自律分散ロボット群の理論モデルに対する実機シミュレーションに関する研究

法政大学大学院　田邉太一
法政大学　　和田幸一
名古屋工業大学　片山喜章

研究背景

自律分散ロボット群の研究では理論モデルが主体で実際に動作させるということが少なく、実現が難しい部分がある

⇒理論モデルを実現する際にどう差を少なくするか

⇒実際のロボットの動作を理論モデルの参考にする

研究内容

アルゴリズムとスケジュールを記述し、スケジュールに従ってロボットを動作させるための動作制御と記述したスケジュールに従ってロボットが動作しているか確認を行うシステムの作成

理論モデル上のロボットの動作

- ロボットはWAIT,LOOK,COMPUTE,MOVEをサイクルとして繰り返す
  - ◆WAIT

    待機状態
  - ◆LOOK

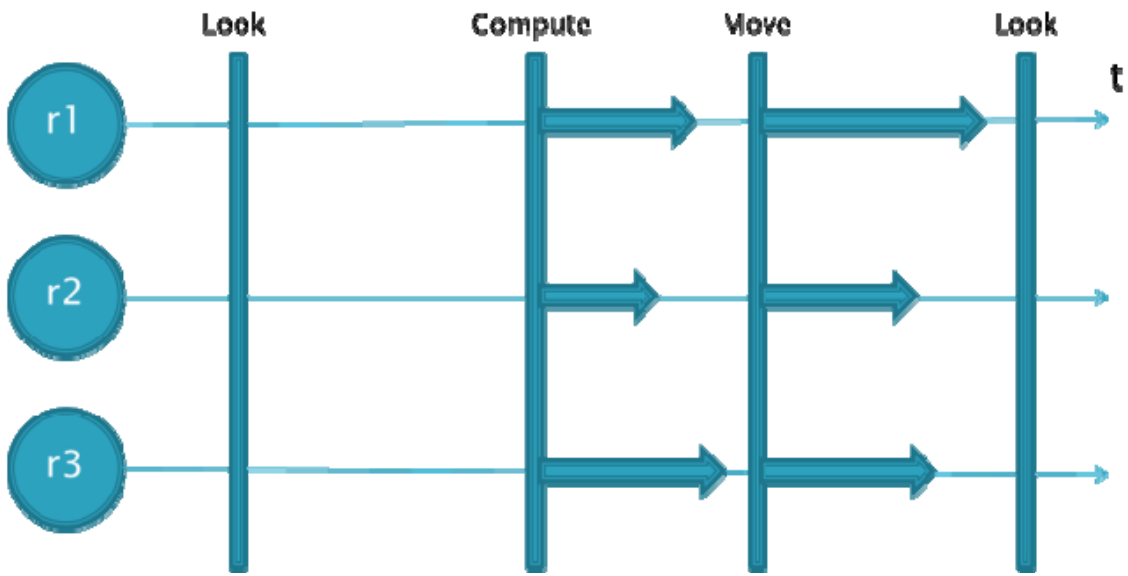
    計測するロボットを原点として、視界に有るロボットの座標を計測する
  - ◆COMPUTE

    LOOKで得た情報を元に目的地を計算する
  - ◆MOVE

    目的地へ移動する

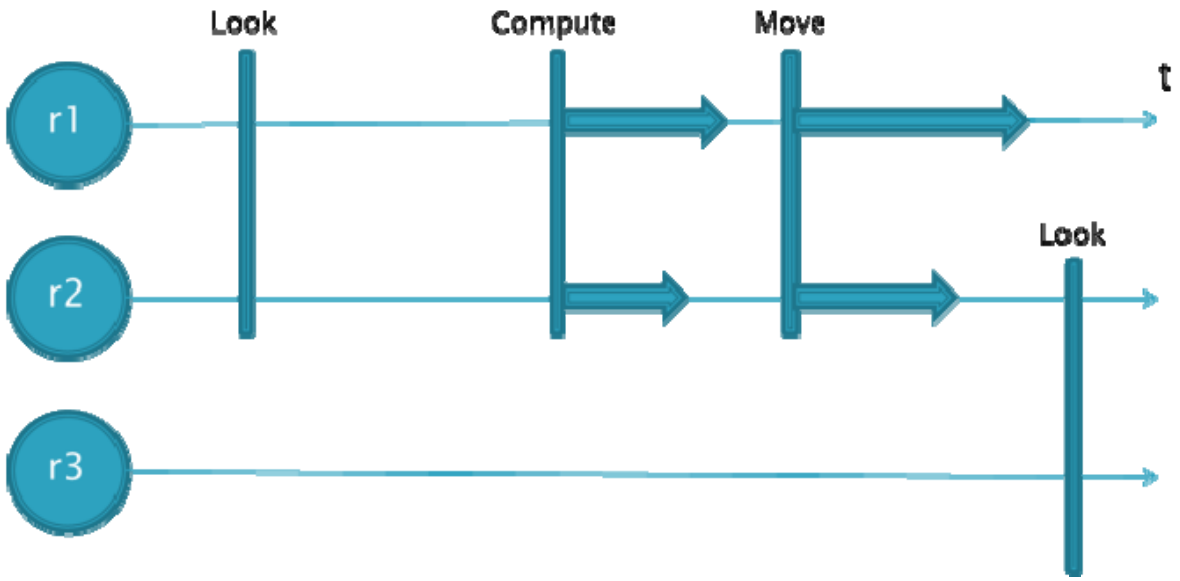
理論モデル上のロボットの同期

- FSYNC（完全同期）



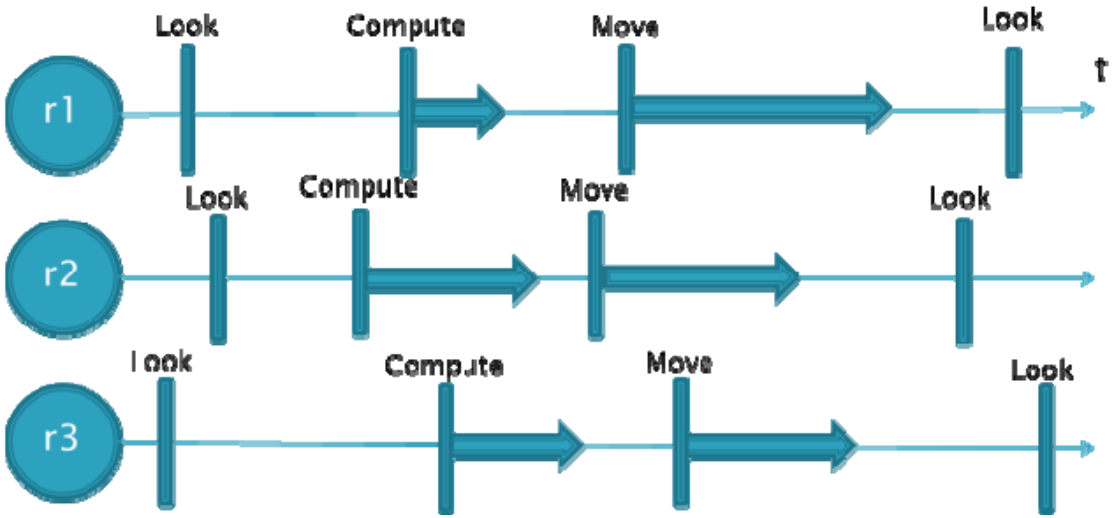
理論モデル上のロボットの同期

- SSYNC（半同期）



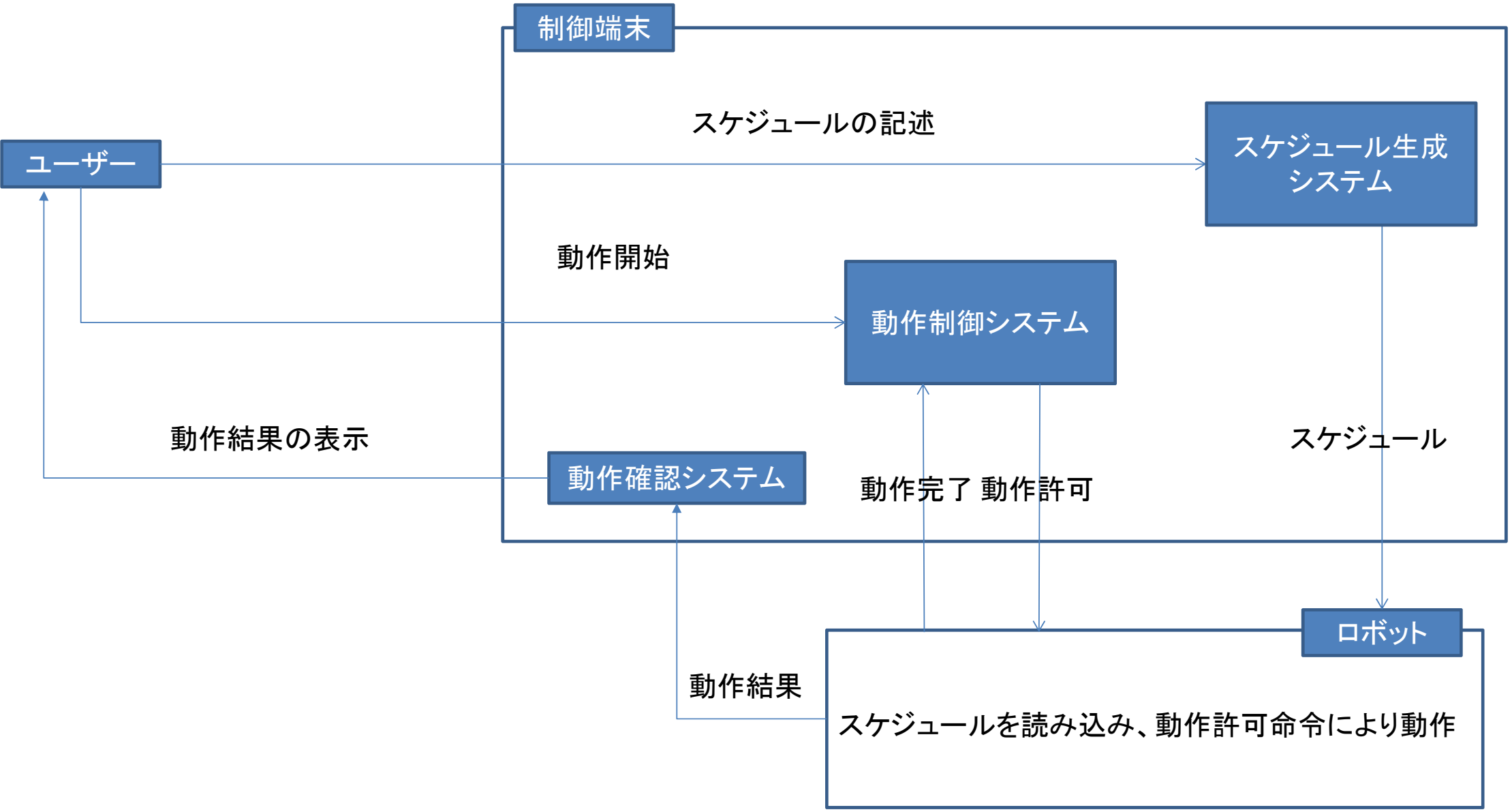
理論モデル上のロボットの同期

- ASYNC（非同期）



システムの各機能

- スケジュール生成システム
  - ユーザーから入力された情報をもとにスケジュールを作成しロボットに送る
- 動作制御システム
  - ロボットの動作の同期をとる際に使用する
- 動作確認システム
  - ロボットから受けとった動作完了信号（ロボットの実動作時間）からロボットの時間ごとの動作を表示する
- ロボット
  - 受け取ったスケジュールと送信制御に従って動作し、動作が終了したら制御端末に送信する


制御端末
スケジュールの記述
ユーザー
スケジュール生成システム
動作開始
動作制御システム
動作結果の表示
スケジュール
動作確認システム
動作完了 動作許可
ロボット
動作結果
スケジュールを読み込み、動作許可命令により動作

ロボットの実現

理論モデルのロボットを以下のもので実現する

- Kinect
  - ◆観察
- PC
  - ◆計算
- ルンバ,Turtlebot
  - ◆移動
- 色紙
  - ◆ロボットの認識

ロボットの認識

Kinectで周囲の画像を取得し,ある程度の大きさの色の塊を発見した場合ロボットとして色の塊の中心をロボットの座標とし,それまでに回転した角度と距離を取得する

動作の実現

- 動作
  - ◆LOOK

    ロボットを１回転させ、Kinectで他のロボットを発見するまでに
    かかった角度と距離を計測する
  - ◆COMPUTE

    LOOKで得た結果をもとに目的地までの角度と距離を計算する
  - ◆MOVE

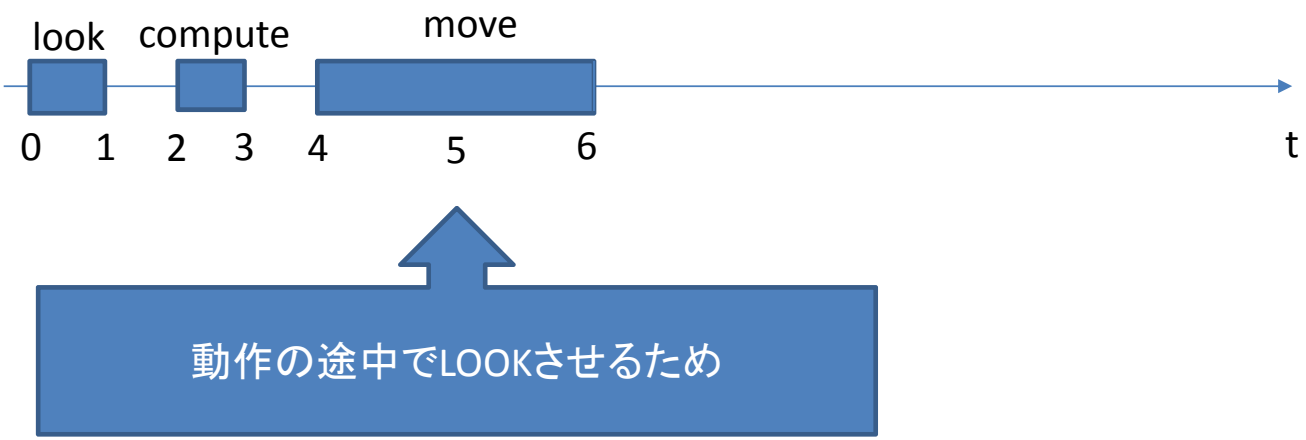
    COMPUTEで得た角度ロボットを回転させ、計算した距離移動する

スケジュール生成システム

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16(単位時間)
robot1																	
robot2																	
robot3																	
robotn																	

単位時間の定義

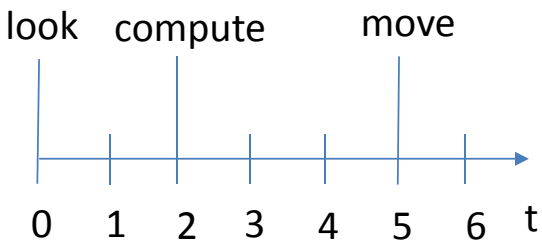
単位時間は1つの動作を開始してから終了するまでを1とする
またMOVEの動作を分割して1とすることもできる



スケジュールの記述形式

サイクルの記述形式は以下のようにする

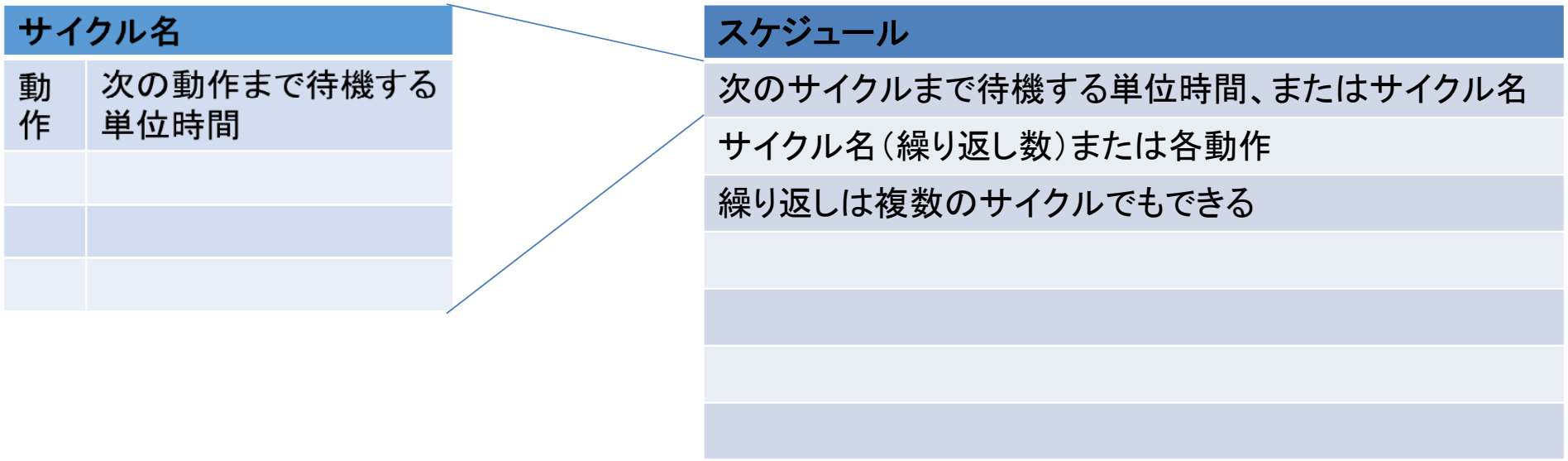
サイクル名	
動作	次の動作まで待機する単位時間



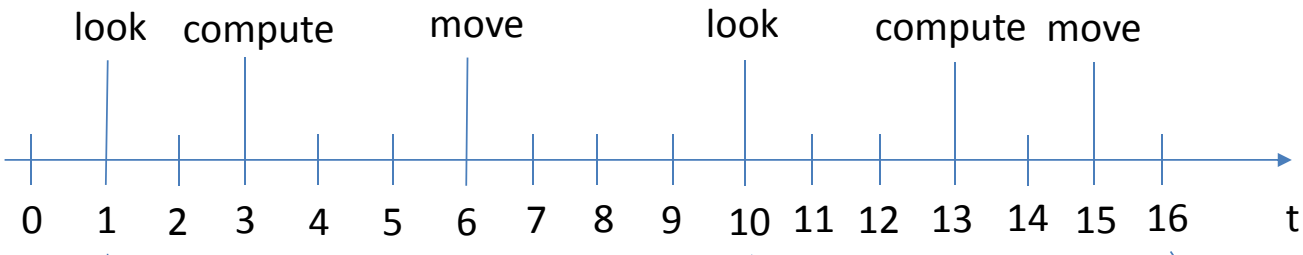
サイクルA	
look	1
compute	2
move	

スケジュールの記述形式

スケジュールの記述形式は以下のようにする



スケジュールの記述例



サイクルA	
look	2
compute	3
move	4

サイクルB	
look	3
compute	2
move	1

スケジュール
1
サイクルA（1）
サイクルB（1）

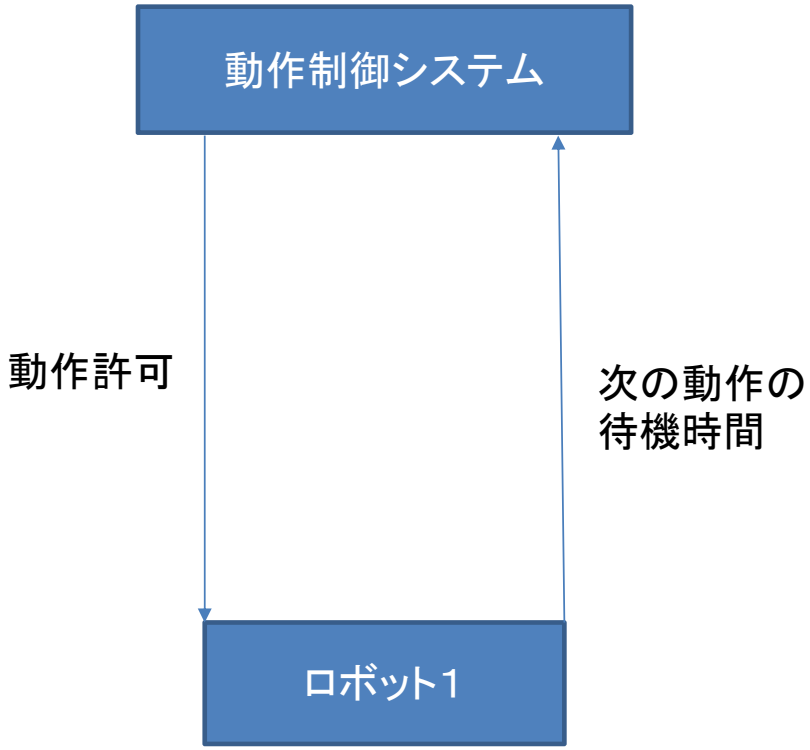
動作制御システム

動作制御システム

・動作制御システムの動作
各ロボットに動作を行う許可を出す

・動作制御システムが記録するもの
1.受け取った次の動作までの待機時間
2.ロボットから動作完了信号を受け取ったか

・動作許可の送信条件
全てのロボットから動作完了を受け取り、待機時間が0のロボットがある場合に待機時間が0のロボットのみに動作許可を送信する



動作制御システムのアルゴリズム

ロボット側

```
While(動作開始命令=false){
}
If(スケジュールの先頭=待機時間){
        制御システムへ送信
}
While(目的達成=false){
        if(動作許可命令=true){
                動作実行
                待機時間を送信
        }
}
```

動作制御システム側

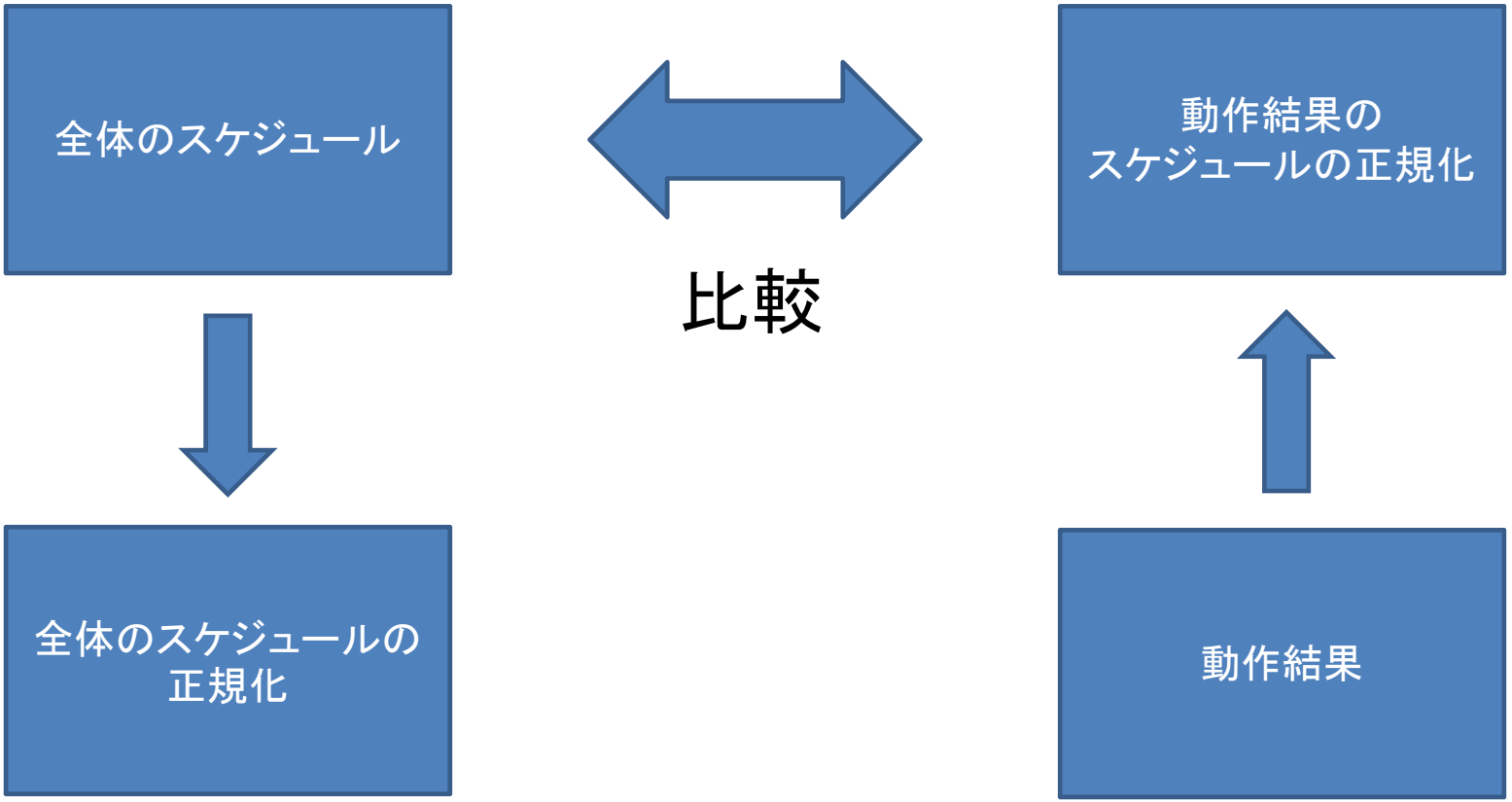
```
動作開始命令を送信
While(待機時間=false){
}
While(目的達成=false){
        if(待機時間=0){
                動作許可命令を送信
                待機時間-1
        }
        else{
                待機時間-1
        }
}
```

動作確認システム

ロボットが動作許可命令を受け取ってから動作完了命令を送信するまでの時間をロボットが動作を行った時間とする

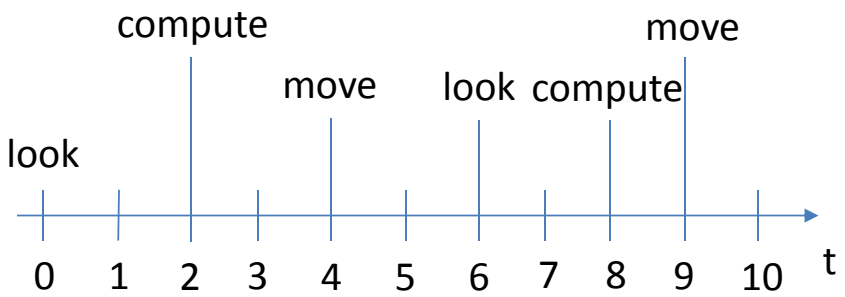
ロボットが目的を達成した時に全てのロボットから動作確認システムに全ての動作を行った時間を送信し、それを元にスケジュールを正規化して記述されたスケジュールに従って動作していたか確認する

動作結果の確認方法

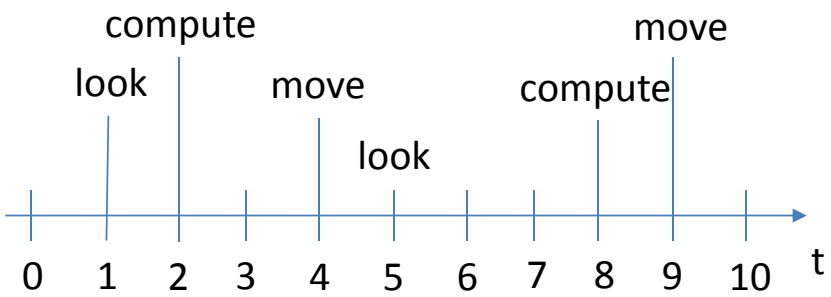


記述したスケジュールの正規化

スケジュール1と2の全てのLOOKに対して,任意のiに対してi番目のLOOKをした時のロボットの配置が等しければロボットの配置が同じ場合スケジュール1と2を同じものとみなす

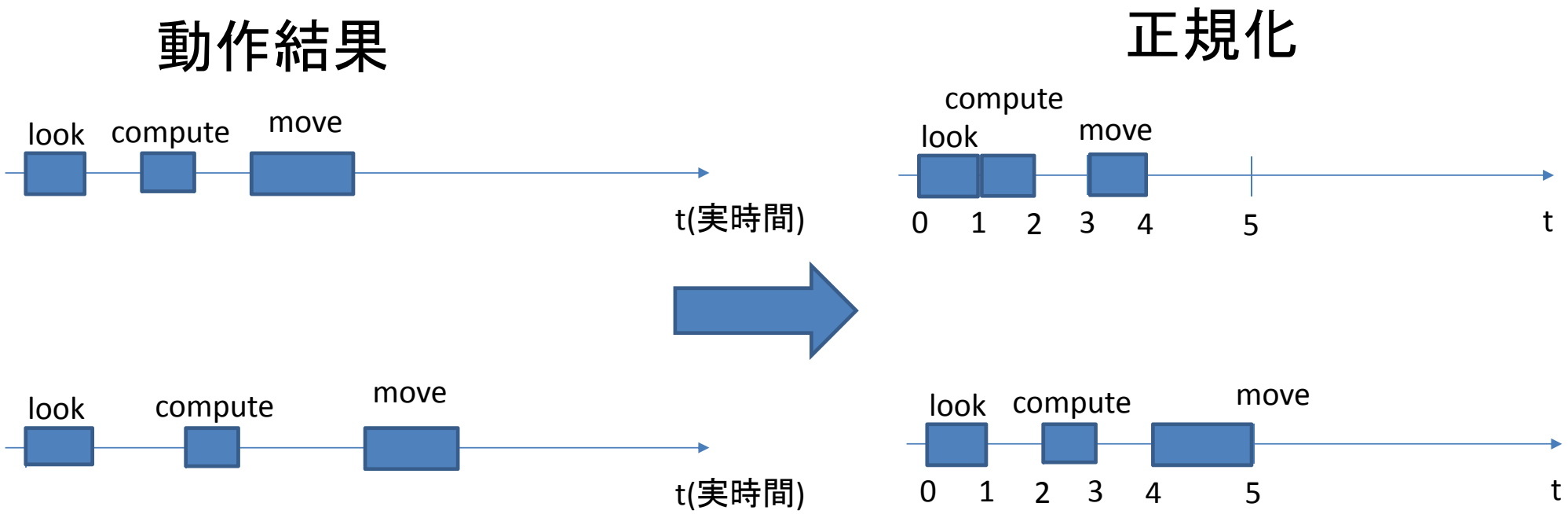


スケジュール1



スケジュール2

動作結果に対するスケジュールへの正規化



スケジュールの実現

ロボットの動作結果からスケジュールを正規化し、記述したスケジュールと正規化されたスケジュールが等価であれば記述したスケジュールが実現されたとする

記述したスケジュール

動作結果

今後の課題

進捗状況

- ロボットの実装は終わっている
- システムの作成
- スケジュールの等価の定義の設定