

分散型拡張カルマンフィルタを用いたマルチロボット SLAM

Multi-robot SLAM via Distributed Extended Kalman Filters

○ 笹岡 俊樹, 岸本 陽介, 鷹羽 浄嗣 (立命館大学)

T. Sasaoka, Y. Kishimoto, K. Takaba (Ritsumeikan University)

Abstract This paper is concerned with the Simultaneous Localization and Mapping (SLAM) problem with multiple mobile robots. To reduce the computational and hardware costs, the distributed extended Kalman filtering (DEKF) technique is applied to the multi-robot SLAM system. In the DEKF, each robot estimates the robots' states and the landmark positions in a distributed manner without any base stations or host robots based on the information from its neighbors. A numerical experiment demonstrates the effectiveness of the proposed method.

1 はじめに

近年、ロボット関連の技術は著しく発展しており、人間と同様に自律して働くロボットの実用化の研究が進んでいる。ロボットが自律的に行動するためには、周囲の環境と自己位置の認識が不可欠である。ロボットが自己位置推定と環境地図の生成を同時に行うことを Simultaneous Localization and Mapping (SLAM) という。近年では、複数のロボットが相互に情報を交換・補完しあうことによってより高精度な推定を行うマルチロボット SLAM の研究が活発に行われている [1].

マルチロボット SLAM は、基地局 (またはホストロボット) に各ロボットの観測値などの情報を集約する方法と基地局を設けずに各ロボットが分散協調的に推定を実行する方法に大別することができる。前者の方法では、基地局のハードウェアコストおよび計算時間が問題となる。筆者ら [2] は、データ融合の考え方に基づいた最適情報融合カルマンフィルタを利用することにより、推定精度を下げずに基地局の計算量を減らす SLAM アルゴリズムを提案した。

本研究では、後者の各ロボットが分散協調的に推定を行う方法について検討を行う。参考文献 [3, 4] ではセンサネットワークに対して、合意アルゴリズムを援用することにより、基地局を必要とせずに隣接するセンサノードからの局所的情報に基づいて分散協調的に推定値を計算する分散型カルマンフィルタが提案されている。ここでは、分散型カルマンフィルタを非線形システムに拡張し、非線形系であるマルチロボットシステムに適用することにより、分散協調型 SLAM アルゴリズムを提案する。さらに、実データを用いた数値実験により、その有効性を検証する。

2 マルチロボット SLAM

同じダイナミクスを持つ n 台のロボットと m 個のランドマークにから成るマルチロボット SLAM について考える (図 1)。このマルチロボット SLAM において、各ロボットは走行を続けながら他ロボット及びランドマークの観測を行う。

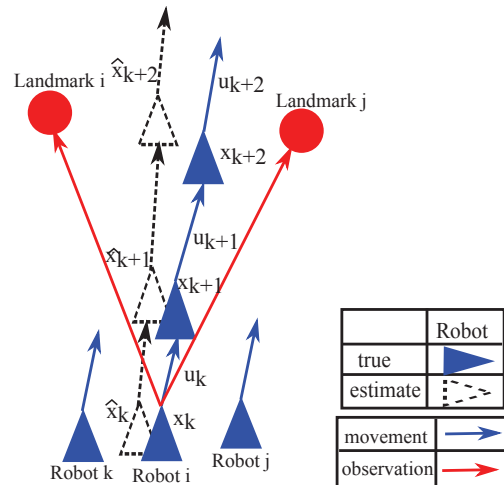


図 1: マルチロボット SLAM の概念図

移動ロボットは二輪車両モデルであるとする。連続時間系でのこのモデルの運動方程式は次式で表される。

$$\begin{bmatrix} \dot{x}_{R_i} \\ \dot{y}_{R_i} \\ \dot{\theta}_{R_i} \end{bmatrix} = \begin{bmatrix} \cos \theta_{R_i} & 0 \\ \sin \theta_{R_i} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_i \\ \omega_i \end{bmatrix}$$

ここで v_i, ω_i はそれぞれ入力速度と入力角速度を表す。オイラー法を用いて (1) 式をサンプル周期 Δt で離散化すると、ロボットの離散時間状態方程式を得る。

$$\begin{bmatrix} x_{Ri,k+1} \\ y_{Ri,k+1} \\ \theta_{Ri,k+1} \end{bmatrix} = \begin{bmatrix} x_{Ri,k} \\ y_{Ri,k} \\ \theta_{Ri,k} \end{bmatrix} + \begin{bmatrix} v_{i,k} \Delta t \cos \theta_{Ri,k} \\ v_{i,k} \Delta t \sin \theta_{Ri,k} \\ \Delta t \omega_{i,k} \end{bmatrix} \quad (1)$$

ただし, $\mathbf{x}_{Ri,k}$ は i 番目のロボットの状態を表し, x_R, y_R, θ_R はロボットの位置座標と姿勢角である. $\omega_{i,k}, v_{i,k}$ は入力角速度と入力速度, Δt はサンプリング周期である.

n 台のロボットの状態ベクトルをまとめて

$$\begin{aligned} \mathbf{x}_{R,k} &= [\mathbf{x}_{R1,k}^T \ \cdots \ \mathbf{x}_{Ri,k}^T \ \cdots \ \mathbf{x}_{Rn,k}^T]^T \\ \mathbf{x}_{Ri} &= [x_{Ri,k} \ y_{Ri,k} \ \theta_{Ri,k}]^T \end{aligned}$$

と定義する. また, 各ランドマークが $x-y$ 座標の状態を持つとすると m 個のランドマークの状態ベクトル \mathbf{L} は

$$\begin{aligned} \mathbf{L}_k &= [x_{L1,k} \ y_{L1,k} \ \cdots \ x_{Lm,k} \ y_{Lm,k}]^T \\ &= [\mathbf{p}_{L1,k}^T \ \cdots \ \mathbf{p}_{Lm,k}^T]^T \end{aligned}$$

と表される. ここで, ランドマークは移動しないので i 番目のランドマークの状態方程式は次のようになる.

$$\mathbf{p}_{Li,k+1} = \mathbf{p}_{Li,k}$$

以上より, $\mathbf{x}_k = [\mathbf{x}_{R,k}^T \ \mathbf{L}_k^T]^T$ とすれば, マルチロボット SLAM における状態方程式は次式のように書ける.

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, v_k, \omega_k) + \mathbf{w}_k \quad (2)$$

$$\begin{aligned} \mathbf{x}_{k+1} &= \begin{bmatrix} \mathbf{x}_{R,k+1} \\ \mathbf{L}_{k+1} \end{bmatrix}, \quad \mathbf{w}_k = \begin{bmatrix} w_{R,k} \\ 0_{2m \times 1} \end{bmatrix} \\ f(\mathbf{x}_k, v_k, \omega_k) &= \begin{bmatrix} f_R(\mathbf{x}_{R,k}, v_k, \omega_k) \\ \mathbf{L}_k \end{bmatrix} \end{aligned}$$

ただし, \mathbf{w}_k は共分散行列を Q_k とする平均 0 のガウス白色雑音である.

また, i 番目のロボットの観測方程式を次のように与える.

$$\mathbf{z}_{i,k} = h_i(\mathbf{x}_{R,k}, \mathbf{L}_k) + \mathbf{v}_{i,k} \quad (3)$$

ここで, $h_i(\mathbf{x}_{Ri,k}, \mathbf{L}_k)$ は観測を表す非線形関数, \mathbf{v}_k は平均 0, 共分散 $R > 0$ のガウス白色雑音である.

各ロボットはレーザスキャナのような測距センサを搭載していると仮定する. ただし簡単のため, センサは全方位測定可能で計測範囲に制限は無いとする. このとき観測関数 h_i は, (4) で与えられる.

$$h_i(\mathbf{x}_R, \mathbf{L}) = \begin{bmatrix} h^L(\mathbf{x}_{Ri}, \mathbf{p}_{L1}) \\ \vdots \\ h^L(\mathbf{x}_{Ri}, \mathbf{p}_{Lm}) \\ h^R(\mathbf{x}_{Ri}, \mathbf{x}_{R1}) \\ \vdots \\ h^R(\mathbf{x}_{Ri}, \mathbf{x}_{Rj}) \\ \vdots \\ h^R(\mathbf{x}_{Ri}, \mathbf{x}_{Rn}) \end{bmatrix} \quad (4)$$

ここで, $h_i^L(\mathbf{x}_{Ri}, \mathbf{p}_j)$ は i 番目のロボットが j 番目のランドマークを観測したときの観測関数

$$h^L(\mathbf{x}_{Ri}, \mathbf{p}_j) = \begin{bmatrix} \sqrt{\Delta x_{ij}^2 + \Delta y_{ij}^2} \\ \tan^{-1} \frac{\Delta y_{ij}}{\Delta x_{ij}} - \theta_{i,k} \end{bmatrix} \quad (5)$$

$$\Delta x_{ij} = x_{Lj} - x_{Ri}, \quad \Delta y_{ij} = y_{Lj} - y_{Ri}$$

である. また, $h^R(\mathbf{x}_{Ri}, \mathbf{x}_{Rj})$ は, i 番目のロボットが j 番目のロボットを観測したときの観測関数である.

$$h^R(\mathbf{x}_{Ri}, \mathbf{x}_{Rj}) = \begin{bmatrix} \sqrt{\Delta x_{ij}^2 + \Delta y_{ij}^2} \\ \tan^{-1} \frac{\Delta y_{ij}}{\Delta x_{ij}} - \theta_{i,k} \end{bmatrix} \quad (6)$$

$$\Delta x_{ij} = x_{Rj} - x_{Ri}, \quad \Delta y_{ij} = y_{Rj} - y_{Ri}$$

マルチロボット SLAM では, 複数のロボットがネットワークを形成し, 相互に情報伝達を行う必要がある. 本研究では, センサを通じて情報伝達が行われるものとする. すなわち, センサにより検知されたロボットとはネットワーク上で隣接するものと見なし, その位置情報を得る事ができる. 以下では簡単のため, ネットワークのトポロジーは時不変であり, 有向全域木を含むグラフで表されると仮定する.

以上のモデリングの下で, 本稿で研究するマルチロボット SLAM 問題は以下のように定式化される.

各ロボットのセンサ測定値およびロボット間の情報伝達に基づいて, 各ロボットの状態及びランドマークの位置を高精度に推定する分散協調アルゴリズムを設計せよ.

3 分散型カルマンフィルタ

各ロボットをセンサと見立てれ, 上述のマルチロボット SLAM のシステムは, マルチセンサネットワークと考えられる. 本稿では, 線形マルチセンサネットワークに対して有効な DKF[3, 4] を非線形系に拡張し, マルチロ

ポット SLAM に適用することを考える。以下では、まず線形システムに対する分散型カルマンフィルタ (DKF) について述べた後、マルチロボット SLAM 問題に適用するため、DKF を非線形システムに拡張する。

3.1 マルチセンサネットワーク推定問題

n 個のセンサノードから成るセンサネットワークを考える。目的は線形システム

$$\mathbf{x}_{k+1} = F_k \mathbf{x}_k + G_k \mathbf{u}_k + \mathbf{w}_k, \quad \mathbf{x}_0 \sim \mathcal{N}(\xi_0, \Pi_0) \quad (7)$$

の状態推定を行うことである。ここで、 $\mathcal{N}(\xi_0, \Pi_0)$ は平均 ξ_0 、共分散 Π_0 のガウス分布を表す。 i 番目のノードの観測方程式は、次式で与える。

$$\mathbf{z}_{i,k} = H_{i,k} \mathbf{x}_k + \mathbf{v}_{i,k}, \quad \mathbf{z}_i \in \mathbb{R}^p \quad (8)$$

\mathbf{u}_k は確定的な制御入力である。 \mathbf{w}_k と $\mathbf{v}_{i,k}$ はそれぞれシステム雑音と観測雑音で、平均 0 のガウス白色雑音である。雑音の共分散を以下に示す。

$$\mathbb{E}[\mathbf{w}_k \mathbf{w}_l^T] = Q_k \delta_{kl}, \quad \mathbb{E}[\mathbf{v}_{i,k} \mathbf{v}_{j,l}^T] = R_k \delta_{kl} \delta_{ij}, \quad \mathbb{E}[\mathbf{w}_k \mathbf{v}_{i,l}^T] = 0$$

ただし、 δ はクロネッカーのデルタを表す。

このセンサネットワーク上の各ノードにおいて、最適な状態推定を行う手法を述べる。集合 N_i をノード i に隣接するノードの添字集合とし、 $J_i = N_i \cup \{i\}$ とする。

最も簡便なノード間の情報伝達は、各ノードの観測値を送受することである。このとき、各ノードで実行するカルマンフィルタを局所カルマンフィルタ (Local Kalman Filter; LKF) と呼ぶ。ノード i では y_j , $j \in J_i$ が利用可能となるので、LKF のアルゴリズムは次式で与えられる [4]。

【局所カルマンフィルタ (LKF)】

初期値：

$$\bar{\mathbf{x}}_0 = \xi_0, \quad P_0 = \Pi_0 \quad (9a)$$

観測更新：

$$\hat{\mathbf{x}}_{i,k} = \bar{\mathbf{x}}_{i,k} + M_{i,k}(y_{i,k} - S_{i,k} \bar{\mathbf{x}}_{i,k}) \quad (9b)$$

$$M_{i,k} = (P_{i,k}^{-1} + S_{i,k})^{-1} \quad (9c)$$

ただし、

$$S_{i,k} = \sum_{j \in J_i} H_{j,k}^T R_{j,k}^{-1} H_{j,k} \quad (9d)$$

$$y_{i,k} = \sum_{j \in J_i} H_{j,k}^T R_{j,k}^{-1} \mathbf{z}_{j,k} \quad (9e)$$

時間更新：

$$\bar{\mathbf{x}}_{i,k+1} = F_k \hat{\mathbf{x}}_{i,k} + G_k \mathbf{u}_k \quad (9f)$$

$$P_{i,k+1} = F_k M_{i,k} F_k^T + Q_k \quad (9g)$$

この LKF アルゴリズムではネットワークポロジが完全グラフでない限り、各ノードの状態推定値が同じになる保証はない。ノード間で推定値に大きな差異が見られる場合、いくつかのノードの推定精度が相対的に低くなっており、センサネットワークに対する推定値としては妥当ではない。そこで、各ノードの推定値の間で合意を取るにより、ネットワーク上で平均的に良好な推定値を与えるアルゴリズムが必要となる。

3.2 分散型カルマンフィルタ

各ノード間では、観測値 y_j に加えて事前推定値 \bar{x}_j も伝達可能であると仮定して、LKF に合意のための修正項を付加することにより分散型カルマンフィルタ (Distributed Kalman Filter; DKF) を得る [3, 4]。

【分散型カルマンフィルタ (DKF)】

初期値：

$$\bar{\mathbf{x}}_0 = \xi_0, \quad P_{i,0} = \Pi_0 \quad (10a)$$

観測更新：

$$\hat{\mathbf{x}}_{i,k} = \bar{\mathbf{x}}_{i,k} + M_{i,k}(y_{i,k} - S_{i,k} \bar{\mathbf{x}}_{i,k}) + \epsilon M_{i,k} \sum_{j \in N_i} (\bar{\mathbf{x}}_{j,k} - \bar{\mathbf{x}}_{i,k}) \quad (10b)$$

$$M_{i,k} = (P_{i,k}^{-1} + S_{i,k})^{-1} \quad (10c)$$

ただし

$$y_{i,k} = \sum_{j \in J_i} H_{j,k}^T R_{j,k}^{-1} \mathbf{z}_{j,k} \quad (10d)$$

$$S_{i,k} = \sum_{j \in J_i} H_{j,k}^T R_{j,k}^{-1} H_{j,k} \quad (10e)$$

時間更新：

$$\bar{\mathbf{x}}_{i,k+1} = F_k \hat{\mathbf{x}}_{i,k} + G_k \mathbf{u}_k \quad (10f)$$

$$P_{i,k+1} = F_k M_{i,k} F_k^T + Q_k \quad (10g)$$

ここに、(10b) 式右辺第 3 項が推定値の合意のための項であり、 ϵ は正定数である。

参考文献 [4] では (10) 式の DKF アルゴリズムの収束性を示すにあたり、連続時間線形システム

$$\dot{\mathbf{x}} = F \mathbf{x} + G \mathbf{u} + \mathbf{w}, \quad \mathbf{x}_0 \sim \mathcal{N}(\xi_0, \Pi_0) \quad (11a)$$

$$\mathbf{z}_i = H_i \mathbf{x} + \mathbf{v}_i \quad i = 1, 2, \dots, n \quad (11b)$$

に対する連続時間分散型カルマンフィルタ

$$\dot{\hat{\mathbf{x}}}_i = F\hat{\mathbf{x}}_i + G\mathbf{u} + K_i(\mathbf{z}_i - H_i\hat{\mathbf{x}}_i) + \epsilon P_i \sum_{j \in N_i} (\hat{\mathbf{x}}_j - \hat{\mathbf{x}}_i) \quad (12a)$$

$$K_i = P_i H_i^T R_i^{-1}, \quad \epsilon > 0 \quad (12b)$$

$$\dot{P}_i = F P_i + P_i F^T + Q - K_i R_i K_i^T \quad (12c)$$

$$P_i(0) = \Pi_0, \quad \hat{\mathbf{x}}(0) = \xi_0 \quad (12d)$$

について、推定値 $\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_n$ の合意が達成されることが示されている。

命題 1 [4]. (12) 式の連続時間分散型カルマンフィルタにおいて、各ノードの推定誤差ダイナミクスは漸近安定である。さらに雑音が存在しない場合、各ノードの状態推定値の合意が漸近的に達成される。

3.3 非線形システムへの拡張

前章の DKF アルゴリズムを非線形システム

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k, \quad \mathbf{x}_0 \sim \mathcal{N}(\xi_0, \Pi_0) \quad (13a)$$

$$\mathbf{z}_{i,k} = h_i(\mathbf{x}_k) + \mathbf{v}_{i,k}, \quad i = 1, 2, \dots, n \quad (13b)$$

に拡張し、マルチロボット SLAM へ適用可能なアルゴリズム、すなわち分散型拡張カルマンフィルタ (Distributed Extended Kalman Filter; DEKF) アルゴリズムを導く。

集中型拡張カルマンフィルタでは、各時刻で状態空間モデルを線形化する。その際、状態遷移関数 f と観測関数 h の状態推定値まわりのヤコビ行列を計算しなければならない [5]。DEKF でも同様に、各ノード i で f および $h_j, j \in J_i$ のヤコビ行列を計算するが、基準となるはノード i 自身における推定値 $\bar{\mathbf{x}}_{i,k}, \hat{\mathbf{x}}_{i,k}$ であることに注意が必要である。すなわち、DEKF アルゴリズムでは、DKF の $F_k, H_{j,k}$ の代わりに次のヤコビ行列を用いる。

$$F_{i,k} = \left. \frac{\partial f(x, u)}{\partial x} \right|_{(x, u) = (\hat{\mathbf{x}}_{i,k}, \mathbf{u}_k)}, \quad H_{j,k}^i = \left. \frac{\partial h_j(x)}{\partial x} \right|_{x = \bar{\mathbf{x}}_{i,k}}$$

したがって、提案する DEKF アルゴリズムは、以下のように与えられる。

【分散型拡張カルマンフィルタ (DEKF)】

初期値：

$$\bar{\mathbf{x}}_{i,0} = \xi_0, \quad P_{i,0} = \Pi_0 \quad (14a)$$

観測更新：

$$\hat{\mathbf{x}}_{i,k} = \bar{\mathbf{x}}_{i,k} + M_{i,k} \left(y_{i,k} - \sum_{j \in J_i} H_{j,k}^T R_j^{-1} h_j(\bar{\mathbf{x}}_{i,k}) \right) + \epsilon M_{i,k} \sum_{j \in N_i} (\bar{\mathbf{x}}_{j,k} - \bar{\mathbf{x}}_{i,k}) \quad (14b)$$

$$M_{i,k} = (P_{i,k}^{-1} + S_{i,k})^{-1} \quad (14c)$$

ただし、

$$y_{i,k} = \sum_{j \in J_i} H_{j,k}^T R_j^{-1} z_{j,k} \quad (14d)$$

$$S_{i,k} = \sum_{j \in J_i} H_{j,k}^T R_j^{-1} H_{j,k} \quad (14e)$$

時間更新：

$$\bar{\mathbf{x}}_{i,k+1} = f(\hat{\mathbf{x}}_{i,k}, \mathbf{u}_k) \quad (14f)$$

$$P_{i,k+1} = F_{i,k} M_{i,k} F_{i,k}^T + Q_k \quad (14g)$$

4 数値実験

提案法の有効性を確認するため、数値実験を行った。実験には 3 台のロボットと 3 個のランドマークを使用した。図 2 に実験環境を示す。各ロボットはレーザーレンジファインダを用いて観測を行い、ロボット 1 からはロボット 3 が、ロボット 3 からはロボット 1 が、それぞれ観測不可能である。表 4,5 に使用したロボット及びレーザーレンジファインダの仕様を示す。各ロボットは y 軸方向に 30 秒間前進し、その後、後退するように入力を与えた。表 1 に実験の設定値を示す。

ロボットの CPU 能力の限界のため、DEKF の実行は、PC 上の MATLAB にてオフラインで実行した。使用した PC の仕様は、OS: Windows7, CPU: Intel Core i5 3.2GHz, メモリ: 8GB である。

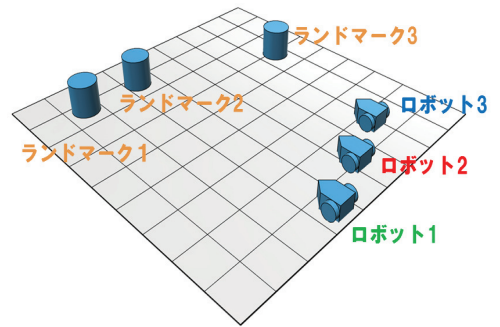


図 2: 実験配置図

図 3 は各ロボットが推定した全ロボットの位置の時刻毎の推定誤差 e_R を示しており、図 4 は各ロボットが推定した全ランドマークの位置の推定誤差 e_L を示している。

$$e_{Ri} = \sqrt{(\hat{x}_{Ri} - x_{Ri})^2 + (\hat{y}_{Ri} - y_{Ri})^2} \quad i = 1, 2, 3$$

$$e_{Li} = \sqrt{(\hat{x}_{Li} - x_{Li})^2 + (\hat{y}_{Li} - y_{Li})^2} \quad i = 1, 2, 3$$

ただし、 x_{Ri}, y_{Ri} は i 番目のロボットの x 座標, y 座標であり、 $\hat{x}_{Ri}, \hat{y}_{Ri}$ はその推定値である。同様に、 x_{Li}, y_{Li}

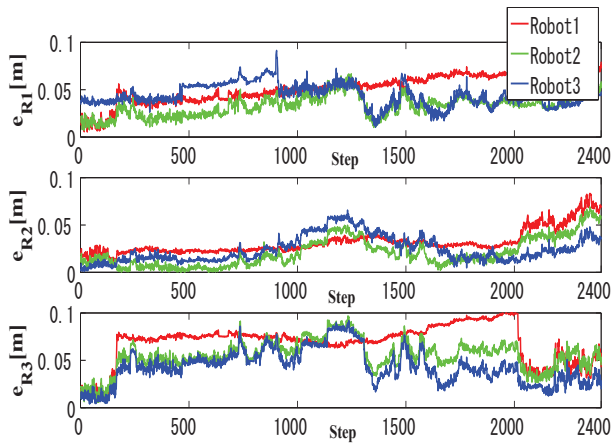


図 3: ロボット位置の推定誤差

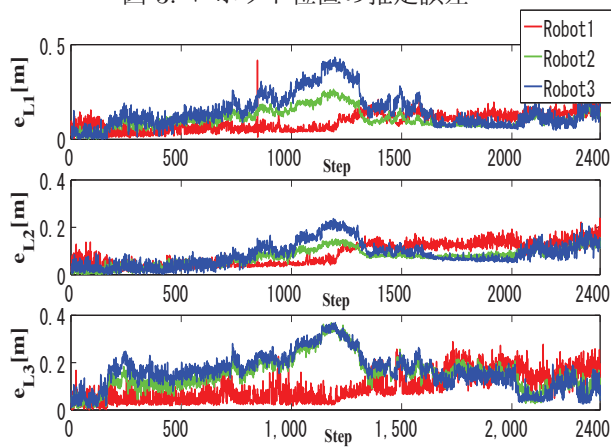


図 4: ランドマーク位置の推定誤差

は i 番目のランドマークの x 座標, y 座標であり, $\hat{x}_{Li}, \hat{y}_{Li}$ はその推定値である.

図 4 から, ロボットの運動が大きく変化する時点 (開始 30s 後) 付近では, それぞれのロボットの推定誤差が大きくなるが, 最終的には誤差がほぼ等しくなり, 合意が取れていることが確認できる.

表 2 と 3 は, DEKF と局所拡張カルマンフィルタ (LEKF) の, 推定時間全体における推定誤差を RMS 誤差で示している. LEKF に比べ, DEKF ではロボットの軌跡に関しては数 cm 程度誤差が大きいが, ランドマークに関してはほぼ全てで 10cm 程度誤差を軽減できたことが確認できる.

計算時間について, LEKF では全てのロボットが 0.30 [ms/step] 程度であったのに対し, DEKF ではロボット 1 とロボット 3 は 0.47 [ms/step], ロボット 2 は 0.62 [ms/step] であった. DEKF では隣接するロボットの台数が多くなるほど, 計算時間が延びることが予想できる.

また, 先行研究 [2] の基地局で推定値を統合する方法では, 各ロボットの計算時間に基地局の計算時間 0.306 [ms/step] を足し合わせるため, 合計で 0.606 [ms/step] だけ計算時間がかかる. また, 通常の集中型 EKF では基地局で全ての計算を行うため 0.738 [ms/step] にかかる. よって, 計算時間の観点では, 提案法は文献 [2] の方法よりも有効であると言える.

5 おわりに

本研究では, 分散型カルマンフィルタ [4] を非線形システムに拡張した分散型拡張カルマンフィルタ (DEKF) を導出し, DEKF を用いたマルチロボット SLAM アルゴリズムを提案した. 実データを用いた数値実験により, 提案法の有効性を検証した.

今後の課題として, 最適性及び収束性の理論的検討も含めた DKF および DEKF アルゴリズムの改良がある. また, DEKF を用いたマルチロボット SLAM において, 各ロボットにおける計算量の低減もまた課題の 1 つである.

参考文献

- [1] S. Saeedi, M. Trentini, M. Seto, and H. Li: "Multi-robot simultaneous localization and mapping: A review," *J. of Field Robotics*, vol. 33, no. 1, pp. 3-46, 2016.
- [2] T. Sasaoka, I. Kimoto, Y. Kishimoto, K. Takaba, and H. Nakashima: "Multi-robot SLAM via Information Fusion Extended Kalman Filters," *Proc. of 6th IFAC Workshop on Distributed Estimation and Control in Networked Systems (NECSYS2016)*, pp.303-308, 2016.
- [3] R. O. -Saber: "Distributed Kalman filter with embedded consensus filters," *Proc. of 44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference (CDC-ECC '05)*, pages 8179-8184, 2005.
- [4] R. O. -Saber: "Distributed Kalman Filtering for Sensor Networks," *Proc. of 46th IEEE Conference on Decision and Control*, pp. 5492-5498, 12-14, 2007.
- [5] 片山: 非線形カルマンフィルタ, 朝倉書店, 2011.

表 1: 設定値

実験時間	60[s]
サンプリング周期	$\Delta t = 0.025$ [s]
ロボットの初期値	$(x_{1,0}, y_{1,0}) = (-61, 30.5), (x_{2,0}, y_{2,0}) = (0, 0), (x_{3,0}, y_{3,0}) = (61, 30.5)$ [cm] $\theta_0 = \pi/2$ [rad]
ランドマークの位置	$(-183, 335.5), (-91.5, 335.5), (122, 244)$ [cm]
ロボットの速度	$v = 5.5$ [cm/s]($t = 0 \sim 30s$) $v = -5.5$ [cm/s]($t = 30 \sim 60s$)
ロボットの角速度	$\omega = 0$ [rad/s]
ロボットの視野	正面を中心に左右 135° ずつ 270°
重み	$\epsilon = 0.025$
プロセス雑音の共分散	$x, y = 1.0 \times 10^{-4}$ [cm], $\theta = 1.0 \times 10^{-2}$ [$^\circ$]
観測雑音の共分散	$r = 1.6 \times 10^{-5}$ [cm], $\phi = 1.6 \times 10^{-2}$ [$^\circ$]
推定誤差共分散の初期値	$P_{ij} = O_{15}(i \neq j), P_{ii} = I_{15}$

表 2: RMSE(DEKF) [mm],[$^\circ$]

RMS Errors	Landmark1XY	Landmark2XY	Landmark3XY	Robot1XY	Robot1 θ	Robot2XY	Robot2 θ	Robot3XY	Robot3 θ
Robot1	74.1194	68.5877	100.5274	33.8367	0.5264	25.4595	1.8522	83.9131	0.4891
Robot2	118.7813	86.0795	167.5418	40.3370	0.8541	29.3260	1.4431	54.8002	2.0840
Robot3	164.5549	100.9139	183.8171	58.3205	0.4636	38.4068	1.4783	48.7184	1.4943

表 3: RMSE(LEKF) [mm],[$^\circ$]

RMS Errors	Landmark1XY	Landmark2XY	Landmark3XY	Robot1XY	Robot1 θ	Robot2XY	Robot2 θ	Robot3XY	Robot3 θ
Robot1	225.6740	206.2046	191.4172	18.0424	1.4886	14.2840	0.0970	-	-
Robot2	265.5475	229.4027	174.8148	34.9569	0.2432	22.0832	2.4300	38.7899	0.3633
Robot3	227.2787	192.1253	111.0759	-	-	17.2199	0.0662	21.2405	0.7909

表 4: Koala2.5(K-TEAM 製) の仕様

OS	Ubuntu
車体寸法	長さ 320[mm]
	幅 320[mm]
	高さ 145[mm]
車体重量	4.5[kg]
駆動輪	右 : 3 輪 左 : 3 輪
車輪径	41.2[mm]
車軸長	280[mm]

表 5: LRF (HOKUYO UTM-30LX) の設定

最大検出距離	30[m]
測定範囲角	180[$^\circ$]
角度分解能	0.25[$^\circ$]
走査時間	25[ms/scan]