

USARSim において自律ロボット エージェントの開発を支援する USADLib の開発

Development of USADLib to support in developing autonomous robot agent on USARSim

佐藤健二・川合徹・黒岩丈介・小高知宏・小倉久和

Kenji SATO , Toru KAWAI , Jousuke Kuroiwa , Tomohiro Odaka , Hisakazu Ogura

福井大学

University of Fukui

ksatou@i.his.fukui-u.ac.jp

Abstract

In this paper, we develop the USADLib(USARSim Agent Development Library) which supports us in developing an autonomous robot agent and performing the simulation of it on USARSim . Since USARSim provides the physical model of various robots , USARSim is effective tool to simulate them . However , USARSim does not prepare tools in developing autonomous robot agents.The USADLib provides the tools;communication tools between the agent and the simulation server based on TCP/IP,method family in programing action rules of the agent and a debug interface.We can develop the autonomous robot agent even though one defined action rule,and performed the simulation by means of USADLib.We conclude that it is possible to simulate various autonomous robots on USARSim.

1 はじめに

RobocupRescue・VirtualRobots リーグでは、ロボットシミュレーションによる競技を行うために USARSim[J.Wang,S.Balakirsky, 2007] [W.C.VUI, 2006] が利用されている。USARSim は、UnrealEngine を利用しロボットやその周辺環境のシミュレーションを行うことができるシミュレーターである。また、USARSim では、UnrealEditor で作成したロボットが利用できるため、様々なロボットのシミュレーションが可能である。しかし、USARSim では、自律動作を行うロボットエージェントを開発するためのツールが十分に提供されておらず、技術力の乏しい初心者が自分の構想したロボットの自律動作を実装し、シミュレーションを行うのは容易ではない。そこで、本研究では、USARSim のロボットエージェント開発を支援する USADLib(USARSim Agent De-

velopment Library) を開発する。

USARSim でロボットエージェントを開発するには以下のような知識が必要とされる。

- USARSim の仕組み
- Server/Client 構成
- TCP/IP プロトコル
- ネットワークプログラミング
- 行動の制御
- 位置情報の処理

そこで、上に挙げたような知識がなくとも、USADLib を用いることによって初心者が容易にロボットエージェントを開発できるようにする。先行するエージェント開発の研究として、FRS-Lib [林, 下羅, 2004] や 4-D/RCS エージェントモデル [J.S.Albus, 2002a] があり、これらを参考に USADLib を開発する。

2 USARSim の概要

USARSim は、3D アクションゲームであるアンリアルトーナメント 2004(以下 UT2004) 及びそれに搭載されている UnrealEngie を利用したロボットシミュレーターである。USARSim では、UT2004 で利用できる機能に加えて VirtualRobots リーグのための様々な要素が追加されている。第一に、AIBO や P2AT といった様々なロボットのシミュレーションモデルである。第二に、実機リーグで使用される被災者のシミュレーションモデルである。第三に、NIST(the National Institute of Standards) によって規格化された、災害現場の様々な特徴を含んだマップである。USARSim は、これら追加された要素を用いてロボットシミュレーションを行う。

USARSim は、Controller とシミュレーションサー

バーから構成されている (Figure.1) . Controller は、ユーザーがロボットを操作するためのインターフェイスである . シミュレーションサーバーは、シミュレーター本体であり、仮想上でのロボットやマップの実体を持つ . ユーザーは、Controller を介してシミュレーションサーバーとインタラクティブに通信しロボットを操作する . シミュレーションサーバーとの通信は、TCP/IP プロトコルに基づく .

USARSim でロボットを動作させるには、シミュレーションサーバーに行動コマンドを送る必要がある . シミュレーションサーバーは、このコマンド情報を用いてロボットの物理モデルに従って動作をシミュレートし、その結果を Controller に返す . また、シミュレーションサーバーからはマップやロボットの情報が送られてくる . この情報は、ロボットに搭載されているセンサーの反応やマップから取得できるロボットの位置情報である .

USER は、これらのロボット情報を元に USER インターフェイスを操作する . この操作により、Controller から行動コマンドがシミュレーションサーバーに送られロボットが動作する .

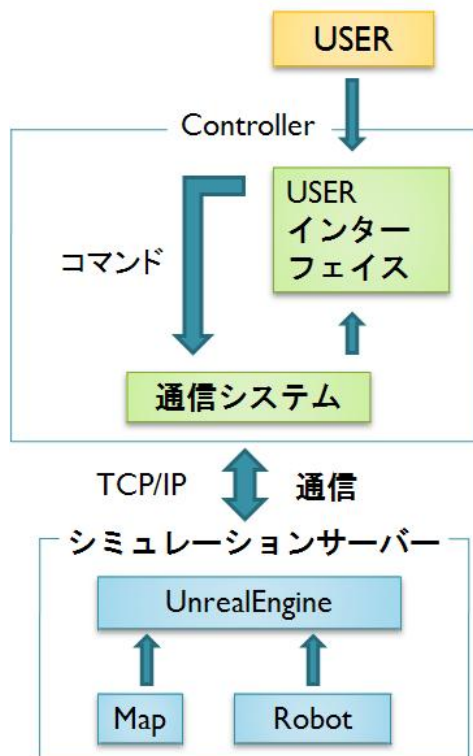


Figure. 1 USARSim の構成

3 USADLib の開発

3.1 USADLib の設計

USARSim においてロボットエージェント開発を行う USADLib を設計する . USARSim でロボットを自分の思い通りに動作させるためには、2章で述べたように、行動コマンドを作成し、それをシミュレーションサーバーに送信するプログラムを作成する必要がある . USADLib では、ロボットを動作させるために必要となる様々な機能を Java のクラスとして提供する . また、それぞれのクラスには、Figure.2 のようなつながりを持たせ、USADLib がロボットエージェントの一部として機能するようにする . 行動コマンド出力ルールは、行動コマンドを生成するメソッドを組み合わせで実装する . また、その他の機能についてもメソッドを組み合わせで実装できるようにする . ユーザーは、ネットワークプログラミングや USARSim の仕組みを知らなくともロボットエージェントを開発できる .

USADLib の機能は役割ごとに分けて Table1 のように実装する . 通信部では、応答文解析と TCP/IP に基づく通信機能を提供する . コマンド部では、エージェントの行動ルールを作成するためのコマンドメソッドを提供する . また、USARSim で提供されるロボットに対応したロボットモデルやエージェントが期待通りの動作をしているか確かめるためのデバッグインターフェイスを提供する .

3.2 USADLib の実装

3.2.1 通信部

シミュレーションサーバーと通信を行う CommunicationInterface クラスを実装した . CommunicationIn-

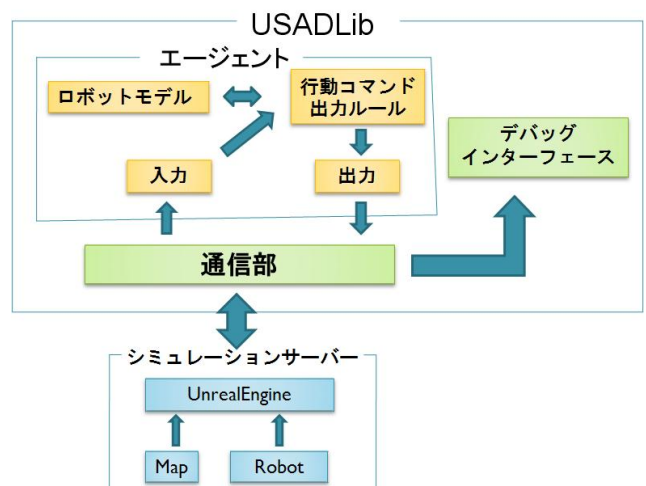


Figure. 2 USADLib の構成

terface クラスは、通信を開くための open メソッド、行動コマンドを送るための send メソッド、応答文を受信するための getMessage メソッド、通信を閉じるための close メソッドを持つ。これら 4 つのメソッドの実装には、JavaAPI の Socket クラスを利用した。行動コマンドや応答文のやりとりには、文字列ストリームを用い、ASCII コードのビット列に変換して通信を行うようにした。これは、UnrealEngine が通信を ASCII コードで行うものと定めているからである。

CommunicationInterface クラスを用いることによってシミュレーションサーバーとの通信が可能となった。しかし、応答文はセンサーごとに形式が異なり、センサーの出力値以外にデバイスのタイプ、センサー名などの情報が記されている。そのため、応答文をセンサーの出力値として演算に利用することができない。

そこで、応答文を解析し目的のセンサー出力値を抽出する MessageAnalyze クラスを実装した。MessageAnalyze クラスは、いくつかの異なる応答文から 1 つの応答文を選び出す get メソッドと、1 つの応答文から数値データのみを抽出する spit メソッドを持つ。シミュレーションサーバーからの応答文は、1 行 (1 応答文) 単位で文字列ストリームから入力される。get メソッドは、引数の文字列を含んだ応答文を返す。split メソッドは、応答文中の数字を切り出し文字列配列として返す。この 2 つのメソッドを用いることでセンサー出力値を利用することができる。

3.2.2 行動コマンド生成部

行動コマンドを生成する CommandMaker クラスを実装した。行動コマンドは、静的な部分と動的な部分から成り立っている。静的な部分は命令や操作すべきアクチュエーターの名前など固定された文字列

である。一方、動的な部分は数値や色などコマンドの引数に当たる部分である。そこで、行動コマンドの静的な部分を文字列で用意し、それをテンプレートとして、引数を行動コマンドに代入する。ユーザーは、行動コマンドの形式に関係なく、メソッドを実行するだけでロボットを動作させることができる。

CommandMaker クラスのメソッドは、基本的に同じ構造をしている。このクラスのメソッドは、動的な部分を引数の変数として与えられ、それを静的な部分に代入しコマンド文を生成する。そして、生成した行動コマンドを CommunicationInterface クラスの send メソッドに引数として与えて実行する。これにより、行動コマンドをサーバーに送信する。以下のようにメソッドを記述すると

```
com.skidDrive(-1,2);
```

次のようなコマンドがシミュレーションサーバーに送られロボットを動作させる。このコマンドは、車輪移動ロボットの車輪を動作させるコマンドで、車輪スピードを左は $-1(\text{radian}/\text{sec})$ に、右は $2(\text{radian}/\text{sec})$ にするというコマンドである。

```
DRIVE{Left -1}{Right 2}
```

3.2.3 ロボットモデル

本研究では、USADLib の開発に利用した P2AT について、ロボットモデルを実装した。ロボットモデルは、シミュレーションサーバーにあるロボットの実体に対応する、エージェント側でのロボット認識である。ロボットエージェントがロボットの動作を決定するとき、ロボットがどのような状態であるか調べ適切な動作を判断する必要がある。本研究で実装した P2ATModel クラスでは、センサー情報、ロボットの位置や周辺の障害物の位置、車輪のスピード、被災者の有無が記録される。また、P2ATModel クラスでは、シミュレーションサーバーから情報を取得しセンサーデータを最新の状態に更新できる sensorUpdate メソッドを持つ。sensorUpdate メソッドは、MessageAnalyze クラスの get メソッドと split メソッドを利用して、各センサーデータをロボットモデルのメンバ配列に格納する。メンバ配列は、sonar, rangeScanner, groundTruth, victim といった P2AT に搭載されているセンサーごとに定義されている。また、RangeScanner のセンサーデータについては、特別な処理をしている。RangeScanner の応答文では左から時計回り方向のセンサー出力値が順に記述されているので、

Table 1 USADLib のクラス

通信部	
通信クラス	CommunicationInterface
応答文解析クラス	MessageAnalyze
行動コマンド生成部	
行動コマンド生成クラス	CommandMaker
行動制御クラス	Act
デバッグインターフェイス	
GUI クラス	DebugInterface
ロボットモデル	
P2AT クラス	P2ATModel

split メソッドをもちいて分割すると配列番号と角度が対応しなくなってしまう．そこで、配列の最後尾から順番に、先頭から配列に並べ直す処理をしている．ロボットの初期条件は、location, rotation, robotID というメンバー配列に記録される．また、ロボットの現在の車輪速度は、メンバー配列 speed に格納する．

3.2.4 デバッグインターフェイス

物体を検知するためのレンジスキャナーとソナーについて反応の視覚化を行った．USARSim ではシミュレーション結果をアンリアルトーナメント 2004 の GUI で見ることができる．しかし、ロボットの動作を見ることができてセンサーの出力値は表示されない．そこで、プログラミングした行動ルールのデバッグを補助するためにセンサーの反応を視覚化する．レンジスキャナーは、レーザー光線を用いて物体との距離を測るセンサーである．レンジスキャナーの応答文は、USARSim と通信することによって、次のように得ることができる．

```
SEN {Time 20.15} {Type RangeScanner} {Name Scanner1} {Resolution 0.0174} {FOV 3.1415} {Range 20.0000,19.9966,19.9803,19.9927,19.9976,.....}
```

応答文の Range 以降に羅列されている数値がレンジスキャナーの出力値である．これらの数値を用いてレンジスキャナーの反応を平面図として表示させる．ロボットを中心とした相対的な座標系を考え、角度と距離から物体の位置座標を計算する．そして、物体の位置座標を直線で結び壁や障害物の概形を図 3-b のように表示させた．Figure.3 は、シミュレーション結果 (a) とレンジスキャナーの反応を視覚化したもの (b) である．Figure.3-b の黒点はロボットの位置である．目盛はロボットの視野を表し、0 度方向がロボットの正面である．赤い線は障害物の反応を表し、円を付けた場所が Figure.3-a の障害物の反応であると考え

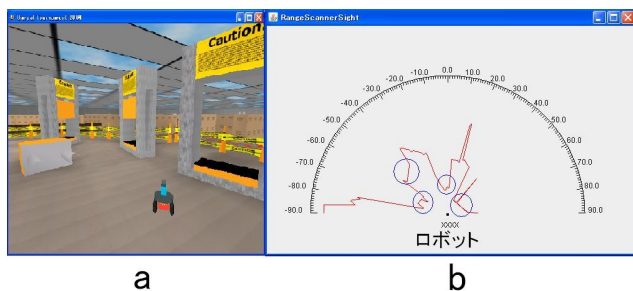


Figure. 3 レンジスキャナーの視覚化

えられる．

同様にソナーの反応についてもインターフェイスを作成し視覚化した．ソナーセンサーの出力値は、レンジスキャナーと同様に、USARSim と通信して応答文から取得することができる．Figure.4 は、シミュレーション結果 (a) とソナーの反応を視覚化したもの (b) である．Figure.4-b の中心にある黒点は、ロボットの位置である．目盛はロボットの視野を表し、0 度方向がロボットの正面である．P2AT にはソナーが 16 個取り付けられている．赤い線は、ソナーが向けられている方向にある物体までの距離を表す．物体の反応がない場合は線がプロットされない．そのため、常に 16 本線がプロットされるわけではない．

さらに、ロボットに搭載されたレンジスキャナーを用いて、地図作成システムを実装した．地図作成システムでは、精密な計測を可能にするためにレンジスキャナーを使用した．物体を検知するセンサーは、レンジスキャナーとソナーの 2 種類がある．レンジスキャナーの視野角は前方 180° で、ソナーの視野角の半分しかない．しかし、センサーの解像度に

当たる走査角度は、1° 刻みでソナーの 20 倍であることから、レンジスキャナーの方が高い解像度を得ることができる．

レンジセンサーの反応値は、角度と距離から物体の

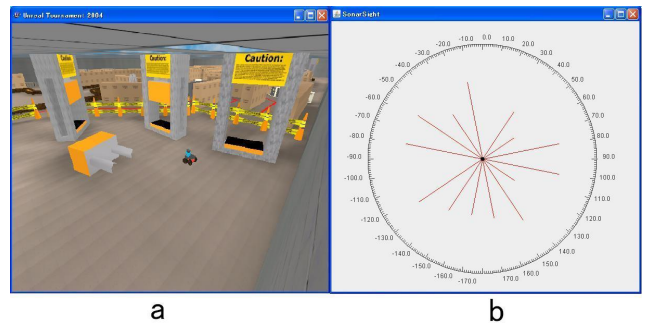


Figure. 4 ソナーの視覚化



Figure. 5 地図描画

位置を把握するため，地図作成には極座標を利用する．地図作成で問題となるのはセンサーを搭載したロボットが常に移動し向きを変えることである．そこで，地図に点をプロットする際には，地図座標系にロボットのセンサー座標系を適合させる．

Figure.5-b は，Figure.5-a の地形から作成した地図である．赤い線が壁や障害物を表す．Figure.5-b では S 字に曲った角が見てとれ，しっかりと地形の特徴を反映していることがわかる．

4 考察

USADLib は，USARSim の仕組みやネットワークプログラミングの知識がない初心者でも，USARSim においてロボットエージェントを開発することを可能にした．レスキューロボットのシミュレーションと聞いて，Server/Client 構成やネットワークプログラミングといったものを連想することはまずない．そのため，USARSim におけるロボットエージェントの開発は，初心者にとって敷居が高いと言わざるを得ない．USADLib を用いれば，こういった知識がなくとも，ある程度のコツと USADLib の使い方を身に付ければ，ロボットエージェントを開発できるようになる．USADLib の利用者は，戦略的行動といった本質的な問題に取り組むことができる．また，すぐに本質的な問題に取り組むことができることによって開発者自身のモチベーションが向上すると考えられる．

5 まとめと今後の展望

USARSim においてロボットエージェント開発を支援するための USADLib を開発した．USADLib の利用者は，USARSim の仕組みやネットワークプログラミングを意識せずにロボットエージェントの開発を行えるようになった．

今後の展望として，本研究では触れなかった二足歩行ロボットのエージェントを開発できるように USADLib を拡張する．

参考文献

- [J.S.Albus, 2002a] J.S.Albus.: 4-D/RCS A Reference Model Architecture for Intelligent aUnmanned Ground Vehicle, Proc.SPIE, Vol.4715,303,2002
- [田所,高橋, 2002b] 田所 諭,高橋 友一: ロボカップレスキュー,日本ファジィ学会誌 Vol.14,No.6,pp.593-601, 2002
- [林,下羅, 2004] 林 央幸,下羅 弘樹,森下 卓哉,石橋 佳征,白井 治彦,高橋 勇,黒岩 丈介,小高 知宏,小倉 久和: RoboCup

サッカーシミュレーションにおけるプレイヤーエージェントの設計 -Fukui RoboCup Soccer Library(FRS-Lib) とそれを利用したエージェント構成法-, Mem.Fac.Eng.Univ.Fukui, Vol.52, No.1, March 2004

[W.C.VUI, 2006] W.C.VUI.: レスキューロボット開発支援用物理シミュレーションシステムの開発,長岡技術科学大学大学院修士課程工学研究科修士論文, 2006

[J.Wang,S.Balakis, 2007] J.Wang, S.Balakis.: USARSimV3.1.1 A Game-based Simulation of mobile robots, <http://sourceforge.net>, 2007