

# 離散事象システムに基づいた人間と調和した マルチロボットによる倉庫自動化に関する研究

大阪市立大学 工学部 電気情報工学科  
エレクトロニクス領域  
A17TN029 野田 健太郎

# 研究背景

無人搬送ロボットを用いた倉庫の自動化の研究が進んでいる。

しかし人間が立ち入る際システムを完全に停止させなければいけない  
これは現在課題となっている。

そこでHuman-in-the-loopを取り入れた，離散事象システムに基づいた  
スーパーバイザ制御<sup>[1]</sup>を考える。

## 先行研究<sup>[2]</sup>

倉庫の自動化についての先行研究は  
いくつかあるが，これらはロボット  
だけを考えた倉庫内の制御について



倉庫のイメージ： <https://rapyuta-qvou.jp/>より

[1] W.M. Wonham and Kai Cai, "Supervisory Control of Discrete-Event Systems", Communications and Control Engineering, Springer, 2019.

[2] 辰本 佑太, 白石 昌大, Kai Cai "Application of Supervisory Control Theory with Warehouse Automation Case Study", Transactions of the Institute of Systems, Control and Information Engineers (ISCIE), Special Issue on Event-Driven Approach to System Design -- Application and Development, vol. 62, no. 6, pp. 203-208, 2018.

# 研究目的

---

ロボットの強み  
(正確性, パワー, スピード)



人間の強み  
(手先の器用さ, 判断力)



予期せぬトラブルやロボットには難しいタスク  
にも対応できるようにする.

ロボットが作業する倉庫内に, 人間も立ち入るとなった場合,  
人間の絶対的な安全を確保しなければならない.

そこで2つの方法を提案します.

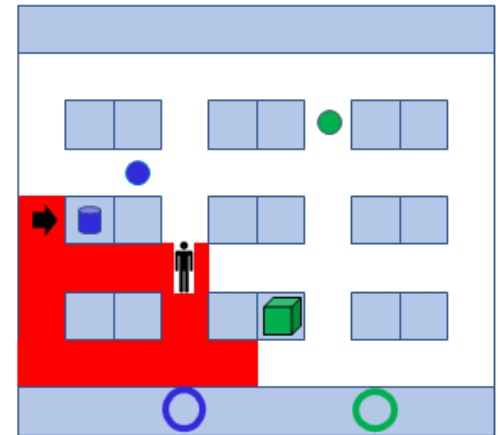
# 安全を確保する方法①

## ■方法1

: 人間の経路上にロボットを進入させない.

ロボットのようには厳密な制御はできない.

⇒人間の行動を不可制御事象で定義する.



■ ロボットの進入を禁止する範囲

### 具体的な計算手順

1. ロボットと人間のモデルを作成
2. 制御要求を作成（同時に同じマスに存在しない）
3. オートマトンを同期合成し，制御対象を作成
4. 制御対象が制御要求を満たすようなスーパーバイザを作成する

## 安全を確保する方法②

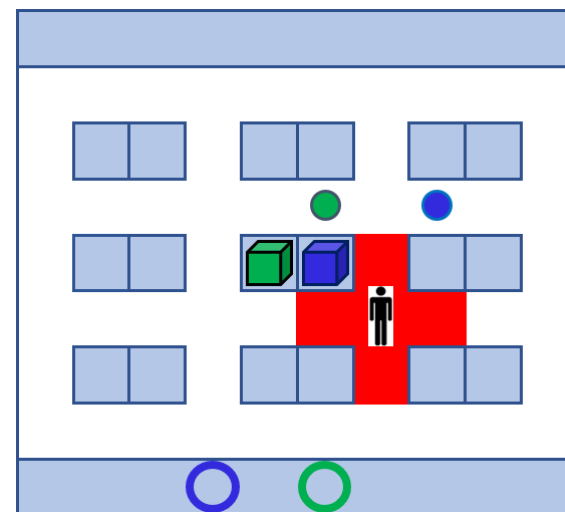
### ■方法2

：ロボットが人間に一定の距離以上近づかない。  
(進入を禁止する範囲を動的に変化させる。)

・制御要求を拡張する。

人間とロボットが近づく距離を変数で置く。  
→さまざまなスペックのロボットに対応可能になる。

$d=1$ のとき



■ ロボットの進入を禁止する範囲

### 具体的な計算手順

1. ロボットと人間のモデルを作成
2. 制御要求を作成 (同時に同じマス+人間の周りに存在しない)
3. オートマトンを同期合成し, 制御対象を作成
4. 制御対象が制御要求を満たすようなスーパーバイザを作成する

## 安全を確保する方法②

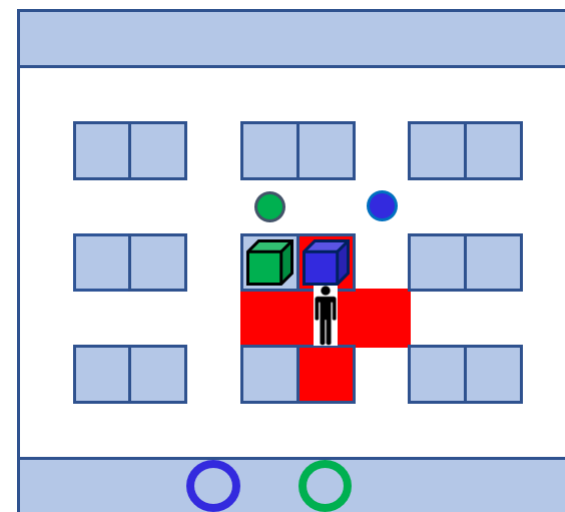
### ■方法2

：ロボットが人間に一定の距離以上近づかない。  
(進入を禁止する範囲を動的に変化させる。)

・制御要求を拡張する。

人間とロボットが近づく距離を変数で置く。  
→さまざまなスペックのロボットに対応可能になる。

$d=1$ のとき



■ ロボットの進入を禁止する範囲

### 具体的な計算手順

1. ロボットと人間のモデルを作成
2. 制御要求を作成 (同時に同じマス+人間の周りに存在しない)
3. オートマトンを同期合成し, 制御対象を作成
4. 制御対象が制御要求を満たすようなスーパーバイザを作成する

## 安全を確保する方法②

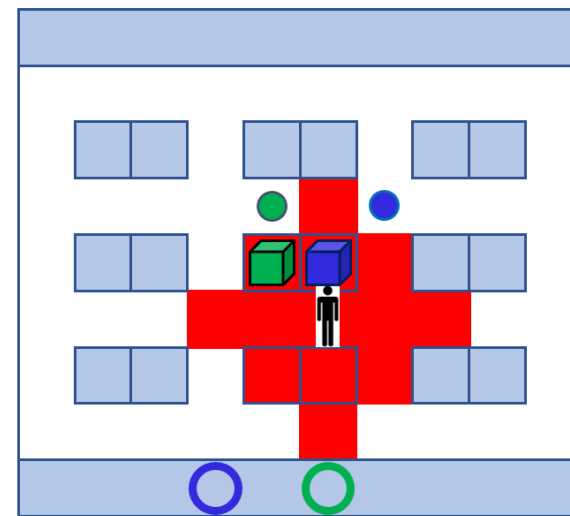
### ■方法2

：ロボットが人間に一定の距離以上近づかない。  
(進入を禁止する範囲を動的に変化させる。)

・制御要求を拡張する。

人間とロボットが近づく距離を変数で置く。  
→さまざまなスペックのロボットに対応可能になる。

$d=2$ のとき



■ ロボットの進入を禁止する範囲

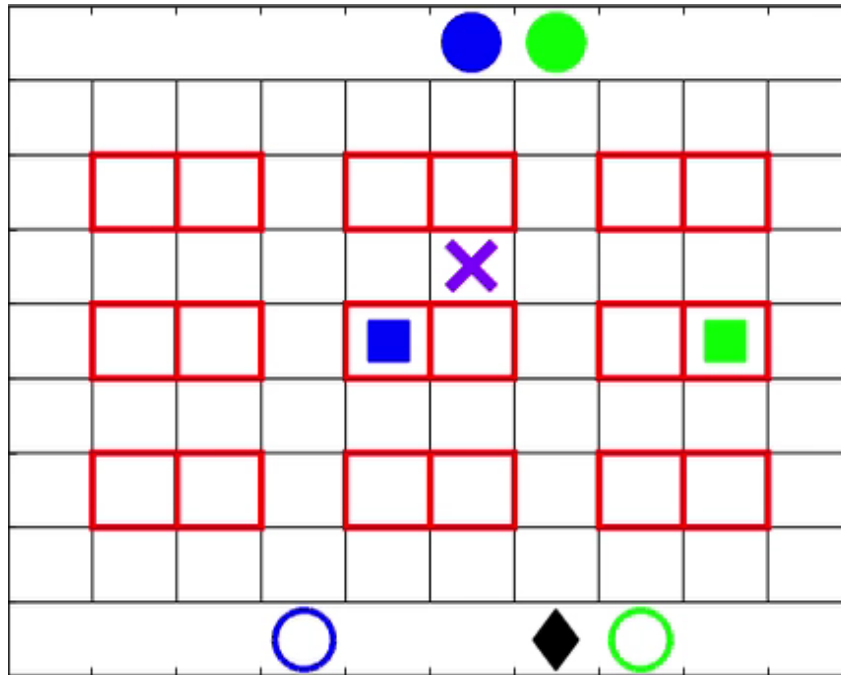
### 具体的な計算手順











1. ロボットと人間のモデルを作成
2. 制御要求を作成 (同時に同じマス+人間の周りに存在しない)
3. オートマトンを同期合成し, 制御対象を作成
4. 制御対象が制御要求を満たすようなスーパーバイザを作成する

# シミュレーション

棚から通路に落ちた商品を人間が棚に戻す場合

(方法2：ロボットが人間に一定の距離以上近づかない)



-  商品が置いてある棚
-  ロボットG1
-  ロボットG2
-  人間
-  G1が積む荷物  
(人間の助けが必要)
-  G2が積む荷物
-  G1のゴール
-  G2のゴール
-  トラブルの場所
-  ロボットの進入禁止エリア



# まとめ と これからの課題

---

## まとめ

方法1 : 人間の経路上にロボットを進入させない方法

方法2 : ロボットを人間に近づかせない方法

これら方法で人間の安全を確保できる.

ロボットの動作中に人間が倉庫内に入れるようになり, 作業効率が上がる.

## これからの課題

- ・ 今回紹介したケースをオンライン制御で解決できるようにする
- ・ ロボットと人間の数を増やす