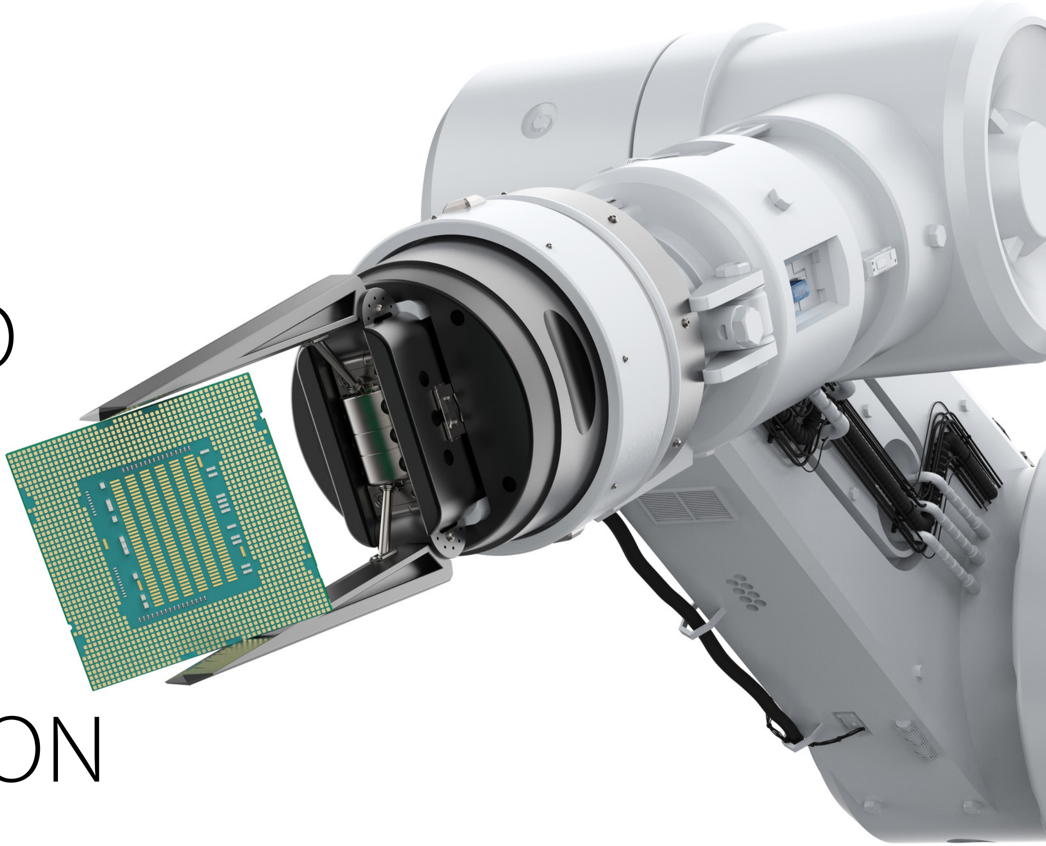
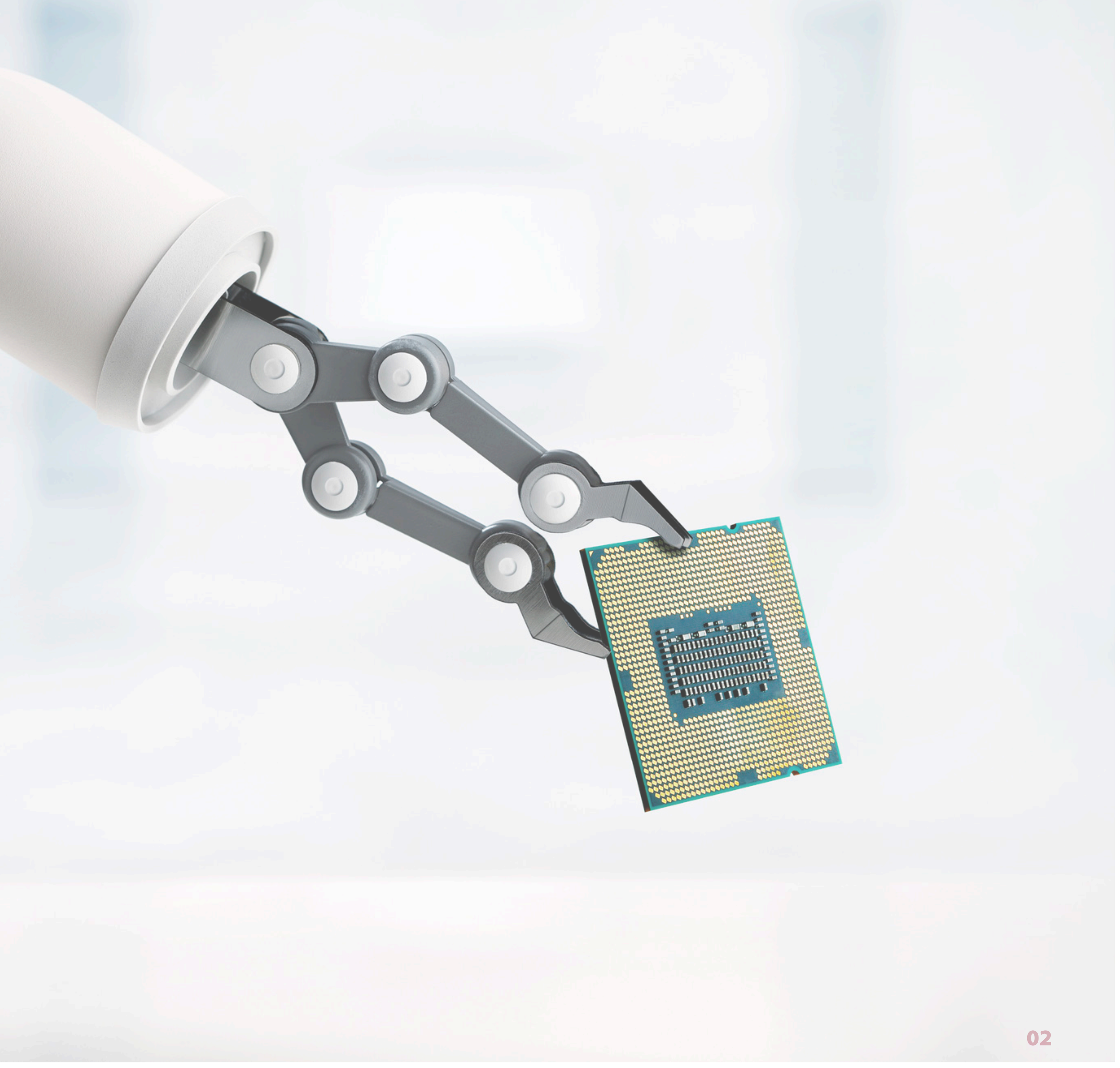
ISSUE 3, 2023



A GUIDE TO ROBOTICS AND AUTOMATION

DISTRELEC



WELCOME

We have already released two informative editions, each delving into the realm of robotics. The first edition provided an introduction to the captivating history and evolution of robots, while the second edition explored their pivotal roles in diverse industries such as automotive, logistics, and agriculture.

Now, we proudly present the third and final part of our Robotics and Automation Guide. This edition offers an extensive overview of crucial components in robot engineering, encompassing everything from power and movement to software and control systems.

At Distrelec, we stand at the forefront of distributing cutting-edge software solutions, empowering our customers to build highly efficient, fully automated workflows for robot design and operation.

Curious about the future of robotics? The upcoming section unravels the exciting prospects, from advancements in artificial intelligence to the rise of collaboration robots - turning once-distant dreams into reality.

This enlightening instalment provides readers with a glimpse into the ongoing trends within the robotics sector, such as AI, Industry 5.0, self-driving cars and more. It also focuses on the persistent engineering challenges that require solutions. Delve into the thrilling new innovations that are tackling these challenges.

Read further to envisage the exciting future of robotics.

Raj Patel

Managing Director

Table of contents

Robot Power and Movement

Energy, Control and Actuation for Modern Robotics.

Providing the power, control and actuation that underpins the ability to move is fundamental to modern robotics, which is why identifying the optimum power source, actuators, motors and drives needs to happen as early as possible in the robot design process.

A fundamental factor guiding selection is the type of robot and its intended use. Within industry, robots take many forms, including cartesian, SCARA, cylindrical, delta, polar, gantry and articulated arm. Target applications include assembly, welding, machine tending, packaging, painting, pick and place, inspection and testing, to list but a few.

The type of power source is a primary consideration, particularly with regard to reliability, size, weight and lifecycle. To achieve optimised motion, robot OEMs also need to think about the most appropriate motor and actuators, and how these are best controlled.

BATTERIES

Batteries are the most commonly selected source of electrical power and the question of which battery type is most suitable hinges on criteria such as safety, life cycle, weight and cost.

Both primary (non-rechargeable) and secondary (rechargeable) batteries are found in industrial robot applications. Although primary batteries present the obvious disadvantage of requiring replacement, they typically provide greater power for their size, making them suitable for certain light-drain applications. More often than not, however, rechargeable batteries are preferable.

Historically, the common types of rechargeable batteries for robot applications are nickel-cadmium (NiCd) and lead-acid. In addition, gelled lead-acid batteries, which are capable of providing power of up to 40 Wh/kg, have sometimes been deployed. Further secondary battery technologies include nickel-metal hydride (NiMH), silver-zinc and lithium-ion.

Now, however, lithium technologies have become the popular choice for today's robot designers. Indeed, the performance, shelf-life and scalability of lithium-ion batteries – lithium battery banks can be scaled to meet most automation applications – have ensured wide appeal for the industrial robot arena. Among the many advantages of lithium-ion technologies is their light weight. Moreover, the lithium element itself is particularly reactive, which means it has the ability to store a lot of energy; typically around 150 Wh of electricity can be stored in 1 kg of battery. This compares favourably to a NiMH battery pack, which can store 60 to 100 Wh.

It is worth mentioning welding robots, commonly found in sectors such as automotive. Here, the same power source that feeds the welder can be used to power the robot's electronic drives and motion-control components, and for these applications inverter power sources prove popular. Some of the latest inverter technologies automatically adjust input power while maintaining a constant output, and provide power surge blockers to ensure performance is unaffected by the simultaneous use of other devices requiring high current.

Clearly, battery geometry is highly influential when it comes to selection and robot type, and form will dictate which battery types should be included in the decision-making process. Similarly, weight is a significant factor and may depend on whether the robot is intended to be portable or fixed.

Also, as competitive advantage can be built on the time that a robot is able to run before additional charge is required, durability and capacity will be key aspects of the life cycle element of the battery selection equation.

PHOTOVOLTAIC (PV) CELLS

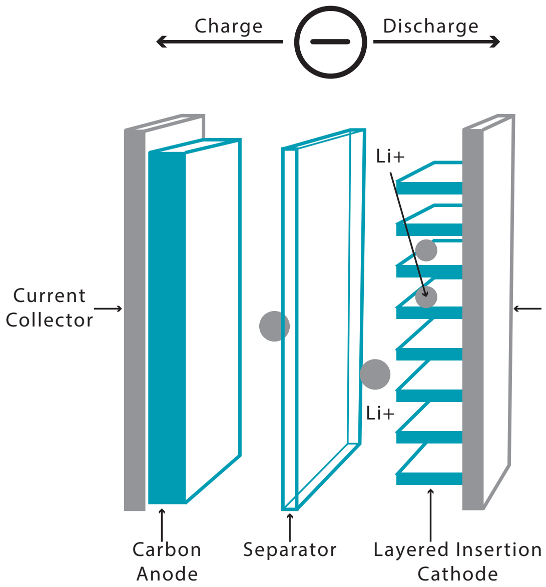
Although the use of solar power has brought benefits to many areas of the engineering world, it has yet to find its place in industrial robotics. Some biology, electronic, aesthetics and mechanics (BEAM) robots, such as automated lawn mowers and vacuum cleaners, do currently use PV technology. The common configuration sees a solar cell, via appropriate circuitry, charge a capacitor to a set voltage level and then discharge it through the motor(s).

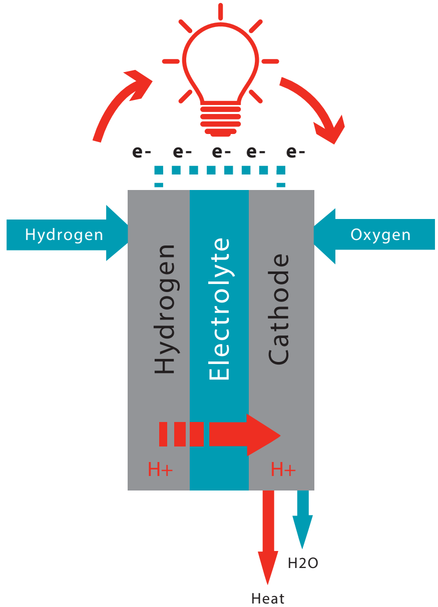
Upscaling this technology to industrial robots has yet to occur on a commercial level. There are a number of reasons behind the lack of progress, but mainly it's due to the rather low power density of solar cells (Wp/m2), which is insufficient for most modern industrial robots.

FUEL CELLS

A far more likely future replacement for conventional batteries in industrial robots is fuel-cell technology, which is able to generate electricity by combining hydrogen, methanol or simple alcohol with oxygen. At present, cost is the restrictive barrier, but this could change as fuel cells are adopted in wider consumer markets.

Fuel cells supply energy by deriving power from a hydrocarbon source at high efficiencies of up to 75%. A typical configuration includes two electrodes located either side of a conductive electrolyte. Current is generated using a concept similar to that of fuel combustion, whereby protons are allowed to pass through the membranes and electrons are forced to bypass from anode to cathode via the electric circuit. Fuel-cell efficiency can be increased even further by making use of waste heat.





Lithium-ion batteries are the cell of choice today, although advances in fuel cell technology could see this change in the future

ACTUATORS

With the decision about the power source complete, attention turns to the actuator technologies and motors needed for linear and rotational motion.

STEPPER MOTORS

Stepper motors are typically found in applications where cost is the primary factor, such as in common pick-and-place robotic devices. Among the principal benefits are high-accuracy positional control – which is why they are often used for systems such as 3D printers and CNC milling machines. This is because stepper motors are specifically designed to offer high holding torque, which in turn provides the ability to step incrementally to the next position. They can be used to advantage in applications where there is a defined need to control rotational angle, speed, position and synchronism. Furthermore, because stepper motors deliver maximum torque at low speeds they are a good choice for applications requiring low speed and high precision.

While stepper motors proved popular in early robot applications, their use has reduced in recent years. Among the reasons for this are factors such as efficiency, the need for encoders or limit switches to establish reference positions, and the potential for missed steps if overloaded. However, just as likely is the advent of more advanced brushless AC servomotor technology.

SERVOMOTORS

While many early electric robots used DC servomotors (as they give reasonable power output with a good degree of speed and positional control), most new industrial robots use brushless AC servomotors. Such motors offer the benefits of higher power output and virtually silent operation, while the absence of a brush means these high-torque devices are highly reliable and require virtually no maintenance.

Servos also have an inherent benefit in that they provide a high degree of angular precision, rotating only as much as requested before waiting for the next signal.

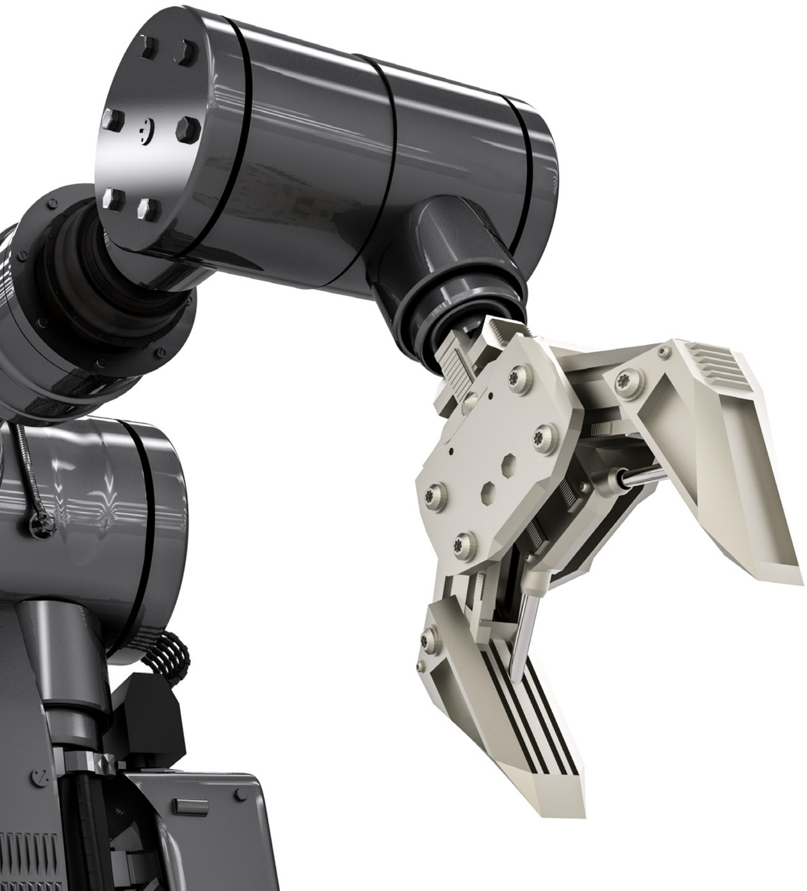
The principal difference between digital and analogue servos is the signal and how it is processed from the receiver to the servo, and how the servo uses this information to send power to the motor. Analogue servos control motor speed by giving on/off voltage signals to the motor, whereas digital servos feature a small microprocessor that analyses the receiver signals and processes them into very high-frequency voltage pulses to the motor.

Unlike analogue servos, which send out 50 pulses per second, an advantage of digital servos is their ability to send pulses upwards of 300 pulses per second. The pulses are shorter in length, but with so many voltage pulses, the motor accelerates faster and provides constant torque. With digital servos, the amount of power sent to the motor can also be adjusted to optimise performance and precision.

EFFICIENCY, SIZE, ACCURACY, RELIABILITY, SPEED AND TORQUE

Selecting the optimum motor is one of the most important parts of a robotics project and is based on considerations that include torque, speed, precision, voltage, cost and form factor.

In a robot, motor torque is typically conveyed to a wheel or actuator that subsequently prompts rotational or linear robotic motion. To estimate the required torque, engineers need to determine the mass of intended maximum payload, as well as the system's static, dynamic and rolling friction.



For wheeled robots, it is important to specify the speed that the wheels need to turn. With faster speeds, there is usually a trade-off with precision. Ultimately, to achieve the accuracy required of robotic arms, servomotors are the popular pick chiefly because they have internal position regulation and are geared down to lower speeds, resulting in very precise position control.

Another important consideration is operating voltage. Before planning what battery packs will be used in the project, it is best to determine the nominal voltage when the motor runs. Usually, the higher the voltage, the higher the motor speed. The 'voltage constant' on the motor datasheet can be used to determine the speed-per-volt.

HYDRAULIC AND PNEUMATIC ALTERNATIVES

Hydraulics were a fairly common sight on early robots as this technology is more rigid and controllable than pneumatics, and could provide greater power than the electric drives available at the time. Hydraulics also offer the potential to create a large reduction ratio.

The principal disadvantage of hydraulics is the comparatively slow pace of operation, while the high pressures involved mean that leaks can be problematic.

In terms of pneumatics, many simple pick-and-place systems are driven using compressed air, which introduces a level of affordability but has the disadvantage of being difficult to control. Essentially, the compressability of air introduces an additional 'dead-time dynamic' to the system that makes control more challenging.

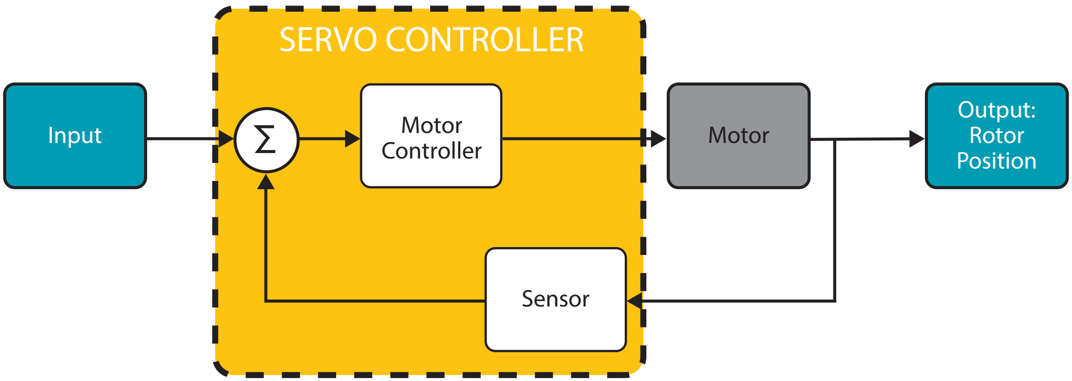
Pneumatics are also used with a number of industrial robots to drive end effectors; pneumatic cylinders can deliver large forces and are a good choice for larger grippers. It is also possible, although fairly uncommon, for some robots to use pneumatic cylinders to move their body using an onboard bottle of pressurised air as the power source. The factor limiting wider adoption tends to be that pneumatics can only produce relatively small and simple back-and-forth motions.

CHOOSING THE RIGHT MOTOR-CONTROL TECHNOLOGIES

Choosing between a servomotor and a stepper motor comes down to a trade-off between complexity and control certainty. A stepper motor is simpler in configuration because, unlike a servomotor, it does not require an encoder. This design concept makes stepper motors simpler to control, but only if the robot has low performance requirements. Any robotics engineer wanting to push stepper motors close to their limit will find they become far more difficult to predictably control.

One of the benefits associated with stepper motors is their ability to be controlled in an open-loop system. Open-loop control means there is no requirement for feedback information concerning position, thus eliminating any need for encoders or resolvers, and the costs these incur. Position is known simply by keeping track of the input step pulses.

SERVOMOTOR

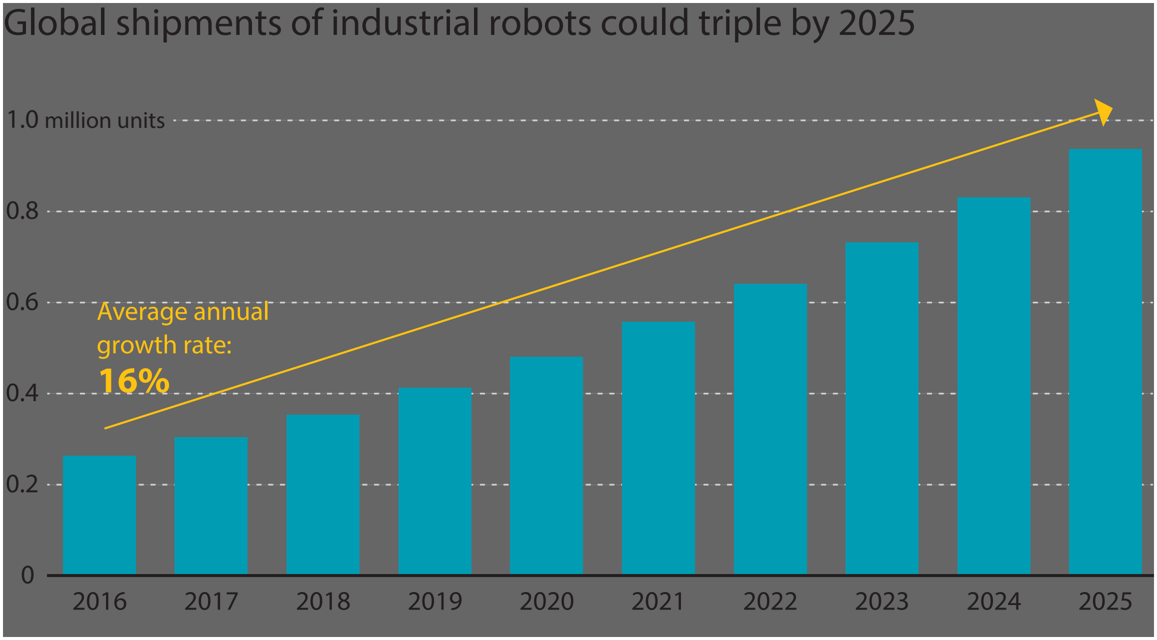


STEPPER MOTOR

Servomotors use a closed-loop control system for positioning, while stepper motors can be positioned to a specific angle of rotation

Servomotors are used in closed-loop systems with a digital controller that sends velocity commands to a driver amplifier, which in turn feeds the servomotor. A feedback device (encoder or resolver) provides information on the servomotor's position and speed. To break it down further, the device is controlled by a feedback signal generated by comparing output signal and reference input signal.

As a result of the closed-loop system, a servomotor can operate with a specific motion profile programmed into the controller. Servomotors are controlled using a principle called pulse width modulation (PWM), with the angle of rotation determined by the duration of the applied pulse.



The global industrial robotics market is set to see a CAGR of 16% (Source: Research and Markets)

KEY FACTORS TO CONSIDER

Power sources

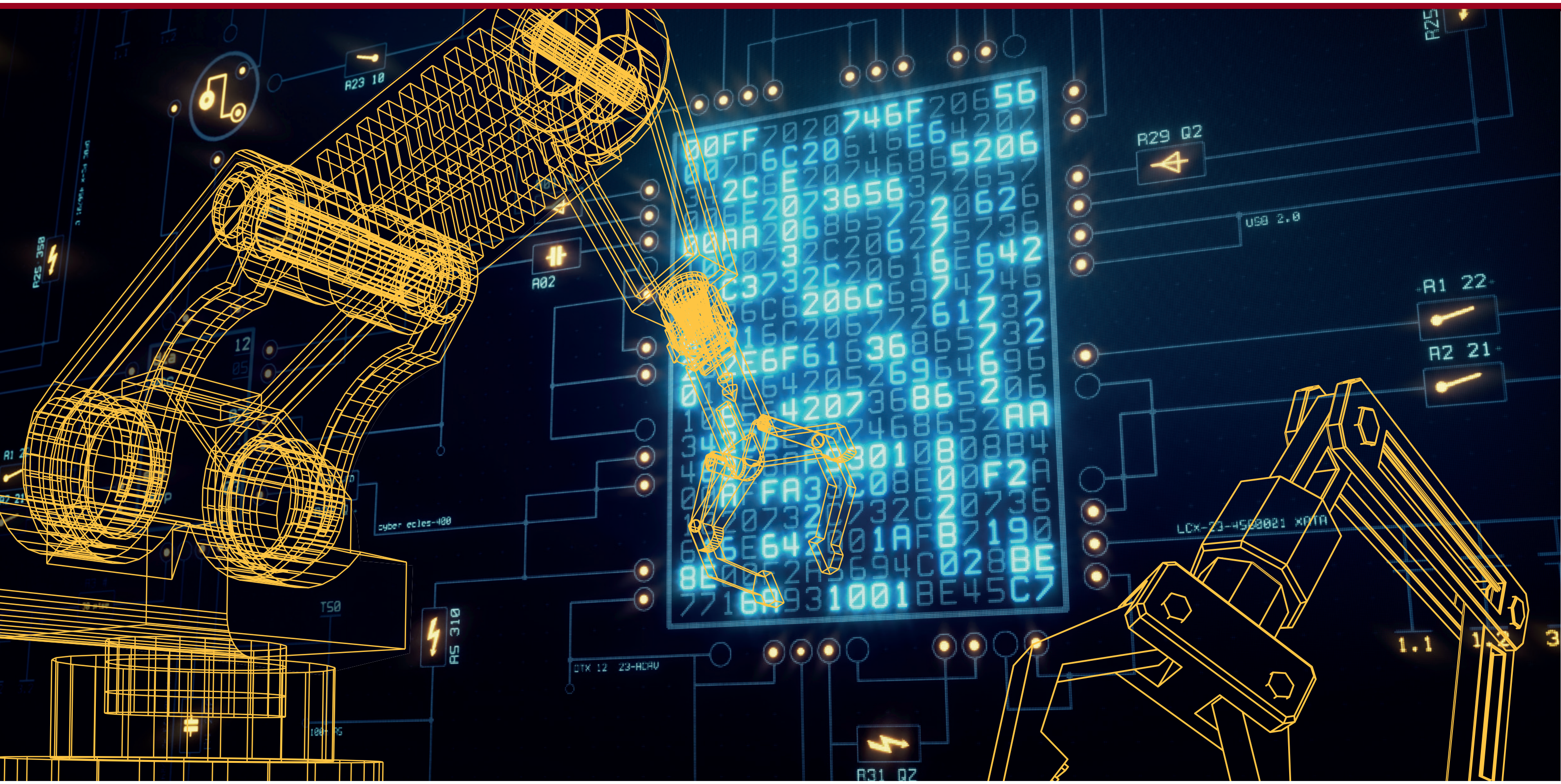
- When it comes to power sources, batteries are the current technology of choice. Both primary (non-rechargeable) and secondary (rechargeable) types can be deployed, although the latter are generally considered the popular pick.
- Types of secondary battery include NiCd, NiMH and lithium-ion. Power density is highest with lithium batteries.
- Important factors in the purchase decision are safety, life cycle, size, weight and cost.
- Alternative power sources include PV, fuel cells, thermoelectric generators, super-capacitors and flywheel energy storage. Extra-large robots weighing several tonnes will require a diesel generator or three-phase mains supply.

Actuators

- Although brushless AC servomotors are generally the first choice for new-build industrial robots as a result of their angular precision, high power output, reliability and low noise, stepper motors can also be used.
- Stepper motors offer good positional control and high holding torque.

Control

- As a stepper motor does not require an encoder, it is simpler to control, although there are limitations in terms of performance.
- Pushing a stepper motor to its limits brings a level of unpredictability to the control equation.
- Servomotors are favoured for industrial robots thanks to their integral position feedback capabilities. If the target position or speed is not reached, this is relayed to the servo loop for correction.



Computational Options for Robotics

Robot architecture and design now spans a wide range of functions and abilities. This strongly affects the processing power and what is used to deliver it. Within the controlled environment of a safety cage, a production robot needs relatively few safeguards and can make use of simple procedural control strategies.

Designers need to ensure that the robot will stop if the cage is opened, or if parts are not aligned as expected. Even so, many of the safety challenges can be met using simple hardware interlocks rather than complex combinations of image sensors and software. The key processing requirements are to ensure efficient and precise control of motion.

This primarily demands the use of microcontrollers or digital signal processors to manage the flow of power to motors and other actuators.

Conventional production robot designs tend to be inflexible. Each of their programs needs to be programmed, simulated and tested extensively before the robot is allowed to proceed. In manufacturing, users want robots to be more flexible so that they can be quickly assigned to different tasks. They also need to be able to move around the production floor, which entails operation outside the safety cage. These requirements call for greater processing power to provide the robot with the ability to navigate without accidentally colliding with objects or harming bystanders.

As a result, robots need to be able to process sensor input in real time and make intelligent decisions on the fly as circumstances change. The further robots move away from the safety cage and the more they interact with humans, the more processing power they will need as they move beyond the relatively controlled environment of the shop-floor. Service robots and delivery drones need to be able to react intelligently to complex situations.

In these more advanced scenarios there is a clear need for greater software sophistication, which goes hand in hand with computational throughput. The designer has a high degree of choice as to how to supply the required processing power, not just in terms of suppliers but overall architecture.

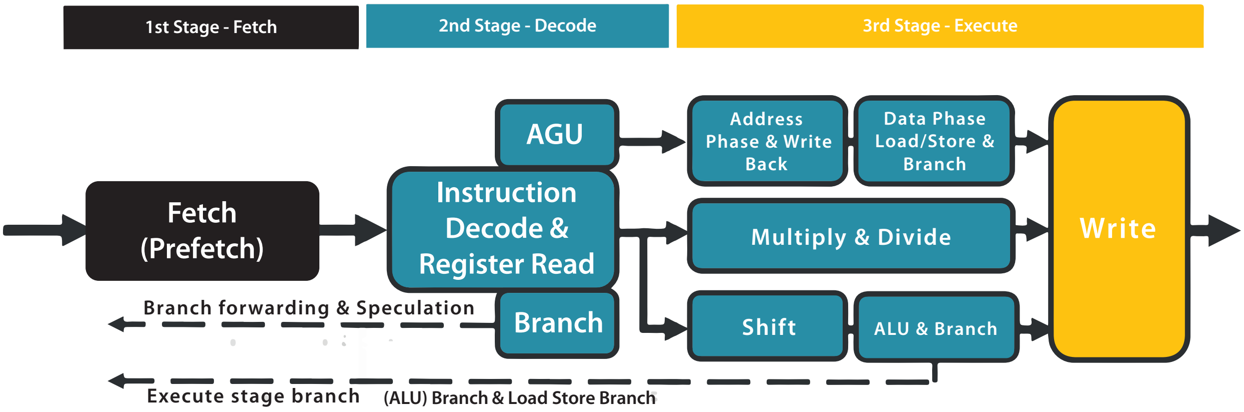
The microcontroller unit (MCU) has for many years been the computational element of choice for basic robots. The core of the MCU is the microprocessor. Initially, the microprocessor cores in MCUs were optimised for simple arithmetic and logic-level control, but since their introduction almost 50 years ago, the performance and data-handling capabilities have expanded dramatically. Today, microprocessor cores that natively operate on 32-bit data words and which offer performance features such as pipelining and Harvard architecture are now offered at a cost level that allows even simple systems to make use of them.

In a typical 32-bit microprocessor core, such as the ARM Cortex-M3, there is an instruction pipeline that separates execution of commands into a number of phases. In the M3 pipeline, first the instruction is fetched from a local cache. If the instruction is not in the cache, it first must be loaded from the main memory. Once in the pipeline, the instruction bytes are decoded to evaluate which functional units need to be activated to execute the instruction. Finally, execution takes place.

Pipelining is used to hide effects such as the latency of memory. It allows execution of multiple instructions to be overlapped and helps boost clock speed, as fewer logic steps are needed per cycle. Faster processor cores use more extensive pipelines that can be ten stages long or more. The drawback of long pipelines is high branch latency. If a branch is taken, it takes time to refill the pipeline with the instructions needed for the new branch.

Support for interrupts allows the processor core to suspend execution of the main program temporarily and handle other tasks. Interrupt handling is a key component for applications that need real-time response to events. Without it, the program code would have to contain loops that continually poll for information on external events, which would be far more wasteful of computational capacity.

A priority scheme employed by most processor cores allows interrupts from relatively unimportant peripherals to be ignored while the processor takes care of critical routines, such as transferring control from one task to another or the input from a critical interrupt. The result is a highly flexible architecture that can handle many different types of real-time application.



The ARM Cortex-M family of processors are the backbone of many microcontroller products

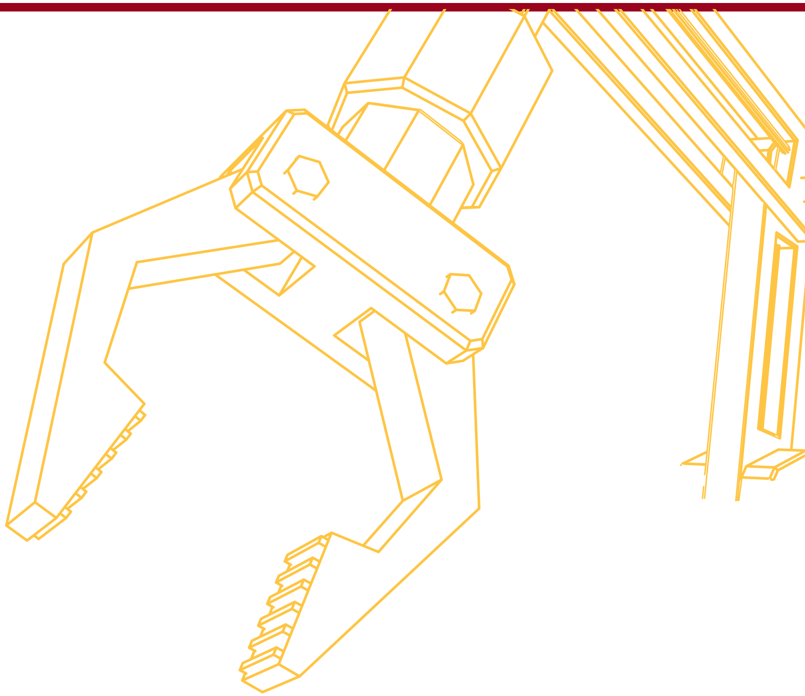
An important and specialised variant of the microprocessor for robot designers is the digital signal processor (DSP).

This is a processor core that adds instructions and execution hardware optimised for signal-processing algorithms such as filters and fast Fourier transforms. Such instructions include fast fused multiply-add operations that are found in practically all DSP algorithms. Because DSP code operates on data structures such as matrices and vectors, it is relatively easy to parallelise the work. This has led to the implementation of single-instruction, multiple-data (SIMD) execution units that perform the same operations – such as multiply-adds – on multiple elements of an array at once. The result is much higher speed for comparatively little additional complexity or cost.

An MCU includes a number of integrated peripherals that are arranged around the processor core. Typically, in an industrial or robotics-oriented MCU, the peripherals range from memory arrays to advanced timer-trigger units, which are used to offload the burden of pulse width modulation (PWM) from the microprocessor. PWM is a core component of almost all motor-control strategies and so features prominently in robotic design. Other system-on-chip (SoC) devices add more features around an MCU such as wireless transceivers, dedicated encryption and authentication logic and graphics accelerators.

The use of intelligent peripherals also illustrates an increasingly important design principle for robots: the exploitation of distributed control and hardware acceleration. A microprocessor can be used to implement PWM control, but it is often a poor allocation of resources. The core of the problem is that the software repeatedly has to switch power between transistors in the half-bridge that controls current flow to a motor after pre-programmed intervals. Interrupts from a real-time clock or counter can readily trigger handlers to switch power state and then configure the timer for the next cycle. But this results in a high interrupt frequency for what is an extremely simple sequence of operations.

A PWM controller combines timer and switching logic, which removes the requirement to have the microprocessor core interrupted for each switching operation. Software only needs to update the timers periodically to set the required



PWM duty cycle. Thanks to a comparatively small amount of additional logic that can operate independently of the processor for substantial periods of time, software efficiency is greatly enhanced. The architecture has a common theme, with other hardware-offload mechanisms that will become increasingly important in robotic design. Hardware peripherals take care of frequent events in real time; software sets policy for those peripherals.

With hardware peripherals, designers are limited to the functions offered by IC suppliers, although the inclusion of sequencers based on hardware state machines increases their flexibility. For example such sequencers can read values from an A/D converter, transfer data values to main memory using direct memory access (DMA), and set and reload timers, all without involving the CPU core. However, the options remain limited.

The field-programmable gate array (FPGA) provides a way to create custom hardware peripherals that can be optimised to specific robotic-control and machine-learning functions.

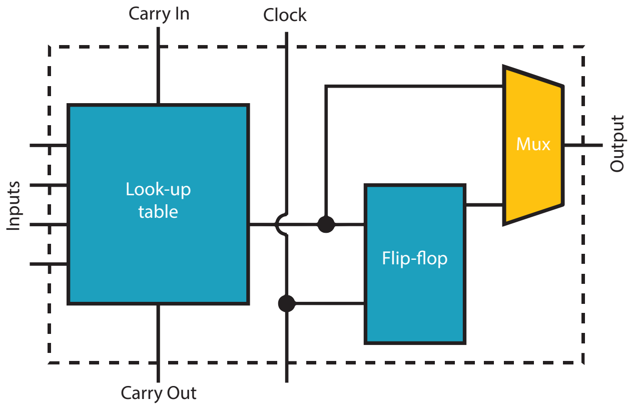
The core of most FPGA architectures is a programmable look-up table that can be configured to implement any logic function that can be expressed as a truth table. Using programmable switches in the interconnect fabric, the look-up tables are wired together into complex combinational logic circuits. Typically, each look-up table is accompanied by one or more registers and additional support logic, such as carry-chain inputs and outputs, to make it possible to implement arithmetic adders efficiently. Together, these functions make up a logic block that is replicated many times across the FPGA.

A drawback compared to fully customised logic is that their silicon efficiency is much lower. It takes 10 to 20 times as much silicon area to accommodate a logic circuit on an FPGA fabric compared to a custom, standard-cell implementation.

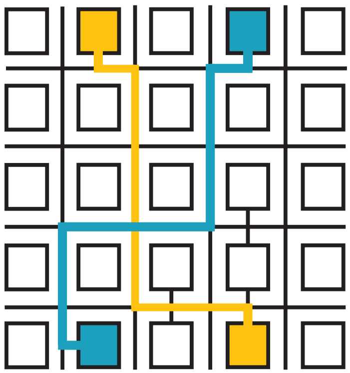
However, most FPGAs support reprogramming of the logic array even in a running system. This makes it possible to share resources by having accelerators dynamically loaded into the fabric only when they are needed. This approach also allows greater flexibility in the end design, enabling it to support new hardware and additional features.

Since their introduction in the 1980s, FPGAs have acquired other features that improve overall density. Memory blocks allow the creation of buffers and caches close to the programmable logic. These have been joined more recently by DSP engines. In many cases, the DSP engines are implemented using a building-block approach, composed of 8-bit or 16-bit units, that allows them to be combined to support higher-precision datatypes.

DSP units make FPGAs highly suitable for processing the inputs from sensors that produce large amounts of data, such as cameras, radar and other types of image sensors. A typical application is to use a combination of DSP units and logic accelerators to handle algorithms such as image warping



The FPGA Look Up Table (LUT) is what gives an FPGA its flexibility



FPGA LUTs are then linked together through a routing matrix to achieve the desired functionality

and lighting compensation that provide more consistent inputs to machine-learning and similar functions. These functions can be coordinated by custom microprocessor cores implemented in the programmable fabric, which act as microsequencers for the different processing primitives.

Another option, particularly for image-processing tasks, is to employ a graphics processing unit (GPU) or vision processing unit (VPU). These contain highly parallelised DSP engines optimised for image processing. For robots that need very high levels of environmental awareness, these dedicated units may be combined with multiple CPUs – sometimes on the same chip, as a heterogeneous multi-core SoC.

The use of multi-processing can also be harnessed to improve overall reliability and safety. A problem for any computer-based design is its reliance on memory technologies that are vulnerable to ionising radiation. When ionising radiation hits the silicon substrate of an IC, it triggers a cascade of free electrons that flip the logic state of a transistor. In combinational-circuit transistors, the effect is usually transitory and captured only rarely. However, memories and registers are more vulnerable to the change because of the way they recycle their contents to prevent stored data leaking away. Error checking and correction (ECC) codes help control the problem. The probability of a single-event upset increases with memory density, which makes it an increasing problem as these ICs continue to scale according to Moore's Law. Also ECC may not catch all of the errors, which can lead to a program acting on incorrect data and, ultimately, a failure in control. In a robot interacting with the public, this cannot be allowed to happen.

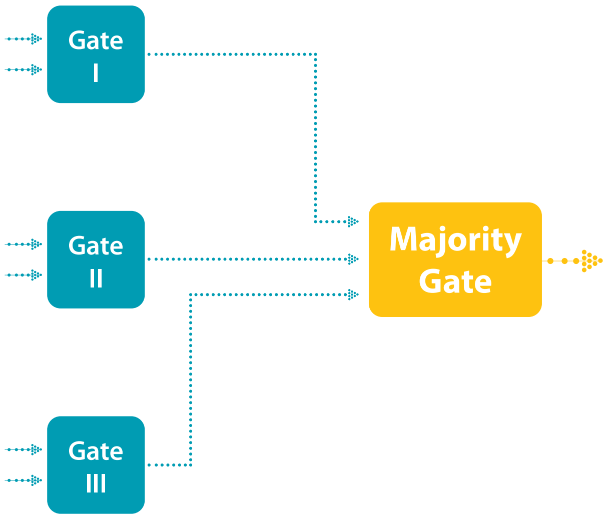
Techniques such as redundancy deal with the problem by having individual processors check each other's work. The processors may be of the same type and run the same code. Checking logic compares their outputs and uses voting to determine which operation to allow or demands that operations are re-run until the processors agree.

The use of three processors with majority voting is more expensive but less intrusive, as re-running operations can incur unwanted delays. Modular redundancy can also be implemented at the gate level.

The processors in a redundant arrangement need not be identical. Some architectures have a less performant processor act as the checking engine. Rather than running the same software, it simply performs consistency checks and forces re-execution if a check fails or, in more extreme cases, a full reset.

To minimise the chances of systematic design errors creeping into the equation, duplicated processors may be designed and implemented in different ways. This is a technique used on some multi-core SoCs developed for automotive safety systems.

The result is that robot designers can now choose from a wide range of architectural options that can take them from simple designs through to highly flexible machines that can react intelligently to problems and obstructions and keep running smoothly.



For applications with high safety or reliability demands, redundancy is built in to mask a failing element

KEY FACTORS TO CONSIDER

01

Microprocessors offer high flexibility but have lower performance than dedicated hardware.

02

Dedicated hardware can be restrictive although integrated MCUs now use microsequencers to coordinate them without processor intervention.

03

By splitting and distributing workloads between microprocessors and hardware, performance can be optimised.

04

Where hardware performance and flexibility are required, FPGAs should be considered.

Software for Robotics

There are multiple levels of software control that need to take place within all but the simplest robots. The microcontroller units (MCUs) and system-on-chip (SoC) solutions responsible for managing sensors and actuators will generally take advantage of a real-time operating system (RTOS) or kernel.

The advantage of using an RTOS is support for multi-tasking. It provides a relatively simple way to schedule numerous activities on a single microprocessor in a way that maximises resources and the ability of the system to react to external events. For example, opening the safety cage needs to trigger a suspension of activities in a way that minimises risk to robot and personnel. Simply removing power is potentially unsafe. An RTOS can trigger all the actions needed to place the robot in a non-moving state but ensure that it does not drop heavy objects or cause damage to anything else. This can be achieved, for example, by moving to a software thread that orders the power electronics circuits to hold motors in predefined positions.

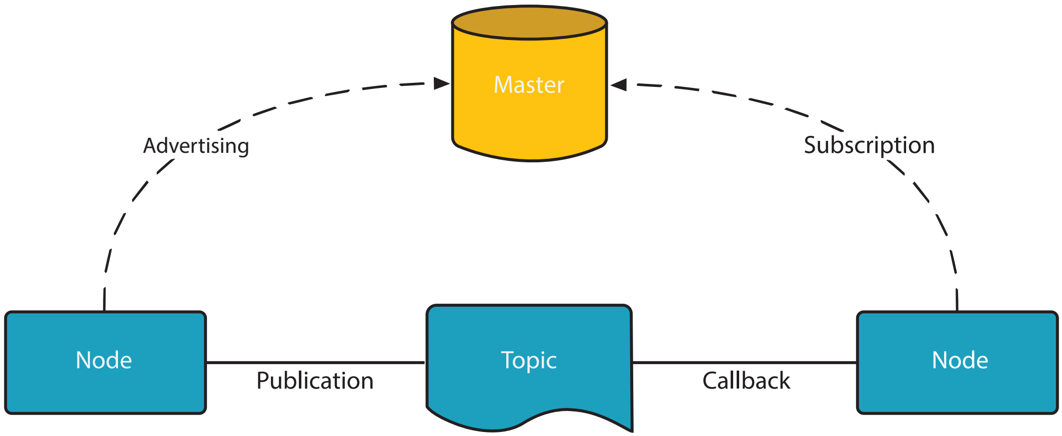
In combination with appropriately designed applications software, an RTOS can provide hard guarantees of the amount of time it takes to react to critical events, which are normally signalled by an external interrupt to the microprocessor. This is normally handled through an interrupt handler which may initiate a software thread that can take action. Through priority-based pre-emptive scheduling, the RTOS guarantees the shortest possible latency for this type of response to the most important issues.

In a robot with multiple microprocessors and hardware accelerators, which is increasingly the case, each of the actuator nodes needs to be controlled by a supervisory system that takes care of task planning and high-level behaviour. This is a role that is typically undertaken by middleware such as the robot operating system (ROS) running on a high-performance microprocessor.

Today, an ROS is designed to run on an operating system such as Linux rather than being an operating system in its own right. ROS also does not demand RTOS behaviour from the underlying operating system as it is performing longer-term tasks than those that need microsecond response times. However, work is underway to build ROS 2.0 implementations that will run on RTOS platforms so that they can offer higher degrees of responsiveness.

The middleware that makes up ROS provides a variety of services. They include hardware abstraction of low-level devices, and support for messages passing between processes to enable multi-processor architectures and the management of software packages. Typically, processes are represented using graphs that link nodes to denote where processing takes place and how the processes communicate with each other. ROS implementations are often open-source packages and make use of Linux platforms to ease the job of managing dependencies between open-source projects. This has the benefit of making ROS software easy to access.

In ROS, nodes are processes or software modules that handle one or more related tasks. For example, a camera and image processing node may process visual data from one or more image sensors. To enable the use of networking infrastructure to interconnect nodes – an architecture now common in automotive systems – ROS supports TCP/IP and UDP for message passing. The various nodes and connections can be described using the Universal Robot Description Format (URDF), which is an XML file format.

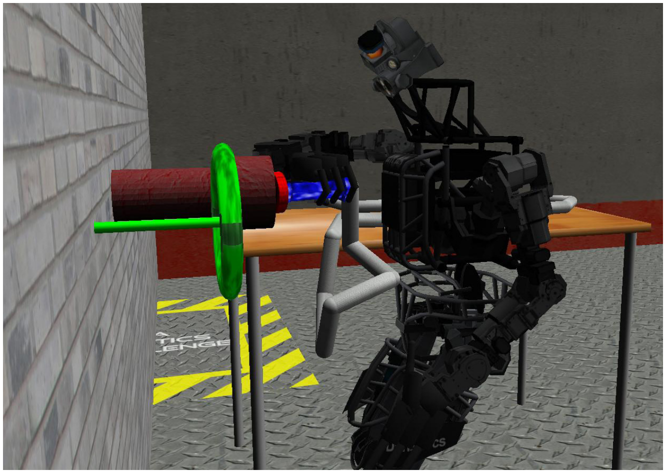


The Robot Operating System (ROS) publish-subscribe mechanism allows nodes to be kept up-to-date on commands or sensor data

To enable efficient sharing of sensor data and commands, ROS employs a publish-subscribe mechanism in which nodes register to be informed of specific topics. Any updates on each topic are sent to all of the subscribed nodes. The ROS Master keeps track of all services and topics. It handles node registration and operates a parameter server to allow nodes to store and retrieve common configuration data.

A major advantage of middleware such as ROS is code reuse and sharing. Code sharing allows all users to have a common base of software, which helps with testing and overall software reliability. ROS is not restricted to physical robots. It also supports simulated robots.

A key requirement of robot design is the ability to simulate its behaviour in the virtual environment before implementation in hardware. The simulator allows for robotics programs to be written and debugged offline. It allows the development of software in a risk-free environment and avoids the problem of damaging the robot or the robot's surrounding environment if the proposed program contains serious errors. The final version of the program can then be tested on an actual robot.



A simulated image of an Atlas robot connecting a hose to a pipe
(The appearance of U.S. Department of Defence (DoD) visual information does not imply or constitute DoD endorsement)

There are further advantages of simulation. Designers can develop in phases, starting with simple high-level models, which is beneficial for complex projects. Such simulations can be used at an early state to establish whether a system is viable. The simulation environments developed for robotics are designed to be compatible with a wide range of programming languages, which supports easy development. And simulation can cut development time as it allows mistakes in application logic to be corrected before they are committed to hardware and so become much more difficult to fix.

There are a number of approaches to robot simulation. Traditionally, simulation was focused on the kinematics of robot movement to demonstrate whether paths and trajectories are feasible and practical.

This type of simulation puts a virtual robot into a 3D space and demonstrates how joints are likely to move in the physical world. The simulation can also help determine whether a robot will be able to lift and manipulate heavy or bulky objects without losing stability.

Some kinematics simulators use a simplified set of calculations and focus primarily on how a program may rotate and move objects to ensure they do not collide with the boundaries of a safety cage or work-cell. Others involve more complex physics simulation to gauge the stresses and other issues that can affect robot performance in the field.

As robots move out of controlled environments protected by safety cages and into areas where people and other robots can move around freely, designers need to take account of possible interactions. For mobile robotics design, simulators that deal with behaviour let designers create, at a high level of abstraction, virtual worlds that contain other objects. A simple behaviour simulation just takes into account the motion of a robot among a set of fixed objects. More complex simulations involve the use of multiple mobile agents or avatars. These behaviour-based simulators help with the design of applications where the robot is likely to be faced with complex environments. They can learn from collisions and other interactions to better deal with obstacles. Physics simulations are important for establishing that the kinematics of the robot are accurately represented.

Simulation environments such as the open-source Gazebo package can generate realistic sensor data that may be corrupted with varying levels of noise. Gazebo makes it possible to tune simulation to the specific requirements of the application – for example by using different physics engines. A maximal coordinate solver such as ODE or Bullet is often chosen when simulating cluttered environments. A Featherstone-based solver like DART or Simbody will find more applications in articulated systems such as humanoid robots or complex manufacturing robots. All of the physics engines are accessed through the same applications programming interface (API).

There are, however, limits to simulation. An application can only simulate characteristics and events for which it is programmed. Internal or external factors are not represented and will not be simulated, which can lead to problems when the design is translated to hardware. It also is often difficult to build sufficiently representative scenarios, especially when it comes to evaluating complex situations and behaviours. However, experience with translating simulated designs into the physical environment can be fed back into future projects, which will reduce errors as time goes on.

As a result, simulation remains one of the most powerful tools in the armoury of the robot engineer.

KEY FACTORS TO CONSIDER

01

Most readily available robot operating system (ROS) implementations do not run on a hard real-time operating system (RTOS), but the components that provide data to the ROS will often need to be based on an RTOS.

02

Engineers can choose from open-source or commercial offerings for ROS, simulation and RTOS options.

03

Simulation environments cover a variety of needs, from basic motion profiles through physics-assisted kinematics to behaviour in complex scenarios.

AI in Robotics

One of the key problems in robot design lies in providing the machine with an understanding of the world around it. The robot needs to be able not only to detect obstacles and dangers but to understand their nature so it can react to each situation appropriately.

A collaborative robot (cobot) is designed to interact with humans in a shared space and, for example, must be able to differentiate between objects that it needs to pick up and move from the people who may be working alongside it.

Although it is possible to build rules-based models that guide the motion of an autonomous system, it has proven difficult to engineer these systems to be robust and effective in the complex situations that they are likely to face in applications such as cobots in the factory or warehouse environment, or delivery robots. Machine learning provides an alternative path to achieving a solution. It has been demonstrated in numerous applications, from drones that can follow paths through a forest, to self-driving vehicles that are reliable enough to be allowed to run in trials on city streets.

The key application for machine learning in robot design is that of perception – providing the robot with the ability to react appropriately to the input from cameras and sensors that image the 3D landscape around it. Sensory artificial intelligence provides the robot with the ability to recognise objects in the surrounding environment. Using that understanding, the robot can use pattern matching to learn appropriate behaviours from past experience. And it may learn new situations as they arise through reinforcement-learning techniques.

AI is becoming more present in our daily lives. Devices like Amazon's Alexa, Google's OK Google and many other web services depend on these complex algorithms, which are run on servers in the cloud. Robot designers will turn to similar approaches both through improvements in hardware performance and the ability to offload some of their processing to the cloud.

Since its inception over 50 years ago, there are now many approaches to the concept of machine learning. The fundamental link between all machine-learning technologies is that they take in data, train a model on that data and then use the derived model to make predictions on new data. The process of training a model is a learning process where the model is exposed to unfamiliar data at each step and is asked to make predictions. Feedback from these predictions in the form of an error term is used to alter the model so that, over time during the training process, the model improves.

Often the model adjustments made for new data will worsen performance on prior samples. So it takes multiple iterations over the training set to achieve consistent performance. Typically, training stops when the predictions of the model reach a point at which the error does not improve – which may be a local or, ideally, a global minimum. As a result, machine learning has strong links to optimisation techniques such as linear regression in which a curve is fitted to a set of data points.

There are many machine-learning algorithms available. An important distinction is between supervised and unsupervised learning. In the latter case, the model is provided with unlabelled data and asked to segment the elements into groups. A common algorithm used for this purpose is k-means clustering. The algorithm works iteratively to assign each data point to one of a number of clusters. The algorithm does this by first estimating centroids for each cluster – often by an initial random selection – and then refining its model based on the distance between data points from each other until it determines the most likely clustering.

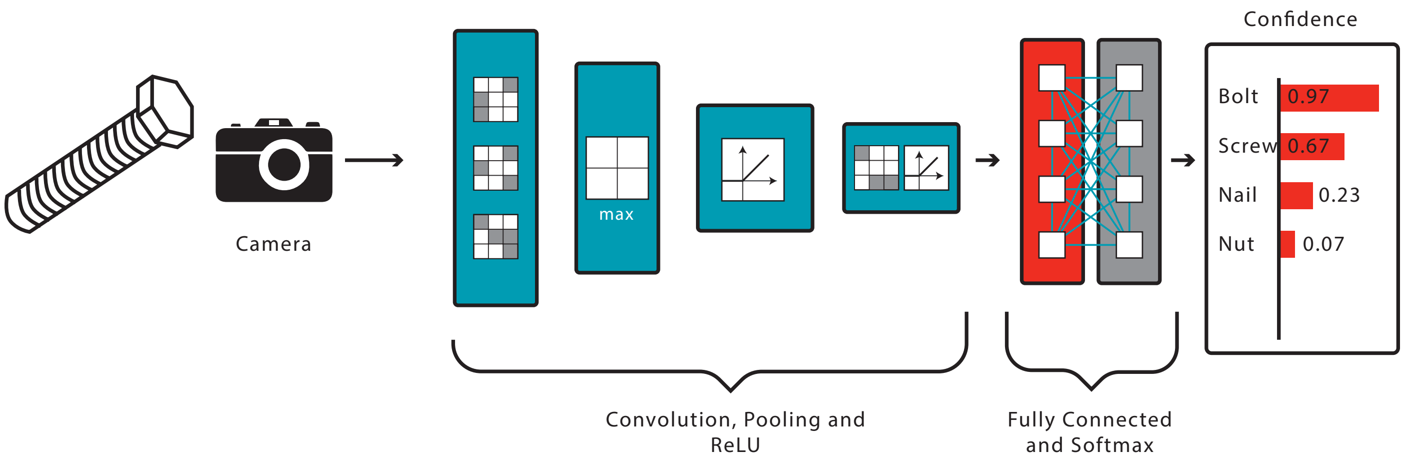
> In robotics, k-means and similar unsupervised clustering approaches have been used to support the automated mapping of unknown spaces by groups of robots. However, for perception-based tasks, supervised learning is currently the most common form of machine learning being applied in research and production robots.

Until recently, one of the most successful techniques for image-recognition tasks was the support vector machine (SVM). This technique is similar to clustering but works with data that has been labelled into two or more classes. The job of the SVM is to determine the parameters that will allow the model to place unlabelled data into the most appropriate class. Although SVMs were used in research for applications such as autonomous vehicles in the late 1990s and early 2000s, their use has largely given way to deep learning.

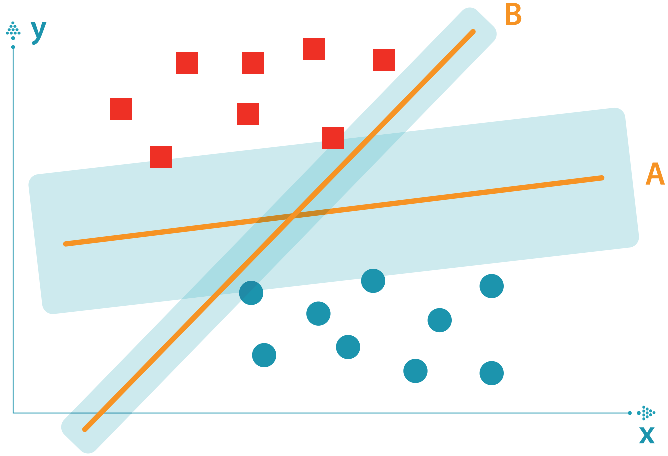
Deep learning is a modification of the artificial neural network (ANN) technology that was highly publicised in the 1980s and 1990s, which itself drew on theories developed more than half a century earlier, which were inspired by the biology of the animal brain. In a traditional ANN design, artificial neurons are arranged in a small number of layers – an input, an output and a hidden layer. Each neuron in the hidden layer takes in data from every neuron in the input layer, performs a weighted sum and applies an activation function, such as the hyperbolic tangent or logistic function, before passing the result to the output layer.

Training of the network is typically performed using backpropagation, an approach to optimisation and error reduction that works from the output back to the input – giving the technique its name. Backpropagation calculates the gradient of the error. This gradient is used to perform gradient descent in an attempt to find a set of weight values that are more likely to reduce the error during each epoch of training. This approach to ANN showed early promise. But the need for intensive computing resources to perform backpropagation and its inability to compete with the SVM meant that ANN slipped into relative obscurity. That situation began to reverse with a reinvigoration of deep networks – ANNs with more than one hidden layer – that were first proposed in the 1960s but which foundered because optimising the network weights proved extremely difficult.

A key development was the application of a more efficient approach to training and backpropagation developed by Geoffrey Hinton and Ruslan Salakhutdinov, working at the University of Toronto in the mid-2000s. The development was aided by the massive improvement in compute performance compared to the early 1990s, first with multi-core CPUs



A deep learning platform delivers a statistical likelihood of the object captured by the camera being the desired object, in this case a bolt and then with GPUs. Increases in model performance came with the application of refinements to the fully connected architecture that had been proposed over the previous two decades. One was to introduce convolutional layers interspersed between fully connected layers.



Two possible partitions for a data set that support vector machine (SVM) training might yield

Convolution is a matrix operation that applies a feature map to an array of data – pixels in the case of image recognition.

The feature map can be regarded as a filter. Convolutions of this kind are frequently used in image processing to blur images or to find sharp edges. They also provide a way of converting data in a spatial domain to a representation based on the time domain, where waves are superimposed on each other to form the overall image. As a result, convolutions make it possible to convert pixel arrays into collections of features that can be worked on independently by the following layers.

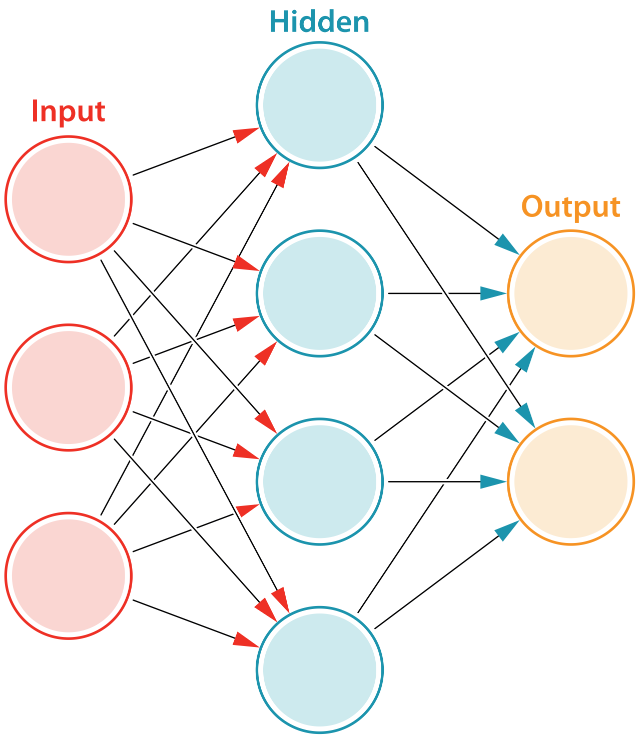
In contrast to the conventional use of convolution in image processing, the feature maps are learned as part of the ANN training process. This makes it possible for the model to adapt to differences in the training set that make it easier to distinguish between examples. For example, feature maps tuned to detect differences in shape will be most appropriate for general image-recognition tasks. Feature maps optimised for colour will be favoured in situations where the objects to be separated have similar shapes but are differentiated by their surface attributes.

One major advantage of the convolutional layer is compute efficiency. It is easier to implement in an ANN as it employs far fewer connections per neuron than fully connected layers, and maps readily to GPUs and other parallel-processing architectures with single-instruction, multiple-data (SIMD) arithmetic units. Another attribute of convolutional layers is that the design resembles the organization of neurons in the visual cortex of the organic brain, which is different to the more highly connected regions used for cognition.

Multiple convolutional layers are often used in series in deep-learning architectures. Each successive layer filters the image for increasingly abstract content. In a convolutional neural network (CNN), a set of convolution layers is often followed by a pooling layer. These pooling layers combine the outputs from multiple neurons to produce a single output – producing a sub-sampling effect – that can be fed to multiple inputs in the following layer. This pooling has the effect of concentrating information and steering it to the most appropriate set of neurons that follow. The benefit of their use is that they improve the performance of recognition operations on images where important features may move around within the input. For example, a person's face may move around in the image field as the robot approaches. Pooling layers help ensure that features activated by the shape and colour consistent with those of a face are steered towards neurons that can perform a more detailed analysis. Training on images in which faces are offset and rotated helps build the connections between the most appropriate neurons.

There are different kinds of pooling operations. A max-pooling layer, for example, takes the maximum value from the inputs and passes that on. The highly influential AlexNet entry to the ImageNet LSCVRC-2010 contest employed these structures. AlexNet comprised five convolution layers, three fully connected layers, and three max-pooling stages.

A further improvement to training performance came with the adoption of stochastic gradient descent (SGD) as the mechanism for calculating gradient during backpropagation. This was primarily a choice made for computational efficiency, as it uses a small sub-set of the training data to estimate gradients. However, the random-walk effect of SGD helps move the optimisation towards a good global minimum faster and more frequently than with previous techniques.



The basic neural network consists of several inputs, a hidden layer, and several output nodes

Not long after deep-learning architectures were first employed, researchers at IDSIA in Switzerland showed that the machines could outperform humans on recognition tasks. In one experiment, the CNN could correctly identify heavily damaged road signs because it was able to make use of visual features that humans would normally ignore. However, this ability to make use of non-obvious features can be a weakness with current approaches based on ANNs.

> Poor selection of training materials can cause the network to train on elements that will lead to mistakes in the field.

Researchers have found in recent years that, simply by changing a single pixel in an image, the network will provide the wrong classification. Analysis of the weights chosen by one CNN indicated that, in trying to classify cats, the network had learned to use unrelated markings in some of the training images as part of the identification. Networks will also sometimes claim a successful classification for an image that is only noise.

The architecture of the CNN should be chosen to fit the application. There is no one-size-fits-all architecture. Decisions as to the number and ordering of convolutional, pooling and fully connected layers have a strong impact on performance. And the feature map and kernel sizes for each of the convolutional layers provide trade-offs between performance, memory usage and compute resources.

The classical feedforward architecture of the basic CNN is far from being the only option, particularly as deep learning moves from classification tasks to control. Feedback is becoming an element of the design in applications such as voice recognition. Recurrent neural networks use feedback loops. Memory networks make use of elements other than neurons to hold temporary data that can be used to store contextual information that is likely to be useful in applications that call for a degree of planning, which may include systems that control robot behaviour and motion. Another option is the adversarial architecture, based on two linked networks. The competition between them helps avoid the risk of a single network making fundamental mistakes. As the technology continues to develop, we can expect other novel architectures to emerge.

Supervised learning is different from the organic experience in that training and execution occur in different phases: the network does not typically learn as it runs. However, in order to ensure the system is able to meet new challenges, it can be important to perform training sessions on recorded data, particularly if the system flags them as situations that led to errors or poor performance.

For control of the core robot functions, reinforcement learning is often employed. This rewards the robot during training for 'good' behaviour and penalises poor decisions. In contrast to simple image-classification tasks, forward planning is a vital component of the process. This calls for the use of discounting techniques to tune rewards for decisions made in a given state. A discounting factor of 0.5, for example, will be just one-eighth of its value after three state changes. This will cause the machine-learning network to pursue near-term rewards. A higher discounting factor will push the network to consider longer-term outcomes.

A key question for designers of robots is where training occurs. The separation of training and the inferencing needed during execution provides an opportunity to offload the most compute-intensive part of the problem to remote servers.

Inferencing can take place in real time using less hardware while servers perform training updates in batches overnight. The cloud environment provides access to standard tools such as Caffe and TensorFlow that can be used to design, build and test different CNN strategies.

With a hardware platform optimised for inferencing, designers can take advantage of some features of CNN architecture to improve processing efficiency. Typically, the backpropagation calculations used during training demand high-precision floating-point arithmetic. This keeps errors to a minimum. The processes of normalisation and regularisation work to reduce the size of individual weights on each neuronal input. These steps are needed to prevent a small number of nodes developing strong weights that reduce overall performance.

As a result of normalisation, some weights will reduce to very low levels and, in the optimisation process, reduce to zero. In the runtime application, these calculations can be dropped entirely. In many of the interneural connections with low significance, the weighted-sum calculation can tolerate increased errors from the use of low-precision fixed-point arithmetic. Often 8-bit fixed-point arithmetic is sufficient. And, for some connections, 4-bit resolution has been found to not increase errors significantly. This favours hardware platforms that offer high flexibility over numeric precision. A number of microprocessors with SIMD execution units will handle low-precision arithmetic operations in parallel. Field-programmable gate arrays (FPGAs) provide the ability to fine-tune arithmetic precision. An upcoming generation of coarse-grained reconfigurable arrays (CGRAs) optimised for deep learning will provide an intermediate solution between microprocessors and FPGAs.

They will help improve performance and make AI-enabled robots and cobots more feasible.

KEY FACTORS TO CONSIDER

01

Machine learning provides a way of building more advanced sensory and control systems for robots compared to traditional path- or rule-based control strategies.

02

Use an appropriate machine-learning algorithm. Deep learning is not necessarily the right answer for all situations.

03

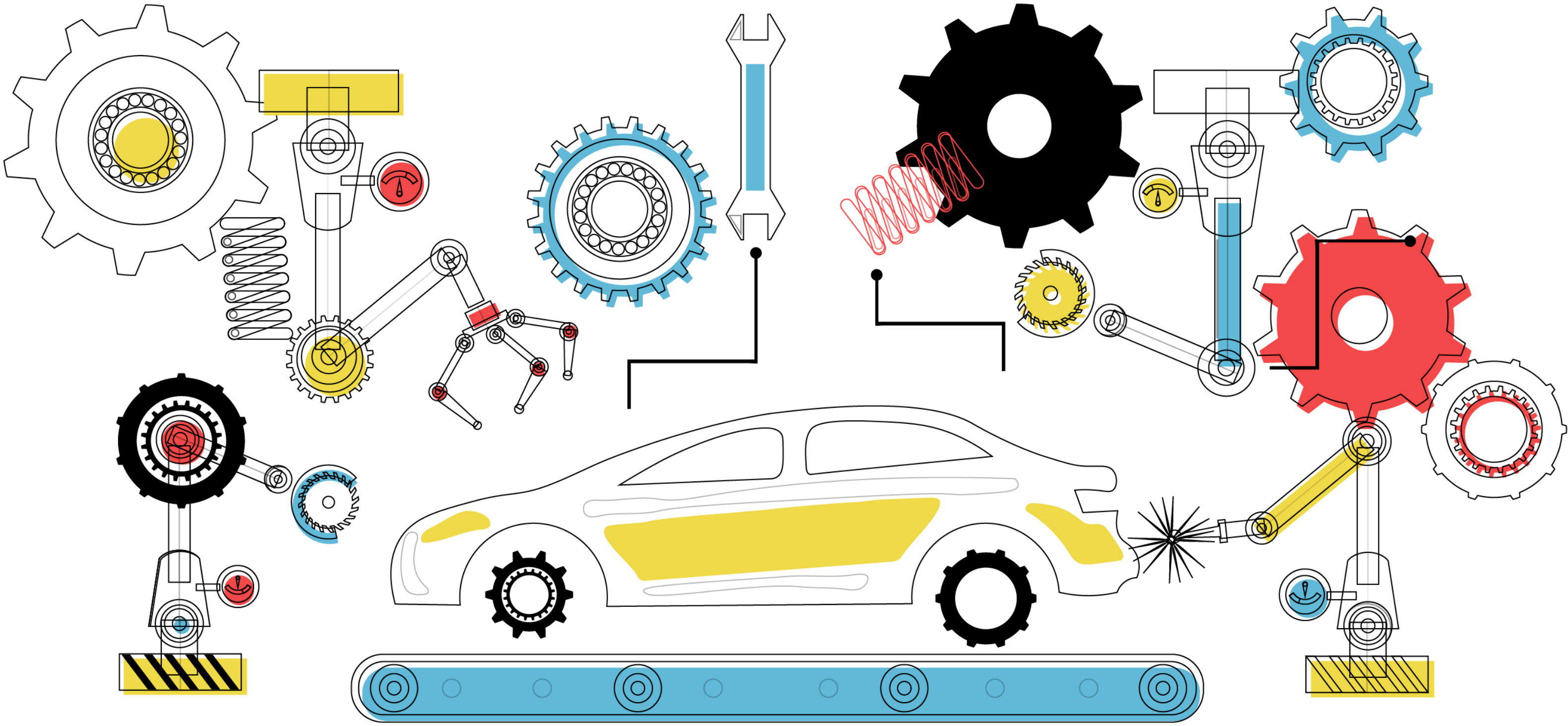
Training and inferencing are separate processes. This can be leveraged by offloading the more compute-intensive operations to the cloud.

04

CNNs can be deployed in many forms. The architecture of the CNN is intimately tied to the type of data it is expected to learn and process.

05

Training data quality is vital. Poor selection of training data can lead to unexpected results.



Effectors

A robot's primary means of interaction with its environment is the effector. The effector is controlled by the robot and can be in the form of wheels, fingers, a tool or any physical construction that makes the robot act within its working envelope.

Effectors are among the most important, most customised parts of any robot, and the robot's purpose defines the requirements for the effector, or effectors.

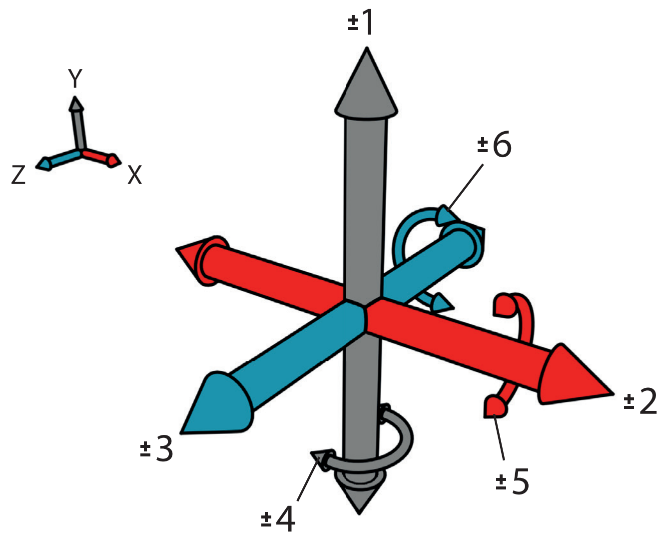
In industrial robotics the end effector is typically either a gripper or a tool. Here the common effector types and their applications and design considerations are described.

BASIC CONCEPTS

In order for actuators to perform a physical movement in an environment an effector is needed to transform (electric) energy into movement. Most simple actuators only control one degree of freedom (DOF), typically linear or rotational.

The task the robot needs to perform defines the system requirements for the effector and, indirectly, the DOF needed. The greater DOF an effector has, the more sensors are needed and the more complex the required programming becomes.

The DOF of a robot can be considered a system constraint. The DOF number of the overall system includes the DOF of the robot plus the additional DOF of the effector. Yet most effectors only add a single DOF to the overall system.



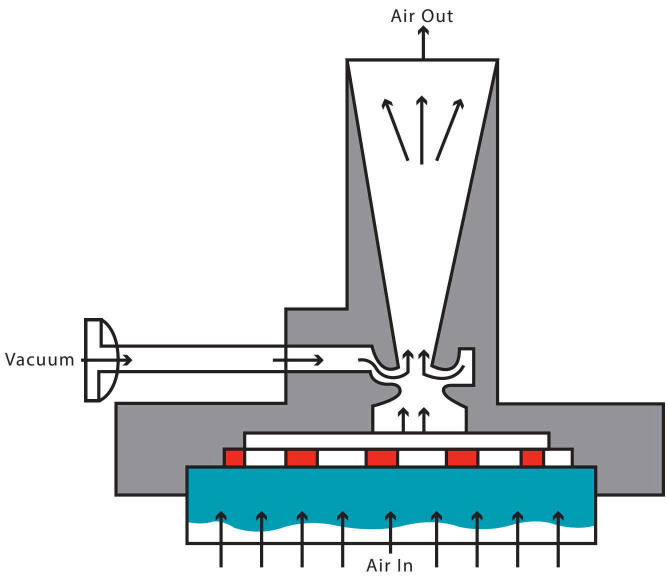
The degrees of freedom (DOF) of an effector defines how flexible it will be in tackling the task at hand

GRIPPERS

The most common effector types in robotics are grippers. Grippers are end effectors that aim at picking, holding and placing an item. Grippers can be unilateral or multilateral, which refers to the number of contact points between the gripper and the target object. There are four general categories of robot grippers:

- **Impactive** – mechanical jaws that use form closure or force closure
- **Ingressive** – pins or any other construction that penetrates the surface of the object (often used for textile and CRFP handling)
- **Astrictive** – suction by means of vacuum, magnetism or electroadhesion
- **Contigutive** – adhesion through direct contact using glue or surface tension

Impactive (mechanical grippers) apply the principle of force closure or form closure to pick and hold items. Force closure grippers are typically multilateral, while form closure grippers can be unilateral.



Suction cups are well suited to handling objects with planar surfaces, such as glass

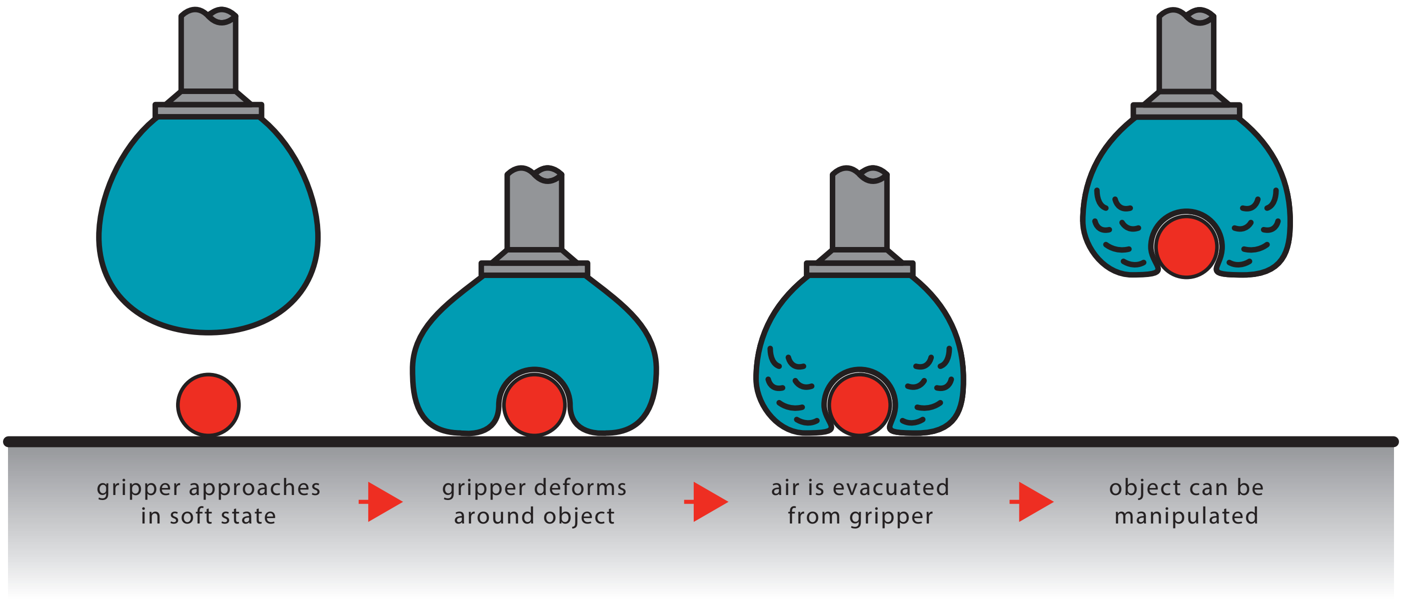
The actuation of those mechanical grippers can be divided into four different mechanisms, all of which require only one actuator:

- **Linkage actuation** – via joints and linkage the lateral movement is transformed, making the grippers close
- **Gear and rack actuation** – a lateral movement is transformed into two counter-rotating rotational movements, shifting the gripper together
- **Cam actuation** – a conical shape is shifted between two ends of a gripper that are fixed, allowing for rotation
- **Screw actuation** – rotational movement is transformed into lateral movement

Mechanically actuated grippers are typically actuated with the means of an electric motor or servo. Other sources of actuation could also be pneumatics or hydraulics, where the source of the activation would come from a cylinder that provides linear movement, which is then transformed into the necessary final movement using linkages.

Types of astrictive grippers include suction and vacuum grippers. These are typically of unilateral nature, meaning they only create one point or surface of contact. They are mechanically rather simple but must be customised according to their specific application.

Suction cups are used to lift objects with simple, mostly planar, surfaces. The suction cup is evacuated of air during the time the object should be bound to the gripper. When the object has to be released or placed, air valves are opened to release the vacuum.



Soft and deformable vacuum grippers can handle a variety of objects

Simple astrictive grippers include adhesive grippers which use only one contact surface, with the advantage that no constant suction is needed to maintain the adhesion. Instead, only lightweight materials can be lifted, and the gripping reliability obviously diminishes over cycle time.

Another concept of grippers with a more universal use is the soft and deformable general-purpose vacuum gripper. This concept can be considered a hybrid between impactive and astrictive. A deformable spherical shape filled with granular material is placed over the object to be lifted. The gripper shape deforms and wraps around the object and then air is evacuated from the gripper head. This makes the gripper head stiff and creates an almost perfect form closure, so the object can be lifted.

TOOLS

Among the most commonly used industrial end effector types are welding torches. Formerly welding robots were general-purpose industrial robots with a high payload capability that were equipped with a welding torch as end effector. But with the increased demand for welding robots, manufacturers have developed distinct welding robot systems that are smaller and cheaper.

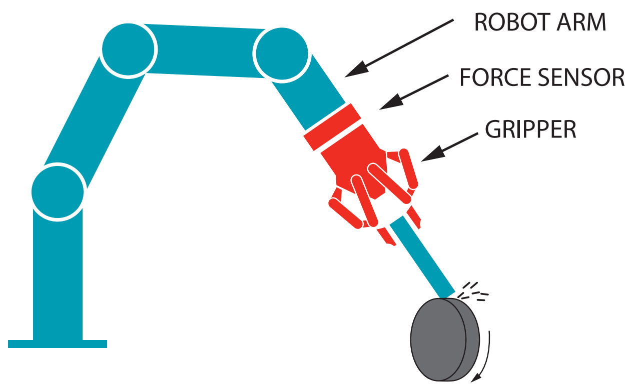
Other types of tools for end effectors include power wrenches, drilling modules and rivet fasteners to name a few.

For production areas where maximum flexibility is required a multi-function end effector (MFEE) can be used. Such MFEEs offer many of the tools mentioned above in one unit, but this comes at a price in terms of volume, weight and general sturdiness.

Depending on the task, a robot head can simply consist of a set of sensors. A robot's head repeatability and accuracy make it ideal for product quality inspection. For example, a camera head with additional laser sensors can check a product for the correct dimensions and surface finish, whereas 3D scanners can measure and digitise the full-size object.

SENSORS

Sensors play a very important role in robotics. But apart from the sensors that are built into the robot, the effectors often come with their own set of sensors that are essential for the robot to safely and accurately perform its tasks. The most commonly used sensor is the 'force torque sensor'. As the name suggests, this device is able to sense forces in all three axes and torque on all three axes. This makes the FT sensor (as it is commonly known) a sensor for all six DOF. The FT sensor is typically placed between the end effector and the wrist of the robot. This way the FT sensor can accurately measure the force exerted on an object.



Force torque sensors enable robots to handle tasks previously only possible by humans, such as deburring and grinding

CONCLUSION

The most commonly used end effectors are grippers. As the requirements for grippers vary with the shape and nature of the object that has to be collected, grippers are often highly customised. Other end effectors are specific or multi-purpose tools for drilling, welding, screw fastening, grinding, to name a few. As the role robots play in production processes attains more importance, robot manufacturers are likely to focus on developing application-specific robots that are cheaper and lighter as opposed to general-purpose robots with customised effectors.

What Does the Future Hold for Robotics?

Although the manufacturing sector is still the biggest purchaser of robots, there is now huge commercial opportunities arising for 'service robots'. This is in part driven by the number of new start-ups, which currently account for approximately 29% of all the robot companies in operation.

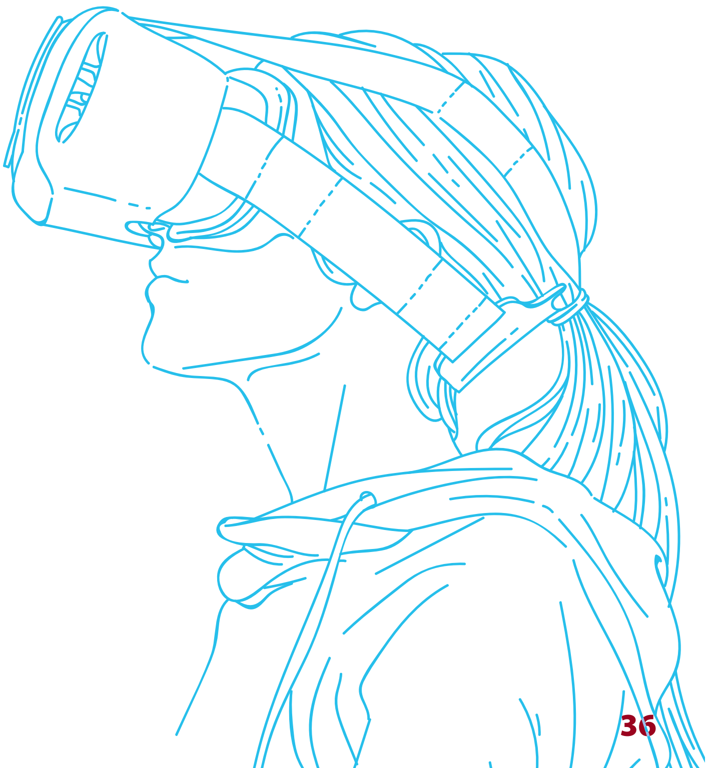
ISO 8373 defines a 'service robot' as a robot 'that performs useful tasks for humans or equipment, excluding industrial automation applications'. According to ISO 8373, service robots require 'a degree of autonomy', or the 'ability to perform intended tasks based on current state and sensing, without human intervention'. The growth in service robots is based on the emergence of 'cobot' (which were discussed earlier). These enabled by innovations in AI and sensor technology. There have been huge advances recently in deep learning, a branch of AI involving building artificial neural networks which attempt to mimic the way organic (living) brains sort and process information. Deep learning is driving innovation at the cutting edge of AI and it can be seen in many applications today, including speech and image recognition.

Healthcare and process manufacturing are high-growth sectors in robotics, with spending here increasing dramatically. Medical diagnostics is one particular field where AI is enabling rapid advances, with prospective applications including:

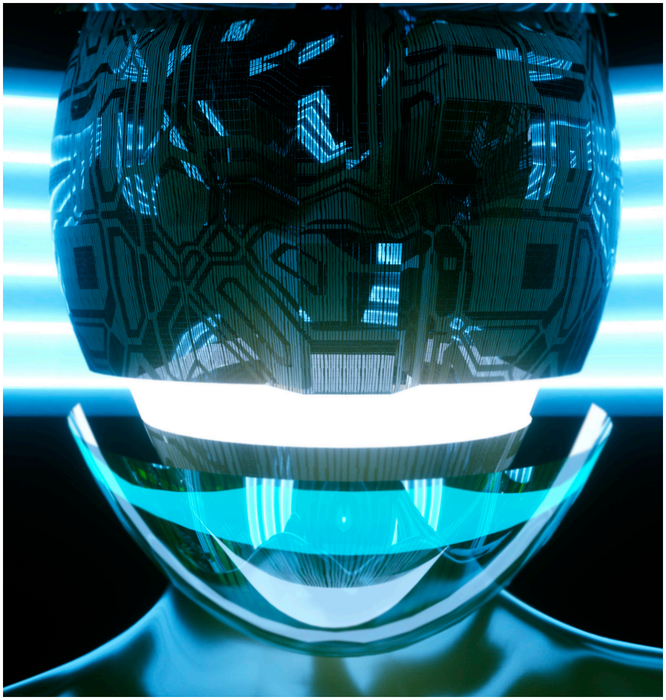
- **Chatbots**: Companies are using AI chatbots with speech recognition capability to identify patterns in patient symptoms and make a potential diagnosis.
- **Oncology**: Researchers are using deep learning to train algorithms to recognise cancerous tissue at a level comparable to trained physicians.
- **Pathology**: Pathology is concerned with the diagnosis of disease based on the laboratory analysis of bodily fluids such as blood and urine, as well as tissues. Machine vision and other machine-learning technologies can enhance the efforts traditionally left only to pathologists with microscopes.
- **Rare diseases**: Facial recognition software is being combined with machine learning to help clinicians diagnose rare diseases. Patient photos are analysed using facial analysis and deep learning to detect phenotypes that correlate with rare genetic diseases.

Service robots are also having a significant impact in areas such as agriculture, surgery, logistics and underwater applications, and the maintenance, security and rescue market.

This section examines the emerging developments in robot technology and the future applications that will be enabled by this. The impact of these innovations on our economy, our society's values and our day-to-day life are also considered.



In Germany, industrial automation is one of ten 'Future Projects' outlined in the government's High-Tech Strategy, to which it has contributed around €200 million of investment. This funding supports a deep base of academic researchers, with projects covering most areas of robotics innovation. Over a dozen major academic institutions are now engaged in various aspects of robotics research, including the Institute of Robotics and Mechatronics, a branch of the German Aerospace Centre. The Institute is developing various types of robot, designed to operate in areas that are inaccessible or dangerous to humans, as well as to support humans in their everyday lives and work. Another German research facility leading the way in robotics is the DFKI Robotics Innovation Centre, which also focuses on technologies for various challenging and dangerous environments, including space and underwater, as well as safety, mobility and cognitive robotics.



PROJECTS, APPLICATIONS AND ENABLING TECHNOLOGIES

A growing number of companies, both start-ups and established players, are developing service robots, with applications across a number of industries. Medineering GmbH, a start-up based in Munich, has developed a collaborative robotic concept that specifically targets the needs of surgeons. Today, surgeons often require the support of a sufficiently trained assistant to position and hold instruments during surgical interventions. The Medineering solution consists of an easy-to-guide Positioning Arm with a mechatronic interface at its end, allowing a variety of surgical robots to be attached to it. The first robot Medineering offers is an Endoscope Guidance Robot, which holds the endoscope during endo- and transnasal interventions.

The Positioning Arm, with its seven joints, can be positioned exactly as required using the touch buttons on its surfaces. Once in place, the position can be saved. Fitted to the end of the arm is the Endoscope Guidance Robot, which is then controlled by the surgeon using a foot pedal during the operation. The robot is designed to make fine movements of low intensity, enabling tremor-free positioning and accurate avoidance of sensitive tissue.

This approach relieves the surgeon or assistant from the task of holding the endoscope, so that surgical resources can be used more efficiently. This reduces the time of such interventions or releases medical support staff for other duties, which leads to cost savings. The Medineering Positioning Arm is also open for approved partners to develop further surgical robots to be fitted, making it useful in other areas of surgery.

In the field of hazardous environments, Forth Engineering Ltd's prototype robot has been developed to address a

The Medineering GmbH Positioning Arm and Endoscope Robot enable a surgeon to operate alone, freeing up medical staff for other activities (Courtesy of Medineering GmbH)

critical issue concerning the treatment of nuclear waste at the Sellafield nuclear site in Cumbria. Forth is collaborating with the University of Manchester to leverage the university's world-leading research in nuclear waste management while developing its robotic 'spider'. The 'spider' is actually a six-legged robot, about the size of a coffee table, and bristling with cameras and sensors, enabling it to see its environment. A large front-mounted pincer grabs and breaks up contaminated material. The robot can crawl up walls using magnets on its feet, and AI software allows a team of the robots to work autonomously, communicating with one another and making their own decisions on how to best complete a task.

Industry 4.0 has been termed the next phase of the Industrial Revolution, characterised by the 'smart factory', in which cyber-physical systems monitor the factory processes and make decentralised decisions. Whereas a next industrial revolution is, the earlier mentioned, Industry 5.0 defines collaboration between humans and robots.

Collaboration – inter-robot and between robots and humans – is a key enabler of the smart factory, allowing people and robots to each contribute their unique strengths: people providing insight and improvisation, and robots offering speed and repetition.

Retailer Ocado's prototype ARMAR-6 robot is an example of this type of collaborative robot, or 'cobot'. ARMAR-6 has been developed with support from four European universities. The robot has a human-looking torso, arms with eight degrees of freedom, hands that can grip and a head with cameras inside. Designed to help engineers fix mechanical faults in its factories, ARMAR-6 uses a three-camera system inside its head to help it detect and recognise humans and objects. Speech recognition helps it understand commands and its hands are able to pick up and grasp objects. The ultimate goal is for the robot to be able to decide what the technician's intentions are and contribute as appropriate at the right point in time.

ETHICAL AND MORAL CONSIDERATIONS

ARMAR-6 collaborates with people in its surroundings, supporting them with maintenance work (Courtesy of Ocado)

Advances in AI and robotics have rekindled two long-standing fears: that machines will cause mass unemployment, and that there will be a *Terminator* scenario in which robots will 'wake up' and do unintended things.

A fierce debate is underway with concerns expressed by, among others, the late Stephen Hawking and Elon Musk, the technology entrepreneur. The more optimistic viewpoint is that throughout history technology has created more jobs than it has destroyed, albeit with some disruption to society, such as the migration of jobs to the cities during the Industrial Revolution.

As ever the truth will probably lie somewhere in between. While AI may not cause mass unemployment, it will certainly create disruption in the labour markets, requiring workers to learn new skills more quickly than in the past. This view is supported by a recent study on the German economy, which has shown that employees in operations deploying robots tend to keep their jobs and benefit from upskilling. At the same time these operations have reduced the number of new hires, leading to a shift in employment away from manufacturing and into other sectors.

A recent report from Citi, produced in conjunction with the University of Oxford, has highlighted the risk of increased automation leading to greater inequality as AI impacts on traditional blue-collar jobs more than white-collar jobs. This highlights the important role of companies and governments in making it easier for workers to acquire new skills and switch jobs as needed. Citi identifies investment in education as the single biggest factor that could mitigate the impact of increased automation and AI.

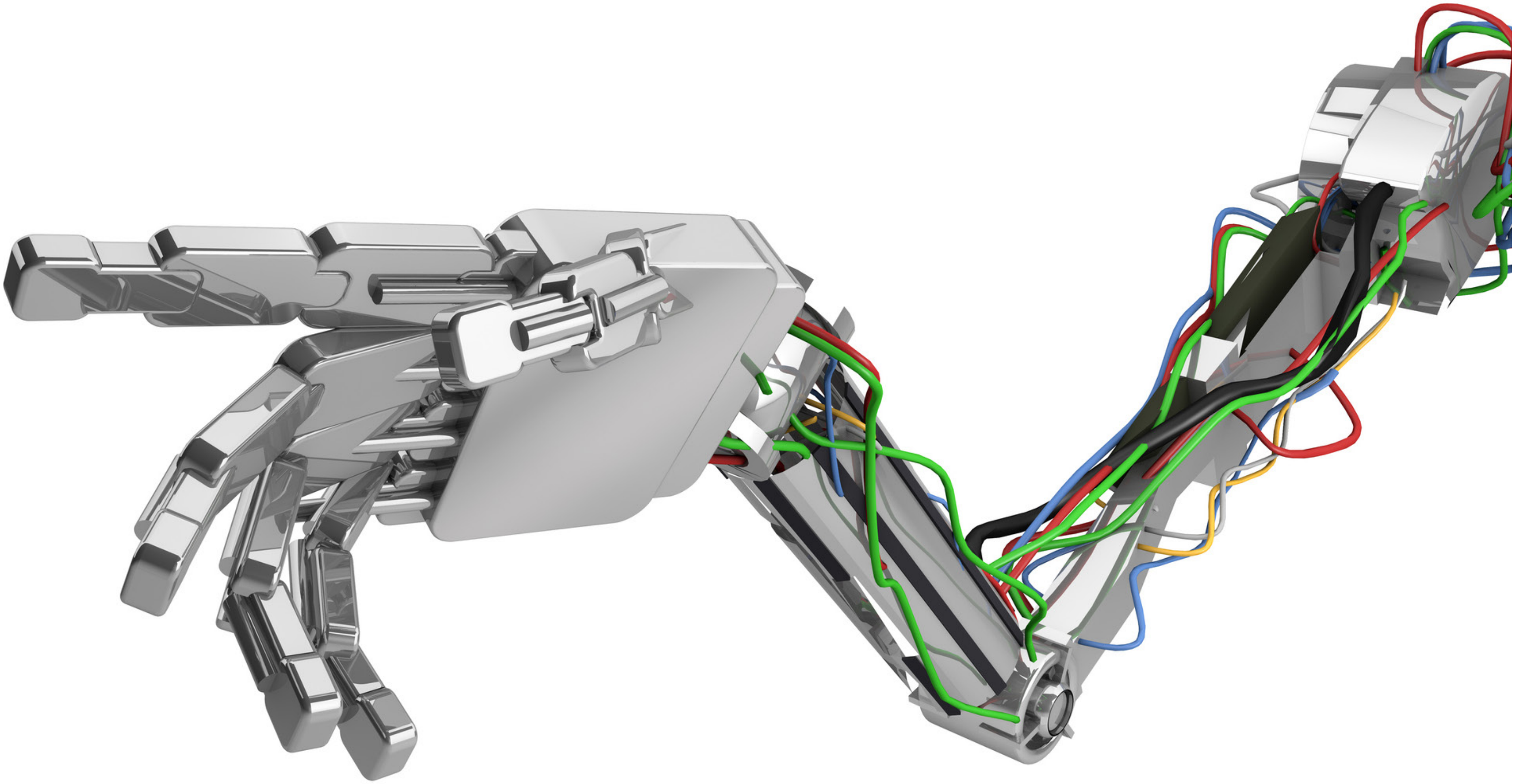
As with any tool, AI has the potential to be used in in both benign and sinister applications. While deep learning has the potential to fight crime it can also allow authoritarian governments to spy on their citizens. Self-driving cars raise other ethical issues, particularly when it comes to how they should behave in emergencies – should the vehicle risk injuring its occupants to avoid hitting a child who steps out in front of it?

It was questions such as these that led to the setting up in 2012 of a two-year, $2.3 million project called RoboLaw, largely funded by the European Union. Consisting of experts in areas such as law, engineering, philosophy, regulation and medicine, the outcome of the project was a recommended set of guidelines for the regulation of robotics.

CONCLUSION

Rapid advances in AI supported by a surge in investment have contributed to a growth in the number and types of robotic applications now being realised. Service robots are set to play an increasingly important role in our day-to-day lives, facilitated by their flexibility and enhanced decision-making capabilities.

With analysts and commentators divided on how these developments will impact upon our society, the consensus is that conscious intervention is required from governments to regulate and manage the disruption to labour markets which will be caused by this technology.



Our thanks go out to…

FOR HELPING WITH THE CREATION OF A GUIDE TO **ROBOTICS AND AUTOMATION**, WE EXTEND OUR **THANKS** TO THE FOLLOWING PEOPLE AND ORGANIZATIONS:

Matthias Heutger – Deutsche Post DHL Group

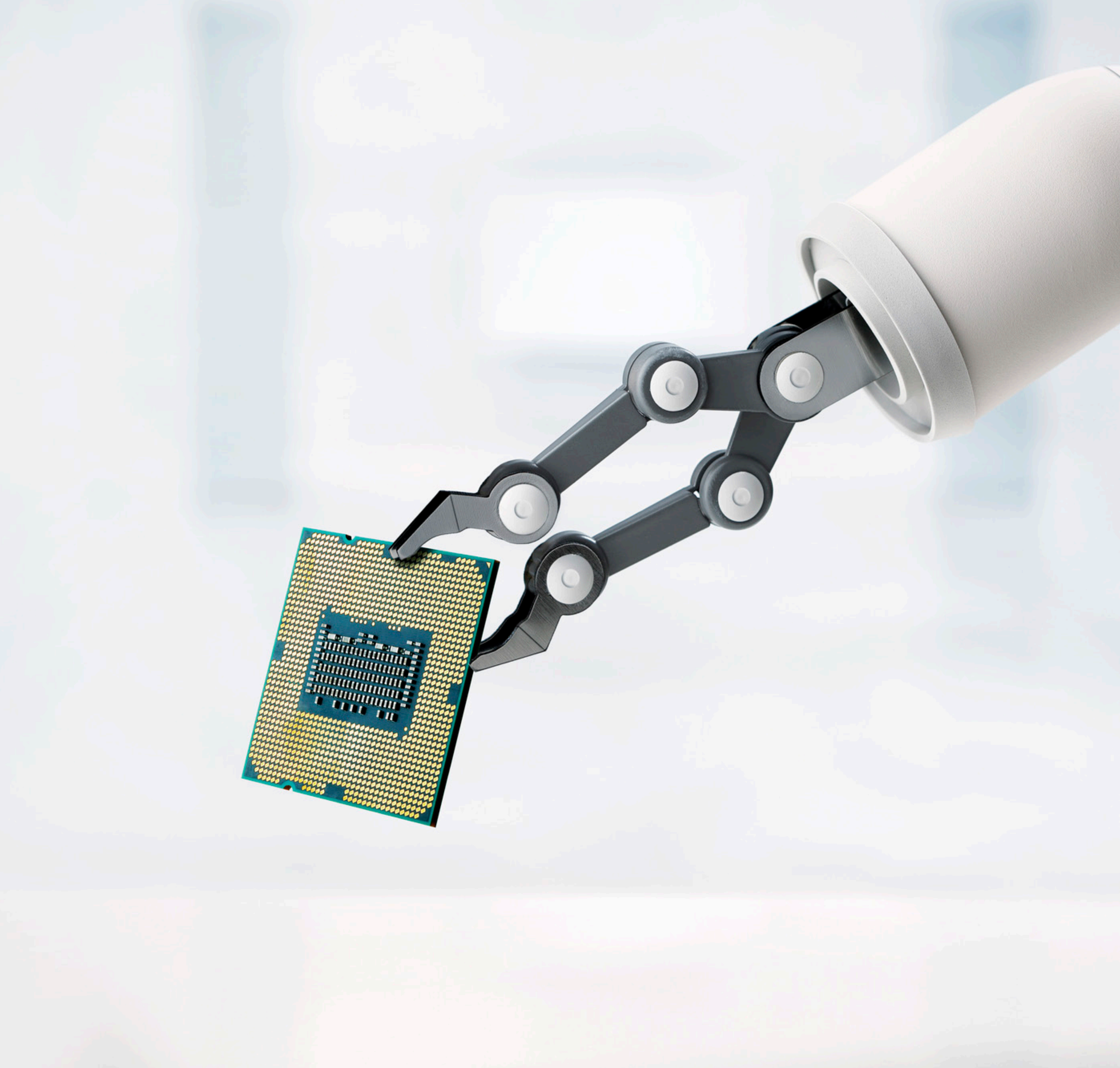
The Media Relations team at Trumpf GmbH

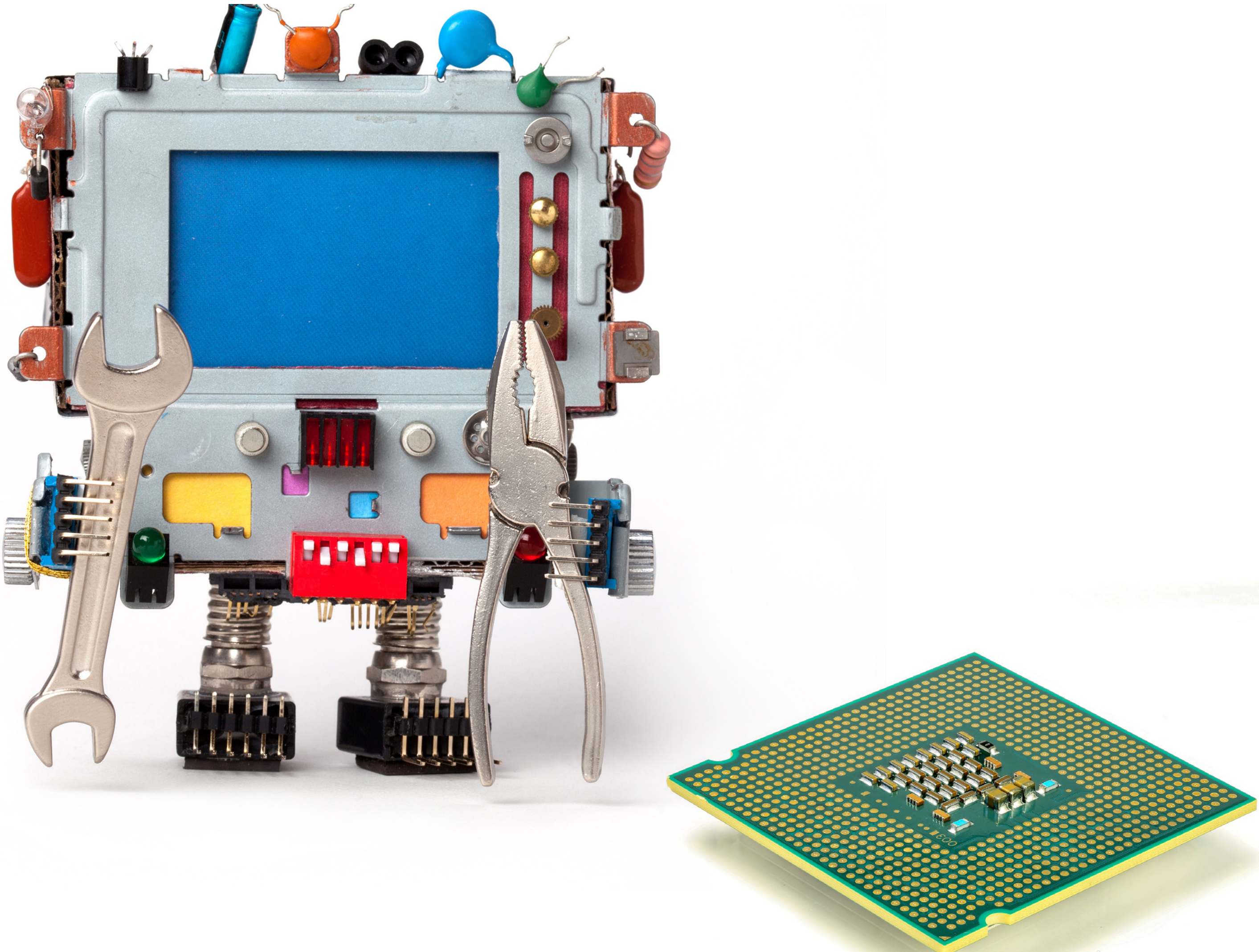
The Technology PR and Communications team at Ocado Technology

The Marketing Team at Medineering GmbH

Publitek

The Marketing Team at Distrelec





KNOWHOW