

ロボットソフトウェア特論 (1-2)

2017.4.12

電気通信大学

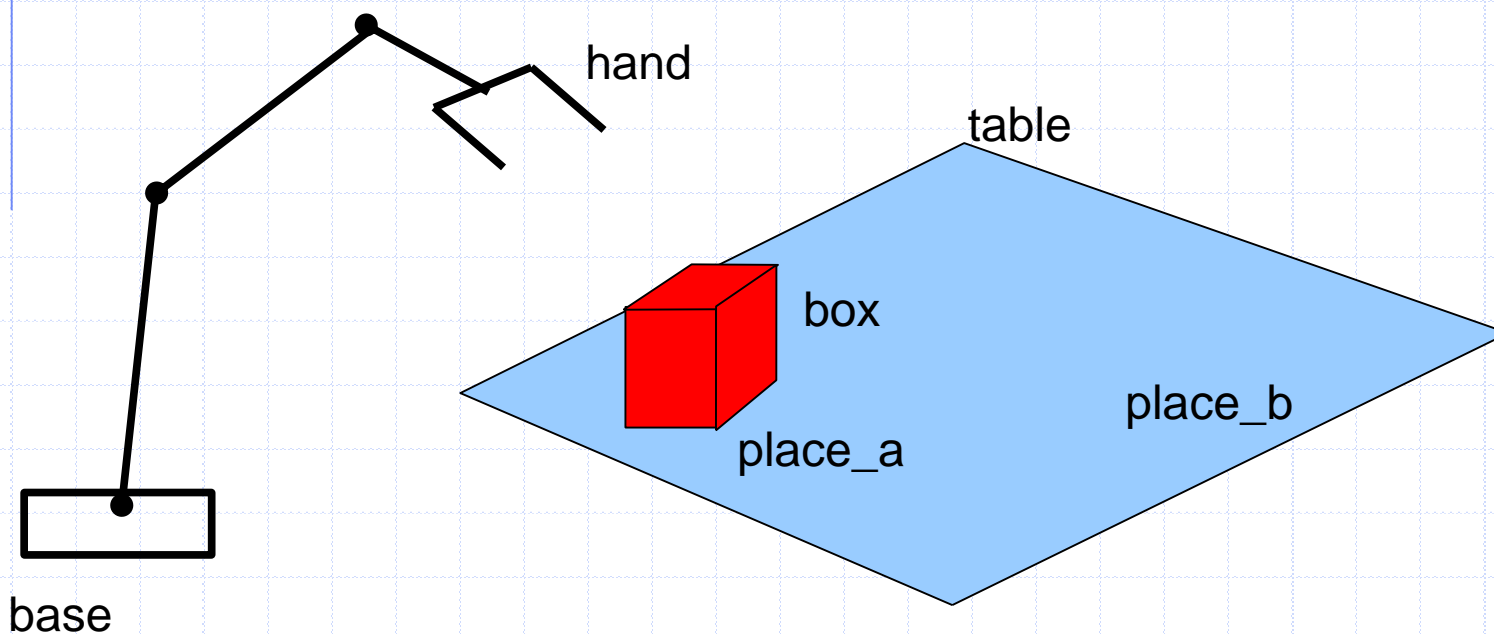
大学院情報理工学研究科

末廣尚士

1. ロボット作業の表現

– ロボットの作業の例

◆ table上のplace_aにあるboxをplace_bに動かす。

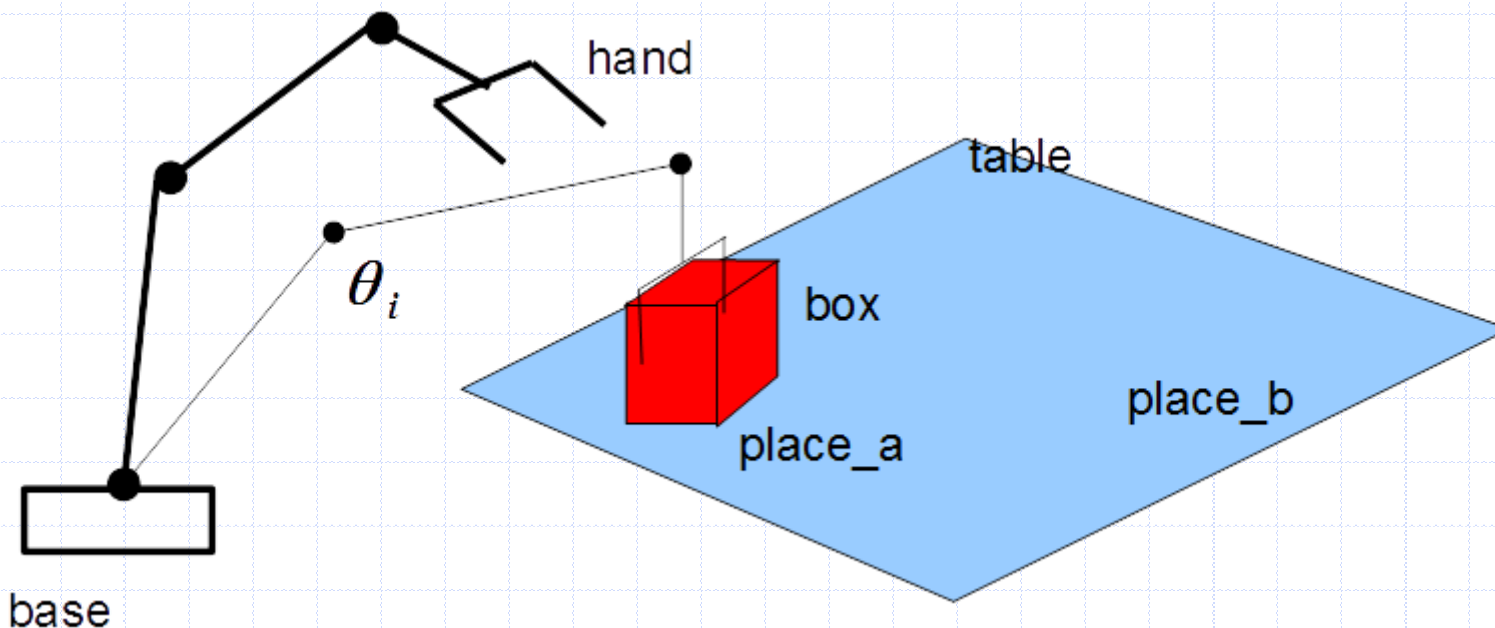


– ロボットの動作制御の方法

- ◆ ティーチング・プレイバック(連続動作)
 - ダイレクト教示
 - 塗装ロボット
- ◆ ティーチング・プレイバック(点列教示)
 - リモート教示。
 - point-to-point動作
 - 溶接ロボット
- ◆ プログラム制御
 - ティーチング・プレイバックの修飾
 - CADデータの利用、オフライン教示、センサ利用
=> ロボット用言語

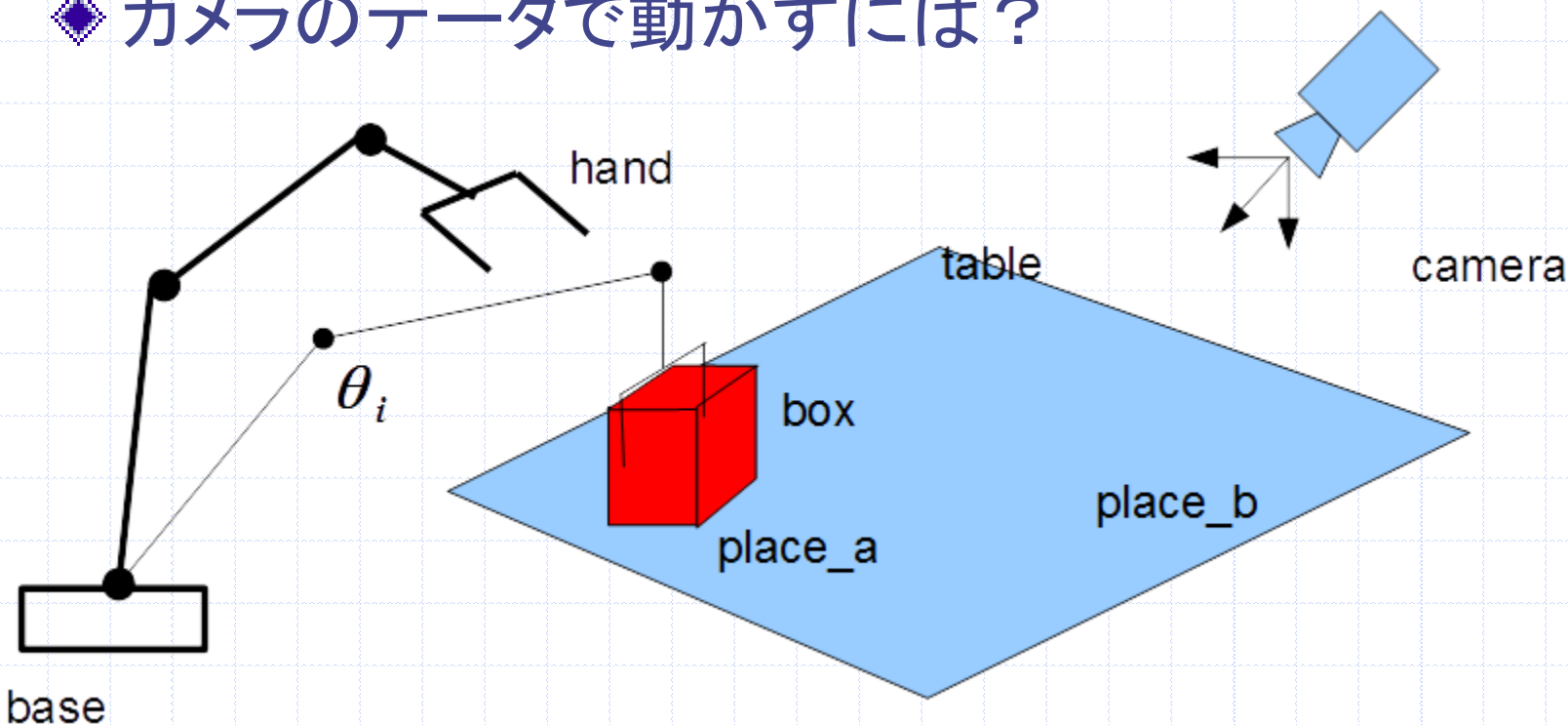
– 関節角度での作業例

- ◆ 要所の関節角を覚えておいてそれを再生する



- 関節角度での作業の問題点

- ◆ ロボットが少しずれたら？
- ◆ 違うロボットだったら？
- ◆ カメラのデータで動かすには？



– ロボット用言語

◆ 動作レベル言語

- ロボットの動作を記述
- VAL, AL

◆ 対象レベル言語

- 操作対象に関する記述
- AUTOPASS、LAMA

◆ 作業レベル言語

- 作業の記述
- 動作するにはプログラミングが必要
- AI研究。積木の世界

– ロボット用言語に要求される機能

- ◆ 動作の記述
- ◆ データ・タイプ
- ◆ センサの利用
- ◆ 作業の同期
- ◆ 環境データの管理
- +
- ◆ 一般のコンピュータ言語と同様な機能
 - 数値演算、フロー制御、、、

- VALのプログラム例

```

1.      SETI N. PRATS=0
2. 100  VPICTURE
3.      VLOCATE PART, 100
4.      APPRO CAMERA:PART:PICK. UP, 50.
5.      MOVES CAMERA:PART:PICK. UP
6.      GRASP 25.
7.      DEPART 50.
8.      APPRO PALLET, 50
9.      MOVES PALLET
10.     OPENI
11.     DEPART 100.
12.     SHIFT PALLET BY 5. 2, 25. 4, 0
13.     SETI N. PARTS=N. PARTS+1
14.     IF N. PARTS NE 10 THEN 100
15.     STOP

```

ループ変数N. PRATSの初期化
 カメラ画像の読み込み
 対象があればPARTに位置が入る
 つかみ位置の上50mmへ移動
 つかみ位置へ移動
 つかむ。指幅が25mm以下なら異常終了
 50mm上へ移動
 PALLETの上50mmへ移動
 PALLETの位置へ移動
 指を開く
 100mm上へ移動
 PALLET位置を(5. 2, 25. 4, 0)シフト
 ループ変数N. PARTSに1加える
 N. PARTSが10以下なら100へ戻る
 停止

- ALのプログラム例

```

BEGIN
  EVENT passed, caught, read_pass;
  FRAME steel_beam, pass, catch;

  COBEGIN
    BEGIN "blue"
      MOVE barm TO steel_beam;
      CENTER barm;
      AFFIX steel_beam TO barm;
      MOVE steel_beam TO pass;
      SIGNAL ready_pass;
      WAIT caught;
      OPEN bhand TO 3.0*inches;
      UNFIX steel_beam FROM barm;
      SIGNAL passed;
    END "blue"
  
```

続き

```

    BEGIN "yellow"
      OPEN yhand TO 3.0*inches;
      MOVE yarm TO catch;
      WAIT ready_pass;
      CENTER yarm;
      SIGNAL caught;
      WAIT passed;
      MOVE yarm TO pallet;
    END "yellow"
  COEND;
END

```

– ロボット作業の座標系による表現

◆ 何をしたいか

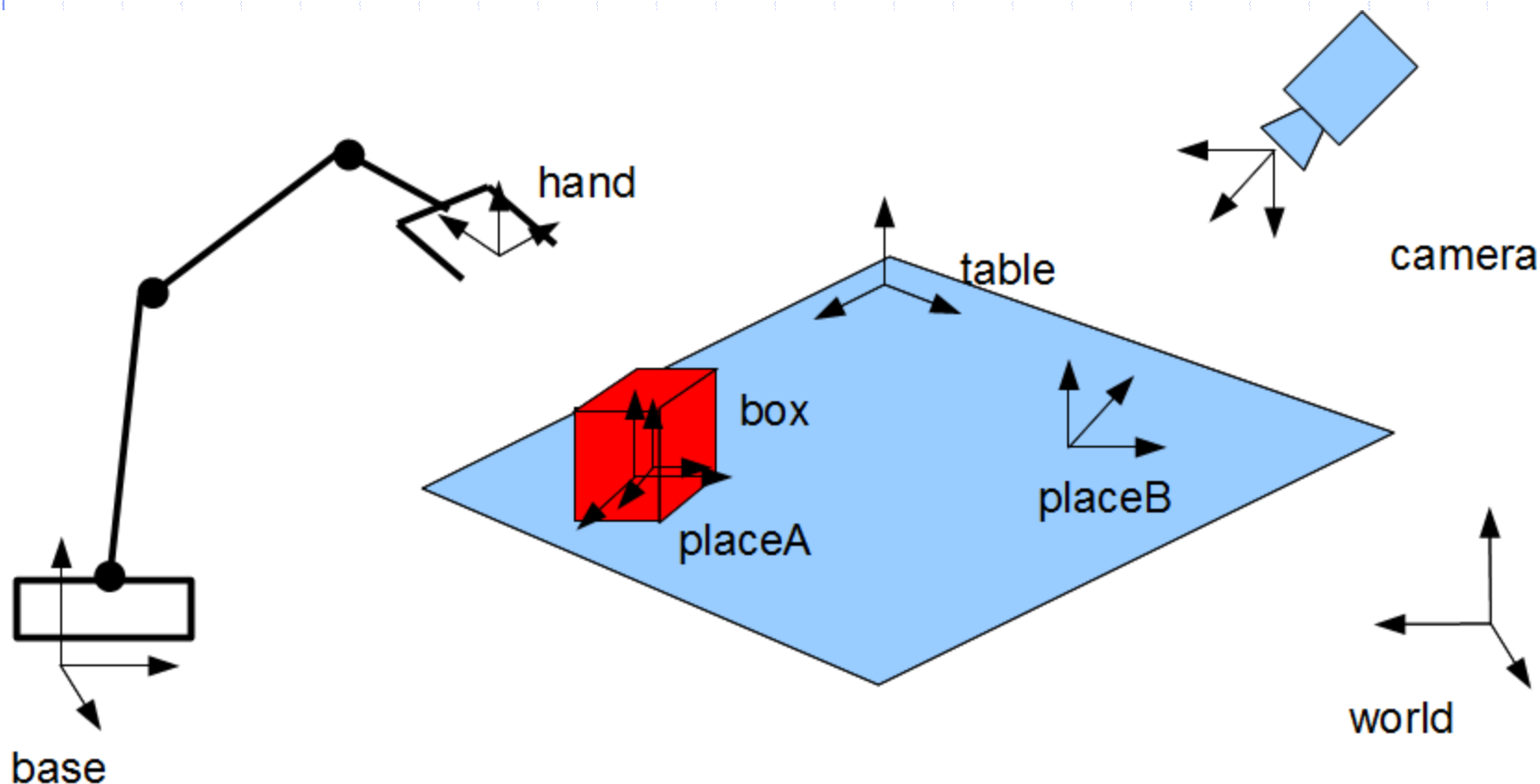
- ロボットの働く環境やロボットの動作を座標系および座標系の連鎖を用いて表す。

◆ 何のために

- ロボットが働く環境を統一的に表現する
- カメラや他のセンサと連携させる。
- ロボットが行う作業をロボットの構造によらずに記述する。
- 局所的な表現と全体的な表現を統一的につなげる。(座標系の連鎖)

– 座標系を用いた表現

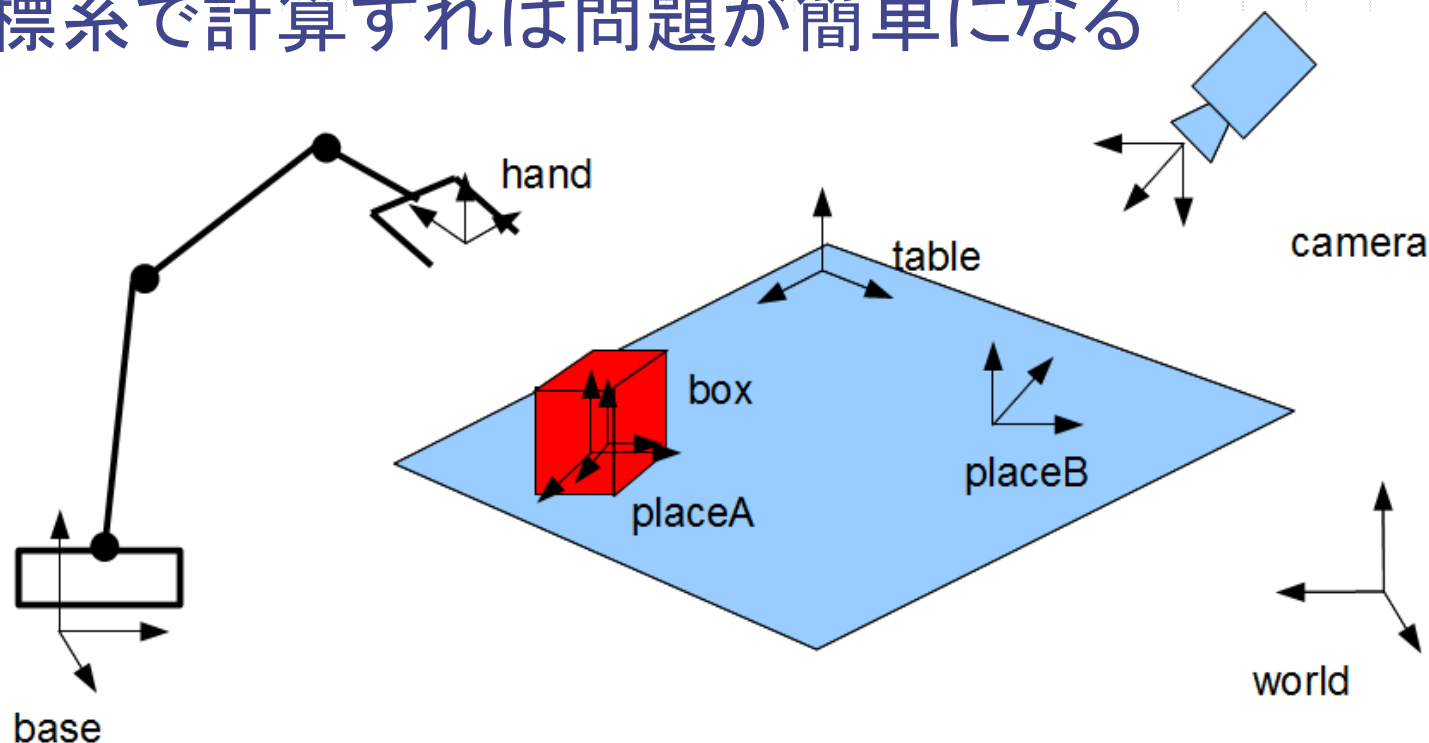
◆ table上のplace_Aにあるboxをplace_Bに動かす。



- 座標系を用いた表現

- ロボットが少しずれたら？
- 違うロボットだったら？
- カメラのデータで動かすには？

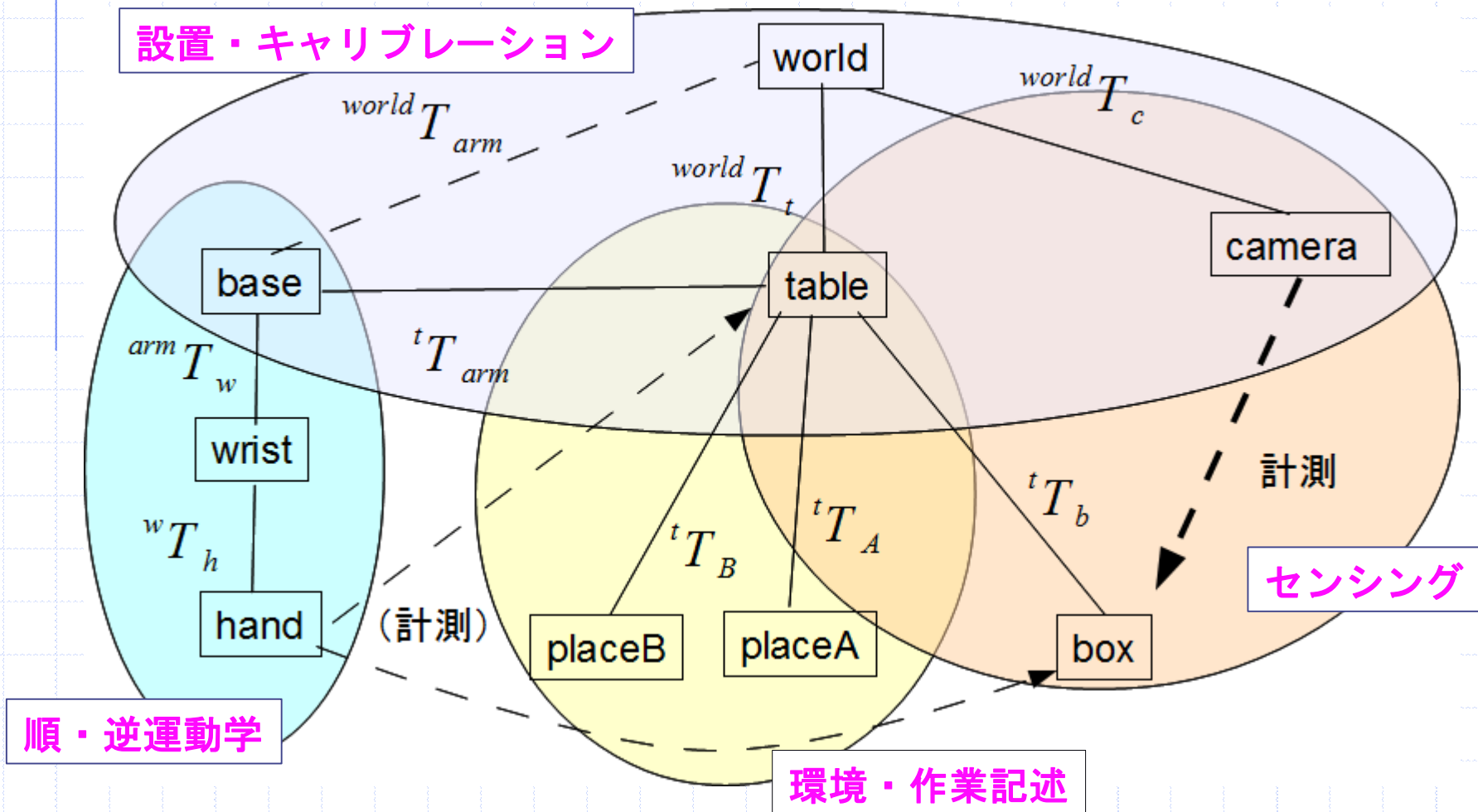
◆ 座標系で計算すれば問題が簡単になる



– 座標系の表現

- ◆ 3次元直交座標系は原点の位置と姿勢とで決まる。どう表現するか？
 - 位置: ある座標系で表した原点の位置: x, y, z
 - 姿勢:
 - ◆ 座標の回転: roll, pitch, yawなど
 - ◆ 各座標軸のセット: 座標回転行列
 - 位置と座標回転行列をセットにした4x4同次変換行列
- ◆ 最終的に基準となる3次元直交座標系: 絶対座標系

座標系の関係の決めり方



一次回予告

◆ python 概要