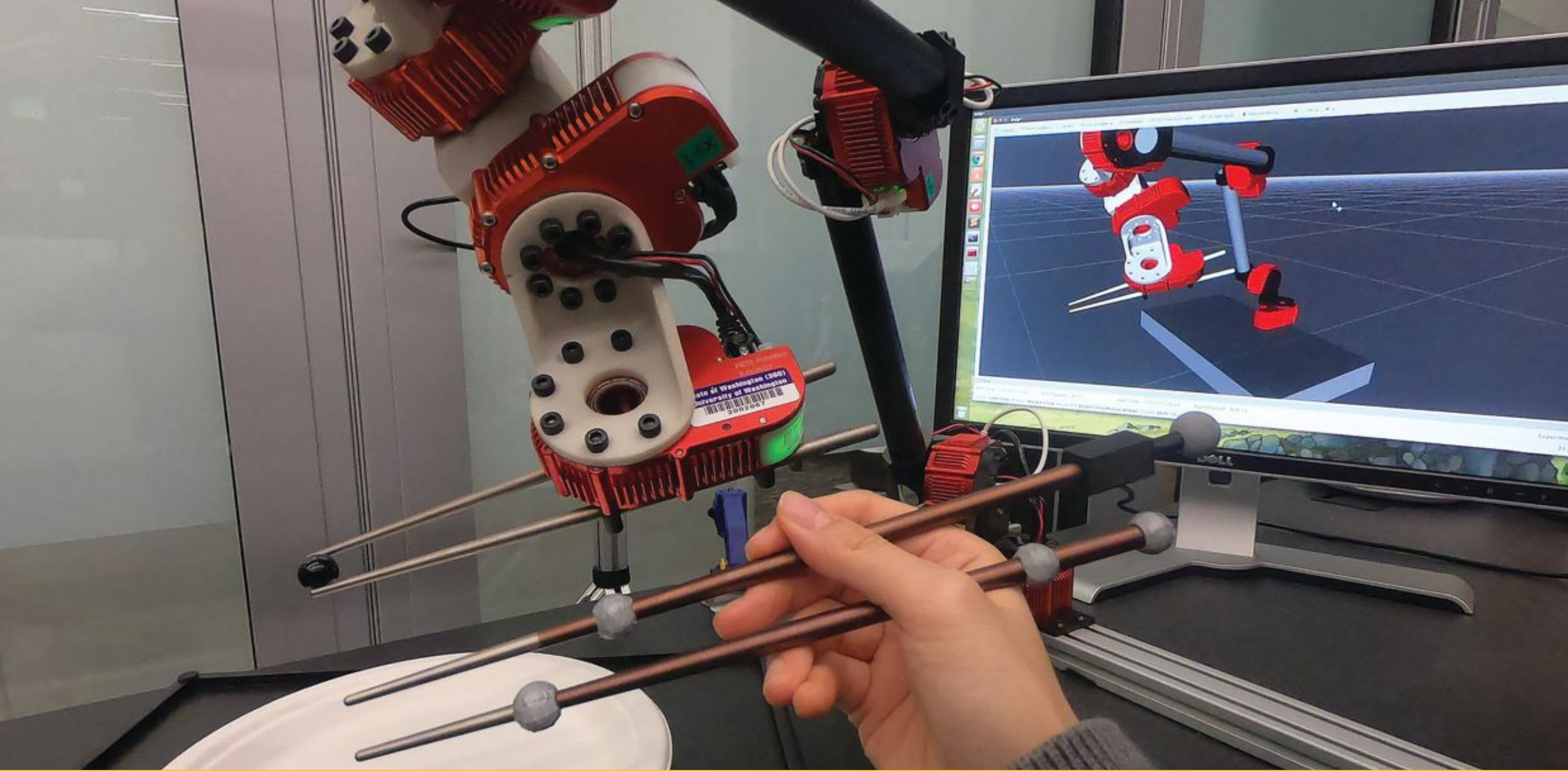

Plug-and-Play Robots: Closer Than You Think

Vendors are developing components, operating systems, and software that make it easier to modify or build robots. Are you ready for robots everywhere?

BY ALAN S. BROWN

Is the market for robots about to explode the way personal computers did in the 1980s?

We may not see robots (other than Roombas) in every household for years to come, but applications of autonomous robots and cobots (robot arms built to work around people) are surging. Suddenly, robots are checking supermarket inventory, cleaning airport floors, disinfecting hospital rooms, and checking petrochemical storage tanks and pipes for corrosion.

The numbers are sometimes surprising. Amazon warehouses alone use more than 200,000 autonomous robots. Brain Corp. claims more than 14,000 cleaning robots use its robotic operating system. In factories, thousands of cobots do jobs ranging from machine tending and driving screws to inspection and pick-and-pack, while autonomous tugs shuttle parts between workstations.

These are not your father's (or mother's) industrial robots. Unlike the mighty but dumb goliaths that weld autobodies, today's robots are smarter. Autonomous vehicles use artificial intelligence to map their location, identify people, and avoid obstacles. Even remotely controlled robots, like drones and maintenance crawlers, leverage AI to make it easier for technicians to control them. AI makes them faster, safer, more economical, and more flexible to deploy than other mobile solutions.

While cobots are essentially smaller and slower versions of their industrial cousins, add-on products make them smart enough to grab workpieces jumbled in a bin and place them in a machine. That task takes seconds, compared to minutes several years ago.

Once, complex functions like autonomy or grabbing unsorted objects would have taken a research team years to develop. Today, small companies with just a handful of employees can build or modify robots to do complex tasks—thanks to the same two factors that drove the rapid growth of PCs in the 1980s.

The first was the rise of killer apps, such as spreadsheets, word processing, presentations, and gaming. They gave everyone a reason to buy a computer.

The second was the growth of standardized components. When IBM released its PC in 1981, it outsourced components from Intel, Microsoft, and others. Once the industry standardized on the IBM AT circuit board communications bus, anyone could enter the market by building a PC from those same standardized plug-and-play components. Hundreds and hundreds of companies did just that.

We are getting close to that point in the robot market now. Pioneers have proven that robots make economic sense and others have followed. Instead of a killer app, there are lots of useful apps.

A research robot at University of Washington's Personal Robotics Lab uses HEBI's platform to teach a robot to use chopsticks.

Meanwhile, an ecosystem of suppliers has emerged to provide everything from cameras, sensors, and arms to autonomy systems, mapping and vision software, and operating systems. Today, this robot ecosytem is sprawling and messy. Sometimes it is more plug-and-pray than plug-and-play. Yet it is already making integration and customization easier for robot developers and users alike.

UP AND RUNNING

This all comes together in UR+, an online marketplace for plug-and-play add-ons created by Universal Robots, the world's first and now largest cobot maker. UR+ resembles the Apple Store, but for robot components, software, and application kits. Its products range from vision systems and grippers to kits that enable robots to perform specific tasks.

Universal's goal is to provide plug-and-play systems. To do that, it shares its mechanical, electrical, and software specifications with developers so they can build compatible components.

"Take a simple gripper," Joe Campbell, Universal's senior manager of application development, said. "If you mount it on the robot faceplate, there's a single cable that goes from the gripper to the robot. Download the software to our teach pendant and we know the gripper's make, model, and operating sequences. It maps all the way up into our software. There is no custom programming. This is true not just for grippers, but for vision systems, tool changers, force sensors, flexible feeders, and other end-effectors. If you develop to those standards, you wind up with a truly plug-and-play gripper," he said.

Other robot arm vendors, such as Kuka and Fanuc, have set up similar marketplaces and encouraged industry vendors to develop components and software for their robots. Denmark's OnRobot, is trying to get a foot in the door of all these markets with its grippers, cameras, tool changers, and other add-ons.

To connect with robots from different vendors, OnRobot developed its "compute box." Kristian Hulgard, OnRobot's general manager for the Americas, likens it to "a virtual Swiss Army knife for communications."

"It's a controller that translates our product language into the robot's language," he said. "If you know how to program your robot, you already know how to program all our tools and sensors because it's all done in the robot's software. You just call up our subroutines."

HEBI'S kits make it easy for researchers to build robots using ROS or other interfaces. According to ABI, ROS will be on 55 percent of all robots sold in 2024.

To switch between OnRobot's tools, users click a button, remove the tool, and snap in a new end-effector. This makes it far easier to redeploy robots and use them as the general-purpose tools they were meant to be.

Hulgard argues that moving a robotic arm to the correct position is fairly easy. His customers are more concerned with how to grip, process, sand, finish, and assemble parts. These tool-specific operations are often complex, and where the robot's true intelligence lies.

In addition to plug-and-play components, UR+ also offers application kits. They include the hardware and software needed for a cobot to perform specific tasks, like welding, sealing, nut driving, and picking. They are applications-in-a-box.

One example is a sanding kit from Canadian firm Robotiq. It includes an orbital head, vacuum system, and software that enables users to finish contoured surfaces without complex programming.

"The old way of doing that took many hours or even days of programming," Campbell said. "Robotiq's software lets you program a complex contour in maybe ten minutes to half an hour."

Universal's impressive ActiNav application enables cobots to quickly pull unsorted parts from a bin and pack, place, or inspect them. The kit includes a 3D vision sensor, an autonomous motion controller, an alignment marker, and all the control software, cables, screws, and brackets needed to get going.

Users train the cobot with a six-step wizard, marking off safe spaces and noting the location of the bin. They scan the end-effector holding the part, and the ActiNav app uses the image to capture the relationship between the end-effector and part. Then, the cobot goes to work. ActiNav reduces the time needed to set up a bin picking—a task most small companies with robots could never even contemplate before—to as little as two hours.

VISION

Many of the same products sold in robot marketplaces help simplify building robots from scratch. Vendors make this easier than ever by bundling hardware with software to provide integrated solutions.

One example is the 3D stereoscopic camera system developed by Germany's Roboception. Its software calculates 3D images by measuring minute differences in the maps and images taken by its two cameras, CEO Michael Suppa explained. Its software then calculates where a cobot can grasp randomly oriented parts, something that often reqires extensive cobot training first.

According to Suppa, the process is breathtakingly fast. "In most cases, we can identify an object in less than one second, pick 1,000 parts per hour, and place 200 to 300 parts per hour" Suppa said.

Roboception also provides simultaneous location and mapping, or SLAM software. It lets autonomous robots use its stereoscopic images to build a map of an unknown environment while locating their position on the map. Combined with a built-in inertial measurement unit (which combines a gyroscope and accelerometer) to double-check the map, the system lets mobile robots see their world—and their location within it—in three dimensions.

Many vendors want to make it easier to solve customer problems, Suppa said. "If the problem is recognizing and grasping the part, does a customer have to source the camera and gripper and integrate them himself?" he said. "Or do we have a system that has both and already solves the problem for you?"

Siera.AI, based in Austin, TX, also offers vision capabilities, but its software grew out of its work on data analytics. It originally developed its S3 vision system to install on existing manually operated forklifts. Linked to the cloud, it analyzes warehouse traffic flows and monitors operator safety. The system helped Siera's customers track driver attentiveness and retrain operators to take sharp turns slower, so loads did not fall off their forklifts, CEO Saurav Agarwal, *NY P '96*, said.

The S3 platform also bolts onto human-driven warehouse forklifts and tugs to provide autonomous control. The system looks for people, obstacles, and moving objects, slowing the vehicle within 5 ft. of them and stopping it completely within 1 ft. It never loses its attention and reacts much faster than people—milliseconds rather than fractions of a second—to unexpected threats, like a forklift coming around a blind corner. This reduces the likelihood of accidental collisions that cost the average warehouse hundreds of thousands of dollars annually.

More recently, Siera launched its own driverless tug. The new T-10 is capable of towing 10,000 lbs. at speeds of up to 4.5 mph. It uses SLAM to learn and map its environment while sending updates to the cloud for analysis. Customers can build and train AI models that enable the T-10 to do things like identify color-coded drums or read a hazmat label.

Although the T-10 embeds Siera's S3 platform, the build relies on standardized parts. These include the ROS robotic operating system for internal communications, off-the-shelf mapping and other software, lasers, high-end infrared sensors, cameras from different vendors (depending on lighting and application), and a vehicle drivetrain and controls.

What makes the robot proprietary is its ability to fuse data from different sensors when making navigation decisions and its cloud-based analytics features.

"THE OLD WAY OF DOING THAT TOOK MANY HOURS OR EVEN DAYS OF PROGRAMMING," CAMPBELL SAID. **"ROBOTIQ'S SOFTWARE LETS YOU PROGRAM A COMPLEX CONTOUR IN MAYBE TEN MINUTES TO HALF AN HOUR."**

This maintenance robot from Baker Hughes inspects and cleans boiler waterwalls without requiring scaffolds or disassembly. Source: Baker Hughes

OPERATING SYSTEM

If components for robot builders are not yet plug-and-play, today's combinations of hardware and software are certainly simpler to integrate than in the past.

Yet imagine how much easier that would be if the industry had standardized operating systems like the one Microsoft developed for IBM-compatible computers. That is exactly what Open Robotics is trying to accomplish with its open-source Robotic Operating System, or ROS.

Company CEO Brian Gerkey, *LA B '98*, would be the first to tell you that ROS is not actually an operating system. It is more like a software development kit (SDK), with libraries of software, tools, and simulation capabilities that enable engineers to build and model functioning robots.

It also provides an abstraction layer that lets programs command cameras, arms, force sensors, or other devices without

A snap-on OnRobot gripper uses a gekko-like surface to lift a machine part. Source: OnRobot

a detailed understanding of their hardware commands. It makes it possible to switch hardware components without reprogramming—as long as vendors provide a device driver.

Agarwal likens ROS to plumbing: it makes it possible for different types of hardware and software—there can be dozens of subroutines running at once—to connect with one another.

Because ROS is open source, it does not tie users to any one proprietary system. The ROS community has thousands of active developers and hundreds of thousands of users. Over the past ten years, volunteers have created an extensive library of documentation and how-to tutorials. Their efforts are mirrored by a cottage industry of churning out ROS technical books and training classes. Most university robotics courses teach ROS, and it enjoys support from major corporations.

Yet ROS has issues that trace back to its history as a Stanford University project to get robot researchers up and running fast. While many roboticists have adapted ROS to commercial applications, weaknesses remain.

For example, the developers wrote it in Ubuntu Linux, which buffers data before processing it. This makes it too slow for some industrial and other applications. Also, ROS runs only on computers and does not support embedded systems for development and commercial production.

"We could have improved ROS around the edges, but we thought it was best to take what we have learned and build a system designed to meet safety-critical production scenarios," Gerkey said.

Enter ROS 2. One big change is a new real-time communications system that prioritizes messages and setting delivery deadlines. This is essential for applications like industrial robots and autonomous vehicles.

Also new is Micro-ROS, a stripped-down version of ROS 2 that runs on microcontrollers where cost, size, weight, and power are constraints, such as drones and consumer robots. Gerkey also promises better documentation and the ability to run on Windows and eventually Apple Mac OS.

Given such advances, ABI research predicts ROS and ROS 2 will be on nearly 55 percent of the 915,000 commercial robots shipped in 2024.

BRAIN

While ROS has an enormous following, it is not the only OS in town. Several companies are developing operating systems of their own and Blackberry and Qualcomm both offer a stripped-down, chip-based OS. Startups A&K Robotics and Qobotix are also developing operating systems.

The leading alternative to ROS is arguably BrainOS from Brain Corp. It comes closer to a true commercial OS for autonomous wheeled systems, and features navigation, cloud-based mapping, sensor integration, and speed control. More than 14,000 mobile robots use BrainOS for floor cleaning, delivery tugs, and store shelf inventory management.

Tennant, a 100-year-old maker of floor care equipment, partnered with Brain to develop robotic floor scrubbers four years ago, while BrainOS was still under development.

A Universal Robot cobot arm uses an ActiNav kit to pick unsorted parts from a bin. Source: Universal Robots

OnRobot has developed robot end-effectors, such as this sander, that are easy to swap and work with a variety of robot arms. Source: OnRobot

A Talos humanoid robot from PAL Robotics based on the open-source Robot Operating System (ROS). Source: PAL Robotics

Mechanical floor scrubbers are complex. They need to regulate several discrete functions, such as water pressure, brush speed, downward pressure, and squeegeeing and vacuuming excess dirty water.

"Floors have a lot of variation," said Tennant vice president of engineering Barb Balinski. "Getting the floors to shine and reflect the way people expect, no matter if you're going straight or turning a corner, that's difficult to do."

It took Brain and Tennant three years to launch its first robotic scrubber, the T7, in 2018, Balinski said. They spent much of that time fine tuning the system to ensure the robot delivered the same performance as a manual system. Even though it was a learning experience, they still needed a year to design Tennant's autonomous T380AMR, which launched in 2020.

While most OS developers focus on autonomous robots, Qobotix is creating an operating system for cobot arms, said CEO Egor Korneev. He argues that robotic arms—and eventually end-effectors and tools—will become commodities. What will count is the software that drives them.

His goal is to emulate Google's Android operating system: Code an OS that any robotic arm maker can use and create an app store to empower the robot to perform certain tasks. Korneev's team has already begun populating the store with apps for variable picking from bins or belts, visual inspection, and packaging.

Not everyone wants a full-blown robot, said Dave Rollinson, a co-founder of HEBI Robotics. HEBI grew out of snake robot research at Carnegie-Mellon University. After struggling to find a market, it morphed into a supplier of compliant motion control actuators and eventually robot kits for researchers. The kits support ROS, and also MathWorks' MATLAB and Python programming language for people who value accessibility over functionality.

"That came out of the lab," Rollinson said. "Using MATLAB really opened up who could contribute to snake robot research. A lot of ideas about how to make things wiggle came from mechanical engineers and even art majors, who know how to think about things spatially in ways most others cannot. They're not going to sit down and start writing in C++. If they could express those wiggles as a sine curve or whatever in MATLAB, there's not a lot else they had to do to test an idea."

ROBOT RUSH

Robotic hardware, software, and operating systems are getting easier to use and integrate. Yet controlling a robot—calculating kinematics, fusing sensor data, and making navigation decisions—is far more complex than building a personal computer. It takes a lot of integration engineering.

Roboception, Siera, and other component vendors may provide hardware-software solutions, but they work closely with customers to implement them. Even with BrainOS, it took a year to get Tennant's second cleaning robot out the door. Open Robotics funds its ROS work by consulting with firms building robots.

Tennant's T380 autonomous scrubber runs on the BrainOS operating system. Brain Corp. says 14,000 autonomous devices use its OS. Source: Tennant

Lee Baldwin can attest to the hard work. He is segment manager for core autonomy of the Autonomy and Positioning Division of Hexagon, a Swedish firm that parlayed its expertise in global navigation satellite systems (GNSS) into control systems for autonomous cars, mining equipment, and tractors.

Hexagon wants to sell kits to make tractors autonomous, so Baldwin built a demonstrator. He started with a farm tractor whose electronic controls ran over a conventional CANbus communications network. Using a network sniffer, his team sussed out how the tractor's controls worked. From this information, they created a set of ROS subroutines to run the tractor using CANbus. Through the ROS abstraction layer, they added GNSS, cameras, and laser distance sensors. ROS also supported the autonomy level.

Safety was a major concern. Not only did the tractor have to recognize obstacles like rocks, ponds, streams, and high-tension towers, but it also needed to avoid partially obscured humans, cows, and other animals moving through a field.

This involved training the system's AI by throwing tens of thousands of images of humans and other hazards at it. Then they tested it on different images and tweaked the algorithms to improve performance. Several companies are also developing virtual reality software to generate test cases, and these are likely to become part of the robot infrastructure one day. Even so, testing takes time because no one wants to unleash smart robots unless they are safe, Baldwin said.

Ultimately, Hexagon wants to develop driverless systems for tractor manufacturers who—like cleaning companies, welding suppliers, automakers, and others—want to provide self-operating versions of existing products.

Add to that the number of entirely new robots in everything from plant maintenance and warehouse operations to supermarket inventory—and demand for robots is sure to grow.

It will only get easier for users and developers to find the tools and components they need to meet that demand. For robot users, plug-and-play tools and solutions are already emerging for cobots and industrial robots. Most are now based on proprietary systems, but vendors are making it even easier to switch between different brands of robots.

For robot builders, the story is more complex. It still takes lots of engineering time and safety testing to launch a new robot. Yet more and more vendors are combining hardware and software to simplify integration. ROS 2 may eventually become the industry's go-to operating system, or perhaps systems like BrainOS will retain important niches. Eventually, engineers will write industry codes that make it even easier to plug into these systems.

Robots may not be ready to explode the way computers did in the 1980s, but they have made major steps towards increased ease of use. If demand keeps growing, true plug-and-play robots cannot be far behind.

ALAN S. BROWN has written broadly about engineering, technology, and science for more than 30 years. He is a board member of Science Writers in New York, a writer for The Kavli Foundation, and a former senior editor of ASME's *Mechanical Engineering* magazine, and contributes to a wide range of publications. He graduated New College at Hofstra University *magna cum laude* in 1974 and can be reached at ***insight01@verizon.net***