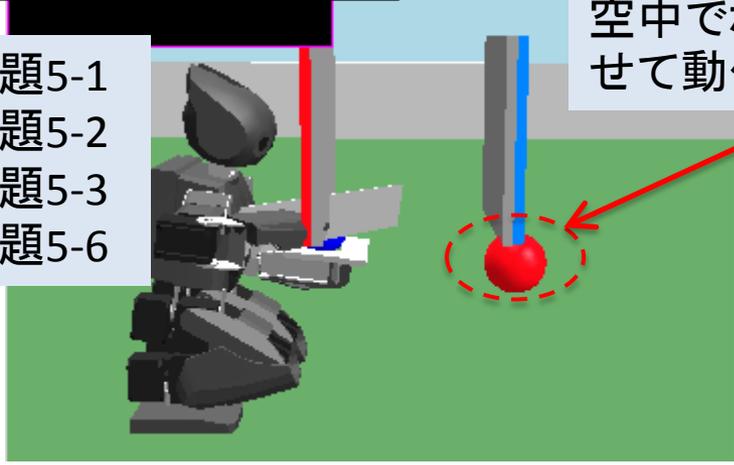


fighter-single.wbt

ボールの検出・追跡、
障害物なしでの動作確認

- 課題5-1
- 課題5-2
- 課題5-3
- 課題5-6



空中でボールを移動させて動作を確認する

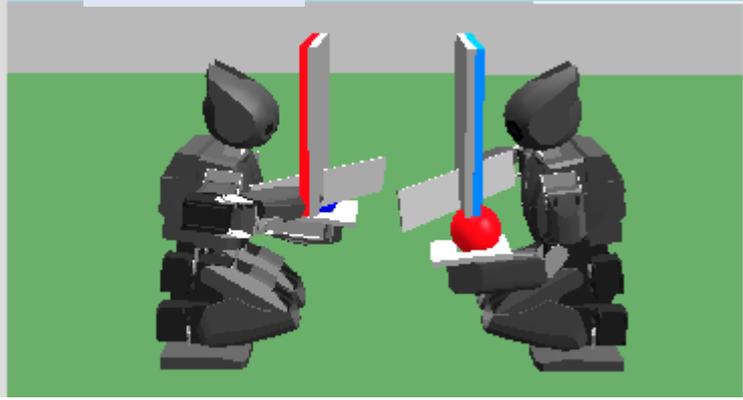
```
Sousei-default.cpp x main.cpp x
544 void SouseiDefault::attack_and_defend() {
545 // =====
// INITIALIZE THE ROBOT.
548
549 // LED
550 setHeadLEDColor(0);
551
552 myStep();
553 enableCamera();
554 enableArmMotorForceFeedback();
```

課題5のプログラムは
attack_and_defend()の
8.2から8.5の部分を
修正して作る

rival-doll.wbt

障害物ありでの動作確認

- 課題5-4
- 課題5-5



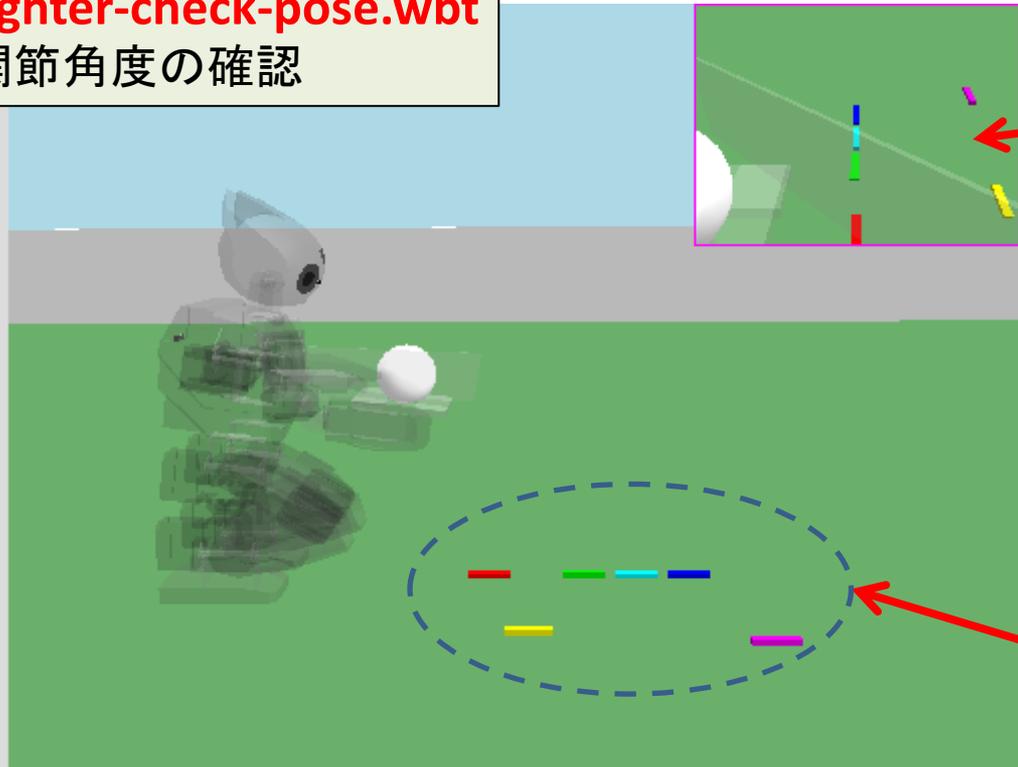
```
main.cpp x Sousei-default.cpp x (doll.cpp x
1 //
2 // You can change the posture of the doll.
3 // The initial posture is defined in the ".
4 //
5
6 #define DOLL_ShoulderL 0.0
7 #define DOLL_ArmUpperL 0.72
8 #define DOLL_ArmLowerL 0.0
9 #define DOLL_ShoulderR 0.0
10 #define DOLL_ArmUpperR 0.72
11 #define DOLL_ArmLowerR 0.9
```

doll.cpp
敵機の姿勢設定
用ファイル

doll.cpp内の
DOLL_ShoulderL等
敵機の姿勢設定
用マクロ
doll.cppをコンパイル
すると適用される

fighter-check-pose.wbt

関節角度の確認



```
Console
INFO: Sousei-student: Starting: ""/home/user/sousei-2014/darwinop-all/control
[Sousei-student] -----
[Sousei-student] Keeper play: find approaching
[Sousei-student] redball and stop it by hand.
[Sousei-student] -----
INFO: fighter check pose: Starting: ""/home/user/sousei-2014/darwinop-all/control
[fighter_check_pose] ShoulderL = -0.187786
[fighter_check_pose] ArmUpperL = 0.77
[fighter_check_pose] ArmLowerL = 0.184857
[fighter_check_pose] ArmUpperR = 0.788714
```

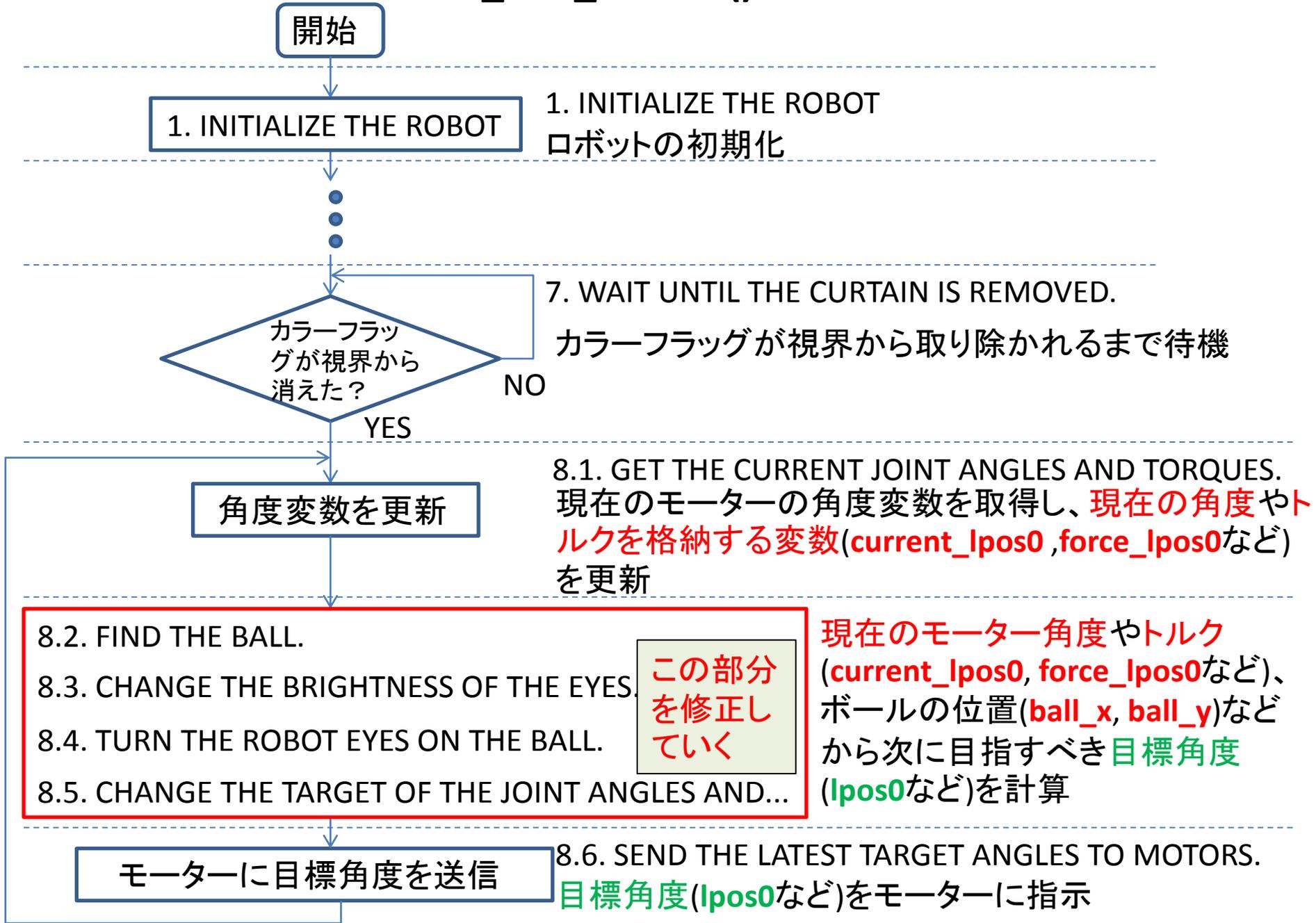
シミュレーション実行時にカメラ画面上で左右に動かすと関節角度が変わる

関節角度を変更するスライダー

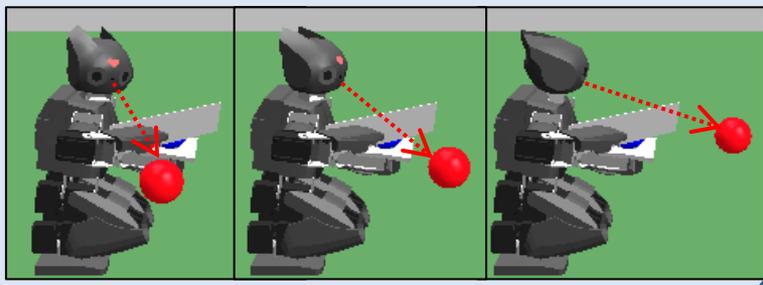
- 赤:ShoulderL
- 黄:ArmUpperL
- 緑:ArmLowerL
- 水色: ShoulderR
- 青: ArmUpperR
- 紫: ArmLowerR

現在の関節角度がコンソールに表示される

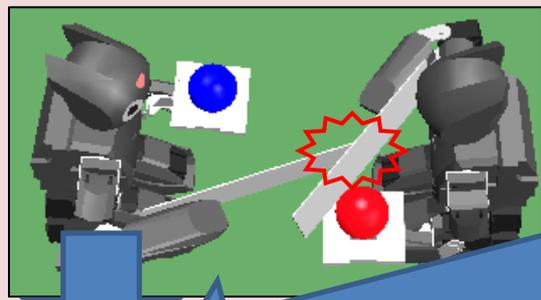
課題5用プログラム attack_and_defend()全体の流れ



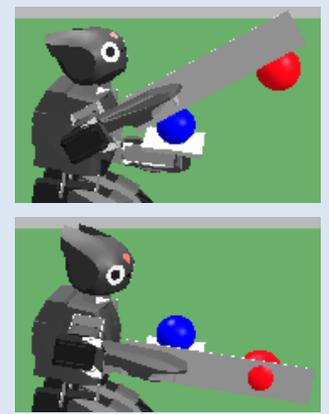
課題5-1. ボールを目で追う機能
(**ボールの方向の把握**)



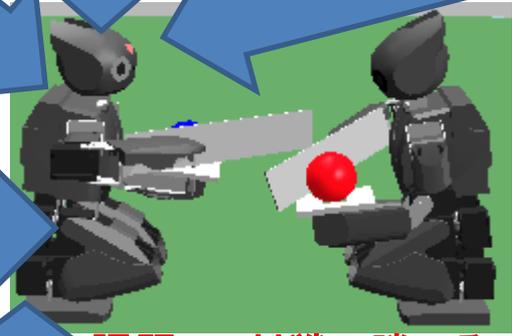
課題5-4. 接触を検知して手を戻す機能
課題5-5. 接触時に相手に衝撃を与えない動作



課題5-6. ボールに合わせて攻撃場所を変える

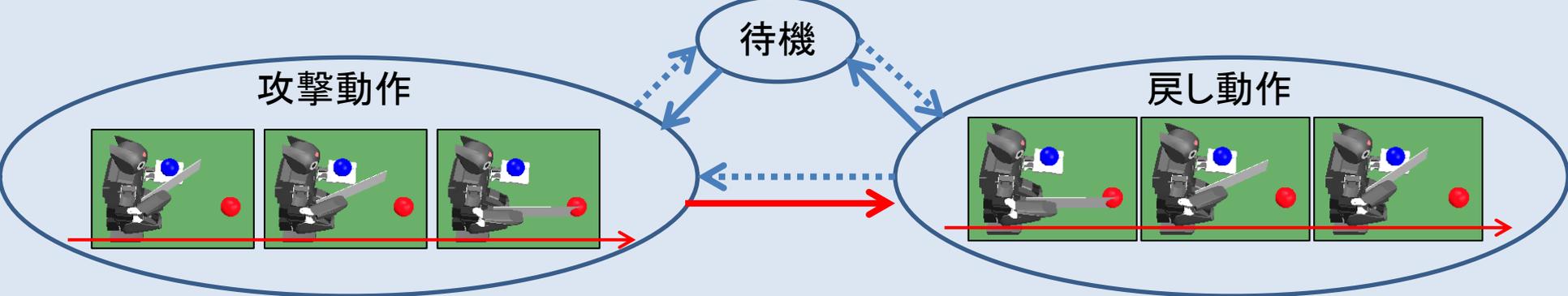


課題5-2. ボールの見えている大きさに
応じた目LEDの明るさ変更
(**見えている大きさの把握**)

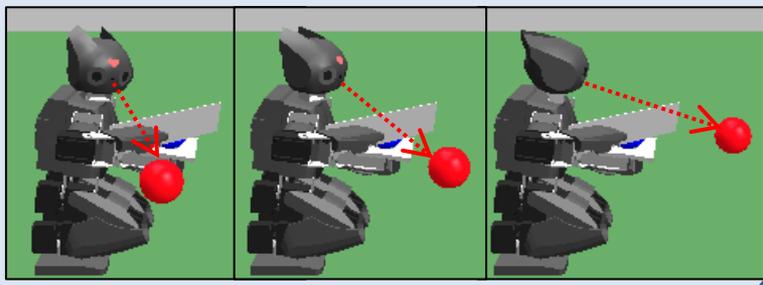


課題5-7. 対戦で勝てるロボット
を目指す

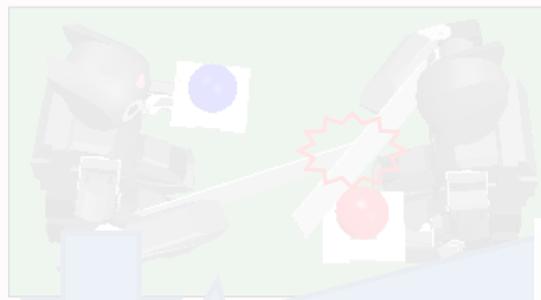
課題5-3. 右手水平往復による攻撃(**状態遷移による攻撃と戻し動作の実現**)
待機、攻撃動作などの現在のロボットの動作に名前を付けて「状態」と呼んでいる。
右手を水平に動かす攻撃動作を行い、それが完了すると戻し動作を行うという状態遷移を追加



課題5-1. ボールを目で追う機能
 (ボールの方向の把握)



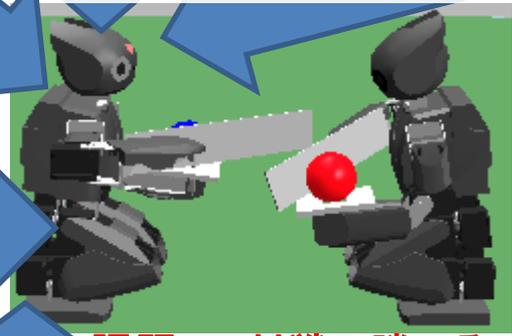
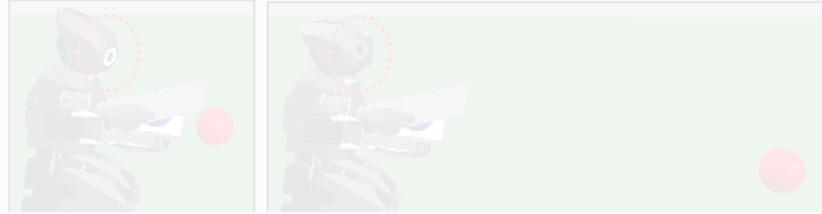
課題5-4. 接触を検知して手を戻す機能
 課題5-5. 接触時に相手に衝撃を与えない動作



課題5-6. ボールに合わせて攻撃場所を変える

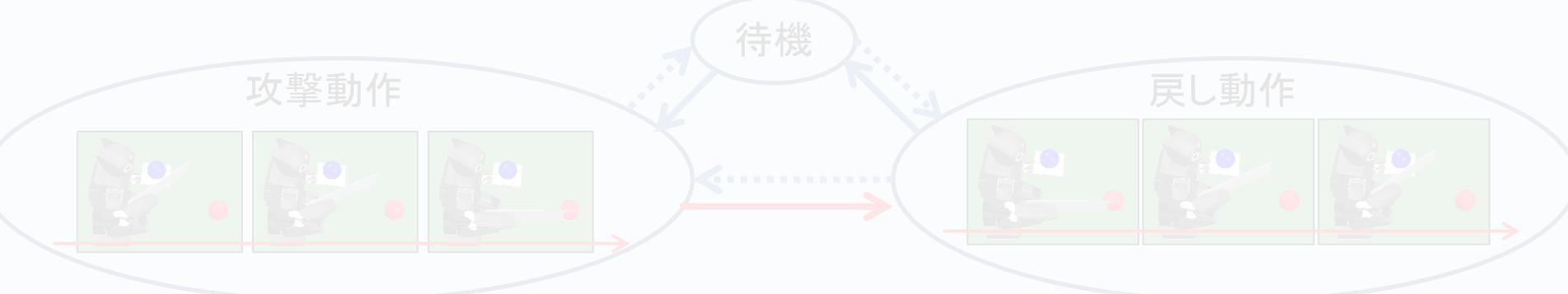


課題5-2. ボールの見えている大きさに
 応じた目LEDの明るさ変更
 (見えている大きさの把握)

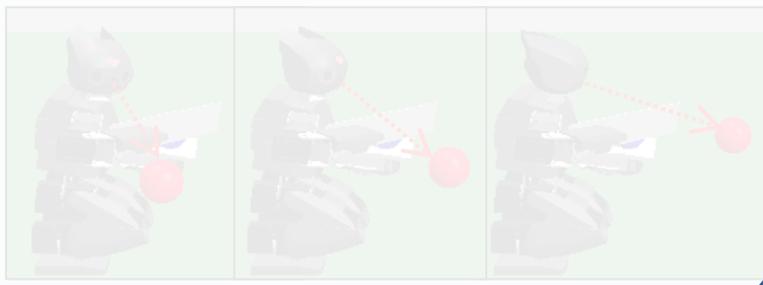


課題5-7. 対戦で勝てるロボット
 を目指す

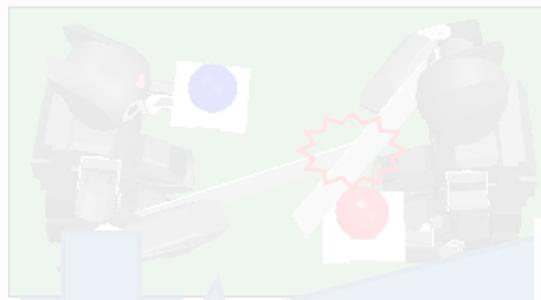
課題5-3. 右手水平往復による攻撃(状態遷移による攻撃と戻し動作の実装)
 待機、攻撃動作などの現在のロボットの動作に名前を付けて「状態」と呼んでいる。
 右手を水平に動かす攻撃動作を行い、それが完了すると戻し動作を行うという状態遷移を追加



課題5-1. ボールを目で追う機能
(ボールの方向の把握)



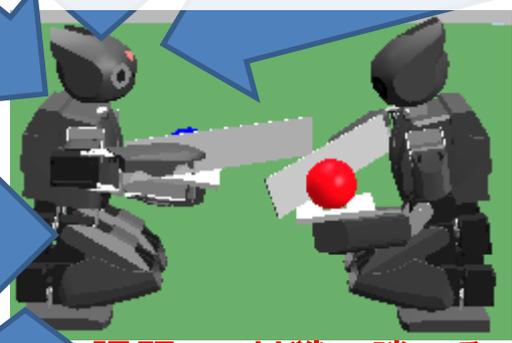
課題5-4. 接触を検知して手を戻す機能
課題5-5. 接触時に相手に衝撃を与えない動作



課題5-6. ボールに合わせて攻撃場所を変える

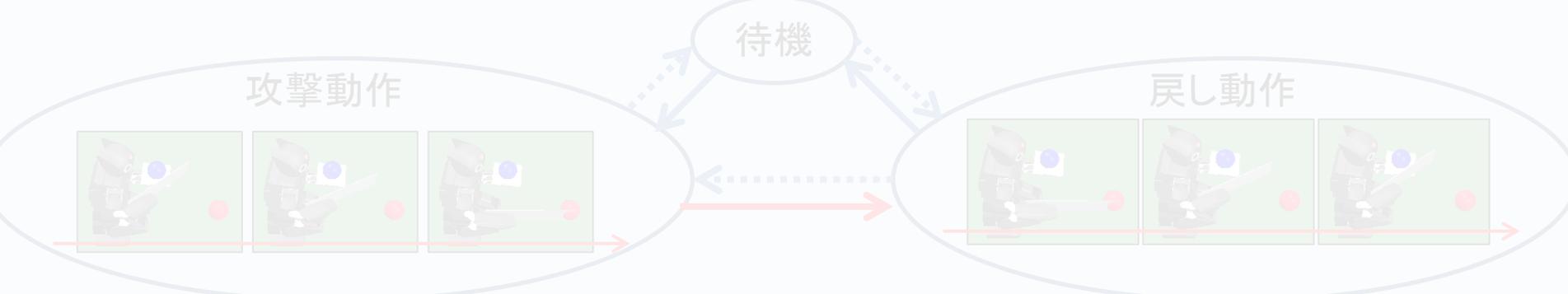


課題5-2. ボールの見えている大きさに
応じた目LEDの明るさ変更
(見えている大きさの把握)

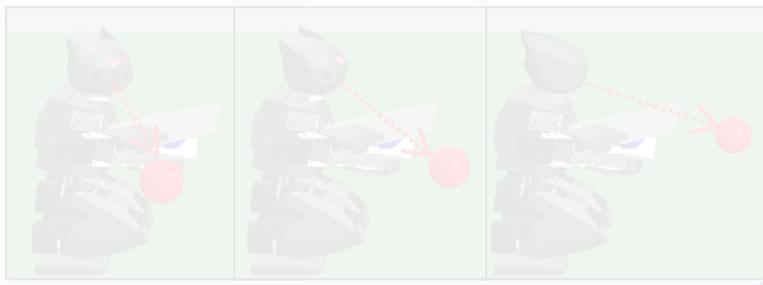


課題5-7. 対戦で勝てるロボット
を目指す

課題5-3. 右手水平往復による攻撃(状態遷移による攻撃と戻し動作の実装)
待機、攻撃動作などの現在のロボットの動作に名前を付けて「状態」と呼んでいる。
右手を水平に動かさず攻撃動作を行い、それが完了すると戻し動作を行うという状態遷移を追加

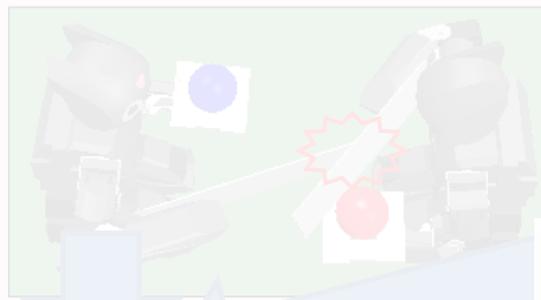


課題5-1. ボールを目で追う機能
(ボールの方向の把握)



課題5-4. 接触を検知して手を戻す機能

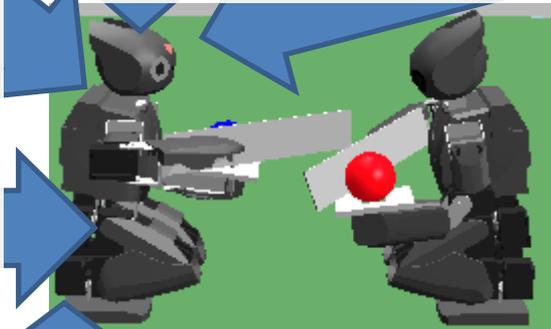
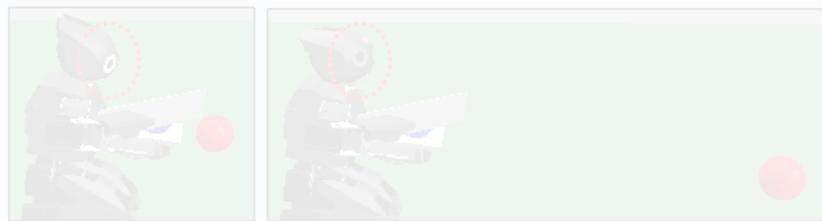
課題5-5. 接触時に相手に衝撃を与えない動作



課題5-6. ボールに合わせて攻撃場所を変える



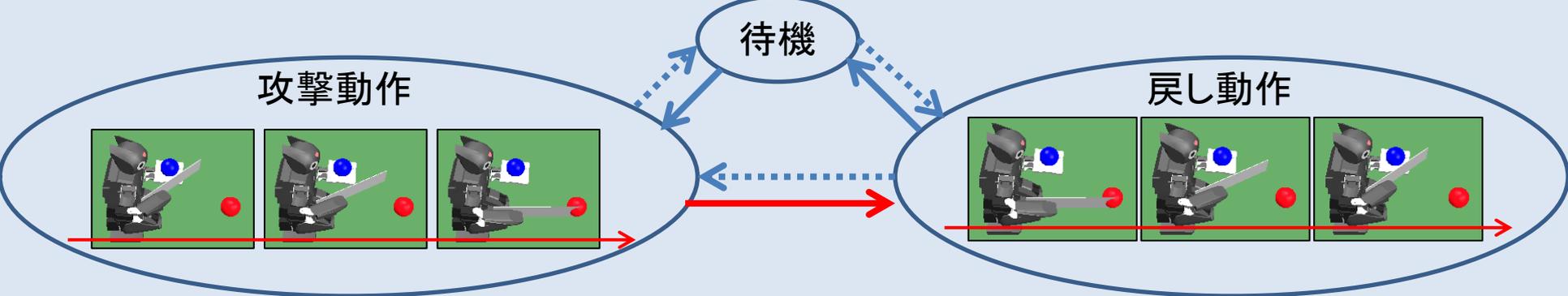
課題5-2. ボールの見えている大きさに
応じた目LEDの明るさ変更
(見えている大きさの把握)



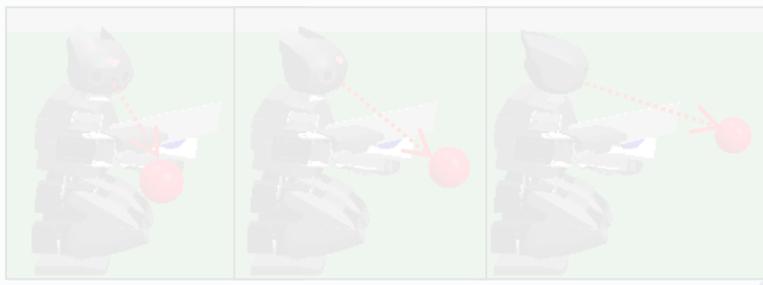
課題5-7. 対戦で勝てるロボット
を目指す

課題5-3. 右手水平往復による攻撃(状態遷移による攻撃と戻し動作の実現)

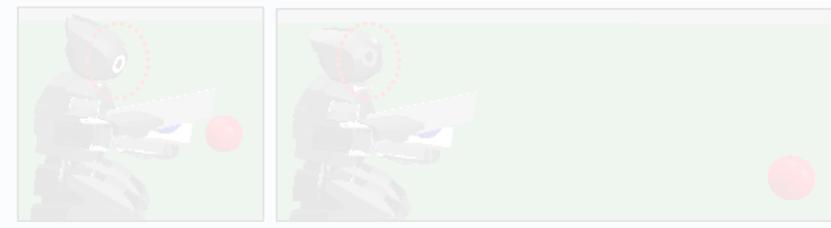
待機、攻撃動作などの現在のロボットの動作に名前を付けて「状態」と呼んでいる。
右手を水平に動かす攻撃動作を行い、それが完了すると戻し動作を行うという状態遷移を追加



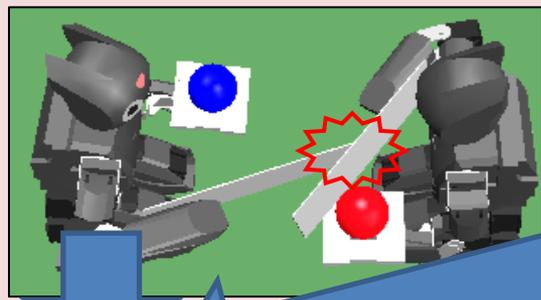
課題5-1. ボールを目で追う機能
(ボールの方向の把握)



課題5-2. ボールの見えている大きさに
応じた目LEDの明るさ変更
(見えている大きさの把握)



課題5-4. 接触を検知して手を戻す機能
課題5-5. 接触時に相手に衝撃を与えない動作



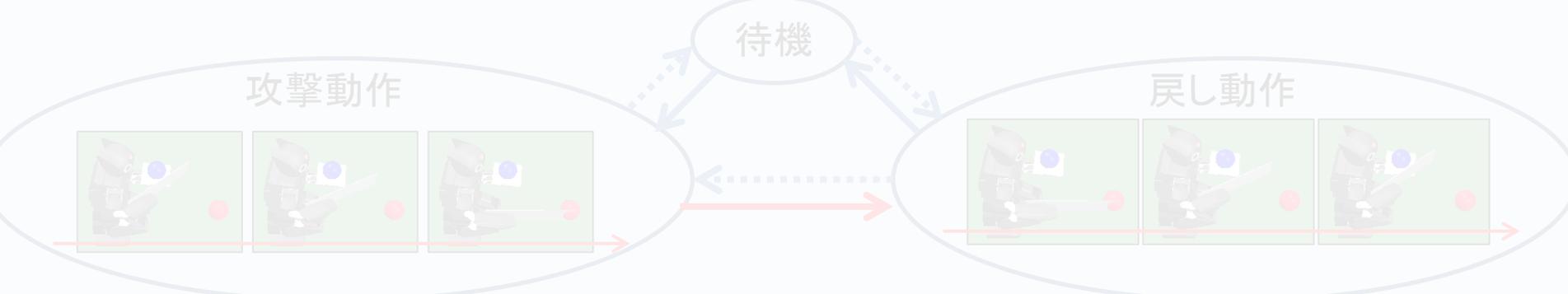
課題5-6. ボールに
合わせて攻撃場所を
変える



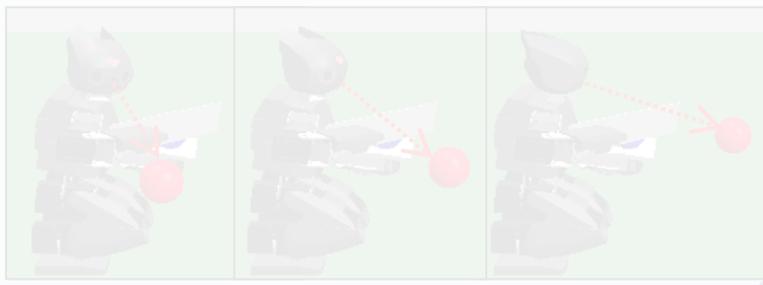
課題5-7. 対戦で勝てるロボット
を目指す

課題5-3. 右手水平往復による攻撃(状態遷移による攻撃と戻し動作の実現)

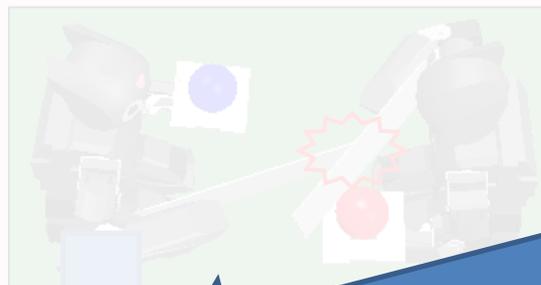
待機、攻撃動作などの現在のロボットの動作に名前を付けて「状態」と呼んでいる。
右手を水平に動かす攻撃動作を行い、それが完了すると戻し動作を行うという状態遷移を追加



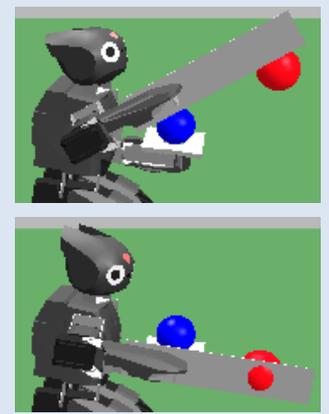
課題5-1. ボールを目で追う機能
(ボールの方向の把握)



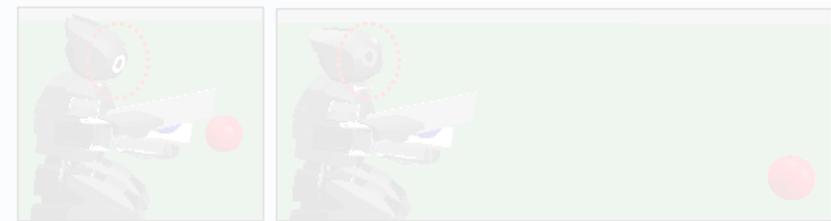
課題5-4. 接触を検知して手を戻す機能
課題5-5. 接触時に相手に衝撃を与えない動作



課題5-6. ボールに合わせて攻撃場所を変える

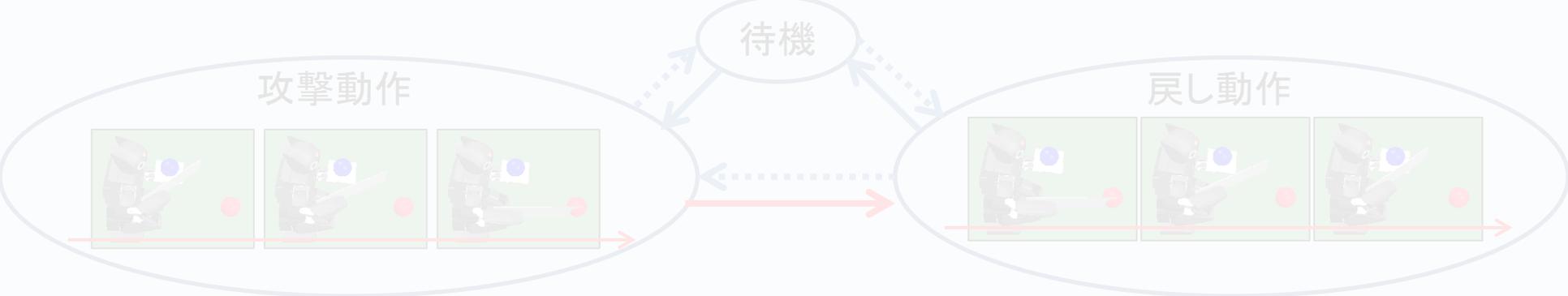


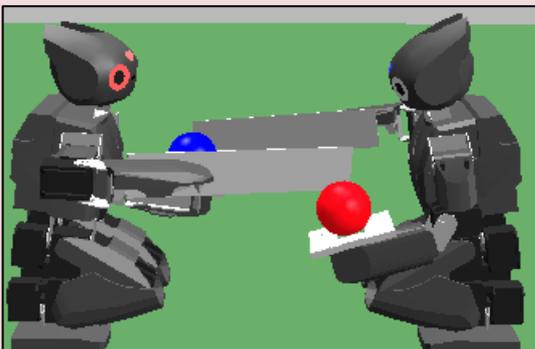
課題5-2. ボールの見えている大きさに
応じた目LEDの明るさ変更
(見えている大きさの把握)



課題5-7. 対戦で勝てるロボット
を目指す

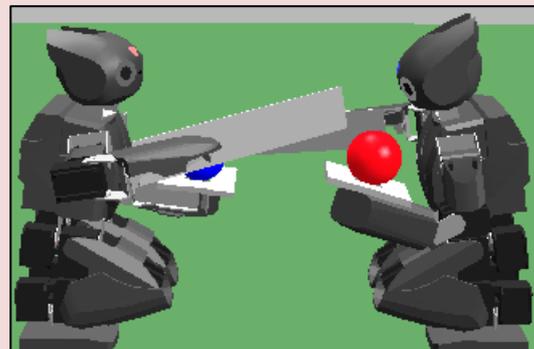
課題5-3. 右手水平往復による攻撃(状態遷移による攻撃と戻し動作の実現)
待機、攻撃動作などの現在のロボットの動作に名前を付けて「状態」と呼んでいる。
右手を水平に動かす攻撃動作を行い、それが完了すると戻し動作を行うという状態遷移を追加





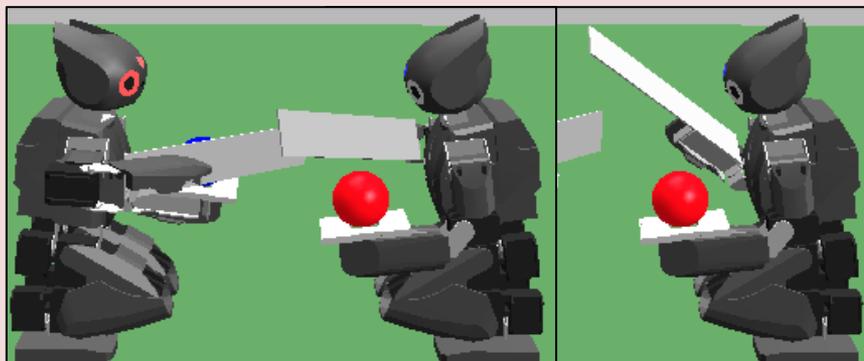
rival-level1.wbt

レベル1. ボールが
ランダムな高さで静
止している
(敵側防御なし)



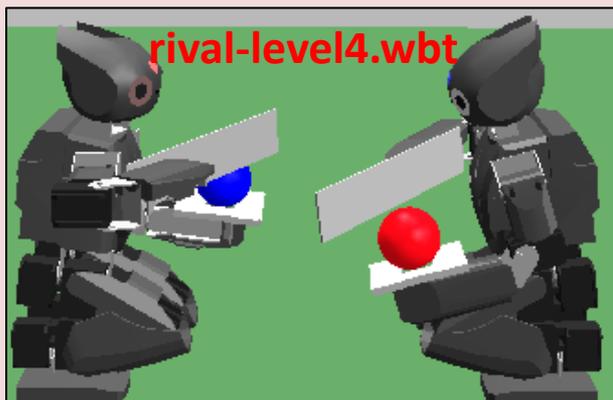
rival-level2.wbt

レベル2. ボールが
上下に移動する
(敵側防御なし)



rival-level3.wbt

レベル3. 敵機が
ボールを隠したり見
せたりする
(ボールは静止)



rival-level4.wbt

レベル4. 敵機がボールを隠しつつ
ボールを上下に移動する



rival-level5.wbt

レベル5. 敵機がボールを隠しつつ
自機のボールを攻撃してくる

課題5-3 modeの移り変わり

赤色の部分を
追加実装する

ボールが検出され、
その面積が十分
大きい時

攻撃

mode==MODE_ATTACK
右腕モーターの目標角
度(rpos0など)を攻撃用
に変更する

待機

mode==MODE_WAIT

現在の角度が目標
角度に十分近づい
たとき

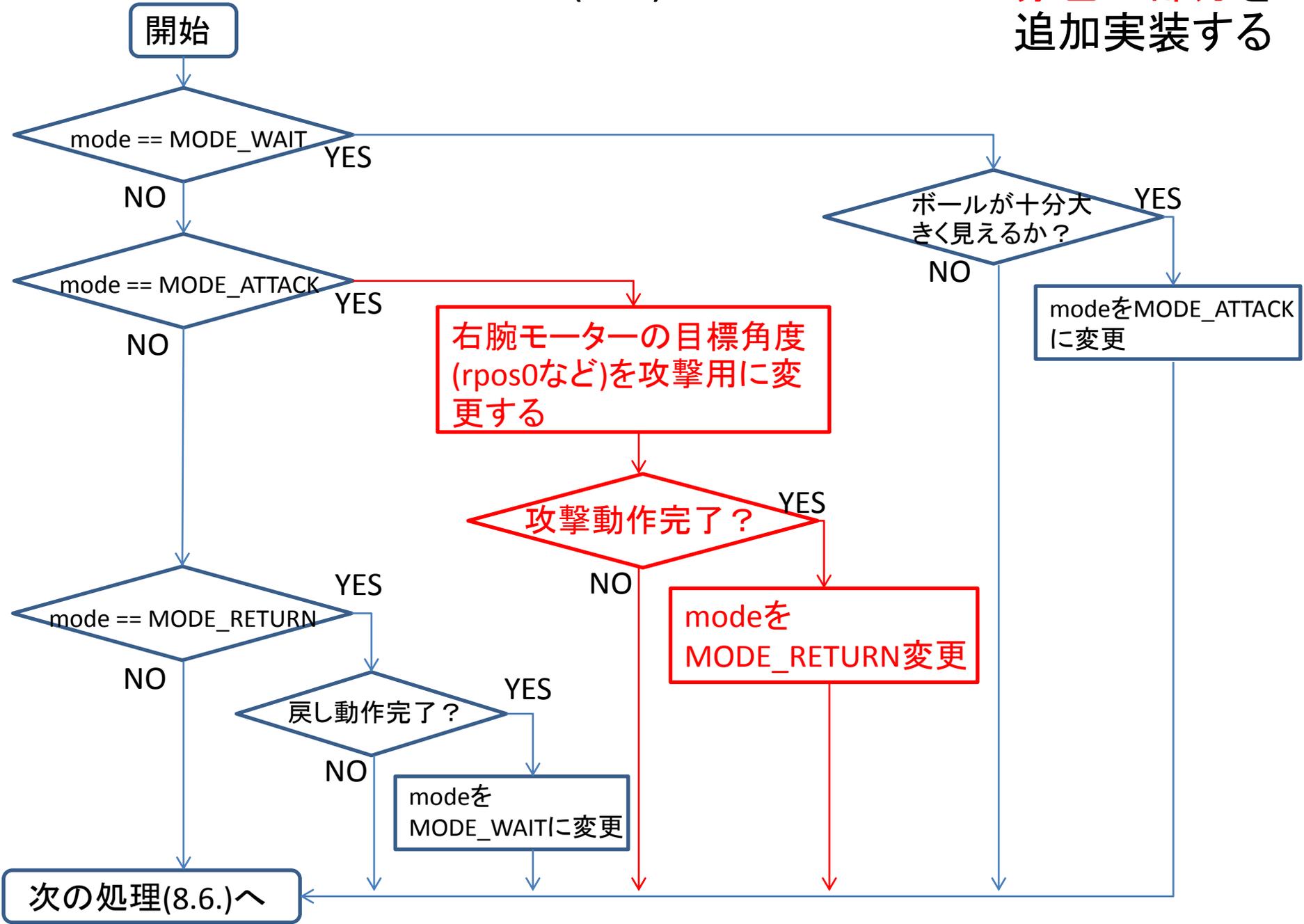
戻し動作

mode==MODE_RETURN
右腕モーターの目標角
度を待機用に変更する

攻撃動作が完了したとき(現在の
モーター角度(current_rpos0など)
が目標角度(rpos0など)に十分近
づいたとき)

状態遷移を実現するプログラム(8.5.)の流れ

赤色の部分を追加実装する



課題5-4,5-5 modeの移り変わり

赤色の部分を
追加実装する

ボールが検出され、
その面積が十分
大きい時

待機
mode==MODE_WAIT

現在の角度が目標
角度に十分近づい
たとき

攻撃

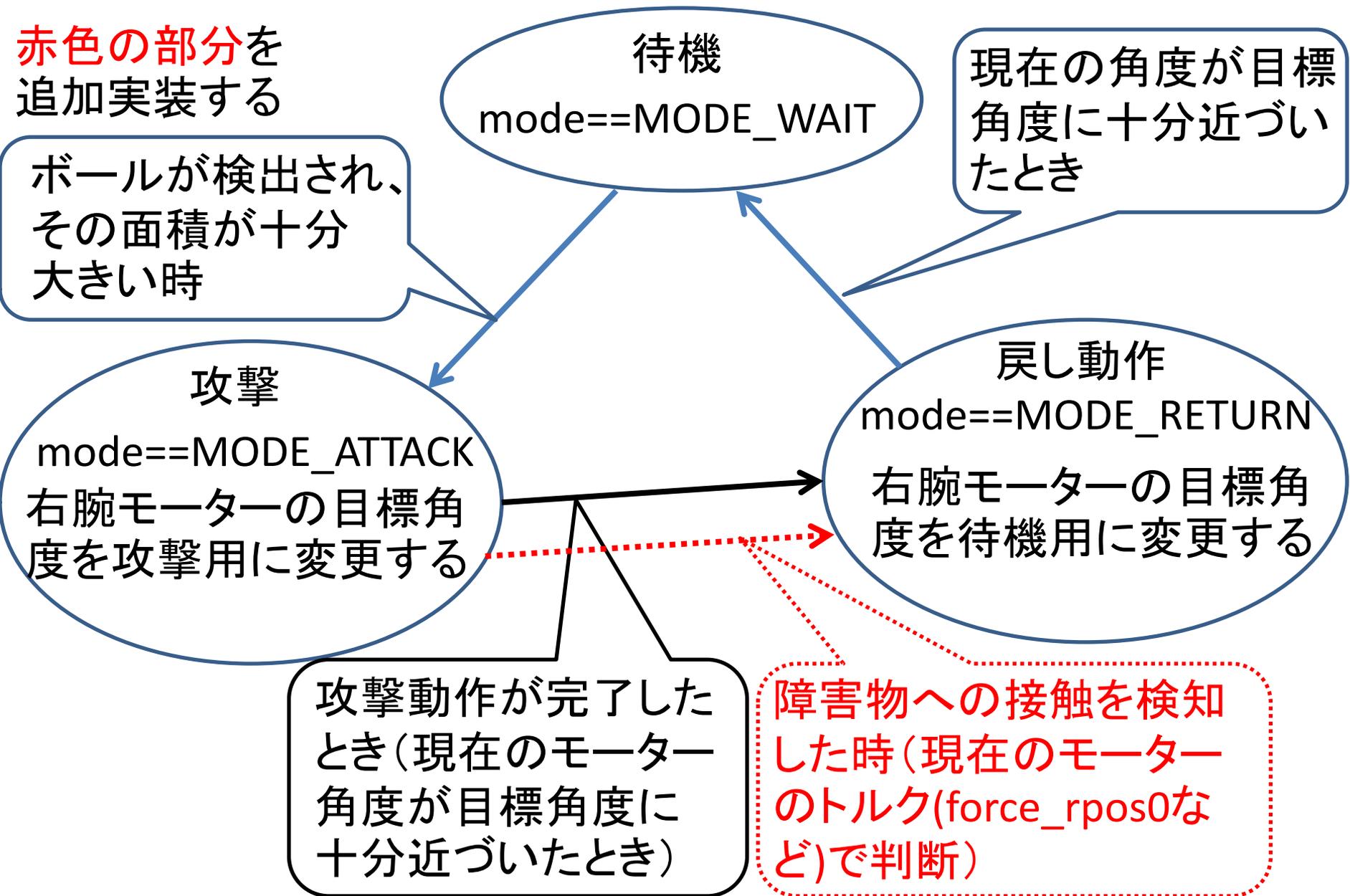
mode==MODE_ATTACK
右腕モーターの目標角
度を攻撃用に変更する

戻し動作

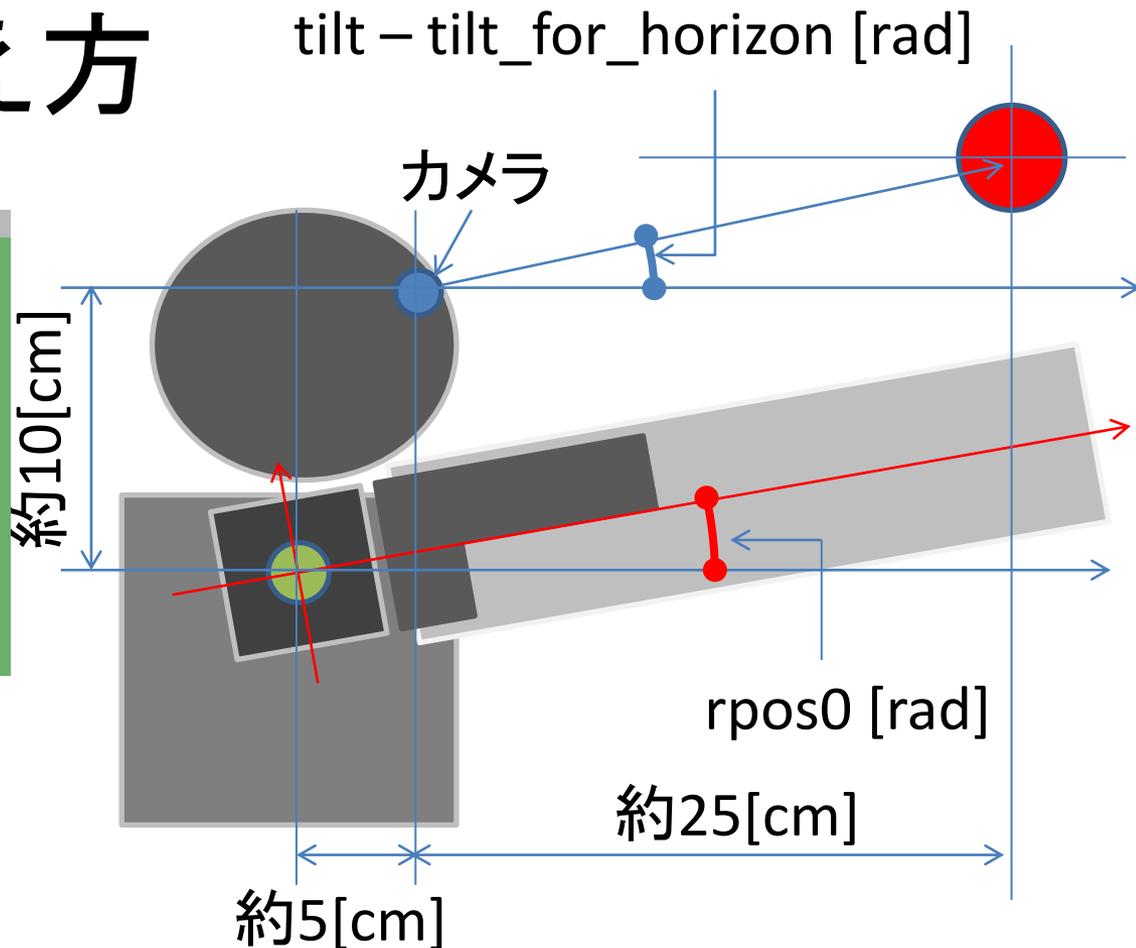
mode==MODE_RETURN
右腕モーターの目標角
度を待機用に変更する

攻撃動作が完了した
とき(現在のモーター
角度が目標角度に
十分近づいたとき)

障害物への接触を検知
した時(現在のモーター
のトルク(force_rpos0な
ど)で判断)



課題5-6の考え方



手先をボールに触れさせるには手先の上下方向の目標角度 rpos0 をどのように設定すればよいか？