

# 複数センサによる静止・移動ターゲットトラッキング手法の実環境における性能調査及び実環境に耐性のある改善手法の検討

代表研究者 塚 本 和 也 九州工業大学情報工学部助教  
 共同研究者 田 村 瞳 九州ヒューマンメディア創造センター専任主任研究員

## 1. 研究の概要

本研究では実環境において実際のセンサノードを用いたネットワークを構築し、その上でこれまでに提案してきた、複数センサが連携することで移動ターゲットをトラッキングするための手法の実環境での性能調査を行った。まず、Atlas を用いてターゲットを検知するための機構を開発し、その上で複数センサからの検知結果に基づく移動ターゲットの現在位置、及び移動履歴（動線）を示すためのアプリケーション開発を行った。これに加え、実環境において Mote センサを用いた際の省電力化を実現するための手法についても検討をすすめ、センシング状態をフィードバックすることで省電力化を実現できることを明らかにした。

また、複数センサによる移動ターゲットトラッキング手法のさらなる改善を行い、その有効性をシミュレーション実験及び近似式導出による解析的アプローチによって評価した。その結果、シミュレーション実験から手法の有効性を検証でき、近似式から提案手法を用いた場合の性能を要求トラッキング確率、配置密度を元に推測出来ることを明らかにした。

これらの結果をまとめたものを、(1)電子情報通信学会 インターネットアーキテクチャ (IA) 研究会において報告し、2007 年度 IA 研究賞を受賞した。また、(2)国際会議 SUPE'07 においても発表を行った。

## 2. 実環境における複数センサによる移動ターゲット捕捉可視化ツールの開発

本節では、Atlas センサノードを利用して開発した、移動ターゲット捕捉センサネットワーク、および、移動ターゲットの位置を推定する可視化アプリケーションの概要を紹介する。

### 2.1. 複数センサノードにおけるターゲット捕捉

#### 2.1.1. センサノードの仕様

本研究で使用するセンサノードは、ヒト感知センサモジュール(株式会社ヘルツ製 HPB-006) とセンサプラットフォーム(Atlas) から構成される。

- センサモジュール: 常に状況を監視し、ターゲット捕捉時は捕捉データを、非捕捉時には非捕捉データを 0.1 秒間隔でセンサプラットフォームに送信する。このセンサモジュールは、一般的に人を検知して照明を点灯制御するような場合に利用されるため、5 秒～3 分のタイマ(実験では 5 秒とした)が設定されており、検知→非検知となった場合でもタイマ時間内は捕捉データを送信し続ける。
- センサプラットフォーム: センサモジュールが取得したデータにノード ID を付加し、データをパケット化して無線通信(IEEE 802.11b)でホストコンピュータにターゲット捕捉情報を転送する。

6 台のセンサノード、および、これらの情報を収集する 1 台のホストコンピュータによってターゲット捕捉センサネットワークを構築する。センサノードは、高さ 16cm の箱の上に設置する。したがって、身長 170cm の人をターゲットとした際のセンサノードの検知範囲は図 2-1 のようになる。

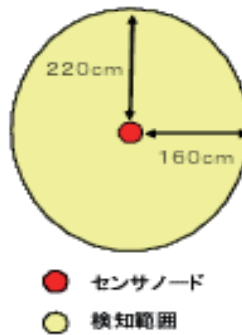


図2-1: センサノードの検知範囲

### 2.1.2. ターゲットの位置特定のための要求条件

前節で述べたように、センサモジュールにはタイマ機能があるため、捕捉情報をそのまま反映すると、実際と異なる位置情報を得る可能性がある。そこで、この性質を考慮して捕捉状態の補正を行い、かつ、複数センサノードの捕捉データを解析して、ターゲットの位置や移動状況を示すことが必要となる。

## 2.2. 捕捉センサネットワークシステム

実験エリアとして、ある地点において、ターゲットを最大4つのセンサノードで捕捉できるように室内に6個のセンサノードを格子状に配置し、各センサノードが無線によってホストコンピュータと通信を行う捕捉センサネットワークシステムを構築した。その構成図を図2-2に示す。Atlasが提供するアプリケーションでは、個別のセンサノードからの情報の確認しかできないため、複数センサノードからの捕捉データを解析して、ターゲットの移動状況を可視化するアプリケーションをホストコンピュータ上に実装した。

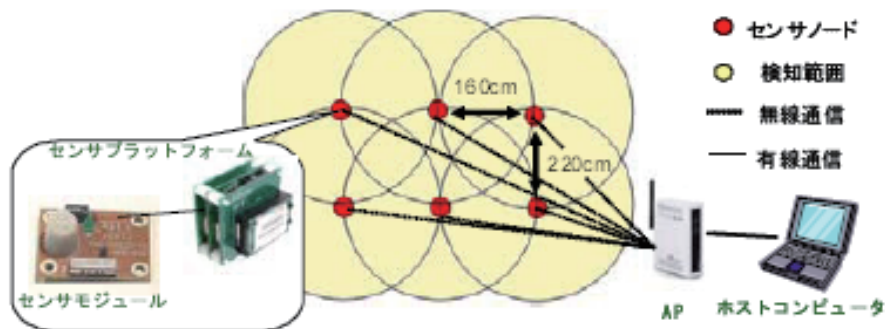


図2-2: センサネットワークの構成図

センサノードがターゲットを捕捉すると、その情報を図2-3のようにGUI (Graphical User Interface) に表示する。このアプリケーションでは、ターゲットの捕捉情報を送信したセンサノードの検知範囲が重なる部分をターゲットの存在位置とみなして推定結果を出力する。つまり、捕捉センサノードの共通重心点がターゲットの推定位置となる (3つのセンサノードで捕捉した場合を図2-3に示す)。センサモジュールのタイマ機能によって、例えば図2-3において、ノード1、2、3、4が捕捉データを送信している際、位置推定の補正として以下の処理を行う。

- 捕捉状態であるセンサノードの捕捉開始時間を解析。
- 捕捉開始時間の最も遅い3つのセンサノードが実際に捕捉していると推定。
- 推定センサノード間の重心推定によりターゲットの位置を表示。

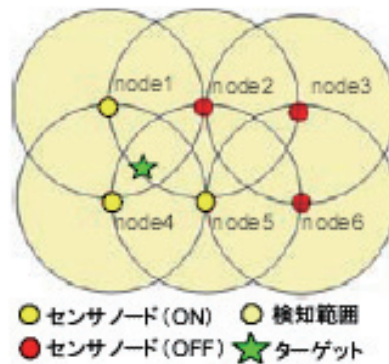


図2-3: 3センサで捕捉した例

### 2.3. 性能評価

図 2-4 に実際のターゲットの移動の軌跡を示し、その際にアプリケーションで得られた移動状況を図 2-5 に示す。ターゲットの移動速度は約 30cm/s である。図 2-4、2-5 を比較すると、可視化ツールにより、ターゲットの移動に沿って存在位置を表示できているようすがわかる。なお、図 2-5 では中央に位置するノード 2、5 で捕捉した場合に位置が表示されていないが、これは、センサモジュールのタイマ機能とターゲットの移動速度の関係に起因する。すなわち、実際にはターゲットがノード 2 と 5 の中間点にいる場合でも移動速度が約 30cm/s と非常に遅く、かつ、5 秒間のタイマ設定により、ターゲットがノード 3、6 の検知範囲から出て中間点に到達した際にもノード 3、6 が捕捉データ送信を続けているため、ノード 2、3、5、6 の 4 点捕捉を行う。また、中間点到達後、ノード 3、6 が捕捉データ送信を続けている間にノード 4 がターゲットを捕捉するため、次にノード 2、3、4、5、6 の 5 点捕捉を行うこととなる。このように実際のターゲットの存在位置を正確に表示することは困難であるが、タイマ機能の補正処理により移動傾向については表示可能であることが確認された。

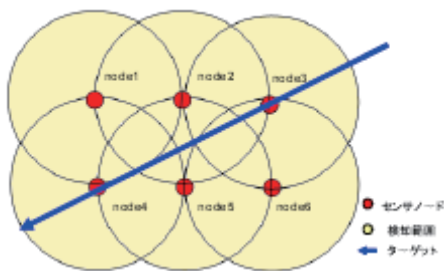


図2-4: 実際のターゲットの軌跡

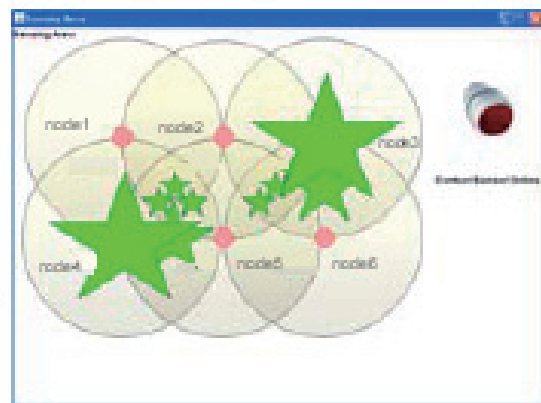


図2-5: アプリケーションにより表示された移動状況

### 3. Mote センサによる省電力化性能についての調査

センサネットワークを構築する上で電池駆動型のセンサノードを用いる場合、長期間にわたりセンサネットワークの性能を維持するためには、周囲の状況に応じたノードの省電力制御が必要不可欠である。しかし、現在最も普及している Crossbow 社のセンサノード「MICA MOTE」(以下 MOTE) には状況に応じた自動で制御を行う機能は実装されておらず、状況に応じた省電力制御の実装が困難である。そこで、MOTE を用いたセンサネットワークを構築し基本性能評価を行った上で、省電力化を目的としたセンサノードの動作制御を自動で行う動的フィードバック制御システムを開発する。

### 3.1. MICA MOTE

MOTE は米カリフォルニア大学バークレー校が発端となり、国際フォーラムで開発されているシステムである。2.4GHz 帯 IEEE 802.15.4 の通信規格の無線デバイスに任意のセンサ基盤を搭載することでセンサネットワークノードとしての運用が可能となる。また、XMesh と呼ばれるプロトコルによりメッシュネットワークの構築を行い、各センサノードは基地局ノードまでの通信経路を確立する。XMesh におけるセンサノードの設定状態には表 3-1 に示すように高電力動作を行う XMesh-HP (HighPower)、および、低電力動作を行う XMesh-LP (Low Power) がある。基地局ノードに集約されたセンサデータはシリアル通信により接続されている PC 上で稼動するサーバシステム XServe へ送られる。XServe は XCommand と呼ばれるコマンドアプリケーションを提供しており、ユーザはこの XCommand を用いて PC 上からセンサノードに対する動作制御を行うことができる。

表3-1: センサノードの状態

パラメータ	XMesh-HP	XMesh-LP
無線待機	常に電源 ON	1 秒に 8 回の励起
ルート更新間隔	36sec.	360sec.
平均消費電流	20-30mA	400 $\mu$ A

### 3.2. 基本性能評価

省電力制御の可能性を確認するために、XMesh-LP においてサンプリング間隔による電池電圧降下量への影響を観測し、その結果を図 3-1 に示す。ここで、サンプリング間隔とはノードからのセンサデータ転送間隔である。図 3-1 よりサンプリング間隔が 1 時間、5 分、1 秒と短くなるほど電池電圧降下が著しいことがわかる。すなわち、センサデータ転送がセンサノードの全電力消費量に与える影響が大きいいため、状況に応じてセンサデータの転送量を制御することによって省電力化が可能であることがわかる。

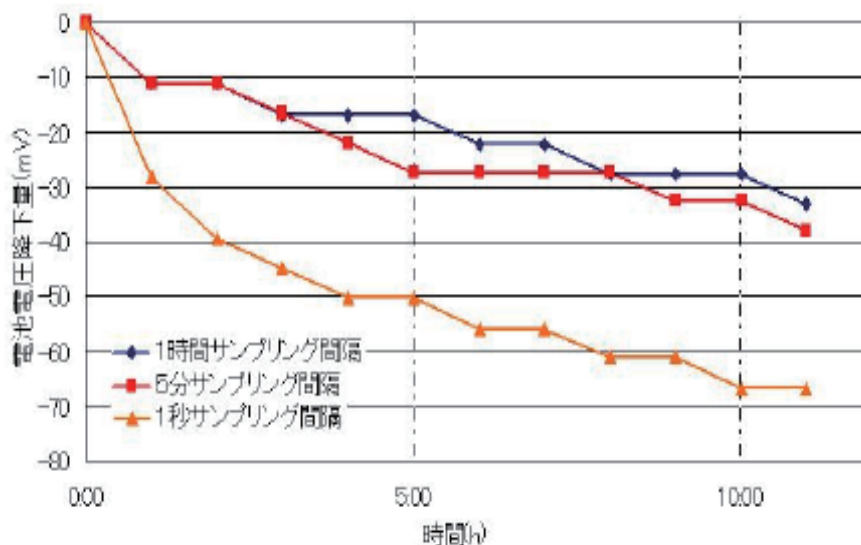


図3-1: サンプリング間隔が電池電圧降下量に及ぼす影響

### 3.3. フィードバックシステム開発

MOTE の取得データを管理する XServe は、個別ノードからのデータパケットを収集し、データベース化するのみで、ユーザはそのデータベースの情報を基に XCommand により各センサノードの設定を手動で行う必要がある。そこで、センサデータの解析とそれに基づく制御情報の作成、制御命令の送信を自動的に行うフィードバックシステムを perl により構築し、図 2 に示すように XServe が稼動する PC 上に実装した。

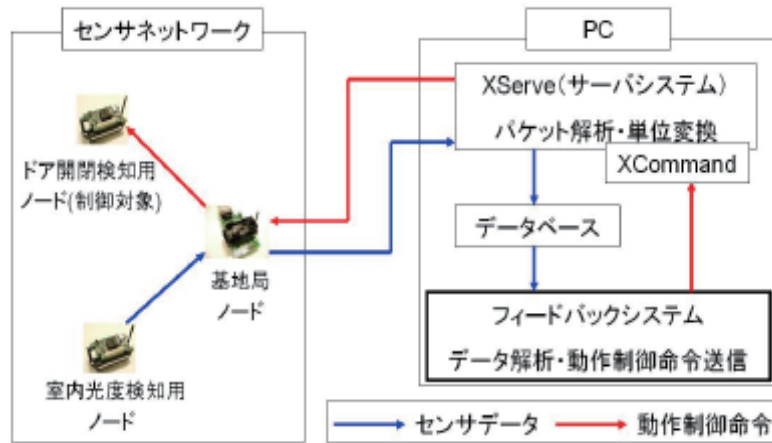


図3-2: システム性能評価調査概要

### 3.4. フィードバックシステムに関する性能評価

開発したフィードバックシステムの動作確認と、センサノードの省電力化を実現するために図 3-2 に示すセンサネットワークを室内に構築した。このネットワークには 2 つのセンサノードが存在し、1 つは室内の光度検知、もう 1 つはドアに設置され加速度センサのデータに基づく開閉検知を行う。一般的に室内の光度が低い時は人が存在せず、出入りも少ない深夜/早朝の時間帯と判断されるため、ドア開閉の頻度は低くなることが予想される。したがって室内光度検知ノードの情報を解析し、ドア開閉検知ノードのサンプリング間隔を調整することで自動的な省電力制御の実現が期待される。なおネットワーク構築におけるセンサノードの状態は XMesh-LP とし、ドア開閉検知時のサンプリング間隔は 1 秒、省電力動作時のサンプリング間隔は 1 時間と設定した。2008 年 1 月 28 日から 29 日の 20 時間にわたって調査を行った際のドア開閉検知ノードの電池電圧降下量を図 3-3 に示す。室内光度検知ノードが午前 2 時 30 分 14 秒から午前 7 時 3 分 41 秒の間で部屋の電気の消灯を検知し、その検知情報によってフィードバックシステムがドア開閉検知ノードへ動作変更命令を送信していることをログファイルから確認している。すなわち、20 時間の稼働の中でドア開閉検知ノードは 4 時間 33 分の間、省電力動作で稼働したことが示されており、電池電圧降下量は省電力制御をしない場合と比べて 11.3 mV 少なくなっている。初期電圧を 3 V、稼働限界電圧を MOTE の仕様である 2.1 V、調査日の消灯・点灯時間の傾向が他日でも同様であると仮定すると、省電力制御をする場合は 5.199 日、省電力制御をしない場合は 4.881 日の連続稼働が可能である。したがって、フィードバックシステムを用いて省電力制御をすることで 7 時間 38 分程度の長期稼働が可能となる。この結果から、開発したフィードバックシステムによる省電力制御が有効であることがわかった。

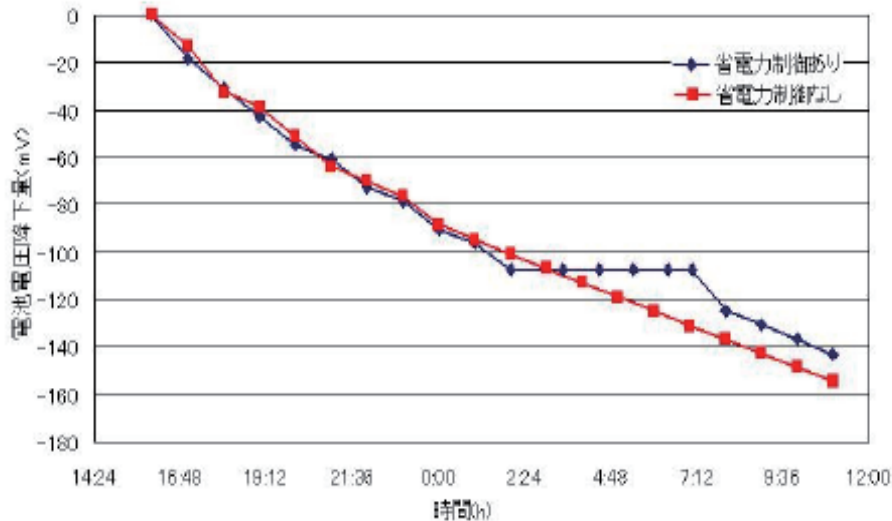


図3-3: ドア開閉検知ノードの電池電圧特性

#### 4. ターゲットの移動性を考慮した複数センサによるトラッキングアルゴリズムのトラッキング特 性と消費電力特性の調査 [1] [2]

センサネットワークでは、各センサが隣接センサと通信を行い協調動作することで、多様なサービスを提供可能で、本節では様々なサービスの内、人や移動体のトラッキングサービスに着目する。センサネットワークは、森林観測や人流観測等、様々な環境で動作することが望まれるため、各センサは小型かつ低コストでなければならない。その結果、電源、メモリ、プロセッサの性能に加え、センサのセンシング能力・通信能力も大きく制限されることになる。よって、個々のセンサが取得可能な情報量は少なく、詳細な情報を取得できない。そこで近年、1つの監視対象オブジェクトを複数のセンサで同時に捕捉し、取得した情報を統合することによって、ターゲットの存在位置や進行方向などの詳細な情報を取得するセンサフュージョン技術が注目されている[3]。一方、このようにセンサネットワークは大量のセンサによって構築されるため、スケーラビリティの観点から各センサが自律的に隣接センサと情報を交換し、動作を決定する必要がある。以上のことから、高精度なターゲットトラッキングの実現には、(1) 近隣センサとの情報交換による自律的な状態決定、(2) 複数センサによるトラッキングアルゴリズムが重要になる。

これまでに我々は、自律的に複数のセンサでターゲットを監視する手法(基本手法)を提案し、ターゲットの移動速度に応じて性能が急激に劣化する問題を明らかにした [4]。これは、基本手法がターゲットの移動に素早く対応できず、ターゲットを監視するセンサの引き継ぎがうまくできないことに起因する。そこで先行研究 [5] では、上記の問題を踏まえた上で基本手法を改良し、ターゲットの移動性を考慮した上で状態を決定する改善手法を提案した。シミュレーション結果から改善手法はターゲットが高速に移動する際においても高いトラッキング性能を達成できることを示した。しかし、改善手法の消費電力、及び要求捕捉センサ数が変化した際の性能については調査していない。そこで本節では、改善手法の省電力性能について調査するとともに、要求捕捉センサ数の変化に対する改善手法の適応度合いについても調査を行う。

##### 4.1. 複数センサによる協調ターゲットトラッキング手法

本節では、想定するセンサの仕様を説明し、複数センサの協調動作による移動ターゲットトラッキング手法[5]の概要について述べる。

###### 4.1.1. センサの仕様

センサは図4-1に示すようにセンシング機能、コミュニケーション機能を有する。各機能について簡単に述べる。

- センシング機能：半径 $R_s$  の円内のセンシング範囲においてターゲットの存在の有無のみを検知可能
- コミュニケーション機能：半径 $R_c = 2R_s$  の円内のコミュニケーション範囲においてブロードキャスト通信のみが可能

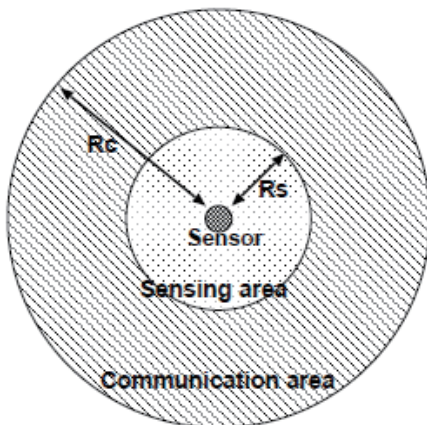


図 4-1：センサの能力

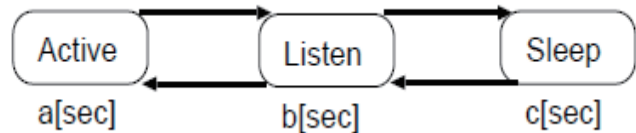


図 4-2：状態遷移モデル

また、全センサは省電力化のために、図4-2に示すようにActive、Listen、Sleepの3状態を一定期間ごとに順に遷移するものとする。ここで、Active状態のセンサは期間中、常にターゲットのセンシング、及び他のセンサとのメッセージの送受信が可能とし、Listen状態では他のセンサからのメッセージの受信のみ

が可能な状態とする。また、Sleep 状態は通信、センシング共に不可能な省電力状態とする。本研究では、各状態滞在期間をそれぞれ a、b、c [sec] と定義する。

#### 4.1.2. トラッキング手法

本節では、前節で述べたセンサによって構成されるセンサネットワークにおける複数センサによる協調ターゲットトラッキング手法[5]について述べる。この手法は、ターゲットの検知を伝えるメッセージ(ALERT、DETECT)に基づき、各センサが自身の状態を決定することで自律的にシステム/ユーザから指定された数(M 個)の捕捉センサを決定可能なこと、各センサの状態遷移サイクルが同期しないことが特徴である。この手法における捕捉センサ決定までの流れについて、図4-3 のフローチャートを用いて説明する。

##### 1. ALERT を受信したActive 状態のセンサ

Active 状態のセンサは、ターゲットを検知可能かどうかによって自身の状態を次のように決定する。

1. I. 検知不可：ALERT メッセージを受信しているため、受信センサは周囲にターゲットが存在すると推測し、Active 状態の残り時間に加えてある一定期間、状態滞在期間を延長しActive 状態を維持する。

1. II. 検知可能：ランダム待ち時間を計算し、ランダム待ち時間経過時点までに受信したALERT 及び DETECT メッセージの合計数によってさらに以下の動作を行う。

1. II. (a) M メッセージ未満：DETECT メッセージを送信し、ターゲットの監視を行う。

1. II. (b) M メッセージ以上：DETECT メッセージを送信せずListen 状態に遷移する。

##### 2. ALERT を受信したListen 状態のセンサ

Active 状態のセンサと異なり即座にランダム待ち時間を決定する。ここで、Listen 状態のセンサのランダム待ち時間はActiveセンサのランダム待ち時間よりも必ず大きくなるように設定する。つまりListen 状態のセンサはActive 状態のセンサの動作が決定した後、ランダム待ち時間経過後の受信メッセージ数に基づき以下の動作決定を行う。

2. I. M メッセージ未満：監視センサが不足していると判断してActive 状態へと遷移し、ターゲット検知可能かどうかによって以下のように自身の状態を決定する。

2. I. (a) 検知不可：Listen 状態へ遷移する。

2. I. (b) 検知可能：DETECT メッセージを直ちに送信しターゲットの監視を行う。

2. II. M メッセージ以上：そのままListen 状態を継続する。

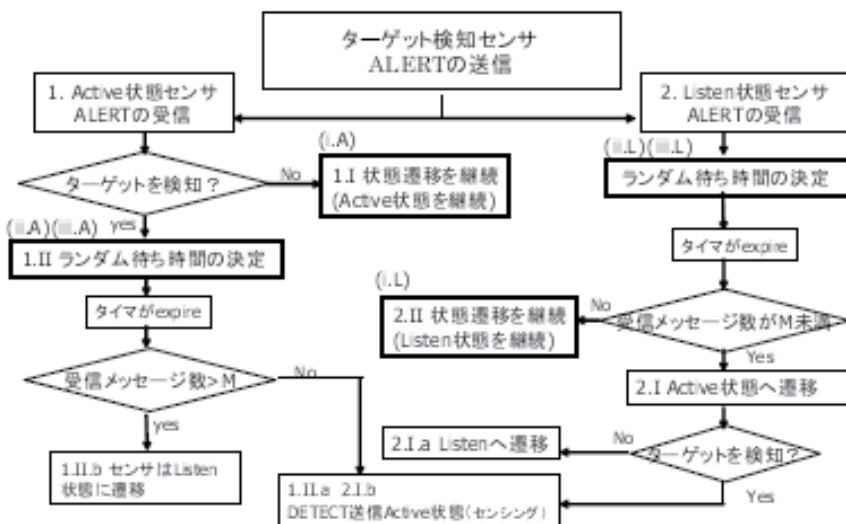


図4-3: ターゲット捕捉アルゴリズム

#### 4.2. シミュレーションモデル

本節では、提案手法を評価するために用いたシミュレーション環境について述べる。シミュレーションパラメータを表4-1 に示す。シミュレーションは、図4-4 に示すように、6000[m] 四方のエリア全体にランダムに100 万個のセンサを配置するものとし、ターゲットは点線上を直進するものとする。ここで、シミュレーション開始時にはActive、Listen、Sleep センサ数が同数存在するように配置する。

表4-1: シミュレーションパラメータ

Simulation time	600 [sec] (10 [min])
Simulation area	6000 [m] × 6000[m]
The number of sensors	N = 1,000,000
Required number of sensors	M = 3
Sensing Range	$R_s = 10$ [m]
Communication Range	$R_c = 20$ [m]
State Transition	Active(a):Listen(2b):Sleep(c) = 1:1:1 (a = 4, b = 2, c = 4 [sec])
Speed of target	0~10 [m/sec]
Message length	10 [bytes]
Slot Time	400 [μsec]
Maximum Slot	ActMax = 30, LisMax = 128
Bandwidth	250 [kb/s]

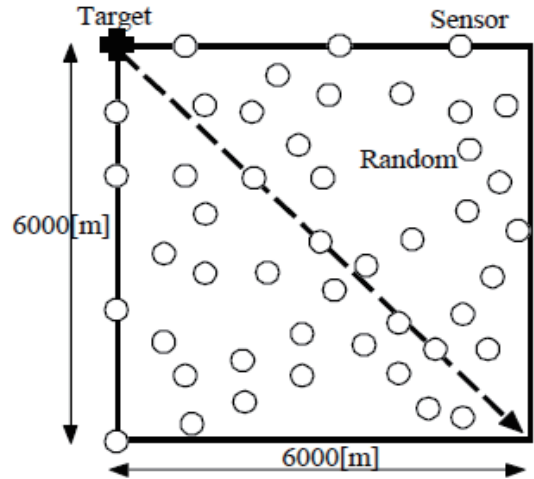


図4-4: シミュレーションモデル

また、本研究では提案手法の評価指標としてトラッキング性能と消費電力を用いる。トラッキング性能は、トラッキング確率 $P_t$ と引き継ぎ効率 $I_h$ によって評価する。消費電力については、状態維持電力 $E_s$ とメッセージ交換電力 $E_c$ を用いて評価を行う。それぞれの算出方法を下記に示す。

$$P_t = \frac{M \text{ 個のセンサで捕捉時間の合計}}{\text{シミュレーション時間}}, \quad (4-1)$$

$$I_h = \frac{M \text{ 個での平均センシング時間}}{\text{平均 ALERT 発生間隔時間}}, \quad (4-2)$$

$$E_s = \sum_{i \in \text{active, listen}} e_i \times T_i, \quad (4-3)$$

$$E_c = \frac{\sum_{j \in \text{tx, rx}} e_j \times C_j}{P_t}. \quad (4-4)$$

平均ALERT発生間隔とM個のセンサによる平均センシング時間の割合である $I_h$ はトラッキング確率に影響を与える指標である。例えばM個のセンサの共通センシングエリアからターゲットが抜けだした後に、ALERTメッセージが発生する場合は $I_h$ は1以下になり、トラッキング確率は減少する。1以上の場合は、センサによるターゲット捕捉中に次のALERTが発生し、ターゲットの捕捉を引き継ぐことが可能である。

一方、 $E_s$ はシミュレーションエリア内においてALERTメッセージを受信したActive、Listenセンサが捕捉センサを決定するまでに状態維持のために消費する電力の合計である。 $e_i$ は1秒間に消費する状態維持電力、 $T_i$ はシミュレーション期間中にトラッキングセンサ決定までに要した時間の総和である。 $E_c$ は、メッセージを受信したセンサ1台がトラッキング確率1%を達成するために必要なメッセージ送受信電力を示す。 $e_j$ はメッセージ送受信1回当たりの電力、 $C_j$ はシミュレーション期間中のメッセージの送受信回数の総和である。

### 4.3. シミュレーション結果

本節では、シミュレーション結果を示す。まず、要求捕捉センサ数Mを3とした場合について、基本手法と改善手法のトラッキング性能について示した上でその改善理由を明確にする。その後、消費電力性能について比較を行い、改善手法の有効性について検証する。最後に、要求捕捉センサ数Mに対する改善手法の適応性能について調査する

#### 4.3.1. トラッキング性能の比較

まず、図4-5に基本手法(図中“Primitive”)、改善手法(図中“Sophisticated”)のターゲットの移動速度の変化に対するトラッキング確率を示す。図より、基本手法はターゲットの移動速度の増加に伴い、トラッキング確率が大きく減少していることがわかる。これは、基本手法におけるセンサの状態遷移の同期問題に



起因する。一方、改善手法ではターゲットの移動を考慮した改善を加えているため、ターゲットの移動速度に関わらず50 %以上の高いトラッキング確率を維持可能であることがわかる。

次に、引き継ぎ効率 $I_h$ を基に状態遷移の同期が改善されメッセージが効率的に発生しているかどうかを確認する。ターゲットの移動速度を変化させた場合の $I_h$ の変化を図4-6に示す。図より、基本手法はセンサの状態同期により、移動速度の増加に伴い $I_h$ が大きく減少し、秒速 2.5 [m]地点で1以下に減少している。つまり、メッセージが効率的に発生できていないといえる。一方、改善手法では、移動速度が増加しても $I_h$ は1以上を維持する。つまり、センサの状態同期問題が改善され、効率的にメッセージが発生していることがわかる。

以上の結果より、改善手法ではセンサの状態同期の問題が改善されているため、それに伴いターゲット捕捉の引き継ぎ効率が向上し、高速で移動するターゲットに対しても高いトラッキング確率を達成可能であることがわかる。

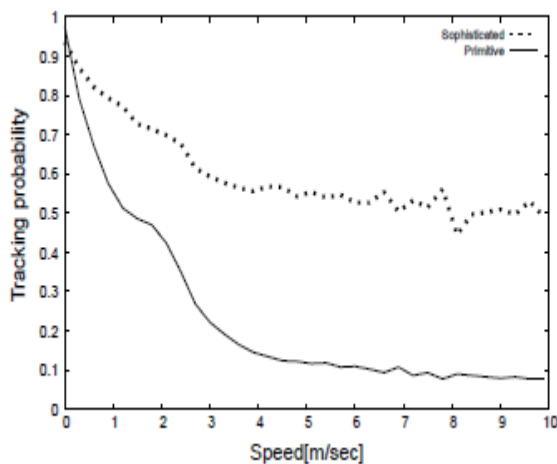


図4-5: トラッキング確率の比較

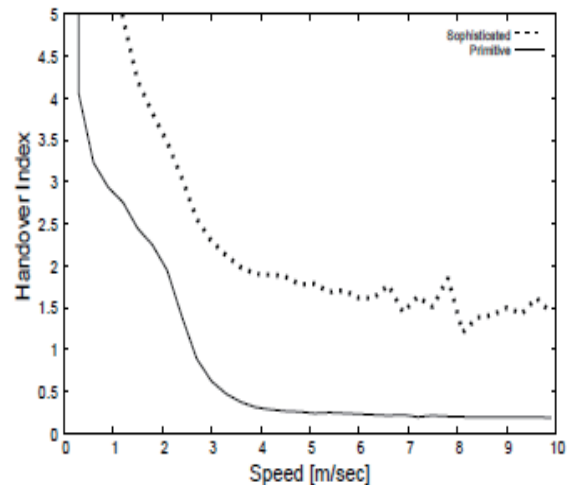


図4-6: 引き継ぎ効率の比較

#### 4.3.2. 消費電力の比較

本節では、基本手法と改善手法の消費電力について述べる。消費電力量を比較するために、センサの状態維持とメッセージ交換に要する電力を調査する。ここで、それぞれの消費電力は式(4-3)、(4-4)を用いて算出し、消費電力量 $e_i$ 、 $e_j$ には、実際のセンサ製品の消費電力である表4-2 [6]、[7]の値を用いる。

表4-2: センサの消費電力モデル

State		Message exchange	
Active	30 mW/s	Transmit	56.7mW/mesg.
Listen	21.6 mW/s	Receive	62.91mW/mesg.

##### 4.3.2.1. 状態維持による消費電力

図4-7 にターゲットの移動速度を変化させた場合の基本手法、改善手法の状態維持に関わる消費電力量について示す。基本手法ではターゲットの移動速度の増加に伴い、消費電力は増加している。基本手法ではトラッキングセンサの決定時に、DETECTメッセージの衝突を回避するために0 ~ 1[sec]の待ち時間を設けている。そのため、トラッキングセンサの決定完了までの時間が改善手法に比べて長くなり、センサの状態維持に要する消費電力量は大きくなる。さらに、ターゲットの移動速度が増加すると、センサが頻繁にターゲットを検知するようになるためALERTメッセージ数が増加し、それに伴いセンサがActive状態に滞在する期間が増加するため、結果的に消費電力量が増加することになる。

一方、改善手法では状態遷移による消費電力量は極めて低い値を示している。これは改善手法では、メッセージ衝突回避時間をメッセージ伝送遅延に基づき計算し、メッセージ送信制御を行っているため、基本手法に比べて捕捉センサ決定までの時間が非常に短くなっているためである。その結果、改善手法は基本手法に比べセンサの状態維持に関わる電力をターゲットの移動速度に依らず大きく削減可能であることがわかる。

##### 4.3.2.2. メッセージ交換による消費電力

図4-8 に基本手法、改善手法のターゲットの移動速度の変化に対するメッセージ交換による消費電力を示す。図より基本手法は改善手法より消費電力が小さい値を示すが、ターゲットの移動速度の増加と共に増加する傾向にあることがわかる。基本手法ではALERTメッセージを受信した際、ターゲットを検知できないセ

ンサはListen 状態へと遷移する。そのため、図4-6 の $I_h$  が低い値となっていることからわかるように、効率的にALERT メッセージを発生させることができず、消費電力量は小さくなる。

一方、改善手法ではALERT メッセージを受信したすべてのセンサのうち、ターゲットを検知できないActive 状態のセンサは状態を継続する。このため、ターゲットの周囲にActiveセンサが数多く存在しターゲットの移動を早期検知でき、図4-6 の $I_h$  が示すようにALERT メッセージが基本手法に比べ多く発生するため消費電力量が大きくなっている。しかし、ターゲットの移動速度が高速になると、改善手法の消費電力は基本手法とほぼ同じ値を示すことから、改善手法は高速移動ターゲットに対して良好な消費電力性能を示すことがわかる。

以上の結果から、改善手法の消費電力は基本手法に比べ、高速移動ターゲットに対しては状態維持電力を大きく削減できる一方で、メッセージ交換に要する電力消費量も同程度の性能を達成することを示した。

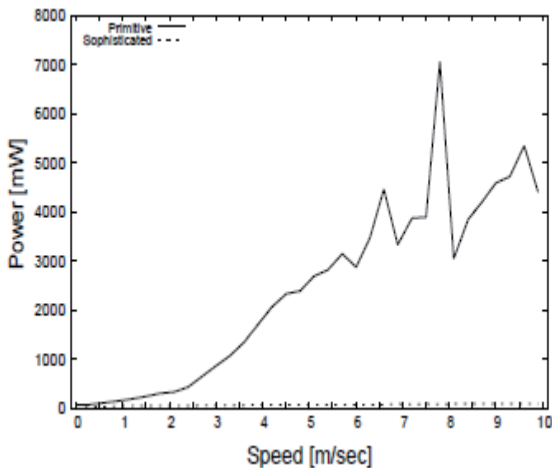


図4-7:状態維持に関わる消費電力量

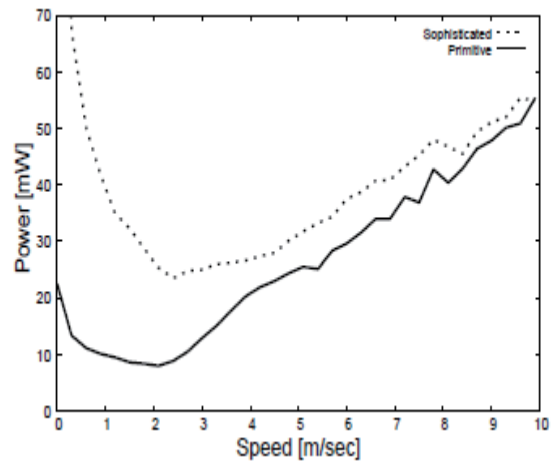


図4-8:メッセージ交換による消費電力量

## 5. 複数センサによる移動ターゲット捕捉のためのセンサネットワーク設計指針の検討～ターゲット

### ト捕捉確率とセンサ配置密度の関係調査～ [8]

ターゲット捕捉センサネットワークにおいて、トラッキング性能や消費電力性能は、センサの配置密度や状態遷移時間、要求ターゲット捕捉センサ数に依存する。一方、実際にサービス提供のためにセンサネットワークを構築する場合、アプリケーション毎に要求が異なるトラッキング性能や消費電力性能に応じて各センサの初期設定や配置個数を検討することが必要になる。そこで、本節ではシミュレーション実験を基に、まずトラッキング性能とセンサの配置密度の関係を調査し、さらにターゲット捕捉センサ数による影響も明らかにする。その後、これらの関係から、トラッキング性能の予測を容易に可能にするための近似式を導出する。最後に、この近似式を用いることで複数センサによる移動体トラッキングのためのセンサネットワークの設計指針について検討する。

#### 5.1. シミュレーションモデル

本節では、提案手法におけるトラッキング確率とセンサ配置密度、及び捕捉センサ数との関係を調査するために用いたシミュレーションモデルについて述べる。主要なシミュレーションパラメータは、特に指定のない限り、表 4-1 と同様である。図 4-4 に示すシミュレーションエリアに、総数  $N_{all}$  (=50 万個(密度: 1.4 個/100m<sup>2</sup>), 100 万個(2.8 個/100m<sup>2</sup>), 200 万個(5.6 個/100m<sup>2</sup>))のセンサをランダムに配置し、センサ配置密度の影響を調査する。センサの状態滞在時間の変化は、センサの配置密度とトラッキング性能の关系到影響するが、ここでは、状態滞在時間は固定とし、一往復した場合のそれぞれの状態滞在時間が等しくなるように、Active、Listen、Sleep の初期設定値をそれぞれ 4 秒、2 秒、4 秒とする。この状態滞在時間を用いた際の、ターゲット捕捉に関係する(提案アルゴリズムに関わる)Active、Listen 状態のセンサの配置密度  $N_{AL}$  を表 5-1 に示す。ここで、シミュレーション開始時には Active、Listen、Sleep センサ数がほぼ同数存在するように配置する。ターゲットは、図 4-4 のシミュレーションエリアの点線上を等速(9.7[m/sec])で移動するものとする。また、評価指標として 4 節と同様にトラッキング確率  $P_t$  を用いる。

表 5-1: 配置センサ数と Active、Listen センサ数

The num. of sensors, $N_{all}$ (Density)	$N_{all} = 2,000,000(5.6/100m^2)$
The num. of Active and Listen sensors, $N_{AL}$ (Density)	$N_{AL} = 1,333,333(3.70/100m^2),$ $N_{all} = 1,000,000(2.8/100m^2)$ $N_{AL} = 666,666(1.85/100m^2),$ $N_{all} = 500,000(1.4/100m^2)$ $N_{AL} = 333,333(0.93/100m^2)$

## 5.2. シミュレーション結果と設計指針の検討

本節では、まず、ユーザ/アプリケーションの要求捕捉センサ数  $M$  が与えられている場合、任意のトラッキング確率を満足するために必要なセンサ配置密度を見積もるために、シミュレーション結果を基にして、トラッキング確率とセンサ配置密度の関係式を導出する。その後、捕捉センサ数  $M$  による影響についても調査し、要求された捕捉センサ数/トラッキング性能を達成するためのセンサ配置密度を推定するための関係式を示すことで、センサネットワークの設計指針を明らかにする。

### 5.2.1. センサ配置密度とトラッキング確率の関係

本節では、要求されるトラッキング確率を満たすためのセンサ配置密度について調査する。ここで、センサ配置密度は(1)ターゲットを捕捉可能な Active 状態のセンサと(2)メッセージを受信可能な Listen 状態のセンサの合計とし、これらのセンサ合計個数で関係の検討を行う。また、ターゲットの検知、メッセージの送受信が不可能な Sleep 状態のセンサは考慮しないものとする。

#### 5.2.1.1. シミュレーション結果

図 5-1 に要求捕捉センサ数  $M = 2 \sim 5$  の場合のセンサの配置密度に対するトラッキング確率を示す。図より、センサ配置密度の増加に伴い、トラッキング確率は急激に増加し、配置密度がさらに増加すると収束して最大になり、それ以上はトラッキング確率に影響を与えていないことがわかる。これは、センサの配置密度が少ない場合は、ターゲットを検知可能なセンサ数が要求捕捉センサ数  $M$  を満足できない場合が多いが、配置密度の増加に伴い、ターゲットの移動を容易に検知することが可能になり、トラッキング確率が急激に増加するためである。一方で、要求センサ数  $M$  の変化に着目すると、 $M$  の増加に従ってトラッキング確率の最大値は低下することがわかる。最大限のトラッキング確率を達成するためのセンサ配置密度について検討する場合、図 5-1 の収束値を達成するセンサ配置密度であれば要求を満足できることになるが、センサ数が過剰な場合、メッセージの衝突によりトラッキング確率が逆に低下する恐れがある。また、コスト面から考えても、センサの配置密度は可能な限り少ない方が望ましい。よって、トラッキング確率が収束し始める最小のセンサの配置密度に着目すると、要求センサ数  $M$  に対して約 6 倍のセンサ個数が必要であることがわかる。このように、図 5-1 からアプリケーション要求を達成するために必要なセンサ数の見積りが可能となり、要求に応じた配置センサ数を明らかにすることができる。

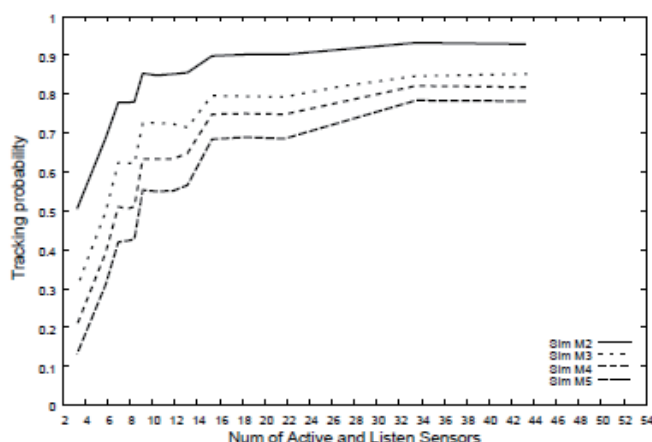


図 5-1: センサの配置密度とトラッキング確率の関係

### 5.2.1.2. 近似式の導出

次に、本節では、図 5-1 の関係を表現する近似式を導出する。式(5-1) を基に図 5-1 のシミュレーションデータを用いて非線形回帰を行い、係数の導出を行った。その結果を表 2 に示す。

$$P_t = A \times (1 - \exp(-B \times x)) \quad (5-1)$$

表 5-2: 回帰分析による係数推定

捕捉センサ数 $M$	A	B	B 減少値
2	0.91371	0.24952	-
3	0.84068	0.17061	0.07891
4	0.83911	0.12265	0.04796
5	0.82692	0.09496	0.02769

表5-2 より、係数 A と B はともに捕捉センサ数  $M$  の増加に伴い減少することがわかる。特に係数 B に関しては、 $M$  の増加に対して B の減少値が約1/2 であることがわかる。この関係式をシミュレーションデータと比較したものを、図5-2 に示す。図より、センサ配置密度が小さい場合かつ、 $M$  が大きい場合に誤差が大きくなっているが、全体的に大まかな特性を近似式により表現できていることがわかる。この式を利用して、捕捉センサ数とトラッキング確率が要求条件として与えられた際に、その条件を満たすために必要なセンサ個数を容易に見積もることが可能となる。

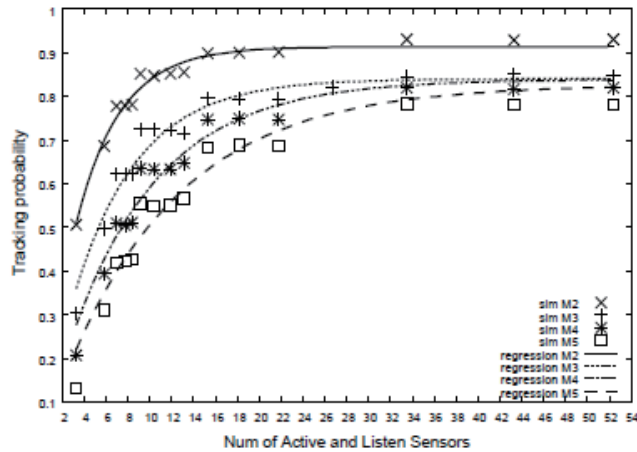


図5-2 : センサ配置密度とトラッキング確率の近似式とシミュレーション結果の比較

### 5.2.2. 捕捉センサ数 $M$ との関係

前節では、センサの配置密度とトラッキング確率の関係について述べたが、捕捉センサ数  $M$  が 2 ~ 5 の場合のみしか対応していない。実際に、センサ配置密度の見積りを行うためには、要求トラッキング確率、要求捕捉センサ数  $M$  を入力として密度を推定することが必要である。そこで、本節ではトラッキング確率と捕捉センサ数  $M$  との関係についての調査を行う。その後、この関係を表す近似式について検討する。

#### 5.2.2.1. シミュレーション結果

図5-3 に、Active、Listen 状態のセンサの配置密度  $N_{AL}$  が、0.93個/100m<sup>2</sup>、1.85 個/100 m<sup>2</sup>、3.70 個/100 m<sup>2</sup> の場合の捕捉センサ数  $M$  とトラッキング確率の関係を示す。図より、捕捉センサ数  $M$  の増加に伴い、トラッキング確率は指数的に減少することがわかる。これは、 $M$  の増加によって  $M$  個のセンサの共通センシング範囲が激減するためである。この関係から、トラッキング確率と捕捉センサ数の関係式として、次式を仮定する。

$$P_t = C \times D^M \quad (5-2)$$

表5-3:回帰分析による係数推定

センサ配置密度 [個/100m <sup>2</sup> ]	C	D
0.93	1.2745	0.6290
1.85	1.2035	0.7504
3.70	1.1315	0.8616

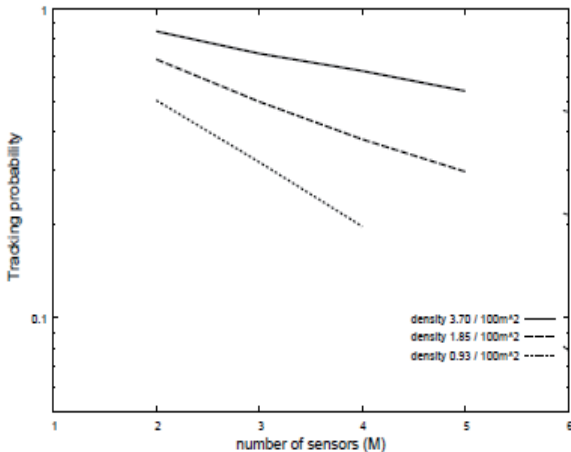


図5-3: 捕捉センサ数Mによるトラッキング確率への影響

### 5.2.2.2. 近似式の導出

次に、式(5-2)の係数C、Dを導出する。図5-3のシミュレーション結果を用いて非線形回帰を行った結果、表5-3の結果が得られた。表5-3より、センサの配置密度が2倍になる毎に係数Cは約0.07減少、係数Dは約0.11増加しており、関係式の各係数はセンサ配置密度に対して一定の増減関係が存在していることがわかる。次に、導出した近似式とシミュレーションデータを比較したものを図5-4に示す。図より、シミュレーションデータと関係式はほぼ一致していることがわかり、捕捉センサ数の変化、センサ配置密度の変化とトラッキング性能の関係を導出した式で示すことができていることがわかる。以上のように、捕捉センサ数Mとトラッキング確率の関係を明らかにし、その近似式を導出することができた。

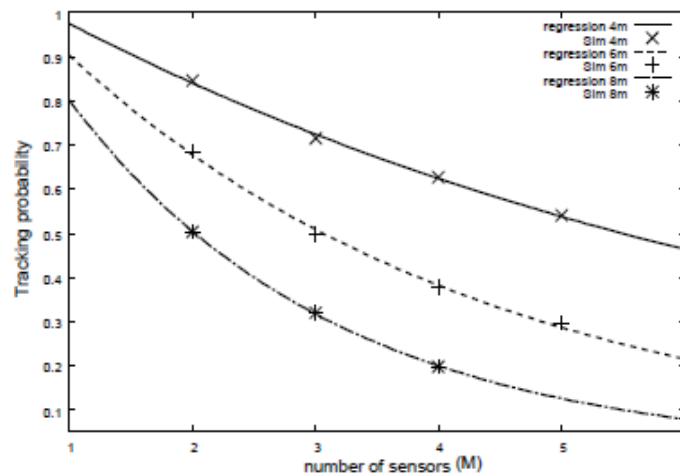


図5-4: 近似式とシミュレーション結果の比較

### 5.2.3. 関係式の統合と設計指針の検討

本節では、センサ配置密度に関するパラメータ(捕捉センサ数とセンサ配置個数)とトラッキング性能の関係について検討し、これまでに導出した近似式を統合して、トラッキング確率と要求捕捉センサ数の変化、双方に対応可能な配置センサ数を導出することで、複数センサの協調動作によるターゲットトラッキングのためのセンサネットワーク設計指針を示す。関係式の統合にあたって、センサ配置密度の関係式である式(5-1)を基に検討する。式(5-1)における係数Aはトラッキング確率の最大値を示しており、捕捉センサ数Mの変化に伴いその値も変化する。よって、係数Aと捕捉センサ数Mの関係として、式(5-2)に示すトラッキング確率と捕捉センサ数Mとの関係を用いると次式が得られる。

$$A = 0.9514 \times 0.9698^M \quad (5-3)$$

次に係数Bについて検討する。表5-1の回帰分析の結果より、係数BはMに対する階差数列であると考え

ることができるため、係数B と捕捉センサ数M の関係として次式を定義する。

$$B = 0.2495 - \sum_{i=2}^M 0.16 \times (1/2)^{i-2} \quad (5-4)$$

以上をまとめると、トラッキング確率に対する捕捉センサ数M、センサ配置個数の関係は次式のようになり、センサの配置密度、捕捉センサ数M、要求トラッキング確率から、それぞれを導出可能となった。

$$P_t = A \times (1 - \exp(-B \times x))$$

$$A = 0.9514 \times 0.9698^M \quad (5-5)$$

$$B = 0.2495 - \sum_{i=2}^M 0.16 \times (1/2)^{i-2}$$

このように、トラッキング確率を最大とするようなセンサの配置密度については、捕捉センサ数 M による影響が支配的であり、その値は M の増加と共に指数的に減少する。また、M が大きい場合にトラッキング確率を高めるためにはより多くのセンサを配置する必要があることが明らかになった。同様に、任意の捕捉センサ数/トラッキング確率を達成するためのセンサ配置密度についても、式(5-5) から導出することが可能である。

## 6. まとめ

本研究では実環境において実機センサノードを用いたネットワークを構築し、その上でこれまでに提案してきた複数センサの連携による移動ターゲットトラッキング手法を実装し、実環境での手法の性能調査を行った。まず、Atlas を用いてターゲットを検知するための機構を開発し、その上で複数センサからの検知結果に基づく移動ターゲットの現在位置、及び移動履歴(動線)を示すためのアプリケーション開発を行った。これに加え、実環境において Mote センサを用いた際の省電力化を実現するための手法についても検討を進め、センシング状態をフィードバックすることで省電力化を実現できることを実環境での実験により明らかにした。

また、複数センサによる移動ターゲットトラッキング手法の更なる改善を行い、その有効性をシミュレーション実験及び近似式導出による解析的アプローチによって評価した。その結果、シミュレーション実験から提案手法の有効性を確認し、近似式から提案手法を用いた際の性能を要求トラッキング確率、配置密度を元に推定出来ることを明らかにした。

## 【参考文献】

[1] K. Tsukamoto, H. Ueda, et.al., "A simple Multi-Point Surveillance Scheme of a Moving Target for Wireless Sensor Networks," SUPE'07, Nov. 2007.

[2] 植田啓文、塚本和也、田村瞳、川原憲治、尾家祐二、須田達也、"ターゲットの移動性を考慮した複数センサによるトラッキングアルゴリズムの消費電力特性の調査、" 電子情報通信学会技術研究報告、vol. 107、no. 423、IN2007-121、pp. 13—18、2008年1月。

[3] 安藤繁、他、"センサネットワーク技術ユビキタス情報環境の構築に向けて、" 東京電気大学出版局、2005。

[4] 植田啓文、塚本和也、田村瞳、川原憲治、尾家祐二、須田達也、"複数センサによる移動ターゲット Tracking 手法の提案と評価、" 電子情報通信学会 2006年総合大会、A-21-22、2006年3月。

[5] 植田啓文、塚本和也、田村瞳、川原憲治、尾家祐二、須田達也、"ターゲットの移動性を考慮した複数センサによるトラッキングアルゴリズムの提案と評価、" 電子情報通信学会技術研究報告、vol. 106、no. 358、IN2006-90、pp. 7--12、2006年11月。

[6] J. Hill, M. Horton, R. Kling, and L. Krishnamurthy, "The Platforms Enabling Wireless Sensor Networks," ACM Communications, Vol. 47, No. 6, pp. 4146, Jun. 2004.

[7] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling Ultra-Low Power Wireless Research,"

Poster session at 4th Int. Conf. Information Processing in Sensor Networks: Track on PlatformTools and Design Methods for Network Embedded Sensors (IPSN/SPOTS), Apr. 2005.

[8] 植田啓文、塚本和也、田村瞳、川原憲治、尾家祐二、“複数センサによる移動ターゲット捕捉のためのセンサネットワーク設計指針の検討 -ターゲット捕捉確率とセンサ配置密度の関係調査、” 電子情報通信学会技術研究報告、vol. 107、no. 449、IA2007-51、pp. 63-68、 2008 年 1 月.

〈発 表 資 料〉

題 名	掲載誌・学会名等	発表年月
A Simple Multi-point Surveillance Scheme of a Moving Target for Wireless Sensor Networks	IEEE Workshop on Service Discovery and Composition in Ubiquitous and Pervasive Environments (SUPE' 07)	2007 年 11 月
複数センサによる移動ターゲット捕捉のためのセンサネットワーク設計指針の検討 -ターゲット捕捉確率とセンサ配置密度の関係調査	電子情報通信学会、技術研究報告、 IA2007-51	2008 年 1 月
ターゲットの移動性を考慮した複数センサによるトラッキングアルゴリズムの消費電力特性の調査	電子情報通信学会、技術研究報告、 IN2007-121	2008 年 1 月
容易に管理可能な無線センサネットワークのための冗長性デザイン	電子情報通信学会、技術研究報告、 IN2007-39	2007 年 7 月