

Unity を用いた Windows 環境下でのロボット制御ソフトウェア開発

新居 徹哉*

Robot control software development under Windows using Unity

Tetsuya Nii*

要旨: 近年の少子高齢化と労働人口の減少から、自動化のためのロボット導入はますます進んでいる。しかし、ロボットの導入をより広げるためにはロボットのプログラム製作環境を安価で使いやすいものとし、またプログラマを増やす必要がある。本研究室では、自作装置やロボット、教育用ロボットの制御プログラム開発プラットフォームとしてゲーム開発エンジン Unity を採用している。Unity を採用することにより、低コストかつ専門的な知識を必要せず、また、多くの人材のいるパソコン・ゲームプログラマも、使い慣れた言語・環境でロボットプログラムに移行できる可能性があり、これからの自動化支援に有用と考える。当研究室での開発事例について紹介する。

キーワード: Unity ロボット アカデミックスカラロボット 制御

1. はじめに

近年、日本のみならず先進諸国において少子化・高齢化に伴う労働人口の減少から生産現場には多くの自動機械が導入され、省力化・無人化が進行してきた。現在、さらなる労働力減少・働き方の見直し・労働市場での労働力の取り合いがますます進行し、自動化は生産現場だけの対策ではなくなっており、社会の幅広い分野で自動化の要求が高まっている。

しかし、食品・店舗・サービスなどの非生産分野における作業では、生産現場のように整った環境で機械を制限なく動かすことができる環境は稀であり、人の代替を求められることが多い。

そのような環境での自動化においては、元々人間の代替作業を目的として開発されたロボットが活躍できる可能性が高い。実際、すでに多くの業種・事業所においてロボットの導入が始まっており、展示会などでは様々な作業をこなすロボットが紹介され、大手チェーン店や空港などの大規模施設においてロボットが稼働しているシーンを見かけるようになってきている。

しかし、実際に市中の店舗などにロボットが導入されているかと言えば、そのような状況にはない。一般にロボットは定型作業については一度プログラム製作・設置を行ってしまえば微調整程度で使い続けられ、費用も保守費程度しかかからない。よって定型作業を置換し、長期間にわたって稼働させるこ

とで導入コストを回収する形が理想となる。工場やチェーン店の業務はこれに当たる。

しかし、製造業でも小規模事業所、サービス系・食品系では非定型作業が多く、また、定型作業についても工数の短い仕事が多い。

ロボットの導入においてはプログラム開発のコストが占める割合が大きい。一般にロボットの導入コスト中では、システム設計・プログラム製作・設置などを担当するシステムインテグレーション費用が半分以上を占める。一例としては、一般的な垂直多関節ロボット1台あたり本体300万円、SI費用520万円などの試算がある。¹⁾

そのような状況では、非定型作業や工数の短い作業を複数、ロボットに実行させようとするれば、プログラム開発にかかるコストは膨らむばかりであり、投資回収が難しいためロボット導入には踏み切れないのが現状である。

しかし、今後もますます人手不足には拍車がかかり、しかも人材の流動化も進むと予想させる中、業務改善・事業継続のためにも、ロボットの導入・自動化はますます必要性が高まると考えられる。

本研究では、小規模かつ簡易的な自動化機械やロボットシステムのシステム設計・導入・プログラミングをパソコンベースのUnityという環境で行うことで、このSIコストを引き下げ、小回りの利くロボットベースの自動化を行えないかと考え、またパソコン・C言語という人材供給量の多い分野からの開発参入につなげることで、さらに広い範囲のロボット活用につなげられると期待し、本校制御機械科学生の卒業研究として取り組んできた内容を紹介する。

* 山形県立産業技術短期大学校庄内校

〒998-0102 山形県酒田市京田三丁目 57-4

e-mail: nii@shonai-cit.ac.jp

* Shonai College of Industry & Technology

3-57-4 Kyoden, Sakata City, Yamagata, 998-0102, Japan

2. Unity とは

Unity とは、Unity Technologies 社が提供するゲーム開発プラットフォームである。ゲームエンジンとしての性質だけでなく、統合開発環境 (IDE) でもある開発ソフトウェア・ツール群である。2D/3D のゲームを Mac OS/Windows/Linux 上で開発可能で、プログラム言語としては、C++言語を用いている。また、クロスプラットフォーム環境であり、上記 OS に加えて Android や iOS 端末、ゲーム機、VR 端末上で実行可能なコード開発を行うことができる。

また、個人、および収益/調達資金が 10 万米ドル以下の企業において無償で開発に利用できるライセンス (Personal ライセンス) があり、しかも基本的なアプリケーション開発機能について利用可能である。

3. Unity の特徴

Unity はゲームプラットフォームである。ゲームというソフトウェアの特徴としては、画面の描画性能などが注目されるが、それ以上に重要になるのが、非同期かつリアルタイムの動作が要求されることである。ゲーム内においては各種物体・キャラクター等が動き回るが、これらの動作は一般的には何らかの内部的なルールや物理法則に従った、同期的な動作である。しかし、ゲームにおいては、それらが人間の操作によってふるまいを変えることである。そして、人間の操作は必ずしも、ゲームの進行に同期的に行われるとは限らない。よって、ゲーム世界中の各種オブジェクトのふるまいは、同期・非同期の混在したシステムとなる。これらを正しく処理するには、非常に複雑な仕組みを用意する必要があるが、Unity においては例えばゲーム中、処理が重くなって追いつかない場合には画面の処理落ちを認めるなど、ソフトリアルタイムシステムであると割り切って、かわりにユーザからはタスク処理を隠蔽し、リアルタイムシステムや非同期システムの知識がないユーザでもそれらの要素を持ったプログラムを実現できるようにしている。

3. Unity の制御システムへの応用

当研究室では、これまでに Unity を 3D グラフィック表示やインタフェースとしてではなく、PC ベー

スの機械・ロボット制御プラットフォームとして使用することを模索してきた。それらの事例について紹介する。

Unity は、初期化動作とループ動作でゲームを記述する。中でも、画面のリフレッシュ動作に合わせたメインループとも呼ばれる Update() メソッドおよび設定した一定時間で起動される FixedUpdate メソッドを設定しておくことで動作 OS のスレッドタイムスライス程度の精度で起動できる。ただし、Windows においてはタイムスライスは数十 ms 程度のため、ミリ秒以下での制御やサンプリング周波数一定の制御には不向きである。

3. Unity のロボット制御への活用

当制御機械科の卒業研究での開発に、表示/制御装置として Unity ベースのアプリケーションを製作した。それらの事例について紹介する。

Unity によるアカデミックスカラロボット制御

アカデミックスカラロボット (以下 ASR) とは、VSTONE 社の販売するロボット教育向けの教材ロボットである。特徴としては、SCARA という水平多関節構造を持ち、一般の垂直多関節ロボットに比べ自由度は下がるものの、3次元空間内の自由な移動と、上下動によるつかみ取り、はめ込み動作が可能である。ロボットのアクチュエータとしてはシリアル通信により制御するロボット用サーボモータを搭載し、USB-シリアル変換ボードを介して USB/I/F に接続・制御が可能である。また、専用ソフトウェアは一般的なロボット制御スキルである、オフライン・オンラインのティーチング動作だけでなく、ロボット用サーボの関節角読み出し機能を用いたダイレクトティーチングにも対応し、幅広いロボット操作の学習が可能となっている。図 1 に ASR の写真と制御用ソフトウェアであるスカラプログラムの画面を示す。



図1 アカデミックスカラロボットとスカラプログラマ
ASR はまた、内部構造・通信プロトコルも公開し、Windows/Linux 用サンプルプログラムも配布されており自作プログラム上から簡単に利用可能である。

当研究室の取り組みとして、ASR と USB カメラを組み合わせたロボットピックアップ装置の開発を行っており、その中で、ロボット制御プログラムの製作環境として Unity を用いている。

Unity 上では、物体の検出・画像処理・ロボットの制御の3つのタスクを処理する必要がある。

今回のシステムでは、物体搬送はベルトコンベアで行っており、その制御はシーケンサで行っている。そのため、物体の検出点への搬送が終わった段階でシーケンサからマイコンを介してシリアル通信で信号を送っている。ここで Unity 側では Windows の .NET framework の serialport クラスを利用する。また、Unity 上で USB カメラは WebCamTexture クラスでアクセス可能であり、取り込んだ画像について画像処理を行い、指示エリア内の物体検出と振り分け判定を行う。仕分け動作については、あらかじめティーチングで作成した ASR の姿勢データをロボット制御用に作ったダイナミックリンクライブラリ (DLL) を介してロボットに姿勢指示を送ることで実現する。また、姿勢の情報は時系列で断続的に送る必要があるが、ここには C#/Unity の持つコルーチンという中断・再開の可能な一種のサブルーチンで、メインルーチンとは独立に継続状況を持つプログラムの記述が可能であり、今回のロボット動作のような一連のシーケンス処理には適している。制御プログラムの概要を図 2 に示す。また、実際にこのシステムを組み込んだ制御機械科制御システム実習で製作したロボットハンドリング装置を組み込んだペットボトルキャップ選別ラインシステムのロボット部²⁾を図 3 に示す。

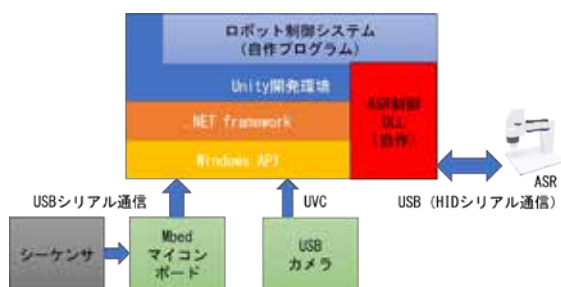


図2 ロボット制御プログラムレイヤー構造図



図3 自動ライン実習装置内のロボット選別部 (ASR)

Unity を用いたロボットティーチングシステム³⁾

産業用ロボットのプログラム作業には、一般的にロボットを実際に動かしてプログラムを行う「オンラインティーチング」とパソコンなどの上でプログラムを作成し、ロボットに転送して利用する「オフラインティーチング」がある。多くの現場では従来、ティーチングペンダントと呼ばれる端末で実際に操縦 (ティーチング) し、その操作を繰り返させる (プレイバック) ことで産業用ロボットを活用してきた。しかし、近年ではロボットによる作業の高度化 (非常に長い曲線の溶接・曲面の塗装など) のため、オンラインティーチングではプログラム困難であり、ロボット制御に CAD/CAM のデータを用いたオフラインティーチングシステムが採用されてきている。

オフラインティーチングの最大のメリットは、複雑な作業をプログラム可能なことと、実際のロボットを動かさずにプログラムできるため、プログラム作成のために作業ラインを止める必要がないことである。さらに、市販のシステムでは、PLM/ERP/BIM などのより高度な生産システムとの連携により、ロボットを用いたライン全般の設計を実現できる Industry4.0 の中核になる工場レベルでのデジタルツインを実現するものも登場してきている。

半面、これら高機能化したシステムではそのコストが問題になり、実際に各社産業用ロボットだけでなく、各種生産機械・装置、さらには作業者も取り込んだシミュレーション上でのロボットプログラムなど、非常に有用、しかし場面によってはオーバースペック・コスト的に導入の難しい場面も考えられる。また、市販ロボット/作業機械を利用するにはデータが用意されているが、独自仕様・一点ものの生産機械、研究ベースの自作ロボットなどについては、データを用意することが難しく対応できない。

今回、卒業研究で取り組んだシステムは、3D CAD で製作したロボットの3DモデルをUnityの3D空間内にインポートし、Unity 上のオブジェクトとして動かす仕組みになっている。

Unity を用いるメリットとして、3DCAD のデータをそのまま利用できる点がある。CAD の独自データや IGES, STEP, parasolid 形式などの CAD 形式のデータは取り込みできないが、OBJ 形式などの広く用いられている 3D データ形式に対応しているため、変換ツールや CAD のデータエクスポート機能を利用することで対応可能である。今回のケースでは、Solidworks2018 のデータを利用し、Fusion360 のデータ変換機能を利用して OBJ 形式に変換、モデルデ

一夕の取り込みを行った。

・モデル上でのオフラインティーチング

Unity を用いて、ティーチングアプリを製作した。Unity にインポートしたモデルの各関節を、作業者の指示で動かしロボットの姿勢を定め、その姿勢を ASR の制御ソフトであるスカラプログラムの形式で保存することで、オフラインでのティーチングを実現する。Unity を用いるメリットとして、入力デバイスとしてゲームコントローラが容易に使えることから、旧来のティーチングペンダントの操作にとらわれず、ゲーム感覚で操縦できるシステムとなったことと、3D 空間内で視点を自由に動かしながら操作できることである。

図 4 に関節角度指令モードでロボットを動かしている様子を示す。ゲームコントローラの十字キーで関節を切り替えながら、関節を正/逆回転させたり、ハンドの開閉が行える。また、視点も上からと横からと切り替えることで、作業に適した視点が確保できる。さらに操作中にどの関節を動かしているかわかるように動かされるリンクに赤く着色するよう工夫している。

図 5 に位置制御モードでロボットを操縦している様子を示す。関節角の指定では、空間内の物体をつかみに行くなどの動作は指示しにくい。そのため、モードを切り替えると仮想的なターゲット球を空間中に表示し、この球を X-Y-Z の 3 軸方向にコントローラで移動させると、ロボットは自動的に追従するようにプログラムされている。また、モードの切り替わりが確認できるよう、ロボット本体は青に着色される。座標から関節角度の座標変換の計算を製作し、自動で計算し指令値とするようになっている。

図 6 にこのプログラムからロボット制御用データを出力した様子を示す。画面上のロボット動作中にコントローラのデータ出力ボタンを押すとデータファイルが作成され、また姿勢を変えて出力ボタンを押すことで、次々とその時点の姿勢がファイルに追記される。この仕組みにより、画面上のロボットの動きを実際のロボットの教示データとして出力することができる。

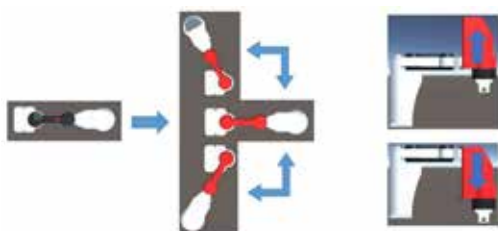


図 4 関節角度指令モードでの制御

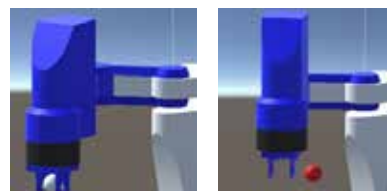


図 5 位置制御モードでの制御

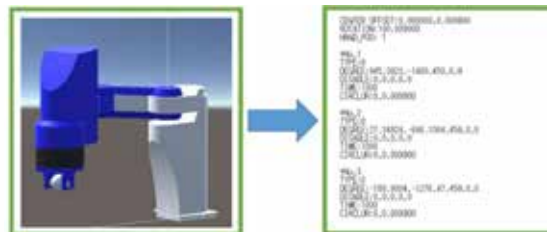


図 6 仮想ロボットを操作し作ったロボット制御データ

4. おわりに

本研究室で取り組んできた、ゲームエンジン Unity 上での制御システム構築について、実際の卒業研究で製作したシステムについて紹介した。Unity はソフトリアルタイムシステムとしては非常に使いやすく、非同期プログラムをそうと意識せず記述できるなど、ロボット制御用として好ましい特徴を持ち、また PC プログラマからも取りつきやすいという優れた特徴を持つ。また、表示機能は特に優れ、3D のみならず VR デバイスを使った表示システムなども簡単に実装可能であり、ロボット制御プログラミングに多視点という価値を付け加える可能性も持っている。今後もこれらの特徴を生かし、自作ロボットなどの制御に活用していきたい。

謝 辞

最後に、今回卒業研究として紹介した内容に取り組んだ新居研究室平成 30 年度卒業生渡部陽介氏、同令和元年度卒業生上野史都氏に感謝いたします。また、アカデミックスカラロボットの 3D モデルについて総合システム実習用に製作したモデルデータをご提供頂いた制御機械科庄司英明教授に感謝いたします。

文 献

- 1) ここが知りたい！ロボット活用の基礎知識，経済産業省，一般社団法人日本ロボット工業会，p5，2016 年
- 2) 渡部陽介：Unity を用いた制御システムの開発，山形県立産業技術短期大学校庄内校，2019 年
- 3) 上野史都：Unity を用いた制御システムの開発，山形県立産業技術短期大学校庄内校，2020 年