動体物の位置を考慮した
複数移動ロボットの動的ナビゲーション

実証実験レポート

2021.3.31　　1.0版

TIS株式会社
会津大学
株式会社日本アドシス

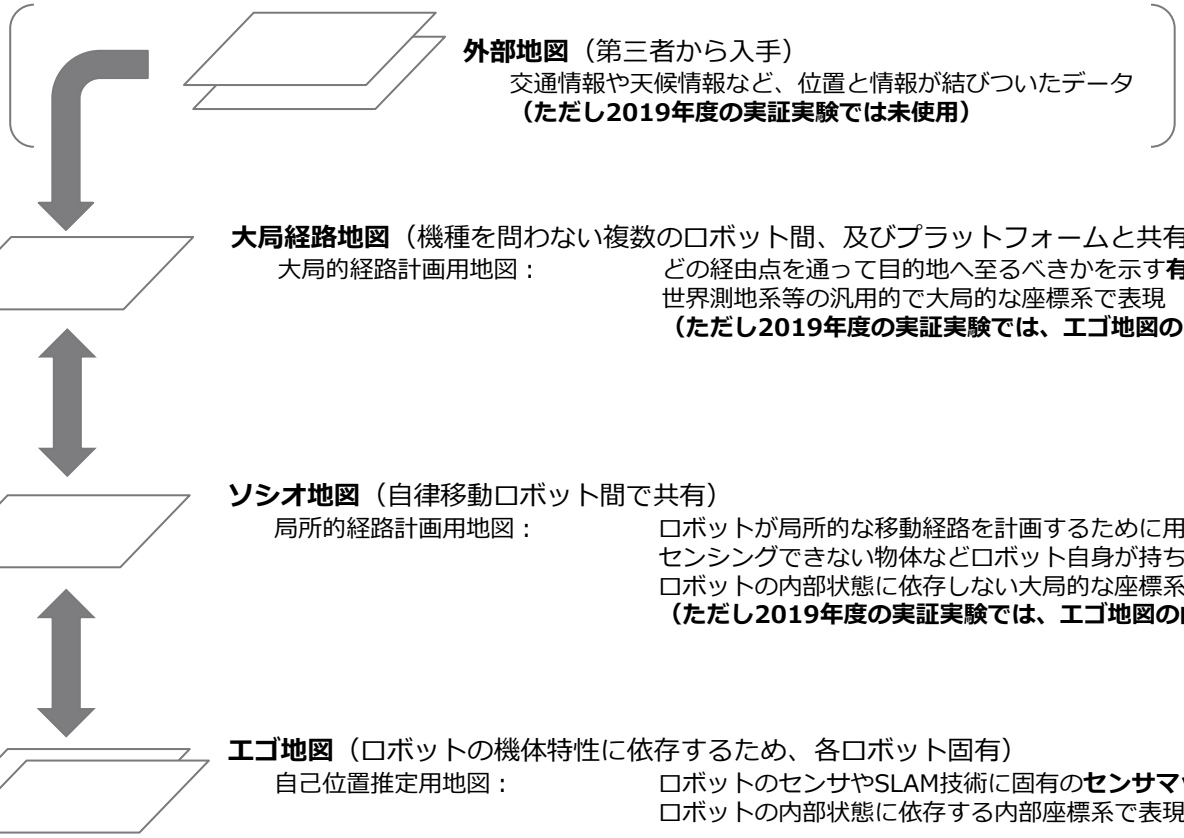
目次：

研究概要

2019年度の実証実験

階層化された地図による自律移動手法の実現性の確認

2019年度の実証実験では、異機種複数ロボットの統合管理システムの有効性を示す一環として、多層構造を持つ地図の概念とそれを「自律移動ロボット用のダイナミックマップ」として機能させる技術確立を行った



外部地図（第三者から入手）
交通情報や天候情報など、位置と情報が結びついたデータ
（ただし2019年度の実証実験では未使用）

プラットフォームが保持

大局経路地図（機種を問わない複数のロボット間、及びプラットフォームと共有）
大局的経路計画用地図：　どの経由点を通って目的地へ至るべきかを示す**有向または無向グラフ**
世界測地系等の汎用的で大局的な座標系で表現
（ただし2019年度の実証実験では、エゴ地図の内部座標系を流用）

プラットフォームが保持し
各ロボットへ供給

**（2019年度の実証実験では
地図管理機能は未実装）**

ソシオ地図（自律移動ロボット間で共有）
局所的経路計画用地図：　ロボットが局所的な移動経路を計画するために用いる**コストマップ**
センシングできない物体などロボット自身が持ちえない情報が含まれる
ロボットの内部状態に依存しない大局的な座標系で表現
（ただし2019年度の実証実験では、エゴ地図の内部座標系を流用）

各ロボットが生成

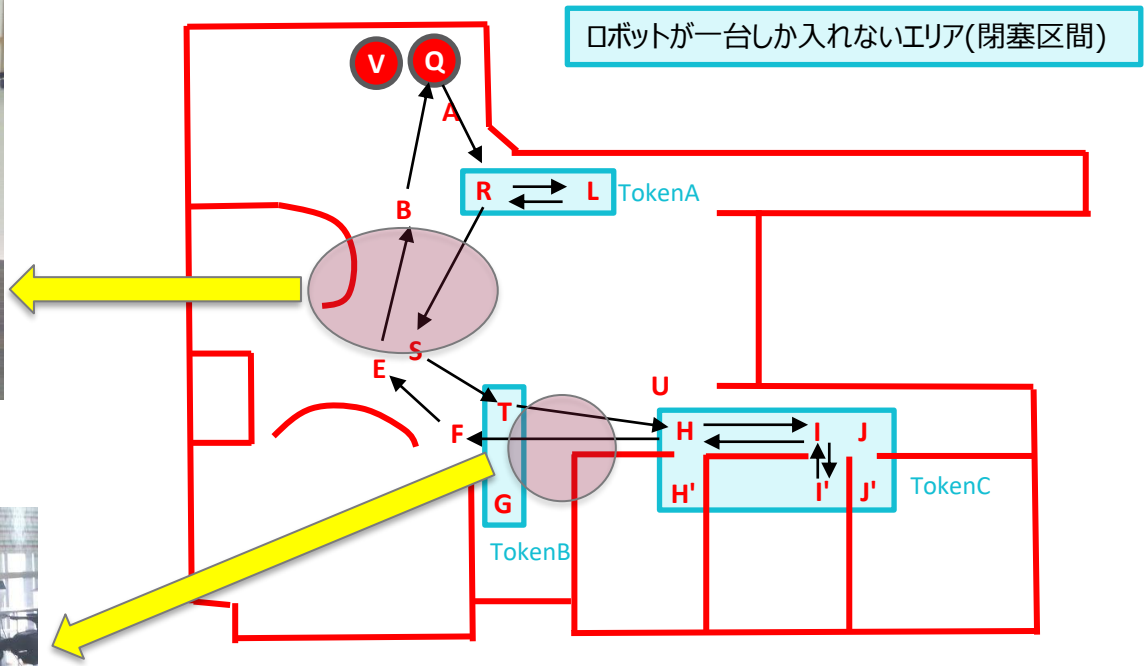
エゴ地図（ロボットの機体特性に依存するため、各ロボット固有）
自己位置推定用地図：　ロボットのセンサやSLAM技術に固有の**センサマップ**
ロボットの内部状態に依存する内部座標系で表現

在庫管理システムと自律移動配送ロボットを連携させたラストワンマイルの自動化に係る実証実験

2019年度の実証実験

大局経路地図の概要

広いスペースではロボットはすれ違い可能



自律移動ロボットが倉庫をめぐって目的地へ至り、出発点へ戻ってくる
大局経路な経路計画をロボット統合管理システムが有向グラフとして保持

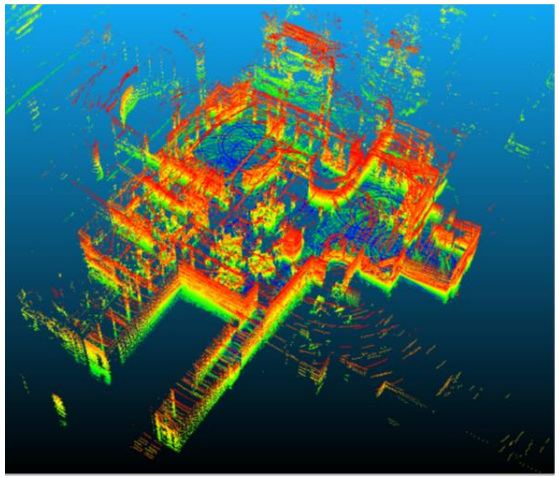
閉塞区間として定義された狭いスペースでは、先行ロボットが
閉塞区間内から脱出するまで後続ロボットは待機

2019年度の実証実験

エゴ地図とソシオ地図の関係

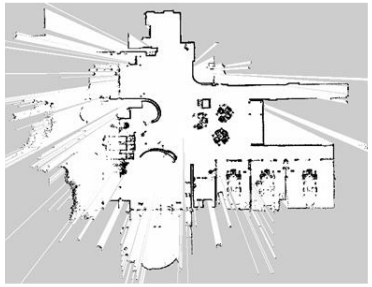
2019年度の実証実験では、三次元点群地図を二次元占有格子地図変換したものをエゴ地図とし、センサーでは見えない物体（ガラス等）を追記したものをソシオ地図として使用

【3D点群地図】



２D変換

【エゴ地図】



ガラス追加・雑点除去

【ソシオ地図】

ガラス部分（赤線）を
手動追加

在庫管理システムと自律移動配送ロボットを連携させたラストワンマイルの自動化に係る実証実験

2020年度の実証実験

問題点と解決方法

・大局経路地図上の各ノードとエッジのデータ登録を手作業で作成したため手間と時間がかかった

⇒ **メトリック地図を基に自律移動ロボットが走行可能な経路としてグラフ地図を作成**

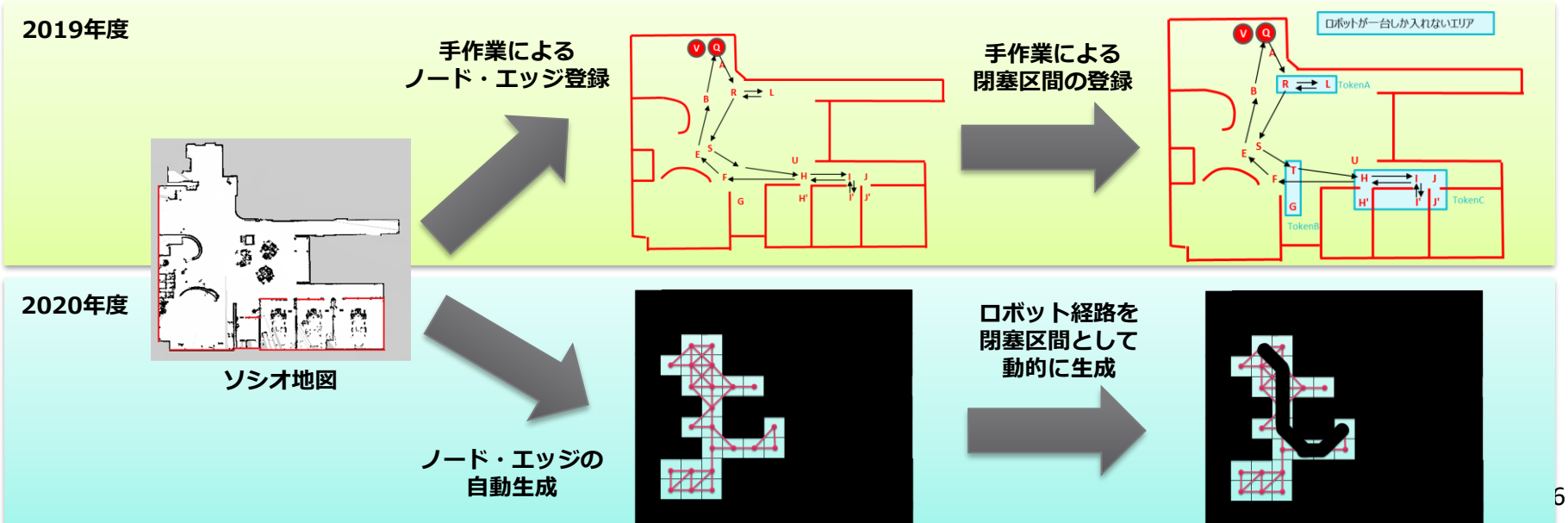
・大局経路地図上の閉塞区間は静的に生成しているため、確認のための無駄な停止などで余計な時間がかかる

⇒ **グラフ地図とロボットの運行状況を基にした大局経路生成**

・通過すべきノードを座標として指定したため、ノード上に障害物があるとロボットがノードに到達できずに立往生することがあった

・ロボットは経由点と目的地の区別がないため、ノード間を直線移動し全ノード上で立ち止まる動きをする。そのため移動に余計な時間がかかる

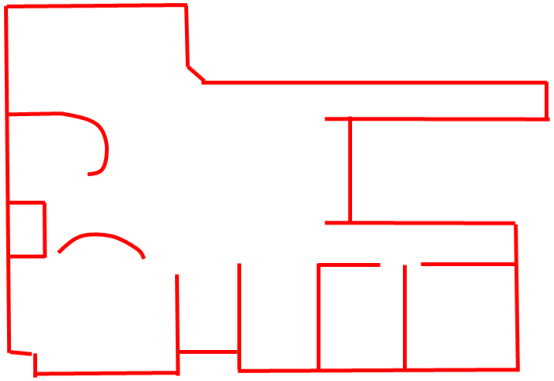
⇒ **移動経路と観測障害物を基にした効率的な局所経路計画**



実施概要

実証実験場所

■会津大学LICTiA（１F）



実証実験時期

■実施：2020/12/16（水）-12/18（金）
■予備：2020/12/21（月）

実証実験対象者

■なし　※サービスレベルの検証は行わないため

実証実験シナリオ

■なし

メガローバー

TurtleBot 3

実施体制と役割

TIS株式会社

ロボットを統合管理および制御するシステムのロボット統合管理システムの開発を担当

- ロボットへの状態管理やタスク管理
- ロボットの動的経路計画の実装

会津大学

学術的知見の提供を担当

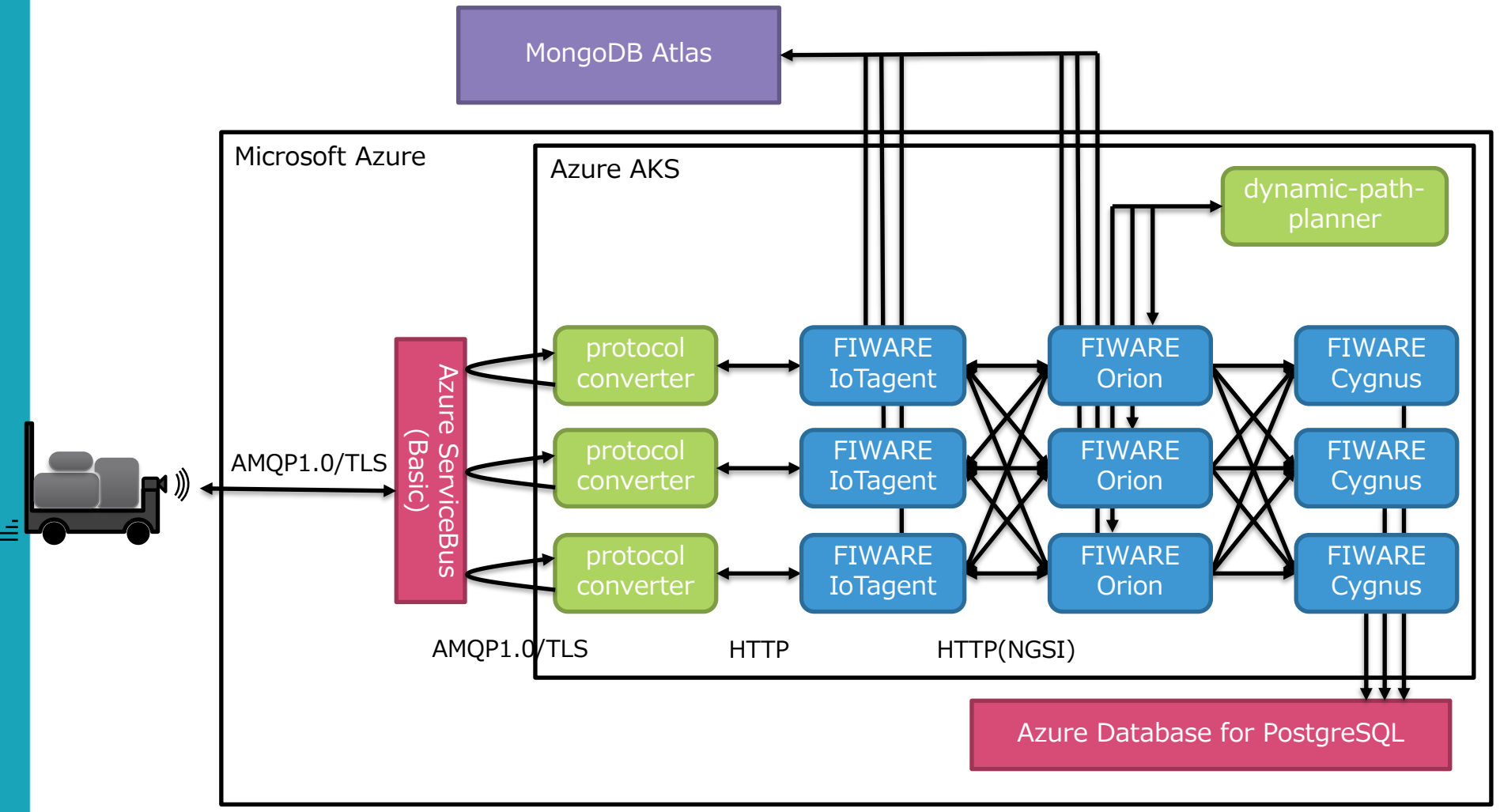
- アーキテクチャ全体の設計
- 動的経路計画アルゴリズムの設計
- グラフ生成アルゴリズムの設計と実装

株式会社日本アドシス

自律移動ロボットのソフトウェア実装とハードウェア環境設定を担当

- 自律移動ロボット内で使用する地図システムの構築
- 自律移動ロボット内で局所経路計画の実装
- 自律移動ロボット（Turtlebot3、メガローバー）の走行環境設定
- コストマップ反映処理の実装

ロボット統合管理システムのソフトウェア構成



自律移動ロボットの使用機器、ソフトウェア構成

No.	項目	ロボット1	ロボット2	備考
1	HW			
2	ロボット	Turtlebot3（waffle_pi）Ver.1.0	メガローバー Ver.2.0	二輪駆動
3	センサー			
4	LiDAR	Velodyne-VLP16 （水平360°/垂直30°、射程:0.50～100m）		FPGA Version:3.0.40.0 F/W Version:3.0.40.0
5	車輪エンコーダ	XM430-W210-Tエンコーダ	40W DC MOTORエンコーダ	
6	SW			
7	下位制御（ロボット稼動、データ取得など）			
8	制御基板	OpenCR1.0（Arduino互換）	VS-WRC021（Arduino互換）	
9	制御プログラム	TB3用Arduinoプログラム	メガローバー用Arduinoプログラム	流用（改修有り）
10	上位制御（地図作成、ナビゲーション、クラウド連携など）			
11	PC	PC：Intel NUC7i5BNH CPU：Intel Core i5（2/4） Memory：16GB	PC：ECS LIVA Z Plus CPU：Intel Core i3（2/4） Memory：16GB	
12	OS	Ubuntu 16.04		
13	フレームワーク	ROS Kinetic		
14	アルゴリズム			
15	地図作成（3D）	LOAM（Laser Odometry and Mapping）		流用（改修有り）/ チューニング
16	ナビゲーション（2D）	Navigation_Stack		
17	自己位置推定	amcl		流用/チューニング
18	大域経路探索	move_base		流用/チューニング
19	局所経路探索			
20	コストデータ反映	multi_robot_layer		新規（costmap_2d plugin実装）
21	ロボット制御統括	delivery_robot_node		新規（ゴール指示、エラー処理など）
22	ソシオ地図更新	map_organizer		新規
23	ROS-MQTTブリッジ	uoa_poc3_bridge		新規（TIS様ご担当）
24	コストデータ変換	Waypoint−Costmap translator		新規（TIS様ご担当）
25	状態表示			
26	PC	PC：DELL G7 CPU：Intel Core i9（6/12） Memory：16GB		
27	OS	Ubuntu 16.04		
28	フレームワーク	ROS Kinetic		
29	可視化プログラム	Rviz（ROS可視化Plugin）		流用/チューニング

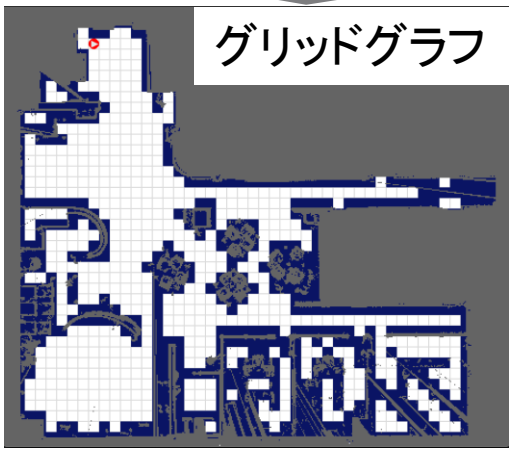
機能実装

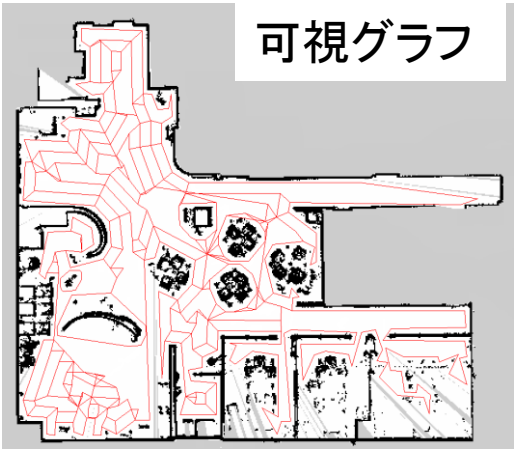
① メトリック地図を基にしたグラフ地図の作成

・以下を候補に生成するグラフ地図を策定

	グリッドグラフ	輪郭を基に生成した可視グラフ
生成例	作業空間を一定サイズの格子で分割し，障害物が含まれていない移動可能な格子をノード、隣接するノード間をエッジとして設定	障害物からある程度の距離をとった輪郭を複数生成し、障害物の含まれないように輪郭間の経路を引いていく
課題	作業空間が大きいときはノード数が多くなり，経路生成に時間がかかることがある	効率的なナビゲーションを実現できるノード位置の発見が難しい

実験で使用する環境は約25m四方、格子サイズは0.8mを使用するため、経路探索時間の問題は解消される。そのため本事象ではグリッドグラフを採用した





機能実装

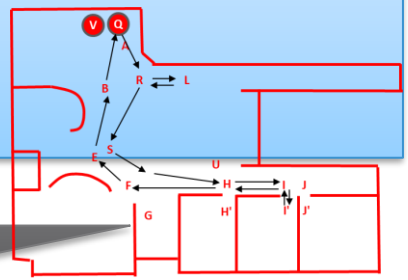
① メトリック地図を基にしたグラフ地図の作成

【2019年度の実装】

- 以下内容を手作業で設定

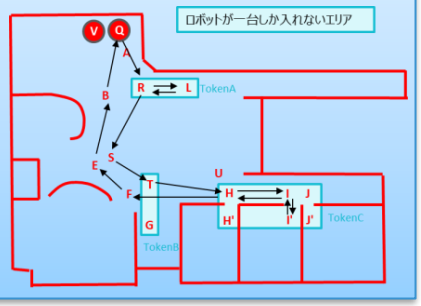
1. コストマップからノード及びエッジを作成

- 実空間の各ポイントをコストマップから探し出す
- ノード・エッジはシナリオに依存するため、作業内容に応じて大局経路地図を作成
- ロボット同士の衝突を避けるため、ロボットの数に応じた複線経路や待機場所を設定



2. 実空間の都合上1台しか入れないエリア(閉塞区間)を設定

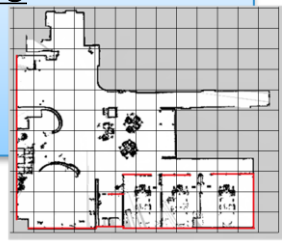
- 各閉塞区間でロボットが衝突しないための設定を大局経路地図に作成



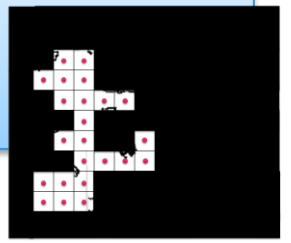
【今年度の実装】

- 以下内容を自動で設定

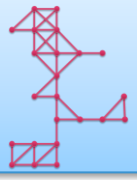
1. コストマップにグリッドを引き、領域を分ける



2. 領域内の障害物の有無から2値化し、障害物のない領域の中心点(ノード)を導出



3. 8近傍内にノードがあれば線を引く(エッジ)



機能実装

① メトリック地図を基にしたグラフ地図の作成

【2019年度の実装】

- 以下内容を手作業で設定

3. 大局経路地図上から目的地に応じた経路を使用

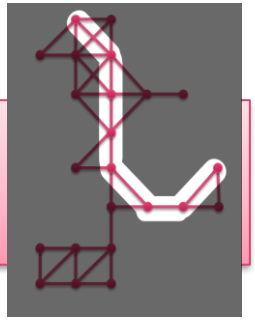
- 大局経路地図上に作成したノード・エッジから移動先に応じた経路を静的に選択
- 閉塞区間の手前で、他ロボットが閉塞区間内に存在するかを問い合わせる

【今年度の実装】

- 以下内容を自動で設定

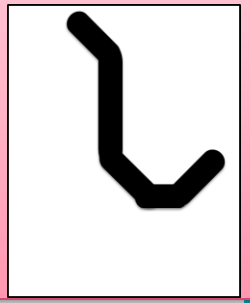
5. 大局経路地図から経路を動的生成

- 経路探索アルゴリズムとしてA*を採用



6. ロボットの位置と移動する経路をポテンシャル場として表現、大局経路地図に反映

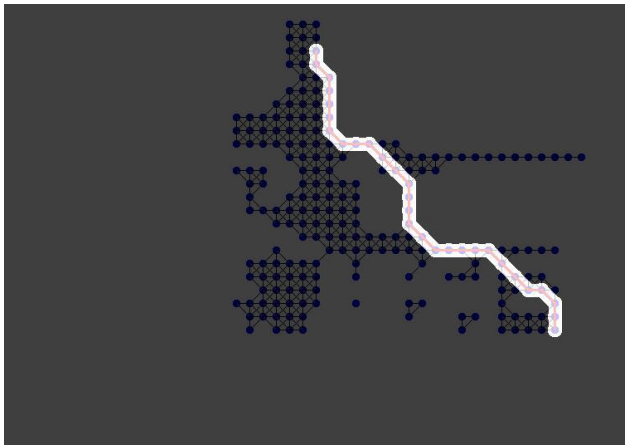
- 既存の**ポテンシャル場を探索時に考慮**することで他ロボットの経路を回避
- 通過済みのポテンシャル場は削除され、他ロボットの経路探索への影響を抑える
- 移動に使用する経路を**閉塞区間として動的に生成**することで、閉塞区間の管理が不要



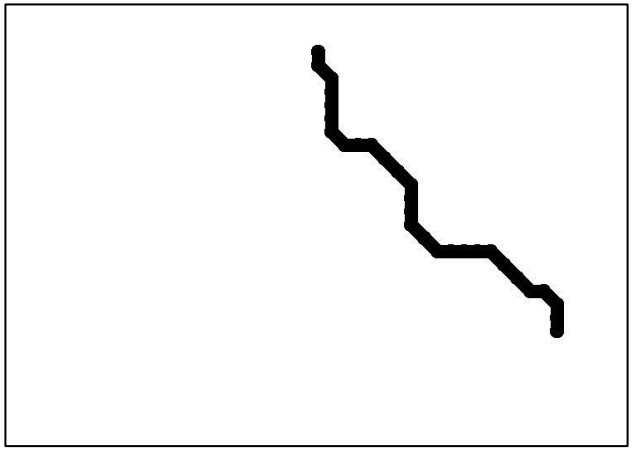
機能実装

② グラフ地図と全ロボットの運行状況を基にした大局経路生成

・グラフ地図から経路を探索
  – 経路探索アルゴリズムとしてA*を採用
・ロボットの位置と移動する経路をポテンシャル場として表現、大局経路地図に反映
  – 既存のポテンシャル場を探索時に考慮することで他ロボットの経路を回避
  – 通過済みのポテンシャル場は削除され、他ロボットの経路探索への影響を抑える
  – 移動する経路を閉塞区間として動的に生成することで、2019年度で行っていた閉塞区間の管理が不要



探索した経路



生成したポテンシャル場

機能実装

③ 移動経路のコストマップ反映

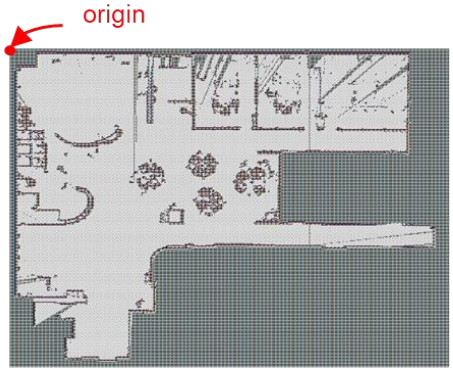
・WPリストのコストデータ変換
- **②大局経路生成**から受信したWPリストをコストマップに変換
- コストマップと目的地情報を移動指示メッセージとしてナビゲーションノードに送信
- コストマップの表現はcostmap_2d形式[1)]を使用

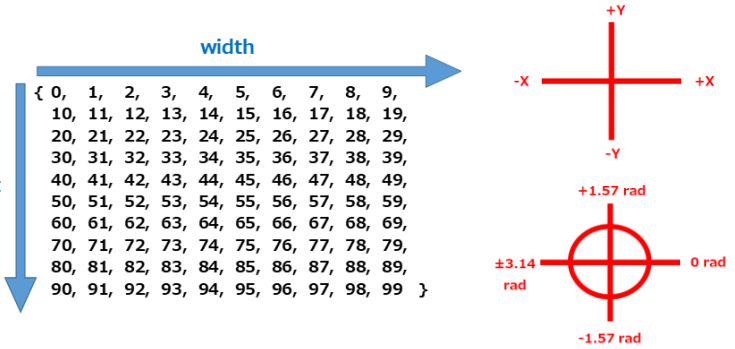
【移動指示メッセージ例】

```
{
 id: "mega_rover_01",
 type: "mega_rover",
 time: "2019-06-07T08:39:40.064+09:00",
 cmd: "navi",
 destination:
 {
    point: { x: 0.503, y: 0.0, z: 0.0 },
    angle_optional: {
      valid: false,
      angle: { roll: 0.0, pitch: 0.0, yaw: 0.0 }
    }
 },
 costmap:{
  resolution: 0.05,
  width: 10,
  height: 10,
  origin: {
    point: { x: 1.0, y: 1.0, z: 0.0 },
    angle: { roll: 0.0, pitch: 0.0, yaw: 0.0 }
  },
  cost_value: [ 0, 1, 0, 1, …
  ]
 }
}
```

- 個々のロボットごとに専用のROS Topicを設定する
- id,type,time,cmdは前回と同じ仕様
- destinationに目標地点の座標・姿勢を入れる
- resolutionはcostmapの1セルの単位
- widthは行数（X軸方向）、heightは列数（Y軸方向）
- originはcostmapの原点座標
- cost_valueは各セルのコスト、[width×height]個の値が入る
（cost_value配列の添え字とXY軸の関係は以下の図参照）
- cost_valueの値の取り方の例（ROS）
  - ・FreeSpace：0
  - ・任意コスト：1~253
  - ・Obstacle：254
  - ・UnknownSpace：255

※同メッセージ受信後、ロボット側のCostmapに253を超えない範囲でコストを加算





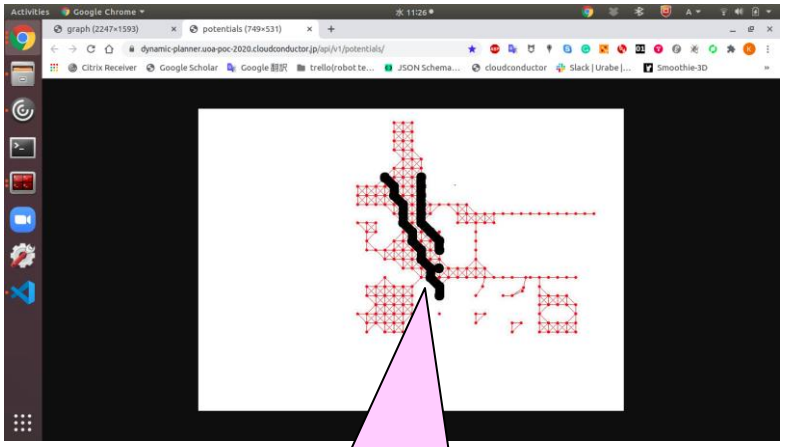
1) http://wiki.ros.org/costmap_2d

機能実装

③ 移動経路のコストマップ反映

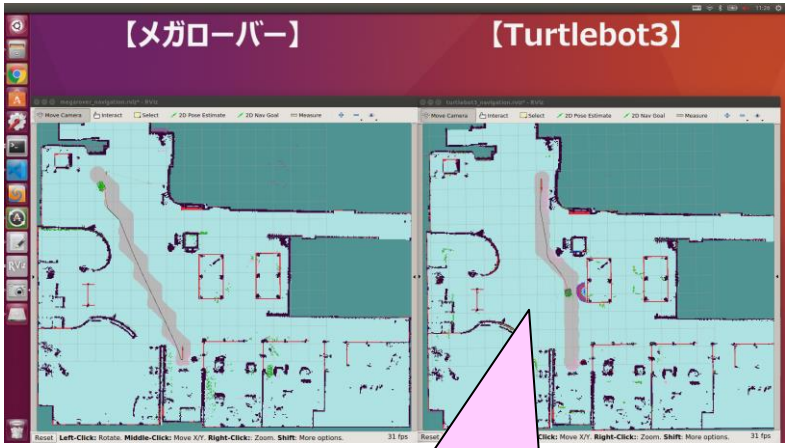
・コストデータの反映
  – 移動指示メッセージを受信したナビゲーションノードがコストデータをコストマップに反映し、ナビゲーションを行う
  – コストデータの反映は、costmap_2dのPlug-inとして実装
  – ポテンシャル場の外側のセルのコストは局所経路計画に使用されない”Inscribed”コストを設定

【上位側－大局経路地図】



黒色部分がポテンシャル場
（ロボットの移動経路）を指す

【ロボット側－コストマップ】



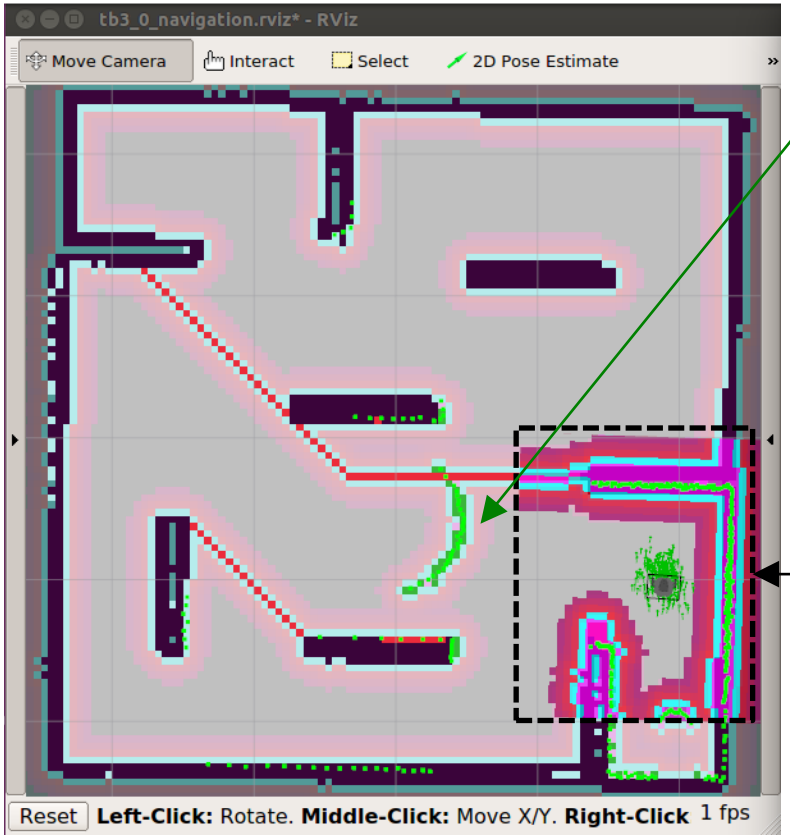
灰色箇所がコストフリー（通行可能）、
水色箇所がコスト高（通行不可）

機能実装

④ 観測障害物の動的地図追加

・観測障害物のコストマップ反映
  – ROS-costmap_2dのObstacle/Inflation layer機能を拡張
  – 15m範囲内の観測データを全てコストマップに反映（観測距離はセンサー測距範囲内で調整可）

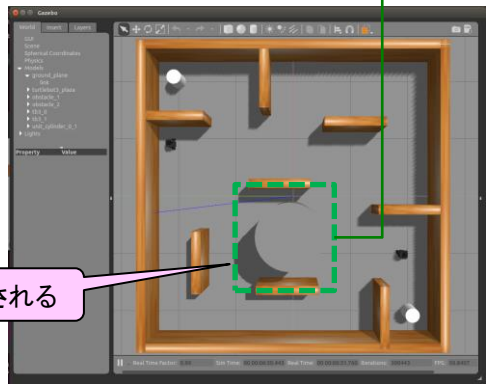
【コストマップ表示画面】



観測データ
（障害物回避機能の範囲外のセンサーデータを
障害物としてコストマップに反映）

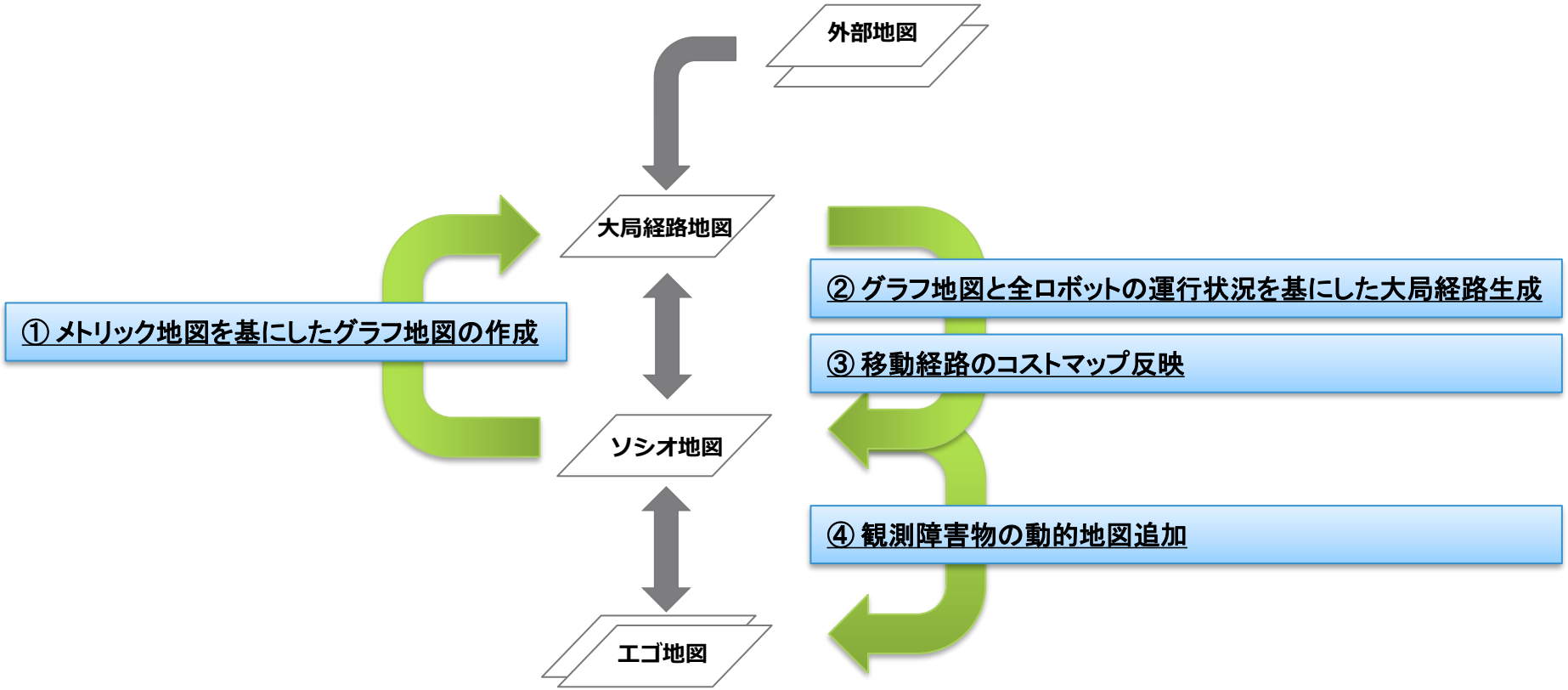
障害物回避機能の範囲
（2019年度は同範囲内でのみ動的障害物を検知）

【実環境】



ナビゲーション中に障害物が設置される

各機能と階層型地図との関係性



実証結果

検証項目

1. 単体ロボットのナビゲーション検証

2019年度の実証実験シナリオに沿った経路を、本実証で作成した機能を使って経路探索・自律移動まで一貫して行えることを確認

2. ロボット同士の衝突回避検証

2019年度では衝突回避のために、2台通行可能な経路では複線経路を、通行不可の経路では閉塞区間を用意しその管理を行っていた

ロボットが衝突する恐れのある状況を再現し、動的経路計画から与えられた経路で衝突回避することを確認

3. 動的経路計画の検証

2019年度では静的な経路を用意していたため、障害物の有無にかかわらず常に同じ経路を選択していた

他ロボットが経路をふさいでいる場合とそうでない場合で異なる経路を選択することを確認

4. 与えられたポテンシャル場内での障害物回避の検証

2019年度では経由点上にロボットがいる場合、ロボットが立ち往生する場合があった

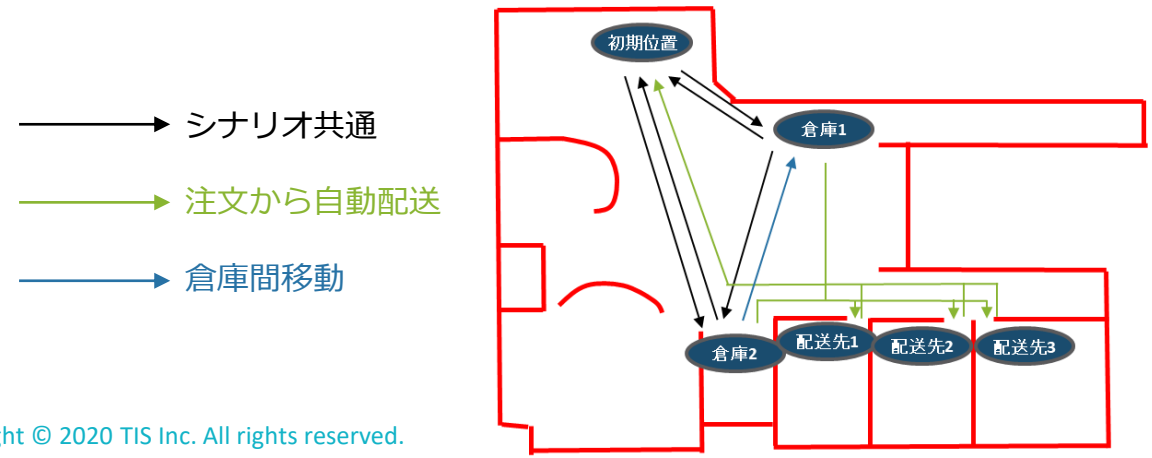
与えられたポテンシャル場内に障害物を配置、回避行動をとることを確認

検証結果
1. 単体ロボットのナビゲーション検証 【検証概要】

2019年度の実証実験シナリオに沿った経路を、本実証で作成した機能を使って経路探索・自律移動まで一貫して行えることを確認

2019年度実証実験のシナリオ

- 注文から自動配送
  1. ロボット管理システムは与えられた注文と配送先住所に応じて、倉庫を経由したウェイポイントを生成し、ロボットへ通知
  2. 配送ロボットはウェイポイントに従って自律移動を行う
- 倉庫間移動
  1. ロボット管理システムは移動させる品物に応じた、倉庫間移動のウェイポイントを生成し、ロボットへ通知
  2. 配送ロボットはウェイポイントに従って自律移動を行う



経路一覧

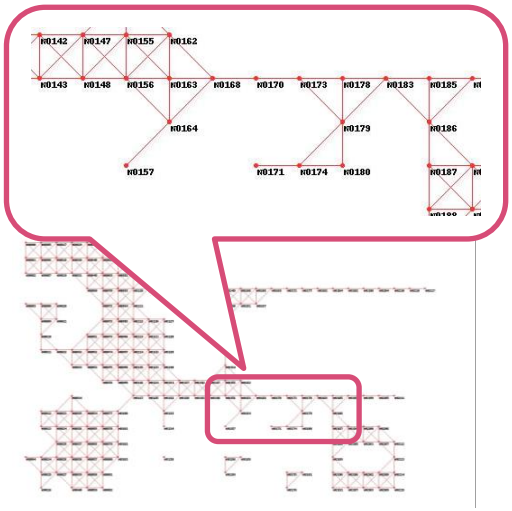
No.	スタート	ゴール	使用するロボット
1	初期位置	倉庫1	メガローバー
2	倉庫1	倉庫2	メガローバー
3	倉庫2	倉庫1	メガローバー
4	倉庫1	配送先1	メガローバー
5	配送先1	初期位置	メガローバー
6	初期位置	倉庫2	メガローバー
7	倉庫2	配送先1	メガローバー
8	倉庫1	配送先2	メガローバー
9	配送先2	初期位置	メガローバー
10	倉庫1	配送先3	メガローバー
11	配送先3	初期位置	メガローバー
12	倉庫2	配送先2	メガローバー
13	倉庫2	配送先3	メガローバー
14	倉庫1	初期位置	メガローバー
15	倉庫2	初期位置	メガローバー
16	初期位置	倉庫1	TurtleBot3
17	倉庫1	倉庫2	TurtleBot3
18	倉庫2	倉庫1	TurtleBot3
19	倉庫1	初期位置	TurtleBot3
20	初期位置	倉庫2	TurtleBot3
21	倉庫2	配送先1	TurtleBot3
22	配送先1	初期位置	TurtleBot3
23	倉庫2	配送先2	TurtleBot3
24	配送先2	初期位置	TurtleBot3
25	倉庫2	配送先3	TurtleBot3
26	配送先3	初期位置	TurtleBot3
27	倉庫1	配送先1	TurtleBot3
28	倉庫1	配送先2	TurtleBot3
29	倉庫1	配送先3	TurtleBot3
30	倉庫2	初期位置	TurtleBot3

検証結果
1. 単体ロボットのナビゲーション検証 【検証結果】

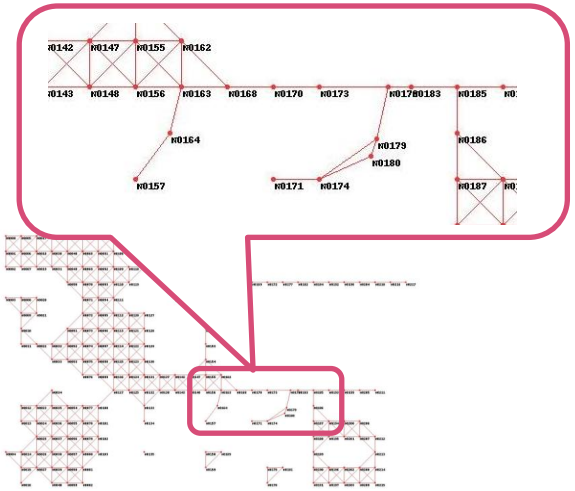
結果：一部ノードの座標を修正し全経路での自律移動を確認

- 狭い場所での移動が困難であることが分かった
  - グリッドグラフを自動生成した際に、会議室ドア付近のような狭い場所で選択できるノードが少なく、適切なノードが適切な場所に無い可能性がある
  - そのため手作業で一部ノードをロボットが通行できるように修正を行った

※ UIを用意しグラフ地図の補正を簡便化



調整前



調整後

各配送先の入口は
ロボットにとっては狭く
経路が限られるため移動が困難

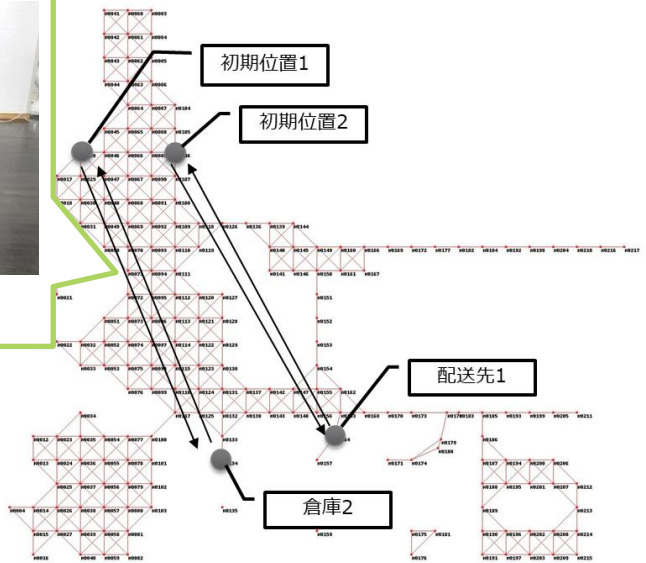
検証結果
2. ロボット同士の衝突回避検証 【検証概要】

2019年度では衝突回避のために、2台通行可能な経路では複線経路を、通行不可の経路では閉塞区間を用意しその管理を行っていた
出発地点と目的地点が重複する状況であっても、各ロボットに対して異なる経路を与えることで、ロボット同士が衝突しうる状況を未然に防止することを確認

（１）同じエリアをロボット同士がすれ違う場合

- 2019年度の経路からすれ違いが起こりうる経路を8パターン選出
- 動作時に各ロボットが同士の衝突を回避することを確認

すれ違いを起こすエリア



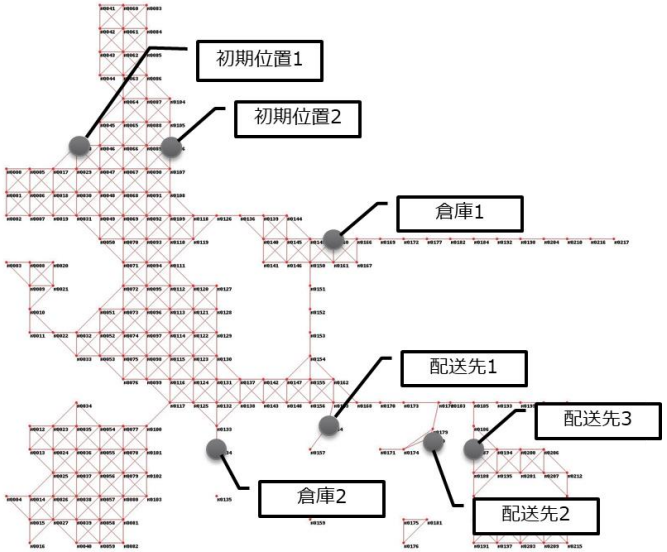
No.	メガローバー		TurtleBot3	
	スタート	ゴール	スタート	ゴール
1	初期位置1	倉庫2	倉庫2	初期位置2
2	倉庫2	初期位置1	初期位置2	倉庫2
3	配送先1	初期位置1	初期位置2	配送先1
4	初期位置1	配送先1	配送先1	初期位置2
5	初期位置1	倉庫2	配送先1	初期位置2
6	倉庫2	初期位置1	初期位置2	配送先1
7	初期位置1	配送先1	倉庫2	初期位置2
8	配送先1	初期位置1	初期位置2	倉庫2

検証結果
2. ロボット同士の衝突回避検証 【検証概要】

2019年度では衝突回避のために、2台通行可能な経路では複線経路を、通行不可の経路では閉塞区間を用意しその管理を行っていた
出発地点と目的地点が重複する状況であっても、各ロボットに対して異なる経路を与えることで、ロボット同士が衝突しうる状況を未然に防止することを確認

（２）ロボットが存在する場所へ他のロボットを移動させた場合

- 2019年度の経路からすれ違いが起こりうる経路を5パターン選出
- 動作時にロボット同士の衝突を回避することを確認



【実施手順】
1.ロボットAを目的地へ移動
2.ロボットBを目的地へ移動
3.ロボットBは移動せずに待機
4.ロボットAが目的地から移動
5.経路が解放されたタイミングでロボットBが目的地へ移動開始

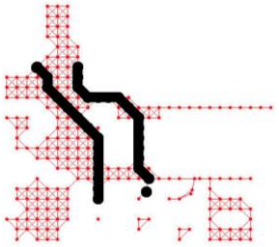
No.	ロボットA	ロボットB	目的地
1	メガローバー	TurtleBot3	倉庫1
2	メガローバー	TurtleBot3	倉庫2
3	メガローバー	TurtleBot3	配送先1
4	メガローバー	TurtleBot3	配送先2
5	メガローバー	TurtleBot3	配送先3

検証結果
2. ロボット同士の衝突回避検証 【検証結果】

（1）同じエリアをロボット同士がすれ違う場合

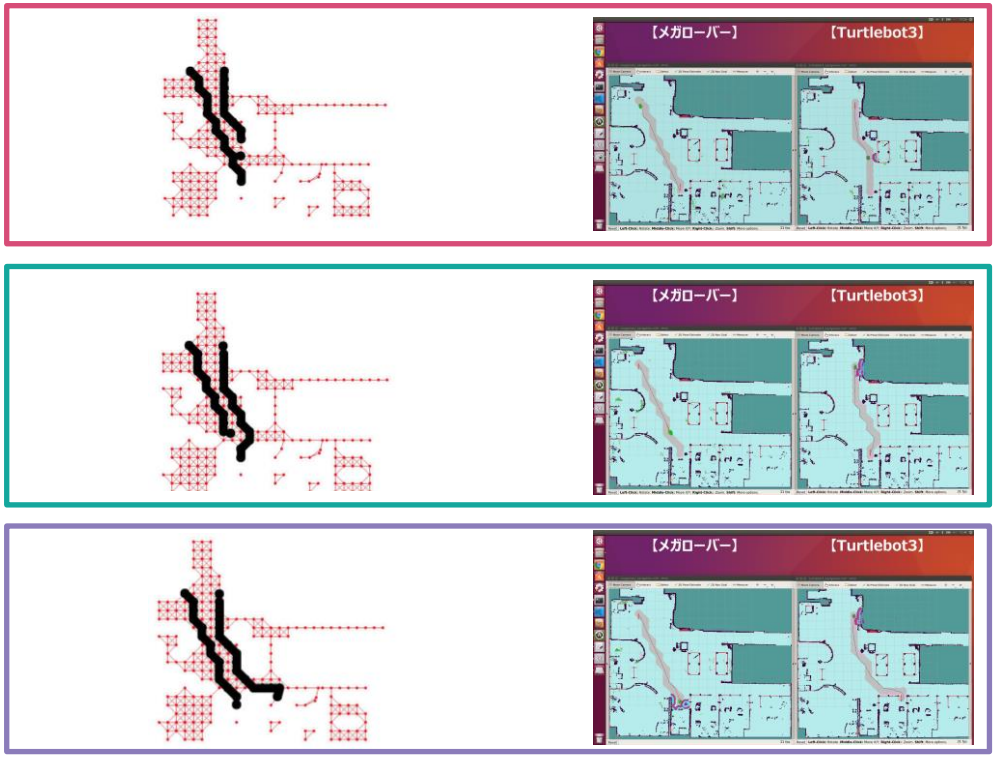
結果：8パターン中、3パターンですれ違い動作を実施

- 残り5パターンでは同エリア内でのすれ違いではなく、異なる通路を選択



異なる通路を選択した場合

- 動的経路計画からロボット同士が衝突しない経路を生成し、経路に従ってすれ違い動作が実現していることを確認



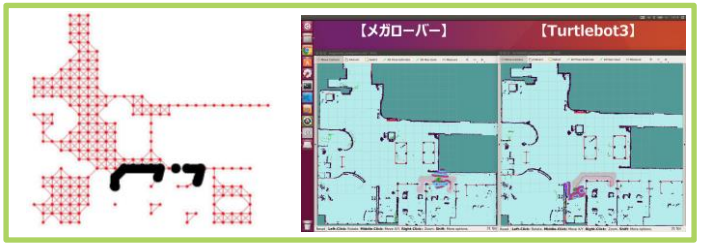
No.	メガローバー		TurtleBot3	
	スタート	ゴール	スタート	ゴール
1	初期位置1	倉庫2	倉庫2	初期位置2
2	倉庫2	初期位置1	初期位置2	倉庫2
3	配送先1	初期位置1	初期位置2	配送先1
4	初期位置1	配送先1	配送先1	初期位置2
5	初期位置1	倉庫2	配送先1	初期位置2
6	倉庫2	初期位置1	初期位置2	配送先1
7	初期位置1	配送先1	倉庫2	初期位置2
8	配送先1	初期位置1	初期位置2	倉庫2

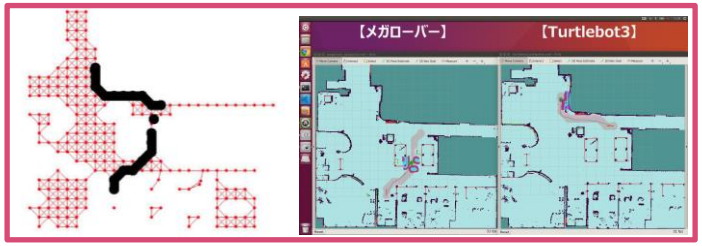
検証結果
2. ロボット同士の衝突回避検証 【検証結果】

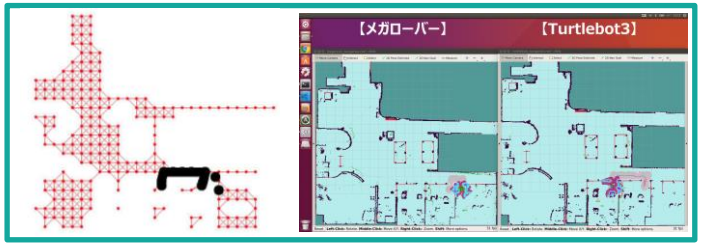
（2）ロボットが存在する場所へ他のロボットを移動させた場合

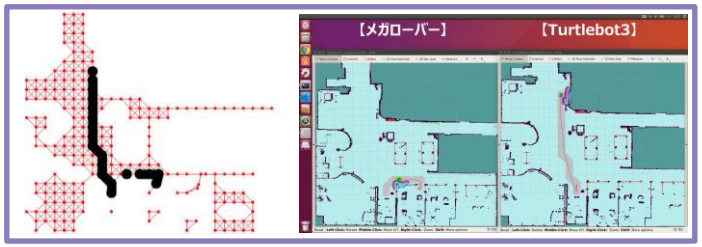
結果：すべてのパターンで待機動作を実施

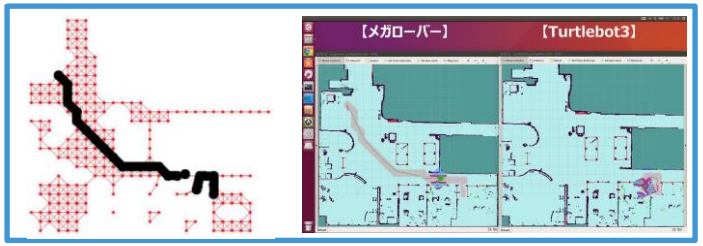
No.	ロボットA	ロボットB	目的地
1	メガローバー	TurtleBot3	倉庫1
2	メガローバー	TurtleBot3	倉庫2
3	メガローバー	TurtleBot3	配送先1
4	メガローバー	TurtleBot3	配送先2
5	メガローバー	TurtleBot3	配送先3





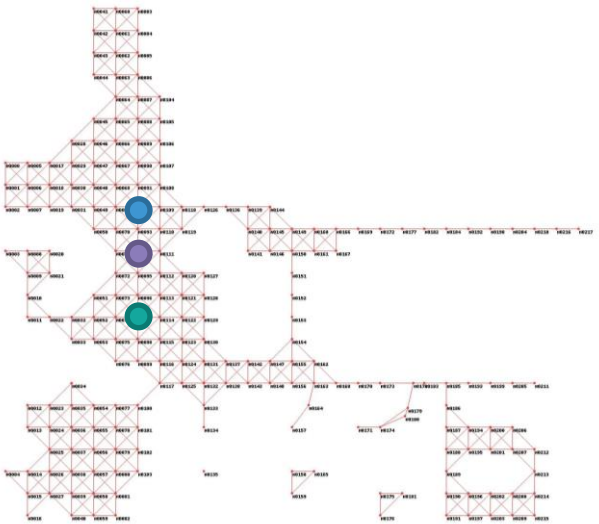






検証結果
3. 動的経路計画の検証 【検証概要】

2019年度では静的な経路を用意していたため、障害物の有無にかかわらず常に同じ経路を選択していた
他ロボットが経路をふさいでいる場合とそうでない場合で異なる経路を選択することを確認

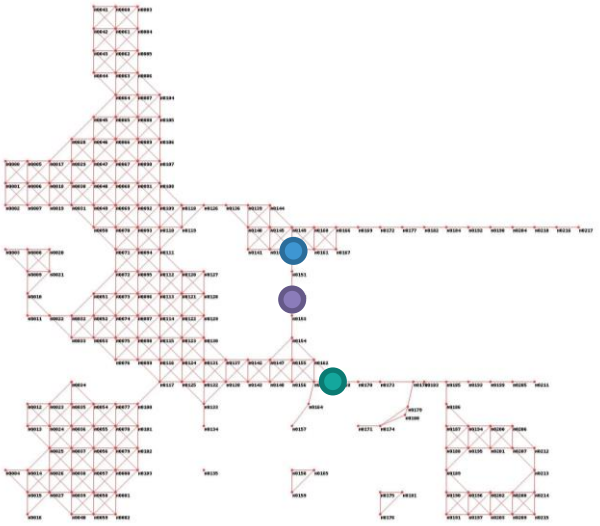


No. 1

移動元

移動先

封鎖箇所

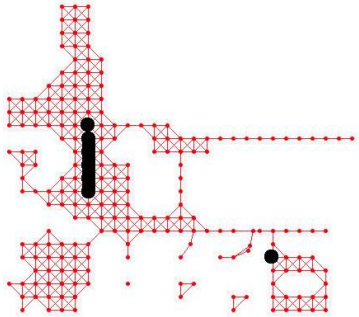
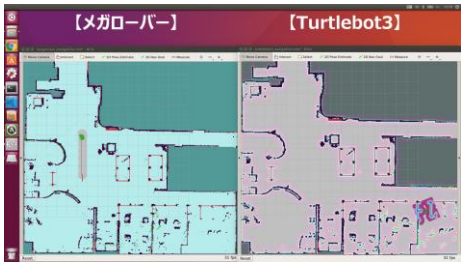
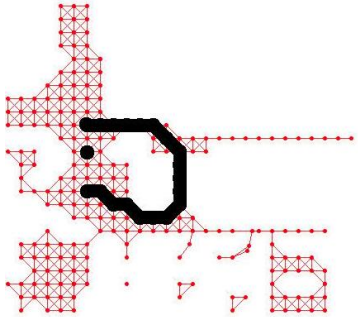
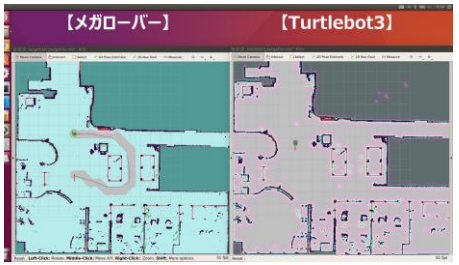
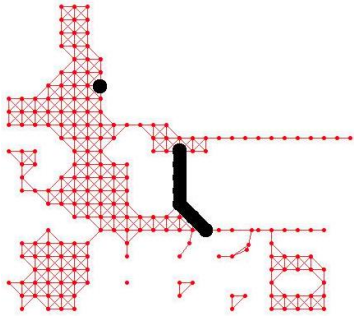
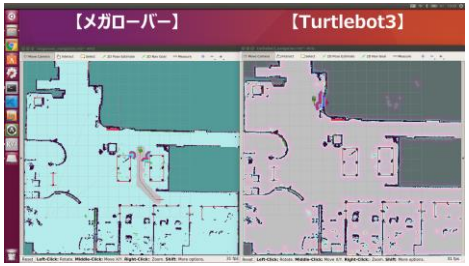
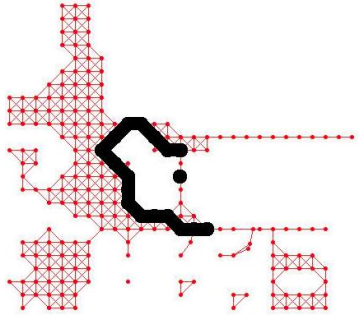
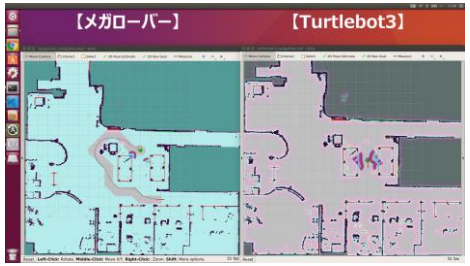


No. 2

検証結果
3. 動的経路計画の検証 【検証結果】

結果：全パターンでロボット位置を基に最適な経路を選択していることを確認

- 他のロボット位置を基に経路を生成することで、ロボット同士の衝突を回避した

	経路をふさいでいない場合		経路をふさいでいる場合	
No. 1				
No. 2				

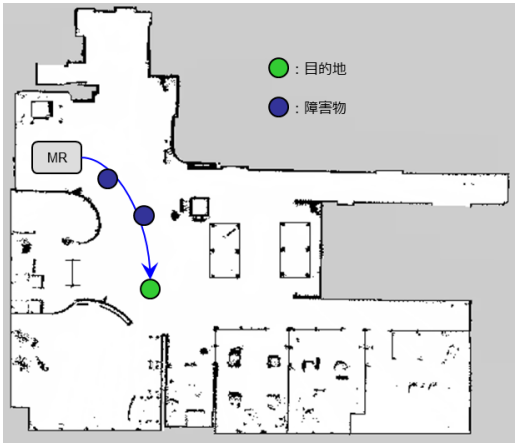
検証結果
4. 与えられたポテンシャル場内での障害物回避の検証 【検証概要】

ポテンシャル場により通行を制限されたエリア内に障害物（人）を配置し、障害物回避できるかを確認する

実施手順

1. ロボットを移動元へ移動
2. ロボットを移動先へ移動
3. 障害物Aが通行可能エリア内でロボットの前に立つ
4. 障害物Bが通行可能エリア内でロボットの前に立つ

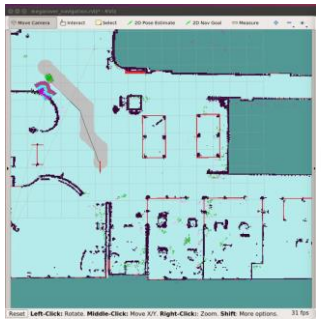
No.	ロボット	移動元	移動先	封鎖箇所
1	メガローバー	玄関前	カフェ前	経路上2か所
2	TurtleBot3	玄関前	カフェ前	経路上2か所



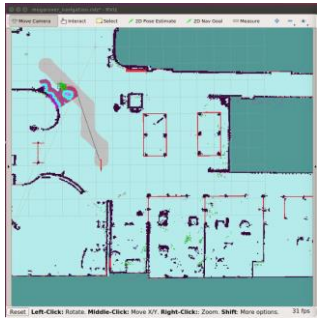
検証結果
4. 与えられたポテンシャル場内での障害物回避の検証 【検証結果】

【結果】

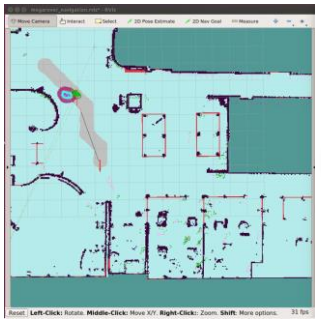
・動作に問題ないことを確認。但しポテンシャル場により通行可能なエリアを制限している状況での障害物回避のため、ロボットが障害物に接触寸前の経路を取る動作となった。



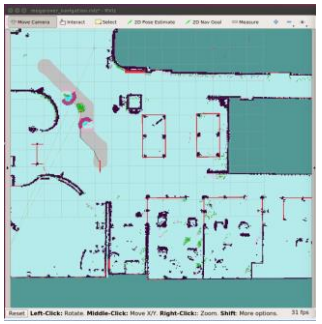
移動開始



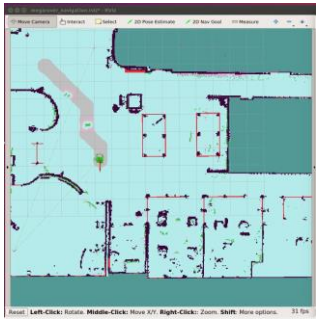
要員Aを検知、経路再探索



要員Aを回避



要員Bを検知、経路再探索



要員Bを回避、ゴール到達

No.	ロボット	移動元	移動先	封鎖箇所
1	メガローバー	玄関前	カフェ前	経路上2か所
2	TurtleBot3	玄関前	カフェ前	経路上2か所

まとめ

まとめ【課題と考察】

1. グラフ生成

【2019年度の課題】
大局経路上の各ノードとエッジのデータ登録を手作業で行ったため手間と時間がかかった

【2019年度から改善した点】
ソシオ地図を基にグラフ地図を自動的に生成した

【課題】
1. 部屋の入り口などの配置によって，単一のグリッド地図では実現可能な経路の表現が困難である
   - 今年度は手作業による調整で対処したため、完全な自動化の実現ではない
2. グラフの位置情報は基となるソシオ地図の座標を使用しているため、ロボットの内部座標が異なると同じグラフが使用できない
3. グリッドグラフ生成時に、障害物の有無にかかわらずエッジが自動設定されてしまう

【次年度以降で活動する予定の解決案】
1-i. 異なるオフセットを持つ複数のグリッド地図を導入し，ロボット位置によってそれらを切り替えて経路計画を行う
1-ii. ロボットのサイズやセンサー精度およびソシオ地図の輪郭を基にした最適なグリッドサイズの推定
1-iii. グラフ生成時にポイントを指定し、それらポイントに到達可能なグラフ地図を自動生成
2. 緯度・経度などの測地系座標等を用いてグラフ地図座標を表現し、ロボット内部座標との相互変換を導入
3. エッジ生成を引く際に障害物の有無を確認する機能の追加

まとめ【課題と考察】

2. 大局経路計画

【2019年度の課題】
大局経路地図上の閉塞区間は静的に生成しているため、確認のための無駄な停止などで余計な時間がかかる

【2019年度から改善した点】
グラフ地図上から閉塞区間を自動で生成できるようにした。ポテンシャル場による経路の有無を判断、動的にロボット(が存在する可能性)をよけた経路を生成するため無駄な立ち止まり等が解消された

【課題】
1. ロボット以外の障害物情報は大局経路地図には反映していないため、障害物がある場合に最適経路の算出ができない場合がある
2. 通過する可能性のある経路すべてを閉塞区間としているため、有効な経路が見つからない場合はスタート地点で経路が空くまで待機する

【次年度以降で活動する予定の解決案】
1-i. ロボットやIoTセンサーから障害物情報を検出し、大局経路地図へ反映
⇒ 動体物・静止物体等の障害物の情報をいつまで保持すべきか、階層化された地図内のどこで管理すべきか等考察が必要
2-i. 経路探索を時間軸へ拡張することでロボット位置の未来予測を行う
⇒ 各ロボットによる閉塞区間と移動時間の短縮が期待できる。しかし他動体物の影響などによる予測の変動なども考慮しなければならないため、安全と効率のバランスが重要となる
2-ii. 目的地までの経路が見つからない場合に、目的地付近まで移動する経路を探索し、閉塞区間が空けるまで待機する
⇒ 移動にかかる時間が短縮されるが、待機位置の場所次第ではデッドロックが起こりうるため、待機場所を限定するなど配慮が必要

まとめ【課題と考察】

3. 局所経路計画

【2019年度の課題】

1. 通過すべきノードを座標として指定したため、ノード上に障害物があるとロボットがノードに到達できずに立往生することがあった
2. ロボットは経由点と目的地の区別がないため、ノード間を直線移動し全ノード上で立ち止まる動きをする。そのため移動に余計な時間がかかる

【2019年度から改善した点】

1. 通過すべきノードをコストマップとして指定したため、コストマップの幅次第では障害物の回避行動が可能。コストマップを障害物が塞いだ場合もポテンシャル場を自主的に拡張し障害物を回避する
2. 複数WP指定による直線的な経路移動と違い、途中停止することなく曲線的な経路を取ることでスムーズな移動が可能

【課題】

1. コストマップが完全に塞がる場合、ロボットが立ち往生する恐れがある

【次年度以降で活動する予定の解決案】

1-i. コストマップ情報を移動可・不可の2択でなく、ロボット側で障害物を避ける判断ができる領域を追加する

1-ii. ロボット側で対処するのではなく、立ち往生時に上位系へ経路の再検索を依頼(大局経路の解決案と同様、障害物の情報を上位系に通知する必要がある)

※ロボットの自主性をどこまで許容するかに関わるため、ロボット性能やシナリオによって解決方法が異なると考えられる

THANK YOU