

軍艦島モニタリングに向けたデータ中心型タスクスケジューリング方式

黒木 琴海[†] 佐藤 匠[†] 小寺 志保[†] 倉田 成人[‡] 濱本 卓司^{‡†} 猿渡 俊介[†]
[†]静岡大学 [‡]筑波技術大学 ^{‡†}東京都市大学

1 はじめに

建築構造物の劣化診断などの技術は発展しているものの、阪神淡路大震災で多発したような建築構造物の崩壊現象は現象自体が稀であるがゆえに知見がまだ少ないのが現状である。これに対して、筆者らは、今まさに崩壊しつつある軍艦島という特殊な状況で崩壊現象のビッグデータを取得する軍艦島モニタリングに取り組んでいる。本稿では、軍艦島モニタリングにおけるセンシングシステムのデータ中心型タスクスケジューリング方式として、ConCon scheduling (Conservative Consumption scheduling) を提案する。ConCon scheduling によれば、単位時間当たりには得られるエネルギー量とその時点で蓄積されているエネルギー量から動的にスリープ時間を調整することで、データ取得量の最大化とデータロス時間の最小化を同時に実現することができる。

2 軍艦島モニタリング

軍艦島は、現在も経年劣化による建築群の複雑な崩壊が発生している世界的にも珍しい環境である。筆者らは、軍艦島において崩壊中の建築群の映像、音声、加速度を取得することで、建築構造解析に貢献することを目指して軍艦島モニタリングプロジェクト [1] を進めている。

軍艦島は電源もネットワークも存在しないため、太陽光発電を用いてセンシングシステムを駆動して無線を用いて本土へとデータを送信する。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。電力を得るためのソーラーパネルとして、Coleman PVS-55 W を用いる。Coleman PVS-55 W の公称最大出力は 55 W である。Coleman PVS-55 W は、400 Wh のバッテリーも具備している。ノート PC には TOUGHBOOK CF-195W1ACS を利用する。CF-195W1ACS にはセンサと無線通信用のデバイスが USB で接続されている。CF-195W1ACS の電力消費量を計測した結果、最大約 100 [W]、最小約 0 [W] であった。すなわち、CF-195W1ACS が常に最大の電力を消費していたとしても、Coleman PVS-55 W のバッテリーがフル充電された状態であれば最低でも 4 時間駆動できる。また、CF-195W1ACS は、サスペンド状態で約 1.2 [W]、映像の取得と本土への無線通信で約 14.1 [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では、ノート PC において太陽光発電で得られた電力を利用してセンサデータを取得するため、限られた電力でどのようにセンシングを実現するかが課題である。具体的には、次の 2 つの要件を同時に満たす必要がある。

1 つ目の要件は、センサから取得するデータ量を最大化することである。取得するデータ量が少ない場合、時間経過によるデータの変化を発見できない可能性がある。例えば、取得できたデータがセンシング開始から 10 分後と 20 分後の 2 つのみであるとすると、15 分後に建物が崩壊するなどによってデータが大きく変化した場合、変化を知ることができない。取得するデータ量が多ければ多いほど、データの変化を発見できる可能性は高くなる。

2 つ目の要件は、最大データロス時間を最小化することである。最大データロス時間とは、センサによってデータを取得していない時間の最大値である。最大データロス時間が大きい場合、取得するデータの時間帯に偏りが生じる。例として、同じデータ量において、最大データロス時間が 1 時間のデータと最大データロス時間が 12 時間のデータを考える。最大データロス時間が 1 時間のデータでは、最悪の場合でも、1 日のうち何時ごろ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、最大データロス時間が 12 時間のデータでは、最悪の場合では、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。

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。赤で示した部分でデータを取得している。蓄積エネルギーがある場合はデータを取得し続けるため、取得できるデータ量は最大となる。しかしながら、蓄積している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ないため、取得できるデータの時間帯に偏りが生じる。図 2 の例では、18 時から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。

取得するデータの時間帯の偏りを減らす方法として、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。赤で示した部分でデータを取得している。一定間隔でデータを取得するため、取得したデータに時間帯の偏りは少なくなる。しかしながら、赤で示した部分以外はデータを取得しないため、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増えない。

4 提案方式: ConCon scheduling

3 節での議論を基に、センサから取得するデータ量の最大化と、最大データロス時間の最小化を両立するタスクスケジューリング方式である「ConCon scheduling」を設計した。ConCon scheduling では、太陽光発電装置が具備しているバッテリーの残量と発電量からスリープ間隔を動的に変化させる。

既存の太陽光発電を用いたセンサネットワークシステムでは、動物の生息環境モニタリングや高山での気候変化モニタリングなど、時間変化が比較的早い現象を対象として太陽光発電のエネルギー利用効率を最適化することを主眼に研究が行われていた [2, 3]。それに対して ConCon scheduling では、軍艦島という特殊な環境における構造物の劣化という時間変化が緩やかなモニタリングにおいて、データ取得量を最大化しつつ最大データロス時間を最小化するデータセントリックなタスクスケジューリングを実現する。

まず、システム全体の消費電力から単位時間あたりに必要なエネルギー量を算出する。単位時間 $T_{interval}$ [sec] あたりに必要なエネルギー量 E_{sense} [J] を式 (1) に示す。

$$E_{sense} = (P_{sleep}T_{sleep} + P_{sense}T_{sense}) \frac{T_{interval}}{(T_{sleep} + T_{sense})} \quad (1)$$

P_{sleep} [W] はスリープ時の消費電力、 T_{sleep} [sec] はスリープ間隔、 P_{sense} [W] はセンシング時の消費電力、 T_{sense} [sec] は 1 回のセンシングにかかる時間を意味している。次に、バッテリーのエネルギー

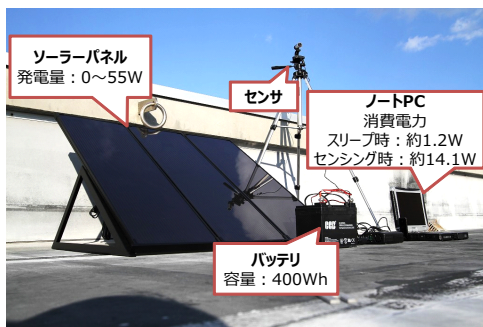


図 1: 軍艦島モニタリングシステムのプロトタイプ

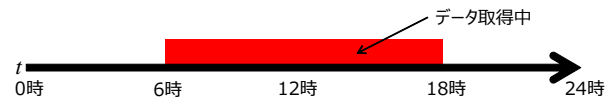


図 2: 蓄積エネルギーがあれば直ちにデータを取得 (ASAP)

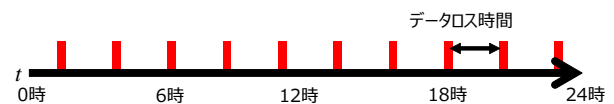


図 3: 一定間隔でデータを取得

Algorithm 1 ConCon scheduling

```

1:  $E_{sense} \leftarrow (P_{sleep}T_{sleep} + P_{sense}T_{sense}) \frac{T_{interval}}{(T_{sleep} + T_{sense})}$ 
2: loop
3:   Get sensor data
4:    $E_{budget} \leftarrow (E_{store} + P_{solar}T_{interval})\alpha$ 
5:    $T_{sleep} \leftarrow \arg \min_{T_{sleep}} (E_{budget} - E_{sense})$ 
6:   Sleep for  $T_{sleep}$  seconds
7: end loop
    
```

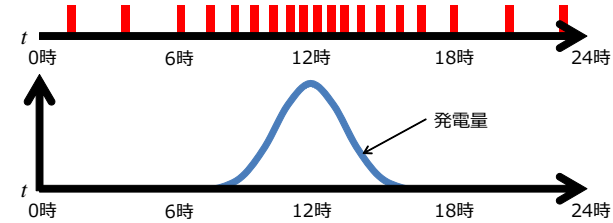


図 4: スリープ間隔と実際の発電量

残量と現在の発電量から、単位時間あたりにシステムで消費できるエネルギー量を決定する。単位時間 $T_{interval}$ [sec] あたりにシステムが消費できるエネルギー量 E_{budget} [J] を式 (2) に示す。

$$E_{budget} = (E_{store} + P_{solar}T_{interval})\alpha \quad (2)$$

E_{store} [J] はバッテリーの充電残量、 P_{solar} [W] はソーラパネルの発電量を意味している。 α は事前のシミュレーションで取得した消費エネルギー量の割合を意味する変数であり、設置環境に応じて調節する。式 (1) の E_{sense} と式 (2) の E_{budget} が最も近似するときの間隔をスリープ間隔とする。ConCon scheduling では、センシングが終了する度に新たなスリープ間隔を決定する。

Algorithm 1 に、ConCon scheduling のスリープ間隔決定アルゴリズムを示す。各変数の意味は前述したとおりである。ConCon scheduling のアルゴリズムは、エネルギー量の予測とスリープ間隔の決定の 2 種類の処理から構成される。

1 つ目の処理であるエネルギー量の予測では、スリープ間隔を変化させ、システムが単位時間あたりに消費するエネルギー量を算出する。具体的には Algorithm 1 の 1 行目で、式 (1) に基づいてエネルギー量 E_{sense} を算出する。今回のプロトタイプでは、0 から 3600 まで T_{sleep} を 1 秒ずつ変化させた場合のそれぞれのスリープ間隔において必要なエネルギー量のテーブルを作成している。1 秒ずつ変化させている理由は、プロトタイプにおいて Linux の `rtctime` を利用してスリープの処理を実装していることに起因する。センシングに必要な電力 P_{sense} とスリープ時に消費する電力 P_{sleep} は 2 節に示したように、プロトタイプの計測データを用いる。例えば、今回のプロトタイプではセンシング時に約 14.1 [W]、スリープ時に約 1.2 [W] を消費する。

2 つ目の処理であるスリープ間隔の決定では、式 (2) に基づいてスリープ間隔を決定する。まず、Algorithm 1 の 3 行目でセンサーデータを取得した後、4 行目において、ノート PC の充電残量と発電量から、式 (2) を用いて単位時間あたりにシステムが消費できるエネルギー量を算出する。次に、5 行目で、式 (1) より算出した単位時間あたりに必要なエネルギー量と、式 (2) より算出した単位時間あたりのシステム消費エネルギー量を比較し、エネルギー量の差が最も小さくなる場合のスリープ間隔 T_{sleep} を算出する。Algorithm 1 の 6 行目でスリープ間隔に従ってシステムをスリープさせる。

図 4 に、ConCon scheduling を用いてスケジューリングした場合のスリープ間隔と太陽光発電による発電量の関係を示す。図 4 上がスリープ間隔、図 4 下が太陽光発電の発電量である。太陽光発電による発電量が多い場合はバッテリーの充電残量も多いため、単位時間あたりのシステム消費エネルギー量は大きくなり、スリープ間隔は短くスケジューリングされる。一方で、太陽光発電による発電量が少ない場合はバッテリーの充電残量も少ないため、単位時間あたりのシステム消費エネルギー量は小さくなり、スリープ間隔は長くスケジューリングされる。

5 性能評価

提案手法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り取得データ量と最大データロス時間を評価した。提案手法の性能を相対的に評価するために、ASAP (As Soon As Possible)、Fix10、ConCon scheduling の 3 つのアプローチを比較した。ASAP はスケジューリングを行わず、バッテリーの残量がある限りセンシングし続ける手法であり、データ取得量をもっとも大きくなることが予想される。Fix10 はセンシングする度に固定値である 10 秒間スリープする手法であり、最大データロス時間の性能を測る尺度となる。ConCon scheduling は 4 節で述べた提案手法である。シミュレーションでは、太陽光発電による発電量の 1 日の推移が午後 12 時を頂点とする正規分布に従うと仮定し、午後 12 時の時点での発電量を 0 [W] から発電上限値 (maximum generated power) までの範囲で天候の変化を模してランダムに決定した。発電上限値は季節による太陽光の入射角の変化を模している。

図 5 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 [MB]、横軸は発電上限値 [W] である。提案手法 (ConCon) が ASAP に漸近する高いデータ取得量を実現していることが分かる。一方で Fix10 のデータ取得量は低い。

図 6 に発電上限値に対する最大データロス時間を示す。縦軸がデータ取得量 [MB]、横軸は発電上限値 [W] である。図 6 より、提案手法は Fix10 に漸近する性能を実現していることが分かる。特に発電上限値が低い領域においては、最大データロス時間は Fix10 よりも提案手法の方が小さい。一方で、ASAP の最大データロス時間は常に最悪である。

6 おわりに

本稿では、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として、ConCon scheduling を提案した。現在、軍艦島に対してセンシングシステムを展開するために、通信環境の調査や、設置場所の検討を進めている (図 7)。

参考文献

- [1] 軍艦島モニタリングプロジェクト, <http://sarulab.inf.shizuoka.ac.jp/battleship/>.
- [2] Zhang, P., Sadler, C. M., Lyon, S. A. and Martonosi, M.: Hardware Design Experiences in ZebraNet, *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (ACM SenSys'04)*, pp. 227–238 (2004).
- [3] 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 (ACM SenSys'14)*, pp. 31–45 (2014).

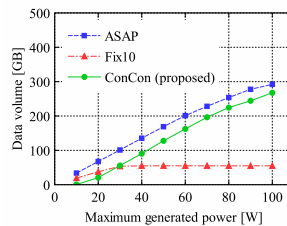


図 5: データ取得量

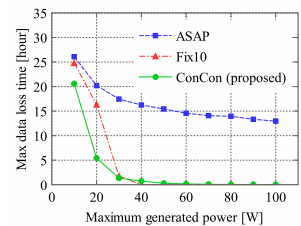


図 6: 最大データロス時間

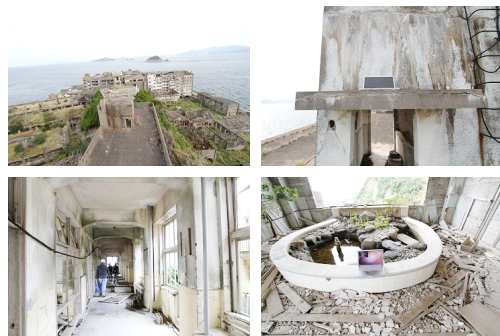


図 7: 軍艦島における調査