混雑環境における移動検問ロボットの開発およびOR手法の応用の検討

佐久間　大，辻田　哲平，冨沢　哲雄

不特定多数の人々が往来する混雑環境において，公衆衛生の確保およびテロリズムの防止の観点から，健康状態や不審物の検査を実施する移動・検問ロボットが必要とされている．本稿では，移動・検問ロボットの開発過程において，OR 手法の応用の検討を行う．その初期段階として，ここでは，ロボットの最適な移動経路を算出するにあたり，整数計画問題による移動経路の導出を行い，その結果を実ロボットに指示するための枠組みを与える．しかしながら，実環境におけるロボットの移動性能については，現実に抱える問題・課題は多いため，それについても述べる．

キーワード：移動検問ロボット，最適移動経路，整数計画問題，ROS，実ロボットの移動制約

1. はじめに

人が密集しやすい市街地や重要施設の周辺では，公衆衛生の確保やテロリズムの防止を目的として，通行する人々に対して健康状態や所持品の確認などが行われることがある．多くの場合，市街地を往来する人々の動線を制約することはできず，検査対象者の行動範囲は広範囲にわたる．ドローンやカメラなどにより広域を観測する方法では，全体を大まかに確認することは期待できるものの，個別の検査精度が十分とはいえない．そのため，各個人に対する精密な検査を実施するためには，十分に接近した状態での検査も必要である．しかしながら，特に危険物の所持が疑われる不審者に対して，人自身が行う検査は危険性を伴う．以上のことから，人々が往来する混雑環境において人の代わりに検問を実施する移動検問ロボットの開発が重要である．

本稿では筆者らが，混雑環境で動作する移動検問ロボット（以下，ロボット）の開発を行ってきた中で，オペレーションズ・リサーチ (OR) 手法の活用について検討してきた一例を紹介する．ここでは特に，多くの歩行者に対する追跡および検問を目的とした下で，それを実現するためのロボットの最適移動経路を，整数計画問題の解として導き（2 節），その解（移動経路）をロボットへ実装するための枠組みである ROS（Robot Operating System [1, 2], ロボット開発に用いるオープンソースソフトウェア）について説明する（3 節）．ここで，OR 手法適用の検討段階は，ロボットの開発における初期段階であり，実ロボットが直面する制約条件（ロボットの移動性能，計測センサー機器の誤差，歩行者の動きの予測など）はかなり複雑である．そのため，本稿の最後（4 節）では，ロボットの移動および関連するセンサ技術が抱える現実問題について述べる．

さくま　ゆたか，つじた　てっぺい
防衛大学校情報工学科
〒 239–8686　神奈川県横須賀市走水 1–10–20
sakuma@nda.ac.jp
t.tsujita@ieee.org
とみざわ　てつお
東京工業高等専門学校機械工学科
〒 193–0997　東京都八王子市椚田町 1220–2
tomizawa@tokyo-ct.ac.jp

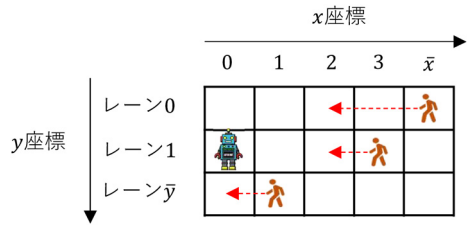

図 1　移動検問ロボットの離散時間モデル

2. OR 手法に基づく最適移動経路の導出

ここでは，複数の歩行者が離散セル領域において等速直線運動する下で，ロボットができるだけ多くの歩行者に接近するための移動経路を整数計画問題の解として導く．モデルの仮定は以下のとおりである．有限セル空間 $\mathbb{X}\times\mathbb{Y}$ ($\mathbb{X}=\{0,1,\ldots,\overline{x}\}$,$\mathbb{Y}=\{0,1,\ldots,\overline{y}\}$) 上に，1 台のロボット（初期位置 $(x_0,y_0)\in\mathbb{X}\times\mathbb{Y}$）と複数の歩行者が存在する離散時間モデルを考える（図 1）．ここで，$\mathbb{X}\times\{y\}(y\in\mathbb{Y})$ をレーン y と呼び，各レーン $y\in\mathbb{Y}$ には 1 人の歩行者（初期位置 $(\hat{x}_y,y)$）が存在し，その歩行者は $(0,y)$ へ向かい速度 $v_y(v_y\in\{1,2,\ldots,\overline{v}\})$ で等速直線運動する．ロボットがセル (x,y) から (x',y') への移動に要する最短時間を $d((x,y),(x',y'))$ で表す．モデルの離散時間軸を $\mathbb{T}=\{0,1,\ldots,\overline{x}\}$ で表し，ある

時点 $t \in \mathbb{T}$ において，ロボットと歩行者が同一セルに位置したとき，検問は成功したものとみなす（つまり，検問は 1 単位時間で済む）．

2.1 整数計画問題による定式化

ロボットの最適移動経路を求めるための，整数計画問題による定式化に関する表記について述べる．これ以降，用語「位置」は主に時点とセルからなる組を指し，特に位置 $a \in \mathbb{S} = \mathbb{T} \times \mathbb{X} \times \mathbb{Y}$ を $a = (a_T, a_X, a_Y)$ で表記する．位置 $a \in \mathbb{S}$ で歩行者を検問したときの利得を p_a（これは，歩行者の初期位置および速度に依存して決まる），レーン $y \in \mathbb{Y}$ において正の利得が得られる位置集合を $\mathbb{P}_y = \{a \in \mathbb{S} \mid a_Y = y, p_a > 0\}$ で表す．また，時点 $t \in \mathbb{T}$ で利得が得られる位置集合を $\mathbb{Q}_t = \{a \in \mathbb{S} \mid a_T = t, p_a > 0\}$ で表す．

次に，ロボットの移動経路を決定するための変数を以下に述べる．位置 $a \in \cup_{y \in \mathbb{Y}} \mathbb{P}_y$ の歩行者への検問実施の有無（1 もしくは 0）を決める変数を $h_a \in \{1, 0\}$ で表す，ここで，同一歩行者に対する重複検問を除外するために，

$$0 \leq \sum_{a \in \mathbb{P}_y} h_a \leq 1, \quad y \in \mathbb{Y} \tag{1}$$

とする．さらに，ロボットの初期位置を考慮し，

$$h_{(0,x,y)} = \begin{cases} 1, & (0,x,y) \in \mathbb{Q}_0 \cap \{(0, x_0, y_0)\} \\ 0, & (0,x,y) \in \mathbb{Q}_0 \setminus \{(0, x_0, y_0)\} \end{cases} \tag{2}$$

とする．つまり，初期時刻において，ロボットが歩行者に検問できるのは，歩行者がロボットの初期位置にいる場合に限ることを意味する．表記を簡単にするために，時点 $t \in \mathbb{T}$ における検問の有無を表す変数 $H_t \in \{1, 0\}$ を用意し，

$$\sum_{a \in \mathbb{Q}_t} h_a = H_t, \quad t \in \mathbb{T} \tag{3}$$

とする．

最後に，ロボットの移動計画を決める条件を述べる．いま，時点 $t \in \mathbb{T}$ に位置 $a \in \mathbb{Q}_t$ で検問し，それから $s \geq 1$ 時間後（つまり，時点 $t + s$）に初めて位置 $b \in \mathbb{Q}_{t+s}$ で検問することを考える．この際，次の制約を満たす必要がある：

$$\begin{aligned} &d((a_X, a_Y), (b_X, b_Y)) h_b \\ &\quad \leq s\overline{d} h_a + M(1 - h_a) + M \sum_{u=1}^{s-1} H_{t+u} \end{aligned} \tag{4}$$

ここで，$\overline{d}$ はロボットの移動速度，M は大きな正の数（たとえば，$M = d((a_X, a_Y), (b_X, b_Y))$ とおけば充分）とする．不等式制約 (4) の意味は以下のとおりである．ロボットが時点 t に位置 a にいないとき（$h_a = 0$ のとき），式 (4) は制約から取り除かれる．一方，位置 a にいるとき（$h_a = 1$ のとき），式 (4) は下記二つの意味をもっている．式 (4) 右辺の第 1 項は，位置 b は a から s 時間以内で到達可能な範囲内で選択されるべきことを意味している．さらに，第 3 項は，時点 $t+1$ から $t+s-1$ の間にどこかで検問済みであるとき（$\sum_{u=1}^{s-1} H_{t+u} \geq 1$ のとき），式 (4) を制約から取り除くことを意味している．

以上より，できるだけ多くの利得を取得できるようなロボットの最適移動経路は，以下の整数計画問題の解として得られる：

$$\max_{h,H} \quad \sum_{y \in \mathbb{Y}} \sum_{a \in \mathbb{P}_y} p_a h_a \tag{5}$$

$$\text{s.t.} \quad \text{式 (1)} \sim \text{(4)} \tag{6}$$

2.2 数値例

ここでは数値例を一つ示す．パラメータなどの設定については以下のとおりである．セル空間のサイズは，$(1+\overline{x}) \times (1+\overline{y}) = 20 \times 15$ とし，歩行者およびロボットの速度上限を $(\overline{v}, \overline{d}) = (2, 4)$ とする．また，ロボットの初期位置を $(x_0, y_0) = (0, \overline{y}/2)$，各レーン $y \in \mathbb{Y}$ の歩行者の初期値および速度 v_y はそれぞれ $\{(1+\overline{x})/2, \ldots, \overline{x}\}$ および $\{1, 2, \ldots, \overline{v}\}$ から一様ランダムに選択する．ロボットの移動規則および検問に関わる利得は以下のとおりである．ロボットは任意方向に動けるものとする，つまり，セル (x, y) から (x', y') への移動に費やす最短時間は $d((x,y),(x',y'))/\overline{d} = \sqrt{(x'-x)^2 + (y'-y)^2}/\overline{d}$ とし，位置 $a \in \mathbb{S}$ で検問したときの利得を $p_a = 1 + a_X$ とする，つまり，セル集合 $\{0\} \times \mathbb{Y}$ から離れた位置での検問ほど利得が高い．この利得の設定の意図は，$\{0\} \times \mathbb{Y}$ を通過した先に重要施設がある状況を想定し，できるだけそこから離れた位置での検問が，重要施設への攻撃リスクが低いことを見込んでのことである．

図 2 の数値結果より，理想的な環境（Ex. 歩行者の初期位置が既知，そして等速直線運動するなど）の下では，最も効率的なロボットの移動経路が得られる．ここで，計算機環境 Intel Core i7 2.3[GHz]，OS Windows 10，メモリ 32[GB] の下で，PuLP に同梱されているフリーの数理計画ソルバー CBC を用いて整数計画問題を解き，計算時間はおよそ 3 秒であった．

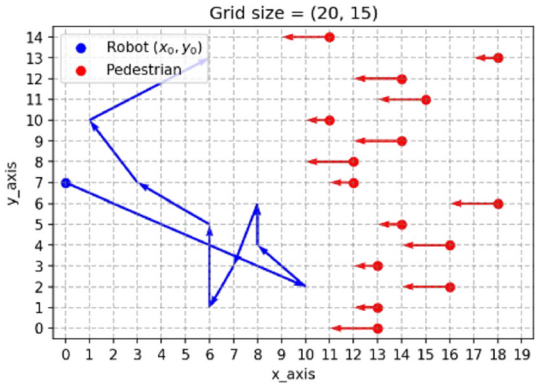

図 2　数値例（移動検問ロボットの経路）

図 3　移動検問ロボットの外観

3.　移動検問ロボットの構成

図 3 に開発した移動検問ロボットの外観を示す．Clearpath Robotics 社製四輪ロボット Husky A200 の上部に Velodyne Lidar 社製 3 次元 LiDAR (Light Detection And Ranging) HDL-32e と OccamVision Groups 社製全方位カメラ Omni Stereo を搭載している．これらのセンサによって，ロボット自身の位置と周囲の人の位置を推定し，2 節で示した最適移動経路に沿ってロボットを移動させ，ロボットに搭載されたテラヘルツセンサ（詳細は本特集の別記事 [3] に示す）で，往来する人の腹部を撮影し，不審物の有無を確認する．撮影の様子の一例として，図 4 に示すように，マネキンの腹部にナイフを取り付け，上から T シャツをかぶせた状態で移動検問ロボットを通過させた．図 5(A) に実験の様子を，図 5(B) にテラヘルツセンサで得られた画像を示す．図 5(e) において，白く光っ

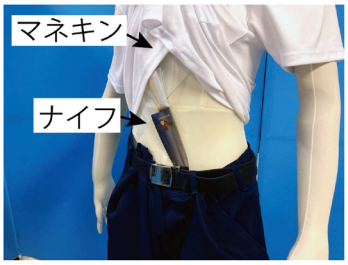

図 4　腹部にナイフを隠したマネキン

ている部分がナイフの刃の部分に該当する．このような画像に対して深層学習器などを用いて，撮影された物体がスマートフォンなのか，ナイフなのか，爆弾物なのかといったような分類を行い，危険物の探知を行う予定である．

本ロボットは複数のセンサとモータからなる複雑なシステムであるため，開発を容易にするために ROS と呼ばれるロボット用アプリケーション作成を支援するライブラリとツール群を使用している．本システムでは，Ubuntu Linux 16.04 LTS 上で動作する ROS Kinetic と呼ばれるバージョンを使用している．ROS が主に対応している開発言語は C++ や Python である．2 節で示した最適移動経路導出プログラムは Python で記述されており，シームレスに本ロボットシステムに実装することができる．また，MathWorks 社製数値計算ソフトウエア MATLAB も RO Toolbox[4] を用いることで ROS と通信することが可能であり，迅速な開発が可能である．ROS 上で動作する各実行ファイルはノードと呼ばれ，メッセージと呼ばれるノード間通信によって連携が可能となっている．roscore は各ノードにネームサービスを提供し，ノードがお互いに検索できるようにする．

図 6 に本ロボットのシステム構成図を示す．本ロボットでは，ROS によって相互接続された 3 台の電子計算機 (PC) を用いている．遠隔操作用 PC は，オペレータがロボットを操作するための端末で，この PC 上で roscore を動作させている．認識用オンボード PC と制御用オンボード PC は四輪ロボット上に搭載されており，遠隔操縦用 PC と無線 LAN で接続されている．

ロボットに搭載されている全方位カメラの RGB 映像において人体検出を行い，対応する箇所の 3 次元位置を LiDAR 情報から取得することで，ロボットの周囲の人の位置をリアルタイムで取得することができる．また，SLAM (Simultaneous Localization and Mapping) 技術（4.2 節で後述）を用いることで，ロボットの自己位置推定を行い，地図上におけるロボットと周

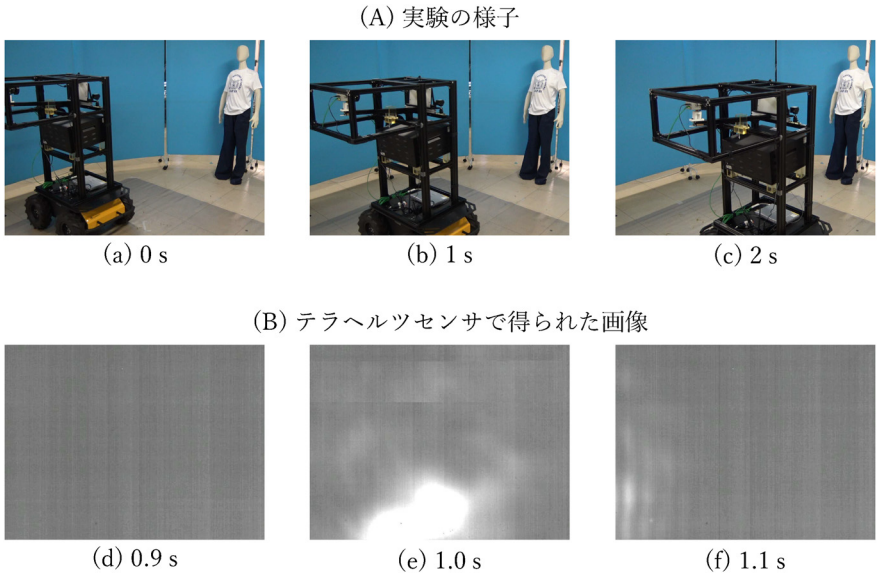

図 5　非接触不審物検査の様子

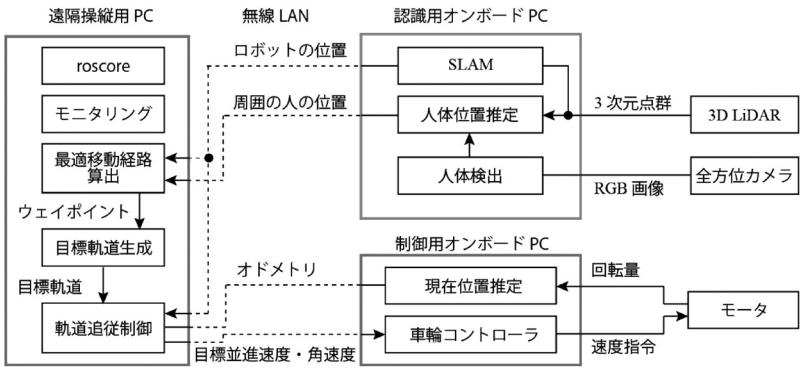

図 6　移動検問ロボットのシステム構成図

囲の人の場所や移動速度を推定することができる．なお，本研究では人体検出に MATLAB 上で実装された YOLO[5, 6] を，SLAM には LeGO-LOAM を用いている [7]．この情報を元に，2 節で示した最適移動経路算出プログラムによって，今後ロボットが移動すべき目標地点（ウェイポイント）と目標時刻を算出する．このウェイポイントと目標時間を補間するように目標軌道を生成し，これを追従するように 4 輪ロボットを制御する．

一般的に，移動ロボットは車輪の滑りなどによって指令したとおりに移動することはない．このため，モータの回転量から計算されるオドメトリ情報と実際の位置には差が生じる．そこで，SLAM によって得られた比較的正確な自己位置情報に基づいて，目標軌道に追従するように制御することで，最適移動経路算出プログラムによって得られたウェイポイントを指定時間で通過するように移動させることが可能となる．

4.　移動ロボットの技術＆実行可能性

2 節で述べた最適移動経路の導出では，ロボットや歩行者の位置や速度はいつでも観測できること，歩行者はおおむね等速直線運動を継続すること，事象や計測誤差の発生頻度は正規分布に基づくこと，ロボットは任意のタイミングで自由に加減速ができることなど，さまざまな前提条件や計算を簡略化するための仮説が設けられている．しかし現実には，これらの前提条件や仮説を担保することは非常に困難な問題である．

4.1　ロボットの自己位置推定

実世界に存在するロボットが行動するには，まず自分自身の位置姿勢や，対象物体との相対位置関係を正

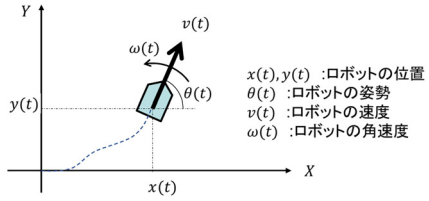

図 7　自己位置の累積計算

確に知る必要がある．人間の場合には，自身が記憶している地図と自分の目で見えるものとの対応を取り，自らの位置や向きを判断する．

ロボットの場合には，大きく分けて二通りの方法で自己位置を計算する．一つは内界センサから得られる自らの速度や角速度情報から累積移動量を計算する手法，もう一つは，外界センサから得られる情報を用いて既知の地図における座標値を計算する手法である．

4.1.1　内界センサによる自己位置推定方法

内界センサとは，ロボット自身がもっている運動量などを観測する装置のことをいい，車輪型の移動機構における各車輪の回転角速度計や，ドローンにおけるジャイロ（加速度計・角加速度計）などを指す．内界センサで得られた情報からは絶対的な位置情報を取得することはできないが，観測された速度や各加速度を積分することで，相対的な位置や姿勢の変化を求めることができる．このように，内界センサ情報の累積により相対的な自己位置移動量を計算することをデッドレコニングといい，特に本稿の 3 節でとりあげたような車輪型ロボットにおいて車輪の回転数から計算された自己位置のことをオドメトリと呼ぶ．

移動ロボットの並進速度を v，回転角速度を ω としたとき，オドメトリによる自己位置 $(x(t), y(t))$ および姿勢 $(\theta(t))$ の累積計算は以下の式であらわすことができる．

$$\theta(t) = \int_{t_0}^{t} \omega(\tau) d\tau + \theta(t_0), \tag{7}$$

$$x(t) = \int_{t_0}^{t} v(\tau) \cos(\theta(\tau)) d\tau + x(t_0), \tag{8}$$

$$y(t) = \int_{t_0}^{t} v(\tau) \sin(\theta(\tau)) d\tau + y(t_0) \tag{9}$$

ここで $x(t_0)$, $y(t_0)$, $\theta(t_0)$ は初期状態におけるロボットの位置と向きである（図 7 も参照）．この方法は，ロボットが自分自身の内部の状態を用いているため，高速かつ連続的にサンプリングできる．そのためほとんどの移動ロボットは何らかのかたちで内界センサの

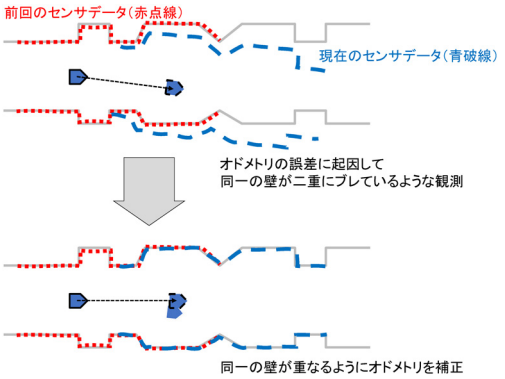

図 8　スキャンマッチング

情報を利用して位置推定を行っている．一方，センサ情報に含まれるわずかな計測誤差，コンピュータで計算する際の量子化誤差，車輪のスリップなどの誤差を徐々に累積してしまうため，長距離を移動する場合には定期的に誤差を検出して補正しなければならない．また，絶対的な位置情報との関連付けを適切に行う必要がある．

4.1.2　外界センサによる自己位置推定方法

外界センサで位置を観測するケースは，2 タイプの計測方法に分類できる．一つは，GPS 信号のように外部から直接絶対位置を得る手法，もう一つは，予め用意した地図情報と，センサで観測した風景や形状との対応を取ることで，地図上における位置姿勢を計算しようとするものである．前者は GPS 受信機さえもっていれば容易に取得することができるため，カーナビやスマートフォンの地図アプリなどで多く採用されている．ただし，衛星からの電波を受信できない環境で利用することはできず，また一般的な GPS は ±5 メートル程度の誤差を含む（1 メートル幅の歩道から逸脱しないように走るには全く精度が足りない）．

後者は，センサを使って周囲環境に存在する物体（壁や樹木など）の表面形状や色情報を抽出し，データ上の地図とのマッチング処理を行う．かつては超音波センサが主流であったが，近年は LiDAR と呼ばれるレーザー距離センサや深度カメラを用いるのが一般的である．

図 8 に，レーザースキャナを用いた場合のスキャンマッチングの処理のイメージを示す．ある地点にいるロボットが観測した周囲の壁の形状を赤点線，そこから一定距離移動した場所で再度観測した壁の形状を青破線で示す．二地点間の相対的な位置関係が正確であれば，同一の壁を観測している部分において赤点線と青破線は完全に重なるはずであるが，デッドレコニングに基づいて二地点間の位置関係を求めようとする

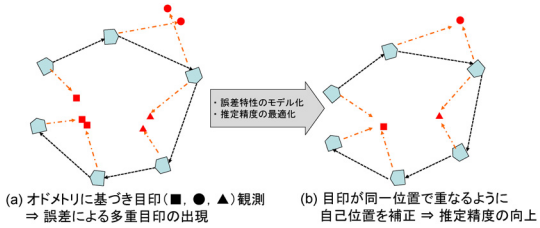

図 9　SLAM

と，同一の壁が二重にブレているような観測結果が得られる．スキャンマッチングは，二地点で得られた周辺形状の重なり具合を定量的に評価しながら，観測地点の位置をずらしていき，重なり度が最大となる場所を求める．もし図 8 のように形状がちょうど重なる箇所があった場合，その位置を正しい観測位置として修正する．マッチングするデータ同士の相対位置の初期値がある程度正確に推定できる場合には ICP (Iterative Closest Point)[8], NDT (Normal Distributions Transform)[9], 3D-NDT[10] といった手法が広く用いられている．

4.2　SLAM

見知らぬ土地で行動することを考えた場合，私たちは頭の中にその土地の地図を構築し，構築した地図上に自分の現在地を思い浮かべ，動き回りながら周囲の情報を取り入れつつ行動するだろう．また，それと同時に獲得した情報から頭の中の地図情報を更新・再構築し，自分の位置をより正確なものにしていく．以上のプロセスによって，うろうろと動き回るうち目的地に到着できる．ロボットにこれと同じことをさせようというのが SLAM (Simultaneous Localization And Mapping) である [10]．直訳すると「自己位置推定と地図の同時構築」である．

図 9 に，SLAM により自己位置と地図を修正する処理のイメージを示す．ロボットはデッドレコニングに基づいて移動しながら，一定の間隔ごとに外界センサを使って周辺のランドマーク（目印）を観測するものとする．デッドレコニングにより得られるロボットの移動量やセンサの観測情報にはそれぞれ誤差が含まれているため，これらの誤差を考慮せずに走行経路やランドマーク位置を座標変換していくと，同一のランドマークが多重に現れたり（図 9(a) 参照），ループ状の経路を一周して元の位置に戻ってきた場合でも推定位置が初期位置に戻っていないケースが生じたりする．SLAM は，デッドレコニングやランドマーク観測における誤差特性をモデル化し，同一のランドマークがなるべく一点で重なるように，自身がこれまでに移動してきた経路や各地点で観測したランドマーク同士の相対位置関係を最適化することで，自己位置推定と地図作成とを同時に実現するものである．

近年はレーザースキャナや深度カメラなどの情報量の多いセンサを用いて，ほぼリアルタイムの地図作成ができるようになり，戦闘地域の偵察や惑星探査などにおいても実用的なレベルにまで達している．そのほかにも，SLAM を用いた自律移動ロボットはさまざまな状況での活躍が期待されており，現在の研究では一般家庭やオフィス環境での作業，災害時の人命救助や被害状況の把握などでの実用化を目指している．

5.　おわりに

本稿では，移動検問ロボット開発における OR 分野の一手法の応用について述べ，実ロボットへの実装方法について述べた．4 節でも述べたように，実環境でのロボット運用には問題が山積みであるが，それは同時に，OR 手法の応用の可能性がまだあることを示唆しているように思える．

参考文献

[1] 岡田慧，“ROS（ロボット・オペレーティング・システム）”，日本ロボット学会誌，**30**, pp. 830–835, 2012.
[2] Open Robotics, ROS Wiki, http://wiki.ros.org/（2022 年 7 月 21 日閲覧）
[3] 山田俊輔，“テラヘルツ波を用いた金属物検知”，オペレーションズ・リサーチ：経営の科学，**67**, pp. 528–534, 2022.
[4] The MathWorks, Inc., ROS Toolbox 製品情報，https://jp.mathworks.com/products/ros.html（2022 年 7 月 21 日閲覧）
[5] The MathWorks, Inc., YOLO v2 入門，https://jp.mathworks.com/help/vision/ug/getting-started-with-yolo-v2.html（2022 年 7 月 21 日閲覧）
[6] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6517–6525 2017.
[7] T. Shan and B. Englot, “LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain,” In *Proceedings of the 2018 IEEE International Conference on Intelligent Robots and Systems*, pp. 4758–4765, 2018.
[8] P. J. Besl and N. McKay, “A method for registration of 3-D shapes,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **14**, pp. 239–256, 1992.
[9] P. Biber and W. Strasser, “The normal distributions transform : A new approach to laser scan matching,” In *Proceedings of IROS 2003*, pp. 2743–2748, 2003.
[10] E. Takeuchi and T. Tsubouchi, “A 3-D scan matching using improved 3-D normal distributions transform for mobile robotic mapping,” In *Proceedings of IROS 2006*, pp. 3068–3073, 2006.