

個体差を考慮した動作割り振りによる 群ロボットの消費電力削減手法

寒竹俊之^{†1} 菅谷みどり^{†1}

概要：近年、災害や飲食業などの現場で、複数台ロボットの活用が検討されているが、その省電力化については、十分議論が行われていない。ロボットは同型の場合でも、行う動作やハードウェアにより消費電力の個体差が大きい。その個体差によっては運用したシステム全体の電力が増加する。この時、まとめて管理し、適切に運用すれば電力をより削減することが可能である。本研究では、複数台のロボットを用いるサービス全体の省電力化手法を提案し、シミュレータによって効果を確認した。その結果、ロボットの台数増加につれ、全体の電力の削減量も増加することが確認できた。しかし、単純な差分での評価だったことから、有効性が確認し辛いという問題が存在した。また既存手法では処理時間の問題から多数のロボットでの評価が困難という課題が存在した。本研究では上記の問題に対し、提案手法を改良、既存手法との比較を行った。その結果、削減効率には改善余地はあるものの処理時間については大幅な削減に成功した。また、有効性の評価指標を差分から、何割削減できたかを示す割合に変更し、既存手法と改良した手法の検証を行った。結果として、既存手法は、全体の電力を最大3%近く削減できることが確認できた。また、改良した手法も最大2.2%の削減が確認できた。この時、両手法も、台数の増加につれて、削減割合が増加しており、ロボットの増加につれて今回の手法は効果が増すことが確認できた。

キーワード：ロボット、ログデータ、タスク分散処理、電力管理、省電力、

Method for reducing power consumption of group robots by action allocation considering individual differences

TOSHIYUK KANTAKE^{†1} MIDORI SUGAYA^{†1}

Abstract: In recent years, utilization of multiple robots has been studied at sites such as disasters and restaurants, but discussion on power saving has not been thoroughly discussed. Even when the robot is of the same type, individual differences in power consumption are large depending on actions to be performed and hardware. Depending on individual differences, the power of the entire system that is operated increases. At this time, it is possible to reduce electric power further by managing it collectively and operating properly. In this research, we propose a power saving method for the whole service using multiple robots and confirmed the effect by simulator. As a result, it was confirmed that as the number of robots increased, the total amount of power reduction also increased. However, because it was an evaluation with a simple difference, there was a problem that the effectiveness was difficult to confirm. Moreover, there are problems that it is difficult to evaluate with many robots from the problem of processing time with the existing method. In this research, we improved the proposed method against the above problem and compared it with the existing method. As a result, although there was room for improvement in reduction efficiency, we succeeded in drastically reducing processing time. We also changed the evaluation index of effectiveness from the difference to a percentage showing how much reduction could be achieved, and verified the existing method and the improved method. As a result, it was confirmed that the existing method can reduce the total power by about 3% at the maximum. We also confirmed reduction of up to 2.2% in the improved method. At this time, the reduction ratio increased as the number of units increased, and it was confirmed that the effectiveness of this method increases as the number of robots increases.

Keywords : Robot, Logdata, Task distributed processing

1. はじめに

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]、様々な場面に応用され、より身近なものになっている[4]。日本では高齢化が進み、将来労働者が不足すると予想されており、介護や福祉、危険な場所など様々な分野で人の代替となるサービスロボットの市場は今後拡大すると予想されている[5]。

無人であるだけでなく自動での作業もロボットなら可能である。この時、作業効率が求められる場合、複数台で

行うことが合理的である。

例えば、災害現場などで扱われるロボットなどは、広範囲で迅速かつ確実に人命救助を行わなければならない状況が存在するが、この時も複数台のロボットを区画ごとに一台とすることにより、効率的な人命救助が可能である。

複数台のロボットの作業において、我々は以下の3つの課題があると考えた。1つ目は、故障管理である。これは、作業中の管理対象の個体の破損状況や、継続支援をさす。[6]。2つ目は、効率的な再配置である。これは、管理対象のロボットの物理的な最適配置により、作業効率を向上さ

^{†1} 芝浦工業大学
Shibaura Institute of Technology

せる支援である[7]。3つ目は、電力管理である。これは、作業効率を上げつつ全体の電力を削減する支援である。1つ目と2つ目の課題はすでに多くの研究がおこなわれている[6, 7]一方、3つ目の課題は十分に議論されていない。

電力削減に関する研究は、動作電圧・周波数の制御を行う DVFS(Dynamic Voltage and Frequency Scaling)[8]や、軽量化や回路の効率化、半導体を使って電力を制御する PWM(Pulse Width Modulation)など様々行われている。また、FPGA(Field Programmable Gate Array)を応用し、センサーの省電力につなげた研究も存在し[9]、省電力を考慮した移動目標認識システム[10]や飛行ドローンの位置認識システム[11]なども研究されている。これらの削減手法は、主にハードウェアに実装されることが効果的である。また、ハード自体の性能ではなく、計算アルゴリズムを工夫することによって省電力を実現する研究[12]も存在する。しかしながら、これらの研究は個々のロボットの消費電力の削減には効果的であると考えられるが、管理目的で利用することは困難である。ソフトウェア側からの時間特性に着目したタスクスケジューリング手法などの省電力手法も数多くなされておき[13, 14]、出荷後のロボットの動作特性に合わせて電力量を削減する手法については、まだ十分に議論されていない[15]。また、複数のロボットを利用するにあたり管理を容易にするための研究も存在する[16]が、あくまで作業効率を向上させるための物でありサービスの質を維持しつつ全体の電力を削減する議論は十分ではない。この課題に対して本研究では、全体の電力削減を目的とした管理手法を提案する。具体的には、ハードウェアと稼働状況に差が生じることから、その傾向を基にロボットと動作の組み合わせを変更することによって全体の電力を削減する手法を提案する。実現のために、まず、ハードウェアの消費電力傾向と稼働状況の差を反映できる予測モデルを構築し、個々の消費電力を予測できるようにした。そのモデルを基に、複数台のロボットの消費電力の総和を予測できるようにした。上記2つの予測を基に使用電力が最小となるロボットと動作の組み合わせを求め、適応させることで全体の電力を削減する。評価では、上記の予測から組み合わせの変更までをシミュレータとして実装し、提案手法を検証した。結果として、全体の電力が削減されていることが確認できた[17]。具体的にはロボットの台数が2倍になると、消費電力の削減量が2倍近くに増え、提案手法の有効性が示された。しかし、課題として、割合ではないため、具体的にどの程度が削減できているのかが分かり辛かった。そのため、削減量(差分)だけでなく、削減率(割合)によって再度評価を行った。

また、シミュレータには台数の増加につれ処理時間が増大するという問題点が存在した。本研究ではその問題点の解決のための改良と評価も行った結果について述べる。

本論文の構成は、第2章で関連研究とその課題、第3章

で提案、第4章でシミュレータでの評価、第5章でまとめと考察とした。

2. 関連研究

電力削減を目的とした技術は供給電圧および動作周波数を適切に制御し、プロセッサなどの電力を削減する DVFS などが存在する[8]。また、電力ロスが少ない PWM 制御なども電力削減手法として挙げられる。これらは、マイコン、CPU などの制御に用いることで最大の削減効果を得ることが出来る方式である。これらの手法は、ロボットなどにも応用可能であり、電力削減の効果が期待できる。一方、現場において様々なロボットが利用されるようなケースにおいて、現場に適応する形での電力削減手法を提供するものではない。また、組み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化発プラットフォームとして TK-SLP(Super Low Power Embedded Software Platform) [18]が存在する。この機能の1つとして消費電力モニタリング機能がある。これは、個別の組み込みシステムの電力量をモニタリングし、システムの管理機能に自動送信する仕組みを提供する。しかしながら、本仕組みは、監視制御はできるものの具体的な削減方式は開発者の実装に委ねられており、本仕組み自体が電力量を削減するものではない。

次に、データセンタなどにおける複数台の計算機を対象とした電力削減技術について書く[13]。この研究では空調に使用され冷却された空気を再度サーバの冷却に用いることや、計算機のタスク配置を変換することにより、約26%の電力削減を実現した。タスク配置の変換などは、複数台ロボットサービスにおける総電力削減へ応用可能と考えられるが、サーバールームなどの固定した空間を前提とした冷却方法などの手法の適用は困難である。

次に、クラウドによって複数のロボットを管理し、連携させながら動作させることでサービス継続を図った研究が存在する[14]。仲川らが行った研究では、クラウドサービスによって、複数のロボットをユーザーが利用可能な環境を実現した。また、タスク分散処理によってロボットサービスの具体的な管理方法と提案、プロトタイプを用いた検証を行っている[14]。提案方式は、バッテリー残量が多い個体により仕事させ、残量の少ない個体の充電できる時間を確保する方式であり、サービス全体の電力も最適化を考慮しているわけではない。そのため、ロボットの充電に必要な電力供給がなくなってしまう場合にはサービスの継続時間が増加するのには未知数である。

3. 提案

3.1 本研究の目的

先に述べたように、従来の研究では、複数台ロボットを対象とした、サービス全体の電力削減手法は十分議論され

ていない。複数台のロボットを運用する場合には、ロボット単体の消費電力を削減するだけでなく、管理により多くの電力削減が見込めると考えられる。このことから、本研究では、複数のロボットを使用した際の各ロボットの消費電力の総和を総電力と定義し、総電力の削減に着目した電力削減手法を提案する。我々は、複数台のロボットを動作させる予備実験を通じ[15]、ロボット使用場所や、提供するサービス状況により、ロボットが消費する電力に差がつくことがわかった。差を発生させる要因を下記の2点であると考えた。

- A) ハードウェアにおける電力消費特性の差
- B) 稼働の程度によって生じる電力消費特性の差

A)は、同じ機能を持っているロボットだとしても、経年劣化やセンサーの種類などによって消費電力に差が生じてしまうことから要因と考えた。次に、B)は、同性能の部品まで同じロボットでも使用される現場や、行う動作によって消費電力が変化することから要因として考えた。以上の二つを本研究では、Aをハードウェアの個体差、Bを動作の個体差と定義した。本提案では、ロボットの稼働状況のログから、上記2つの個体差を把握する。それを基に個々のロボットの消費電力予測式を構築する。次に、予測式から得た予測値から、各ロボットの動作を総電力が最小となる組み合わせを選ぶことで、総電力を削減するものとした。

次に具体的に実現するための提案手法について述べる。

3.2 提案手法

本研究では、目的を実現するために、具体的に次の(1)(2)の提案を行う。

(1)消費電力予測式の構築:

総電力消費を抑えるようにロボットのタスクを再分配する為には、精確な電力消費予測をする必要がある。その際、ハードウェアの個体差と動作の個体差の分割が容易なように構築した。その際、ロボットを実際に動作させて得たログデータを解析し、ハードウェアの個体差を把握した予測式を構築した。

(2)予測式をもとにした最適配置の計算:

(1)で得られた予測式を基に、総電力が最小となる組み合わせを求める方法を考案した。具体的には、移動量を示す定数が大きく、かつ動作当たりの電力消費傾向が大きい個体と、その逆に、定数が小さく、かつ動作当たりの電力消費傾向が小さい個体の稼働量を入れ替える。これにより、全体として電力消費量を削減できると考えられる。

本研究では(1)のために、実際にロボットから取得した値により、個体差が生じることを検証した。

次に(2)をシミュレータ上で設計、実装し、削減効果をシミュレータにより検証した。

3.2.1 消費電力予測式

ハードウェアの個体差と、動作の個体差を変数上で区別できるように予測式を構築した。まず、単純化のためにロ

ボット1台の動作が移動のみとし、速度を一定とした場合で予測式を構築した。このとき、あるロボットの消費電力(MC_x)[%]は、下記の予測式(1)により表すことができる。

$$MC_x = CPD_x^V \times D_y + CPS_x^V \times S_y \cdot \cdot \cdot (1)$$

予測式(1)は、ある個体 x における、ある速度(V)[cm/s]の移動動作による消費電力を1単位(ここでは1cm当たり)の消費電力(CPD_x^V)[%]とし、この値と稼働量である動作 y の累計移動距離(D_y)[cm]の積として求め、消費電力としたものである。停止状態からの動作による消費電力を1回当たりの停止からの動作による消費電力(CPS_x^V)[%]とある動作内の停止回数(S_y)の積として求めている。次に、総電力(MC_{all})[%]は予測式(2)を用いて消費電力を予測する。

$$MC_{all} = \sum_i^x MC_i \cdot \cdot \cdot (2)$$

予測式(2)はある個体 (i) の消費電力 (MC_x) の総量を求めることで総電力を予測する。

3.3 消費電力予測の変数

本節では、予測式の変数について説明する。先述した通りに、予測式においては、個体と動作の組み合わせの変更が容易とするため、ハードウェアの個体差と動作の個体差を分離できる形で構築した。このとき、3.1節 A)のハードウェアの個体差に相当する変数は CPD_x^V 、 CPS_x^V の2つとした。理由としては、実際にロボットを利用し行った実験[17]より、ロボットから取得したログ情報から、電力消費に影響力が高い物は以下の2点としたためである。

- A) 移動距離: 予備実験[15]により、ロボットの移動距離が消費電力に大きく影響していることが確認されている。また、実際にロボットを利用し行った実験[17]により、個体毎に差が生じていることが確認できた。よって、消費電力予測式には3.1節 A)のハードウェアの個体差を示す変数として CPD_x^V 、次に3.1節 B)の動作の個体差を示す変数として移動距離 D_y を加えた。
- B) 停止状態からの動作: 予備実験[15]において、障害物に衝突した時の回避行動の回数の増加が消費電力に影響を与えていることが確認できた。この時、回避行動において、通常の直線移動と異なる点は停止している状態から動作を行っている点である。よって、増加の原因として、停止状態からの動作と考えた。これより停止回数が上昇するにつれて消費電力が上昇すると考えた。このことより、よって、消費電力予測式には3.1節 A)のハードウェアの個体差を示す変数として CPS_x^V 、次に3.1節 B)の動作の個体差を示す変数として停止回数 S_y を加えた。
- C) 速度: 実際にロボットを利用し行った実験[17]により、速度の変化によって上記2つの個体差が変化することが確認できた。そのため、ハードウェアの個体差を示す変数 CPD_x^V 、 CPS_x^V の2つには速度による区別を

するため V を付加した. また, この V は動作の個体差を示す変数でもある.

上記の3点より, 式(1)には3.1節 A)のハードウェアの個体差として, 変数の CPD_x^V , CPS_x^V の2つを加え, 対応する動作の個体差として, 変数の D_y , S_y , V の3つを加えた.

3.3.1 シミュレータの設計

電力削減にあたっては3.2節に示した, 総電力が最小となる組み合わせを求める. 本手法の有効性はシミュレータを用いて検証する. 検証にあたり, シミュレータにて提案手法を実装し, 予測式で得られた値を入力として用いる方法で行う. 予測値はそれぞれ, 1個体の消費電力を求めるには式(1)を用いて求め, 総電力を求めるためには式(2)を用いて求める. まず, 1個体のロボットの消費電力の計算方法について述べる. 個体の消費電力は式(1)において, ハードウェアの個体差を示す変数(CPD_x^V , CPS_x^V)の値に対して, 動作の個体差を示す変数(D_y , S_y)の値を割り当て計算する. 次に総電力の計算方法について述べる. 総電力は式(1)の計算を全ての個体で行い, 式(2)によって総電力を計算する.

本研究における個体と動作の入れ替えというものは, 式(1)において変数である CPD_x^V と CPS_x^V に対して, 別の動作 y の変数 D_y と S_y を別の値と入れ替えることを示す. また, 個体と動作の組み合わせは式(1)において, 変数である CPD_x^V と CPS_x^V に対して, 変数 D_y と S_y の組み合わせのことである.

まず, 提案で削減可能な総電力を検証するために, 個体と動作の組み合わせが最悪となった場合も検証した. このとき, 総電力が最小となる個体と動作の組み合わせを求め, 総電力が最大となる個体と動作の組み合わせを求め, 総電力の最大値を求める. また, 個体と動作の組み合わせが最悪の場合ではない場合を検証するため, 最初に設定された個体と動作の入れ替えを行う前の総電力を基準値とした. 予備的な実装では基準値に対しての最小値の差分, 最大値に対しての最小値の差分の2つを検証することによって, 本研究の手法が有効かどうかを検証した[17]. しかし, 削減量(差分)のみでは全体の電力がどれほど削減できているのかの把握は困難である. そこで, 今回は, 評価指標を削減量(差分)だけではなく, 削減率(削減されている割合)も採用し再度検証を行い, 削減量と削減率の推移を確認した.

図1は改良前のシミュレータをブロック図として示した.

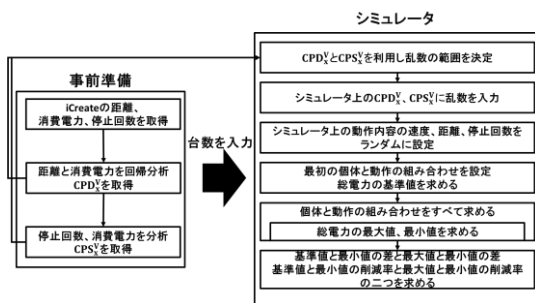


図1 シミュレータのシステム概要図

Figure 1 System outline of simulator

3.3.2 シミュレータ内での計算方法

次に, 上記のシミュレータ内での計算方法について述べる. 最初に, 変数の説明を行う. シミュレータにおいてハードウェアの個体差を示す変数を $SimCPD_x^V$ と $SimCPS_x^V$ とした. これはある個体 x のときの変数である.

次に, 動作の個体差を示す変数を $WD_y^{WV_y}$, $WS_y^{WV_y}$, WV_y とした. また, 個体 x の消費電力を MC_x とした. 次にその総電力の変数についてだが, これは4種類存在する. まず, 単純に総電力を示す変数を MC_{all} とする. このとき, 本手法を適応する前の MC_{all} を $MC_{all}Start$, シミュレータ上で求めた MC_{all} の最大値を $MC_{all}Max$, 最小値を $MC_{all}Min$ とした. 変数の意味については表1に記載した.

表1 変数の説明

Table 1 Variable of Description

変数	説明
CPD_x^V	個体 x の速度 V での 1cm 当たりの消費電力
CPS_x^V	個体 x の速度 V での 1 回当たりの停止からの動作による消費電力
$SimCPD_x^V$	シミュレータ上での個体 x の速度 V での 1cm 当たりの消費電力
$SimCPS_x^V$	シミュレータ上での個体 x の速度 V での 1 回当たりの停止からの動作による消費電力
WV_y	仕事ナンバリー y での速度
$WD_y^{WV_y}$	仕事ナンバリー y の速度 WV_y での距離
$WS_y^{WV_y}$	仕事ナンバリー y の速度 WV_y での停止回数
MC_x	ある個体 x の消費電力
MC_{all}	全個体の消費電力の合計
MC_{all}	手法適応前の個体と動作の組み合わせの
$MC_{all}Max$	総電力が最大となる組み合わせの MC_{all}
$MC_{all}Min$	総電力が最小となる組み合わせの MC_{all}

上記の内, ハードウェアの個体差を示す変数は, シミュレータによって式(3)と式(4)の範囲で乱数を設定される. 次に, 動作の個体差を示す変数については, 5m 移動して 800 回衝突などの実際のデータには程遠い値を設定されないようにする必要がある. そのため, ユーザーが設定できる基準値を設定し, その値にさらにある程度操作できるランダムな数値を足すことにより設定される.

$$SimCPD_x^V = CPD_{ave}^V + CPD_{random}(-CPD_{sta-dev}^V \leq CPD_{random} \leq CPD_{sta-dev}^V) \dots (3)$$

$$SimCPS_x^V = CPS_{ave}^V + CPS_{random}(-CPS_{sta-dev}^V \leq CPS_{random} \leq CPS_{sta-dev}^V) \dots (4)$$

上記の式(3)と式(4)について補足する. この2つの式は以前の研究[17]において実機の動作実験のログデータから取得した値を基に変数を構成されている. 具体的には CPD_{ave}^V と $CPD_{sta-dev}^V$ はログデータにて取得した CPD_x^V の平均と標準偏差を用いており, CPS_{ave}^V と $CPS_{sta-dev}^V$ も同様である. これは可能な限り実機に近い値が出力されるように範囲を設定しているためである.

次に、個体の消費電力の計算と総電力の計算方法について述べる。個体の消費電力の計算は式(1)を用いて行い、総電力の計算は式(2)を用いて行う。具体的には、個体 x の消費電力はシミュレータにおいてのハードウェアの個体差を示す変数と、動作の個体差の変数を用いて計算した。総電力は、個体の消費電力を最初に入力された台数 N 分の計算し、総和を求めることで計算した

3.3.3 シミュレータの動作

ここでは、シミュレータにユーザーから台数 N が入力されてからのシミュレータの動作を説明する。シミュレータは入力された N 個分の SimCPD_x^V と SimCPS_x^V と $\text{WD}_y^{\text{WV}_y}$, $\text{WS}_y^{\text{WV}_y}$, WV_y を生成する。その際には、3.3.2 節で述べたように値が設定される。次に、生成した変数をそれぞれ組合せ、式(1)を使い、全ての個体の消費電力 MC_x を計算する。次に、式(2)を使い、総電力 MC_{all} を求める。その時、既存の最大値と比較し、値が大きかった場合は、その MC_{all} を最大値とする。最小値はその反対を行う。その後式(1)においての SimCPD_x^V と SimCPS_x^V の2つの変数に対して $\text{WD}_y^{\text{WV}_y}$, $\text{WS}_y^{\text{WV}_y}$, WV_y の3つの変数の組み合わせを変更し、再度計算を行う。それを全ての組み合わせが検証されるまで続ける。最後に、 $\text{MC}_{\text{allStart}}$ と $\text{MC}_{\text{allMin}}$, $\text{MC}_{\text{allMax}}$ と $\text{MC}_{\text{allMin}}$ の差分と削減率を求め終了する。

次に、評価実験について述べる。

4. 評価実験

4.1 目的

先の提案手法を実装したシミュレータを用いて、以下の検証を行う。

1. 個体毎の消費電力の差を考慮し動作を入れ替えることで総電力の削減が可能か
2. 同時に扱う個体の台数が増加した時の削減量の数値を比較により、複数台のロボットを扱うサービスに適しているか
3. 総電力が最小となる場合の個体の消費電力と総電力が最大となる個体の消費電力の比較

4.2 評価方法

前回の研究[17]では、検証は5台から11台までで行ったが今回は、10台まで行った。検証した速度は、シミュレータが設定できる7つの中からランダムに設定した。移動距離と停止回数は前回行ったものと同様に範囲を設定した。つまり移動距離は5000cm~10000cmの範囲とし、停止回数は300~800の範囲で設定した。また、試行回数としては台数毎に500回行った。その平均を取ることで、乱数の影響をなるべく減らした。

4.2.1 評価結果

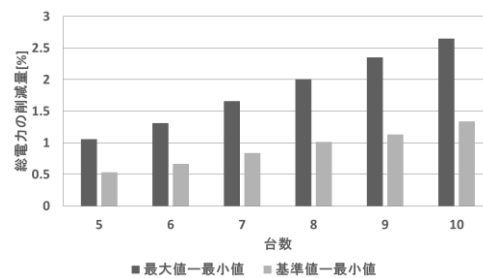


図 2 シミュレータの結果：削減量にて表示

Figure 2 Simulator result: Displayed in reduction amount

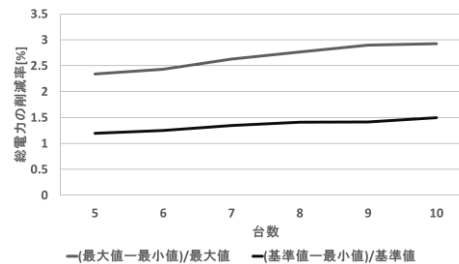


図 3 シミュレータの結果：削減率にて表示

Figure 3 Result of simulator: displayed in reduction rate

結果を図 2, 図 3 それぞれに示した。まず、図 2 について説明する。図 2 は、縦軸が削減量を示しており、横軸が台数を示している。使用したログデータが期待のバッテリー残量を百分率で示していたために単位は%となっている。凡例の「最大値-最小値」は総電力の最大値と最小値の差分である。これは本研究の提案手法を適用しなかった時の最悪値と提案手法で求めた最小値を比較したものである。

次に、凡例の「基準値-最小値」は手法適用前の総電力の基準値と最小値の差異を示している。以前の研究[17]で述べたが、この結果より台数が増えるにつれ削減量の上昇が確認できた。5台の時の削減量と10台の時の削減量を比較すると、2倍以上の差が確認できる。

次に、図 3 について説明する。図 3 は、縦軸が削減率を示しており、横軸が台数を示している。凡例の「(最大値-最小値)/最大値」は最小値との比較対象が最大値の時の削減率であり、凡例の「(基準値-最小値)/基準値」は同様に基準値の時の削減率である。縦軸の単位が図 2 と同様に%だが、図 3 の場合は最小値が比較している値と比べ、何%削減できたかを示している。台数の増加につれて、緩やかに削減率が上昇していることが確認できる。

4.3 考察

4.2 節により本研究の提案手法では、個体数が増加するにつれて総電力の削減量が増加することが確認できた。この結果は以前にも同様の結果が得られていた[17]。この時は、最小の個体と動作の組み合わせを求めることは、最終的に1個体の消費電力を削減することに繋がり、結果として、台数の増加とともに削減量が増えると考えていた。しかし、新しい個体が増えるに連れて比例に近い増加量とな

るはずであった。しかしながら、図2より、10台の時の値は5台の時の値より2倍以上となっており、疑問であった。ここで新たな評価指標である削減率の実験結果を見ると、こちらの値も台数の増加とともに、緩やかながら値が上昇している。このことから、台数の増加によって単純な比例より大きい形で削減量が上昇していったと考えられる。

4.4 問題点と改良

上記のシミュレータの問題点としては処理時間の増加が挙げられる。10台以降のシミュレータの処理時間を以下に示す(図4)。

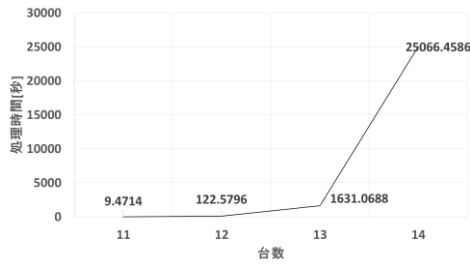


図4 シミュレータの処理時間

Figure 4 Processing time of the simulator

図4について説明する。図4において、縦軸は処理時間(単位は秒)で示しており、横軸はシミュレートした台数を示している。処理時間の推移をみると台数が追加するにつれて、大きく増加していることが確認できた。シミュレータが組み合わせを求め際の計算量が原因と考えられた。現シミュレータの計算量Oを見てみると、全ての個体と動作の組み合わせを求めて、最小値を計算していることから、計算量はO(n!)である。この値は、巡回セールスマン問題の総当たりによる計算量と同値であり、台数を増やした際のシミュレートが困難である。ここで、我々は処理時間を削減したシミュレータを製作した。

手法としては、式(1)において、ハードウェアの個体差を示す変数である CPD_x^V と CPS_x^V のどちらかの値と、それに対応する動作の個体差を示す変数である D_y と S_y のどちらかの値をソートしておく。次に、ハードウェアの個体差を示す変数の数値が高い順にそれに対応する動作の個体差の変数の値が高い物を組み合わせることで最大値とする。最小値は、最大値と反対に、動作内容の変数の値が少ない物を順に組み合わせることで求める。この手法において、処理時間の要因はソート部分となる。ソートの部分はクイックソートによって実現しており、この最悪値は $O(n^2)$ である。シミュレータにおいては2回ソートを行うことから、計算量は $O(2n^2)$ となる、これは、手法1に比べ大幅に削減された値であり、処理時間の削減が見込める。

図5は、改良後の手法で作成したシミュレータのブロック図として示した内容である。事前準備部分は変更していないため割愛する。

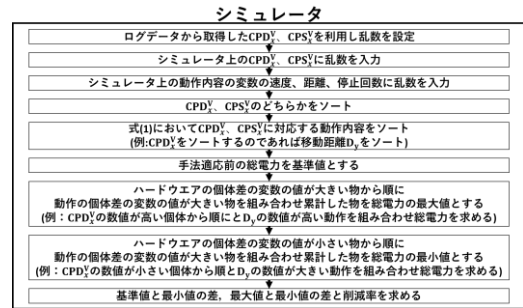


図5 改良したシミュレータのシステム概要図

Figure 5 System schematic of the improved simulator

次に、改良前の手法(以下手法1)で作成したシミュレータと改良後の手法(以下手法2)で作成したシミュレータを比較し、検証を行った。

4.5 検証

本研究の目的としては、総電力の削減を行っているため、

1. 削減効率
2. 処理時間

動作内容においての速度に値するパラメータは、今回はシミュレータにおいて、検証できる最も低速な値である5cm/sで固定した。このシミュレータは実際にロボットを利用し行った実験[17]の値を反映しているため、低速な値の時が最も値が変動する。これは、削減効率を検証するにあたり、値の変動が多いことで、新旧の比較がしやすいと考えたからである。

4.5.1 削減効率の検証

削減率を比較した図を以下に示す(図6)。

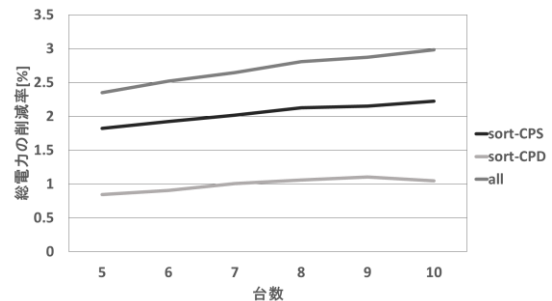


図6 削減率の比較

Figure 6 Comparison of reduction rates

図6は縦軸が最大値に対して、最小値が何%削減されているかを示しており、横軸がシミュレートにおいて設定した台数である。sortと付いている凡例は手法2で組み合わせを求めたときの値である。Sort-CPSは、式(1)の変数の CPS_x^V を手法2で適用して求めた値を示している。sort-CPDも同様である。それぞれを比較すると、手法1が最も削減率が大きく、手法2をsort-CPDで適応させたときの削減率が最も小さい。また、手法2で求めた値のsort-CPS、sort-CPDはそれぞれ値が異なることが確認できる。この中でも、手法1が最も削減できている。

次に個体毎の消費電力の比較を行った(図7、図8)。

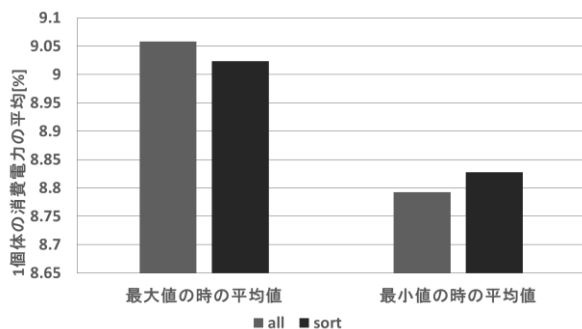


図7 1個体の消費電力の平均値

Figure 7 Average power consumption of one individual

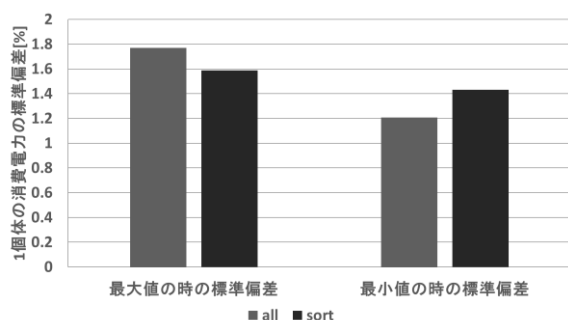


図8 1個体の消費電力の標準偏差

Figure 8 Standard Deviation of Power Consumption of One Individual

まず、図7の説明を行う。図8において、縦軸は1個体の消費電力の平均を示している。横軸がそれぞれ、最大値の時の個体の平均値と最小値の個体の平均値である。凡例の all は手法1の時の値を示し、凡例の sort は手法2の時の値を示している。手法1と手法2の値に共通することは、最大値よりも最小値の時の1個体の消費電力の平均値の値が小さいことである。ただし、最大値の時の値は手法1より手法2の値が小さく、最小値の時の値はその逆となっていることが確認できる。

次に、図8についての説明を行う。図8において、縦軸は1個体の消費電力の標準偏差を示している。横軸はそれぞれ、最大値の時の個体の標準偏差と最小値の時の物を示しており、凡例は図7と同じである。図7と同様に、最大値の時の値よりも最小値の時の値の方が小さい。手法1と手法2についての比較も同様で最大値の時の値は手法1の値が手法2の値より大きく、最小値の時はその逆となっている。

4.5.2 削減効率についての考察

最初に、図6においてどの手法2よりも、手法1の方が削減率は高い要因として考えられることを述べる。要因としては、手法1は組み合わせを全て求めているため、最小値と最大値を確実に求めることが可能であるためだと考えられる。次に、手法2においてソートした変数によって結果が異なる要因について述べる。考えられる要因としては、ソートする変数の影響の大きさが原因と考えられる。この

シミュレータでは、実際のロボットを用いた実験[17]によって得たデータを使って乱数を設定している。このことが原因で CPS_x^V の値は CPD_x^V よりも大きく変動するように設定されている。よって、 CPD_x^V をソートしても CPS_x^V の値とそれに対応する動作を示す変数 S_y の影響が大きいくまくだ削減が出来ていないと考えられる。逆に CPS_x^V をソートしたときはその影響の大きさから削減率が上昇した物と考えられる。影響の大きさによって削減率が変わってしまう課題は、影響の大きさによって値が変わるスコアのような物を判断基準とすることで改善できる可能性がある。

次に、1個体の消費電力についてだが、平均値と標準偏差両方に共通することとして、最大値の値は最小値の値より大きいことと、手法1の最大値の値は手法2の値より大きく、最小値の時は逆であることが挙げられる。以前の研究[17]でも、述べたように、最小となる動作と組み合わせを求めることは、全ての個体の消費電力を削減させることに繋がり、個体の消費電力を平均化させることにも繋がる。手法2の総電力の実験結果を見ると、削減率が低いことから、うまく削減できていない。このことが、結果として、個体毎の消費電力に影響を与えていると考えられる。

4.5.3 処理時間の検証

次に処理時間を比較した結果を以下に示す(図9, 図10)。

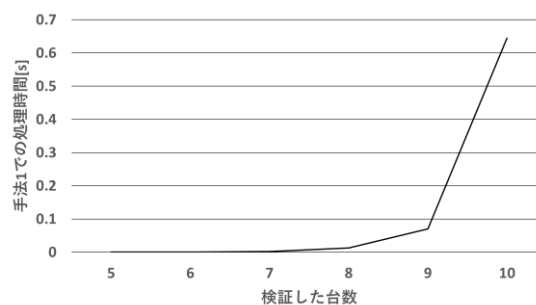


図9 手法1でのシミュレータの処理時間の推移

Figure 9 Transition of processing time of simulator in method 1

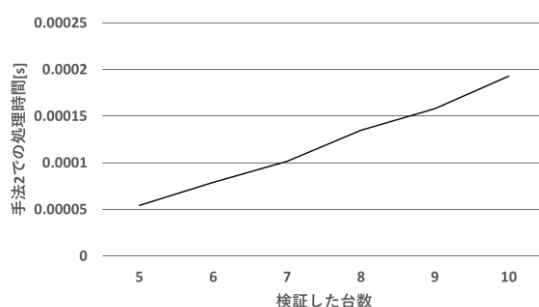


図10 手法2でのシミュレータの処理時間の推移

Figure 10 Transition of processing time of simulator in method 2

図9, 図10ともに、縦軸が処理時間[秒]を示しており、横軸がシミュレータにおいて検証した台数である。図9は手法1における処理時間の推移となっており、図10は手法2における処理時間の推移となっている。手法

1は入力する台数が増加するにつれて処理時間が指数関数的に増大している。次に、手法2の処理時間は、台数の増加に比例して、増大している。しかしながら、手法1と比べ、処理時間の上昇は一定でゆるやかである。また、10台の時の処理時間を比較すると、手法1と比べ、手法2は大幅に削減出来ていることが確認できる。

5. まとめ

研究では、総電力の削減の手法としてロボットの動作に使用する消費電力の差異を考慮して動作を入れ替えるという手法を提案した。改良した手法では、1つの変数をそのままソートしていたため他の変数の影響が考慮できなかったが、式に与える影響によって重みを付け、第三のスコアによってソートすることによって改善が可能だと考えられる。今後は課題を解決しながら、データをリアルタイムにクラウド上に存在するサーバに送信し、サーバでデータを分析し動的に動作を割り振ることによって総電力の削減を目指していきたい。また、現状ではiRobotCreate1のみでの実験[17]だが、空中を飛行するドローンなども用いて再度実験を行い、この手法で効果的に削減が可能かなどを検証していきたい。また、現在の提案手法は総電力の削減を目的としているが、稼働時間に影響があった場合にどちらを優先するべきかなども今後は考えたい。

謝辞 本研究を行うにあたり、省電力研究の芝浦工業大学、工学部情報工学科宇佐美公良先生、農工大学、工学部、情報工学科並木美太郎先生、産業技術大学院大学、中野美由紀様からアドバイスをいただき感謝いたします。

尚、本研究はJSPS 科研費 15K00105 の助成により実施いたしました。合わせて感謝申し上げます。

参考文献

- [1] Burgner-Kahrs, J. et al. : Continuum Robots for Medical Applications: A Survey, IEEE Transactions on Robotics, VOL. 31, NO. 6, pp. 1261-1280, 2015.
- [2] Teodoros, T. & Hu, H. : 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] Soakage, T. 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] Maneewarn, T. et al. : Survey of Social Robots in Thailand, International Electrical Engineering Congress (iEECON), pp. 19-21(2014).
- [5] 経済産業省. ロボット新戦略. 平成 27 年度版 .2015<http://www.meti.go.jp/press/2014/01/20150123004/20150123004b.pdf>(参照 2017-07-25)
- [6] Otani, M. et al. :Cooperation among multiple robots by

collecting broken robots in large-scale structure assembly, IEEE/SICE International Symposium on System Integration (SII), pp. 20-22(2011).

[7] 今泉貴敬ほか : ボロノイ分割を応用した無線中継移動ロボットの配置制御, 電気学会論文誌 D(産業応用部門誌), VOL. 132, NO. 3, p.381-388(2012)

[8] 高須雅義ほか : 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

[9] Albo-Canals, J. et al. : Embedded low-power low-cost Camera Sensor based on FPGA and its applications in mobile robots, 19th IEEE International Conference on Electronics, Circuits, and Systems (ICECS), pp. 336-339,(2012).

[10] Pan, S 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).

[11] Eckert, J. 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).

[12] Kim, Y. et al. : BRAIN: A Low-Power Deep Search Engine for Autonomous Robots”, IEEE Micro, VOL. 37, NO. 5, pp. 11-19(2017).

[13] 菅沼孝二ほか : サーバへのタスク配置がデータセンタの総消費電力に与える影響の評価, 信学技報, vol.113, no.472, p.185-190(2014)

[14] 仲川幸子ほか : 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, マルチメディア、分散、協調とモバイル(DICOMO2013)シンポジウム, p.3-p.10(2013)

[15] 清水航平ほか : 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).

[16] Ravankar, A. 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).

[17] 寒竹俊之ほか : 複数台の移動ロボットのバッテリー残量を考慮した総電力削減手法, 組込みシステムシンポジウム 2017(2017)

[18] 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-Kernel Super-LowPower)の研究開発. 平成 26 年 .2014.http://www.soumu.go.jp/main_content/000323271.pdf(参照 2017-07-25)