「マルチメディア，分散，協調とモバイル
(DICOMO2021)シンポジウム」 令和3年6月-7月

人と共存する自律移動ロボットにおける安全性と効率性を考慮したナビゲーション手法の評価

天野 加奈子[1]　加藤 由花[1]

概要：近年，人と共存する自律移動ロボットに対する期待が高まっている．これらサービスロボットには，安全性に配慮した動作が求められ，特に，歩行者が存在する動的環境下で行動する移動ロボットの場合，ロボット自身が歩行者を認識し，回避行動をとる必要がある．我々はこれまで，歩行者を回避しつつ効率的に目標位置に到達するロボットの行動を強化学習により獲得し，それを経路計画に利用する手法を提案してきた．ここでは，時間の経過と歩行経路への侵入に対して負の報酬を与えることで，安全性と効率性の双方を考慮した行動を学習させた．本稿では，提案したナビゲーション手法を小型車輪型ロボットに実装し，実機においても提案手法が有効に機能することを検証する．ここでは，学習機構と合わせて，センサーによる人物追従機能，環境地図作成機能，自己位置推定機能をロボットに実装し，評価実験を行う．

1. はじめに

近年，人と共存する自律移動ロボットに対する期待が高まっている．例えば，商業施設や空港，博物館などで，案内・運搬・警備など様々なサービスを提供するロボットである．これらサービスロボットには，安全性に配慮した動作が求められ，特に，歩行者が存在する動的環境下で行動する移動ロボットの場合，ロボット自身が歩行者を認識し，回避行動をとる必要がある．また，移動ロボットは，自身に搭載したバッテリーを動力源とするのが一般的であり，不要な動作による電力消費を可能な限り抑えることが望ましい．そのため，安全かつ効率的な移動経路の計画は，移動ロボットにとって重要な研究課題になっている．

動的環境における移動ロボットの経路計画については，これまでも多くの研究開発が行われてきた．障害物とゴールにポテンシャル関数を定義することで進行方向を決定するポテンシャル法 [1] や，ロボットを配備する環境下で計測した歩行者軌跡データを用いて時刻ごとの人存在確率を予測し，経路計画に利用する手法 [2], [3] などである．前者は，ロボットの近傍から得られる情報のみを利用するため，未知の環境においても障害物との衝突を回避する行動を策定できる一方，目的地から遠ざかる非効率な経路が生成される場合がある．後者は，環境内の人の行動特性に従った経路を計画できる一方，ロボットを配備する環境ごとにあらかじめ軌跡データを収集する必要がある．

[1] 東京女子大学
Tokyo Woman's Christian University

これらの問題を解決するために，我々はこれまで，歩行者を回避しつつ効率的に目標位置に到達するロボットの行動を強化学習により獲得し，それを経路計画に利用する手法を提案してきた [4]．強化学習は，行動の主体であるエージェントが環境の状態を観測し，何らかの行動により報酬を得るという一連の流れを繰り返すことで，得られる報酬の期待値を最大化する行動を獲得する手法である．文献 [4] では，時間の経過と歩行経路への侵入に対して負の報酬を与えることで，安全性と効率性の双方を考慮した行動を，シミュレータ上で学習させた．

本稿では，提案したナビゲーション手法を小型車輪型ロボットに実装し，実機においても提案手法が有効に機能することを検証する．ここでは，学習機構と合わせて，センサーによる人物追従機能，環境地図作成機能，自己位置推定機能をロボットに実装し，評価実験を行う．

2. 関連研究

2.1 局所的な経路探索手法

移動障害物が存在する環境では，ロボットの近傍の情報を利用して，局所的な経路決定，移動をゴール到達まで繰り返す局所的経路探索手法（例えば [5] など）がよく用いられる．これらの手法は，突発的な移動障害物の出現に対応可能である一方，移動障害物の経路によってはロボットが不要な回避や待機を行うことがある．また，短い周期で経路を更新するため移動効率が悪化したり，急に向きを変えるなど，人間にとって親和的でない行動を取る場合もある．

© 2021 Information Processing Society of Japan

2.2 歩行者経路予測と組み合わせた大域的な経路計画

環境内の人の移動傾向を考慮しつつ，効率的に目標地点に到達するために，人移動軌跡データを取り入れた経路計画手法が提案されている [6]．これらの手法は，環境内の人の移動傾向から歩行経路を時系列データとしてリアルタイムに予測し，その結果を経路計画に反映する．

例えば，特定の環境下で観測された軌跡データから歩行者移動モデルを生成することで歩行者経路を予測し，これを経路計画に用いる手法 [3] や，動的環境において大域的経路探索を可能にするために，時空間グラフに障害物移動経路を不可侵領域として組み込む手法 [7] などが提案されている．前者の手法は，ロボットの速度や回転などの制約を考慮していないため実用的な環境への適用が難しく，後者の手法は，施設案内（美術館など）のように人の移動経路が定まっている環境を前提としているため，人が自由に行き来するような環境に適用できないという問題がある．歩行者経路予測など，他の手法と組み合わせる必要がある．

従来の RRT*（Rapidly exploring Random Tree star）ベースのアルゴリズムの拡張として，LSTM（Long-Short Term memory）を用いた歩行者経路予測を組み込んだ手法も提案されている [8]．RTT は空間的な情報に加えて時間的な情報をプランニングに反映させることができるため，より効率的な経路計画が可能である．しかし，常に最適な経路が生成されるとは限らないという問題がある．

2.3 強化学習を用いた経路計画

機械学習や深層学習の進展により，強化学習を用いた経路計画に関する研究も活発に行われている．強化学習は，ロボット自身の試行錯誤により，目標を達成するためのモデルを獲得するアルゴリズムであり [9]，未知・不確実な環境に適応した行動の獲得が可能である．事前に大量の学習が必要であるが，学習後は高速に経路生成が可能である．

ロボットの経路計画に関しては，学習結果を用いて経路探索アルゴリズムや深層強化学習の目的関数，報酬関数のパラメータを決定する手法が多く提案されている [10], [11], [12]．ロボットへの動作指令を強化学習により直接獲得する研究もある．静的環境においては，環境地図を事前に取得せずに動作計画を実現するために，深層強化学習を用いてセンサー情報と目標位置から動作命令を直接獲得する手法が提案されている [13]．動的環境においては，移動障害物の次時刻遷移先の予測と実際の観測点の差異を評価して行動選択を行う予測型強化学習による行動計画が提案されている [14]．この手法は，環境に依存しない回避行動獲得が可能であるが，衝突予測に基づいた危険度によってのみ行動を決定するため，危険度が同じ場合の行動優先順位を手動で設定しないとゴールに到達できないという問題がある．

本稿では，強化学習を用いることで，動的環境において適切な経路計画を実現する手法を対象に，実装評価を行う．

3. マルコフ決定過程と Q 学習

まず，対象とする経路計画手法で用いるマルコフ決定過程と Q 学習 [15] について説明する．

3.1 マルコフ決定過程

マルコフ決定過程は，環境の取り得る状態の有限集合 $\mathcal{S}$ とエージェントが取る行動の有限集合 $\mathcal{A}$ によって定義される．時刻 t において状態 $s \in \mathcal{S}$ にあるロボットが，エージェントに与えられた制御指令により行動 $a \in \mathcal{A}$ を実行し，時刻 $t+1$ において状態 $s' \in \mathcal{S}$ に遷移するとき，状態遷移確率は $P(s' \mid s, a)$ と表される．このとき，エージェントが環境から受け取る報酬 r は確率的に決定され，その期待値は $R(s, a, s')$ と表される．ここで，状態遷移確率，報酬の期待値はともにもマルコフ性を持ち，時刻 t 以前の状態や行動履歴に依存しないとする．エージェントは何かしらの行動ルール（方策 Π）に従って行動 a を選択し，行動の評価値を最大化するような行動選択の規則を見出していく．評価値は遷移後の状態 s' における報酬と価値の和の期待値から成り，状態価値関数は，

$$V^{\Pi}(s) = \mathbb{E}_{P(s'|s,a)}\left[R(s,a,s') + V^{\Pi}(s')\right]. \qquad (1)$$

と表される．ある状態 s において既存の行動 $a = \Pi(s)$ よりも高い評価値を得られる行動があれば，エージェントは行動を変えた方が良いことになる．このとき，状態価値関数の行動 a を変数として，行動価値関数を，

$$Q^{\Pi}(s,a) = \mathbb{E}_{P(s'|s,a)}\left[R(s,a,s') + V^{\Pi}(s')\right] \qquad (2)$$

と定義する．すると行動の書き換えの手続き（方策改善）は，

$$\Pi(s) = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\, Q^{\Pi}(s,a) \qquad (3)$$

となる．全行動に対する $Q^{\Pi}(s,a)$ を求め，最大の行動価値を $V(s)$ に代入することを繰り返し，最終的に得られる状態価値関数 V^* の示す方策が最適方策 Π^* となる．

3.2 Q 学習

Q 学習は，ロボットが行動をとった後に得られる情報から行動価値関数を更新していくアルゴリズムの一つである．すべての状態 $s \in \mathcal{S}$，行動 $a \in \mathcal{A}$ に対する行動価値関数の値 $Q(s,a)$（以下，Q 値と呼ぶ）を任意の初期値に設定して学習を開始し，エージェントの試行錯誤により，以下のように Q 値を更新していく．

$$Q(s,a) \longleftarrow (1-\alpha)Q(s,a) + \alpha\left[r + \max_{a'} Q(s',a')\right]. \qquad (4)$$

ここで α $(0 < \alpha < 1)$ は学習率である．$\max_{a'} Q(s',a')$ は遷移先の状態 s' において一番価値の高い行動 a' を選んだときの Q 値を指し，このような行動を選択する方策はグリーディ方策と呼ばれる．本稿では，確率 ε でランダムに行動を選択する ε-グリーディ方策を採用する．

© 2021 Information Processing Society of Japan

4. ロボットシステムの設計

4.1 前提条件

本稿で対象とする経路計画手法は，博物館や空港，大型商業施設など，歩行者が自由に行き来できる通路がある環境において，自律移動ロボットが歩行者との衝突を回避しながら目的地に到達するためのものである．これらを考慮し，ナビゲーション手法を実装するロボットシステムとして，以下の前提条件を仮定する．

- 2 輪型移動ロボットを対象とする．
- 環境情報のセンシングには LiDAR を用いる．
- ロボットへの制御指令は $u = (\nu, \omega)^\mathsf{T}$ で与える．
  ν [m/s] は前進速度，ω [rad/s] は中心角速度である．
- 環境中に存在する歩行者数は 1 人とする．
- 目標位置（ゴール）は既知であるとする．

4.2 システムの概要

システムの構成を **図 1** に示す．ロボットシステムは，以下に示す 6 つの機能からなるものとする．

- 環境情報の取得
- 環境地図の構築
- 自己位置推定
- 障害物の検知・追跡
- 経路計画
- モータの制御

環境情報はロボットに搭載された LiDAR により取得し，環境地図の構築，自己位置の推定，および障害物（本稿では歩行者）の検知・追跡に利用する．そして，ロボットの自己位置推定結果，障害物検知結果から現在の状態（前章で定義した環境の取り得る状態）を把握し，シミュレータを用いた事前学習により獲得しておいた方策を用いて，ロボットの経路を生成する．その後，経路計画に従った移動命令を，ロボット実機のモータ制御機能に入力することで，ロボットは歩行者を回避しながらゴールまで移動する．

以下，それぞれの機能について詳細を説明する．

4.3 機能の詳細

4.3.1 環境情報の取得

ロボットに搭載された LiDAR を用いて，環境の形状（LiDAR から物体までの距離と角度）を点群データとして取得する．取得したデータは，環境地図の構築に用いる他，実環境にロボットを配備してサービスを提供するときの自己位置推定，および障害物検知・追跡に用いる．

4.3.2 環境地図の構築

環境地図の構築には，既存の SLAM（Simultaneous Localization and Mapping）アルゴリズムを用いる．環境情報の取得に LiDAR を用いるため，LiDAR SLAM の一種であり，周辺環境を点群データとして計測し，ベイズフィルタを用いて地図の点群を生成する Gmapping（Grid Mapping）[16] を採用する．対象となる環境で，LiDAR により環境データを観測しながらロボットを遠隔操作することにより，事前に環境地図を構築しておく．

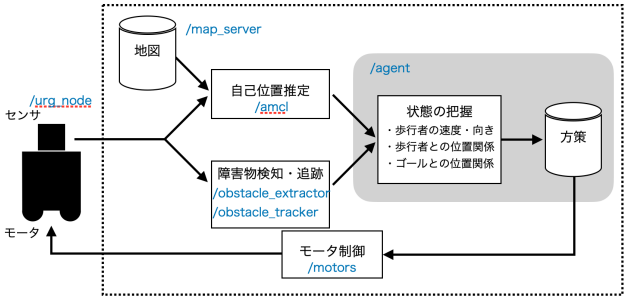

図 1: システムの構成．SLAM により予め構築しておいた環境地図と，ロボットに搭載されたセンサーから取得したデータを用いて，自己位置推定を行いながら，障害物（この場合は歩行者）の検知と追跡を行う．それらのデータから状態を把握し，事前学習により獲得した方策を用いて，経路計画を行う．経路計画に従いモータ制御を行うことで，ロボットは歩行者を回避しながらゴールに到着する．青字は，実装時に用いる ROS パッケージの名称である．

4.3.3 自己位置推定

与えられた地図，ロボットの遷移モデル，センサーの観測モデルを用いて自己位置推定を行う．ここでは，ランドマーク m_j（ランドマークごとの位置座標として定義，j はランドマークの ID である）の集合として地図 $\mathcal{M}$ を定義する．ロボットの遷移モデルとしては，時刻 $t-1$ に状態 s_{t-1}（ロボットの二次元座標と姿勢を合わせたものとして定義）にあるロボットに移動命令 a_t が作用することで s_t に遷移する確率を，確率分布 $p(s_t \mid s_{t-1}, a_t)$ として定めておく．センサーの観測モデルとしては，時刻 t におけるロボットの状態が s_t であるとき z_t が観測される確率を，確率分布 $p(z_t \mid s_t, \mathcal{M})$ として与えておく．

これらの条件の下で，$z_{1:t}$，$a_{1:t}$，$\mathcal{M}$ から s_t を推定する．ここで，ロボットの初期状態は既知であり，状態の更新は再帰的に行われると仮定すると，s_t の確率分布は，

$$p(s_t \mid z_{1:t}, a_{1:t}, \mathcal{M}) = \alpha\, p(z_t \mid s_t, \mathcal{M}) \times \int p(s_t \mid s_{t-1}, a_t)\, p(s_{t-1} \mid z_{1:t-1}, a_{1:t-1}, \mathcal{M})\, ds_{t-1} \tag{5}$$

となり，逐次推定が可能になる．ここで，ベイズ理論とマルコフ性の仮定を用いた．なお，α は正規化定数である．

4.3.4 障害物の検知・追跡

LiDAR の計測範囲内に入った歩行者を補足し，物体までの距離と角度を逐次計測することにより追跡を行う．ロボットの自己位置推定と同様，物体の位置を確率分布として表現し，追跡対象に応じた遷移モデルとセンサーの観測モデルに従い，再帰的に追跡対象の位置に関する確率分布を推定していく．前項と同様，ベイズフィルタを用いる．

© 2021 Information Processing Society of Japan

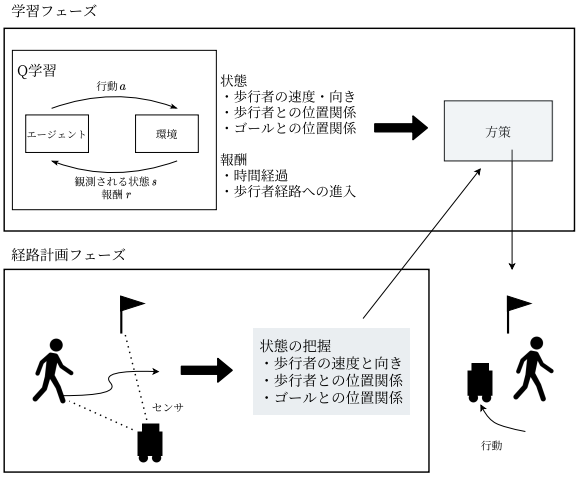

図 2: 経路計画の概要 [4]．方策は Q 学習を用いて事前に生成しておく．エージェントは，方策と自律移動ロボットが計測したセンシングデータを用いて，歩行者との衝突を回避し，目標位置に到達するための行動を獲得する．

なお，歩行者の位置座標は，LiDAR により極座標形式で計測されるため，これを直行座標形式に変換することで，単位時間ごとの人位置データが算出される．

4.3.5 経路計画

経路計画は，文献 [4] に示したものと同じ手法を用いる．手法の概要を **図 2** に示す．手法は Q 学習による学習フェーズと経路計画フェーズの 2 段階で構成される．学習フェーズでは，シミュレータ上の環境で人移動軌跡データを使用し，ロボットと歩行者，目標地点の位置関係や歩行者の速度，移動方向などの状態に応じてエージェントがロボットに与える制御指令を学習する．経路計画フェーズでは，事前に学習フェーズで獲得した方策を参照することにより目標地点到達までの経路計画を行う．

（1）学習フェーズ

環境の状態は，以下の 5 つの値で定義する（全てロボットに設定されたローカル座標系での値）：ロボットから歩行者までの距離（l_p）；角度（ϕ_p）；歩行者の速度（v_p）；歩行者の方向（θ_p）；ゴールの角度（ϕ_g）．センサーにより直接計測されるのは l_p と ϕ_p であり，残りの値を計測データから導出することで，状態ベクトルを以下のように定義する．

$$\boldsymbol{x} = (l_p,\ \phi_p,\ v_p,\ \theta_p,\ \phi_g). \tag{6}$$

状態に対応する Q 値を参照するにあたり，連続空間における状態を離散化する．具体的には，状態空間の各次元をそれぞれ幅 $w_{l_p}, w_{\phi_p}, w_{v_p}, w_{\theta_p}, w_{\phi_g}$ によりグリッド化し，各離散状態 s に対して価値を求める．各グリッドに番号を振り，これが $\boldsymbol{i} = (i_{l_p},\ i_{\phi_p},\ i_{v_p},\ i_{\theta_p},\ i_{\phi_g})$ である離散状態を $s_{\boldsymbol{i}}$ と表すことにする．そして，現在の状態 $\boldsymbol{x}$ から，離散空間における番号の組 $\boldsymbol{i}$ を求める．各離散状態において各行動 a の Q 値が記録されているものとし，ε-グリーディ方策に基づいて $s_{\boldsymbol{i}}$ における行動 a を選択する．

報酬については，経過時間（r_t）と歩行経路への侵入（r_c）に負の報酬を設定する．r_t は以下のように表現する．

$$r_t(s, a, s') = -\Delta t. \tag{7}$$

r_c については，現時刻におけるロボットの位置と，1 から n ステップ先までの歩行者位置について衝突判定を行う．ここでは，歩行者の位置を中心として，ロボットの半径 rad_{rob} と歩行者の半径 rad_{ped} の和の 2 倍を一辺とする正方形領域を衝突範囲とし，ロボットが衝突範囲内に位置する場合を衝突とみなす．衝突と判定された歩行者位置が未来の時刻のものであればあるほど衝突危険度は低くなるという考えの下，$i \in \{1, 2, \cdots, n\}$ ステップ先の歩行者と衝突した際の報酬を以下のように設定する．

$$r_c(s, a, s') = \beta^{i-1} \qquad (0 < \beta < 1). \tag{8}$$

報酬モデル R は，r_t と r_c の和として，

$$R(s, a, s') = r_t(s, a, s') + c \cdot r_c(s, a, s') \tag{9}$$

と定義する．ただし，c を歩行経路侵入に対するペナルティーの大きさを決める係数である．

行動により得られた報酬 R と遷移先の状態 s' を用いて，1 ステップ前の状態 s と行動 a の組の価値 $Q(s, a)$ を式 (4) により更新する．ただし，終端状態である場合は，$\max_{a'} Q(s', a')$ の代わりに終端状態の価値を使用する．終端状態は，ロボットが目標地点に到達した場合，および歩行者と衝突した場合とし，終端価値を設定しておく．

（2）計画フェーズ

学習フェーズと同様に，$l_p,\ \phi_p,\ v_p,\ \theta_p,\ \phi_g$ を計算し，現在の状態ベクトル $\boldsymbol{x} = (l_p, \phi_p, v_p, \theta_p, \phi_g)$ を得る．歩行者を観測している場合は，前項で算出した現在の状態 $\boldsymbol{x}$ に対する離散状態空間におけるインデックス $\boldsymbol{i}$ を求める．そして，前項で獲得した行動価値関数を参照し，$s_{\boldsymbol{i}}$ における行動をグリーディ方策により選択する．

4.3.6 モータの駆動

得られた方策に従い行動 a を決定し，行動 a に従った移動命令を，ロボット実機のモータ制御機能に入力することで，ロボットは歩行者を回避しながらゴールまで移動することが可能になる．実機の種類により制御命令は異なる．

5. 実装と評価

5.1 ロボットシステムの実装

前章で述べた設計結果に従い，実ロボットに機能を実装し，ナビゲーション手法が有効に機能することを検証する．今回は **図 3** に示す小型車輪型ロボット（RT 製 Raspberry Pi Mouse[*1]に HOKUYO 製 URG-04LX-UG01 を搭載したロボット）を用いることにした．

[*1] Raspberry Pi Mouse:
https://rt-net.jp/products/raspberrypimousev3/

© 2021 Information Processing Society of Japan

表 1: 利用する ROS パッケージ

パッケージ名	対応する機能	説明
urg_node	環境情報の取得	URG からのデータ取得
slam_gmappig	環境地図の構築	SLAM による地図生成，gmapping を拡張したもの（https://github.com/ryuichiueda/pimouse_slam）
map_server	環境地図の構築	地図の管理
amcl	自己位置推定	パーティクルフィルタによる自己位置推定
obstacle_extractor	物体検知	円形障害物の抽出（https://github.com/tysik/obstacle_detector）
obstacle_tracker	物体追跡	カルマンフィルタによる円形障害物の追跡（https://github.com/tysik/obstacle_detector）
agent	経路計画	自作したエージェントを ROS パッケージ化したもの
motors	モータの駆動	Raspberry Pi Mouse 用の ROS パッケージを利用

図 3: 本研究で利用するロボット．小型車輪型ロボット（RT 製 Raspberry Pi Mouse に HOKUYO 製 URG-04LX-UG01 を搭載したロボット）を実験に利用する．

各機能，アルゴリズムの実装にはロボット用ソフトウェアプラットフォームである ROS（Robot Operating System）[*2]を利用した．ROS では，提供されている様々なパッケージを組み合わせてシステムを構成することが可能である．本実験で利用する ROS パッケージの一覧を **表 1** に示す．URL を提示していないものについては，ROS のデフォルトパッケージを用いている．実装する機能とパッケージの関係は 図 1 にも青字で示した．

5.2 方策の学習

文献 [4] と同様に，Python で自作したシミュレータを用いて方策を学習し，agent パッケージに組み込んだ．

5.2.1 シミュレーション環境

シミュレーション環境は，10 [m] × 10 [m] の正方形領域内に，正方形の中心を原点とする二次元直行座標系を設定し，環境内に 1 台のロボット，1 つのゴール，1 人の歩行者を配置することで構築した．ロボットの半径は $r_{rob} = 0.2$ [m]，歩行者の半径は，歩行者自身と個人空間を考慮して $r_{ped} = 0.5$ [m] とした．歩行者は，シミュレータ外部から与える歩行者移動軌跡データを使用して移動し，ロボットはセンサによりこれを観測できるものとする．

ロボットに搭載するセンサについては，URG-04LX-UG01 の製品仕様を参考に，観測距離を $0.5 \sim 4.0$ [m]，観測角度を，ロボット正面を 0 [rad] として $-2\pi/3 \sim 2\pi/3$ [rad] とした．ロボットの行動 $a \in \mathcal{A}$ は，ロボットへの制御指令である前方方向への速度 ν [m/s] と中心の角速度 ω [rad/s] の組 $(\nu, \omega)^\mathsf{T}$ をそのまま利用し，$\mathcal{A} = \{(0.0, 2.0), (1.0, 0.0), (0.0, -2.0)\}$（それぞれ，左回転，直進，右回転に相当）とした．離散状態の設定は，センサの観測範囲に基づき決定される各状態の最小値，最大値と，離散幅により決定した．離散化のパラメータを **表 2** に示す．

5.2.2 歩行者移動軌跡データ

本稿では，博物館や大型商業施設のように道幅が一定程度あり，通路内を歩行者が行き来するような環境を想定している． そこで，想定と近い環境の人移動軌跡データセットとして ETH Dataset [17] を用いることにした．これは，市街地の歩行者を鳥瞰視点で撮影したシーンからなるデータセットで，4，5 人が並んで歩けるほどの道幅で双方向に移動可能な通路における人移動軌跡データから構成されている．ここでは，動画像から各歩行者の軌跡を抽出し，位置座標系列に変換された後の移動軌跡データを利用した．

学習用データセットを構築するために，まず，移動距離の短い歩行者や停留している歩行者の軌跡を除外し，残ったデータをシミュレーション環境の領域を通過するように平行移動させた．なお，シミュレーション環境の時間軸の離散幅は 0.1 s であるが，ETH Dataset は 0.4 s であるため，各フレーム間では歩行者は等速直線運動をしていると仮定し，データを補間した．次に，学習用のデータ数を増やすため，正規化後の各データの位置座標を原点に関して対称移動させたデータを作成し，データセットに追加した．

表 2: 離散化のパラメータ

	l_p	ϕ_p	v_p	θ_p	ϕ_g
最小値	0.5	$-2\pi/3$	0.5	$-\pi$	$-2\pi/3$
最大値	4.0	$2\pi/3$	2.5	π	$2\pi/3$
離散幅	0.5	$\pi/6$	1.0	$\pi/6$	$\pi/6$

[*2] ROS: https://www.ros.org/

© 2021 Information Processing Society of Japan

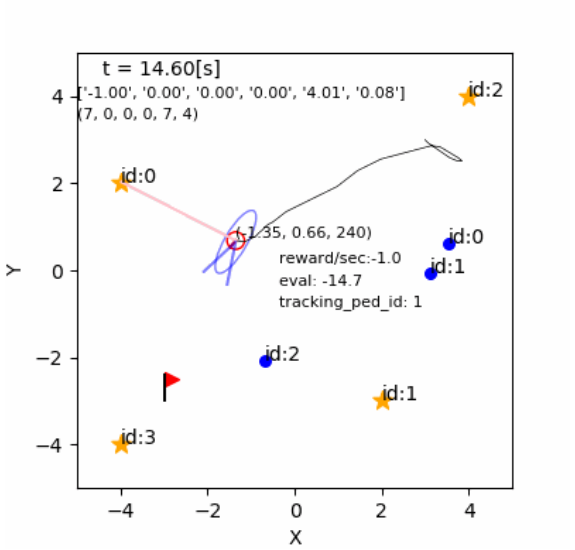

図 4: シミュレータによる学習の例 [4]．赤丸はロボット，青丸は歩行者，旗は目標地点（ゴール）である．星印は自己位置推定用のランドマークであり，実験結果に直接影響を及ぼすものではない．歩行者とランドマークには識別用の ID を割り当てている．黒線はロボットの移動軌跡を表し，ピンク色の線分はセンサによる観測を意味する．青色の楕円はカルマンフィルタによる自己位置推定における信念分布を表し，ロボットの右上に推定姿勢 (x, y, θ) が表示されている．$reward/sec$ は時刻 t に獲得した報酬，$eval$ は時刻 t までの累計報酬を表す．

5.2.3 強化学習による方策の学習

構築したデータセットを用いて，4 章に示した手順に従って学習を行った．シミュレータにおける学習の例を **図 4** に示す．本稿では，環境内の歩行者は 1 人であると仮定しているが，学習時は歩行者を観測する機会を増やすために同時に 3 人の歩行者を環境内に配置した．ただし，最初に観測した歩行者のみを観測可能とし，その歩行者の移動が終了するまで状態の更新および衝突判定にその歩行者の情報を用いる．以下の場合にタスク終了と判定し，目標位置，ロボットの姿勢，歩行者軌跡などをリセットする．

- ロボットが目標地点（ゴール）に到達した場合
- ロボットが歩行者と衝突した場合
- 試行開始から 30 秒経過した場合

リセット時には，目標位置，ロボットの初期姿勢（位置・向き）はランダムに設定する．歩行者軌跡データは，移動終了時およびリセット時にデータセットの中からランダムに選択する．報酬モデルにおけるパラメータは $n = 20$, $\beta = 0.8$, $c = 200$ とした．本稿では，歩行者回避行動の獲得が主な目的であるため，Q 値の初期値は目標位置に向かう行動の価値が他の行動の価値よりも高くなるよう設定し，35 万秒分の学習を行った．

学習後，得られた方策を agent パッケージに実装した．

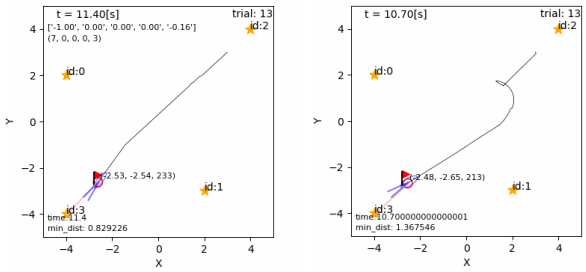

図 5: 提案手法により生成されるロボットの軌跡（左図）とポテンシャル法による軌跡（右図）[4]．いずれも，シミュレーション実験により得られた軌跡である．各 30 回試行した結果から代表的なものを 1 つずつ示した．

5.3 結果と考察

構築したロボットシステムを利用し，シミュレーション環境と同様の正方形の区域において，ロボットの初期位置と目標位置を設定し，対象領域内に歩行者が侵入した際のロボットの行動を観察した．具体的には，ロボットの初期位置を $(0.0, 0.0)$，目標位置を $(2.0, 0.0)$ に設定し，歩行者が y 軸の負の方向から領域内に侵入する（ロボットの左側から出発し，ロボットの初期位置の方向に進む）条件で実験を行った．その結果，ロボットは歩行者を観測した後，歩行者の進行方向と逆向きに回転しながら歩行者の通過を待ち，その後ゴール地点まで到達する行動を取ることが確認できた．ゴール到着の成功率，平均時間等，より詳細な評価実験は今後の課題であるが，本実験により，提案手法が実機においても有効に機能することが検証できた．

文献 [4] では，UCY Dataset [18] を用いて，シミュレータ上で上記と同様の評価実験を行っている．ここではロボットの初期位置を $(3.0, 3.0)$，目標位置を $(-2.8, -2.8)$ に設定し，歩行者はロボットから一定の距離を保った位置で，ロボットの前方から接近してくる．このときのロボットの軌跡を **図 5** に示す．30 回試行した結果から代表的なものを 1 つ取り出し，比較のためにポテンシャル法（障害物と目標位置にポテンシャル関数を定義し，ポテンシャル場の勾配からロボットの行動を決定する手法）により生成される軌跡を併記している．これらの結果からは，ポテンシャル法では進路を引き返す行動やその場で複数回回転する行動，大きく迂回する経路などが見られる一方，提案手法は歩行者が通過するまでその場で回転する動きを取ることで，初期位置から目標位置までをほぼ直線的に結ぶ経路を進んでいることがわかる．本稿における実機実験でも，同様の軌跡が観測された．

なお，学習で得た制御指令値をそのまま用いると，制御指令が頻繁に変わり，ロボットがうまく前進しない．そのため，台形駆動の処理（初期値は小さい値に設定し，徐々に加速しながら制御指令値に到達する処理）を agent に追記している．

© 2021 Information Processing Society of Japan

6. おわりに

本稿では，文献 [4] で提案した経路計画手法を小型車輪型ロボットに実装し，実機においても提案手法が有効に機能することを検証した．本稿の結論は以下の 2 点である．

- 歩行者との衝突を回避しながら自律移動するロボットを実現するために，強化学習による経路計画機能を内包したロボットシステムを設計した．
- 設計結果を小型車輪型ロボットに実装し，自律移動実験を行った．その結果，シミュレータで学習した方策を組み込むことで，歩行者を回避しながら，効率的な経路を生成可能であることがわかった．

今後，様々な条件の下でのより詳細な実機実験，複数歩行者への対応，歩行者経路予測手法との組み合わせによる経路計画手法の拡張等を行っていく予定である．

謝辞 本研究の一部は，電気通信普及財団，JSPS 科研費 20K11776，20K12011 の助成を受けたものである．

参考文献

[1] Li, G. et al.: An Efficient Improved Artificial Potential Field Based Regression Search Method for Robot Path Planning, *Proc. of IEEE International Conference on Mechatronics and Automation*, pp. 1227–1232 (2012).

[2] Ferguson, D. et al.: Replanning with RRTs, *Proc. of 2006 IEEE International Conference on Robotics and Automation 2006*, pp. 1243–1248 (2006).

[3] Noguchi, H. et al.: Mobile Robot Path Planning Using Human Prediction Model Based on Massive Trajectories, *Proc. of IEEE International Conference on Networked Sensing Systems*, pp. 1–7 (2012).

[4] 一色春香，天野加奈子，加藤由花：歩行者の移動傾向を考慮した強化学習による自律移動ロボットナビゲーション，情報処理学会研究報告 DPS186，pp. 1–9 (2021).

[5] Kato, Y., Nagano, Y. and Yokoyama, H.: A Pedestrian Model in Human-Robot Coexisting Environment for Mobile Robot Navigation, *Proc. IEEE/SICE International Symposium on System Integrationd 2017*, pp. 992–997 (2017).

[6] Akabane, R. and Kato, Y.: Pedestrian Trajectory Prediction Using Pre-trained Machine Learning Model for Human-Following Mobile Robot, *Proc. IEEE Big Data Workshop (IoTDA) 2020*, pp. 3453–3458 (2020).

[7] Iwase, M., Toda, Y. and Kubota, N.: The Return Way Path Planning of an Autonomous Mobile Robot considering Traveling Risk of the Road, *Proc. of 19th International Symposium on Advanced Intelligent Systems*, pp. 1406–1409 (2018).

[8] Todi, V., Sengupta, G. and Bhattacharya, S.: Probabilistic Path Planning using Obstacle Trajectory Prediction, *Proc. of ACM CODS-COMAD 2019*, pp. 36–43 (2019).

[9] Sutton, R. S. and Barto, A. G.: *Reinforcement Learning: An Introduction*, The MIT Press (2018).

[10] Chen, Y. F., Everett, M., Liu, M. and How, J. P.: Socially aware motion planning with deep reinforcement learning, *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1343–1350 (2017).

[11] Wulfmeler, M., Rao, D., Wang, D. Z., Ondruska, P. and Posner, I.: Large-scale cost function learning for path planning using deep inverse reinforcement learning, *The Internatilnal Journal of Robotics Research*, Vol. 36, No. 10, pp. 1073–1087 (2017).

[12] Jing, Y., Chen, Y., Jiao, M., Huand, J., Niu, B. and Zheng, W.: Mobile Robot Path Planning Based on Improved Reinforcement Learning Optimization, *Proc. of the 2019 International Conference on Robotics Systems and Vehicle Technology*, pp. 138–143 (2019).

[13] Tai, L., Paolo, G. and Liu, M.: Virtual-to-Real Deep Reinforcement Learning: Continuous Control of Mobile Robots for Mapless Navigation, *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 31–36 (2017).

[14] Everett, M., Chen, Y. F. and How, J. P.: Motion Planning Among Dynamic, Decision-Making Agents with Deep Reinforcement Learning, *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3052–3059 (2018).

[15] Watkins, C. J. and Dayan, P.: Q-learning, *Machine Learning*, Vol. 8, No. 3–4, pp. 279–292 (1992).

[16] Grisetti, G., Stachniss, C. and Burgard, W.: Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters, *IEEE Transactions on Robotics*, Vol. 23, No. 1, pp. 34–46 (2007).

[17] Pellegrini, S., Ess, A., Schindler, K. and Cool, L.: You'll Never Walk Alone: Modeling Social Behavior for Multi-target Tracking, *Proc. of ICCV 2009*, pp. 261–268 (2009).

[18] Lerner, A., Chrysanthou, Y. and Lischinski, D.: Crowds by Example, *Computer Graphics Forum*, Vol. 26, No. 3, pp. 655–664 (2007).

© 2021 Information Processing Society of Japan