

Simple, flexible implementation of applications with ROS (Robot Operating System)

Design robotics to be open

The flexibility to assemble your own robot controller from a variety of software packages: the open source framework ROS makes it possible. With the appropriate interfaces and an open concept, robot applications are simple to implement with ROS. What's so special about it? No specific expert knowledge is required to use ROS packages for programming, offering entirely new possibilities for the use of robots.

Just as in automation, the trend in robotics is heading towards openness and interoperability. To satisfy these requirements you need a robot controller that can be used across a range of manufacturers. Until now many robot manufacturers have relied on their own proprietary controllers. Often these operate with pre-programmed paths, so if any obstacles should suddenly appear, it is very difficult if not impossible to modify the path. In contrast, ROS shows its strengths in dynamic environments, such as when navigating automated guided vehicles (AGV), collision avoidance or grasping objects for example, and so offers greater flexibility.

A cross-vendor software framework

ROS stands for Robot Operating System and is an open source framework for writing software for robotics applications. This programming framework consists of a collection of functionalities,

drivers and a communication layer. Programmers create their own packages within this "framework". The completed ROS packages contain certain functionalities and drivers and are made available to users via a shared community. It may be a path planning function, for example, which users then adapt to their own individual application, such as a pick-and-place operation. The individual packages are modular and therefore versatile to use; they are also compatible with hardware from a variety of manufacturers. As a result, users can exchange the manipulator they used previously and continue using the the ROS packages with new manipulator. The application itself remains the same.

Bundled knowledge open and available

ROS emerged in 2007 in a university research environment. There it has developed into the standard for research projects within robotics. One advantage of the open source framework is the collaboration within the ROS community between specialists from various sectors – from research institutions to robot manufacturers. Together they are able to program even complex robotics applications simply. This online community collaborates on ROS packages, conducts reviews and tests and provides mutual support. This includes providing detailed documentation for the packages, processing "pull requests", in other words suggested improvements for the code, and creating tutorials.

Suitable for industry

In addition to its open source nature, the benefits of ROS also

include the open availability of the source text and the use of modern programming languages such as Python or C++. As such, ROS is suitable for use in industrial applications. It is the appropriate framework for complex applications in which various sensors and actuators from any manufacturer undertake a variety of tasks and control complex algorithms. This requires a standardised communication layer. So ROS can be used across a range of manufacturers, providing a networked, interoperable system completely in the spirit of Industrie 4.0.

Service robotics on the factory floor

With its Pilz service robotics modules, the automation expert Pilz from Ostfildern is introducing a set of modular building blocks, from which users can assemble their own individual robot application (see Infobox). That's because the demand in robotics is for flexible solutions – for compact, versatile assistants rather than massive assembly robots. As a result, the boundaries between service and industrial robots are increasingly breaking down. Service robots are intended to assist humans and so ease the strain in the case of physically demanding or monotonous work. A service robot can be equally at home supporting a doctor on his rounds as helping a technician in industry during maintenance work, for example. The market for service robots is being driven by a large number of start-ups and so is highly dynamic and innovative. That's why Pilz has opted for an open approach for its service robotics modules, with the corresponding physical and virtual interfaces.

ROS by Pilz

For its service robotics modules Pilz uses ROS packages with selected functions, which are required as the robot controller for the manipulator module. Many robot manufacturers have their ROS packages developed externally. In contrast, with Pilz the packages come directly from the manufacturer and take control – and not just in terms of issuing commands to the manufacturer-specific robot control system. The drive electronics are integrated within the manipulator and react directly to the set values in the ROS packages. This gives customers every freedom in terms of path and motion planning. Pilz is also an active supporter of the ROS community, not only by programming packages but also with pull requests and documentation of central code in the ROS wiki. Pilz's customers receive product support for its own ROS packages.

Open, user-friendly and versatile

But how do users benefit from this openness? Users can become familiar with the packages very quickly thanks to supporting tutorials. With ROS, even mechanical integrators with no expert knowledge of programming are capable of customising their own robot application. This may be of interest to small and medium-sized enterprises, for example, who wish to use robots to increase the level of automation in their production. With the framework it is easier for system integrators to integrate components, irrespective of the manufacturer, and implement applications. With ROS it is also possible to implement robot applications such as palletising, conveyor tracking or even camera-based object recognition.

Speaks a simple language

For users who are not familiar with the special industrial programming languages in accordance with EN 61131 it is also beneficial that ROS packages can be implemented with the Python programming language, among others, because Python is easy to learn. What's more, with a Python programming interface it is easy to use the MoveIt! interface – a tool for path and motion planning. MoveIt! uses an environment model and the target position to plan the path of the Pilz manipulator. Pilz provides the robot kinematics, so that the specific manipulator application can be modelled in the 3D visualisation tool RViz or in a simulation environment such as Gazebo, for example, before the purchasing decision is made. This saves customers time and costs with virtual commissioning of the actual robot. The modular nature of the Pilz service robotics range also supports rapid commissioning in accordance with the plug-and-play principle. So even new users quickly succeed in setting up their service robotics application.

High quality and standards for industry

Open source also poses some challenges: essentially ROS packages come from a variety of authors within the community. For that reason the quality of the packages ranges from undocumented blocks to professional, high-quality projects. It is important to Pilz that its ROS modules are of a high quality standard, so the software is developed and tested in accordance with industrial quality criteria and the requirements of the ROS

Industrial Consortium. High-quality ROS packages make system integration easier for users, as the packages are well documented and supporting tutorials are available.

As a result, even users with no expert knowledge of programming can easily implement their own individual robot application using ROS. The modular approach allows you to combine a variety of ROS packages for an application, thereby offering a high level of flexibility for the design of robotics applications. As the open source framework can be used across a range of manufacturers, it offers the openness required in order to implement robot applications in the spirit of Industrie 4.0.

Characters: 8,481

Author
Dr. Manuel Schön
Product Management Controller
Pilz GmbH & Co. KG

Infobox: Building blocks for service robotics

With the service robotics modules from Pilz, users can assemble their own individual service robot application. Initially the package includes the manipulator module PRBT, the control module PRCM, the operating module PRTM and the ROS modules. Key features include openness, the software framework ROS, user-friendly operation and fast commissioning. All together the manipulator

module, control module and operating module form a DGUV-certified package in accordance with ISO10218-1 and so provide all the prerequisites for implementing safe robot applications. This makes it easier to achieve the obligatory CE marking. Application areas include pick-and-place applications as well as modular, industrial robot cells, for example. The six-axis manipulator module is a robot arm developed by Pilz; it has a load capacity of 6kg and weighs in at 19kg.

Characters in infobox: 903

Infobox: ROS Industrial Consortium

The ROS Industrial Consortium is an alliance of industrial companies that use ROS for applications in industry and wish to promote it. The initiative has around 65 members worldwide, from start-ups to corporations and from research institutions to industrial end users. Activities in Europe are coordinated by Fraunhofer IPA. For example, activities include the research project Easy Cohmo (Ergonomics Assistance Systems for Contactless Human-Machine-Operation), in which Fraunhofer IPA is working with Pilz and other affiliated partners to develop gesture-controlled programming. Through its membership of the ROS Industrial Consortium, cooperation with research institutions and collaboration with ROSIN, an EU-funded project to increase the quality of ROS, Pilz is setting standards for its use in the industrial sector.

Characters in infobox: 888