



第4章

通信機能の信頼性アップ、リアルタイム性の向上、組み込み対応…機能面で選ぶならコレ

進化中! 次世代バージョン「ROS 2」

高瀬 英希 Hideki Takase

第1章で紹介したように、ROSは新世代のROS 2に移りつつあります。本特集では、すぐに使えるという観点から、主に旧世代のROS(本稿ではROS 2に対してROS 1と呼ぶ)を紹介しています。

ROS 2がなぜ開発され、どのように違うのかを紹介します。ROS 1とROS 2は互換性がないので、ロボット開発はどちらかで行うことになります。どちらを選ぶべきなのか、判断のヒントも紹介します。

〈編集部〉

ROS誕生の経緯とこれまでの変遷

● 最初の正式版が出てから10年以上経過

ROSがSourceForge.net(オープンソース・ソフトウェアの公開や管理ができるWebサイト)に初めてcommitされたのは2007年11月⁽¹⁾で、最初の正式版であるBox Turtleがリリースされたのは2010年3月です。ROSはすでに10年以上の歴史をもち、ロボット開発の加速に多大な貢献をしています。

ROS 1はもともと、Willow Garage Inc.によるロボットPR2の開発環境として誕生しました。そのときのシステム構成や動作環境は、単体ロボットであり、ワークステーション級の計算資源を有し、理想的かつ安定的なネットワーク環境を有すること、などが想定されていました。主な用途は研究であり、新たな動作計画アルゴリズムや制御技術をすぐに試せる環境として利用されてきました。

ロボットを自作するときのラピッド・プロトタイプングにも便利に採用されました。また、企業での商用製品に採用されることも多くなっています。製品に採用された事例として日本で有名なのは、ソニーのペットロボットaiboです⁽²⁾。

● 研究用から商品開発、企業採用へ利用が広がる

十年一昔とは言ったもので、ロボット開発を取り巻く背景や、ROSのユースケースが大きく変わってきました⁽³⁾。

コンピュータの性能は向上していますし、その一方で、組み込み規模の、つまり計算資源に限られるシス

テム環境での稼働や、リアルタイム制御への要求が高まってきました。また、ロボットの動作環境においてはしばしばネットワークが不安定で、遅延やパケット損失が発生する状況への対応も求められるようになりました。開発環境および実行環境がほぼUbuntu一択というのも難点でした。

オープンソースを商用製品に採用する場合、やはり品質も重要視されます。ROS 1では独自の通信プロトコルが実装されていて、その明確な仕様書やドキュメントが不足していた、という問題もありました。

このような、ロボットの開発事情や周囲事情の変化を受けて、ROSの新世代バージョンであるROS 2の開発が始まりました。

ROS 1をベースとするのではなく基本設計から見直し、API互換性さえも捨てて、フルスクラッチで新たに開発されています。商用利用を見据えて、アーキテクチャや開発プロセスも見直されています。ROS 2は2015年8月に最初のalpha distributionがリリースされて、現在も精力的に開発が進められています。

ROS 2の新たな特徴と機能

● 対象OSが拡大

ROS 2では、WindowsやmacOSに加えて組み込みLinuxもサポート対象になりました。これによってROSの適用対象と選択肢の幅が広がりました。例えば、Windowsのみに対応してデバイス・ドライバが開発・提供されていたモジュールを扱えるようになりますし、UWP(Universal Windows Platform)アプリやiOS/macOSアプリとROS 2が連携することもできるようになります。

ROS 2の最新版であるFoxyにおいてサポートされている対象のプラットフォーム(CPUアーキテクチャとOS)を表1に示します。

Tier 1はOpen Roboticsが公式にサポート、Tier 2は公式ではあるもののサポートが限定的、Tier 3はコミュニティ・ベースでのみサポートされます。

以降、ROS 2の新たな特徴と機能を紹介していきます。

1

2

3

4

5

6

7

8

9