

# ロボットソフトウェア特論

## 課題5(最終課題)

2017.7.26

電気通信大学

大学院情報理工学研究科

末廣尚士

# 最終課題 提出方法

## ◆ 提出物

(1) プログラムファイル(提出した自分の環境で動くこと)

- ◆ report5\_学籍番号.py
- ◆ task\_final\_学籍番号.py

(2) pdfファイル:

- ◆ report5\_学籍番号.pdf

(3) 前の課題から変更, その他独自ファイル

- ◆ arm\_学籍番号.py, hand\_学籍番号 など

◆ 提出先: [suehiro@is.uec.ac.jp](mailto:suehiro@is.uec.ac.jp)

◆ 表題:「ロボットソフトウェア特論 課題5」

◆ 〆切: 8/9(水)

# 最終課題で必要なプログラム

- geo.py
- object\_model\_v.py
- parts.py
- larm\_w\_hand.py
- hand.py (自作ハンドでもOK)
- face\_fit.py
  
- task\_final\_base.py
- task\_final\_tmpl.py (提出ファイルのテンプレート)
- face\_points\_final.py
  
- arm\_学籍番号.py

# 問題1 二点間の緯度, 経度 => 距離(角度)

(1-1) 北緯を正, 南緯を負の角度, 同様に東経を正, 西経を負の角度とし, 地球の大圏の長さを40,000[km]として, 以下の関数を作成せよ

- ・ 関数名: lalo2km
- ・ 引数: 二つの緯度経度のリスト, [la1, lo1], [la2, lo2]
- ・ 返り値: 上記, 二地点間の大圏距離(km)

(この関数はreport5\_学籍番号.pyにまとめる)

# 問題1 二点間の緯度, 経度 => 距離(角度)

(1-2) その関数を用いて, 下記2地点間の大圏距離を求めよ

- 電通大(北緯35.6561, 東経139.5439)
- スカイツリー(北緯35.7101, 東経139.8107)

(1-3) 同様に, 電通大とホノルル空港(北緯21.3479, 西経157.8974)の大圏距離を求めよ

(これらは実行結果を含めてreport5\_学籍番号.pdfにまとめる)

# 問題1 ヒント

1. 地球中心から以下の二つの地点への単位ベクトルを求める.
2. 上記, 二つのベクトルの間の角度を求める.
3. 大圏距離は, それに半径を掛ければ求められる.

◆ emacs で “M-x shell” とやると, emacs の中で shell が走るのので, そこで python を実行すると実行結果を簡単にテキストファイルにできる.

## 問題2 作業 & 計測対象モデル作成

(2-1)  $0.3 \times 0.24 \times 0.04$  の直方体(stand)を作成せよ.

- 座標系の座標系は長さ0.3の辺の方向をx軸,
- 0.2の辺の方向をy軸とし,
- xy平面を底面とし, 底面の中心を原点とする.

(2-2) standの上面中央にstand.top座標系を設定せよ

(2-3) place\_c(task\_final\_base.pyで生成済み)の位置にstandを配置せよ

(この部分はtask\_final\_学籍番号.pyにまとめる)

# 問題2のヒント

vpythonの画面,

◆ table上にstandが配置され,

◆ stand上面にstand.top座標系が表示されること,  
を確認する.



## 問題3 standの位置・姿勢計測

standをdepthカメラ(camera)で計測した結果がface\_points\_final.pyに入っている。

注: 計測はstand形状の中心. standの代表座標が底面であることに注意すること

(3-1) cameraからみたstand形状の座標変換を求めよ

(3-2) 上記の結果からtableからみたstandの座標変換を求めよ

(3-3) 計測結果を用いてstandを置きなおせ

(この部分はtask\_final\_学籍番号.pyにまとめる)

## 問題3のヒント

- ◆ 基本的に「ロボットソフトウェア特論(14)」(7/19)の `face_fit_box.py` を利用すれば良い
- ◆ ただし、位置計測の際に直方体の原点を中心に置いたことと、環境モデルでの `stand` の原点が底面にあることの違いを考慮する必要がある
- ◆ `stand` を置きなおすときは、まず `unfix` することを忘れないように

## 問題4 作業プログラム

自身の作成したアームを用いて以下の作業プログラムを作成し、実行せよ。

作業プログラム `do_task()`:

- ◆ `cup_b`から`cup_a`にそそぐ
- ◆ `cup_a`を`saucer`の上に載せる
- ◆ `saucer`を持って`cup_a`とともに`stand.top`に置く

(この部分は`task_final_学籍番号.py`にまとめる)

## 問題4のヒント

- ◆ task\_final\_tmpl.pyを書き換える.
- ◆ 自身のアーム(arm\_学籍番号.py)を読み込む.
- ◆ 作業実行中に不具合を発見したら, place\_a, place\_b, place\_cやアームの設置位置を適切に調整する.  
必要ならばpourその他の設定も変更して良い.
- ◆ またアームやハンドの改造を行ってもよい.
- ◆ 変更点についてはpdfファイルに記述すること
- ◆ 提出済みのプログラムを変更した場合は併せて, 再提出すること

# 問題4 直したい不具合の例

