

クラウドロボティクスにおける ロボットの消費電力を考慮したタスク分散処理

中川幸子^{†1} 成田雅彦^{†1} 土屋陽介^{†1} 加藤由花^{†1}

近年、インターネット分野とロボット分野の融合が加速しており、クラウド環境への適用を前提とした様々なプラットフォームが提案されている。ただし、アクチュエーションを伴うロボットサービスを対象とした場合、実世界に発生する仮想環境との不整合の吸収や、同期タイミング、タスク実行中の別タスク発生などの不確実な実環境をクラウド側で考慮する仕組みが必要である。本稿では、ロボットサービス向けプロトコル仕様である RSNP (Robot Service Network Protocol) を利用し、ケアサービス分野へのロボットサービス適用をめざした複数ロボットでのタスク分散処理によるケアロボットサービスを提案する。サービスではロボットの消費電力を考慮し、ユーザによるロボット遠隔操作を可能にしながら、複数ロボットの連携によるサービス継続を行う。さらにプロトタイプシステムを実装し、その有効性を検証する。

A distributed processing considering power consumption for Cloud-based Robot Services

SACHIKO NAKAGAWA^{†1} MASAHIKO NARITA^{†1}
YOSUKE TSUCHIYA^{†1} YUKA KATO^{†1}

1. はじめに

少子高齢化、労働力不足等の社会問題が顕在化する中、ロボットは商業施設や公共施設等の公共空間や、家庭・職場などの生活空間に導入されつつあり、病院・介護・福祉サービス等のケアサービス分野へのロボット活用への期待が寄せられている。とくにサービスロボット分野では商業施設や福祉施設における対人サービスがある。この場合、サービス対象は人であり、ロボットには高度なコミュニケーションと人とのインタラクションが要求される。一方で、公共・生活空間のロボットには、従来どおりの連続的な繰り返し作業の実行による労働負担の軽減や、より便利で快適な社会への貢献も望まれる。掃除ロボットや荷物運搬ロボットはこの代表例であり、この場合は必ずしもサービス対象は人ではなく、人的労働の替わりとして物理的な作業を達成する能力が要求される。とくに後者の場合、ロボットへの作業指示者であるユーザは、ロボット稼働現場ではなく遠隔から複数のロボットを利用する状況が考えられる。このような状況下において、遠隔ユーザへの情報提供とユーザによる遠隔操作を可能とすることは、ロボットをネットワークに接続する利点の一つである。

ロボットのネットワーク化については、これまでも RTミドルウェア[1]や ROS[2]などの活動があったが、近年のロボットの汎用化と小型化やロボット周辺のオープン技術に発展に対応して、クラウドコンピューティングを活用したロボットサービスの研究がすすめられている。例えば、

2011年の Google/IO では、ROS-java を搭載した Android-OS の携帯端末とロボットとの連携デモンストレーションによるクラウドロボティクスのコンセプト[3]や、Kamei らによるクラウドネットワークロボティクスの概念[4]が発表された。このような背景のもと、筆者らはロボットサービスのインターネット化を目指して RSNP (Robot Service Network Protocol) [5][6]の研究をおこなってきた[7]。RSNP は、2004年に設立された業界団体である RSi (Robot Service Initiative) [8]により仕様化されている。RSi は、インターネットを活用したロボットサービス向けのソフトウェア基盤の仕様化としてライブラリ開発、高信頼通信、他のプラットフォームとの連携や、ロボットサービスをクラウド基盤に適用するための研究開発環境とそのサービス提供手法として RSi-Cloud[9][10]を提案している。本稿ではこの RSNP を利用し、ケアサービス分野へのロボットサービス適用をめざした Publish/Subscribe 型分散処理によるケアロボットサービスを提案する。サービスは複数のタスクの組み合わせで構成されており、これを複数ロボット間でタスク分散する。解決すべき課題は、次の3点である。

- (1) 複数のロボットをサービスに接続し、ロボットのアクチュエーションを考慮したタスク分散処理を可能にする。これにより、インターネット経由での施設管理者 (User) へ情報提供と人的労働の替わりとしての物理的な作業の双方を行う仕組みを実現する。
- (2) タスク処理中のロボットに対する User の遠隔操作を可能とする。また、タスク処理中に、別タスクが発生した場合の処理や不確実な実環境への対応が可能な仕組みを構築する。さらに、別タスクの発生により定常タスクを処理するロボット台数が変化する状況下

^{†1} 産業技術大学院大学産業技術研究所
1-10-40 Higashi-Ohi, Shinagawa-Ku, Tokyo 101-0062, Japan

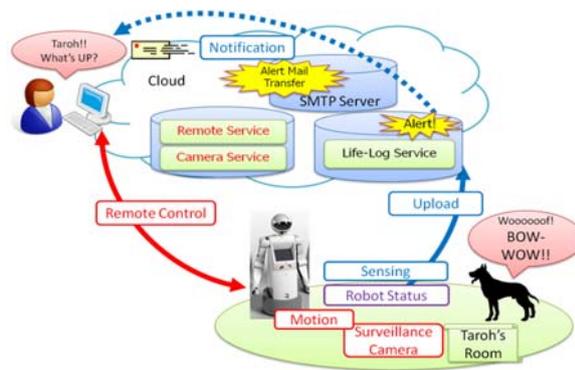


図1 サービスの概要

Figure 1 Overview of a service

でも、適切な再プランニングのできる仕組みを構築し、サービスを継続する。

- (3) ロボットや IT の専門家ではない導入組織における運用コストを最小限にとどめることを目指す。そのため、クラウドによるサービス提供を前提とする。

本稿では、(1) の課題解決のため、タスク分散機構を導入する。実ロボットの走行時には、慣性や滑りや床面の凸凹により生じる誤差の発生があり、インターネット経由でのロボットへのアクセス時には、実世界に発生する仮想環境との不整合の吸収をクラウドで考慮する必要がある。タスク分散機構は、ユーザが要求するサービスの呼出し、最適なロボットの選択、サービスに応じたロボットの走行経路計画、及び同期タイミングを考慮したスケジューリングを行う。(2) の課題解決のため、定常タスク実行中のロボットに発生する別タスクとして、a) User からの遠隔操作の要求の受付と実行(遠隔作業タスク)と、b) ロボットのバッテリー残量低下時の充電行動(充電タスク)を設定する。これは、生活空間で稼働するロボットが電力を使い果たすことで立ち往生した場合、ユーザはロボットの稼働現場まで行き、手でロボットを充電ステーションまで運ぶ必要があり運用コストは増大する、また、この状態はロボットの非専門家には不具合と映ることすらあるため、バッテリー残量低下時の充電プランはサービスロボットに必須のタスクプランと考えたためである。また、バッテリー残量低下時の充電ステーションへの移動行動は、ロボットの実環境稼働時に発生するタスクの一つであり、ロボットサービスに特徴的といえる。(3) の課題を解決するためには、ロボットとクラウド間の通信に RSNP を利用する。また、本稿は、公共施設や生活空間での複数のサービス要求に対応するため、汎用のサービスロボットを利用することを前提とする。さらにプロトタイプシステムを実装した結果を報告する。

2. 関連研究

ロボティクス分野でのクラウド利用では、PaaS 型フレームワークを目指した RoboEarth[11]のクラウドエンジン

rapyuta[12]、M2M/M2C のアーキテクチャを提唱する CloudRobotics[13]、サービス対象の学習と認識のために Google ゴーグル[14]のオブジェクト認識エンジンを Google のクラウドストレージに搭載した把持ロボットサービス[15]、Hadoop によるデータ処理で FastSLAM アルゴリズム計算を行い ROS への適用を目指した DAVinci[16]などが提案されている。これらの提案では、クラウドを介した遠隔操作は考慮されていない。また、ROS と非 ROS の互換ツールである rosbridge[17]による、Web サービスによるロボット遠隔操作ラボが提供されているが、複数ロボットの継続的なサービス提供に伴うタスク分散は考慮されていない。

一方、複数ロボット間におけるタスク割り当てについては、契約ネットプロトコル[18]や群知能ロボット[19]によるマルチエージェントシステム分野での研究は、従来から多く行われている。これらの関心はロボット間の連携や知識共有による一定のタスクの高速な処理やロボット協調による挙動生成の考察に焦点があり、汎用のサービスロボットでのタスク分散の考察は少ない。また、MAMET などの複数モバイル端末間でのタスク分散[20]では、タスク実行中に非定常的な別タスク要求を受け付けるためのアクチュエーションの考慮は対象外である。しかし、日常空間でのサービスでは都合による不確実な要求や、ロボットならではの行動要求を処理しながらサービスを継続する必要がある。とくに、ケアサービス分野では、ロボットによる対象の異常検知やロボットを端末とした患者からの呼出しを契機として、医師や看護師などの施設スタッフがロボットの遠隔操作等により対象とコンタクトを行うなどの利用が想定される。また、ケアサービス分野では、長時間のサービス継続が求められる。

これにクラウドを利用した Web サービスとして対応するため、本稿では、従来の研究成果をとりいれながら、複数ロボットによるタスク実行中に遠隔ユーザによるタスク要求と、ロボットの充電行動が発生する環境下でもサービス継続可能な仕組みを提案する。

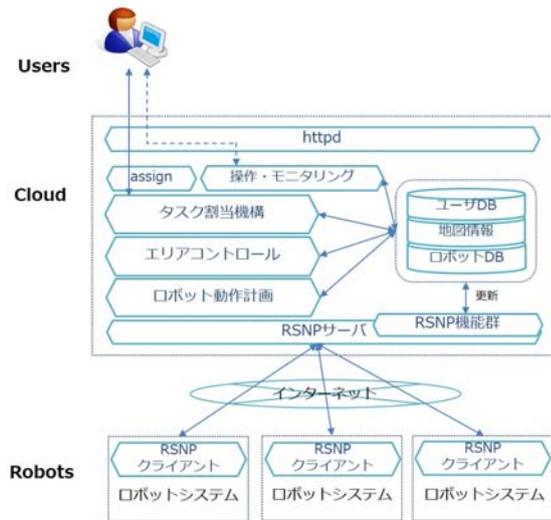


図 2 システム構成の概要
Figure 2 System configuration

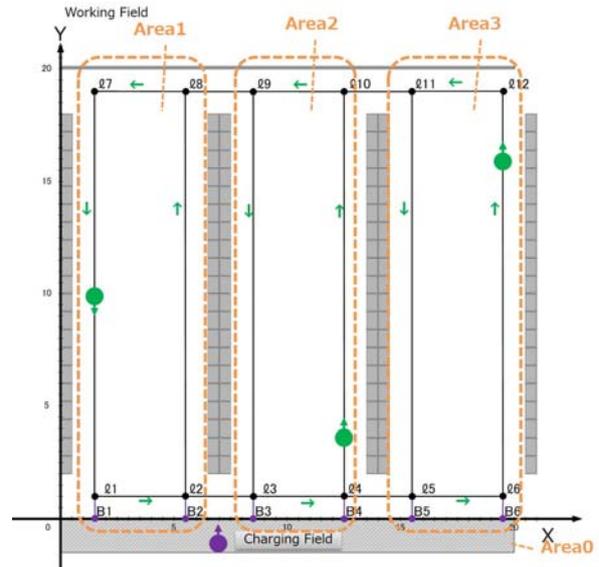


図 3 ロボットの活動フィールド
Figure 3 Running field of robot

3. 想定環境

3.1 サービス概要

本稿では、複数ロボットで構成されたアクチュエーションを伴うタスク実行中のロボットへの別タスク割当てを考察する。そのため、ケアサービスに発生する定常作業と、遠隔地にいるロボットへの作業指示者であるユーザに対しケア対象の情報提供を行う「ケアロボット」として施設に導入されるシナリオを想定する。具体的には、ペット預かり施設で動物の見守り（監視・巡回など）や世話（給餌・掃除など）を行うロボットの自律移動と、遠隔ユーザによるインターネット経由でのロボット操作を組み合わせたペットシッターサービスを設計する。ケア対象はペットである。サービスの概要を図1に示した。

ロボットは汎用のサービスロボットを利用する。ロボットは、カメラによる画像配信機能、センサによる環境データ取得機能、車輪による移動機能を持ち、巡回監視ができる。この監視ログはサーバへ蓄積され、サーバが異常を検知した場合は遠隔ユーザへアラートが配信される。また、台車による荷物運搬機能及びアームとハンドをもつ場合、簡易作業もできる。簡易作業は、給餌、掃除とする。また、巡回タスク実行中のロボットに、ユーザからの遠隔操作要求と、ロボットの充電行動タスクの割当てするためのタスク割当て機構をサーバに導入する。システム構成の概要を図2に示す。タスク割当て機構は、遠隔作業タスクに対しては、要求を実行するのに最適なロボットの選択を行い、遠隔作業タスクをロボットへ割り当てる。充電タスクに対しては、充電が必要なロボットを選択し、充電エリアへの走行経路計画を設定する。この時、充電エリアに格納される

ロボットの台数に応じて、作業フィールドで活動するロボットの台数は増減する。サーバは、ロボットの作業フィールドの地図情報はあらかじめ保持し、作業フィールドでのロボットの稼働台数に応じてロボットの作業エリアの分担、つまり、可動域を動的に設定することで、複数ロボットによるサービス提供の調整を行う。エリア内のロボットの移動モデルはサーバで計画され、この移動モデルをもとにタスクの割当てが行われる。尚、サービスに参加するロボット及びユーザは既知であり、ロボット及びユーザにはあらかじめ識別 ID が付与され、認証によりサービスに参加する。これは RSNP により仕様化されている。

3.2 想定環境

24時間常時稼働の巡回タスクを定常タスク、遠隔操作サービスを遠隔作業タスク、ロボットの充電行動を充電タスクとして定義し、想定環境の詳細を以下に示す。

(1) フィールド：

- ロボットの稼働フィールドを図3に示す。ロボットは $20 * 20$ 単位の正方形の作業エリアで活動する。作業エリアは座標 $(0, 0)$ から座標 $(20, 20)$ の範囲で表され、フィールド内にはペットの部屋が配置されて、ペットは部屋の中にいる。ペットの部屋には $0.8 * 0.6 * 0.5$ 単位程度の檻を想定し、ペットが格納される部屋を $room_i$ ($i = 1, 2, \dots, 120$) とする。
- $room_i$ の座標は、サーバ上の全体地図が保持する。また、 $room_i$ へのアプローチに最適な走行パスも、サーバがあらかじめ保持する。
- 図3の灰色部を充電フィールド (Area0) とする。

表 1 各ロボットのタスク実行状態

Table 1 Task execution state of a robot

稼働エリア	稼働範囲	状態
作業フィールド	Area_1, Area_2,...	巡回タスク実行中, 遠隔作業タスク実行中, 遠隔作業割当て中, 充電要求中, 充電準備中
充電フィールド	Area_0	充電待機中, 充電中, 出動待機中, 巡回タスク割当て中

- 作業フィールドでは n 台のロボットが稼働し, 充電フィールド (Area0) には m 台のロボットが待機する.
 - 各ロボットは, 無線 LAN 通信でサーバと接続するが, ここではフィールドには適切に無線 AP が配置されており, 正常時の無線 LAN 通信の死角はなく常時接続可能とする.
- (2) サーバ :
- サーバは, サービスに参加する各ロボットのタスクの実行状態を, 可動域に応じて保持する. 実行状態のパラメータを, 表 1 に示す.
 - サーバは, あらかじめフィールドの全体の地図情報を持ち, 巡回タスクの走行経路のウェイトポイント座標を端点 ℓ_j ($j = 1, 2, \dots, 12$) として, 充電フィールドへの入口座標を端点 B_g ($g = 1, 3, 5$) として, 出口へ座標を端点 B_s ($s = 2, 4, 6$) として保持する.
 - サーバは巡回タスク, 遠隔操作タスク, 充電タスクの割当て基準と, 移動モデルを保持する. 移動モデルについては (4) を参照のこと.
 - サーバは, 実行タスクに応じたエネルギー消費量をもつ. エネルギー消費については, (6) を参照のこと.
 - サーバは充電タスクの割当て基準となる, バッテリ残量の閾値 (z) をもつ.
 - サーバは, サービス継続を保証する巡回タスク実行中のロボットの下限台数の閾値 $field_min$ をあらかじめ保持する. これは, 遠隔作業タスク完了後のロボットは, タスク完了後に一度充電フィールドに格納されるため, フィールド全体のロボット総数ではなく, 巡回タスク実行中のロボット台数を, タスク割当ての条件とするためである.
- (3) ロボット :
- ロボットはバッテリーで駆動される.
 - ロボットの車輪半径 (r), ロボット重心から車輪までの距離 (d) は既知とする.
 - ロボットはすべて均質であり, それぞれが個別の識別 ID として $robot_id$ をもつ. $robot_id$ を $\{robot_0, robot_1, \dots\}$ とする.
 - 各ロボットは, IEEE802.11 に基づく無線 LAN 通信機能を持ち, サーバが割当てするタスクプランにしたがってタスクを実行する.
 - 各ロボットは, カメラによる画像配信機能, センサによる環境データ取得機能, 車輪による移動機能を持ち,

前進と後進を行うことができる. 車輪は左右独立駆動型であり, その場で指定した角度だけ旋回ができる. 尚, 自律移動中の画像配信用カメラの向きは, ロボット向きと同じとする.

(4) 移動モデル :

- 各ロボットの稼働エリア (Area_n) は, 作業フィールドのロボット台数 (n) に応じて分割される. したがって, 各ロボットの稼働範囲は, ロボット台数に応じて変動する. 稼働範囲内の走行ルートは, スタート位置ポイントからゴール位置ポイントと, その移動距離の走行パスの組み合わせとして, サーバが計画する.
- ロボットは, 端点 B_g より充電フィールド (Area0) へ入り, 端点 B_s より作業フィールドへ出動する. 充電フィールド (Area0) に向かうロボットには, 走行パス ($\ell_g \rightarrow B_g$) がサーバにより与えられる. ゴール位置ポイント B_g に到着したロボットは, 自動的に充電ステーションに接続される. 一方, フィールドへの出動するロボットには, 走行パス ($B_s \rightarrow \ell_s$) が与えられ, スタート位置ポイント B_s より作業フィールドへ出動する.
- ロボットは障害検知が可能であり, 障害 (例えば, 他の移動中ロボット) を進行方向に検知した場合, 障害を自律回避して走行を続ける. また, 端点で外界センサによる自己位置補正を行うことができる.

(5) 遠隔操作 :

- 遠隔ユーザは, 個別にユーザ ID として $user_id$ をもち, $user_id$ を $\{user_0, user_1, \dots\}$ とする.
- 遠隔ユーザは, フィールド内のロボットの遠隔操作権を取得することができる. この時, 遠隔ユーザは特定のロボットを指定するのではなく, 観察したいペットのいる部屋番号 ($room_i$) で指定し, サーバが $room_i$ でのサービスに最適なロボット選出する.
- 遠隔ユーザは, 遠隔操作権をもつロボットに対し, 画像配信を要求する. この時, ロボットのカメラ向きの調整と指定された範囲の移動操作をおこなうことができる.
- サーバは, サービス継続を保証する遠隔操作権の割当て上限数と 1 ユーザの利用可能時間をあらかじめ保持しており, ユーザの遠隔操作要求時にはユーザへ遠隔操作権を付与する.
- 遠隔作業タスク完了後の, ロボットについては, 直後に充電タスクを割当て, 充電フィールドに格納する.

(6) バッテリ消費と充電 :

- 各ロボットは, 実行タスクに応じて一定のエネルギーを消費する. 尚, ここでは簡単のため, 作業フィールドは平地として扱い, 移動によるエネルギー消費量は移動距離に比例する. また, 想定環境では, どのタスクも, 移動, 画像処理, サーバとの通信との組み合わせ

せで構成されるため、フィールドで移動する各ロボットのタスク実行中のエネルギー消費量は定数 e とし、時間経過に比例する。

4. タスク分散機構

本稿では、遠隔ユーザからの操作要求発生時には、巡回タスク実行中のロボットから、各ロボットのバッテリー電力残量と位置情報に応じて最適なロボットを選出し、遠隔作業タスクを割当てる。よって、遠隔ユーザはロボット状態を直接意識することなく、タスク要求を行うのみで最適なロボットを利用できる。また、電力残量が低下したロボットに対しては、充電タスクを割当てる。この時、ロボット側で充電タイミングと充電エリアへの経路を算出するのではなく、サーバ側で電力残量と充電開始のタイミング、及び充電フィールドへの経路計画を保持する。これにより、フィールドのロボット台数の増減する環境下でもサービスを継続する仕組みを構築する。

以下では、まず、ロボットの動作計画の手順を説明する。これは、サービスの基本状態である巡回タスク実行中の動作に該当する。次に、遠隔作業タスクと充電タスクの割当て手順について述べる。

4.1 タスク割当手順

各ロボットからは一定の規則でセンサ情報による位置情報、例えばロボット側でのオドメトリを用いた動作結果が通知され、これとの同期によりロボット情報は更新される。ただし、位置推定のためにここでポーリングによる情報取得を行う場合、ロボットの動的な情報取得の精度を上げるためにはポーリング頻度を上げる必要があり、パケット数は増大化し、データは冗長化する。一方、ポーリング頻度を下げた場合は、動作契機となる端点でのロボット状態情報の精度が下がることがあり、次の指定パスに対する誤差が拡大する。また、オドメトリをクラウド側で計算する場合、ロボットの速度を状態変数に組み込む必要があるが、ここで取得される速度は動作結果に対する速度となるため、不安定な実環境と通信遅延もあわせて考慮した場合、やはり誤差は拡大する。よって、本稿のタスク割当て機構では、目標速度 (v) と目標回転速度 (w) を利用した速度動作モデルによりサーバでロボットの事後状態位置を計算し、これをもとに遠隔作業タスク、及び充電タスクをロボットに割当てる。また、各ロボットのエネルギー消費量をもとに、時間経過 (t) に伴う電力残量を算出し、タスク割当てに利用する。したがって、タスク割当てに利用する情報は、各ロボットの位置情報、電力残量、巡回タスク実行中

のロボット台数である。

また、本稿では、走行パスのゴール位置ポイントを同期ポイントとする。ロボットは、走行パス移動完了時、つまり、自身がゴール位置ポイントへ到着したと判断したタイミングで、サーバへ動作結果通知を行う。この時、同時に環境センサによる自己位置補正を行い、自己位置推定結果を再度サーバへ通知する。これにより、サーバ側のサーバ側のロボット情報と、ロボット側の自己位置推定情報の誤差を最小限にする。また、この時、電力残量についても通知し、サーバ側の情報と同期を行う。

4.2 遠隔タスクの割当て

電力を使い果たすロボットの発生をできる限り抑制し、作業フィールド内で稼働するロボット数及び稼働率を最大化するために、遠隔タスクをロボットへ割り当てるにあたって、以下の3つの選択方法を用いる。

- エリア選択法：ユーザの指定する部屋のエリアにいるロボット、つまり距離の近いロボットを選択する。この方法では、同一エリア内のロボットを利用するため巡回タスクの経路計画をそのまま適用した状態でタスク割当てが可能のため、サービス開始が迅速となる。ただし、電力残量の少なくなったロボットが選択されてしまう場合があり、ロボットが電力を早く使い果たしてしまいサービスが中断することがある。
- 電力残量最大法：電力残量 (pw) が最大のロボットを選択する。この方法では、電力残量 (pw) の少なくなったロボットが選択されることによりロボットが電力を早く使い果たすことは抑止できる。しかし、ロボットの作業フィールドには檻が設置されており、異なるエリアのロボットが選択された場合、檻を迂回して部屋を目指すため、移動距離は増加する。また、遠隔ユーザへ遠隔操作権を発行する前に経路計画の再計算が必要になり、サービス開始に時間がかかる。また、移動距離が増加すれば電力は消費されるため、サービス対象の部屋に到着しサービスを開始する時には、電力残量はすでに低下している場合がある。
- 電力-エリア選択法：電力残量 (pw) が閾値 z (z は固定値) 以上のロボットで、ユーザの指定する部屋のエリアにいるロボットを選択する。該当ロボットが存在しない場合は、電力残量最大法を用いてロボットを選択する。この方法では、各ロボットの移動体の電力残量を均一化しながらロボットの選択を行うため、エリア選択法と電力残量最大法の欠点が緩和される。

電力-エリア選択法の初期状態では、ロボットの電力残量が多いため、ほとんどのロボットの電力残量 (pw) が z 以上となり、エリア選択法と同じ動作となる。時間が経過すると、閾値以上の電力残量を持つロボットが少なくなり、電力残量最大法と同じ動作となる。ただし、電力-エリア選択法を導入しても、そのままバッテリーの充電を行わない場合は、ロボットの電力残量は時間の経過にともない減少するため、適切な閾値 z の設定は困難な問題となる。提案手法では、ここにロボット要求による充電行動をひとつのタスクと位置づけ、充電タスク導入することによりこれを回避する。

4.3 充電タスクの割当て

充電タスクの割当てと、作業フィールドでのサービスへの復帰について述べる。充電タスクは、ロボットからの要求として発生するが、本稿では、サーバでロボットの電力残量 (pw) を予測し、これをもとに充電タスクの割当てを行う。電力残量 (pw) が閾値 y ($y < z$, y は固定値) 以下になったロボットのステータスは、充電要求中となる。サーバは、充電エリア (Area0) への分岐経路を持つ端点 l_g での、ステータスが充電要求中であった場合、このロボットの次の走行パスに、 $l_g \rightarrow B_g$ を設定し、ロボットは充電フィールド (Area0) へ格納する。

次に、作業フィールドへの復帰について記載する。充電フィールド内の $robot_0$ から電力残量が Full になったことを通知された場合、サーバは、電力残量 (pw) が、閾値 z 以下の定常タスク実行中のロボットを検索する。電力残量が、 z 以下の $robot_1$ が発見された場合、サーバは $robot_1$ のステータスを充電要求中へ変更し、充電タスクの割当てを行う。この割当て条件は、すなわち、巡回タスクは継続可能だが、遠隔タスクへの割当てが不可能の状態を示す。電力残量 (pw) が閾値 y 以下のロボットが複数台発見された場合は、電力残量が最も低下しているロボットを選択する。発見されなかった場合は $robot_0$ を充電フィールド内に待機させ、フィールド内の電力残量が z 以下になった $robot_2$ に対して、充電タスクを割当てする。同時に、 $robot_0$ には、 $robot_2$ が稼働する走行ルートへのスタートポイント B_s を設定し、 $robot_0$ はフィールドへ出動する。

一方、複数台のロボットが、充電フィールド内に格納された場合、巡回タスク実行中のロボット台数が $field_min$ 以下、かつ、電力残量が閾値 z を上回ったロボットが、フィールドへ出動する。

5. 実装

提案手法の実装検証を行うために、ペットシッターサービスのプロトタイプシステムを構築した。本章ではその実装結果を記載する。

構築したシステムのネットワーク構成を図4に示す。実装には RSNP を利用した。インターネットやシステム構築

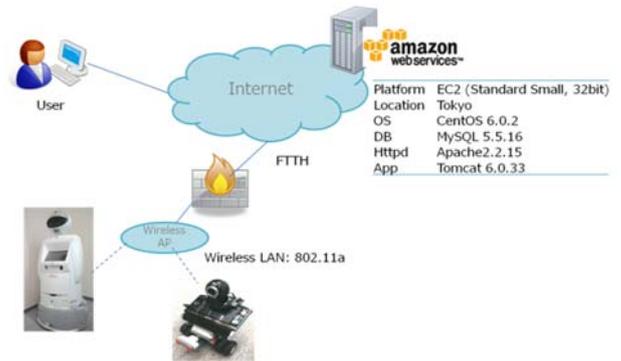


図4 プロトタイプシステムのネットワーク構成

Figure 4 Network configuration of the prototype system

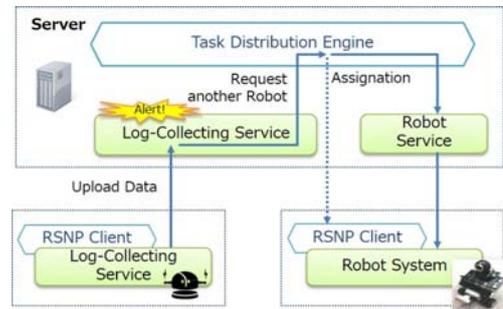


図5 ロボットが取得したセンサ情報をもとに他のロボットをアサインする仕組み

Figure 5 Mechanism to assign other robot based on the robot sensor information

向け通信基盤である Web サービス基盤を利用しているため、インターネットとの整合性の高い標準化された機能を利用できる。サーバは、OS に CentOS6.0.2、Web アプリケーションサーバに Tomcat6.0.33 を利用し、AmazonEC2 (Tokyo) 上に実装した。サービスアプリケーションは、サーバ上に常住する Java アプリケーションであり、ユーザに対する認証、操作画面を提供するとともに、ユーザから要求された作業のデータベースへの保存を行う。また、ロボットに対しては、ロボットの認証、ロボットへのタスク割当て、ロボットが収集する画像・ログデータのデータベースに保存を行う。データベースは、MySQL5.5.16 を利用し、ユーザ認証、ロボット認証、サービス要求管理、ロボットリソース情報管理、セッション管理の 5 種類のテーブルを定義した。なお、ロボットの認証は本来第三者機関を通じて行うことが RSNP で推奨されているが、プロトタイプシステムはデータベース内にテーブルを作成して行っている。

遠隔作業タスクを実行するためのロボットとして、enon と LEGOMINDSTORMS NXT を実装した。NXT にはロボットを制御する役割を担う WindowsPC を Bluetooth により接続し、PC 上の無線 LAN アダプタにより RSNP サーバへ

接続する。また、富士通製サービスロボット enon には、ロボット内に実装された Windows 上に RSNP 機能を構築し、ロボットが持つ無線 LAN を経由して RSNP サーバに接続した。ロボットで提供するサービスは、ロボットのカメラで撮影した画像データをサーバへ送信する機能 (Multimedia Profile) と移動方向と移動量を指定しロボットを遠隔操作する機能 (Motion Profile) である。enon は両方の機能を持つことが可能であり、NXT には接続先 PC に Web カメラを接続することにより両方の機能をもたせた。

さらにロボット側の自己位置推定情報やバッテリー状態を取得し、サーバへ送信する仕組みを疑似的に実装した(図 5)。本来 Multimedia_Profile#get_sensor_info でロボットの内部情報を取得しサーバへ送信するべきだが、現時点では Contents_Profile でランダム値を生成してサーバへ送信している。ここでは、ロボットからの情報をサーバ側で蓄積及び分析し、閾値を超えた場合には、タスク分散機構へ要求を行い、他のロボットをアサインする仕組みを実装した。これは、本稿の、ロボット割当て機構のロボットへの充電タスク割当てに該当する。

6. 考察

本稿は、ロボット自身のアルゴリズムの高速化や計算量の低減により、ロボット負荷の軽減を目指すのではなく、サーバ側で複数ロボットの状態情報をもとにタスク計画を行うことでロボット側の計算処理のための電力消費を低減し、さらに充電を考慮したタスク割当てを行うことで、ロボットサービスの継続を目指している。具体的には、ランダムに発生する複数ロボットの電力残量の低下による充電行動と、ユーザによる遠隔作業を、サーバ側で各ロボットの状態を管理すること割当てを調整し、タスク分散を行った。このタスク分散に必要なロボット情報をサーバに保持するために、サーバ側に各ロボットの状態推定機能を持たせた。類似手法は、環境クローニングとして[12][13]において提案されているが、さらに、走行経路計画の変更を伴う移動ロボットによるサービスでは、走行パス単位の端点での同期が必要であることを示し、サーバとロボットの推定誤差を最小限にするためには、アクチュエーションのための同期タイミングの考慮を行う必要があることを示した。

また、複数ロボットでのサービス提供の場合、サービス全体の一貫性が求められる。クラウドによるデータ共有を利用しない群ロボットでは、このような場合、リーダーロボットを選出し、リーダーロボットにデータを集約することでロボット間の調整を行う。この場合、リーダーロボットに多くのリソースが要求される。本稿では、タスクプランをクラウドで行うことにより、複数ロボット間の協調条件をサーバで付与することを可能にし、ハイエンドな機能をもたない汎用ロボットでもサービス可能な仕組みとした。

ロボットの遠隔作業タスクの割当てには、充電フィール

ドであらかじめ待機状態にあるロボットを、遠隔作業に割当てるのがもっとも単純な割当て手法となり、そのようにスケジューリングを組むことのほうが効率的な場合もあるが、本稿では、人間活動を絡めたサービス環境においては、より不確定な要素が増大し、スケジュール通り充電計画に即さないケースがあると考え、任意のタイミングでロボットが充電を行っても、サービス継続が可能な仕組みを採用した。

7. おわりに

本稿では、ケアサービス分野へのロボットサービス適用をめざし、複数ロボット間でのタスク分散処理によるケアロボットサービスを提案した。その中で、ロボットの消費電力を考慮した仕組みを設計した。また、プロトタイプシステムを構築し、その有効性を確認した。その中で、消費電力を考慮したタスク分散は有効に働くが、タスク割当て中のロボット側の状態変化、通信遅延の考慮が必要なが明らかになった。

今後、シミュレーション評価を行いながら、分散処理の検討を進めていく予定である。

謝辞 本稿の執筆にあたり、五十嵐登さん、大山直人さん、齊藤由香利さん、阪口和明さん、角田龍太さん、中山央士さんに貴重なご意見を頂きました。ここに感謝の意を表します。

参考文献

- [1] Ando N., Suehiro T., Kitaguchi K., Kotoku T. and Yoon W.: RT-Middleware: Distributed Component Middleware for RT (Robot Tecnology), IEEE/RSJ International Conference on Intelligent Robots and Systems 2005 (IROS2005), pp3933-3938 (2005)
- [2] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, Andrew Ng.: ROS: an open-source Robot Operating System. Proc.of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics, (2009)
- [3] Google I/O 2011: Cloud Robotics, ROS for Java and Android, (2011.5.12) [Online]. Available: <http://www.willowgarage.com/blog/2011/05/12/google-io-2011-cloud-robotics-ros-java-and-android?page=5>
- [4] Koji Kamei, Shuichi Nishio, and Norihiro Hagita.: Cloud Networked Robotics, IEEE Network May/June-2012, pp.28-34, (2012).
- [5] 成田雅彦, 村川賀彦, 植木美和, 中本啓之, 平野線治, 蔵田英之, 加藤由花: 普及期のロボットサービス基盤を目指す RSNP (Robot Service Network Protocol) 2.0 の開発, 日本ロボット学会誌, Vol. 27, No. 8, pp. 857-867(2009).
- [6] 成田雅彦, 村川賀彦, 植木美和, 岡林桂樹, 秋口忠三, 日浦亮太, 蔵田英之, 加藤由花: インターネットを活用したロボットサービスの実現と開発を支援する RSi (RobotService Initiative) の取り組み, 日本ロボット学会誌, Vol. 28, No. 7, pp. 829-840 (2010).
- [7] Nakagawa S., Ohyama N., Sakaguchi K., Nakayama H., Igarashi N., Tsunoda R., Shimizu S., Narita M., and Kato Y., "A Distributed Service Framework for Integrating Robots with Internet Services," in The 26th IEEE International Conference on Advanced Information Networking and Applications (AINA2012), pp. 31-37. (2012)
- [8] RSi Robot Service initiative: [Online]. Available: <http://robotsservices.org/>
- [9] Kato Y., Izui T., Tsuchiya Y., Narita M., Ueki M., Murakawa Y., and

- Okabayashi K., "RSi-Cloud for Integrating Robot Services with Internet Services," in The 37th Annual Conference of the IEEE Industrial Electronics Society (IECON2011), 2011, pp. 2164–2169.
- [10] 加藤由花, 岡部泉, 村川賀彦, 岡林桂樹, 植木美和, 土屋陽介, 成田雅彦: インターネットワークサービスプラットフォームにおけるロボットサービス提供手法, 情報処理学会論文誌, vol.54, No.2, pp.659-pp.671.(2013)
- [11] Markus Waibel, Michael Beetz, Javier Civera, Raffaello D'Andrea, Jos Elfring, Dorian Galvez-Lopez, Kai-Haussermann, Rob Janssen, J.M.M. Montiel, Alexander Perzylo, Bjorn Schiessle, Moritz Tenorth, Oliver Zweigle, and Rene van de Molengraft, "Robo Earth - World Wide Web for Robots -", Robotics & Automation Magazine, IEEE, June 2011, pp.69-82, (2011)
- [12] Dominique Hunziker, Mohanarajah Gajamohan, Markus Waibel, and Raffaello D'Andrea, "Rapyuta: The RoboEarth Cloud Engine", IEEE the International Conference on Robotics and Automation, (2013)
- [13] Guoqiang Hu, Wee Peng Tay, and Yonggang Wen.: Cloud Robotics: Architecture, Challenges and Applications, IEEE Network May/June-2012, pp.21-27 (2012)
- [14] Google Goggles: [Online]. Available: <http://www.google.com/mobile/goggles/>
- [15] Ben Kehoe, Akihiro Matsukawa, Sal Candido, James Kuffner, Ken Goldberg. :Cloud-Based Robot Grasping with the Google Object Recognition Engine, IEEE International Conference on Robotics and Automation. (2013).
- [16] Rajesh Arumugam, Reddy Enti, Liu Bingbing, Wu Xiaojun, Krishnamoorthy Baskaran, Foong Foo Kong, A.Senthil Kumar, Kang Dee Meng, and Goh Wai KitVikas. "DAVINCI: A Cloud Computing Framework for Service Robots". IEEE International Conference on Robotics and Automation, (2010).
- [17] G. T. Jay, "brown remotelab: rosbridge," 2012. [Online]. Available: <http://www.rosbridge.org/>
- [18] R. G. Smith.:The Contract Net Protocol :High-Level Communication and Control in a Distributed Problem Solver, IEEE Transactions on Computers, pp.1104-1113.(1980)
- [19] Jun Ota, Multi-agent robot systems as distributed autonomous systems, Advanced Engineering Informatics 20 (2006) 59–70
- [20] 吉田 幹, 奥田 剛, 寺西 裕一, 春本 要, 下條 真司: マルチオーバレイと分散エージェントの機構を統合した P2P プラットフォーム PIAX 情報処理学会論文誌, vol.49, No.1, pp.402-413,(2008).