

# 災害時の無線通信路確立のための エージェント技術に基づくロボット運用方式の提案

林 幸汰<sup>1</sup> 打矢 隆弘<sup>1</sup> 内匠 逸<sup>1</sup> 木下 哲男<sup>2</sup>

**概要：**近年，災害に強い通信インフラとして無線 LAN(Local Area Network) が注目されている．災害発生時，電話回線が急激な需要増加に伴う輻輳により利用できない場合にも，無線 LAN はインターネットにアクセスできるため，災害時に有効な通信手段とされる．災害時の無線 LAN 利用における問題点として，被災によるアクセスポイントの故障から発生するデッドスポットが挙げられる．デッドスポット内は外部との連絡が取れないため非常に危険である．このようなデッドスポットに対する解決策として，無線中継機により，隣接するアクセスポイントから新たな無線通信路を確立し，デッドスポットをカバーする方法が考えられる．本研究では，エージェント技術に基づきロボットを運用することで，災害時に人の手を介することなく迅速に無線通信路を確立するシステムを提案する．

## Robot Utilization Method using Agent Technology for Establishment of Wireless Communication Path in Event of Disaster

Kouta Hayashi<sup>1</sup> Takahiro Uchiya<sup>1</sup> Ichi Takumi<sup>1</sup> Tetsuo Kinoshita<sup>2</sup>

### 1. はじめに

近年，災害に強い通信インフラとして無線 LAN (Local Area Network) が注目されている．災害発生時，電話回線が急激な需要増加に伴う輻輳により利用できない場合にも，無線 LAN はインターネットにアクセスできるため，災害時に有効な通信手段とされる．災害時の無線 LAN 利用における問題点として，被災によるアクセスポイントの故障から発生するデッドスポットが挙げられる (図 1)．デッドスポット内は外部との連絡が取れないため非常に危険である．

被災により発生したデッドスポットに対する解決策として，無線中継機により，隣接するアクセスポイントから新たな無線通信路を確立し，デッドスポットをカバーする方法が挙げられる．災害時に無線通信路を確立する上で以下の二つが問題となる．

**(P1)** 災害時という状況下を考慮すると，無線中継機を人が運ぶのは危険が伴うため困難である．

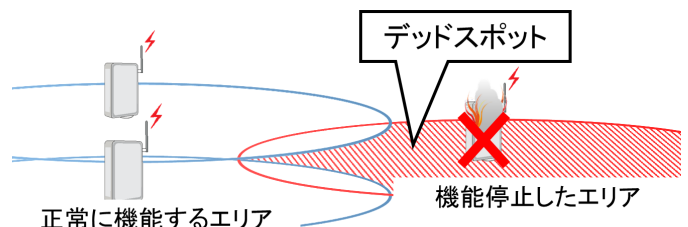


図 1 デッドスポット

**(P2)** 無線通信路の確立を迅速に行わなければ，安否確認・救助に支障をきたす．

本研究の目的は，上記の二つの問題点を解決する無線通信路確立システムを実現することである．提案システムではエージェント技術に基づきロボットを運用する．これによって，問題点 (P1)，(P2) の解決を図る．具体的には，エージェントがロボットを制御し，ロボットによる無線中継機の運搬を行う．さらに，マルチエージェントにより，ロボットの制御だけでなく，アクセスポイントの稼働状況の監視など，無線通信路確立に当たり必要となる全工程を自動化し，人の手を介することのない迅速な無線通信路の確立を実現する．これで (P1)，(P2) どちらも解決される．

<sup>1</sup> 名古屋工業大学 大学院 工学研究科 情報工学専攻

<sup>2</sup> 東北大学電気通信研究所

本稿では、前提知識としてエージェント技術と利用するロボットについて説明した後に、提案システムについて説明する。最後に評価実験を通して提案システムの実用性を検証し評価を行う。

## 2. エージェント技術

本章では、まずエージェント技術に関する基本的な知識について説明する。次に、エージェントフレームワーク DASH[1] について説明する。

### 2.1 エージェント

エージェントとは、外部環境、人、他のエージェント、他のプログラムとの相互作用によって自律的に動作するプログラムである。単体のエージェントは、単純な機能しか持たないが、複数のエージェントが分散・協調することで複雑な機能を持つ一つのシステムを構築することが出来る。エージェントという概念は現在、明確な定義が無く、様々なエージェントの定義が存在する。共通した概念として以下のようなものが挙げられる。

**自律性** 他のエージェントや人から直接干渉を受けずとも、自身で判断し動作する。

**社会性** 他のエージェントや他のプログラムと相互作用を行う。協調することで一つのタスクを処理する。

**反応性** 環境を認識し、変化に対し応答できる。

**自発性** 目的達成のために、能動的にシステムに参加し、他のエージェントと協調を行う。

### 2.2 エージェントシステム

エージェントシステムとは複数のエージェントによって構成されたシステムである。各エージェントは単純な機能しか持たない。エージェントシステムでは、問題を部分問題に分割し、各エージェントは部分問題を解決する。処理された部分解を統合することで大きな一つの問題の解決が可能になる。複数のエージェントが協調することで、処理の分散や構造の変更が容易になり、システムは柔軟性を持つことができる。

### 2.3 エージェントフレームワーク DASH

エージェントフレームワークとはエージェントシステムの効率的な開発や運用を実現するために必要な枠組みのことをいう。エージェントフレームワークは 2.1 節で紹介したエージェントの性質を再現し、エージェント間相互作用のための通信プロトコルや、自律性・自発性を得るための推論機構を提供する。現在、DASH や JADE[2] といったエージェントフレームワークが存在している。本節では、本研究で用いる DASH について詳しく述べる。

DASH とは「Distributed Agent System based on Hybrid architecture」の略であり、リポジトリ型エージェン

トフレームワークと呼ばれる。本節では、DASH の特徴・機能を説明する。

**概要** DASH は開発したエージェントを保存するリポジトリとエージェントが実際に動作する環境であるワークスペースで構成される (図 2)。リポジトリはワークスペースからの要求を受け、ワークスペースにエージェントを生成する。DASH ではエージェントが状況に応じてリポジトリに新たなエージェントの生成を要求することで、システムが柔軟性を持つ。

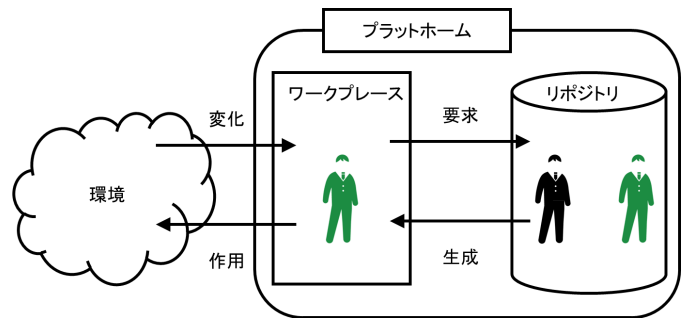


図 2 リポジトリ型エージェントフレームワーク

**ネームサーバ** ネームサーバとは、DASH におけるエージェント管理用のサーバである。ネームサーバはエージェントやリポジトリ、ワークスペースについての情報を保持する。分散環境上にエージェントシステムを構築する場合、異なる PC 上のエージェントとの通信にネームサーバを利用する。ネームサーバは一つのネットワークに一つだけ存在する。ネームサーバは以下の情報を保持する。

**環境情報** リポジトリ、ワークスペース環境名

**エージェント情報** エージェント名、生成時間、生成された環境名、現在の環境名、機能名

**DASH エージェント** DASH で作成するエージェントは、拡張子.dasj のテキストファイルとして記述される。DASH エージェントは、通信機構、情報処理機構、動作機構、ベースプロセスから構成される。エージェントは if-then 型で記述されたルールを複数持ち、ワーキングメモリに蓄積されたファクトとパターンマッチングを行うことで、ルールが発火しエージェントの動作が決定する。DASH エージェントは、ベースプロセスと呼ばれる Java プロセスを持ち、Java プロセスを制御することができる。ベースプロセスを用いることで、DASH 言語では行うことができない処理を Java で処理することができる。

**組織構成** エージェントシステムは、複数のエージェントで組織を構成する場合がある。DASH エージェントが通信を行うためには、相手のエージェント名、環境名が必要となる。組織構成の過程で各エージェントは、他エージェントのエージェント名、環境名を把握する。

### 3. 提案システムで利用するロボット

提案システムでは、無線中継機の運搬にロボットを利用する。提案システムで利用するロボットを選定する上で以下の二点を想定する。

- 屋外と屋内では運搬における障害が大きく異なる。
- 運搬が必要な地点は広範囲にわたって複数生じる。

上記二点の解決法として (S1), (S2) を提案する。

(S1) 屋外と屋内で異なるロボットを用いる。

(S2) 安価なロボットを複数台用意し、エージェントシステムにより効率よくロボット群を運用する。

#### 3.1 屋外で利用するロボット

提案システムにおいて屋外での無線中継機の運搬に利用するロボットとその開発ツール群について説明する。

**UAV** UAV(Unmanned Aerial Vehicle : 無人航空機) は飛行能力を有するロボットである。災害時、屋外の運搬における問題点として、瓦礫による悪路が挙げられる。そのため、陸路を利用しての運搬は高価な専用のロボットが必要と考えられる。UAV を利用し空路から運搬を行うことでこの問題を解決する。UAV として提案システムではマルチコプタ (図 3) を利用する。

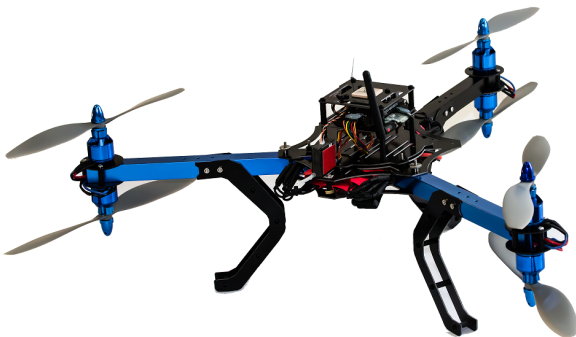


図 3 マルチコプタ

**DroneKit-python** DroneKit-python[3] は 3DRobotics 社 [4] により提供される UAV 制御用のライブラリである。DroneKit-python では Python を用いることにより、MAVLink<sup>\*1</sup>による UAV とコンピュータ間の通信を行うプログラムを記述することができる。DroneKit-python は ArduPilot<sup>\*2</sup>に基づいた機体の制御を行える。例として、フライトプランに基づく UAV への自律飛行の命令や、UAV から GPS 等のセンサ情報の取得などが行える。UAV を用いたシステムの

<sup>\*1</sup> 小型の UAV に広く用いられている制御用プロトコル

<sup>\*2</sup> MAVLink に基づくオープンソースの UAV プロジェクト

開発にはシミュレータを用いるのが一般的であり、DroneKit-python では実機だけでなくシミュレータ上の UAV を制御することもできる。本研究では次に説明する SITL シミュレータを用いて開発を行う。

**SITL シミュレータ** SITL(Software in the Loop) シミュレータとは仮想環境から機体まで全てをシミュレータによりまかなうシミュレータの方式である。DroneKit-python の開発で用いる SITL シミュレータ [5] では回転翼機や固定翼機の飛行・制御をシミュレートすることができる。また、MAVProxy[6] というソフトウェアを用いることで図 4 のように衛星写真でシミュレータ上の UAV の位置を知ることができる。

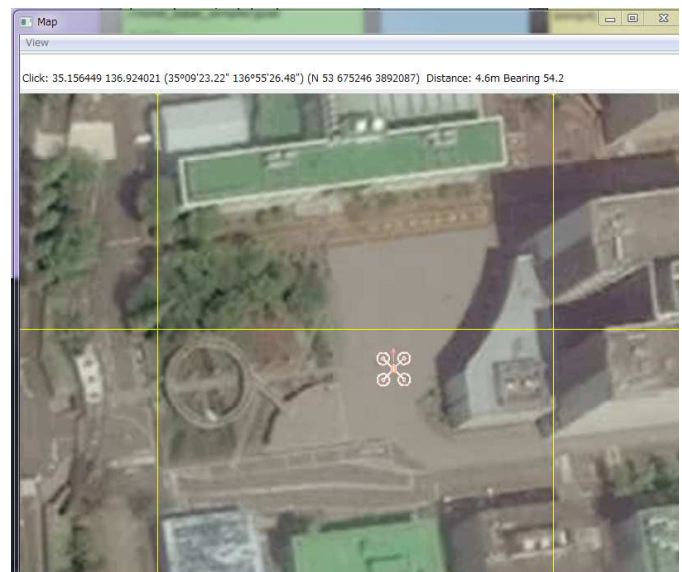


図 4 SITL と MAVProxy の利用

#### 3.2 屋内で利用するロボット

本節では、提案システムにおいて屋内での無線中継機の運搬に利用するロボットとその開発ツール群について説明する。

**UGV** UGV(Unmanned Ground Vehicle : 無人地上車両) は地上を走行するロボットである。屋内での運搬における問題点として、壁や障害物を避けながら移動する必要があることが挙げられる。提案システムでは、屋内での運搬には外界センサにより壁・障害物を避けながら安全に移動できる UGV を用いることでこの問題を解決する。なお、UGV は建物の各フロアに一台配備する。これは、階段を利用することで災害時の避難の妨げになることを避けるためである。

**Turtlebot2** 本研究では UGV として Turtlebot2[7] を利用する (図 5)。Turtlebot2 は後述する ROS[8] の標準プラットフォームとしてサポートされている UGV であり、ROS のライブラリを用いて自律移動を行える。

Turtlebot2は内界センサ\*3としてジャイロセンサとホイールエンコーダ，外界センサ\*4として Microsoft's Kinect[9] を搭載する．制御は Ubuntu\*5を搭載したラップトップから行える．移動部はホイールにより移動を行う構造となっており，段差を想定していない．

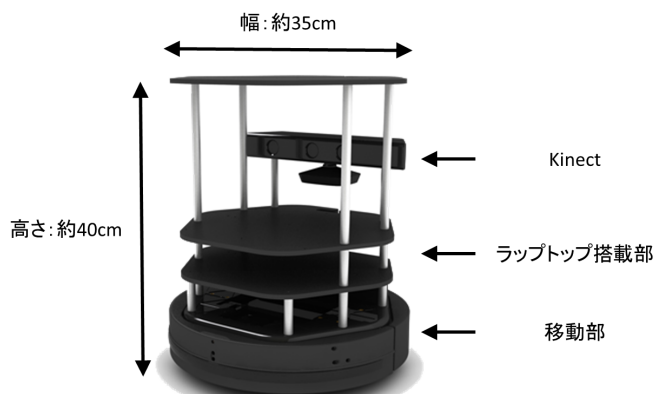


図 5 Turtlebot2

**ROS** ROS(Robot Operating System) は豊富なライブラリを備えるロボット開発用ミドルウェアである．ROSの開発は主に C++と Python により行う．ROSは世界中でロボット開発に広く用いられており，ROSに対応したロボットは多く存在する．ROSを利用してUGVの制御プログラムを開発することでシステムの汎用性を高めることができる．ROSを利用する利点として開発段階においては安価なロボットを利用し，実利用の段階ではより高価で高性能なロボットを利用できることが挙げられる．また，提案システムではUGVを建物の各フロアに配置するため，必要なUGVの数が増える．システムの汎用性を高めれば，フロアの大きさや重要度にあわせたスペックのUGVを用いることができるため，コストを抑えることができる．

**UGVにおける自律移動** UGVが自律移動を行うためには，目標地点がわかるマップと，マップ上の自己位置を知る手段が必要となる．屋外ではマップとして衛星写真を，自己位置の取得法としてGPSによる緯度・経度情報が利用できるが，屋内ではどちらも利用できない．また，屋内では壁や障害物を避けながら目標地点まで移動する経路を計画する手法も必要となる．ROSでは屋内マップを生成する手法としてSLAM(Simultaneous Localization And Mapping)，自己位置推定の手法としてAMCL(Adaptive Monte Carlo Localization)，経路計画のためのプランニングのライブラリが用意されている．

\*3 ロボット自身の状態を知るためのセンサ：ジャイロセンサ・加速度センサなど

\*4 ロボットの外の情報を知るためのセンサ：カメラ・レーダーなど

\*5 Linux ディストリビューションの一つ

## 4. 提案システム

無線通信路確立に当たり，以下の手順が必要となる．

- (1) デッドスポット出現確認のためのAPの監視
- (2) 無線中継機の運搬地点の決定
- (3) 運搬に最適なロボットの選択
- (4) 運搬地点までのロボットの制御

提案システムはこの四つの手順を行うために，ロボット，管理サーバ，エージェントシステムから構成される(図6)．エージェントシステムの開発・運用にはエージェントフレームワーク DASHを用いる．

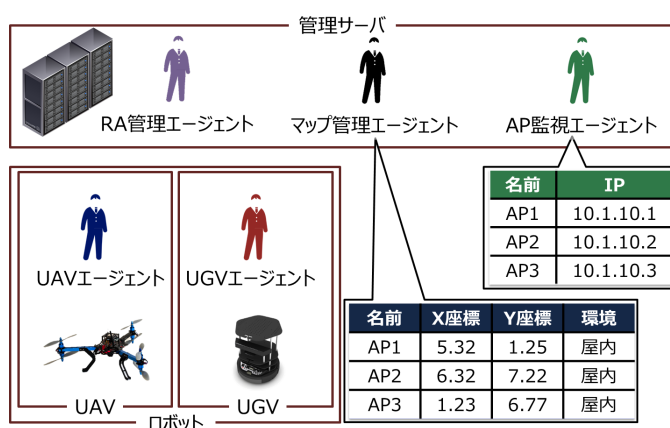


図 6 提案システムの全体図

**ロボット** 提案システムでは屋外において UAV，屋内において UGV を用いて無線通信路の確立を行う．本研究で用いるロボットは，UAVはコンパニオンコンピュータ\*6，UGVはラップトップをそれぞれ搭載しロボットの制御を行う．UAVはDroneKit-python，UGVではROSによって開発されたプログラムにより制御を行う．主に管理サーバとの通信を行うため，搭載しているコンピュータ上でDASHエージェントを稼働させる．エージェントと制御プログラムは共有メモリを介して通信する．

**管理サーバ** 提案システムに管理サーバを設置する．管理サーバは安定して通信を行える環境に設置され，ネームサーバを起動する．また，高い処理能力を持ち手順1～3を行う．提案システムをロボットとロボットに搭載されたコンピュータ上で動作するエージェントシステムだけで構成する場合，以下のような問題点が挙げられるため，管理サーバ導入によりこれを回避する．

### ネームサーバの設置

DASH エージェントの通信にはネームサーバの参照が必要である．ネームサーバはエージェントシステムの一つであり，安定して通信が行える環境に設置

\*6 ロボットの制御以外の処理を行うために搭載されるコンピュータ

される必要がある。特定のロボット上のコンピュータにネームサーバを設置する場合、ロボットがデッドスポット内にいる場合やロボットが故障した際に、システム全体が停止してしまう。

#### ロボット上のコンピュータの処理性能

提案システムでは安価なロボットを複数台運用するため、ロボット上のコンピュータの処理性能はあまり高くできない。手順1~3は高性能なコンピュータ上で行うのが望ましく、ロボット上のコンピュータで行うのに適さない。

**エージェントシステム** 提案システムのエージェントシステムは、異なる機能を有する複数のエージェントから構成される。エージェントは役割に適した環境上で動作する。動作する環境に応じてエージェントは以下の二つに分類される。

#### ロボットエージェント

UAV エージェント、UGV エージェント

#### 管理サーバエージェント

アクセスポイント監視エージェント、マップ管理エージェント、ロボット管理エージェント

### 4.1 アクセスポイントの監視

提案システムでは、デッドスポットが発生した場合、隣接するアクセスポイントの無線通信路を、運搬した無線中継機によって拡張することでカバーする。そのためには、デッドスポットがどこで発生したかを検知する方法が必要となる。管理サーバでは監視対象のアクセスポイントのIPアドレスを保持し、一定間隔でPingを送ることで稼働状況を監視する。Pingが一定回数連続して返ってこない場合、アクセスポイントが停止しデッドスポットが発生したと判断する。また、アクセスポイントの位置情報を保持することで、位置情報を基にデッドスポットがどこに発生したかを特定する。

### 4.2 運搬地点の決定

デッドスポットを特定した後は、無線中継機の運搬地点を決める必要がある。無線中継機の運搬地点は停止したアクセスポイントと隣接するアクセスポイントの間、かつ隣接するアクセスポイントの無線エリア内である必要がある。理想としてはアクセスポイントの座標を基に動的に運搬地点を決定することが望ましいが、以下の二つの制約がある。

- 運搬地点が隣接アクセスポイントの無線エリア内にならなければならない

図7のように停止したアクセスポイントと隣接するアクセスポイントの無線エリアが十分に重なっている場合、動的に中間点を運搬地点として問題ない。しかし、図8のように無線エリアが十分に重なっていない

場合、隣接アクセスポイントの無線エリア内に運搬地点が来るようにする必要がある。電波は壁の干渉などで減衰するため、無線エリアは環境によって変化する。動的な決定ではこのような場合には運搬地点を決めることができない。

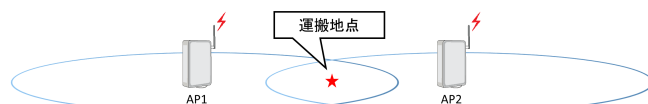


図7 無線エリアが十分に重なっている場合

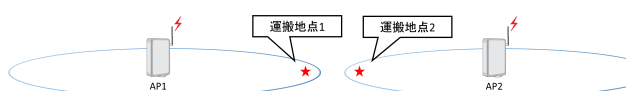


図8 無線エリアが重なっていない場合

- 運搬地点に障害物があるといけない

運搬地点に障害物があったり、運搬地点が壁の向こう側である場合が考えられる(図9)。このような場合運搬を行っても通信が確立できないか、運搬を行えない。

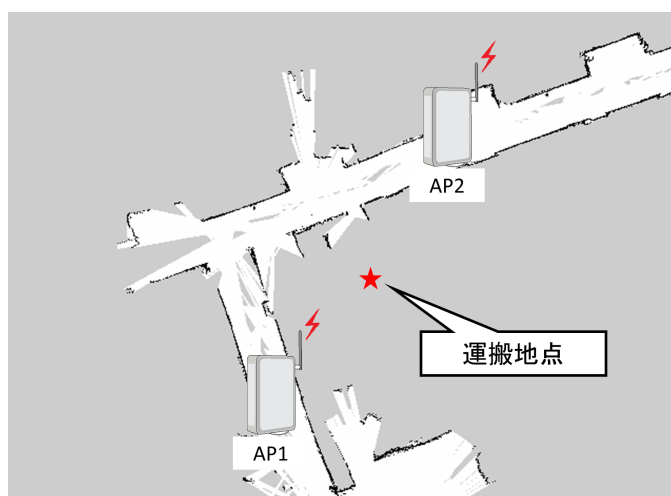


図9 運搬地点に障害物がある場合

以上の問題点を踏まえて、提案システムでは運搬地点の決定を静的に行う。

### 4.3 ロボットの選択

提案システムでは複数台のロボットを運用するため、運搬に最適なロボットの選択を行う必要がある。ロボットの選択は以下に示す考えの基に行う。

**屋外** UAVは空路を用いるため、壁などの障害物の影響を受けない。これは全てのUAVが全ての屋外運搬に対応できることを意味する。しかし、UAVには稼

働時間が短いという欠点がある。本研究で用いるマルチコプタは飛行時間が 30 分前後のものが市販されているものでは一般的である。これらのことから、屋外でのロボット選択では、AP との距離、バッテリー状況を考慮した選択が重要となる (図 10)。

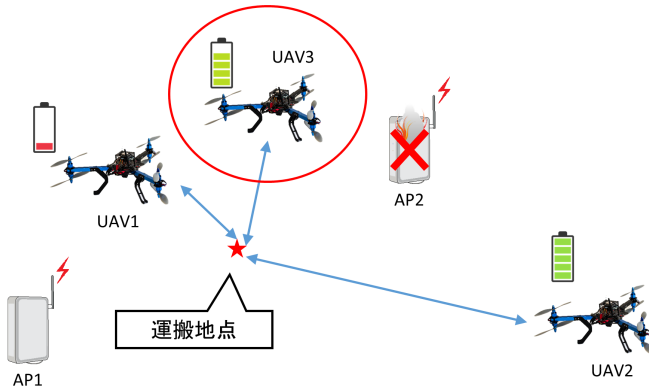


図 10 屋外でのロボット選択

**屋内** 本システムでは UGV を各階に最低一台を配置することを想定している。これは階段を移動する機能をもつ UGV は高価であり、各階ごとに安価な UGV を配置するほうがコストがかからないためである。また、階段は災害時に避難の経路として利用されるため、ロボットにより避難が滞ることはあってはならない。

上記の状況を考慮すると、屋内でのロボット選択において重要なのは、どの建物の何階にいいのかというマップ情報である (図 11)。UGV は UAV と比べて稼働時間の制約が少ないため、各階に多くの UGV を配置する必要もないと考えるため、座標情報、バッテリー残量は選択時の重要度が低い。

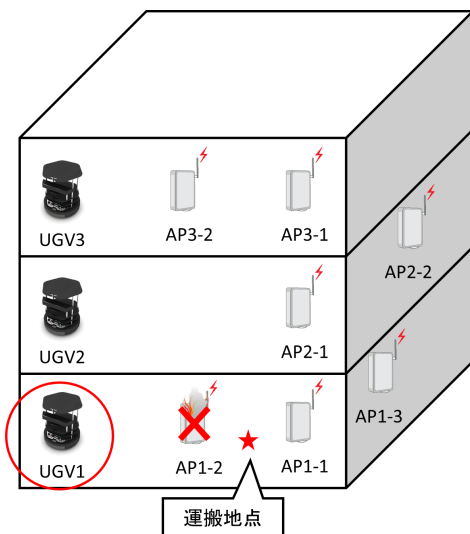


図 11 屋内でのロボット選択

#### 4.4 提案システムの処理の流れ

以下、提案システムの処理の流れを述べる (図 12)。

##### 1. 故障したアクセスポイントの特定

アクセスポイント監視エージェントは Ping が返ってこないことを AMA 管理エージェントに通知する。AMA 管理エージェントは IP アドレスから停止したアクセスポイント特定する。

##### 2. アクセスポイントの停止を通知

AMA 管理エージェントはマップ管理エージェントにアクセスポイントの停止を通知する。マップ管理エージェントは停止したアクセスポイントの地点にデッドスポットが出現したと判定する。

##### 3. 無線中継機の運搬地点の決定

マップ管理エージェントは停止したアクセスポイントと隣接するアクセスポイントの稼働を確認する。稼働していた隣接するアクセスポイントとの中間点を無線中継機の運搬地点とする。

##### 4. 適切なロボットエージェントの選択

マップ管理エージェントは RA 管理エージェントに運搬地点の情報を提供し、運搬に適したロボットエージェントの探索を依頼する。RA 管理エージェントは第 4 章で説明した選択基準に基づきロボットエージェントを選択する。マップ管理エージェントにロボットエージェントの名前と環境名を通知する。

##### 5. ロボットエージェントに無線中継機の運搬を要請

マップ管理エージェントは RA 管理エージェントが選択したロボットエージェントに無線中継機の運搬を要請する。

##### 6. ロボットを制御し無線中継機を運搬

運搬の要請を受けたロボットエージェントは名前付きパイプを使いロボット制御プログラムに運搬地点を入力する。ロボットが指定された運搬地点へ移動する。

##### 7. デッドスポットをカバー

無線中継機が運搬されることで無線通信路が確立されデッドスポットがカバーされる。

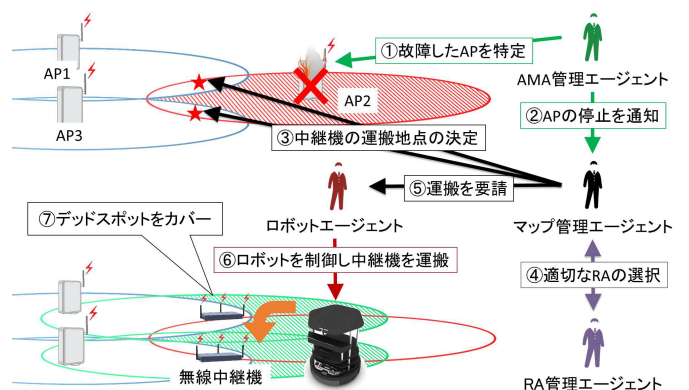


図 12 提案システムの流れ

## 5. 評価実験

アクセスポイント停止時にシステムが正しく動作することをテストする。また、所要時間を測定することで提案システムで迅速な無線通信路の確立が行えるかを検証する。

### 5.1 実験 1

名古屋工業大学の御器所キャンパスで提案システムを利用する場合を想定する。なお、UAV とアクセスポイントは図 13 のような配置である。図中の矢印の先端が UAV、アクセスポイントの図がアクセスポイント、星マークが静的に決定されている運搬地点 (CP : Carry Point) の位置である。複数の UAV が存在する場合を実験するために、シミュレータ上の UAV を一機とダミーロボットエージェントによる架空の UAV を二機利用する。UAV2 としてシミュレータ上の UAV を利用し、UAV1 と UAV3 はダミーロボットエージェントを利用する。UAV の情報を表 1 に示す。

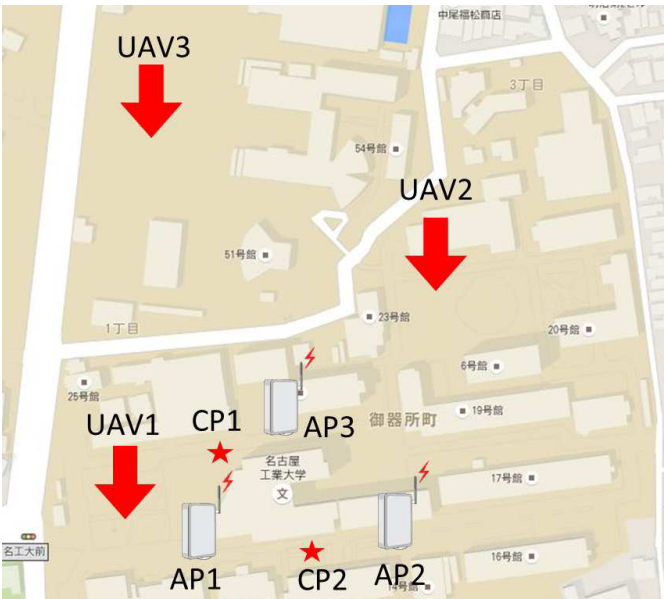


図 13 名古屋工業大学御器所キャンパス

表 1 UAV ロボットデータ

| Name | 座標 1      | 座標 2       | バッテリー |
|------|-----------|------------|-------|
| UAV1 | 35.156544 | 136.923977 | 10 %  |
| UAV2 | 35.157682 | 136.925838 | 100 % |
| UAV3 | 35.158612 | 136.923817 | 70 %  |

### 実験 1 の結果

測定結果を表 2 に示す。実験は 5 回行った。AP2 停止後に UAV2 が運搬地点 2 へ到達することが確認できた。全体稼働時間から、今回想定した状況では提案システムによ

り、一分以内に無線通信路を確立できることがわかった。稼働時間のうちメッセージ送信時間とロボット稼働時間は提案システムの性能の影響が少ない。表 2 に示すエージェントの稼働時間から、提案システムの動作にかかわる時間は 0.1 秒以内に抑えられていることがわかった。迅速な無線通信路確立といって差し支えない結果である。

表 2 屋外テストの所要時間

| 項目          | 平均時間 [ms] | 標本標準偏差 |
|-------------|-----------|--------|
| エージェントの動作時間 | 97.2      | 21.440 |
| メッセージの送信時間  | 56.4      | 9.555  |
| ロボットの稼働時間   | 45294.6   | 35.500 |
| 全体稼働時間      | 45448.2   | 53.013 |

### 5.2 実験 2

名古屋工業大学の二号館で提案システムを利用する場合を想定する。UGV2 とアクセスポイントは二階に配置され、図 14 のような配置である。図中の矢印の先端が UGV、アクセスポイントの図がアクセスポイント、星マークが静的に決定されている運搬地点 (CP : Carry Point) の位置である。また、UGV1 は一階、UGV3 は三階に配置する。UGV2 として Turtlebot2 を利用し、UGV1 と UGV3 はダミーロボットエージェントを利用する。UGV の情報を表 3 に示す。

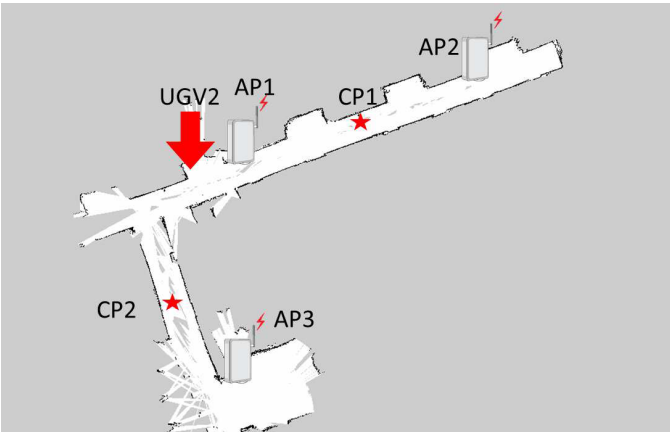


図 14 名古屋工業大学二号館二階打矢研究室前

表 3 UGV ロボットデータ

| Name | マップ名         |
|------|--------------|
| UGV1 | 名古屋工業大学二号館一階 |
| UGV2 | 名古屋工業大学二号館二階 |
| UGV3 | 名古屋工業大学二号館三階 |

## 実験 2 の結果

測定結果を表 4 に示す。実験は 5 回行った。AP2 停止後に UGV2 が運搬地点 1 へ到達することが確認できた。全体稼働時間から、今回想定した状況では提案システムにより、30 秒程度で無線通信路を確立できることがわかった。こちらも、迅速な無線通信路確立といって差し支えない結果である。

表 4 屋外テストの所要時間

| 項目          | 平均時間 [ms] | 標本標準偏差   |
|-------------|-----------|----------|
| エージェントの動作時間 | 138       | 36.571   |
| メッセージの送信時間  | 111.1     | 18.722   |
| ロボットの稼働時間   | 32077     | 1581.373 |
| 全体稼働時間      | 32326.1   | 1611.527 |

## 5.3 考察

今回の実験より、提案システムは運搬地点が一箇所しか現れないような、小規模な環境で迅速に動作できることがわかった。現在のシステムは、アクセスポイントの停止から新たな無線通信路確立のために無線中継器を運搬するまでの機能を実装することにとどまっている。提案システムの実用性を高めるためには、複数の運搬地点が同時または連続して現れるような大規模な環境で正しく動作できる必要がある。

## 6. おわりに

本研究の目的は、ロボットとマルチエージェントを組み合わせることで、災害時に人の手を介することなく、迅速に無線通信路を確立するシステムを構築することである。

提案システムでは、ロボットにより無線中継機を運搬することで、人が運搬することによる危険性を排除した。また、無線通信路確立までの手順をマルチエージェントにより全自動化することで迅速に無線通信路を確立できるようにした。

実験として、提案システムの動作の確認と所要時間の測定を行った。小規模な環境において提案システムが迅速に動作することを確認できた。今後は提案システムをより大規模な環境で動作させるため、以下の様な改良を行っていく。

**ロボットの状態管理の強化** RA 管理エージェントが管理する RA の状態に Busy を追加し、Busy 状態の RA を選ばない選択肢を与えることでタスクの分散を促進できるようにする。

**運搬命令のストック機能の追加** RA 管理エージェントに運搬命令ストック機能を追加し、適切なロボットエージェントが見つからなかった場合に、運搬命令を保持できるようにする。

**ロボットの連続した運搬への対応** ロボットエージェント

はタスク終了時に RA 管理エージェントに対し、通知を行い Busy 状態の解除を行えるようにする。また UAV のロボットエージェントはタスク終了時緯度経度情報も通知するようにする。

**ストックされたタスクの探索機能の導入** ロボットエージェントが自律的にストックされたタスクを発見・実行できるようにする。

## 参考文献

- [1] Takahiro Uchiya, Hideki Hara, Kenj Sugawara, Tet-suo Kinoshita, “Repository-Based Multiagent Framework for Developing Agent Systems”, (book chapter) in Y. Wang (Ed.), Transdisciplinary Advancements in Cognitive Mechanisms and Human Information Processing, IGI Global, Ch.4, pp.60-79, 2011.
- [2] Fabio Bellifemine, Giovanni Caire, and Dominic Greenwood, “Developing Multi-Agent Systems with JADE”, John Wiley & Sons, Ltd, 2007.
- [3] “DroneKit-python”, <http://python.dronekit.io/>
- [4] “3D Robotics”, <https://3dr.com/>
- [5] “SITL Simulator (Software in the Loop)”, <http://dev.ardupilot.com/wiki/sitl-simulator-software-in-the-loop/>
- [6] “MAVProxy”, <http://dev.ardupilot.com/wiki/mavproxy-developer-gcs/>
- [7] “Turtlebot2”, <http://www.turtlebot.com/>
- [8] “Robot Operating System”, <http://wiki.ros.org/>
- [9] “Microsoft’s Kinect”, <http://www.xbox.com/ja-JP/kinect>