

東京大学
情報理工学系研究科 創造情報学専攻
博士論文

長期経験に基づくタスク具体化機能を備えた
日常生活支援ロボットシステムの構成法
(System Architecture for Home Service Robots based on
Task Instantiation using Long-term Experience)

古田 悠貴

指導教員 稲葉 雅幸 教授

2018年12月

概要

日常生活支援ロボットシステムには予め与えられた知識をもとに自律的に計画・実行する機能に加えてロボットが自律的に各環境に固有の知識を経験的に獲得し、タスクを環境に適応的に変えていく仕組みが必要である。というのも、日常生活におけるタスクは目的そのものが多様であるだけでなく、目的を達成するための手段や目標状態自体が、環境や環境に共生する人の知識や経験によって定義されるからである。従来では、日常生活タスクは実世界の環境と一致するよう具体的、常識的で環境に依存した知識を用いて注意深く書き下された固定的な記述を用いて表現されていた。本論文では、事前に具体的に定義することが困難な日常生活タスクを、環境非依存の知識を用いた記述によって与え、ロボットがタスクを行う環境において経験的に獲得した知識を用いて各環境における局所的な合理性を満たした具体的なタスクを自律的に計画・実行することが可能なロボットシステムの構成法について論じる。

このようなロボットシステムを構成する上で必要な基盤的要素として、ロボットが環境においてタスクを行っていくうちに長期的に経験を蓄積する機構の必要性、および要求機能について議論する。蓄積した経験から、環境に非依存なタスクを環境に適した記述へ具体化するための環境における局所合理性を満たす知識を自律的に抽出する機能を構成するための要素の分類と必要な学習手法・学習モデルについての考察を行い、実際のロボットに提案システムを搭載し、これらの知識の自律的獲得と環境に適したタスクの計画・実行が可能であるシステムとなっていることを示す。

さらに、日常生活環境におけるロボットシステムに要求される構成要素として、共生する人からの指示を随時受け付け対応する機能についても論じる。日常生活環境はロボット自身や共生する人など様々な要因で常に変化するため、失敗可能性を考慮するロボットシステムであることが求められる。このとき、日常生活環境に習熟した人から随時指示やロボットの判断の修正を受け付ける機能はロボットのタスク成功率を向上するだけでなく、人にとって安全・安心感を与える上で必須となる。また長期経験からの自律的獲得によって得た知識は、予め与えられた知識に対して常に誤りを含む可能性があるため、ロボットが判断の修正指示を随時受け付けられるシステムとなっていることは有用である。このようなロボットに対して判断の修正を人が指示する際には意思伝達が正確に行われる必要があるが、人とロボットにおける指示の伝達には常に曖昧性がつきまとう。本論文では人からの指示に含まれる曖昧性について分類し、ロボットや環境に関する知識を用いて曖昧性を解決する機能についても論じる。

本研究は、ロボットが長期的に蓄積した経験を用いて環境に固有の局所的な合理性に基づく知識を獲得し、環境に非依存な形で与えられた日常生活タスクを、環境に適応する具体的なタスクとして自律的に計画・実行し、人による随時指示可能なシステムの構成により、環境ごとに異なるタスクの内容を獲得し、実行していく自律適応的な日常生活ロボットシステムの構成法を示すものである。

目次

第 1 章	序論	1
1.1	本研究の背景と目的	1
1.2	本研究の特色	3
1.3	本論文の構成	3
第 2 章	長期経験に基づくタスク具体化機能を備えた日常生活支援ロボットシステム	6
2.1	はじめに	6
2.2	日常生活環境におけるタスク	6
2.3	日常生活タスクの自律計画実行システム	10
2.4	日常生活タスクの記述	12
2.5	環境における局所合理性に基づく知識	17
2.6	ロボットシステムにおける長期経験獲得機能	17
2.7	曖昧な指示・修正を随時受け付け対応可能な日常生活タスク実行システム	20
2.8	長期経験に基づくタスク具体化機能を備えた日常生活支援ロボットシステムの構成	23
2.9	おわりに	24
第 3 章	日常生活支援タスクを行う自律ロボットシステムの構成	26
3.1	はじめに	26
3.2	本研究で用いたロボットハードウェアとロボットシステム	26
3.3	日常生活支援ロボットのためのソフトウェア構成	35
3.4	おわりに	46
第 4 章	ロボットにおけるエピソード記憶に基づく長期経験の蓄積機構	47
4.1	はじめに	47
4.2	関連研究	47
4.3	時系列データの構造化	48
4.4	蓄積するデータの選択	49
4.5	蓄積するデータの保存	51
4.6	蓄積経験の抽出	57

iv 目次

4.7	エピソード記憶に基づく長期経験の蓄積機構を用いたロボットシステム評価 実験	57
4.8	おわりに	59
第 5 章	長期経験を用いた環境の局所合理性に基づく知識の自律獲得機構	61
5.1	はじめに	61
5.2	関連研究	61
5.3	人とスポットの相互作用に基づく定位置操作モデルの経験的獲得	64
5.4	アイテムとスポットの相互作用に基づく確率的物体配置地図の経験獲得	94
5.5	人とアイテムの相互作用に基づく嗜好モデルの経験的獲得	97
5.6	おわりに	99
第 6 章	環境に非依存な状況レベルタスク記述と局所合理性に基づく具体化機構	108
6.1	はじめに	108
6.2	関連研究	110
6.3	日常生活タスクの状況レベル記述	114
6.4	定位置操作モデルを用いた場所の局所合理性に基づく具体化	116
6.5	嗜好モデルを用いた人と物の局所合理性に基づく具体化	117
6.6	確率物体配置地図を用いた物・場所の局所合理性に基づく具体化	117
6.7	状況レベルタスク記述の具体化に基づくタスク計画実行実験	118
6.8	おわりに	122
第 7 章	曖昧な指示・修正を随時受け付け対応可能な日常生活タスク実行システム	127
7.1	はじめに	127
7.2	関連研究	128
7.3	曖昧性解消を行う音声対話を用いた自然言語によるタスク指示機能	130
7.4	随時指示・修正を受け付ける日常生活タスク実行システム	139
7.5	おわりに	154
第 8 章	日常生活タスクの局所合理性に基づく具体化機能を用いたタスク計画実行シ ステムの評価実験	155
8.1	はじめに	155
8.2	食事の準備タスクにおける局所合理性を用いた具体化タスクの計画・実行	155
8.3	おわりに	159
第 9 章	結論	168
発表論文		172
参考文献		175

付録 A	Appendix	189
A.1	PDDL を用いた日常生活支援行動計画のための環境状態記述	189

第 1 章

序論

1.1 本研究の背景と目的

ロボットは環境を認識するセンサと環境に作用するアクチュエータを備えた計算機プラットフォームである。また単語の意味について考える [1] と、今まで人間の行ってきた作業を代替、または協調的に行う機械であるという社会的性質を帯びる。実際にロボットは自動車や船舶の溶接作業、部品の組み立て・検品を始めとする作業を行い、実行可能な作業内容としては単純ながらも、人間を超える高精度かつ高速な生産を実現することにより商業的な成功を収めている。また最近では DARPA Robotics Challenge[2] や Google Lunar XPRIZE[3] といった人間が通常近づくことの難しい過酷な環境においてロボットが人間の代わりとなる実体として遠隔地から操作を送る人間の手足となり作業を行うテレオペレーションの本格的実用化を念頭においた大会が開催され、注目を浴びている。

日常生活環境においても、今まで人間の行ってきた洗濯や調理などの家事や高齢者などの生活支援タスクをロボットが代替して行うことが、少子高齢社会が問題提起されるに従って、大きく期待されている。

家庭環境においてタスクを行うロボット実現を目指したロボット競技である Robocup@Home[4] では、競技種目の設計指針として今までロボットが扱ってきたタスクと比較した家庭環境におけるタスクを以下のように特徴づけている。

- 完全な自律制御システムを要求する
- 人とロボットの自然なインタラクションがタスク目的達成に必須である
- 認知的なシステムを要求する
- アドホックな解決が不可能なほど多様である

このように複雑な環境においてタスクを遂行するためのロボット知能システムの研究が今まで数多く行われてきている。岡田らは川田工業社製等身大ヒューマノイドロボット HRP-2[5] を用いた皿洗いやお茶汲みタスク [6] を実現しモデルベースのロボットシステムにおける実世界ロボットのための認識行動システムを提案した。岩田らは早稲田大学において開発された等身大双腕ロボット TWENDY-ONE[7] を用いた調理支援タスクを用いて人との共生環境に馴

染むロボットをハードウェア・ソフトウェアの連携のもと提案した。また本田技研工業社製等身大ヒューマノイドロボット ASIMO[8] を用いたカート押し動作を伴うコーヒー給仕タスクを行った。

このように日常生活環境における数多くのタスクを実現できる身体能力をロボットが持つことが示されることと並行して、ロボットが日常生活環境で人間と同様にタスクを行う上で必要となるもう一つの要素である思考能力についても人工知能分野で研究が長く行われてきている。ロボットが目的を達成するための行動を計画するためには目的となる状態をロボットシステムで解釈可能な形で記述する必要がある。様々な状態記述の方法がこれまで提案されてきているが、代表的なものとしてシンボリックな技術が挙げられる。シンボリックな記述を用いたシステムにおいて行動計画は、事前に if-then となるルール知識を与えておき、現在の状態記述から目標状態へ至る状態遷移の集合を求める問題となり、人工知能分野では推論とも呼ばれる。推論は一階述語論理を利用した探索問題として扱われている [9, 10]。このようなシステムはエキスパートシステムと呼ばれ、医療における症例診断 [11] や工業製品の故障原因診断 [12] など、対象となる分野における専門的知識を予め入力し、推論によって最適な問題解決を行う手段として利用されている。ロボットシステムへの応用として実用的なタスクの実現も行われている。Beetz らは Willow Garage 社製双腕モバイルマニピュレーションロボット PR2[13] を用いて、パンケーキの調理 [14] を題材に、調理手順や物品の認識・把持方法などタスクを行うために必要となる膨大な知識を、インターネット上に集積された大規模知識データベースを用いた推論を行い自律的に実現するシステムを提案した。Kunze らは、同ロボットを用いてエレベータを使ったビル移動を伴うサンドイッチの購入 [15] を行い、複雑な内部状態を持つ構造物を利用した行動実現を達成した。

このようにロボットによって日常生活環境における数多くのタスクの実現可能性が示されている一方で、日常生活支援環境において実用的にタスクを行うロボットが普及するには至っていない。原因の一つとして日常生活環境におけるタスクが多様であることが考えられる。例として「部屋の片付け」をロボットの行動目的として与えることを考える。リビングテーブルに食器や食べ物、テレビリモコンが置いてあったとして、ある家では食器と食べ物のみを片付ければよい場合もあれば、テレビリモコンも別の特定の場所へ運びテーブル上になにもないような状態にする必要がある場合もある。また片付ける物品がそれぞれどこに運ばれるべきなのかは環境によって異なる。さらに食べ物は冷蔵庫に片付けられるという場合には冷蔵庫の場所や扉などの操作法などの知識をロボットはタスクを行うために持っている必要がある。これらは環境において固有の知識であり、環境中において共生する人の経験や文化的な背景知識に依存するため、事前にロボットに対して統一的に与えておくことが不可能である。

したがって、日常生活環境におけるロボットシステムには、事前に与えることのできる固定的・一般的な知識に加えて、各環境において後天的に得られる経験に基づく知識を自律的に獲得していく機能、安全かつ効率的にタスクを遂行するために失敗を未然に防ぎ修正することのできる機能が必要不可欠である。ロボットがタスクを行っている最中に発生する失敗として、ロボットの認識や動作計画時に目標状態にいたる動作を現在の状態から計画不能になることによる失敗、ロボットが持つ環境の状態記述と現実世界の状態に差異が生じることによる失敗に

大別される．前者の失敗は例えばロボットの自律移動を行う際に障害物の影響などにより経路探索の解がない場合などが挙げられるが，失敗したこと自体はロボットシステムにとって明示的に検知可能である．この場合は失敗に対してタスクを続行できる状態への遷移を行う動作を予め定義しておくことで，失敗状態からの復帰を行うことができる．例として障害物をどかす動作などが挙げられる．一方で後者の失敗はロボットが失敗をした事自体を認識すること自体が困難であり，ロボットは自力で正常状態に復帰することはできない．一方でロボットが行う日常生活におけるタスクは人間によって見慣れたものであるため，人間にとってロボットが行動に失敗していることがわかることが多い．したがって，ロボットが失敗を防ぐためには，タスクを実行中にも共生する人間に対して，常に指示を受け入れてロボットのもつ知識や判断を随時修正可能にする機能を備えるシステムとなっている必要がある．

本論文では，このようにロボットが日常生活環境において活動を行ううちに環境の認識に基づく経験を蓄積し，随時人の指示を受けながら日常生活タスクを行うために必要な知識を獲得していく仕組みを備えたロボットシステムの構成法を明らかにすることを目的に論じていく．

1.2 本研究の特色

近年では Graphic Processing Unit(GPU) を活用した大規模並列演算やセンサの性能向上，計算機の演算高速化や省電力化といったハードウェアの技術進歩と，深層学習や動作計画といったアルゴリズムの改善による精度・速度の向上により，ロボットが活躍することができる環境やタスクが増えてきている．従来ロボットが得意としてきた，ロボットが特定のタスクを行うためにデザインされた環境で，与えられた単純なタスクを人間よりも正確・高速に行うという範疇を超えて活躍するロボットが期待される．環境が動的で不確実性が大きくなるに連れて，認識や動作計画そのものの性能向上が必要となるだけでなく，知識を持ち，部分的な観測を用いて現在の状況を推論し，高度な意思決定を行っていくロボスタな知能システムが必要となる．人と共生する環境においてロボットが正しく振る舞うためには，タスクを行うために必要な認識や計画機能を構成する普遍的な知識を備えた上で，環境に固有の構造物の操作方法や，対象物の配置といった経験的に得られる情報を自律的に獲得する必要がある．本論文ではこのような長期的な経験に基づく環境固有の知識を自律的に獲得し，目的は同じであっても環境に応じて達成する目標状態が多様となる日常生活タスクを，環境に非依存な形で抽象的に記述されたタスクから環境に適した計画・実行可能なタスク記述へと具体化し，自律的に実行していくシステムをシンボリックな状態記述レベルだけでなく，環境操作の具体的な手法などのサブシンボリックなレベルも含めて統合的に扱う研究は未だなされていない．

1.3 本論文の構成

本論文は全9章から構成される．図1.1に本論文の構成を図示する．

第1章「序論」では，過去に行われてきた日常生活支援タスクのロボットによる実現に関する研究についてまとめ，日常生活支援タスクをロボットが実現するために必要となる要素と課

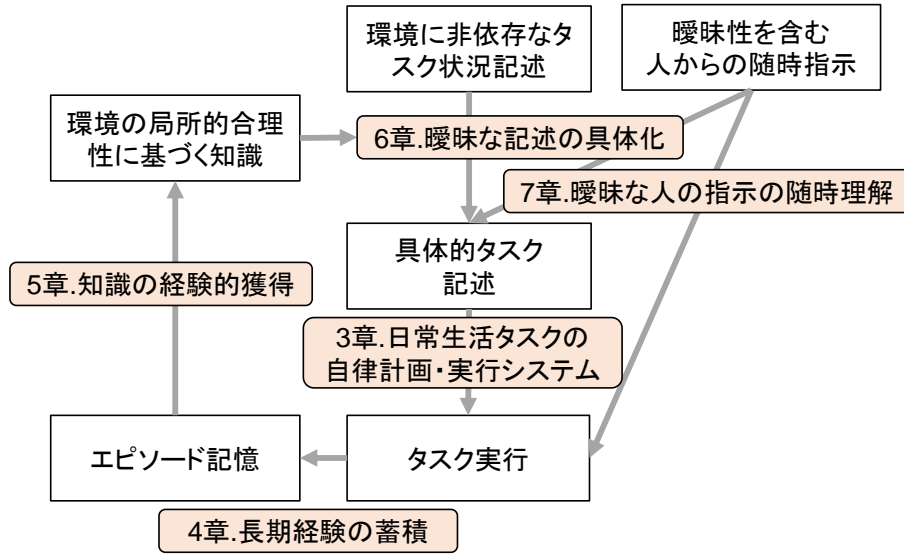


図 1.1: 本論文における構成と章の関係

題について整理し、本研究で題材とする課題と目的についてまとめた。

第2章「長期経験に基づくタスク具体化機能を備えた日常生活支援ロボットシステム」では、ロボットが日常生活支援タスクを自律的に行うロボットシステムの構成法について考察する。日常生活タスクを自律的に行うための課題として日常生活タスクの目的に対する目標状態の多様性について指摘し、今まで与えられてきた固定的なタスク記述をそのような多様な状況を許容する形で記述可能にした状況レベルにおける記述の必要性についてタスク記述の階層的な分類を行い説明する。経験から得た環境固有の局所的な合理性に基づき状況レベルタスク記述をロボットがタスクを実行する各環境に応じて具体化する仕組みを提案する。また日常生活環境の特徴として人との共生環境であることに着目し、誤りを含む可能性のある経験から得た知識を人が随時指示・修正することが可能であるロボットシステムの構成法と、人の指示における曖昧性解決法についても考察する。

第3章「日常生活支援タスクを行う自律ロボットシステムの構成」では、本研究において扱う日常生活タスクを行うロボットのアーキテクチャである分散型モデルベースロボットシステムの構成法をはじめ、行動計画器・物体認識器・動作計画器といった日常生活タスクの自律計画・実行をするために必要な機能の構成や、人による随時指示を実現するために必要となる音声対話機能や人のロボットへの直接接触における指示機能の構成要素について、具体的なハードウェア・ソフトウェアに関する説明を行う。

第4章「ロボットにおけるエピソード記憶に基づく長期経験の蓄積機構」では、ロボットが環境ごとに異なる日常生活タスクの合理性に関する知識を後天的に得るために、ロボットシステムにおいて経験を蓄積する仕組みを構成する方法について述べる。経験の形式として認知心理学的な記憶の分類における出来事・経験を表す記憶であるエピソード記憶の性質をもとに、ロボットシステムにおいて扱われる時系列データをロボットにおける出来事であるシンボリックな状態記述や状態遷移と関連付けながら記憶する仕組みを提案し、経験からの知識獲得時に

必要となるデータの集約を、システム全体のパフォーマンスを低下させないように行う仕組みを導入する。

第5章「長期経験を用いた環境の局所合理性に基づく知識の自律獲得機構」では、環境に非依存な状況レベルの日常生活タスクを具体化するために必要な局所合理性を含む経験的知識として、人が日常生活を送る際の物や場所との相互作用に伴う状態変化に着目して、人の物に対する相互作用傾向を表す嗜好モデル、人の特定の場所に対する相互作用傾向を表す定位置操作モデル、物と場所の関係性における傾向を表す確率的物体配置地図としてモデル化し、各々の獲得手法を提案する。実ロボットにシステムを搭載し、経験から学習的に知識を獲得可能であることを示す。

第6章「環境に非依存な状況レベルタスク記述と局所合理性に基づく具体化機構」では、状況レベル日常生活タスク記述を行うための要件について考察し、記述したタスク記述を局所合理性を含む経験的知識を用いて具体化し、実際にタスクを計画・実行するための仕組みについて提案する。

第7章「曖昧な指示・修正を随時受け付け対応可能な日常生活タスク実行システム」では、日常生活環境でタスクを行うロボットに必要となる、共生する人から常に指示や修正を受け入れ対応することが可能なシステムについて提案する。ロボットがタスクを実行中に行われる可能性のある割り込み指示について、ロボットの実行制御システムを階層的に分類し、割り込み制御機能を組み込むことで、人からの随時指示に対応する仕組みを導入する。これにより誤りを含む経験的な知識を利用したタスクも人間が随時中断・修正を行えることで、タスクを失敗せずに続行することが可能となる。人による随時指示手段として音声指示と直接接触による指示を挙げ、音声指示において発生する曖昧性と解決手法についても説明する。

第8章「日常生活タスクの局所合理性に基づく具体化機能を用いたタスク計画実行システムの評価実験」では、本研究における提案システムの有効性を検証するために、実ロボットに提案システムを実装し、提案手法により環境に非依存な状況レベルの日常生活タスクが環境の局所合理性を満たす具体的なタスクとして実行可能であることを実験を通じて示す。

第9章「結論」では、各章の内容から本研究を総括し、本研究の成果と今後の展望についてまとめる。

第2章

長期経験に基づくタスク具体化機能を備えた日常生活支援ロボットシステム

2.1 はじめに

ロボットシステムとは与えられたタスクを受け取り、環境認識、行動計画、操作制御を組み合わせることで環境に作用しながら状態を変化させていく機能を有するシステムである。ロボットが与えられる役割や環境の動的性質によって、行動目的となる状態記述の抽象度は変わるが、抽象度が大きくなるにつれて、実際の動作に必要な具体的状態遷移列を得るために必要な周辺知識が多く求められる。日常生活環境において目的を達成するために必要な周辺知識は各環境に固有の文化や習慣によって根拠付けられているため、予め知識としてロボットシステムに与えておくことは難しく、ロボットはおかれた各環境において自身が得た経験から知識を自律的に獲得していく必要がある。本章ではこのようにロボットが工場で予め具体的に記述しておくことが困難である、環境や実行時の状況に応じて多様な日常生活タスクを環境に非依存な形で記述し、各環境における局所的な合理性に従う計画・実行可能なタスク記述として具体化可能なロボットシステムの構成法について説明する。

2.2 日常生活環境におけるタスク

タスクとは、ロボットが動作を行うことによって動作実行前の環境の初期状態をタスクの目的を満した環境の目標状態へ変化させるための状態遷移のことをいう。例えば「物の運搬」を目的とするタスクでは、目標状態は「その物が運搬目標地点に配置されている」状態として表現される。日常生活において行われるタスクの目的として家事が挙げられる。家事の具体的な例として以下のものが挙げられている [16]。

- 家の掃除

、物品の整頓、掃除機などの道具を用いた床の清掃、壁や窓の拭き掃除、机などの拭き掃除、ゴミ捨て、台所やトイレなどの掃除など

- 食事の準備

調理やそのために必要な食材の調理準備作業、調理を行ったあとのごみや調理器具の後片付け、出来上がった料理や食器の運搬、飲み物などの給仕など

- 食事の後片付け

食器の運搬、食器洗い、机などの拭き掃除、食器洗い機の操作など

- 洗濯

洗濯衣類の集積、洗濯機の使用、衣類の折りたたみ、アイロンがけ、衣類の仕分け収納など

- 買い物

在庫不足物の確認、買い物、宅配便の受け取りなど

例として、物品の整頓を目的とするタスクの目標状態を環境中に配置された各々の物品が本来あるべき地点に配置されている状態と定義する。タスクを実行するロボットは予め整頓対象である物品やテーブルなどの物品を配置する構造物を環境において識別するための知識を持っている必要がある。次に各物品についてその物品が整頓されている状態に関する知識を必要とする。この知識は物品がタスクを行う初期状態において整頓されているかの判断、および整頓されている環境状態へ遷移するための動作をロボットが決定する上で必須となる。整頓されていない物品を整頓されている状態へ遷移するために必要な動作は物品自体の特性や物品が現在配置されている場所、物品を配置すべき場所に依存する。例えばテーブルから別のテーブルへ物品を運搬する場合には物品の把持、移動、物品の配置の動作が必要になるが、対象物品の配置先が冷蔵庫などのように、扉を開けるといった配置のための予備動作を必要とする場合は予備動作を行う必要があるかを判断するための操作対象の状態推定や捜査対象の状態を遷移させる予備動作に必要な動作パラメタが必要となる。ここで動作とはロボットが実行することによって起こる環境の状態遷移の最小単位として定義される。

このタスクを実行するために必要な知識を時間的・空間的な有効範囲によって分類すると以下のようなになる。

静的な知識 タスクの実行開始から終了まで常に真であるような知識。

環境の大域的な合理性に従う知識 タスクを行う環境・時間・状況などが変わっても不変であるような知識。例として、物品や環境における構造物を識別するために必要なラベルや、重力加速度・運動方程式といった物理的な法則に従う知識。

環境の局所的な合理性に従う知識 ある環境においてのみ真であるような局所的な合理性を満たす知識。例として、棚の配置や操作法、コップが整頓されるべき場所、リビングテーブルに常に置いてあるべき物品に関する知識。

動的な知識 タスクの実行中に真偽が変化する可能性のある知識。環境認識によって得られるある時点における物品の状態やタスクの行為者自身の状態が該当する。

このうち環境の局所的な合理性に従う知識は予めエージェントに対して与えておくことが不可能な知識であるため、環境においてロボットが経験を蓄積し自律的にこれらの知識を獲得する機能が必要とされる。

Anderson[17]はこのような日常生活におけるタスク (Everyday Activities) を知能エージェントが扱うことの問題点を、チェスを解くことといった今までに人工知能分野において扱われてきたタスクと比較して以下のように特徴づけている。

- 現実世界では不可逆な失敗により危険な状態に陥る可能性が大きい。
- 日常生活におけるタスクは最適解よりも範囲によって目標状態が与えられる。
- 日常生活におけるタスクは環境において何回も反復的に観測もしくは経験されるものであり、その行動のいかなる状態も予期可能である。予期できないことが起こる行為は日常的とはいわない。
- 日常生活におけるタスクの合理性は文化・経験・先例によって定義される。その日常生活におけるタスクの行為者が自身の知識に照らして、タスクの目標を達成に導くことができるかと十分期待できるような意思決定は合理的であるとされる。

環境において局所的に得られる合理性は環境においてロボットが共生する人の持つ文化や経験によって定義される。したがって、ロボットは環境において行われる人の動作や操作対象となる物品や場所を観察した記憶から、間接的に局所的な合理性に基づく知識を獲得する必要がある。間接的に獲得した知識は誤りを含む可能性をはらむ。誤った知識に基づいて実行したタスクは失敗する。したがって失敗に対してそれを未然に防ぐ仕組みが必要とされる。

以上をもとに日常生活環境におけるタスクの特徴をまとめると以下ようになる。

複雑な手順を要する 洗濯機やエレベータなど日常生活環境において扱われる家電・構造物を扱うためにはその操作方法や効果を知らなければならない。冷蔵庫の中から食べ物を取り出すためには扉の開閉動作を物体の把持の前後に追加することにより、冷蔵庫の扉が開き把持可能であるという状態に遷移する必要がある。このように環境において内的状態を持つ物体操作が含まれるタスクでは依存関係の解決や環境状態の推論が必要になる。

目標状態が環境に固有の局所的な合理性に従う 掃除や食事の準備といったタスクはタスクを行う環境固有の知識なしに目標状態を定義することができない。タスクの目標状態が合理的である根拠は環境において生活する人の経験・文化・慣習によって支えられるからである。食事の準備にコーヒーは含まれるか、コーヒーをサーブするときにミルクを添える必要があるのか、砂糖は入れる必要があるのかといった知識は事前に与えることは難しく、各環境において生活を営んでいるうちに観測もしくは人間とのインタラクションを通じて得られる経験に基づき獲得され、その知識を用いることで初めて具体的なタスクとして計画・実行可能になる。

失敗が致命的である 冷蔵庫のドアを閉めることに失敗した場合、冷蔵庫内の食材が痛み使用不能になる場合がある。アイロンがけタスクにおいてアイロンを床に落とした場合、家

事になる可能性がある．このようにタスク実行中の失敗は取り返しが付かない可能性があり，安全のため失敗復帰を計画可能な機能が必要である．また環境において共生する人間によって動作を中断できる仕組みも必要になる．

このような特徴をもつタスクを自律的に計画・実行可能なシステムとして，以下のような機能が必要とされる．

失敗復帰を考慮したタスク計画・実行機能 ロボットは，タスクごとに家電操作など環境に依存した複雑な手順を含む動作を行わなければならない．またタスクの種類は多様である．このような日常生活環境におけるタスクをロボットが自律的に行うためには各タスクの実現動作列を書き下すシステムは非効率であり，行動に必要な動作をモジュール化し，タスクの内容に応じて必要な動作を組み合わせる行動計画の仕組みが必須である．日常生活環境はロボット以外にも共生する人などによって常に変化し，ロボットにとって環境における情報を一度に完全に観測することはできないため，計画された行動は実世界と環境モデルの誤差により失敗する可能性が付きまとう．したがって，失敗を予め考慮したロバストな行動実行機能が必要とされる．第 2.3 節にて本機能について関連研究と本研究におけるアプローチを説明する．

環境非依存なタスク記述機能 ロボットが自律的，あるいは環境に共生する人の指示を受けて日常生活タスクを行うためにはそのタスク自体の名前や目的をロボットが知識として持っている必要がある．日常生活におけるタスクは同じ目的でも環境に応じて内容が多様になるため，どの環境においても共通する大域的な合理性を持ち，あとから環境固有の局所的な合理性に応じて具体的なタスクの記述を生成することができるような記述が必要とされる．第 2.4 節にて本機能についての関連研究と本研究におけるアプローチを説明する．

タスクの具体化機能 大域的な合理性に従うタスク記述はそのままではタスク実行に必要な具体的な動作を計画することはできない．各環境における局所的な合理性を含む知識を用いてタスク記述を行動計画が可能な具体的な記述へと具体化する仕組みが必要とされる．

経験に基づき局所的な合理性を獲得する機能 局所的な合理性は環境に暮らす人の経験によって根拠付けられる．したがって人や人が暮らすことによって相互作用のある環境をロボットが観察することによって得た経験から環境に固有の局所的な合理性をモデル化し，自律的に獲得する仕組みが必要となる．

自律的に経験を蓄積する機能 経験的な知識を得るためにはロボットが自律的に経験として抽出可能な長期的記憶を持つ必要がある．ロボットシステムにおいて得られるデータは時系列情報から経験として有意なデータを獲得するための仕組みが必要となる．ロボットシステムにおいて扱われるデータは多様で膨大だが，日常生活環境において多目的にタスクをこなすロボットは物理的な制約から計算資源や記憶資源に制限がある．タスクの実行性能に影響を与えないような記憶の仕組みも必要とされる．

人の指示を随時受け入れ対応する機能 予め与えられた知識に対して自律的な経験蓄積に基づ

き得られた知識は必ずしも真であるとは限らない。誤った知識を元に計画されたタスクは失敗を誘発する。タスクを正しく計画したとしても環境の変化に伴いロボットはタスク実行に失敗する可能性がつきまとう。一方で環境において共生する人には、日常生活タスクに関する経験や知識を十分に持っているため、ロボットのタスク失敗の予兆に人が気がついた場合にその場でロボットに指示を与えることで、ロボットが失敗から復帰しタスクを失敗することなく続行可能な仕組みを備えることは、タスクの成功率や安全性を向上する他にも人に安心感を与える上で重要な機能である。タスクの実行中には環境やロボット自身の状態遷移を伴うため、人の指示を受けた時点でタスクを再開可能であるように適切に中断し、タスクの目標状態や動作を修正・再開する仕組みが必要となる。また人から指示を得る際に発生する意思伝達にも曖昧性がつきまとうため、人の指示を正確に指示を受け付けるための曖昧性解決機能も必要となる。

2.3 日常生活タスクの自律計画実行システム

2.3.1 行動計画手法

古典的アプローチ

古典的な行動計画手法の特徴としてシンボリックな状態記述言語を用いることが挙げられる。ここではロボットが行うべきタスク計画はロボットが置かれる現在の状態（初期状態）からタスクの達成条件が満たされた状態（目標状態）に対して状態を遷移させる経路を探索する問題として捉えられる。シンボリックな状態記述言語としてよく用いられるものに Planning Domain Description Language(PDDL) がある。PDDL は Stanford Research Institute Problem Solver(STRIPS) 型行動計画器を実装する言語の一つである。PDDL には複数のバージョンがあり、STRIPS 型行動計画器においては対応していなかった概念がいくつか拡張され利用可能になっている。例えば、STRIPS では宣言されていない命題はすべて偽とする閉世界仮説が前提となっていたが、STRIPS の拡張である Action Description Language(ADL) における文法を PDDL に取り入れることにより、開世界仮説に基づく計画も可能になっている。また、従来では記述に離散値のみが扱われていたが、数値を扱うことにより、コスト関数を定義することができる。これにより目標を満たす状態遷移が複数存在するときに効用を最大化する拘束を計画に与えることが可能となっている。複雑な手順を要するタスクを定義することができ、環境の認識に応じた初期状態を与えることで柔軟な行動計画が行える一方で、行動計画に必要な状態記述を知識として事前に全て与えておく必要があるため、環境やタスクの複雑さによって膨大な記述を必要とすること、またそれに伴い探索を行う状態空間が広がり、計画に時間を要すること、タスクが増えるにつれ状態記述をなすシンボルやシンボル間の関係性を慎重に設計する必要があることが問題として議論の対象になっている。

反射的アプローチ

ロボットのセンサ情報からの入力に対して、ロボットが取るべき動作を予め対応付けることによって自律的に振る舞うロボットシステムである。代表的なものとしてサブサンプションアーキテクチャが挙げられる [18]。サブサンプションアーキテクチャは「測距センサの値が十分大きければ前進する」、「前方に障害物があれば 90 度回転する」といった単純な if-then ルールによってセンサ入力とアクチュエータ出力を対応付ける並列動作モジュールを組み合わせることにより、システム全体として意思決定を自律的に行うシステムである。各モジュールは相対的に優先度を持っており、優先順位の高いモジュールにおける決定はそれより下位の決定よりも優先されることで、システム全体の意思決定として一貫性をもたせることで、アクチュエータの資源競合や矛盾を回避している。各モジュールはセンサ入力に対して行動計画を行うことなく対応する行動を反射的に出力するため、動的な環境においてロボットが意思決定をすばやく行うことができる利点がある。このシステムは iRobot 社の自律掃除ロボットである Roomba に搭載され、実際に家庭環境に広く普及している。しかしながら、家電操作や道具の利用といった環境状態に深く依存し、複雑な手順を要する家事支援タスクをこのシステムで実現するためには、多数のモジュールを用意する必要があるほか、ある行動中における意思決定が正しく行われているかを正しく設計・修正するためにはモジュール全体の振る舞いを開発者が把握し、優先度を注意深く設計していかなければならない。これは動的に環境が変化していく中で学習により自律的にモデルを修正していく自律システムを構築することが難しいことも意味する。モジュール単位の認識と意思決定に関する情報を高度にしていくと、全体に必要なモジュール数を減らすことが可能となるが、認識処理時間の増大を招くなどサブサンプションアーキテクチャの利点である反射的で高速な意思決定を行いにくなくなってしまう。

本研究では日常生活支援環境における環境依存のタスクや経験をういた知識の獲得による具体化を題材とする上で、タスクを環境に非依存な形で記述する仕組みが必要とされる。またロボットが自律的に蓄積した記憶データから経験を抽出するためには、記憶を行った際の出来事に関する手がかりをもとにデータの選択を行う必要がある。反射的なアプローチに比べて古典的なアプローチではロボットが現在置かれている状況と、対応するロボットの意思決定が明確に記述される。本研究では古典的なアプローチにおいて扱われるモデルベースな行動記述・計画手法をもとに議論を行う。

2.3.2 動作実行手法

動的な環境においてロボットが行うタスクには常に失敗がつきまとう。チェスなどの可逆的な失敗とは異なり、日常生活環境では失敗状態への遷移が取り返しがつかない損害を環境に与える可能性もあるため、ロボットは環境の変化に適応して失敗を未然に防ぎながらタスクを遂行していく必要がある。このようなロバストな自律動作実行を行うための仕組みとして、リアクティブプログラミングが挙げられ、航空宇宙分野を始め多くのシステムで適用されてきている [19]。リアクティブプログラミングではある条件を満たすときに指定された処理を実行する

ルールを複数定義し、状態が適用可能なルールを並列に実行する反射的な行動記述方式である。航空宇宙分野におけるロケットエンジンの制御システムなどのように、システムとして正常な定常状態に対して、何らかの異常な状態が起きた時に、もとの正常な状態へ復帰するルールを記述することなどに適している。ルール全体の条件をもとに有限オートマトンを生成することができ、状態に対して全ルールの一致を確認する必要がないような効率化を行うことができる。一方で環境依存操作を含む日常生活環境では定常状態が定義されないため、記述が複雑化してしまう。

岡田らは古典的な STRIPS 型シンボリック状態記述言語を用いた行動計画器の出力である動作列に対して、動作の前提条件 (Precondition) を同一、影響 (Effects) を失敗状態へ書き直した失敗動作を定義し、失敗状態から目標状態への計画を行った動作列をグラフ形式に結合する仕組みを提案した [20]。この手法ではタスクを実現する動作間の中間状態を明確に記述することが可能であった。ロボットがタスクを行う最中の状態を個別に識別可能であることで、環境において人から随時割り込みを受けた場合でも割り込みを受けた際の状態をもとに適切な中断・復帰行動を行うことが可能となる。本研究ではモデルベースな行動記述・計画手法を用いた [20] の方法を用いて、具体化によって得られる環境に適したタスク目標状態を満たす動作遷移に対して失敗復帰動作を加えることで失敗を考慮した自律動作実行が実現される。

2.4 日常生活タスクの記述

2.4.1 ロボットのタスク記述の階層性

本節では日常生活環境におけるタスクをロボットシステムにおいてどのように記述し扱うかについて述べる。ロボットシステムとは与えられたタスクを受け取り、環境認識、行動計画、操作制御を組み合わせることで環境に作用しながら状態を変化させていく機能を有するシステムである。1台のロボットで多様なタスクをこなすためには多様な機能を備えていることが必要となる。

- ロボットの置かれた環境においてタスクの目的を満たす動作列を考慮する行動計画機能
- 環境において目標となる地点へ自身を移動するための自律移動機能
- 物体操作対象の把持点や移動における障害物回避を行うための環境認識機能
- 対象物の把持や構造物の操作を行うための動作計画機能

のように多数の機能を必要に応じて組み合わせることでロボットはタスクを実現していくシステムである。日常生活タスクのように家電操作や物体運搬・道具の使用など多数の手順からなり、環境に依存する複雑な動作からなるタスクをロボットシステムに置いて実装するためには、タスク間や動作間において共通して使用される機能の組み合わせやそのやり方をまとめてカプセル化し、階層的に記述可能な仕組みを用意することが重要である。

ロボットにおける言語の階層性について、井上はロボットアームを用いた作業を題材に5つの階層構造に整理している [21]。

- レベル 0: コマンド・レベル ロボットに対する最も単純な指令値である目標関節角度列や台車アクチュエータのモータ指令値を記述するレベル．例として ML 言語 [22] がある．
- レベル 1: 原始的動作レベル ロボットの手先位置・姿勢によって作業を記述するレベル．座標系の変換や逆運動学計算によってコマンドレベルへ変換可能になる．例として VAL 言語 [23] がある．
- レベル 2: 構造的動作レベル ロボットの手先位置・姿勢列やロボットのセンサ値などを利用した条件分岐によって構成される手先軌道によって作業を記述するレベル．軌道を時間的にサンプリングすることにより原始的動作レベルによる記述へ変換可能になる．例として RAPT 言語 [24] がある．
- レベル 3: 対象物状態レベル ロボットの行う動作を動作の対象物の状態変化として記述するレベル．環境における物体の運動制約やロボットとの干渉計算を行うことで構造的動作レベルの記述へ変換可能になる．
- レベル 4: 作業目標レベル ロボットの行うタスクの最終目標状態のみを記述するレベル．最終的な目標状態に至る手順の推論を行うことで対象物状態レベルの記述へ変換される．シンボリックな環境状態記述．

上位レベルの階層における記述によって下位の目標が決定される構造となっている．このとき下位の目標は上位レベルの目標を満たし，かつロボットの動作という観点から上位レベルより具体的な記述であることから，上位レベルにおける目標記述に従い，下位レベルにおける目標記述を表すことを具体化と呼ぶ．例えばロボットの座標系における目標手先位置・姿勢をもとに逆運動学計算を用いてロボットの各アクチュエータにおける目標指令値を決定することは原始的動作レベルにおける具体化である．具体化を行うためには上位の目標を満たす範囲のうち下位における目標を決定するための情報を与える必要がある．先程の例では目標手先位置・姿勢に対して，関節のリンク情報や関節の可動域などの情報を含むロボット自身の幾何モデルや逆運動学計算の評価関数を与えることで目標関節指令値が決定される．日常生活におけるタスクをロボットシステムにおいて記述することを考えるとき，タスクの最終目標状態には大域的な合理性に従う知識の他に，局所的な合理性に従う知識やタスクの実行ごとに異なる動的な知識による状態表現が含まれており，同じ目的のタスクであっても実行ごとに具体的な記述として与えられる必要があることを意味する．ロボットが様々な環境や状況においてタスクを実行するためにはこれらに非依存な記述をもとに環境や状況に適応した動作をすることが必要とされる．環境や実行状況に依存しない目標状態記述を，環境における具体的な状態に依存しない記述であることから，本研究では状況レベルにおける記述と呼ぶ．本研究において扱われるロボットシステムの階層性は表 2.1 のように整理される．

ここで作業目標レベルにおける記述は環境に依存しているため事前にはロボットシステムに記述として与えておくことができないのに対して，状況レベルにおける記述はタスクの目的を達成する状態のうちの一部を表現する記述である点は変わらない一方で，環境に依存する知識が含まれないため，システムに予め記述することが可能である．本研究ではこの階層構造のうち，具体化を行うために環境において経験的に知識を獲得することが必要となる状況レベルと

表 2.1: 日常生活タスクを行うロボットシステムの階層的枠組み

目的レベル	目標	タスクの目的そのものを表すシンボル .
	例	(tidyup-room)
	具体化	環境の大域的な合理性に従う知識
状況レベル	目標	タスクの目的を満たす状態遷移の最終目標状態を環境に非依存な形で抽象化した記述
	例	(forall ?item (on ?item ?usual-spot))
	具体化	環境の局所的な合理性に従う知識
作業目標レベル	目標	シンボリックな作業の最終目標状態
	例	(and (in bottle fridge) (on tray kitchen))
	具体化	行動計画
対象物状態レベル	目標	各環境状態遷移における操作対象物の目標状態
	例	(in bottle fridge)
	具体化	環境操作モデル
構造的動作レベル	目標	センサ座標系における操作対象物の位置姿勢
	例	#<coords :camera->bottle 770 0 800 / 0.0 0.0 0.0>
	具体化	動作計画
原始的動作レベル	目標	ロボット座標系におけるエンドエフェクタの手先位置姿勢
	例	#<coords :rarm 370 -197 782 / 0.904 -0.947 2.073>
	具体化	逆運動学計算
コマンドレベル	目標	ロボットのアクチュエータの指令値
	例	#f(10.0 0.0 20.0 0.0 12.0 9.0 -2.0)

対象物状態レベルを扱う .

2.4.2 状況レベルのタスク記述

第 2.4.1 節において、状況レベルのタスク記述は、日常生活におけるタスクの目的を満たす、環境に非依存なタスクの最終目標状態記述であり、環境に固有の局所的な合理性に従う知識を用いて各環境に適した具体的な目標状態である作業目標レベルのタスク記述へと具体化することが可能となるような記述として定義された。状況レベルのタスク記述のように、その記述の解釈がそれ単独では定まらず、各環境における状況を参照して初めて解釈可能になるという性質を持つ表現を推論する枠組みは状況理論 (Situation Theory) の分野で議論されてきた [25, 26, 27, 28]。状況理論は従来では普遍的に真偽が与えられる論理式について、状況、個体、関係の 3 種の要素を用いて捉え直し、特定の状況においてのみ成立する事実や関係を表現するための理論である [29]。中島による状況理論に基づく論理式の表記 [26] を用いると、例えば

「鳥が飛ぶ」という知識は、本来の論理式では

$$\forall x \text{fly}(x) \subset \text{bird}(x) \quad (2.1)$$

となるが、状況理論の枠組みにおいては

$$s_{\text{bird}} \models \langle \langle \text{fly} \rangle \rangle \quad (2.2)$$

のように表記される。「鳥」という状況 s_{bird} において「飛ぶ」という性質が無条件で成立することを意味する。

日常生活タスクにおける記述をこの枠組で説明していく。「テーブルの上にあるものを片付ける」タスク t を考える。ある環境 s_1 においてテーブルの上にあるものを片付けるタスクの作業目標レベルにおける記述、すなわちタスクの最終目標状態は以下のように記述されるとする。

$$\text{on}(\text{cup}, \text{kitchen}) \quad (2.3)$$

これはテーブルの上にあるコップをキッチンへ移動することによってタスクの目的が達成されることを意味する。

別の環境 s_2 においては以下のような記述として与えられるとする。

$$\text{on}(\text{cup}, \text{dishwasher}) \quad (2.4)$$

これはテーブルの上にあるコップを食器洗い機へ移動することによってタスクの目的が達成されることを意味する。

上記の2つの式は同じタスクを表現する式であるが、各々の環境においてのみ支持されるため、異なる環境に対しては異なる式によってタスクの目標状態を与える必要がある。ここで kitchen や dishwasher を環境において片付けられているときに cup が存在する場所 p_{cup} を用いると、

$$s \models \text{on}(\text{cup}, p_{\text{cup}}) \text{ s.t. } s_1, s_2 \in s \quad (2.5)$$

として環境に非依存な形でタスクの目標状態を記述することが可能となる。このとき p_{cup} は各環境の状況が与えられて初めて決定することから変数と捉えることが可能である。変数には変数が用いられる式における他の個体や関係による拘束を受ける。ここでは変数に代入できる個体として「コップが片付けられたときに置かれている場所」という拘束が与えられている。本研究ではこのような変数に対して環境の状況が与えられて初めて決定する拘束条件を環境に固有の局所的な合理性と呼ぶ。本研究において提案されるロボットシステムではこの局所的な合理性をロボットが環境において得られる経験的な知識から自律的に獲得することを目的とする。獲得の対象となる局所的な合理性に基づく知識は第2.5節および第6章において説明される。

2.4.3 日常生活タスク記述のための概念

本小節では日常生活タスクを記述するためにロボットシステムとして備えておくべき概念について説明する。

第2.2節に分類したタスクは、マニピュレーションの観点からみるとロボット自身の環境における移動を伴う環境操作の繰り返しと捉えることができる。移動を伴う環境操作を扱うロボットシステムにおけるシンボリックな計画に利用可能な環境状態表現として、岡田らはスポット (spot) というシンボル概念を提案し、複雑な日常生活タスクのロボットによる自律実行を行っている [6]。spot は環境に対して特定の操作を行うために意味を持つ場所を表す概念で、いわゆる“定位置”を表す概念である。spot 概念が定義されることにより、環境に対する操作はコップやペットボトルといった環境においてタスク実行中に運搬対象となる物体と冷蔵庫やテーブルのようなタスク実行中に運搬対象とならない環境構造物に分類される。ここで区別のため前者をアイテムと呼ぶ。

特定の操作を行う場所は、各場所においてロボットが物理的に操作可能な物体との位置関係によって決定される。また文化的な制約（花などを飾るディスプレイ用のテーブルの上には他の物を置いてはいけないというような物理的な合理性を伴わない条件）など多様な条件がスポットに対して与えられる。このような spot はタスクや環境ごとに手作業で定義されていた。

以上を元に、移動を伴う環境操作からなる行動を計画するために必要となる概念を以下にまとめる。なおシンボリックな状態記述を行うことを前提に物体や場所と言った存在に関する概念をシンボルと呼ぶ。

シンボル アイテム ロボットがタスクを行う上で扱うアイテムを表すシンボルである。

スポット ロボットが動作を行うための前提条件として存在するべき意味的な地点 spot を表すシンボルである。

人 ロボットが動作を行う対象としての人を表すシンボルである。

ロボットの資源 ロボットの身体的な資源を選択する計画を行うためのシンボル。本研究では双腕台車移動ロボットを用いるため、選択可能な資源は右腕、左腕、両腕の3つである。

シンボルの関係 アイテムとアイテムの関係 アイテムについて特定の他のアイテムとの意味的な接触関係を表す。

アイテムとスポットの関係 アイテムについて特定の spot との意味的な接触関係を表す。

ロボットとスポットの関係 ロボットの現在の存在位置関係を spot によって表された関係。

ロボットとアイテムの関係 ロボットとアイテムとの接触関係を表す。ロボットの移動を伴う物体操作によってアイテムと spot の関係にも変化を及ぼす。

- 人とアイテムの関係 アイテムと人の意味的な関係として所有や嗜好を表す。
- スポットの状態関係 特定の spot において可能な動作により変化する状態を表す。
- アイテムの状態関係 特定のアイテムそのものの状態を識別するための関係。

これらのシンボルとシンボルの関係を組み合わせることでロボットがタスクを行う環境状態が記述される。タスクを表す記述は初期状態と目標状態の2つの環境状態からなり、この2つの状態を遷移する動作列が計画される。したがって、シンボルやその関係はタスクにおける各動作の適用条件や状態の変化を十分に記述可能であるような表現能力を備えている必要がある。

2.5 環境における局所合理性に基づく知識

本節では、前節において取り上げた、環境に非依存な日常生活タスクを表す状況タスク記述を具体化するための知識として、経験的に得られる環境依存の知識について考察する。Andersonによれば日常生活タスクにおける合理性は各環境に暮らす人の経験や知識によって定義される。したがって、理想的には、タスクを具体化する上で必要な局所合理性は環境においてロボットと共生する人から得ることによって計画可能な具体的タスクを得ることができる。しかし、人にすべての不足事項について質問などによって知識を得ることは人の時間や労力を割くため、本来の日常生活支援という目的を鑑みるとなじまない。そこで、人や人が普段生活を送る環境における傾向や規則性をロボットがタスクを実行していくうちに観察・経験することで自律的に獲得していく仕組みが必要となる。

本研究では人が日常生活タスクを行う際におこる相互作用として、第2.4.3節における分類を意識し、人、アイテム、スポットに日常生活環境の要素を分類し(図2.1)、以下のようにそれぞれの相互作用についてモデル化を行い、経験から自律的にパラメタを獲得することで各環境に適応したタスクの具体化を行うことを可能にする。

- 人とアイテム 人がアイテムを扱う時の傾向から得られる嗜好性に関する知識。
- 人とスポット 人がある特定の spot において動作を行う傾向から得られる定位置操作に関する知識。
- アイテムとスポット あるアイテムが特定の spot と関係を持つ傾向から得られるアイテムのよくある配置を表す知識。

2.6 ロボットシステムにおける長期経験獲得機能

ロボットが日常生活環境において適応的にタスクを計画・実行するために必要となる環境の局所的な合理性を表すモデルを獲得する際には、ロボットが環境において共生する人や環境の観察や相互作用の経験によって得られる知識が必要となる。そのためには、膨大なセンサ情報や認識結果などの時系列データを扱うロボットシステムにおいてこのような経験を蓄積する機

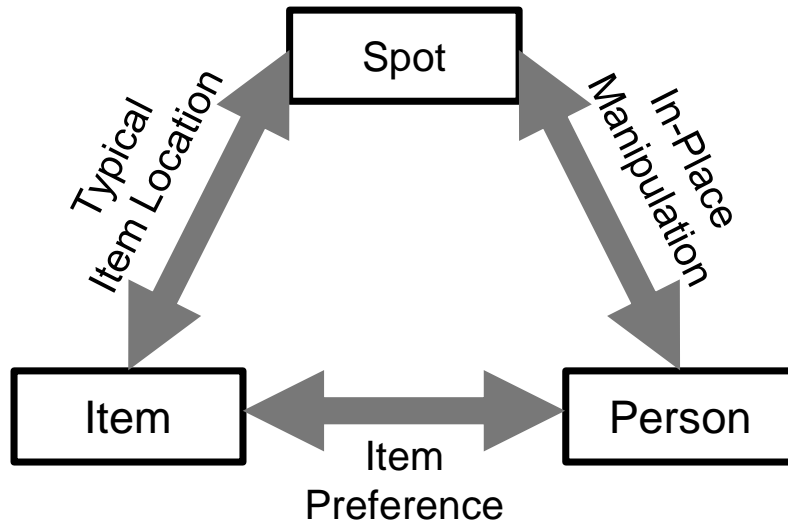


図 2.1: 蓄積経過時間に伴い獲得対象となる知識の分類

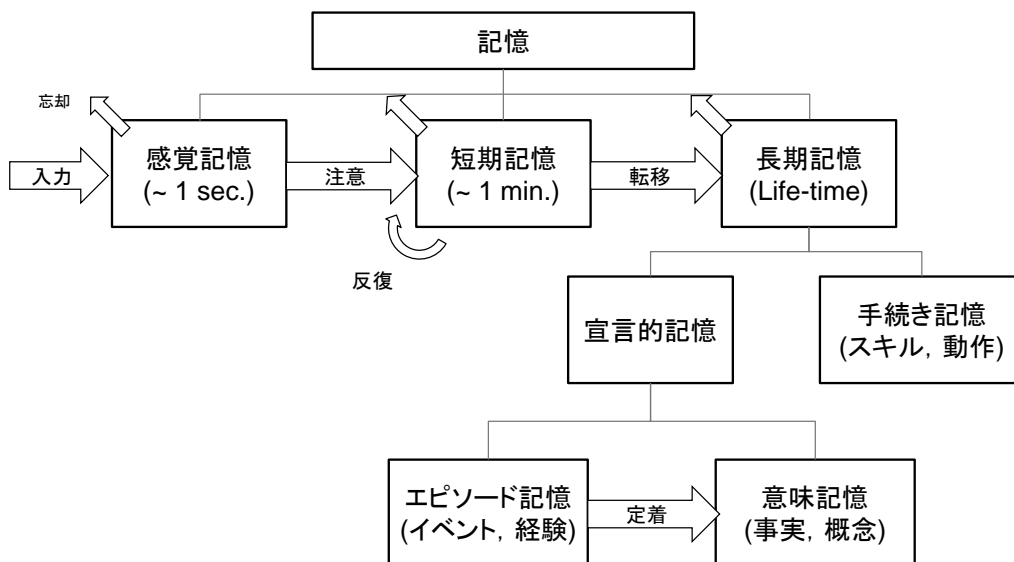


図 2.2: 蓄積時間に注目した記憶の分類とその関係モデル

能を構成する必要がある。本節では本研究において扱うべき記憶の性質について分類を行う。

2.6.1 認知心理学における記憶の分類

経験とは実際に知覚を通じて物事を体験する事によって得られる知識である。

Atkinson と Shiffrin はそれまで画一的に捉えられていた人間の記憶をその記憶蓄積期間の長さに注目して以下の 3 つに分類した ([30], 図 2.2)。

感覚記憶 感覚器官によって知覚した環境からの刺激が入力されたときの記憶である。各感覚器官につき 1 つ存在し、短期記憶層において処理される知覚情報の認識と保持を行う。

[30] では sensory register と定義されている．そのうち注意を与えられた情報のみが短期記憶へと渡され，それ以外は即座に失われる．モデルベースなロボットシステムにおいてはセンサの生データに相当する．

短期記憶 短期記憶は working memory の分野の範疇である．即時記憶 (immediate memory) とも呼ばれる．感覚記憶と同様揮発性を持つが，喪失に要する時間は比較的長い．またその容量も単純なものに制限される．感覚記憶によって保持された情報を用いるが，その種類は一致するとは限らない．例えば映像情報を知覚したとしても聴覚的な短期記憶を持つこともある．モデルベースなロボットシステムでは認識モジュールにおける処理結果など扱われるデータのほとんどが短期記憶に相当する．

長期記憶 短期記憶のうち一部が長期記憶として保持される．保持期間は最も長く，その容量にも制限はない．ロボットシステムではロボットが持つ環境の地図情報やタスクに関するシンボル情報などがこれに相当する．

さらに Squire ら [31] や Tulving ら [32] は長期記憶をその内容から以下のように分類した．

陳述記憶 宣言的記憶とも呼ばれ，言語などにより内容を意識上に想起できる記憶を指す．

意味記憶 いわゆる知識に相当する．通常いつ・どこで得た情報であるといった付加的な情報は喪失され，内容のみが記憶として残る．ロボットシステムにおいてシンボルとその意味的性質などがこれに当たる．

エピソード記憶 特定の日時や場所などの状況に関連した個人的な出来事に関連する記憶である．例えば数日前の夕飯の献立やニュースなどがこれに当たる．出来事に関する内容そのものと，時間，コンテキスト，知覚情報（出来事の内容と関連性がなくても良い），当時の自身の身体・心理的状态など，それを経験したときのさまざまな付随的情報の両方が関連づけられて記憶されていることを特徴とする．

手続き記憶 ピアノの演奏や自転車の乗り方など同じ経験を反復することで形成される．いわゆる身体が覚えているというときの記憶である．ロボットシステムにおいては人間による教示などによりロボットが手順を覚える際に獲得を目標とする記憶がこれに相当する [33] ．

第 1 章にて取り上げた Anderson における日常生活タスクの特色に関する指摘では，各日常生活タスクを反復的に行うことにより得られる経験からくる合理性をもとに目標状態が決定されるというものであった．以上の記憶に関する分類ではこのような経験は長期記憶の一種であるエピソード記憶に該当する．

2.7 曖昧な指示・修正を随時受け付け対応可能な日常生活タスク実行システム

2.7.1 人による割り込みを随時受け付ける自律タスク実行システム

ロボットがコップを運ぶタスクを考える．ロボットがコップを運んでいる途中に人が割り込むとき，最も単純な中断の方法はロボットが行っている動作をすべて停止することである．このときコップを把持している手先の制御やロボット台車の移動制御を考慮していないシステムでは，コップを落としたり，ロボットの移動中の慣性により環境と衝突するなど危険であるばかりか，ロボットシステムにおける環境モデルと実世界の状況が大きく乖離するため，タスクの再開や別のタスクを行うための再計画を行うことも困難になる．したがって，ロボットは安全のため，割り込み指示を受けた時点のロボット自身の状況を踏まえて適切に中断状態に遷移する動作を行う必要がある．また手順が複雑になり，目標達成に時間のかかるようなタスクでは，割り込みを行った時点でタスクの進捗状況を可能な限り損なわずに途中から再開可能になるような中断を行う機能が求められる．

人による随時割り込みを実現するためには，ロボットは割り込みを受けた時点で人による指示内容を素早く正確に理解し，ロボット自身の判断を修正する必要がある．割り込み指示自体に割り込みを行う時点におけるロボットや環境の状態といったコンテキストが含まれるからである．

割り込み指示にはロボットの判断を人が修正する意図により行われることから，ロボットにおける失敗を分類することで割り込み指示の対象についてロボットシステムにおける Tso らの障害分類 [34] を参考にまとめる．

System Level Fault

タスクを行うロボットのハードウェア障害である．例えばマニピュレータの故障や通信ケーブルの断線などがこれに相当する．

Task Level Fault

ロボットが行動実行の中で生じた障害である．例えば物体把持タスクにおいて，物体認識の結果が適切でなく，把持に成功しなかった場合がこれに相当する．

ここで扱われる失敗は Task Level Fault である．さらに Task Level Fault における復帰方法について Tso らは詳細に分類している．

Retry

行動の再実行を行う．何らかの行動が失敗した場合，もう 1 度その行動を実行することで失敗の解決を図る．例えばロボットの移動先を歩行中の人間が一時的に遮り，ロボットが目標位置に到達不能（失敗）となったとき，移動動作を再実行することで目標位置への到達が達成できる可能性は高い．

New Parameters

ロボットの行動を構成する動作のパラメータが適切に設定されていないことで行動に失敗する．そのためパラメータを変更し動作をもう 1 度実行することで失敗の解決を図る．

Corrective Actions

動作の失敗を検知した際に，その失敗状態が計画された状態遷移におけるいずれかに復帰可能である場合，その状態へ遷移するような追加の動作を行い，さらにタスク実現行動を続行する．例えば物体把持の最中に把持している物体が滑ってしまった場合に再度把持を実行することがこれに相当する．

World Model Update

実世界の状況とロボットシステムが持つモデル世界との間に差分が生じた際にそれを更新する．例えば，物体把持タスクの中で，把持対象物が存在しないことが確認できればその対象物をモデル世界から削除することがこれに相当する．

Replan

行動計画による動作列を再計画することによってタスクの失敗状態へ遷移しないようにする．

Reconfiguration

ハードウェア障害によって，その行動が実行不可能になっていないかを確認する．例えば マニピュレータのアクチュエータ故障によって，把持が不可能になっていないかの確認がこれに相当する．

植田 [35] は失敗復帰の戦略を Tso らの分類を元に以下のように分類した．

Global Error Recovery Strategy

再度行動計画を実行することを Global Error Recovery Strategy と呼ぶ．この Global Error Recovery Strategy はロボットシステムが持つモデル世界のシンボル表現の変更と共に行われる．Tso らの分類では World Model Update および Replan に相当する．

Local Error Recovery Strategy

再度行動計画を実行しない失敗復帰を Local Error Recovery Strategy と呼ぶ．Local Error Recovery は行動計画の結果得られる行動列に対する変更として定義できる．Local Error Recovery Strategy はいくつかの Error Recovery Strategy に分類する．Tso らの分類では Retry, Corrective Actions および New Parameters などに相当する．

これはシンボリックな状態記述レベルでの分類であるため，以下のようにサブシンボリックなレベルに拡張する．

タスクレベル

タスク記述の目標状態レベルの割り込みである．例：他のタスクの割り込み．中断動作と割り込みタスク，割り込まれるタスクの再開を行う新たな目標状態を用いて行動の再

計画を行う．Global Error Recovery Strategy に相当する．

アクションレベル

行動計画によって得られる状態遷移列において人によって状態遷移を変更するレベルの割り込みである．例として実世界とロボットの環境モデルの差異によって生じる失敗の防止などが挙げられる．行動の再計画は行われなため Local Error Recovery Strategy に相当．

モーションレベル

ロボットの動作における具体的な関節角度列や目標手先位置を修正するレベルにおける割り込みである．アクションレベルの割り込みに伴って適切に状態遷移を変更するために必要とされる．

2.7.2 音声を介した自然言語による指示の曖昧性

音声による意思伝達は，伝達事項を言語を用いて記述し，記述に対応する音素列を発信することで情報を伝達する行為である．画像などの視覚的な情報に比べて，音声の性質上情報の伝達により大きな時間を要するしたがって，人間が音声を用いて対話を行うとき，伝達事項を説明するために必要な知識のうち，意識・無意識的にかかわらず，相手との関係において，お互いが共通して理解していると想定される知識についての説明は省略される．また音声と自然言語の変換，自然言語とエージェントの理解する知識表現との変換を経て得た情報にはその間において概念の曖昧性による指示の欠損や誤解が発生する可能性もある．このような音声を介した自然言語による指示の曖昧性は多義性とも呼ばれる．Russell らは多義性解消に用いる知識のモデルとして以下を挙げている [36] ．

世界モデル ある命題が世界で起こる尤度．

メンタルモデル 話し手と聞き手が相互に信じていると思う仮定に基づき，伝えようと意図する尤度

言語モデル 事実を伝える前提のもとある単語列が選ばれる尤度

音響モデル 話し手がある単語列を選ぶ過程で特定の音列が生成される尤度

この分類を意識してロボットに対する人の指示における曖昧性を階層的に分類すると以下のようになる．

音響レベル 音声認識の誤認識による曖昧性である．音声認識とは音声データから音響モデルを利用して一致度の高い対応する自然言語の単語列を探索する手法である．音声に対応する自然言語語彙の探索空間を小さくすることによりより単語の一致尤度を大きくすることが可能となる．

言語レベル 人の指示に対する仮定が得られた元で指示の実行に必要なパラメタの省略や欠損による曖昧性．省略されたパラメタを現在の状況から補完したり，未知概念を割り当てることにより解決が可能となる．

メンタルレベル 指示内容について予備動作など本来必要な前提知識を省略することに関する曖昧性．指示内容に基づき指示タスクの実行可能性を評価，予備動作を推論することにより省略指示やパラメタの補完が可能となる．

世界レベル 指示内容に含まれる未知概念により発生する曖昧性．未知概念の場合にもロボットがタスクを実行可能になるような追加情報の取得を行う必要がある．

音声を用いて正しくロボットエージェントが人の指示を理解するためにこれらの曖昧性について解決する機能が必要とされる．

2.8 長期経験に基づくタスク具体化機能を備えた日常生活支援ロボットシステムの構成

図 2.3 に本論文における提案システムの全体図を示す．システムを構成する要素について説明する．

- エピソード記憶を用いた長期経験の蓄積機能
あいまいな目標状態記述からロボットが実際に行動計画時に解釈可能な目標状態へと具体化するために，ロボットは概念的な知識に加えて，ロボットがタスクを行う各環境に依存した局所的な合理性に基づく知識が用いられる．事前に与えられることのない知識を，ロボットが日常生活環境にいるうちに知覚した経験をもとに形成するためのための機能である．
- 長期経験を用いた環境の局所合理性に基づく知識の自律獲得機能
環境における長期的な経験から，環境において局所的な合理性に従う知識を自律的に得るための知識のモデル化と経験からモデルの獲得を行うための機構である．
- 環境に非依存な状況レベルの日常生活タスク記述と局所合理性に基づく具体化機構
朝の食卓に並ぶ料理や食器が千差万別であるように，日常生活環境において概念的には同じタスクであっても実際にタスクを遂行するための手順や目標となる状態は家庭環境や環境に暮らす人の文化，時間など無数の要素によって決定される．従来ではタスクの目的は静的に与えられていたため，このようなタスクの具体的な目標状態がタスクを指示された時点ではじめて決定可能であるような性質を持つ日常生活タスクを表現するための記述である．
- 曖昧な指示・修正を随時受け付け対応可能な日常生活タスク実行システム
ロボットが行う意思決定や動作に対して，ロボットと共生するユーザから随時割込を受け付けるための機能である．ロボットがタスクを行っている最中でもユーザの割込を検知するための監視モジュール，割り込みにより判断を変更するための対話や直接接触を通じたインタラクションを行うためのモジュールに加えて，インタラクションを行う際に発生しうる，ユーザとロボットシステム間での意図伝達の曖昧性を解決する機能が含まれる．

2.9 おわりに

本章では、日常生活タスクを自律的に行うロボットシステムの構成に必要な要素や概念について考察しまとめた。日常生活タスクを自律的に行うための課題として日常生活タスクを構成する知識の環境における局所性について指摘し、今まで与えられてきた固定的で環境に依存したタスク記述を環境固有の知識に非依存な形で記述可能にし、経験から得た環境固有の局所的な合理性に従う知識をもとにタスク記述をロボットがタスクを実行する各環境に応じて具体化する仕組みの必要性を指摘し、これを可能にするシステムの構成法について提案した。また日常生活環境の特徴として人との共生環境であることに着目し、誤りを含む可能性のある経験から得た知識を人が随時指示・修正することが可能であるロボットシステムの構成法と、人の指示における曖昧性やその解決法について提案した。

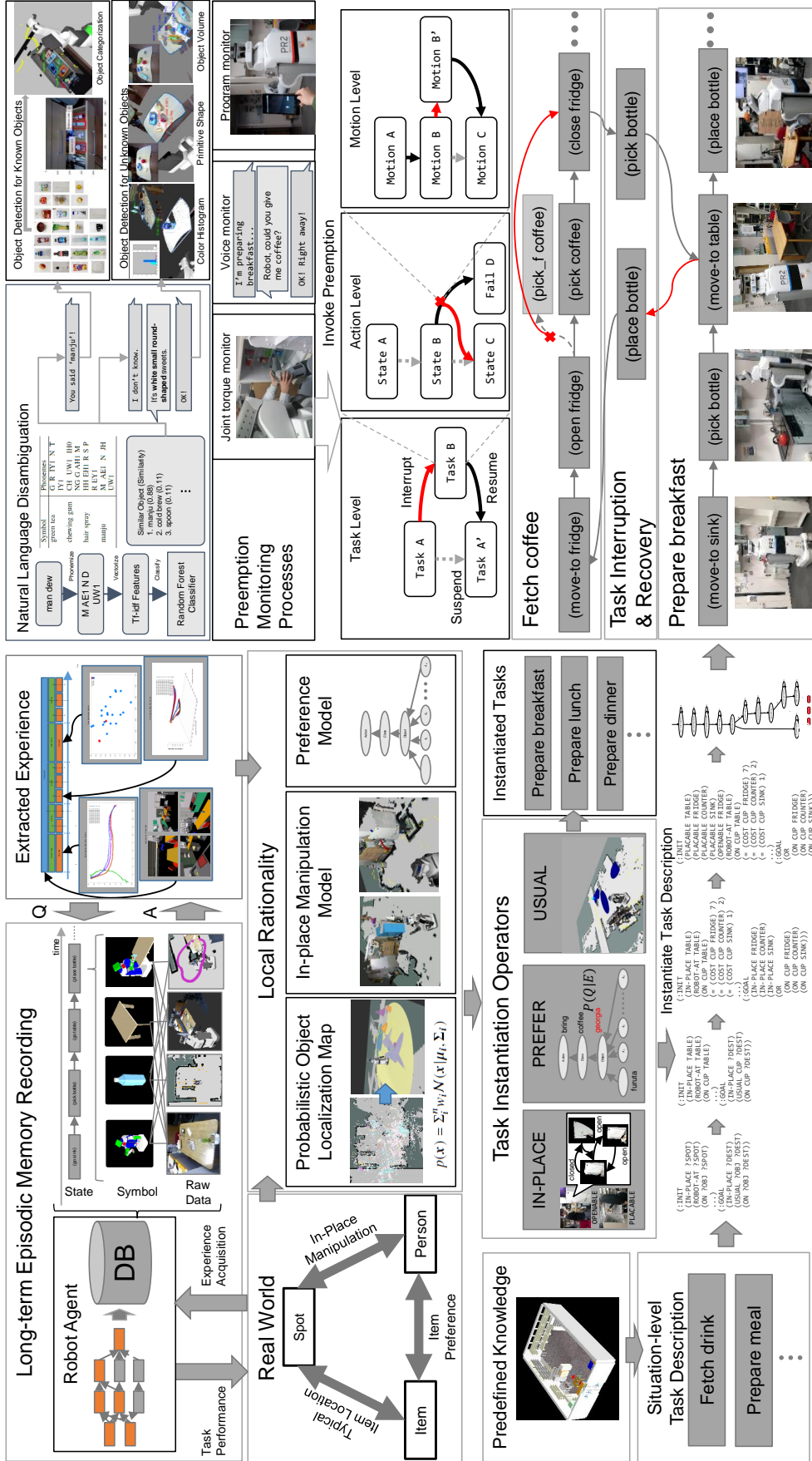


図 2.3: 本論文において提案するシステムの構成図

第 3 章

日常生活支援タスクを行う自律ロボットシステムの構成

3.1 はじめに

本章では、まずはじめに本研究における実験に使用したロボットの説明を行い、次に本研究で提案するシステムを構成するために必要なアーキテクチャ・構成要素について説明する。

3.2 本研究で用いたロボットハードウェアとロボットシステム

実験に用いたロボットハードウェアの構成と、ロボット開発ミドルウェアである ROS(Robot Operating System)[37] において実現されているロボットシステムについて説明する。

本研究では日常生活支援ロボットとして米 Willow Garage 社によって開発された全方位台車型双腕ロボットである PR2(Personal Robot 2) を元に、筆者らが計算機・センサ・通信装置などを交換し機能を向上させたものを用いる (図 3.1)。後者を前者から区別するために JSK-PR2 と呼ぶ。PR2 は Salisbury らによって開発された PR1(Personal Robot 1)[38] におけるカウンターバランス機構を基礎として、各種センサやベース移動などの機能を強化した物である。そのハードウェアは無線 LAN ルータやカメラなど、既製のサードパーティハードウェアを統合することで、比較的ユーザが多い製品から構成されるため、問題が発生しても膨大なオープンソースコミュニティにおいてサポートや解決策を知ることが出来る点でメンテナンスの難易度・期間ともに優れている。

3.2.1 等身大日常生活支援ロボット PR2 のハードウェア仕様

PR2 のハードウェア仕様の詳細は PR2 マニュアル [39] に記載されている。PR2 本体は重量約 220kg であり、高さは 1330mm であるが、腰部にある直動可動リンクにより、最大 1645mm となる。PR2 の両腕はそれぞれグリッパーを含めて 8 自由度で 1.8kg までのものを腕に載せることが出来る。この機構によって家事支援タスクなどを行う際の細かな操作を行う

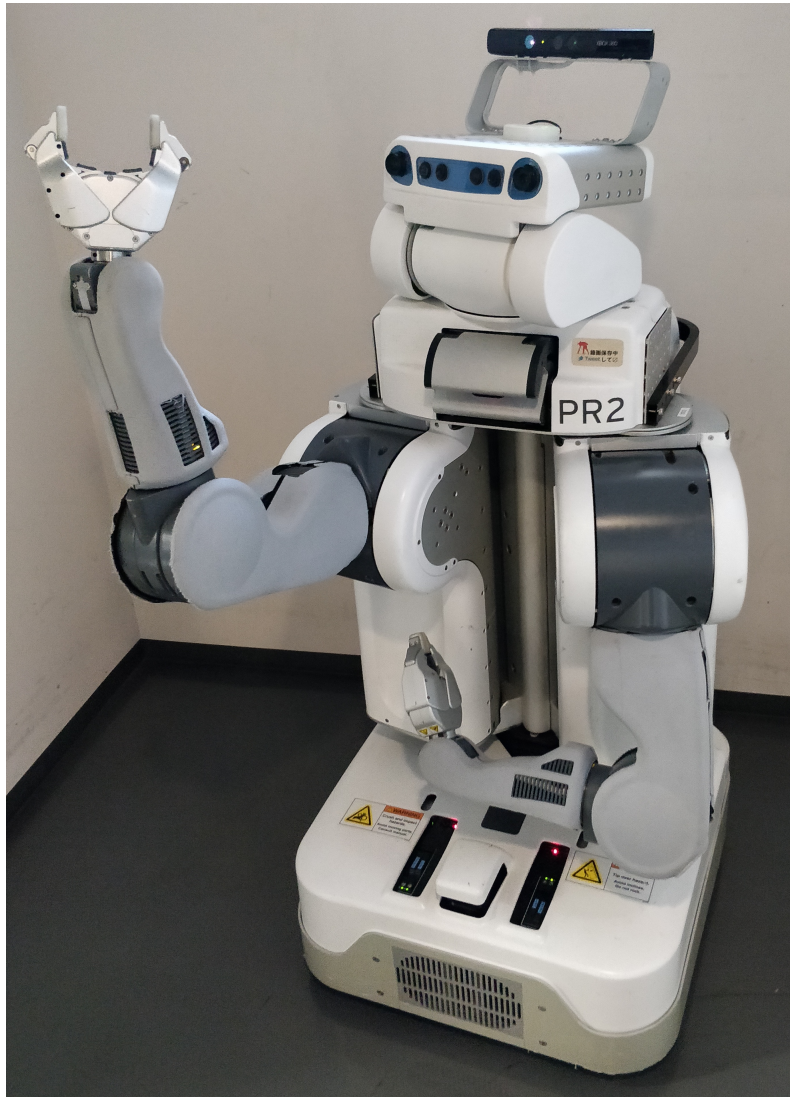


図 3.1: 移動台車型等身大日常生活支援ロボット JSK-PR2

事ができ、現実世界において物体の姿勢を 1 意に特定するためのパラメタ数である自由度よりアームの自由度の方が大きく、PR2 の両腕はある目標位置に対して冗長性を有している。

図 3.2 は Diankov らによる動力学ライブラリ OpenRAVE[40] を用いて算出された、PR2 ロボットの双腕のうち右手先の到達範囲を可視化した様子である。領域の色はその領域における到達可能性がどの程度であるのかを表しており、ある目標手先位置に対して収束した関節角度列の数が多いほど赤色になっている。

PR2 ロボット全体の関節自由度は、移動台車部分を除いて 20 である。詳細を以下に記載する。

- 胴体 1 自由度 (直動リンク)
- 頭部 2 自由度
- 腕部 7 自由度 (計 14 自由度)

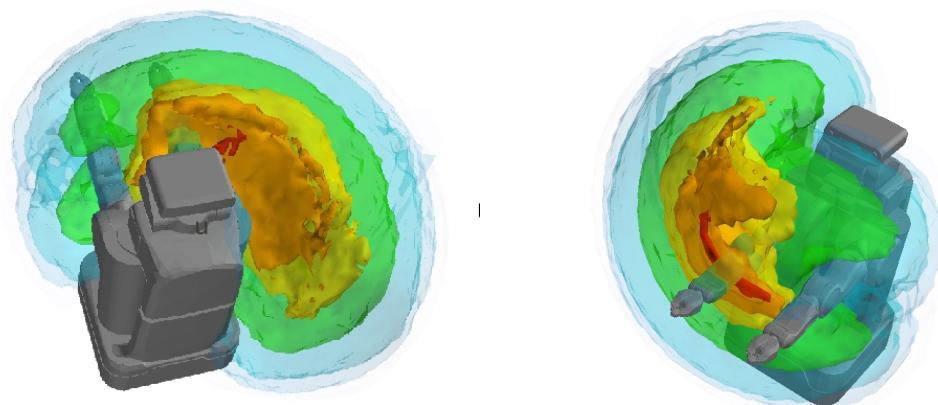


図 3.2: PR2 ロボット右腕の運動学的到達範囲 (OpenRAVE Tutorial より引用 [41])

- ハンド 1 自由度 (直動式グリッパー)
- 距離センサ駆動台座 1 自由度

表 3.1: PR2 の関節軸仕様 ([39] を改変) . Velocity の単位は rad/s または m/s である.

Joint Name	Type	Limit(+)	Limit(-)	Velocity	Torque (Nm or N)
torso_lift_joint	prismatic	310mm	0mm	0.013	10000
laser_tilt_joint	revolute	85°	-45°	10.00	0.65
head_pan_joint	revolute	168°	-168°	6.00	2.65
head_tilt_joint	revolute	60°	-30°	5.00	15.00
l_shoulder_pan_joint	revolute	130°	40°	2.10	30.00
r_shoulder_pan_joint	revolute	40°	-130°	2.10	30.00
{l,r}_shoulder_lift_joint	revolute	80°	-30°	2.10	30.00
l_upper_arm_roll_joint	revolute	224°	-44°	3.27	30.00
r_upper_arm_roll_joint	revolute	-44°	224°	3.27	30.00
{l,r}_elbow_flex_joint	revolute	133°	0°	3.30	30.00
{l,r}_forearm_roll_joint	continuous	-	-	3.60	30.00
{l,r}_wrist_flex_joint	revolute	130°	0°	3.10	10.00
{l,r}_wrist_roll_joint	continuous	-	-	3.60	10.00
{l,r}_gripper_joint	prismatic	86mm	0mm	0.20	1000

3.2.2 日常生活支援ロボットにおける環境センシングモジュール

日常生活においてロボットが作業を行うためにはロボットが置かれた環境において操作対象の幾何形状や構造, 回避すべき障害物や人間などを正しく認識する必要がある. 環境を知覚す

のためのセンサとして、大きく視覚センサと聴覚センサ、接触センサに分けられる。

このうち最も利用されているのが視覚センサであり、それぞれ1次元平面情報を得ることができるレーザ測距センサ、2次元画像情報を得ることが出来るカメラセンサ、センサからの奥行き距離情報を得ることが出来る、2次元画像を得るカメラセンサと組み合わせることで3次元幾何情報を点の集合という形で得ることが出来る3次元点群センサなどがある。

聴覚センサとしてはマイクロフォンが挙げられる。マイクロフォンを複数並べて配置することで音のマイクロフォンへの到着時差や大きさの違いから音源定位などを行う事が出来る。

また接触センサではセンサに加えられた力やモーメントを、金属の歪みなどを利用して計測する事ができる力センサの他にもモータへかかるトルクをモータに付けられた電流センサから計算して求めるものなどがある。

以下では本研究に用いたロボットにおいて装備された環境センシングモジュールの特性について説明する。

レーザ測距センサ

日常生活支援ロボット PR2 には Hokuyo 社製レーザ測距センサである Top-URG(UTM-30LX) が2つ取り付けられている(図 3.3)。このセンサは内部からレーザを照射し、照射したレーザ光が物体に反射してセンサに再び入力されるまでの経過時間 (TOF:Time of Flight) を元にセンサから物体までの距離を算出している。センサの内部に回転体がありレーザを照射する方向を変えながら距離を測定しているため、レーザ光自体は1次元の点であるが、2次元平面上を走査することが出来る。計測可能範囲は最大 30m までで、平面上の 270° の範囲を角度分解能 0.25°、距離分解能 50mm、周波数 40Hz で計測することができる。なお、このセンサに使われているレーザは FDA の基準で最も安全な Class 1(通常の使用における危険は無い)であり安全であることが確認されている。

このセンサのうち一つは地面と水平になるよう台車に固定されていて Base Laser と呼ばれる(図 3.4)。Base Laser はロボットのリンクが動いても地面からの相対位置は不変であるように設置されている。ロボットが環境における自身の位置を特定するために用いる自己位置同定においては、ロボットがセンサから得た情報から時間的に不変な情報と地理的なロボットの位置関係を用いて地図情報を予め作成し、現在のロボットのセンサ値からロボットの現在位置を推定するが、このときセンサがロボットの可動リンクより先にある場合には、ロボットが同じ位置にいるにも関わらずロボットの姿勢によって得られるセンサ値が変わってしまう。したがって、Base Laser が地面から可動リンクを経由しないことはロボットが自身の位置を得るために重要である。

もう一つのレーザセンサは、内部の回転体によって水平に2次元走査する機構のものを、垂直方向に回転するチルティング機構を有するリンクの先に設置することで、3次元空間における形状測定ができるものである。これは Tilting Laser と呼ばれる(図 3.5)。計測する平面の振幅は仰角 45° 俯角 90° である。

さらに IMU が Tilting Laser に隣接して取り付けられている。加速度の計測範囲は $\pm 5g$ 、ジャイロ計測範囲は $300^\circ/s$ である。



図 3.3: Hokuyo Top-URG



図 3.4: ロボット台車部に取り付けられたレーザーレンジファインダ



図 3.5: ロボット胸部に取り付けられた回転機構を伴うレーザーレンジファインダ

カメラセンサ

カメラは頭部上方に 1 台，頭部全面に 5 台，両腕に各 1 台の計 8 台が取り付けられている．頭部上方の深度カメラは USB 接続であるが，他のカメラはすべて Ethernet ケーブルによってロボットの内部ネットワークに接続されており，EtherCAT プロトコルを介してロボット内蔵の PC によって画像を取得している．頭部のカメラ 5 台を説明する（図 3.6）．広角ステレオカメラは，画素数が 752×480 で，画角はおよそ 90° のカラーカメラである．望遠のステレオカメラは，画素数が 752×480 で，画角はおよそ 55° である．なお望遠ステレオカメラはモノクロカメラである．それらステレオカメラと並んで Allied Vision Technologies 社製高解像度カメラ Prosilica が 1 つ取り付けられており，画素数が 2448×2050 で，画角はおよそ 36° のカラーイメージセンサを搭載したカメラである．

また，左右の前腕部にも手元を写すことができるカラーカメラが取り付けられている（図 3.7）．画素数が 752×480 で，画角はおよそ 55° である．



図 3.6: 頭部に搭載したカメラ

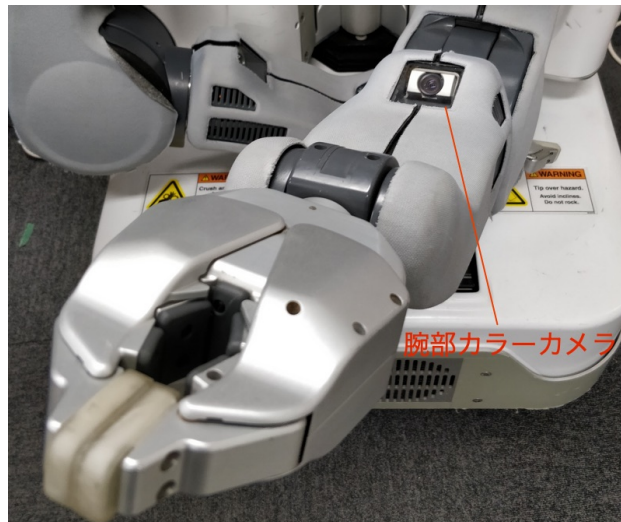


図 3.7: 腕部に搭載したカメラ

3次元点群センサ

頭部上方には、3次元点群センサとカメラ、複数の音声マイクロフォンが一体となったセンサモジュールである Microsoft 社製 Kinect[42] が取り付けられている (図 3.8) . Kinect の測距センサは Light Coding と呼ばれ、赤外パターン光をカメラ前方に照射し、物体に当たって生じたパターンの歪みを赤外線カメラによって読み取ることで物体までの距離を計測している . 距離情報は深度を各画素における数値の大きさとした単色 2次元画像として表現され、赤外線カメラとその隣りに取り付けられたカラーカメラのカメラパラメータや位置関係を計算し、カラー情報と距離情報を同期することで、色つきの点の集合である 3次元点群情報を生成している . カメラはそれぞれ 30fps で距離センサの取得可能範囲は 0.8m から 4.0m までであり、カメラセンサ位置から前方に対して水平方向に 57° , 垂直方向に 43° の範囲の物体の距離を測定できる . このセンサでは赤外光を物体に対して照射しているため、ステレオカメラなどのよ

うに2つのカメラの視差を用いて3次元形状を求める手法では対応する点を見つけることが難しかった、色特徴のない物体でも正確に形状を測定できることが特徴である。

頭部全方位型マイク

JSK-PR2 ロボットにおいて、頭部上方 Kinect の前方には Seeed Studio 社の ReSpeaker Mic Array v2.0 と呼ばれる全方位型のマイクが取り付けられている(図 3.8)。Respeaker Mic Array v2.0 は、61dB の S/N 比を有する MEMS 型デジタルマイクロフォン ST 社製 MP34DT01-M を水平に 90° ごとに円状に並べた 4 チャンネルマイクロフォンアレイと、XMOS 社製 DSP モジュールである XVF-3000、および 12 個のカラー LED インジケータからなる USB 接続型の音声入力デバイスである。XVF-3000 の機能を用いることで、雑音環境下での発話検知機能、音源方向検知、ノイズ除去機能を備えている。

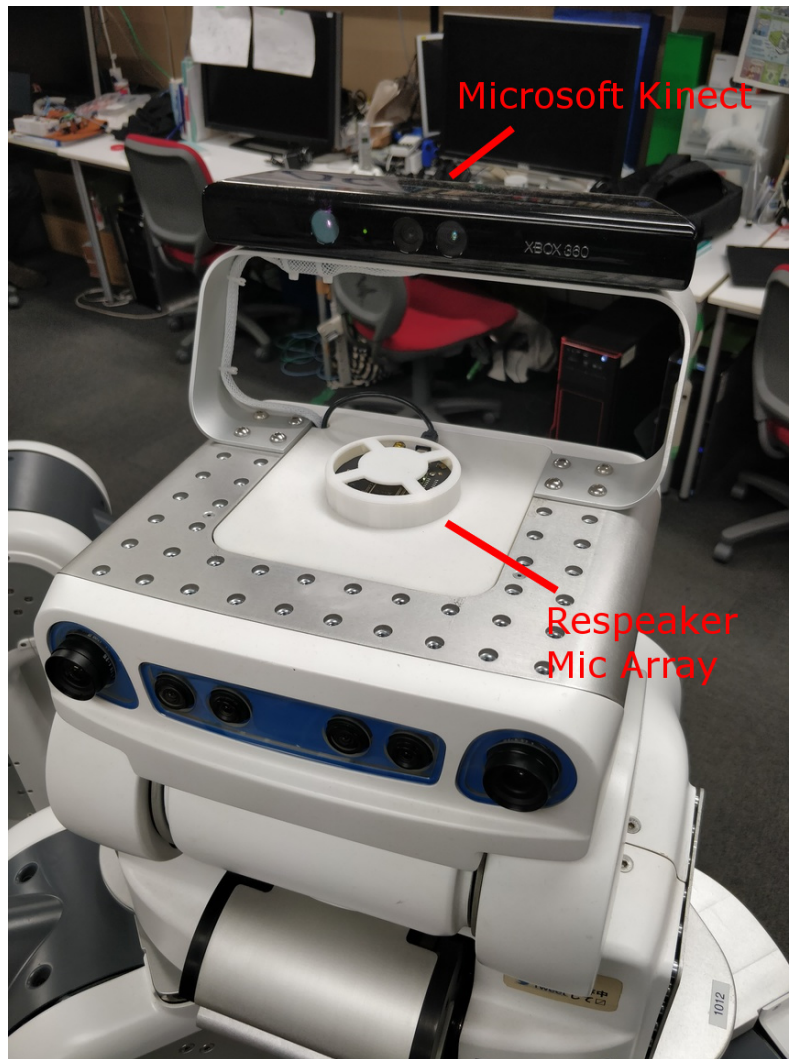


図 3.8: Kinect/Respeaker を頭部に搭載した PR2

3.2.3 日常生活支援ロボットの計算装置

JSK-PR2 では脚部に 2 台の計算機を内蔵している（以降それぞれ c1, c2 という）。それぞれには Linux ディストリビューションの 1 つである Ubuntu 14.04.5 LTS が用いられており、メモリは各計算機につき 32GB、CPU は単一コアあたり 3.80GHz の演算能力を備え、4 コア 8 スレッドの演算装置を搭載した米 Intel 社製 Xeon E3-1275V5 を搭載している。また JSK-PR2 にはロボットの脚部前方にリムーバブル HDD と呼ばれる着脱式の HDD ドライブがあり、2.5 インチのストレージが各 2 つそれぞれの計算機に接続されている。モバイルロボットのシステムにおいて扱われるセンサ情報を逐一ネットワークを通じて通信することはネットワークの帯域を大きく消費するどころか、移動中に通信が切断される可能性があり、システムが不安定になる。動作中の認識結果や動作ログ情報などは通信によって逐次得る代わりにこのリムーバブル HDD に保存することで、データ通信を介することなくロボット本体から取得することが出来る。

c1 はロボット体内における主計算機となっており、モータの制御のようなリアルタイム性の高いプロセスやセンサの処理などロボットのほぼ全ての機能を提供している。EtherCAT 通信のようにリアルタイム性の高いプロセスを安定的に動作させるため、lowlatency カーネルを使用している。

c2 は c1 に対する副計算機となっており、システム領域を記憶装置に持たない。ブート時に DHCP による IP 割当てと TFTP によるデータ通信によって、c1 で使用されている Linux カーネルおよびオペレーティングシステムを利用し、ネットワークブートすることによって、オペレーティングシステムを起動している。また、システムおよびユーザ領域は UnionFS[43] ファイルシステムを用いて c1 で利用されているシステム・ユーザ領域を読み込み専用でマウントし、さらに別の書き込み可能なブランチをマージすることでコピーオンライト環境を形成することで、c1 と c2 の間で基本的なソフトウェア環境を同期させながらも各コンピュータごとの独自の設定を可能にしている。この構成を実現する際の制限として、コンピュータの起動に順序依存がある。

また、レーザ測距センサや ReSpeaker Mic Array v2 などはリアルタイム性の高いモータ制御等の処理が多い c1 計算機に対する負荷分散を目的として c2 に取り付けられている。また、頭部 Kinect センサは USB コネクタの接続制限により c1 計算機に接続されているが、センサデバイスより得られる深度画像と RGB カラー画像から、3 次元点群を生成するための処理は c1 計算機で行わず、画像をネットワークを介して c2 計算機へ転送したうえでやっている。このように 2 つの計算装置を用いることで、モータ制御などの高いリアルタイム性が要求される処理と計画や認識などのそれ以外の処理をそれぞれ分散させることが出来る。

3.2.4 家事支援ロボットのためのアクチュエータ機構

日常生活環境では物体把持や運搬など、ロボットが環境に対して何らかの操作を伴う事がほとんどである。このとき単にロボットが認識によって予測した目標位置にロボットの手先が向かうようにアクチュエータを動かす制御のみである場合は、認識誤り、アクチュエータの制御誤差などに対して冗長性を確保することが出来ず、過大な負荷により環境中の物体やロボット自身のモータなどを破損する危険性がある。

そのため、誤差がある場合にも環境やロボットを保護するような仕組みがハードウェアでも補償されることは有用である。本研究で用いられたロボット PR2 では、腕部のいくつかの関節と頭部の関節に使われているモータにおいて、負荷トルクに応じて関節角度を変化させる制御が行われており、環境との接触に対して馴染むようなモータの動作をすることでモータに過大な負荷がかからないようバックドライバビリティを備えている。そのため従来の剛性の高いロボットにおいて必須であった衝突判定をほとんどの場合で行う必要がなくなった。

また通常ではロボットの腕部では高い自由度を持たせるためにモータをチェーン上に設置することから重量が極めて大きくなる。そのため腕部の自重を支えるために腕部は固く、人間などに衝突したことを検知する為には力センサなどを入れる必要があったが、PR2 においては腕部の重量を補償するカウンターバランスが肩部に備わっているため、わずかなトルクで腕部を持ち上げることが可能となっている。そのため人間によってロボットのアームが力を受けた際にもアクチュエータへ過負荷を与えることなく馴染むことが出来、モータ保護や安全性の確保が行われている。

PR2 の全てのモータは EtherCAT[44] 通信によって制御されている。EtherCAT とは、Ethernet for Control Automation Technology の略であり、Beckhoff Automation 社によって開発されたリアルタイム通信が可能な工業用通信規格である。Ethernet ネットワークによって繋がれた各デバイスのうちマスタとなるコントローラ 1 つのみがデータを送信する。TCP/IP といった通常の Ethernet 通信との違いは、スレーブとなる各デバイス（アクチュエータなど）ごとにデータを送るのではなく、マスタが送信したデータを各スレーブが受信後透過的に他スレーブへ送信することで、ネットワーク上に仮想的なリングネットワークを作り、マスタが最終通信結果を受け取る点である。各スレーブはデータ中の自身に割り当てられた領域へのオフセットを一意に決め、その領域だけのデータ送受信をオンザフライで行う。したがって、ハンドシェイクなどの複雑な通信が不要である他、マスタはデータの往復所要時間を計測し遅延時間を推定でき、スレーブとの同期精度を向上することができる。

3.2.5 日常生活支援ロボットにおけるバッテリー設計

PR2 は Inspired Energy 社製 14.4V, 6600mAh のリチウムイオンバッテリーを 16 個搭載しており、最大およそ 4 時間の連続駆動が可能である。また外部電源供給の有無に応じて電流の入力元を自動で選択する機構により、電源を切ることなく連続運用が可能となっている。各

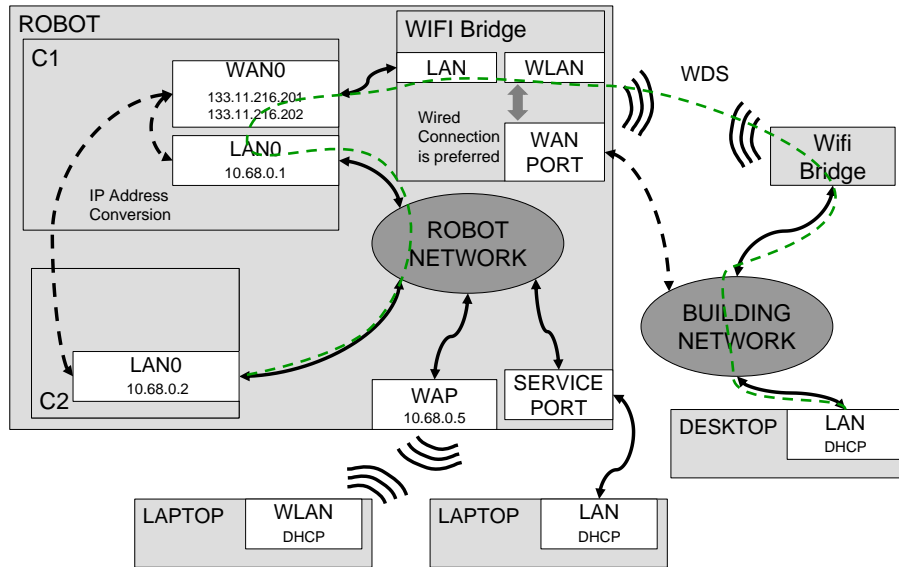


図 3.9: JSK-PR2 のネットワーク構成

バッテリーモジュールには制御基板が組み込まれており，RS-232C によるシリアル通信を通じて，バッテリーセルの温度や充放電サイクル回数などの履歴，現在の充電容量などバッテリーの状態を主計算機から取得することができる．タスクを実行している途中にバッテリー容量が低下した場合，タスクの実行を中断するなどの機能に利用可能である．

3.3 日常生活支援ロボットのためのソフトウェア構成

3.3.1 ネットワーク通信の構成

JSK-PR2 はネットワークへの接続として外部に向けて有線 LAN/WAN と，802.11ac 規格に基づく無線 LAN アクセスポイント及び 802.11ac に対応したクライアントを有しており，状況に合わせてインターネットに対して接続方法を柔軟に選択できる．無線 LAN による通信では環境に固定された無線 LAN アクセスポイントとロボットに内蔵されたアクセスポイント間で WDS(Wireless Distribution System) 機能を利用し，無線通信区間を有線通信同様にリレーすることによって実現されている．アクセスポイントは RouterOS によって基本的なルーティング機能が実現されているが，スクリプト機能により予め決められたルーティング機能の変更をプログラムすることができる．この仕組みを利用して，アクセスポイントに対しての有線 LAN ケーブルの接続・切断をトリガとして，無線・有線経由の通信を切り替えている．従来では同一のネットワークに対して複数の通信経路が存在する場合，計算機に対して複数のアドレスを割り当てる必要があるが，この切り替えを行うことにより，同一の IP アドレスを計算機に割り当てたまま通信の切り替えを行うことができる．無線通信では約 400Mbps，有線通信では約 1000Mbps における通信を行うことができ，用途に応じて使い分けることができる (図 3.9) ．

3.3.2 分散型ロボットミドルウェア

家電操作や物体認識・計画など複雑な手順からなる家事支援タスクをロボットが自律的に行うためには、動作や計画と言った各要素を再利用可能な形でモジュール化し、モジュールを組み合わせることで複雑なタスクを実現していく手法が生産性・メンテナンス性向上の観点で有用である。

オープンソースロボット開発ミドルウェア ROS

本研究で用いた日常生活支援ロボット PR2 はオープンソースなロボット動作ミドルウェアである ROS(Robot Operating System)[37] のモジュールとして制御器や計画器などの構成要素を組み込む事で行動制御を行っている。本研究のシステムもその大部分を ROS のモジュールとして既存のロボット動作モジュール群に統合する形で実現されている。

Publish-Subscribe 型の非同期通信とリモート関数呼び出し型の同期通信をサポートする通信ライブラリを中心とするプラットフォームである。

ROS は以下の 5 つの設計思想を目標としている。

- Peer-to-peer
- Tool-based
- Multi-lingual
- Thin
- Free and Open-Source

ROS の最大の特徴はこのうち Tool-based であること、すなわちロボットに高度なタスクを行わせる上で必要な各構成要素をモジュールとして用意することで、これらを組み合わせて単純な機能から大規模な制御に至るまでを、新たに特定の用途に特化したプログラムを作成することなく実現することが出来るということである。ROS のモジュールはオープンソースプロジェクトとして公開し、世界の誰でもが利用することができるようなソフトウェア頒布の仕組みも含まれている。そのため開発者はシステムを実現する上で自分が実装したい機能だけの開発に集中することが出来、開発期間・難易度を大幅に軽減することに成功している。

ROS で動作するモジュールはプロセスを同じくせず、各々が非同期・並列に動作するためそれぞれが別のプログラムとして独立している。これらのモジュール間における通信のために ROS ではまず始めに roscore と呼ばれるサーバプロセスを起動する。roscore はネームサーバを備えていて、各モジュールは通信を行いたいモジュールの情報をネームサーバに問い合わせることでモジュール間の通信を開始することが出来る。また roscore はパラメータサーバの機能も備えていて、ここに登録されたパラメータは同じ ROS のネットワークに属する全てのモジュールが参照・変更出来る。各プロセスは roscore から目的の通信を行うことができる URI リストを取得し、それぞれのプロセスと直接通信を行う。ROS におけるプロセス間通信やパッケージ情報を扱うユーティリティツールが各種用意されている。

非同期並列プロセス分散型ロボットシステムにおけるプロセス間通信

ROS のような分散プロセス型のロボットシステムにおいては、各動作プロセスが並列に処理を実行する中で他のプロセスと適宜通信が行われる。通信されるデータの性質に応じてプロセスは同期的・非同期的にデータの通信を使い分けられている。図 3.10 にその概念図を示す。

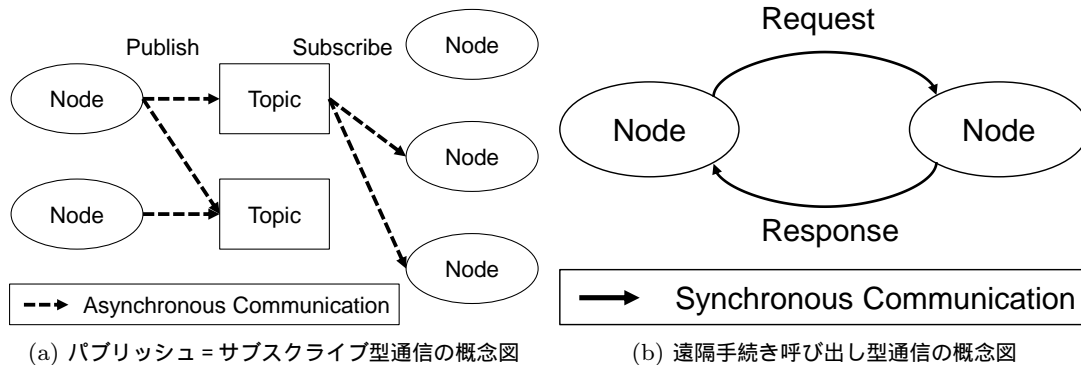


図 3.10: 分散プロセス型ロボットシステムにおけるプロセス間通信

非同期通信の仕組みとして広く使われているものとして、パブリッシュ=サブスクライブ型データ通信がある。パブリッシュ=サブスクライブ型通信 (図 3.10a) では、メッセージと呼ばれるデータ通信の単位を用いて非同期にプロセス間通信を行う。この手法はイベント駆動型とも呼ばれる。メッセージは、トピックと呼ばれる文字列をキーとしたデータチャンネルによってプロセス間で送受信される。1つのトピックには1つのメッセージ型がそれぞれ対応している。メッセージを受信すると予め登録されたコールバック関数が呼ばれ、受信ノードは受信したメッセージを入力として処理を行う。送信ノードは受信ノードにおいてデータが受け取られたかどうかを待たずに次の処理を行う。受信ノードにはトピックごとにキューを受信バッファとして持つ。単位時間あたりのデータの受信速度がコールバック関数の呼び出しによるデータの処理速度を上回った時、データは受信バッファには格納されず破棄される。

同期通信は (図 3.10b) のようにデータの受信側プロセスが送信側プロセスに対してデータの要求を行い、データを処理した結果を得る仕組みで、サーバ=クライアント型、遠隔手続き呼び出し型、RPC(Remote Procedure Call)とも呼ばれる。データの要求を行うプロセスは、呼び出されたプロセスにおける処理が終わり、戻り値を受け取るまで処理を停止するため、プログラム上は通常の関数呼び出しと等価に見える。

分散型ロボットシステムにおける3次元幾何関係の相互参照の仕組み

分散型ロボットシステムでは基準となる絶対座標系はなく、すべての座標系はある座標系に対する相対的な座標系として表現される。変換が行われる座標系の対をノードとすると、座標変換をエッジとする有向木構造となる [45]。任意の座標系変換はこの木構造を目的の座標系ノードまでを探索することで実現される。また、座標系の変換には変換が有効となる時限が付

加されているため、キャッシュとしてデータが生存する限り過去に遡って座標変換を得ることができる。この機構によってデータの主保存先としてのサーバが存在しない分散プロセス環境においても、負荷が1つのノードに集中することなくプロセス全体でロボットの持つあらゆる座標の関係を参照することが出来る。座標系の数が増えても負荷がほとんど変わらないため拡張性に優れている。

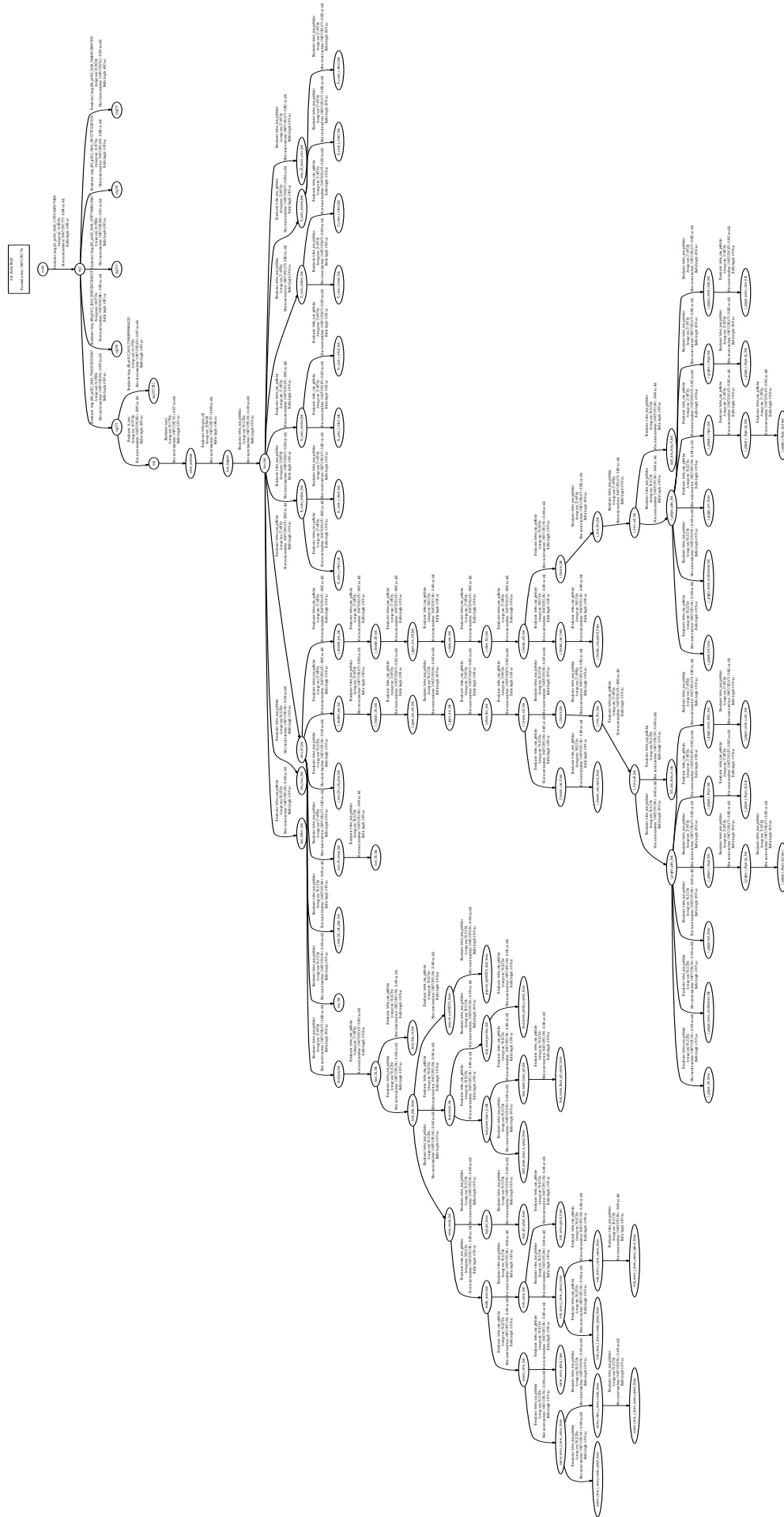


図 3.11: 本研究において用いられるロボットシステムの座標系変換系構造

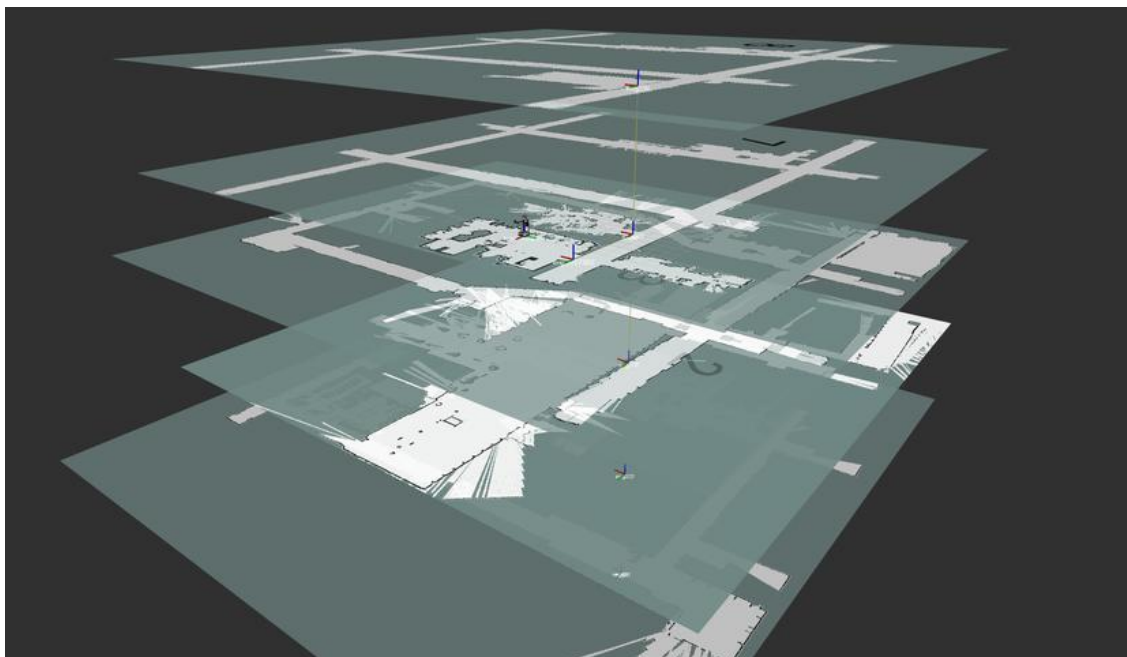


図 3.12: 多層階構造に対応した座標系と障害物地図

図 3.11 に本研究におけるロボットシステムにおいて扱われる座標系の本構造を示す。ロボットの自律移動経路計画などに用いる地図の原点は/map によって与えられる。自己位置推定の結果など、環境に固定された座標系から見た任意の地点は/map 座標系、および/map 座標系から静的に変換可能な座標系に相対して定義される。図 3.11 では/map は/eng2/7f から位置・回転変位が 0 の座標系変換が定義されている。ロボットは環境にあるエレベータを利用した階層の移動に伴い、例えば 2 階へ移動した場合は/eng2/2f から/map へ位置・回転が 0 の座標系変換を定義することで、自己位置推定モジュールなどへの影響を与えることなく、多層階に対応した位置表現を利用可能にしている (図 3.12)[46]。

分散プロセスロボットシステムにおける非同期リモート関数呼び出し

actionlib[47] は呼び出し側が処理を依頼し、呼び出される側が計算をして結果を呼び出し元に返す点でサービス型の通信と近いが、呼び出し元は呼び出される側が計算途中でも処理をブロックされない。その代わりに呼び出し元は呼び出される側で計算がどの程度行われているのかという進捗を知ることが出来る。また結果を受け取る前に処理の中断を指令したり新たな命令で上書きすることもできる点でより柔軟性がある。本研究における人間による随時割り込み機能の実現において利用されている。これはプロセス間で複数の役割を持つトピックを扱うことで実現している。

例えばロボットアームを動かすためのパッケージである arm_controller では、クライアントがアームの目標関節角度列をサーバへ送信する。サーバではロボットアームを現在の状態から目標の状態になるような軌道を、補間軌道を生成しながら計画し、アームのアクチュエータへ関節角度指令値を送信する。アームは指令値に従い動く。アームが目標状態になるまでの

間，呼び出し元は `arm_controller` 全体の状態や各時点における実際の関節角度列や目標角度列との誤差を得ることが出来る．最後にサーバにおける処理が完了すると，呼び出し元はその結果を受け取る．

3.3.3 日常生活環境におけるロボットの自律移動実現のための構成

日常生活環境において，環境の中でロボットが自分の位置を推定するための機能は単に経路探索のためだけでなく，行動計画における意思決定を行う上でも必須である．また日常生活環境には人間や壁，椅子や棚などを壊さないよう安全・安心に十分考慮した移動を行う仕組みが必要となる．また環境は動的に変化し，部分的にしか観測できないため，環境における地図の更新を随時行っていくことも求められる．Simultaneously Localization and Mapping(SLAM)[48]は環境に関する地図の生成と生成した地図中における自己位置の推定を同時に行うことで，ロボットの自律移動を実現しようとするアプローチである．障害物地図の表現としてはロボットが位置する平面をグリッド上に区切り，各セルが障害物に占有されているかどうかをロボットのセンサを用いて検出・マークしていく．今度はロボットのセンサ情報と地図中のセル占有情報に基づき，パーティクルフィルタなどの最尤推定によって環境中の自己位置を推定していく．レーザレンジファインダなどのセンサによって一時的に障害物があると判定された障害物地図上の領域は，安全上障害物がセンサによる検出空間から消失していても，一定時間が経過しないと移動経路として使用されない．一方で，例えばレーザセンサの距離情報とカメラ画像などにおける顔認識などを組み合わせて人間と人間以外の障害物を区別することで，歩行中の人間に対しては，人間の移動後に即座にロボットの移動経路として利用可能にするなど障害物地図は目的ごとに分割し多層的に用いられる．

自律移動を行うための経路探索も SLAM によって作成された地図を用いて行われる．障害物回避の多くは目標となる座標までの経路探索を大域的計画と局所的計画の2つに分割して解決する(図 3.13)．大域的計画ではダイクストラ法 [49] や A*アルゴリズムなどにより現在地点から目標地点までの探索を行う．最適化関数としては移動距離の最小化や移動速度の最大化などが考えられる．この探索は計算コストが比較的大きく，ロボットの移動モデル(加速・減速性能など)が考慮されないため，ロボットが現実に正確に追従することは難しい．また自律移動中台車に速度指令を与えるにあたって必要なのはロボットの現在位置周辺における経路のみであるため，計算に無駄がある．したがって，大域的計画によって，大まかな経路を計画し，探索空間をロボットの近傍領域のみに絞った局所的計画によってロボットの移動性能や障害物の動的回避を行う方式になっている．局所的計画において利用される探索アルゴリズムは Dynamic Window Approach(DWA)[50]などが挙げられる．

ロボットは地図上の目標座標やロボットに対する相対的な目標座標を与えることで自律移動を行う方法に加えて，ジョイスティックやタブレットの操作を入力とする目標速度指令に対しても移動を行う遠隔操縦機能を有する．遠隔操縦時のロボットと障害物との衝突を防止するために，速度指令入力に対して，一定時間後の予想目標座標を前述の自律移動機能に対して入力することで，衝突回避を実現している．

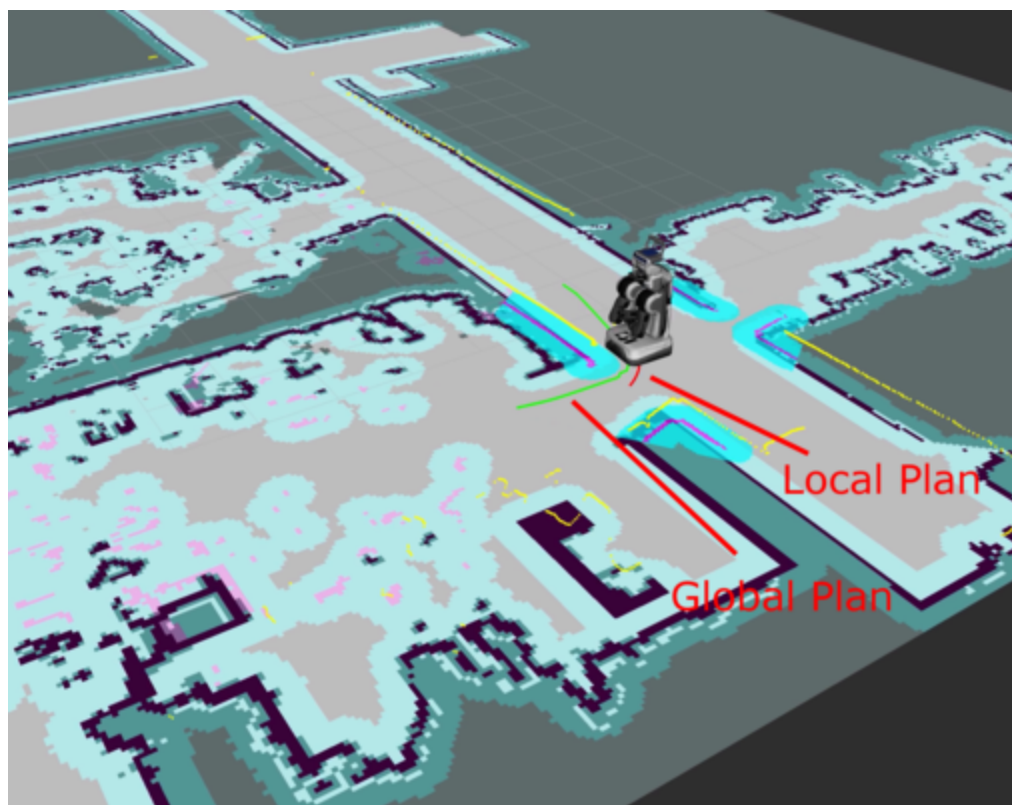


図 3.13: 台車移動経路の大域的計画と局所的計画結果

以上のような自律移動機能の最終的な出力は、全方位移動台車に対する水平 2 軸方向と回転の 3 次元速度ベクトルとなる。分散型ロボットシステムにおいてはモータ制御などの特別なプロセスを除き、各プロセスにおけるリアルタイム性が保証されないため、自律移動機能の出力は断続的にモータ制御プロセスに与えられることが前提とされる。このときモータ制御プロセスでは断続的な速度指令に対して補間を行う必要があるが、安全のため一定時間速度入力が与えられない場合、自動的に移動速度をゼロにして停止するような機能を備えることで、計算の高負荷時やセンサ情報更新の遅延などによりロボットが意図しない移動を行う事態を防止している。またロボットが充電中の時、充電ケーブルには電流が流れているため、そのまま移動を行うとケーブルやコネクタの焼損の危険がある。したがって、バッテリーへの電流入力を監視して、ケーブルがロボットへ接続されている時には速度指令をモータ制御プロセスに送信されないようにするフェイルセーフ機能を実現している。

3.3.4 日常生活支援ロボットシステムにおける動作制御機構

日常生活環境におけるタスクとしてテーブルの上にあるコップを取る行動を考える。ロボットは環境中のテーブルやコップの把持点を認識したあと、コップに対して手先を近づけ、グリッパを閉じることでコップを取る動作を実現する。コップに対して手先を安全に近づけるために、目標となる姿勢にロボットを動かす途中の姿勢を補間し、障害物に腕をぶつけないよう

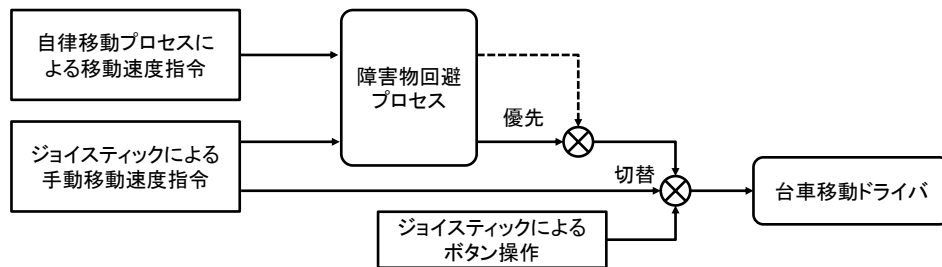


図 3.14: 複数の入力を受け入れる台車操縦用のシステム図

に動作の各時刻について環境と接触しないよう干渉回避を行う必要がある。このような動作経路の探索を高速に行うための手法として RRT-Connect アルゴリズム [51] などが挙げられる。環境との干渉計算を行う際は軽量な環境の障害物地図として Octomap[52] などが用いられる。

3.3.5 日常生活環境における操作物品認識機能

日常生活において物体運搬や家電操作は最も基本的な行動の一つである。物体操作を行うためには環境の中から操作対象となる物体を正しく認識する必要がある。このような物体認識問題はロボット分野だけでなく広く研究が行われている。

ロボットが搭載するセンサの種類に応じて認識手法や出力も異なる。カメラを用いた撮像によって得られるカラー画像をベースにした認識と、ステレオカメラや深度センサ、レーザーレンジファインダと可動機構を組み合わせたセンサなどにより得られる三次元点群を用いた手法に大きく分けることができる。

画像を入力とする認識器としては、画像領域から得られる特徴ベクトルを用いて対象物体を表す参照画像から得られる特徴ベクトルと比較を行う手法が有名である。特徴を算出する対象は HSV 色ヒストグラム [53] や HOG 特徴量 [54] などのように対象領域全体を用いるものや、SIFT[55] や ORB[56] などのように画像領域内において輝度変化などに頑強な点を予め選択して特徴量を算出するもの、点ではなく線 (エッジ) を用いてマッチングを行うものなどがある [6, 57]。これらの手法はマッチングの探索空間において、操作対象物体とそれ以外のものをよく分割できるような特徴を予め与えることによって機能する。特徴量の分類性能は対象物体だけでなく、環境の照明条件やセンサの特性にも影響を受ける。しかしながら、後述の深層学習でクラス識別モデルを直接得る手法では、学習データセットにない未知物体を扱うことが難

しい。

一方で、近年の演算技術の向上を背景に、ニューラルネットワークをベースとした認識器が多数研究され、ベンチマークテストにおいて高い認識性能を示している。Faster R-CNN[58] や YOLO(You Only Look Once)[59]、SSD(Single Shot Multibox Detector)[60] などに代表されるネットワークでは、畳み込み層 (Convolution Layer) が持つ画像の位置偏在性をプーリングにおいて保存する特徴を活かして、高次特徴ベクトル表現を用いた次元削減を経て認識対象を表す出力を直接求める手法である。ユーザは自身で特徴量を選択する必要がなく、対象物体を含むクラスの識別が行われた正解データセットを与え、学習によってモデルをフィッティングさせることで、認識機能を構成する。前述のネットワークでは画像を入力として、画像内の物体候補領域 (Region Proposals) と各候補領域のクラスラベルを出力とするものであり、一般物体認識と呼ばれる。対して、物体を矩形領域ではなく、画像の画素レベルで把握する物体認識問題はセマンティックセグメンテーションと呼ばれる。この問題に対する手法として FCN(Fully Convolutional Network)[61] や SegNet[62]、PSPNet(Pyramid Scene Parsing Network)[63] などがある。この問題は入力画像を要素の大きさがそれぞれ (C, H, W) (ただし C は画像のチャンネル数で RGB カラー画像では 3、 H, W はそれぞれ画像の高さと幅を表す) となる 3 階テンソルとみなし、出力は (L, H, W) の 3 階テンソル (L はネットワークの識別するクラス数) として、入力画像の各ピクセルに対して、そのピクセルが識別される物体である Confidence を表すベクトルとなるようなネットワークとしてフィッティングを行うことで構成される。多くはピクセルごとに最大の値を持つクラスがそのピクセルの属するクラスとして扱われる。セマンティックセグメンテーションでは画像中のピクセルごとに物体のクラスを識別できるが、複数の同じクラスに属する物体が重なりあうような場合に書く物体を識別することができない。この問題をインスタンスセグメンテーションという。Mask-RCNN[64] の手法ではまず物体候補領域を検出し、各領域に対してセマンティックセグメンテーションを行い、結果を結合することでこの問題を解決している。

三次元点群を用いる手法では認識対象となる物体の 3 次元幾何データをもとに、環境中の点群を部分的にマッチングする手法が考えられる。マッチング手法として代表的なものに Iterative Closest Point(ICP) アルゴリズム [65] が挙げられる。奥行き情報を持たない画像と異なり、三次元点群では物体を観測する方向や物体との距離などを考慮する必要はない。一方で深度センサやステレオカメラなどによって得られる点群情報はオクルージョンのない表面形状のみであるため、正確にマッチングを行うためには遮蔽領域の補間などを行う必要がある。

3.3.6 行動計画

複雑な手順や目的状態を持ち、環境に深く依存する家事支援タスクをロボットが自律的に行うためには失敗復帰高度な行動計画機構が必須である。

行動計画に関連する機能として以下の物がある。

smach 動作をノード、動作の結果による条件分岐をエッジとした階層構造をもつ

	ステートマシンを管理する [66] .
pddl_planner	シンボリックなタスク言語である PDDL[10] を用いて記述されたタスクから動作列を生成し, 条件分岐ノードを別プランとして解き, 結果をマージした状態遷移グラフを作る .
task_compiler	pddl_planner を用いて作られた状態遷移グラフに失敗動作をマージして条件分岐ノードを含む状態遷移グラフを作る [20] .

3.3.7 ヒューマンロボットインタラクション

日常生活環境ではロボットは共生する人間に常に気を配り, 人間の指示を受け取ったり, 人間に対してインタラクションを行うことで, 環境に関する知識を獲得できる機能を備える必要がある. 人間とインタラクションを行う上でロボットシステムに必要な機能を説明する .

Text to speech	自然言語の音声発話や音の発声を行い, 人間に対して注意の誘導や質問を行う .
Speech to text	音声信号から自然言語における発話を認識する機能である .
Mannequin mode	ロボットの腕関節に対して外部から力が与えられるときに馴染むように関節角度列を変更していく制御機能 [67]
Tablet	タブレット端末を用いたグラフィカルユーザインターフェースによるロボットへの指示や情報の表示を行う機能

3.3.8 行動制御

ロボットが何らかのタスクを与えられ, タスクの目標状態に至る動作列を計画している状態, または計画した動作を行っている状態をタスク実行中と呼ぶ. ロボットがタスク実行中ではない状態をアイドル状態と呼ぶ. ロボットがアイドル状態にある間にロボットは任意の行動をすることができるが, 分散型ロボットシステムにおいては複数の動作の並列実行によってアームなどのロボットの動作資源が競合する他, 音声によるタスク指示などロボットをアイドル状態からタスク実行中状態へ移行するためのロボットシステムへの行動指示に随時応答することができるようになっている必要がある .

本研究におけるロボットシステムでは, 以上のような要求に応じるために, アイドル時のロボットの動作を管理する `idle_behavior` プロセスを並列に動作させている. `idle_behavior` プロセスは以下の機能からなる .

Attention	ロボットの反応を引き起こす注意を環境を知覚するセンサ入力から検出する機能
Behavior	Attention を受け取ったときにロボットが開始する動作の単位. Attention の種類に応じて一つの Behavior が選択される
Launcher	Attention によって得られるロボットが向けた注意対象や, 注意対象との Behavior

表 3.2: アイドル状態の注意と振る舞い

Attention	条件	対応する Behavior
音源定位	パワースペクトルが閾値以上	音源方向へ頭を向ける
音声認識	特定の単語による呼びかけを受けた	音声対話を開始
人認識	人を認識した	人へ頭を向ける
時間経過	一定時間 Attention を受けていない	周りを見渡す
物体認識	物体が認識された	配置を確認する (第 5.3.2 節で説明)
物体・人認識	物体への人の接触が認識された	操作を確認する (第 5.3.2 節で説明)

を通じた相互作用によって必要になったときに何らかのタスクを開始する機能。タスクが開始されるとき、アイドル状態からタスク実行中状態に移行し、タスク実行を阻害しないよう Behavior は停止される。タスクが開始されたあとはタスクの実行状態を監視し、開始されたタスクが終了し再びアイドル状態に戻るときに Attention に基づいて Behavior を選択し、アイドル時の動作を再開する。

本研究におけるロボットシステムでは表 3.2 のように Attention と Behavior を与えている。

3.4 おわりに

本章では、本研究における題材となる日常生活環境において生活支援タスクを行う自律ロボットシステムの構成について、ハードウェア・ソフトウェア両面から提案システムを構成するために必要な機能、構成要素を本研究における実験に使用したロボットの具体的な説明を交えて説明した。

第4章

ロボットにおけるエピソード記憶に基づく長期経験の蓄積機構

4.1 はじめに

本章では日常生活環境におけるタスク計画を実現するための知識を獲得するための、長期経験蓄積機能の構成法について述べる。日常生活環境のように事前にタスクに関する知識を与えることができない環境においては、環境においてタスクを行っていくうちに知識を後天的に獲得していく仕組みが必要となる。ロボットがタスクを行う上で日々環境を知覚したデータを記憶し、類似した状況やその時にロボットが選択した行動とその結果を参照することで、ロボットは自身の行動を振り返ることが出来る。複雑なタスクシーケンスを実行する日常生活支援ロボットシステムは高度な認識・計画・動作機能の組み合わせにより実現されるが、カプセル化され、非同期・並列に動作するプロセスによって構成されるシステムにおいては各プロセスが参照できるものは入力となるデータに限られるため、環境状態を用いた高度な意思決定や他のプロセスの状態を用いて各プロセスが学習を行うことは難しい。本章ではロボットシステムにおいて扱われるデータを、ロボットの内的な環境モデル表現であるシンボリックな状態記述と関連付けて記憶する機能を導入することで、過去の時点における意味的な状態を手がかりにデータを抽出し学習を行うための方法を説明していく。また実世界ロボットならではの問題として計算・記憶資源の有限性にも注意しながら構成を明らかにする。本研究では経験の蓄積機能を、(a) ロボットシステムにおいて扱われる情報のうち蓄積するデータの選択、(b) 蓄積の対象となるデータをどのようにストレージへ保存するか、(c) 保存したデータを経験としてどのように抽出するかの3つに焦点をあてて説明していく。

4.2 関連研究

経験を扱う知能エージェントとして SOAR[68] や ACT-R[69] が挙げられる。両者ともにロボットシステムとしてというより、認知心理学的観点から人間の記憶メカニズムをモデル化することが目的とされているシステムである。SOAR では、人間の記憶メカニズムにおいて、

情報を一時的に保ち操作を行うための構造的な概念であるワーキングメモリが用いられた。ワーキングメモリは知覚から 50 ミリ秒以下の短期間保持される記憶 [30] で、SOAR ではシンボリックな記述のグラフとして表現される。またシステムとして行動目標に対して if-then ルールを複数与えておき、ワーキングメモリ上の状態とマッチしたルールを適用していく、プロダクションシステムに近いアーキテクチャを採用しているが、このようなシステムではある状況に対し適用できるルールが存在しないときにエージェントが行動を取ることができなくなってしまう。このような状況を SOAR では袋小路 (impasses) と呼び、適用可能なルールを行動目標に関係のないものまで対象を広げることで、解決を図ろうとするアーキテクチャになっている。ルール自体が増加することはなく、ルールの条件適用は並列・非同期に行われるため、複雑な手順を要するタスクでは目標到達のためのルール設計を慎重に行う必要があった。一方 ACT-R では if-then ルールの条件適用はシーケンシャルに行われる。エージェントが観察や自らの体験に基づきルール適用によって動作を管理していくうちに、ルール適用のチェーン法則性を監視することによって、より大きな複合ルールを生成していくシステムである。SOAR と異なり、現在の状態を解決するための知識が不足しているときの振る舞いについては考えられていない。

実世界ロボットシステムとして経験蓄積機能を搭載することを考える場合には計算資源やデータ蓄積先であるストレージの有限性も考慮に入れる必要がある。Winkler らは反射型行動エージェントを備えるロボットシステム上で現在のエージェントの状態とセンサデータを関連付けて保存する仕組みを提案している。このシステムでは蓄積対象のデータを固定的に与えた上で、データの種類ごとに適当な圧縮を行うことによりデータ容量の増加を小さくする機能を有していた。一方で、本研究において用いられるシステムのように、再利用性が意識され、多数のノードが非同期・並列に動作することで構成される分散ロボットシステムにおいてはタスクの目的や操作対象物体によって必要になる認識器や動作計画器が異なるため、予め蓄積するデータを固定的に与えることが難しい。そこで本研究ではノード間におけるデータの送受信の関係性を監視し、状況に応じて関係のあるデータを自動的に選択・蓄積する手法を提案する。

4.3 時系列データの構造化

人間は曖昧な情報や行動を選択するために必要な知識が不足した状況下でも、過去に経験した類似する状況から類推することで不足した情報を補い、適切な判断を行うことができるロバストな意思決定を行う機能を有する。人間は知覚した情報を単に時系列情報として保管するのではなく、経験した出来事の内容と、自身の感覚や心理的状态などの付随情報とともに記憶することでこの機能を実現している。

本研究ではロボットシステムにおける現在の状況の内的表現としてシンボリックな状態記述を用いる。各シンボルやシンボルとその関係性によって表される状態と、センサや認識器などのプロセスから得られる時系列データを互いに関連付けることによってエピソード記憶の蓄積をロボットシステムにおいて実現する。

図 4.1 にエピソード記憶蓄積によって関連付けられるデータの概略を示す。

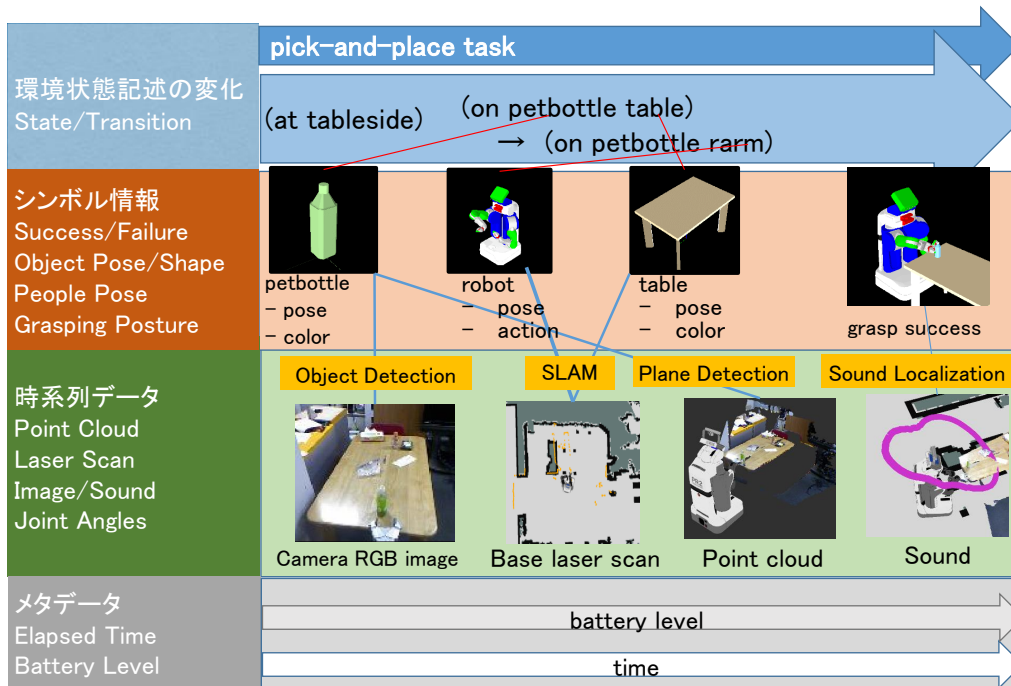


図 4.1: 時系列データと環境状態記述の関連付け ([70] より一部改変)

4.4 蓄積するデータの選択

この節では、ロボットがタスクを行う間にロボットシステム内で扱われるデータのうち蓄積するデータを選択する機能の構成について述べる。

ロボットシステムにおいて記憶すべき情報はセンサから得られるデータやセンサデータを元に得られる認識結果、意思決定を行う根拠となるシンボリックな環境状態記述など多くの種類がある。これらのデータをすべて蓄積することは、ロボットがタスクを遂行する上で必要となる計算資源やデータ記憶資源を圧迫するだけでなく、蓄積した情報をあとから抽出する上で探索コストが非常に大きくなるため、非現実的である。人間は知覚した情報をすべて記憶する代わりに、感覚記憶、短期記憶を経て反復を行ううちに長期記憶として定着する過程の中で、入力した情報の中で蓄積されるべき情報を取捨選択することで、このような記憶の爆発を防いでいる。以降、同様の記憶の取捨選択機能をロボットシステムにおいて実現する方法をロボットアーキテクチャを意識しながら議論していく。

ROS に代表されるロボットミドルウェアでは、ロボットシステムを構成するセンサドライバ・認識・計画・動作などのプロセスは各々がモジュール化され、各プロセス間の情報伝達をネットワークによる通信において行うことで全体のシステムを構成していく。本研究では、ロボットシステムを構成要素として独立した状態を持つ最小機能単位をノードと呼ぶ。ハードウェアであるセンサから信号を受け取り、決められたデータ型として他のノードへデータを送信する機能はセンサノード、センサデータを受け取り、認識処理を行い認識結果を出力する機

能は認識ノードである．その他には速度司令などを受け取り，ロボットに内蔵されたアクチュエータを制御する動作ノードなど多数のノードがある．ロボットシステムはこれらのノードを組み合わせることによって構成される．

各ノードは上位のノードから送信されるデータを受け取り，順次処理して下位のノードに対してデータを送信する．送受信を行うノード間でやり取りされるデータ形式さえ同じであれば，ノードの中でアルゴリズムを変更したり，システムにおいてあるノードを別のノードに変更をしても，それ以外の部分を変える必要がなく，再利用性が高いシステムとなっている．また各ノードは非同期・並列に動作するため，複数の演算装置や複数の計算機からなるロボットシステムにおいても容易に並列化による高速化の効果を得られる．このようなアーキテクチャはパブリッシュサブスクライブ (Publish-Subscribe) 型アーキテクチャと呼ばれる．ここでは各ノードは必要としているデータがどのノードによって出力されるのか，いつ出力されるのか，処理したデータをどのノードに対して送信する必要があるのかについての情報を持たず，トピックと呼ばれる，全ノードによって共通の仮想的なチャンネルに対して，データを送受信する．あるノードからチャンネルへデータが送信されるとそのチャンネルを購読 (Subscribe) するノードのコールバック関数が呼ばれ，処理が開始される．コールバック関数内で処理されたデータがチャンネルに送信されると，それが新たに別のノードのコールバック関数を呼ぶ．以上を繰り返す中で，意思決定ノードが判断に必要なデータを受信し，必要な動作ノードへ目標指令を送信することでタスクが実行されていく．ここでは各ノードが他のノードの状態管理を不要となる一方で，ロボットがタスクを行う上で必要とされていないデータが処理されてしまう問題がある．例えば，テーブルにあるコップをキッチンの棚へ運搬するタスクを考える．このタスクを行うためには，カメラ画像を一定の周期で出力するカメラセンサノード，カメラ画像のうちコップの位置を検出するコップ認識ノード，ロボット自身を含む環境状態を管理し，ロボットの取る行動の意思決定を行う意思決定ノード，ロボットのアクチュエータを動かす動作ノードが必要となる．タスクを実行する動作列としては以下ようになる．

1. テーブルへ移動する
2. テーブルの上のコップの位置を認識し，把持する
3. キッチンへ移動する
4. コップを置く

ここでコップ認識ノードの出力である環境におけるコップの位置情報は上の動作列のうち，コップの把持を行う動作においてのみ必要とされる情報である．一方でカメラセンサノードは画像を一定の周期で出力するため，この動作以外の動作を実行しているときにはコップ認識結果が下位のノードによって利用されていない，すなわちロボットシステムにとって不要であるにもかかわらず，認識処理が行われている．計算資源が十分確保することができれば良いが，日常生活支援ロボットのようにモバイルマニピュレーションを行うロボットは，ロボットが体内に持つことのできる計算資源や記憶容量，バッテリー容量などに制約があるため，不要な認識処理によってロボットシステム全体のパフォーマンスが低下する可能性がある．

一方で，前述の例の通り，モデルベースなロボットシステムでは環境の状態モデルを内部に

持ち、システム全体の意思決定を行うノードが存在する。これを意思決定ノードと呼ぶ。意思決定ノードはセンサから得られる認識情報をもとに環境状態を更新し、次にロボットが行うべき動作を決定し、動作計画ノードへ伝達する。意思決定ノードを中心にこのシステム構成を整理すると、センサドライバや認識、動作計画などのノードに対して、意思決定ノードが現在の状況に応じてオンデマンドに他ノードに対してデータを要求するシステムと考えることができる。このとき、各状態において、意思決定ノードによって直接的・間接的にデータを要求されていないノードでは出力データが直接意思決定や動作に使われることがないことがわかる(図 4.3a)。従来では入力データの受信をトリガとして処理を開始していたところを、本研究では出力が下位のノードによって要求された時にのみ上位のノードに対して出力を要求する機能を導入する。

各ノードのデータ送受信の流れを図 4.2 に示す。従来型のノードではデータの受信を行うノードの購読開始・終了にかかわらず、受信データを処理し、送信を行っている(図 4.2a)。一方で、遅延購読型のノードではノードが送信するチャンネルを購読するノードが出現した時点で、自身が必要とするチャンネルの購読を開始する(図 4.2b)。上位のノードにおいても同様に購読の時期が遅延されることで、ロボットシステム全体に必要なデータのみが必要なときに処理されるシステムとなる(図 4.3b)。

ロボットのタスク実行中に蓄積が必要となるデータの選択をするためには、意思決定ノードが購読を行っているすべてのノードを再帰的にたどり得られるノードの中から、動作の各時点において必要なデータを取捨選択する必要があった。ここで、本研究では遅延購読型ノードを導入することで、ロボットシステムにおける通信のうち必要なデータのみが逐次ノード間で送受信されるシステムとなっているため、動作の各時点において、意思決定ノードと各ノードの出版・購読が動的に行われるため、この関係を追跡することで蓄積情報の取捨選択を実現することができる。

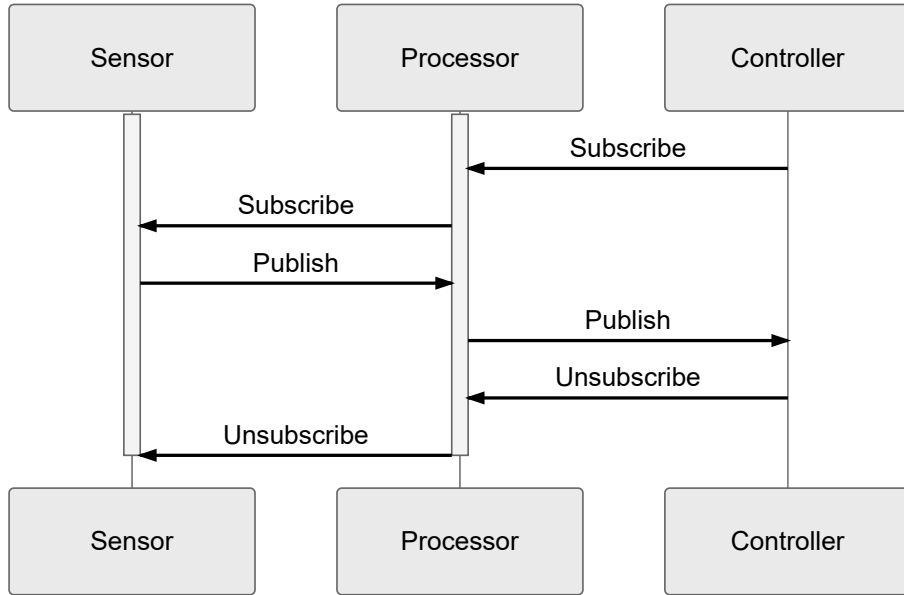
4.5 蓄積するデータの保存

本節では、経験を構成する蓄積データの保存を行うための具体的な実装について説明する。

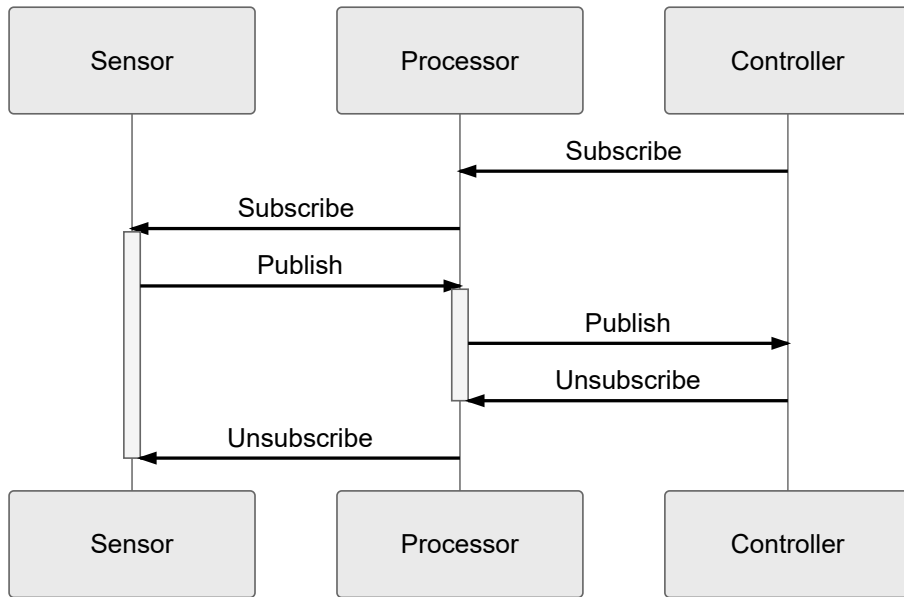
4.5.1 多重連想配列データ構造を用いたロボットエピソード記憶の保存

ロボットはタスク実行時にデータをその場で保存する必要があるが、データの種類はロボットの関節角度列や温度などの単純な数値列であるもの、ロボットの 2 次元平面上における位置や姿勢といったユークリッド空間における幾何学的意味を持つもの、画像などのバイナリ値をもつもの、タスクや動作対象物の関係性と言ったシンボリックな高次表現など多岐にわたる。また複数のロボットが協調して作業する場合などに備えてデータ構造の拡張可能性を考慮しなければならない。

データベースとして最も広く使われているものに関係データベースがある。関係データベースではデータは表(テーブル)と呼ばれる構造によって定義される。表はそれぞれのデータ値



(a) 従来の購読型ノードのデータ処理フロー



(b) 遅延購読型ノードのデータ処理フロー

図 4.2: 分散プロセス型ロボットシステムにおけるデータ送受信フロー

を表す組 (タプル) と属性 (アトリビュート, データのそれぞれの値が何を示すか) からなる。関係データベースにおいてロボットの姿勢を表す属性は表 4.1 のようになる。

関係データベースではこれらの表同士の関係を記述することで, SQL などに代表される問い合わせ言語を用いて論理演算を行い求める結果をデータベースから抽出する。

関係データベースではこのようにデータを挿入する前にデータ構造を定義する必要があり, 様々な種類のセンサ値を扱うロボットのデータベースとしてはなじまない。全てのデータを何らかの規則に従って文字列に変換 (シリアライズ) し, データベースへ保存する方法もある

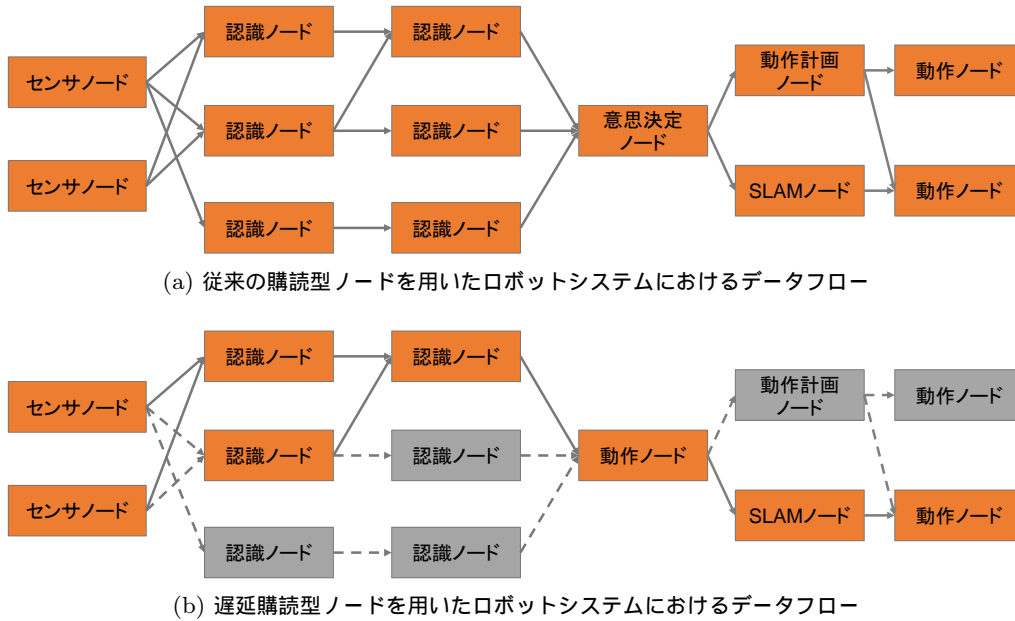


図 4.3: ロボットシステムにおけるデータフロー．橙色のノードは実際に処理を行うノード．遅延購読型システムでは動作ノードによって認識ノードによる処理結果の一部が使用されていないときはサブスクライブ処理を停止することによってそのノードと親ノードのうち必要がないノードでは処理が停止する．停止したノードは灰色，切断された通信は破線で表される．

表 4.1: 関係データベースにおけるロボット姿勢のデータ属性 ([71] より改変)

Column	Type	Modifiers
id	integer	not null default
header_stamp	bigint	
header_frame_id	text	
child_frame_id	text	
transform_translation_x	real	
transform_translation_y	real	
transform_translation_z	real	
transform_rotation_x	real	
transform_rotation_y	real	
transform_rotation_z	real	
transform_rotation_w	real	

図 4.1: ドキュメント指向データベースにおけるロボットの姿勢データ例

```

{
  _id: "554a56878e8b843665000c6b",
  header: {
    frame_id : "eng2/7f",
    stamp: 1540478457515947103,
  },
  child_frame_id: "base_link",
  transform: {
    translation: { x: 0.0, y: 1.0, z: 0.5 },
    rotation: { x: 0.0, y: 0.0, z: 0.0, w: 1.0 }
  },
  _meta: {
    state: [{predicate: "AT", operands: ["FRIDGE"]}],
    action: {predicate: "OPEN-DOOR", operands: ["RARM", "FRIDGE"]},
    task: { _id: "5be4922d80b94dafa104dcf6" },
    type: "geometry_msgs/TransformStamped",
    robot: { name: "pr1012", type: "pr2" },
  }
}

```

[46] が、これは実際にデータベースからデータを取り出すまでデータの内容を知ることができず、データベースが用意している検索のためのインターフェースを利用出来なくなってしまうためコストが大きい。

本研究ではロボットの蓄積データ保存機構としてドキュメント指向データベースである MongoDB[72] を用いた。MongoDB はデータ構造としてスキーマを必要としないという意味で SQL と異なる、いわゆる NoSQL データベースの 1 種である。MongoDB では 1 つのデータはドキュメントと呼ばれる連想配列のリストによって表される。連想配列はキーを文字列、値を文字列、数値、入れ子となるドキュメント、バイナリ列などを扱うことが出来る。

各ドキュメントは `_id` というデータベース内で固有の識別子を持つ。この識別子によってデータは一意に特定される。またデータ構造の事前定義を必要としないため、例えば関節角度列などのようにロボットによって数が異なるデータや種類のことなるセンサのデータなどもそのままデータベースに挿入することが出来る。したがって、シリアライズが不要であり、データベースシステムにおいてデータの要素を条件に含む検索を行う事が出来る。

stamp は各データの表す事象が起きた時点の時刻を表す．この情報は認識データと認識に用いられたデータの検索や比較に用いられる．例えば顔認識の結果データにおける stamp に対応する画像データを取得するときは同じ stamp の値を持つカメラ画像を検索することで行われる．

_meta はデータに付随する情報を格納するために用いられる．_meta.state にはシンボリックな状態記述言語によって表現される環境状態表現が格納される．図 4.1 ではデータとして環境の構造物上の座標系 eng2/7f から見たロボットの基準リンク base_link の相対位置・姿勢を表すが，_meta.state によって，ロボットが冷蔵庫の前にいる時の環境から見た立ち位置，という意味的な情報が付加されている．またデータの種別を識別する _meta.type や経験したロボットを識別する _meta.robot などの付随情報も保存時に付与される．

4.5.2 蓄積データの重要性に応じた保存機能

実際のデータ保存はデータベースエンジンのドライバを備え，ロボットシステムにおいて環境の状態表現と関連付けられたデータを受信すると同時に逐次データベースエンジンへ保存するインターフェースの役割を持つノードによって行われる．このノードをデータ保存ノードと呼ぶ．このノードは内部にバッファを備えるが，このとき単位時間あたりのデータベースへの挿入データ数よりも受信データ数が大きいと，バッファの容量を超えた場合にデータが欠損してしまう．特に，保存したデータをあとから抽出する際のキーとなる環境状態表現や座標変換の情報が失われた場合は，失われたデータ自身だけでなく，そのデータと関連付けられるすべてのデータを取り出すことができなくなってしまう．そこで本研究ではデータの保存方法として 2 種類の方法によってインターフェースを構成している．

一つは保存するデータをノンブロッキングにデータ保存ノードに送信する方法である．送信するノードはデータの保存が行われたかを待つことなく次の処理に進むため，データが確実に保存されたことの確認を行わない無手順方式で，データ保存ノードへのデータ転送が行われる．データ保存ノードはデータを受信すると受信キューにデータを一時保存する．次に受信キューからデータを取り出し逐次データベースへの挿入を行う．受信キューの上限を超えてデータが一時保存されているときは受信データはデータベースへ挿入されず破棄される．無手順方式のデータ転送ではデータの送信ノードにおいて待ち時間が発生しないため保存データを送信するノードのパフォーマンスを低下することを防ぐことが可能である一方で，保存データが確実に保存される保証がない．

もう一つは保存するデータをデータ保存ノードに送信し，データ保存ノードにおける保存が終了するまで処理を待つ方法である．データベースへの挿入はノンブロッキングな挿入に割り込む形で高速に行われる．この方法ではデータの保存が保証される一方で，大量のデータを高速に保存する場合，パフォーマンスが低下する．

本研究では，環境の状態表現や座標変換情報など検索クエリを構成するデータに関しては後者の方法，それ以外のデータは前者の方法を用いてデータ保存を行うことで信頼性を向上している．

4.5.3 データの性質に応じた圧縮保存機能

ロボットが家事支援行動を行うときに扱うデータは多岐にわたる．実際に Willow Garage 社製の日常生活支援ロボットである PR2 によってエレベータのボタンを押す動作を行う際のデータの種類と特徴を 4.2 に示す．

表 4.2: ロボットシステムにおいて扱われるデータの例 ([71] より引用)

データ種別	単位データサイズ (kB)	単位時間当たりのデータ数 (/sec)
base laser scan	4.227	19.6
kinect rgb sensor	921.677	30.0
kinect point cloud	9830.536	8.7
tf frame	5.235	206.5
joint states	2.289	99.3

このようにデータの種類やその利用の仕方によってデータの取得周期や 1 データあたりのサイズが異なる．本小節ではデータの性質に応じて保存するデータを圧縮し，記憶リソースや蓄積データの挿入に際してかかる計算負荷を低減する方法について述べる．

バイナリデータ

カメラ画像など，データあたりの情報のほとんどがバイナリであるデータがある．バイナリ情報は可読性がなく，データの検索キーとして利用することが難しい．したがって，バイナリ情報のみを圧縮することでデータを削減することは有効である．バイナリの汎用的な圧縮法としては Deflate 圧縮法を利用した gzip[73] や LZX アルゴリズムを利用した CAB 形式など数多く存在する．これらの手法は可逆圧縮として知られ，逆符号化を行うことで元のデータを完全に復元できる．また画像の JPEG 圧縮などデータの性質や用途に応じて，影響が出ない範囲でデータを劣化することで通常の可逆圧縮よりも高圧縮率を実現する非可逆圧縮手法が提案されている．

本研究において使用されるカラーカメラ画像を実際に JPEG 画像に圧縮することにより圧縮前は 921.677kB であったファイルサイズは 4.8397kB へとデータサイズが削減された．

時系列数値データ

ロボットシステムでは様々な時系列データがやり取りされている．データによって必要とされる通信周期は異なるが，高周期で通信することが要求されるデータとして，ロボットの関節角度列や座標系変換行列情報が挙げられる (表 4.2)．高周期で通信されるデータは 1 データあたりのデータサイズは比較的小さいが，単位時間あたりの通信データサイズは大きく，またデータベースへの挿入に関する計算負荷が無視できなくなってくる．このような場合には，データ中における値の変化を監視し，差分が生じた時にのみデータを保存する手法が有効とな

る．データを取り出すときには要求する時刻以前の最も要求時刻に近いデータを使う．

4.6 蓄積経験の抽出

本節ではロボットシステムにおいて扱われる時系列データに対してシンボリックな状態記述言語によって表現される環境状態モデルとの関連付けによって意味的な出来事として記憶していく仕組みによって蓄積されたデータを各ノードが学習に使うために抽出する方法について説明する．

本研究における行動計画システムではロボットが行う各動作は階層的に制御される．最上位がタスク (例：冷蔵庫から缶を取ってくるなど行動目的を表す)，タスクを構成する動作列となるアクション (例：缶を把持する，冷蔵庫のドアを開けるなどの状態遷移を含む最小動作)，アクションを構成する指令単位を表すモーション (例：手先を缶に向けて近づける目標関節角度指令など) である．エピソード記憶の抽出はこれらの階層的記述を用いた演算式を組み合わせで行う．演算式は MongoDB の検索クエリに変換され，データベースエンジンによって評価される．各データのキーも検索クエリとして使用可能である．また MongoDB の機能としてデータのソートやデータ数のカウント，不要なキー値の省略，データ数の上限設定を行うことができる．

4.7 エピソード記憶に基づく長期経験の蓄積機構を用いたロボットシステム評価実験

4.7.1 時系列数値データの差分値保存

ロボットの関節角度列を表すデータについて，データ間の差分のみの保存を行う機能の有無に応じた保存データサイズの時間変化を図 4.4 に示す．差分のみの保存を行わない場合は保存データサイズが一定の割合で増加し，タスク実行終了時には 45.45MB のデータサイズとなった．一方で差分のみの保存を行った場合は保存データサイズはロボットが関節角度指示を行った時間のみ保存が行われ，タスク実行終了時には 7.336MB のデータサイズとなった．

4.7.2 環境状態記述を用いた蓄積経験の抽出

例えば「冷蔵庫のドアを開け終わった時の手先位置」を蓄積した経験として抽出するクエリは

```
'(:_meta.stored_type TransformStamped
  :header.stamp ?stamp
  :header.frame_id "base_link"
  :child_frame_id "r_gripper_frame")
```

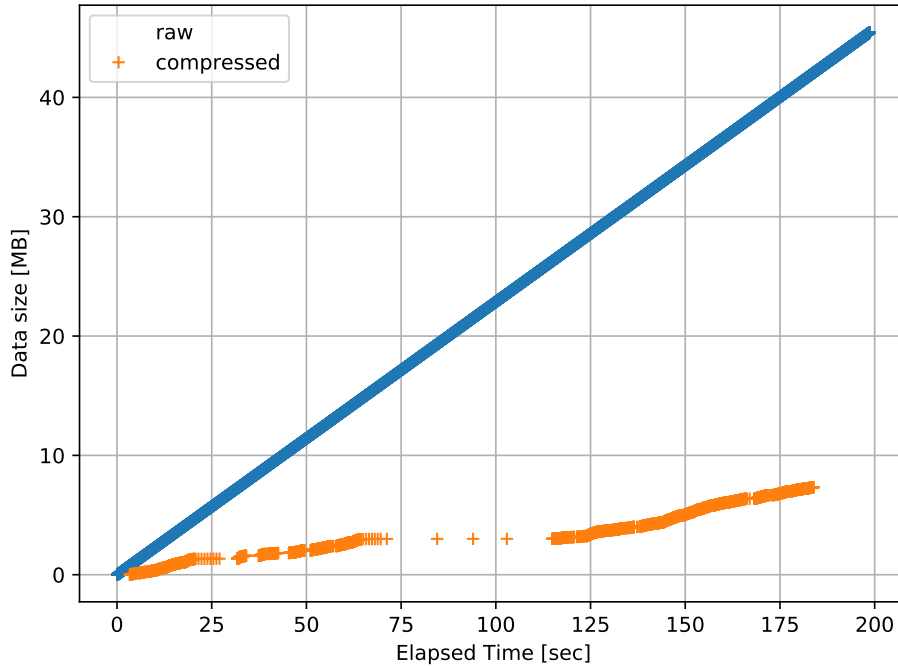


図 4.4: 冷蔵庫から飲み物を取るタスク実行中におけるロボットの関節角度列の保存データサイズ

```
forall ?stamp in
(bind :_meta.end_at ?stamp
'(:_meta.stored_type Action
:name (open-door fridge rarm))
```

と表現される。

```
'(:_meta.stored_type TransformStamped
:header.stamp ?stamp
:child_frame_id "r_gripper_frame")
```

は抽出する対象のデータに関する指定である。データの種別として座標系を表す `TransformStamped` を指定している。親リンクはロボットの台車リンクを表す `base_link` であり、子リンクとして右手先リンクである `r_gripper_frame` を指定している。また取得するデータが蓄積された時刻として `?stamp` により制約が与えられている。`?stamp` の具体的な値は以下で定義される。

```
(bind :_meta.end_at ?stamp
'(:_meta.stored_type Action
:name (open-door fridge rarm))
```

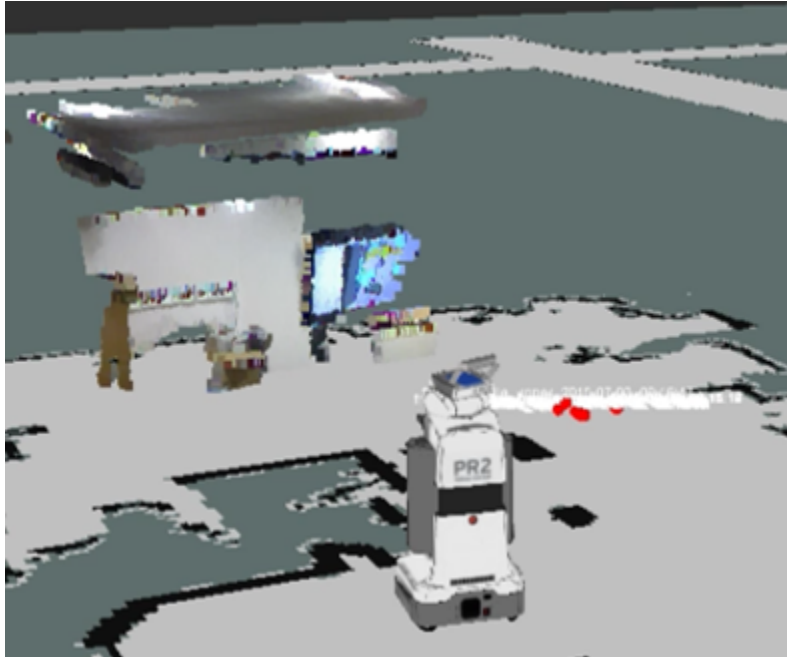



図 4.5: 冷蔵庫のドアを開ける動作後の右手先位置をエピソード記憶蓄積機構から抽出例 ([71]より引用)

ここではロボットがタスクを構成する動作の開始時，終了時にそれぞれ出版される行動実現動作の遷移を表すデータ種別である Action を指定している．動作の具体的な内容として冷蔵庫の扉を右手で開ける動作を表す (open-door fridge rarm) が指定されている．この条件を満たす各データについて，動作の終了時刻を表すフィールド `_meta.end_at` を `?stamp` に割り当てる bind 関数を用いて，動作の終了時刻を `?stamp` に束縛している．

座標変換はある 2 つのリンク間における変換をエッジ，リンクをノードとした木構造として表されるため，ロボットの台車リンク `base_link` から右手先リンクへの座標変換が直接データとして保存されていない場合は，2 つのリンク間を結ぶエッジを木構造を用いてたどり，変換を適用していく．木構造は図 3.11 を用いた．すべてのエッジは非同期に得られた座標変換結果を表しているため，木構造を利用して得られた座標変換の有効な時刻は，計算に用いられた全エッジのうち最も古い時刻のものとなり，要求される時刻における座標変換と誤差が生じる原因となる．ここでは座標変換として要求された時刻から 10 秒以上古いエッジを削除し，求める 2 リンク間の座標変換が計算不能であるときは無効なデータとした．得られたデータを可視化した図を図 4.5 に示す．

4.8 おわりに

本章では日常生活環境におけるタスク計画を実現するための知識を獲得するための，長期経験蓄積機能の構成法について述べた．ロボットシステムにおいて経験蓄積機能を構成するために記憶の認知心理学的分類におけるエピソード記憶に着目し，センサや認識器などから出力さ

れる時系列データと、行動計画器によって扱われる環境状態記述を関連付けてデータを保存していくことで、あとから特定の出来事に関するデータを抽出することができるシステムを構成した。実世界ロボットシステムにおいて長期的に経験を蓄積する際に発生しうる発生する問題としてタスク実行中のロボットの演算・記憶性能の低下を取り上げ、解決する手法として分散プロセス型ロボットシステムにおけるプロセス間通信経路の動的結合性を利用して蓄積するデータの自律的な選択と保存データの圧縮のための方法を提案した。蓄積した経験を出来事やロボットシステムにおける環境モデルの状態記述を用いて抽出する方法についても述べた。実ロボットにおいて提案するシステムを実装し、日常生活環境においてデータを蓄積することで、提案システムの有効性を示した。

第 5 章

長期経験を用いた環境の局所合理性に基づく知識の自律獲得機構

5.1 はじめに

本章では日常生活環境においてロボットがタスクをこなす、あるいは待機中に観察行動を行うことを通じてエピソード記憶として蓄積してきた経験データベースから、日常生活におけるタスクの行動計画に活用できる知識として抽出する機能について述べる。第 2 章において議論された、長期的な経験に基づく知識を獲得するための具体的な実装に関する説明となる。

5.2 関連研究

5.2.1 経験学習機能を備えるロボットアーキテクチャ研究

STRANDS プロジェクト [74] では、日常生活環境が壁や天井などの構造物などは静的であるが、照明やドアなど動的なものも混在する環境であることに着目し、複数の研究機関において長期的にパトロールタスクをロボットが行うことにより、このような準静的な環境における周期的な変化を地図に埋め込むことで、時間的な環境の変化を考慮した移動経路計画を行う仕組みを提案している。Kirsch ら [75] は RPL を拡張して、タスクを構成する動作ごとに学習するセンサデータ (Experience)、学習手法 (Learning algorithm)、学習対象 (Use of result) を定義することで、ロボットが動作を行っている最中に対応するセンサデータを蓄積し、学習対象となるパラメタを改善していくアーキテクチャを提案した。ここでは動作ごとに学習が独立して行われていたため、環境状態や行動意思の決定などのより上位の内的モデルについては考慮されていない。桑山らはヒューマノイドロボットにおける複雑な立ち上がり動作のパラメタを学習する手法として、制御モデルを立てることが不要な強化学習やニューラルネットワークがあるが、学習によって得られたモデルは可読性がなく、一度形成したモデルを修正することが困難である点を指摘している [76]。Winkler らは CRAM アーキテクチャにおける行動計画言語である CRAM Plan Language(CPL) を用いたロボットシステムにおいて、木構造を持つ

タスク実行マシンの各ノードにあたる動作に対して、動作成功時のパラメタセットを表す期待モデル (Expectation Model) を定義し、経験蓄積によってパラメタセットの取りうる範囲を推定した [77]。学習済み期待モデルと実際の動作結果を比較することで、動作の失敗判定や因果関係を推論可能にしている。期待モデルを元に因果関係の推論を行い、次に行う行動を経験を元に判断していく適応的なシステムとなっている。ここでは大規模データ解析を行うデータマイニングの手法である決定木学習が経験学習の手法として用いられた。決定木学習 [78, 79] は動作パラメタやセンサ値などの数値ベクトルがあるときに、動作の成功や失敗などを表す離散的な値をパラメタの従属変数とみなして、従属変数を説明する木構造を求める手法である。ニューラルネットワークなどを用いた分類などと比較して、木構造を見ることによって分類判断の根拠が明示的にわかるなど可読性に優れ、判断の修正を用意に行うことができる。

人間の意思決定や行動に注目して長期的観察によって得られる経験的知識は嗜好と呼ばれる。Abdo らは物品の棚への片付けタスクにおいて、クラウドソーシングサービスを用いて大規模な人間の物品収納嗜好知識を収集し、スペクトラルクラスタリング [80] などの統計処理を行ったモデルを利用して物体の適切な収納場所を推定した。ここで、推薦エンジンを構成するアルゴリズムとして利用される協調フィルタリング [81] を導入することで、未知物体に対しても学習モデルを最も支持するクラスタへ収納するモデルを提案した [82]。

前述の手法はロボットがすでに環境や行動モデルをシステムとして備えている前提で、行動の意思決定や動作のパラメタを経験によって環境に馴染むように改善していくものであった。日常生活におけるタスクのように人間の行ってきたタスクをロボットが代替する場合は人間を観察することにより、環境や行動モデルを形成可能であるという前提に立った研究もなされてきている [83, 84, 33]。この分野は人間の行っている行動がロボットにとっての学習対象、すなわちデモンストレーションであるとして Learning from Demonstration (LfD) と呼ばれる。

國吉らは人間を観察することによって物体の操作に関する知識を学習するタスクを Learning by watching とよび、ブロック積み上げタスクを題材としてタスクを実現するためのシステムに必要な構成要素を提案している [85]。

Seeing 初期状態と環境モデルの構築，手先の観察，操作対象の探索，操作による状態変化検出
Understanding 操作シーケンスの分節化，操作の分類，依存関係の分析，計画の記述
Doing 初期状態の認識，動作計画，フィードバック制御，失敗検出・復帰

ここでは環境モデルとしてすべてが可観測であることが前提となっていた。日常生活環境におけるタスクではロボットのセンサの物理的制約やオクルージョンにより、すべての環境状態を常に認識することは難しい。したがって、ロボットは部分的な観測に基づき、自身の持つ環境モデルに照らして現在の状態を推定しなければならない。例としてロボットの環境中における自己位置推定が挙げられる。第 3.3.3 節において述べたように、SLAM に用いられる地図は 2 次元あるいは 3 次元空間上を等間隔に分割した各セルに対して障害物の有無を記録したものが多く、このような環境の幾何的な形状を用いた地図は障害物回避などの移動に関するタスクに有用とされる一方で、日常生活環境のように日々動的に変化する環境では、自律移動や自己位置推定の精度が環境の動的性質に大きく依存してしまう問題がある。特にドアハンドルの

把持など高い精度が要求される操作においてロボットの自己位置推定による誤差の影響は大きい。Stulp らはこのような動的な環境において生じるモデルと実世界の乖離によって生じる誤差に対処するシステムとして、物体操作を行う移動ロボットの立ち位置に関する概念である Action-Related Places (ARPlace) を提案した [86]。ARPlace は、従来では環境モデル中に固定的に与えられていた操作を行うための立ち位置を、操作物体と操作者であるロボット自身の位置関係、及び各立ち位置における操作成功体験をパラメタとした多変量ガウス分布として表したものである。このシステムではロボットの立ち位置は物理的には操作対象とロボット自身の関係性によって決まり、自己位置推定による誤差の影響を受けない利点がある一方で、ロボットの操作成功体験に基づき誤差を経験的に吸収するモデルとなっているため、モデル獲得のためにロボット自身による試行錯誤を行う必要があり、安全上懸念がある。一方でロボットが操作対象物体が見えるような位置へ近づき、ロボットがセンサ座標系における認識のように自己位置推定による誤差が含まれないような座標系で位置合わせを行うことでこの問題は解決することができる。

ロボットの物体操作の分類を行うためには日常生活環境において扱われる物体について考える必要がある。食事の準備タスクを例にすると、食器棚や引き出しなどのハンドルとドアによる単純な開閉動作を伴うもの、ボタン操作によって扉が開閉する冷蔵庫や食器洗い機のように操作と物理的な効果が直接対応しないもの、電子レンジやエレベータのように対象が内的状態を複数持ち、ボタンなどのインターフェースを操作することによって状態を遷移させることが必要となるものもある。このような情報を扱う概念としてセマンティックマップが挙げられる。セマンティックマップでは環境上のものや場所の関係性を表す地図として一般化され、従来の環境上の幾何的な情報は地図の持つプロパティの一つとして扱われる。Pitzer らは、環境上における概念をシンボルとしてシンボル間の関係性を一階述語論理に従って記述した階層的な有向グラフ構造と各シンボルの具体的な位置や姿勢を関連付けた地図である Semantic Objects Maps を用いることで、環境中における冷蔵庫の位置やドアハンドルなどの部分形状、可動ジョイントの構造といった環境の意味的な分割を管理する仕組みを提案した [87]。「冷蔵庫は家電である」、「冷蔵庫にはドアがある」といった環境上における概念や関係性は、そのほとんどが事前にロボットシステムに知識として与えることができるが、各家庭における冷蔵庫のドアハンドル形状と言った環境に固有な情報はロボットがあとから自律的に獲得しなければならない情報である。小島らは操作可能な環境構造物について、幾何的な拘束がモデルとして与えられたときに、三次元エッジ抽出を利用した操作による対象の移動量認識によって可動構造のパラメタ推定を行った [88]。また垣内らは環境の三次元点群から抽出された、対象の各平面の操作による移動量の変化と、操作者の手先位置から得られるドアハンドル位置を人間の操作を観察することによって、パラメタに加えて三次元点群中のハンドル位置やモデルの形状を獲得している [89]。

日常生活環境における可動構造をもつ操作対象が認識されたときに、対象についてどのような操作が必要かについては遠隔操作の研究においてもまとめられている。東らはタブレット端末を用いた家具・家電操作のグラフィカルユーザインタフェースを実現するために、操作を以下のように分類した [90, 91]。

- 引く動作
- 押す動作
- 回転する動作

引く動作は、棚などの直動可動機構を有するドアのハンドルを引くための動作である。押す動作は、引く動作により開いた棚などを元の状態に戻すための動作や、家電のボタン操作に用いられる動作である。回転する動作は冷蔵庫の大きな扉や電子レンジ、食器棚などにあるヒンジによって取り付けられた扉を、ハンドルを用いて回転しながら開閉するときの動作である。

5.3 人とスポットの相互作用に基づく定位置操作モデルの経験的獲得

本節では日常生活環境において人がある特定の場所において行う傾向を観測した経験から、ロボットが日常生活タスクを計画する上で必要な意味的な場所に関する知識を獲得する方法について説明する。日常生活環境におけるタスクをロボット自身の移動を伴った環境操作の連続と捉えるとき、ロボットが移動した先の各地点において行う動作は、タスクを行うために必要なコップやペットボトルなどのアイテムを把持・配置したり、冷蔵庫の扉の開閉やボタン操作などの環境中の構造物に対する操作からなる。このうち環境構造物に関する操作を行うための地点は環境依存の知識として定義され、シンボリックな状態記述言語では各々の場所がシンボルとして与えられる。この知識を *spot* と呼ぶ。*spot* は実世界においてある特定の幾何的な場所と関連付けられるが、意味的には環境における構造物の配置に依存する知識であることから、*spot* を用いて表現される知識や動作を定位置における操作を表すモデルと考えられる。場所に関するシンボル *spot* を用いてロボットがタスクを計画・実行するためには、シンボルが依存する操作対象物体について以下の情報が必要となる。

- 操作物体を他と区別するラベルと幾何形状
- 操作物体の環境における配置
- 操作物体の取りうる状態
- 操作物体の状態を遷移させる動作方法・手順・パラメタ

例としてテーブルに対して物を配置する動作を伴うタスクを考える。ロボットはテーブルという物が識別可能であり、テーブルに対して物を上に置くことができることを知識として持っている必要がある。このとき *spot* としてシンボル *table* を定義し、(*placable table*) のように物を置くことができるというシンボルに対する属性を *placable* という関係式によって記述することで記述可能となる。冷蔵庫などのような物体自体が可動構造を持ち、複数の状態を遷移する物体については、物体が持つ各状態について関係式を定義することでタスク記述が可能となる。例として冷蔵庫 *fridge* の扉が開けられるということに関して (*openable fridge*) という操作可能性を表す関係式が与えられる。この関係式を用いてロボットが環境操作を行い状態を遷移する必要がある場合には、操作物体の状態変化に応じ

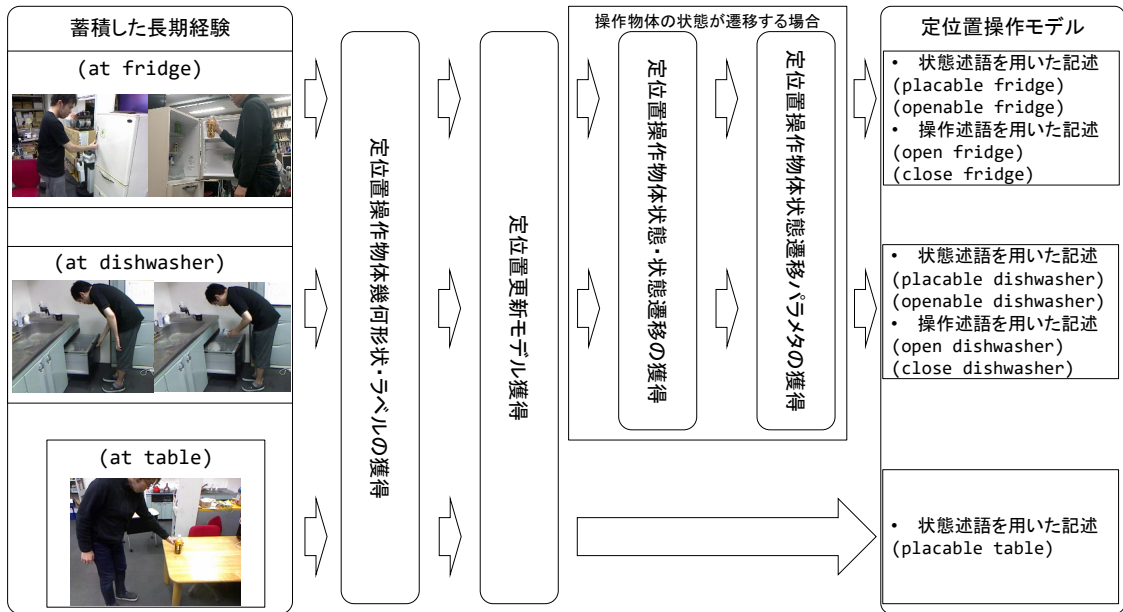


図 5.1: 定位置操作モデルの獲得フロー図

た関係式を与える必要がある．例えば冷蔵庫の扉を開けるためには冷蔵庫の扉が閉じられた状態と区別するための関係式として (open fridge) を与えるとすると，扉を開ける動作は任意の spot を表すシンボル SPOT に対して，(openable SPOT) である条件のもとで (open SPOT) という影響を与える状態遷移と定義される．実際にこの遷移を行うためには状態遷移に対応する実世界でのロボットの動作を決定するための手順やパラメタが必要となる．openable や placeble といった spot 自体の状態を表す述語を状態述語，open のような spot における操作によって変化する操作物体の状態を表す述語を操作述語と呼ぶ．このような spot に関する知識をまとめて定位置操作モデル (In-place manipulation model) と呼ぶ．

ロボットがあとから各環境において自律的に更新する利点として以下が挙げられる．

- 洗濯機や冷蔵庫など，日常生活環境におけるタスクで操作の対象となる物品は，機能的には限定的であるものの，操作方法に個体差があるため，事前にすべての対象物に関する知識を与えることは難しい．したがって，ロボットがタスクを行う各環境で後天的に獲得できると良い．
- 環境の観察によってモデルを逐次更新することで，環境が変化してもタスクを継続的に行うことができるようになる．
- 環境を共有する人間は操作対象に関する知識を十分に持っているため，観察や対話などの手段により比較的知識を獲得しやすい点

本節では第 3.3.8 節において述べたロボットのアイドル状態における振る舞いとして，ロボットが定位置操作モデルを持たない対象について，対話や観察によりロボットエージェントが経験的にモデルを獲得する方法について説明する．

5.3.1 定位置操作物体の状態述語獲得

タスクの複雑さに応じて必要とされる状態述語は異なるが、この述語は環境に依存しないため、事前に定義することが可能である。本研究では spot に対する状態述語として、以下の2つを定義する。

placable その地点に置いて物体を持ったり置いたりすることが可能であることを示す状態述語。

openable その地点に置いて物体を取り出したり、格納する際に扉の開閉などの物体操作が可能であることを示す状態述語。

placable については各 spot における操作物体において接触関係にある物が現れる・なくなるという時系列的な変化をロボットが経験した記憶から抽出することで、この述語を適用する対象の spot シンボルを得ることが可能である。

一方 openable 述語に関しては操作物体自体の状態変化を認識する必要がある。またロボットが実際にこの述語が必要とされるタスクを実行するときはロボットが自律的に操作物体の状態遷移を行う必要がある。以降では操作物体に状態変化が発生する場合に、変化しうる状態と状態を遷移するための操作手順・パラメタを経験から獲得する手法について取り上げる。

5.3.2 定位置操作物体の操作述語獲得

操作述語を用いたタスクをロボットが扱うために必要な知識は以下にまとめられる。

物体の名前 動作指示や知識の管理のために他の操作物と区別をするための情報である。

物体の幾何的な特徴 環境認識において操作すべき物体を識別するために必要となる情報である。カラーカメラや深度センサを用いて識別するためのカラー画像や三次元点群などの外見的特徴などが挙げられる。

物体の幾何的配置 物体が環境中において配置されている位置や姿勢といった情報である。ロボットが環境操作を行うために移動を行うときに用いられる。

物体の状態 物体について操作を行うことにより遷移される状態に関する情報である。冷蔵庫のドアを開ける場合では、冷蔵庫のドアが開いていない状態、ドアが開いている状態などが挙げられる。目標状態に至るまで複数の状態を経由する必要があるときはその依存関係についても含まれる。

物体の操作 物体の状態を遷移させるためにロボットがとる動作に関する情報である。冷蔵庫のドアを開ける例ではドアハンドルの位置、ドアハンドルを操作する際の軌道・関節角度列などが挙げられる。

操作モデルの状態遷移に関する知識獲得のフロー図を図 5.2 に示す。

以降これらの情報を得る方法について述べていく。

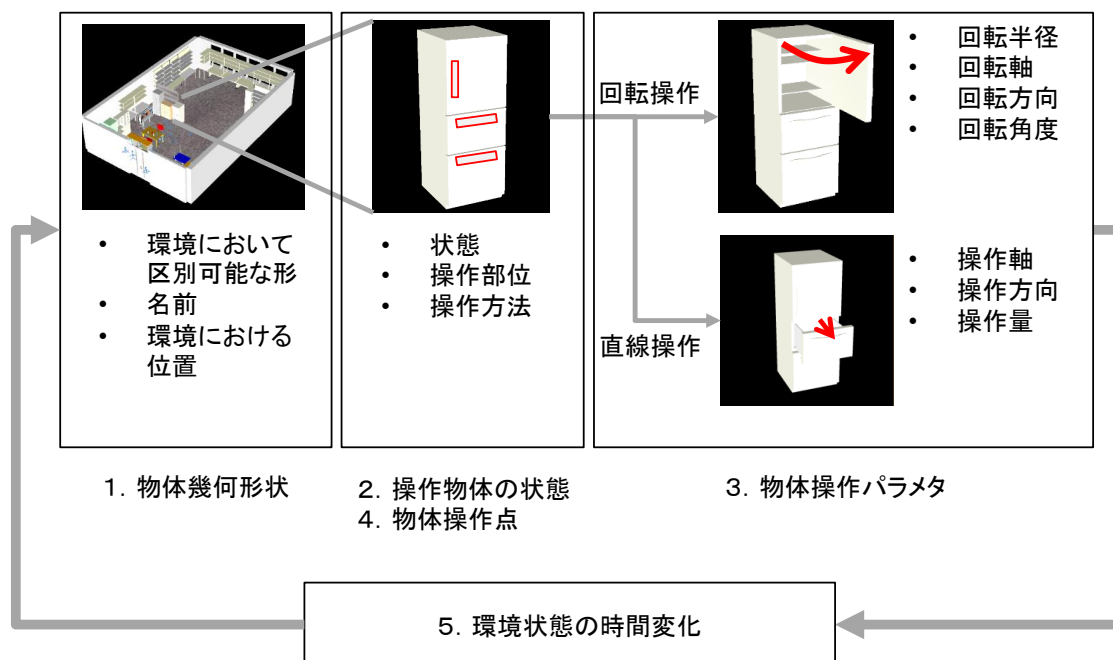


図 5.2: 定位置操作モデルの操作述語知識の獲得フロー図

操作モデルの幾何形状獲得

センサによって知覚した環境の幾何特徴から、操作物体を認識することはタスク実行中に操作対象とロボットの位置関係を把握したり、人間の物体に対する操作を観察する際に必要となる。

Achanta らは k-means による教師なしクラスタリング手法をカラー画像配列のピクセル値に対して適用することで、画像領域全体を色情報に基づいたピクセルの集合 (Superpixel) に分割する手法を提案した [92]。

このような画像中のピクセルごとに物体のクラスを当てはめる写像は非線形関数となるが、多層的なニューラルネットワークに大容量のデータセットを教師データとして与えて学習を行うことで、このタスクを実現する研究があり、このタスクはセマンティックセグメンテーションとして盛んに研究されているほか、競技会が開催され性能改善が行われている。この手法の問題として、出力となる扱える物体の種類が学習時のデータセットにおいて定義されたクラスに制限されてしまうことと、自身でデータセットを作成しようとしたときに教師データを多く集めなければならない点が挙げられる。

本研究では、操作モデルの幾何形状を得る手法として、事前に学習によって与えられたクラスセットによるセマンティックセグメンテーションによって識別可能な物体を用いる方法、それ以外の物体についてはユーザから環境中の物体領域を指さしによって指示する方法により獲得する手法を採用する。

ロボットは何もするべきタスク目標を与えられていない状態において、人間が行う環境への操作を観察する。このとき環境の三次元的な幾何形状は以下の三種類に分類される。

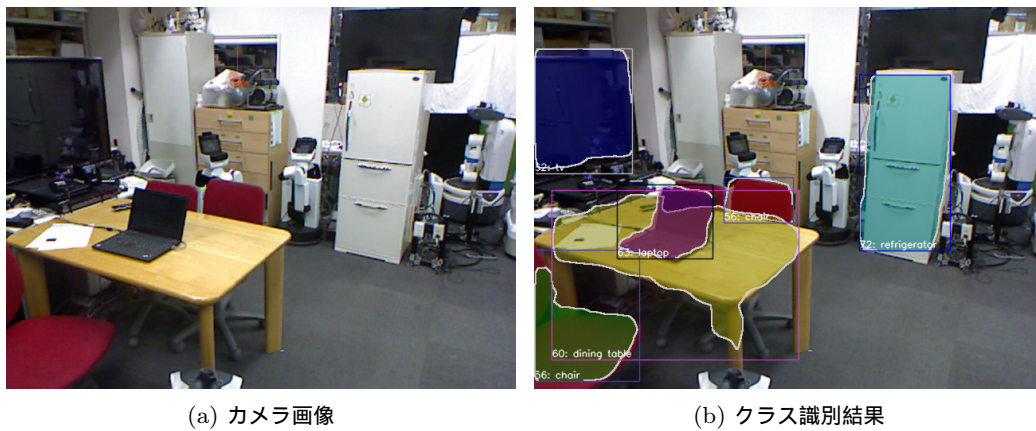


図 5.3: Mask-RCNN モデルを用いたセマンティックセグメンテーション

ラベル付き操作物体 セマンティックセグメンテーションによって識別可能であるが、物体に関する操作モデルは未知である物体

ラベルなし操作物体 セマンティックセグメンテーションによって識別不能であるが、環境において人間が操作を行っている物体．物体の占める領域，および操作モデルはロボットにとって未知である．

非操作物体 入力センサ空間上において上記以外の領域はこれに分類される

ラベル付き物体を得るために、本研究では Mask-RCNN モデルを用いたセマンティックセグメンテーションを行う (図 5.3)．学習データセットとして MS-COCO データセットを用いた．MS-COCO データセットには 80 クラスのセグメントされた物体領域によって構成されているが、環境中の操作可能物体として以下のようなクラスがある．

- tv
- microwave
- oven
- toaster
- refrigerator

一方で、これらのクラスに分類されないラベルなし操作物体については、環境から物体の領域をセグメントする必要がある．本研究ではロボットが何もタスクをしていない状態で環境中に共生する人間を観察する振る舞いをロボットが持ち、観察の中心人間がラベルのない環境の構造物と接触時に音声対話により物体領域とラベルを獲得する仕組みをロボットに備えるシステムとした．

まずは人間が環境中の操作対象物へアプローチしているかを判定するために、環境観察において得られる三次元点群情報のうち人間とそれ以外を区別する必要がある．環境中の人間の検出には Openpose[93] を用いた．Openpose は RGB カラー画像を入力として、画像中の人間

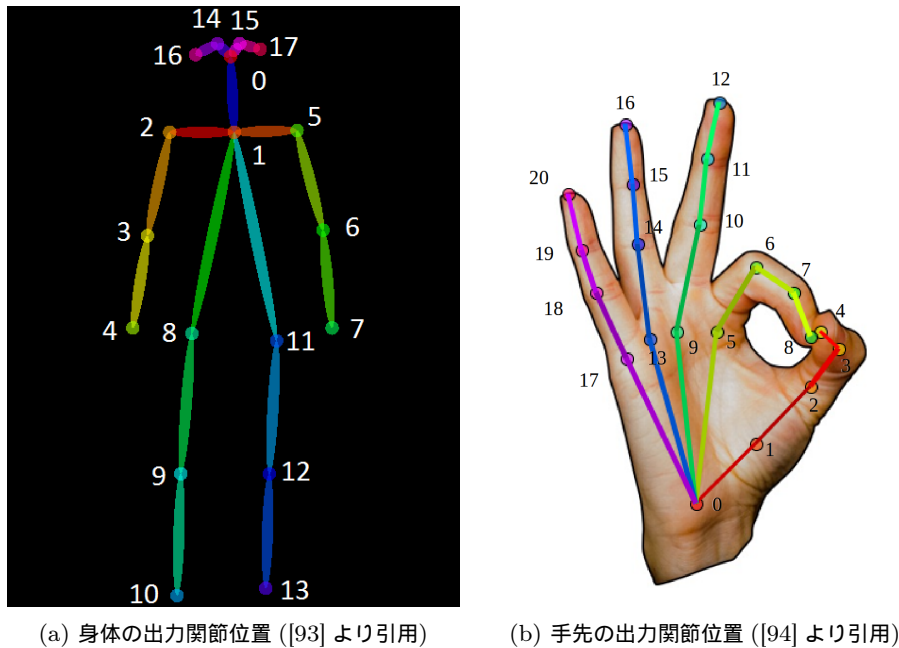


図 5.4: Openpose モデルの出カ関節位置

の関節位置推定を行う深層学習モデルである．このモデルの出カは人間の関節位置に相当する画像上の 2 次元空間における確信度マップと関節間の接続を表すベクトル場マップで，確信度マップから得られる人間の関節位置に対してカラー画像と対応の取れた深度画像上にそれらを投影することによりセンサ相対座標系における 3 次元空間上の関節位置を得る．ベクトル場マップを併用して対応する肘から手首までの三次元ベクトルを算出し，手首より先端の手先領域をもとの RGB カラー画像から切り出した部分画像を得る．ここで肘関節，手首関節はそれぞれ図 5.4a における 3, 6 番, 4, 7 番の位置を用いた．

得られた部分画像を入力として，手先の関節に関する確信度マップを出カとするモデル [94] を使用することで手先関節のカラー画像上の二次元平面上的座標が得られるので，元の画像に対する部分画像の位置を考慮し，深度画像に投影することで手先関節の三次元的な位置を得る．ここで手先関節位置は図 5.4b における 5, 6, 7, 8 番目の平均位置を用いている．

得られた手先関節の三次元位置と環境中の物体が接触するときに操作モデルを獲得するための音声対話が始まる．

音声対話のフローを以下に示す．

1. Excuse me. What is that?
2. Please, point out the left upper corner of the object.
3. Please, point out the right bottom corner of the object.

最初の対話では操作モデルを表す名前を人間から獲得する．次に人に対して操作対象の手前左上隅，手前右下隅を指す位置をそれぞれ指差す操作を誘導することにより得る．3 次元空間上に得られた 2 点を対角線とし，地面に対して垂直な平面上にある長方形を底面とした直方体

Algorithm 1 定位置操作物体の状態推定手法**Input:** *geom* : Geometric model of the manipulable object**Input:** *state* : Initial state**Output:** *states* : States of the manipulated object

```



```

をバウンディングボックスとして、この直方体内に含まれる三次元点群のうち、人間を除いた部分を操作対象の幾何形状として記憶する。

操作モデルの可動状態推定

得られた操作物体の幾何形状を用いて、可動構造を持つ操作モデルの状態推定を行う (アルゴリズム 1)。まず操作物体の三次元点群に対応する RGB カラー画像ピクセルのみを切り出したマスク画像に対し、グリッド上に配置した点について、Optical flow を計算し、人間の操作時における操作物体領域の部分的な移動量を算出する。次に、*estimateFlowThreshold* では人が操作を行っていない状態のオプティカルフローの変化量の最大値にオフセット z_f を加えたものを閾値として設定し、操作中に閾値を上回った時点から、操作が終了し閾値を再び下回った時点の状態遷移とする (アルゴリズム 2)。操作の前後で同じマスク領域に対して SIFT 特徴量を用いた画像マッチングを行い、操作前後の各状態の類似度を比較する (アルゴリズム 3)。類似度が閾値を下回る場合、操作後の状態を操作対象の持つ新しい状態として登録する。

操作種別・パラメタの推定

操作対象の状態遷移グラフの各エッジについて、必要となる操作の種類とパラメタを求める。押す動作と引く動作は線分により近似される。線分をなすパラメタは始点・終点・線分長

Algorithm 2 定位置操作物体の状態遷移認識に用いるオプティカルフローの計算手法

```

prevImg ← null                                ▷ Color image on previous time step
prevMask ← null                                ▷ Mask image on previous time step
function CALCFLOW(img, mask)
  if prevImg is null then
    prevImg ← img
    prevMask ← mask
    return null
  else
    flows ← calcOpticalFlow(img, prevImg)
    factor ← sizeof(mask)/sizeof(img)
    velocities ← []
    for each flow ∈ flows do
      if pointInMask(mask, flow)pointInMask(prevMask, flow) then
        velocity ← getVelocity(flow)
        velocities ← append(velocities, norm(velocity))
      end if
    end for
    nVelocities ← sizeof(velocities)
    velocities ← sort(velocities):[nVelocities/2]
    return getAverage(velocities)
  end if
end function

```

である．回転する動作は円弧により近似される．円弧をなすパラメタは円の中心点，半径，円上の軌道開始点を示す角度，終点を示す角度，円をなす平面に垂直な三次元ベクトルとなる．操作の種類とパラメタ，第 5.3.2 節において得られた操作中の人間の手先軌道から操作種類とパラメタのフィッティングを行う．フィッティングを行うアルゴリズムをアルゴリズム 4 に示す．

操作モデルの各状態を遷移する間の人間の手先軌道について，三次元空間上における線分と円弧にフィッティングを行う．人の手先軌道の抽出は第 5.3.2 節同様 Openpose を用いた．

線分のフィッティングは手先軌道に対して重心を原点とした点列の座標を列とする行列に対して特異値分解を適用することにより，主軸を算出する．重心を通り，主軸ベクトルを方向とする直線に対して手先軌道を投影し，重心から最も遠い二点をそれぞれ直線の始点と終点とすることで，線分の長さを特定する．フィッティングを行った線分に対して手先軌道の各点との距離を算出し，合計したものを線分フィッティングのエラー値とする．ここで，主軸の算出には外れ値に対して頑強な結果が得られる RANSAC アルゴリズムを利用する [95]．

Algorithm 3 定位置操作モデルの静止状態の一致判定法

Input: *matchPointMax* : The maximum number of matches used**Input:** *stateThreshold* : Threshold for matching states**Output:** *states* : States of the manipulable object

```

function ESTIMATESTATE(img, mask)
  keypoints  $\leftarrow$  detectKeypoints(img, mask)
  description  $\leftarrow$  extractPoints(img, keypoints)
  averageDists  $\leftarrow$  []
  for each rImg, rKey, rDesc  $\in$  states do
    matches  $\leftarrow$  matchKeypoints(keypoints, rKey)
    matches  $\leftarrow$  sortByDistance(matches)
    matches  $\leftarrow$  subseq(matches, matchPointMax)
    distances  $\leftarrow$  computeDistance(matches)
    averageDist  $\leftarrow$  average(distances)
    averageDists  $\leftarrow$  append(averageDists, averageDist)
  end for
  minDist  $\leftarrow$  min(averageDists)
  if minDist < stateThreshold then
    stateIndex  $\leftarrow$  argmin(averageDists)
    return stateIndex
  else
    stateIndex  $\leftarrow$  sizeof(states)
    states  $\leftarrow$  append(states, (img, keypoints, description))
    return stateIndex
  end if
end function

```

Algorithm 4 定位置操作種別・パラメタの推定法

Input: *trajectory* : Trajectory during manipulation*line, lineError* \leftarrow *fitToLine*(*trajectory*)*circle, circleError* \leftarrow *fitToCircle*(*trajectory*)**if** *lineError* < *circleError* **then** **return** *line***else** **return** *circle***end if**

円弧のフィッティングについては、まず手先軌道をなす点列について、二乗距離の合計を最小にする平面を特異値分解により求める。求めた平面上に点列を投影した点列に対して二乗距離の合計を最小にする円の中心座標と半径を求める。

$$(x - x_c)^2 + (y - y_c)^2 = r^2 \quad (5.1)$$

$$2x_c x + 2y_c y + r^2 - x_c^2 - y_c^2 = x^2 + y^2 \quad (5.2)$$

$$Ac = b \quad (5.3)$$

$$\tilde{c} = \operatorname{argmin} \|Ac - b\|^2 \quad (5.4)$$

$$\tilde{x}_c = \tilde{c}(0.5, 0, 0)^T \quad (5.5)$$

$$\tilde{y}_c = \tilde{c}(0, 0.5, 0)^T \quad (5.6)$$

$$\tilde{r} = \sqrt{\tilde{c}(0, 0, 1)^T + \tilde{x}_c^2 + \tilde{y}_c^2} \quad (5.7)$$

$$\text{where } A = [x, y, 1]^T$$

$$c = [2x_c, 2y_c, r^2 - x_c^2 - y_c^2]^T$$

$$b = [x^2 + y^2]^T$$

得られた円に関して円の中心点と平面上に投影した点列との距離に対する円の半径との差の合計をエラー値として、線分の場合と同じく RANSAC アルゴリズムを用いてパラメタを得る。

操作開始点の学習

最後に対象物体の操作開始点を獲得する。操作開始点とは冷蔵庫のドアを開ける場合ではドアのハンドル位置に相当する。操作開始点は単純には人間の操作を観察する際の操作軌道において操作対象に初めて接触する点として観測可能であるが、この点を環境の原点から見た座標系において保存、再利用することを考えると、自己位置推定による誤差やオドメトリ誤差の影響を受けてしまう。これらの誤差の影響を排除するためにロボットと操作対象物体との相対関係において開始点を取得する必要がある。本研究ではこのような第三人称視点の観察に基づいて操作点を学習し、ロボットが実際に操作するときに第一人称視点において操作開始点を得る方法として、深層学習モデルの一種である Fully Convolutional Network[61] を利用した画像中の操作開始点の学習を用いる。

学習データとして、ロボットが人間の操作を観察して得られる画像シーケンスのうち、対象物体が状態遷移を行う時点を t_s とすると、 $t_s - t_{offset}$ の時点から t_s までに得られる画像を利用する。各画像について対象物体の領域以外の要素を 0 としたマスク済み画像とし、inpainting によって操作点付近にある人間の手の領域を、手がないような画像となるように置換する [96]。得られた各画像に対して、以下の処理を一定の確率で適用することで Data Augmentation を行った (表 5.1)。

表 5.1: 操作開始点認識モデル学習データセットにおける Data Augmentation

処理	変更パラメタ	適用する割合
RGB 色空間におけるチャンネル値乗算	チャンネル毎の値の乗算値	0.5
HSV 色空間におけるチャンネル値乗算	チャンネル毎の値の乗算値	0.5
ガウシアンブラー	ガウス関数パラメタ σ	0.5
アフィン変換	スケール, 平行移動, せん断	0.8
透視投影変換	変換をなす元画像上の 4 点に対してガウス分布を加算し変換画像の対応点とするときのパラメタ μ, σ	0.8

出力は操作ごとに開始点となる画像領域の要素のみを 1 としたマスクをチャンネルとする画像となるよう, [61] における FCN-8s モデルの出力に対してシグモイド関数をピクセルワイズに適用する. 出力の各チャンネルに対して, 各操作種別の操作開始点である確信度を表すマップとなっている. 確信度マップに対して閾値 c を超えたものを 1, それ以外を 0 とするマスク画像を各チャンネルごとに作り, 教師データとのクロスエントロピー誤差を損失関数 E として計算することで学習を行う. 勾配法として確率的勾配降下法に慣性項を加えた Momentum SGD[97] を用いた.

Momentum SGD は通常の勾配法で起こる局所最適解への収束問題を解消するための手法で, 以下のようにネットワークの重み w を更新していく.

$$w^{t+1} \leftarrow w^t - \eta \frac{\partial E(w^t)}{\partial w^t} + \alpha \Delta w^t \quad (5.8)$$

ここで学習係数 η , 慣性項係数 α はそれぞれハイパーパラメタである.

評価実験

日常生活環境における操作モデルの獲得を行う例として, 冷蔵庫のドアを開ける際の定位置操作モデルの獲得を行った. MS-COCO データセットを用いてセマンティックセグメンテーションの学習を行った Mask-RCNN モデルを用いて操作モデルの幾何形状を取得した (図 5.3).

また操作物体領域のカテゴリが学習済みモデルに含まれない場合は音声対話誘導によって操作物体領域を含むバウンディングボックスを人から取得する. ここでは MS-COCO データセットにおけるカテゴリにはない, 食器洗い機 (dishwasher) カテゴリを取得した (図 5.5).

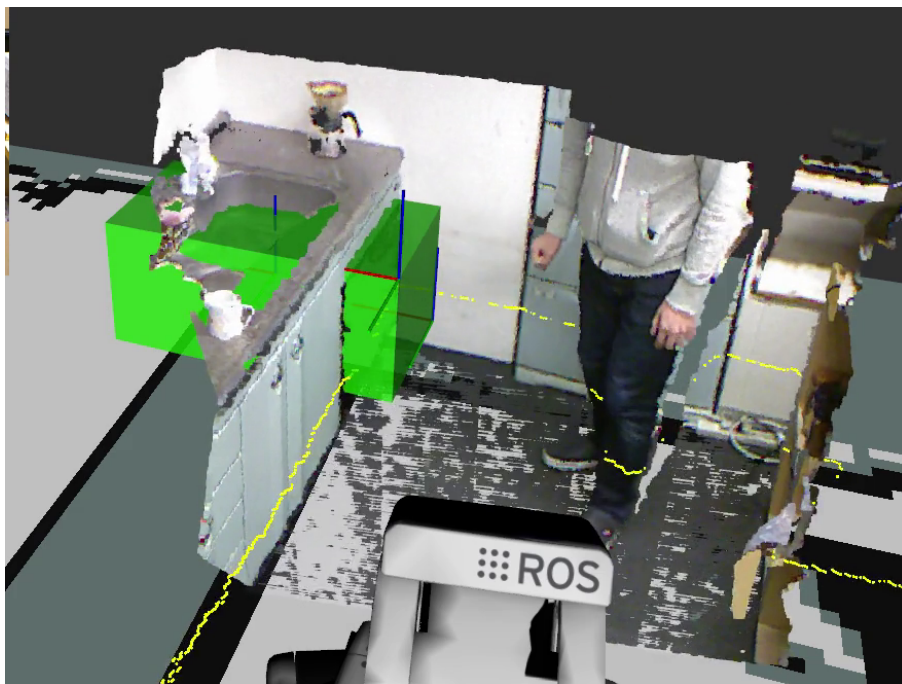
以上によって環境において得られる操作物体の幾何形状と配置が得られた. ロボットは指示されたタスクを実行していない状態のときに, 得られたモデルのうち操作方法を取得していないものについて, 人と操作対象物体を観察することにより, 操作物体の取りうる状態と操作パラメタを取得する.

前述の通り得られた冷蔵庫について, ロボットが操作方法を未取得である状態で, 人が冷蔵



(a) 操作対象手前左上隅位置の認識．上はカメラセンサから得られる画像．下は入力画像を Openpose モデルに入力して推定された人の関節モデルの位置と関係を入力画像上に頂上表示したもの

(b) 操作対象手前右下隅位置の認識



(c) 獲得操作物体領域の可視化

図 5.5: 音声対話誘導を用いた操作対象物体領域の獲得

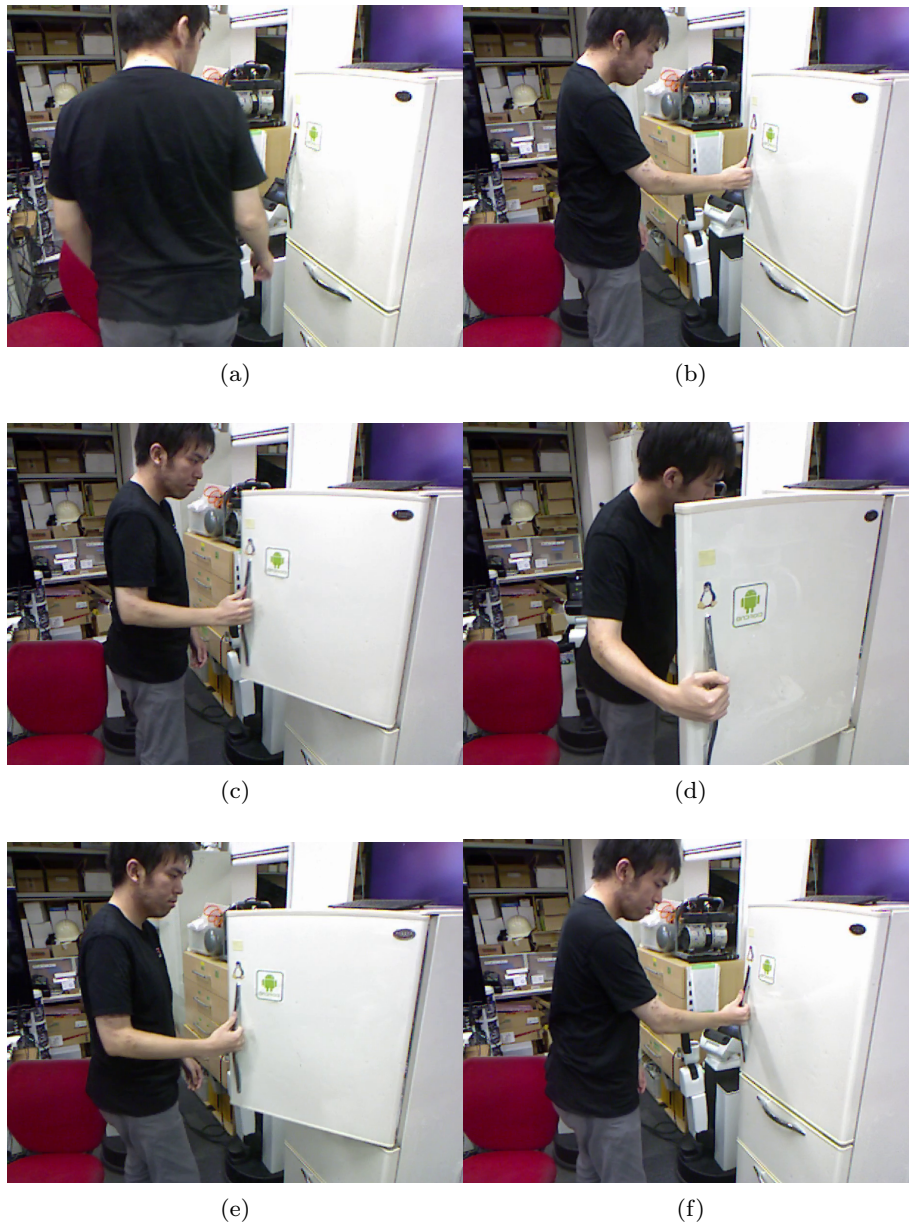


図 5.6: 人による冷蔵庫の操作の観察 (1/2)

庫に操作を行っている方法を観察した (図 5.6) .

操作対象物体の状態は第 5.3.2 節に従い, 入力画像からまず対象物体のみを抽出し, Optical flow を検出することで行った. 図 5.9 に対象物体領域のみを抽出した画像についてグリッド状に配置した点について, Optical flow を算出した結果を示す. 操作が行われていない図 5.9(a) を初期状態として, 連続する 2 フレーム間を用いて Optical flow が算出される. 人により冷蔵庫上部の回転扉が操作とき, 操作開始時には対象物体領域の各点における Optical flow 移動量が大きくなる (図 5.7). 操作が終わり, 扉が開いた状態になると移動量が再び小さくなる. ここでフローの閾値を推定する際のオフセットパラメタ z_f は実験的に 5.0 と決定した. 人に



図 5.6: 人による冷蔵庫の操作の観察 (2/2)

よる冷蔵庫の操作シーケンスから得られる Optical flow 移動量および閾値は図 5.7 のように図示される。

アルゴリズム 1 によって移動量と SIFT 特徴量を用いて移動量が小さいときの対象物体領域の状態が分類される。このときの状態ごとの一致誤差は図 5.8 のようになった。状態推定開始時は状態が State 0 の 1 つのみ存在する。操作が行われていない時は状態一致誤差が小さくなっている (図 5.810 秒以前の赤線)。人が操作を開始すると、Optical flow 移動量が閾値 Flow threshold を超えるため、状態推定は行われなくなる。人の操作が終わり、Optical flow 移動量が再び閾値を下回ると、状態推定は再開される (図 5.81 の間)。状態推定再開時に状態

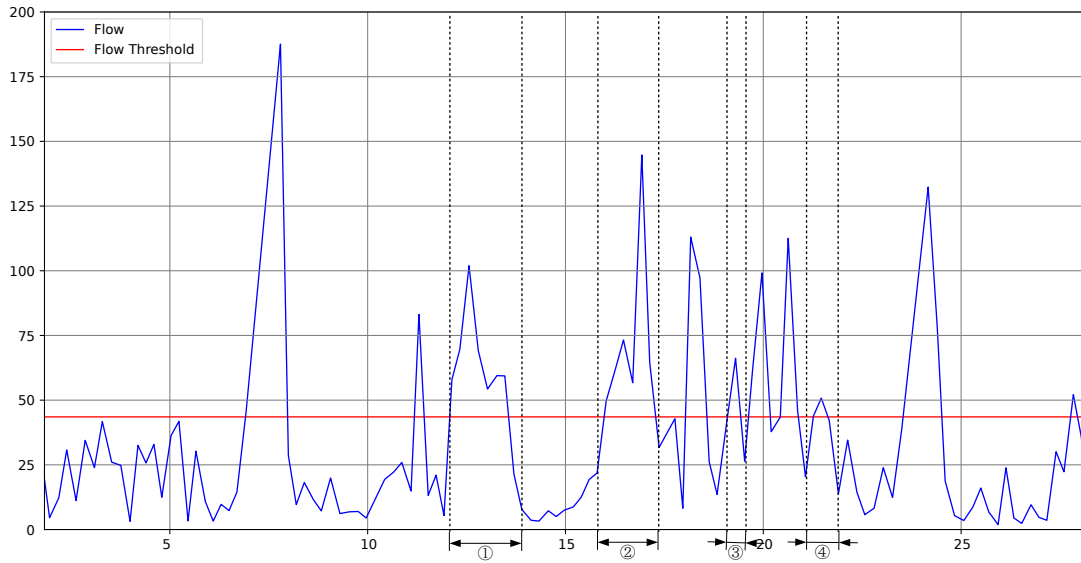


図 5.7: Optical flow を用いた操作物体領域の移動量推移．横軸は時間，縦軸は移動量を表す．青線で表される移動量が赤線で表される閾値を下回った時，操作対象物の状態が静的であると判断し，状態推定を行う．

State 0 の一致誤差が閾値 State Error Threshold を超えているため，新しい状態 State 1 に変化したものとみなされる (図 5.81 右端)．以後 Optical flow 移動量が閾値を下回るときに 2 つの状態一致誤差が計算され，より誤差が小さい状態を操作対象物体の推定状態とされる．2 回めの人の操作の前後で，前では状態 State 1 の，後では状態 State 0 の状態一致誤差が小さくなり，状態が遷移したことが検出された (図 5.82)．このとき既存の状態 State 0 が閾値 State Error Threshold を下回っているため，新しい状態はまだ存在しない．一方で 3 回目の人の操作の後の状態推定では State 0, State 1 に対する状態一致誤差がともに閾値を上回っているため，既存の状態とは異なる新しい状態 State 2 に遷移したものととして新しく状態が登録された (図 5.83)．4 回めの人の操作では同様に状態 State 2 から State 0 への遷移が検出された (図 5.84)．

以上のように上部の回転扉と下部の引き出しが操作されたときに検出された状態の遷移を図 5.10 に図示する．

次にセグメントされた状態間の遷移を行う操作を人の観察によって取得していく (図 5.11)．

第 5.3.2 節による方法で状態間遷移時の手先関節軌道の円・直線近似を行い，パラメタを求める．

また近似したパラメタを利用した軌道と実軌道を 2 次元平面上に投影した結果を図 5.12 に示した．

操作開始点獲得のためのネットワーク学習に用いる教師データセットは，対象物体の状態遷移開始直前の手先関節位置とカメラ画像を用いた． t_{offset} は実験的に 0.5[sec] とした．得られた画像に対して Data augmentation を行い，学習データセットとした．

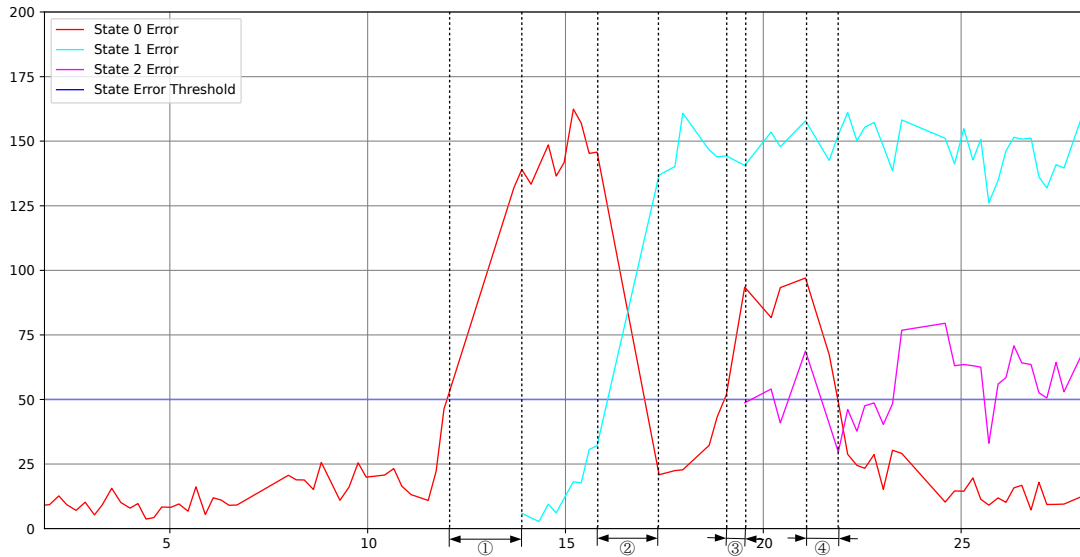


図 5.8: SIFT を用いた操作物体領域の状態一致度推移．横軸は時間，縦軸は状態一致誤差を表す．赤，水，桃色の線はそれぞれ操作対象物体の状態ごとの一致誤差を表す．青線は事前に与えられた状態一致誤差の閾値を表す．すでにある状態すべての一致誤差が閾値を超えた場合，操作物体が新しい状態にあるとみなされる．

使用した FCN モデルの学習時には Momentum SGD のパラメタはそれぞれ $\eta = 0.0001$ ，慣性項係数 $\alpha = 0.99$ と実験的に決定した．学習の進行に伴う損失の推移を図 5.13a に示す．学習の進行に伴い損失が小さく収束していく様子がわかる．評価データセットを用いた学習した重みを用いた認識器の性能評価を行った (図 5.13b)．評価指標として Standard mean Intersection-Over-Union (mIOU) を利用した．mIOU は画像に対して semantic segmentation を行う際に，カテゴリ i に属する画素領域 A とカテゴリ i に判定された画素領域 B が与えられたとき，以下の式で表される．

$$mIOU = \frac{A \wedge B}{A \vee B} \quad (5.9)$$

以上において得られた定位置操作モデルを実ロボット PR2 に搭載し，冷蔵庫のドア開け実験を行った．

1. 図 5.14a: ロボットが操作対象である冷蔵庫にカメラを向ける．冷蔵庫の位置算出には自己位置推定の結果をもとに得られるロボット自身の環境における位置と定位置操作モデルの配置が利用される．
2. 図 5.14b: 操作開始点の学習モデルを利用し，冷蔵庫のドアハンドルを認識し，ハンドルを操作する位置へ台車移動を行う．
3. 図 5.14c: 操作開始点へアームのエンドエフェクタを移動し，グリッパを開く．
4. 図 5.14d: 手先にかかる反力を監視しながらエンドエフェクタを前に出す．反力に変

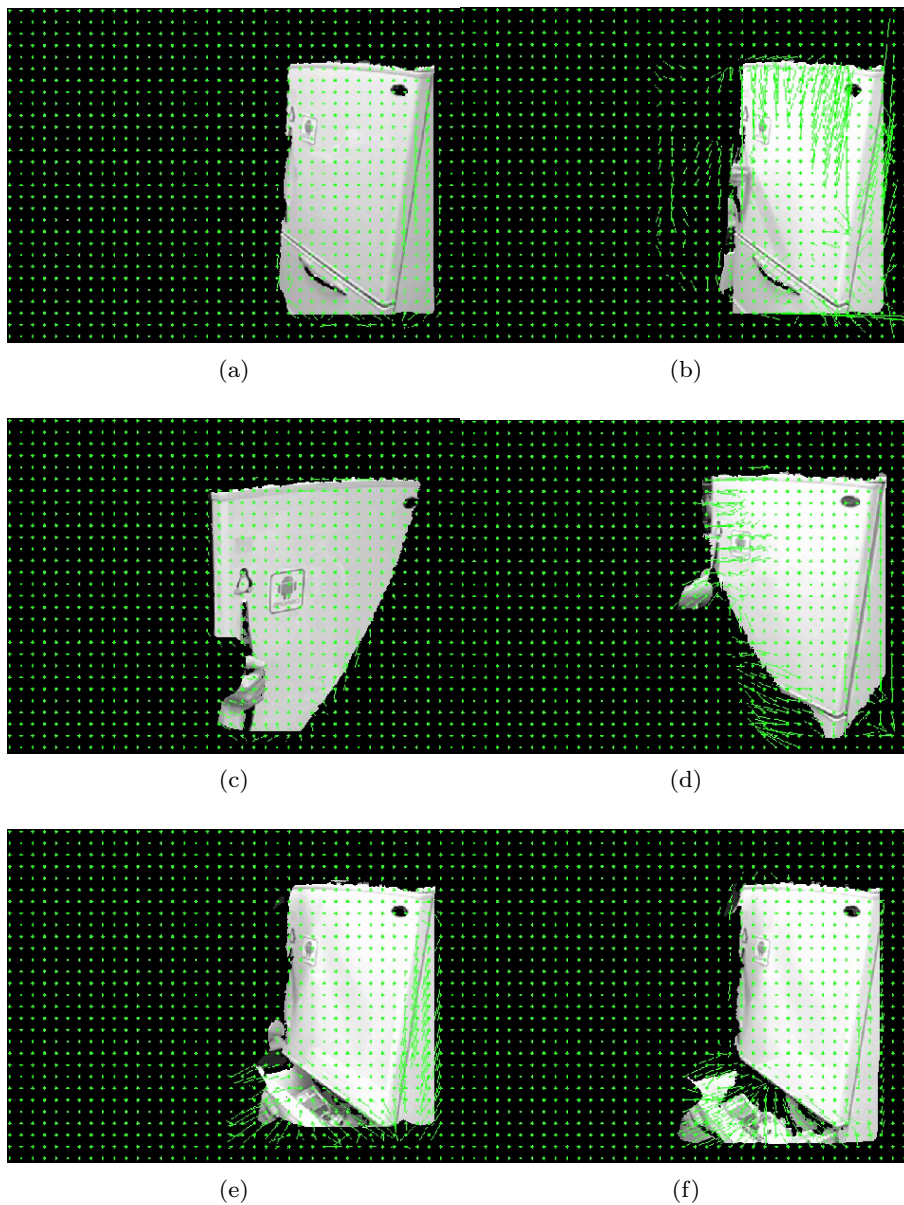


図 5.9: Optical flow を用いた対象物体状態のセグメンテーション

化があったら，グリッパの把持位置がドアハンドルと一致するように手先位置を調整する．

5. 図 5.14e: 学習した定位置操作パラメタを用いてドアを開ける関節軌道を再生する
6. 図 5.14f: 操作物体の可動範囲の境界に達したことによる手先反力の変化か，定位置操作モデルパラメタの学習によって得られた目標操作量に達した時点で動作を終了する．

以上により経験から獲得した定位置操作モデルを用いて冷蔵庫のスポットにおける操作であるドア開けを自律的に行うことが可能であることが示された．

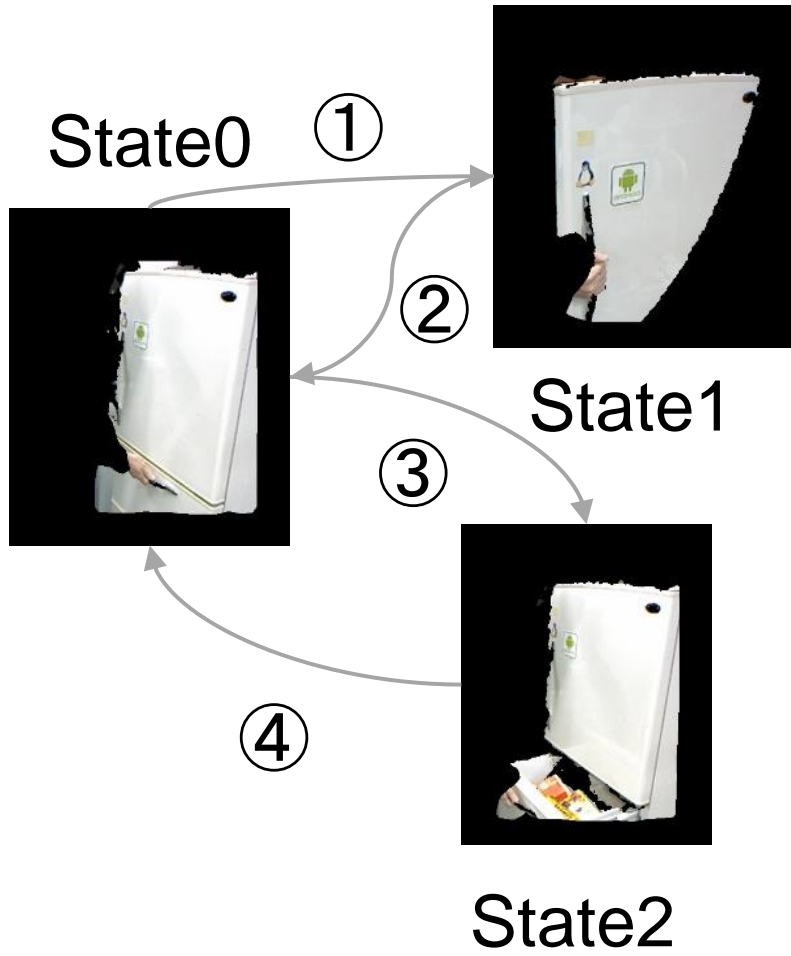


図 5.10: セグメンテーションによる操作対象物体の状態遷移

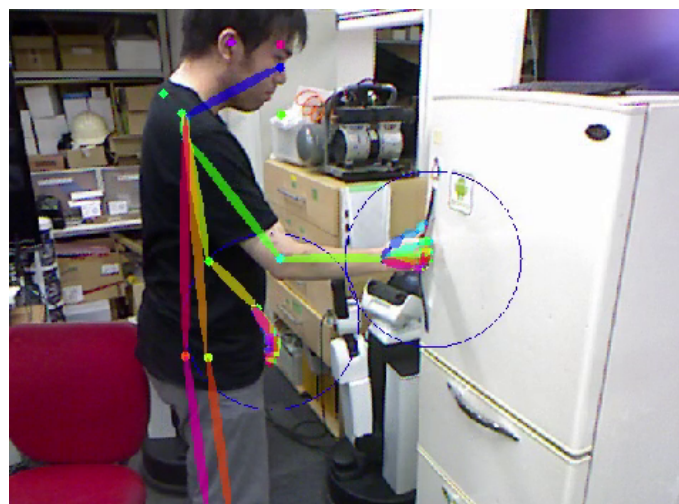


図 5.11: 物体操作時の人の手先軌道観察

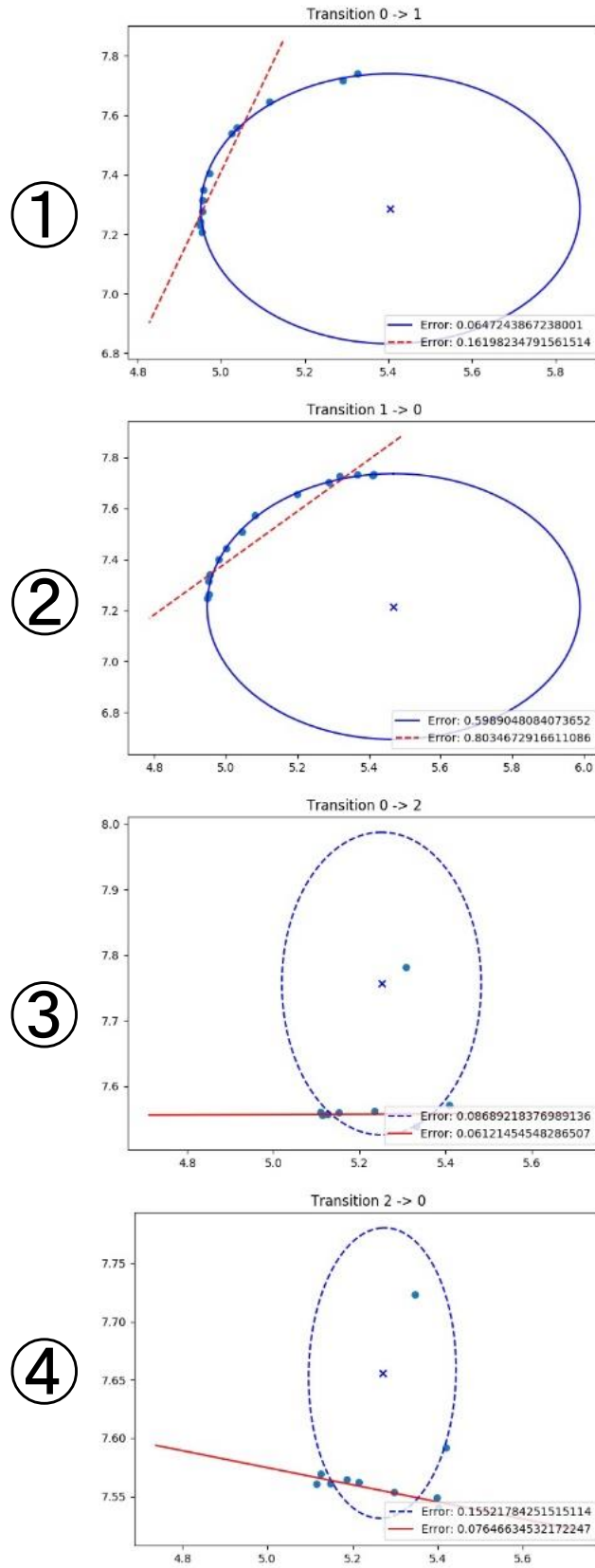


図 5.12: 状態間遷移時の手先関節軌道の円・直線近似

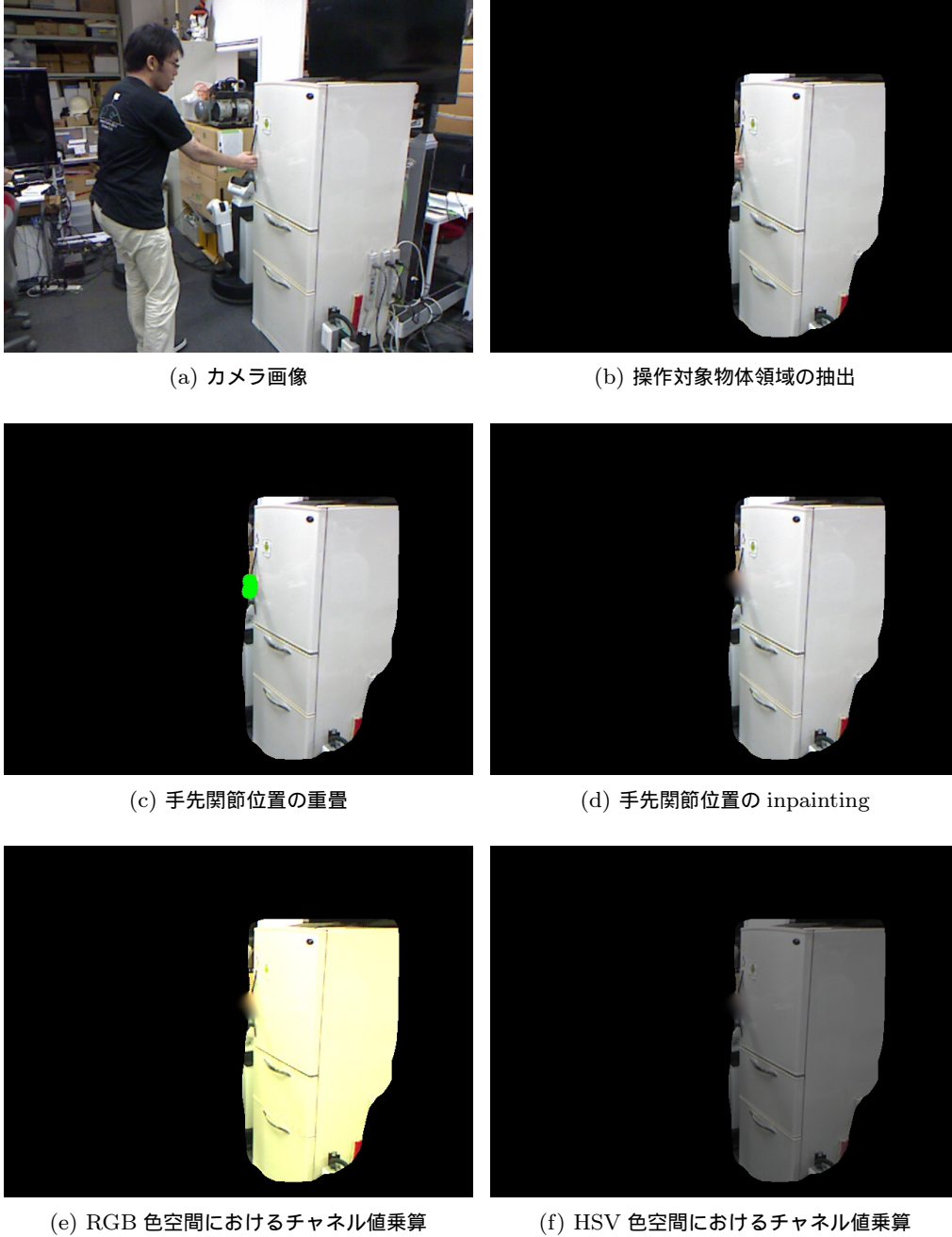
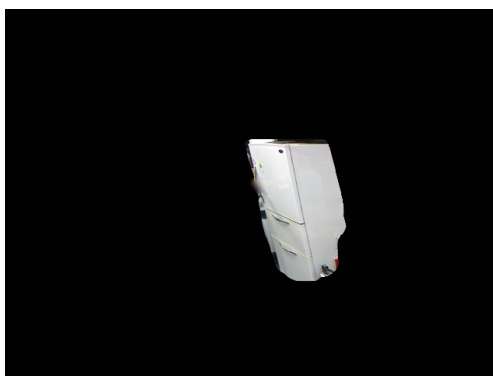


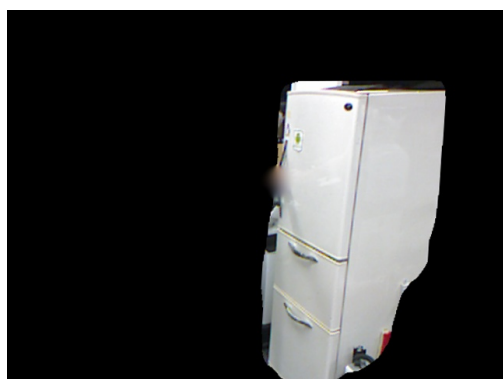
図 5.13: 操作開始点獲得ネットワークの学習に用いるデータとデータ拡張



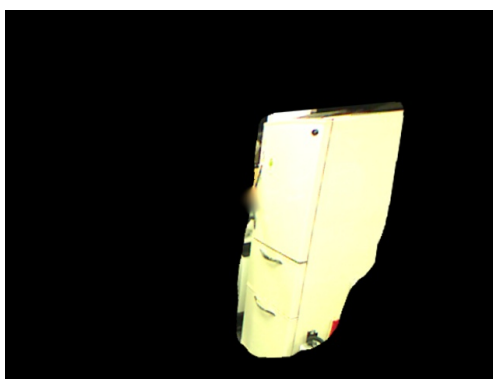
(g) ガウシアンブラー



(h) アフィン変換



(i) 透視投影変換

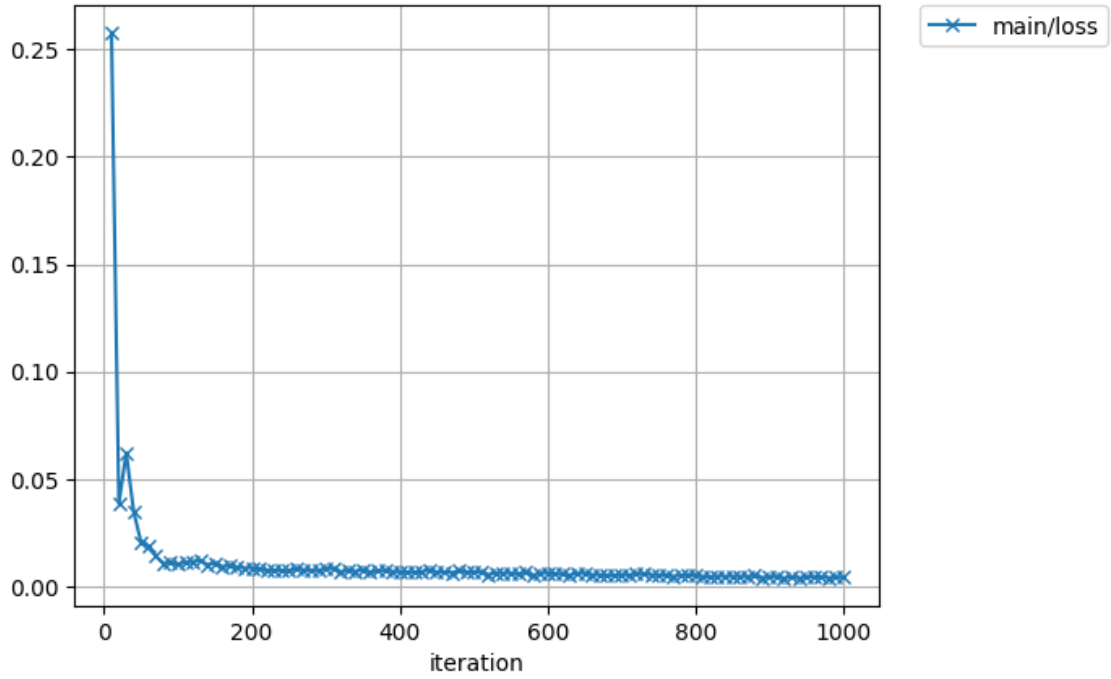


(j) データ拡張によって生成されるデータ例

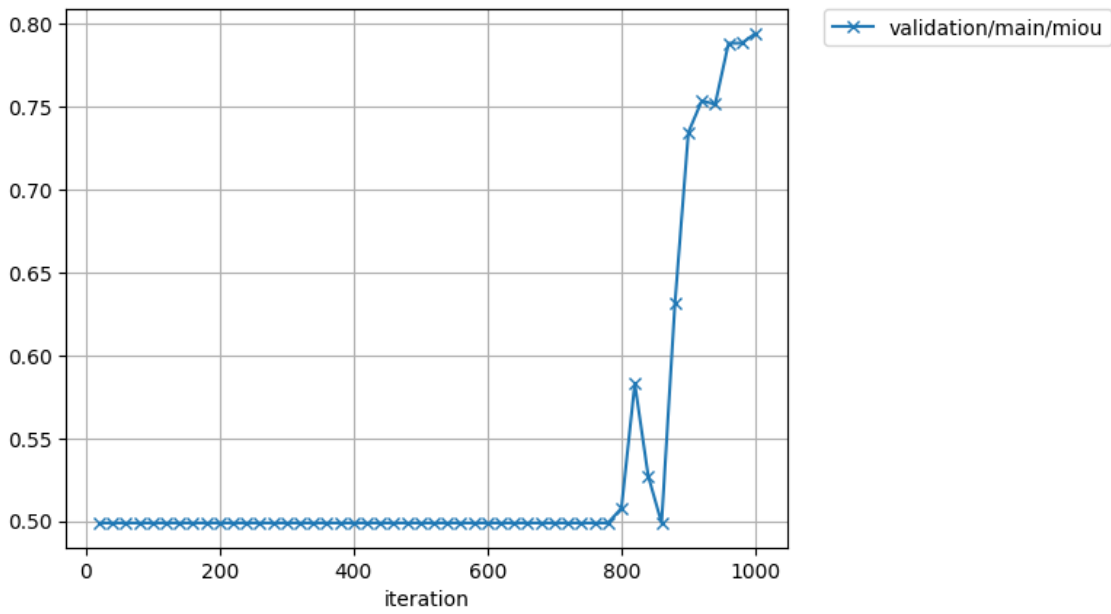
5.3.3 定位置操作モデルの操作対象物体配置の更新

前小節において経験から環境における定位置操作モデルが獲得された。本小節では、日常生活環境の特徴である動的な環境に対応するために、獲得した定位置操作モデルが、模様替えや家具の配置変更などにより変化した環境においても持続的に使用可能になるようなシステムについて説明する。従来では SLAM などによって生成される環境地図に対して、環境モデルや環境モデルを操作するためにロボットが移動すべき目標地点 $spot[6]$ を配置することにより、物体操作を行うための移動は自律移動経路計画によってまず $spot$ へ移動したあと、物体操作を行うために必要な精度で操作物体認識を行うことによって達成されていた。ここでモデルや $spot$ は環境地図上に相対して定義されるため、操作物体の配置変更に対応することが難しい。SLAM で用いられる環境地図は、環境における自己位置を高精度でロバストに推定するためにロボット周辺の情報を広範囲で入力する必要性から、ロボットを支持する地面に水平にロボットへ取り付けられた 2 次元走査レーザレンジファインダなどを通じて得られるが、センサの取り付け位置や、例えば二足歩行機構のように移動中垂直方向の動作が発生するなどロボットの種類などによって、共通して使用することが難しく、物体操作モデルの知識を共有することが難しい。

そこで、本研究では獲得した定位置操作モデルに対して、ロボットがタスクを行っている最



(a) 操作開始点認識モデル学習時の損失の推移



(b) 操作開始点認識モデル学習時のモデル評価推移

図 5.13: 操作開始点認識モデルの学習

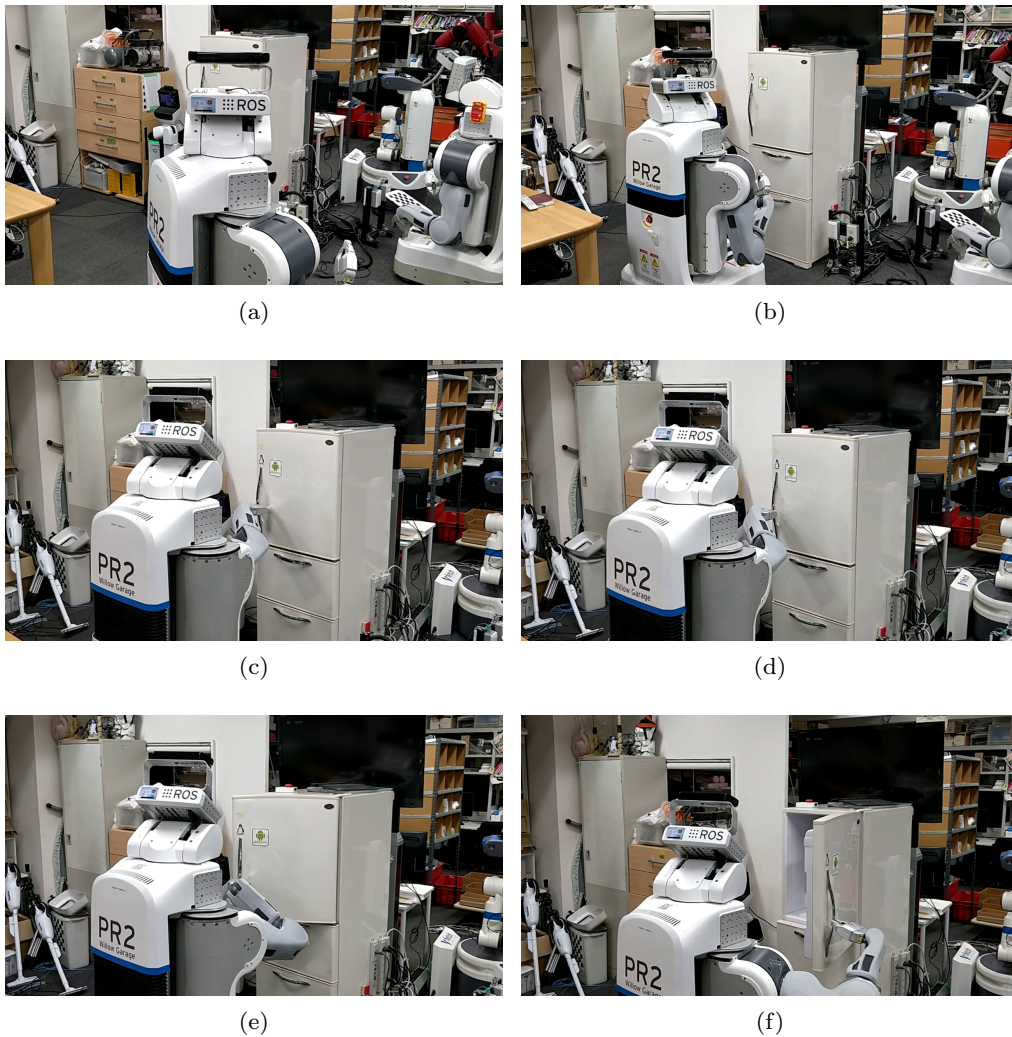


図 5.14: 定位置操作モデルの経験的獲得に基づくロボットの家電操作実験

中に蓄積した経験において観測した外観から認識モデルを自律的に学習し、環境中で物体の配置が変更されても物体を再認識することで、モデルを更新していく。

操作物体認識器学習のためのデータセット獲得

操作物体ラベル画像の獲得 操作対象物体は任意の外観的特徴を持ち、適切な認識モデルを事前に与えることは難しいため、パラメタの学習のみを行うことで認識モデルを形成する深層学習モデルを用いる。深層学習を用いた手法では学習のために大規模な教師データが必要とされる。教師データの作成にはロボットのカメラセンサによって得られる画像データのうち認識対象物体の領域を選択するアノテーションによって行われ、手動で行うことは大きな時間と精度を要する。

そこで、本研究では第 5.3.2 節における定位置操作モデルの獲得によって利用可能となる操作対象物体領域のバウンディングボックスを画像中における操作物体領域のラベル付けに用い

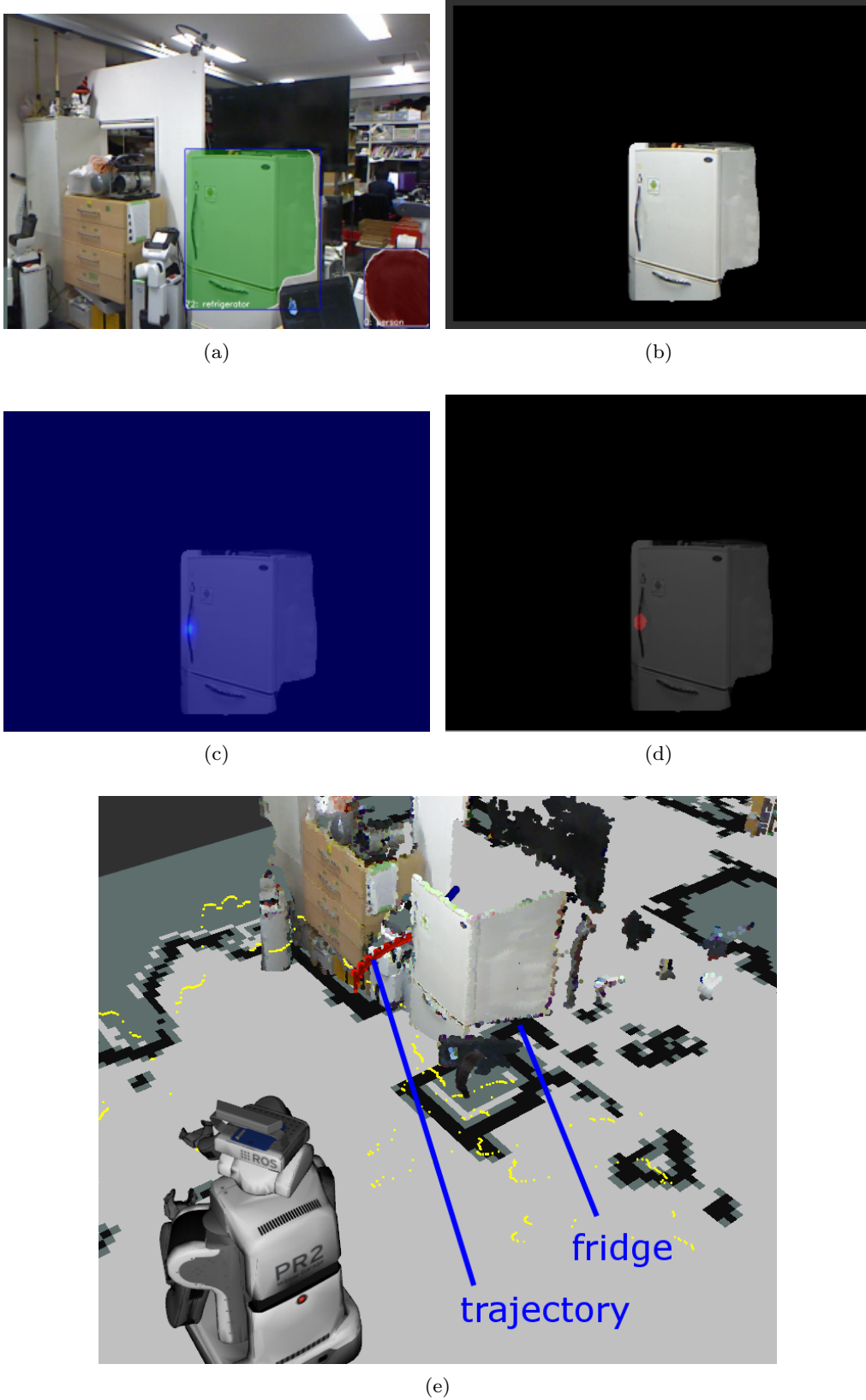


図 5.15: 定位置操作モデルの認識実験

る(図 5.16(a)). バウンディングボックスの頂点を透視投影変換により画像座標系における領域として使うことが最も単純な手法だが, これでは対象物体とセンサ間にある物体によって遮蔽された領域を考慮することができない(図 5.17). したがって, カラー画像に加えて深度画像を用いて得られる三次元点群を用いて, 蓄積した経験に含まれるセンサ画像から操作物体領域のうち空間を占める領域に当たる点列を選択し, 透視投影変換によって画像座標とする. 操作物体ごとに対応するラベルを与え, 画像座標のみをラベル値, その他は0とするラベル画像とする(図 5.16(b)).

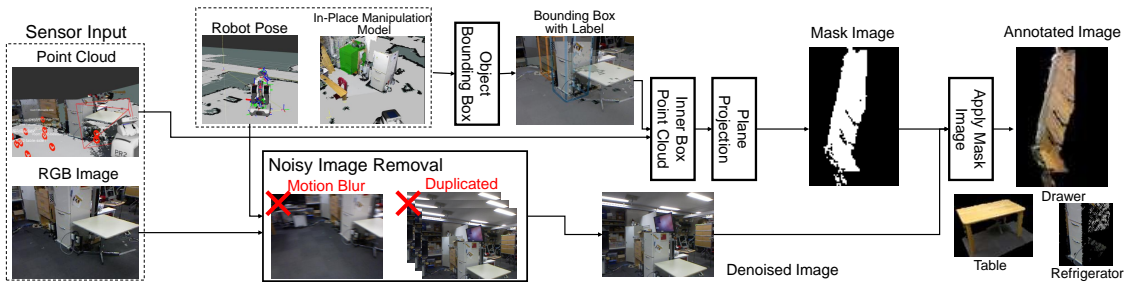


図 5.16: 操作物体ラベル画像データセットの獲得フロー図 ([98] より改変)



図 5.17: 2次元画像における定位置操作物体認識器学習データセットの自動生成時に発生するオクルージョン ([98] より引用)

ロボット座標を用いたノイズありラベル画像の削除 得られたラベル画像にはロボットが動作中のものも含まれるため, 結像中のセンサの物理的移動によりモーションブラーなどのノイズが発生する可能性がある. 一方でロボットが移動を行っていない時にはロボットと環境中の物体の移動がないときにデータセット中の重複画像の割合が大きくなってしまふ [99]. これらのノイズや重複を含む画像は, ロボットの移動台車オドメトリと, 台車とカメラセンサ間の関節角度列およびその時間変位によって得られる環境座標系相対のセンサの位置, 速度を用いて削除されるアルゴリズム 5.

Algorithm 5 ロボットの関節状態情報を用いたノイズを含む画像の除去 ([98] より引用)

```

1: maxVel: threshold for filtering by the velocity of camera device
2: minPos: threshold for filtering duplicated images captured in the same camera pose
3: hasImageInPos  $\leftarrow$  false ▷ Implicit False
4: dataset  $\leftarrow$  Empty
5: camPose  $\leftarrow$  pose(robot, map)
6: prevCamPose  $\leftarrow$  camPose
7: while image  $\leftarrow$  annotator do
8:   camPose  $\leftarrow$  pose(robot, map)
9:   if abs(camVel) < maxVel then
10:    if hasImageInPos = false then
11:      dataset  $\leftarrow$  append(dataset, image)
12:      prevCamPose  $\leftarrow$  camPose
13:      hasImageInPos  $\leftarrow$  true
14:    else
15:      diffPose  $\leftarrow$  norm(camPose - prevCamPose)
16:      if diffPose > minPos then
17:        dataset  $\leftarrow$  append(dataset, image)
18:        prevCamPose  $\leftarrow$  camPose
19:        hasImageInPos  $\leftarrow$  false
20:      end if
21:    end if
22:  end if
23: end while
24: return dataset

```

操作物体認識モデルの学習

得られたデータセットを用いてセマンティックセグメンテーションのための認識モデル学習を行う。学習モデルとして Fully Convolutional Networks(FCN) を用いた (図 5.18)。

深層ニューラルネットワークを用いてセマンティックセグメンテーションなどの識別問題を解くための学習において、識別性能は学習時に用いたデータセットの品質の影響を大きく受けるため、では通常人の手によって慎重にラベル付けを行ったデータセットが用いられる [100]。一方で本研究で作成されたデータセットには図 5.19 のようにラベル値に誤りを含むようなデータも含まれる。

このような座標変換の遅延やカメラセンサの検出不能な動き、自己位置推定誤差、台車オドメトリのノイズなど複合的な要因で発生する誤ったラベルが付与された教師信号を考慮した学

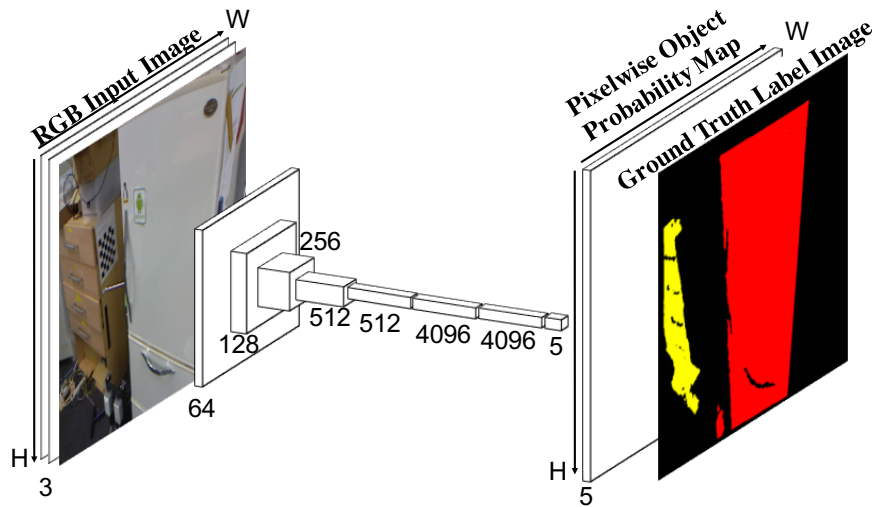


図 5.18: Fully convolutional network のアーキテクチャ図.16 の畳み込み層と 6 の max pooling 層からなる . 画像中の数字は各畳み込み層における出力のチャンネル数を表す . (H, W) はそれぞれ 2 次元画像の高さと幅を表す . ([98] より引用)

習を行うために , Gradient clipping[101] を適用する . Gradient clipping は学習時のネットワーク勾配計算において , 計算後に値が閾値を超えているときにより小さな値に修正することで勾配の発散を防ぐ手法である . 本研究では閾値を計算する指標として勾配の L2 ノルム値を用いた .

評価実験

評価実験では , ロボットはタスクを支持されていない時に頭部をランダムに動かし , 頭部カメラセンサによって環境の観察を行う動作を断続的に行った . また物体運搬や環境構造物操作などのタスクを行っている最中のデータも使用した . 第 4 章を利用して蓄積された経験データベースから頭部カメラセンサから得られる 450,973 のカラー画像と 445,611 深度画像のうち , 第 5.3.3 節によって 417 の画像データセットが自動的に生成された (表 5.2) . ここでノイズの除去のために用いたカメラセンサの位置 , 速度のしきい値はそれぞれ実験から $0.03m$, $0.01m/s$ とした .

得られたデータセットを 8:2 の割合で分割し , それぞれパラメタの学習と評価のために用いた . 深層学習はモデルの出力と教師データの誤差を評価する関数を損失関数として定義し , パラメタを損失関数が小さくなるように勾配法を用いて反復的に更新していくことで行われる . 勾配法として Adam(Adaptive moment estimation)[102] を用いた . Adam は以下の式にしたがいパラメタの更新を行う .

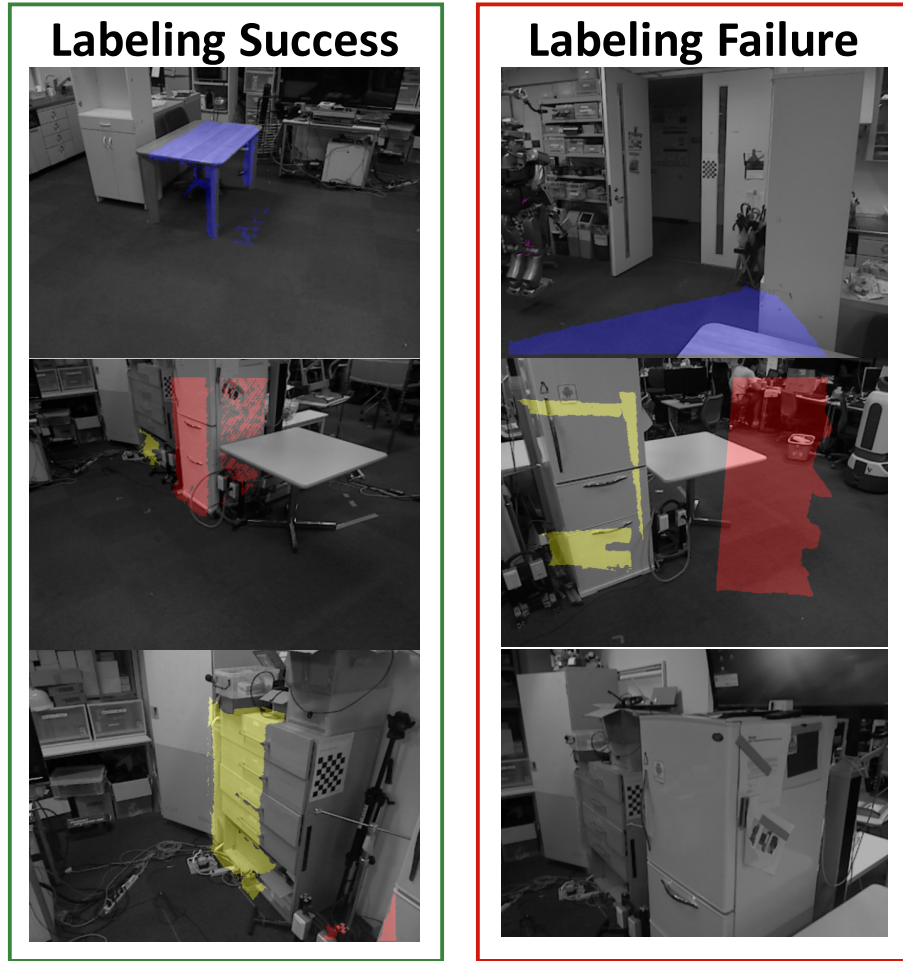


図 5.19: 第 5.3.3 節において説明した方法を用いて生成されるデータセットの例．左側は正しくラベル付けが行われた画像を示す．右側は誤ったラベルのついた画像である．赤色，青色，黄色はそれぞれ冷蔵庫，テーブル，棚としてラベル付された領域，色のついていない領域は背景を表す（[98] より引用）

$$m_{t+1} = \beta_1 m_t + (1 - \beta_1) \nabla E(\mathbf{w}^t) \quad (5.10)$$

$$v_{t+1} = \beta_2 v_t + (1 - \beta_2) \nabla E(\mathbf{w}^t)^2 \quad (5.11)$$

$$\hat{m} = \frac{m_{t+1}}{1 - \beta_1^t} \quad (5.12)$$

$$\hat{v} = \frac{v_{t+1}}{1 - \beta_2^t} \quad (5.13)$$

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \alpha \frac{\hat{m}}{\sqrt{\hat{v} + \epsilon}} \quad (5.14)$$

$\alpha, \beta_1, \beta_2, \epsilon$ はハイパラメタとなるが，本研究では $\alpha = 1e - 5$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e - 8$ を実験的に決定し，用いた．また Gradient Clipping のしきい値の値は 5.0 と実験的に決定した．

表 5.2: ラベルの種類とラベル付けされたデータ数 ([98] より引用)

Object Name	Number of Images	Rate
Background	417	100.0%
Refrigerator	162	38.85%
Drawer	119	28.54%
Table	81	19.42%
Door	63	15.10%
All class	417	100.0%

学習を行った際の学習曲線を図 5.20 に示す．予測精度が学習・評価時ともに上昇した．また学習中のモデルのパラメタを利用して推論を行った結果についても示した．

本実験では日常生活環境として第 5.3 節において経験から定位置操作モデルを得た時の配置と一致する環境 (図 5.21a) と，家電・家具の配置を変更 (図 5.21b) した環境において「冷蔵庫からテーブルへ飲み物を持ってくる」タスクの計画実行をすることを通じて環境モデルに手動で変更を加えることなく自律的にタスクを遂行することを確認した．

本タスクは以下のような手順で行われた．

1. 図 5.22a: ロボットが操作対象である冷蔵庫に接近する
2. 図 5.22b: 冷蔵庫のドアハンドルを認識することで冷蔵庫の扉を開ける
3. 図 5.22c-図 5.22d: 冷蔵庫内にある缶コーヒーを把持し，冷蔵庫扉を閉める
4. 図 5.22e-図 5.22f: テーブルを認識し接近する
5. 図 5.22g-図 5.22h: テーブル上に缶を置く

タスクを遂行するための行動計画には失敗復帰を考慮した行動計画器である `task_compiler` を用いた．得られた動作遷移グラフを図 5.23 に示す．家具・家電配置の変更後，(find-object fridge) においてロボットはまずロボットの定位置操作モデルとして記憶されている位置がカメラの視野内に入るように姿勢を変更する．冷蔵庫を使用しない他のタスクの実行中や他のタスクを行っていない時の環境の観察によって変更された冷蔵庫が認識された時にはすでに定位置操作モデルの操作物体配置は更新されるが，ここでは更新を行わず，前回すなわち配置変更前の配置のままとする．モデルを参照した結果あるべき場所に冷蔵庫を認識できなかった場合，(find-object fridge) は終わり，(look-around) 動作を行う．この動作ではロボットが姿勢を変更し，カメラセンサの視野を変更する．(look-around) 動作は再び (find-object fridge) において視野内の操作対象物体認識を行い，認識が成功すると，操作対象物体を操作可能な位置へロボットの台車移動を行う (approach-to fridge) 動作を行う．移動が成功したあとロボットは実際の物体操作 (open-door fridge) を開始する．

また同じタスクを二足歩行等身大ヒューマノイドロボット HRP2-JSKNT を用いて実行し，

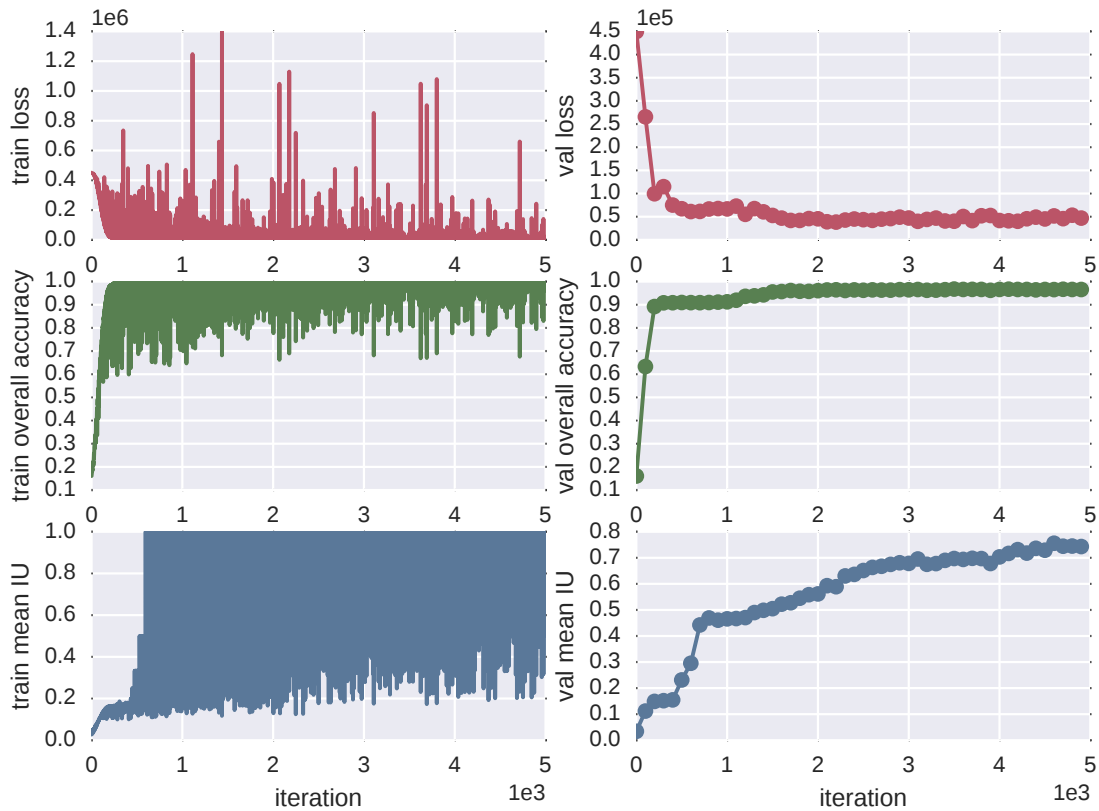
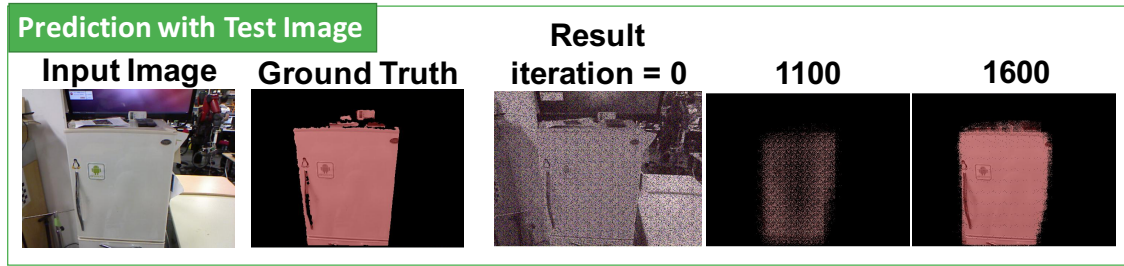


図 5.20: FCN モデルの学習曲線とテストデータにおけるセグメンテーション結果．上部は学習中の各イテレーションにおけるパラメタを用いてセグメンテーションを行った結果を可視化した画像を表す．下部は損失関数，正確度，mean IU 値を学習用データセット（左側）と評価用データセット（右側）を用いた場合についてプロットした図．（[98] より引用）

ロボットの物理的制約を受ける環境地図を用いないことによって，異なる種類のロボットにおいてもモデルを共通にタスクを実行可能であることを示した（図 5.24）．

5.4 アイテムとスポットの相互作用に基づく確率的物体配置地図の経験獲得

本節では、従来の知識データベースでは固定的に与えられてきたものの配置に関する知識を、ロボットがタスクを行っていくうちに蓄積された経験から獲得する方法について説明する。ロボットが扱うものの場所 spot に関する知識を、ロボットシステムにおける行動計画器が扱えるようにするために必要となる情報は、環境に固定された座標系における位置・姿勢といった幾何的な情報だけでなく、環境における構造物や他の物体との関係に関する知識も必要となる。本研究では、幾何的な情報によって物体の経験的な存在位置を確率モデルとして表現し、経験から獲得する。これを確率的物体配置地図 (Probabilistic Object Localization Map) という。

5.4.1 蓄積経験からの物体配置情報の抽出

本節ではロボットの経験的知識の獲得のうち物体に関する知識獲得のための仕組みについて説明する。第4章において蓄積された情報のうち、タスク実行中に認識された物体に関する情報を抽出する具体的な実装となる。蓄積した経験のうち得られるすべての物体認識のうち、ロボットが持っている最中の物体のように配置されていない状況にある物体の情報は除かれる。長期的に蓄積された経験から抽出したデータのうち物体に関するラベルと位置情報を環境地図上にプロットしていく。物体のラベルと位置情報を検出するフローを図5.25に示す。

物体のラベル識別はカラーカメラセンサから得られる画像を用いて行う。ここでは第3.3.5節において述べたSSD物体検出器を例として用いている。得られた物体のラベルに対して位置情報を付加する方法として最も単純なものは物体候補領域のある点に対応する深度画像の要素値から得られる深度情報とカメラパラメータを用いる方法だが、以下のような問題がある。

1. 深度情報は対象物体の表面における値であるため、把持操作を行う上では位置にズレが生じる。
2. 複雑な表面形状を持つ物体ではロボットと物体間での小さなブレや移動による誤差で深度が大きく変化してしまう。
3. 物体候補領域のすべての画素値が物体の占める領域であるとは限らない

一方で、(1)に関しては壁にかかっている物や押し動作を伴う物体に関しては問題とならない。したがって、物体が支持平面の上に置かれているか、それ以外の配置状況にあるかに応じてラベルとの対応付けに使用する位置情報を変える。支持平面上に物体が置かれているかを確認するために、深度画像を用いて平面検出を行う[104]。検出された平面のうちロボット自身の支持平面に対して傾きが一定値以下の面を物体支持平面とする。深度画像から三次元点群を生成し、物体支持領域をなす平面に対してロボットの床面から受ける反力ベクトルと内積が負にならないような法線方向の正の向きにある点群のみを抽出する。次に抽出した点群から物体

支持平面の点群を除く．得られた点群は支持平面領域上に支持された物体表面となる．この点群をユークリッド空間においてクラスタリングすることで，物体ごとの点群が得られる．物体ごとの点群に対応する深度画像領域とカラー画像から検出された物体のラベルに対応する物体候補領域の重複が大きな点群に対し，重心位置を求めることにより，ラベルに対応する位置情報とする．一方でカラー画像における物体候補領域が平面検出によって得られた物体支持領域にない場合は物体候補領域の中心位置に対応する三次元点群の点に対して，kd-tree[105]を用いて近傍点列を探索する．得られた点群に対して同様に重心位置を求めることによりラベルに対応する位置情報とする．

この時 SSD モデルは MS-COCO データセットを用いて学習したパラメタを用いた．したがってクラスラベルは MS-COCO データセットのものを使用した．

次に経験から得られた 3 次元空間における物体のラベルと位置情報を，多変量混合ガウスモデル (Multi-Variate Gaussian Mixture Model, MVGMM) を用いてクラスタリングを行う．

d 次元データをクラスタリングする際の高変量混合ガウスモデルをなすクラスタ数 N のうち i 番目のクラスタは多変量ガウスモデルで表され，パラメタ $\boldsymbol{\mu}_i \in \mathbb{R}^d, \boldsymbol{\Sigma}_i \in \mathbb{R}^{d \times d}$ は以下のようなになる．

$$P(\boldsymbol{x}) = \sum_i^N w_i p_i(\boldsymbol{x}) \quad (5.15)$$

$$p_i(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x} | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \quad (5.16)$$

$$= \frac{1}{(2\pi)^d |\boldsymbol{\Sigma}_i|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_i)\right) \quad (5.17)$$

$$\boldsymbol{\mu}_i = \frac{1}{n_i} \sum_{j=0}^{n_i-1} \boldsymbol{X}_j \quad (5.18)$$

$$\boldsymbol{\Sigma}_i = \sum_{j=0}^{n_i-1} (\boldsymbol{X}_{i,j} - \boldsymbol{\mu}_i)^T (\boldsymbol{X}_{i,j} - \boldsymbol{\mu}_i) \quad (5.19)$$

確率密度関数の対数尤度は観測されたデータ数 n のもとで以下のように表され，

$$\ln P(\boldsymbol{x} | \boldsymbol{w}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \ln\left(\prod_{i=0}^{D-1} p(\boldsymbol{x}_i | \boldsymbol{w}, \boldsymbol{\mu}, \boldsymbol{\Sigma})\right) \quad (5.20)$$

$$= \sum_{j=0}^{n-1} \ln p(\boldsymbol{x}_j | \boldsymbol{w}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad (5.21)$$

$$= \sum_{j=0}^{n-1} \ln\left(\sum_{i=0}^N w_i \mathcal{N}(\boldsymbol{x}_j | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\right) \quad (5.22)$$

ガウス分布の総和の対数となるので，尤度の最大化問題を陽に解決することは難しい．そこで，反復計算手法の一つである EM (Expectation-Maximization) アルゴリズムを用いて尤度を最大化させるパラメタ $\boldsymbol{w}_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j$ を推定する．クラス識別性能の評価にはベイズ情報量

Algorithm 6 EM アルゴリズムとベイズ情報量規準を用いた多変量混合ガウスモデルのフィッティング ([103] より引用)

```

1: procedure FITMVGMM(data, maxComponents)
2:   bestModel  $\leftarrow$  Null
3:   minBIC  $\leftarrow$  infinity
4:   modelTypes  $\leftarrow$  ["full", "diag", "spherical", "tied"]
5:   for each N in 1..maxComponents do
6:     for each T in modelTypes do
7:       params  $\leftarrow$  random(N, T)
8:       model  $\leftarrow$  GMM(N, T, params)
9:       model  $\leftarrow$  EMFitting(model, data)
10:      bic  $\leftarrow$  BIC(model, data)
11:      if bic < minBIC then
12:        minBIC  $\leftarrow$  bic
13:        bestModel  $\leftarrow$  model
14:      end if
15:    end for
16:  end for
17:  return bestModel
18: end procedure

```

規準 (Bayesian Information Criterion, BIC) を用いた。クラスタ数は最大となるクラスタ数 *maxComponents* が与えられるとき、1 から *maxComponents* までの各クラスタ数となるモデルを用いて推定を行い、BIC 値が最小となるクラスタ数を選択する。また、このとき学習に用いるデータ数が小さいと過学習を起こしてしまうため、パラメタのうち、各クラスタの共分散行列に対して以下の拘束を加えたモデルも同時に用いて推定を行う。

full 各クラスタの共分散行列は任意

diag 各クラスタの共分散行列が任意の対角行列で与えられる

spherical 各クラスタの共分散行列が単位行列とスカラ定数の積で与えられる

tied 各クラスタの共分散行列はクラスタ間で共通の任意の行列で与えられる

ここで *d* はデータが 3 次元空間上で与えられることから 3, 最大クラスタ数 *numComponents* は 5 と実験的に決定した。

5.4.2 評価実験

本節において説明したシステムを実際にロボットへ搭載し、実世界でタスクを行っている間に蓄積した長期経験を用いて、確率的物体配置地図としての知識が獲得可能であるこ

とを検証した．本実験では Willow Garage 社製等身大双腕移動台車ロボット JSK-PR2 と Fetchrobotics 社製単腕移動台車ロボット Fetch[106] を用いて 41 日間日常生活環境において経験の蓄積を行った．経験の蓄積を行っている間，ロボットは第 3.3.8 節によって行われるアイドル動作や，人のタスク指示によって環境内での自律移動や物体操作を含む日常生活支援タスクを行った．各ロボットの頭部に搭載されている RGB カメラと深度カメラセンサから得られる 2 次元画像を使用した．

全蓄積期間中において 29 カテゴリの物体が合計 13848 回認識された．各カテゴリの認識回数は図 5.26 のようになった．ロボットがタスクを行った環境における 2 次元地図上に投影した認識物体の配置情報は図 5.27b のようになった．

$maxComponents = 5$ のもと，認識された各カテゴリについて，認識によって得られた物体配置情報から多変量混合ガウスモデルを用いたフィッティングを行い，確率的物体配置地図を得る．

カテゴリ refrigerator の確率的物体配置地図は各重みパラメタ w が以下のような 5 つのガウス分布からなるモデルとなった．

$$\mathbf{w}_{refrigerator} = [0.519, 0.187, 0.170, 0.083, 0.0408] \quad (5.23)$$

図 5.28 はフィッティングされたガウスモデルを 2 次元環境地図上にプロットしたもので， $d = 3$ であるが，以下の式によって投影したものである．

$$z = \sum_{i=0}^n w_i \mathcal{N}((\mu_{xi}, \mu_{yi}, 0), \Sigma_i) \quad (5.24)$$

同様にすべてのカテゴリについて確率物体配置地図を計算し，プロットしたものを図 5.29 に示す．

5.5 人とアイテムの相互作用に基づく嗜好モデルの経験的獲得

本節では，日常生活を送る人間を長期的に観察することによって蓄積される経験に基づいて獲得される人の嗜好に関する知識とその獲得方法について説明する．人の嗜好モデルを獲得することで，はじめはロボットに対して逐一日常生活タスク内容を指示，あるいは対話によって曖昧性の中から人が希望する目的状態を選択する必要があるものを，人ごとの嗜好性に基づき人に問い合わせることなく推論し，タスクを実行可能になる．

タスクの実行時に省略される内容，すなわち嗜好モデルに基づき推論すべき内容が，ロボットがタスク実行に必要な内容のうちどの程度を占めるかはタスクの内容や人の経験，文化など様々な要因に応じて変化するため，ロボットへの指示のうち指示の任意の部分が省略されても推論が可能なモデルであることが要求される．

タスク指示において得られるアイテムの指示には対象となるアイテムを一意に特定可能なシンボルだけでなく，アイテムの属するカテゴリやタスクを指示する人，アイテムに関連する動

作，アイテムの性質など多岐にわたる．

このような人からの指示による任意の部分的事前情報が与えられたときに，タスクを行う上で必要な知識としてもっともらしいものを推定する手法としてベイジアンネットワークが挙げられる．ベイジアンネットワーク [107] は複数の変数に対する依存関係を条件付き確率と有向非循環グラフ構造によって説明する確率モデルの1つである．変数間の依存関係を元に不確実で・欠損のある情報が与えられたときの状況や原因の予測を行うことに使われる [108] ．

本研究では人の嗜好性を獲得するために人からロボットが受けたタスク指示の履歴を用いた．

図 5.30 のようにアイテムのシンボルを表す変数ノードに対して子ノードにアイテムに付随する特徴を説明する変数ノードを接続，アイテムシンボルに対する親ノードとしてアイテムのカテゴリと動作を表す変数ノードを接続したモデルを与える．アイテムの特徴を表すノードとしてアイテムの外観から抽出可能な色ヒストグラム (第 7.3.5 節) と基本形状 (第 7.3.5 節)，容積 (第 7.3.5 節) を用いた．

5.5.1 タスク指示受付履歴に基づくベイジアンネットワークを用いた嗜好モデル学習実験

日常生活環境におけるタスク指示として指示をロボットが受け付けた経験データから嗜好モデルのパラメタ学習を行った．各指示においてロボットは人の識別を行い，User 確率変数に対応する値を獲得する．人は指示としてロボットにしてもらいたい操作と操作の対象となるアイテムを指定する．会話フローの例を表 5.3a に示す．操作は第 7.3.1 節における deliver, place の 2 種類，カテゴリは tea, coffee, sweets, hamburger, milk, shampoo の 6 種類，カテゴリに属するアイテムは iemon, georgia, wanda, boss, manju, bigmac, morinaga, merit とした．それぞれのアイテムは複数の外観的特徴を持っている．色特徴 (Color) は green, blue, red, white，基本形状特徴 (Shape) として box, circle，容積特徴 (Volume) としては short, small, big, tall がそれぞれ変数の取る値とした．

ロボットが指示を受けた経験から図 5.30 のグラフにおける条件付き確率表を最尤推定法によって学習 [107] したモデルを用いて，人による省略指示を行った．会話のフローを表 5.3b に示す．得られた指示において操作対象アイテムが具体的に指示されていないため，ロボットは嗜好モデルを用いて指示における対象アイテムを推論する．指示や顔認識結果により得られるモデルにおけるノードの観測値から求める Item ノードの確率変数が各値を取る事後確率を求める．ここでは観測値として Category ノードの値と User ノードの値をエビデンスとして与えた．表 5.4 に事後確率を示す．ここで Item ノードの値として確率が最も大きい Georgia を対象アイテムとして推論してタスクを実行可能となった．

表 5.3: 対話に基づくアイテム操作タスク指示の会話フロー

Robot	Hi, may I help you?	Robot	Hi, may I help you?
User	Coffee, please.	User	Coffee, please.
Robot	You want Georgia, Wonda or Boss?	Robot	OK, wait a moment...
User	Wonda please.		
Robot	OK, wait a moment...		

(a) 嗜好モデル不使用時の対話フロー

(b) 嗜好モデル使用時の対話フロー

表 5.4: 人の指示による観測と学習済みベイジアンネットワークに基づく Item ノードの事後確率表

No.	Symbol	Probability
1	georgia	0.80
2	wanda	0.10
3	boss	0.10
4	iemon	0.00
5	morinaga	0.00

5.6 おわりに

本章では、従来では固定的に与えられてきた環境状態記述をロボット自身の経験に基づいて自律的に獲得するための手段として、ロボットが人との共生を行う日常生活環境において、タスクを行っていくうちに蓄積した経験から人や、人やロボット自身によって常に動的に変化する環境、物体について獲得可能な知識を、経験において注目する対象を人とタスクを行う上で環境中において意味のある場所であるスポット、タスクにおいて扱われる物体を表す概念、アイテムとの関係性から分類し、それぞれについて知識獲得の方法を説明した。

人とスポットに関して経験的に得られる知識として、環境における定位置における操作物体とその操作方法に関する知識を表す定位置操作モデルを挙げ、その獲得手法を示した。また環境が動的に変化する中で、操作物体の配置が変化しても対応する知識を更新していく方法についても示した。

人とアイテムに関して経験的に得られる知識として、人の嗜好に関する知識を獲得する方法について説明した。アイテムに関して人から指示を受けた経験に基づき、同一カテゴリ内における物体の人ごとの嗜好を反映した物体性質モデルを構成する手法について示した。

アイテムとスポットに関して経験的に得られる知識として、環境における確率的物体配置地図を定義した。ロボットが扱う物体をタスク実行中に認識した経験を用いてカテゴリごとの配置情報を抽出し、物体がよくある位置を確率モデルを用いて表す方法を示した。

またそれぞれの知識獲得手法を実ロボットを用いて長期的に蓄積した経験から獲得可能であることを実験により検証した。

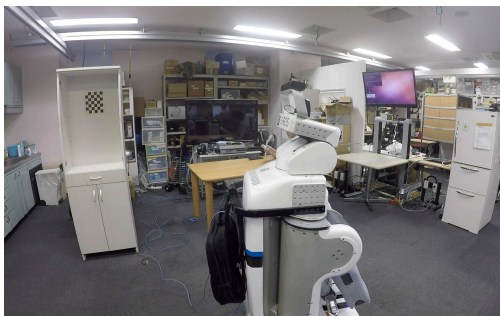


(a) 模様替え前の環境 ([98] より引用)



(b) 模様替え後の環境 ([98] より引用)

図 5.21: 模様替え前後の実験環境



(a)



(b)



(c)



(d)



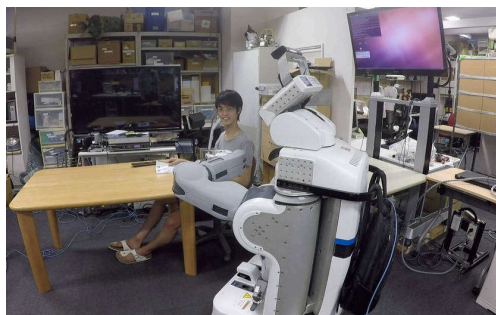
(e)



(f)



(g)



(h)

図 5.22: 模様替えを行った環境における冷蔵庫から缶を取ってくるタスクの実行 ([98] より引用)

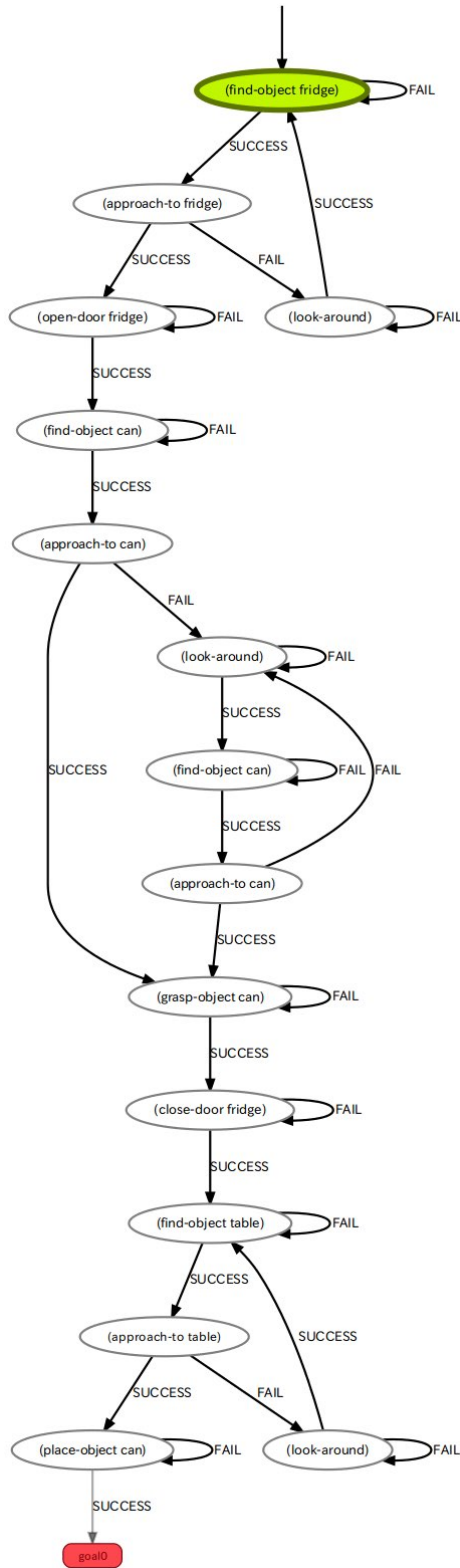


図 5.23: 冷蔵庫から缶を取ってくるタスクの動作遷移グラフ ([98] より引用)

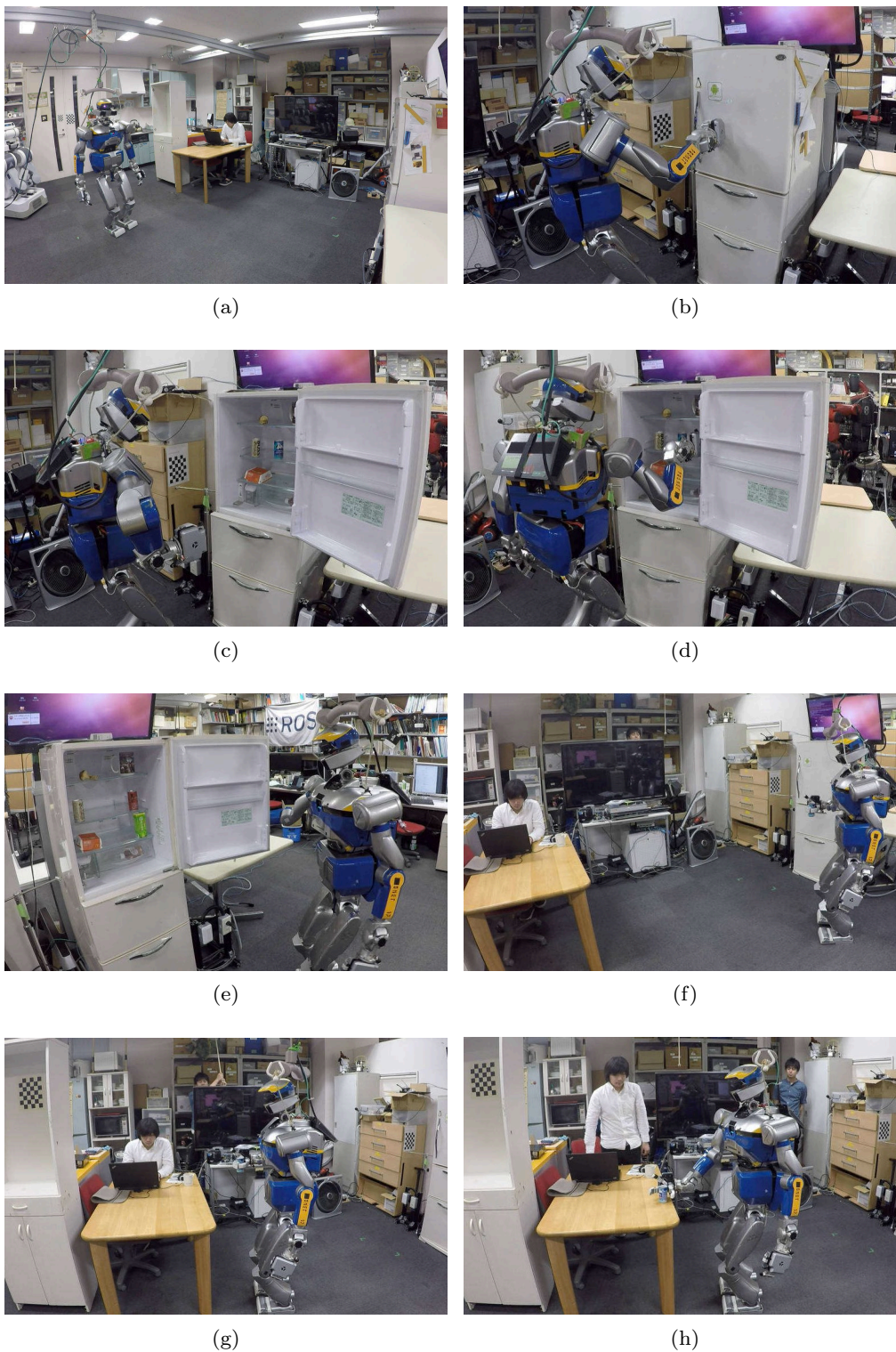


図 5.24: 二足歩行等身大ヒューマノイドロボット HRP2-JSKNT において提案システムを用いて冷蔵庫から缶を取るタスクを実行 ([98] より引用)

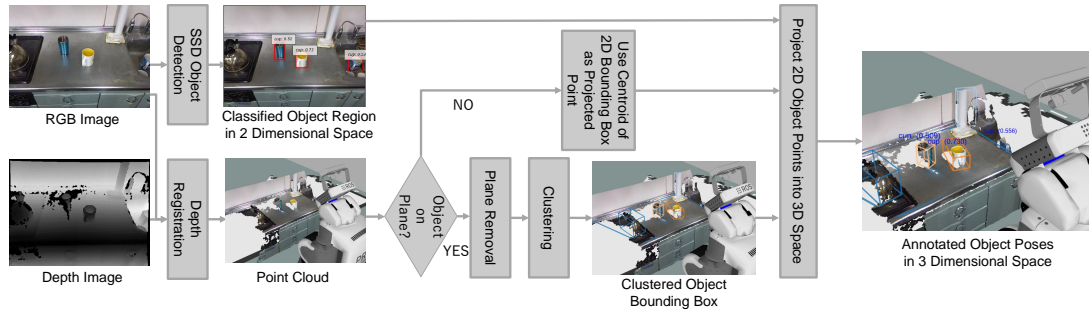


図 5.25: 蓄積経験からラベル付けされた物品の位置情報を集約するフロー（[103] より引用）

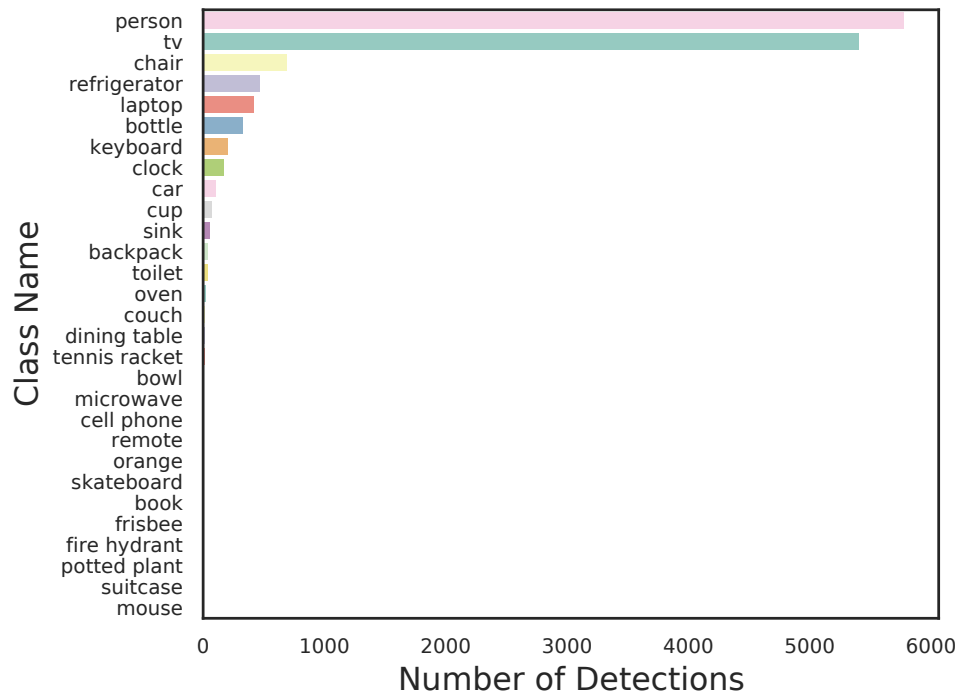


図 5.26: 長期蓄積経験におけるカテゴリ別物体認識回数（[103] より引用）

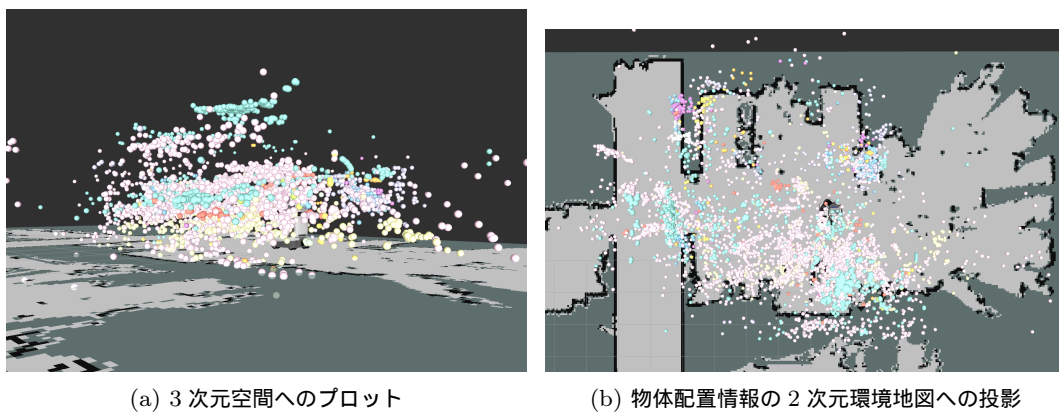


図 5.27: 認識物体のカテゴリ別配置情報 ([103] より引用)

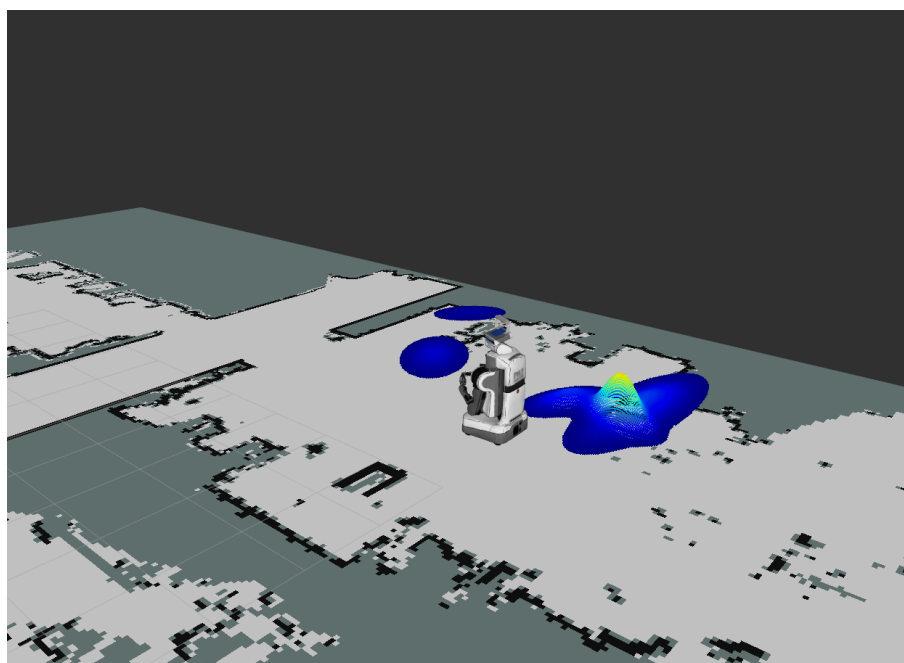


図 5.28: 冷蔵庫の確率的物体配置地図

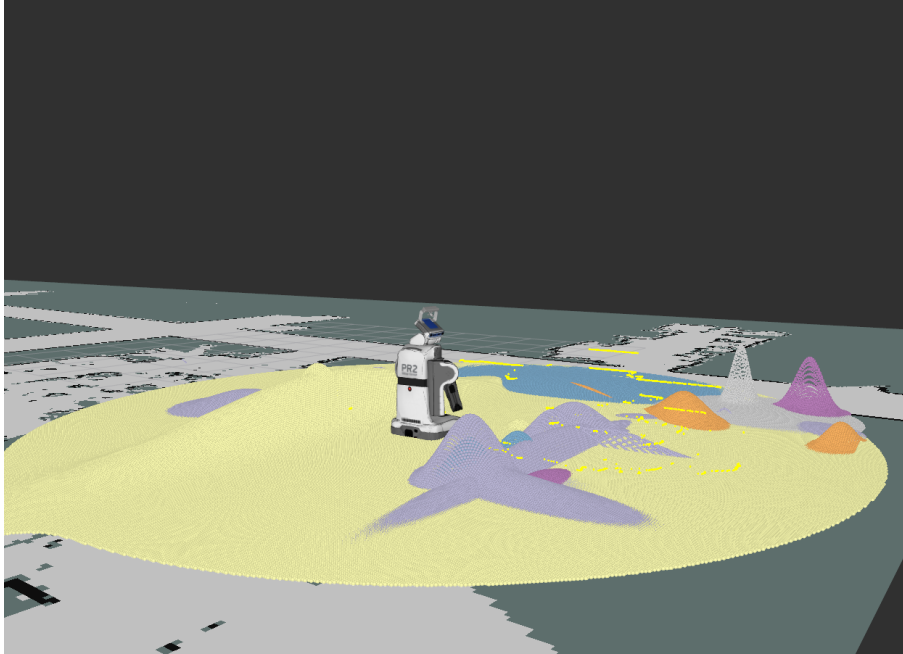


図 5.29: 確率的物体配置地図 ([103] より引用)

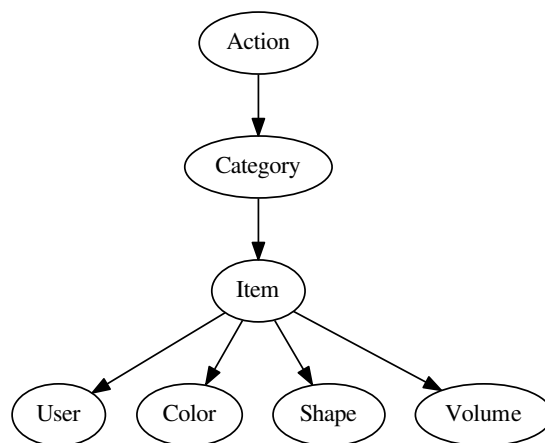


図 5.30: 省略を含むタスク指示における対象アイテムの推論を行うベイジアンネットワーク

第 6 章

環境に非依存な状況レベルタスク記述と局所合理性に基づく具体化機構

6.1 はじめに

本章では、事前に与えられる知識のみで記述することが困難な日常生活行動を、ロボットが環境におけるタスク実行を通じて得た環境や共生する人間に関する局所的な合理性を有する経験的知識を用いて、環境に即した具体的なタスク記述に展開可能である、環境に非依存なタスク記述とその具体化機能について説明する。本章の内容は第 2 章における環境に非依存な状況レベルタスク記述と局所合理性に基づく具体化機能のための具体的な実装にあたる。

本章ではまずシンボリックな環境状態・動作遷移記述を用いた日常生活タスクの行動計画と制御について説明し、従来では固定的に与えられてきたタスク記述に加えて、環境から経験的に得られる知識を用いた計画が可能になるよう記述を拡張する。第 5 章で説明した経験による獲得可能知識の分類に基づき、第 5 章で経験的に獲得した知識モデルを用いて、対応する状況レベルタスク記述を実世界で計画・実行可能なタスク目標状態へと具体化するモジュールを順に適用することで、従来の行動計画器に入力可能なタスクプランを生成する (図 6.1)。本章では状況レベルタスク記述を環境状態に即した具体的なタスク記述へ変換するための手がかりとなる状態への制約として具体化演算子を定義し、具体化演算子をもとにタスクを行動計画器へ入力可能な具体的な記述へ変換していくシステムの構成方法について説明していく。

また、提案機能を実際のロボットへ実装・長期エピソード記憶を蓄積し、蓄積したデータをもとに片付けタスクを行うことで、提案システムの評価を行う。

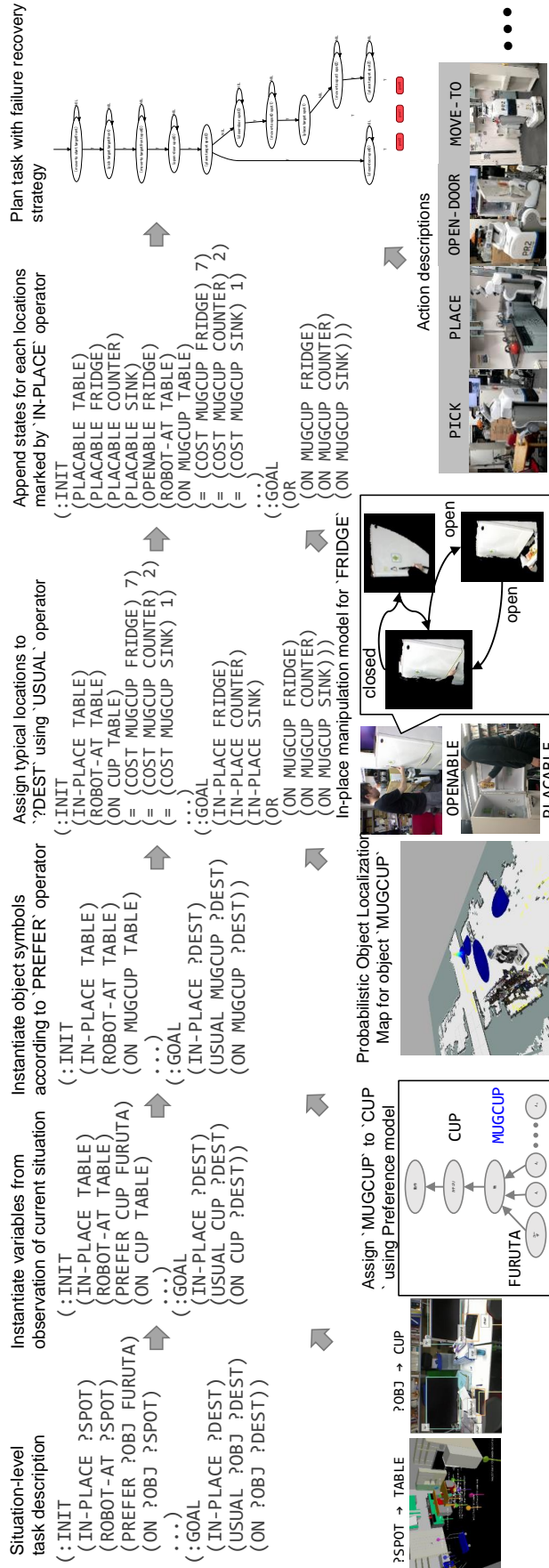


図 6.1: 状況レベルタスク記述の具体化機能に基づく日常生活タスク計画機能の全体図 ([103] より一部引用)

6.2 関連研究

6.2.1 自律ロボットシステム

反射主導型自律エージェント

ロボットのセンサ情報からの入力に対して、ロボットが取るべき動作を予め対応付けることによって自律的に振る舞うロボットシステムである。代表的なものとしてサブサンクションアーキテクチャが挙げられる [18]。サブサンクションアーキテクチャは「測距センサの値が十分大きければ前進する」、「前方に障害物があれば 90 度回転する」といった単純な if-then ルールによってセンサ入力とアクチュエータ出力を対応付ける並列動作モジュールを組み合わせることにより、システム全体として意思決定を自律的に行うシステムである。各モジュールは相対的に優先度を持っており、優先順位の高いモジュールにおける決定はそれより下位の決定よりも優先されることで、システム全体の意思決定として一貫性をもたせることで、アクチュエータの資源競合や矛盾を回避している。各モジュールはセンサ入力に対して行動計画を行うことなく対応する行動を反射的に出力するため、動的な環境においてロボットが意思決定をすばやく行うことができる利点がある。このシステムは iRobot 社の自律掃除ロボットである Roomba [109] に搭載され、実際に家庭環境に広く普及している。

しかしながら、家電操作や道具の利用といった環境状態に深く依存し、複雑な手順を要する家事支援タスクをこのシステムで実現するためには、多数のモジュールを用意する必要があるほか、ある行動中における意思決定が正しく行われているかを正しく設計・修正するためにはモジュール全体の振る舞いを開発者が把握し、優先度を注意深く設計していかなければならない。モジュール単位の認識と意思決定に関する情報を高度にしていくと、全体で必要なモジュール数を減らすことが可能となるが、認識処理時間の増大を招くなどサブサンクションアーキテクチャの利点である反射的で高速な意思決定を行いにくなくなってしまう。

計画主導型自律エージェント

ロボットが環境や自身に対して内的なモデルを持ち、センサによる環境の知覚をもとに状態モデルを逐次更新し、与えられた目標状態に至る状態遷移を行う行動を計画していくシステムである。自律ロボットシステムとしては最も古典的で研究が盛んに行われてきた [110, 111, 112, 113]。タスクを行う環境に関する現在状態や目標状態全体を不足なくデータとして表すことは不可能であるため、何らかの記述形式をとることになるが、代表的なものとしてシンボリックな状態記述が用いられてきた。シンボリックな状態記述では、「コップ」(cup) や「テーブル」(open) などの物や「ドアが開いている」(open) といった環境物の状態、ロボット自身の機能的な概念（例えば双腕マニピュレーションロボットにとっての「右腕」(rarm)）などの名前を伴う知識を代表するシンボルとして、それらのシンボルの関係性を表す述語を連言や選言、否定などにより集合的に表すことで状態を表現する。例えば「コップがテーブルの上にある」という状態は (on cup table) となるように 2 つのシンボルと on という述語から

なる式で記述される．この状態記述を用いて日常生活支援を遂行するためには行動計画を用いてロボットの行動意思決定を行う．行動計画はこのような環境状態記述として初期状態（現在の状態記述）と目標状態（あるタスクが与えられたときにそのタスクを達成したとみなせる時点での状態記述）が与えられたときに初期状態から目標状態に至る状態遷移，すなわち動作列を推論する事である．このとき可能な状態遷移はロボットエージェントにとって実行可能な動作のリストとして定義される．このような行動計画を行うための技術として論理型プログラミング言語 Prolog[114] や PDDL[10] に代表される一階述語論理に基づいて記述された命題を推論する計画器が多数存在する [115, 116, 117, 118] ．

PDDL は AI(Artificial Intelligence) プランニングの言語における標準化の一環で，IPC(International Planning Competition) と呼ばれるプランニングのための国際競技のために開発され，PDDL によって記述された問題を解決するための高性能なソルバーが開発されてきている．言語の使用としては同じくシンボリックプランニング言語である STRIPS[119] や ADL[120, 121] などの影響を受けている．

以下シンボリックなタスク記述言語を利用したロボットのタスク計画システムの構成法について説明する．

PDDL ではまず解くべき問題記述を以下の 2 つに大別する．

1. ドメイン記述 (Domain description)

問題を記述するためのシンボルとそれらシンボルの関係を示す述語の定義を記述する．

2. 問題記述 (Problem description)

Domain Description において宣言したシンボルを用いて表現された問題の初期状態と目標状態を記述する．1 つの Domain Description について，複数の Problem Description を記述することができる．

Domain Description では以下の事項を定義する

型 (Object types) プランナが解くタスク空間において扱われる物体シンボルの種類を記述する

述語 (Predicates) シンボルの状態や関係性を記述する

アクション (Actions) シンボルと述語の組み合わせによって表現される状態を遷移させる動作を記述する

また Problem Description では以下の事項を定義する．

オブジェクト (Objects) 状態の表現に使われるシンボルを記述する．オブジェクトは必ず Domain Description で宣言したいずれかの型に属する

初期状態 (Initial condition) シンボルと述語の組み合わせによって表現される．プランナが探索を開始する時点での状態．

目標状態 (Goal condition) プランナが目標とする状態．

以下に PDDL を用いたロボットのタスク記述の例を示す．ここではロボットがとる

ことのできる動作，すなわち状態遷移として MOVE, PICK, PLACE が定義されている．状態遷移はそのものの名前，動作に必要なパラメタと，状態遷移を適用可能な状態の条件 (precondition)，状態が遷移したときに状態がどのように変更されたかを示す影響 (effect) からなる．初期状態として与えられた (AT TABLESIDE) はロボットの初期位置を表し，(ON BOTTLE TABLESIDE) はテーブルサイドにカップがあるという環境状態を意味している．目標状態 (ON BOTTLE KITCHEN) はカップがキッチンにある状態を意味する．コップをテーブルからキッチンへと片付けるタスクはこのように目標状態として与えられる．完全な初期状態をロボットシステムが得るためには環境をタスクを行う前に完全に認識する必要があるが，実際にはオクルージョンなどの物理的制約はもとより，環境を表現するために必要な述語やシンボルが膨大になってしまうため難しい．したがって多くの行動計画器の処理系では記述されていない状態に関してはすべて偽であるとする閉世界仮説を前提としている．

DOMAIN:

name: TIDYUP

type: (ITEM SPOT)

predicates: ((ON ?OBJ - ITEM ?SPOT - SPOT)

(ONHAND ?OBJ - ITEM)

(AT ?SPOT - SPOT))

actions:

- (MOVE ?FROM - SPOT ?TO - SPOT)

precondition: ((NOT (AT ?TO)))

effect: ((NOT (AT ?FROM))

(AT ?TO))

- (PICK ?OBJ - ITEM ?AT - SPOT)

precondition: ((AT ?AT)

(ON ?OBJ ?AT)

(FORALL (?OBJ2 - ITEM) (NOT (ONHAND ?OBJ2))))

effect: ((NOT (ON ?OBJ ?AT))

(ONHAND ?OBJ))

- (PLACE ?OBJ - ITEM ?AT - SPOT)

precondition: ((AT ?AT)

(ONHAND ?OBJ))

effect: ((NOT (ONHAND ?OBJ))

(ON ?OBJ ?AT))

PROBLEM:

objects:

```

BOTTLE - ITEM
TABLESIDE - SPOT
KITCHEN - SPOT
initial condition: ((AT TABLESIDE)
                   (ON BOTTLE TABLESIDE))
goal condition: (ON BOTTLE KITCHEN)

```

この記述を行動計画器へ入力することで、以下のように状態遷移列が得られる。

1. (PICK BOTTLE TABLESIDE)
2. (MOVE TABLESIDE KITCHEN)
3. (PLACE BOTTLE KITCHEN)

シンボリックな状態記述を用いて日常生活タスクの目標状態を与えることでタスクを遂行していくロボットシステムでは、実行中の時点でロボットが現在の状態をモデルとして所有・更新していることから、動的な環境において考慮しなければならない失敗に関してその復帰手順を陽に記述することができるという利点が挙げられる。岡田らは状態遷移リストの記述に加えて、記述の影響 (effect) を失敗状態に変更したものを失敗動作として定義して計画を行うことで、失敗状態への遷移を含む行動計画を行った。さらに計画によって生成された失敗状態を初期状態としてもとの目標状態までの行動計画を別途行って得られる状態遷移列を通常の状態遷移列に対して結合していくことで、失敗と正常状態への復帰遷移を含む状態遷移グラフを生成して、失敗を考慮した行動計画を実現した [20]。

2002年に登場した PDDL2.1[122]では、プランの選択時に任意の実行コストを設定できるようになり、目標状態を達成する解が複数存在するときにコストを最適化するような条件を指定できる。

本研究では計画主導型のロボットシステムとしてシンボリックな状態記述言語を用いた目標状態記述によりタスクを記述する。

6.2.2 タスクの抽象表現

プランニングの分野においてタスクを抽象的に記述することは過去に行われてきている。Fikesらは STRIPS プランナに対して与える状態遷移オペレータと各オペレータのパラメータセットを用いて、計画中に繰り返し出現するオペレータのセットを1つの複合的なマクロオペレータとみなすことで、過去に行った探索結果を再利用する仕組みを提案した [123]。Mokhtariらはロボットに対してタスクの名前とプランナのオペレータとパラメータに対応する指示を人が複数回与えた時に、タスク実行内容のオペレータとパラメータの共通性から、変数を用いた抽象タスク記述の獲得法を提案している [124]。これらの手法によって既存のオペレータと定数パラメータの順序付きセットに対して新しく複合的なオペレータを定義することで一般化されたタスクを表現することが可能となる一方で、抽象化されたタスク記述に対して環境に

において共生する人の経験や文化といった環境固有の知識が含まれているかどうか抽象タスクの一般性に依存しているため、日常生活タスクの目的のように環境に依存しない常識によって定義可能な知識としてタスクを記述することが難しい。

タスクを状態記述によって表現するのではなく、状態遷移を行う動作を使ってタスクを記述する手法として HTN(Hierarchical Task Network) がある [125]。HTN ではタスクはサブタスクの集合として定義され、多層構造になっている。プランナは階層構造を持つタスクの各階層における状態の粒度で計画を行えばよいため、環境に関する完全な記述を持たなくても計画を開始することが可能となっている。各階層における記述の粒度をプログラマが調節可能であるため、上位の階層において環境に依存しない記述を行うことも可能である。一方で、状態記述によってタスクを表現しないため、第7章において説明される人によるタスクの随時割り込みや再開を行った時に必要となる中断状態や中断からの復帰動作を伴う作業の続行・再開を表現することが難しい。

本研究では古典的な状態としてのタスク記述をベースに具体化演算子を導入することで、環境に非依存な記述を行うための仕組みを提案する。

6.3 日常生活タスクの状況レベル記述

ロボットが自律的にタスクを計画・実行するためにはタスクの目的をロボットの処理系に対して記述として与えておく必要がある。第2.4節において従来自律ロボットエージェントに対して与えられてきた行動の記述において用いられている知識はタスク実行状況依存性、環境依存性に基づき分類されることを述べた。本節では日常生活タスク記述として従来では固定的に記述されていたものをタスク実行状況や環境に依存しない形で記述するための仕組みについて述べる。この記述を行うことによって環境や毎回変化するタスクの初期状態に依存しない抽象的なタスクをロボットに与えることが可能となる。一方で抽象的な記述はロボットが実際にその場の環境においてタスクの動作列を計画するのに十分な表現を持っていない。したがって、ロボットが自律的に経験から局所的な合理性に基づく環境依存の知識を獲得し、状況レベルタスク記述に対して適用することで、具体的なタスク記述を得ることが必要となる。本節では第5章において説明した、ロボットが自律的に経験から得られる局所的な合理性を伴う環境依存の知識表現をもとに状況レベルの日常生活タスクを記述する仕組みを提案する。

抽象度の大きいタスク記述を行うために、従来のシンボリックな状態記述言語を用いたタスク記述に対して、具体化演算子と変数を導入する。具体化演算子は状況レベルタスク記述において、任意のシンボルを割り当て可能な変数に対して局所的な合理性に基づく制約を与える記述となる。

具体化演算子は第5章で説明した環境から得られる局所合理性を含む知識に基づき、以下を定義する。

USUAL 演算子 場所と物の関係に基づく局所合理性を与える演算子である。第5.4節において得られた確率的物体配置地図に基づき物が存在する配置に関する制約を与える。

IN-PLACE 演算子 特定の場所に基づく局所合理性を与える演算子である．第 5.3 節において得られる定位置操作モデルに基づき場所に関する状態の制約を与える．

PREFER 演算子 人の物に関する局所合理性を与える演算子である．第 5.5 節において得られる人の嗜好モデルに基づき物の種類に関する制約を与える演算子である．

6.3.1 USUAL 演算子

USUAL 演算子は物が存在する場所との局所的な合理性を与える演算子であり以下のようなインターフェースを持つ．

```
(defoperator USUAL (ITEM SPOT &key AFTER BEFORE))
```

スポットとアイテムに属する変数または定数シンボルを引数としてとり，アイテムが経験的によく配置されている可能性が高いスポットを表す．アイテムとスポットが定数を表すシンボルであるか，変数であるかに応じて具体化される意味が異なる．以下に具体的説明を示す．

例えば (USUAL CUP ?SPOT) のようにアイテムを表すシンボルを定数，スポットを変数としたとき，シンボル?SPOT は CUP に対する経験的なよくあるスポットを表すシンボルとなる制約を受ける．(USUAL ?ITEM TABLE) のようにスポットを表すシンボルを変数，アイテムを変数とした時，?ITEM は TABLE に存在する確率が大きいアイテムのシンボルとなる制約を受ける．クラスタ数 n が 2 以上であるとき，アイテムを表すシンボルが変数のときは各クラスタの平均からなる地点に対象物体が配置される状態どうしの論理和 (OR) 結合として具体化される．スポットを表すシンボルが変数のときはスポットに対して接触関係のあるアイテムを列挙し，それらの記述を論理積 (AND) 結合として具体化される．

また:AFTER や:BEFORE とともに時間的な制約がさらに加えられる．例えば，(USUAL ?ITEM TABLE :AFTER MORNING) のように時間に関する制約を加えることで，確率的物体配置地図の生成に用いられる経験に対して時間的な制約が与えられる．

6.3.2 IN-PLACE 演算子

IN-PLACE 演算子は特定の場所に基づく局所合理性を与える演算子であり，以下のインターフェースを持つ．

```
(defoperator IN-PLACE (SPOT))
```

場所に関する変数または定数シンボルを引数としてとり，シンボルの表す特定の場所において可能な定位置操作に関する制約を表す．制約として与えられる定位置操作の関係は第 5.3 節によって得られた知識を用いる．引数のシンボルは他の演算子により具体化されたものである必要がある．

6.3.3 PREFER 演算子

PREFER 演算子は人の物に関する局所合理性を与える演算子であり、以下のインターフェースを持つ。

```
(defoperator PREFER (CATEGORY PERSON))
```

物のカテゴリや人に関する変数または定数シンボルを引数としてとり、人の物に対する嗜好性を制約として加える。物のカテゴリは定数シンボルであり、人は変数または定数シンボルである必要がある。

6.3.4 状況レベルのタスク記述例

シンボリックな状態記述に対して以上の具体化演算子、変数を導入したタスク記述を行う例を以下に示す。以下の記述は TABLE の上にある物をすべて普段ある場所に片付けるタスクを環境に非依存な形で表したものである。?OBJ は初期状態において TABLE の上にある物の認識から具体的な定数シンボルへと置換される。次に (USUAL ?OBJ ?SPOT) において、定数シンボルとなった?OBJ から確率的物体配置地図を用いて?SPOT が具体化されることで、変数の解決された具体的なタスク記述を得る。

```
(: INIT
  (ROBOT-AT START)
  (ON ?OBJ TABLE))
(: GOAL
  (USUAL ?OBJ ?SPOT)
  (ON ?OBJ ?SPOT))
```

次節では具体化手法について説明していく。

6.4 定位置操作モデルを用いた場所の局所合理性に基づく具体化

日常生活におけるタスクを行うときにはロボットの台車移動と物体や環境のマニピュレーションが含まれる。例えば冷蔵庫から飲み物を取ることを考える。単純な場所と物の関係のみを考えると、このタスクを満たす動作は冷蔵庫へ移動し、飲み物を把持し、目的の場所へ移動し、飲み物を置くことになる。しかし、飲み物をロボットが把持するためには冷蔵庫に付いている扉を開ける必要がある。冷蔵庫の扉の開閉のような操作は物を把持する動作を行う場所が冷蔵庫の前であったときに発生する特定の場所における付随動作である。このような特定の場

所に関連付けられた操作を行う必要性に関する知識を状況レベルタスク記述に対して付加することが具体的タスク記述を得るために必要とされる．そのような特定の位置における操作や状態変化に関する知識を具体化する手法として，IN-PLACE 演算子をもとに初期状態に場所に対する状態記述を追加する．

例えば冷蔵庫の前を表す FRIDGE における定位置操作としてものを置くことが可能であることを示す演算子 PLACABLE と扉を開けることが可能であることを示す演算子 OPENABLE が得られた時，(IN-PLACE FRIDGE) によって，場所シンボル FRIDGE に対して，タスクの初期状態に (PLACABLE FRIDGE) (OPENABLE FRIDGE) のように可能な定位置操作を表す関係式を加えるような制約が与えられる．

OPENABLE などの状態述語によってスポットに対応付けられた定位置操作物体が環境操作によって遷移する複数の状態を持つことがタスク記述として定義される．OPEN などの操作述語によってタスク実行中のロボットの物体操作前後における操作物体の状態が記述される．以上の2つの述語によってシンボリックな状態記述において状態遷移が表現されているが，実際にロボットが該当する状態遷移のために環境に対して行う操作も環境固有の知識として獲得する必要がある．第 5.3.2 節において環境における人と定位置操作物体との相互作用の観察の結果得られる物体操作モデルをシンボリックな状態記述における操作述語と対応付いた手続き知識としてこの状態遷移に必要な操作として用いることで，本研究ではシンボリックな状態記述の具体化のみでなく，環境の状態遷移に必要な適切な環境物体操作を行うためにロボットが必要とするパラメタについても具体化される．

6.5 嗜好モデルを用いた人と物の局所合理性に基づく具体化

(PREFER ?CATEGORY ?PERSON) のように関係式が与えられるとき，第 5.5 節におけるタスク指示をする人を表す User ノードの確率変数に?PERSON の値，指示に含まれる対象物のカテゴリを表すノード Category の確率変数に?CATEGORY の値をそれぞれエビデンスとして与えた時に対象物を表すノード Item の確率変数に関する最大事後確率推定を行う．

例えば (PREFER COFFEE FURUTA) のような関係が与えられた時，COFFEE は事後確率の最大値を取る物 GEORGIA へと置換される．

6.6 確率物体配置地図を用いた物・場所の局所合理性に基づく具体化

この解決器は定位置操作モデルを用いた具体化演算子を含む関係式の具体化が行われたあとの記述を用いる．初期状態および目標状態記述において，解決済みの物体シンボルに関して，第 5.4 節において確率的物体配置地図として経験的に得られた物体の存在確率分布を用いて，シンボルの指す物体が通常存在している場所を表すシンボルを解決する．

確率的物体配置地図において，各物体の存在の信念は以下のように与えられている．

$$p(\mathbf{x}) = \sum_i^n w_i \mathcal{N}(\mathbf{x} | \mu_i, \Sigma_i) \quad (6.1)$$

ここで μ_i は各クラスをなす GMM の平均, Σ_i は共分散, w_i は各クラスの重み, n はクラス数数をそれぞれ示している.

MVGMM の各モデルの次元は 3 次元となり, これを環境上の固定点を座標系とする 3 次元空間上に表すことで確率的物体配置地図が得られる.

このときクラスタリングされた各 GMM の平均 μ_i は対象クラスの物体が普段配置される確率が高い地点を示していることに着目し, GMM の平均値を目標状態における物体配置候補点として利用する. クラスタ数 n が 2 以上であるとき, 目標状態は各クラスの平均からなる地点に対象物体が配置される状態の論理和によって定義される. また各クラスにおける重みを行動計画に反映するため, 本研究では PDDL2.1 において追加された numeric fluents という状態遷移についてコスト関数を設定可能な機能を用いて行動計画の最適化を行う.

$$cost_{obj}(\mathbf{x}) = floor(k(1.0 - p(\mathbf{x}))) \quad (6.2)$$

これまでの確率的物体配置地図を利用したタスクの記述はジオメトリックな情報のみが扱われていた. しかし, 現実環境において物体を配置するためには対象物体だけでなく環境の状態を必要がある場合がある. 例えば冷蔵庫の中に物体を配置するためには冷蔵庫の扉の開閉といった操作を行う必要がある.

この問題に対処するために, 提案システムにおいては定位置操作モデルとの統合を行う. 定位置操作モデルには操作対象物体の操作可能構造や操作対象物体の幾何的な包含領域 (Bounding Box) が含まれる. 幾何的な位置情報に対して, 定位置操作モデルを参照することで解決対象物体と定位置操作モデルの包含関係を考慮し, 単純な接触関係と区別された記述として解決される. 実際の行動計画時における計画の最適性評価には, 式 6.2 によって算出されるコスト関数の他にも, ロボットの場所移動に伴う所要時間を表すコストや環境操作における時間, 失敗リスクを表すコストなどが用いられ, タスク全体でのコスト最小化が目標として与えられる.

6.7 状況レベルタスク記述の具体化に基づくタスク計画実行 実験

本節では本章において扱ってきた日常生活環境におけるタスクの抽象的行動記述と, ロボットが環境において経験的に得られる知識を用いて抽象的行動記述を行動計画器が計画可能な具体的行動記述に展開する機能について, 「片付けをする」タスクを題材に実ロボットを用いた実験を通して評価する.

6.2.1 における片付けタスクの状態記述表現は以下のようであった.

(: INIT (AND (AT TABLESIDE)

```
(ON BOTTLE TABLESIDE))
:GOAL (ON BOTTLE KITCHEN))
```

この表現は以下の点が課題としてあげられる。

- 場所 TABLESIDE, KITCHEN や片付けを行う物体 BOTTLE が固定的に書き下されているため、他の物体や場所に対応できない。また複数の物体を扱うことができない。
- ロボットの初期位置が TABLESIDE に固定されているため、汎用性がない。
- BOTTLE の初期位置が TABLESIDE に固定されているため、汎用性がない。
- BOTTLE が片付けられるべき位置が KITCHEN として事前に定義されている。BOTTLE が KITCHEN に常に片付けられるべきであるとは限らない。

一方で、本章において説明された状況レベルタスク記述を用いた片付けタスクは以下のように定義される。

```
(:INIT (ROBOT-AT ?SPOT)
      (IN-PLACE ?SPOT)
      (ON ?OBJ ?SPOT))
(:GOAL (USUAL ?OBJ ?DEST)
      (IN-PLACE ?DEST)
      (ON ?OBJ ?DEST))
```

この記述は初期状態において、ロボットが現在いる場所?SPOT にあるもの?OBJ について、目標状態では?OBJ がいつもある場所?DEST にあるべきである、と表現する記述である。ただし、この記述は行動計画器が理解することができない変数が含まれるため、経験的知識を用いて具体化していく。

はじめにタスク実行状況に依存する変数を置換する。ここではロボットはテーブルにおいてペットボトルがテーブルの上にあることを認識したことから、?SPOT は TABLE, ?OBJ が BOTTLE へと具体化される。

```
(:INIT (ROBOT-AT TABLE)
      (IN-PLACE TABLE)
      (ON BOTTLE TABLE))
(:GOAL (USUAL BOTTLE ?DEST)
      (IN-PLACE ?DEST)
      (ON BOTTLE ?DEST))
```

目標状態における具体化演算子 USUAL を含む関係式に注目し、?DEST を経験的に得られる BOTTLE の配置される場所シンボルで置換する。確率的物体配置地図において BOTTLE シ

ンボルに対応するカテゴリ `bottle` のクラスタ数は3であり、それぞれは`?DEST0`、`?DEST1`、`?DEST2`とシンボライズされる。またクラスタごとの重みパラメタ w から算出されるコストを初期状態において定義する。各シンボルのコストはそれぞれ第6.6節における手法により7,2,1となった。

```
(:INIT (ROBOT-AT TABLE)
      (IN-PLACE TABLE)
      (ON BOTTLE TABLE)
      (= (OBJ-COST BOTTLE ?DEST0) 30)
      (= (OBJ-COST BOTTLE ?DEST1) 80)
      (= (OBJ-COST BOTTLE ?DEST2) 90))

(:GOAL (IN-PLACE ?DEST0)
      (IN-PLACE ?DEST1)
      (IN-PLACE ?DEST2)
      (OR (ON BOTTLE ?DEST0)
          (ON BOTTLE ?DEST1)
          (ON BOTTLE ?DEST2)))
```

次に確率的物体配置地図の各クラスタに対応するシンボル`?DEST0`、`?DEST1`、`?DEST2`について、その幾何的な位置情報と経験から獲得した定位置物体操作モデルを用いて具体的なシンボルに置換する。

```
(:INIT (ROBOT-AT TABLE)
      (IN-PLACE TABLE)
      (ON BOTTLE TABLE)
      (= (OBJ-COST BOTTLE FRIDGE) 30)
      (= (OBJ-COST BOTTLE COUNTER) 80)
      (= (OBJ-COST BOTTLE SINK) 90))

(:GOAL (IN-PLACE FRIDGE)
      (IN-PLACE COUNTER)
      (IN-PLACE SINK)
      (OR (ON BOTTLE FRIDGE)
          (ON BOTTLE COUNTER)
          (ON BOTTLE SINK)))
```

最後に `IN-PLACE` を定位置操作モデルをもとに置換する。このとき定位置物体操作モデルの幾何情報・操作物体状態情報を元に物体の包含関係が反映される。

```
(: INIT (ROBOT-AT TABLE)
        (PLACABLE TABLE)
        (PLACABLE FRIDGE)
        (OPENABLE FRIDGE)
        (PLACABLE COUNTER)
        (PLACABLE SINK)
        (ON BOTTLE TABLE)
        (= (OBJ-COST BOTTLE FRIDGE) 30)
        (= (OBJ-COST BOTTLE COUNTER) 80)
        (= (OBJ-COST BOTTLE SINK) 90))
(: GOAL (OR (ON BOTTLE FRIDGE)
            (ON BOTTLE COUNTER)
            (ON BOTTLE SINK)))
```

以上ですべての変数が具体的なシンボルに置換された記述が得られた．ここではコスト関数 $cost_{obj}$ について，係数 k を実験的に 100 とした．本実験では行動計画における動作列の評価指標として用いられるコスト関数として，OBJ-COST に加えて，表 6.1 にあるコスト関数を用いている．

表 6.1: 行動計画時のコスト関数

コスト関数	説明
OBJ-COST $?obj ?place$	物体配置信念を表すコスト
MOVE-COST $?from ?to$	台車移動時に加算されるコスト
ARM-COST $?arm$	単腕・双腕を用いた動作時に加算されるコスト
FIND-COST	物体認識時に加算されるコスト
MANIP-COST	定位置物体操作時に加算されるコスト

MOVE-COST コスト関数は (MOVE-COST ?FROM ?TO) のように移動動作における始点と終点を表す 2 つの場所シンボルを引数に取る関数として定義される．コスト値はシンボル値が表す場所間の実際の距離に定数を乗算した数値を用いた．ARM-COST コスト関数は物体把持やテーブルなどへの物体を置く動作に際して単腕のみを使用する場合と双腕を使用する場合でかかる失敗リスクと時間の変化を表す．ロボットの腕を表すシンボルは左腕を表す $larm$ ，右腕を表す $rarm$ ，双腕を表す $arms$ があるが， $larm, rarm$ については定数 10， $arms$ に関しては 30 を実験的に決定して与えた．FIND-COST は物体認識や物体を置く動作において支持平面を認識によって探索するときにかかる時間的コストを表す．本実験では定数値 50 を実験的に与えた．MANIP-COST コスト関数は環境における物体操作を行うことによる失敗リスクを表すコスト関

数で、本実験では定数 10 をコスト値として与えた。

以上を踏まえて実際に行動計画器に与えられる状態記述を含む問題記述は以下のように与えられる。なお本実験に用いるドメイン記述は第 A.1 節に記載した。

図 6.2 は行動計画器における出力である動作遷移列を、動作をエッジ、動作前後の状態をノードとしたグラフに表したものである。ドメイン記述にある失敗動作については find-object に対して find-object_f のように末尾に_f 記号をつけたもので、同一のグラフに統一されている。

このグラフを動作をノード、状態遷移をエッジとするグラフへ変換したものを図 6.3 に示す。このグラフは失敗と復帰を考慮した動作制御ステートマシンとなり、ロボットはこのステートマシンに沿って行動目標を達成していく。

1. 図 6.4a: ロボットがテーブルの上にある片付け対象物である、ペットボトルの位置を認識する
2. 図 6.4b: 動作計画を行いペットボトルを把持する
3. 図 6.4c: ペットボトルの片付け目標場所である冷蔵庫へ移動する
4. 図 6.4d: 冷蔵庫のドアを開ける
5. 図 6.4e: 冷蔵庫内においてペットボトルを置くことが可能な支持平面を認識し、認識した平面上にペットボトルを置く
6. 図 6.4f: 冷蔵庫の扉を閉める

6.8 おわりに

本章では、事前に与えられる知識のみで記述することが困難な日常生活行動を、ロボットが環境におけるタスク実行を通じて得た環境や共生する人間に関する局所合理性を含む経験的知識を用いて、環境に即した具体的なタスク記述に展開可能な状況レベルのタスク記述と具体化機能について説明した。固定的に書き下されてきた操作対象アイテムやスポットシンボルを変数を用いて抽象的に記述し、タスクの実行指示が与えられた時に経験や認識によって得られる知識を用いて変数を具体的なシンボルへ置換する機能を持つ具体化手法を適用することで、行動計画器に入力可能な具体的なタスク記述を得る仕組みについて説明した。また、環境において長期経験を持つ実ロボットに本機能を搭載して片付けタスクを行い、本機能が、予め与えられた知識に加えて、環境に長期的にいることによって得られる経験的知識を利用し環境に適応した行動を計画・実行可能であることを示した。

図 6.1: 片付けタスクの問題記述

```

(define (problem tidyup)
  (:domain room-domain)
  (:objects (counter kitchen ... fridge - spot)
            (bottle - item)
            (larm rarm - arm))
  (:init
   ;; cost functions
   (= (find-cost) 50)
   (= (manip-cost) 10)
   (= (total-cost) 0)
   (= (move-cost fridge sink) 3)
   (= (move-cost fridge drawer) 30)
   ...
   (= (move-cost counter fridge) 30)
   (= (obj-cost bottle fridge) 30)
   (= (obj-cost bottle counter) 90)
   ...
   (= (obj-cost bottle sink) 80)
   (= (arm-cost rarm) 10)
   (= (arm-cost larm) 10)
   (= (arm-cost arms) 30)
   ;; initial states
   (robot-at table)
   (on bottle table)
   (openable dishwasher)
   (placable dishwasher)
   (openable fridge)
   (placable fridge)
   (placable sink)
   (placable counter)
   (placable kitchen)
   (placable table)
   (placable bookshelf)
   (placable drawer)))
  (:goal
   (and
    (or (on bottle fridge) (on bottle counter) (on bottle sink))
    (forall (?at - spot) (not (open ?at)))))
  (:metric minimize (total-cost)))

```

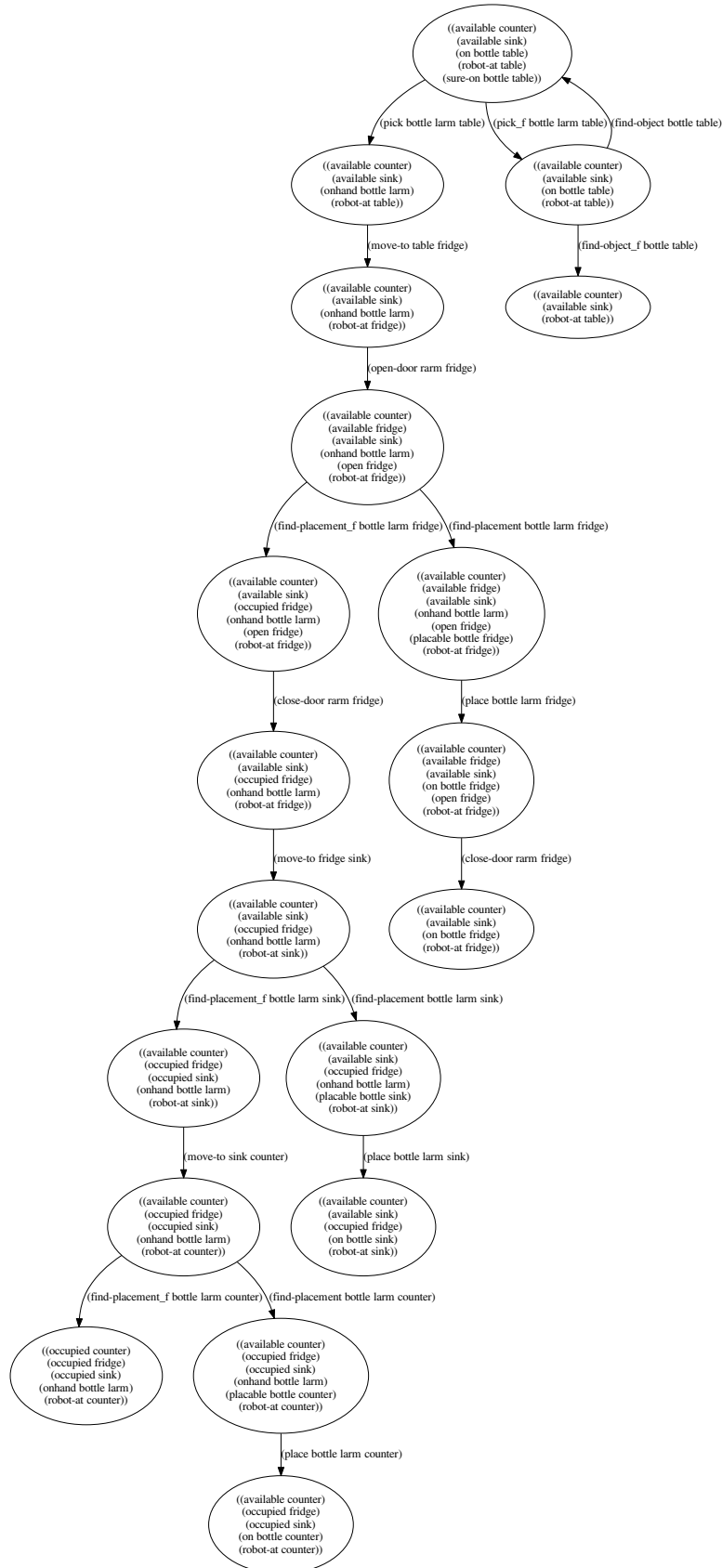


図 6.2: 片付けタスクの行動計画

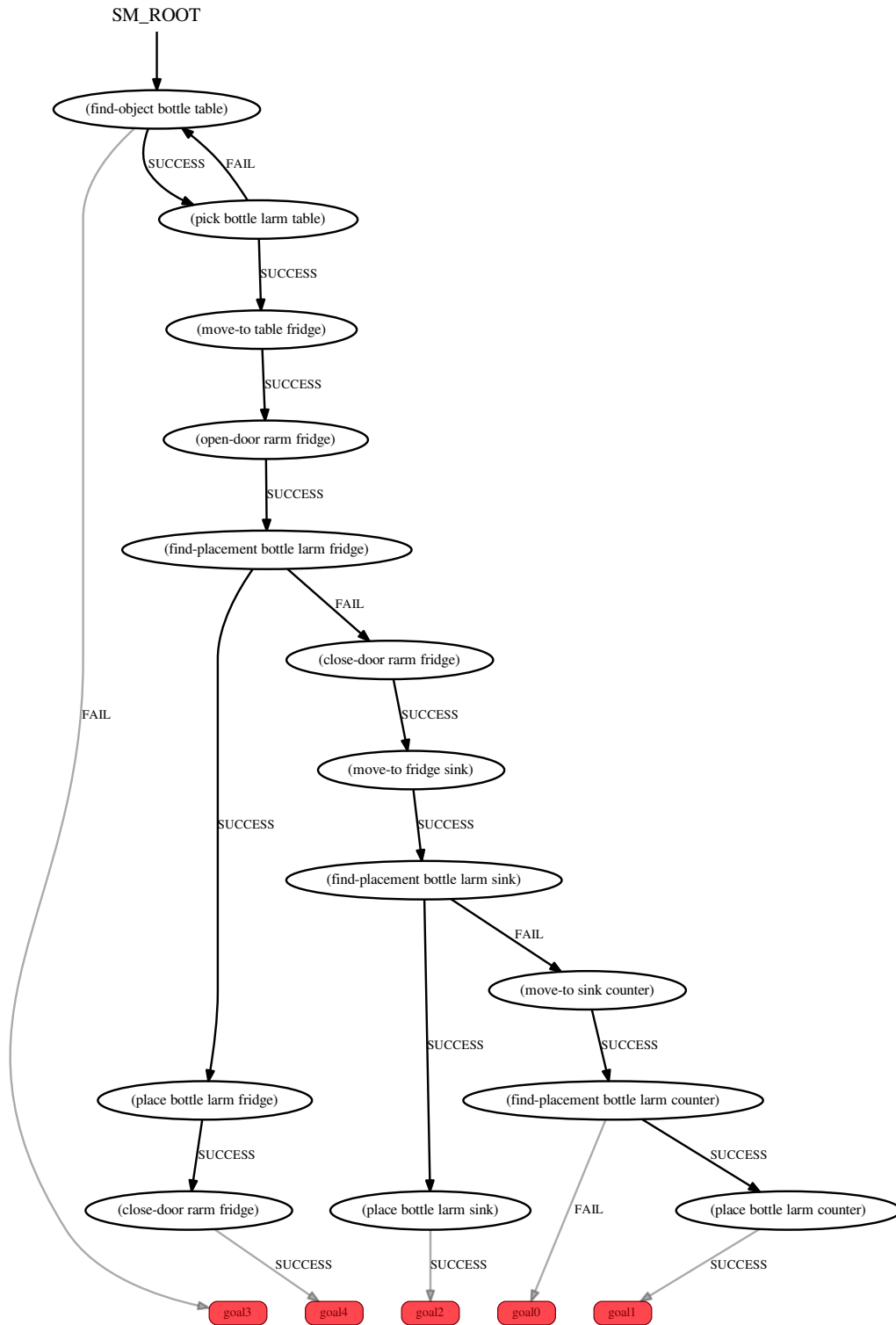


図 6.3: task_compiler による失敗復帰可能な動作制御ステートマシン



(a) ペットボトルの認識



(b) ペットボトルの把持



(c) 冷蔵庫への台車移動



(d) 冷蔵庫ドアハンドルを把持し、扉を開ける



(e) 冷蔵庫内の物体支持平面を認識し、冷蔵庫内にペットボトルを置く



(f) 冷蔵庫の扉を閉める

図 6.4: 片付けタスクの実験 ([126] より引用)

第7章

曖昧な指示・修正を随時受け付け対応可能な日常生活タスク実行システム

7.1 はじめに

本研究では日常生活環境において、自律的にタスクを遂行するロボットシステムの構築を目的とするが、一方で日常生活環境の特色として、ロボットはユーザとなる人間と同じ環境において共生することが前提となる。そのためロボットは人間の活動を妨げないように常に共生するユーザに気を配り、指示を受け付け対応するような機能が必要となる。また各環境において生活を営む人間は日常生活タスクに関する経験や知識を十分に持つエキスパートとみなすことができる。したがって、インタラクションを行うことによってロボットシステムにとって環境を観察する以外の経験を獲得する手段として活用されることが期待される他、ロボットが自律的にタスクを実行している最中に起こりうる潜在的な失敗に関して人間からの指摘を随時受け入れるような機能を備えることで、失敗を未然に防ぎ、動作を修正することが可能となる。

本章ではロボットが人間とのインタラクションを行いながら自律的にタスクを計画実行していくシステムを構築する上で必要な要素を検討し、その構成法について述べる。

ロボットに対して人間が指示を伝達する手法として次のようなものが挙げられる。

プログラムによる指示 ロボットに搭載された計算機に対して、タスクを行うプログラム名やパラメタを直接指定することで指示する方式である [127]。指示を送る段階で、指示を送るロボットとタスク遂行に必要な構成要素を指定することで、ロボットエージェントと人間の間で指示の誤解が発生する余地はないため、最も正確に指示を伝達する手段であるが、ユーザである人間がロボットシステムの内部構造に十分詳しい専門的知識を持ち、ロボットシステムに対してプログラム名やパラメタを指示するためのインターフェースを用意する必要がある。

GUIによる指示 プログラムによる指示と形式的には同様であるが、ロボットシステムの開発者が事前にロボットに対する専門的知識を有しないユーザのために簡易な指示のためのグラフィカルインターフェースを用意する方式である [91]。プログラムによる指示に対

して、指示可能なパラメタの自由度は小さいが、ロボットシステムの提供する主要な機能をすばやく使用可能にすることができる。一方でロボットとの指示の度にインターフェースを備えるデバイスをユーザが使用する必要がある。

音声による指示 音声によってユーザである人間が対象のロボットに対して指示を送る方式である [128]。ユーザが特別な道具を必要とせず、人間同士の意思疎通と同じ手段で指示を行うため、ユーザに対して要求する条件が最も少ない一方で、音声による指示は時間的に一方的な意思伝達であるため、長いパラメタを含む指示を伝達することは難しい。したがって、人間同士の音声対話による意思伝達においても参加者全員が共通する知識を持っている前提において、伝達内容が適宜省略あるいは換言される。このときロボットエージェントは人間が共有を仮定している知識がエージェントの持つ知識に存在しているか否かを確認する必要がある。さらに、ユーザ・ロボットエージェント双方が意味的に知識を共有していたとしても、音声により伝達する際に用いられる記号が一致していないときに、エージェントは適切に記号を変換したうえで解釈を行わなければならない。

直接接触による指示 ロボットのアームなどに対して人が外力を加えることなどにより、ロボットを人が直接操作することで指示を送る方式である [33, 129]。実環境においてロボットに目的とする指示を直接的に指示可能であり、人にとって専門的な知識や正確な数値などを扱うことなくロボットへ指示を送ることが可能である。タスクの目標状態など、複雑な概念を伝えることよりも動作の目標手先位置など、指示する内容が限定される。

本章では日常生活環境において随時指示を送ることを前提としているため、人にインターフェースを必要とせず、指示を送るための予備動作時間を必要としない、音声による指示と直接操作による指示を題材に、前述した曖昧性やタスク遂行中においてロボットがユーザの随時的な指示を受けたときにも対応するロボットシステムの構成について述べていく。

7.2 関連研究

7.2.1 音声によるタスク指示の曖昧性

タスク指示の曖昧性解決システム

このような対話の参加者が理解を前提としている知識は文脈と呼ばれる。音声対話において文脈がどの程度利用されるかは単に相手との関係性のみならず、対話に使用される言語 [130] など多様な要素によって決定される。人間とロボットエージェントが対話を行う上で、このような文脈が含まれる情報が伝達されるとき、前提として共有されていない知識を含む曖昧な情報をエージェントは理解することができない。

この課題に対して考えられるアプローチの1つとして、文脈を用いない対話を用いる方法が考えられる。対話を行うエージェントが理解することができる指示や用語のリストを決めておき、ユーザはリストにある用語のみを用いて対話を行うというものである。このシステムは、

ロボットが行うべきタスクが、手順が少なく、単純であるときに有効で、ロボットシステムとして文脈を理解する必要がない一方で、日常生活タスクのような手順が複雑で、反復的である行動、すなわち文脈を活用することで伝達すべき情報量が小さくなりやすいタスクに対しては、かえって煩雑な指示となってしまう。

Rybski らは、曖昧な自然言語指示を入力として、ロボットエージェントが現在の環境状態に応じて選択可能な動作を手がかりに記号設置を推定し、行動指示を行う仕組みを提案している [131]。Thomason らは確率グラフィカルモデルを用いて、対話について情報からロボットが要求されたタスクの確信度を推定し、対話数が少なくても確信度が大きくなる質問を行う手法を提案している [132]。これらは事前に決まったドメインにおける命令セットと環境知識のみが使用可能であり、ロボットに事前に与えられた知識にない文法、単語は扱うことができなかった。

音声認識機能

タスクの指示は音声を用いた自然言語記述によって行われるため、指示の解釈を行うためにロボットエージェントはまず音声を入力として自然言語を認識する機能を備える必要がある。音声認識のための代表的なフレームワークとして Julius[133] や Sphinx[134] が挙げられる。これらのフレームワークでは、音声から音素と呼ばれる音声の最小文節単位への変換である音響モデルと音素列と単語の対応表、すなわち語彙を与えることで、ある音声区間に対して語彙に含まれる単語列を推定する。事前に与えられた語彙に関して最も確からしい認識を行うため、レストランにおける自動応対など限定されたドメインにおける利用に適している一方で、日常生活支援ロボットシステムのように環境が動的で語彙を限定されない環境下においては、適さない。一方で、最近では Google 社を始め複数のサーバクライアントモデルを採用した大規模な語彙モデルを用いた音声認識サービスが利用可能である [135]。このようなサービスは語彙が大規模であり、あらゆる話題における自然言語を認識可能である。

一方で、似た発音を持つ単語・単語列への誤認識に対して頑強である必要がある。また認識した自然言語の語彙がロボットエージェントの知識にとって未知である場合に適切に対応を行わなければならない。

7.2.2 割り込み可能なロボットシステム

一般に行動計画器は初期状態と目標状態、状態遷移オペレータが与えられた時に初期状態から目標状態へ至る動作遷移列を解くもので、初期状態、目標状態は、それらの情報のみで行動を計画可能となるような十分かつ完全に記述を行う必要があった。しかし、完全な環境の状態を計画時点で与えなければならないこの前提は、特に環境が動的に変化する日常生活環境においては適用が難しかった。一方で、はじめにプランをすべて生成せず、現在の状況に応じて目標状態を達成するための動作を逐次計画、実行していく、行動計画と動作制御を組み合わせた手法としてオンラインプランニングが研究されている。割り込みを考慮したロボットシステムでは、割り込みを受けた際に割り込みタスクが実現可能であるかを検証する必要がある他、割

り込みタスクを実行することで、割り込まれたタスクのプランに影響が生じる可能性についても考慮する必要がある。

カーネギーメロン大学において開発された移動ロボット Xavier に搭載された行動計画器 ROGUE[136] では、タスクの実行中に割り込みタスクが新たに指示されるとき、割り込まれるタスクの目標状態と割り込みタスクの目標状態の両方を満たす新たな目標状態を作成し、新しいタスクとして実行する。このとき目標状態に衝突が生じる可能性があるが、ROGUE ではタスクごとに優先度を設定することによって優先度が高いタスクの目標状態が優先されることによって解決している。この方法では割り込みタスクと割り込まれるタスクの目標達成に対する優先度は考慮されるものの、新しいタスクの目標状態において、既存の割り込みタスクや割り込まれるタスクの達成状況が考慮されていない。

林らは階層的行動計画手法の一つである HTN(Hierarchical Task Network) を拡張し、緊急割り込みに対応した行動計画器を提案している [137]。林らの手法では割り込みタスクの目標状態を満たす計画を先に実行し、その過程で起こる状態変化を監視し、割り込まれるタスクの状態を変更する。この方法では割り込みタスクを緊急度の高いタスクであるという前提としているため、割り込まれるタスクの達成状況にかかわらず割り込みタスクが行われる。

またこれらの手法は行動計画をオンラインで変更する手法を提案しているもので、割り込みによってどのように変更を人間がロボットに対して指示するかについては扱われていなかった。

7.3 曖昧性解消を行う音声対話を用いた自然言語によるタスク指示機能

本節では大規模な語彙データベースを用いる音声認識器をもとに得られる自然言語に対して、省略によって生じる曖昧性や誤認識、欠落による不完全性、未知の知識を考慮した自然言語解釈を行う機能について述べる。

本研究における音声指示理解の流れをアルゴリズム 7 に示す。

音声信号から自然言語を認識するためには音声信号の時系列的なデータから人の発生する連続区間を切り出し (Voice Activity Detection, VAD), 音声認識器へ入力し, 認識を行う (Speech Recognition, SR)。VAD としては W3C によって標準化が進められている WebRTC プロトコルにおいて実装されているものを使用した。また SR については Google 社のサービスとして提供されている Google Speech Recognition を利用した。

7.3.1 メンタルレベルにおける曖昧性の解消

まずはじめに、指示内容について予備動作など本来必要な前提知識を省略することに関する曖昧性を解消する仕組みについて説明する。指示可能な動作のリストとして表 7.1 を定義する。各動作は動作指示を表す自然言語上の単語列 (Synsets)、動作を行うために満たされるべき前提条件 (Precondition)、動作を行うことによる状態の変化 (Effects) を STRIPS 型のシン

Algorithm 7 音声による自然言語指示の理解手法

Input: *Audio* : Input audio data
Input: *AS* : Audio Segments
Input: *ST* : Statements
Input: *CMD* : Parsed Instructions

```

1: AS ← vad(Audio)
2: for all as ∈ AS do
3:   ST ← speech_recognition(as)
4:   for all st ∈ ST do
5:     found, act ← find_action(inst)
6:     if found then
7:       args ← find_arguments(st, act)
8:       if ¬arg_satisfied(act, args) then
9:         cand, likelihood ← disambiguate(st)
10:        if likelihood < threshold then
11:          cand ← ask(act, cand)
12:        end if
13:        args ← merge(args, cand)
14:      end if
15:      CMD ← (act, args)
16:    end if
17:  end for
18: end for
19: CMD ← reason(CMD)

```

ボリックな状態記述によって定義されている．各動作記述のシンボルは一階述語論理に従い，次のような型情報を持つ．*?loc* は空間上の地点を表すシンボル，*?obj* は物体を表すシンボル，*?ppl* は人を表すシンボルを意味する．指示内容に基づき動作が選択されると動作の影響が指示の目標状態として与えられる．指示の目標状態を満たす状態へ遷移する動作列を閉世界仮説に基づき計画することで，指示内容が実行可能であるか，もし可能でない場合に必要となる予備動作を補完することができる．

7.3.2 言語レベルにおける曖昧性解消

音声認識によって得られた自然言語の各指示文に対して，言語理解 (Language Understanding) を行う．言語理解では例文と文中の単語のラベルを与えることで，未知の文における単語間の関係性から各単語のラベルを推測する問題である．本研究では言語理解のための機能として Microsoft の LUIS[139] を用いた．

表 7.1: ロボットに指示可能な動作リスト ([138] を改変)

Action	Description	Synsets	Precondition	Effect
move-to <i>?to</i>	Move robot base to the location	navigate, enter	(not (at <i>?to</i>))	(at <i>?to</i>)
find-object <i>?obj ?loc</i>	Find object near the location	find, locate, look for	(not (on <i>?obj ?loc</i>)) (at <i>?loc</i>)	(on <i>?obj ?loc</i>)
find-person <i>?ppl ?loc</i>	Find person near the location	find, look for	(not (see <i>?ppl ?loc</i>)) (at <i>?loc</i>)	(see <i>?ppl ?loc</i>)
ask <i>?ppl ?loc</i>	Ask the person at the location	answer, tell, say	(not (asked <i>?ppl</i>)) (see <i>?ppl ?loc</i>) (at <i>?loc</i>)	(asked <i>?ppl</i>)
pick <i>?obj ?loc</i>	Grasp the object	take, get, pick up, grasp	(on <i>?obj ?loc</i>) (at <i>?loc</i>)	(onhand <i>?obj</i>)
place <i>?obj ?loc</i>	Put the object	put	(onhand <i>?obj</i>) (at <i>?loc</i>)	(on <i>?obj ?loc</i>)
deliver <i>?obj ?ppl</i>	Deliver the object to the person	hand, give, bring	(onhand <i>?obj</i>) (see <i>?ppl ?loc</i>) (at <i>?loc</i>)	(delivered <i>?obj ?ppl</i>)

言語理解を行うことにより文章に対して、対応する動作シンボルを表 7.1 を参照することで得る (*find_action*)。次に動作シンボルに対して要求されるパラメタを、各パラメタの型情報をもとに指示文から解釈する (*find_arguments*)。ここで動作シンボルに対して、実行するために必要なすべてのパラメタを解釈できた場合は次の指示文へと処理を続けていく。

7.3.3 音響レベルにおける曖昧性解消

言語レベルにおける曖昧性解消時に不足するパラメタがある場合は以下のように分類できる。

- 不足するパラメタの型に該当するシンボルがエージェントの知識として存在するが、誤ったシンボルとして解釈されている場合
- 解釈されたシンボルがエージェントの知識に存在せず、未知の概念を意味する場合

前者の場合は自然言語において類似した発音のシンボルとして誤認識された場合である。このような場合は不足するパラメタについて、ロボットエージェントが知識として既知のシンボルに正しく対応付ける必要がある。本研究ではまず指示文のうち不足するパラメタ以外の解釈に用いられた部分を除く部分に対して、音素列へ変換する。自然言語から音素列への変換は Grapheme-to-Phoneme 変換として知られている。本研究では Transformer モデル [140] を利用した深層学習ベースの変換器を利用した。自然言語を変換して得られる音素の例を表 7.2 に示す。

得られた音素列とロボットエージェントの既知のシンボルに対応する音素列のリスト

表 7.2: 単語の音素 ([138] から引用)

Symbol	Phonemes	Symbol	Phonemes
living room	L IH1 V IH0	green tea	G R IY1 N
	NG R UW1		T IY1
	M		
kitchen	K IH1 CH	chewing gum	CH UW1
	AH0 N		IH0 NG G
fridge	F R IH1 JH		hair spray
sofa	S OW1 F	manju	HH EH1 R
	AH0		S P R EY1
entrance	EH1 N T R		M AE1 N
	AH0 N S	JH UW1	

(a) 場所を表す単語

(b) 物品を表す単語

に対して類似度を計算していく。各音素について Term Frequency Inverse Document Frequency (TF-IDF) 特徴量を算出する。既存知識シンボルに対応する各音素列に対して TF-IDF 特徴量が要素となる特徴ベクトルが得られると、ベクトルのリストを教師なしクラス識別器の一つである、ランダムフォレストを用いてモデルの学習を行う。ここで各シンボルを 1 つのクラスとみなす。得られた学習済みモデルを用いて、指示分の未解釈部分の音素列より得られる特徴量とのクラス識別を行う。学習済みモデルの各クラスについて計算される類似度が最も大きく、かつ閾値を超えたものについては、類似する既存知識をパラメタとして採用する。

類似度が閾値を下回ったときは不足パラメタをロボットエージェントの知識にとって未知のシンボルとみなす。

7.3.4 世界レベルの曖昧性の解消

未知概念として判定されたシンボルは、ロボットシステムにとって情報を持たない概念であるため、このままではタスクを実行することができない。そこで、ユーザとの対話を通じてタスクを行うために必要な情報である未知シンボルの性質を明らかにする。追加で得られた性質を未知シンボルと関連付けることによって、タスク実行中の認識を行う。

最後に自然言語における指示をすべて行った状態に対して、ロボットエージェントが置かれた現在の状態からの遷移を行動計画器へ入力することで、音声指示の中に現れた動作シンボルのみではなく、各動作シンボルを含み全体のタスクを実行可能となるために必要な予備動作とパラメタを推論し、行動全体の動作列を得る。

7.3.5 未知概念物体の認識手法

本節ではロボットがタスクを行う上で必要な物体認識について、ロボットシステムがすでに知識としてモデルを与えられた物体、すなわち既知物体と、ロボットがモデルを予め与えられていない未知物体について統一的に扱うためのシステムについて述べる。

物体の認識を行う際に必要となる認識器は、センサ入力の特徴だけでなく、認識を行う環境によっても使い分けが必要となる。例としてシャンプーのボトルを認識するタスクを考える。ロボットシステムの知識として特定のシャンプーの物体モデルを持っている場合はSSD(Single-shot multibox Detector)[60]などに代表される物体検出を行う深層学習モデルを利用してセンサ空間中において認識対象物体をエンドツーエンドに求めることができる。しかし、シャンプーボトルをロボットが知識として持っていないとき、ロボットエージェントはシャンプーボトルの色や形状など、ロボットシステムが理解可能な尺度をもつ部分的な性質をもとに認識対象物体を推論していくことになる。例えばシャンプーと石鹸がセンサ入力として与えられたときには、ロボットは背が高い物体という尺度をもとにシャンプーを推測できるが、シャンプーとリンスのボトル、すなわちどちらも背が高い物体が並んでいるときには2つのものを区別可能な別の尺度が必要とされる。このようにある物体を認識しようとするときに認識対象の物体だけでなく、認識を行う周囲の環境も影響を与えうるため、事前にロボットが必要となる認識器を決定することは難しい。岡田らは複数の認識器と粒子フィルタを組み合わせ、認識物体のもっともらしさを各認識器における尤度の積として表すことで、複数の並列に動作する認識器の結果を統合的に評価する方法を提案した [57]。この手法では物体認識結果の確率的な出力が各認識器の尤度の単純な積で表されるため、認識対象の物体に対して各認識器において期待される出力が完全に得られない場合には認識を行うことができなかった。

本研究では物体の性質を出力とする認識器を並列に動作することにより、個々の認識物体ごとに複数の指標による識別結果を得る。物体の未知概念に対して追加で人と対話を行うことにより得られた物体の性質の組に対して、物体ごとの認識結果の組との一致数が最も多いものを対象候補物体とする。各検出器における結果が集約され、対象候補物体が1つになるとき、その候補物体を対象物体とする。

本研究では検出器として色ヒストグラム、基本形状推定、容積を用いた。物体領域の抽出は3次元点群の入力に対して平面検出を行い、平面上の3次元点群をユークリッド距離を用いてクラスタリングすることで得られる。平面検出にはRANSACを用いる方法がよく使用されている [141]。

色ヒストグラムを用いた色検出器

各物体の3次元点群情報から色情報を抽出し、HLS色空間に変換する。HLS色空間は色相(Hue)、輝度(Luminance)、彩度(Saturation)の成分からなる。色相は360度の角度によって連続的に表される。輝度は0.0～1.0までの値を取り、0.0が黒色、1.0が白色を表す。色相を12のビンに等分割し、これに白色と黒色を加えた14のビンに物体領域の各画素値を振り分

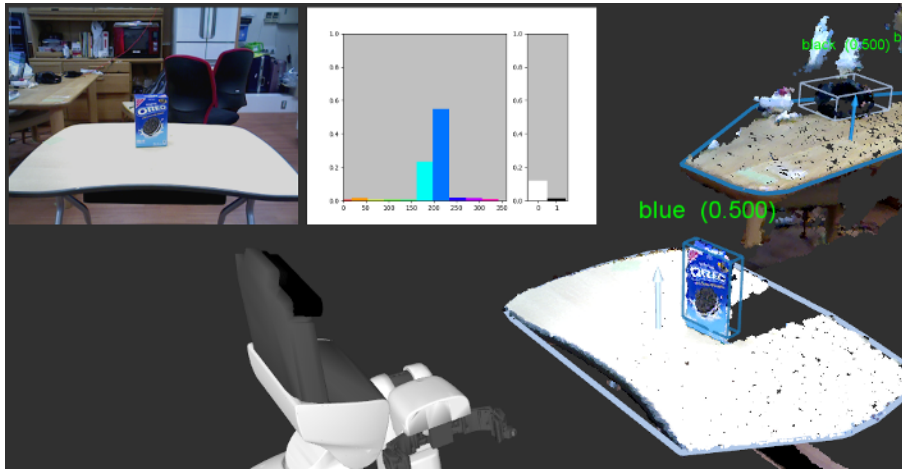


図 7.1: 色ヒストグラム検出器を用いた物体の性質認識 ([138] から引用)

け、ヒストグラムを生成する．ここで、輝度が一定の閾値より小さい、あるいは大きい画素値は色相にかかわらず白色、黒色に振り分ける．本研究では実験的に 0.1 以下、0.9 以上をそれぞれ白色、黒色とした．得られたヒストグラムに対して最大値となるピンをその物体の色として出力する．

物体基本形状検出器

丸いもの、四角形のものといった物体の外観の基本形状による近似を行うことにより得られる特徴の検出を行う検出器である．各物体の 3 次元点群について、3 次元的なエッジを、3 次元点群における各点の近傍点との法線ベクトルのなす角度を集計し、最小と最大値の差が閾値以上である点をエッジ上の点とみなし、3 次元点群上のエッジ上における点群を抽出する．次に物体を指示する平面上にエッジ上の点群を射影する．射影した同一平面上の点列に対して、RANSAC を用いて円と長方形に対するフィッティングを行う．モデルの一致誤差が小さいものをそれぞれ “round”, “cube” とラベル付する．

物体容積検出器

大きい、小さい、背が高い、背が低いなどの物体の容積に関する特徴を検出する検出器である．各物体の 3 次元点群について、すべての点を内包する容積最小のバウンディングボックスを計算する．各物体の 3 次元点群を支持平面上に射影し、同一平面上の点列に対して主成分分析を行う．主軸方向を長辺とし、すべての点列が含まれる最小の長方形を求める．得られた長方形を底面として、支持平面から最も距離が大きい点までの距離を高さとする直方体を物体のバウンディングボックスとする．バウンディングボックスに対して、長辺に対する 3 辺の長さの平均値の比が閾値を超えると背が高い (“tall”, 背が低い (“short”), 閾値以下の場合には大きい (“big”), 小さい (“small”) とした．“tall” と “short”, “big” と “short” はバウンディングボックスの体積が閾値を超えると前者、それ以外は後者とした．本研究では前者の閾値を 0.4, 後者を $0.1[m^3]$ に実験的に設定した．



図 7.2: 物体基本形状検出器を用いた物体の性質認識 ([138] から引用)

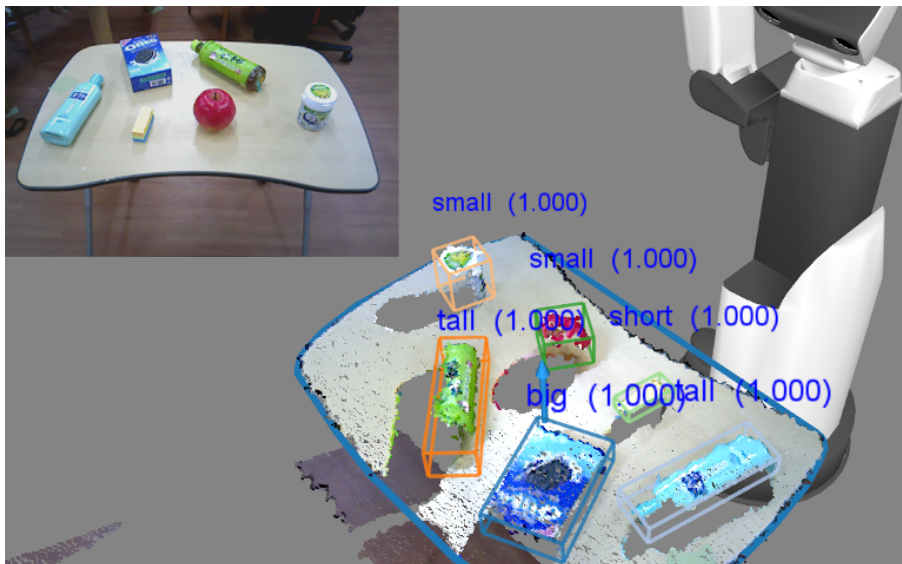


図 7.3: 物体容積検出器を用いた物体の性質認識 ([138] から引用)

7.3.6 音声自然言語指示を伴う日常生活タスクの指示実験

本節において説明した音声を通じての自然言語による曖昧性解決機能を有する行動指示機能について、実ロボットに搭載し、日常生活環境における指示を行い有効性を検証した。

日常生活環境におけるタスク指示として、家事支援ロボットのための国際競技会である Robocup@Home 2017 における競技種目 General Purpose Service Robot におけるタスク指示文を生成するアプリケーション [142] を用いた。得られたタスクの指示文章を複数人で読み上げ、ロボットの頭部に備えられたマイクロホンによって認識することで実験を行った。言語理解モデルの学習データとして指示文を表 7.1 にしたがって、199 文ラベル付したものをを用い

表 7.3: 人の音声によるタスク指示認識結果における認識失敗例 ([138] から一部引用)

Ground Truth	Recognized Text
Pick up the green tea from the fridge	pick up the green T from the fritz
tell me how many tea there are on the kitchen table	tell me how many teeth are on the kitchen table
find the Pringles in the bedroom	find the Rings in the bedroom
find the Manju in the kitchen	find the man dew in the kitchen

た．既知の概念として 26 の物体 (図 7.4) と 48 地点の場所を与えた状態で，ランダムに生成されるタスク指示を 128 回行った中で，98 回のタスクにおいて目標状態を満たす行動を行った．



図 7.4: 本実験において使用された物品 ([138] から引用)

タスク自体は成功したが，音声によってロボットにタスクを指示した際に誤って認識された自然言語の文字列の例を表 7.3 に示す．

タスク生成器によって得られた文章 “Pick up the green tea from the fridge.” について，音声指示を行った時，音声認識の結果から自然言語の単語列 “pick up the green T from the fritz.” が得られた．ここで，表 7.1 と学習した言語理解により “pick up” から動作 “pick” が指示されたことがわかる．“pick” は $?obj$ ， $?loc$ の 2 つをパラメタとするが，自然言語の単語列に既知の単語がないため，パラメタの割当は行われぬ．ここで音響レベルにおける曖昧性を解消するため，単語中における名詞を Grapheme-to-Phoneme 変換によって音素に変換し，

Tf-idf 特徴量とランダムフォレスト分類器を用いて既存知識の単語列との一致度を比較する．ここで未知概念とみなすしきい値は 0.5 に実験的に決定した．

単語列 “green T” は音素列 “G R IY1 N T IY1” に，“fritz” は “F R IH1 T S” にそれぞれ変換された．学習済み分類器へ入力した結果を表 7.4 に示す．“green T” は物体を表すシンボル “green tea” へ，“fritz” は場所を表すシンボル “fridge” という既知の概念にそれぞれ置換された．ここで再度動作 “pick” に対するパラメタを確かめると，*?obj* には “green tea”，*?loc* には “fridge” が割り当てられる．

さらに動作 “pick” を行うために必要な予備動作を推論することで，以下のような動作列を得る．

```
(move-to fridge)
(find-object green-tea fridge)
(pick green-tea fridge)
```

また未知物体概念を扱う場合として，ロボットの知識には無い物体である manju を持ってくるタスクの指示を行った．表 7.5 に音声対話を示す．“manju” は音素 “M AE1 N D UW1” に変換され，分類器へ入力される．結果を表 7.4c に示す．未知物体概念とみなす閾値 0.5 を下回っているため，“manju” は未知概念として扱われる．未知概念の性質を得るために追加の対話を行い，色の性質である “white” と大きさの情報である “small” を得た．

予備動作の推論によって以下の動作列が得られる．

```
(move-to living-drawer)
(find-object manju living-drawer)
(pick manju living-drawer)
(move-to start)
(deliver manju me)
```

ここで find-object における未知概念の場合における認識の様子を図 7.5 に示す．各物体について既存の物体において学習を行った物体検出器，色ヒストグラム検出器，物体容積検出器，物体基本形状検出器による検出結果を統合したラベルが表示されている．各検出器は非同期・並列に動作し，検出器ごとに計算コストが異なるため，各検出器における検出が終わり結果が得られた時点ごとに，対話によって得られた物体の性質ラベルとの一致を行う．初めて認識物体の中から一つの物体が特定された時点で把持を行う．図 7.5 では右から 2 番めの物体に対して “white” と “small” というラベルが色ヒストグラム検出器，物体容積検出器からの出力として得られ，これにより検出された物体のうち未知概念の物体特徴に一致する物体が唯一特定されるため，この物体を対象物体とみなし，把持・運搬を行うことでたすくの目標状態を満たす動作列を実行することができた．

表 7.4: 分類器を用いて認識したシンボルの推定類似度

No.	Symbol	Phoneme	Similarity
1	green tea	G R IY1 N T IY1	0.70
2	fries	F R AY1 Z	0.04
3	cleaning stuff	K L IY1 N IH0 NG S T AH1 F	0.03
4	snack	S N AE1 K	0.03
5	plate	P L EY1 T	0.03

(a) green T

No.	Symbol	Phoneme	Similarity
1	fridge	F R IH1 JH	0.54
2	bed	B D	0.21
3	mini desk	M IH1 N IY0 D EH1 S K	0.07
4	microwave	M AY1 K R OW0 W EY2 V	0.03
5	kitchen sink	K IH1 CH AH0 N S IH1 NG K	0.03

(b) fritz

No.	Symbol	Phoneme	Similarity
1	snack	S N AE1 K	0.34
2	cold brew	K OW1 L D B R UW1	0.17
3	apple	AE1 P AH0 L	0.12
4	fruit	F R UW1 T	0.10
5	spoon	S P UW1 N	0.09

(c) manju

7.4 随時指示・修正を受け付ける日常生活タスク実行システム

前節において、ロボットが日常生活環境においてユーザから音声対話による自然言語指示を受け取り、タスクを自律的に遂行するための機能について説明した。本節ではロボットが日常生活環境においてタスクを実行中にロボットに対して共生する人間が随時指示を送り、ロボットの行動や判断に対して割り込み・修正を行うことが可能となる機能の構成法について第2.7.1節において説明した階層的な分類に従い説明していく。

表 7.5: 音声によるタスク指示の対話 ([138] から引用)

Robot	Hi, may I help you?
User	Please give me the manju from the living drawer.
Robot	Now I'm thinking...
Robot	Sorry, I don't know the manju. What is manju?
User	It's white, small sweets.
Robot	OK.



図 7.5: 未知概念物体の認識 ([138] から引用)

7.4.1 タスクレベル

ロボットがタスクを実行中に別のタスクを割り込むというレベルでの割り込みである。ロボットは割り込みを指示された時点で実行中のタスクを中断する。このとき最も単純な割り込みの方法は、現在のタスクをその場で停止して、割り込みタスクをそのまま開始するというものである。これは割り込まれた時点で行っているタスクの動作がタスクの初期状態と変わらないときにしか有効ではない。なぜならタスクが中断された状態が保存されないため、割り込みタスクを行ったあとに元のタスクを再開することができなくなるからである。したがって、ロボットは割り込まれた時点で行っているタスクを再開可能な状態に中断したあと中断状態を記憶し、割り込みタスクを行った後に記憶した中断状態からもとのタスクを再開する必要がある。また、ロボットがタスクを実行中、ロボット自身の状態に応じて別のタスクを割り込まれるときにロボットが現在置かれている状態が割り込むタスクと競合する可能性がでてくる。例として「トレイを運搬する」タスクを実行中のロボットに対して「冷蔵庫からコーヒー缶を取って人に渡す」というタスクを行うような指示を割り込みする場合を考える。使用するロ

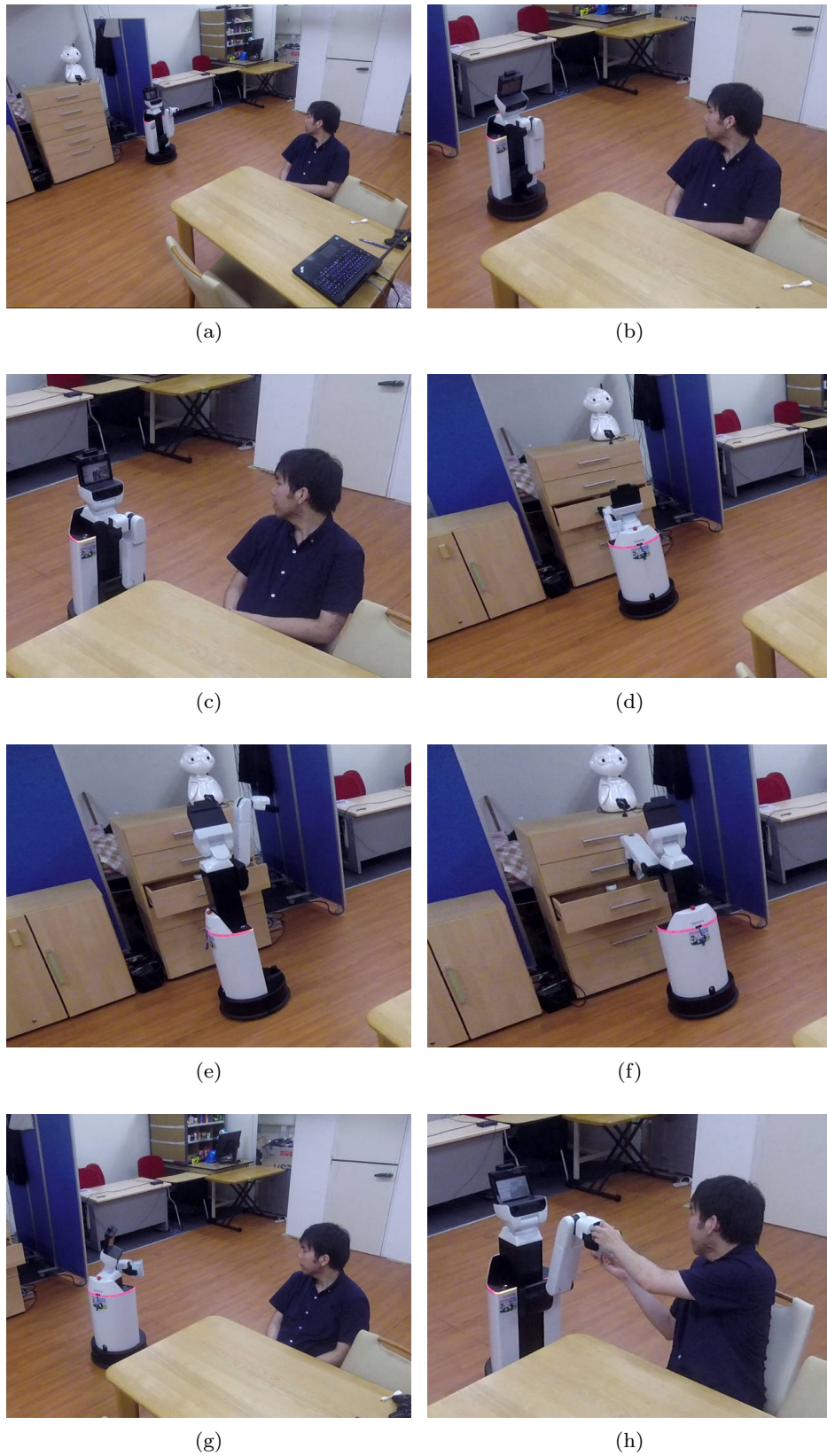


図 7.6: 音声対話による自然言語指示による日常生活支援タスク実験 ([138] から引用)

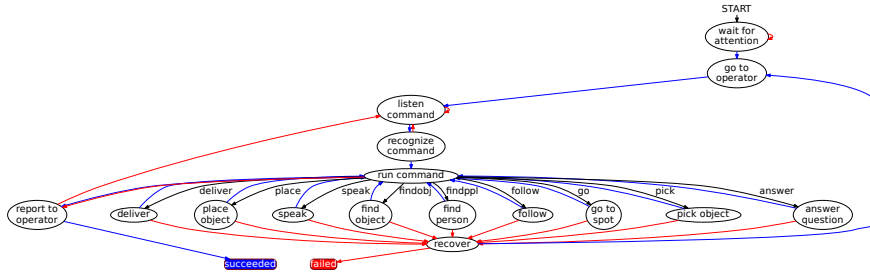


図 7.7: 本実験で使われたロボットの動作ステートマシン ([138] から引用)

ロボットは双腕台車型移動ロボットであり，トレイの運搬は両手を使用する．一方で冷蔵庫を開け，コーヒー缶を取る動作をもつ割り込みタスクも両手を使用するため，割り込みタスクを行うときの状態によってはロボットの物理的資源が競合するおそれがある．このときロボットは割り込むタスクの内容に応じて中断動作を柔軟に変える必要がある．以上をまとめると，タスクレベルの割り込みは以下の条件を満たすような問題を解くことになる．

1. 割り込まれたタスクを再開可能な状態にするよう適切な中断動作を行う
2. 割り込まれたタスクの進捗状況を可能な限り保存した状態で中断・再開する
3. 割り込むタスクが実行可能な状態になるような中断動作を行う
4. 最終的に割り込むタスクと割り込まれるタスクの双方の目標状態を満たす状態にする

ここで進捗状況を可能な限り保存した状態で中断を行うためには割り込みが行われた時点での状態から最も実行コストの小さい中断可能な状態へ遷移することである．本研究では割り込みが行われた時点での状態に対して，割り込みタスクの初期状態を結合した新しい初期状態を定義する．ここで，初期状態に含まれるコスト関数は割り込み時点でのコストの値に更新する．また目標状態については割り込みタスクと割り込まれるタスクの結合を行い，双方の目標状態を満たす状態とする．これらの初期状態と目標状態を行動計画器に入力することで，割り込まれるタスクの中断動作，割り込みタスク，割り込まれるタスクの再開動作を含む行動実現動作列が計画される．

例として，「トレイをキッチンへ片付ける」タスクの実行中に「コーヒーを取ってくる」タスクを割り込むことを考える．トレイをキッチンへ片付けるタスクの状態記述として以下のようなシンボルと関係式を用いて表すことができる．

```
(:init
  (and
    (= (move-cost start table) 1)
    (= (move-cost start counter) 4)
    (= (move-cost start kitchen) 7)
    (= (move-cost start fridge) 4)
```

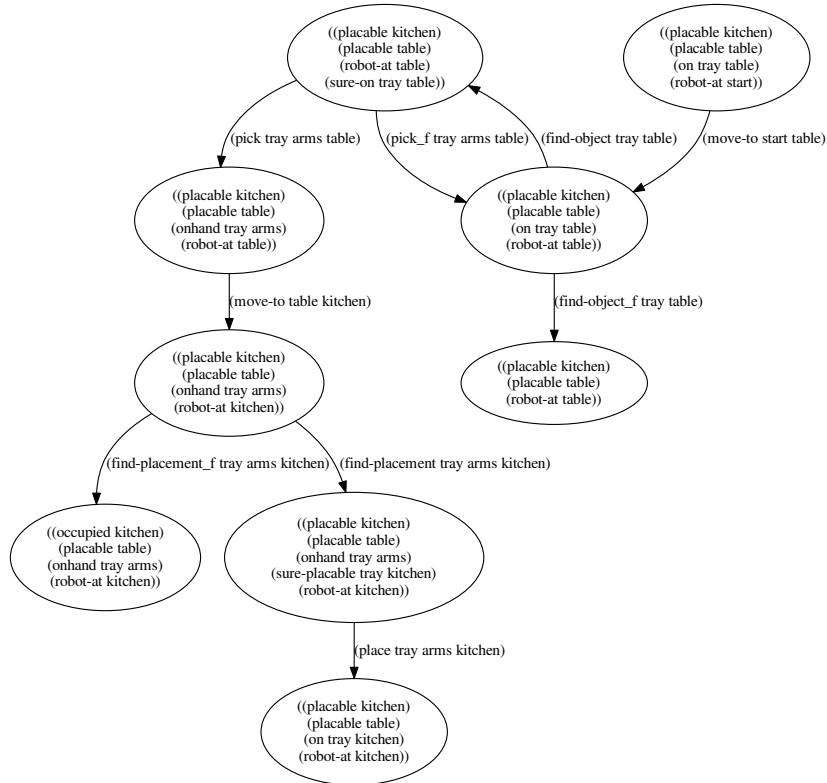


図 7.8: トレイ片付けタスクの状態遷移記述

```

... ; ; 他の2地点同士の組み合わせについてもコストを定義
    (at start)
    (on tray table)))
(:goal
 (on tray kitchen))

```

ここで tray はトレイを表すシンボル, kitchen, table, start は環境上の意味的な場所を表すシンボルである。また関係式 (at ?spot) は場所シンボルを引数にとり, ロボットが環境中のどこにいるかを表す関係式である。また (on ?obj ?spot) は場所と物のシンボルが接触関係にあることを表す関係式である。move-cost は2地点間を移動する際の移動時間に対応するコスト関数であり, ロボットが初期状態においている位置を表す start から各場所シンボルへの距離は現在のロボットの実際の環境における位置情報から行動計画時に動的に算出される。

以上の記述を行動計画器に入力した結果, 図 7.8 のような動作遷移グラフを得た。失敗復帰動作を除く動作列は以下ようになる。

```
(move-to start table)
(find-object tray table)
(pick tray arms table)
(move-to table kitchen)
(find-placement tray arms kitchen)
(place tray arms kitchen)
```

ここでロボットが (move-to table kitchen) 動作を実行中に音声による割り込みを行う。割り込みが行われた時点での環境状態記述は以下のようになる。

```
(onhand tray arms)
(robot-at start)
```

ここで start はロボットが割り込まれた時点において存在する環境における地点を意味するシンボルである。ロボットは移動中にトレイを両腕を使って把持している状態を表している。

ここで、中断を行う時点を移動動作の開始初期、中盤、終盤に場合分けし、それぞれの割り込み時点におけるロボットシステムの状態記述をもとにタスクレベルの割り込み手順について説明していく。

動作開始初期における割り込みタスク指示 ロボットが移動を開始した直後に割り込みを行った場合について説明する。割り込み時点での環境状態記述に対して、初期状態において定義されたコスト関数を現在のロボットの位置に基づいて更新する。ここでは割り込み時点が移動動作開始初期であることに留意して、以下のよう数値が更新された。

```
(= (move-cost start table) 2)
(= (move-cost start counter) 3)
(= (move-cost start kitchen) 6)
(= (move-cost start fridge) 5)
... ;; 他の2地点同士の組み合わせについてもコストを定義
(onhand tray arms)
(robot-at start)
```

行動計画器から出力される割り込みタスクの実現動作列として以下の記述が得られた。

```
(move-to start table)
(find-placement tray arms table)
```

```

(place tray arms table)
;; 冷蔵庫からコーヒーを取るタスク
(move-to table fridge)
(open-door rarm fridge)
(find-object coffee fridge)
(pick coffee larm fridge)
(close-door rarm fridge)
(move-to fridge table)
(find-placement coffee larm table)
(place coffee larm table)
;; 冷蔵庫からコーヒーを取るタスクの目標状態が満たされた
;; トレーを片付けるタスク
(find-object tray table)
(pick tray arms table)
(move-to table kitchen)
(find-placement tray arms kitchen)
(place tray arms kitchen)

```

動作開始初期に割り込みが行われ、ロボットの位置が移動の初期位置と変わらなかったため、割り込まれるタスクの初期状態へと遷移する中断動作が行われ、割り込みタスクを実行したのちに再びもとのタスクの動作列を最初から実行するような行動となった。

動作開始中盤における割り込みタスク指示 ロボットが移動を開始後、移動が行われている最中に割り込みを行った。割り込み時点での環境状態記述は以下ようになった。

```

(= (move-cost start table) 3)
(= (move-cost start counter) 1)
(= (move-cost start kitchen) 3)
(= (move-cost start fridge) 6)
... ;; 他の2地点同士の組み合わせについてもコストを定義
(onhand tray arms)
(robot-at start)

```

行動計画器から出力される割り込みタスクの実現動作列として以下の記述が得られた。

```

(move-to start counter)
(find-placement tray arms counter)
(place tray arms counter)

```

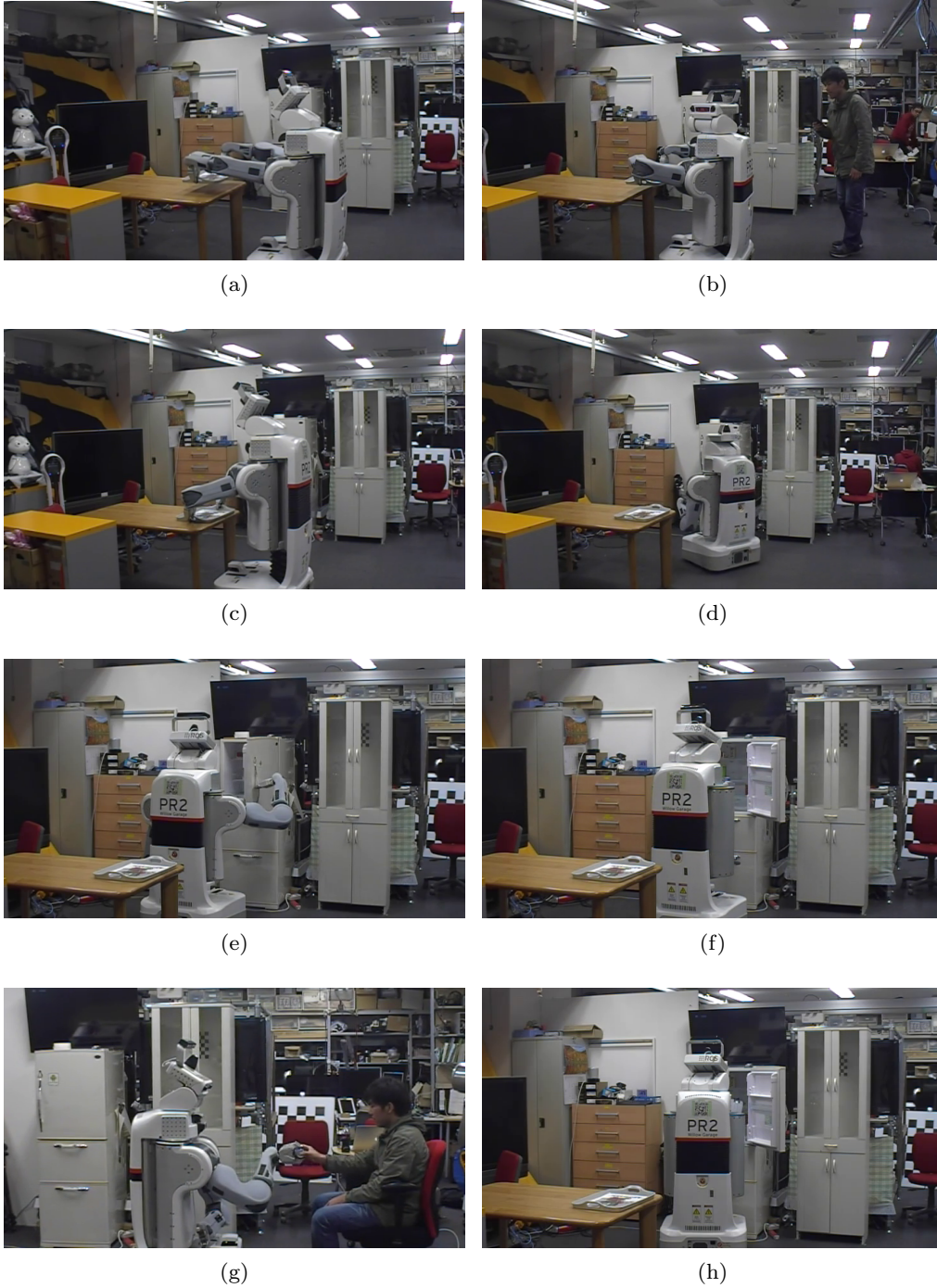



図 7.9: トレイ運搬タスク移動動作初期におけるタスク割り込み・再開実験


```
;; 冷蔵庫からコーヒーを取るタスク
(move-to table fridge)
(open-door rarm fridge)
(find-object coffee fridge)
(pick coffee larm fridge)
(close-door rarm fridge)
(move-to fridge table)
(find-placement coffee larm table)
(place coffee larm table)
;; 冷蔵庫からコーヒーを取るタスクの目標状態が満たされた
;; トレーを片付けるタスク
(move-to table counter)
(find-object tray counter)
(pick tray arms counter)
(move-to counter kitchen)
(find-placement tray arms kitchen)
(place tray arms kitchen)
```

移動動作中盤での割り込みでは、ロボットが初期位置である table や目標地点である kitchen とともに近くない一方、移動経路上において近い counter に対して、移動コスト値が最も小さくなっている。したがって、ロボットは中断動作として、現在までのタスク進行状況を最も保存する中断動作として tray を counter へ置く動作を計画し、割り込みタスクを行った後に再び counter から tray の運搬を再開する行動となった。

動作開始中盤における割り込みタスク指示　ロボットが移動を開始し、目的地へ近づいた時点で割り込みを行った。割り込み時点での環境状態記述は以下のようになった。

```
(= (move-cost start table) 5)
(= (move-cost start counter) 3)
(= (move-cost start kitchen) 1)
(= (move-cost start fridge) 10)
... ;; 他の2地点同士の組み合わせについてもコストを定義
(onhand tray arms)
(robot-at start)
```

行動計画器から出力される割り込みタスクの実現動作列として以下の記述が得られた。

```
(move-to start kitchen)
```

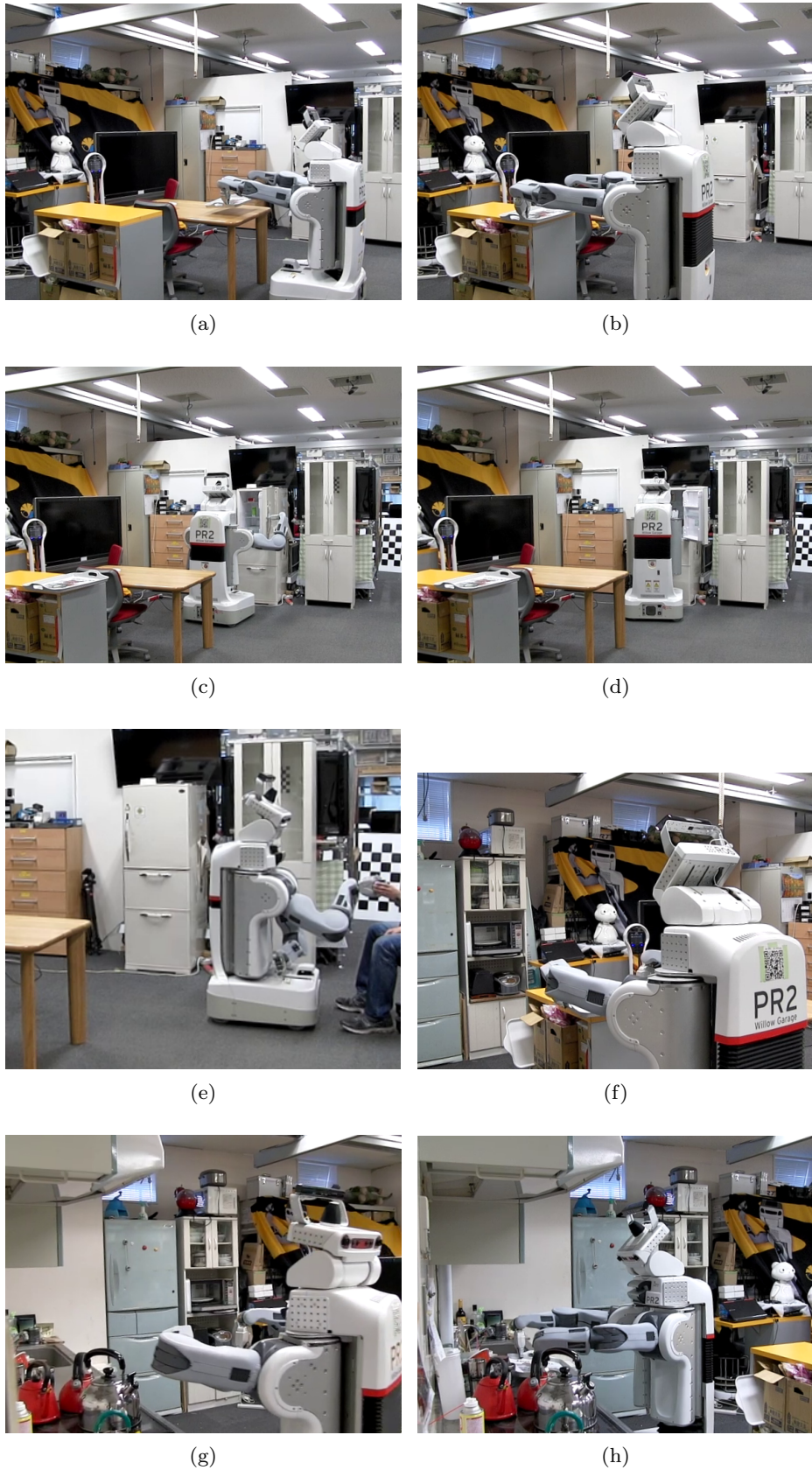


図 7.10: トレイ運搬タスク移動動作中盤におけるタスク割り込み・再開実験

```

(find-placement tray arms kitchen)
(place tray arms kitchen)
;; トレイを片付けるタスクの目標状態が満たされた
;; 冷蔵庫からコーヒーを取るタスク
(move-to kitchen fridge)
(open-door rarm fridge)
(find-object coffee fridge)
(pick coffee larm fridge)
(close-door rarm fridge)
(move-to fridge table)
(find-placement coffee larm table)
(place coffee larm table)
;; 冷蔵庫からコーヒーを取るタスクの目標状態が満たされた

```

移動動作終盤での割り込み時の状態記述では、ロボットはトレイを片付けるタスクの目標地点である kitchen に最も近くなるようなコスト値となっている。ロボットは再び kitchen へ移動を行い、トレイを片付けるタスクの目標状態を満たしたあと、割り込みタスクを続けて実行することで双方のタスクの目標状態を満たす行動列が計画された。

7.4.2 アクションレベル

行動計画によって得られる状態遷移列において人によって状態遷移を変更するレベルの割り込みである。失敗復帰戦略を考慮したロボットシステムであってもロボットは自己位置推定の局所最適解や認識の false-positive, 確認動作における確認漏れなどにより、ロボットシステムにおいて意図されていない失敗が起こった場合はロボットが失敗した事実に気づかないため、復帰が不可能である。この時に人が気づいた場合にロボットに割り込むことでロボットの動作を停止し、本来のタスクを失敗せずに続行可能になるような機能が必要とされる。この場合、再開する時の状態は割り込みが行われた時に行っていた動作を正常に終了した時の状態とすれば良い。またシステムによって想定された失敗状態に移行しようとしているロボットに対しても人が割り込むことによって、失敗復帰動作を行うリスクや時間的コストをなくすことが可能となる。この場合は失敗復帰動作を行い、正常に終了した時の状態とすることでタスクを続行可能になる。

例としてロボットが冷蔵庫からコーヒーを取るタスクを考える。このタスクの動作遷移グラフは図 7.12 となる。

ロボットが冷蔵庫を開けた時にコーヒーボトルが倒れていて、ロボットはお茶のボトルを誤認識し、把持しようとしている。そこで人が誤った把持に気が付き、ロボットに対して直接接触により割り込みを指示、本来のコーヒーボトルを把持するように修正し、タスクの再開を行う。割り込みが行われた時点の動作は (pick coffee larm fridge) であり、ロボットは

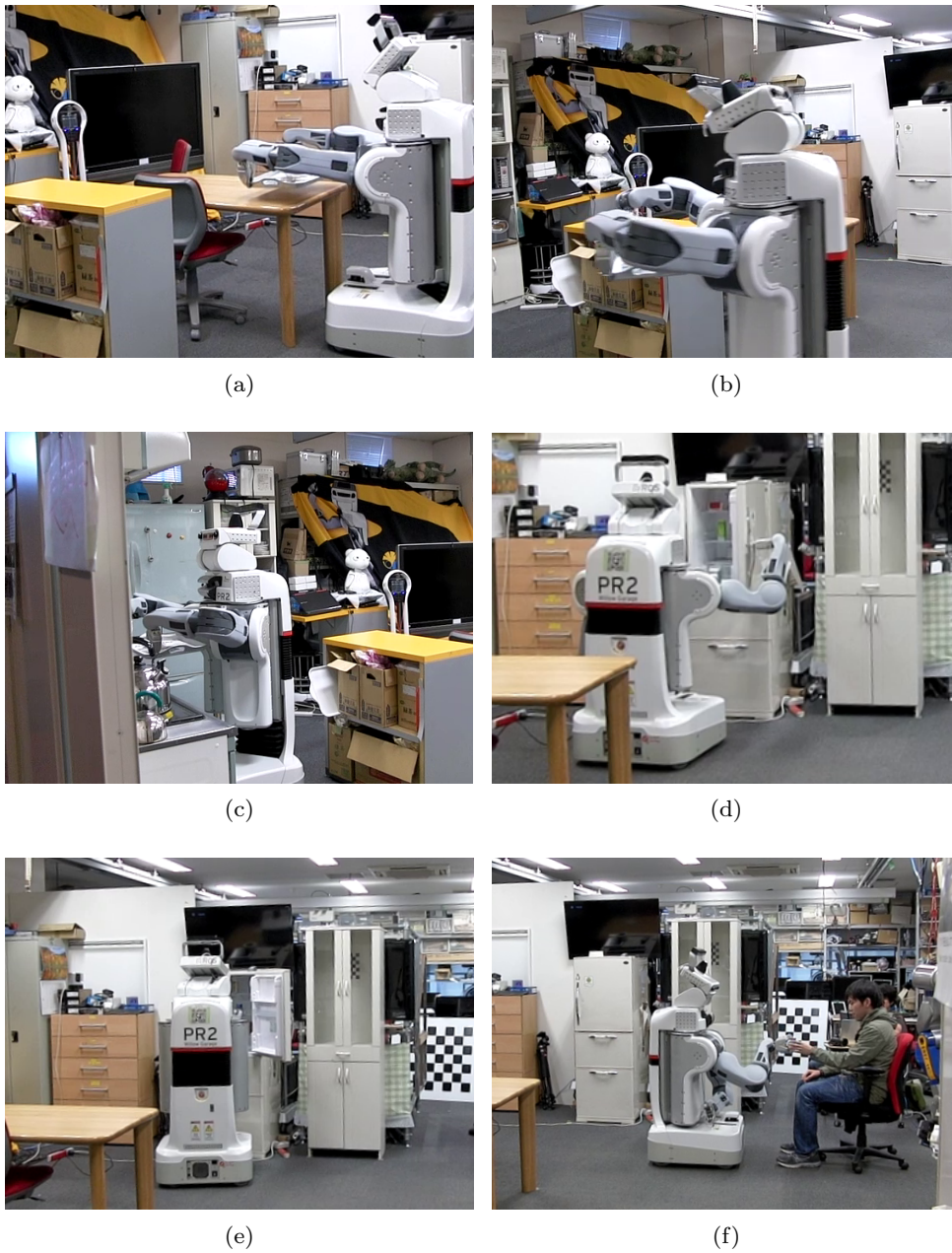


図 7.11: トレイ運搬タスク移動動作終盤におけるタスク割り込み・再開実験

この動作の正常終了時の状態

- ((openable fridge)
- (placable fridge)
- (placable table)
- (sure-placable fridge)

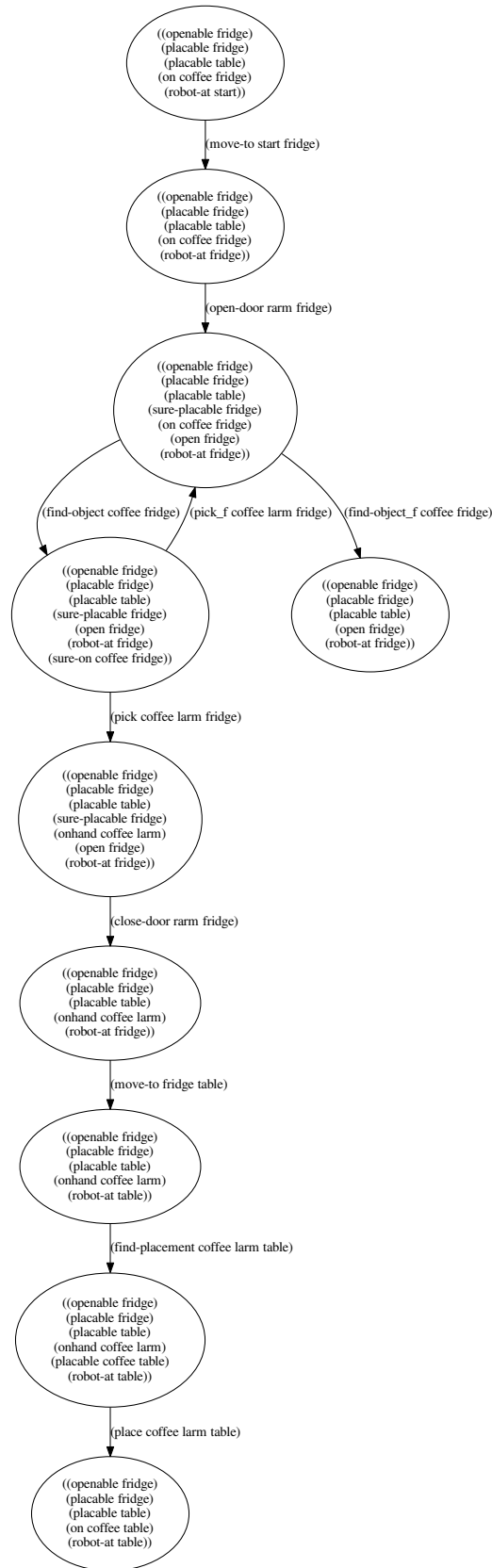


図 7.12: コーヒーを冷蔵庫から取るタスクの状態遷移記述


```
(onhand coffee larm)
(open fridge)
(robot-at fridge))
```

から再開し、動作 (close-door rarm fridge) を開始する。

7.4.3 モーションレベル

ロボットがアクチュエータを作動させている最中の関節角軌道に対して割り込みを行うことで動作を一時停止し、軌道を修正した後に再開を行うというレベルである。物体の把持失敗をユーザが予期したときにロボットの動作を一時停止し、ロボットのアームを直接接触することにより関節角軌道を修正することで、タスク全体を失敗することなく続行することができる。

第7.4.2節において、本来把持対象ではないお茶のボトルへ向かった手先目標指令に基づく関節動作を、直接接触におけるロボットアームの操作によりコーヒーボトルを把持した姿勢へと修正することがモーションレベルにおける割り込みとなる。

7.4.4 割り込み監視機能を備えるロボットシステムの実装

前節までに説明された割り込みを実現するためにはロボットがタスクを実行中の任意の時点でユーザからの割り込みや判断の修正指示を受け入れる機能がロボットシステムに必要とされる。本研究ではこの機能を並列分散型ロボットアーキテクチャにおいて、通常のタスクを行うプロセスに加えて割り込み指示を監視するプロセスを起動することで実現している。

次に割り込みを行うための条件を説明する。

音声認識による割り込み

ロボット頭部に取り付けられたマルチアレイマイクから得られる周囲の音声に対し、音源定位と音声のパワースペクトル密度を計算することにより、日常生活環境音に対して閾値を超えた大きな音が認識された際に割り込みが行われるときの条件である。割り込みが行われたのち、音声発話と音声の自然言語認識を用いた対話を行うことで、先に説明された各レベルにおけるロボット目標状態の修正を行っていく。

直接接触による割り込み

ユーザがタスクを実行中のロボットに対して直接的に接触することにより割り込みを指示する条件である。ロボットは行おうとしている動作において通常は発生しないような関節トルクを検知したときに割り込みが行われる。割り込み後には、ロボットの関節に対して外部から与えられる力に対して追従を行う(第3.3.7節)ことで、モーションレベルの割り込みにおいて修正する対象となる関節軌道を直接指示することが可能となっている。

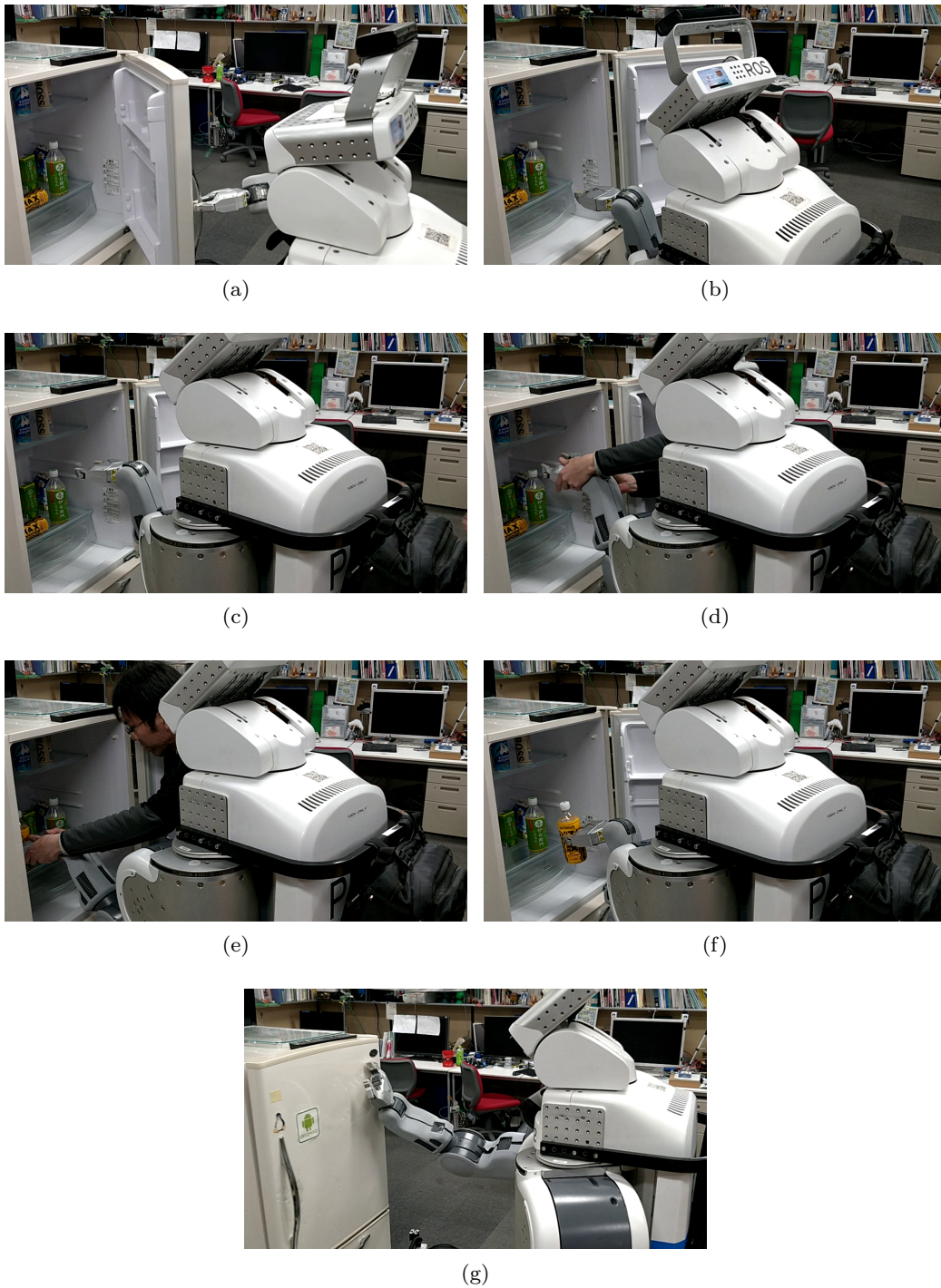


図 7.13: コーヒーを冷蔵庫から取るタスクにおけるアクションレベル割り込み・モーションレベル修正・再開実験

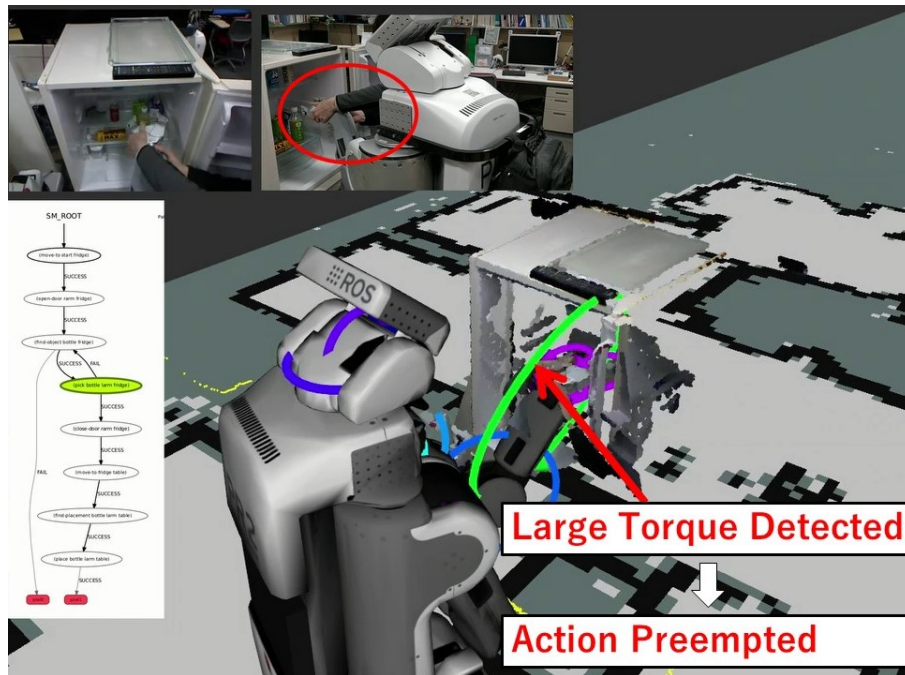


図 7.14: 関節トルク値の監視に基づく直接接触による割り込み

7.5 おわりに

本章ではロボットがユーザからタスクを受け入れる際の指示を受け付けるために必要な機能として音声を用いた自然言語による指示を取り上げ、日常生活環境における指示に含まれる曖昧性やロボットシステムの持つ知識との不一致が含まれる指示を用いても正確に意思伝達を行うことのできる機能について述べた。また指示を受けとり、タスクを行っている最中にも生活環境を共有する人間に常に気を配り、ユーザから随時割り込みや修正を受け入れる機能について、ロボットシステムにおける行動制御の観点から割り込みの条件や対象を分類し、割り込みによって中断・復帰を自律的に行うシステムを提案した。本章で提案された機能を実際のロボットシステムに統合的に搭載し、曖昧性を含む音声指示による日常生活タスクの自律計画実行、および実行中タスクの随時指示・判断修正を行い、その有効性を検証した。

第 8 章

日常生活タスクの局所合理性に基づく具体化機能を用いたタスク計画実行システムの評価実験

8.1 はじめに

本章では、本研究において提案する長期経験蓄積機能に基づく日常生活支援ロボットシステムの有効性を評価するため、提案システムを実ロボットへ搭載し、環境に非依存な形で記述された状況レベルの日常生活タスクをロボットが環境において経験的に得た局所合理性を含む知識を用いて具体化し、自律的に行動を計画・実行することができることを検証していく。

8.2 食事の準備タスクにおける局所合理性を用いた具体化タスクの計画・実行

本章における実験で行ったタスクは食事の準備である。食事の準備に必要なことは人が食事をするために必要な食べ物、飲み物、皿やコップなどの容器をテーブルへ置き、人がすぐに食事を開始可能になるように準備を行うことである。食べ物や飲み物として具体的に何を準備すればいいのかは、ロボットやロボットと共生する人の属する環境や文化、季節や時間帯や人ごとの嗜好など様々な要因の影響を受ける。また食品や食器を収納してある場所や、収納してあるものを取り出す方法など、環境に固有で予めロボットに与えておくことが困難な知識がタスクの計画・実行のために必要になる。

食事の準備を行う状況レベルタスク記述を以下の通り与える。この記述は後述の実験を通じて変更の必要がない環境非依存な記述となっている。

(: INIT

(ROBOT-AT START)

```

(IN-PLACE ?SPOT)
(USUAL ?OBJ ?SPOT :BEFORE ?STAMP)
(ON ?OBJ ?SPOT))
(: GOAL
(IN-PLACE TABLE)
(USUAL ?OBJ TABLE :AFTER ?STAMP)
(ON ?OBJ TABLE))

```

ここで変数?OBJは(USUAL ?OBJ TABLE :AFTER ?STAMP)に注目すると、テーブルによくある物であることがわかる。?SPOTは初期状態における(ON ?OBJ ?SPOT)によって、目標状態においてテーブルにあるものが初期状態にある場所を表す。さらに(USUAL ?OBJ ?SPOT :BEFORE ?STAMP)によって?OBJがよくある場所を示すことがわかる。?STAMPは?OBJがよくある場所について時間的な制約を与える変数となっている。

?STAMPにおける時間的な制約として、朝食、昼食、夕食の時間に関するシンボルを与えることで、同じタスク記述を用いて書く食事の環境に依存する局所的な合理性を踏まえたタスクとして具体化し、食事の準備を行っていく。

8.2.1 朝食の準備

?STAMPとして朝食を行う時間を与えることにより、目標状態における(USUAL ?OBJ TABLE :AFTER ?STAMP)は朝食を終えた時間における場所 TABLEによくある物であるという制約をシンボル?OBJに対して与える。確率的物体配置地図を用いると、TABLEにおいて存在確率が大きい物として?OBJはBOWL, SERIAL, MILKと具体化可能であることがわかる(図8.1b, 図8.2b, 図8.3b)。ここで複数の物のシンボルに対してUSUALによって具体化を行う場合は連言によって?OBJを用いる関連式が結合される。

```

(: INIT
(ROBOT-AT START)
(IN-PLACE ?SPOT0)
(IN-PLACE ?SPOT1)
(IN-PLACE ?SPOT2)
(USUAL BOWL ?SPOT0 :BEFORE BREAKFAST)
(USUAL SERIAL ?SPOT1 :BEFORE BREAKFAST)
(USUAL MILK ?SPOT2 :BEFORE BREAKFAST)
(ON BOWL ?SPOT0)
(ON SERIAL ?SPOT1)
(ON MILK ?SPOT2))

```

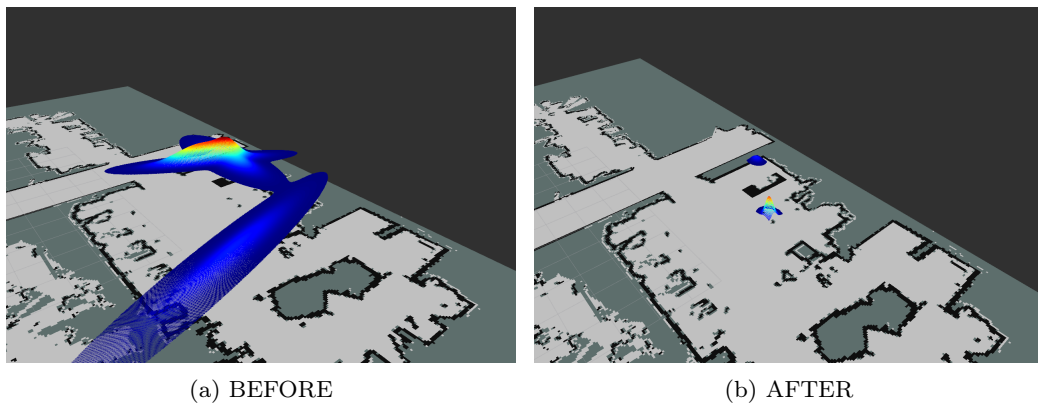


図 8.1: BOWL シンボルの朝食時間前後の確率的物体配置地図の 2 次元地図上への投影

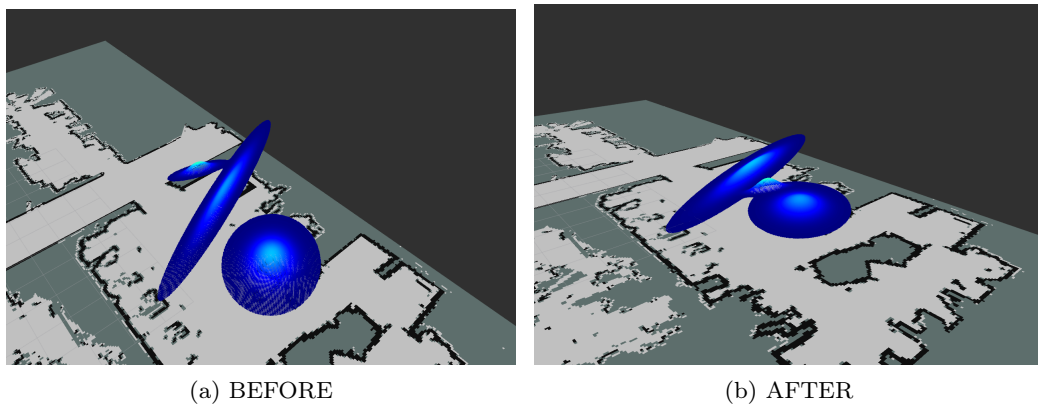


図 8.2: SERIAL シンボルの朝食時間前後の確率的物体配置地図の 2 次元地図上への投影

```
(: GOAL
  (IN-PLACE TABLE)
  (ON BOWL TABLE)
  (ON SERIAL TABLE)
  (ON MILK TABLE))
```

次に初期状態における USUAL を含む関係式における変数を物の定数シンボルと確率的物体配置地図を用いて具体化する。BOWL, SERIAL, MILK についての時刻 BREAKFAST 前の確率的物体配置地図から、それぞれの場所変数シンボルは KITCHEN, KITCHEN, FRIDGE と具体化される (図 8.1a, 図 8.2a, 図 8.3a)。

最後に各場所における定位置動作を表す関係式を初期状態へ加える具体化を行い、以下の具体的なタスク記述を得る。

```
(: INIT
```

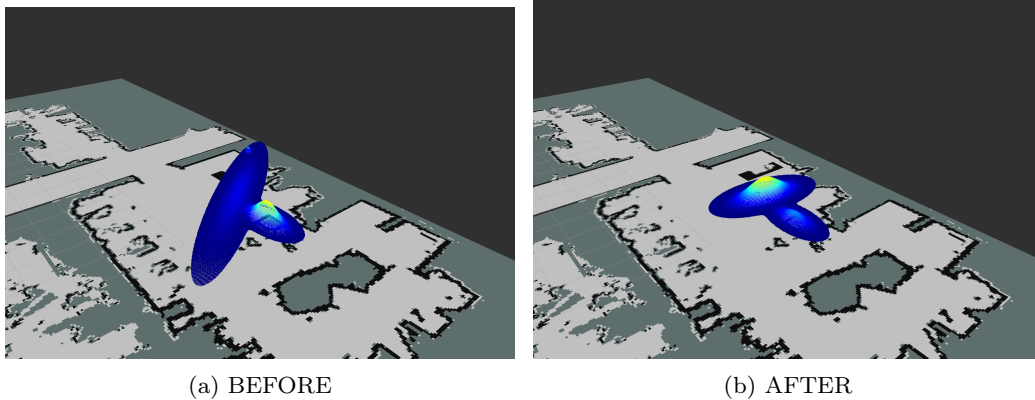


図 8.3: MILK シンボルの朝食時間前後の確率的物体配置地図の 2 次元地図上への投影

```
(ROBOT-AT START)
(PLACABLE KITCHEN)
(PLACABLE FRIDGE)
(OPENABLE FRIDGE)
(PLACABLE TABLE)
(ON BOWL KITCHEN)
(ON SERIAL KITCHEN)
(ON MILK FRIDGE))
(: GOAL
(ON BOWL TABLE)
(ON SERIAL TABLE)
(ON MILK TABLE))
```

8.2.2 昼食の準備

第 8.2.1 節と同様に?STAMP に対して昼食の時間を意味するシンボルによって制約を与える。同様の手順で具体化を行った。確率的物体配置地図を用いると、TABLE において存在確率が大い物として?OBJ は BOTTLE,BOWL,CUP と具体化された (図 8.6b, 図 8.7b, 図 8.8b)。

第 8.2.1 節と同様に具体化演算子に基づき具体化を行い、最終的なタスク記述として以下のようになった。

```
(: INIT
(ROBOT-AT START)
(PLACABLE FRIDGE)
```

```

(OPENABLE FRIDGE)
(PLACABLE DISHWASHER)
(OPENABLE DISHWASHER)
(PLACABLE KITCHEN)
(PLACABLE TABLE)
(ON BOTTLE FRIDGE)
(ON BOWL DISHWASHER)
(ON CUP KITCHEN))
(: GOAL
(ON BOTTLE TABLE)
(ON BOWL TABLE)
(ON CUP TABLE))

```

8.2.3 夕食の準備

第 8.2.1 節と同様に?STAMP に対して夕食の時間を意味するシンボルによって制約を与える．同様の手順で具体化を行った．確率的物体配置地図を用いると，TABLE において存在確率が大きい物として?OBJ は NOODLES と具体化された (図 8.11b)．

第 8.2.1 節と同様に具体化演算子に基づき具体化を行い，最終的なタスク記述として以下のようになった．

```

(: INIT
(ROBOT-AT START)
(PLACABLE COUNTER)
(PLACABLE TABLE)
(ON NOODLES COUNTER))
(: GOAL
(ON NOODLES TABLE))

```

8.3 おわりに

本章では，本研究における提案システムの有効性を検証するために，実ロボットに提案システムを実装し，提案手法により環境に非依存な形で記述された状況レベルの日常生活タスクが環境の局所合理性を満たす具体的なタスクとして実行可能であることを実験を通じて示した．

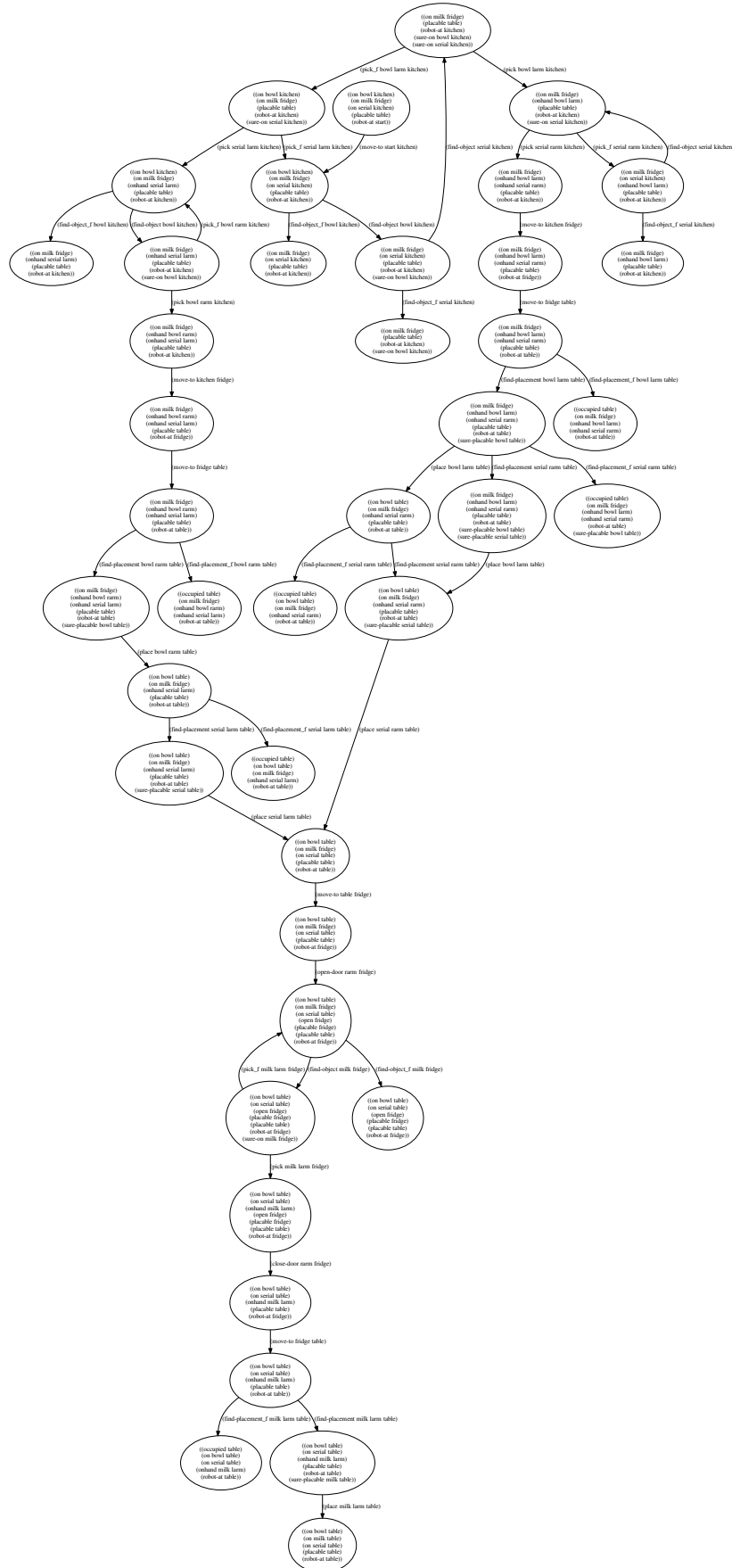


図 8.4: 朝食の準備タスクの状態遷移グラフ



(a)



(b)



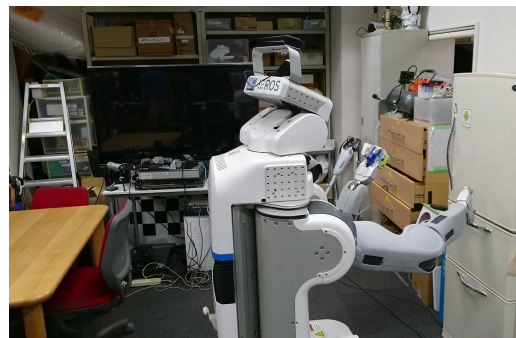
(c)



(d)



(e)



(f)



(g)



(h)

図 8.5: 状況レベルタスク記述の具体化・朝食の準備実験

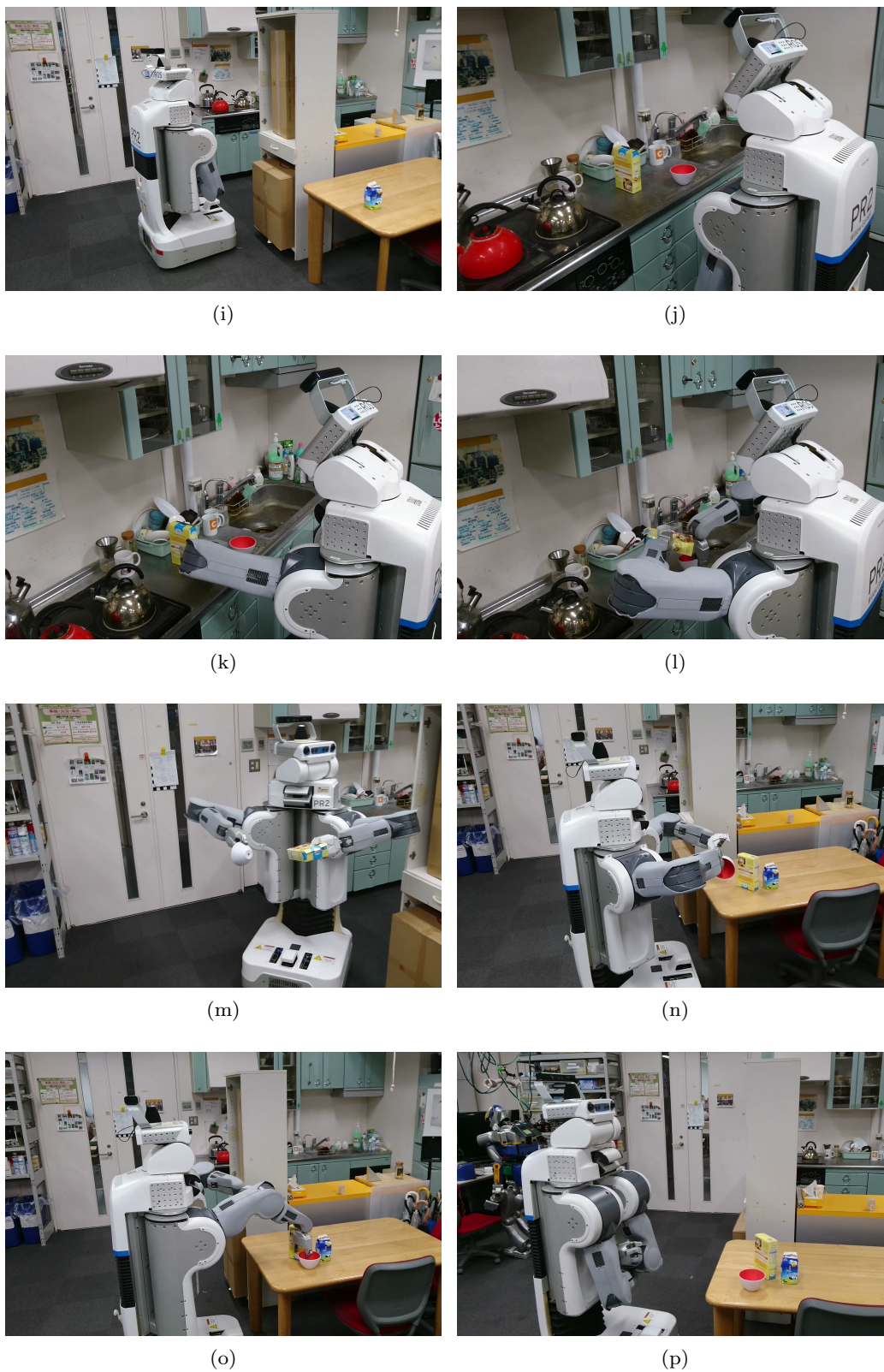


図 8.5: 状況レベルタスク記述の具体化・朝食の準備実験

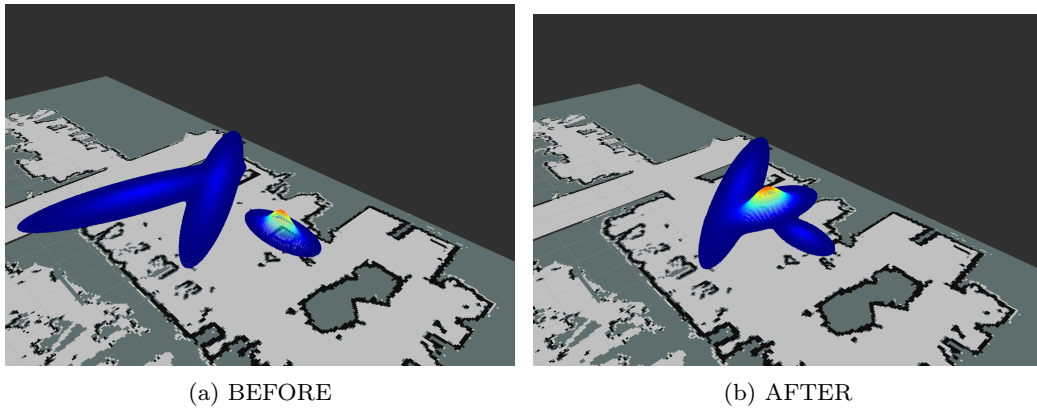


図 8.6: BOTTLE シンボルの昼食時間前後の確率的物体配置地図の 2 次元地図上への投影

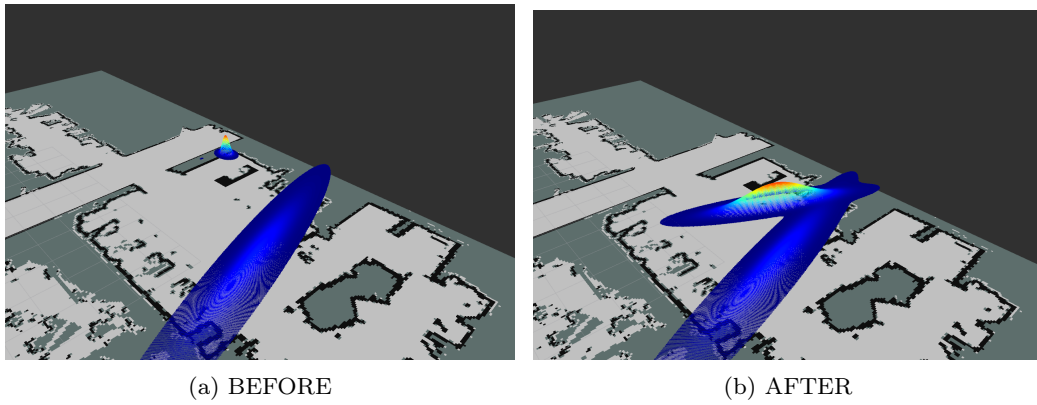


図 8.7: BOWL シンボルの昼食時間前後の確率的物体配置地図の 2 次元地図上への投影

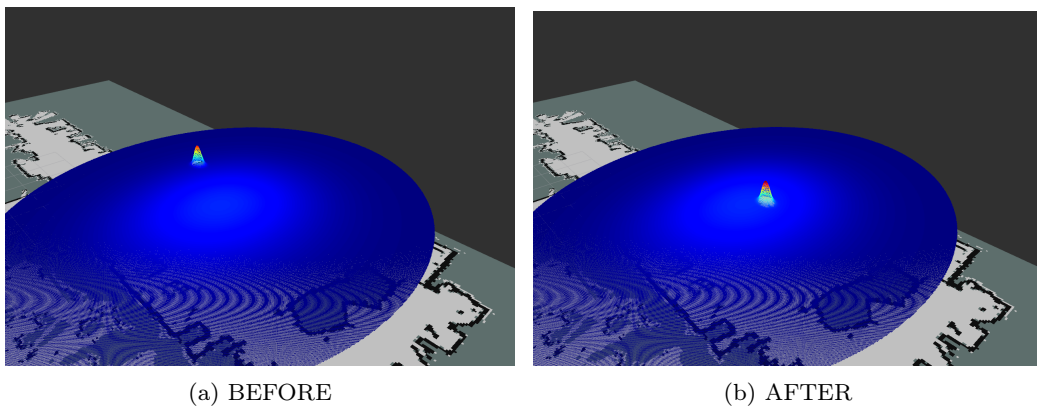


図 8.8: CUP シンボルの昼食時間前後の確率的物体配置地図の 2 次元地図上への投影

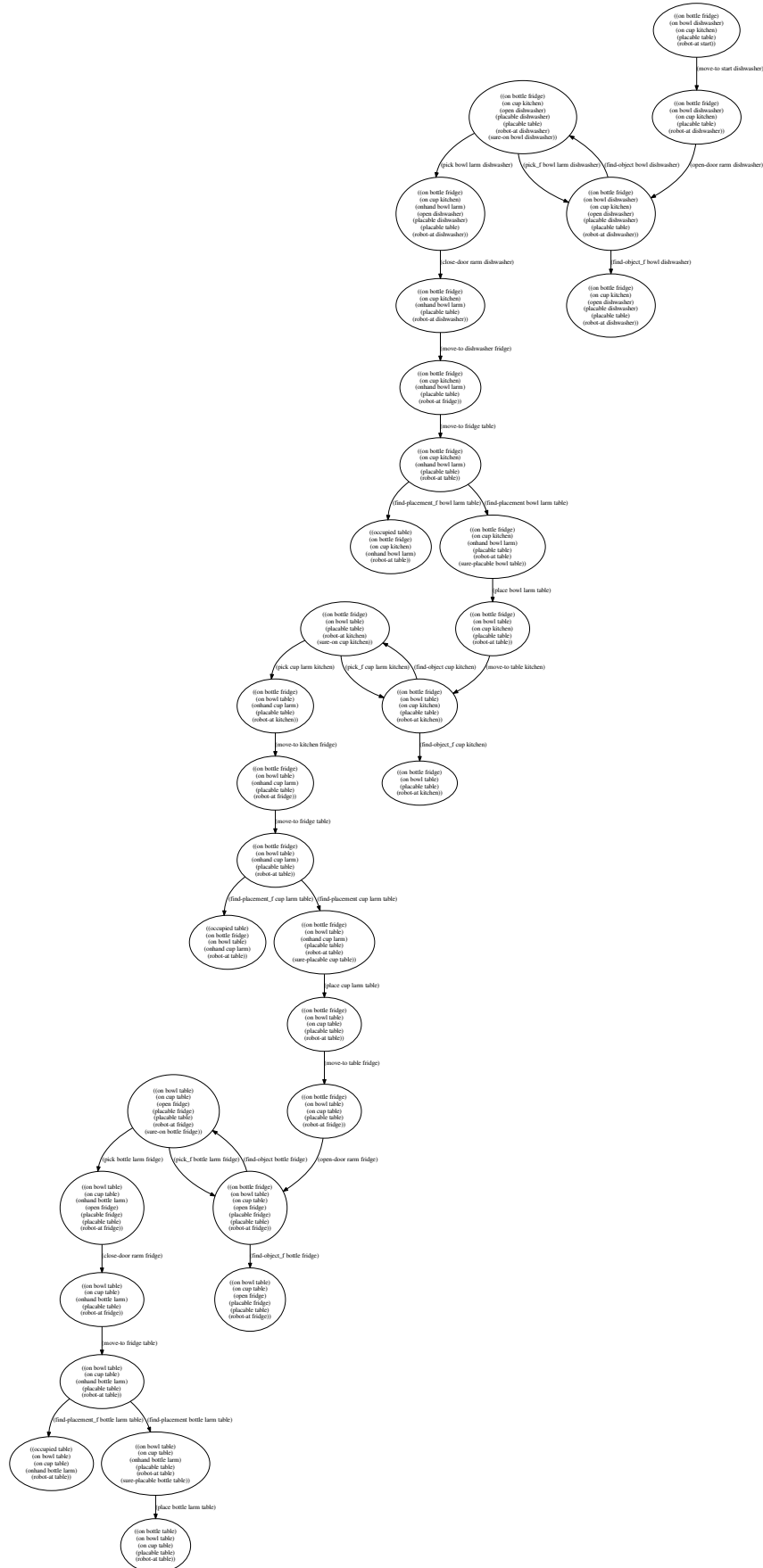
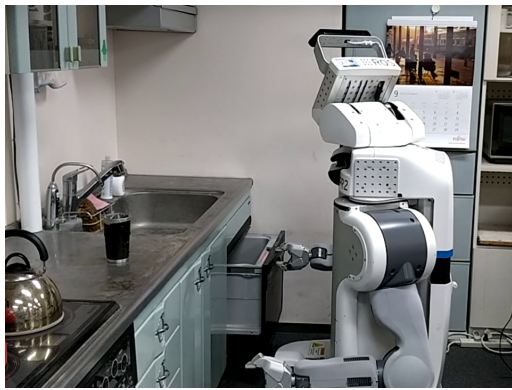


図 8.9: 昼食の準備タスクの状態遷移グラフ



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)

図 8.10: 状況レベルタスク記述の具体化・昼食の準備実験

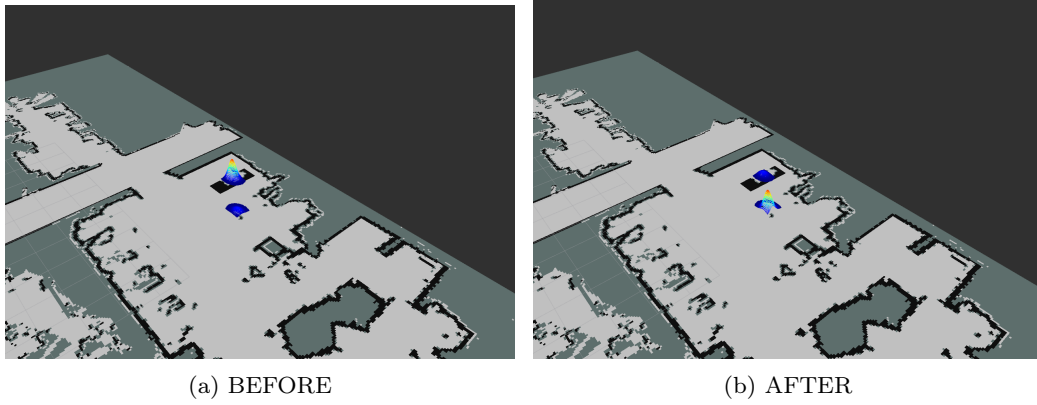


図 8.11: NOODLES シンボルの夜食時間前後の確率的物体配置地図の 2 次元地図上への投影

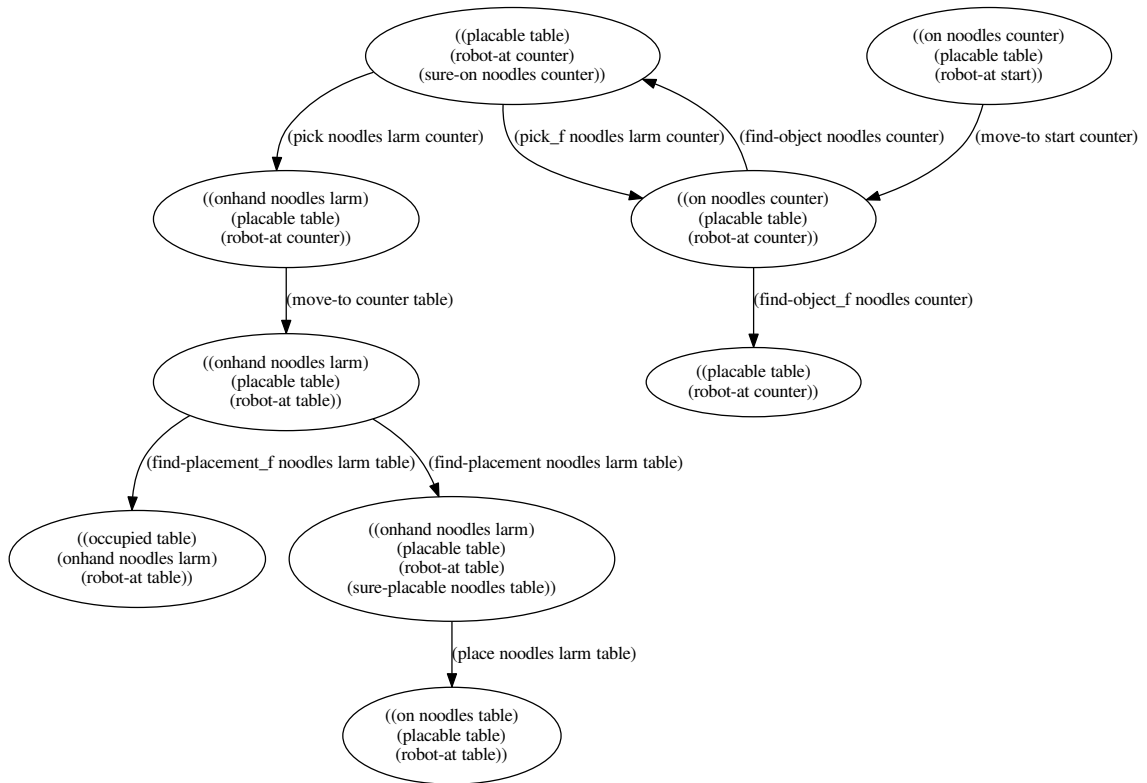


図 8.12: 夕食の準備タスクの状態遷移グラフ

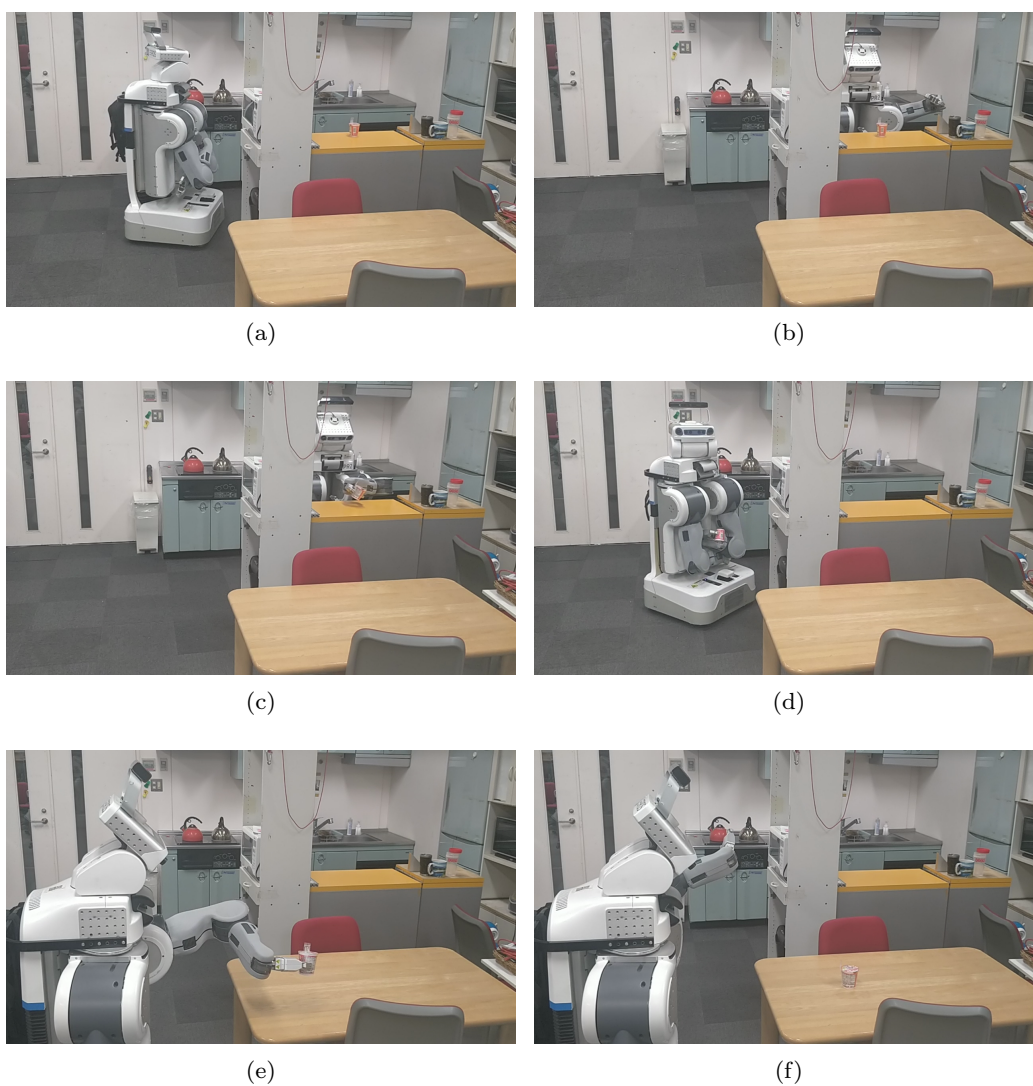


図 8.13: 状況レベルタスク記述の具体化・夕食の準備実験

第9章

結論

本論文は日常生活支援ロボットのための統合システムを構成するにあたって要求される、環境に非依存な日常生活タスクの記述と環境固有の局所的な合理性を経験から獲得し、具体化する機能を備え、環境や環境において共生する人に対して柔軟に適応するロボットシステムの構成法を主題に取り組んだものであった。

第1章「序論」では、統合的ロボットシステムの研究について体系的に分類し、日常生活タスクを行うことを目的とするロボットにとって、ロボットが固有の環境で過ごしていく中で自律的に長期経験を獲得し、自身の行動を環境に適応していく機能をロボットシステムに持たせることの必要性について述べた。また関連研究との対比を通じて本論文の位置づけを行った。

第2章「長期経験に基づくタスク具体化機能を備えた日常生活支援ロボットシステム」では、自律的にタスクを計画・実行するロボットシステムにとって重要な日常生活タスクの特徴として、目的が同じでも扱う物や手段、目標状態が常に変化し、その合理性が環境に依存することを取り上げ、予めシステムに与えておくことが困難な環境依存知識の獲得方策として、人との共生環境を前提として、環境における人と場所、物品との長期的な関連性に基づく経験知識の獲得を行う機能を備えることで、従来では固定的に書き下されてきたタスク記述を環境や実行時の状況に非依存な形で記述可能にすること、それをロボットが自律的に獲得した経験的知識を用いて環境に適した形に具体化する機能の必要性について説明した。さらに自律的に獲得された経験的知識には誤りを含むため、失敗可能性を考慮し、人による随時割り込みによる指示を受け付けることで、失敗を防ぐ機能の必要性と実現方策について説明した。また人の指示に含まれる曖昧性についても分類し、解決法についても説明した。

第3章「日常生活支援タスクを行う自律ロボットシステムの構成」では、本研究における実ロボットを用いた実験において使用されるロボットのハードウェア・ソフトウェア要素の説明を通じて、本研究において扱う日常生活タスクの自律計画認識行動を行うロボットシステムの特徴である分散プロセス型アーキテクチャとプロセス間の通信方式やシンボリックな状態記述言語を用いた行動計画機能の概要をまとめ、他の章で前提となっている機能や用語を説明した。

第4章「ロボットにおけるエピソード記憶に基づく長期経験の蓄積機構」では、日常生活環境におけるタスク計画を実現するための知識を獲得するための、長期経験蓄積機能の構成法に

ついて述べた。ロボットシステムにおいて経験蓄積機能を構成するために記憶の認知心理学的分類におけるエピソード記憶に着目し、センサや認識器などから出力される時系列データと、行動計画器によって扱われる環境状態記述を関連付けてデータを保存していくことで、あとから特定の出来事に関するデータを抽出することができるシステムを構成した。ロボットシステムにおいて扱われる膨大なデータの中で保存すべきデータの選択を自律的に行う手段として、分散プロセス型ロボットシステムにおけるプロセス間通信における遅延購読の導入によって実現可能となるノード間の動的結合性を利用し、ロボットシステム全体の性能低下やプログラムの変更をすることなく、経験蓄積可能な機能を提案した。蓄積した経験を出来事やロボットシステムにおける環境モデルの状態記述を用いて集約的に抽出するための方法についても述べた。実ロボットにおいて提案するシステムを実装し、日常生活環境において経験を蓄積することで、提案する機能の有効性を示した。

第5章「長期経験を用いた環境の局所合理性に基づく知識の自律獲得機構」では、ロボットが蓄積した経験に基づき、環境に非依存な状況レベルのタスク記述を具体化する上で必要となる、環境に固有の局所的な合理性を経験から抽出するための知識獲得モデルを日常生活環境における人とスポット・アイテムの関係性に注目しながら分類し、その獲得法を提案した。人とスポットに関して経験的に得られる知識として、環境における定位置における操作物体とその操作方法に関する知識を表す定位置操作モデルを挙げ、その獲得手法を示した。また環境が動的に変化する中で、操作物体の配置が変化しても対応する知識を更新していく方法についても示した。人とアイテムに関して経験的に得られる知識として、人の嗜好に関する知識を獲得する方法について説明した。アイテムに関して人から指示を受けた経験に基づき、同一カテゴリ内における物体の人ごとの嗜好を反映した嗜好モデルを構成する手法について示した。アイテムとスポットに関して経験的に得られる知識として、環境における確率的物体配置地図を定義した。ロボットが扱う物体をタスク実行中に認識した経験を用いてカテゴリごとの配置情報を抽出し、物体がよくある位置を確率モデルを用いて表す方法を示した。またそれぞれの知識獲得手法を実ロボットを用いて長期的に蓄積した経験から獲得可能であることを実験により検証した。

第6章「環境に非依存な状況レベルタスク記述と局所合理性に基づく具体化機構」では、従来では環境に依存した状態が固定的に書き下され、汎用性が小さかったロボットの作業目標レベルタスク記述に対して、環境に非依存な形で記述され、環境に即した具体的なタスク記述に展開可能な状況レベルタスク記述と具体化機能を提案した。固定的に書き下されてきた操作対象アイテムやスポットシンボルを変数や具体化のための演算子を用いて環境・実行時の状況に依存しない形で記述し、タスクの実行指示が与えられた時に経験や認識によって得られる知識を用いて変数を具体的なシンボルへ置換する機能を持つ具体化手法を適用することで、行動計画器に入力可能な具体的な形で各環境に適したタスク記述を自律的に獲得する仕組みについて説明した。またシンボリックな状態記述だけでなく、手続き的な知識における環境固有の構造物に対する物体操作についても、環境においてロボットが人と操作物体との相互作用を観察することで蓄積される長期経験から抽出される局所的な合理性に従った知識として操作のパラメータを獲得することで、ロボットの環境状態遷移を行う動作を増やすことが可能な仕組みを提案

した．この機能を備えるシステムにすることで，一般的に理解される日常生活における行為の目的を達成しながらも，各環境における経験や文化に基づく局所的な合理性をも満たす，環境に対して適応的なタスク計画実行が可能となった．

第7章「曖昧な指示・修正を随時受け付け対応可能な日常生活タスク実行システム」では，人間と共生する環境下におけるロボットとして，人間からの音声や直接接触を伴うインタラクションを題材に，ロボットシステムが行動を計画・実行するために持つ内部知識とロボットへ指示を行う人間が持つ知識との不一致を解消する機能をロボットシステムに持たせることによって，ロボットが行動を実行する際に起こりうる失敗を防ぎ，随時人間の指示による行動の変化に対応可能であることを示した．

第8章「日常生活タスクの局所合理性に基づく具体化機能を用いたタスク計画実行システムの評価実験」では，提案システムを備えたロボットを用いて，環境に非依存な状況レベルタスク記述を用いた日常生活タスクを行うことを通じて，同一のタスク記述から環境状態や環境から長期的に獲得した経験を用いて多様なタスクに具体化可能であるような自律適応的なシステムとなっていることを実験的に示した．

本論文の成果は以下のようにまとめられる．日常生活タスクは決まりきったルーチンワークである一方で，同じ目的でも環境に応じて多様な目標状態が考えられるため，具体的なタスクとしてシステムに定義を与えることが難しかった．というのも日常生活タスクを行う日常生活環境は人と共生する環境であり，ロボットや人によって常に変化するからである．一方で，人はこのような多様な記述が可能な日常生活タスクを識別・分類する知識を備えている．本論文では日常生活タスクを各環境において自律的に計画・実行するためのシステムを構成する方法として，従来では固定的に与えられてきた環境に暮らす人間の経験や文化に基づく局所的な合理性を，ロボットが人と共生する環境において自律的に経験し，知識として獲得する仕組みを提案した．局所合理性を形成する経験的獲得知識として，日常生活タスクを移動を伴う環境操作の連続と捉えることにより，日常生活タスクの行為主体である人と環境における移動を必要とする場所や扱う物品との相互作用をモデル化した．局所合理性をモデル化することにより，日常生活タスクを環境に依存しない抽象的な記述を用いてロボットに与えることができ，さらにロボットが自律的に得たモデルを用いて具体的なタスク記述へ特化する機能を備えるロボットシステムとなったことを示した．ロボットが自律的に得た知識は誤りを含む可能性があるため，その知識を利用して判断を行う場合には失敗を考慮したシステムにする必要がある．本論文では環境において共生する人からロボットがタスク中に随時指示を受け，判断を修正し，作業を続行する仕組みを備え，人から指示を受ける際の意思伝達における曖昧性についても考慮することで，ロボットの失敗を未然に防ぐことができるシステムとなっていることを実験を通じて確認した．本論文は，環境固有の知識や環境において共生する人の経験や文化に関する知識によって合理性が定義される日常生活タスクを，ロボット自らが環境において経験的知識から局所合理性を自律的に獲得し，環境に非依存な形で定義された抽象的なタスク記述を元に状況に適したタスク記述へと変化させ，計画・実行可能であることを示したのみでなく，タスク実行中に人による随時指示を受け入れ，対応することが可能な機能を導入することにより，人との共生が前提となる日常生活環境において人を支援するためのタスクを適応的自律的に行う

ことが可能なロボットシステムとして有効であることを示したものとなっている。

発表論文

査読あり発表論文

- (1) 垣内洋平, 古田悠貴, 三喜田浩行, 野沢峻一, 岡田 慧, 稲葉 雅幸: 家事支援ロボットにおける割り込み可能な階層化されたタスク実行マシンの自動生成システム, in 第 19 回ロボティクスシンポジウム予稿集, pp.631-636, 2014.
- (2) Ryohei Ueda, Masaki Murooka , Yu Ohara , Iori Kumagai , Ryo Terasawa , Yuki Furuta, Kunio Kojima , Tatsuhi Karasawa , Fumihito Sugai , Satoshi Iwaishi , Shunichi Nozawa , Yohei Kakiuchi , Kei Okada , Masayuki Inaba: Humanoid Integrated UI System for Supervised Autonomy with Massive Data Visualization over Narrow and Unreliable Network Communication for DRC Competition, in Proceedings of the 2015 IEEE-RAS International Conference on Humanoid Robots (Humanoids 2015), pp.797-804, 2015.
- (3) Yuki Furuta, Kazuhiro Sasabuchi, Yusuke Niitani, Kotaro Nagahama, Hiroaki Yaguchi, Kei Okada, Masayuki Inaba: Bring me manju from the drawer: Task Acquisition Framework under Incompleteness and Ambiguity using Interaction and Semantic Knowledge-enabled Perception, in 2017 IEEE/RSJ IROS Workshop Machine Learning Methods for High-Level Cognitive Capabilities in Robotics, 2017.
- (4) Yuki Furuta, Yuto Inagaki, Kei Okada, Masayuki Inaba: Self-improving Robot Action Management System with Probabilistic Graphical Model based on Task Related Memories, in Proceedings of the 14th International Conference on Intelligent Autonomous Systems, pp.811-823, 2016.
- (5) Yuki Furuta, Kentaro Wada, Masaki Murooka, Shunichi Nozawa, Yohei Kakiuchi, Kei Okada, Masayuki Inaba: Transformable Semantic Map Based Navigation using Autonomous Deep Learning Object Segmentation, in Proceedings of the 2016 IEEE-RAS International Conference on Humanoid Robots (Humanoids 2016), pp.614-620, 2016.
- (6) Asil Kaan Bozcuoglu, Gayane Kazhoyan, Yuki Furuta, Simon Stelter, Michael Beetz, Kei Okada, Masayuki Inaba, "The Exchange of Knowledge using Cloud Robotics", In International Conference on Robotics and Automation (ICRA), Bris-

bane, Australia, 2018.

- (7) Asil Kaan Bozcuoglu, Gayane Kazhoyan, Yuki Furuta, Simon Stelter, Michael Beetz, Kei Okada, Masayuki Inaba: The Exchange of Knowledge Using Cloud Robotics, IEEE Robotics and Automation Letters, Vol.3, No.2, pp.1072-1079, 2018.
- (8) Yuki Furuta, Kei Okada, Yohei Kakiuchi, Masayuki Inaba: An Everyday Robotic System That Maintains Local Rules Using Semantic Map Based on Long-Term Episodic Memory, in Proceedings of The 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.7641-7647, 2018.

国内学会・シンポジウム等における発表

- (1) 古田悠貴, 稲垣祐人, 垣内洋平, 岡田慧, 稲葉雅幸: IRT ホームアシスタントロボットによる掃除片付け作業シーケンスの PR2 による実現, in 第 31 回日本ロボット学会学術講演会講演論文集, 1I2-02, 2013.
- (2) 古田悠貴, 垣内洋平, 三喜田浩行, 植田亮平, 岡田慧, 稲葉雅幸: 日常生活支援ロボットによるオンサイト教示可能なロボットシステム, in 日本機械学会ロボティクス・メカトロニクス講演会'14 講演論文集, 1P2-Q06, 2014.
- (3) 東風上奏絵, 稲垣祐人, 古田悠貴, 植田亮平, 垣内洋平, 岡田慧, 稲葉雅幸: 具体的操作期の子供との対話を目指した視聴覚対話協調システムに関する研究, in 第 32 回日本ロボット学会学術講演会講演論文集, 1C3-08, 2014.
- (4) 大坪諭史, 古田悠貴, 稲垣祐人, 笹淵一宏, 室岡雅樹, 垣内洋平, 岡田慧, 稲葉雅幸: 動力学シミュレータを用いたホームアシスタントロボットによる家事支援タスク遂行, in 第 33 回日本ロボット学会学術講演会講演論文集, 1J3-06, 2015.
- (5) 鎌田一史, 古田悠貴, 植田亮平, 垣内洋平, 岡田慧, 稲葉雅幸: 日常生活支援ロボットにおける認識信頼度を利用した動作遷移による動作並列化システムに関する研究, in 第 16 回 SICE システムインテグレーション部門講演会講演概要集, pp.2625-2630, 2015.
- (6) 鎌田一史, 古田悠貴, 垣内洋平, 岡田慧, 稲葉雅幸: 日常生活支援ロボットにおける認識信頼度に応じた動作の遷移による動作並列化システムの研究, in 日本機械学会ロボティクス・メカトロニクス講演会'15 講演論文集, 1P2-T10, 2015.
- (7) 竹田棕, 古田悠貴, 長濱虎太郎, 矢口裕明, 稲葉雅幸: 生活支援ロボットにおける類似物探索による日用品収納行動に関する研究, in 第 35 回日本ロボット学会学術講演会講演論文集, 1I2-02, 2017.
- (8) 竹田棕, 二井谷勇佑, 古田悠貴, 長濱虎太郎, 矢口裕明, 岡田慧, 稲葉雅幸: 生活支援ロボットによる配置方法識別収納システムに関する研究, in 第 18 回 SICE システムインテグレーション部門講演会講演概要集, 2017.
- (9) 室岡貴之, 古田悠貴, 岡田慧, 稲葉雅幸: 実時間制御のオンライン学習による双腕台車型ロボットの中国ゴマ操作の獲得, in 第 36 回日本ロボット学会学術講演会講演論文集,

1P2-02, 2018.

- (10) 古田悠貴, 岡田慧, 稲葉雅幸: 日常生活支援ロボットにおける長期記憶蓄積に基づく文脈適応行動計画・動作システムを用いた片付けタスクの構成法, in 第32回人工知能学会全国大会, 2018, 4L2-02, 2018.

参考文献

- [1] Asimov Issac. *The Vocabulary of Science Fiction*. Asimov's Science Fiction, 1979.
- [2] Eric Krotkov, Douglas Hackett, Larry Jackel, Michael Perschbacher, James Pippine, Jesse Strauss, Gill Pratt, and Christopher Orłowski. The darpa robotics challenge finals: results and perspectives. *Journal of Field Robotics*, Vol. 34, No. 2, pp. 229–240, 2017.
- [3] A Hall, A Barton, and A Heward. Google lunar xprize: Newspace entrepreneurs go back to the moon-for good. In *European Planetary Science Congress*, Vol. 8, 2013.
- [4] Luca Iocchi, Dirk Holz, Javier Ruiz del Solar, Komei Sugiura, and Tijn van der Zant. Robocup@home: Analysis and results of evolving competitions for domestic and service robots. *Artificial Intelligence*, Vol. 229, pp. 258 – 281, 2015.
- [5] 五十棲隆勝, 赤地一彦, 平田勝, 金子健二, 梶田秀司, 比留川博久. ヒューマノイドロボット hrp-2 の開発. *日本ロボット学会誌*, Vol. 22, No. 8, pp. 1004–1012, 2004.
- [6] Kei Okada, Mitsuharu Kojima, Yuichi Sagawa, Toshiyuki Ichino, Kenji Sato, and Masayuki Inaba. Vision based behavior verification system of humanoid robot for daily environment tasks. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pp. 7–12. IEEE, 2006.
- [7] Hiroyasu Iwata and Shigeki Sugano. Design of human symbiotic robot twenty-one. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 580–586. IEEE, 2009.
- [8] Yoshiaki Sakagami, Ryujin Watanabe, Chiaki Aoyama, Shinichi Matsunaga, Nobuo Higaki, and Kikuo Fujimura. The intelligent asimo: System overview and integration. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, Vol. 3, pp. 2478–2483. IEEE, 2002.
- [9] François Félix Ingrand, Raja Chatila, Rachid Alami, and Frédéric Robert. Prs: A high level supervision and control language for autonomous mobile robots. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, Vol. 1, pp. 43–49. IEEE, 1996.
- [10] D. Mcdermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, and D. Wilkins. Pddl - the planning domain definition language. Technical Report

- TR-98-003, Yale Center for Computational Vision and Control,, 1998.
- [11] Fernando Alonso Amo, A Gómez-Pérez, Genoveva López Gómez, and C Montes. An expert systems for homeopathic glaucoma treatment (seho). *Expert Systems with applications*, Vol. 8, No. 1, pp. 89–99, 1995.
- [12] 柳吉沫, 志村正道ほか. 故障診断用エキスパートシステムにおける知識獲得. *人工知能学会誌*, Vol. 1, No. 1, pp. 93–100, 1986.
- [13] Pr2 overview. <http://www.willowgarage.com/pages/pr2/overview>.
- [14] Michael Beetz, Ulrich Klank, Ingo Kresse, Alexis Maldonado, Lorenz Mösenlechner, Dejan Pangercic, Thomas Rühr, and Moritz Tenorth. Robotic roommates making pancakes. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pp. 529–536. IEEE, 2011.
- [15] Lars Kunze, Michael Beetz, Manabu Saito, Haseru Azuma, Kei Okada, and Masayuki Inaba. Searching objects in large-scale indoor environments: A decision-theoretic approach. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 4385–4390. Citeseer, 2012.
- [16] 江藤文夫伊藤利之. 新版 日常生活活動 (ADL) 評価と支援の実際. 医歯薬出版株式会社, 2010.
- [17] John Eric Anderson. Constraint-directed improvisation for everyday activities. 1995.
- [18] Rodney Brooks. A robust layered control system for a mobile robot. *IEEE journal on robotics and automation*, Vol. 2, No. 1, pp. 14–23, 1986.
- [19] Michel Ingham, Robert Ragno, and Brian C Williams. A reactive model-based programming language for robotic space explorers.
- [20] Kei Okada, Yohei Kakiuchi, Haseru Azuma, Hiroyuki Mikita, Kazuto Murase, and Masayuki Inaba. Task compiler: Transferring high-level task description to behavior state machine with failure recovery mechanism. In *ICRA Workshop on Combining Task and Motion Planning*. Citeseer, 2013.
- [21] 井上博允. ロボット言語の研究課題. *日本ロボット学会誌*, Vol. 2, No. 2, pp. 87–90, 1984.
- [22] Peter M Will and David D Grossman. An experimental system for computer controlled mechanical assembly. *IEEE Transactions on computers*, No. 9, pp. 879–888, 1975.
- [23] Bruce Shimano. Val: a versatile robot programming and control system. In *Computer Software and Applications Conference, 1979. Proceedings. COMPSAC 79. The IEEE Computer Society's Third International*, pp. 878–883. IEEE, 1979.
- [24] RJ Popplestone, AP Ambler, and I Bellos. Rapt: A language for describing assemblies. *Industrial Robot: An International Journal*, Vol. 5, No. 3, pp. 131–137, 1978.
- [25] Center for the Study of Language, Information (US). Center for the Study of Lan-

- guage, Information, and J Barwise. *Situations and small worlds*. 1989.
- [26] 秀之中島. 状況に依存した推論 (<小特集>「状況意味論」). 人工知能学会誌, Vol. 7, No. 3, pp. 392–398, may 1992.
- [27] 明典阿部. 状況理論に基づく学習. Technical Report 92(1992-ICS-085), NTT コミュニケーション科学研究所, nov 1992.
- [28] 明典阿部. 状況理論に基づく知識獲得における状況の再構成. 全国大会講演論文集, 第47回, pp. 27–28, sep 1993.
- [29] 鈴木浩之ほか. 状況理論の基礎概念 (<小特集>「状況意味論」). 人工知能学会誌, Vol. 7, No. 3, pp. 385–391, 1992.
- [30] Richard C Atkinson and Richard M Shiffrin. Human memory: A proposed system and its control processes. *The psychology of learning and motivation*, Vol. 2, pp. 89–195, 1968.
- [31] Larry R Squire and Stuart M Zola. Structure and function of declarative and non-declarative memory systems. *Proceedings of the National Academy of Sciences*, Vol. 93, No. 24, pp. 13515–13522, 1996.
- [32] Endel Tulving. Episodic and semantic memory 1. *Organization of Memory*. London: Academic, Vol. 381, No. e402, p. 4, 1972.
- [33] Scott Niekum, Sachin Chitta, Andrew G Barto, Bhaskara Marthi, and Sarah Osentoski. Incremental semantically grounded learning from demonstration. In *Robotics: Science and Systems*, Vol. 9, 2013.
- [34] Kam S Tso, Myron Hecht, Neville Marzwell, et al. Fault-tolerant robotic system for critical applications. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pp. 691–696. IEEE, 1993.
- [35] 植田亮平. 失敗予測評価に基づく行動動作計画機能を有するヒューマノイドの統合システム構成法. 修士論文, 東京大学情報理工学系研究科 創造情報学専攻, 2011.
- [36] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.
- [37] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. ROS: an open-source Robot Operating System. In *ICRA workshop on Open-Source Software*, 2009.
- [38] K.A. Wyrobek, E.H. Berger, H.F.M. Van der Loos, and J.K. Salisbury. Towards a personal robotics development platform: Rationale and design of an intrinsically safe personal robot. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 2165–2170, 2008.
- [39] Willow Garage. PR2 user manual rev.307, 2010.
- [40] Rosen Diankov. *Automated Construction of Robotic Manipulation Programs*. PhD thesis, Carnegie Mellon University, Robotics Institute, August 2010.
- [41] OpenRAVE Tutorials inverse reachability. <http://openrave.org/docs/0.6.6/>

- openravepy/examples.tutorial_inversereachability/.
- [42] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE MultiMedia*, Vol. 19, No. 2, pp. 4–10, April 2012.
- [43] C. P. Wright and E. Zadok. Unionfs: Bringing file systems together. *Linux Journal*, Vol. 2004, No. 128, pp. 24–29, December 2004.
- [44] Industrial Communication NetworksFieldbus SpecificationsPart 312: Data-Link Layer Service DefinitionPart 412: Data-Link Layer Protocol SpecificationType 12 Elements, 2007.
- [45] Tully Foote. Ros tf. <http://wiki.ros.org/tf>.
- [46] 齊藤学. フロア間自律移動を行う学習推論機能を有する認識行動ロボットの構成法. 東京大学 情報理工学系研究科 創造情報学専攻 修士論文, 2012.
- [47] E Marder-Eppstein and V Pradeep. Ros actionlib. <http://wiki.ros.org/actionlib>.
- [48] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE transactions on Robotics*, Vol. 23, No. 1, pp. 34–46, 2007.
- [49] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, Vol. 1, No. 1, pp. 269–271, 1959.
- [50] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, Vol. 4, No. 1, pp. 23–33, 1997.
- [51] James J Kuffner and Steven M LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, Vol. 2, pp. 995–1001. IEEE, 2000.
- [52] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, Vol. 34, No. 3, pp. 189–206, 2013.
- [53] 垣内洋平, 岡田慧, 稲葉雅幸. 色ヒストグラムを用いたカラー距離画像の高速位置合わせによる未知物体のモデリングを伴う生活環境地図作成. *日本ロボット学会誌*, Vol. 29, No. 8, pp. 694–701, 2011.
- [54] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, Vol. 1, pp. 886–893. IEEE, 2005.
- [55] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, Vol. 60, No. 2, pp. 91–110, 2004.
- [56] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pp. 2564–2571. IEEE, 2011.

- [57] Kei Okada, Mitsuharu Kojima, Satoru Tokutsu, Toshiaki Maki, Yuto Mori, and Masayuki Inaba. Multi-cue 3d object recognition in knowledge-based vision-guided humanoid robot system. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 3217–3222. IEEE, 2007.
- [58] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pp. 91–99, 2015.
- [59] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.
- [60] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pp. 21–37. Springer, 2016.
- [61] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *arXiv preprint arXiv:1605.06211*, 2016.
- [62] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.
- [63] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network.
- [64] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pp. 2980–2988. IEEE, 2017.
- [65] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, Vol. 1611, pp. 586–607. International Society for Optics and Photonics, 1992.
- [66] Jonathan Bohren and Steve Cousins. The smach high-level executive [ros news]. *IEEE Robotics & Automation Magazine*, Vol. 17, No. 4, pp. 18–20, 2010.
- [67] Vijay Pradeep. Ros pr2 mannequin mode. http://wiki.ros.org/pr2_mannequin_mode.
- [68] John E Laird, Allen Newell, and Paul S Rosenbloom. Soar: An architecture for general intelligence. *Artificial intelligence*, Vol. 33, No. 1, pp. 1–64, 1987.
- [69] John R Anderson, Michael Matessa, and Christian Lebiere. Act-r: A theory of higher level cognition and its relation to visual attention. *Human-Computer Interaction*, Vol. 12, No. 4, pp. 439–462, 1997.
- [70] Yuki Furuta, Yuto Inagaki, Kei Okada, and Masayuki Inaba. Self-improving robot action management system with probabilistic graphical model based on task related memories. In *Proceedings of the 14th International Conference on Intelligent*

- Autonomous Systems*, pp. 811–823, 5 2016.
- [71] 古田悠貴. 長期タスク行動試行と改善機構に基づく家事支援ロボットシステムに関する研究. 修士論文, 東京大学, 2016.
- [72] Kristina Chodorow. *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage*. " O'Reilly Media, Inc.", 2013.
- [73] Peter Deutsch. Gzip file format specification version 4.3. Technical report, 1996.
- [74] Nick Hawes, Christopher Burbridge, Ferdian Jovan, Lars Kunze, Bruno Lacerda, Lenka Mudrova, Jay Young, Jeremy Wyatt, Denise Hebesberger, Tobias Kortner, et al. The strands project: Long-term autonomy in everyday environments. *IEEE Robotics & Automation Magazine*, Vol. 24, No. 3, pp. 146–156, 2017.
- [75] Alexandra Kirsch. Robot learning language integrating programming and learning for cognitive systems. *Robotics and Autonomous Systems*, Vol. 57, No. 9, pp. 943–954, 2009.
- [76] 桑山清丈, 加藤昇平, 伊藤英則. 経験からの知識獲得に基づいたヒューマノイドの運動制御. 日本ロボット学会誌, Vol. 25, No. 3, pp. 402–409, 2007.
- [77] Jan Winkler and Michael Beetz. Robot action plans that form and maintain expectations. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 5174–5180. IEEE, 2015.
- [78] J. Ross Quinlan. Induction of decision trees. *Machine learning*, Vol. 1, No. 1, pp. 81–106, 1986.
- [79] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Unsupervised learning. In *The elements of statistical learning*, pp. 485–585. Springer, 2009.
- [80] Charles J Alpert and So-Zen Yao. Spectral partitioning: the more eigenvectors, the better. In *Proceedings of the 32nd annual ACM/IEEE Design Automation Conference*, pp. 195–200. ACM, 1995.
- [81] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. In *The adaptive web*, pp. 291–324. Springer, 2007.
- [82] Nichola Abdo, Cyrill Stachniss, Luciano Spinello, and Wolfram Burgard. Robot, organize my shelves! tidying up objects by predicting user preferences. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 1557–1564. IEEE, 2015.
- [83] Aude Billard, Sylvain Calinon, Ruediger Dillmann, and Stefan Schaal. Robot programming by demonstration. In *Springer handbook of robotics*, pp. 1371–1394. Springer, 2008.
- [84] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, Vol. 57, No. 5, pp. 469–483, 2009.
- [85] Yasuo Kuniyoshi, Masayuki Inaba, and Hirochika Inoue. Learning by watching:

- Extracting reusable task knowledge from visual observation of human performance. *IEEE transactions on robotics and automation*, Vol. 10, No. 6, pp. 799–822, 1994.
- [86] Freek Stulp, Andreas Fedrizzi, Lorenz Mösenlechner, and Michael Beetz. Learning and reasoning with action-related places for robust mobile manipulation. *Journal of Artificial Intelligence Research*, Vol. 43, pp. 1–42, 2012.
- [87] Dejan Pangercic, Benjamin Pitzer, Moritz Tenorth, and Michael Beetz. Semantic object maps for robotic housework-representation, acquisition and use. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 4644–4651. IEEE, 2012.
- [88] Mitsuharu Kojima, Kei Okada, and Masayuki Inaba. Manipulation and recognition of objects incorporating joints by a humanoid robot for daily assistive tasks. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pp. 1564–1569. IEEE, 2008.
- [89] 垣内洋平. 三次元視覚による認識操作モデルの獲得機能を有するヒューマノイドシステムの研究. PhD thesis, 東京大学情報理工学系研究科 知能機械情報学専攻, 2011.
- [90] 東馳, 垣内洋平, 岡田慧, 稲葉雅幸. 対話画面インタフェースでのマルチタッチジェスチャ指示と操作候補提示に基づくロボット遠隔操作システム. *日本ロボット学会誌*, Vol. 31, No. 4, pp. 409–415, 2013.
- [91] Haseru Chen, Yohei Kakiuchi, Manabu Saito, Kei Okada, and Masayuki Inaba. View-based multi-touch gesture interface for furniture manipulation robots. In *Advanced Robotics and its Social Impacts (ARSO), 2011 IEEE Workshop on*, pp. 39–42. IEEE, 2011.
- [92] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels. Technical report, 2010.
- [93] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.
- [94] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017.
- [95] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, Vol. 24, No. 6, pp. 381–395, 1981.
- [96] SB Damelin and NS Hoang. On surface completion and image inpainting by bi-harmonic functions: Numerical aspects. *International Journal of Mathematics and Mathematical Sciences*, Vol. 2018, , 2018.
- [97] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, Vol. 4, No. 5, pp. 1–17, 1964.
- [98] Yuki Furuta, Kentaro Wada, Masaki Murooka, Shunichi Nozawa, Yohei Kakiuchi,

- Kei Okada, and Masayuki Inaba. Transformable semantic map based navigation using autonomous deep learning object segmentation. In *Proceedings of the 2016 IEEE-RAS International Conference on Humanoid Robots (Humanoids 2016)*, pp. 614–620, November 2016.
- [99] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge & Data Engineering*, No. 9, pp. 1263–1284, 2008.
- [100] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, Vol. 88, No. 2, pp. 303–338, 2010.
- [101] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. Understanding the exploding gradient problem. *CoRR*, *abs/1211.5063*, 2012.
- [102] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [103] Yuki Furuta, Kei Okada, Yohei Kakiuchi, and Masayuki Inaba. An everyday robotic system that maintains local rules using semantic map based on long-term episodic memory. In *Proceedings of The 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1072–1079, october 2018.
- [104] Alexander JB Trevor, Suat Gedikli, Radu B Rusu, and Henrik I Christensen. Efficient organized point cloud segmentation with connected components.
- [105] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, Vol. 18, No. 9, pp. 509–517, 1975.
- [106] Fetch Mobile Manipulator and Freight Mobile Robot Base. <https://fetchrobotics.com/robotics-platforms/>.
- [107] Judea Pearl and Stuart Russell. *Bayesian networks*. Computer Science Department, University of California, 1998.
- [108] 稲嶋哲也. 人間とロボットの対話に基づく行動知能形成に関する研究. 博士論文, 東京大学, 2000.
- [109] roomba. <http://www.irobot.com/For-the-Home/Vacuum-Cleaning/Roomba.aspx>.
- [110] Nicola Muscettola, P Pandurang Nayak, Barney Pell, and Brian C Williams. Remote agent: To boldly go where no ai system has gone before. *Artificial intelligence*, Vol. 103, No. 1-2, pp. 5–47, 1998.
- [111] Patrick Doherty, Gösta Granlund, Krzysztof Kuchcinski, Erik Sandewall, Klas Nordberg, Erik Skarman, and Johan Wiklund. The witas unmanned aerial vehicle project.
- [112] Reid Simmons, Richard Goodwin, Sven Koenig, and Greg Armstrong. Xavier: An autonomous mobile robot on the web. *Beyond Webcams: an introduction to online robots*, p. 81.
- [113] Reid Simmons, Dani Goldberg, Adam Goode, Michael Montemerlo, Nicholas Roy,

- Alan C Schultz, Myriam Abramson, Ian Horswill, David Kortenkamp, and Bruce Maxwell. Grace: An autonomous robot for the aai robot challenge. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA, 2003.
- [114] Alain Colmerauer and Philippe Roussel. The birth of prolog. In *History of programming languages—II*, pp. 331–367. ACM, 1996.
- [115] Malte Helmert. The fast downward planning system. *Journal of Artificial Intelligence Research*, Vol. 26, pp. 191–246, 2006.
- [116] Jörg Hoffmann and Bernhard Nebel. The ff planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, Vol. 14, pp. 253–302, 2001.
- [117] Silvia Richter and Matthias Westphal. The lama planner: Guiding cost-based any-time planning with landmarks. *Journal of Artificial Intelligence Research*, Vol. 39, pp. 127–177, 2010.
- [118] Alfonso Gerevini and Ivan Serina. Lpg: A planner based on local search for planning graphs with action costs. 2002.
- [119] Richard E. Fikes and Nils J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. In *Proceedings of the 2Nd International Joint Conference on Artificial Intelligence*, IJCAI’71, pp. 608–620, San Francisco, CA, USA, 1971. Morgan Kaufmann Publishers Inc.
- [120] E. P. D. Pednault. Formulating multiagent, dynamic-world problems in the classical planning framework. In M. P. Georgeff and A. L. Lansky, editors, *Reasoning about Actions and Plans*, pp. 47–82. Kaufmann, Los Altos, CA, 1987.
- [121] Michael Gelfond and Vladimir Lifschitz. Action languages. *Electronic Transactions on AI*, Vol. 3, No. 16, 1998.
- [122] Maria Fox and Derek Long. Pddl2. 1: An extension to pddl for expressing temporal planning domains. *Journal of artificial intelligence research*, Vol. 20, pp. 61–124, 2003.
- [123] Richard E Fikes, Peter E Hart, and Nils J Nilsson. Learning and executing generalized robot plans. *Artificial intelligence*, Vol. 3, pp. 251–288, 1972.
- [124] Vahid Mokhtari, Luís Seabra Lopes, and Armando J Pinho. Experience-based robot task learning and planning with goal inference. 2016.
- [125] Raphaël Lallement, Lavindra De Silva, and Rachid Alami. Hatp: An htn planner for robotics. *arXiv preprint arXiv:1405.5345*, 2014.
- [126] 古田悠貴, 岡田慧, 稲葉雅幸. 日常生活支援ロボットにおける長期記憶蓄積に基づく文脈適応行動計画・動作システムを用いた片付けタスクの構成法. 第 32 回 人工知能学会全国大会, 2018, pp. 4L2–02, Jun 2018.
- [127] Kimitoshi Yamazaki, Ryohei Ueda, Shunichi Nozawa, Yuto Mori, Toshiaki Maki, Naotaka Hatao, Kei Okada, and Masayuki Inaba. Tidying and cleaning rooms

- using a daily assistive robot—an integrated system for doing chores in the real world. *Paladyn, Journal of Behavioral Robotics*, Vol. 1, No. 4, pp. 231–239, 2010.
- [128] Daniel Nyga, Subhro Roy, Rohan Paul, Daehyung Park, Mihai Pomarlan, Michael Beetz, and Nicholas Roy. Grounding Robot Plans from Natural Language Instructions with Incomplete World Knowledge. In *2nd Conference on Robot Learning (CoRL 2018)*, Zurich, Switzerland, 2018.
- [129] Sylvain Calinon, Florent Guenter, and Aude Billard. On learning, representing, and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 37, No. 2, pp. 286–298, 2007.
- [130] Edward T Hall. *Beyond culture*. 1976.
- [131] Paul E Rybski, Jeremy Stolarz, Kevin Yoon, and Manuela Veloso. Using dialog and human observations to dictate tasks to a learning robot assistant. *Intelligent Service Robotics*, Vol. 1, No. 2, pp. 159–167, 2008.
- [132] J Thomason, S Zhang, R Mooney, and P Stone. Learning to interpret natural language commands through human-robot dialog. In *IJCAI*, pp. 1923–1929, 2015.
- [133] Akinobu Lee, Tatsuya Kawahara, and Kiyohiro Shikano. Julius—an open source real-time large vocabulary recognition engine. 2001.
- [134] Willie Walker, Paul Lamere, Philip Kwok, Bhiksha Raj, Rita Singh, Evandro Gouvea, Peter Wolf, and Joe Woelfel. Sphinx-4: A flexible open source framework for speech recognition. 2004.
- [135] Johan Schalkwyk, Doug Beeferman, Françoise Beaufays, Bill Byrne, Ciprian Chelba, Mike Cohen, Maryam Kamvar, and Brian Strope. “your word is my command” : Google search by voice: a case study. In *Advances in speech recognition*, pp. 61–90. Springer, 2010.
- [136] Karen Zita Haigh and Manuela M Veloso. Planning, execution and learning in a robotic agent. 1998.
- [137] 林久志, 十倉征司, 尾崎文夫, 長谷川哲夫. Dynagent : 割込み htn プランニングエージェント. *知能と情報*, Vol. 21, No. 5, pp. 856–869, 2009.
- [138] Yuki Furuta, Kazuhiro Sasabuchi, Yusuke Niitani, Kotaro Nagahama, Hiroaki Yaguchi, Kei Okada, and Masayuki Inaba. Bring me manju from the drawer: Task acquisition framework under incompleteness and ambiguity using interaction and semantic knowledge-enabled perception. In *2017 IEEE/RSJ IROS Workshop Machine Learning Methods for High-Level Cognitive Capabilities in Robotics*, 2017.
- [139] J Williams, E Kamal, M Ashour, H Amr, J Miller, and G Zweig. Fast and easy language understanding for dialog systems with microsoft language understanding intelligent service (luis). In *Proceedings of 2015 SIGDIAL Conference, Prague. ACL – Association for Computational Linguistics*, September 2015.
- [140] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N

- Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.
- [141] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Mihai Dolha, and Michael Beetz. Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems*, Vol. 56, No. 11, pp. 927–941, 2008.
- [142] Command Generator for Robocup@Home. <https://github.com/kyordhel/GPSRCmdGen>.

謝辞

本論文は筆者が東京大学大学院創造情報学系研究科博士課程に在籍中、稲葉雅幸教授のご指導のもと情報システム工学研究室において行った研究をまとめたものである。

稲葉雅幸教授には学部4年生から6年間の長きにわたりご指導いただきました。先生の広大な視点から見た研究に関する展望や意見は、うかがうたびに筆者を研究活動へと駆り立てる強い動機付けとなったのみならず、研究者、教育者として筆者のことを指導・サポートしていただきました。ありがとうございました。

岡田慧教授には研究のみならず、技術的な課題や議論について直接ご指導いただく機会が多く、煩雑なロボットのシステムを前に右往左往する筆者に対して道を示していただきました。先生の底の見えないオプティミスティックながらも本質を逃さぬご助言や議論は学ぶ所大変多く、とても楽しい時間でした。ありがとうございました。

その他、ご卒業された先輩・後輩を含め、筆者が研究室でともに過ごし、喧々諤々の議論や夜を徹してのロボットの修理作業など非常に多くの時間を通じて、時に辛くも、厳しいロボット研究三昧の生活を楽しいものにしてくださったすべての方々のサポートのおかげでこの論文を執筆し終えることができました。大変感謝しております。

最後に、都合9年間にも及ぶ長い長い学生生活を理解し、筆者のことを精神的かつ経済的にも支えてくださった両親、家族、妻、そして、筆者を大学と自宅との往復のみの生活から連れ出し、新鮮な刺激を与えてくれた友人たちに心から感謝いたします。

平成30年11月 古田悠貴

付録 A

Appendix

A.1 PDDL を用いた日常生活支援行動計画のための環境状態記述

第 8 章における実験で用いた行動計画における環境状態記述は以下であった。

```
1 (define (domain room-domain)
2   (:requirements :adl :typing :action-costs :negative-preconditions)
3   (:types spot arm item person)
4   (:constants cleaner - item somewhere - spot someone - person)
5   (:predicates
6     (robot-at ?at - spot)
7     (sure-on ?obj - item ?at - spot)
8     (on ?obj - item ?at - spot)
9     (sure-see ?who - person ?at - spot)
10    (see ?who - person ?at - spot)
11    (delivered ?obj - item ?who - person)
12    (asked ?who - person)
13    (onhand ?obj - item ?arm - arm)
14    (open ?at - spot)
15    (placable ?at - spot)
16    (occupied ?at - spot)
17    (sure-placable ?obj - item ?at - spot)
18    (openable ?arm - arm ?at - spot)
19    (clean))
20  (:functions
```

```
21 (total-cost)
22 (move-cost ?from ?to - spot)
23 (arm-cost ?arm - arm)
24 (obj-cost ?obj - item ?at - spot)
25 (find-cost)
26 (manip-cost))
27 (:action
28 "move-to"
29 :parameters
30 (?from ?to - spot)
31 :precondition
32 (and (robot-at ?from) (not (open ?from)))
33 :effect
34 (and
35 (not (robot-at ?from))
36 (robot-at ?to)
37 (forall (?obj - item) (not (sure-on ?obj ?from)))
38 (increase (total-cost) (move-cost ?from ?to))))
39 (:action
40 "find-object"
41 :parameters
42 (?obj - item ?at - spot)
43 :precondition
44 (and (robot-at ?at) (on ?obj ?at) (placable ?at))
45 :effect
46 (and
47 (sure-on ?obj ?at)
48 (not (on ?obj ?at))
49 (increase (total-cost) (find-cost))))
50 (:action
51 "pick"
52 :parameters
53 (?obj - item ?arm - arm ?at - spot)
54 :precondition
55 (and
```

```

56     (forall (?other - item) (not (onhand ?other ?arm)))
57     (robot-at ?at)
58     (sure-on ?obj ?at)
59     (placable ?at))
60 :effect
61 (and
62     (onhand ?obj ?arm)
63     (not (sure-on ?obj ?at))
64     (not (on ?obj ?at))
65     (not (occupied ?at))
66     (increase (total-cost) (arm-cost ?arm))
67     (increase (total-cost) (obj-cost ?obj ?at))))
68 (:action
69     "find-placement"
70     :parameters
71     (?obj - item ?arm - arm ?at - spot)
72     :precondition
73     (and (onhand ?obj ?arm) (robot-at ?at) (placable ?at))
74     :effect
75     (and (sure-placable ?obj ?at) (increase (total-cost) (find-cost))))
76 (:action
77     "place"
78     :parameters
79     (?obj - item ?arm - arm ?at - spot)
80     :precondition
81     (and
82         (onhand ?obj ?arm)
83         (robot-at ?at)
84         (placable ?at)
85         (sure-placable ?obj ?at)
86         (not (occupied ?at)))
87     :effect
88     (and
89         (not (onhand ?obj ?arm))
90         (on ?obj ?at)

```



```
91      (not (sure-placable ?obj ?at))
92      (increase (total-cost) (arm-cost ?arm))
93      (increase (total-cost) (obj-cost ?obj ?at))))
94 (:action
95  "open-door"
96  :parameters
97  (?arm - arm ?at - spot)
98  :precondition
99  (and
100   (forall (?obj - item) (not (onhand ?obj ?arm)))
101   (robot-at ?at)
102   (not (placable ?at))
103   (not (open ?at))
104   (openable ?arm ?at))
105  :effect
106  (and (placable ?at) (open ?at) (increase (total-cost) (manip-cost))))
107 (:action
108  "close-door"
109  :parameters
110  (?arm - arm ?at - spot)
111  :precondition
112  (and
113   (forall (?obj - item) (not (onhand ?obj ?arm)))
114   (robot-at ?at)
115   (openable ?arm ?at)
116   (open ?at))
117  :effect
118  (and
119   (not (placable ?at))
120   (not (open ?at))
121   (increase (total-cost) (manip-cost))))
122 (:action
123  "clean-room"
124  :parameters
125  (?arm - arm ?at - spot)
```

```

126 :precondition
127 (and (onhand cleaner ?arm) (robot-at ?at))
128 :effect
129 (and (clean) (not (robot-at ?at)) (robot-at somewhere)))
130 (:action
131 "find-person"
132 :parameters
133 (?who - person ?at - spot)
134 :precondition
135 (and (robot-at ?at) (see ?who ?at) (placable ?at))
136 :effect
137 (and (sure-see ?who ?at) (not (see ?who ?at))))
138 (:action
139 "deliver"
140 :parameters
141 (?obj - item ?arm - arm ?who - person ?at - spot)
142 :precondition
143 (and
144     (onhand ?obj ?arm)
145     (robot-at ?at)
146     (sure-see ?who ?at)
147     (placable ?at))
148 :effect
149 (and (not (onhand ?obj ?arm)) (delivered ?obj ?who)))
150 (:action
151 "ask"
152 :parameters
153 (?who - person ?at - spot)
154 :precondition
155 (and (robot-at ?at) (sure-see ?who ?at) (placable ?at))
156 :effect
157 (and (asked ?who)))
158 (:action
159 "find-object_f"
160 :parameters

```

```
161  (?obj - item ?at - spot)
162  :precondition
163  (and (robot-at ?at) (on ?obj ?at) (placable ?at))
164  :effect
165  (and (not (on ?obj ?at)) (increase (total-cost) (find-cost))))
166 (:action
167  "pick_f"
168  :parameters
169  (?obj - item ?arm - arm ?at - spot)
170  :precondition
171  (and
172    (forall (?other - item) (not (onhand ?other ?arm)))
173    (robot-at ?at)
174    (sure-on ?obj ?at)
175    (placable ?at))
176  :effect
177  (and
178    (on ?obj ?at)
179    (not (sure-on ?obj ?at))
180    (increase (total-cost) (arm-cost ?arm))
181    (increase (total-cost) (obj-cost ?obj ?at))))
182 (:action
183  "find-placement_f"
184  :parameters
185  (?obj - item ?arm - arm ?at - spot)
186  :precondition
187  (and (onhand ?obj ?arm) (robot-at ?at) (placable ?at))
188  :effect
189  (and
190    (not (placable ?at))
191    (not (openable ?arm ?at))
192    (occupied ?at)
193    (increase (total-cost) (find-cost))))
194 (:action
195  "find-person_f"
```

```
196 :parameters
197 (?who - person ?at - spot)
198 :precondition
199 (and (robot-at ?at) (see ?who ?at) (placable ?at))
200 :effect
201 (and (not (see ?who ?at))))
```