

IR-UWB を用いた自己組織化ノード位置推定方式における改良アルゴリズムとその精度改善効果

森 流星¹ 北之馬 貴正² 川田 千尋¹ 滝沢 泰久³

概要: 我々は屋内空間内に遍在する多数のスマートデバイスの位置を定点 3 点のみで高精度に推定する集約型自己組織化ノード位置推定方式 (SOL: Self-Organizing Localization) を提案している。集約型 SOL では各ノード間の相対距離を用いてノード位置を推定する。ノード間相対距離に IR-UWB を用いた場合の位置推定精度は約 30~40cm 程度である。集約型 SOL は相対座標系で位置推定を行い、定点 3 点を用いて絶対座標系に変換する。そのため定点 3 点の位置推定精度に応じて精度が劣化する問題が存在する。本論文ではこの問題を解決するため、IR-UWB の測距特性に適応した集約型 SOL の改良アルゴリズムを提案し、その精度改善効果を実機実験により示す。

1. はじめに

近年、ダイナミックセル生産方式の工場や建設現場のような障害物の位置や空間レイアウトが頻繁に変わる動的環境の作業現場において、自律型サービスロボットの導入検討が進められている。このような作業現場では、複数台の自律型サービスロボットが協働し人の労働力の代替として、モノの運搬、設備配置、清掃などの作業を行うことが考えられる。ダイナミックセル方式の生産現場や建設現場でのロボットは、ライン生産方式の工場に代表されるようなロボットの位置が固定、または移動経路は事前に決定して作業を行うのではなく、作業状況や現場状況に応じて移動し作業を行うため、業務管理上において複数台の自律型サービスロボットの移動状況とその軌跡把握が必要となる。また、移動軌跡の位置精度は、一人一人が停止して作業を継続する範囲を肩幅相当 50cm を最小範囲とした場合、数十 cm オーダーの位置精度が必要である。

自律移動ロボットの研究において、ロボットと周辺との相対的な距離把握機能の実現のためカメラやレーザーレンジファインダを用いた方式が多数提案されている [1]。しかし、この方式は大域的な自己位置把握機能に欠けるため、自律移動ロボットは自身の現在位置 (所在場所) を把握で

きない。そのため、移動空間にカメラの配置や移動空間のマップ (ランドマーク) を用意することにより、自律移動ロボットが現在位置を把握するとともに、業務管理においてもその移動軌跡を把握可能とする。しかし、ダイナミックセル生産方式の工場や建設現場では、作業工程の組み替えにより作業空間のレイアウトは頻繁に変動するため、前述のようなカメラやマップなどの移動環境内の測位設備は、その都度、再構成する必要があり、大きなコストが発生し、持続可能性が低い。従って、測位設備への依存性が低い大域的な位置推定方式が必要である。

現在、大域的な位置推定を行う方法として電波を用いる iBeacon[2] や Indoor Messaging System (IMES)[3]、各種センサを用いるフィンガープリンティング [16]、カメラ画像を用いる手法[?]がある。電波を用いる手法では環境内に多数の定点ビーコンを必要とし、フィンガープリンティングでは事前にセンサ取得情報に応じた環境の物理情報マップの作成が必要である。また、これら方式の位置推定精度は数 m オーダーであり、自律移動ロボットの移動軌跡把握の精度要件を満たせない。高精度な位置推定方式としてカメラによる多点計測の方式があり、数 cm オーダーでの位置把握が可能とされている。しかし、高精度な位置推定を行うには相当数のカメラと事前の十分なデータによる位置推定機能の学習を必要とし、また見通し環境を確保する必要がある。すなわち、いずれにおいても、位置推定のための十分な設備や事前準備を必要とするため、頻繁にレイアウト変更があるダイナミックセル生産環境への適用は困難である。

北之馬らは、屋内の集約型自己組織化スマートデバ

¹ 関西大学 理工学研究科
Graduate School of Engineering, Kansai University

² 関西大学 先端科学技術推進機構
Organization for Research and Development of Innovative Science and Technology, Kansai University

³ 関西大学 環境都市工学部
Faculty of Environmental and Urban Engineering, Kansai University

イス位置推定方式 SmartFinder[5][6] を改良している。SmartFinder は、スマートデバイス間の隣接情報を用いて定点 3 点のみで、多数のスマートデバイスの位置を取得する自律型屋内測位技術であり、測位設備の依存性が極めて低い。SmartFinder は、スマートデバイス間の通信を Bluetooth Low Energy (BLE) とし、集約型自己組織化ノード位置推定方式 (Self-Organizing Localization: SOL) を用いてスマートデバイスの位置を推定する。その位置推定精度は 2m 程度である。

ダイナミックセル方式の生産現場や建設現場に代表される屋内レイアウトが動的にレイアウト変更・更新される屋内環境において、自律移動ロボットの大域的な移動軌跡をリアルタイムで把握可能とするため、前述の集約型 SOL に IR-UWB を適用する方式を提案している [7]。本方式は、集約型 SOL と同様に 3 定点のみで位置推定可能であり、また IR-UWB によりその位置推定精度は 30~40cm である。本方式は、集約型 SOL と同様に、相対座標で全デバイスの位置推定し、3 定点の推定位置と真位置から相対座標を絶対座標へ変換して、全デバイスの絶対座標位置を取得する。そのため、3 定点の相対座標位置の精度が劣化すると、全デバイスの位置推定精度が劣化する問題がある。本稿では、この問題を解決するため IR-UWB の測距特性に適應した集約型 SOL の改良アルゴリズムを提案する。さらに、実験から改良アルゴリズムの有用性を示す。

2. 関連研究

現在利用もしくは研究されている位置推定方式は、電波を用いる方式とセンサ情報を用いる方式が存在する。電波を用いる方式はノード間を用いるレンジベース位置推定方式 (Range-based) とノード間距離を用いないレンジフリー位置推定方式 (Range-free) に分類できる。本章では上記の例の概説をする。

2.1 Range-Based 方式

Range-Based 方式ではノード間距離を利用するため測距機能を持った無線ノードが必要となる。ノード間距離の測距方式は TOA (Time of arrival), TDOA (Time Difference Of Arrival), RSSI (Received Signal Strength Indicator) が利用されている。

2.1.1 TOA 利用方式

TOA 方式は、送信側から受信側に信号が到着するまでの時間を計測し、転送媒体 (超音波、電波等) の転送速度からノード間距離を計算する方式である。TOA 方式を利用した主な位置推定方式として Global Positioning System (GPS) [8] や Ultra Wide Band(UWB) [9] など存在する。GPS では GPS 衛星からの信号を受信し時刻データに電波の伝搬速度を掛け合わせることで GPS 衛星との距離を算出する。3 個以上の GPS 衛星との通信ができれば 3 点測

量の要領で位置推定が可能である。しかし屋内や山間部では GPS 衛星と通信が困難なために位置推定精度の劣化が発生する。

UWB は非常に短いパルスを用いることにより高精度なノード間測距が可能である。しかし電波の通信カバレッジが短いので位置推定を行うためには多数のアンカーノードを必要とする。

2.1.2 TDOA 利用方式

TDOA 方式は、異なる複数の伝送媒体からの信号の到着時間の差を用いて測位を行う。一般的には電波や音波が用いられるが、各伝送媒体による通信カバレッジや環境依存特性の違いから利用環境が限定的になる。TDOA 方式を利用した主な位置推定方式として Active Bat[10], Cricket[11] などが挙げられる。

2.1.3 RSSI 利用方式

RSSI 方式は電波強度を利用しノード間距離を測距する。RSSI を用いた位置推定方式として RADAR, SpotON がある。電波強度のみでノード間の測距が可能のため比較的単純な機能で実装が可能である。しかしこの方式ではフェージングの影響で電波の強弱が変動することがあるため、正確な測距を行うことが難しいとされる。そのため正確な位置推定を行うためには大量のアンカーノードや事前の電波状況の計測が必要となる問題が存在する。

2.1.4 Range-Free 方式

Range-Free 方式ではセンサノードが測距機能を持つ必要がないため、Range-Based 方式と比較し安価で単純な実装が可能である。Range-Free 方式を用いた位置推定方式は、Centroid[12], APIT[13] がある。Centroid 方式は、通信可能なアンカーノードの位置情報を取得しそれらの重心を用いて位置を推定する。Centroid 方式は仕組みが単純であるが精度は低い。APIT 方式は、複数個のアンカーノードの組み合わせから作成可能な全ての三角形に対して、位置推定対象のノードが外側にあるか内側にあるかを判定することで自身の位置を推定する方式である。APIT 方式では精度が構成可能な三角形の数に依存するため、アンカーノード数に精度が依存するとされる。

2.2 センサを用いた方式

センサを用いた手法では主に Dead Reckoning(DR)[14], Simultaneous Localization And Mapping(SLAM)[15] などが該当する。DR 法ではジャイロセンサや加速度センサなどの各種センサの情報を元に移動距離、移動方向を算出し過去の自己位置からの相対位置を算出する方式である。DR 法では過去の自己位置と比較を行うためセンサ値の誤差による位置推定誤差の蓄積が問題となる。そのため、誤差の補正と絶対位置を得るために、相当数の定点が必要となる。SLAM は初期に大域的な自己位置推定機能または環境マップのいずれかが必要である。大域的な自己位置推定

機能を有する場合はこれを用いて動的に環境マップを作成することは可能である。一方、大域的な自己位置推定機能を有しない場合は、事前に用意した環境マップから大域的自己位置推定を行い、これをフィードバックして環境マップを更新する。しかし、環境が動的に変わる状況には適用が困難である。従って、SLAMは環境が動的に変わる状況において大域的な自己位置推定機能が必須となる。

3. IR-UWB を用いた集約型 SOL

IR-UWB を用いた集約型 SOL ははスマートデバイスモジュールとサーバモジュールから構成される。以下では IR-UWB を用いた集約型 SOL を概説する。

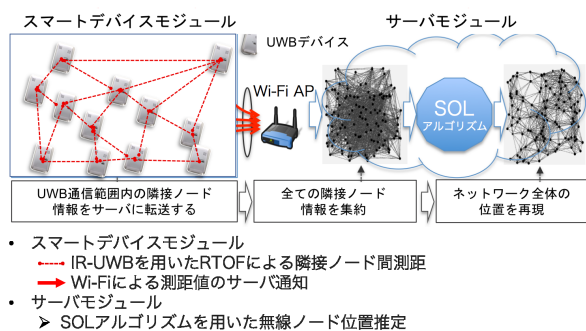


図 1 IR-UWB を適用した集約型 SOL の構成

3.1 IR-UWB を用いた集約型 SOL システム構成

IR-UWB を用いた集約型 SOL は主に多数のスマートデバイスモジュールとサーバモジュールから構成される (図??)。それぞれの主な動作は、スマートデバイスモジュールでは隣接ノード ID 情報及び隣接ノード間距離の取得と、Wi-Fi/LTE を用いたサーバへの情報集約を行う。サーバモジュールでは集約された隣接ノード ID 及びノード間距離から仮想メッシュネットワークの構成、集約型 SOL によるスマートデバイスの位置推定を行う。

3.1.1 スマートデバイスモジュール

集約型 SOL に用いる IR-UWB デバイスとして日本ジー・アイ・ティー社製の IR-UWB デバイス²を用いた。本デバイスでは 7.25-10.25GHz 帯において端末間で RTOF により測距を行い、誤差数 cm 以内の高精度な測距が可能である。以下に UWB デバイス間での測距アルゴリズムを示す。

- 相互のブロードキャスト通信で隣接ノード情報を取得。
- 上記隣接ノードへユニキャスト送信。
- ユニキャスト受信ノードは一定時間後にユニキャストで返信。
- 返信受信ノードで通信往復時間により距離を算出 (式??)。
- 端末間で数十回の測距を行い測距成功時の平均を距離として算出する。

以上の手順で測距したノード間距離を Wi-Fi を用いてサーバモジュールへと転送する。また、UWB の通信は implus Radio で極めて短い送信時間であることから、アクセス制御は実施しないことし、通信制御を軽減する。

3.1.2 サーバモジュール

サーバモジュールではスマートデバイスモジュールから集約された隣接ノード ID 及びノード間距離情報を元に仮想メッシュネットワークの構成/更新を行い集約型 SOL を適用することでスマートデバイスの位置を推定する。

3.2 サーバモジュールにおける仮想ネットワークの構成及び更新

サーバモジュールではスマートデバイスから集約した隣接ノード情報及びノード間距離情報を用いて、隣接ノード情報の欠損を考慮した仮想メッシュネットワークの構成/更新を行う。

3.2.1 隣接ノード情報の欠損の考慮

UWB の通信においてアクセス制御を実施しないため、電波干渉によるノード間距離計測の失敗の可能性がある。隣接ノード間距離計測の起点となるブロードキャスト通信が干渉要因などにより失敗した場合、多くの隣接とのノード間距離計測ができなくなる。この場合を想定して、計測に成功した直近のノード間距離を保持する。直近のノード間距離を短期間で破棄した場合、ノード間リンクを失う。集約型 SOL はノード間距離への誤差耐性が高い特徴を有することから、ノード間距離データが少なくなるより、過去のノード間距離、すなわち、誤差を含むノード間距離を採用してノード間距離データを確保することが位置推定精度に有利と考える。以上を踏まえ隣接ノード情報保持期間 t_g はノード間距離計測周期より十分に長い時間とする。また、より直近に計測したノード間距離を用いるため、2ノードでそれぞれ計測したノード間距離においてより最新のノード間距離を2ノードのノード間距離として採用する。

3.2.2 仮想メッシュネットワークの構成

隣接ノード間距離情報を用いてサーバモジュールの構成を行う。まず隣接ノード情報及びノード間距離情報から仮想メッシュネットワークの構成を行う。式 (1) を用いて、仮想メッシュネットワークにおいてノード相対距離の定義を行う。ノード i とノード j を結ぶ経路において、ノード間リンク長の和が最短となる経路をノード i とノード j のノード間経路とし、その距離をノード間距離 d_{ij} とする。

$$d_{ij} = \min(d(p_{ij}) | S(p_{ij})), d(p_{ij}) = \sum_{l_{xy} \in p_{ij}} l_{xy} \quad (1)$$

ただし、 l_{xy} はノード x とノード y 間のリンク、 p_{ij} はノード i から j までの経路となるリンク l_{xy} の集合、 $d(p_{ij})$ はノード i からノード j までの経路長、 $S(p_{ij})$ は p_{ij} の集合

とする。

3.3 集約型 SOL によるスマートデバイスの位置推定

集約型 SOL は隣接ノード間相対距離を元に位置情報の修正を繰り返し行いネットワークの形状の再現を行う。以後、位置推定のための1回の計算処理を1ステップとする。また各ステップでの計算過程の推定位置を仮位置とする。隣接ノード ID 情報の取得/集約、仮装メッシュネットワークの構成、集約型 SOL の計算終了までの過程を1サイクルとし、集約型 SOL の推定結果を位置推定結果とする。

ノード間経路長の近傍範囲

位置修正に用いる近傍ノードをノード間経路長から選択するため、選択の基準となるノード間経路長の近傍半径を算出する。全ての1次近傍ノード間におけるノード間経路長から、通信半径に相当するノード間経路長の近傍半径を算出し、これを位置修正に用いる近傍ノード選択の基準に用いる。ノード間経路長の近傍半径を R とすると、1次近傍ノード間におけるノード間実距離の期待値は以下の式2により算出される。

$$E[x] = \int_0^R x \frac{2\pi x}{\pi R^2} dx = \frac{2}{3}R \quad (2)$$

ここで、1次近傍ノード間におけるノード間経路長の平均を \bar{d} とすると、ノード間経路長の期待値から算出した近傍半径 R は式14となり、 H 近傍ノードは $(H-1)R$ 以上 HR 未満のノード間経路長に相当する。

$$R = 1.5\bar{d} \quad (3)$$

これにより、ノード間経路長から位置修正段階に応じたノード選択ができるため、任意のホップ数で構成されたトポロジにおいても高精度な位置推定が可能となる。

3.4 集約型 SOL によるスマートデバイス位置推定

集約型 SOL では仮想メッシュネットワーク全体の停止ノードを用いて高精度に位置推定を行う。集約型 SOL では数十秒程度の十分な推定時間を利用できるとし、数十秒の周期で位置推定を実施する。集約型 SOL のアルゴリズムは以下の手順で分けられる。

- 集約型 SOL アルゴリズムによる位置推定
- 絶対座標変換
- 推定ジオメトリの領域判定値算出
- 最小値の領域判定値と推定ジオメトリを記憶

集約型 SOL における位置推定アルゴリズムは以下の手順で実施される。

[Step.1] 各ノードの推定位置をランダムに生成する。以降、 t 回目の修正におけるノード i の推定位置を $w_i(t)$ とする。

[Step.2] ノード i において位置修正を行う対象ノードを選択する。位置修正対象ノードは集約型 SOL におけるノード間相対距離が t 回目の修正における閾値 $\gamma^g(t)$ 以下のノード

からランダムに1つを選択して、これをノード m とし、ノード i と m との位置推定を行う場合、ノード間相対距離 $d_{im}^{\{\gamma^g(t)\}}$ を用いてノード i の修正ベクトル $V_{im}^{\{\gamma^g(t)\}}(t)$ を、次のように定義する。

$$V_{im}^{\{\gamma^g(t)\}}(t) = \frac{d_{im}^{\{\gamma^g(t)\}} - |w_i(t) - w_m(t)|}{|w_i(t) - w_m(t)|} (w_i(t) - w_m(t)) \quad (4)$$

この修正ベクトル $V_{im}^{\{\gamma^g(t)\}}(t)$ を用い、ノード i の位置修正は次のように行う。

$$w_i(t+1) = w_i(t) + \alpha_i(t) \cdot (V_{im}^{\{1.0\}}(t) + V_{im}^{\{\gamma^g(t)\}}(t)) \quad (5)$$

$$\alpha_i(t) = \eta \alpha_i(t-1) \quad (0 < \eta < 1). \quad (6)$$

$\alpha_i(t)$ はノード i の t 回目の修正における学習係数を表す。初期の段階では全体的なメッシュネットワークの形状を再現を行い、位置修正が進むほどに局所的な位置推定を行う。そのため仮位置修正回数に応じて $\gamma^g(t)$ の値を減少させる必要がある。以上を考慮すると $\gamma^g(t)$ は式(7)により決定する。

$$\gamma^g(t) = \begin{cases} d^{\max} - \frac{t(d^{\max} - 2R)}{S^g} & (d^{\max} > 2R) \\ 2R & (\text{otherwise}) \end{cases} \quad (7)$$

S^g は集約型 SOL における仮位置修正回数を表す。 d^{\max} は全ノード間におけるコスト距離最大値を表す。

[Step3] アンカーノードを基準とした絶対座標変換を行い、全ノードの絶対位置を算出する。相似形を再現した仮装メッシュネットワークを3点のアンカーノードの絶対位置と推定位置を用いて絶対座標へと変換する。アンカーノードの真位置 $W_A = (X_A, Y_A)$ は推定位置 $w_A = (x_A, y_A)$ を用いて以下のように表される。

$$\begin{aligned} X_A &= ax_A + by_A + t_x \\ Y_A &= cx_A + dy_A + t_y \end{aligned} \quad (8)$$

3つのアンカーノードから構成される連立方程式(8)から6つの係数 a, b, t_x, c, d, t_y を得ることにより、すべてのノードは以下のように推定位置 $w_i = (x_i, y_i)$ から絶対座標 $\hat{w}_i = (\hat{x}_i, \hat{y}_i)$ へ変換し絶対位置を得る。

$$\begin{pmatrix} \hat{x}_i \\ \hat{y}_i \\ 1 \end{pmatrix} = \begin{pmatrix} a & b & t_x \\ c & d & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \quad (9)$$

矛盾の少ないジオメトリを推定するために相対距離矛盾率の算出による推定ジオメトリ評価を用いる(図??)。ノード i において、ランダムに選択したノードを2次近傍ノード相当のノード l とし、ノード i とノード l に対するノード間コスト距離がノード i とノード l 間のコスト距離未満となるノード、すなわち、 $(d_{ij} \leq d_{il})$ かつ $(d_{jl} \leq d_{il})$ のノードを1次近傍ノード相当のノード j とする。 w_i と w_j を基準点とし、線分 $\overline{w_j - w_i}$ の垂直二等分線を用いて w_i と w_j

のいずれかに近い領域に空間を2分割し、 w_i に近い領域にある ($|w_l - w_i| \leq |w_l - w_j|$) 場合、コスト距離矛盾と判定する。

これにより、任意のノード間相対距離に対して相対距離の矛盾を検知することが可能となる。さらに、検知領域を拡大するため、上記と同様に全ての共通1次近傍相当のノード群による複数の分割空間を重ね合わせて相対距離矛盾の検知範囲を拡大し、誤推定検知の可能性を高める。全ての2次近傍ノードとの組み合わせを領域判定値とし、これにより最小の領域判定値のジオメトリを選出し、位置推定結果とする。

4. 集約型 SOL 改良アルゴリズム

IR-UWB を用いた集約型 SOL では、相対座標で全デバイスの位置推定し、3定点の推定位置と真位置から相対座標を絶対座標へ変換して、全デバイスの絶対座標位置を取得する。そのため、3定点の相対座標位置の精度が劣化すると、全デバイスの位置推定精度が劣化する問題がある。本問題を解決するため、以下に IR-UWB を用いた集約型 SOL における位置推定アルゴリズムの改良方針を示す。

- 全隣接 (1 ホップ) ノードによる更新

UWB で計測された1ホップ間ノード環境は高い精度・高解像度である、一方、多ホップノード間距離は各ノード間計測距離の和とするため多ホップノード間の中間ノードが直線上に位置する距離となる。すなわち、多ホップノード間距離は実距離より大きな値となり、ホップ数が増えとこの距離誤差が増大する。従って、多ホップノード間距離を排除し、高精度・高解像度の1ホップノード間距離を多く採用するため、全隣接 (1 ホップ) ノード間距離を用いることとする。

- 局所移動を想定した位置推定処理量の抑制

移動ノードは人との協働を想定した場合、人の歩行速度相当の移動速度と想定される。すなわち、移動ノードの移動範囲は、SOL の位置推定出力周期において、局所にとどまる。従って、継続的な位置出力では直近の移動位置を初期値として位置推定を開始し、位置推定処理量を削減する。この削減により、全隣接ノードによる位置推定処理量増加を相殺し、位置推定出力遅延を抑制する。

- ノード間距離誤差の補正による精度劣化抑制

SOL はアンカーノードを含めて相対座標において位置推定する。その後、アンカーノードの相対座標における推定位置と絶対座標における真位置から全ノードの相対座標の推定位置を絶対座標へ変換する。この座標変換においてノード間距離誤差によりアンカーノード推定位置精度が劣化する場合、座標変換により全ノードの精度誤差を拡大する。UWB 計測のノード間距離は絶対距離であるので、誤差がなければ相対座標

のノード位置集合と絶対座標のノード位置集合のそれぞれの凸包面積は一致する。しかし、UWB 計測距離は実距離より大きくなる傾向にある。従って、これが一致しない場合は一致するようにノード間距離を修正し、かつアンカーノードを真位置に固定し、絶対座標にてノード位置推定する。

4.0.1 集約型 SOL の位置推定改良アルゴリズム

上記に基づいて集約型 SOL の位置推定改良アルゴリズムは以下の手順で実施する。以降、位置推定出力周期 t 回目におけるノード i の位置推定過程の n 回目の修正時のノード i の仮位置を $w_i^t(n)$ とする。

[Step1] 各ノードの初期仮位置をランダムに設定する。

[Step2] $w_i^t(n+1)$ を、ノード i の全隣接ノードの集合 $neighbor_i$ から次のように求める

$$V_{im}^t(n) = \frac{d_{im}^t - |w_i^t(n) - w_m^t(n)|}{|w_i^t(n) - w_m^t(n)|} (w_i^t(n) - w_m^t(n)) \quad (10)$$

$$w_i^t(n+1) = w_i^t(n) + \sum_{m \in neighbor_i} \alpha_i^t(n) \cdot V_{im}^t(n) \quad (11)$$

$V_{im}^t(n)$ はノード i の位置推定出力周期 t 回目における n 回目の修正におけるノード m による位置修正ベクトル、 d_{im}^t は位置推定出力周期 t 回目におけるノード i と m の UWB 計測ノード間距離、 $\alpha_i^t(n)$ はノード i の位置推定出力周期 t 回目における n 回目の修正の学習係数である。式 (1)(2) を全てのノードに適用して、各ノードの仮位置を修正する。この修正を全てのノードで N 回繰り返す、全ノードの相対座標における位置を推定する。

[Step3] アンカーノードを基準とした絶対座標変換を行い、全ノードの絶対位置を算出する。

[Step4] 相対座標におけるノード位置集合と絶対座標におけるノード位置集合のそれぞれの凸包面積を a^t, A^t とし、ノード間距離誤差を排除するため、それぞれの凸包面積を用いてノード間距離を次のように修正する。

$$D_{im}^t = d_{im}^t \cdot \sqrt{\frac{A^t}{a^t}} \quad (12)$$

[Step5] 座標変換による誤差拡大を回避するため、Step2 においてアンカーノードを真位置に固定して、式 (1) の d_{im}^t を式 (3) の D_{im}^t とし、絶対座標において位置推定を行う。また、全隣接ノードによる位置推定処理量増加を相殺するため、局所な移動を想定して初期仮位置を直近 (周期 $t-1$ 回目) の推定位置とし、初期学習係数を小さな値として、各ノードの仮位置修正回数を少なくする。これにより得られた位置を周期 t 回目のノード推定位置として出力する。

5. 評価

5.1 改良アルゴリズムによる精度向上評価

IR-UWB デバイスを移動体 Roomba600series[18] に搭載し Roomba の位置推定を行う。Roomba は全8台用意し4台を停止体、4台を移動体とし移動させる。Roomba は

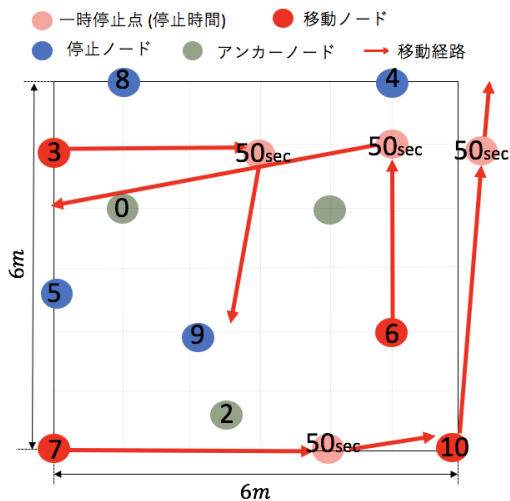


図2 ノード配置図



図3 実験風景

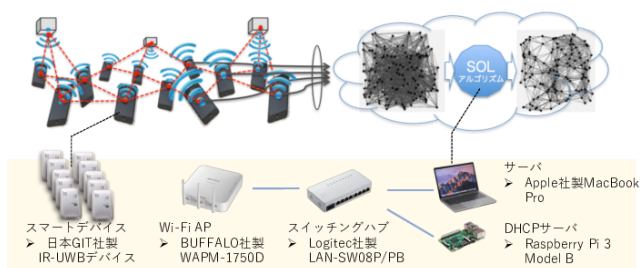


図4 実装評価におけるシステム構成図

予め決められたルートを移動するとし、Roombaの移動シナリオを図2に示す。ID3, 6, 7, 10のRoombaを移動体とし。ID4, 5, 8, 9のRoombaを停止体とする。薄い赤丸と数値は一時停止場所と停止時間を示す。図3に実験風景、表2に実験諸元を示す。改良アルゴリズムによる精度向上評価では通信媒体をUWBとして以下の表2(方式U-C~z)の比較を行う。提案改良方式U-Uの3つの改良点、全隣接更新、更新量抑制、ノード間距離補正をそれぞれ個別に用いる方式をU-x, U-y, U-zとする。

位置推定精度評価は、実時刻 t において推定された各ノードの位置と同実時刻 t におけるユークリッド距離の平

均である位置推定誤差 Err^t (式13を用いて評価する。

$$Err^t = \frac{1}{|N|} \sum_{i=1}^N |W_i^t - w_i^t| \quad (13)$$

実時刻 t の仮想メッシュネットワークにおけるノード i の1ホップノード間距離を SD_i^t を用いて評価する。 SD_i^t は仮想メッシュネットワークにおける隣接ノード距離とオリジナルネットワーク形状における近傍との距離精度の誤差分散を正規化して評価する。 SD_i^t は次の式により評価する。

$$r_{ij}^t = \frac{d_{ij}^t}{D_{ij}^t}, \quad \bar{r}^t = \frac{1}{|N|C_2} \sum_{i=1}^{N-1} \sum_{j=i+1}^N r_{ij}^t \quad (14)$$

$$SD_i^t = \frac{\sqrt{\sum_{j=i+1}^N (\bar{r}^t - r_{ij}^t)^2}}{Link_i^t} \quad (15)$$

$Link_i^t$ は実時刻 t におけるノード i の1ホップノード数を表す。 d_{ij}^t, D_{ij}^t はそれぞれ実時刻 t でのノード ij 間における真距離、仮想メッシュネットワークにおけるノード間距離を示す。 \bar{r}^t が1に、 SD_i^t が0に近いほど測距精度が正確であることを示す。

5.1.1 改良アルゴリズムによる精度向上評価結果

図5に従来アルゴリズムを示す方式U-Cと改良アルゴリズムを全て用いた場合を示す方式U-Uの比較結果を示す。全時間において方式U-Uは方式U-Cより精度が良くなっていることが確認できる。特に2度目の移動期間である120~165secにおいては大きな位置推定精度が向上の確認ができた。以下では提案改良アルゴリズムの優位性を確認するため個別評価結果を図6, 8, 7に示す。全隣接更新単

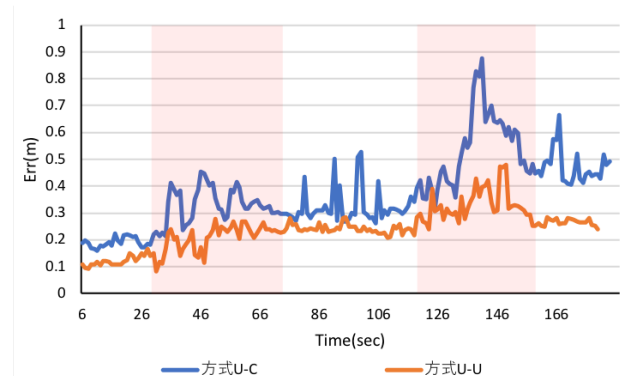


図5 方式U-Cと方式U-Uにおける Err^t 推移

独の効果を示すため、図6に従来アルゴリズムを示す方式U-Cと従来アルゴリズム+全隣接更新を示す方式U-xの比較結果を示す。全隣接更新にすることで2hopノードによる位置更新がなくなるため高解像度の1hopノード間距離での更新のみとなる。そのため2hopノードによるノード間誤差の大きい更新が無い場合位置推定精度が安定している。特に移動期間(30~70sec, 120~165sec)において、方

表 1 改良アルゴリズムによる精度改善評価結果

	従来アルゴリズム	全隣接更新	位置推定処理量抑制	ノード間距離誤差補正
評価方式 U-C	○	×	×	×
評価方式 U-U	○	○	○	○
評価方式 U-x	○	○	×	×
評価方式 U-y	○	×	○	×
評価方式 U-z	○	×	×	○

表 2 移動体実験環境

フィールド範囲 (m^2)	49
アンカーノード数	3
停止ノード数	4(ID : 4, 5, 8, 9)
移動ノード数	4(ID : 3, 6, 7, 10)
移動ノード移動速度 (cm/sec)	10
実験時間 <i>cycle</i> (sec)	180

表 3 実機実験におけるパラメータ値

	方式 U-C, U-x, U-z	方式 U-U, U-y
SOL の推定位置出力周期	1	1
隣接ノード情報保持期間	5	5
SOL の仮位置修正回数	600	200
学習係数初期値	1.0	0.1
減衰定数	0.992	0.992

式 U-C と比較して精度安定効果が強く現れている。移動に伴って UWB の測距精度がばらつく場合、2 ホットボロジはそのばらつきが累積されるためばらつきが増幅され影響を受けやすいと考えられる。一方、全隣接更新の場合は全隣接を用いることから平均効果によりばらつきによる影響を相殺して精度安定に有利に機能すると考えられる。

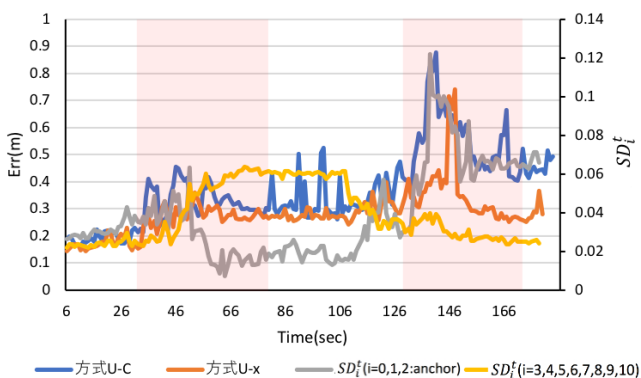


図 6 方式 U-C と方式 U-x 比較

ノード間距離誤差補正単独の効果を検証するため、図 7 に従来アルゴリズムである方式 U-C と従来アルゴリズム+ノード間距離誤差の補正を示す方式 U-z の比較結果を示す。方式 U-z のノード間距離誤差の補正はアンカーノード近傍の測距精度が高い場合 (66~110sec) に、おおよそ 0.2m の誤差精度になり高精度化の効果を有する。これは、ノード間距離補正は相対座標と絶対座標のスケール比を求める場合、アンカーノードの推定位置と真位置の相関を用

いるため、アンカーノード推定位置の精度の高い場合は、これを用いることにより他のノードのノード間距離誤差を補正することができ、高精度化に寄与していると考えられる。

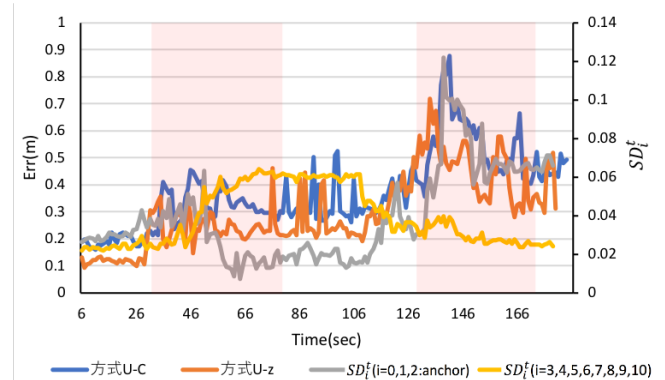


図 7 方式 U-C と方式 U-z 比較

図 8 に位置推定処理量による精度劣化がないことを検証するため、図 25 に従来アルゴリズムである方式 U-C と従来アルゴリズム+位置推定処理量抑制の方式 U-y の位置推定精度を示す。方式 U-y では従来のステップ数 600 からステップ数を 66%削減した 200 ステップで平均位置推定精度 0.382m となった。従来アルゴリズム方式 U-C の位置推定制度から 0.02m 精度が劣化しているが同等位置推定精度を保ちつつ位置推定処理量抑制が出来ていると考えられる。

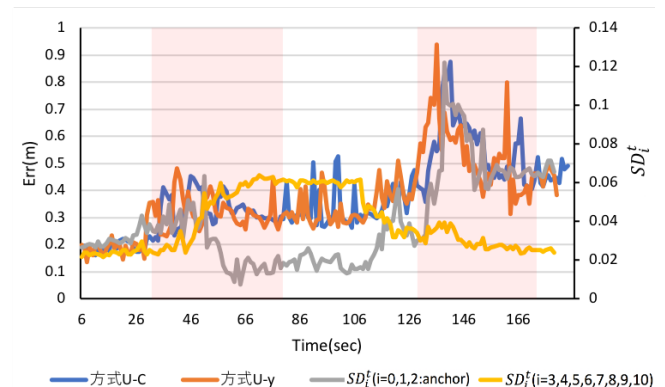


図 8 方式 U-C と方式 U-y 比較

以上の結果から、全隣接更新は位置推定精度の安定効果を有し、ノード間距離補正は位置推定精度の高精度化効果を有し、さらに位置更新処理量抑制は精度を劣化なしに、処理量を抑制すると考える。表 5 にこれらの各方式の平均位置推定精度を示す。改良アルゴリズムを用いた方式 U-U

がもっとも良い精度となっており各アルゴリズムを複合した場合にも有効に働くことがわかる。

表 4 改良アルゴリズムによる平均位置推定精度結果

	方式 U-C	方式 U-U	方式 U-x	方式 U-y	方式 U-z
Err(m)	0.372	0.239	0.279	0.382	0.314

6. まとめ

本稿ではダイナミックセル方式の生産現場や建設現場などの動的環境における自律型サービスロボットの利用を想定し、作業管理上において必要となる自律移動ロボットの移動軌跡を数十 cm で軽微な測位設備で推定する屋内測位方式として、集約型 SOL へ IR-UWB デバイス適用する改良方式を提案した。さらに Roomba を用いた実装評価を行った。改良アルゴリズムの精度改善評価における移動・停止を繰り返す実機実験では IR-UWB を用いた改良アルゴリズムは平均位置推定精度誤差が約 0.239m となる結果が確認できた。今後の課題として IR-UWB は遮蔽物の影響を受けやすいため実環境を想定して障害物混在環境における実機検証を行うことを検討する。

謝辞

本研究開発は総務省 SCOPE(受付番号 181507001) の委託を受けたものです。

参考文献

[1] S. Thrun; D.Hahnel; D.Ferguson; M.Montemerlo; R.Triebel; W.Bergard; C.Baker; Z.Omohundro; S.Thayer; W.Whittaker: "A system for volumetric robotic mapping of abandoned mines" IEEE International Conference on Robotics and Automation (Cat. No.03CH37422) (2003)

[2] Nic, N.: "Apple iBeacon technology briefing." Journal of Direct, Data and Digital Marketing Practice 15.3, pp.222-225 (2014).

[3] 石井真, 小暮聡, 神武直彦, 海老沼拓史: IMES (Indoor Messaging System) の原理と課題及びその解決について, GPS/GNSS Symposium 2009 テキスト, pp.120-125 (2009).

[4] Vandermeulen, D., Vercauteren, C., and Weyn, M.: Indoor localization using a magnetic flux density map of a building, In The Third International Conference on Ambient Computing, Applications, Services and Technologies, pp.42-49 (2013).

[5] Kitanouma, T., Takashima, Y., Adachi, N., Takizawa, Y.: Cloud-based Self-Organizing Localization for wireless sensor networks in mixture environments of LOS and NLOS, Proc. IEEE IWCMC 2015, pp.1230-1235 (2015).

[6] 北之馬貴正, 新居英志, 安達直世, 滝沢泰久, "SmartFinder: 大規模屋内施設における集約型自己組織化スマートデバイス位置推定方式とその評価", 情報処理学会論文誌, Vol.59, No2, pp.462-472, (2018)

[7] 森 流星, 北之馬 貴正, 川田 千尋, 滝沢 泰久: IR-UWB を用いた自己組織化ノード位置推定方式の実装評価, 研究報告マルチメディア通信と分散処理 (DPS) ,2019-DPS-178, 28,

[8] Hofmann-Wellenhof, B., Lichtenegger, H., and Collins,

J.: Global Positioning System, Theory and Practice, 4th ed. (1997).

[9] Molisch, A., Cassioli, D., Chong, CC., Emami, S., Fort, A., Kannan, B., Karedal, J., Knish, J., Schnoz, H., Siwiak, K. and Win, M.: A Comprehensive Standardized Model for Ultrawideband Propagation Channels, in IEEE Transactions on Antennas and Propagation, vol.54, no.11, pp.3151-3166 (2006).

[10] Harter, A., Hopper, A., Steggle, P., Ward, A., and Webster, P.: The anatomy of a context-aware mobile applications, Proc. ACM/IEEE MobiCom 99, Vol.8, pp.187-197 (1999).

[11] Priyantha, N., Miu, A., Balakrishnan, H., and Teller, S.: The cricket compass for context-aware mobile applications, Proc. MOBICOM 2001 (2001).

[12] Bulusu, N., Heidemann, J., and Estrin, D.: GPS-less low cost outdoor localization for very small devices, IEEE Pers. Commun., Vol.7, No.5, pp.28-34 (2000).

[13] He, C., Huang, C., M.Blum, B., A.Stankovic, J., and F.Abdelzaher, T.: Range-free localization and its impact on large scale sensor networks, ACM TECS, Vol.4, No.4, pp.877-906 (2005).

[14] Li, F., Zhao, C., Ding, G., Gong, J., Liu, C., and Zhao, F.: A reliable and accurate indoor localization method using phone inertial sensors, Proc. the 2012 ACM Conference on Ubiquitous Computing (UbiComp '12), pp.421-430 (2012).

[15] Stefan Kohlbrecher ; Oskar von Stryk ; Johannes Meyer ; Uwe Klingauf "A flexible and scalable SLAM system with full 3D motion estimation" IEEE International Symposium on Safety, Security, and Rescue Robotics (2011).

[16] Kawauchi, K. and Rekimoto, J.: FineMesh: High-Density Sampling Platform Using an Autonomous Robot, Green Computing and Communications (Green-Com), 2012 IEEE International Conference on, Besancon, pp.477-486 (2012).

[17] M. Okutomi ; T. Kanade. "A multiple-baseline stereo" IEEE Transactions on Pattern Analysis and Machine Intelligence, pp.353 - 363 (Volume: 15 , Issue: 4 , Apr 1993)

[18] iRobot Japan G.K. (<http://www.irobot-jp.com/>)