解答例

第 1 章

チャレンジ 1.1

1.3.4 節に従って作業を行うだけなので，特に答えはありません．

第 2 章

チャレンジ 2.1

`ros2 サブコマンド -h` のサブコマンドに `action`，`bag`，`launch`，`node`，`pkg`，`run`，`service`，`topic` を 1 つずつ入れて実行してください．

チャレンジ 2.2

ハンズオンと同様に実行してください．

チャレンジ 2.3

端末を開いて 3 分割してください．

1. 上段の端末で `turtlesim_node` を起動してください．

```
ros2 run turtlesim turtlesim_node
```

   中段の端末で `turtle_teleop_key` ノードを起動してください．

```
ros2 run turtlesim turtle_teleop_key
```

2. ros2 bag record コマンドで全トピックの記録
   下段の端末で次のコマンドを実行してください．

```
ros2 bag record -a -o bagfile
```

3. キー操作によりタートルの移動
   中段の端末で矢印キー（↑, ↓, ←, →）を使いタートルを動かしてください．
4. 記録の停止
   下段の端末で Ctrl+C キーを押してください．
5. 全ノードの終了
   上段と中段の端末で Ctrl+C キーを押してください．
6. `turtlesim_node` ノードの起動
   上段の端末で次のコマンドを実行してください．

```
ros2 run turtlesim turtlesim_node
```

7. 保存したバッグファイルの再生

上段の端末で次のコマンドを実行してください．turtle がスイスイ泳ぎ出します．

```
ros2 bag play bagfile
```

チャレンジ 2.4

本文で説明したとおりです．

チャレンジ 2.5

```
cd ~/happy_ws
codium .
```

~/happy_ws/src/chapter2/my_happy/my_happy/setup.py を開き，プログラムリスト 2.6 の内容に書き換えて保存します．~/happy_ws/src/chapter2/my_happy/my_happy/happy_node2.py を開き，プログラムリスト 2.5 の内容に書き換えて保存します．次のコマンドを実行すると結果を確認できます．

```
colcon build
ros2 run my_happy happy_node2
```

チャレンジ 2.6

サポートサイトを参照してください．

- https://github.com/AI-Robot-Book-Humble/answers/tree/main/chapter2

チャレンジ 2.7

まず，コピーしてソースファイル time_subscriber_node.py を作ります．

```
cd ~/happy_ws
cp ~/airobot_ws/src/chapter2/happy_topic/happy_topic/happy_subscriber_node.py
   ~/happy_ws/src/chapter2/time_topic/time_topic/time_subscriber_node.py
codium .
```

1. **ソースコードの作成：**time_subscriber_node.py **をエディタで開き，9 行目のノード名を** time_subscriber_node **に書き換えて保存する．**
2. **package.xml の編集：前のチャレンジで編集したので，ここではしない．**
3. **setup.py ファイルの編集：24 行目を次のように書き換えて保存する．**

   'time_subscriber_node = time_topic.time_subscriber_node:main',

端末を 2 分割し，上の端末でサブスクライバノードを起動して，メッセージを待ちます．

```
cd ~/happy_ws
source install/setup.bash
ros2 run time_topic time_subscriber_node
```

下の端末でパブリッシャノードを起動して，図 2.18 のように表示されているかを確認してください．

```
cd ~/happy_ws
source install/setup.bash
ros2 run time_topic time_publisher_node
```

チャレンジ 2.8

この本の説明に従って作業を行うだけなので，特に答えはありません．

チャレンジ 2.9

サポートサイトを参照してください．

- https://github.com/AI-Robot-Book-Humble/answers/tree/main/chapter2

チャレンジ 2.10

この本の説明に従って作業を行うだけなので，特に答えはありません．

チャレンジ 2.11

サポートサイトの次のパッケージが実装例です．参考にしてください．

- https://github.com/AI-Robot-Book-Humble/chapter2/tree/main/happy_service

第 3 章

【クイズ 3.1】

1. 音声認識モデルを構成する 3 つの要素は，音素モデル，単語辞書，言語モデルです．音素モデルは，音声特徴量と音素列の間の確率を計算するモデルです．音素とは，母音や子音で構成される音の最小要素です．単語辞書とは，音素列と単語の対応を扱うモデルです．具体的には，単語とその音素列が記述されたリストです．言語モデルは，ある単語に対して，その単語が発話される確率を計算するモデルです．「僕の」の後には「トップ」よりも「コップ」のほうがきやすいといった確率を計算します．
2. MFCC は，メル周波数に基づいて得られた対数メルフィルタバンク特徴量に対して，離散コサイン変換を適用して得られた成分です．メル周波数とは，人間の聴覚における音の聞こえ方に基づいたメル尺度により設計された周波数スケールであり，低周波数域は分解能が高く，高周波数域は分解能が低い対数スケールになっています．
3. 音声認識の問題は，以下のように定式化できます．

$$\hat{w} = \mathrm{argmax}_w P(w|x)$$

ここで，x は音声認識モデルの入力となる音声特徴量，w は出力となるテキスト候補です．$P(w|x)$ は，音声特徴量 x が与えられた下でのテキスト候補 w の条件付き確率であり，argmax_w はこの確率を最大にする w を返すことを意味しています．つまり，音声認識の問題は，音声特徴量が与えられた下でのテキスト候補の条件付き確率をすべて計算し，この確率を最大化するテキスト候補を出力する関数として定式化できます．

4. RNN では，遠い過去のフレームの情報が勾配消失によって学習に影響しなくなる問題がありました．この問題に対して LSTM は，前の隠れ層の値に対する勾配を常に 1 にするメモリセルを導入することで長期間の情報を扱えるようにしました．

【クイズ 3.2】

1. 統計的音声合成は，言語特徴量，音響特徴量，音声波形の 3 つの予測問題に分けられます．1 つ目は，与えられたテキストから音節や語，句におけるアクセントの情報を持つ言語特徴量を予測する問題です．2 つ目は，予測された言語特徴量から話者の特性に依存する音声の特徴を表す音響特徴量を予測する問題です．3 つ目は，予測された音響特徴量を基本周波数，メルケプストラムなどの音声パラメータに分離し，ボコーダを用いて音声波形を予測する問題です．
2. ニューラルネットワークでは，系列長が異なる特徴量間の変換を行うのは困難です．そこで必要になるのが，音素単位の言語特徴量を入力として音素継続長を予測する継続長モデルと，フレーム単位の言語特徴量から音響特徴量を予測する音響モデルです．
3. テキストと言語特徴量のみでは，話者の特性を考慮した音声合成はできません．同じ語や句であっても，男性と女性，大人と子ども，話者によって音声は異なります．このような話者の特性に依存する音声の物理的な特徴を表すパラメータが音響特徴量です．

チャレンジ 3.1

チャレンジの解答例はプログラムなのでここに掲載できません．サポートサイトの解答例を参照してください．

- `https://github.com/AI-Robot-Book-Humble/answers/tree/main/chapter3`

第 4 章

チャレンジ 4.1〜4.16

第 4 章の全チャレンジの解答例はプログラムなのでここに掲載できません．サポートサイトの解答例を参照してください．

- `https://github.com/AI-Robot-Book-Humble/answers/tree/main/chapter4`

【クイズ 4.1】

式 (4.18) と式 (4.19) を連立して姿勢角 θ を消して式を整理すると，円の方程式が導出され，ロボットの位置 x, y はランドマーク 1, 2 を通る円の円周上になります．同様に式 (4.18) と式 (4.20)，式 (4.19) と式 (4.20) から円の方程式が 2 つ導出され，距離を使う方法（113 ページ）と同じ計算手法で自己位置を求めることができます．

【クイズ 4.2】

LiDAR の正確な取り付け高さは地図だけからはわかりませんが，テーブルの脚と思われる障害物があるので天板より低い位置にあると思われます[注1]．なお，地図ではテーブルの天板がわかりませんの

注 1　比較するために house.pgm を UBUNTU のアプリ「ファイル」から開いてください．ダブルクリックすると開き，マウスのホイールで拡大縮小します．

で，LiDAR センサしか搭載していない場合は，天板より背の高いロボットはテーブルに衝突する可能性があります．それを避けるためには，この house.pgm ファイルを gimp[注2] などのお絵描きソフトで修正する必要があります．多くの場合，SLAM で作成した地図をそのまま使わず，地図にない障害物を追加したり，地図上のノイズなどを消したり，ロボットに行ってほしくない領域は不明領域などに設定しています．なお，家のサイズはピクセル数を数えればわかるので省略します[注3]．

【クイズ 4.3】

このシミュレーションは，地面は完全な平面で，車輪のスリップもなく，ロボットの重心も台車の中央にある理想的な環境です．リアルワールドでは地面は凸凹していて，車輪もスリップしますし，ロボットの重心も多少偏っています．特に，地面が凸凹している影響は大きく，これによりロボットの向きの推定に誤差が出ます．この誤差は時間が進むにつれて雪だるま式に大きくなるのです．そのため，つくばチャレンジに参加している多くのチームは，デッドレコニングの計算でロボットの向きに関しては地磁気センサやジャイロセンサなどの姿勢センサからの値をそのまま使っています．

第 5 章

【クイズ 5.1】

1. **a.** `haarcascade_fullbody.xml`, **b.** `haarcascade_smile.xml`, **c.** `haarcascade_upperbody.xml`
2. `haarcascade_smile.xml` を使って，`face_detection.py` の目検出部分を置き換えることで，笑顔検出のアプリケーションを開発できます．笑顔よりも口元を検出するほうが，検出性能は高くなります．アプリケーションの開発にあたっては，サポートサイトの `quiz5_1_smile_detection.py` を参照してください．`https://github.com/AI-Robot-Book-Humble/answers/blob/main/chapter5/quiz5_1_smile_detection.py`

チャレンジ 5.1

サンプルプログラム `face_detection.py` をもとに，カメラに人の顔が写っているかどうかを判断します．顔が検出されたら，第 3 章の音声合成の `speech_synthesis_server` と `speech_synthesis_client` を使って，あいさつができます．アプリケーションの開発については，サポートサイトの `challenge5_1_say_hello.py` を参照してください．`https://github.com/AI-Robot-Book-Humble/answers/blob/main/chapter5/challenge5_1_say_hello.py`

チャレンジ 5.2

サンプルプログラム `aruco_node_tf.py` をもとに，コップ上の aruco マーカの tf 情報を取得できます．この tf 情報からコップの向きを判断し，端末上でテキストで表現したり，第 3 章の音声合成の `speech_synthesis_server` と `speech_synthesis_client` を使って音声で発話したりすることができます．アプリケーションの開発については，サポートサイトの `challenge5_2_cup_direction.py` を参照してください．`https://github.com/AI-Robot-Book-Humble/answers/`

注 2　Linux ではよく使われているオープンソースのお絵描きソフト．プアマンズ・フォトショップといわれるほど機能は高いです．

注 3　このような大きな家に住みたいものです．なお，日本の住環境では背の高い生活支援ロボットは使えないでしょう．昔，実家に Pepper があったとき，部屋間の敷居を乗り越えることができず，手を広げてダンスなどされたときは，テーブルのコップなどにぶつかって大変なことになりました．お掃除ロボットサイズが 1 つの答えかもしれません．

blob/main/chapter5/challenge5_2_cup_direction.py

第 6 章

【クイズ 6.1】

2 次元の場合，手先の位置は 2 変数，姿勢は 1 変数ですから，合計 3 変数です．したがって，ロボットアームには，最低限 3 自由度が必要です．

【クイズ 6.2】

$$
\begin{aligned}
x &= L_1 \cos(\pi/2) + L_2 \cos(\pi/2 + \pi/2) = L_1 \\
y &= L_1 \sin(\pi/2) + L_2 \sin(\pi/2 + \pi/2) = -L_2
\end{aligned}
$$

【クイズ 6.3】

$$
\begin{aligned}
d &= \sqrt{L_1^2 + L_2^2} \\
q_2 &= \pm \cos^{-1} 0 = \pm \pi/2 \\
\psi &= \phi = \mathrm{atan2}(L_2, L_1) \\
q_1 &= \mathrm{atan2}(L_2, L_1) \mp \mathrm{atan2}(L_2, L_1) \\
(q_1, q_2) &= (0, \pi/2),\ (2\mathrm{atan2}(L_2, L_1), -\pi/2)
\end{aligned}
$$

【クイズ 6.4】

$d > 0$, $L_1 > 0$, $L_2 > 0$ を前提に以下のように式を変形できます．

$$
\begin{aligned}
\frac{d^2 - L_1^2 - L_2^2}{2L_1L_2} &< -1 \\
d^2 - L_1^2 - L_2^2 &< -2L_1L_2 \\
d^2 &< L_1^2 - 2L_1L_2 + L_2^2 = (L_1 - L_2)^2 \\
d &< |L_1 - L_2|
\end{aligned}
$$

$$
\begin{aligned}
\frac{d^2 - L_1^2 - L_2^2}{2L_1L_2} &> 1 \\
d^2 - L_1^2 - L_2^2 &> 2L_1L_2 \\
d^2 &> L_1^2 + 2L_1L_2 + L_2^2 = (L_1 + L_2)^2 \\
d &> L_1 + L_2
\end{aligned}
$$

【クイズ 6.5】

順運動学

$$x = -L_1 \sin q_1 + q_2 \cos q_1$$
$$y = L_1 \cos q_1 + q_2 \sin q_1$$

逆運動学（$q_2 > 0$ に限定すれば解は 1 個）

$$d = \sqrt{x^2 + y^2}$$
$$\phi = \mathrm{atan2}(y, x)$$
$$q_2 = \sqrt{d^2 - L_1^2}$$
$$q_1 = \phi - \sin^{-1}\left(\frac{L_1}{d}\right)$$

$d < L_1$ ならば解なし．

チャレンジ 6.1

1. 動作時間を表す変数 dt の値は，commander1.py の以下で設定します．

```
88              # 目標関節値とともに送る目標時間
89              dt = 0.2
```

キーを 1 回押すたびに変化する角度は同じですから，dt を小さくすると動きが速くなり，大きくすると動きが遅くなります．また，大きくした場合は，dt で指定した時間が経過しないと，次の指令が効きません．

2. commander1.py を変更して，キー入力の種類を増やしたプログラムを以下で提供します．
https://github.com/AI-Robot-Book-Humble/answers/blob/main/chapter6/challenge6_1.py

チャレンジ 6.2

1. 例えば，初期状態からキーボードの a キーを押して，手先の x 座標を増やしていくと，腕がほぼ伸びきった状態で「逆運動学の解なし」と表示されます．これは，commander2.py の以下の部分で処理しています．

```
157                 joint = inverse_kinematics([x, y, z, pitch], elbow_up)
158                 if joint is None:
159                     print('逆運動学の解なし')
160                     joint = joint_prev.copy()
```

一方，初期状態からキーボードの z キーを押して，手先の x 座標を減らしていくと，「関節指令値が範囲外」と表示されます．これは，commander2.py の以下の部分で処理しています．逆運動学の解は得られた後に，関節が可動範囲に入っていないことが検出されています．

```
165             if not all(joint_in_range(joint)):
166                 print('関節指令値が範囲外')
167                 joint = joint_prev.copy()
168                 elbow_up = elbow_up_prev
```

2. アームと周囲の物体の干渉を一般的に扱おうとすると大変難しい問題になりますので，手先と机の面の干渉だけを扱いましょう．そのためには，手先の部品の z 座標が 0 より小さい（机の面より下にある）かどうかを調べます．関節角度を与えると，机の面との干渉の有無を調べてくれるような関数を kinematics.py に追加するといいでしょう．その関数を追加したプログラムと，commander2.py を変更してそれを利用するようにしたプログラムを以下で提供します．

   https://github.com/AI-Robot-Book-Humble/answers/blob/main/chapter6/kinematics6_2.py

   https://github.com/AI-Robot-Book-Humble/answers/blob/main/chapter6/challenge6_2.py

3. 上のプログラムでは，手先を動かすキー入力の種類も増やしています．

チャレンジ 6.3

commander3.py では，動作時間を表す変数 dt の設定は以下にあります．

```
117                 # 目標関節値とともに送る目標時間
118                 dt = 0.2
```

この値を例えば，5[s] にしてみましょう．タイマコールバックによって，0.5[s] 間隔で関節値が表示されていますので，1 回キーを押すと，対応する関節の値が徐々に変化し，10 回目で指令の角度に達することが確認できるはずです．

チャレンジ 6.4

1. commander4.py を変更して，保持している目標のポーズを増やすには，辞書型の変数 goals に要素を増やします．例えば，以下のように行を追加します．

```
114         goals['halves'] = [0.5, 0.5, 0.5, 0.5]
```

2. commander1.py と commander4.py を合体させたプログラムを以下で提供します．

   https://github.com/AI-Robot-Book-Humble/answers/blob/main/chapter6/challenge6_4.py

3. 上のプログラムには，キーボードの r キーを押すと現在のポーズを登録する機能も追加してあります．

チャレンジ 6.5

commander5.py を変更して，target 座標系で与えられた位置に手先を動かしグリッパを閉じて運搬に適したポーズへ動かすようにしたプログラムを以下で提供します．

https://github.com/AI-Robot-Book-Humble/answers/blob/main/chapter6/challenge6_5.py

チャレンジ 6.6

commander2_moveit.py を変更して，MoveIt のモデルの中の机の天板に対応する位置に直方体を配置するようにしたプログラムを以下で提供します．

https://github.com/AI-Robot-Book-Humble/answers/blob/main/chapter6/challenge6_6.py

このプログラムでは，C キーを押すと直方体が削除され，A キーを押すと再び直方体が追加されます．直方体は，RViz の中で緑色に表示されているので確認できます．直方体がある場合とない場合でアームの手先を机の天板に近づけたときの違いを調べてください．

チャレンジ 6.7

commander5_moveit.py を変更して，target 座標系で与えられた位置に手先を動かしグリッパを閉じて運搬に適したポーズへ動かすようにしたプログラムを以下で提供します．

https://github.com/AI-Robot-Book-Humble/answers/blob/main/chapter6/challenge6_7.py

動きはチャレンジ 6.5 とほぼ同じですが，プログラムの違いを調べてください．

チャレンジ 6.8

commander6_moveit.py を変更して，「set_endtip 名前」（名前は tf に登録された座標系の名前）という文字列を受け取ると，指定された名前の座標系の位置・姿勢に手先を動かすアクションを追加したプログラムを以下で提供します．

https://github.com/AI-Robot-Book-Humble/answers/blob/main/chapter6/challenge6_8.py

第 7 章

【クイズ 7.1】

1. 不確実な現実の世界では，事物が確率的にしか決定せず，その場の状況に応じて動的に行動を計画しなければなりません．また，複雑なタスクにおいては，想定すべき行動列が膨大になります．ステートマシンは，行動を実行する状態とイベントに基づく状態遷移により複雑な行動列を簡素に表現できます．これにより，実世界の複雑なタスクを達成する行動列の記述と管理を容易にします．
2. カプセル化は，用意した操作を使ってしか状態の内部にあるデータにアクセスできないようにする仕組みです．これにより，状態間での予期しないデータの書き換えの防止や他のタスクへの状態の再利用を容易にします．
3. プロダクションシステムでは，振る舞いの規則を IF 文と THEN で記述しますが，動的で不確実な実世界において，あらゆる可能性を考慮した規則を記述しきることは極めて困難であったためです．

チャレンジ 7.1

第 3 章のチャレンジ 3.1 で作成したアクションサーバを音声の疑似コードに置き換えます．

チャレンジの解答例はプログラムなのでここに掲載できません．サポートサイトの解答例を参照してください．

- https://github.com/AI-Robot-Book-Humble/answers/tree/main/chapter7