
と座標変換の掛け算で計算される。





## - 座標系の連鎖の利点

- ◆ たとえばtableを動かしたときに、その上の場所や上に乗っている物はtableと一緒に動く(tableとの相対関係は変わらない)のでtableの座標変換だけ変更すればよい。



# – 座標変換の視点の指定

- ◆ ワールドや親からだけでなく、その他いろいろな座標系からの相対変換を知りたくなる/表現したくなることがある。
- ◆ どの座標系からの変換を考えているのかを  
wrt: with respect to と表現する。
- ◆ A wrt B とは  
Bから見たAつまり  ${}^B T_A$  のこと
  - Bが何でもよい変換行列の場合は省略する
  - Bがworldの場合にも省略することが多い

## – 座標変換と座標系の補足

- ◆ 座標変換行列自身は二つの座標系の相対関係を表している(z方向1メートルなど)だけで, 特定の座標系を表しているわけではない.
- ◆ 特定の座標系を表現するためには, どの座標系からの座標変換かを明示する必要がある.
- ◆ 「座標系Aからみた座標系B」というのを”B with respect to A”, 略して”B wrt A”と表現する.
- ◆ これを座標変換行列では,  ${}^A T_B$ と表す.

# - 座標系の連鎖の計算

◆ placeA wrt table :  ${}^{\text{table}}T_{\text{placeA}}$

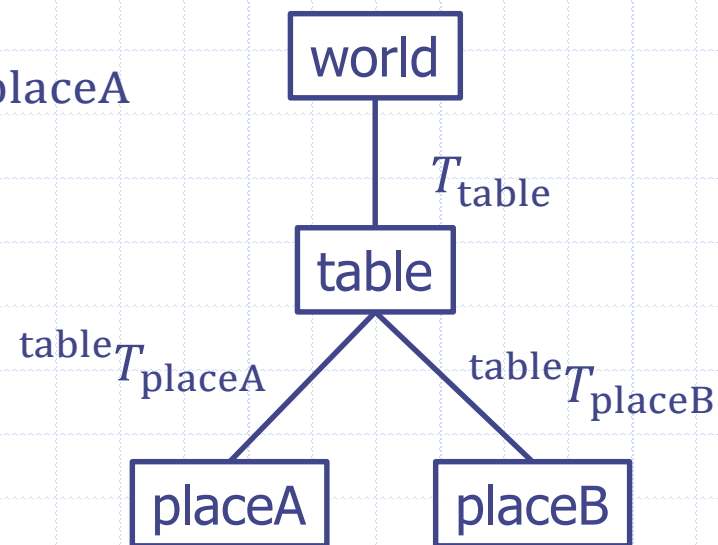
◆ placeA wrt world :  ${}^{\text{world}}T_{\text{placeA}} = {}^{\text{world}}T_{\text{table}} {}^{\text{table}}T_{\text{placeA}}$

◆ placeA wrt placeB :

$$\begin{aligned} {}^{\text{placeB}}T_{\text{placeA}} &= {}^{\text{placeB}}T_{\text{table}} {}^{\text{table}}T_{\text{placeA}} \\ &= \left( {}^{\text{table}}T_{\text{placeB}} \right)^{-1} {}^{\text{table}}T_{\text{placeA}} \end{aligned}$$

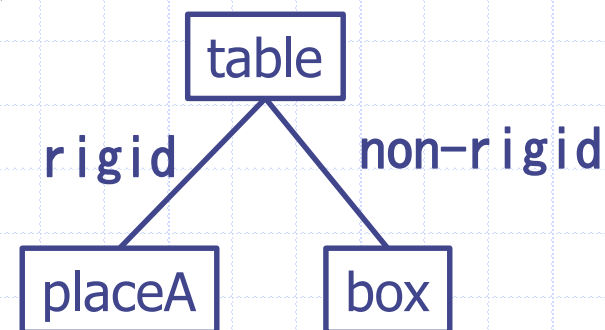
良く使う関係:

$${}^{\text{xx}}T_{\text{yy}} = \left( {}^{\text{yy}}T_{\text{xx}} \right)^{-1} = \left( {}^{\text{yy}}T_{\text{xx}} \right)^T$$



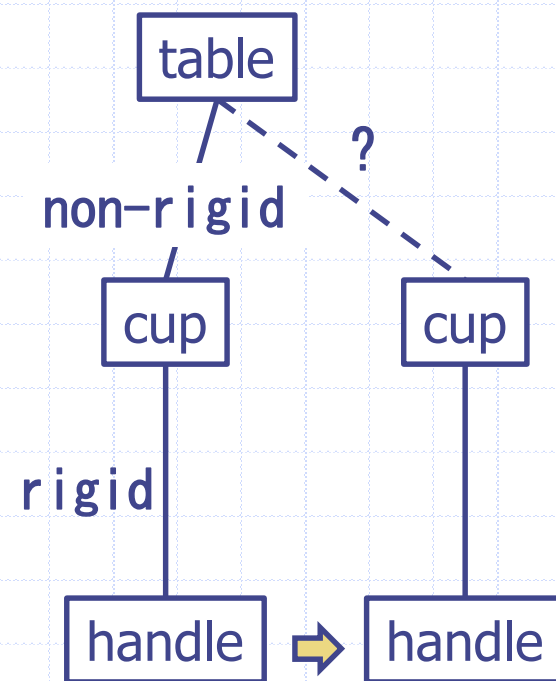
## – 座標系の親子関係の種類

- ◆ 親子関係には、tableの上のboxのように相対関係を簡単に動かせるもの(non-rigid)と場所を表すplaceAのように動かせないもの(rigid)がある。
- ◆ 動かせるものを動かすときは親との相対座標を変更する
- ◆ 動かないものを動かすときは、..
  - 親をしかるべく動かす
  - または動かせない(親を動かす)



# - 親をしかるべく動かすとは

- ◆ カップの取っ手を持ってカップを動かしたとき何をどう変更すればよいか.
- ◆ rigidでない関係までたどってそれを「正しく」動かす.
- ◆ たとえば、取っ手を  ${}^{\text{world}}T_{\text{handle}}$  にしようとする  
と、カップは、  ${}^{\text{world}}T_{\text{handle}} ({}^{\text{cup}}T_{\text{handle}})^{-1}$  になる.
- ◆ 一方、テーブルが  ${}^{\text{world}}T_{\text{table}}$  だとするとテーブルとカップの変換は、  
 $({}^{\text{world}}T_{\text{table}})^{-1} {}^{\text{world}}T_{\text{handle}} ({}^{\text{cup}}T_{\text{handle}})^{-1}$   
と変更されることになる。



## – 例題6-1:座標系の生成

- ◆ ワールド座標系から見て、テーブル面の位置が  $(1.0 \ 2.0 \ 0.7)^T$ ，姿勢はワールドと一致しているとしたときの座標変換  $T_t$ を作成せよ。
- ◆ テーブル面上で、位置  $(0.1 \ 0.2 \ 0)^T$ ，姿勢がz軸まわりに30度回転した座標系A，および位置  $(0.3 \ 0.3 \ 0)^T$ ，姿勢がz軸まわりに45度回転した座標系Bのテーブルからの座標変換  $T_{a,t}$ ， $T_{b,t}$ を作成せよ。

## – 例題6-1の解答例

```

>>> r2d=180.0/pi
>>> d2r=pi/180.0
>>> Tt=FRAME(xyzabc=[1,2,0.7,0,0,0])
>>> Ta_t=FRAME(xyzabc=[0.1,0.2,0,0,0,30*d2r])
>>> Ta_t
f:(m:[[0.86602540378443871, -0.4999999999999999994,
0.0], [0.4999999999999999994, 0.86602540378443871,
0.0], [0.0, 0.0, 1.0]],v:[0.10000000000000000001,
0.20000000000000000001, 0.0])
>>> Tb_t=FRAME(xyzabc=[0.3,0.3,0,0,0,45*d2r])
>>> Tb_t
f:(m:[[0.70710678118654757, -0.70710678118654746,
0.0], [0.70710678118654746, 0.70710678118654757,
0.0], [0.0, 0.0, 1.0]],v:[0.2999999999999999999,
0.2999999999999999999, 0.0])

```



## – 例題6-2:座標系連鎖の計算

- ◆  $T_t$ ,  $T_{a_t}$ ,  $T_{b_t}$ 用いて, 座標系Aおよび座標系Bのワールド座標系からの座標変換 $T_a$ ,  $T_b$ を求めよ.
- ◆  $T_a$ ,  $T_b$ を位置 $x$ ,  $y$ ,  $z$ および $\alpha$ ,  $\beta$ ,  $\gamma$ のオイラー角の6つ組みデータで表現せよ.
- ◆ ワールドからの姿勢の回転がそれぞれ $z$ 軸まわりに30度, 45度になっていることを確認せよ.

## – 例題6-2の解答例

```

>>> Ta=Tt*Ta_t
>>> Tb=Tt*Tb_t
>>> Ta.xyzabc()
[1.1000000000000001, 2.2000000000000002,
0.69999999999999996, 0.0, 0.0, 0.52359877559829882]
>>> Tb.xyzabc()
[1.3, 2.2999999999999998, 0.69999999999999996, 0.0,
0.0, 0.78539816339744828]
>>>
>>> Ta.xyzabc()[5]*r2d
29.999999999999996
>>> Tb.xyzabc()[5]*r2d
45.0

```

## – 例題6-3:親座標系の移動

- ◆ ワールド座標系から見て、テーブル面の位置は  $(1.0 \quad 2.0 \quad 0.7)^T$  で、姿勢がz軸回りに90度回転した座標変換を新たに  $T_t$  として作成せよ.
- ◆ 変更後の座標系Aおよび座標系Bのワールド座標系からの座標変換  $T_a$ ,  $T_b$  を求めよ.
- ◆  $T_a$ ,  $T_b$  を位置  $x$ ,  $y$ ,  $z$  および  $\alpha$ ,  $\beta$ ,  $\gamma$  のオイラー角の6つ組みデータで表現せよ.
- ◆  $T_a$ ,  $T_b$  のワールドからの姿勢の回転がどのようになっているか確認せよ.

## – 例題6-3の解答例

```

>>> Tt=FRAME(xyzabc=[1,2,0.7,0,0,90*d2r])
>>> Ta=Tt*Ta_t
>>> Tb=Tt*Tb_t
>>> Ta.xyzabc()
[0.80000000000000000004, 2.100000000000000001,
0.6999999999999999996, 0.0, 0.0, 2.0943951023931953]
>>> Ta.xyzabc()[5]*r2d
119.99999999999999
>>> Tb.xyzabc()
[0.6999999999999999996, 2.299999999999999998,
0.6999999999999999996, 0.0, 0.0, 2.3561944901923448]
>>> Tb.xyzabc()[5]*r2d
135.0

```

## – 例題6-4:親をしかるべく動かす

- ◆ 最初カップのテーブル上の座標 $T_{c\_t}$ が座標系Aと一致しているとする.
- ◆ カップから見て取っ手の座標が位置は $(0.03 \quad 0.0 \quad 0.04)^T$ で姿勢はカップと一致しているとして、その座標変換 $T_{h\_c}$ を作成せよ.
- ◆ 取っ手を座標系Bの上0.04としたときのワールドからの座標変換 $T_h$ を求めよ.
- ◆ そうなるためには、テーブルから見たカップの座標系 $T_{c\_t2}$ をどのようにしなくてはならないか計算せよ.

## – 例題6-4の解答例

```

>>> Tc_t=Ta_t
>>> Th_c=FRAME(xyzabc=[0.03,0.0,0.04,0,0,0])
>>> Th=Tb*FRAME(xyzabc=[0,0,0.04,0,0,0])
>>> Tc_t2=(-Tt)*Th*(-Th_c)
>>> Tc_t2.xyzabc()
[0.27878679656440342, 0.27878679656440364,
3.4694469519536142e-017, 0.0, 0.0, 0.78539816339744828]

```

## – 例題6-4の確認

```

>>> tmp=Tt*Tc_t2*Tch
>>> tmp.xyzabc()
[0.69999999999999999996, 2.29999999999999999998,
0.73999999999999999999, 0.0, 0.0, 2.3561944901923448]
>>> Th.xyzabc()
[0.69999999999999999996, 2.29999999999999999998,
0.73999999999999999999, 0.0, 0.0, 2.3561944901923448]
>>>

```

# – 次回予告

- ◆ シリアルリンク ロボットアーム
- ◆ ロボットアームのモジュール
  
- ◆ ホームページから,  
`object_model_v.py`, `larm_w_hand.py`,  
`hand.py`, `arm3dof.py`, `arm6dof.py`  
 をダウンロードしておく.