

1. 選択した競技の説明

自由競技で、ロボットと人間でチャンバラを行わせた。人間の持つ赤い棒状のスポンジを認識し、アームで持っている青いスポンジを当てにいくプログラムを実装した。

2.

ノート PC では以下のプログラムを起動した。

- minimal.launch
- ps3_teleop.launch
- 3dsensor.launch
- hsi_color_filter.launch
- rviz
- tsukamu_last.l

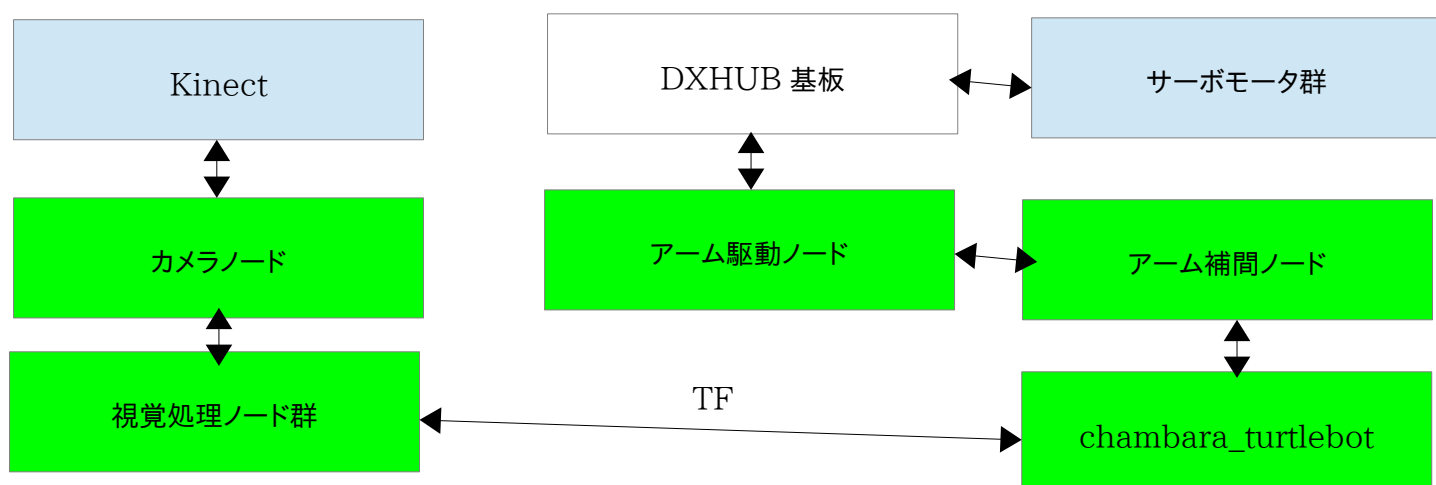
最後の tsukamu_last.l が自分たちで作成したプログラムである。それ以外の 5 つのプログラムを利用して、タートルボットの移動、物体認識を行う。tsukamu_last.l では、画像認識で得られた TF を読み込み、自身の持つスポンジを当てにいくように逆運動学を解く。

プログラム間のデータのやりとりはすべて ROS トピックを利用する。tsukamu_last.l では chambara_turtlebot というノードを立ち上げ、目標の TF を表す

/camera/depth_registered/cluster_decomposer/boxes というトピックを購読する。アームへの命令は euslisp を利用して送る。

tsukamu_last.l で扱うデータやデバイスを簡単に図にまとめると図 1 のようになる。

図1. Turtlebot システム構成



プログラムでは、相手（人間）の持つ赤いスポンジの剣に対して、自分の持つ青いスポンジ剣の中心を当てにいく動作を実装した。これにより、人間とturtlebotでチャンバラができるようになった。

私は主にハードウェアの改造と、ソースコードのうち、逆運動学を解く際の剣の位置の補正を担当した。最初の状態では、Kinectの認識範囲が遠く、認識範囲の一番手前がアームを限界まで伸ばした位置だった。そのため、演習室内にあった端材を用いて、アームの取り付け位置を20cm前方に移動した。これにより認識範囲内でアームを振ることができるようになった。ソースコードについては、メール添付したファイルの30～32行目にあたる部分を担当した。

実際に動かしてみると、うまく青いスポンジの中心が赤いスポンジに当たる場合と、そうでない場合があった。これは、アーム取り付けに使った端材の剛性が低かったために、アームの根元の角度が回転してしまったのが主な理由と考えられる。制御のソフト面については、概ね目的どおりのものが作れた。

3. ロボット行動プログラミング演習の感想

euslispに対する習熟度の違いにより、最終日の課題で取り組める内容の選択肢が大きく変わってしまっている様子でした。特に逆運動学を解くプログラムを実装できていない（自分たちで書けていない）ために苦労しているグループが多かったように感じます。前回までに行った内容については、解答や解説を加えると全体的な理解度が向上するのではないかと思います。