

惑星探査機の誘導法と UKF-SLAM による自己位置推定

Guidance Method and Localization using UKF-SLAM for Space Rover

○平野智太¹, 澁谷浩平², 内山賢治³*Tomohiro Hirano¹, Kohei Shibuya², Kenji Uchiyama³

Abstract : We introduce the design of a system for a space rover. Simultaneous localization and mapping (SLAM) technique is useful for planetary exploration without GPS. We combine a reliable estimation ability of unscented Kalman filter (UKF) and SLAM. The superior performance of UKF-SLAM is validated by numerical simulation.

1. 諸言

従来の惑星探査ローバの自己位置推定法では、惑星を周回する人工衛星及び探査ローバから得られる二種の画像データを照合し自己位置推定を行っている。しかし、この手法では人工衛星からの情報を受信できない環境下において、自己位置推定を行うことが困難である。この問題に対し、探査ローバのみでミッションを行うことができる SLAM (Simultaneous Localization and Mapping) という手法が有効に働く。SLAM とは探査ローバに搭載されたセンサ情報から自己位置推定と地図構築を同時に行う手法である。このとき、センサ情報に混入したノイズにより実際の位置と観測値の間に差異が生じる¹⁾。この影響を抑えるために幾つかの有効な方法があるが、物体の非線形な運動を推定することが可能な手法として EKF (Extended Kalman Filter)がある。しかし、EKF-SLAM では多数のランドマークを避けて誘導する際に推定誤差が大きくなる問題が見られる。そこで、生じる推定誤差を小さくするために UKF (Unscented Kalman Filter)を適用する手法を提案する。

本稿では、未知の環境下における探査ローバの誘導ミッションを仮定し、誘導則に計算負荷の軽いポテンシャル関数誘導法を用いて、目的地への誘導を試みる。最後に数値シミュレーションを行い、提案手法の有効性を確認する。

2. 探査ローバの誘導

2.1. UKF-SLAM

SLAM は、GPS のような絶対的な位置情報を得ることができない遠方の惑星等を想定した環境で、正確な自己位置推定と地図構築を行うための手法である。本研究では、UKF を適用した SLAM により自己位置推定を行う。Fig.1 に SLAM の概念図を示す。

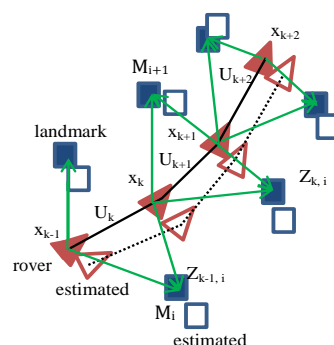


Figure 1. Structure of SLAM

- \mathbf{x}_k : 時刻 k における探査ローバの状態ベクトル
- \mathbf{U}_k : 状態 \mathbf{x}_k へ遷移するために時間 $(k-1)$ で探査ローバに与える制御ベクトル
- \mathbf{M} : ランドマークの位置ベクトル
- \mathbf{Z} : ランドマークの推定位置

SLAM は地図情報 m や観測値の時系列 \mathbf{z}_k が得られる時の探査ローバ位置 \mathbf{x}_k の確立密度 $p(\mathbf{x}_k | \mathbf{z}_{1:k}, \mathbf{U}_{1:t}, m)$ の推定問題として次式のように表される²⁾。

$$p(\mathbf{x}_k, m | \mathbf{z}_{1:k}, \mathbf{U}_{1:k}) = \alpha p(\mathbf{z}_k | \mathbf{x}_k, m) \times \int p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{U}_k) p(\mathbf{x}_{k-1}, m | \mathbf{z}_{1:k-1}, \mathbf{U}_{k-1}) d\mathbf{x}_{k-1} \quad (1)$$

これは漸化式であり、UKF を用い、状態量ベクトルとランドマークの状態量をまとめて一つの状態量 \mathbf{X} として扱うことにより実装することができる。UKF を適用することで状態量の推定値 $\hat{\mathbf{X}}_k$ は次式のように計算される。

$$\hat{\mathbf{X}}_k = \hat{\mathbf{X}}_k^- + \mathbf{g}_k \{ \mathbf{y}_k - \hat{\mathbf{y}}_k^- \} \quad (2)$$

ここで \mathbf{g}_k はカルマンゲインであり、 \mathbf{y}_k は観測値である。また $\hat{\cdot}^-$ は事前推定値を表している。

2.2. 誘導則

本研究では、誘導則に仮想的なポテンシャル場の勾配を用いて機体を誘導するポテンシャル関数誘導法を適用する。この手法は、障害物に高ポテンシャル場を、目的地に低ポテンシャル場を設定することで、障害物を回避しながら目的地へと誘導を行う^[4]。

ポテンシャル場は誘導ポテンシャル U^S と反発ポテンシャル U_j^R の和として表される。 \mathbf{x} は位置ベクトルを表し、 \mathbf{x}_j は障害物の位置ベクトルを表す。また $\mathbf{x}_{ij} = |\mathbf{x} - \mathbf{x}_j|$ は機体と障害物間の相対距離とする。探査ローバの誘導則には、ポテンシャル関数から導かれる速度場を利用する。 x 方向と y 方向の速度指令値は、それぞれ次式のように導出される。

$$V_x = -\frac{\partial U^S(\mathbf{x})_x}{\partial x} - \frac{\partial U_j^R(\mathbf{x}_{ij})_x}{\partial x} \quad (3)$$

$$V_y = -\frac{\partial U^S(\mathbf{x})_y}{\partial y} - \frac{\partial U_j^R(\mathbf{x}_{ij})_y}{\partial y} \quad (4)$$

以上より、速度指令値 V_c と指令方位角 φ_c は以下のよう
に与えられる。

$$V_c = \sqrt{V_x^2 + V_y^2} \quad (5)$$

$$\varphi_c = \tan^{-1}\left(\frac{V_y}{V_x}\right) \quad (6)$$

3. 数値シミュレーション結果

提案する手法の有効性を検証するために数値シミュレーションを行う。 Fig.2 はポテンシャル関数誘導法に UKF-SLAM を適用し、探査ローバの誘導を行ったものである。

Fig.2 中の実線は探査ローバの軌跡、*はランドマークを表している。また、赤は実際の位置、青は推定位置を表している。 Fig.3 は探査ローバが各時刻に実際の位置と推定位置にどれだけ誤差が生じているか、EKF-SLAM と UKF-SLAM のそれぞれの結果を表している。

Fig.2 より、提案手法で探査ローバを目的地に誘導できていることが確かめられる。また、 Fig.3 から EKF-SLAM での推定値に比べて UKF-SLAM での推定値の誤差は 4 分の 1 程度になっていることが見て取れる。このような差が生じた原因としてポテンシャル関数誘導法によって得られる探査ローバの制御入力急激に変化したため、運動の非線形性が強くなったことが考えられる^[4]。

4. 結言

探査ローバの誘導における自己位置推定問題の内、SLAM に EKF を適用した際に生じる誤差の問題に対して UKF を適用した手法を提案し、数値シミュレーションにより検証を行った。その結果、複数のランドマークを避けながら目的地に誘導する必要性があるミッションでは EKF-SLAM に比べ UKF-SLAM の方が有効に働いていると考えられる。

参考文献

- [1] 森本祐介, 正滑川徹, “拡張カルマンフィルタを用いた移動ロボットの自己位置推定と環境認識”, 日本機械学会運動と振動の制御シンポジウム講演論文集, pp.2-3, 2009.
- [2] 友納正裕, “移動ロボットのための確立的な自己位置推定と地図構築”, 日本ロボット学会誌, Vol.29, No.5, pp.423-426, 2011.
- [3] 鈴木真之, 内山賢治, Derek Bennet, and Colin R. McInnes, “速度場に分岐理論を適用した UAV の 3 次元フォーメーションフライト”, 日本航空宇宙学会論文集, Vol.59, No.693, pp.259-265, 2011.
- [4] 足立修一, 丸田一郎, “カルマンフィルタの基礎”, 東京電機大学出版局, pp152-190, 2012

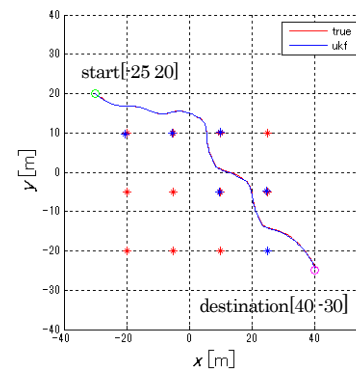


Figure 2. Trajectory of the rover by UKF

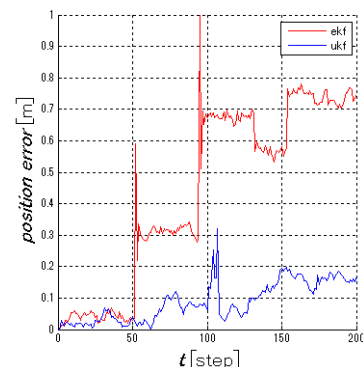


Figure 3. Time history of position error