

## 特別研究報告

題目

サイボーグインセクトによる迅速な被災者発見のための自己組織型  
移動制御手法の提案

指導教員

若宮 直紀 教授

報告者

北浦 直

2019 年 2 月 12 日

大阪大学 基礎工学部 情報科学科

平成 30 年度 特別研究報告

サイボーグインセクトによる迅速な被災者発見のための自己組織型移動制御手法の提案

北浦 直

内容梗概

主な用語

## 目次

<b>1</b>	<b>はじめに</b>	<b>3</b>
<b>2</b>	<b>サイボーグインセクト</b>	<b>4</b>
<b>3</b>	<b>サイボーグインセクトのモデル</b>	<b>5</b>
3.1	ランダムに方向を決定する . . . . .	5
3.2	進行方向ベクトルを変更するかの判定 . . . . .	5
3.3	回転角の決定 . . . . .	5
3.4	1step 分前進 . . . . .	6
3.5	面と面が構成する境界線に接したかの判定 . . . . .	6
3.6	次の進行方向ベクトルの決定 . . . . .	6
3.6.1	境界線に平行なベクトル . . . . .	7
3.6.2	境界線に平行なベクトルの逆ベクトル . . . . .	7
3.6.3	現在の方向を保ちつつ, 新しく接した面に平行なベクトル . . . . .	7
3.6.4	ベクトルの決定 . . . . .	8
<b>4</b>	<b>フロッキング</b>	<b>9</b>
<b>5</b>	<b>制御モデル</b>	<b>10</b>
5.1	パラメータ設定 . . . . .	10
<b>6</b>	<b>実験結果</b>	<b>11</b>
<b>7</b>	<b>おわりに</b>	<b>12</b>
	<b>参考文献</b>	<b>13</b>

## 1 はじめに

災害が起きた場合、倒壊した建物内に被災者が取り残されることが起こりうる。倒壊した建物内に捕らわれた被災者の探索において、現在は災害救助犬やスコープカメラなどの人間の能力を補助するような手法が広く使われている。しかし、倒壊した建物内は人間や災害救助犬が入れないような環境であることが多い。また、救助活動の中で2次災害が起きてしまう事例も少なくない。そこで、現在の探索手法では探索できないような狭い空間を探索可能で、探索中の2次災害の危険性を低くすることができるサイボーグインセクトを用いた被災者探索の研究がなされている。

サイボーグインセクトを被災者探索に活用するための研究として CINEMa(Cyborg Insect Networks for Exploration and Mapping) があげられる。この研究の中では、サイボーグインセクトへ制御を与えて任意の方向へ向かわせたり、サイボーグインセクトが位置推定をするためのアルゴリズムの実験がされていたりする。しかし、CINEMa はがれきなどが散乱するような悪条件下における被災者探索のための構成要素の確立は行われているが、それらを用いて被災者探索を効率的に行うためのアルゴリズムなどは提案されていない。つまり、群で探索しているにも拘らず複数個体がほぼ同一の場所を探索することで空間全体を探索するのにかかる時間が増加してしまう恐れがある。

また、自律的なレスキューロボットが行う自律的なアルゴリズムの1つであるフロッキングという手法がある。フロッキングとは、周囲のロボットに対して、Separation, Alignment, Chesionの3つの動作をとることで複数個体が群れとなって動くようなアルゴリズムである。このフロッキングを被災者探索に適用される研究が行われているが、フロッキングをサイボーグインセクトに適用して被災者探索を実現している研究はなされていない。また、これらの研究でなされているフロッキングは基本的に連続で操作を行っており、断続的に制御を行っている研究は少ない。しかし、サイボーグインセクトに与えるフロッキングの制御は生物に電気信号を流すためできるだけ頻度が低いことが望ましい。

そこで、本研究では CINEMa で提案されているようなサイボーグインセクトの群れに対するフロッキング制御手法の提案を行う。ここでは、サイボーグインセクトに対する断続的なフロッキング制御でも探索にかかる時間の短縮ができることを確認する。

以降、第章では、

## 2 サイボーグインセクト

この章では、本研究で考えるサイボーグインセクトの概要と、サイボーグインセクトに取り付けられる機材、サイボーグインセクトに関する条件の説明を行う。

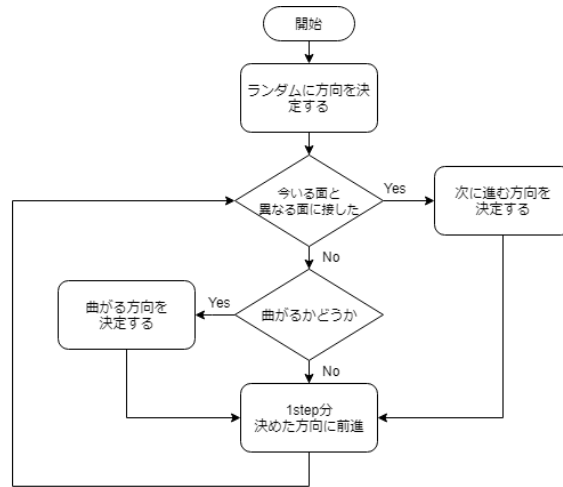


図 1: 制御されていないサイボーグインセクトのアルゴリズムのフローチャート

### 3 サイボーグインセクトのモデル

この章では，サイボーグインセクトが制御を受けていない場合の動きのモデルについて説明を行う．

制御をされていないサイボーグインセクトは，図 1 のフローチャートに示されているアルゴリズムに従って行動する．このアルゴリズムでの 1step は 0.1 秒とする．また，サイボーグインセクトの速度は 60cm/s とする [?].

#### 3.1 ランダムに方向を決定する

探索が開始されたときに実行される．どの方向に進むのかをランダムに決定する．この時に決定された向きの単位ベクトルを進行方向ベクトルと呼ぶ．

#### 3.2 進行方向ベクトルを変更するかの判定

サイボーグインセクトは，1step 毎に現在の進行方向ベクトルを別の進行方向ベクトルに変更するかの判定を行う．この時に，進行方向ベクトルを変更する確率は 1%とする．

#### 3.3 回転角の決定

現在の進行方向ベクトルからどれだけ回転するかの決定を行う．この時，現在の進行方向ベクトルから角度が大きく異なる進行方向ベクトルほど選択する確率が低くなるように設定した．また，現在接している面から別の面へ移動するようなベクトルには回転しないものと

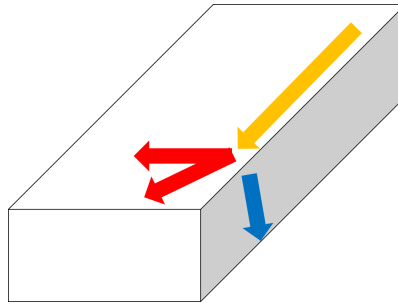


図 2: 回転できるベクトル (赤) と回転できないベクトル (青) の例

する。図 2 のように赤い矢印であらわされるベクトルには回転する可能性が存在し、青のベクトルのように接する面が変わるベクトルには回転しないものとする。

ここで、回転する角度は、平均 0, 分散  $\frac{\pi}{6}$  の正規分布に従って決定される。

### 3.4 1step 分前進

サイボーグインセクトは、進行方向のベクトルに 1step 分前進する。また、途中で障害物があったり接している面が途切れたりなどで 1step 分の距離を進めない場合は、進めるところまで進み、その位置で停止する。

### 3.5 面と面が構成する境界線に接したかの判定

サイボーグインセクトは、現在いる場所が面と面が構成する境界線に接しているかどうかを 1step 毎に判定を行う。

### 3.6 次の進行方向ベクトルの決定

サイボーグインセクトは次の進行方向ベクトルを計算する。次に進行方向ベクトルの候補として、図 3a のような 3 種類のベクトルを計算する。この時、サイボーグインセクトがすでに境界線に沿って進んでいる場合は、図 3b のように、新しく接した境界線に沿うような 2 つのベクトルを計算する。

1. 境界線に平行なベクトル
2. 境界線に平行な 1. の逆ベクトル
3. 現在の方向を保ちつつ、新しく接した面に平行なベクトル

それぞれの具体的な計算方法は以下の章で説明する。

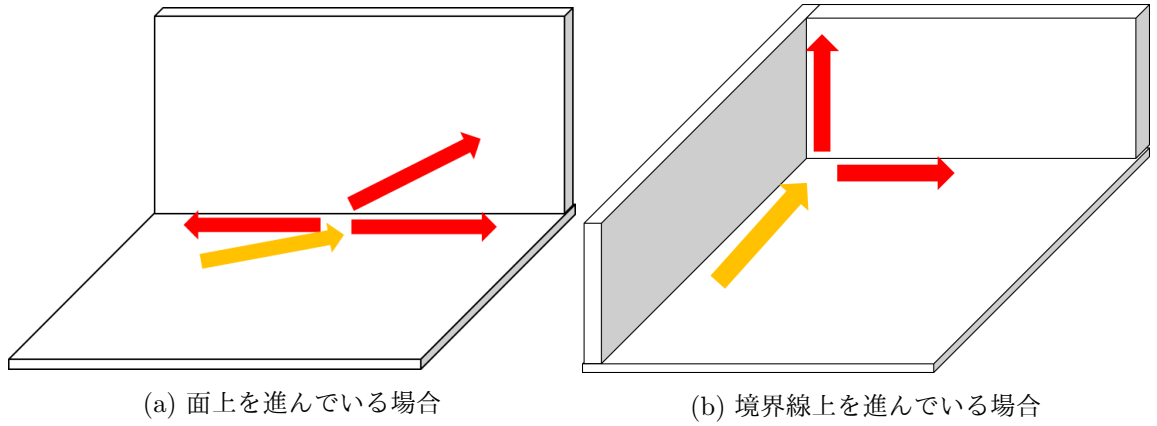


図 3: 次に進む方向のベクトル

### 3.6.1 境界線に平行なベクトル

新たに接した境界線からベクトルを求めて、そのベクトルに平行な単位ベクトルを次の進行方向ベクトルの候補とする。3b のように境界線上を進んでいる場合、新たに接した境界線が 2 つあるため、その 2 つを次の進行方向ベクトルの候補とする。

### 3.6.2 境界線に平行なベクトルの逆ベクトル

3.6.1 章で求めたベクトルの逆ベクトルを次の進行方向ベクトルの候補とする。

### 3.6.3 現在の方向を保ちつつ、新しく接した面に平行なベクトル

このベクトルは、今進んでいる方向ベクトルを接した境界線を軸にして回転することで得られる。今進んでいる方向ベクトルの境界線に垂直な成分と、新しく触れた面のベクトルがなす角を  $\phi$  としたときに、 $\phi$  だけ回転することで、次に進む方向ベクトル  $\vec{x}$  が得られる。 $\phi$  の求め方は図 4 のようになる。

境界線に平行なベクトルを  $\vec{b}$  とした際に、以下のようにあらわされる。

$$\vec{x} = R_{\vec{b}}(\phi)\vec{v} \quad (1)$$

また、回転行列  $R_{\vec{b}}(\phi)$  は以下のようになる。

$$R_{\vec{b}}(\phi) = \begin{pmatrix} b_x^2(1 - \cos \phi) + \cos \phi & b_x b_y(1 - \cos \phi) - b_z \sin \phi & b_z b_x(1 - \cos \phi) + b_y \sin \phi \\ b_x b_y(1 - \cos \phi) + b_z \sin \phi & b_y^2(1 - \cos \phi) + \cos \phi & b_y b_z(1 - \cos \phi) - b_x \sin \phi \\ b_z b_x(1 - \cos \phi) - b_y \sin \phi & b_y b_z(1 - \cos \phi) + b_x \sin \phi & b_z^2(1 - \cos \phi) + \cos \phi \end{pmatrix} \quad (2)$$



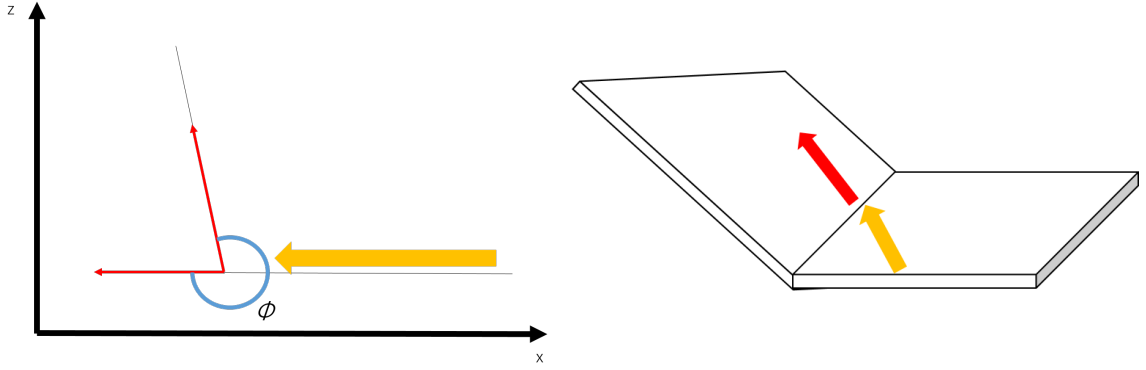


図 4: なす角  $\phi$  の位置

#### 3.6.4 ベクトルの決定

3.6.1 章から 3.6.3 章で計算したベクトル  $\vec{x}_i$  から、次に進む方向ベクトルにするものを決定する。次に進む方向ベクトルは、基準ベクトルとの差異による確率に従って決定される。基準ベクトルは、現在の方向ベクトルと重力ベクトルとの和によって計算される。まず、現在の方向ベクトル  $\vec{v}$  と重力ベクトル  $\vec{g}$  から基準となるベクトル  $\vec{a}$  を計算する式は以下の通りである。

$$\vec{a} = \vec{v} + \frac{49}{60}\vec{g} \quad (3)$$

ここで、 $\vec{g}$  の係数は、0.1 秒分の自由落下の計算から求めている。

次に、ベクトルを選ぶ計算は以下の式から求められる。

$$P(\vec{x}_i) = \frac{1}{n-1} \left( 1 - \frac{|\vec{a} - \vec{x}_i|}{\sum_{k=1}^n |\vec{a} - \vec{x}_k|} \right) \quad (4)$$

ここで、 $\vec{x}_i$  は 3.6.1 章から 3.6.3 章で計算したベクトルで、 $i = (1, \dots, n)$  である。

## 4 フロッピング

## 5 制御モデル

### 5.1 パラメータ設定

## 6 実験結果

## 7 おわりに

## 参考文献

[1]