

無線センサーネットワークの位置決めと時刻同期手法の開発 ～ 機動的な多点計測の実現に向けて ～

Development of localization and time-synchronization methods for wireless sensor network

～ For mobile measurement of structures ～

本多弘明*・小国健二**・井上純哉***・堀宗朗****

Hiroaki HONDA, Kenji OGUNI, Junya INOUE and Muneo HORI

*工修 東京大学大学院博士課程 工学系研究科 社会基盤学専攻 (〒113-8656 東京都文京区本郷 7-3-1)

**Ph.D 東京大学助教授 東京大学地震研究所 (〒113-0032 東京都文京区弥生 1-1-1)

***工博 東京大学助教授 東京大学大学院工学系研究科 マテリアル工学専攻 (〒113-8656 東京都文京区本郷 7-3-1)

****Ph.D 東京大学教授 東京大学地震研究所 (〒113-0032 東京都文京区弥生 1-1-1)

This paper presents an implementation of localization(i.e., determination of positions) and time synchronization methods for sensor network. Relative positions between sensor nodes are estimated based on acoustic ranging through *Inverse Delaunay Algorithm*. Noise tolerant acoustic ranging algorithm that employs digital signal processing techniques is implemented in an off-the-shelf sensor platform (Mica2). Field experiment was conducted with 21 sensor nodes to evaluate the accuracy of the localization. Besides a time synchronization method between sensor nodes based on time stamp is developed. Experiments show that the shift of synchronization is under 2ms for 500Hz sampling.

Key Words : *Network sensing, Acoustic ranging, Localization, Time Synchronization*

キーワード：ネットワークセンシング，音響測距，位置決め，時刻同期

1. はじめに

ワンチップマイコンと呼ばれる CPU とメモリを併せ持つチップは，家電や工業機械などの制御に一般的に用いられている，省電力かつ安価な電子デバイスである^{1) 2)}。このワンチップマイコンと小型センサーが無線端末に搭載されたセンサープラットフォームを小型・安価・省電力という条件で製造することが近年の無線通信技術とセンシング技術の進歩に伴い可能となっている。

現在製品化されているセンサープラットフォーム（以下，センサーノード）の多くは，省電力化の結果小型バッテリーで駆動可能となっている。そのため，通信や電源確保に必要なケーブル類が一切不要となり，多点計測で問題となる設置コストが大幅に抑えられる。また，各センサーノードに搭載されたマイコンによる計測・通信の制御や計測データの簡易処理も容易に実装可能である。

例えば TinyOS³⁾ というセンサーノード用の組み込みソフトウェアがインストールされたセンサーノードを用いると，センサーノード群が自律的に通信ネットワークを構成し，どのセンサーノード間でも通信経路が複数のセンサーノードを介してマルチホップで確保される。つまり，基地局との通信を介さずに，局地的なセンサーノード間の通信が可能である。そのため，基地局とその

周辺のセンサーノードの通信負荷・電力消費が抑制され，さらに，センサーノード群をネットワークとして用いる多様なアプリケーションの実装が可能となっている。

このようなセンサーノードを用いた構造物の稠密計測を想定すると，100～1000 個程度のセンサーノードが必要となるため，各センサーノードの位置を短時間に自動的に測定する機能がなければ，計測全体に掛ける時間・労力の多くをセンサーノードの位置決めに費やすことになり非効率的である。また有線でデータを収集する場合とは異なり，多数のセンサーノードの時刻同期を無線環境下で確保する方法を実装する必要がある。本研究では，前者に対しては，センサーノード間の音響測距の実装と距離データを基にした位置決めアルゴリズムの構築を行い，後者に対しては，半径 10m 程度の小規模エリアに展開されるセンサーネットワーク（10～20 個に相当）に適用可能な，簡易な時刻同期手法を実装した。

本論文のうち，音響測距の実装と位置決めアルゴリズム (IDA) に関しては既発表⁴⁾ だが，さらに IDA と同期計測が実装されることで，機動的な多点計測のための基本技術が揃うことになるため，これらを IDA の計算量の評価と併せて本論文を作成した。本論文で述べる位置決め可能なセンサーノードに，高精度に位置決めされた GPS 受信機を上層に加えた階層型位置決めシステムについては Saeki M, et al.⁵⁾ に述べられている。

2. 音響測距機能の実装

センサーノード間の測距の方法として、センサーノード間を伝達する電波や音波の到達時間 (TOA: Time Of Arrival) から距離を逆算する方法がある。この方法は、電磁波強度 (RSSI: Received Signal Strength Indicator) の測定を基にした方法⁶⁾ などと比較して、高密度なセンサーネットワークにおける高精度な測距を目的とした方法として有用であるとされている。特に、伝搬速度が電波より十分に遅く比較的安価な測定装置で計測可能な、音波を用いた測距技術の開発が多く行われている⁷⁾⁸⁾。本研究では可聴波の TOA を用いた測距方法を Mica2⁹⁾ に実装した。Mica2 の仕様を表-1 に示す。本研究で用いた音響デバイスはいずれも Mica2 にデフォルトで搭載されているものである。

表-1 Mica2 の主な仕様

One-Chip MicroComputer	
Chip	ATmega 128L
CPU	7.3MHz
RAM	4kB
Flash	128kB
ADC	28.6kHz 10bit
RF Transceiver	
Chip	Chipcom CC1000
Frequency	315MHz
FSK data rate	38.4Kbaud
Microphone	
Chip	WM-62A
SNR	more than 58dB
Sounder	
Chip	PS14T40A
Frequency	4.5kHz

2.1 計測方法

TOA を基に距離計測を行うためには、音波が発信された時刻と受信された時刻の差を正確に知る必要がある。このため、音響測距を行う際に、発信ノードと受信ノードの時刻同期が必要である。本研究では、発信ノードは音波と同時に無線信号を発信し、受信ノードはその無線信号の受信と同時に音波のサンプリングを開始する、という方法で同期をとった。この方法によると、無線信号と音波の伝播時間の差 (TDOA: Time Difference Of Arrival) が計測されることになる (図-1) が、無線信号は音波よりも遥かに速く (約 10^6 倍で) 到達するため、

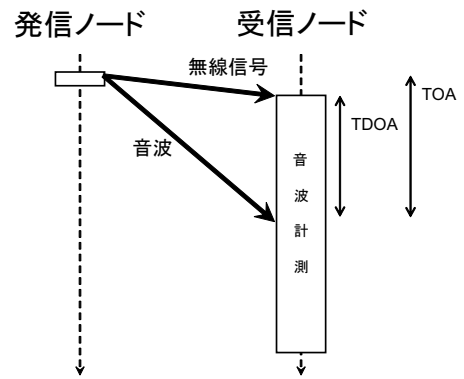


図-1 TDOA と TOA

音波で測距を行うことを考えると、測定された時間は音波の TOA とほぼ同一と見做せる。特に、本研究で用いた ADC (Analog to Digital Converter) のサンプリング間隔 (約 $35\mu s$) を考慮すると、例えばセンサーノード間が $10m$ 離れていた場合には無線信号の到達時間は約 $33ns$ となり、計測データに影響は現れない。

さらに、マイクロフォンで得られた波形データの S/N 比を改善するためノイズを除去する必要がある。音波に含まれるノイズには周辺環境から偶発的に発せられるランダムなノイズと、壁などにより生じる反射波がある (図-2 参照)。まず、同一の計測を複数回スタッキング

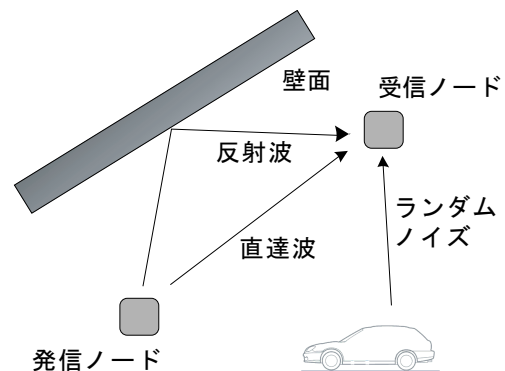


図-2 ノイズの発生要因

する方法をとることで、偶発的なノイズを除去した。S/N 比はスタッキング回数を N として $10\log(N)dB$ で改善される。本研究では 64 回スタッキングを行ったため、S/N 比の改善は $18dB$ となる。なお、一回の計測に掛かった時間は約 4 秒であった。一方、反射波は常に同じタイミングで計測される為、単純なスタッキングによっては除去できない。特に本研究では短時間に多数回音波

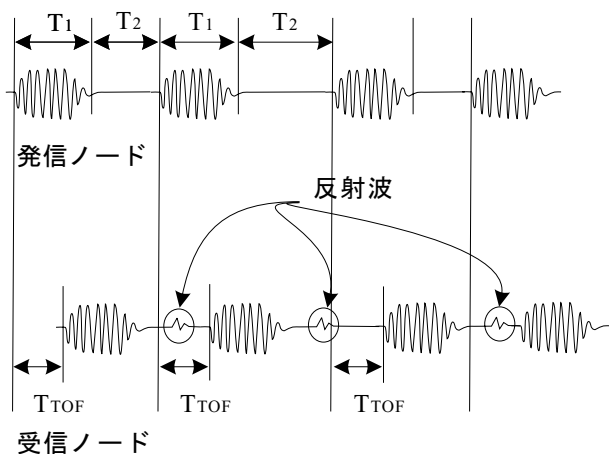


図-3 スタッキングによるノイズの除去: T_1 一定, T_2 ランダム

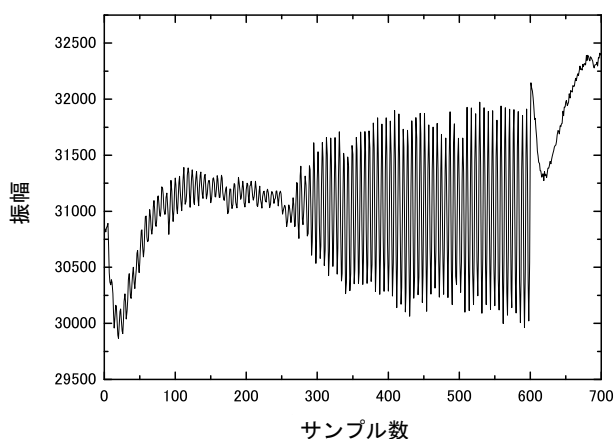


図-4 スタッキング後の波形データ

を発信するため、例えば一つの音波によって発生した反射波が以降の音波の到達直前に観測されるケースがあり、このケースでは、実際よりも短く距離が算定されてしまう。そこで本研究では 図-3 に示すようにサンプリングの間隔をランダムに設定することで、複数回のサンプリング時に生じた反射波の位相をランダムにずらし、反射波の影響を除去した。図-4 にスタッキング後の波形データを示す。この図から、計測データには顕著な低周波及び高周波のノイズが含まれていることが分かる。なお、図には得られた 10 ビットのデジタルデータをそのまま掲載しているため、単位は付していない (図-5・図-6 も同様)。これらの図ではともに 64 回スタッキング後のデータであるため振幅の絶対値が大きくなっている。

2.2 データ処理方法

まず、スタッキング後の波形データからブザー音の周波数帯域 (4.0 ~ 5.0kHz) を FIR バンドパスフィルタに

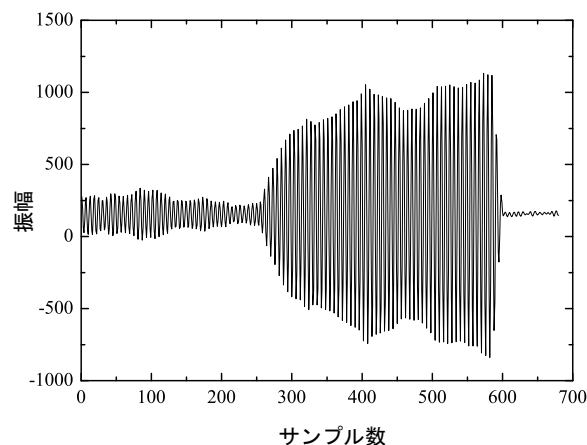


図-5 デジタルフィルタ適用後の波形データ

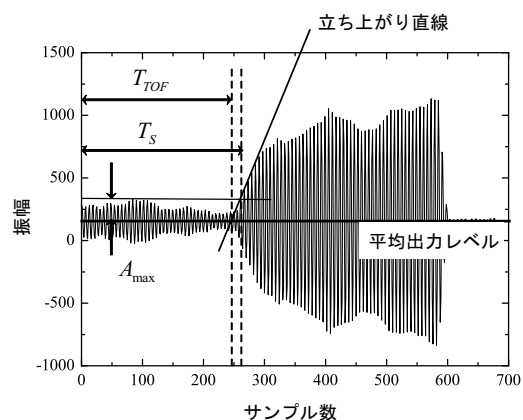
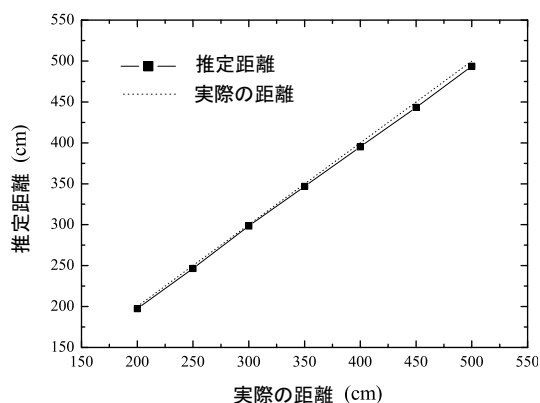


図-6 直達波到達時間抽出方法

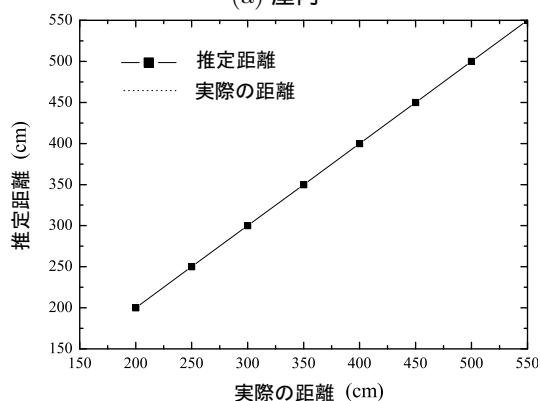
より抽出する。この結果、アナログ回路固有の低周波ノイズと、他の電子機器や内部クロックに起因する高周波ノイズが抑えられる。図-5 にデジタルフィルタ通過後の波形データを示す。

次に、波形データから音波到達時刻を抽出する。このためには、ノイズを除去した波形データで最初に発生する顕著なピークの時刻を得れば良い。しかし、実際にはブザーに電圧を与えてから最大音量が出力されるまでに時間が掛かるため、図-5 に見られるように、振幅が極大値をとる時刻は第一波が到達した時刻よりも後となり、第一波の振幅はノイズ振幅より小さく検出できない。このタイムラグの長さはブザー間で異なるため、本研究では全てのブザーで振幅が線形に立ち上がると仮定し、以下の方法により音波の TOA を求めた (図-6 参照)。

1. 静音時におけるノイズの抽出: サンプリング開始から 40 番目のピークまでを静音領域とし、設置された環境中の最大ノイズレベルを 40 番目までのピーク中の最大値 A_{max} とする。
2. ブザー音到達領域の抽出: 連続してピークレベルが

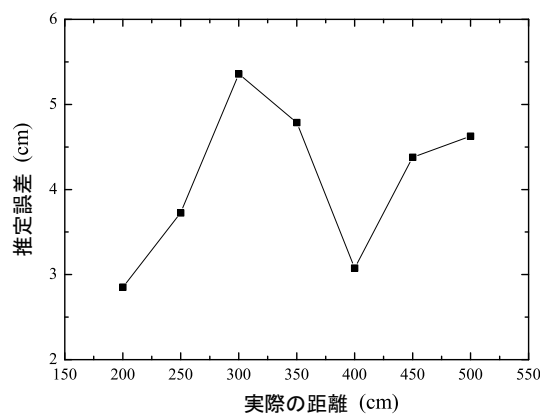


(a) 屋内

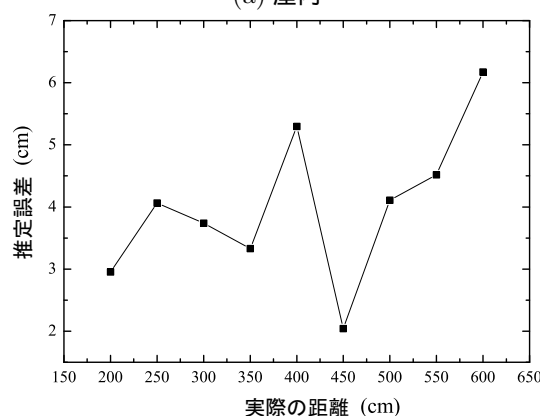


(a) 屋外

図-7 推定距離と実際の距離の関係



(a) 屋内



(b) 屋外

図-8 推定誤差 (標準偏差) と実際の距離の関係

A_{max} を超える領域をブザー音到達領域とし、最初に A_{max} を超える時間を T_s とする。

3. 第一波到達時刻の推定: ADC の平均出力レベルと T_s から始まる振幅の立ち上がりの交点を、第一波到達時刻 T_{TOF} とする。 T_{TOF} は簡単なカルマンフィルタを用い、 T_s から 20 サンプル過ぎたところを起点として時間を逆方向に推定した。

2.3 精度検証試験

交通騒音の大きな屋外環境、及び残響の大きな屋内環境、それぞれにおいて検証試験を行った。計測距離の範囲は屋内で 2.0 ~ 5.0m, 屋外で 2.0 ~ 6.0m と設定し、それぞれの距離・環境で 250 回計測を行った。

図-7 に実際の距離と計測値の関係を、図-8 に実際の距離と計測誤差の関係を示す。いずれの環境においても、本研究で開発された手法は測定距離 5m 以下であれば 6cm 以下の測定誤差で測距が可能であることがわかる。屋外と比較し屋内は環境騒音の影響は少ないが、閉塞空間であることから残響が大きいいため、測定距離が 3m 程度で屋外よりも測定誤差が大きくなっている。計測誤差の標準偏差は実際の距離の 1% 前後となっている。

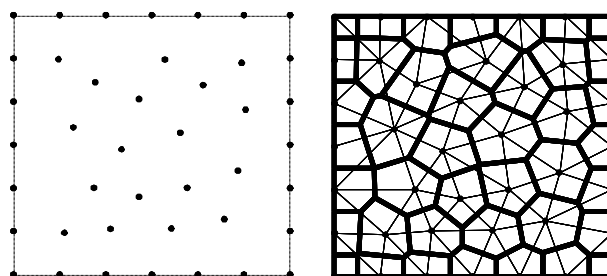


図-9 左: 母点 (位置決めされるべきセンサーノードに対応), 右: Voronoi 分割 (太線) と Delaunay 分割 (細線)

3. Delaunay 分割を用いた位置決めアルゴリズム – *Inverse Delaunay Algorithm*

本節では、センサーネットワーク内で計測された個々のセンサーノード間の距離の情報のみに基づいて、全てのセンサーノードの位置を同定するためのアルゴリズム、*Inverse Delaunay Algorithm* (以下、IDA) について簡単に説明する (以下図-9 参照)。

アルゴリズムの説明の前に、Delaunay 分割など、本アルゴリズムで利用する空間分割の手法について概説する（詳しくは¹⁰⁾ 参照）。以下の説明では、カッコ内に 2 次元平面を想定した場合に対応する用語を記述した。

与えられた有限個の点「母点」を基にして空間（平面）を分割する方法に Voronoi 分割と Delaunay 分割という分割方法がある。母点ごとに、その母点が最も近い母点であるような点の集合で空間を分割すると、母点一点を含む凸多面体（凸多角形）によって分割される。この分割のことを Voronoi 分割と呼び、分割の一単位を Voronoi 領域と呼ぶ。一方、Voronoi 分割の各領域の面（辺）を共有する母点どうしをつなぐことにより分割を行うと、各母点を頂点とする多面体（多角形）により分割される。この分割のことを Delaunay 分割と呼び、各分割領域である多面体（多角形）は一般的に四面体（三角形）となり、Delaunay 四面体（三角形）と呼ばれる。

Delaunay 分割には、「領域を分割する Delaunay 四面体（三角形）の外接球（円）内部には他の母点は存在しない」という特徴があり、以下で述べる IDA の「ローカルクラスターの作成」の際に利用する。ここで、ローカルクラスターとは、相対的な位置関係が定まった複数個のノードの集合のことである。ローカルクラスターどうしの位置関係をノード間の距離データを基に計算し、各ローカルクラスターの座標軸を統合することにより、全ノードの相対位置が決定される。IDA は、

1. ローカルクラスターの作成
2. ローカルクラスターのつなぎあわせ

の 2 つの手続きからなる。以下にそれぞれの手続きの概要を述べる。なお、簡単のため以下の説明は二次元的に配置された点群の相対位置決定問題を例にとって進めるが、三次元問題への拡張は理論的には容易である。

3.1 ローカル・クラスターの作成

図-10 にローカル・クラスターの例を示す。図中 CN 及び 1~6 の 7 点で構成される領域が 1 つのローカル・クラスター（クラスター A）に対応する。計測対象となる平面上におけるローカル・クラスターの位置が既に決まっているように描かれているが、実際には位置決めが終了したときにローカル・クラスターの平面上の位置が決定される。

まず、個々のローカル・クラスターに対し、クラスター中央のノード（「センター・ノード」と呼ぶ）をこの座標系の原点として、二次元の Euclid 座標系が定義される。センター・ノードを頂点の一つとする Delaunay 三角形は一般的に複数存在するが、この Delaunay 三角形の頂点でセンター・ノード以外のものを「サテライト・ノード」と呼ぶ。このセンター・ノードとサテライト・ノード

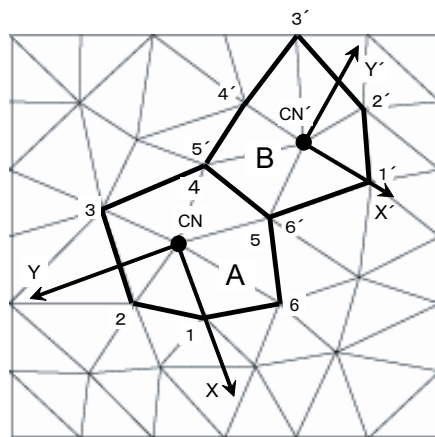


図-10 ローカル・クラスターの例 (CN:CenterNode)

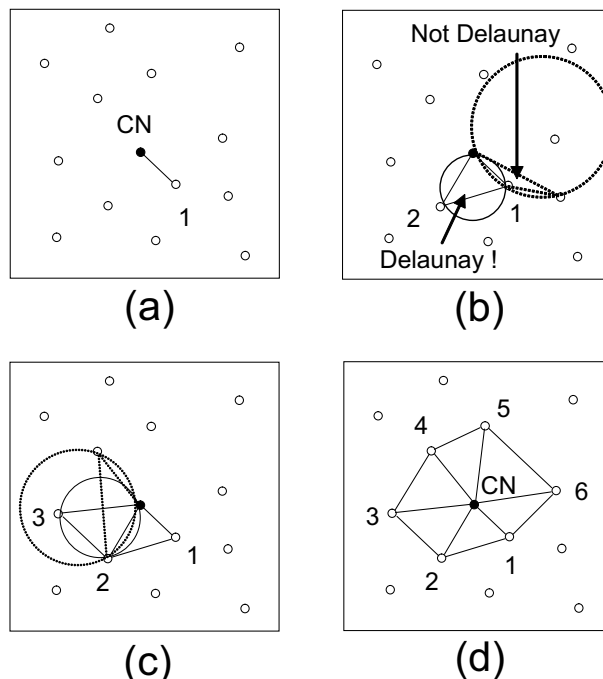


図-11 Delaunay クラスターの生成

ドが一つのローカル・クラスターを構成するメンバーとなる。サテライト・ノードのローカル座標系における座標値は、ノード間の距離データを基に計算され、さらにサテライト・ノードにはローカル座標系上で反時計回りの順番で番号が付けられる。一つのノードは、一つのローカル・クラスターのセンター・ノードであり、同時に一つ又は複数のローカル・クラスターのサテライト・ノードでもある。従って一つのノードの属性から、そのノードを共有しているクラスターどうしの関係が分かる。例えば図-10 において、クラスター A のサテライト・ノード 4, 5 が、クラスター B のサテライト・ノードとして 5, 6 と番号付けられているため、クラスター A とクラスター B とでは面の向きが反対であることが分かる。

個々のローカル・クラスターは図-11 に示す手順に

従って生成される。

1. 音響測距によって得られたセンサー・ノード間の距離情報を基に、センター・ノードに最も近い点を同定する(図-11 (a))。このクラスターのローカル座標系は、この第1のサテライト・ノードが+ X 軸上に位置するように設定される。
2. センター・ノード付近のノードの中から、上記2ノードと Delaunay 三角形を構成するノードを第2のサテライト・ノードとする。(図-11 (b) 中の点線で描かれた三角形は外接円の中に他のノードを含むため Delaunay 三角形ではない。従って、ローカル・クラスターの構成要素にはなりえず、却下される。) なお、第2のサテライト・ノードの y 座標が正となるようにローカル座標系を定める。
3. センター・ノード付近のノードの中から、第2のサテライト・ノードとセンター・ノードとで Delaunay 三角形を構成するノードを第3のサテライト・ノードとする。以降のサテライト・ノードも同様に決定していく。(図-11 (c))。
4. センター・ノードが Delaunay 三角形で取り囲まれたら、サテライト・ノードに番号を割り当てる。(注: センター・ノードがノード群の辺縁部に位置している場合、Delaunay 三角形で取り囲めないことがあるが、センター・ノードを頂点にもつ Delaunay 三角形が最低ひとつ存在すればローカル・クラスターとして成立する。)

3.2 ローカル・クラスターのつなぎ合わせ

前節の手順に従い、全てのノードに対してローカル(Delaunay) クラスターが生成されたら、次は、全てのセンサーノードの位置を同一の座標で表現するために、ローカル・クラスターの座標系を統合することになる。同じノードを共有するローカル・クラスターの座標系のずれ(並進・回転・反転)を計算し、座標系の統合を行うことになるが、座標統合に全てのローカル・クラスターを用いる必要は無い。座標統合が行われるローカル・クラスター群の中に全てのセンサーノードが含まれていれば、全てのセンサーノードの位置を決めることができることに着目し、以下の様な手順でセンサーノードの相対位置を決めることにした。

1. 全てのローカル・クラスターの中から、座標統合に用いるクラスターを選ぶ。このとき、選ばれるクラスターは2種類に分けられる。ひとつは座標統合の核となる Atomic・Cluster(AC)、もうひとつは AC どうしの橋渡しに用いられる Bridging・Cluster(BC) である。全ての AC, BC の領域は、お互いに重なり合うことはない。また、AC どうし

で、サテライトノードを共有することもない。BC は、AC とサテライト・ノードを共有し、AC のメンバーでないノードをセンター・ノードとするクラスターである(図-12 参照)。

2. 複数の AC 座標系(あるいは AC, BC がいくつか統合されたクラスター・グループ(次頁図-13)の座標系)を BC を基に統合する。このとき、統合の要となる BC と統合されるクラスター(またはクラスター・グループ、以下省略)がサテライトノードを共有している必要がある。このとき、対応するサテライト・ノードの番号付けの順序が同じであれば、つながれるクラスターの面の方向が互いに逆向きであることを意味するため、どちらかのクラスターの座標系を反転させる必要がある。例えば図-12 において、AC と BC の対応する辺は AC の中では $1 \rightarrow 2$ と番号がつけられ、BC の中では $2 \rightarrow 3$ と番号がつけられている。どちらも昇順であり、面の方向が互いに反対である。最終的に、この例では図-12 に示すように BC が反転させられる。
3. クラスター間の座標のずれを計算し、必要な並進・回転・反転に対応する計算を座標値の変更が必要なら

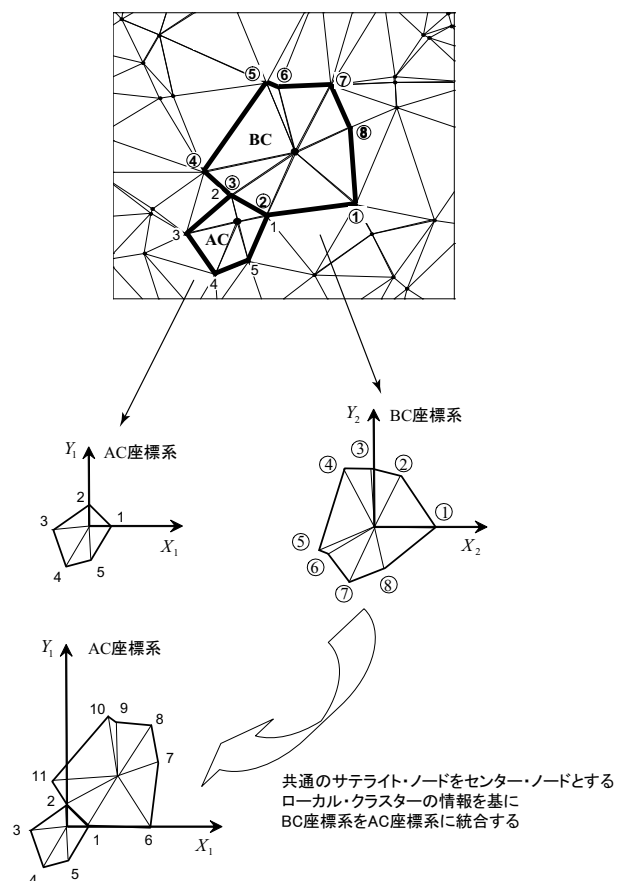


図-12 ローカルクラスターのつなぎ合わせ

ノード群に施す。

4. 全てのノードがひとつの座標系に統合されるまで，
2 番目と 3 番目の手順を繰り返す。

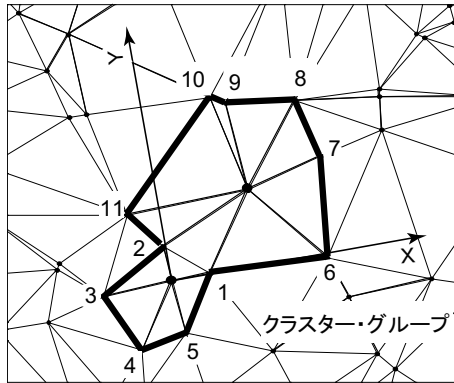


図-13 クラスタ・グループの例 (図-12 で統合されたクラスターがひとつのクラスターとして扱われる)

3.3 IDA における座標統合の計算量の評価

多数のセンサーノードで構成される計測システムへ IDA を適用する場合を想定し，IDA のスケーラビリティ（多数のセンサーノードからなるシステムへの拡張性）を評価する。

IDA 全体の計算量は，ローカルクラスターの作成とローカルクラスターのつなぎ合わせに要する計算量の総和であるが，ローカルクラスターの作成は各センサーノードに関して並列的に行われると仮定すると，スケーラビリティに関連するのはローカルクラスターのつなぎ合わせ，つまりローカルクラスターの座標統合に要する計算量である。

そこで，座標統合に要する計算量のセンサーノード数に対するスケーラビリティを評価した。具体的には，前節までに説明した位置決めアルゴリズムをコーディングし，位置決め計算を行う過程で，座標統合に要する計算量，すなわち，座標変換時の演算（回転・並進・反転）に要する計算量を算出した。

センサーノードを設置する領域として Euclid 平面上で， $-100 \leq X \leq 100$ ， $-100 \leq Y \leq 100$ を頂点とする 200×200 の正方形領域を仮定した（距離の単位は全て [m] とする）。ノード総数 (N_{all}) を

$$N_{all} = 10, 50, 100, 500, 1000$$

と変化させ，各 N_{all} について異なる配置を用いて多数回計算を行った平均値を結果として用いた。

結果は図-14 に示す通りである。 $N \leq 1000$ のノード総数に対して，演算回数はいずれも線形関係にある。既往研究¹¹⁾において $O(N^2)$ と見積もられていた計算回

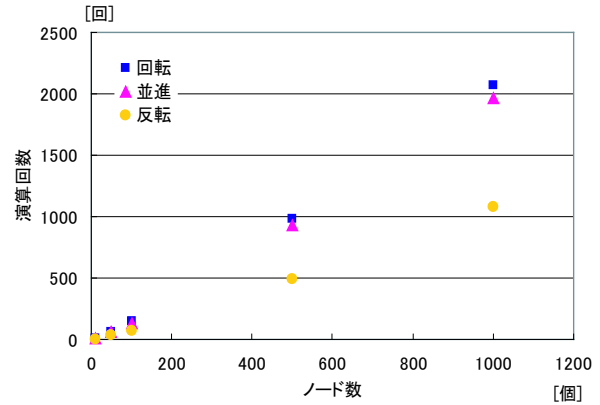


図-14 演算回数とノード総数の関係

数が，本アルゴリズムでは $O(N)$ となっており，IDA のスケーラビリティが既往研究より高くなっていることが示される。これは，座標統合計算を行う前に AC，BC の選定を行ったため，座標統合に用いるローカルクラスターの総数が少なく抑えられたためであると考えられる。

測距データに含まれる誤差は座標変換の計算を重ねるごとに蓄積していく。従って，位置決め精度は得られた測距データの精度だけでなく，測距データに対する演算回数にも影響される。IDA では計算量が従来の手法よりも少なく抑えられていることが上記の結果から分かるため，位置決め計算過程での測距誤差の蓄積も従来の手法よりも低く抑えられていることが予想される。

3.4 IDA の特長

通常の計算幾何の問題では，座標を与えられた点の集合に対して一意に与えられる空間分割である Delaunay 分割を求めることが目的となる。これに対し，本研究で提案するアルゴリズムは互いの距離のみを与えられた点の集合の位置を決定するものであり，graph realization の問題に対するアルゴリズムである。本研究では Delaunay 三角形で構成されるローカルクラスターを用いて位置決めを行うため，特に Inverse Delaunay Algorithm と命名した。

計算量が抑制されること以外の本アルゴリズムの利点として，距離計測の誤差に対するロバスト性の高さが挙げられる。これを端的にあらわす例として，次頁 図-15 に示す状況を考える。四角形 ABCD は対角線を含む 6 辺が計測されている四角形であり，分散型位置決めアルゴリズムで広く最小単位として用いられている。このとき，距離 BD の計測の信頼度が低いと，図-15 の右側に示すように三角形 ACD が反転し，四角形 ABCD の

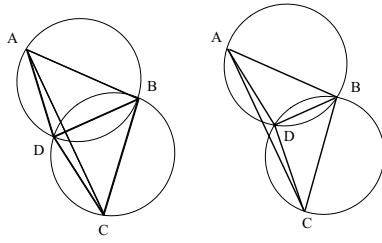


図-15 距離計測誤差の影響

凸性が失われる可能性がある。このような反転の曖昧さは分散型位置決めアルゴリズムの最大の誤差要因であり、四角形 ABCD を最小単位とする位置決めアルゴリズムの多くはこの四角形を位置決め計算に用いない。結果として、測距精度の低下に応じて計算に用いることのできる基本単位の数が増加することで、位置を確定できないノード数が増大し、アルゴリズムが破綻することが多い。

これに対し、図に示すように、同程度の距離 BD の計測誤差に対して、IDA におけるローカルクラスター生成では使用可能な Delaunay 三角形の個数は減少しない。これは反転可能性のある三角形 ACD は外接円の半径が大きいため Delaunay 三角形とはならず、この図では ABD・BCD の組が Delaunay 三角形となっているためである。このように、測距精度に対するロバスト性の低い三角形は Delaunay 性を判定する段階で排除され、そのかわり別な三角形が利用されるため、Delaunay 三角形を基にしたクラスターを利用して位置決めを行う IDA は測距誤差に対してロバストとなっている。

4. センサーノードの位置決め実験

第2節で説明した音響測距機能と第3節で説明した位置決めアルゴリズムを組み合わせることで実際にセンサーノードの位置決めが可能であることを実証する目的で以下のような実験を行った。

まず、音響測距機能を実装したセンサーノード 21 個を平面上に設置した。第2節で述べたとおり各センサーノードの距離が極端に近くなると測距不能になるため、隣接するセンサーノードの間隔を 1 m 以上とした。センサーノード配置の様子を図-16 に示す。次に、近隣のセンサーノードどうしの音響測距を実行させた。具体的には、若い番号のセンサーノードから順番に図-2 における発信ノードの役割を担わせ、発信ノードとその近隣のセンサーノード間の距離を測定した。受信ノードにスタックされたデータは各計測直後処理され、TOA が算出された後 PC に回収された。全てのセンサーノードが発

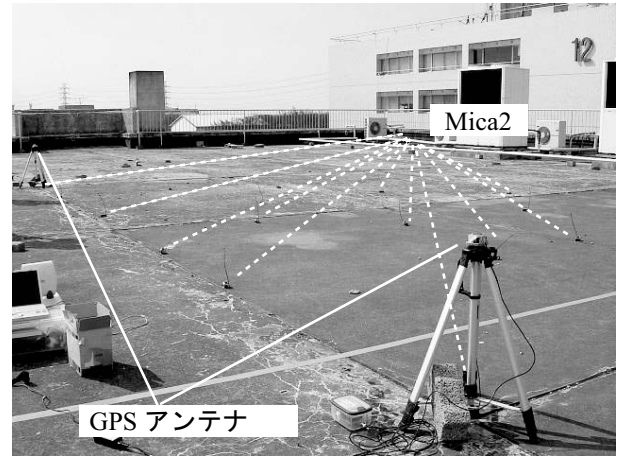


図-16 配置の様子

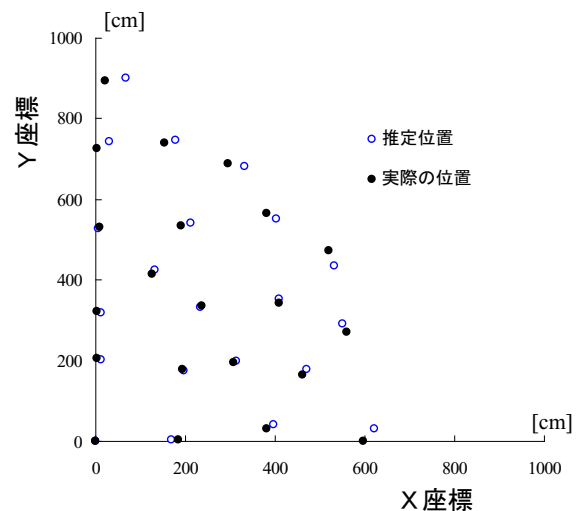


図-17 実際の位置と推定位置の比較

信ノードとしての役割を終えた後、PC で測距データを基に第3節で述べた IDA を用いた位置決め計算を行い、全センサーノードの相対位置を決定した。そして、総体としての回転・並進・反転を $(0, 0) \cdot (600, 0) \cdot (0, 892)$ に設置されたセンサーノードと同じ位置に別途設置した GPS 受信解析装置により得られる絶対位置に従い決定した。センサーノードの実際の配置と IDA による推定位置との比較を図-17 に、位置決め計算をする際に用いたクラスターを次頁図-18 にそれぞれ示す。辺縁部分にあるノードの位置決め誤差が大きくなっているのは、原点のノードの位置を参照点と合わせたためである。音響測距の誤差は平均で $\pm 3.8\text{cm}$ (最大 16.8cm) であり、これはノード間の平均距離 321.6cm に対して 1~5% 程度となっている。このような誤差を含んだ計測距離を基に位置決め計算を行ったにもかかわらず、情報量が少な

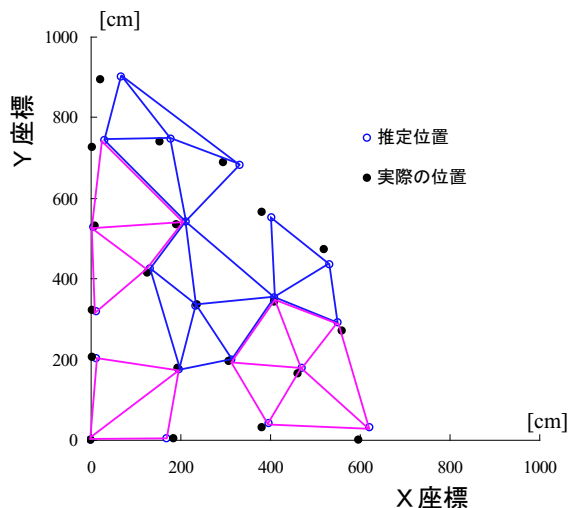


図-18 位置決めに使ったクラスター (赤:AC, 青:BC)

位置関係が全く異なるものになるなど、位置決めに失敗したノードは存在せず、位置決め誤差は妥当な範囲に収まっていることから、手法の測距誤差に対するロバスト性の高さが示唆される。

ただし 1000 個程度以上のセンサーノードを用いた実計測に本手法を適用するためには、以下のような改良・検討が必要であると思われる。

- IDA による位置決め精度の定量的評価：センサーノード数・配置形状・音響測距の精度と位置決め精度との関係を定量的に検討する必要がある。
- 音響測距の高精度化：実験結果から、分散型の位置決め手法である IDA を用いても測距誤差の影響は取り除けていないことが分かる。従って、1000 個程度のセンサーノードの位置決めのためにはさらなる測距精度の向上が必要である。
- IDA の三次元化：本研究では 2 次元配置を仮定した実装を行ったが、実計測ではセンサーノードは 3 次元的に配置されることが多く、手法の三次元化が必要である。
- 音響測距の高速化：実験で用いた方法では、音響測距に掛かる時間はセンサーノード数に比例する。複数箇所と同時に音響測距が行われるような工夫が必要である。

5. 時刻同期機能の実装

センサーノード間の時刻同期をとる手法を音響測距の実装の際に用いた Mica2 に実装した。無線センサーネットワークを対象とした時刻同期手法として、参照信号の受信時刻を合わせることで同期をとる方法 RBS(Reference Broadcast Synchronization)¹²⁾ がある。

名である。RBS では各センサーノードが参照信号を送信するが、本研究では RBS よりも簡略化した方法として、特定のセンサーノードが一定間隔で発信するタイムスタンプを基に計測データの時刻を求める方法を実装した。

5.1 システムの構成

同期計測システムの構成要素を以下に挙げる。

- ファンクション・ジェネレータ
- 「時刻ノード」(Mica2)
- 「計測ノード」(Mica2)
- 「データ受信ノード」(Mica2)

これらの構成要素によって、以下のように同期計測を行う。

ファンクション・ジェネレータから送信されるパルス波の立ち上がりトリガーとして、「時刻ノード」からタイムスタンプが定期的 ($1 \sim 10[\text{Hz}]$ 程度) に発信される。このタイムスタンプには、時刻に対応した通し番号が搭載されており、収集された計測データ間で同期をとる際に、この通し番号が利用される。

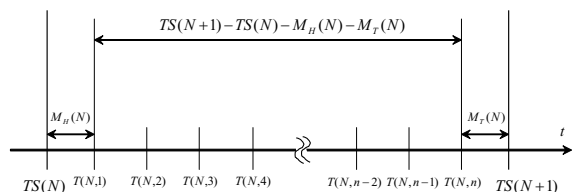
初回のタイムスタンプの受信を計測開始の合図として、各「計測ノード」が計測を開始する。「計測ノード」では、指定されたサンプリングレートで計測が行われ、計測データは、Mica2 に搭載されているメモリに時系列に記録される。

「計測ノード」は、タイムスタンプを受信した時に、図-19 に示すような時刻データと計測データを含むバケット(「計測データバケット」)を無線で発信する。

「データ受信ノード」は、計測データバケットを受信し、サーバに転送する。

具体的に N 番目のバケットに搭載されている主な情報は以下のとおりである。

- センサーノードの ID
- N 番目、 $N+1$ 番目のタイムスタンプ ($TS(N)$, $TS(N+1)$)
- 「前マージン」、「後マージン」(M_H , M_T)
- 計測データ数 (n)
- 計測データ配列



$TS(N)$: N 番目のタイムスタンプ
 $T(N,i)$: i 番目のサンプリング時刻
 $M_H(N)$: $TS(N)$ から $T(N,1)$ までのマージン
 n : $TS(N) \sim TS(N+1)$ の計測データ数
 $M_T(N)$: $T(N,n)$ から $TS(N+1)$ までのマージン

図-19 N 番目のバケット

ここでマージンとは、タイムスタンプの受信時刻と計測データのサンプリング時刻を補正するために計測ノードの内部クロックでカウントされる時間間隔であり、最大でもサンプリング時間間隔以内である。タイムスタンプ・マージン・計測データ数を基に各サンプリング時刻が求められる(図-19)。計測開始からデータ回収までを時系列で表現すると、図-20 のようになる。前述の通り、タイムスタンプの受信が計測データパケットの生成・発信のトリガーとなっている。この計測システムを用いることで、時刻同期のとれた多点計測データをリアルタイムに取得することが可能となる。

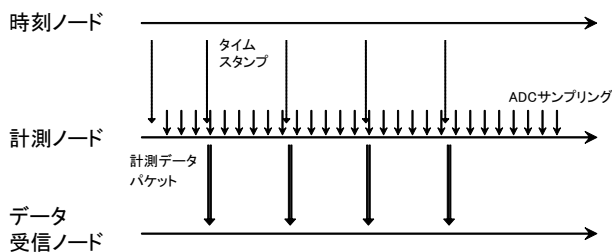


図-20 同期計測の流れ

5.2 同期精度の検証

同期精度を検証する目的で、以下のような検証実験を行った。構築した同期計測システムにおいて、2 個の計測ノードのセンサー入力部に同一の機器から同時にパルス波を入力しつつ、サンプリングレート 500Hz で計測を行った。得られた時系列データから、パルス波の立ち上がり時刻(イベント時刻)を求め、2 ノード間での差をとった(図-21 参照)。

離散的サンプリングを行っている場合、図-21 に示すように、実際のイベント時刻とイベント検出時刻には違

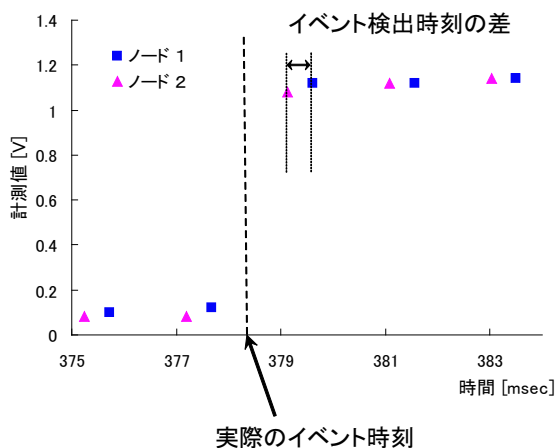


図-21 計測データの一部(イベント時刻付近)

いがあり、むしろ実際のイベント時刻にイベント検出時刻が一致する例はまれである。このような離散的サンプリングの限界により、500Hz でサンプリングを行う本検証実験では原理的には最大 2ms のずれが生じる可能性がある。

約 450 回のイベント回数に関して検出時刻の差を計測し、ヒストグラムを作成したところ、図-22 に示される結果を得た。イベント時刻の差は、全てのイベントに関して 2ms 以下となっており、時刻同期誤差はこの実験で検証できる同期精度の分解能(2ms)以下であることが分かった。

サンプリングレートをさらに高くすると、時刻同期誤差はサンプリングレートに対応するサンプリング時間間隔内に収まることが示される。

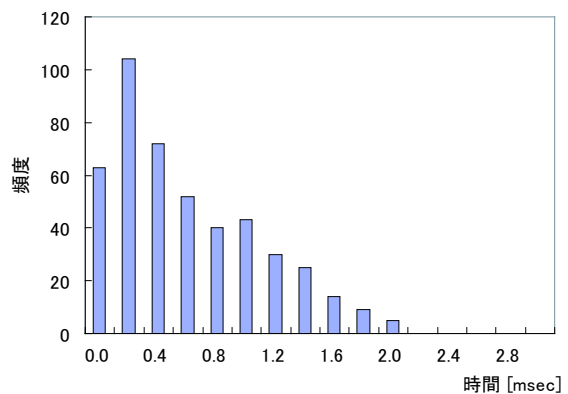


図-22 イベント検出時刻の差

6. まとめ・今後の課題

本研究では、ネットワークセンシング技術を構造工学の問題へ適用する際に生じる Localization Problem を解決するため、安価なデバイスを用いた音響測距と分散型位置決めアルゴリズム(IDA)を組み合わせたシステムの開発を行った。測距試験を行ったところ、5m 以下の距離計測では測距誤差が 6cm 以下に抑えられることが分かった。IDA のセンサーノード数に対するスケラビリティを評価したところ、位置決めに要する計算量はセンサーノード数と同じオーダーに抑えられることが分かった。測距機能を実装した 21 個のセンサーノードを用いた性能検証実験を行ったところ、センサーノードの位置決めが良好に行われた。

センサーノード間の時刻同期をとるため、タイムスタンプを利用した簡易な時刻同期方法を Mica2 に実装した。2 センサーノード間の同期精度を検証したところ、同期誤差がサンプリング時間間隔以下に抑えられるこ

とが分かった．センサーノード数を増加させても，同じ Mica2 から発信されるタイムスタンプを受信できる範囲にセンサーノードを設置すれば，設置した全センサーノード間で，上記と同様な同期精度が得られる．従って，原理的には無線の届く範囲（半径 10m 以内）に展開されたセンサーネットワークに関しては，いくつでも拡張することができる．

本研究により，小規模エリアに展開される数十個程度のセンサーノードによって構成されるセンサーネットワークにおいて，各センサーノードの位置決めから，多点同期計測までを自動的に行わせるために必要な基本技術（測距方法・位置決め方法・同期方法）が開発されたことになる．これらの機能を，ひとつのアプリケーションとしてセンサーノードに実装することにより，センサーノードを適当に設置するだけで，位置情報と結びついた同期計測データを収集できるようになり，安価なセンサーノードを多数個用いて構造物を機動的に計測することが可能になると予想される．

参考文献

- 1) <http://www.microchip.com>
- 2) <http://www.atmel.com>
- 3) <http://www.tinyos.net>
- 4) 許国豪, 井上純哉, 本多弘明, 小国健二 (2005). センサーネットワークの位置決めのための音響測距の実装と分散型アルゴリズムの提案応用力学論文集, Vol.8, pp.653-662.
- 5) Saeki, M., Inoue, J., Khor, K.H., Kousaka, T., Honda, H., Oguni, K. & Hori, M. (2006). Hierarchical localization algorithm based on inverse Delaunay tessellation, *Wireless Sensor Networks*, Proceedings Lecture Notes in Computer Science 3868, SPRINGER-VERLAG BERLIN 180-195.
- 6) Bulusu, N., Heidemann, J. & Estrin, D. (2000), GPS-less low cost outdoor localization for very small devices, *IEEE Personal Communications Magazine*, Vol.7, (5), pp.28-34.
- 7) Ward, A., Jones, A. & Hopper, A. (1997), A New Localization Technique for the Active Office, *IEEE Personal Communications Magazine*, Vol.4, (5), pp.42-47.
- 8) Priyantha, N., Chaakraborty, A. & Balakrishnan, H. (2000), The Cricket Location-Support System, *6th ACM International Conference on Mobile Computing and Networking*.
- 9) Hill, J. & Culler, D. (2002), Mica: A Wireless Platform for Deeply Embedded Networks, *IEEE Micro*, 22(6), pp.12-24.
- 10) 杉原厚吉 (1994), 計算幾何工学, 培風館.
- 11) Moore, D., Leonard, J., Rus, D. & Teller, S. (2004), Robust Distributed Network Localization with Noisy Range Measurements, *Proc. Second ACM SenSys*.
- 12) Elson, J. and Estrin, D. (2002), Fine-Grained Network Time Synchronization using Reference Broadcasts, *Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002)*, Boston, MA. December 2002.

(2006 年 9 月 11 日 受付)