

## 1. はじめに

近年、品種改良によるブランド化や、メルカリなどの誰でも手軽に出品することができるフリマアプリの登場により、農作物の盗難被害が増加している。農林水産省が公表している令和元年の農作物盗難に関するデータによると、圃場での盗難が多発しており、屋外で栽培されている農作物が狙われやすいことがわかる [1]。これに対して、監視カメラやドローン、屋外移動ロボットなどを活用した盗難対策が求められている。

これまで GNSS や SLAM を用いたロボットの走行制御は広く研究されてきたが、GNSS による制御の場合、木の下や建物の近くなど、GNSS の受信環境が悪い場所では位置推定精度が低下してしまうという課題がある。一方 SLAM による制御の場合、特徴的な物体 (木や建物など) が存在する環境では高精度な位置推定が可能であるが、開けた場所では特徴量が不足し、精度が低下してしまうという課題がある。このように GNSS と SLAM にはそれぞれ異なる弱点が存在するが、これらを補完的に活用する方法の研究は限られている。

そこで、本研究では GNSS と SLAM の弱点を補完し合う形で切り替えを行い、圃場のような多様な環境において安定したロボットの走行制御の実現を目指す。具体的には、受信環境が良好な場合は GNSS による制御を行い、受信環境が悪化した場合には SLAM による制御に切り替えることで屋外のどの経路でも安定した経路追従が可能となる走行制御の実現を目指し、非 GNSS 下における圃場巡回ロボットの自律走行制御システムの構築を目的とする。

## 2. SLAM による自律走行制御

### 2.1 テンプレートマッチング法

Fig.1 のように、測定された周囲の状況の測定点群において、測定元となるセンサの位置を中心とする  $100 \times 100\text{cm}$  の範囲を設定する。Fig.2 にテンプレートマッチングの様子を示しており、設定された範囲の左上の位置にテンプレートの中心を設置し、その上でテンプレートを  $2^\circ$  ずつ  $-60^\circ \sim 60^\circ$  回転させ、テンプレートと測定点群の類似度を求める。その後、設定された範囲内において  $2\text{cm}$  ずつテンプレートの中心とする点をずらしていき、テンプレートを設置するごとにテンプレートを  $2^\circ$  ずつ  $-60^\circ \sim 60^\circ$  回転させ、テンプレートと測定点群の類似度を求めるという作業を繰り返す。範囲内でテンプレートマッチングが終了後、最も類似度が高い場合の位

置と角度をロボットの位置推定結果とする。本研究では  $1\text{m}$  進むごとにテンプレートマッチングによる自己位置推定を行うことで、 $1\text{m}$  間隔でロボットの軌道修正ができるようにする。

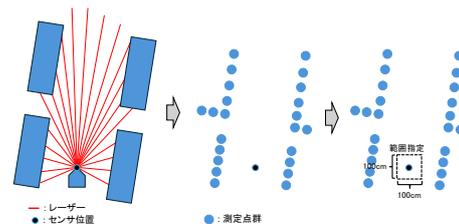


Fig. 1 範囲指定イメージ



Fig. 2 テンプレートマッチング

### 2.2 類似度の定義

テンプレートマッチングにおける類似度について述べる。まず、座標の定義について、測定点群  $P$  に属する測定点  $p_i$  についてそれぞれの座標は

$$p_i = (x_i, y_i) \quad (1)$$

と表す。また、テンプレートを設定した範囲で位置  $g$  に配置し、角度  $\theta$  だけ回転させた時の点群  $Q$  は

$$q_i = (\hat{x}_i^{g, \theta}, \hat{y}_i^{g, \theta}) \quad (2)$$

と定義する。このとき、測定点群  $P$  とテンプレート  $Q$  の類似度  $\text{sim}(P, Q)$  を以下のように定義する。

$$\text{sim}(P, Q) = \sum_{i=1}^n \delta(p_i, q_i) \quad (3)$$

$$\delta(p_i, q_i) = \begin{cases} 1 & \text{if}(\|q_i - p_i\| \leq 2[\text{cm}]) \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

類似度  $\text{sim}(P, Q)$  は  $1$  から  $n$  までの  $q_i$  から  $p_i$  までの距離が  $2\text{cm}$  以上であった数の総和を表しており、この総和の算出を設定した範囲内で移動と回転をさ

せながら繰り返し行う。そして設定した範囲で繰り返し作業が終わった際に、類似度  $sim(P, Q)$  が最大となる位置と角度を位置推定結果に決定する。

### 3. ロボット制御則

目標経路の作成イメージを Fig.3 に、ロボットの経路追従制御イメージを Fig.4 に示す。目標経路を作成する際は目標経路上に 1m 間隔で基準となる位置を設定し、それぞれの位置に対応するテンプレートを事前にセンサデータから取得して保存しておく。走行中はロボットが 1m 進むごとに事前に取得した対応するテンプレートを用いて位置と角度の推定を行う。その推定結果に基づき、ロボットの軌道を適切に修正することで、目標経路に正確に追従しながら走行することが可能となる。

目標経路への追従は位置推定結果の  $\hat{y}$ ,  $\hat{\theta}$  を利用する。位置推定結果の値は事前に取得したテンプレートと走行中に取得されるスキャンデータのずれを表している。つまり、位置推定結果  $(\hat{x}, \hat{y}, \hat{\theta}) = (0, 0, 0)$  の場合、スキャンデータとテンプレートとのずれがないことを意味する。本研究ではロボットが目標経路上のテンプレートに沿って正確に走行することを目的としているため、走行制御をする際は位置推定結果の値が 0 に近づくようにする。

したがって、右車輪の速度  $V_R$ , 左車輪の速度  $V_L$  を求める式は位置推定結果  $\hat{y}$ ,  $\hat{\theta}$  の値を用いると

$$\omega = K_y(-\hat{y}) + K_\theta(-\hat{\theta}) \quad (5)$$

$$V_R = V + \frac{W}{2}\omega \quad (6)$$

$$V_L = V - \frac{W}{2}\omega \quad (7)$$

となる。ここで  $\omega$  は回転速度、 $K_y$  および  $K_\theta$  はそれぞれ  $\hat{y}$ ,  $\hat{\theta}$  に対応するゲイン、 $V$  は直進速度、 $W$  は車体幅を表す。また、 $K_y$  および  $K_\theta$  の値は任意に設定する。

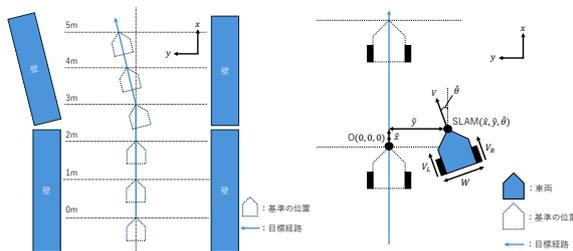


Fig. 3 目標経路作成

Fig. 4 経路追従制御

## 4. 実機実験

### 4.1 実験概要

Fig.5 に実験風景、Fig.6 に走行経路を示す。非 GNSS 環境を想定した目標経路を作成し、実機を用いて走行実験を実施する。この実験によって非 GNSS 環境下でもロボットが目標経路に追従しながら走行できるかを確認する。



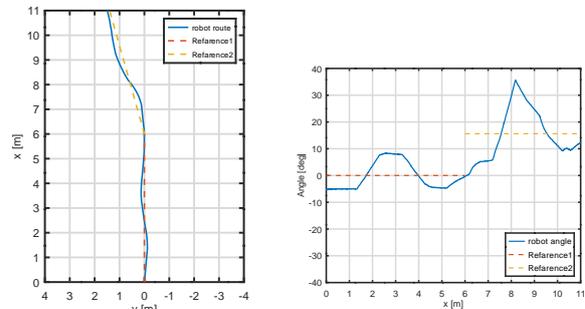
Fig. 5 実験風景



Fig. 6 走行経路

### 4.2 実験結果

Fig.7(a) にロボットの走行経路を、Fig.7(b) にロボットの角度の推移を示す。Fig.7(a) より、ロボットは目標経路に沿うように走行しており、左に曲がった後も経路に沿って走行していることが確認できる。また、Fig.7(b) より、ロボットの角度が目標角度に向かって振動しながら収束するよう調整されている様子が確認できる。実験結果より、非 GNSS 環境下を想定した経路であっても、ロボットが目標経路に沿って安定して走行できることが確認できた。



(a) 走行経路

(b) 角度の推移

Fig. 7 実験結果

## 5. まとめ

本研究では圃場のような多様な環境において安定したロボットの自律走行制御の実現を目指し、非 GNSS 下における圃場巡回ロボットの自律走行制御システムを実装した。結果として、ロボットは作成した目標経路に沿って正確に走行することができ、非 GNSS 下の環境であっても自律走行が可能であることを確認した。今後は GNSS による走行制御と SLAM による走行制御を切り替えるシステムを実装し、実際の圃場などの実環境での走行精度や経路追従性能を評価、検証する必要がある。

## 参考文献

- [1] 農林水産省生産局園芸作物課 (2019) 「農作物の盗難の実態と対応策」 <https://www.maff.go.jp/j/seisan/ryutu/engei/tounan.html> (参照日 2025 年 1 月 14 日)
- [2] 見玉泰伸, 増本雅之, 神崎僚太, 留森賢一, 松井敦史: 「Lidar SLAM 技術による ICT フォワードと小型 UGV の森林作業道自動走行の実現」, 森林利用学会誌, Vol. 37, No. 4, pp.193-200 (2022)
- [3] 有水賢吾: 「測域センサおよび RGBD カメラを用いた森林内自己位置推定」, 森林利用学会誌, Vol.36, No.1 pp.63-68 (2021)