

循環行列を用いたセンサノード上への圧縮センシングの実装と消費電力の評価

佐々木 達哉¹ 川原 圭博^{1,2} 浅見 徹¹

概要：本研究では、メモリ容量が極めて制限されたセンサノード上でセンシングデータを圧縮することを目標に、循環行列を用いた圧縮センシング手法を提案する。無線センサネットワークにおいて、センサノードに最も大きな負荷がかかるのは通信時であり、センシングデータを圧縮して通信時のデータを削減することで、センサノードを長寿命化することが可能である。しかしながら、圧縮センシングはその原理上、観測行列のサイズが大きくなりがちであり、単一のセンサノード上に実装する際に、しばしばセンサノードのメモリ容量が不足するといった問題が発生する。このため、観測行列に循環行列を用いることにより、圧縮センシングの実装に際して要求されるメモリ量を $N \times M$ から $N + M$ にまで低減できることを示す。また、無線センサモジュールである eZ430-RF2500 に圧縮センシングを実装し、圧縮計算時及び通信時の各動作時におけるセンサモジュールの消費エネルギーを評価した。評価の結果、観測対象の信号サイズ N 、圧縮後の信号サイズ M に対して、 $M \leq N / (1 + 2.82 \times 10^{-3} \times N)$ を満たすならば、提案手法による消費エネルギーの削減効果が見込めることがわかった。

1. はじめに

無線センサネットワーク (WSN) は空間中に多数のセンサノードを配置し、センシングしたデータをノード間で無線通信し、データを処理する Sink ノード、ベースステーションへと収集する技術である。データを無線で通信することにより、計測機器を配線する手間を削減し、計測機器を配置する自由度を大幅に向上させることができる。また、配線による制限がなく、かつ遠隔地点のデータを自動的に収集することができるため、森林地帯や砂漠などの計測機器を持ち込むことや、人力での計測が困難な地域の環境情報をモニタリングすることができる。このような利点に注目して、WSN は環境モニタリングや防災、農業、軍事などの幅広い分野への応用が期待され、様々なアプリケーションが研究されてきた。

WSN の実現にあたり、最も大きな課題の一つとしてセンサノードの電源確保が挙げられる。WSN では遠隔地に広範囲にセンサノードを配置するという性質上、センサノードはバッテリー駆動であることを前提としている。このため、バッテリーの寿命はノード及びネットワークのリンク

寿命と直結しており、ノードがバッテリーを使い果たした際にはバッテリーの交換を行うか、新たにノードを配置しなければならず、管理コストやノードの購入コストが大きくなってしまふ。そのため、WSN をできる限り低コストで運用するためには、バッテリーの負荷を低減しセンサノードを長寿命化させることが不可欠である。特に、センサノードの動作のうち最もバッテリーを最も大きく消費するのは無線通信時であり、Estrin らの研究ではセンサノードが通信時に消費する電力は、センシングや CPU によるセンシングデータの処理といった動作に比べて約 10 倍以上も大きいという結果が報告されている [1]。また、Handy らによって 1,000 bit のデータを 100 m 離れた地点に送信するのにかかるエネルギーは CPU で 3,000,000 個分の命令を実行する際に消費されるエネルギーに匹敵することが明らかになっている [2]。このため、センサノードの省電力化のためにはセンシングデータを圧縮することで、通信時の消費エネルギーを削減することが効果的である。センシングデータを圧縮する研究はこれまで数多く提案されており、センシングデータの結合エントロピーやウェーブレット変換を用いてデータを圧縮する方法などがある [3], [4]。しかし、圧縮計算が複雑であるため圧縮にかえて大きなエネルギーが必要になってしまうという問題があった。

そこで近年、センサノードでのセンシングデータの圧縮に圧縮センシングの応用が注目されている [5]。圧縮センシ

¹ 東京大学 大学院情報理工学系研究科
Graduate School of Information Science and Technology,
The University of Tokyo

² School of Electrical and Computer Engineering, Georgia Institute of Technology

ングは信号のスパース性に注目することで、極めて少数のサンプリングデータから元信号を復元する技術である。圧縮センシングは圧縮計算が行列積のみで行われるため圧縮計算が容易であり、計算にかかるエネルギーが小さくて済むということが期待される。しかしながら、単一のセンサノード上で圧縮センシングを行う場合、 $N \times M$ のサイズを持つ観測行列をセンサノードに搭載されたメモリ上に展開しなければならず、メモリの容量が不足してしまうといった問題が発生する。一般的なセンサノードに搭載されたメモリ量は数 kB 程度であるため、観測対象の信号サイズが 100 程度の場合には圧縮率が 10% でも観測行列に 1 kB を必要とするため、 $N \times M$ の観測行列をセンサノード上へ実装するのは、メモリ容量の制限上困難である。そこで、本稿では観測行列に循環行列を用い、また循環行列を用いることによる行列のランダム性及び復元効率の低下を乱数配列によって補償することによって、観測行列のサイズを $N \times M$ から $N + M$ に低減し、センサノード上へ実装可能な圧縮センシングの手法を提案する。また、実際の無線センサモジュールである eZ430-RF2500 に提案手法による圧縮センシングを実装し、圧縮計算と通信時にノードで消費されるエネルギーを計測し、観測対象の信号サイズ N 、圧縮後の信号サイズ M に対して、 $M \leq N / (1 + 2.82 \times 10^{-3} \times N)$ を満たすならば、提案手法による消費エネルギーの削減効果があることを示す。

本稿の構成は以下のものである。第 2 では圧縮センシングの原理について紹介し、実際のセンサノードへの実装におけるメモリ使用量の問題について述べる。第 3 節では WSN への圧縮センシングの応用事例について紹介し、第 4 節で循環行列を用いたメモリ使用量の低減手法について述べる。第 5 節で提案手法を実際にセンサノードへ実装した際の、圧縮計算と通信時の消費エネルギーの計測と評価について述べ、第 6 節で本稿についてまとめる。

2. 圧縮センシング

本節では、圧縮センシングの数学的な原理について簡単に紹介し、センサノードへの圧縮センシングの実装における問題点である、メモリ使用量について指摘する。

2.1 スパース性と信号の圧縮

従来の信号圧縮では、一度すべてのデータをサンプリングした後に、データの冗長性に注目しながら符号化を行うなどの処理を施すことで圧縮が行われ、圧縮の過程でサンプリングされたデータの大多数は破棄されてしまっていた。圧縮センシングは Donoho や Candès によって提唱された、従来の枠組みを大きく変える信号の圧縮・復元技術である [6], [7]。

観測対象が N 次元のスパース性を持つ信号 x であるとき、信号 x のスパースな表現を N 次元ベクトル s で表す

と、元信号 x は基底変換の $N \times N$ 行列 Ψ を用いて、式 (1) のように表すことができる。

$$x = \Psi s \quad (1)$$

この時 s の要素は、大部分が 0 となっている。このように、信号の要素の中で非ゼロ成分が占める割合が小さい信号をスパースな信号と呼び、信号 x はスパース性を持つ信号となる。 s の非ゼロ成分の数を K とすると、 x は基底 Ψ の下で K スパースな信号である。また実在の信号では、信号が完全にスパースとなることはほぼないが、多くの場合、適切な基底のもとで信号の大部分の要素が 0 に近い値となり、信号の一部にのみ情報が集中する。このような信号を圧縮可能性と呼び、スパース性と類似した性質を持っている。本稿では、信号が完全にスパースな場合、圧縮可能性を持つ場合の両方を、スパース性を持つ信号として表現する。

圧縮センシングでは、観測の過程でランダムプロジェクションまたは、ランダムサンプリングによる信号の圧縮が行われる。これらの圧縮は N 次元ユークリッド空間から M 次元ユークリッド空間へのランダムな写像であるので、圧縮過程は $M \times N$ 行列 Φ で表すことができる。この時、観測されるデータは M 次元ベクトル d によって式 (2) のように表され、 $M \ll N$ が成り立つならば、観測されたデータの次元は元信号の次元よりも極めて小さいため、情報は圧縮されることになる。

$$d = \Phi x = \Phi \Psi s \quad (2)$$

2.2 復元手法

式 (2) において、式 (2) において、 Φ, Ψ, d は既知であるので、 s を一意に推定することができれば元信号 x を復元することができるが、式 (2) は既知数に対して未知数の方が多い不良設定問題であり、通常は一意な解を求めることができない。しかし、信号 x がスパース性をもつことを仮定すれば、式 (3) のように式 (2) を満たす無数のベクトル s の中から、 l_0 ノルムが最小となり s のスパース性が最も高くなるような \hat{s} を推定することで元信号 x を復元することができる。ここで、ベクトル s の l_n ノルムを $|s|_n$ で表すことにする。 l_0 ノルムは信号 s の非ゼロ成分の数を示す。

$$\text{minimize } |\hat{s}|_0 \quad \text{subject to } d = \Phi \Psi \hat{s} \quad (3)$$

しかし、 l_0 ノルム最小化は一般的に NP 困難な問題であることが知られている [8]。そこで、式 (4) のように、 l_0 ノルムの代わりに、各次元の絶対和である l_1 ノルムを最小化することで、信号 s の推定が行われる [5], [8]。

$$\text{minimize } |\hat{s}|_1 \quad \text{subject to } d = \Phi \Psi \hat{s} \quad (4)$$

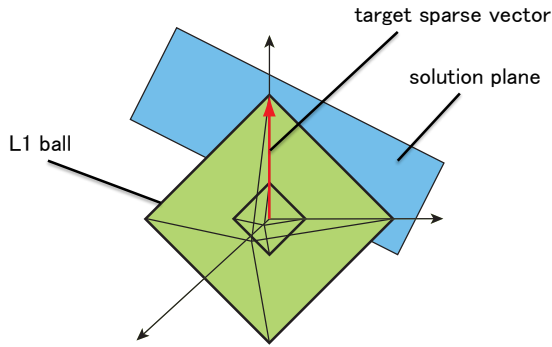


図 1 l_1 ノルム最小化によるスパース信号の復元

元信号 x を再現するために必要な観測数 M が満たすべき条件は、式 (5) で与えられる [5] .

$$M \geq c \cdot \mu^2(\Phi, \Psi) \cdot K \cdot \log N \quad (5)$$

ここで、 c は正の定数であり、 $\mu(\Phi, \Psi)$ は Ψ の第 j 列ベクトルを ψ_j 、 Φ の第 i 行ベクトルを ϕ_i として、式 (6) のように定義される [5] . 式 (6) の $|\langle \phi_i, \psi_j \rangle|$ は ϕ_i と ψ_j の内積である .

$$\mu(\Phi, \Psi) = \sqrt{N} \max_{1 \leq i, j \leq N} |\langle \phi_i, \psi_j \rangle| \in [1, \sqrt{N}] \quad (6)$$

ここで、 l_1 ノルム最小化による復元が可能である理由を、視覚的に説明する . 図 1 は 3 次元の場合におけるスパース信号と、 l_1 ノルム最小化による信号の復元の様子を表している . 図における赤線を 3 次元の元信号として、2 次元の観測データから元信号を復元する場合を考える . このとき、元信号は z 成分のみが非ゼロ要素となる 1 スパースな信号であり、観測数 $M = 2$ 、元信号の次元数 $N = 3$ 、信号のスパース度合い $K = 1$ となる . 2 次元の観測信号から 3 次元の元信号を復元すると、不良設定問題であるため無数の解が出現し、その解の存在範囲は図 1 における青い平面で表すことができる . ここで、 l_1 ノルムは原点からのマンハッタン距離であるため、原点からの l_1 ノルムが等しい点の集合は図 1 における、緑色の正八面体で表される . l_1 ノルム最小化による復元は、この正八面体が初めて解の存在範囲である平面に接触した点が、復元結果として得られるベクトルとなる . このように、 l_1 ノルム最小化による復元では、信号のスパース度合いが最も大きくなる方向 (図 1 では x, y, z 軸方向) への復元が、真っ先に行われる . このため、信号にスパース性がある場合には、少ない観測データからでも高い確率で元の信号が復元されることになる .

2.3 観測行列の選択

式 (6) において $\mu(\Phi, \Psi) = 1$ となる時、行列 Φ, Ψ は互いにインコヒーレントであるといい、復元に必要な観測数を最も小さくすることができる . このような観測行列の選び方として、Candès はガウス乱数に基づく行列、各要

表 1 代表的な無線センサノードにおける MCU 性能

Sensor Node		MICAz	IRIS	TelosB	eZ430-RF2500
Micro Controller	chip	ATMega 128L	ATMega 128	MSP430 F1611	MSP430 F2274
	TYPE	7.37MHz 8bit	7.37MHz 8bit	8MHz 16bit	16MHz 16bit
	SRAM	4kB	8kB	10kB	1kB

素が ± 1 となるようなランダム行列、フーリエ基底行列の行をランダムに抜き出すような行列などを挙げている [5] .

さらに、式 (4) による解が式 (3) による解と一致することを保証する、制限等長性 (Restricted Isometry Property, RIP) という性質がある . 制限等長性とは、任意の K スパースな信号 v に対して式 (7) を満たす定数 $\delta_K \in (0, 1)$ が存在するというものである .

$$1 - \delta_K \leq \frac{|\Phi \Psi v|_2^2}{|v|_2^2} \leq 1 + \delta_K \quad (7)$$

Candès は $\delta_{2K} < \sqrt{2} - 1$ であるとき、任意の K スパースな信号に対して、式 (4) の結果が正しい解を与えることを示しており [9]、先に挙げた、ガウス乱数に基づく行列、各要素が ± 1 であるランダム行列、フーリエ基底行列の行をランダムに抜き出す 3 つの観測行列はこの条件を高確率で満たすことがそれぞれ証明されている [10], [11] .

2.4 センサノードのメモリ制限

これまでに紹介した性質上、圧縮センシングは無線センサネットワークへの応用が期待されており、センサノード上でのデータ圧縮手法が多数報告されている . しかし、単一のセンサノード上で圧縮センシングによりデータを圧縮する際には、観測行列をノード上に実装しなければならず、センサノードのメモリ容量の制限が問題となる . 観測行列のサイズは $N \times M$ であるため、例えば $N = 128$ の信号を観測する場合、 $M = 8, 32, 64$ でもそれぞれ 1kB, 4kB, 8kB のメモリ容量が必要となる . ここで、表 1 に代表的な無線センサノードが搭載している Micro Controller Unit(MCU) とメモリ量を示す . 表 1 からわかるように、一般的なセンサノードを用いる場合、搭載されているメモリ容量は 1~10kB 程度であるため、 $N \times M$ ものサイズを持つ観測行列をセンサノードのメモリ上に実装することは困難であり、単一のセンサノード上で圧縮センシングを行うためには、観測行列のサイズを低減することが不可欠である .

3. WSN への圧縮センシングの応用に関する関連研究

本節では圧縮センシングを WSN に応用した関連研究について紹介する .

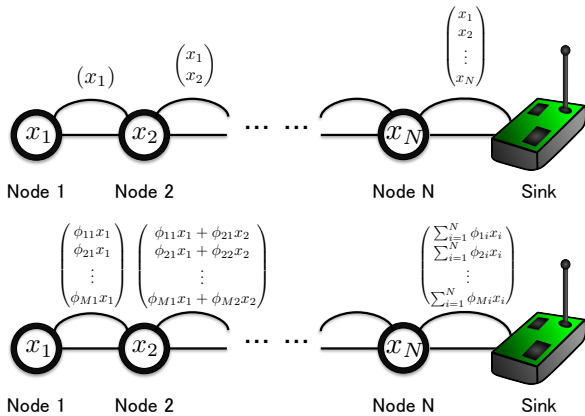


図 2 圧縮センシングを利用したマルチホップ WSN の負荷分散

3.1 シングルホップの WSN への応用例

シングルホップの WSN への圧縮センシングの応用例として, Bajwa らによって提唱された Compressive Wireless Sensing がある [12]. これは空間中に多数のセンサノードを配置し, シングルホップでデータを Fusion Center へ送信する際に圧縮センシングを用いて送信データ量の削減を行っている. このとき FC で復元されたデータと元のセンシングデータとの間に生じる誤差 D と, ノードの消費電力 P の関係をオーダ単位で比較し, シミュレーションにより圧縮センシングを用いることの有効性を述べている.

また, Mahmudimanesh らは同様の環境において, センサノードの ID を並び替えることによりセンシングデータの順序が変更され, データのスパース性が上昇するという事を利用して, センシングデータに準最適なスパース性を与えられるようなノード ID の並び替えの手法を提案している [13].

3.2 マルチホップの WSN への応用例

マルチホップの WSN に対して圧縮センシングを応用した例として, Chong らの Compressive Data Gathering が挙げられる [14]. マルチホップでデータを収集する際には, 図 2 の上図のようにセンサノードはパケツリレーの方式で直前のノードから送られてきたデータに加えて, 自らのデータも送信することになるため, Sink 近くのノードほど送信しなければならないデータ量が, Sink から末端のノードまでのホップ数 N に対して $O(N^2)$ で増加していく. そこで, Chong ら図 2 の下図のように, 各ノードで得られたデータ x_j に対して M 個の乱数をかけ合わせ, 前のノードから送られてきたデータとの加算を行うことで, 全てのノードが M 個のデータを送るようにしている [14]. このとき, 各ノードがセンシングするデータにスパース性があれば, $M \ll N$ の場合でも圧縮センシングによりデータの復元が可能であり, ネットワーク全体でのノードの負荷分散と, データの圧縮が実現できる.

また, Nguyen らはアドホックネットワークのようにルー

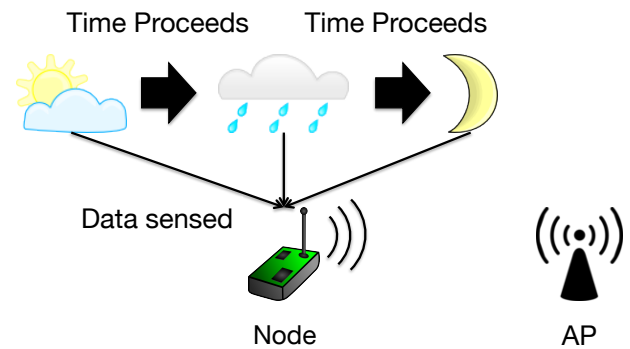


図 3 本研究での WSN の想定環境

ティングが変化するネットワークにおいて, ネットワークコーディングと圧縮センシングを組み合わせることにより, 動的なルーティング変化に対応できる圧縮センシングの手法, NETCOMPRESS を提案している [15].

3.3 本研究での想定環境との相違点

関連研究での WSN への圧縮センシングの応用は全て, 多数のノードが隣接している状態において, 隣接ノード間のセンシングデータに存在するスパース性を利用することで圧縮センシングを応用し, 各ノードまたはネットワーク全体としての通信データの圧縮を行うものである. この場合, データ圧縮は隣接ノードとの間で行われるため, 各ノードは観測行列の中で自分が使う一部のみを保持しておけばよい. しかし, 図 3 のように時々刻々と変化する気象情報を単一のデータでセンシングし, AP まで送信するという状況に圧縮センシングを応用する場合には, $N \times M$ の観測行列を全て一つのノード上に実装しなければ, データの圧縮を行うことができない. 本稿ではこのような WSN の利用形態を想定して, 圧縮センシングを単一のセンサ上で行うことを目標としている.

4. 循環行列を用いた観測行列サイズの低減

第 2.4 節で述べたように, 単一のセンサノードに圧縮センシングを実装するためには, 観測行列のサイズを低減しなければならない. 本節では, 循環行列を用いることで観測行列のサイズを $N \times M$ から一般的なセンサノードに実装可能な $N + M$ にまで低減する手法について説明する.

4.1 循環行列による観測

観測行列 Φ のサイズを低減するため, 観測行列に循環行列を用いた圧縮センシング手法を提案する. 循環行列とは, 式 (8) に示されるように, $\phi_i(j)$ を行ベクトル ϕ_i の j 番目の要素とした時, $1 \leq k \leq N, 1 \leq i \leq N$ に対して, $\phi_i(\text{mod}(N, k + i - 1)) = \phi_1(k)$ となる要素から成る行列のことである. ここで, $\text{mod}(a, b)$ は a を法とする b の剰余を表す. 循環行列では, 最初の 1 行のみを保存することで

残りの行のすべての要素を、行のシフト操作によって再構築することができるため、観測行列を実装するために必要なメモリ量は観測対象の信号サイズ N にまで低減することができる。観測行列を循環行列にすると、式 (7) の RIP を満たさなくなることがある。しかし、そのような場合でも完全にランダムな行列を観測行列として用いる通常の圧縮センシングと同様に、信号を復元することができることが Bajwa らによって確認されている [16]。

$$\Phi = \begin{pmatrix} \phi_{1,1} & \phi_{1,2} & \phi_{1,3} & \cdots & \phi_{1,N} \\ \phi_{1,2} & \phi_{1,3} & \phi_{1,4} & \cdots & \phi_{1,1} \\ \vdots & \vdots & \vdots & & \vdots \\ \phi_{1,M} & \phi_{1,M+1} & \phi_{1,M+2} & \cdots & \phi_{1,M-1} \end{pmatrix} \quad (8)$$

4.2 循環行列のランダム性

上節のように、循環行列を用いることでセンサノード上に実装しなければならない観測行列のサイズを N にまで低減することができる。しかし、圧縮センシングにおいては、観測行列のランダム性は復元効率を左右する重要な要素となる。循環行列は通常の圧縮センシングで用いられる完全ランダム行列と比べて、ランダム性が低いため、観測対象の信号によっては復元精度が劣化してしまうことがある [17]。そこで、観測行列にランダム性を与えるために、行列を循環シフトする幅を 1 から N まででランダムに変更することを考える。すなわち、循環行列の行をランダムに M 行、抜き出すことにより、観測行列 Φ を生成する。このためには、第一行目のベクトルだけでなく、どれだけ要素をシフトさせるかの乱数が M 個必要となる。これより、観測行列を実装するために必要なメモリ量は、 $N + M$ となる。

4.3 圧縮のアルゴリズム

以上のような観測行列を用いたときの、センサノード上でのデータ圧縮のアルゴリズムをアルゴリズム 1 に示す。圧縮センシングではセンシングデータ d における k 番目の要素 d_k は、観測対象の信号 x と圧縮行列 Φ の第 k 行ベクトル $\phi_k = (\phi_{k,1}\phi_{k,2}\dots\phi_{k,N})$ を用いて、 $d_k = \sum_{i=0}^N x_k\phi_{k,i}$ と表される。ここで、アルゴリズム 1 のように、 i 番目のセンシングデータ x_i をセンシングした際に、圧縮行列 Φ の第 i 列ベクトル $\phi_i = (\phi_{1,i}\phi_{2,i}\dots\phi_{N,i})^T$ との積 $x_i\phi_i$ を計算し、以前得られたセンシングデータ d へ加算する、という計算を N 回繰り返すという形で圧縮計算の順序を変更する。これにより、圧縮センシングにおける圧縮と観測を同時に行うことができるという性質を利用している。

5. センサノード上における圧縮計算およびデータ送信時の消費電力測定

本節では第 4 にて提案した観測行列を、実際の無線セン

Algorithm 1 Compression

```

x[N] : Signal to sense
d[M] : Sensing data
Φ[N] : Measurement matrix
idx[M] : Random number from 0 to N - 1
i = 0, j = 0 : Iteration indices
if ith data x[i] sensed then
  while j < M do
    d[j] ← x[i] × Φ[(i+idx[j]),N]
    j ← j + 1
  end while
  i ← i + 1
end if

```



図 4 消費電力測定の様子

サノードである eZ430-RF2500[18] に実装し、圧縮計算とデータ送信で消費される電力を測定し、評価する。

5.1 測定環境

センサノード上での圧縮計算にかかる消費電力を測定するにあたり、無線センサノードとして Texas Instruments 製の eZ430-RF2500 を使用し、10 分に一度センシングを行い、1 日分の 144 個のデータを圧縮して送信する場合、すなわち元信号のサイズ $N = 144$ となる場合を想定して、圧縮計算及びデータ送信にかかる消費電力を測定した。eZ430-RF2500 に第 4.2 節にて提案した観測行列 Φ を実装し、ノード上で圧縮計算を行うときの消費電力を測定する。ここで、今回用いた観測行列は各要素に ± 1 が等確率でランダムに現れる、 $\Phi = \{+1, -1\}$ となる行列である。以下、本稿ではこの行列をバイナリ行列と呼ぶ。測定は Agilent 社のマルチメータ 34410A/11A を用いて行い、図 4 のように eZ430-RF2500 に 3V 電源を接続し、電源とノード間にマルチメータを電流計として接続することにより、圧縮計算時及びデータ送信時にノードに流れる電流を測定した。

図 5 に送信データのサイズ $M = 30$ にまで圧縮した場合の、圧縮計算とデータ送信で消費される電流を示す。約 35~200 ms の間に圧縮計算が行われ、200~250 ms の間でデータ送信が行われている。圧縮計算では平均で約 3 mA

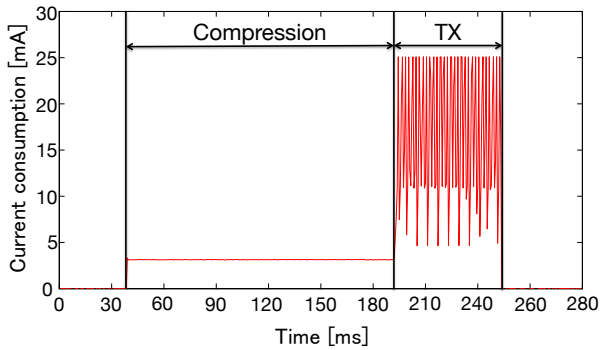


図 5 $M = 30$ に圧縮したときの電流

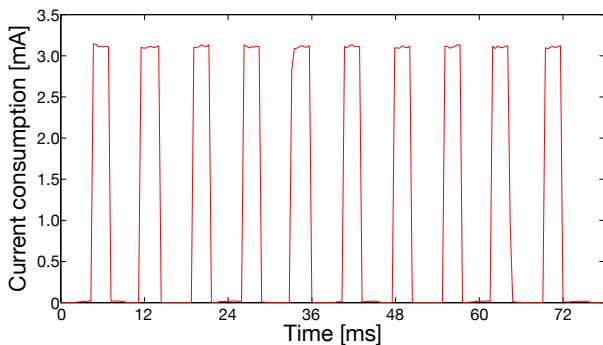


図 6 $M = 1$ の場合における圧縮計算で流れる電流

の電流が流れるのに対して、データの送信では平均で約 16 mA の電流が流れており、圧縮計算を行う場合と比べてデータの送信には約 5.3 倍の電流がノードに流れていることがわかる。しかしながら、図から明らかであるように圧縮計算にかかる時間はデータ送信と比べて 100 ms 以上も多くの時間を必要としている。また、データ送信では一つのデータを送るたびに電流の変動が大きく出ており、一つのデータを送るために必要な電力にばらつきが大きい。このため、電力の瞬時値だけでなく、実際の計算と送信にかかる時間を考慮して消費エネルギーの比較を行う。

5.2 圧縮計算時の消費エネルギー

圧縮後のサイズ $M = 1$ となる場合について、圧縮計算でノードに流れる電流を測定する。測定は 10 回行い、10 回それぞれの場合において流れた電流を図 6 に示す。それぞれの測定において、圧縮にかかった時間は約 4.2 ms であり、消費エネルギーは図 6 の電流が流れた部分の面積に、電源電圧 3 V をかけることによって計算する。計算の結果、 $M = 1$ の場合において、圧縮時にノードで消費されるエネルギーは平均で $40.5 \mu\text{J}$ であった。ここで、図 5 のように、圧縮計算時に流れる電流はほぼ一定値であり、また圧縮計算は行列積であるため圧縮にかかる時間も圧縮後のサイズ M に比例して増加していくと考えられる。これより、圧縮時の消費エネルギーは、圧縮後のサイズ M に対して $40.5 \times M \mu\text{J}$ であると考えられる。

また、圧縮計算の計算量は行列積であることから $O(NM)$

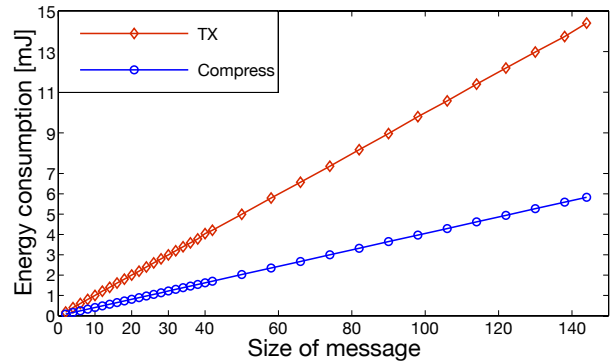


図 7 データ送信時の消費エネルギー

であるため、観測対象の大きさ N に対して、一度の圧縮で消費されるエネルギーは $40.5 \times \frac{N}{144} \mu\text{J}$ となると考えられる。これより、 N, M に対して消費されるエネルギーは一般的に、 $0.281 \times M \times N \mu\text{J}$ となる。

5.3 データ送信時の消費エネルギー

次に、データ送信時における消費エネルギーを測定する。送信するデータのサイズ M を 2~144 まで変化させ、それぞれの M についてデータの送信時にノードに流れる電流を 10 回測定し、消費エネルギーを計算した。10 回の測定で得られた消費エネルギーの平均を図 7 の赤線に示す。データ送信時には $M = 2 \sim 144$ の間で平均 $0.18 \sim 14.4 \text{ mJ}$ のエネルギーが消費されている。一方、図 7 の青線は前節にて述べた圧縮計算で消費されるエネルギー $40.5 \times M \mu\text{J}$ を表したものである。この図より、圧縮計算とデータ送信では、データ送信で消費されるエネルギーの方が約 2 倍程度大きく、また送信するデータサイズが大きくなるほどデータ送信の方がより消費するエネルギーが増大していることがわかる。

また、144 個すべてのデータを圧縮計算を行わずに送信する場合の消費エネルギー、すなわち 14.4 mJ を 100 で正規化した場合に、提案手法の圧縮センシングによりデータの送信を行った場合の消費エネルギーの関係を図 8 に示す。このとき、圧縮後のメッセージサイズ M が 106 を超えると、圧縮計算による消費エネルギーの増大がメッセージサイズを小さくすることによる送信時の消費エネルギーの削減効果を上回ってしまう。これより、ノードの省電力化を行うためには、 $M < 106$ すなわち圧縮率 $1 - \frac{M}{N}$ が 24.6% 以上である必要がある。

ここで、一般的な場合を考える。図 7 において、送信時の消費エネルギーを最小二乗法により直線で近似する。このとき得られる直線の傾きは 0.0998 である。これより、提案手法による総消費エネルギーは観測対象のサイズ N と圧縮後の信号サイズ M に対して $0.281 \times 10^{-3} \times M \times N + 0.0998 \times M \text{ mJ}$ となる。提案手法により消費エネルギーの削減を行うためには、 $0.281 \times 10^{-3} \times M \times N + 0.0998 \times M \leq 0.0998 \times N$

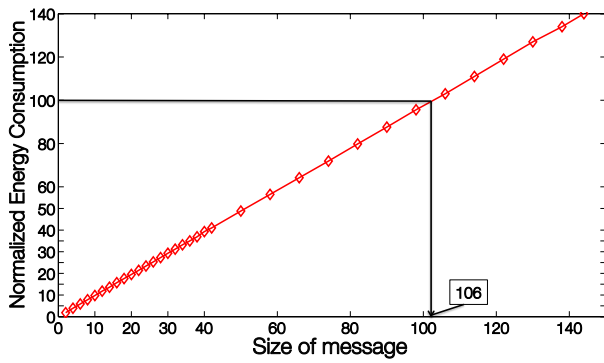


図 8 圧縮を行わない場合で正規化した消費エネルギー

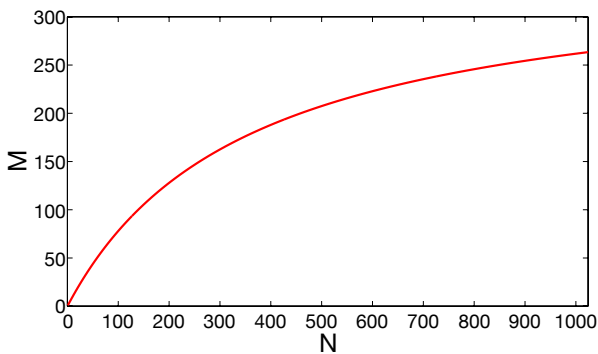


図 9 N に対する M の境界条件

を満たす必要がある．これより，圧縮後の信号サイズ M が満たすべき式は $M \leq N / (1 + 2.82 \times 10^{-3} \times N)$ となり， M の境界条件は図 9 のように与えられる．図 9 のように N が大きくなるほど満たすべき M の増加は鈍化していくため，圧縮による消費エネルギーの削減効果を得るためには，信号の圧縮率 $1 - \frac{M}{N}$ を大きくする必要がある．

5.4 実測の土壤水分データへの適用

実測データに対するセンサノードの省電力化の効果を確認するため，土壤水分データへの提案手法による圧縮センシングの適用を行う．観測対象となる水分データは図 10 のようにして収集した．土壤水分は DECAGON 社の土壤水分センサ EC-5 を用いて 10 分毎に計測を行い，データロガー Em5b にデータを集積する [19]．図 11 は 2012/9/23 00:00 から 2012/9/29 11:59 までの土壤水分データである．9/26，9/27 の 10:00 及び，9/29 の 12:00 に土壤に水を与えており，給水直後は水分含有量が大きく上昇し，その後緩やかに減少していることがわかる．また，9/24 のように土壤への水やりが行われていないときにも含水量が上昇している時刻が存在する．図 10 からわかるように，土壤水分は急峻に上昇をする場合を除いて，緩やかに変化することが大半である．そのため，時系列で連続した 2 つのデータの差分をとることで，データをスパースにすることが可能である．

得られたデータを 1 日分の $N = 144$ 個に分割し，それぞ

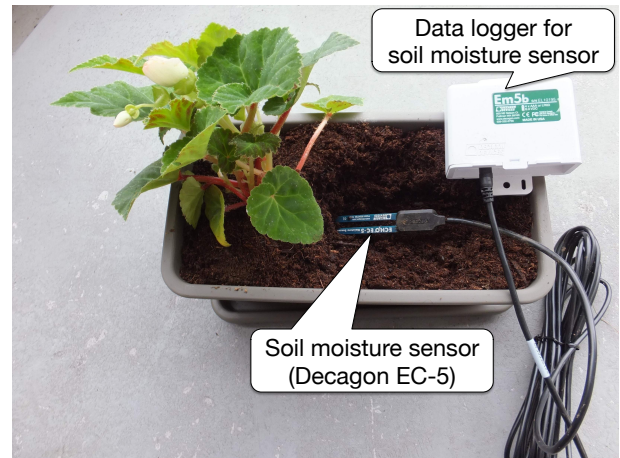


図 10 土壤水分計測の環境

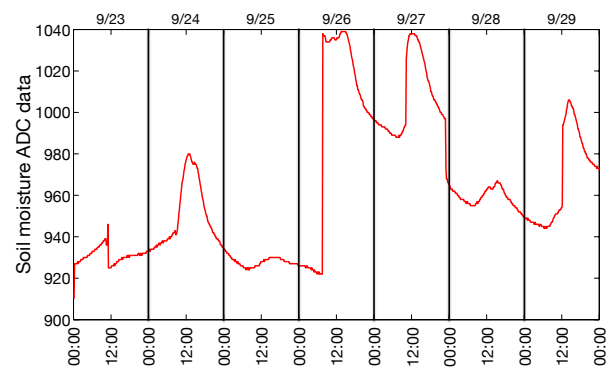


図 11 土壤水分データ

れに対して圧縮後のサイズ M を変更しながら提案手法による復元を行った．ここで，復元後に生じる誤差に関しては土壤水分の急激な変化部分を正確に復元することを目的に，正規化平均二乗誤差 (Normalized Mean Square Error, 以下 NMSE) を 10^{-5} 以下とすることを目標とした．図 12 は元データに対して $NMSE=10^{-5}$ となる復元結果を表している．図の赤線は元の土壤水分データを，青線が復元されたデータを表している．ここで，復元結果は，9/23 のはじめの部分や 9/28~9/29 にかけての緩やかに変化をしている部分についての誤差が大きく出ている．しかしながら，図の印をつけた部分のように，水分量が急峻に上昇をする部分については精度よく復元ができており，重要な変化を生じている部分についての復元ができています．これは，信号が差分でスパースであるため急峻な変化があると変化部分のスパース表現における係数が大きくなりエネルギーがその部分に集中するため， l_1 最小化の復元によって係数の大きい変化部分がより復元されやすいためであると考えられる．

ここで，表 2 に $NMSE=10^{-5}$ を初めて満たす M と，提案手法により消費されたエネルギー及び，削減された消費エネルギーの割合を示す．図 9 から消費エネルギーを削減できる M の上限値は $N = 144$ に対して， $M = 102$ であり，いずれもこの条件を満足している．削減されたエネル

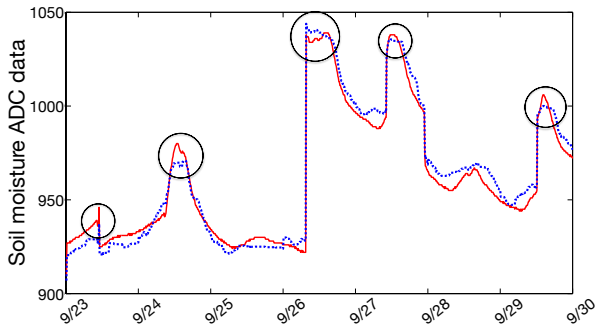


図 12 NMSE=10⁻⁵ で復元された土壌水分データ

表 2 NMSE=10⁻⁵ を満たす M と削減された消費エネルギー

Date	M	Calc. energy [mJ]	TX energy [mJ]	Total energy [mJ]	Saved energy [%]
Sept.23	31	1.26	3.09	4.35	69.8
Sept.24	79	3.20	7.88	11.08	23.0
Sept.25	33	1.34	3.29	4.63	67.8
Sept.26	73	2.96	7.29	10.24	28.9
Sept.27	83	3.36	8.28	11.64	19.1
Sept.28	57	2.31	5.69	8.00	44.5
Sept.29	71	2.88	7.09	9.96	30.8

ギーの割合は、圧縮を行わず $N = 144$ 個のデータをもそのまま送信したときのエネルギーに対して、どれだけエネルギーが削減されたかを百分率で示している。表からわかるように、24、26、27、29日の水分量が急激に変化する日には、圧縮後のサイズ M がより大きく必要であり、削減できるエネルギーは 20 ~ 30%程度である。実際の運用上は、ノード内では一日の水分量変化が急峻なものになるかどうかは判断不明であるため M は固定となる。これより、 M は最大の $M = 83$ で設定することになり、全体でおよそ 19%の消費エネルギーの削減を行うことができる。

6. おわりに

本稿では、循環行列に注目し観測行列のサイズを $N \times M$ から $N + M$ にまで低減することで、メモリ量の乏しい実際の無線センサノードに実装可能となる圧縮センシングの手法を提案した。また、実際の無線センサモジュールである eZ430-RF2500 に提案手法を実装し、圧縮計算とメッセージ送信に消費されるエネルギーを計測した。計測の結果から元の信号サイズ N 、圧縮後のサイズ M に対して $M \leq N / (1 + 2.82 \times 10^{-3} \times N)$ を満たすならば、提案手法による消費エネルギーの削減効果があることを示した。

本研究は NEDO 平成 24 年度産業技術研究助成事業の一環として実施された。

参考文献

[1] Estrin, D., Sayeed, A. and Srivastava, M.: Wireless Sensor Networks, *ACM Mobicom, Tutorial*, Atlanta, USA (2002).
 [2] Handy, M. J., Hasse, M. and Timmermann, D.: Low Energy Adaptive Clustering Hierarchy with Deterministic Cluster-Head Selection, *4th International Workshop on Mobile and Wireless Communications Network*, pp.

368–372 (2002).
 [3] Cristescu, R., Beferull-Lozano, B., Vetterli, M. and Wattenhofer, R.: Network correlated data gathering with explicit communication: NP-completeness and algorithms, *Networking, IEEE/ACM Trans. on*, Vol. 14, pp. 41–54 (2006).
 [4] Ciancio, A., Pattem, S., Ortega, A. and Krishnamachari, B.: Energy-efficient data representation and routing for wireless sensor networks based on a distributed wavelet compression algorithm, *Proc. of the 5th international conf. on IPSN*, Tennessee, USA, pp. 309–316 (2006).
 [5] Candès, E. and Wakin, M.: An introduction to compressive sampling, *IEEE Signal Processing Magazine*, Vol. 25, No. 2, pp. 21–30 (2008).
 [6] Donoho, D.: Compressed sensing, *Information Theory, IEEE Transactions on*, Vol. 52, No. 4, pp. 1289–1306 (2006).
 [7] Candès, E.: Compressive Sampling, *Proceedings of the International Congress of Mathematicians*, Vol. 3, pp. 1433–1452 (2006).
 [8] Baraniuk, R.: Compressive sensing, *IEEE Signal Processing Magazine*, Vol. 24, No. 4, p. 118 (2007).
 [9] Candès, E.: The restricted isometry property and its implications for compressed sensing, *Comptes Rendus Mathématique*, Vol. 346, No. 9–10, pp. 589–592 (2008).
 [10] Baraniuk, R., Davenport, M., DeVore, R. and Wakin, M.: A Simple Proof of the Restricted Isometry Property for Random Matrices, *Constructive Approximation*, Vol. 28, No. 3, pp. 253–263 (2008).
 [11] Rudelson, M. and Vershynin, R.: Sparse reconstruction by convex relaxation: Fourier and Gaussian measurements, *Information Sciences and Systems, 2006 40th Annual Conference on*, pp. 207–212 (2006).
 [12] Bajwa, W., Haupt, J., Sayeed, A. and Nowak, R.: Compressive Wireless Sensing, *Proc. of the 5th international conf. on IPSN*, Tennessee, USA, pp. 134–142 (2006).
 [13] Mahmudimanesh, M., Khelil, A. and Suri, N.: Reordering for Better Compressibility: Efficient Spatial Sampling in Wireless Sensor Networks, *2010 IEEE International Conf. on Sensor Networks, Ubiquitous, and Trustworthy Computing*, California, USA, pp. 50–57 (2010).
 [14] Luo, C., Wu, F., Sun, J. and Chen, C. W.: Compressive Data Gathering for Large-Scale Wireless Sensor Networks, *Proc. of the 15th Annual Int. Conf. on Mobicom*, pp. 145–156 (2009).
 [15] Nguyen, N., Jones, D. and Krishnamurthy, S.: Netcompress: Coupling network coding and compressed sensing for efficient data communication in wireless sensor networks, *IEEE Workshop on Signal Processing Systems (SIPS)*, San Francisco, USA, pp. 356–361 (2010).
 [16] Bajwa, W. U., Haupt, J. D., Raz, G. M., Wright, S. J. and Nowak, R. D.: Toeplitz-Structured Compressed Sensing Matrices, *Statistical Signal Processing, IEEE/SP 14th Workshop on*, pp. 294–298 (2007).
 [17] Yin, W., Morgan, S., Yang, J. and Zhang, Y.: Practical Compressive Sensing with Toeplitz and Circulant Matrices (2010).
 [18] TEXAS INSTRUMENTS: eZ430-RF2500 Development Tool User's Guide, <http://www.tij.co.jp/lit/ug/slau227e/slau227e.pdf>.
 [19] DECAGON: DECAGON DEVICES, <http://www.decagon.com/products/sensors/soil-moisture-sensors/ec-5-soil-moisture-small-area-of-influence/>.