

離散制御器合成によるサービスロボット向け行動計画自動生成手法

笠井栄良¹ 鄭顕志²

概要: 近年サービス分野への応用が期待されるモバイルマニピュレータ向けのソフトウェアシステムの課題の一つである行動計画・動作生成に関して、離散制御器合成により生成した制御器により動作計画を行う手法を提案した。さらにサービス用途向けのアプリケーションである World Robot Summit (WRS) フューチャーコンビニエンスストアチャレンジの補充・廃棄タスクを対象としてケーススタディを実施した。まず WRS の競技規則をもとにタスク分析・ユースケース抽出を実施し、LTS および FLTL による環境・ロボット・動作仕様の段階的なモデル化を実施し、MTSA により離散制御器の合成を行うことでスケーラビリティに関する評価を行った。結果、生成された離散制御器の有効性を確認することができたものの、現実的な規模のアプリケーションに対しては合成時間が増大するという課題があるという知見を得た。

Automated Motion Plan Generation for Service Robot by Discrete Controller Synthesis

TAKARA KASAI¹ KENJI TEI²

1. 背景

近年、サービスロボットの实用化に向けた開発が活発に行われている。特に生活分野や医療福祉分野への応用が期待されるモバイルマニピュレータ（移動台車に1つ以上のアームが搭載されたロボットの事、以下 MM と記載）への注目度は高く、これまで HSR[1], PR2[2]や TIAGo[3]等に代表される研究・開発用プラットフォームが製品化されてきた。ここ数年でアバターロボット ugo[4], 配膳・下膳ロボット ACUR-C[5]などの比較的少数の動作により構成される実アプリケーションを対象とした実証試験に向けた開発が行われており、事業化に向けた企業主体の開発が活発化している。今後は複数の動作を組み合わせた複雑なアプリケーションへの適用が加速すると考えられ、様々な動作を状況に合わせて選択するための行動計画技術を伴った自律型の MM への期待が高まっている。

2. 課題

自律的に動作する MM では一般に図 1 に示すようなソフトウェア構造が採用される。この構造は各種機能を提供するコンポーネントとコンポーネント間の通信を提供する通信ミドルウェアにより構成される。具体的には各種センサを利用して周囲の環境中の障害物・操作対象物を認識する環境認識、認識結果とロボットの身体モデルに基づいてよりロボットの詳細な動作を生成・実行する運動計画・制御、環境モデルと周囲の状況に応じて環境認識部および運動計画・制御部に要求を出す行動計画などのコンポーネン

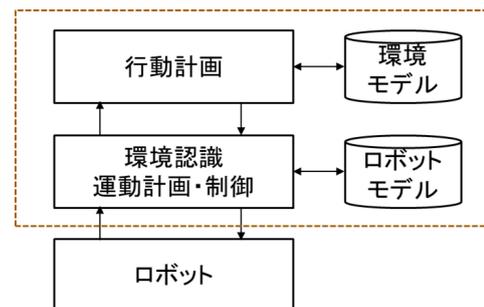


図 1 MM のソフトウェア構造 (概要)

Figure 1 Software Structure Overview for MM

トにより構成される。このうちコンポーネント間の通信機能を提供する通信ミドルウェアや各種機能を実現するコンポーネントにおいては事実上の標準といえるものが開発されている。例えば通信ミドルウェアとしてはロボット向け汎用通信ミドルウェア OSS である ROS[6]や ROS2[7]を採用する例が多い。ロボットの身体的モデル表現としては urdf や sdf 等のフォーマットが広く利用されている。また運動計画・運動制御を実現するソフトウェアとしては MoveIt![8]および ros_control [9]が事実上の標準である。これらを活用することにより多様な身体性をもつ MM を統一的方法で扱うことが可能となっている。

一方で自律型の MM を実現するための行動計画に関する技術としては自動プランナや離散制御器合成技術[10]などが選択肢として挙げられる。しかしながら具体的なコンポーネントとしては rosplan [11] や SkiROS [12] といった自動プランナに由来するものに限定され、離散制御器合成技

1 ソニーグループ株式会社

2 早稲田大学

術の応用は筆者の知る限り活発ではない。そのため、ROS/ROS2 を前提とした場合の行動計画手段の選択肢は限定的であり、サービス分野を対象としたモバイルコンピュータ向けの実アプリケーションを想定した適用事例・検討事例も少ない。また自律型の MM では周囲の環境や状況に応じて適切に動作を選択し実行する行動計画機能が重要である。特にサービスロボットは複雑な環境や状況の下で動作するため、網羅的に仕様を満たす行動計画機能の開発が困難であり課題であった。

3. 目的

本研究ではサービスロボット、特に自律型 MM 向けの行動計画手法に対し離散制御器合成[13]を導入することで、行動計画を自動生成する手法を提案する。特に実サービスロボットアプリケーションを対象としたケーススタディを通じ、適用可能性やスケーラビリティを評価する。これにより、離散制御器合成技術のサービスロボットへの応用に関する知見を得ることを目的とした。

4. アプローチ

離散制御器合成の概要を説明したうえで、当該技術をサービスロボットのソフトウェアシステムに導入する方針について説明する。

4.1 離散制御器合成

離散制御器合成を行うツールである MTSA (Modal Transition System Analyzer) は動作環境や制御対象の状態遷移モデルと要求仕様記述から、想定される状況下で網羅的に要求仕様を満足する様な制御器を自動生成する技術である。具体的には環境モデルと呼ばれる動作環境や制御対象(ロボットなど)の状態遷移を表す LTS (Labeled Transition System) 記述と要求仕様を表す FLTL (Fluent Linear Temporal Logic) 記述による時相論理式を入力する(図 2)。その結果 MTSA は当該環境モデルの状態遷移に関して想定されるいかなる状況下においても要求仕様を満足する様な状態遷移制御器を生成する。

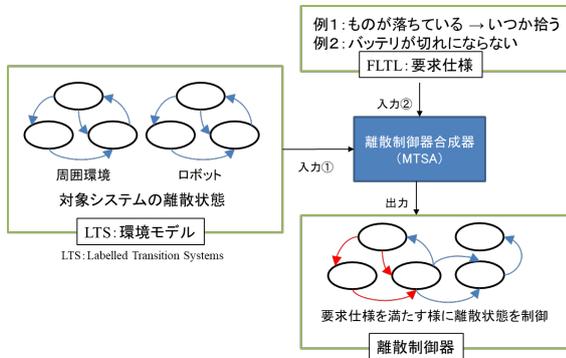


図 2 離散制御器合成器 MTSA

Figure 2 MTSA: Discrete Controller Synthesis Tool

4.2 ロボットシステムへの離散制御器合成の導入

離散制御器合成技術を自律型 MM のソフトウェアシステムへ導入した場合の構成を図 3 に示す。ロボットが観測可能な周辺環境とロボットが実現可能な動作を環境モデルとして LTS によりモデル化する。さらにロボットに実現してほしい振る舞いを要求仕様モデルとして FLTL により記述する。これらを MTSA に入力し、得られた離散制御器の仕様と環境モデルの観測・動作仕様を満たす様にロボットシステムが振舞うことで、指定された要求を満たすロボットを構成することが可能となる。

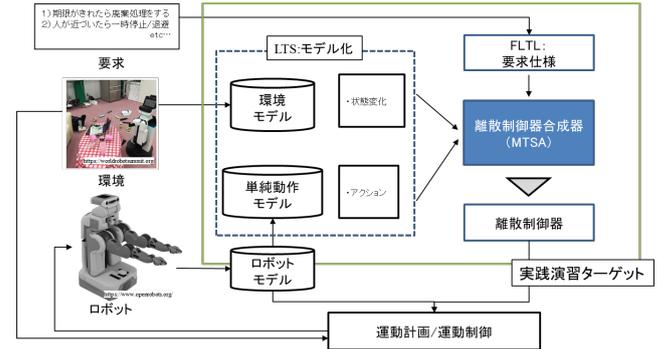


図 3 MTSA のロボットシステムへの適用

Figure 3 Application of MTSA to Robot System

5. 対象アプリケーション

5.1 World Robot Summit

World Robot Summit [14]はロボットの活躍が期待される様々な分野において世界中から終結したチームがロボットに関する技術やアイデアを競う競技大会 (WRC) と展示会 (WRE) から構成されるイベントである。WRC ではものづくり、サービス、インフラ・災害対応などの4つのカテゴリに関して今後実現が期待される製品組立、介助、コンビニの軽作業、プラント災害予防・トンネル事故災害対応などの各種アプリケーションが競技対象として設定されている。このうち本研究では特に MM の活用が期待されると考えられるサービスカテゴリのフューチャーコンビニエンスストアチャレンジ (FCSC) をケーススタディの対象とした。

5.2 フューチャーコンビニエンスストアチャレンジ

FCSC はロボット技術による従業員負担の軽減と消費者に対して新たなサービスを提供する未来のコンビニエンスストアを実現することを目的とした世界初の競技である。コンビニエンスストア従業員が行う作業のうち、食品など複数種類の商品の品出し・入替、客や従業員とのインタラクション、トイレの掃除を対象として、1) 陳列・廃棄タスク、2) 接客タスク、3) 清掃タスクの3つのタスクが設定されている。特に今回は MM を特徴づけるマニピュレーション要素および移動要素の両者が最も活用されるタス

クをケーススタディの対象とし陳列・廃棄タスクに着目しケーススタディを実施することとした。

5.3 模擬店舗環境

陳列・廃棄タスクを含むすべてのタスクは実店舗を意識して設計された模擬店舗環境により実施される。模擬店舗レイアウトに記載された模擬店舗環境を図 4 および図 5 に示す。



図 4 FCSC フィールドコンセプトイメージ
Figure 4 FCSC Field Concept Image

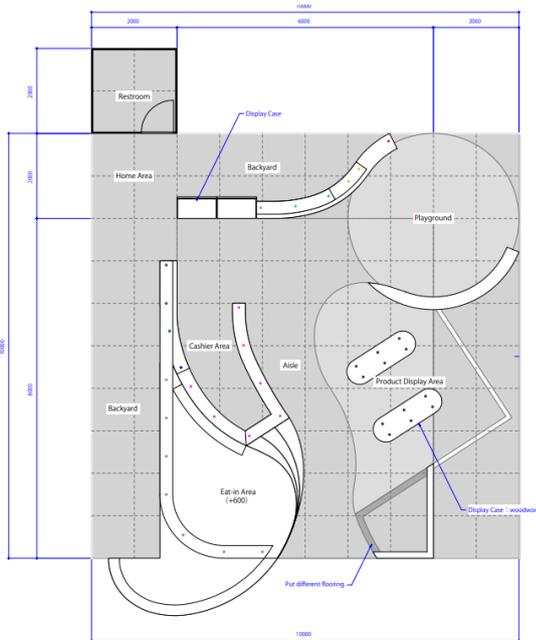


図 5 模擬店舗平面図
Figure 5 Floor plan of simulated store

5.4 陳列・廃棄タスク

陳列・廃棄タスクでは、コンビニエンスストアで扱われる日常商品の自動補充およびフェイスアップと賞味期限切れ商品の自動廃棄を行う。対象とする商品は 6 種 20 個で識別番号 (ID) が設定されている (表 1)。また本タスクは模擬店舗に設定されたバックヤードエリア、ホーム、通路、陳列棚を使用し、ホームエリアに置かれたコンテナ内に収められている商品を陳列棚の所定の位置に並べる“陳列”，あらかじめ陳列棚に並べられている商品を整頓する“整列”，廃棄品を回収してホームまで運ぶ“廃棄”と呼ばれる作業

に分解される。またロボットは顧客の動作を妨げない様にふるまわなければならない。ロボットの周囲 1 m に接近した場合には作業の停止、音声による報知、空きスペースへの退避等の回避動作を実施する必要がある。

表 1 FCSC で使用されるオブジェクト

Table 1 Set of Objects in FCSC.

対象物名	個数
Plum rice balls	4
Salmon rice balls	4
Sandwiches	3
Deli Food Pouches	3
Drinks	3
Lunch Boxes	3

6. モデル化

本タスクに離散制御器合成を適用するにあたって実施したモデル化の詳細について説明する。陳列・廃棄タスクの競技詳細を分析し、ユースケース図とユースケース記述を作成したので説明する。そのうえでロボット周囲の環境モデルおよびロボットの動作モデル、動作仕様モデル記述と記述の際に捨象した要素についても記載する。

6.1 ユースケース

陳列・廃棄タスクのユースケース図を図 6 に示す。アクタとして顧客と店員が存在し、顧客は商品をかごに入れる動作と買い物かごを渡してレジで決済する動作を行う。一方で店員 (ロボット) は商品棚に商品を補充する動作、消費期限の切れた商品を回収・廃棄する動作、商品をフェイスアップして整列する動作、コンテナに商品を出し入れする動作を想定している。このうちコンテナに出し入れする動作は補充・廃棄いずれの動作にも表れることから包含関係にある。

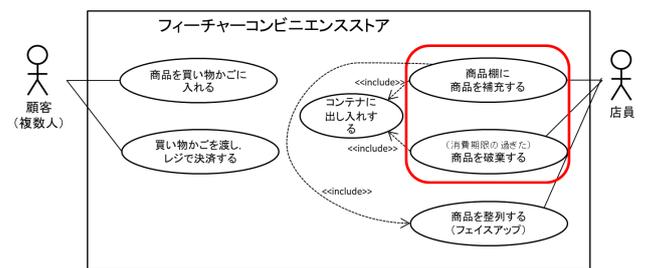


図 6 陳列・廃棄タスクユースケース

Figure 6 Use Case of Display and Disposal Tasks

本研究ではコンテナへの出し入れは対象物の物体認識や位置・形状認識および把持位置計画・運動計画等の複雑な動作シーケンスを伴うことから離散制御器による状態制御からは切り離して扱うべきと考え捨象した。扱う商品の種類に関しても同様に、状態制御ではなく運動計画や運動

制御のコンポーネントが扱うべき内容であると考え捨象した。また商品を整列する動作に関してもモデル化を簡略化するために以降のモデル化の対象からは省いた。以上を踏まえ、ケーススタディではロボットに関する4つの動作のうち特に補充・廃棄に関する2つの動作についてモデル化と離散制御器合成を適用した。以降はこの2つの動作(図8の赤枠部)に絞って説明する。

6.2 ユースケース記述

商品の補充と廃棄に関するユースケース記述を図7および図8に示す。捨象した動作に関連する記述は灰色表記にしている。

- ユースケース名: 商品棚に商品を補充する
- 起動アクタ: ロボット(サービスロボット)
- 事前条件: いずれかの商品棚に欠品が存在する
- 事後条件: すべての商品棚に商品が補充されている
- 基本系列:
 1. ロボットは補充する商品が入ったコンテナをもってバックヤードを出る
 2. ロボットは商品棚前に移動する
 3. ロボットはコンテナを商品棚横の床に置く
 4. ロボットは商品をコンテナから取り出して補充し、商品の整列(フェイスアップ)を行う
 5. 2~4を繰り返すすべての商品を補充する
- 例外系列:
 1. 基本系列4.において、顧客が接近(1m以内)した場合には、作業を一時停止する。(音声で報知する)。
 2. 例外系列1.において、作業の一時停止の時間が3秒を超えたら、顧客に場所を明け渡すように退避する。
 3. 例外系列2.において、退避後に顧客がいなくなったら作業を再開する。

図7 商品の補充に関するユースケース記述

Figure 7 Use Case Description for Display Task

- ユースケース名: (消費期限の過ぎた)商品を破棄する
- 起動アクタ: ロボット(サービスロボット)
- 事前条件: いずれかの商品棚に消費期限切れの商品が存在する
- 事後条件: すべての商品棚から消費期限切れの商品が撤去されている
- 基本系列:
 1. ロボットは補充する空のコンテナをもってバックヤードを出る
 2. ロボットは商品棚前に移動する
 3. ロボットはコンテナを商品棚横の床に置く
 4. ロボットは商品を賞味期限切れの商品を棚から取り出す(コンテナに入れる)
 5. 2~4を繰り返すすべての商品を補充する
- 例外系列:
 1. 基本系列4.において、顧客が接近(1m以内)した場合には、作業を一時停止する。(音声で報知する)。
 2. 例外系列1.において、作業の一時停止の時間が3秒を超えたら、顧客に場所を明け渡すように退避する。
 3. 例外系列2.において、退避後に顧客がいなくなったら作業を再開する。

図8 商品の廃棄に関するユースケース記述

Figure 8 Use Case Description for Disposal Task

6.3 環境モデル

陳列・廃棄タスクで利用する環境構造物はバックヤード、棚、通路の3種で構成されていると考えることができ、図9に示すような多重度関係をもつ。



図9 環境要素の多重度関係

Figure 9 Relationship of environmental elements

これを踏まえて離散制御器合成を導入に関して段階的に環境モデルを複雑化しつつ評価を実施可能であるようにモデル化をおこなった。具体的には通路数を変化させることによりMMの台車移動範囲を調整できるようにした。また同一の通路に設置される棚の数を変化させることで通路の両側に棚が設置されている様な実環境に近い構成を設定できるようにした。陳列・廃棄タスクに関するLTSによる環境モデル記述については棚(SHELF)、通路(AISLE)、バックヤード(BACKYARD)に分割し表現した。

6.3.1 棚(SHELF)のLTS記述

棚は図10に示すようにEMPTY、NOT_EMPTYの2つの離散状態を持つ。EMPTYは商品の補充が必要な状態、NOT_EMPTYは十分な数の商品が補充されている状態をそれぞれ意味する。また顧客による商品取得や期限切れの発生をそれぞれイベントtake_outおよびtime_elapsedで表現している。MMが直接関与する廃棄処理の実行と商品補充はそれぞれ動作の成否情報を含むrefill_success、refill_error、dispose_success、dispose_errorの4種のイベントを用意した。

```

SHELF = EMPTY,
EMPTY = (
  refill_error -> EMPTY
| refill_success -> NOT_EMPTY
| dispose_error -> EMPTY
),
NOT_EMPTY = (
  take_out -> EMPTY
| refill_error -> NOT_EMPTY
| time_elapsed -> NOT_EMPTY
| dispose -> dispose_success -> EMPTY
).
  
```

図10 棚(SHELF)のLTS記述

Figure 10 LTS Description for the Shelf

6.3.2 通路 (AISLE) の LTS 記述

通路のモデルを図 1 1 に示す. 当該通路に設置された棚の数をパラメータとして受け取ることで設置されたすべての棚の補充 (refill) 及び廃棄 (dispose) のイベントと関連付けられる. 一方, 離散状態としてロボットが当該通路を占有しているか否かを EMPTY および OCCUPIED の 2 種の状態として持つ.

```

AISLE(N=1) = EMPTY,
EMPTY = (
    arrived -> OCCUPIED
),
OCCUPIED = (
    Left -> EMPTY
| shelf[1..N].refill -> OCCUPIED
| shelf[1..N].dispose -> OCCUPIED
).
    
```

図 1 1 通路 (AISLE) の LTS 記述
Figure 11 LTS Description for the AISLE

6.3.3 バックヤード (BACKYARD) の LTS 記述

バックヤードのモデルは図 1 2 の様にロボットの存在を示す EMPTY および OCCUPIED の 2 つの状態を持つ. また初期状態においてロボットはバックヤードにて待機しているため初期状態は OCCUPIED である. 関連するイベントはロボットがバックヤードに入る arrived_BY とバックヤードから退去する left_BY の 2 種のみである.

```

BACKYARD = OCCUPIED,
EMPTY = (
    arrived_BY -> OCCUPIED
),
OCCUPIED = (
    left_BY -> EMPTY
).
    
```

図 1 2 バックヤード (BACKYARD) の LTS 記述
Figure 12 LTS description for the Backyard

6.4 ロボットモデルの LTS 記述

ロボットモデルには MM の台車移動に関するイベントが含まれる. ロボットが通路ないしバックヤードのどこに存在しているかをロボットモデルの離散状態として設定した. そのため通路数に比例した離散状態をもつ. 例えば通路数が 2 の場合にはバックヤードを含めて 3 つの離散状態を持つことになる. そのうえでそれぞれの位置 (通路およびバックヤード) において MM がいずれの通路に移動可能なのかをイベントにより定義することで模擬店舗の環境を表現した. したがって模擬店舗の構成ごとにロボットモデ

ルを定義する必要がある.

なお顧客とのすれ違いや環境との衝突などの動作は運動計画や運動制御の責務であることからロボットモデルからは捨象している. それ以外のユースケース記述から MM の動作計画に必要な動作である商品の補充・廃棄, 退避動作を MM が通路に位置している場合のイベントとして定義している. また簡単のために商品が消費期限か否かはロボットが即時観測できるものとしている.

6.4.1 通路数 1 の記述例

模擬店舗を通路数 1 でモデル化した場合の概念図を図 1 3 に示す. この構成のロボットモデルに関する LTS 記述例を図 1 4 に示す. 通路数とバックヤード数分の離散状態があるので MM がバックヤードに位置する状態 (AT_BY) および通路に位置する状態 (AT_A1) の状態がある. MM がバックヤードに位置する場合, 顧客がバックヤードに入ることにはないと期待できることから MM はバックヤードから通路に移動する行動のみが可能である. そのため通路に移動を開始 (go_A1), バックヤードから退去 (left_BY), 通路に到着 (aisle[1].arrived) のイベントのみを定義している. 一方 MM が通路に位置している場合に実行可能な行動は 1) バックヤードへの退去, 2) 棚に関する動作, 3) 顧客検知時の例外動作の 3 種である. バックヤードへの退去に関連するイベントはバックヤードへの移動開始 (go_BY), 通路からの退去 (aisle.left), バックヤードへの到着 (arrived_BY) である. また棚に関する動作としては商品の補充 (refill, refill_success, refill_fail) と廃棄 (dispose, dispose_success, dispose_fail) があり, ユースケース記述の例外系列に関するイベントとしては顧客への報知・中断・退避・待機開始 (それぞれ alert_human, suspend, evacuate, timer_start) がある. また MM が実行可能なイベントではないものの検知可能 (uncontrollable) なイベントとして例外系列に関連したイベントを定義している. 顧客の接近を検知 (detect_human_appeared), 顧客の退去を検知 (detect_human_disappeared), 待機時間 (3 秒) の経過 (timer_3sec_expired) の 3 つのイベントがこれに該当する.

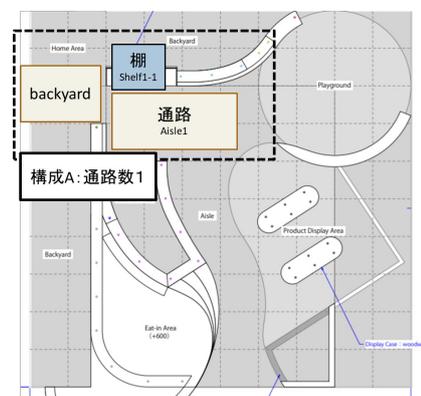


図 1 3 構成例 (通路数 1)
Figure 13 Sample Configuration: One Aisle

```

ROBOT_AISLE1(SNUM=1) = AT_BY,
AT_BY = (
  /* controllable events */
  robot.go_A1
  -> left_BY -> aisle.arrived -> AT_A1
),
AT_A1 = (
  /* controllable events */
  /* 1) move to the backyard */
  robot.go_BY
  -> aisle.left -> arrived_BY -> AT_BY
  /* 2) action to the shelf */
  | aisle.shelf[i:1..SNUM].refill
  -> (
    aisle.shelf[i].refill_success -> AT_A1
    | aisle.shelf[i].refill_fail -> AT_A1)
  | aisle.shelf[i:1..SNUM].dispose -> (
    aisle.shelf[i].dispose_success -> AT_A1)
  /* 2) action for exception behavior */
  | robot.alert_human -> AT_AISLE[no]
  | robot.suspend -> AT_AISLE[no]
  | robot.evacuate -> AT_AISLE[no]
  | robot.timer_start -> AT_AISLE[no]
  /* uncontrollable events */
  | robot.detect_human_appeared -> AT_AISLE[no]
  | robot.detect_human_disappeared -> AT_AISLE[no]
  | robot.timer_3sec_expired -> AT_AISLE[no]
).

```

図 14 ロボットの LTS 記述 (通路数 1 の場合)

Figure 14 LTS Description for Robot: One Aisle

6.5 環境全体に関するモデルの LTS 記述

棚, 通路, バックヤードおよびロボットのモデルを合成して環境全体モデルを構築する LTS 記述を図 15 に示す。

```

||ENV_WRS_ROBOT1 =
(
  (aisle:(shelf[1..SNUM]:SHELF || AISLE)) ||
  BACKYARD ||
  ROBOT_AISLE1(1)
).

```

図 15 環境モデル全体に関する LTS 記述

Figure 15 LTS Description for the Entire Environment Model

6.6 動作仕様の FLTL 記述

ユースケース記述により抽出された MM が満たすべき以下の 5 つの仕様を抽出し, FLTL により制御器の要求仕様を記述した。この時 1), 2) に関しては安全性特性とし

て, 3), 4), 5) に関しては活性特性として表現した。

- 1) いつか必ず商品は補充される
- 2) 既に商品がある棚に過剰に補充はしない
- 3) 消費期限切れの商品は破棄される
- 4) 付近に人を検知したら作業を中断・報知し, 人の退去を待って作業を再開する
- 5) 4. で人が退去しない場合には退避し, 再度人の退去を待って作業を再開する

7. 評価と考察

離散制御器合成をロボットのシステムに導入し行動計画機能として利用する場合, あらかじめ環境モデルと動作仕様記述とから離散制御器を生成する。離散制御器は環境モデルに基づき, 状況に応じた適切なイベントを即時出力する。これはシステム稼働中に実施する行動計画において探索時間は必要とされないことを意味する。一方で離散制御器を生成する際に環境モデルや動作仕様モデルの複雑さ(状態数)に応じた合成時間が必要とされる。これを踏まえて段階的に環境モデルの状態数を増加させた場合の合成時間の変化を観察することでスケーラビリティの観点での評価を試みた。また評価を行った環境を表 2 に示す。

表 2 評価環境

Table 2 Evaluation Environment

項目	Version
離散制御器合成ツール	MTSA Latest at 2020/11/05
OS	Windows 10 バージョン 1909
プロセッサ	Intel Core i7-9750 2.60 GHz
メモリ	32 GB

7.1 評価対象モデル

通路数と通路数当たりの棚数を変化させることで環境モデルの複雑さを変化させた。具体的には, 通路数あたりの棚数を 1 および 2 とし, 通路数を 1~5 まで段階的に変化させた環境モデルを構築し, 離散制御器合成を実施した。構成例として通路数が 2 までの通路数と通路数当たりの棚数の組み合わせを表 3 に示す。このうち通路数当たりの棚数が 1 の場合の模擬店舗の平面図イメージを図 13 に, 通路数当たりの棚数が 2 の場合のイメージを図 14 に示す。

表 3 環境モデルの変化例

Table 3 Example of Environmental Models

構成	通路数	通路数当たりの棚数
構成 A	1	1
構成 B	2	1
構成 C	1	2
構成 D	2	2

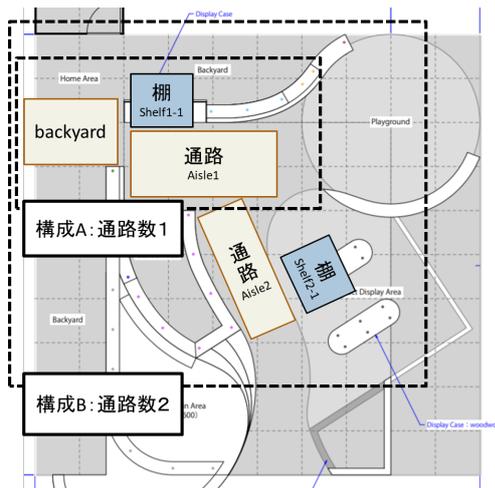


図 16 環境モデルにおける通路数変化例

Figure 16 Environmental Model Changes: number of Aisles

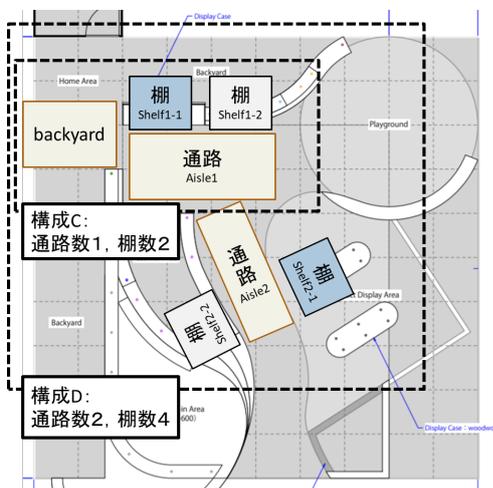


図 17 環境モデルにおける通路数・棚数変化例

Figure 17 Model Changes: number of Aisles and Shelves

7.2 評価結果

7.2.1 通路数あたり棚数1のモデル

通路数あたりの棚数1のモデルについて、離散制御器合成ツールが扱う状態数と制御器を合成するのに必要とした時間を表3に示す。通路数1の場合に146である状態数は通路数変化に伴い増加し通路数5では9299に達することがわかる。この時、図18に示すように通路数の変化に対して状態数が指数関数的に増加していることがわかる。また合成時間も同様に増加し通路数が5の場合に10819秒に達する。

7.2.2 通路数あたり棚数2のモデル

通路数あたりの棚数2のモデルについて、離散制御器合成ツールが扱う状態数を表4に示す。離散制御器合成については通路数が1の場合は0.338秒程度で合成ができたものの、通路数2の時点で状態数が10752と棚数1で通路数が5の場合よりも状態数が多くなる。これを受けて通路数

2以上の場合には数時間以内での生成ができなかったため結果を省略している。

表4 状態数および合成時間：通路数あたりの棚数1

Table 4 States and Synthesis Time: One Shelf per Aisle

通路数	状態数	合成時間 [sec]
1	146	0.053
2	372	0.75
3	1344	50.29
4	3488	3245.85
5	9299	10819.23

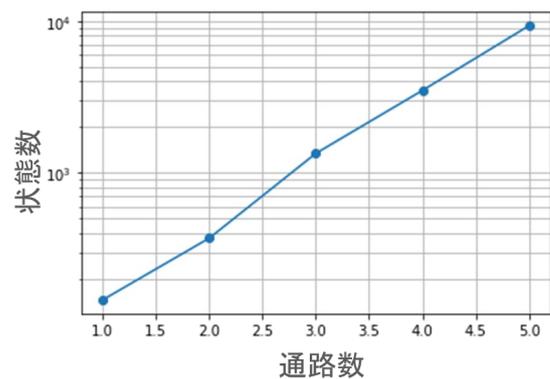


図 18 通路数に対する状態数の変化

Figure 18 Number of States with Respect to Number of Aisles

表5 状態数および合成時間：通路数あたりの棚数2

Table 5 States and Synthesis Time: Two Shelves per Aisle

通路数	状態数	合成時間 [sec]
1	392	0.338
2	10752	-
3	243712	-
4	1004346	-

7.3 考察

離散制御器合成は内部的にモデル解析・検査を実施することから、状態数に対して指数関数的に計算時間が増大することが推測できるのに対し、今回の評価においても通路数や棚数を増加させることによって同様に計算時間が増加する傾向にあることが分かった。さらに、棚数、通路数がそれぞれ1台、4本の比較的単純な構成であっても合成時間が3000[sec]を超えることから、より現実的なアプリケーションの規模に対して数時間程度の現実的な時間で生成を実施することは困難であると考えられる。このことからスケーラビリティの観点で課題があり状態空間や合成時間を削減するための工夫や妥協が必要である。

さらに今回はMMが1台であるという前提でケーススタディを実施したが、サービス分野での実アプリケーションでは複数台のロボットを扱うことも十分考えられる。その

場合扱うべき状態数及び合成時間はさらに増大すると考えられる。

8. まとめと展望

本研究では MM 向けのソフトウェアシステムの課題の一つである行動計画に関して、離散制御器合成により生成した離散制御器を動作計画に用いる手法を提案した。さらにサービスロボット向けのアプリケーションである WRS の補充・廃棄タスクを対象としたケーススタディを実施した。まず WRS の競技規則からタスク分析・ユースケース抽出を実施し、その結果をもとに LTS および FLTL による環境・ロボット・動作仕様のモデル化を実施した。当該モデルは通路数や柵数を可変できるように構成し、段階的にモデルを複雑化しつつ離散制御器合成実施することで、スケラビリティの観点から提案手法の評価を実施した。その結果、生成された離散制御器の有効性を確認することができたものの、現実的な規模のアプリケーションに対して適用に対しては合成時間が増大するという課題があることが判明した。

現状のモデル化のアプローチでは現実的な時間での離散制御器の合成が期待できない。そのため、今後はより状態数や遷移数を抑制可能なモデル化方針の検討や対象とする領域毎に制御器を分割する検討などを実施し、現実的な問題に適用するための手段を検討したい。

また MTSA に限らず新しい技術をロボットシステムに統合するうえで、ロボット向けシステムソフトウェアのデファクトスタンダードである ROS/ROS2 との統合は避けては通れない。しかしながら現状では、MTSA により生成された離散制御器を ROS/ROS2 上で扱うフレームワークは十分には構築されておらず、また離散制御器の導入を想定したソフトウェア設計されていないことも多い。そのため、業務への活用にはフレームワークの構築や既存ソフトウェア構造の見直しを実施することによりこれらの解消を図らなければならないと考える。

参考文献

- [1] Yamaguchi, U., Saito, F., Ikeda, K., Yamamoto, T.: Hsr, human support robot as research and development platform. In: The Abstracts of the International Conference on Advanced Mechatronics: Toward Evolutionary Fusion of IT and Mechatronics: ICAM 2015.6, pp 39–40. The Japan Society of Mechanical Engineers (2015)
- [2] CLEARPATH, “PR2”, <https://support.clearpathrobotics.com/hc/en-us/categories/200217239-PR2>
- [3] Jordi Pages, Luca Marchionni, and Francesco Ferro, ‘Tiago: the modular robot that adapts to different research needs’, in Proc. of the IROS Workshop on Robot Modularity, (2016).
- [4] ugo Inc., “avator robot ugo”, <https://www.ugo.plus/#about>
- [5] Smile Robotics, “ACUR-C”, <https://www.smilerobotics.com/>
- [6] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. B. Foote, J. Leibs, R., Wheeler, and A.Y. Ng, “ROS: An open-source robot operating system,” in Proc. ICRA Open-Source Softw. Workshop, 2009.
- [7] A. Barciś, M. Barciś and C. Bettstetter, "Robots that Sync and Swarm: A Proof of Concept in ROS 2," 2019 International

- Symposium on Multi-Robot and Multi-Agent Systems (MRS), New Brunswick, NJ, USA, 2019, pp. 98–104, doi: 10.1109/MRS.2019.8901095.
- [8] Chitta, S., Sucas, I., Cousins, S (2012) MoveIt![ROS topics]. IEEE Robotics and Automation Magazine 19(1): 18–19.
- [9] Chitta S, Marder-Eppstein E, Meeussen W, Pradeep V, Rodríguez Tsouroukdissian A, Bohren J, Coleman D, Magyar B, Raiola G, Lüdtke M, Fernández Perdomo E (2017) ros_control: a generic and simple control framework for ros. J Open Source Softw.
- [10] V. Braberman, N. D'Ippolito, N. Piterman, D. Sykes and S. Ucritei, "Controller synthesis: From modelling to enactment," 2013 35th International Conference on Software Engineering (ICSE), San Francisco, CA, 2013, pp. 1347-1350, doi: 10.1109/ICSE.2013.6606714.
- [11] Cashmore, M., Fox, M., Long, D., Magazzeni, D., Carrera, A., Palomeras, N., Hurtos, N. and Carreras, M., ROSPlan: Planning in the Robot Operating System, Proceedings of the 25th International Conference on Automated Planning and Scheduling (ICAPS 2015)
- [12] Roviada, F., Crosby, M., Holz, D., Polydoros, A.S., Großmann, B., Petrick, R.P.A., Krüger, V.: SkiROS—a skill-based robot control platform on top of ROS. In: Koubaa, A. (ed.) Robot Operating System (ROS). SCI, vol. 707, pp. 121–160. Springer, Cham (2017).
- [13] N.D'Ippolito et.al., Synthesis of Live Behaviour, pp.77-86, FSE2010
- [14] World Robot Summit, “Feature Convenience Store Challenge”, <https://worldrobotsummit.org/wrs2020/challenge/service/fcsc>