

マルチエージェントにおける位置に応じた タスク管理方法に関する一考察 Task Managements Depending on Positions of Multi Robots

内海 孝亮, 植村 渉

Kosuke Utsumi and Wataru Uemura

龍谷大学大学院理工学研究科電子情報学専攻
Ryukoku University

utsumi@vega.elec.ryukoku.ac.jp, wataru@rins.ryukoku.ac.jp

概要

本研究では移動ロボットを用いた競技大会の一つである RoboCup Logistics League におけるロボットのタスク管理方法について議論する。タスクを管理するロボットが非常停止や調整によって試合から退場すると、管理が放棄される。それを抑えるために、ロボットが退場しやすい位置などの条件を検討し、管理破棄が生じにくい方法を提案する。

1 はじめに

自律移動ロボットによるサッカーや災害現場での人命救助などを目的とした競技大会として RoboCup がある。RoboCup は毎年世界各地で開催され、2017 年は日本の名古屋で開催される。本研究ではその中の工場のオートメーション化を背景とする RoboCup Logistics League (以下 RCLL と表記) におけるロボットのタスク管理方法について議論する。RCLL では指定されたオーダーに従い複数のロボットが製品を生産・加工し、完成した製品を目的地へ搬送する [Utsumi 2016]。その中でロボットはタスクを遂行するために競技用のマシンを使用していく。このとき他のロボットとのタスクの重複やリソースの使用権の衝突を避けるためにタスクとリソース管理を行う必要がある。一般に、これらの管理はリーダー選挙などを通して選ばれたロボットが行うことが多いが、ロボットの非常停止や調整などで試合から退場することによって管理が放棄される可能性がある。本研究では非常停止や調整などが起きにくいロボットが管理者となることで、管理破棄が生じにくい方法を提案する。そのためにはロボットが退場しやすい位置などの条件を試合のログから解析する。

2 RoboCup Logistics League

RCLL では各チームは FESTO 社製の移動ロボットと、運営側の競技用の Module Production System (以下 MPS と表記) を用いて試合を行う。MPS は全部で 4 種類あり、製品の基礎部分の供給を行う Base Station (BS), 中間工程の加工を行う Ring Station (RS),

完成のための加工を行う Cap Station (CS), 完成した製品の格納を行う Delivery Station (DS) に分かれる。MPS は 1 チームに BS と DS を 1 台ずつ, RS と CS を 2 台ずつで全 6 台与えられ, フィールド上には両チームの MPS が全部で 12 台存在する。そして MPS はフィールド上に左右対称に配置される (Figure 1) [Uemura 2015]。Figure 1 の青線は壁を表しており, フィールド全てを囲んでいるわけではなく, 一部壁のない部分もある。さらにフィールドは 24 個の $1.5 \times 2\text{m}$ のゾーンで構成されている。また試合の基本的な情報は審判用のプログラムである RefBox によって管理される。

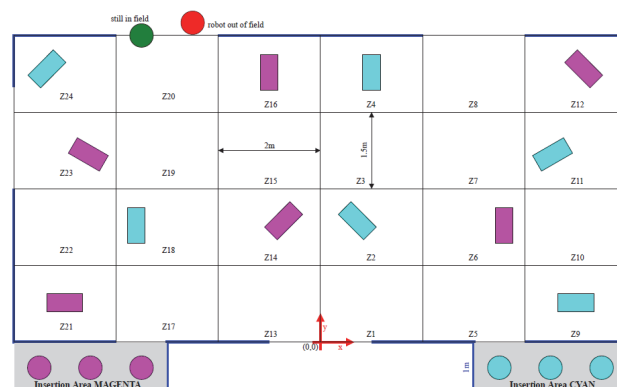


Figure 1 RCLL のフィールド例

2.1 競技の目的

RCLL は, 自律移動ロボットによって製品を生産して, 完成品を目的地まで搬送することを目的とした競技である。3 台のロボットを用いて製品の加工を MPS で繰り返し行い, 完成した製品を格納用の MPS へ搬送する。

競技はセットアップ, 探索, 生産の 3 つのフェーズで構成される。まずはセットアップフェーズから始まり, 運営側が MPS のセットアップを行う。次に探索フェーズへと移り, MPS の種類を示すマーカや状態を示すシグナルパターンをロボットが探索し, RefBox に報告する。最後に生産フェーズがあり, 製品を加工していき, 完成した製品を搬送する。

2.2 使用ロボット

RCLL ではドイツの FESTO 社が扱う移動型ロボットである Robotino 3 を使用して競技を行う (Figure 2). Robotino 3 は組込み型の小型計算機を搭載しており、OS として Ubuntu が動いている。そして Ubuntu 上の API を経由して Robotino 3 を制御する。Robotino 3 は USB や複数の I/O ポートを持ち、カメラや外部センサを取り付けることができる。その結果、周囲の環境が把握できて複雑な動作が可能となる。また付属品の金属製支柱やテーブルを取り付けることで製品を掴むためのグリッパが取り付けられるようになる。

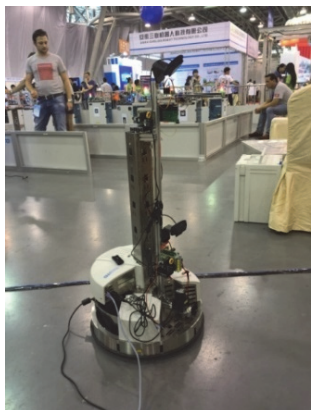


Figure 2 競技で使用する Robotino 3

2.3 得点方法

RCLL には探索フェーズと生産フェーズがあり、それぞれで得点の扱いが異なる。探索フェーズでは MPS に貼り付けられたマーカを認識することによって MPS の種類を認識し、シグナルパターンはカメラなどを用いて認識する。認識結果を RefBox に正しく報告できると点が得られる。誤った報告をした場合は減点されるので注意が必要である。生産フェーズでは、試合中にどのような製品を作るべきかというオーダーが RefBox によってリアルタイムでアナウンスされる。そのオーダーに従い製品を一定の状態まで加工すると得点する。完成した製品を正しく搬送できるとさらに点が得られる。

3 ロボットの行動エンジンについて

本章では RoboCup2016 の Logistics League の優勝チームである Carologistics のタスク管理方法及び生産フェーズでロボットが製品を生産・加工していく上でどのような流れでタスク管理を行っているかを紹介する。

3.1 階層型行動エンジン

Carologistics は随時変化する RCLL 環境に素早く対応できるようにロボットの行動要素を 3 つの階層に分けている (Figure 3) [Tim 2016]。最下層では、認識やアクチュエータの処理、中間層では行動の実行、そして最上位層ではロボットの行動意思決定や

プランニングなどを扱っている。この最上位層のプランニングを行っていく上で、ロボットはタスク管理をする必要がある。

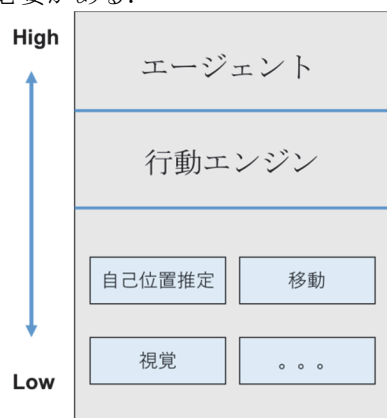


Figure 3 ロボットの行動構造

3.2 Carologistics のタスク管理方法

RCLL の生産フェーズではロボットは RefBox によってアナウンスされたオーダーに従い製品を MPS で加工して生産していく。このとき、RCLL では最大 3 台の自チームのロボットが同時に同環境下で動くため、タスクの重複やリソースの使用権の衝突を引き起こす可能性がある。そのような状況为了避免するために、ロボットはタスクやリソースの使用権を管理する必要がある。

Carologistics ではタスク管理方法としてマスター/スレーブ方式を採用している。これはリーダー選挙を通して選ばれた一台のロボットがマスターとなり、そのマスターがタスク管理を行う方法である。そして他の二台のロボット、つまりスレーブはタスクやリソースのロックをマスターに要求する。マスターはそれらの要求を受け取るとロックが可能かどうかを確認する。可能であった場合、要求は承認されてタスクまたはリソースはロックされる。またロックしたタスクが終了すると、スレーブはタスクとリソースのロック解放をマスターに要求し、マスターは解放を実行する。

4 マスターの選択方法

RCLL ではロボットは生産フェーズで製品を生産していく上でタスクやリソースの管理を行う必要がある。それらの管理はリーダー選挙などを通して選ばれた一台のロボットによって行われるマスター/スレーブ方式を取ることが多い。しかし、マスターが非常停止や調整などが原因で試合から急遽退場した場合、今まで行っていたタスク管理が放棄されることになる。このときロボット間でタスクの重複やリソースの使用権で衝突が起こる可能性がある。

RCLL においてロボットが退場する状況として、1) 敵ロボットや MPS、フィールドの壁などとの衝突によって故障した場合と、2) 移動中や MPS などに接近しているときにデッドロックに陥った場合などが考えられる。そのような状況に陥りやすい位置とし

て、まずは MPS 周辺がある。これは MPS への接近時には複雑な処理が必要なためである。次に敵チームの主要サイドが挙げられる。RCLL ではフィールド内に主要サイドが存在し、これはフィールド y 軸で分割し、自チームのロボット挿入エリアがあるサイドが主要サイドとなる。つまりもう一方のサイドは敵チームの主要サイドということである。Figure 1 ではシアンチームならば y 軸から右側が主要サイドになる。RCLL では自チームの MPS は主要サイドに 4 台、敵チームの主要サイドに 2 台というように配置される。このような MPS の配置方法から、ロボットは自チームの主要サイドでの活動が多くなると考えられる。さらにフィールドの壁際でもロボットは動きが制限されるので、退場の可能性は高くなると考えられる。

マスターのロボットを選択する際に、これらの問題が起きにくいロボットを優先して選ぶ方法を提案する。これらの問題がフィールド上で発生する可能性を計算し、それらを合成したフィールド情報を元に、各ロボットの退場の可能性を算出する。MPS へ接近しているときに退場する可能性をポテンシャル関数で扱う [D.H.Kim 2006]。6 個の MPS の内の j 番目の MPS の中心座標を (x_j, y_j) として以下にポテンシャル関数 U_m を示す。

$$U_m = \sum_{j \in N_m} c_m \exp \left\{ -\frac{(x - x_j)^2 + (y - y_j)^2}{l_m^2} \right\} \quad \dots \text{式(1)}$$

ここで c_m , l_m はポテンシャル関数のパラメータであり、本研究ではそれぞれともに 1 として扱ったため、

$$U_m = \sum_{j \in N_m} \exp \left\{ -(x - x_j)^2 - (y - y_j)^2 \right\} \quad \dots \text{式(2)}$$

となる。

式(2)による MPS ポテンシャル場を Figure 4 に示す。

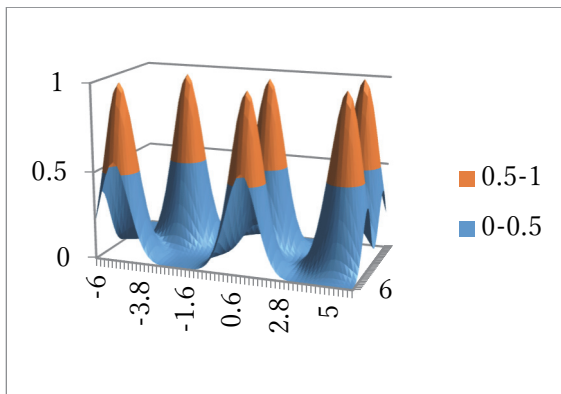


Figure 4 RCLL フィールドの MPS のポテンシャル場

さらに敵主要サイドと壁際のポテンシャル値を 1 として MPS のポテンシャル場と足し合わせたものを Figure 5 に示す。

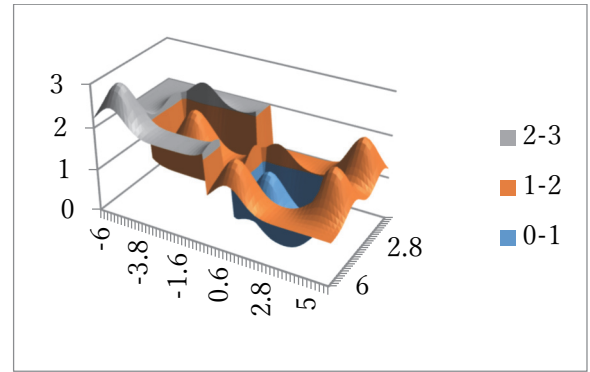


Figure 5 敵主要サイド、壁際のポテンシャル、そして MPS のポテンシャルを考慮したポテンシャル場

Figure 5 のポテンシャル場を基に、ポテンシャルが低い位置にいるロボットにマスター権を与えることでタスク管理破棄を回避する。

5 おわりに

本研究では RCLL を背景として、マルチエージェントのタスク管理方法について議論した。今後は提案手法を基に実際にロボットにマスター権を与えて実験していく必要がある。

謝辞

本研究は JSPS 科研費 15K16313 の助成を受けたものです。

参考文献

- [Utsumi 2016] 内海孝亮, RoboCup2016 に参加して, 理工ジャーナル, (印刷中)
- [Uemura 2015] Wataru Uemura, et. Al., RoboCup Logistics League Rules and Regulations 2016, <http://www.robocup-logistics.org/rules>
- [Tim 2016] Tim Niemueller, et. Al., The Carologistics RoboCup Logistics League Team 2016, http://www.robocup2016.org/media/symposium/Team-Description-Papers/Logistics/RoboCup_2016_Logistics_TDP_carologistics.pdf
- [Fawkes] https://trac.fawkesrobotics.org/wiki/Carologistics/Agent_Architecture
- [D.H.Kim 2006] D.H.Kim and S.Shin, Local path planning using a new artificial potential function configuration and its analytical design guidelines, Advanced Robotics 20, pp. 115 - 135, (2006).
- [Shimakura 2009] 島倉諭, 項警宇, 稲垣信吉, 鈴木達也, ポテンシャルを組み込んだパーティクルコントロールによる自律移動ロボットの障害物回避制御, ロボティクス・メカトロニクス講演会講演概要集 2009, 2A1-G05(1-4).