

環境発電型センサシステムのための データ中心型タスクスケジューリング方式

黒木 琴海^{1,a)} 小寺 志保^{2,b)} 倉田 成人^{3,c)} 濱本 卓司^{4,d)} 猿渡 俊介^{5,e)}

受付日 2016年2月8日, 採録日 2016年8月12日

概要: 本稿では, 筆者らが進めている軍艦島モニタリングプロジェクトにおけるタスクスケジューリングについて述べる. 軍艦島モニタリングプロジェクトは, 建築構造物の崩壊が進んでいる軍艦島において映像や音声, 加速度データといった崩壊現象のデータを収集し, 建築構造解析に貢献することを目的としている. 軍艦島モニタリングでは太陽光発電を電源として用いるため, 限られたエネルギーを効率的に利用する仕組みが求められる. 本稿では DC-LQ (Data Centric LQ-Tracker) と BLT allocation (Battery Level Task allocation) の2つの手法から構成されるデータ中心型のタスクスケジューリング方式として, BAAD scheduling (BAattery Aware Data centric scheduling) を提案する. DC-LQ によってセンサノードのスリープ時間の変動を抑えることで, 限られたエネルギーでデータを偏りなく取得する. さらに BLT allocation によって, エネルギー残量に応じて実行するタスクを決定する. シミュレーションによる評価によって, BAAD scheduling が先行研究と比較して優れた性能を備えていることを示す.

キーワード: 軍艦島モニタリング, タスクスケジューリング

Data Centric Task Scheduling for Energy Harvesting Sensor Systems

KOTOMI KUROKI^{1,a)} SHIHO KODERA^{2,b)} NARITO KURATA^{3,c)} TAKUJI HAMAMOTO^{4,d)}
SHUNSUKE SARUWATARI^{5,e)}

Received: February 8, 2016, Accepted: August 12, 2016

Abstract: Energy harvesting sensor systems, like our Battleship Island monitoring system, need to acquire a lot of data and use energy efficiently. Our Battleship Island monitoring system acquires data of images, sounds, and acceleration of collapsing buildings for structural health monitoring. The system is powered by solar energy. To enable the efficient use of solar-powered energy, this paper proposes BAAD scheduling (BAattery Aware Data centric scheduling) which consists of DC-LQ (Data Centric LQ-Tracker) and BLT allocation (Battery Level Task allocation). DC-LQ decides sleep time as the fluctuation is decreased. BLT allocation decides to perform tasks by the battery level. Evaluations using computer simulation show that BAAD scheduling outperforms conventional task scheduling.

Keywords: Battleship Island monitoring, task scheduling

¹ 静岡大学大学院総合科学技術研究科
² 静岡大学大学院情報学研究科
³ 筑波技術大学産業技術学部産業情報学科
⁴ 東京都市大学工学部建築学科
⁵ 大阪大学大学院情報科学研究科
a) kuroki@aurum.cs.inf.shizuoka.ac.jp
b) koder@aurum.cs.inf.shizuoka.ac.jp
c) kurata@a.tsukuba-tech.ac.jp
d) thama@tcu.ac.jp
e) saru@ist.osaka-u.ac.jp

1. はじめに

センサネットワークの発達によって, 実空間における様々なデータを取得できるようになった. さらに, センサネットワークを他の分野の学問と組み合わせることで, 人々の生活に新たな価値をもたらすことも可能になりつつある. センサネットワークと建築の分野を融合した技術に, 構造ヘルスマニタリング [1-5] がある. 構造へ

ルスモニタリングとは、建築構造物にセンサを張り巡らせ、センサから取得したデータを用いて建築構造物の安全性を分析する技術である。構造ヘルスマニタリングがありとあらゆる建築構造物に展開された場合、崩壊の予兆を十分早く検知することができれば、崩壊の前に人々を避難させることも可能になる。

しかしながら、現状では、建築構造物が崩壊する際の基準となるデータが少なく、崩壊現象の予兆を検知することは極めて困難である。実際の建築構造物において崩壊時のデータを取得しようとする、人命が危険にさらされる可能性があり、データを取得することができない。現在はシミュレーションを用いて人為的に崩壊現象を発生させているが [6]、経年劣化などによる複雑な崩壊現象までは網羅できない。

実際の建築構造物が崩壊する際のデータを取得することを目的として、筆者らは、軍艦島で崩壊現象のビッグデータを収集する軍艦島モニタリングプロジェクト [7] に取り組んでいる。軍艦島は、今まさに建築構造物の崩壊が進んでいる環境であり、経年劣化などによる建築構造物の複雑な崩壊現象のデータが取得できる。軍艦島モニタリングプロジェクトでは、建築構造物にセンサやカメラを設置して、映像や音声、加速度などのデータを取得する。取得したデータは無線通信を用いて本土に送信する。しかしながら、軍艦島には発電所がなく、軍艦島モニタリングシステムに必要な電力は全て太陽光発電で供給しているため、使用できる電力には限りがある。

本稿では、限られたエネルギーから効率良くセンサデータを取得するためのタスクスケジューリング方式を検討する。各センサノードはソーラーパネルとバッテリーを具備しており、利用するエネルギーはノード毎に独立しているため、タスクスケジューリングはセンサノード毎に独立して行われる。限られたエネルギーから効率良くセンサデータを取得するためには、以下の3つの要件を考慮する必要がある。1つ目は、複数のタスクをスケジューリングすることである。各センサノードには、データの取得や送信などの複数のタスクが存在する。複数のタスクをどのようにスケジューリングするかによって、取得できるデータや消費電力が変化する。2つ目は、センサから取得するデータ量を最大化することである。より多くのデータが取得できれば、詳細な分析が可能になる。3つ目は、データロス時間の最大値を最小化することである。データロス時間を短縮することができれば、データが取得された時間帯のばらつきを削減することができる。

本稿では、3つの要件を考慮したデータ中心型のタスクスケジューリング方式として、BAAD scheduling (Battery Aware Data centric scheduling) を提案する。BAAD scheduling は、DC-LQ (Data Centric LQ-Tracker) と BLT allocation (Battery Level Task allocation) の2つから構

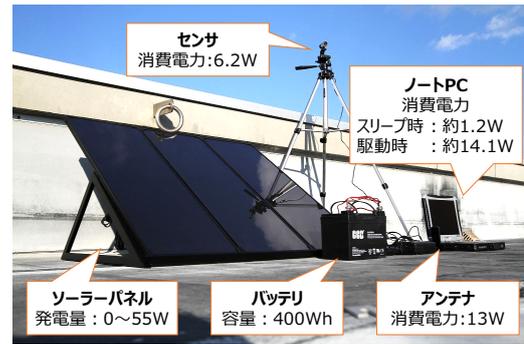


図 1 モニタリングシステム
Fig. 1 Monitoring system

成される。DC-LQ では、エネルギーが枯渇しないように、センサノードの駆動時間とスリープ時間を決定する。BLT allocation では、センシングや通信といった複数のタスクから、システムのバッテリー残量を元に行うタスクを決定する。シミュレーションを用いて評価した結果、BAAD scheduling が多くのデータを取得しつつ、データロス時間を短縮できることが分かった。

本稿の構成は以下の通りである。2 節では、軍艦島モニタリングシステムの詳細を述べる。3 節では、タスクスケジューリングにおける課題について述べる。4 節では、提案手法である BAAD scheduling と、BAAD scheduling を構成する DC-LQ と BLT allocation について述べる。5 節では、シミュレーションを用いて、既存タスクスケジューリング方式と BAAD scheduling を比較する。6 節では、既存研究について述べる。7 節では、DC-LQ における平滑化や軍艦島モニタリングにおけるタスクについて議論する。最後に 8 節でまとめとする。

2. 軍艦島モニタリング

軍艦島モニタリングプロジェクト [7] は、軍艦島において崩壊中の建築構造物の映像や音声、加速度といったデータを収集することで、建築構造解析に貢献することを目指すプロジェクトである。軍艦島は、長崎県にある無人島であり、現在も経年劣化による建築構造物の複雑な崩壊現象が発生している。軍艦島でデータを取得することで、人為的に発生させた崩壊現象の検証では得られなかった、建築構造物が実際に崩壊する瞬間のデータを取得することができる。

図 1 に、軍艦島モニタリングで利用するハードウェアと、それぞれのハードウェアにおける電力供給量、電力消費量を示す。軍艦島には電源やネットワークがないため、太陽光発電を用いてセンシングシステムを駆動し、無線を用いて本土へとデータを送信する必要がある。電力を得るためのソーラーパネルとして公称最大出力が 55 [W] の Coleman PVS-55 W, 400 [Wh] のバッテリー、ノート PC として TOUGHBOOK CF-195W1ACS を使用する。CF-

195W1ACS にはセンサと無線通信のデバイスが USB で接続されている。CF-195W1ACS の電力消費量を計測した結果、スリープ時で約 1.2 [W]、駆動時で約 14.1 [W] であった。さらにセンサデータの取得に 6.2 [W]、本土への無線通信に 13 [W] の電力を消費する。

3. タスクスケジューリングにおける課題

軍艦島モニタリングに向けたタスクスケジューリングでは、センサノード上にあるセンシングや無線通信を用いたデータ送信などの複数のタスクを扱う。また、軍艦島モニタリングシステムは太陽光発電で得られる電力を利用しており、限られた電力をどのように利用するかが課題である。

太陽光発電を用いる以上、悪天候が続いて長時間エネルギーがほとんど得られなくなる可能性がある。7.3 節でも述べる通り、実際に悪天候の日が 12 日間続く場合もある。このような場合に常時モニタリングを行うためには、通信を行わない場合でもノード毎に約 480 [Wh] のバッテリーと約 630 [W] の太陽光パネルが必要となる。現在使用しているシステムと比較すると太陽光パネルの大きさは約 11 倍となる。加えて、常に通信をも行う場合には約 650 [Wh] のバッテリーと約 1100 [W] の太陽光パネルが必要となる。

コストが非常に高くなること、長崎市の許可を取った上で小さな漁船で軍艦島に上陸すること、軍艦島は崩壊中の建物が入り組んだ地形であること、ソーラーパネルは建物の屋上に設置する必要があることなどの理由から、ソーラーパネルを何枚も設置するのは困難な状況にある。すなわち、限られたエネルギーを効率的に利用するタスクスケジューリングは必須となる。

軍艦島モニタリングの目的は建物の崩壊の過程を解析することにあるため、崩壊現象が発生する瞬間のデータを取得することが何より重要になる。エネルギーの制約がなければ、常にモニタリングシステムを駆動してあらゆるデータを取得するというのが理想である。最終的には、カメラ、音声センサ、加速度センサを用いて崩壊現象の予兆を検知できるようにして、崩壊現象の瞬間を逃さないようにスケジューリングすることを想定している。崩壊現象の予兆検知のためにも、まずはあらゆる時間のデータを万遍なく、より多く取得する必要がある。

以上を踏まえて、軍艦島モニタリングに向けたタスクスケジューリング方式は次の 3 つの要件を考慮する必要がある。1 つ目は、センシングや通信といった複数のタスクをスケジューリングすることである。タスクによって電力消費量が異なり、センシングと通信をそれぞれどのようにスケジューリングするかによってエネルギーの消費量が変化する。また、センサデータの取得量や、センシングしてから本土へデータを送信するまでの遅延時間などが大幅に変化する。

2 つ目は、センサから取得するデータ量を最大化するこ

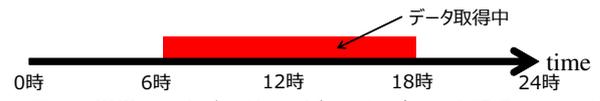


図 2 蓄積エネルギーがあれば直ちにデータを取得 (ASAP)
Fig. 2 Acquire data immediately if the energy remains (ASAP)

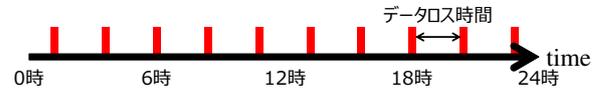


図 3 一定間隔でデータを取得
Fig. 3 Acquire data at fixed intervals

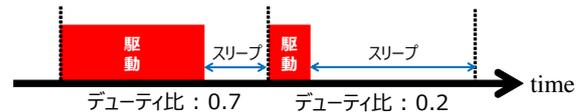


図 4 LQ-Tracker
Fig. 4 LQ-Tracker

とである。より多くのデータを収集することで、崩壊現象の過程をより詳細に分析することが可能となる。例えば、5 秒間データを取得した場合、10 秒間の崩壊現象に対しては現象の発生状況を一部しか知ることができない。一方で、20 秒間データを取得した場合、10 秒間の崩壊現象に対しては現象の発生状況を全て克明に知ることができる。

3 つ目は、最悪データロス時間を最小化することである。データロス時間とはセンサノードが連続してデータを取得していない時間であり、最悪データロス時間とはデータロス時間の最大値である。例えば、同じデータ量において、最悪データロス時間が 1 時間のデータと 12 時間のデータを考える。最悪データロス時間が 1 時間のデータでは、最悪の場合でも、1 日のうち何時頃その崩壊現象が起こったかを把握することができる。一方で、最悪データロス時間が 12 時間のデータでは、半日単位でしか崩壊現象のタイミングを把握することはできない。

2 つ目と 3 つ目の「データ取得量の最大化」と「最悪データロス時間の最小化」はトレードオフの関係にある。データ取得量を最大化することで最悪データロス時間が大きくなり、最悪データロス時間を最小化することでデータ取得量が小さくなる。

データ取得量を最大化するには全てのエネルギーをデータ取得のために使用しなければならない。スリープ状態からの復帰やスリープ状態の継続のためにはエネルギーが必要であることから、一度起動したら一気にデータを取得して、出来る限り起動する頻度を少なくする必要がある。その場合、データ取得量は最大となるが、起動の間隔が長くなることでデータロス時間は長くなる。

一方で、最悪データロス時間を最小化するためには、起動とスリープの間隔を可能な限り短くして頻繁に起動とスリープを繰り返す必要がある。起動時にはオーバーヘッドが

発生するため、頻繁な起動はより多くのエネルギーを消費する。結果として、データの取得に使用できるエネルギーが減ることになるため、1回の起動でデータを取得する時間を短くしなければならない。

「データ取得量の最大化」と「最悪データロス時間の最小化」のそれぞれの要件で定量的に必要とされる性能はセンサの種類や目的に依存する。データ取得量と最悪データロス時間のどちらを重視するかによって適切なスケジューリングは異なる。本稿では、最悪データロス時間の短縮に重きを置きつつも、データ取得量も可能な限り増やすタスクスケジューリングを模索する。

センサから取得するデータ量を最大化する方法として、蓄積エネルギーがあれば直ちにデータを取得する方法 (ASAP: As Soon As Possible) が考えられる。図2に ASAP の例を示す。時間軸上に四角で示した部分でデータを取得する。ASAP は蓄積エネルギーがある限りデータを取得し続けており、取得できるデータ量は最大となる。しかしながら、蓄積エネルギーが枯渇した場合、新たにエネルギーが蓄積されるまでデータを取得できず、取得できるデータの時間帯に偏りが生じる。取得できるデータの偏りは最悪データロス時間を大きくする。図2の例では、18時以降のデータが全く取得できておらず、取得したデータはすべて日中のデータとなっている。

最悪データロス時間を小さくする方法として、一定間隔でデータを取得する方法が考えられる。図3に一定間隔でデータを取得する場合の例を示す。時間軸上に四角で示した部分でデータを取得する。一定間隔でデータを取得すると、データが取得された時間帯の偏りがなくなり、最悪データロス時間は小さくなる。しかしながら、蓄積エネルギーに余裕がある場合でも、取得できるデータ量は変化しない。

センサネットワークにおいて、エネルギーの残量によって動作を制御する研究として、LQ-Tracker [8] が挙げられる。LQ-Tracker は、適応制御理論を用いてセンサノードのデューティ比を決定することで、バッテリーレベルを目標値に維持する。デューティ比とは、センサノードが駆動とスリープを繰り返す中で1つのサイクルにおける駆動時間の割合のことである。図4に LQ-Tracker の動作例を示す。デューティ比が0.7の場合、1サイクルの70%の期間はセンサノードを駆動させる。1サイクルごとにその時のバッテリー残量に従ってデューティ比を決定することで、常にバッテリー残量に応じたタスクスケジューリングが可能となる。しかしながら、LQ-Tracker はデューティ比が変わるたびにデータロス時間が変動するという課題がある。図4に示した LQ-Tracker では、1サイクルの長さが固定であるため、デューティ比が0.7から0.2に変化すると、スリープ時間が急増する。スリープ時間の変動が大きくなると取得データに時間的な偏りが生じ、最悪データロス時間が増

Algorithm 1 BAAD scheduling

```

1: loop
2:    $B \leftarrow$  current battery level  $\in [0, 1]$ 
3:    $T_{work}, T_{sleep} \leftarrow$  dclq( $B$ )
4:   blt( $B, T_{work}$ )
5:   Sleep for  $T_{sleep}$  seconds
6: end loop
    
```

表 1 Algorithm 1 で使用する変数、関数

Table 1 Variables and functions used in Algorithm 1

変数、関数	説明
B	センサノードの現在のバッテリーレベル
T_{work}	タスクの実行時間
T_{sleep}	センサノードのスリープ時間
dclq(B)	B を元に T_{work} と T_{sleep} を決定する関数 詳細は 4.1 節
blt(B, T_{work})	B を元に実行するタスクを決定・実行する関数 詳細は 4.2 節

加する。

4. BAAD scheduling

3節での議論を元に、センサから取得するデータ量の最大化と、最悪データロス時間の最小化を考慮した手法として、データ中心型タスクスケジューリング方式、BAAD scheduling を設計した。BAAD scheduling は、DC-LQ と BLT allocation から構成される。DC-LQ は、センサノード上のタスクの実行時間とスリープ時間を決定する。BLT allocation は、センサノードが起動したときにバッテリー残量に応じてセンサノードで実行するタスクを決定・実行する。

Algorithm 1 に BAAD scheduling のアルゴリズム、表1に Algorithm 1 で使用する関数と変数を示す。 B はセンサノードの現在のバッテリーレベルで、0 から 1 の範囲で表される。 T_{work} はタスクの実行時間、 T_{sleep} はセンサノードのスリープ時間である。dclq(B) は B を元に T_{work} と T_{sleep} を決定する関数である。詳細は 4.1 節で述べる。blt(B, T_{work}) は B と T_{work} を元に実行するタスクを決定・実行する関数である。詳細は 4.2 節で述べる。

Algorithm 1 では、まず DC-LQ を用いてタスクの実行時間とセンサノードのスリープ時間を決定する。Algorithm 1 の 2 行目で現在のバッテリーレベルを取得して、3 行目で T_{work} と T_{sleep} を決定する。次に、決定された実行時間とスリープ時間に応じて、BLT allocation を用いてタスクを実行する。タスクが終了するとセンサノードをスリープさせる。Algorithm 1 の 4 行目で B を元に実行するタスクを決定、 T_{work} [sec] の間タスクを実行する。タスクが終了すると、6 行目で T_{sleep} [sec] の間センサノードをスリープ

させる． T_{sleep} [sec] が経過するとセンサノードを再び起動させて *Algorithm 1* の 2 行目から繰り返す．

4.1 DC-LQ

3 節で説明した通り，LQ-Tracker はバッテリーレベルを元にセンサノードのデューティ比を決定する手法である．DC-LQ では，3 節で述べた LQ-Tracker で動的に決定されるデューティ比を用いて，タスクの実行時間とセンサノードのスリープ時間を決定する．ただし，3 節で述べた通り，LQ-Tracker はデューティ比が変わるたびにスリープ時間が変動する．それに対して，DC-LQ ではスリープ時間の変動が小さくなるように 1 サイクルの時間を調節する．LQ-Tracker が 1 サイクルの時間を固定して駆動とスリープの時間を調節するのに対して，DC-LQ ではデューティ比とスリープ時間に合わせて 1 サイクルの時間を変動させることでスリープ時間の変動を抑える．

まず，センサノードがスリープ状態から復帰するたびに LQ-Tracker によってデューティ比を決定する．次に，決定したデューティ比を元にスリープ時間を算出する．スリープ時間には，ある一定の時間を 1 サイクルとして，デューティ比に従った場合のスリープ時間の指数移動平均を使用する．指数移動平均を取ることで，直近のエネルギーの変動に対応しつつ，スリープ時間の変動を抑えることが可能となる．スリープ時間 T_{sleep} [sec] を式 (1) に示す．

$$T_{\text{sleep}} = \begin{cases} T_{\text{cyclebase}}(1 - D) & \text{(初回)} \\ T_{\text{cyclebase}}(1 - D)\alpha + T'_{\text{sleep}}(1 - \alpha) & \text{(2 回目以降)} \end{cases} \quad (1)$$

$T_{\text{cyclebase}}$ [sec] は 1 サイクルの基準となるあらかじめ決められた時間， D は LQ-Tracker によって決定されたデューティ比を意味している． α は指数移動平均法における平滑化係数であり，スリープ時間の変動をどの程度許容するかを表す． α は 0 から 1 の範囲であらかじめ設定する． α が小さいと，新たなデューティ比によるスリープ時間 $T_{\text{cyclebase}}(1 - D)$ より前回のスリープ時間 T'_{sleep} に重きを置いて T_{sleep} を算出するため， T_{sleep} の変動が小さくなる．一方で， α が大きいと， T'_{sleep} より $T_{\text{cyclebase}}(1 - D)$ に重きを置いて T_{sleep} が決定され， T_{sleep} の変動が大きくなる． α と $T_{\text{cyclebase}}$ に関してはそれぞれ 5.4 節，5.5 節で詳細に検証する．

初回の T_{sleep} は事前に決定する $T_{\text{cyclebase}}$ を 1 サイクルとして，デューティ比に従った場合のスリープ時間を設定する． $T_{\text{cyclebase}}$ が短い場合，頻繁に駆動とスリープが繰り返されることによって，データロス時間は短くなる．しかしながら，頻繁なスリープからの起動が電力を消費するため，取得できるデータ量が少なくなる．一方で， $T_{\text{cyclebase}}$ が長い場合，駆動とスリープの繰り返しがなくなり，データロス時間は長くなる．しかしながら，一度センサノード

Algorithm 2 DC-LQ

Input: $B \in [0, 1]$
Output: $T_{\text{work}}, T_{\text{sleep}}$

- 1: $D \leftarrow \text{lq_tracker}(B)$
- 2: $T'_{\text{sleep}} \leftarrow$ previous sleep time
- 3: **if** T'_{sleep} does not exist **then**
- 4: $T_{\text{sleep}} \leftarrow T_{\text{cyclebase}}(1 - D)$
- 5: **else**
- 6: $T_{\text{sleep}} \leftarrow T_{\text{cyclebase}}(1 - D)\alpha + T'_{\text{sleep}}(1 - \alpha)$
- 7: **end if**
- 8: $T_{\text{work}} \leftarrow \frac{T_{\text{sleep}}D}{1 - D} - T_{\text{wake}}$

表 2 *Algorithm 2* で使用する変数，関数

Table 2 Variables and functions used in *Algorithm 2*

変数，関数	説明
B	センサノードの現在のバッテリーレベル
T_{work}	タスクの実行時間
T_{sleep}	センサノードのスリープ時間
D	センサノードのデューティ比
$\text{lq_tracker}(B)$	B を元に LQ-Tracker を用いてデューティ比を決定する関数
T'_{sleep}	前回のスリープ時間
$T_{\text{cyclebase}}$	1 サイクルの基準となる時間
α	平滑化係数
T_{wake}	センサノードが起動にかかる時間

が起動すると長時間データを取得できる．2 回目以降は， $T_{\text{cyclebase}}$ と新たなデューティ比から決定したスリープ時間と前回のスリープ時間 T'_{sleep} から指数移動平均を計算して新たなスリープ時間として設定する．

式 (1) によりスリープ時間が決定されると，スリープ時間とデューティ比に応じて 1 サイクルの時間 T_{cycle} [sec] を決定する． T_{cycle} を式 (2) に示す．

$$T_{\text{cycle}} = \frac{T_{\text{sleep}}}{1 - D} \quad (2)$$

T_{sleep} が T_{cycle} に対するスリープ時間の割合である $1 - D$ を占めるように T_{cycle} を決定する． T_{cycle} が決定されると，タスクの実行時間 T_{work} [sec] を決定する．式 (3) に T_{work} を示す．

$$T_{\text{work}} = T_{\text{cycle}}D - T_{\text{wake}} \quad (3)$$

T_{wake} [sec] はセンサノードの起動にかかるシステム固有の時間を意味している． $T_{\text{cycle}}D$ はセンサノードが駆動している時間全体である．センサノードが駆動している時間全体から T_{wake} を引いた値が T_{work} となる．式 (2) と式 (3) より， T_{work} は式 (4) となる．

$$T_{\text{work}} = \frac{T_{\text{sleep}}}{1 - D}D - T_{\text{wake}} \quad (4)$$

Algorithm 2 に *Algorithm 1* の関数 dclq に相当する DC-LQ のアルゴリズムを，表 2 に *Algorithm 2* で使用する変数を示す． B はセンサノードの現在のバッテリーレベ

Algorithm 3 BLT allocation

Input: $B \in [0, 1], T_{\text{work}}$

```

1: for  $i = 1$  to  $n$  do
2:   if  $B \geq B_i$  then
3:     invoke  $\tau_i$ 
4:   end if
5: end for
6: Run invoked tasks for  $T_{\text{work}}$  seconds
    
```

表 3 Algorithm 3 で使用する変数, 関数

Table 3 Variables and functions used in Algorithm 3

変数	説明
B	センサノードの現在のバッテリーレベル
T_{work}	タスクの実行時間
n	タスクの数
$\tau = \{\tau_1, \dots, \tau_n\}$	スケジューリングの対象となるタスク
B_i	タスク τ_i の実行バッテリーレベル

ル, T_{work} はタスクの実行時間, T_{sleep} はセンサノードのスリープ時間, D はセンサノードのデューティ比である. $\text{lq_tracker}(B)$ はバッテリーレベル B を元に LQ-Tracker を用いてデューティ比を決定する関数である. T'_{sleep} は前回のスリープ時間を表している. 前述した通り, $T_{\text{cyclebase}}$ は 1 サイクルの基準となる時間, α はスリープ時間の変動を許容する割合を表す平滑化係数であり, それぞれ事前に決定される. T_{wake} はセンサノードが起動にかかる時間を意味するシステム固有の値である.

DC-LQ はバッテリーレベル B を入力とし, 実行時間 T_{work} とスリープ時間 T_{sleep} を出力とする. Algorithm 2 の 1 行目ではバッテリーレベル B を元に LQ-Tracker を用いてデューティ比 D を決定する. T'_{sleep} が存在しない場合, Algorithm 2 の 4 行目で式 (1) の初回の式を用いて T_{sleep} を算出する. T'_{sleep} が存在する場合, Algorithm 2 の 5 行目から 6 行目で式 (1) の 2 回目以降の式を用いて T_{sleep} を算出する. T_{sleep} が決定されると, Algorithm 2 の 8 行目で式 (4) に従って T_{work} を決定する.

4.2 BLT allocation

BLT allocation では, センサノードがスリープから復帰した際に, バッテリーレベルに応じて実行するタスクを決定して実行する. 頻繁に実行する必要のないタスクはバッテリーレベルの低いときには実行しないようにすることで, エネルギー消費を抑えることができる.

Algorithm 3 に Algorithm 1 における関数 blt に相当する BLT allocation のアルゴリズムを, 表 3 に Algorithm 3 で使用する変数を示す. B は現在のバッテリーレベルである. n はタスクの数, τ はスケジューリングの対象となるタスクの集合を表している. B_i はタスク τ_i の実行バッテリーレ

表 4 タスクの対応関係の例

Table 4 Example of correspondence of tasks

タスク	実行時間 C_i	実行バッテリーレベル B_i
τ_1	T_{work}	0
τ_2	T_{work}	0.8

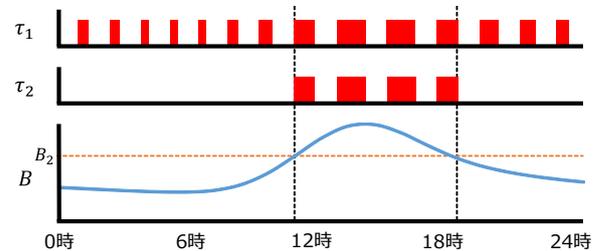


図 5 BLT allocation

Fig. 5 BLT allocation

ルである. センサノード上にタスク $\tau = \{\tau_1, \tau_2, \dots, \tau_n\}$ が存在する場合, それぞれのタスク τ_i は, 実行時間 C_i [sec] と実行バッテリーレベル $B_i \in [0, 1]$ で特徴づけられる. 各タスクの実行時間は DC-LQ によって決定された T_{work} となる.

Algorithm 3 では, 現在のバッテリーレベル B と DC-LQ によって決定された T_{work} を入力として用いる. Algorithm 3 の 3 行目で現在のバッテリーレベル B と i 番目のタスクの実行バッテリーレベル B_i を比較する. 現在のバッテリーレベルが実行バッテリーレベル以上だった場合, Algorithm 3 の 4 行目で i 番目のタスクを起動する. Algorithm 3 の 3 行目から 5 行目をタスクの数だけ繰り返す. 最後に 6 行目で起動したタスクを T_{work} [sec] の間実行する.

例として, 軍艦島モニタリングにおける, タスク τ_i と実行バッテリーレベル B_i , 実行時間 C_i の対応関係を表 4 に示す. τ_1 は映像や加速度データのセンシング, τ_2 は本土とのデータの通信のタスクを想定している. タスク τ_1 の実行バッテリーレベル B_1 は 0, タスク τ_2 の実行バッテリーレベル B_2 は 0.8 とする. 軍艦島モニタリングでは, より多くのデータを取得することが重要であるが, データをリアルタイムで本土に送信する必要はない. 2 つのタスクは表 4 に基づき, センシング τ_1 は常に実行され, 通信 τ_2 はバッテリーに余裕がある場合にのみ実行されることになる.

図 5 に表 4 のタスクを対象とした軍艦島モニタリングにおける BLT allocation の動作を示す. 図 5 では, 1 日のバッテリーレベルが下のグラフの実線のように推移した場合, それぞれのタスクがどのようにスケジューリングされているかを表している. 通信のタスク τ_2 は, バッテリーレベルが実行バッテリーレベル B_2 以上となる日中のみ実行されている. センシングのタスク τ_1 は実行バッテリーレベル B_1 が 0 であるため, バッテリーレベルに関係なく実行されている.

表 5 評価パラメータ

Table 5 Evaluation parameters

駆動時の消費電力	14.1 W
スリープ時の消費電力	1.2 W
センシングの消費電力	6.2 W
通信の消費電力	13 W
ソーラーパネルの最大出力	55 W
バッテリーの容量	400 Wh
T_{wake}	60 sec
1秒で取得できるデータ量	1 MB

5. 評価

5.1 評価環境

BAAD scheduling の有効性を確認するために計算機シミュレーションによってデータ取得量と最悪データロス時間を評価した。BAAD scheduling の性能を相対的に評価するために、以下の3つのタスクスケジューリング方式を比較した。

(1) ASAP (As Soon As Possible)

タスクスケジューリングを行わず、バッテリー残量がある限りセンシングと通信をし続ける手法である。各デバイスの起動時に要するオーバーヘッドが少ないため、データ取得量の目標値となる。

(2) LQ-Tracker

3節で述べた、適応制御理論を用いた LQ-Tracker [8] を前提としたタスクスケジューリング手法である。文献 [8] での評価と同様に、1 サイクルの時間を 1800 秒、デューティ比の平滑化に用いる平滑化係数を 10^{-3} としている。

(3) BAAD scheduling

4節で述べた提案手法である。DC-LQ と BLT allocation を用いて、データ取得量の最大化と最悪データロス時間の最小化を実現する。

共通のパラメータを表 5 に示す。表 5 のパラメータは全て 2 節で示した軍艦島モニタリングシステムを想定している。システムの消費電力は駆動時が 14.1 [W]、スリープ時が 1.2 [W] とした。システムは、センシング時に駆動時の消費電力に加えてさらに 6.2 [W]、通信時に駆動時の消費電力に加えてさらに 13 [W] 消費する。また、ソーラーパネルの最大出力は 55 [W]、バッテリーの容量は 400 [Wh]、センサノードの起動にかかる時間 T_{wake} を 60 [sec]、1 秒で取得できるデータ量を 1 [MB] とした。BLT allocation における通信の実行バッテリーレベルは、軍艦島モニタリングを想定して 0.8 とした。

評価では、1 日の発電量の推移をモデル化することで、時間、季節、天候による太陽光発電の発電量を模擬した。図 6 に、太陽光発電による 1 日の発電量の推移の例を示す。

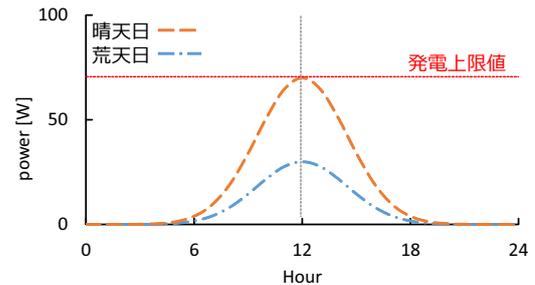


図 6 シミュレーションにおける発電量の推移

Fig. 6 Output power transition in the simulation

発電量は午後 12 時を頂点とする正規分布に従うと仮定することで、時間における発電量の推移を模擬した。また、発電上限値 (maximum generated power) を季節に応じて変化させることで、季節における太陽光の入射角の変化を模擬した。さらに、午後 12 時時点での発電量を 0 [W] から発電上限値までの範囲で 1 日の始めにランダムに決定することで、天候による発電量の違いを模擬した。図 6 の場合、晴天日は午後 12 時に発電上限値に到達するように発電量が推移するが、荒天日は午後 12 時時点での発電量が少なくなるように推移する。LQ-Tracker や ASAP に対する BAAD scheduling の相対的な比較を行うことが目的であるため、7 日間のシミュレーションを 10^4 回実行した際の平均値を評価結果として用いた。

5.2 発電上限値に対するデータ取得量の評価

BAAD scheduling のデータ取得量を評価するために、発電上限値に対するデータ取得量を取得した。発電上限値は 10 [W] から 100 [W] の範囲を 10 [W] 刻みで変化させた。また、BAAD scheduling における $T_{cyclebase}$ は 1800 [sec]、 α は 0.01 とした。図 7 に、発電上限値に対するデータ取得量を示す。縦軸がデータ取得量 [GB]、横軸は発電上限値 [W] である。図 7 から以下の 2 つのことが分かる。

1 つ目は、BAAD scheduling のデータ取得量が LQ-Tracker のデータ取得量を上回っていることである。例えば、発電上限値が 50 [W] のとき、BAAD scheduling は LQ-Tracker と比較して約 1.43 倍のデータ取得量を実現している。これは、BAAD scheduling が BLT allocation によってバッテリー残量が少ないときには通信を行わずにセンシングのみを行っているからだと考えられる。常にセンシングと同時に通信を行う LQ-Tracker よりもエネルギーの消費量が少なくなるため、より多くのエネルギーをデータの取得に利用できる。

2 つ目は、発電上限値が大きくなるにつれて ASAP のデータ取得量が BAAD scheduling のデータ取得量を上回っていることである。例えば、発電上限値が 30 [W] のときは BAAD scheduling が ASAP より約 1.22 倍のデータ取得量を実現しているが、発電上限値が 80 [W] のときには約 0.93 倍と少なくなっている。これは、発電上限値が大き

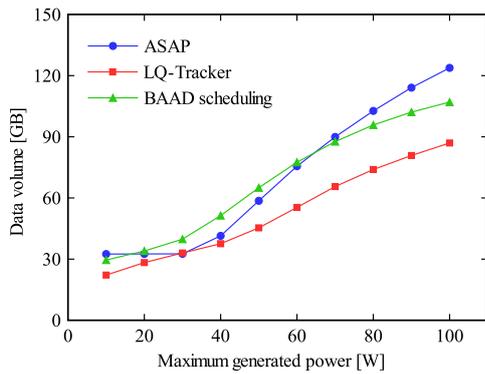


図 7 データ取得量

Fig. 7 Amount of acquired data

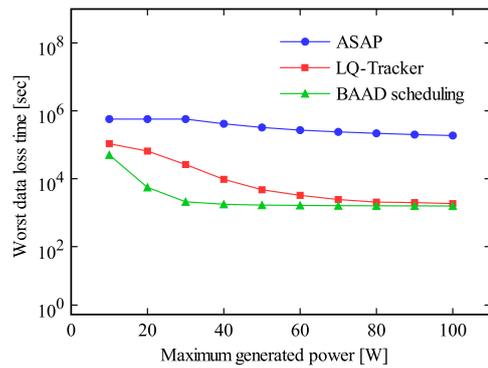


図 8 最悪データロス時間

Fig. 8 Worst data loss time

なるにつれて、ASAP ではセンサノードの起動時のオーバーヘッドが小さくなったからだと考えられる。発電上限値が小さい場合、ASAP ではセンサノードを起動させるだけでエネルギーが枯渇することが多くなってデータ取得量は少なくなる。一方で、発電上限値が大きい場合、センサノードを駆動している期間が長くなり、起動時のオーバーヘッドが相対的に少なくなった結果として多くのデータを取得できる。

5.3 発電上限値に対するデータロス時間の評価

BAAD scheduling の最悪データロス時間を評価するために、発電上限値に対する最悪データロス時間を取得した。発電上限値は 10 [W] から 100 [W] の範囲を 10 [W] 刻みで変化させた。BAAD scheduling における $T_{\text{cyclebase}}$ は 1800 [sec]、 α は 0.01 とした。図 8 に発電上限値に対する最悪データロス時間を示す。縦軸が最悪データロス時間 [sec]、横軸が発電上限値 [W] である。図 8 から、以下の 2 つのことが分かる。

1 つ目は、BAAD scheduling が常に最も小さい最悪データロス時間を達成していることである。特に発電上限値が小さい領域では、他の 2 つの手法より大幅に最悪データロス時間が小さくなっている。例えば、発電上限値が 30 [W] のとき、BAAD scheduling は ASAP と比較すると約 99.6%、LQ-Tracker と比較すると約 92.0%の最悪データロス時間を短縮できている。これは、BAAD scheduling が DC-LQ によってスリープ時間の変動を小さくしているからだと考えられる。

2 つ目は、発電上限値が大きくなるにつれて、LQ-Tracker の最悪データロス時間が小さくなっていることである。3 節で述べた通り、LQ-Tracker は 1 サイクルが固定であることで最悪データロス時間が大きくなりやすい。しかしながら、発電上限値が大きい場合、デューティ比が大きくなることで毎回のスリープ時間が相対的に短くなり、結果としてデータロス時間も短くなったと考えられる。

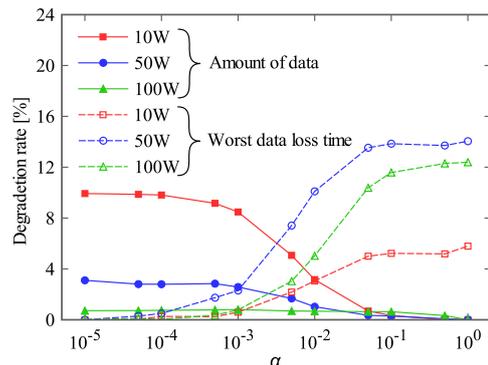


図 9 α を変化させた場合の評価

Fig. 9 Evaluation with varying α

5.4 α を変えた場合の評価

5.2, 5.3 節の評価では、BAAD scheduling における α を 0.01 とした。本節では α の値が BAAD scheduling の性能に与える影響を検証する。具体的には、 α を 10^{-5} から 10^0 まで変化させた場合のデータ取得量と最悪データロス時間を取得した。 α の変化による結果の変動を相対的に評価することを目的として、データ取得量と最悪データロス時間のそれぞれについて結果の悪化率を評価した。悪化率の算出手順を以下に示す。

- (1) 発電上限値が 10 [W] の場合において、 α を 10^{-5} から 10^0 まで変化させて 7 日間のシミュレーションを 10^5 回実行し、データ取得量と最悪データロス時間を計測する。
- (2) α を 10^{-5} から 10^0 まで変化させた結果の中で、データ取得量の最大値と最悪データロス時間の最小値をそれぞれ最も良い結果とする。
- (3) データ取得量と最悪データロス時間の両方について、最も良い結果とそれ以外の結果の差分を算出する。差分を最も良い結果で割った値を結果の悪化率とする。
- (4) 発電上限値が 50 [W]、100 [W] の場合も同様に算出する。

データ取得量と最悪データロス時間のどちらにおいても、悪化率が小さいほど高い性能であることを意味する。

図9に、 α の値を 10^{-5} から 10^0 の範囲で変化させた場合のデータ取得量と最悪データロス時間の悪化率を示す。 $T_{\text{cyclebase}}$ は全て1800 [sec]である。横軸が α 、縦軸が結果の悪化率である。図9から、以下の2つのことが分かる。

1つ目は、 α が小さくなるにつれて最悪データロス時間の悪化率が低下していることである。例えば発電上限値が50 [W]の場合、 α が 10^{-1} のときの最悪データロス時間の悪化率は約13.8%であるが、 α が 10^{-4} のときの悪化率は約0.5%である。これは、 α を小さくすることで、スリープ時間の変動が抑えられたからだと考えられる。 α が小さくなると、スリープ時間の算出では新たなデューティ比によるスリープ時間より、過去のスリープ時間が重視される。スリープ時間が前回のスリープ時間とほぼ同じ値になることで、スリープ時間の変動が小さくなり、最悪データロス時間が短縮できている。

2つ目は、発電上限値が小さい場合に、 α が小さくなるにつれてデータ取得量の悪化率が上昇していることである。例えば発電上限値が50 [W]の場合、 α が 10^{-1} のときデータ取得量の悪化率は約0.3%であるが、 α が 10^{-4} のとき悪化率は約2.8%である。これは、デューティ比の変動に応じて1サイクルの時間が変動することに起因すると考えられる。1サイクルの時間はスリープ時間とデューティ比から決定されるが、 α が小さいことでスリープ時間の変動も小さくなる場合、1サイクルの時間はデューティ比のみによって変動する。発電上限値が小さい場合、エネルギー残量が少なくなることでデューティ比も全体的に小さくなる。デューティ比が小さくなると1サイクルの時間も小さくなり、頻繁に駆動とスリープを繰り返して起動時に消費するエネルギーが大きくなる。すなわち、1サイクルに対する起動のオーバーヘッドが大きくなることで、データの取得に利用できるエネルギーが減少し、結果としてデータ取得量が小さくなる。

5.5 $T_{\text{cyclebase}}$ を変えた場合の評価

5.2, 5.3節の評価では、BAAD schedulingにおける $T_{\text{cyclebase}}$ を1800 [sec]とした。本節では $T_{\text{cyclebase}}$ の値がBAAD schedulingの性能に与える影響を検証する。具体的には、 $T_{\text{cyclebase}}$ を800 [sec]から2000 [sec]まで変化させた場合のデータ取得量と最悪データロス時間を取得する。 $T_{\text{cyclebase}}$ の変化による結果の変動を相対的に評価するために、データ取得量と最悪データロス時間のそれぞれについて結果の悪化率を評価した。悪化率の算出手順は5.4節の手順において α を $T_{\text{cyclebase}}$ に置き換えたものである。データ取得量と最悪データロス時間のどちらにおいても、悪化率が小さいほど高い性能であることを意味する。

図10に、 $T_{\text{cyclebase}}$ の値を800 [sec]から2000 [sec]の範囲で変化させた場合のデータ取得量と最悪データロス時間の悪化率を示す。 α の値は全て0.01である。横軸が

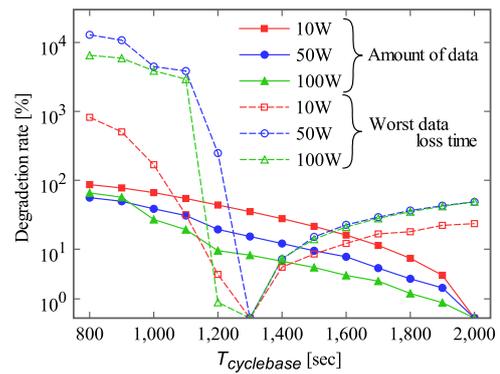


図10 $T_{\text{cyclebase}}$ を変化させた場合の評価
Fig. 10 Evaluation with varying $T_{\text{cyclebase}}$

$T_{\text{cyclebase}}$ 、縦軸が結果の悪化率である。図10から、以下の2つのことが分かる。

1つ目は、 $T_{\text{cyclebase}}$ が大きくなるにつれてデータ取得量の悪化率が低下していることである。例えば発電上限値が50 [W]の場合、 $T_{\text{cyclebase}}$ が1000 [sec]のときのデータ取得量の悪化率は約39%であるが、 $T_{\text{cyclebase}}$ が1800 [sec]のときの悪化率は約3%である。これは、 $T_{\text{cyclebase}}$ が大きくなるにつれて1サイクルの時間が増加することでセンサの起動時のオーバーヘッドが相対的に減少したからだと考えられる。DC-LQでは式(1)に示した通り $T_{\text{cyclebase}}$ を基準としてスリープ時間を算出するため、 α の値に関わらず $T_{\text{cyclebase}}$ が大きくなるとスリープ時間は増加する。スリープ時間が増加すると、同じデューティ比でも1サイクルの時間も増加する。例えばデューティ比0.5において、スリープ時間が100 [sec]の場合の1サイクルの時間は200 [sec]であるが、スリープ時間が500 [sec]の場合の1サイクルの時間は1000 [sec]となる。1サイクルの時間が増加すると、起動とスリープを繰り返す頻度が少なくなることによって起動に要するオーバーヘッドも少なくなり、結果としてより多くのデータを取得できる。

2つ目は、 $T_{\text{cyclebase}}$ が1300 [sec]の場合に最悪データロス時間の悪化率が最も低くなっていることである。これは、センサノードの起動時のオーバーヘッドに起因すると考えられる。 $T_{\text{cyclebase}}$ が1300 [sec]未満の場合、1サイクルの時間が短くなりすぎてセンサノードが起動した後のセンシング時間が十分に確保できない。センサノードが起動してもデータを取得しなければ、最悪データロス時間は増加する。一方で、 $T_{\text{cyclebase}}$ が1300 [sec]より大きい場合、1サイクルの時間と共にスリープ時間も長くなり、結果として最悪データロス時間が増加する。

5.6 要素毎の詳細な比較

5.5節までは、DC-LQとBLT allocationを組み合わせてBAAD schedulingとしての評価を行った。本節ではBAAD schedulingの各要素の効果を検証する。BAAD scheduling

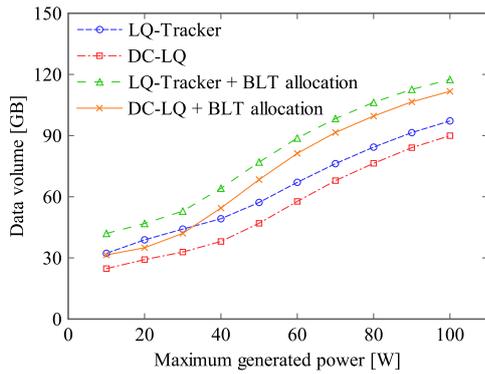


図 11 データ取得量 (要素毎の比較)

Fig. 11 Amount of acquired data (comparison of each element)

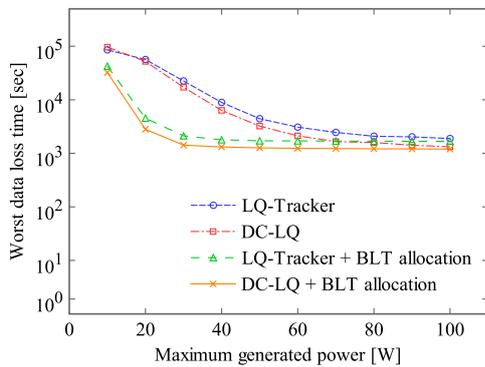


図 12 最悪データロス時間 (要素毎の比較)

Fig. 12 Worst data loss time (comparison of each element)

の各要素ごとに詳細な比較を行うことを目的として、

- (1) LQ-Tracker 単体
- (2) LQ-Tracker + BLT allocation
- (3) DC-LQ 単体
- (4) DC-LQ + BLT allocation

の 4 つのデータ取得量と最悪データロス時間を比較した。

データ取得量の結果を図 11 に示す。5.4 節の評価に基づいて、DC-LQ における α の値は 10^{-3} 、LQ-Tracker におけるデューティ比の平滑化係数は 10^{-3} とした。

図 11 の評価結果から、以下の 2 つのことが分かる。1 つ目は、BLT allocation を適用することでデータ取得量が増加することである。LQ-Tracker 単体と LQ-Tracker + BLT allocation、DC-LQ 単体と DC-LQ + BLT allocation をそれぞれ比較すると、最大で約 46% データ取得量が大きくなっている。2 つ目は、DC-LQ を適用することで LQ-Tracker よりもデータ取得量が少なくなることである。LQ-Tracker 単体と DC-LQ 単体、LQ-Tracker + BLT allocation と DC-LQ + BLT allocation をそれぞれ比較すると、最も差がある場合には DC-LQ を用いた方が約 25% データ取得量が少なくなっている。5.4 節における、 α を小さくしてスリープ時間を平滑化することでデータ取得量は悪化するという評価に沿った結果となっている。

最悪データロス時間の結果を図 12 に示す。図 12 の評価

結果から以下の 2 つのことが分かる。1 つ目は、DC-LQ によって最悪データロス時間が短縮されていることである。LQ-Tracker 単体と DC-LQ 単体、LQ-Tracker + BLT allocation と DC-LQ + BLT allocation をそれぞれ比較して DC-LQ の効果を算出すると、最大で約 38% の最悪データロス時間を短縮できている。2 つ目は、BLT allocation によっても最悪データロス時間が短縮されていることである。LQ-Tracker 単体と LQ-Tracker + BLT allocation、DC-LQ 単体と DC-LQ + BLT allocation をそれぞれ比較して BLT allocation の効果を算出すると、最大で約 95% の最悪データロス時間を短縮できている。

6. 関連研究

本研究は、構造ヘルスマニタリングと環境発電を用いたセンサネットワークシステムの研究に関連する。

構造ヘルスマニタリングは、建築構造物にセンサを設置してその建築構造物の性能を診断する技術である。建築構造分野においては、構造ヘルスマニタリングの理論面での研究は既に完成の域にあり [1-5]、現在では、センサネットワークを構造ヘルスマニタリングに活用する研究 [9-11] や、サービスの提供 [12] が進められている。本研究でも、センサネットワークを用いて実際の建築構造物の崩壊現象をモニタリングすることで実データを元にした建築構造解析技術への貢献を目指している。

環境発電を用いたセンサネットワークシステムを利用する研究としては、太陽光発電を用いた動物の生息環境モニタリング [13-15] が挙げられる。これらの研究は動物の生息状況という時間変化が比較的早い現象を対象としている。それに対して本研究では、軍艦島での建築構造物の劣化という時間変化が緩やかな現象を対象としてエネルギーの利用を効率化することを目的としている。

環境発電を用いたセンサネットワークシステムにおけるエネルギー利用の効率化を目的とした研究として、ハードウェアの設計によってエネルギー利用を効率化する研究 [16, 17] や、センサノードのデューティサイクリングの研究 [8, 16, 18] が挙げられる。本研究ではデューティサイクリングに着目している。

デューティサイクリングの研究 [16, 18] では、発電量の予測を元にデューティ比を決定する手法が提案されている。例えば、文献 [16] では、発電量の予測値と実際の発電量を用いて年単位の長期的な稼働を視野に入れてデューティ比を決定している。しかしながら、消費電力や発電量を正確に予測するためには、事前に使用するセンサノードの消費電力やシステムの発電量を知る必要がある。それに対して本研究では、バッテリーレベルのみを用いてデューティ比を決定している。

デューティサイクリングを利用したものとして、B-MAC [19]、A-MAC [20]、X-MAC [21]、ContikiMAC [22]

表 6 LQ-Tracker と DC-LQ の違い

Table 6 Difference of the LQ-Tracker and the DC-LQ

	LQ-Tracker	基本的な アイデア	DC-LQ
デューティ比	変動 + 平滑化	変動 + 平滑化	変動 + 平滑化
スリープ時間	変動	固定	変動 + 平滑化
1 サイクルの時間	固定	変動	変動

などの通信プロトコルの研究が挙げられる．例えば，B-MAC では，一定間隔のスリープとウェイクを繰り返してウェイクした際にプリアンプを受信したら完全にウェイクしてデータを受信する仕組みを導入することで低消費電力な通信を実現している．これらの研究は基本的には通信に要するエネルギーの最小化を目的としている．それに対して，本研究では，獲得できるエネルギー量の変動に対してデータ取得量の最大化と最悪データロス時間の最小化を目的としている．

7. 議論

7.1 平滑化に関する議論

4.1 節で述べた通り，提案手法の 1 つである DC-LQ はスリープ時間の平滑化を行う．また，DC-LQ でデューティ比を決定する際に使用している LQ-Tracker も，内部でデューティ比の平滑化を行っている．

LQ-Tracker も DC-LQ も平滑化している目的は同じである．文献 [8] に明記されている通り，LQ-Tracker において平滑化をしているのは，イベントの検出を行う場合にはできるだけ一定間隔でセンシングをした方が良いとの理由からである．これは本研究のモチベーションと同じである．

表 6 に，LQ-Tracker と DC-LQ の違いを示す．LQ-Tracker と DC-LQ の違いを表すのに重要な要素はデューティ比，スリープ時間，1 サイクルの時間の 3 点である．

表 6 に示しているように，LQ-Tracker ではデューティ比のみを平滑化することでスリープ時間の変動を抑えようとしている．仮にデューティ比を平滑化しない場合には，1 サイクルの時間が固定であることに起因してデューティ比に応じてスリープ時間の変動が大きくなる．一方で，文献 [8] での平滑化に関するパラメータ設定を変化させた場合の評価が示している通り，平滑化の度合いを大きくすることでデューティ比の分散は小さくなるものの，ある値を超えるとエネルギーを使い切って電源が OFF になる事態が発生し，結果として最悪データロス時間が増加する．このエネルギーが足りなくなるという問題は，本来はエネルギーを蓄えるのに必要なデューティ比と平滑化されたデューティ比が乖離することに起因している．

このような LQ-Tracker の平滑化の制約に対して，DC-LQ のアイデアは 2 つある．1 つ目は，表 6 に基本的なアイデアとして示しているように，LQ-Tracker で変動

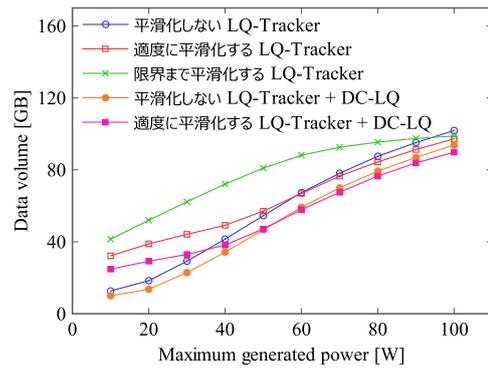


図 13 データ取得量 (平滑化の評価)

Fig. 13 Amount of acquired data (evaluation of smoothing)

させていたスリープ時間を固定にして，固定していた 1 サイクルの時間を変動させるというものである．スリープ時間を固定して 1 サイクルの時間をデューティ比に応じて決定することでスリープ時間の変動が抑えられる．しかしながら，この基本的なアイデアのままだと，固定とするスリープ時間をどのような値にすべきかという問題が発生する．

そこで提案手法の一部である DC-LQ では，2 つ目のアイデアとしてデューティ比の変化に応じてスリープ時間を変動させつつ，スリープ時間を平滑化している．このようなアプローチを採用することで，LQ-Tracker による平滑化にさらなる平滑化を加えることが可能となり，最悪データロス時間をより少なくすることが可能となる．

より厳密に平滑化の効果を評価することを目的として，

- (1) 平滑化しない LQ-Tracker
- (2) 適度に平滑化する LQ-Tracker
- (3) 限界まで平滑化する LQ-Tracker
- (4) 平滑化しない LQ-Tracker + DC-LQ
- (5) 適度に平滑化する LQ-Tracker + DC-LQ

の 5 つの比較を行った．平滑化しない LQ-Tracker は平滑化係数を 1，適度に平滑化する LQ-Tracker は平滑化係数を 10^{-3} ，限界まで平滑化する LQ-Tracker は平滑化係数を 10^{-5} とした．

図 13 にデータ取得量の比較結果を示す．図 13 から次の 3 つのことが分かる．1 つ目は，限界までデューティ比を平滑化した LQ-Tracker が最もデータ取得量が大きくなることである．例えば，発電上限値が 50 [W] の場合，平滑化しない LQ-Tracker と比較するとデータ取得量は約 1.5 倍となっている．2 つ目は，発電上限値が低い場合には，平滑化しない LQ-Tracker よりも適度に平滑化する LQ-Tracker の方がデータ取得量が大きくなることである．LQ-Tracker と DC-LQ のそれぞれにおいて LQ-Tracker で平滑化しないものと適度に平滑化するものを比較すると，発電上限値が 10 [W] の場合には両方とも適度に平滑化の方が約 2.5 倍のデータ取得量となっている．3 つ目は，DC-LQ を適用する

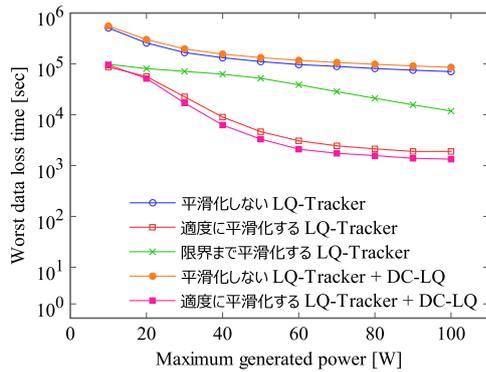


図 14 最悪データロス時間 (平滑化の評価)

Fig. 14 Worst data loss time (evaluation of smoothing)

ことでデータ取得量が小さくなることである。LQ-Tracker のみの場合と比較して、最大で約 26%データ取得量が小さくなっている。

図 14 に最悪データロス時間の比較結果を示す。図 14 から次の 2 つのことが分かる。1 つ目は、適度に平滑化する LQ-Tracker + DC-LQ が最も最悪データロス時間が短くなっていることである。平滑化しない LQ-Tracker 単体と比較すると最大で約 98%、限界まで平滑化する LQ-Tracker 単体と比較すると最大で 94%、適度に平滑化する LQ-Tracker 単体と比較すると最大で約 32%の最悪データロス時間を短縮できている。LQ-Tracker と DC-LQ の平滑化を組み合わせることで、単体で使用するよりも大きな効果が出ていることが分かる。2 つ目は、LQ-Tracker で限界までデューティ比を平滑化すると最悪データロス時間が大きくなることである。適度に平滑化する LQ-Tracker 単体と比較すると、最悪データロス時間は最大で約 12.7 倍となっている。前述したとおり、デューティ比の平滑化の度合いを極端に大きくしたことでエネルギーの変動に適応できず電源が OFF になる事態が発生していると考えられる。

7.2 軍艦島モニタリングにおけるタスクに関する議論

4.2 節の BLT allocation の説明や 5 節での評価では、センサード上のタスクをセンシングと通信の 2 つのみとして扱っていた。本節では、軍艦島モニタリングにおいて実際にスケジューリングの対象となり得るタスクについて議論する。

図 15 に、現段階での軍艦島におけるセンサードの設置場所を示す。軍艦島モニタリングでは、カメラ、音声センサ、加速度センサといった複数の種類のセンサを用いている。各センサードでは、センサや通信がそれぞれ異なる構成で構築されているため、センシングと通信のタスクも異なる。図 15 の 3 号棟のセンサードでは、カメラ 3 台、マイク 3 台を用いてセンサデータを取得して、無線通信を介してセンサデータを送信している [23]。図 15 の 70 号棟のセンサードでは、加速度センサ 5 台、マイク 1 台

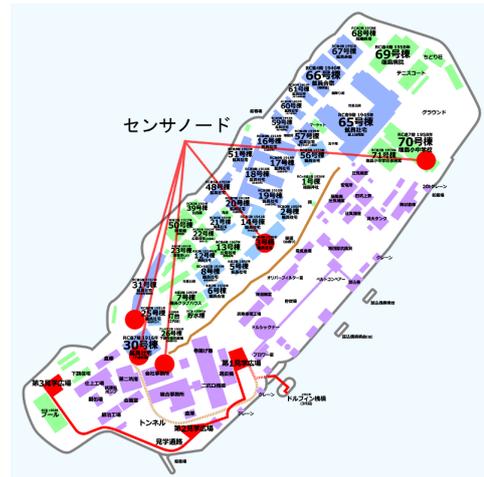


図 15 センサードの設置場所 [7]

Fig. 15 Installation location of sensor nodes [7]

でセンサデータを取得して、無線通信を介してセンサデータを送信している [25]。図 15 の 30 号棟のセンサードでは、加速度センサ 9 台、マイク 1 台でセンサデータを取得して、有線通信を介してセンサデータを送信している [24]。図 15 の 31 号棟のセンサードでは、加速度センサ 6 台、マイク 1 台でセンサデータを取得して、有線通信と無線通信を併用してセンサデータを送信している [24]。図 15 の 30 号棟前のセンサードでは、カメラ 1 台、マイク 1 台でセンサデータを取得して、無線通信を介してセンサデータを送信している [26]。

また、取得したデータはそのまま送信するのではなく、特徴量の抽出や圧縮といった処理を施してから送信するため、データ処理のタスクも存在する。データの種類によって重要度や必要とされるリアルタイム性が変化するため、データの種類ごとにそれぞれスケジューリングが必要となる。例えば、加速度データは建物の構造解析のために最も重要となるが、蓄積されたデータを後から一括して解析するため、頻繁に通信を行う必要はない。それに対して、崩壊現象の発生の有無など、最新の軍艦島の様子を出来る限り把握するために、加速度データよりも映像データを優先して送信することも想定している。

7.3 軍艦島の天候に関する議論

5 節ではモデル化した天候を前提として性能を評価していた。本節では、実際の軍艦島の天候下における BAAD scheduling の挙動について議論する。

太陽光発電において最も発電量が少なくなるのは積雪によって光が遮られる場合であるが、軍艦島がある野母崎ではほとんど積雪がない。すなわち、雨天日が続く場合が最もエネルギーを得られない状況となる。2015 年に悪天候 (雨天) の日が最も長く続いた期間は 6 月 7 日から 6 月 18 日までの 12 日間であった。この期間にモニタリングシス

テムを駆動させた場合、12日間でのデータ取得量は約56 [GB]、最悪データロス時間は約9時間となる。すなわち、悪天候が12日間続いた場合でも1日あたり2 [GB]以上のデータを取得でき、約9時間に1回以上はセンシングできるということになる。雨天時は発電量が少なくなるが、一切発電されないというわけではないため、悪天候の期間がさらに長くなった場合でも約11時間に1回はデータが取得できる。一方で、タスクスケジューリングを全く導入しない場合、すなわち3節で述べたASAPを用いてスケジューリングした場合には、データ取得量は約46 [GB]、最悪データロス時間は約275時間となる。

最も多くのエネルギーを得られる状況は、晴天日が続く場合である。2015年に晴天日が最も長く続いた期間は7月30日から8月11日までの13日間であった。この期間にモニタリングシステムを駆動させた場合、13日間でのデータ取得量は約205 [GB]、最悪データロス時間は約14分となる。すなわち、天候の良い日が続く場合は約14分に1回は必ずデータが取得でき、1日あたり約17 [GB]のデータが取得できるということになる。一方で、タスクスケジューリングを導入しない場合、すなわち3節で述べたASAPを用いてスケジューリングした場合には、データ取得量は約383 [GB]、最悪データロス時間は約17時間となる。

8. おわりに

本稿では、軍艦島モニタリングにおけるデータ中心型タスクスケジューリング方式として、BAAD schedulingを提案した。BAAD schedulingでは、DC-LQによってスリープ時間の変動を抑えるようにセンサノードの駆動時間とスリープ時間を決定する。また、BLT allocationによってバッテリー残量に応じて実行するタスクを決定する。シミュレーションによる性能評価から、BAAD schedulingが既存方式と比較してデータ取得量の減少を抑えつつ、最悪データロス時間の短縮を実現できていることが分かった。

謝辞

本研究は科学研究費補助金(26289194, 代表: 濱本卓司)の助成を受けたものである。本研究の遂行をサポートして下さった長崎市世界遺産推進室、日本航空電子の富岡昭浩氏に感謝致します。

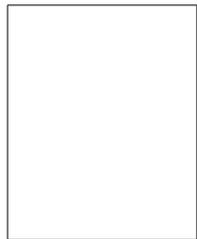
参考文献

[1] Natke, H. G., Tomlinson, G. R. and Yao, J. T. P.: *Safety Evaluation Based on Identification Approaches*, Vieweg (1993).
 [2] Natke, H. G. and Cempel, C.: *Model-Aided Diagnosis of Mechanical Systems*, Springer (1997).
 [3] Haldar, A.: *Health Assessment of Engineered Structures*, World Scientific (2013).
 [4] 濱本卓司: 建築物の耐震性能評価のためのモニタリング技術, 計測自動制御学会計測と制御, Vol. 46, No. 8, pp.

605-611 (2007).
 [5] 濱本卓司: 建築物の構造ヘルスマニタリング, 基礎工 特集 基礎工におけるモニタリングとその活用, Vol. 43, No. 11, pp. 17-20 (2015).
 [6] 金裕錫, 壁谷澤寿海, 松森泰造, 壁谷澤寿一: E - ディフェンスによる実大6層鉄筋コンクリート耐震壁フレーム構造の破壊過程究明に関する解析的研究, 日本建築学会構造系論文集, Vol. 74, No. 641, pp. 1327-1334 (2009).
 [7] Battleship Island monitoring, <http://sarulab.inf.shizuoka.ac.jp/battleship/>.
 [8] Vigorito, C. M., Ganesan, D. and Barto, A. G.: Adaptive Control of Duty Cycling in Energy-Harvesting Wireless Sensor Networks, *Proceedings of the 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON'07)*, pp. 21-30 (2007).
 [9] Xu, N., Rangwala, S., Chintalapudi, K. K., Ganesan, D., Broad, A., Govindan, R. and Estrin, D.: A Wireless Sensor Network for Structural Monitoring, *Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys'04)*, pp. 13-24 (2004).
 [10] 長井望, 三田彰, 矢向高弘, 佐藤忠信: 構造ヘルスマニタリング用ワイヤレスセンサに関する研究, 日本地震工学会論文集, Vol. 3, No. 4, pp. 1-13 (2003).
 [11] Kurata, N., Suzuki, M., Saruwatari, S. and Morikawa, H.: Actual Application of Ubiquitous Structural Monitoring System using Wireless Sensor Networks, *Proceedings of the 14th World Conference on Earthquake Engineering (14WCEE)*, pp. 1-9 (2008).
 [12] NTT データ: Xrosscloud 橋梁監視ソリューション (BRIMOS), <http://www.nttdata.com/jp/ja/lineup/brimos/>.
 [13] Zhang, P., Sadler, C. M., Lyon, S. A. and Martonosi, M.: Hardware Design Experiences in ZebraNet, *Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys'04)*, pp. 227-238 (2004).
 [14] Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R. and Anderson, J.: Wireless Sensor Networks for Habitat Monitoring, *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, pp. 88-97 (2002).
 [15] Szewczyk, R., Osterweil, E., Polastre, J., Hamilton, M., Mainwaring, A. and Estrin, D.: Habitat Monitoring with Sensor Networks, *Communications of the ACM*, Vol. 47, No. 6, pp. 34-40 (2004).
 [16] Buchli, B., Sutton, F., Beutel, J. and Thiele, L.: Dynamic Power Management for Long-Term Energy Neutral Operation of Solar Energy Harvesting Systems, *Proceedings of the 12th ACM Conference on Embedded Networked Sensor Systems (SenSys'14)*, pp. 31-45 (2014).
 [17] Alippi, C. and Galperti, C.: An Adaptive System for Optimal Solar Energy Harvesting in Wireless Sensor Network Nodes, *IEEE Transactions on Circuits and Systems*, Vol. 55, No. 6, pp. 1742-1750 (2008).
 [18] Kansal, A., Hsu, J., Zahedi, S. and Srivastava, M. B.: Power Management in Energy Harvesting Sensor Networks, *ACM Transactions on Embedded Computing Systems (TECS)*, Vol. 6, No. 4 (2007).
 [19] Polastre, J., Hill, J. and Culler, D.: Versatile Low Power Media Access for Wireless Sensor Networks, *Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys'04)*, pp. 95-107 (2004).
 [20] Dutta, P., Haggerty, S. D., Chen, Y., Liang, C. J. M. and Terzis, A.: Design and Evaluation of a Versatile and Effi-

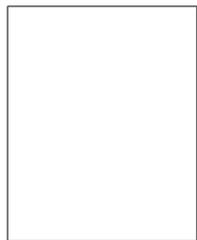
cient Receiver-initiated Link Layer for Low-power Wireless, *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys'10)*, pp. 1-14 (2010).

- [21] Buettner, M., Yee, G. V., Anderson, E. and Han, R.: X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks, *Proceedings of the 4th ACM Conference on Embedded Networked Sensor Systems (SenSys'06)*, pp. 307-320 (2006).
- [22] Dunkels, A.: The ContikiMAC Radio Duty Cycling Protocol, Technical Report T2011:13, Swedish Institute of Computer Science (SICS) (2011).
- [23] 濱本卓司, 倉田成人, 猿渡俊介, 富岡昭浩: 軍艦島モニタリングプロジェクト その1: 研究計画と予備計測/長期計測, 社団法人日本建築学会学術講演梗概集 (2015).
- [24] 富岡昭浩, 濱本卓司, 倉田成人, 猿渡俊介: 軍艦島モニタリングプロジェクト その2: 長期振動計測システム, 社団法人日本建築学会学術講演梗概集 (2016).
- [25] 関根明日香, 濱本卓司, 富岡昭浩, 倉田成人, 猿渡俊介: 軍艦島モニタリングプロジェクト その3: 長期モニタリングに基づく軍艦島70号棟の動的挙動に関する考察, 社団法人日本建築学会学術講演梗概集 (2016).
- [26] 倉田成人, 濱本卓司, 猿渡俊介, 富岡昭浩: 軍艦島モニタリングプロジェクト その4: 日本最古の鉄筋コンクリート集合住宅30号棟の画像モニタリング, 社団法人日本建築学会学術講演梗概集 (2016).



黒木 琴海 (学生会員)

2016年静岡大学情報学部情報科学科卒業。現在、静岡大学大学院総合科学技術研究科情報学専攻に在籍。センサネットワークに関する研究に従事。2015年情報処理学会 MBL 研究会優秀論文賞。情報処理学会学生会員。



小寺 志保 (正会員)

2014年静岡大学情報学部情報科学科卒業。2016年静岡大学大学院情報学研究科情報学専攻修士課程修了。同年東芝研究開発センター入社。映像伝送, 計算機ネットワークに関する研究に従事。2013年情報処理学会 MBL 研究会優秀論文賞。2014年情報処理学会山下記念研究賞。静岡大学学長賞。情報処理学会正会員。



倉田 成人

1986年東京大学大学院工学系研究科建築学専門課程(地震研究所)修士課程修了。博士(工学)。1986~2010年鹿島建設・小堀研究室。2010~2014年同・技術研究所。2014年より筑波技術大学産業技術学部産業情報学科教授。建築情報学, 構造工学, 地震工学の研究に従事。日本建築学会, 計測自動制御学会, IEEE, 電子情報通信学会各会員。



濱本 卓司

1981年早稲田大学大学院理工学研究科博士課程修了。博士(工学)。1982~1990年大林組。1986~1988年イリノイ大学客員研究員。1990年武蔵工業大学工学部建築学科助教授。1996年武蔵工業大学工学部建築学科教授。2009年より東京都市大学工学部建築学科教授。構造工学, 地震工学, 維持管理工学, 建築構造・材料, 船舶海洋工学の研究に従事。1999年日本建築学会賞(論文)受賞。



猿渡 俊介 (正会員)

2007年東京大学大学院博士課程修了。博士(科学)。2003~2004年IPA未踏ソフトウェア創造事業。2006~2008年日本学術振興会学振特別研究員。2007~2008年イリノイ大学客員研究員。2008~2012年東京大学先端科学技術研究センター助教。2012~2015年静岡大学大学院情報学研究科助教。2013年より理化学研究所と産業技術総合研究所にて客員研究員。2015~2016年静岡大学情報学部講師。2016年より大阪大学大学院情報科学研究科准教授。専門はワイヤレスネットワーク, センサネットワーク, システムソフトウェア等。2009年電子情報通信学会論文賞。2010年情報処理学会山下記念研究賞。電子情報通信学会, IEEE, ACM 各会員。