

AI チャレンジ研究会(第47回)

Proceedings of the 47th Meeting of Special Interest Group on AI Challenges

CONTENTS

- RoboCup サッカーにおける枝刈りを用いた行動制御…………… 1
田中 翔†, 中島 智晴†, 秋山 英久‡
†大阪府立大学, ‡福岡大学
- RoboCup Logistics Leagueにおける一方通行経路設定に対する格子状経路の追加に関する一考察 …… 7
辻 和輝, 津田 諒太, 木綱 智, 内海 孝亮, 植村 涉
龍谷大学
- 可視光通信を用いたロボット制御のための信号強度取得に関する一考察…………… 10
清水 謙汰, 端山 稜人, 福森 康洋, 植村 涉
龍谷大学
- 可視光通信の利得制御方法に関する一考察…………… 14
端山 稜人, 植村 涉
龍谷大学
- ROS を使って移動ロボットを実現するためのマイコン用通信ライブラリの実装と移動ロボットの実現例
…………… 18
光永 法明, 杉本 梓
大阪教育大学

日時:2017年3月30日

場所: 玉川大学 (東京都町田市) 大学教育棟 5F 503室

Tamagawa University, March 30th, 2017

一般社団法人 人工知能学会

Japanese Society for Artificial Intelligence

RoboCup サッカーにおける枝刈りを用いた行動制御

Controlling Actions with Pruning in RoboCup Soccer

田中 翔[†], 中島 智晴[†], 秋山 英久[‡]

Sho Tanaka[†], Tomoharu Nakashima[†], Hidehisa Akiyama[‡]

大阪府立大学[†], 福岡大学[‡]

Osaka prefecture University[†], Fukuoka University[‡]

{sho.tanaka@cs, tomoharu.nakashima@kis}.osakafu-u.ac.jp, akym@fukuoka-u.ac.jp

Abstract

This paper proposes a team strategy implementation method by directly controlling its behavior. The team behavior depends mainly on the decisions of the ball owner who selects the best action in an action space. Unfortunately, the action space is generally too large to explore in a reasonably short time. Furthermore, many solutions might be irrelevant to the strategy decided by the developer and can be ignored. In order to reduce the search space and make the ball owner focus only on relevant actions, a binary classifier is used. Actions classified as negative by the classifier are pruned from the search space. Computational experiments are conducted in order to show that the reduced search space produces tendencies in the team behavior that correspond to the strategy of interest.

1 はじめに

ロボット工学と人工知能の領域横断型研究プロジェクトとして RoboCup [1] が知られている。RoboCup には様々なリーグが存在しており、それぞれにおいて活発な研究、開発が行われている。本論文では、RoboCup のリーグの一つである RoboCup サッカーシミュレーション 2D リーグを研究対象とする。RoboCup サッカーでは、競技大会で勝利することを目的としている。ランダムな思いつきで作られた戦略を使って勝利するよりも、緻密なデータ分析や機械学習によるモデル化を活用して勝利に繋がる戦術を生成することが RoboCup の理念と合致する。

RoboCup サッカーシミュレーション 2D リーグでは、毎年各チームが変化を加えることにより、フィールドの片側

に選手を固め攻撃する戦術や守備を偏重する戦術など、多種多様な戦術が存在している。戦術の多様性と相性のため、1つの戦術で全てのチームに勝つことは困難である。そのため、複数の戦術を用意し、相手に合わせて戦術を変更することで試合を有利に進めることが求められる。複数の戦術を用意するには、どのような方策で行動するかを決定し、それに合わせてプレイヤーの行動を制御する必要がある。本研究では、プレイヤーの行動制御に着目する。

プレイヤーの行動制御に関して、探索木を用いたプレイヤーの意思決定法が提案されている [2]。探索木で用いられる評価関数では、敵ゴールからの距離や敵との距離といったサッカー特有の特徴量の重み付き総和により、状態を評価する。そのため、評価関数における特徴量の重みを変更することでプレイヤーの行動を制御出来る。しかし、評価関数の重みは人手により調整されているため、戦略や戦術を表現する評価関数を新たに試作するだけでも、高い時間的コストが必要である。

プレイヤーの行動を制御する他の方法として、探索木生成段階で不要な行動列を直接制限することが考えられる。本論文では、この方針により探索木生成のプロセスを変更することで、評価関数を調整することなく、設計者の意図通りに戦術を実装する方法を提案する。

提案手法の概要を説明する。探索木生成段階において、行動が設計者の意図を反映しているかどうかを判断し、枝刈りを行う。枝刈り判定のため、サポート・ベクタ・マシン (SVM) により識別モデルを構築する。識別モデルの構築は教師あり学習に基づくため、モデルの構築には学習用データが必要である。学習用データを作成するため、まず試合を行い試合ログを作成する。作成した試合ログからプレイヤーの行った行動列を抽出しクラスタリングする。作成したクラスタを GUI アプリケーションにより人手でラベル付けすることで、学習用データを作成する。

数値実験では、10-分割交差検証により SVM の識別精度を調査する。また、学習した枝刈りによる行動制御への

有効性を示すため、実際に枝刈りを組み込んだチームで試合を行いキックの軌跡と性能を調査する。

2 RoboCup

RoboCup は、ロボット工学と人工知能の発展が目的の自律移動型ロボットによるサッカーやレスキュー、家庭内作業などを題材とした研究プロジェクトである。RoboCup には「西暦 2050 年までに、サッカーの世界チャンピオンチームに勝てる自律型ロボットチームを作る」という目標があり、この目標に向けて盛んに研究が行われている。

RoboCup にはサッカー以外にも、大規模災害への対応のシミュレーションや災害現場で活躍するロボットの開発を促進するレスキューリーグ、日常生活で人間を支援する自律ロボットによる競技を通じて、人とコミュニケーションしながら役に立つロボットの実現を目指す@ホームリーグの他に、次世代のロボット技術者育成を目的としているジュニアリーグも存在する。本論文では、RoboCup サッカーシミュレーションリーグを研究の対象とする。シミュレーションリーグは RoboCup 創設当初から存在する最も古いリーグの 1 つである。サッカーシミュレーションでは、実機を使用せずに、コンピュータ内に用意された仮想フィールド上でサッカー競技を行う。サッカーシミュレーションには 2D リーグと 3D リーグがある。Figure 1, 2 に 2D リーグと 3D リーグの試合の様子を示す。2D リーグでは、基本的な動作（キックやドリブルなど）はコマンドとして実装されている。そのため 2D リーグでは高レベルな意思決定を主な研究対象としている。一方、3D リーグでは、エージェントはヒューマノイドロボットで形成されているため、基本的な動作を関節から制御する必要があり、基本的な動作が非常に重要である。本論文では 2D リーグを扱う。2D リーグでは、二次元平面を仮想サッカーフィールドとし、円形のエージェントをプレイヤーとして競技を行う。また、プレイヤーやボールの位置と速度は全て二次元のベクトルとして表される。試合は前後半 3000 サイクルずつ合計 6000 サイクルからなる。1 サイクルは 0.1 秒で離散化されている。各プレイヤーはそれぞれ独立したエージェントとしてプログラムされており、制限された視覚情報や聴覚情報からドリブルやパス等の行動選択を行う。

3 関連研究

RoboCup サッカーシミュレーション 2D リーグにおいて、プレイヤーの行動制御に関する様々な研究が行われている。例えばプレイヤーレベルでの制御として、Gabel ら [3] は、強化学習を用いた Neuro Hassel により、ディフェンス時における積極的なボールのインターセプト動作獲得を行っている。しかし、各プレイヤーでの行動が洗練される

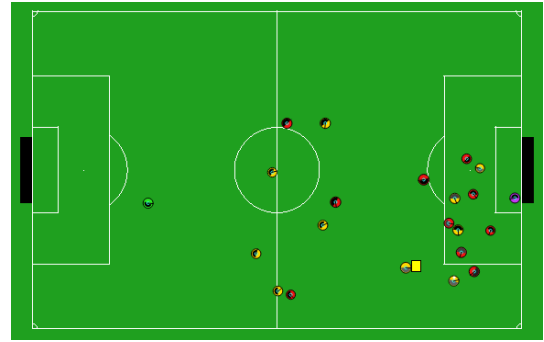


図 1: 2D Simulation League



図 2: 3D Simulation League

ことにより、チーム力は向上するが戦術レベルの行動制御に至ることは難しい。

チームレベルでの行動制御に関する研究として、Stone ら [4] は、特定の条件下において実行すべき戦術を事前に用意し共有することで、戦術レベルの行動をプレイヤーに知らせる Locker Room Agreement を提案している。Locker Room Agreement により、限定的な状況で戦術レベルの行動制御が可能であるが、戦術が固定的であるため、柔軟性に乏しいという問題がある。一方で、状況に応じて柔軟に対応し戦術レベルでプレイヤーの行動制御を行う方法も提案されている。Akiyama ら [5] は、Delaunay Triangulation を用いたボール位置に基づく直感的にわかりやすいプレイヤーのポジショニング手法を提案している。Akiyama ら [2] は、行動連鎖と呼ばれる探索木を生成し探索と評価により行動決定を行う木探索によるオンラインでの行動プランニングを提案している。また、Akiyama ら [6] は、上記 2 つの手法が実装された agent2d をオープンソースとして公開している。agent2d を基本としてプレイヤー開発を行っているチームは多く、行動制御に関する研究にもよく使われている。特に、チームの戦術を表現するよう行動制御を行うため、行動プランニング時に用いられる評価関数に関する研究が多く行われている。

評価関数に関する研究として、例えば三船ら [7] は、Single Input Rule Modules ファジィシステムを用いて評価

関数を自動作成する手法を提案している。田川ら [8] は、強化学習の 1 つである方策勾配法と人間の投票システムにより評価関数で用いる重みを学習する手法を提案している。牛腸ら [9] は、ランク SVM を用いたランキング学習と人間の教示による評価関数の作成手法を提案している。しかし、自動作成された評価関数が競技会に用いられることは少ない。このことから、人手により作成した評価関数のほうが優れており、自動作成された評価関数は未だ実践的なレベルではないと考えられる。そのため、複数の戦術を評価関数の自動作成により用意することは意思決定の質を落とす要因になる。そこで、評価関数を変更することなく、複数の戦術を用意する必要がある。

4 行動プランニングと枝刈りによる行動制御

本論文で使用するチームは、探索木により行動列を生成する。プレイヤーの行動を制御するため、行動プランニングにおける枝刈りを提案する。本章では、まず行動プランニングについて説明し、次に枝刈りによる行動制御について説明する。

4.1 行動プランニング

本論文では、行動意思決定のモデルとして秋山ら [2] の提案した木探索による協調行動プランニングを用いる。このモデルでは、ボールキック時において探索木を生成し探索することにより、行動プランを作成している。行動プランを次にプレイヤーが行うべき一定数の長さを持つ行動列と定義する。

本論文で使用するチームでは、探索木の走査アルゴリズムとして、最良優先探索を用いる。以下の手順により行動プランを作成する。まず、ルートノードに現在状態を格納する。そして、ルートノードから行動の候補を生成する。このとき、プレイヤーが観測した現在状態や予測状態を入力とし、自分と他者を含めた複数のエージェントによって実行される行動（パスやドリブル、シュートなど）を生成する。この時、実現可能な行動が計算し、不可能だと判断した行動は削除されるため、現実性のある行動のみ生成される。生成された行動を評価関数により評価値を計算し、行動と状態、評価値を探索木へ子ノードとして格納する。すべてのノードが追加された後、評価値が最大のノードを選択し、そのノードにおける予測状態からさらに行動の候補を生成する。これを繰り返すことで、探索ノード数があらかじめ設定された最大値に達するまで探索木を成長させる。ただし、木の深さがあらかじめ設定した値を越える場合や、ノードの予測状態から行動が生成できない場合、そして行動列の終了条件に設定されている行動（シュート）が生成された場合は、その葉ノードでの子ノード生成は行わないものとする。構築された木構造の中からノード列をつなげると、行動列が得られる。

探索終了後、生成された行動列の中で評価値が最大のものを行動プランとする。行動プランニングにより、プレイヤーは数手先の状況を考慮し、より戦術的価値が高い行動を選択することが可能となる。

4.2 枝刈りによる行動制御

行動プランニングにおいて、行動の意思決定は評価関数に強く依存する。新たに戦略や戦術を反映した行動選択のためには、それに対応した評価関数を作成する必要がある。評価関数は、ゴールからの距離、ペナルティエリアへの侵入、サイド攻撃といったサッカー特有の特徴量の重み付き総和により評価値を算出している。しかし、この重みは人手による調整の繰り返しによって定められているため、有用な評価関数を作成には高い時間的コストや開発の経験が必要であり困難である。評価関数を修正せずに行動制御を行う他の方法として、探索木生成段階で行動を制限することが考えられる。そこで、木探索時に枝刈りを行い、探索の選択と集中を行う手法を考える。本論文では、SVM により枝刈りの判断を行う。本節では、まず位置座標の離散化について説明し、SVM を用いた枝刈りについて説明する。

4.2.1 位置情報の離散化

本論文では、戦術を行動列によって表現している。そこで、行動の遷移を明確にするため、 x と y の 2 変数の連続値により表現される位置座標を離散化する。離散化のため、フィールドを $n \times m$ 個のグリッドに分割する。グリッド状に分割したフィールドにおいて、位置座標に対応したグリッドのみを 1、それ以外では 0 の値をとる $n \times m$ 次元ベクトルにより位置座標を表現する。つまり、仮に k 番目のマス目内でキックした際、その位置座標を離散化して表現する場合、 $n \times m$ 次元ベクトルにおいて k 番目の要素のみ 1 の値をとり、残りの要素は 0 の値をとる、 $n \times m$ 次元ベクトルとなる。

4.2.2 SVM を用いた枝刈り

探索木生成段階において、設計者の意図と反する行動を枝刈りする。SVM をラベル付けした行動列により学習させる。ラベル付けの方法は 5 章で説明する。枝刈りの結果、設計者の意図した戦術を表現する行動列だけが残り、SVM への入力には、離散化したキック座標による行動列を用いる。行動列の長さは一定ではないため、可能なすべての長さに対する SVM を構築し、枝刈りを判断する。

5 GUI を用いた学習用データへのラベル付け

SVM は教師あり学習であり、モデルを構築するための学習用データが必要である。本研究では、学習用パターンを試合ログから生成する。まず、多数の試合を行い試合ログを作成する。次に、作成した試合ログからプレイヤー

が実行した行動列を抽出し、その行動列から長さ4の連続した部分行動列を生成する。これは、行動プランニングで生成される探索木の深さが最大4であるためである。そして、抽出した行動列にラベル付けを行い学習用データとする。学習用データへのラベル付けにおいて、製作者の意図した戦術を表現する行動列かどうかを完全に自動で判断することは難しい。そのため、既存研究において人間の教示による学習用データの作成が行われてきた。例えば、田川らの研究では人間による投票システムを用いた行動に対する報酬を決定するシステム [8] を作成している。また、牛腸らは GUI アプリケーションを用いて行動にランク付けを行うシステム [9] を作成している。そこで、本研究においても、GUI アプリケーションを用いることにより、人間による直観的なラベル付けを行う。しかし、いずれの研究においても人間による選択に必要な時間的コストが問題となっている。そこで、本研究ではクラスタリングにより似た行動列をまとめることにより、選択に必要なコストを削減する。クラスタリングには、ガウス混合分布と EM アルゴリズムを用いる。また、最適クラス数数を決定する基準として、ベイズ情報量規準を用いる。作成した GUI アプリケーションを Figure 3 に示す。GUI アプリケーションでは、Figure 3 のように、ウィンドウ上に一つのクラスに纏められた行動列が表示される。そして、下部に設置されたボタンから、設計者の意図した戦術を表現する行動列には 1、それ以外には -1 をラベル付けする。

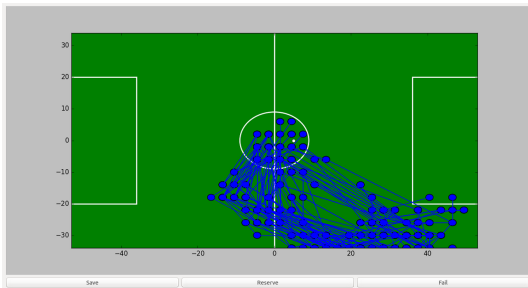


図 3: Screenshot of the developed GUI

6 数値実験

フィールドの片側サイドから攻撃するような学習用データを使って SVM を学習させ、識別精度を調査する。次に、枝刈りによる行動制御の有効性を示すため、枝刈りを組み込んだ HELIOS [10] で試合を行い、枝刈りを組み込んでいない従来の HELIOS と比較することで、行動の変化、チーム性能の変化について調査する。HELIOS は、本研究室と福岡大学により共同開発を行っている世界大会参加用のチームである。本実験において、SVM に入力する素性ベクトルは、4.2.1 項の離散化処理を行ったキック位置とする。また、行動プランニングにおいて、探索する

最大の深さを 4、最大探索ノード数を 1000 とする。連続するキックの中で、次にキックするプレイヤーが、キックしたプレイヤーと異なる場合をパス、同一プレイヤーが続けてキックを行いプレイヤーの位置座標が変化したものをドリブルと定義する。ドリブルは、キックによる位置座標の変化量が少なく、戦術的な意味合いが低いと考えられる。したがって、本論文の実験ではパスのみを枝刈りの対象とする。行動生成時の枝刈りにおいて、仮に [パス, パス, ドリブル, パス] といった長さ 4 の行動列が生成された場合、[パス, パス, パス] といったドリブルを排除した長さ 3 の行動列を識別対象とする。

6.1 SVM の学習と識別

HELIOS と Shiraz, FURY, CSU_Yunlu, InfoGraphics, Ziziphus, HERMES, Ri-one, agent2d の 8 チームとそれぞれ 100 試合を行い試合ログを作成する。agent2d [6] はベースチームとして公開されているチームであり、InfoGraphics は RoboCup2014 に参加したチーム、HERMES は RoboCup2015 に参加したチーム、Shiraz, FURY, CSU_Yunlu, Ziziphus, Ri-one は RoboCup2016 に参加したチームである。5 章で説明した操作により学習用データを作成する作成した学習用データを用いて、SVM を学習させる。作成した学習用データを用いて、10-分割交差検証を行う。グリッド数による識別精度への影響を調査するため、フィールドを 7×7 個のグリッドに分割した際の識別結果を Table 1, フィールドを 23×26 個のグリッドに分割した際の識別結果を Table 2 に示す。また、この時使用したカーネルは線形カーネルと RBF カーネルである。Table 1, 2 から、カーネルで比較した際、線形カーネルの識別精度が良いことがわかる。また、グリッド数で比較した際、 23×26 個のグリッドに分割した際に識別結果が良いことがわかる。

RoboCup サッカーシミュレーション 2D リーグにおいて、意思決定は 1 サイクル (0.1 秒) の間に行わなければならない。そのため、提案手法を組み込んだ SVM を HELIOS に組み込み、agent2d と 10 試合行い学習した SVM の実行速度を調査した。SVM の識別一回当たりの平均実行速度は、線形カーネルでは 0.0024(msec)、RBF カーネルでは 4.7474(msec) であった。このことから、実行速度に関しても線形カーネルの方が有効であると言える。

表 1: SVM's accuracy rates for a 7×7 grid

Kernel	Depth : 1	Depth : 2	Depth : 3	Depth : 4
Linear	80.31	81.98	83.65	84.60
RBF	67.93	67.93	67.93	67.93

表 2: SVM's accuracy rates for a 23×26 grid

Kernel	Depth : 1	Depth : 2	Depth : 3	Depth : 4
Linear	83.44	85.08	86.58	87.45
RBF	67.93	67.93	67.93	67.93

6.2 枝刈りによる行動制御

提案手法では、行動プランニング時の探索において、生成された行動列が設計者の意図に適しているかどうかSVMにより判断し、枝刈りを行う。6.1節の実験により得られた知見から、識別モデルとして線形カーネルを用いたSVMを使用する。また、フィールドを 23×26 個のグリッドに分割してキック位置を離散化する。HELIOSに提案手法を組み込み、Shiraz, FURY, CSU_Yunlu, InfoGraphics, Ziziphus, HERMES, Ri-one, agent2dの8チームとそれぞれ50試合ずつ行う。行動の変化を表す例として、対FURY戦における枝刈りを組み込んでいない従来のチームでのパスの軌跡と提案手法を組み込んだチームでのパスの軌跡をFigure 4に示す。また、従来のチームでの試合結果をTable 3, 枝刈りを組み込んだチームでの試合結果をTable 4に示す。

Figure 4において、上側の図が従来のチームでのパスの軌跡を表しており、下側の図が枝刈りを組み込んだチームでのパスの軌跡を表している。また、左側が自陣、右側が敵陣である。赤い線がパスの軌跡を表しており、線の色の濃さにより頻度を表している。Figure 4から、枝刈りを組み込んだチームは、従来のチームと比較して片側のサイドにパスを出す傾向が強くなっていることがわかる。Table 3, 4において、Win rateは勝率、Goals forは平均得点、Goals againstは平均失点、Through passは平均スルーパス回数、Shootは平均シュート回数、Penalty areaは敵ペナルティエリアへの平均侵入回数を表している。この時、スルーパスを、敵のオフサイドラインより自陣側からキックを行い、次にキックする位置が一つ前の敵のオフサイドラインを超えており、なおかつフィールドを3分割した際の敵陣側でキックしたものと定義する。Table 3, 4から提案手法を組み込んだチームでは、従来のチームと比較した際、得点においてCSU_Yunlu, Ziziphusでは同等程度、それ以外では減少している。得点が減少した原因として、スルーパスの回数が減少や、学習用データの作成において誤ったラベル付けをした可能性や誤識別の可能性などが考えられる。InfoGraphicsとの試合では、スルーパスの回数が著しく減少している。これは、InfoGraphicsの守備が、InfoGraphics側に向けてパスをさせないよう一列に並ぶという特徴に起因すると考える。従来は、これに対しサイドチェンジを繰り返し行い、スルーパスを行う機会を作り、突破していた。しかし、枝刈りを組み込んだチームでは、片方のサイドに行動を寄せているため、スルーパスを行う機会が減り、その結果としてシュート回数や得点が減少したことが考えられる。一方で、Ziziphusでは、ゴール前にプレイヤーを集め、守備を行うという特徴を持つ。そのため、片方のサイドから攻撃した場合でも、同様に得点に繋がる行動を行ったと考えられる。しかし、勝率で見ると、HERMES以外ではあまり変化してい

ない。このことから、勝利が可能な行動選択ができていえる。HERMESに対し勝利数が減少した原因として、HERMESは守備に偏重した陣形をとっており、得点が難しかったことが考えられる。

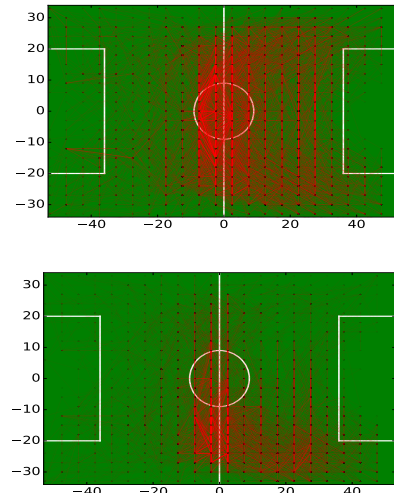


図 4: Pass distribution against FURY

7 おわりに

本論文では、評価関数の修正なしに行動制御をするため、探索木生成段階で枝刈りすることにより行動列を制御する手法を提案した。戦術を複数用意することが出来れば、より有利に試合を進めることが出来る。また、設計者の意図を反映した学習用データを作成するため、GUIアプリケーションによりラベル付けを行った。今後の課題として、ユーザビリティを考えたGUIアプリケーションの作成や、識別精度を高めるため識別モデルや入力に用いる特徴量の検討などが考えられる。

参考文献

- [1] Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, Eiichi Osawa and Hitoshi Matsubara, "RoboCup: A Challenge Problem for AI," *AI Magazine*, Vol. 18, No. 1, pp. 73-85, 1997.
- [2] 秋山 英久, "連続状態行動空間での木探索によるオンライン協調行動プランニング", 研究報告ゲーム情報学, 2012-GI-27 巻, 11号, 2012.
- [3] Thomas Gabel, Martin Riedmiller, and Florian Trost, "A Case Study on Improving Defense Behavior in Soccer Simulation 2D: The NeuroHassle Approach", *RoboCup 2007: Robot Soccer World Cup XII*, pp. 61-72, 2008.
- [4] Peter Stone and Manuela Veloso, "Task Decomposition, Dynamic Role Assignment, and Low-Bandwidth Communication for Real-Time Strategic

表 3: Performance without the proposed method

Opponent	Win rate	Goals for	Goals against	Through pass	Shoot	Penalty area
Shiraz	0.94	4.1	0.21	20.04	6.1	11.27
FURY	0.78	2.63	0.1	8.67	3.69	6.18
CSU_Yunlu	0.88	2.49	0.06	11.63	4.24	7.47
InfoGraphics	0.96	3.55	0.0	11.35	4.96	9.06
Ziziphus	0.9	1.7	0.0	25.44	3.92	13.88
HERMES	0.92	1.88	0.14	11.44	3.22	6.64
Ri-one	0.84	2.06	0.1	10.44	2.72	5.66
agent2d	0.98	9.55	0.12	19.0	11.49	17.39

表 4: Performance with the proposed method

Opponent	Win rate	Goals for	Goals against	Through pass	Shoot	Penalty area
Shiraz	0.86	3.85	0.3	21.17	6.21	11.34
FURY	0.76	2.02	0.06	8.18	3.06	5.69
CSU_Yunlu	0.86	2.36	0.0	9.42	3.58	6.34
InfoGraphics	0.86	2.2	0.0	6.78	2.78	5.33
Ziziphus	0.96	1.74	0.0	24.18	3.48	12.58
HERMES	0.54	1.16	0.22	9.72	2.4	5.72
Ri-one	0.74	1.48	0.04	9.12	2.12	4.5
agent2d	0.98	8.2	0.14	17.37	10.69	16.27

Teamwork”, *Artificial Intelligence*, Vol. 110, No. 2, pp. 241-273, 1999.

Germany, 6 pages, 2015.

- [5] Hidehisa Akiyama and Itsuki Noda, “Multi-Agent Positioning Mechanism in the Dynamic Environment”, *RoboCup 2007 : Robot Soccer World Cup XI*, 2008.
- [6] Hidehisa Akiyama and Tomoharu Nakashima, “HELIOS Base: An Open Source Package for the RoboCup Soccer 2D Simulation”, *RoboCup 2013 : Robot World Cup XVII*, pp. 528-535, 2014.
- [7] 三船 哲史, 中島 智晴, 秋山 英久, 関 宏理, “RoboCup サッカーにおける SIRM’s ファジィシステムを用いたログからの行動評価”, 第 40 回人工知能学会 AI チャレンジ研究会論文集, pp. 20-25, 2014 .
- [8] 田川 諒, 五十嵐 治一, “サッカーエージェントにおける局面評価関数の強化学習”, 第 20 回ゲームプログラミングワークショップ 2015 論文集, pp. 78-83, 2015 .
- [9] 牛腸 俊寛, 秋山 英久, 荒牧 重登, “人間の行動選択ログを用いたサッカーエージェントの行動評価関数獲得”, 第 32 回ファジィシステムシンポジウム論文集, pp. 61-64, 2016 .
- [10] Hidehisa Akiyama, Tomoharu Nakashima, Jordan Henrio, Thomas Henn, Sho Tanaka, Tomonari Nakade and Takuya Fukushima, “HELIOS2016 Team Description Paper”, *RoboCup 2016 Leipzig*,

RoboCup Logistics Leagueにおける一方通行経路設定に対する格子状経路の追加に関する一考察

About One-Way Routing with Mesh Points at RoboCup Logistics League

○辻 和輝, 津田 諒太, 木綱 智, 内海 孝亮, 植村 渉

Kazuki TSUJI, Ryota TSUDA, Tomo KIZUNA, Kosuke UTSUMI and Wataru UEMURA

龍谷大学

Ryukoku University

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

Abstract

現在、生産現場では作業の効率化を図って機械による Factory Automation (FA) 化が進んでいる。多品種少量生産が必要とされており、それに伴い、加工機械の入れ替わりが頻繁に起きるようになってきている。製品を搬送する移動式ロボットは、それら加工機器間を移動する必要があり、移動経路を状況に応じて設計する必要がある。特に複数台のロボットが混在する環境では、お互いのロボットの経路計画を考慮するのが難しい場合もあり、ロボット間通信を行わずに経路設計をする必要がある。一方通行ルールをロボットに導入することで、それら通信問題の影響を受けない経路設定方法が考えられる。本研究では、一方通行の経路設定に対し、格子状の経路を追加することで、その効果を評価する。

1 はじめに

現在の生産現場においては人間が行っている作業を機械が代わりに行う Factory Automation (FA) 化が進んでいる。さらに、顧客のニーズの多様化や市場の変化に対応するべく、多品種少量生産の生産形態を取ることが多く、それに伴い加工機械の頻繁な入れ替えが想定される。それらの環境において部品や製品を搬送する移動式ロボットは、移動経路計画を状況に応じて設計する方法が必要である。各ロボットが移動する際に電波状況の問題や管理サーバの不調等により、各ロボット間の通信が困難な場合がある。ロボット間通信を行わない場合、他のロボットの経路に干渉して、渋滞やデッドロックを引き起こす可能性がある。そこで解決方法として一方通行の移動ルールを与えた移動経路計画を検討している[5]。

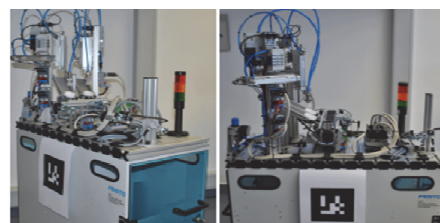


Figure 1: 競技で使う加工マシン (Module Production System)

2 RoboCup Logistics League (RCLL)

2050年に人型ロボットでワールドカップ・チャンピオンにサッカーで勝つことを目的として、ロボットの自律制御技術を競う競技会である RoboCup が 1997 年より開催されている。RoboCup には現在サッカー以外を扱う競技もあり、その中の一つである RoboCup Industrial には、Logistics League と呼ばれるリーグがある。RoboCup Logistics League (RCLL) では、Festo 社製の移動式ロボットである Robotino 3 を用いて、生産現場における FA 化を想定した環境で競技を行っている。RCLL では作業環境の探索、4 種類の Module Production System (MPS) (Figure 1 による製品の加工、製品の納品の工程を競技としている。

RCLL の試合は 3 つの Phase に分かれており、Setup Phase, Exploration Phase, Production Phase である。実際に Robotino 3 が行動をするのは Exploration Phase と Production Phase になる。RCLL におけるゲームフィールドを Figure 2 に示す。

RCLL では 1 チーム最大 3 台の Robotino を用いて、2 チームが同一のフィールドで競技を行う。Figure 2 のシアーンとマゼンタの長方形が MPS を示しており、フィールド内に各チーム 6 台、計 12 台の MPS が配置される。Figure 2 における MPS の位置、角度は一例を示したものであり、

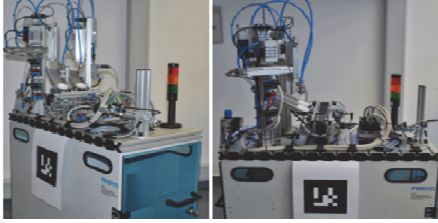


Figure 2: RCLL2016 のフィールド図

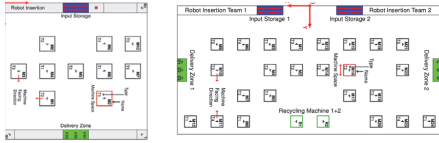


Figure 3: RCLL の 2013 年 (左) と 2014 年 (右) のフィールド図

配置されるゾーンは固定されているが、実際の競技においてはフィールドの y 軸に沿って線対称になるように MPS の種類とゾーン内の位置と角度が毎試合毎にランダムで配置される。

2.1 RCLL における移動経路計画

RCLL では 2015 年からルールの変更があり現在のフィールドの形が取られている。2014 年までは MPS を用いずに加工用マシンを直接床上に配置していた。2013 年のフィールドは $5.6m \times 5.6m$ であり、2014 年のフィールドは $11.25m \times 5.6m$ であった。それぞれのフィールドの周囲の一部には高さ $0.5m$ の壁がある。それぞれの年度のフィールド図を Figure 3 に示す。フィールド中の正方形が作業機械を示しており、ゲームごとに作業機械の配置場所は固定であるが 90 度ごとの向きは変更するようになっている。2014 年までのルールではチームごとにフィールドが存在しており、相手チームは別のフィールドを利用していたため、同一フィールド上で移動するロボットの情報は既知であった。また、作業機械の位置も固定であったため、フィールドをあらかじめマス目状に区切ることが可能であった。著者らのチームである BabyTigers-R では、マス目に沿って移動方向を設定することでロボット間の衝突回避を行った (Figure 4) [2]。

2015 年から新たなルールとなり、加工機械として MPS が用いられることになった。MPS の配置場所がランダムになったため、フィールドをマス目で区切ることが難しくなった。Exploration Phase においては Robotino がフィールドを探索し、MPS が置いてあるエリアを探し、MPS の種類とエリアの情報を RefBox へと報告する必要がある。MPS が配置エリアのどこに配置されているかわからない状態でセンサで環境情報を検知しながら移動するため経路設計においても MPS の検知に合わせて設定をし直し

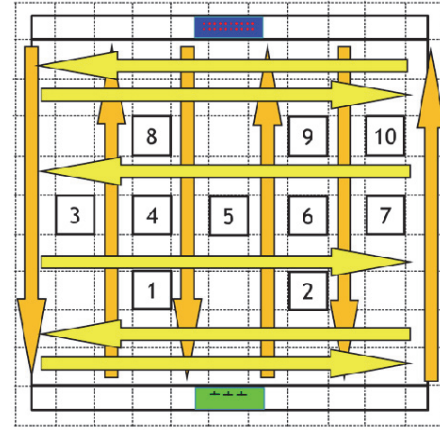


Figure 4: 2013 年の移動経路

ていく必要がある。そこで環境情報の更新に応じて移動経路の再設定をする一方通行ルールを検討した[5]。

3 作成する移動経路

現在の RCLL では試合ごとに MPS の向きと種類がランダムでフィールドに設定されるが、配置されるエリアとチームごとの台数は固定されている。そのため、MPS が配置されることがなく、試合ごとに MPS の配置の影響を受けることのないエリアが存在している。Figure 5 の Z17 から Z5 までの並んだエリアと、Z20 から Z8 までの凹型のエリアが該当する。それらのエリア間においては常に固定の経路を設定することができる (Figure 5)。そこで、それら 2 つの閉じた経路に対して一方通行を設定し、さらに閉じた経路間をつなぐ経路を用意する方法を提案した[5]。しかし、この方法では、現在地と目標地の組み合わせによっては、遠回りの経路を選択することになり、効率が悪い。

そこで、本研究ではそれぞれの一方通行経路内においてショートカットとなる経路を追加することで、効率の改善を図る。今回は経路の追加方法として 2 つのパターンを検討する。

一つ目の方法として、Figure 6 の上図であり、左右対称に経路を追加する方法である。これを提案法 1 とする。二つ目の方法として、Figure 6 の下図であり、上方向きの経路を追加する方法である。これを提案法 2 とする。これらの効果を確認する。ここでは、実際に MPS やそれに類似する寸法の物体の設置や複数台のロボットを用意することが難しいため、シミュレータである Gazebo を用いて検証する。

4 実験

RCLL においてロボット間通信を行わない状況でのロボットの移動時間を比較する。一方通行経路内にショートカットとなる経路を追加し、その効果を確認する。比較用の移

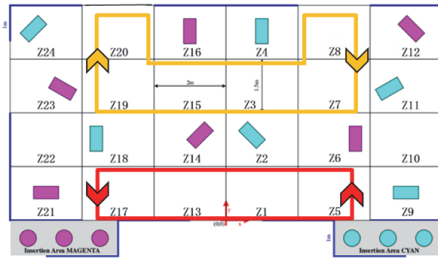


Figure 5: 一方通行の移動経路

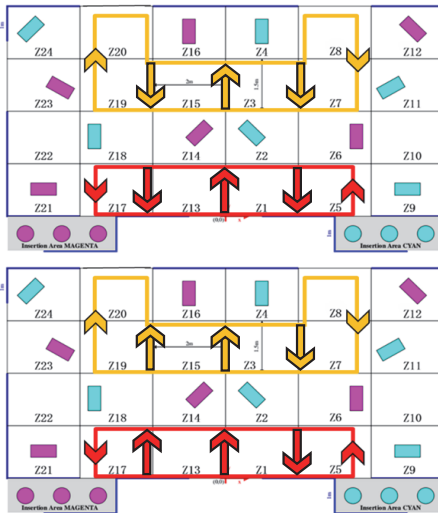


Figure 6: 追加した経路

動経路として、RCLLの優勝チームであるアーヘン工科大学チーム Carologistics のプログラムを用いる (Figure 7) [3].

結果を Figure 8 に示す. 提案法 1 が Figure 6 の上図, 提案法 2 が Figure 6 の下図の結果を示している.

5 おわりに

本研究では Logistics League の環境において一方通行ルールを実装し, さらに閉じた経路間にショートカットとなる経路を加えることを提案した. ロボットが移動を行った際に移動時間がどのように変化するかを比較した. 今後の課題としては German Open2017 および, 名古屋世界大会において実環境において動作させることである.

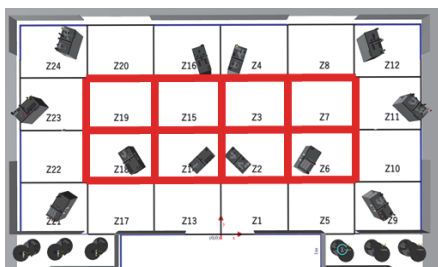


Figure 7: Carologistics の経路計画の骨格となるルート

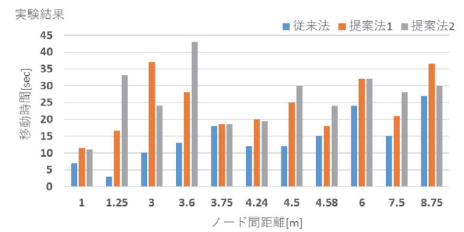


Figure 8: 実験結果

謝辞

本研究は JSPS 科研費 15K16313 の助成を受けたものである.

参考文献

- [1] <http://www.robocup.org>
- [2] Wataru Uemura, et. Al, "Team Description Paper. BabyTigers - R", RoboCup 2013: Robot World Cup XVII, 2013.
- [3] Tim Niemueller, et. Al, "The Carologistics RoboCup Logistics Team 2013", RWTH Aachen University and Aachen University of Applied Sciences, 2013.
- [4] Tim Niemueller, et. Al, "Fawkes for the RoboCup Logistics League", RoboCup Symposium, Hefei, China, 2015, <https://www.fawkesrobotics.org/projects/rc112015-release>.
- [5] 辻和輝, 植村 渉, "Fawkes を用いたマルチロボットにおける一方通行ルールの提案と評価", 人工知能学会第 45 回 SIG-Challenge 研究会, pp. 10 - 12, 2016.
- [6] 小田翔平, 植村 渉, "可視光通信における信号定位に関する一考察", 第 19 回創発システムシンポジウム, p. 34, 2013.

可視光通信を用いたロボット制御のための信号強度取得に関する一考察

Signal Level with the Pulse Modulation in Visible Light Communication for Controlling Robots.

○清水 謙汰, 端山 稜人, 福森 康洋, 植村 渉

Kenta SHIMIZU, Takato HAYAMA, Yasuhiro FUKUMORI and Wataru UEMURA

龍谷大学

Ryukoku University

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

Abstract

工場において、少量多品種の生産ラインへの対応が求められており、汎用型移動式ロボットが必要とされている。本研究では、複数台の同一種のロボットを無線コントローラで遠隔制御する時の通信方法を検討する。

複数台の同一種のロボットをコントローラで遠隔操作するとき、各ロボットの識別番号を確認しないと、操縦対象がわからなくなる。目に見える無線通信として注目されている可視光通信をロボットのコントローラに使うことで、対象物を明確に把握できるようになる。

一方、無線でロボットを操縦するときには、操縦体系にもよるが、方向に関しては右や左といった相対的な扱いをすることが多く、リモコン操縦者の向きと操縦対象物の向きが一致せず、操縦者に一定の慣れとスキルが必要となる。これを防ぐためには、相対的な移動指示ではなく、絶対的な移動指示が必要となる。つまり、受信機側であるロボットが、操縦者との位置関係を獲得しないとイケない。この方法には、いくつかの方法が考えられるが、ここでは可視光通信を使っていることを利用し、受光素子の受信強度分布から発信源の向きを推定する方法を検討する。

本研究では、パルスを用いた可視光通信における、受信信号強度の取得方法について検討する。

1 はじめに

工場において、少量多品種の生産ラインへの対応が求められており[2]、Industrie 4.0におけるスマートファクトリーを代表とするファクトリーオートメーション化が注

目されている。そのような工場では、汎用型移動式ロボットが必要とされている[3]。

本研究では、複数台の同一種のロボットをコントローラで遠隔制御する時の通信方法に注目する。複数台の同一種のロボットをコントローラで遠隔操作するとき、有線通信であれば一台一台にコントローラをつなぐ必要があり、ケーブルの取り回しが複雑になる。Wi-Fiに代表されるIEEE802.11などの無線通信を用いた場合、ケーブルの取り回しの問題は解決するが、各ロボットの識別番号を確認しないと、操縦対象がどれかわからなくなる。これは、無線通信の接続状況が目に見えないため、どのロボットと通信しているが把握できないからである。

そこで、本研究では、目に見える無線通信として注目されている可視光通信をロボットのコントローラに使うことを検討する。通常のリモコンでも同様であるが、無線でロボットを操縦するときには右や左といった相対的な扱いをとると、リモコン操縦者の向きと操縦対象物の向きが一致せず、操縦者に一定の慣れとスキルが必要となる[4]。これを防ぐためには、相対的な移動指示ではなく、絶対的な移動指示が必要となる。つまり、受信機側であるロボットが、操縦者との位置関係を獲得しないとイケない。この方法には、いくつかの方法が考えられるが、ここでは可視光通信を使っていることを利用し、受光素子の受信強度分布から発信源の向きを推定する方法を検討する。

しかし、可視光通信で情報を光に重畳すると、光源が高速で点滅するため受信強度の変動が起こる。本研究では、パルスを用いた可視光通信における、受信信号強度の取得方法について検討する。

以下、2章ではパルスを用いた可視光通信について紹介し、変調時の受信信号強度の取得における問題点を挙げる。発光素子、受光素子の過渡現象の影響を明らかにし、3章にて矩形波の立ち上がりを検知することで過渡現象の影響のない時間帯に信号強度を獲得する方法を提案し、4章にてその効果を実験にて確認する。そして、5章で本

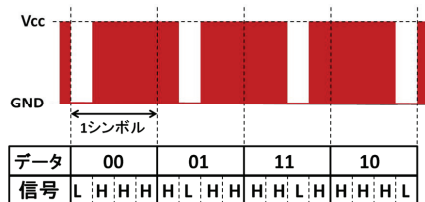


Figure 1: 4 値-IPPM による情報と信号の関係の一例

論文をまとめる。

2 可視光通信

可視光通信は、LED などの白色光の点滅パターンに情報を重畳する通信方法である。現在のところ多重化を考慮する場面が少ないため、可視光通信や赤外線通信では隣接チャンネルへの影響を考慮しなくて良く、帯域を制限する必要が無い。そのため、送信側でフィルタを通す必要が無く、正弦波ではなくパルスを用いた通信が可能となる。それゆえ、マイコンのデジタル出力ピンを用いて、簡単に送信波形を作り出すことができる。一般に、赤外線通信ではパルスの幅を用いて情報を送る Pulse Width Modulation (PWM) が主流であるが、可視光通信ではちらつきの影響を抑えるために単位時間あたりの点灯時間が一定になる Pulse Position Modulation (PPM)¹ の利用が一般的である (図 1)。

可視光通信は、光照射範囲内が受信可能範囲となるため、通信対象物を目で確認できる特徴がある。これをロボットの制御に応用すると、ロボットを懐中電灯で照らすような感じで、コントロールすることができる。また、レンズなどを用いて光照射範囲を広くしたり狭くしたりして、複数のロボットを照らすことができれば、それら照らされた全部のロボットを一度に操縦することも可能となる。

一方、無線でロボットを操縦するときには、操縦者はほぼ動かないことが多く、リモコン操縦者の向きと操縦対象物の向きが一致しないことが頻繁に発生し、操縦者に一定の慣れとスキルが必要となる。

また、複数台のロボットを同時に動かす場合、それぞれのロボットに対して相対的な向きを用いて操縦すると、向きの異なるロボットは、それぞれ異なる方位を基準とし、結果としてばらばらな動きになる。例えば、「右に進む」という操縦をした場合、操縦者と同じ向きを向いているロボットは操縦者から見て右に進むが、操縦者の方を向いているロボットは操縦者から見て左に進む。

これを防ぐためには、相対的な移動指示ではなく、絶対的な移動指示が必要となる。つまり、受信機側であるロ

¹ 照明として利用する場合は、点灯割合が長くなる Inverse PPM (IPPM) が一般的であり、4 値-IPPM がよく使われるが、本研究の実験では簡単のため 2 値 PPM を用いる

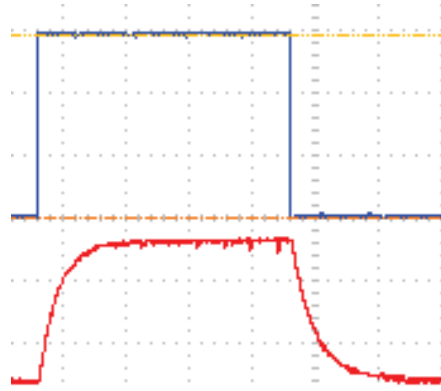


Figure 2: 発光素子と受光素子の過渡現象による送信波形の矩形波の波形変化の一例

ボットが、操縦者との位置関係を獲得しないといけない。この方法には、いくつかの方法が考えられるが、ここでは可視光を使っていることを利用し、受光の明るさつまり受光素子の受信強度分布から発信源の向きを推定する方法を検討する。

電波や電磁波では、送受信機間距離が離れると、受信信号強度は指数的に減少する。さらに電波の場合は波長の関係から、腹と節が生じて単調に減少しない[5]。また、マルチパスの影響も加わり、受信強度から送受信機間距離を推定するのは困難である[6]。一方、可視光領域の電磁波では波長が短いため、マルチパス等の影響が出にくく、受信信号は単調減少となり、複数の受光素子を用いることで、送信源の方向を知ることができる[7]。

本研究では、そのようなパルスを用いた可視光通信における、受信信号強度の取得方法について検討する。

3 過渡現象が受信信号強度に与える影響

まず、可視光通信の変調の影響を確認する。矩形波で情報を送るため、受信信号強度は点灯時と消灯時で異なる。以後、これらを HIGH の時と LOW の時と呼ぶ。当然ながら、送受信機間距離を推定するには、HIGH の時の受信強度が必要である。移動体の場合、HIGH と LOW のいずれの受信強度も変動する。送受信機間距離によって HIGH の強度が変化し、受信機のある場所の明るさによって HIGH と LOW の両方の強度が変化する。そのため、HIGH と LOW の時の受信強度を区別するには、何度か強度を測定することで、HIGH の時と LOW の時の 2 つのグループに分けることができる。ここで通信速度を上げて、受光側のサンプリング周波数を上げると、図 2 のように発光素子や受光素子の応答速度の影響が出てくる。このとき、過渡特性による電圧変動が占める割合が大きくなり、本来の HIGH の受信強度よりも低い値を測定することが多くなる。この過渡現象の影響を確認する。送受信機間距離を 10cm から 40cm まで 10cm ずつ離れた状態で 100 回測定

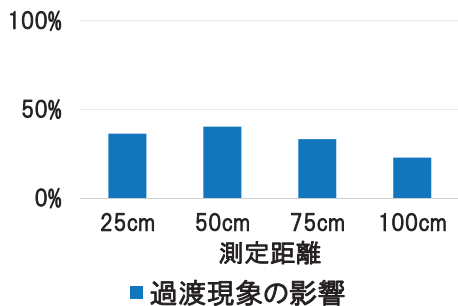


Figure 3: 発光素子と受光素子の過渡現象による送信波形の矩形波の波形変化の一例

を行う。その状態における HIGH 信号と LOW 信号の判定基準を知るために、無変調の点灯状態と消灯状態の受信信号強度を、変調の影響のない強度として扱う。ただし、外乱光などのノイズの影響があるため、それぞれ 10,000 回測定を行い、HIGH 信号に対しては過渡現象の影響を考慮して最小値の 63.2% ($=1 - (1/e)$) の電圧を、LOW 信号に対しては最大値を基準値として扱う。PPM で情報を送ったときに、測定した受信信号強度が HIGH の基準値以上、もしくは LOW の基準値以下であるときは正しく復号できるが、それ以外の値のときは復号結果は誤りとなる。2 値 PPM で情報を送信したときの誤りの割合を図 3 に示す。距離によって、基準電圧が変わるため、過渡現象の影響も変化しているが、過渡現象の生じる時間割合と同様の比率で、復号結果の誤りが生じていることがわかる。

4 変調を考慮した受信強度の取得方法の提案

受信信号強度取得の際に、過渡現象の影響を受けている状態を避ける必要があるため、本研究では、信号強度の変化を観測して変化後に信号強度を取得する方法を提案する。

過渡現象が生じるのは、信号が LOW から HIGH、もしくは HIGH から LOW に変化する時である。また、LOW の時には受信信号強度が測定できないため、LOW から HIGH に変わる部分の検知が重要である。過渡現象は、通信速度に依存し、通信速度を決めれば、過渡現象が生じる時間を推測することができる。そこで、受信信号強度を測定するタイミングとして、LOW から HIGH に変化したのを確認した後、一定時間経過後に AD コンバータにて強度を測定する方法を提案する。これにより、過渡現象の影響のある時間領域を避けることができ、適切な受信強度を測定することができる。

5 実験と考察

変調による波形の変化を考慮して受信強度の測定タイミングを調整した提案方法の効果を確認する。先の実験と

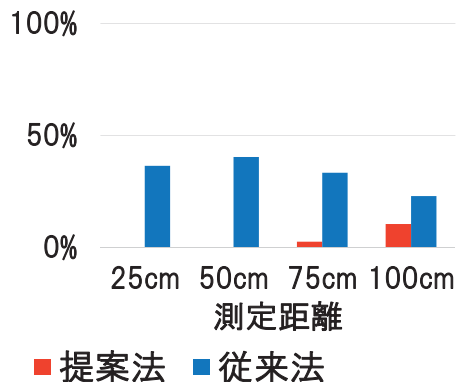


Figure 4: 変調による波形の変化を考慮して受信強度の測定タイミングの調整の有無による受信強度の変動割合

同じ条件で測定した結果を図 4 に示す。過渡現象の生じる時間帯を避けていることが確認できる。一方で、通信距離が離れると信号強度そのものが小さいため、一定時間経過時には強度が減少し、LOW と判定するため誤りの割合が上昇した。

6 おわりに

ロボットの制御に可視光通信を用いるとき、送信機と受信機間の相対的な向きを知ることができれば、絶対的な角度による制御が可能となり、操縦者の負担が大きく軽減できる。送受信機間の向きを知るには、受光素子の受信強度を使うことで送信機の方向を得ることができる。ここで、矩形波を用いた可視光通信において、通信速度を上げると過渡現象の影響が現れて矩形波の HIGH と LOW の間の電圧値をとる時間割合が大きくなり、受信強度が本来の値より小さくなる。本研究では、その受信強度変化が発光素子と受光素子による過渡現象であると考え、その影響を避けるために変調を考慮して受信強度の測定を行う方法を提案した。結果、近距離では過渡現象の影響を避けることができたが、遠距離になると信号強度の減衰の影響がでてきて、適切に強度を測定できなかった。今後は、距離に応じて、過渡現象の影響時間を調整することで、遠距離時にも適切に信号強度を測定できるようにし、可視光通信によるロボットの制御を実現したい。

謝辞

本研究は JSPS 科研費 15K16313 の助成を受けたものである。

参考文献

[1] 中川 正雄, 可視光通信コンソーシアム, “可視光通信の世界—LED で拓く「あかりコミュニケーション」”, 工業調査会, 2006.

- [2] RoboCup Logistics League, <http://www.robocup-logistics.org/>
- [3] World Robot Summit, <http://www.worldrobotsummit.org/>
- [4] 辻和輝, 植村渉, “全方位移動ロボットに対する移動指示インタフェースの評価”, 人工知能学会第 42 回 SIG-Challenge 研究会, pp. 28 – 33, 2015.
- [5] 植村渉, “ad-hoc ネットワークにおける音波を用いた端末間距離測定に関する一考察”, 人工知能学会第 23 回 SIG-Challenge 研究会, pp. 29 – 33, 2006.
- [6] 上東朋寛, 植村渉, 村田正, 阿部宏尹, “ZigBee を利用したベイズ推定による 2 次元位置推定の精度の評価と考察”, 平成 22 年電気関係学会関西連合大会, pp, 3A303 – 28, 2010.
- [7] 小田翔平, 植村渉, “可視光通信における信号定位に関する一考察”, 第 19 回創発システムシンポジウム, p. 34, 2013.

可視光通信の利得制御方法に関する一考察

About gain controll in in Visible Light Communication.

○端山 稜人, 植村 渉

Takato HAYAMA and Wataru UEMURA

龍谷大学

Ryukoku University

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

Abstract

目に見える無線通信として注目されている可視光通信をロボットのコントローラに使うことが検討されている。このとき、外乱光等のノイズに対応するためにフィルタを使うことが多いが、パルスを用いた通信の場合、フィルタを通すと波形が変化するため、情報が失われる危険性がある。

本研究では、パルスを用いた可視光通信において、ハイパスフィルタの特性を考慮して、フィルタ通過後の波形から元波形を復元する方法を検討する。受光素子の感度調整抵抗が受信波形に与える影響を考慮し、自動調整する方法を提案する。

1 はじめに

スマートファクトリーに代表されるように工場内でのロボットによる自動化が注目されている。複数のロボットが独立して動いているときに、それらを無線通信でリモートコントロールする方法として、目に見える無線通信である可視光通信の利用が提案されている[2]。

可視光通信の送信機では、発光素子として Light Emitting Diode (LED) を使用する。LED は応答速度が速いため、高速点滅が可能であり、その点滅パターンを変えることで情報を重畳できる。また、LED の制御はマイコンの出力ピンで簡単に扱えるため、可視光通信では矩形波通信を用いることが多い。矩形波のパルスの位置や長さを変えて情報を送るが、この場合、ノイズ除去のためにハイパスフィルタを通すと、パルスの形が変化し、情報を失う。一方、元の波形がパルスであることが分かっているため、波形の立ち上がりや立ち下がり点さえわかれば、パルスを復元することができる。このとき、受信信号の利得を適切

に制御しないと、立ち上がりや立ち下がり点が変わる。そこで本研究では、受光素子感度の自動調整を検討し、受信信号の利得制御方法を提案する。

2 可視光通信

可視光通信は無線通信の一種であり、光の点滅パターンを変えることで、情報を送信する (Figure 1) [3]。送信用の発光素子として LED を用いることが多い。LED は、白熱電球や蛍光灯の照明などと比較して省電力、長寿命、高速応答、小型といった特徴を持っている。

一方、受光用の受光素子に使う可視光センサとして、フォトダイオードやフォトトランジスタがある。これらは明るさに応じて素子を流れる電流が変化し、抵抗の変化や電圧の変化として取得できる。電圧の変化を読み取るためには、抵抗を介して分圧することが一般的である。この抵抗の抵抗値で受光素子の光に対する感度および応答速度が決定する。

抵抗の抵抗値が大きいと電圧変化の範囲が広くなり、感度が良くなる。それゆえ、送受信機間距離が離れても信号を受信できるが、送受信機間距離が短いと HIGH から LOW への電圧変化に時間がかかり、応答速度が遅くなる。一方、抵抗値が小さいと応答速度は速いが、送受信機間距離が離れると受信信号が弱くなり信号を正しく復号できなくなる (Figure 2)。

可視光通信では、照明としての利用も考えると送信光として白色を使うことが多い。隣接周波数への影響を考慮しなくて良いため、送信波形の帯域は自由に扱える。それゆえ、正弦波ではなく矩形波のままで情報を送信することが可能であり、パルスを用いた変調が主流である。その際に、ちらつきを考慮する必要がある。単位時間あたりの点滅のデューティ比が変化すると明るさに変化が生じるため、人の目から見てちらつきを感じやすい。Pulse Position Modulation (PPM) 変調はパルスの位置の変化で情報を送るため、1 シンボルあたりの点灯と消灯時間は



Figure 1: 可視光通信の一例

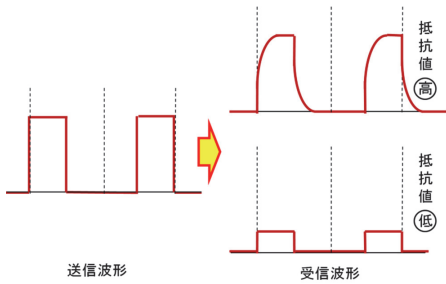


Figure 2: 抵抗値の変化による波形の変化

変化せず、人の目から見てちらつきを感じにくい (Figure 3). 本研究では、2 値 PPM 変調に対して説明を行うが、4 値 PPM や 4 値 I-PPM など他の PPM でも同様である。

また、可視光通信には自然光や人工光などのノイズへの対策も必要である。一般にノイズ対策として、フィルタの利用があげられる。太陽光などの自然光は、点滅しておらず直流成分として扱える。蛍光灯などの照明は、 $50Hz$ もしくは $60Hz$ で点滅しているが、低い周波数である。一方、可視光通信は高速点滅の波形であるため高周波成分として扱える。それゆえ、自然光や照明光を除去するためには、ハイパスフィルタが有効である。簡単なハイパスフィルタとしてコンデンサと抵抗で構成された CR ハイパスフィルタがある。CR ハイパスフィルタは、入力信号に並列する抵抗と直列するコンデンサから成り立っている。抵抗値と容量値の積は時定数 τ といい、遮断周波数に逆比例する。またこの回路は電圧が変化したときに大きく反応するため、微分回路でもある。それゆえ、矩形波

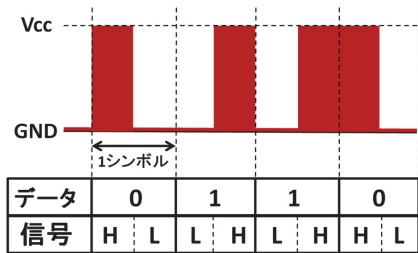


Figure 3: 2 値 PPM 変調による情報の送信方法

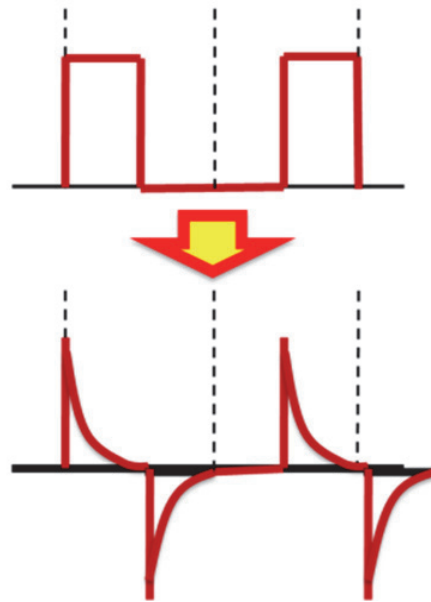


Figure 4: ハイパスフィルタによる波形の変化

を入力すると、立ち上がり立ち下りの部分に対してパルスが生じ、その後電圧が減少する波形となる (Figure 4). 可視光通信においてハイパスフィルタを使用すると、矩形波の形が変化するため、矩形波の長さや位置の情報が失われるため、復号が困難になる。

3 受光素子感度の自動調整方法の提案

電波通信などの無線通信と同様、可視光通信においても送受信機間距離が離れると、受信電圧が下がり、矩形波通信の場合は閾値電圧を下回ると復号できなくなる。その場合、受信機側の受信信号の増幅率を上げたり、受光感度を上げたりして、閾値電圧よりも高くなるように調整する。なお、増幅率を上げることは、相対的に閾値電圧を下げることに等価である。

ここでは、受光感度に注目し、自動調整の方法を提案する。受光感度を可変するために、デジタルポテンシオメータを用いる。デジタルポテンシオメータとはアナログ可変抵抗の抵抗体の上を可動する接点をデジタル信号で制御できる可変抵抗である。フォトトランジスタに使用するプルダウン用の抵抗にデジタルポテンシオメータを使用する。これにより受光素子感度の自動調節を行うことが可能になる。

感度が変わることで応答速度が変化するが、矩形波通信の場合、その波形の立ち上がり立ち下りの開始点は変化しない。そのため、立ち上がり開始時と立ち下り開始時の変化点を適切に扱うことで、元の矩形波を復元することが可能である (Figure 5)。

しかし、感度を強くすると、立ち下りの開始点が遅れることがある。波形のピークが V_{cc} の電圧値を超える時、

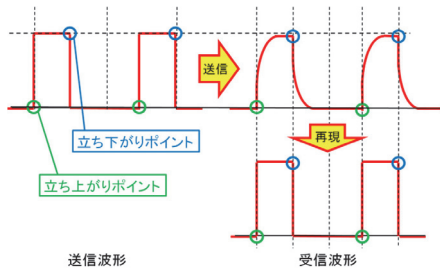


Figure 5: 立ち上がり立ち下がりポイントからの元波形の復元

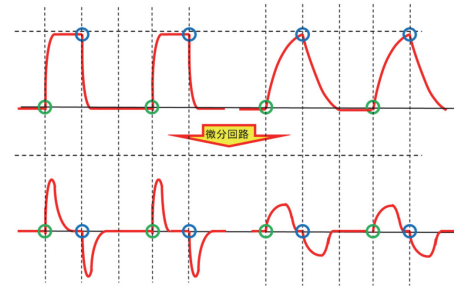


Figure 7: ハイパスフィルタを通した時の立ち上がり開始点と立ち下がり開始点の変化

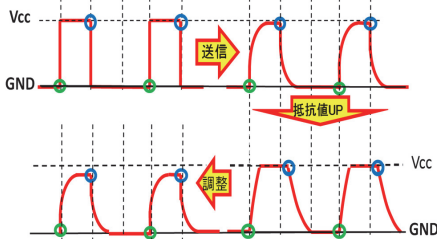


Figure 6: 感度を強くしたときの立ち下がり開始点の遅れ

その間の電圧は V_{cc} となる。このとき、立ち下がりの開始点が遅れ、パルスの位置情報が変化する (Figure 6)。つまり、波形のピークが V_{cc} の電圧値を越えないように、受信感度を調整する必要がある。

立ち上がり立ち下がりの開始点は、それぞれ GND と VCC からの電圧変化点である。受信波形の復元にマイコンを使っているとき、通常は HIGH と LOW の信号が判別できれば良いため、デジタル入力ピンで対応できる。しかし、立ち上がり立ち下がりの開始点を見つけるには、アナログ値の変化を確認する必要があるため、マイコンでは動作速度の低下が起こる。立ち上がり開始点に対しては、外部のコンパレータを利用し、比較電圧として GND より少し高い電圧を与えれば、判別が可能である。一方、立ち下がり開始点に対しては、その時の矩形波の電圧値からの変動となり、その電圧値は送受信機間距離に依存するため、設定が難しい。

ここでハイパスフィルタに注目する。ハイパスフィルタには微分の働きがあり、立ち上がり開始点と立ち下がり開始点の変化を抽出することができる。また、ハイパスフィルタを通して波形が変化しても、立ち上がり立ち下がり開始点は変化しない (Figure 7)。そして、それら立ち上がり開始点と立ち下がり開始点の電圧を基準電圧 (GND) として扱うことができ、そこからの正の方向に変化する点が立ち上がり開始点であり、負の方向に変化する点が立ち下がり開始点となる。

立ち上がり開始点は、コンパレータで抽出できるが、立ち下がり開始点は負電圧のため、抽出回路が複雑になる。そこで、抵抗を用いた電圧を印加することで、基準電圧を

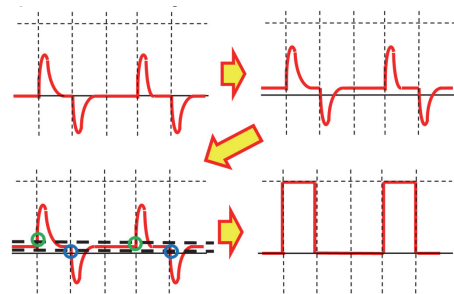


Figure 8: 立ち上がり開始点と立ち下がり開始点の抽出方法

少し高くすることで、立ち下がり開始点も正電圧として扱う。これにより、立ち下がり開始点からの電圧変化もコンパレータで抽出できるようになる (Figure 8)。

ただし、ハイパスフィルタを通った波形は、パルスの時間幅が短くなるため、立ち上がり開始点や立ち下がり開始点から基準電圧に戻るまでの時間が短くなる。この時間がマイコンの処理速度よりも短い場合、パルスを抽出できない場合が生じる。このパルス時間幅は、ハイパスフィルタの性能に依存し、フィルタの抵抗値およびコンデンサの容量に比例して長くなる。そこで、 CR ハイパスフィルタのカットオフ周波数 ($= 1/2\pi CR$) との関係の時定数と比較し Figure 9 に示す。使用するマイコンの処理速度に応じて、ハイパスフィルタのカットオフ周波数を決定する必要がある。

この方法により、立ち上がり開始点と立ち下がり開始点を適切に抽出することで、送信波形を復元でき、矩形波通信の復調が可能となる。

4 実験と考察

提案する受光感度の自動調整の効果を確認するために、受光感度を固定した方法との比較実験を行う。受光素子の感度調整抵抗値が 0Ω から $27k\Omega$ まで 128 段階で設定できるデジタルポテンショメータを用いる。一方、比較対象として抵抗値を $10k\Omega$ と $27k\Omega$ に固定した受信機を用いる。比較対象の回路は、コンパレータの閾値を、無信号時のノイズ電圧の少し上に設定し、遠距離の通信に適した状態に

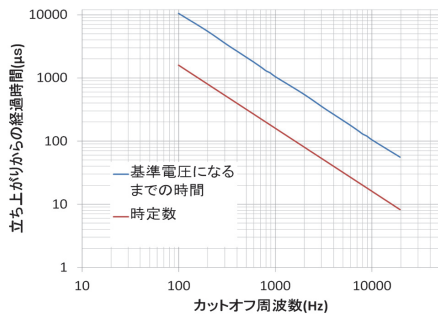


Figure 9: ハイパスフィルタのカットオフ周波数とパルスの時間幅の関係

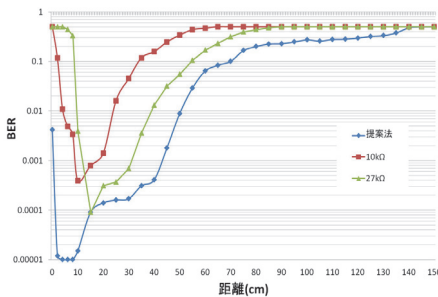


Figure 10: 通信速度 6.25kbps での実験結果

する。送受信機間距離を 0cm から 150cm まで変えながら、100万ビットの通信によるビット・エラー・レート (BER) を測定する。なお測定点として、送受信機間距離が 0cm から 10cm の間は 2cm ごとに、 10cm 以上は 5cm ごととする。また、通信速度の影響を確認するために 6.25kbps , 8.30kbps , 12.50kbps の3つの速度で情報を送信して、違いを測定する。

通信速度が 8.30kbps と 6.25kbps の場合、提案法では近距離と遠距離のいずれの距離でも抵抗値を適切に調整できており、さらに立ち下がり開始点の抽出の効果も加わり、抵抗値を固定した場合よりも良い性能になっている (Figure 10, 11)。一方、通信速度が 12.5kbps のときは、送受信機間距離が 20cm 以上になると、提案法の方が悪くなった (Figure 12)。これは、パルスの時間幅が約 $100\mu\text{s}$ と短く、マイコンが立ち上がり開始点や立ち下がり開始点

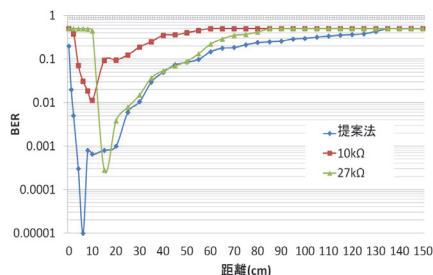


Figure 11: 通信速度 8.30kbps での実験結果

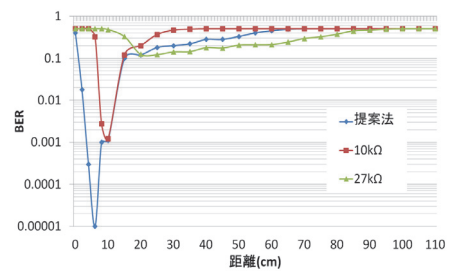


Figure 12: 通信速度 12.5kbps での実験結果

を取りこぼしているためである。今回は、 16MHz で動作するマイコンを用いたが、このマイコンの性能の限界であり、これを用いた場合の通信速度の上限であると考えられる。

5 おわりに

目に見える無線通信として注目されている可視光通信をロボットのコントローラに使うことが検討されている。本研究では、パルスを用いた可視光通信において、受光素子の感度を調整する方法について検討した。ハイパスフィルタの特性を考慮して、フィルタ通過後の波形から元波形を復元する方法を提案した。波形の立ち上がり開始点と立ち下がり開始点を抽出することで、元波形を復元した。近距離と遠距離のいずれにおいても、感度を固定した時より良い結果となった。今後は、この受信機をロボットの可視光通信の受信機として用いる必要がある。

謝辞

本研究は JSPS 科研費 15K16313 の助成を受けたものである。

参考文献

- [1] 中川 正雄, 可視光通信コンソーシアム, “可視光通信の世界—LED で拓く「あかりコミュニケーション」”, 工業調査会, 2006.
- [2] 清水 謙汰, 端山 稜人, 福森 康洋, 植村 涉, “可視光通信を用いたロボット制御のための信号強度取得に関する一考察”, 人工知能学会第 47 回 SIG-Challenge 研究会, (印刷中), 2017.
- [3] 小田翔平, 植村涉, “可視光通信における信号定位に関する一考察”, 第 19 回創発システムシンポジウム, p. 34, 2013.

ROSを使って移動ロボットを実現するための マイコン用通信ライブラリの実装と移動ロボットの実現例

光永 法明, 杉本 梓 (大阪教育大学)

Noriaki Mitsunaga and Azusa Sugimoto (Osaka Kyoiku University)

mitunaga@cc.osaka-kyoiku.ac.jp

概要

本論文では ROS (Robot Operating System) を使って移動ロボットを実現する際に必要となる ROS パッケージと通信するための、ハードウェアを制御するマイコン用ライブラリを実装したので報告する。Yujin Robot 社の移動ロボット Kobuki の通信プロトコルを実装しているため、ROS の Kobuki 用パッケージが少しの改変で利用できる。実装したライブラリを移動台車を制御するマイコンに載せ移動ロボットを製作した。移動ロボットは Kobuki 用メタパッケージを少し改変して利用することで、本学内を約 50m 移動できた。

1 はじめに

最近ではロボットの開発用フレームワークとして ROS (Robot Operating System) [1]が広く使われている。ROS は研究用プラットフォームとして開発されてきたことから、パッケージという単位で機能を分割することができ、研究成果をパッケージとして公開している研究者も多い。また複数のパッケージを組み合わせて複合的な機能を実現したメタパッケージも公開されており、市販のハードウェアプラットフォーム用のメタパッケージの中には基本的な移動ロボットのナビゲーションを実現するものもある。ハードウェアにメタパッケージが対応していれば、基本的な機能の実現についてはメタパッケージに頼り、目的とする問題に素早く取り組むことが出来る。一方で、市販のハードウェアプラットフォームを利用しない場合には、どのようにパッケージを組み合わせればいいのか ROS になじみがないと戸惑う面もある。

そこで、自作のハードウェアについて、既存のハードウェアプラットフォーム用メタパッケージを利用できるようにすることを考える。ROS のパッケージは Ubuntu 等の PC 上の OS で動作し、ハードウェア側のアクチュエータやセンサとは何らかの通信を行うことが一般的である。この通信を模擬できれば、既存のハードウェアア

ラットホーム用メタパッケージが利用できる。

本論文では、Yujin Robot 社の移動ロボット Kobuki [2]用のメタパッケージが利用できるようにすることとし、その通信プロトコルを模倣するライブラリを実装する。そして実装したライブラリを用いたマイコンボードでモータ制御等をする移動ロボットを製作し、表らの著書 [3]内で紹介されている Kobuki 用のメタパッケージを利用して、本学内で試走させる。以下、まず Kobuki の通信プロトコルを説明し、ライブラリと移動ロボットの実装方法を述べる。そして、必要になった Kobuki 用パッケージの改変と試走結果を述べる。

2 Kobuki の通信プロトコル

Kobuki と ROS のパッケージ (kobuki_driver パッケージ, kobuki_core メタパッケージに含まれる) 間の通信プロトコルは公開 [4]されており、非同期シリアル (115.2kbps, 8bit, non parity, stop bit 1) で通信する。非同期シリアル上にはパケット (図 1) を載せる。パケットの長さは可変長で、プロトコル上の最短は、サブペイロード数が 1, サブペイロードの長さが 3 のときで 7 バイトである。最長はペイロードの長さが 255 バイトのときで 259 バイトである。

サブペイロードには ROS から Kobuki へ送るコマンドパケット (最大 14 種類で、現在規定されているのは 7 種類, 表 1) と Kobuki から ROS へ送るフィードバックパケット (最大 21 種類で、現在規定されているのは 11 種類, 表 2) がある。Kobuki は ID が 1,3,4,5,6,13,16 のフィードバックパケットを ROS からの要求がなくても 20ms 周期 (50Hz) で送信を続ける。ほかのフィードバックパケットは ROS からの要求があるときにのみ返す。

ROS から Kobuki へ送られるコマンドパケットの ID 1 である Base Control には台車の直進速度と回転速度が含まれ単位は mm/s である。したがってマイコン側で、モータの回転数 (エンコーダパルス数) に変換する必要がある。一方で Kobuki から ROS へ送るフィード

Headers		Length	Payload					Checksum
Header 0	Header 1		Sub-Payload 0	Sub-Payload 1	Sub-Payload 2	...	Sub-Payload N-1	

図 1: Kobuki と ROS のパッケージ間のパケット ([4] より引用)

表 1: ROS から Kobuki へ送るコマンドパケット

ID	Name
1	Base Control
3	Sound
4	Sound Sequence
9	Request Extara
12	General Puropse Output
13	Set Controller Gain
14	Get Controller Gain

表 2: Kobuki から ROS へ送るフィードバックパケット

ID	Name
1	Basic Sensor Data
3	Docking IR
4	Inertial Sensor
5	Cliff
6	Current
10	Hardware Version
11	Firmware Version
13	Raw data of 3-axis gyro
16	General Purpose Input
19	Unique Device Identifier (UDID)
21	Controller Info (PID gain)

バックパケットの ID 1 には左右のモータのエンコーダの値 (0~65535) が含まれる。エンコーダの値から台車の移動量への変換は ROS のパッケージ `kobuki_driver` で行っているため、台車に合わせて `kobuki_driver` 内のパラメータを改変する必要がある。

3 ライブラリの実装

ライブラリは Arduino IDE で利用するものとして C++ のクラスとして実装する。ライブラリは主に次のクラス内関数で構成されている。シリアルポートから受け取ったバイト列をパケットとして構築する関数 `recv()`、パケット内のコマンドパケット (サブペイロード) を解釈

```
while (Serial1.available() > 0) {
  char c = Serial1.read();

  if (fbRecv.recv(c)) {
    // パケットを処理する
    fbRecv.handle();

    if (fbRecv.isSet(fbRecv.flagSetPID)) {
      // PID ゲインの更新
      kPIDs = fbRecv.kPIDs;
      fbRecv.clearFlag(fbRecv.flagSetPID);
    }

    if (fbRecv.isSet(fbRecv.flagUpdateVel)) {
      // 目標速度の更新
      setWheel(fbRecv.spd, fbRecv.radius);
      fbRecv.clearFlag(fbRecv.flagUpdateVel);
    }

    fbRecv.reset();
    break;
  }
}
```

図 2: パケットを受信するプログラム例

してクラス内変数に代入する関数 `handle()` と、シリアルポートへ送るフィードバックパケットをサブペイロードとして詰め込む関数 `addBasicSensor()`, `addDocikngIR()`, `addInertialSensor()` 等とチェックサム計算用関数 `setCheckSum()` である。ただし `handle()` 関数でコマンドパケットのうち ID 3, 4, 12 を無視している。

コマンドパケットを受信し処理する例を図 2 に示す。シリアルポートから 1 バイトずつ受信し、ステートマシンで構成された `recv()` 関数に渡す。 `recv()` 関数の戻り値が真であればパケットを受信しているので `handle` 関数で処理している。

次にフィードバックパケットを送信する例を図 3 に示す。まず Basic Sensor Data パケットを組み立て、その後フィードバックパケットをリセットしてから順に ROS へ送信するサブペイロードとして詰め込んでいく。最後に `setCheckSum()` 関数でチェックサムを計算・追加し、シリアルポートからパケットを送信する。

```

bs.t = millis();
bs.leftEncoder = encL.read();
bs.rightEncoder = -encR.read();

// パケットの組み立て
fbPacket.reset();
fbPacket.addBasicSensor(bs);
fbPacket.addDockingIR(0, 0, 0);
fbPacket.addInertialSensor(0, 0);
fbPacket.addCliff(0, 0, 0);
fbPacket.addCurrent(0, 0);
fbPacket.addRawGyro2(gyroID, 0, 0, 0, 0, 0, 0);
    // Gyro のタイムスタンプをダミーで更新
gyroID += 2;
fbPacket.addGeneralPurposeInput(0, 0, 0, 0, 0);

// PC から要求されているパケットがあれば追加
if(fbRecv.isSet(fbRecv.flagHardwareVersion)){
    fbPacket.addHardwareVersion(1, 1, 0);
    fbRecv.clearFlag(fbRecv.flagHardwareVersion);
}
if(fbRecv.isSet(fbRecv.flagFirmwareVersion)){
    fbPacket.addFirmwareVersion(1, 2, 9);
    fbRecv.clearFlag(fbRecv.flagFirmwareVersion);
}
if (fbRecv.isSet(fbRecv.flagUDID)) {
    fbPacket.addUDID(1, 2, 3);
    fbRecv.clearFlag(fbRecv.flagUDID);
}
if (fbRecv.isSet(fbRecv.flagPID)) {
    fbPacket.addControllerInfo(1, kPIDs.Kp,
                               kPIDs.Ki, kPIDs.Kd);
    fbRecv.clearFlag(fbRecv.flagUDID);
}
// チェックサムの追加
fbPacket.setChecksum();

// パケットの送信
Serial1.write(fbPacket.packet,
              fbPacket.len());

```

図 3: フィードバックパケットを送信するプログラム例

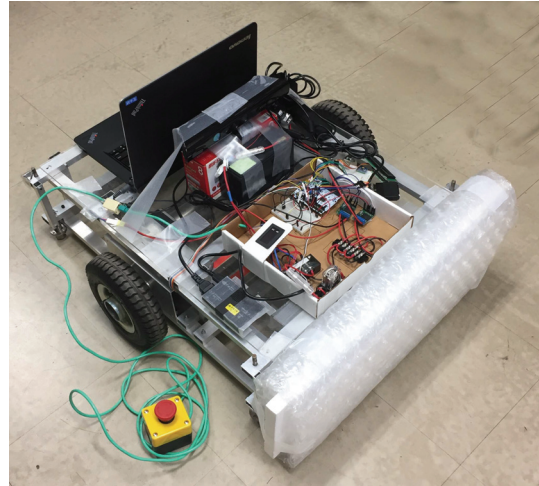


図 4: PC や Kinect を載せた移動ロボット

4 移動ロボットの実装

メカニカルな部分 (モータ・エンコーダを含む) は著者等が以前に製作したロボット [5]の台車部分を利用している (図 4). Kobuki との差異を表 3 に示す. これらの差異を制御マイコンもしくは, ROS のメタパッケージで吸収する必要がある. 台車には ROS を載せた PC (Lenovo ThinkPad E420, Ubuntu 14.04 LTS, ROS Indigo), Microsoft Kinect, モータと Kinect, PC 駆動用バッテリー (12V, 7.2Ah), 12V から AC100V をつくるインバータ, モータ制御用マイコンなどが載っている.

モータ制御用マイコンには STmicro の Nucleo-F103RBT6 マイコンボード (MCU: Coretex-M3, max 72MHz, ROM 128kB, RAM 20kB) を選択した. 理由はハードウェアエンコーダを内蔵しており, 3.3V 動作でありながら 5V トレラントピンが豊富であり, Arduino IDE で開発できて十分な速度で動作するからである. 配線図を図 5 と図 6 に示す. 配線図にはないがマイコンボード上の USB シリアルでは 115.2.kbps での連続通信時に不具合が生じたので FTDI の FT232 ベースの USB シリアル変換ケーブルを利用している. また ROS 非利用時にリモートコントロールできるようにヴィストンのゲームパッド VS-C1 用受信機を載せているが, 衝突検出スイッチやジャイロなどは搭載していない. そのため, センサ類はダミーのデータを返す. マイコン上ではモータの PID 制御 (制御周期 50ms), Kobuki の通信処理 (周期 20ms) と VS-C1 の受信をすべてメインループ内に記述している. VS-C1 の受信にはライブラリ [6] を利用している.

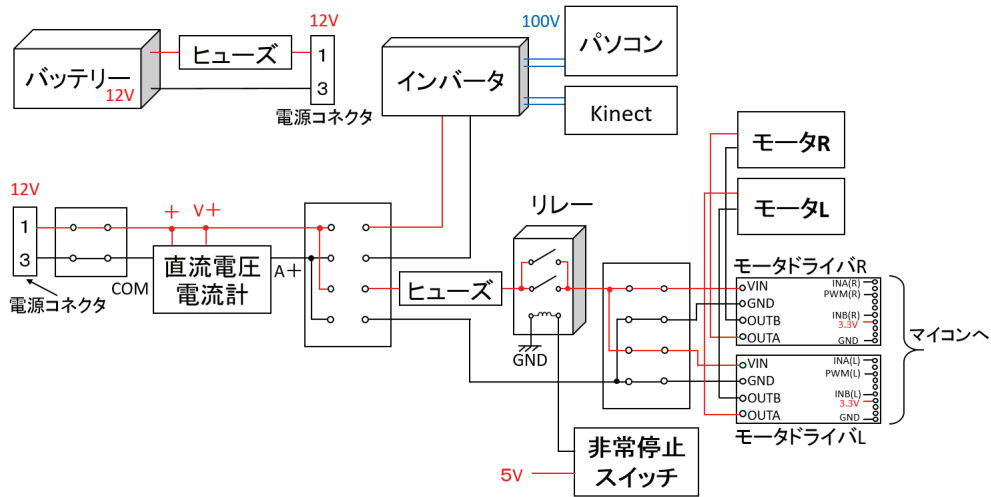


図 5: 電源とモータドライバ周りの配線図

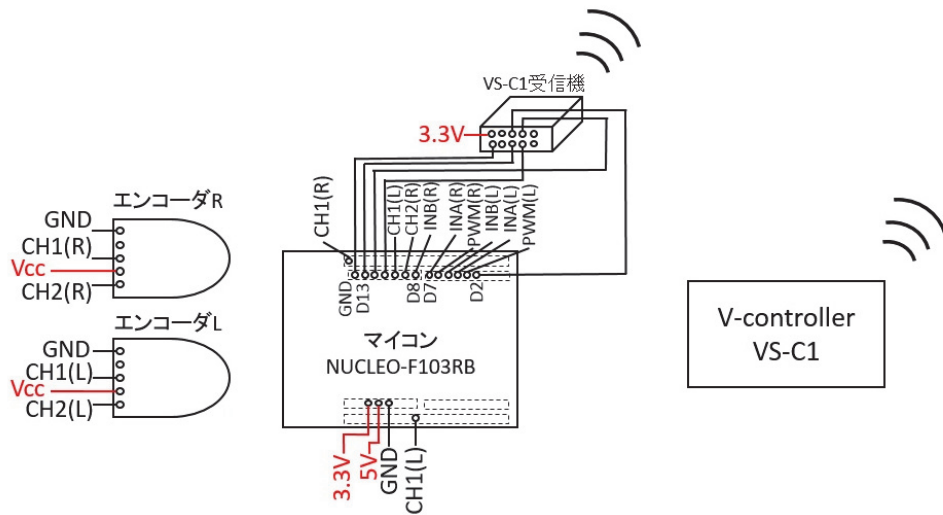


図 6: エンコーダとマイコン周りの配線図

5 Kobuki用パッケージの改変

製作した移動ロボットを屋内でナビゲーションさせるために表らの著書 [3]にある `kobuki_navigation` メタパッケージを利用することにした。これは公開されている git リポジトリ [7]に含まれている。このメタパッケージのほかに `kobuki_node` メタパッケージ (`kobuki_driver` パッケージ含む)が必要である。また `kobuki_navigation` では Kobuki に北陽電機の LRF である UTM-30LX を載せることを前提としている。

表 3 の差異を吸収し LRF ではなく Microsoft Kinect を利用するために次の改変を必要とした。ロボットの形状と大きさについては形状は円形のままとしたが、経路計画時のパラメータを決める `costmap_common_params.yaml` にあるロボットの半

径 (`robot_radius`) と衝突防止のための半径 (`inflation_radius`) を変更した。駆動輪直径、間隔、エンコーダ係数については `kobuki_driver` パッケージ内の `diff_driver.cpp` に記述があるので変更した。

Kinect を利用するためには、ロボット上での Kinect の位置を定義する `tf` を発行する必要がある。そのため `kobuki_navigation` メタパッケージに含まれる `kobuki_tf` パッケージ内の `tf_broadcaster.cpp` に追加した。また Kinect で得られるのは 3次元距離画像 (point cloud とよばれる点群画像) で、ナビゲーションに必要とされる LRF 等で得られる 2次元距離画像ではない。そこで `depthimage_to_laserscan` パッケージ [8]を利用し変換する。

ほかにシリアル通信部分 (パケットの受信部分) に特定のタイミングで顕在化する不具合があり `kobuki_driver` パッケージの `packet_finder.cpp` を修正する必要があった。

表 3: Kobuki と製作したロボットの主要な差異

	Kobuki	製作したロボット
形状	円形	矩形
大きさ [mm]	直径 352	幅 600 × 全長 700
駆動輪直径 [mm]	70	210
駆動輪間隔 [mm]	230	640
エンコーダ係数	0.0024369...	0.00026329...
ジャイロ バンパーセンサ 脱輪センサ クリフセンサ 発音など	あり	なし

6 本学内での試走

製作した移動ロボットを図 7 の地図上の A 点と B 点間を試走させる。場所は本学内の C4 棟 2F の屋内である。A 点と B 点間の距離は約 50m である。ナビゲーション用の地図は環境を実測したものであり、図 7 はナビゲーション用地図に部屋名等を記入したものである。環境中に移動ロボットが乗り越えにくい段差があったため、そこに簡易スロープをとりつけた。またローカルプランナがグローバルプランナの計画した経路を追従するパラメータ等を事前に調整した。

A 点から B 点 (往路) と、B 点から A 点 (復路) のナビゲーションを各 2 回試した。その結果、往路は 2 回ともナビゲーションができたが、復路は 1 回のみであった。その原因は点 A のある廊下の幅がロボットの幅に対して狭く、廊下への進入角度が悪いため壁に近づきすぎてしまったためである。これはロボットの大きさを変えるか、ナビゲーションに使っているローカルプランナを改変する必要があると考える。一方で、通信ライブラリを含めロボットの動作に大きな不具合はみられなかった。

7 まとめ

本論文では既存のロボットプラットフォーム Kobuki 用の ROS パッケージが利用出来る移動ロボットを実現するためのライブラリを実装を提案し、それを利用した移動ロボットがナビゲーションできることを示した。本稿が ROS パッケージを利用した移動ロボットの製作の参考になれば幸いである。ライブラリについては github 等で公開していきたいと考えている。

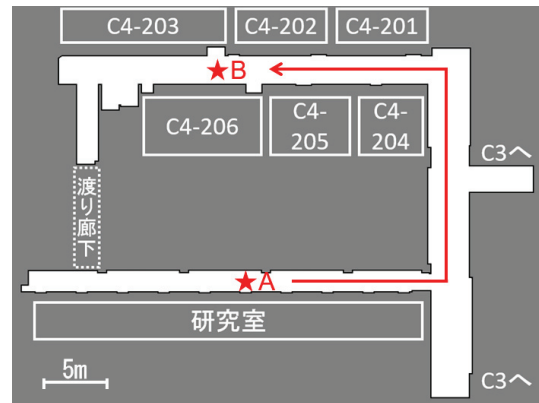


図 7: 試走した環境の地図。ナビゲーション用に用意したものに部屋名等を記載している。C4 は試走環境の棟を C3 は隣の棟を表す。

参考文献

- [1] ROS: <http://wiki.ros.org/> (2017/3/1 閲覧)
- [2] Yujin robot: iClebo Kobuki, <http://kobuki.yujinrobot.com/about2/> (2017/3/1 閲覧)
- [3] 表 允哲, 倉爪 亮, 渡邊 裕太: 詳説 ROS ロボットプログラミング. Kurazume laboratory, 2015. (<http://irvs.github.io/rosbook-jp/>)(2017/3/1 閲覧)
- [4] D. Stonier, Y. Ju, and J. S. Simon: kobuki_driver documentation, Appendix: Protocol specification, (<https://yujinrobot.github.io/kobuki/doxygen/enAppendixProtocolSpecification.html>) (2017/3/1 閲覧)
- [5] 光永法明, 江崎大樹, 伊藤直也, 按田翔悟: 人を乗せて案内をするロボットの提案と試作. ロボティクス・メカトロニクス講演会'10, 2P1-E30, 2010.
- [6] Bill Porter: PlayStation 2 Controller Arduino Library v1.0, <http://www.billporter.info/2010/06/05/playstation-2-controller-arduino-library-v1-0/> (2017/3/1 閲覧)
- [7] rosbook_kobuki の git リポジトリ: https://github.com/irvs/rosbook_kobuki.git (2017/3/1 閲覧)
- [8] depthimage_to_laserscan パッケージ: http://wiki.ros.org/depthimage_to_laserscan (2017/3/1 閲覧)

© 2017 Special Interest Group on AI Challenges
Japanese Society for Artificial Intelligence
一般社団法人 人工知能学会 AI チャレンジ研究会

〒162 東京都新宿区津久戸町4-7 OSビル402号室 03-5261-3401 Fax: 03-5261-3402

(本研究会についてのお問い合わせは下記にお願いします.)

AI チャレンジ研究会

主査

光永 法明

大阪教育大学 教員養成課程 技術教育講座

主幹事

中臺 一博

(株)ホンダ・リサーチ・インスティテュート・
ジャパン / 東京工業大学 大学院
情報理工学研究科

担当幹事

植村 渉

龍谷大学 理工学部 電子情報学科

幹事

公文 誠

熊本大学 大学院 自然科学研究科

幹事

中村 圭佑

(株)ホンダ・リサーチ・インスティテュート・
ジャパン

Executive Committee

Chair

Noriaki Mitsunaga

Department of Technology Education
Osaka Kyoiku University
mitsunaga @ cc.osaka-kyoiku.ac.jp

Secretary

Kazuhiro Nakadai

Honda Research Institute Japan Co., Ltd.
/ Graduate School of Information
Science and Engineering,
Tokyo Institute of Technology

Wataru Uemura

Department of Electronics and Informat-
ics, Faculty of Science and Technology,
Ryukoku University

Makoto Kumon

Graduate School of Science and
Technology,
Kumamoto University

Keisuke Nakamura

Honda Research Institute Japan Co., Ltd.

SIG-AI-Challenges web page; <http://www.osaka-kyoiku.ac.jp/~challeng/>