

複数台ロボット向け消費電力量削減手法及び ROS によるエッジサーバの提案

三宮 夏帆^{1,a)} 天野 英晴³ 菅谷 みどり²

概要:近年、ロボットは様々な場面に応用され、より身近なものとなっている。応用場面の拡大に伴い、作業効率の上昇のために、複数台のロボットを用いたサービスの増加が予想される。その際の課題の 1 つとして、サービス実施ロボットの増加に伴う、これらの省電力化によるサービスの持続的提供性があげられる。CPU を対象とした電力のばらつきのモデル化および計算機を対象とした省電力の研究は数多く提案されているが、移動などのモーターのアクチュエーションなどと連動する計算機を対象としたばらつきのモデル化および、複数の機器を対象とした省電力は達成されていない。本研究では、汎用性の高い電力予測モデルの提案と、予測モデルに基づくタスク交換の最適化処理をサーバーにオフロードするシステムを提案する。実環境での評価を目指し、ROS による実装によりロボットがサーバーと透過的に連携しながら処理するシステムを開発し、評価により有効性の傾向が得られたことを示す。

キーワード:組み込みシステム、省電力化

1. 研究の背景と目的

近年、世界中でロボットの活躍の場が工場などの製造業から、医療分野[1]、警備[2]、災害現場[3]、輸送現場[4]など、様々なサービス分野に応用され、より身近なものになっている[5]。具体的な医療分野の使用例としては、薬剤や検体の搬送用ロボットが試験的に導入されており[6]、夜間や休日の人手不足の軽減に貢献している。警備分野では、無人のパトロールロボットや監視ドローンが開発、試験中であり[7]、今後も発展していくと考えられる。災害現場においては、生身の人間で作業が困難な汚染区域の探索やがれきの撤去を目的に探査機[8]や無人機が開発されている。輸送現場においては輸送用ドローンの試験的導入[9]や、倉庫内での荷物移動ロボットの導入が行われている[10]。日本では少子高齢化に伴う労働人口減少対策の一環でロボットの活用が促進されており[11]、介護や福祉、危険な場所での作業など様々な分野で人の代替となる複数台ロボットサービスの市場は今後拡大すると予想されている[12]。

複数台のロボットを用いたサービスは人的コストの削減を実現する一方、サービス全体の消費電力量の増加、実施するサービス規模に比例して使用されるロボットの台数の増加による消費電力量の増加がある。警備ロボットや宅配ドローンの実際の運用を想定した場合、使用されるロボットの台数は膨大なものとなり、環境負荷低減のためにもサービス全体の省電力化が必須である。しかし、複数台ロボットの制御システムで電力管理による全体の消費電力の削減手法は十分に議論されていない。

全体の消費電力を検討するには、個々のロボットの電力モデル予測モデルが課題となる。既に計算機や、CPU や GPU

を対象とした電力消費のモデル化やそれを実現する手法の研究は数多く提案されている。しかし、移動などのモーターのアクチュエーションなどと連動する計算機を対象としたモデル化については十分に議論されていない。また、複数台を対象としているものは少ない。清水[13]、寒竹[14]は、ロボットの電力消費予測モデルを提案し、予測モデルを元にタスクの割振りの最適化による複数台のロボットの累計電力量の削減をシミュレーションにより示した[14]。しかし、消費電力量予測モデルの汎用性および、実環境での評価が不十分で、有効性に課題がある。

そこで、本研究では、先行研究の課題をもとに(1) ロボットの電力消費の予測モデルおよび、(2) 予測モデルに基づくタスク交換の最適化処理をサーバーにオフロードすることで、実環境に適したシステムを提案する。提案に基づき、システムの設計・実装を行い、複数台ロボット制御向け消費電力量削減手法の有効性を示した。

2. 参考文献

計算機科学における電力削減に関する技術は、主にハードウェアでの電力削減技術が提案されている。プロセッサの電圧・周波数制御では、供給電圧及び動作周波数を適切に制御する DVFS(Dynamic Voltage and Frequency Scaling)アルゴリズムは多くのプロセッサなどで使用されている [15]。近年では高度な計算機でのモーター制御が一般的になり、アクチュエータ制御の際に、入力信号の大きさに応じたパルス波の幅(デューティ・サイクル)の制御、アクチュエータの動作を PWM (Pulse with Modulation) 制御する方法も、消費電力削減手法の一つとして挙げられる[15]。しかし CPU への適用とモーター制御双方でロボットの消費電力制御に

ついて、一般化は行われていない。

ロボットで使われるセンサー類の省電力化の研究としては省電力を考慮した移動目標認識システムにおける省電力化[16]、飛行ドローンの位置認識システムにおける省電力化[17]がある。これらは、処理負荷に応じた動的電力管理手法や低電力のセンサーノード利用による電力削減など、多様な方法で電力削減手法が提案されている。しかし、これらは、主にハードウェア側での電力削減技術であり、出荷後のサービスに依存した電力消費量の変化は考慮していない。

より柔軟な電力削減を実現するためには、ソフトウェアでの技術も重要である[18,19]。Y.Kim らは、マルチスレッドコアアーキテクチャにおいて、プロセッサ間の重複検索を検出して回避する手法を提案し、DVFS 手法により性能を維持したまま消費電力を最小限に抑える結果を得た[18]。A. Hosangadi らは、計算命令を自動置換処理による配列演算での計算数削減による消費電力の削減技術を提案した[19]。ソフトウェアでの電力削減は、ハードウェアほどの削減効果はないものの、ハードウェアの変更が困難なケースや、出荷後であってもソフトウェアの制御により削減効果が得られるため、出荷後のサービスに柔軟に対応したい場合にメリットが大きい。

これらのメリットを生かした複数台のロボット全体の消費電力を削減する技術として、個別の消費電力量をモニタリングし、使用者へ通知することで、節電操作の促進を図るシステムといった電力削減手法が行われている。坂村健らは、TK-SLP(Super Low Power Embedded Software Platform)という組込み機器向けの消費電力モニタリング機能などを備えた省電力システム開発プラットフォームを提案した[20]。本プラットフォームの機能の 1 つとして節電操作を促進するために消費電力モニタリング機能により個別の組込みシステムの電力量をシステムの利用者に自動送信する仕組みが提供されている。しかし、機能はあくまで監視のみであり、この仕組み自体が消費電力量を削減するものではない。

複数のロボットの動作効率化を観点とした管理システムの研究はいくつか存在する。中川らはクラウドによって複数のロボットを効率的に連携させることでサービスの継続を行う管理手法を提案した[21]。バッテリー残量の多い個体に仕事を多く割り当て、残量の少ない個体の充電時間を確保する方式である。しかし、予測モデルの具体性が低い課題がある。

Abhijeet Ravankar らは、ロボットに割り当てられたタスクの優先度、工作中的のバッテリー消費量、バッテリー残量から、優先して充電を行うロボットを設定することで効率的に充電を行う方法を提案している[22]。しかし、この 2 つの研究ではあくまで充電の最適化によるサービス継続を目的としており、複数台ロボット消費電力量の削減については検証が行われていない。

3. ロボットの省電力化手法の課題

清水らは、複数台ロボットを用いたシステムの全体の消費電力量の削減手法を提案した[13]。ロボットの動作特性の差異による電力消費特性の分析が、同機種のロボットでも消費電力の差が出ることに着目し、各個体の移動距離当たりの平均バッテリー消費率と時間当たりの平均バッテリー消費率を把握した。具体的には、移動距離当たりの平均バッテリー消費率と時間当たりの平均バッテリー消費率が高いグループの動作を移動距離当たりの平均バッテリー消費率と時間当たりの平均バッテリー消費率が低いグループの動作と入れ替えることで、総電力の削減を提案した[13]。

清水らの提案手法は、ロボット単体の電力削減ではなく、複数台のロボットを対象とした電力削減を行う点で、今後実際に行われるロボットサービスにも有用な手法であったが、好ましい結果が得られなかった。これに対し、寒竹らは、この研究を元に、ロボットの個体差に着目した予測モデルを提案し、よりの確に動作を交換する最適化手法を提案した[14]。

寒竹らは、同一機能のロボットでも内部のハードウェアが異なった場合、同じタスクを行った際の消費電力には差異が生じる、また、同一のロボット個体でも、そのロボットで行うタスクが異なれば消費電力量には差異が生じることに着目し、ロボット毎の差異をそれぞれ「ハードウェアの個体差」、「動作の個体差」と定義し、定義に基づく電力消費の予測モデルを作成し、作成した予測モデルに従い、清水らが提案したタスク交換を行うことにより、ロボットが動作し、提供するサービス全体の消費電力量を削減することに成功した[14]。しかし、消費電力量予測モデルの汎用性および、実環境での評価が不十分で、有効性に課題があった。

4. 提案

4.1 概要

本研究では、課題を解決するために(1)アクチュエータを持つ計算機向けの省電力予測モデルの再検討、(2)複数台のロボットのタスク交換を前提とした省電力化のための最適化処理とそれを実現するシステムの設計・実装 (3) 実機による検証・評価、を提案する。(2)(3)については、(1)の予測モデルの算出をサーバーにオフロードする際の重回帰分析は、ロボットの台数と分析要素の増加による計算量の爆発的増加が課題となることから、FPGA を用いたエッジによる高速計算システム[23]により処理時間の削減検証(4)を実施する。

4.2 消費電力の定義

電力消費予測モデルと異なるハードウェアでの実機検証とを行う。先行研究の予測について述べる。個体差を移動距離と、基礎的な電力消費の 2 項目に分け、さらに移動と、停止から走行の 2 つの状態をモデル化した[14]。

ある個体 x の消費電力(W_x)は、

$$W_x = d \times CPD_x + r \times CPS_x \text{ (式 1)}$$

の式により求める[3].ここで d (移動距離), r (停止状態から走行状態への移行回数), CPD_x (個体 x のある速度における移動距離単位の消費電力量), CPS_x (個体 x の停止状態から移動状態への移行回数当たりの消費電力量)と定義した.

4.2.1 バッテリー電圧の変化による移動距離の推定

寒竹らのモデルでは移動距離を特徴量としていた[14].しかしロボットの移動距離は, 動作する物理環境条件により変化する. また, 移動距離を直接計測できないロボットも多く存在することから, 移動時に消費する電力エネルギーで評価した方が, 汎用性の高いモデルとなると考えた.

そこで, 移動時に消費する電力エネルギーを, 消費電力量(W)とし, バッテリーの電圧値の降下量で表すものとした. バッテリー電圧: $V(v)$, モータードライバにかかる電圧との比率: B , 車輪の直径: $R(cm)$, 左右のモーターにかかるデューティ比の平均: D , モーターの最大回転数とその時の電圧との比例定数: X , 動作時間: $T(s)$ を掛け, 個体 x の移動距離: $L_x(cm)$ を求めるものとした (式 2).

$$L_x = \sum_{i=0}^T V \times B \times X \times R \times \pi \times D \text{ (式 2)}$$

4.2.2 実機による予測式の検証 1

新たに提案した予測式(式 2)の検証を目的に, 8 台の Zumo を用いて実機検証を行った. Zumo[6]は, 単三型電池を用いるため充電と充電の確保が容易なこと. 小型ロボット故に電池に対して駆動時間が長く, Arduino もしくは mbed による自由度の高いプログラミングが可能である点が評価できる. 実験のため Zumo には電圧取得が可能な拡張を行った (図 1)



図 1:電圧取得基盤と Zumo+ラズパイ構成

上記の式 2 を元に Zumo を 8 台, 100cm 走行させ, 移動距離の真値の平均, 誤差を計測した結果を図 2 に示した.

実走行時には路面との転がり抵抗や自重によるモーター軸への負担が影響するため, 理論値(推定距離)が真値(実際の走行距離)よりも大きくなり, 平均して 34.2%の誤差と, ロボット毎にばらつきを確認した. 本結果を重回帰分析し, 理論値 L_x と計測中の電圧の変化量 VL_x , 計測中の最大電圧 VmL_x から以下のロボットごとの移動距離の誤差を補正する値を算出した (式 3).

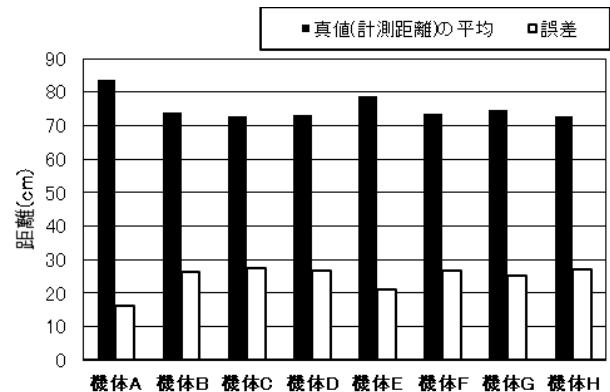


図 2: 機体ごとの移動距離理論値と真値の誤差比較

$$\begin{aligned} \text{補正距離} = & -38.084 + 0.643 \times L_x - 20.487 \times VL_x \\ & + 10.003 \times VmL_x \text{ (式 3)} \end{aligned}$$

式 3 によって先ほどの真値を補正した結果, 平均誤差率が 4.28%と, 約 87%の誤差の削減を達成した. 重回帰式の評価において重決定係数が 0.941, 有意 F 値も低く, 補正式の精度が高い結果が得られた(表 1).

表 1:重回帰式の解析結果

重回帰係数*	重決定係数*	F値*
0.970	0.941	F<0.000001

また, 理論値 (L_x), 計測中の電圧の変化量 (VL_x), 計測中の最大電圧 (VmL_x)の 3 つの要素の中で理論値と計測中の最大電圧が移動距離に対し影響度が大きい. 一方計測中の電圧変化量に対しては, 一回の計測時間が短く, 電圧の変化量が小さいことから影響度は小さい. ただし 3 つの要素とも有意確率が 5%を下回っているのである程度関連性はあるとし, 式 3 を有意なもの結論付けた (表 2).

表 2:各要素の重回帰式における影響

L_x : 理論値(推定距離)		
標準偏差	t値	p-値
0.022	29.136	p<0.0001
VL_x : 計測中の電圧変化量		
標準偏差	t値	p-値
8.134	-2.518	0.013
VmL_x : 計測中の最大電圧		
標準偏差	t値	p-値
4.19	2.382	0.018

4.2.3 実機による予測式の検証 2

消費電力式(W_x)から個体差の検出が可能かを検証するため, 小型無限軌道ロボット Zumo を用いた実機検証を行った. Zumo でバッテリーの電圧値を取得し, 4.2.2 節により計測した移動距離(d)と停止状態からの移行状態の回数(r)を 10 回計測した値を元に式 2 により算出し, 結果を消費電力

(W) の予測式(式 1)に当てはめ、累計の電圧変化量を求めた。

4.2.4 結果

求めた予測式を、 $d=1000(\text{cm})$ 、 $r=50$ として 0.1 秒ごとの電圧変化量の総和をシミュレーションし、累計の消費電力量が小さい順に並べた結果を示した(図 3)。

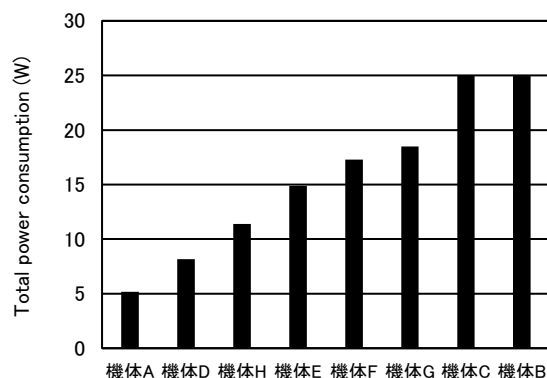


図 3:累計の電圧変化量(W)

Zumo の消費電力量のばらつきの分散は 6.78(W)であり、十分にばらつきがあることから、予測式の汎用性を確認できた。グラフは消費電力量が小さい順に並べているが、停止回数を固定とし、移動距離を増やしても順序の変化に違いがなかった。一方で移動距離を固定とし、停止回数を増やすと順番が変わることがあった。これは、Zumo においてモーターを回し始める際に必要な起電力が、個体毎に大きく差があることが、消費電力量に影響していることが原因と考えられる。

5. オフロードサーバによる最適化処理

5.1 タスク割り振りによる削減手法

改善した消費電力予測モデルで算出した値をもとに算出したロボットの消費電力量に応じ、タスクを割り振る動作割り振り最適化システムによってサービス全体の消費電力量の削減を図る(図 4)。

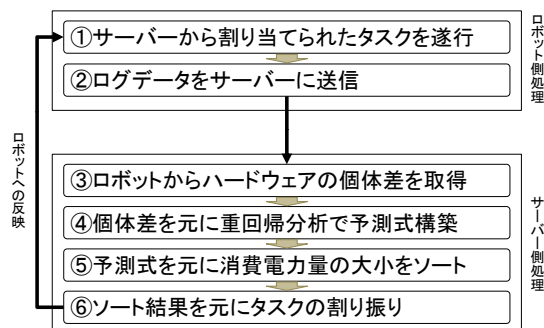


図 4:システム全体の流れ

本研究では、先行研究で未実施であった実機による、動作時の予測値を元にした最適なタスク割り振りを検証する。最適なタスク割り振りは、個々のロボットからデータを収

集し、予測式を元に最適な動作割り振りを決定する。具体的には、消費電力が多い個体に少ないタスク、消費電力が少ない個体に多いタスクを割り振る手法である[14]。図 4 にシステム全体の流れを示した。

5.2 システム設計と実装

図 4 に示したように、動作割り振り処理は全体の結果を集計しソートを適用することから、一極集中処理が望ましい。そのため、サーバ・クライアント形式の非同期分散処理システムを開発するものとした。サーバ上での処理は(3)(4)主に個々のロボットから収集したデータを元にした予測式の計算、(5)(6)予測式に基づく、割り振りの最適化の 2 段階である。処理の詳細を表 3 にまとめた。

表 3 サーバ・クライアント処理の詳細

ロボット
(1)サーバから送信されたタスク命令を受け取り、指示に従いタスクを遂行。
(2) 1 タスクの終了毎に、移動距離、停止状態から動作状態への移行回数、バッテリー電圧の 3 つのデータをサーバに送信。
サーバ
(3)送られた情報をもとに
(4)各ロボットの移動距離、停止状態から動作状態への移行回数、バッテリー電圧を元に重回帰分析を行い消費電力量の予測式を算出。
(5)求めた予測式から、ロボットごとの消費電力値をもとに消費電力に基づいた最適なタスクの組み合わせを計算する。最適なタスクは消費電力量の少ない順にクイックソートの実施。
(6) タスク割り振りの決定と結果の送信

ロボットは、モーターを備えたロボット本体と制御マイコン(Arduino)、通信・制御(Raspberry Pi)で構成する。

また、命令の発行とデータ送信を汎用的な通信ミドルウェア ROS(Robot Operating System)で行うことで、非同期通信により制御できるようにした。図 5 に実装したシステムの図を示した。サーバ側と、クライアントは双方に Linux 上に ROS を構成し pub/sub 通信でデータの送受信を実現した。

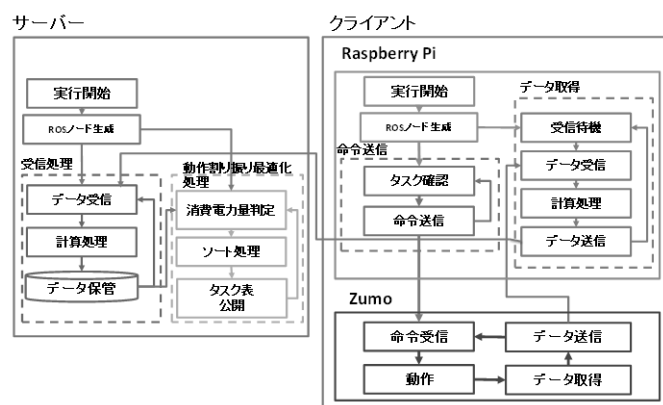


図 5:システム設計図

6. 評価

6.1 概要

消費電力量予測式の評価モデルをもとに、ROS の pub/sub

を実装し、開発したシステムでタスク割り振り最適化処理と累計の消費電力量の削減効果の評価する。Zumo10 台を用いてタスク割り振り最適化処理を行った時と行わなかった時で累計の消費電力量を比較し、最適化効果の検証を行う。システムの構成を以下に示す。

表 4 評価環境

	CPU	Memory	OS
サーバー	Intel® Core™ i7-4600U (1.8GHz)	8.0GB	Ubuntu 16.04LTS
クライアント (Raspberry Pi)	ARM Cortex- A53(1.4Ghz)	1.0GB	Ubuntu Mate 18.04 LTS

本実験にてロボットに割り当てるタスクは前進、後退、左旋回、右旋回の 4 つの行動を 1 セットとして構成されている。セット内の各行動にかかる時間及び総セット回数の違いによって総走行時間(移動距離)と総移行回数の変化を再現している(表 5)。

表 5:タスク一覧

タスク	1セット目	2セット目	3セット目	4セット目	5セット目	総走行時間	総移行回数
1	1秒	—	—	—	—	4秒	4回
2	2秒	—	—	—	—	8秒	4回
3	1秒	2秒	—	—	—	12秒	8回
4	2秒	2秒	—	—	—	16秒	8回
5	1秒	2秒	2秒	—	—	20秒	12回
6	2秒	2秒	2秒	—	—	24秒	12回
7	1秒	2秒	2秒	2秒	—	28秒	16回
8	2秒	2秒	2秒	2秒	—	32秒	16回
9	1秒	2秒	2秒	2秒	2秒	36秒	20回
10	2秒	2秒	2秒	2秒	2秒	40秒	20回

本実験でのタスク割り振り最適化処理では、消費電力の大きい順にソートを行い、消費電力が最も大きいロボットに仕事量が一番少ないタスクを、消費電力が最も小さいロボットにタスク 10 に割り当てる形で割り振り最適化処理を行う。

6.2 実験結果

ニッケル水素電池での動作による計測開始から 4 時間の累計の消費電力量とその削減効果の割合を示した (図 6)。

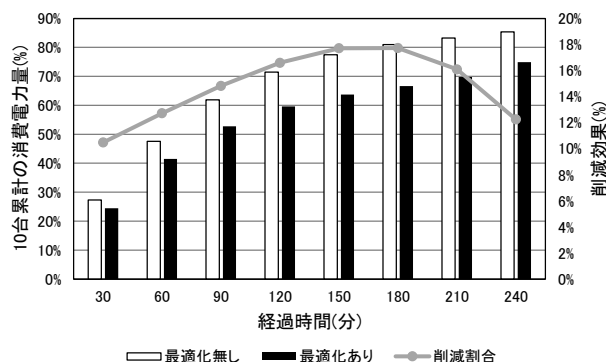


図 6:タスク割り振り最適化処理時の消費電力量の比較

削減効果の割合は開始から約 3 時間(180 分)でピークに達

した。一方でピーク後は削減効果が減少、つまり削減効果がみられなくなった。全体の削減効果は、4 時間全体での削減効果の平均は 14.81%であり、iRobot での実験で確認された削減効果の 0.22%と比較して、十分大きな削減効果を得られた。これは、先行研究で用いられた iRobot Create よりも Zumo のハードウェアの個体差が顕著であるのではないかと考えられる。また、iRobot Create では最大動作時間の 1 時間に対し、4 時間という長時間のデータ取得ができることによる母数の増加が精度の高い予測式を生み出すことができたことも一因と考えられる。

また Zumo ではアルカリ電池での動作が可能のため、電池の種類による削減効果の影響を調べた。図 7 は図 6 と同等の実験をアルカリ電池で行った時の結果である。

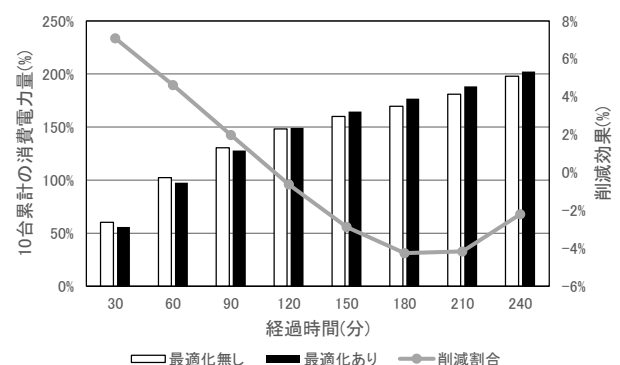


図 7:アルカリ電池によるタスク割り振り最適化処理の有無による消費電力量の比較

アルカリ電池利用時の消費電力量はニッケル水素電池と比べて約 2 倍以上になった。削減効果は、開始時に約 7%程度であったのが、時間の経過とともに減少していった。4 時間全体での平均の削減効果は約-0.06%と削減効果がほとんどないことが判明した。ニッケル水素と比べ、アルカリ電池の方が時間当たりの消費電力量が高いことが削減効果の表れなかったことに関係していると考えられる。

7. FPGA ボードを用いたシステム評価

本研究の過程で複数台のロボットに対応可能な汎用性の高いタスク割り振り最適化システムを実装できた。一方で計算量の問題が存在する。

7.1 重回帰分析の計算量

タスク割り振り最適化システムにおいて、ロボットのログデータから消費電力の予測式を算出する過程において重回帰分析が用いられている。

本研究での実験では、電圧の変化量を Y 値、移動距離、移行回数を X 値として 2 値の要素を元に重回帰分析を行ってきた。この X 値の数が、重回帰分析の計算量に大きく影響を示す。計算するロボットの総数を n とした場合、X 値が 1 つの時、その計算量のオーダーは $O(n \log n)$ となるが、X 値

が 2 つの場合計算量のオーダーは $O(n!)$ となる。以下の図 11 は n の値の変化による計算量の変化を示したグラフである。

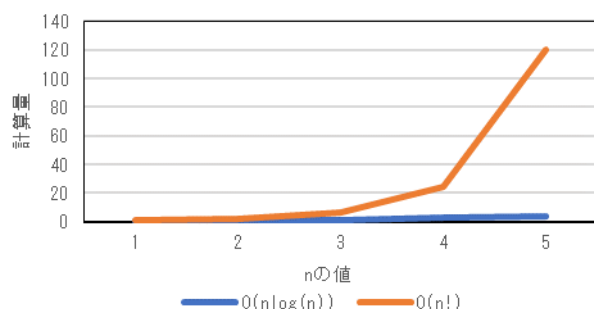


図 11: 計算量の変化

X 値が 2 つの時ではロボット総数の増加で計算量が爆発的に増加していく形となる。本研究の実験では 10 台のロボット分の計算量 3,628,800 に近い計算を以下の表 8 のスペックのサーバーで行っていたが、処理に 1 秒以上かかるなどの問題は見られなかった。

表 8: 計算用サーバーのスペック

CPU	Intel® Core™ i7-4600U (1.8GHz)
RAM	8.0GB
OS	Ubuntu 16.04 LTS
補助記憶装置	SATA3 接続 SSD

しかし、ここから更にロボットの台数が増加すると計算量は莫大になる。計算量を減らすのは非常に困難であるため、計算量の増加に耐えることのできる専用の計算システムが必要となる。そこで MEC システム上の FPGA を用いた高速計算システムで計算を行い、通常のコンピュータで計算処理を行った場合との処理時間の違いを検証する。

7.2 FiC ボード

本研究で用いた MEC システムである FiC ボードを用いた。FiC ボードは慶應義塾大学の天野英晴教授が開発したボード(図 12)であり、内蔵された Raspberry Pi から GPIO 経由で FPGA を制御するボードである。



図 12: FiC ボード

7.3 実験手順

以下図 13 の重回帰分析のプログラムを実行し、その実行

時間を比較する。なお FiC ボードで動作するプログラムのコンバートを行った。

```
//定数は 0, X 値は 2 つとする
for i = 0 からロボット総数
  ロボット i のデータを読み込み
  //データは 1 行ごとに "y,x1,x2" で定義されている
  //求める係数は x1 の係数を k1, x2 の係数を k2 とする
  //
  for j = 0 から最大の行数
    a += y と x1 の積
    b += y と x2 の積
    c += x1 と x2 の積
    d += x1 の 2 乗
    e += x2 の 2 乗
  a-d*k1-c*k2=0
  b-c*k1-e*k2=0
  上記 2 つの連立方程式から係数 k1 と k2 を求める
```

図 13: 重回帰分析のアルゴリズム

FiC ボードと比較対象に使うコンピュータのスペックは表 5 と同一のものである。以下の表 9 は行数 539 行の CSV ファイルを用いて重回帰分析 1 回にかかった時間の平均の比較である。なお画面表示等の計算に必要な余分な処理やシステムによるキャッシュの影響を排除するように実験を行った。

表 9: 重回帰分析実行時間の比較

	FiC	PC
平均時間	32.28(μ 秒)	2410.3(μ 秒)

FiC ボードによる処理の方が PC と比べて 70 倍以上の計算処理速度とできた。このことからエッジサーバの FiC による高速化により制御に近いフィードバックを実現することが可能となることが示唆された。

8. まとめと今後の課題

先行研究の課題をもとに Zumo を用いた実証実験を実施し、動作割り振り最適化手法に累計の消費電力量の削減効果があることを示した。また、無線通信によるロボット消費電力を消費電力予測式に組み込み、精度を向上させることができた。今後は、それらの予測式を元にした実機での実証実験を条件の変更とともにを行い、最も削減効果を高くできる最適化処理のタイミングなどの検証を行っていく。

本研究で行ったタスク割り振り処理の検証において、ROS 通信に必要な Raspberry Pi の関係上、外部からの電源をつなげる必要から、ロボットを実際に走行させるのではなく、空転させる手法で行ったため、実走行時の影響を考慮できていない問題が存在する。システムの改良や使用ロボットの種類を変更し、それらの問題改善に努めていきたい。

また、FiC ボードの実機検証については、実際のタスク割

り振り最適化システムに置いては処理に用いるロボットの台数分のデータを用いる必要があるが、FPGA ではまだ CSV ファイル 1 つでの計算にしか対応していない。今後は、複数のファイルを同時に処理できるプログラムにて比較を行う

また、FiC ボードでの計算においては現段階では、計算処理に関係ない部分の処理速度にボトルネックが存在する。先述のように FiC ボードは内蔵された Raspberry Pi による GPIO 経由で制御が行われる。GPIO はかなり低速であるため、データを FiC ボードに送る過程で秒単位の時間がかかる課題がある。今後のハードウェアの改善を元に検討する。

謝辞

本研究は科学技術振興機構戦略的研究推進事業(JST), CREST, JPMJCR19K1 の支援を受けたものである。

参考文献

- [1] J. Burgner-Kahrs et al. "Continuum Robots for Medical Applications: A Survey", IEEE Transactions on Robotics, VOL. 31, NO. 6, pp. 1261-1280, 2015.
- [2] T. Teodoros & H.Hu. "Toward Intelligent Security Robots: A Survey", IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), VOL. 42, NO. 6, pp. 1261-1280, 2012.
- [3] T. Sakaue et al. "Survey in Fukushima Daiichi NPS by Combination of Human and Remotely-Controlled Robot", IEEE International Symposium on Safety, Security and Rescue Robotics, pp. 11-13, 2017.
- [4] "Kiva Systems: Three Engineers, Hundreds of Robots, One Warehouse" - IEEE Spectrum . <https://spectrum.ieee.org/robotics/robotics-software/three-engineers-hundreds-of-robots-one-warehouse>(参照 2020-01-07)
- [5] T. Maneewarn et al. "Survey of Social Robots in Thailand", International Electrical Engineering Congress (iEECON), pp. 19-21, 2014.
- [6] ホスビー | Panasonic Collaboration 共同制作 | 松下記念病院 <http://phio.panasonic.co.jp/kinen/pc/hospi/index.htm> (参照 2020-01-07)
- [7] "日本初、発進から飛行、帰還・充電までを完全自律で行う自律型飛行監視ロボット「セコムドローン」の「巡回監視サービス」を運用開始セコムが代表企業として参画する PFI 刑務所「美祢社会復帰促進センター」で3月1日から開始". セコム株式会社. https://www.secom.co.jp/corporate/release/2017/nr_20180301.html(参照 2020-01-07)
- [8] "災害対応ロボット Quince|TadoLab". 東北大学 Human-Robot Informatics Laboratory. https://www.rm.is.tohoku.ac.jp/quince_mech/#Quince_1 (参照 2020-01-07)
- [9] "小型無人航空機を用いた郵便局間輸送の開始". 日本郵便 . https://www.post.japanpost.jp/notification/pressrelease/2018/00_honsha/1030_01.html(参照 2020-01-07)
- [10] "Amazon Uses 800 Robots to Run This Warehouse" - IEEE Spectrum . <https://spectrum.ieee.org/automaton/robotics/industrial-robots/amazon-introduces-two-new-warehouse-robots> (参照 2020-01-07)

- [11] 経済産業省. ロボット新戦略. 平成 27 年度版. 2015 http://warp.da.ndl.go.jp/info:ndljp/pid/11181294/www.meti.go.jp/committee/sankoushin/seizou/pdf/003_s01_03.pdf(参照 2019-12-25)
- [12] 国立研究開発法人新エネルギー・産業技術総合開発機構. 2035 年に向けたロボット産業の将来市場予測. 2010. <https://www.nedo.go.jp/content/100080673.pdf>(参照 2019-12-25)
- [13] 清水航平他. "複数台の分散移動ロボットの動作と電力消費特性に基づく管理手法の提案", 情報処理学会第 78 回全国大会, 2015.
- [14] Toshiyuki Kantake "Distributed Task Processing Considering Rmption of multiple robots" Embedded Systems Symposium 2018 Proceedings. 2018, 108-109p.
- [15] 高須雅義他. "組込みプロセッサにおける省電力機構の実機評価", 組込みシステムシンポジウム 2012, 2012, p.79-p.86
- [16] S. Pan et al. "A Low-power SoC-based Moving Target Detection System for Amphibious Spherical Robot", IEEE International Conference on Mechatronics and Automation (ICMA), pp. 1116-1121, 2015.
- [17] J. Eckert et al. "An Indoor Localization Framework for Four-Rotor Flying Robots Using Low-Power Sensor Nodes", IEEE Transactions on Instrumentation and Measurement, VOL. 60, NO. 2, pp. 336-344, 2011.
- [18] Y. Kim, et al. "BRAIN: A Low-Power Deep Search Engine for Autonomous Robots", IEEE Micro VOL. 37, NO. 5, pp. 11-19, 2017.
- [19] A. Hosangadi et al. "Energy efficient hardware synthesis of polynomial expressions", 18th International Conference on VLSI Design pp.653-658. 2005.
- [20] 坂村健. 超低消費電力組込みソフトウェアプラットフォーム: TK-SLP(T-KernelSuper-LowPower)の研究開発. 平成 26 年 . 2014. http://www.soumu.go.jp/main_content/000323271.pdf (参照 2020-01-07)
- [21] Sachiko Nakagawa "Distributed Task Processing Considering Robot Power Consumption in Cloud Robotics" DICOMO2013. 2013, 3-10p.
- [22] A. Ravankar et al. "An Intelligent Docking Station Manager for Multiple Mobile Service Robots", 15th International Conference on Control, Automation and Systems (ICCAS), pp. 72-78, 2015.
- [23] Weisong Shi et al. "Edge Computing: Vision and Challenges" IEEE Internet of Things Journal. 2016 637-646p
- [24] "Create 2 Programmable Robot" | iRobot . <https://www.irobot.com/about-irobot/stem/create-2> (参照 2020-01-07)
- [25] "Pololu - Zumo Robot for Arduino (Assembled with 75:1 HP Motors)". <https://www.pololu.com/product/2506> (参照 2020-01-07)

正誤表

下記の箇所に誤りがございました。お詫びして訂正いたします。

訂正箇所	誤	正
1 ページ 最下部	(所属表記なし)	1. 芝浦工業大学工学部情報工学科 Faculty Engineering Shibaura Institute of Technology 2. 芝浦工業大学工学部情報工学科 Faculty Engineering Shibaura Institute of Technology 3. 慶應義塾大学理工学部情報工学科 Faculty of Science and Engineering Keio University
5 ページ 71 行目	重回帰分析の計算量	タスク割り振り最適化処理の計算量