ROBOTIQ

FIND THE IDEAL FORCE TORQUE SENSOR FOR YOUR APPLICATION

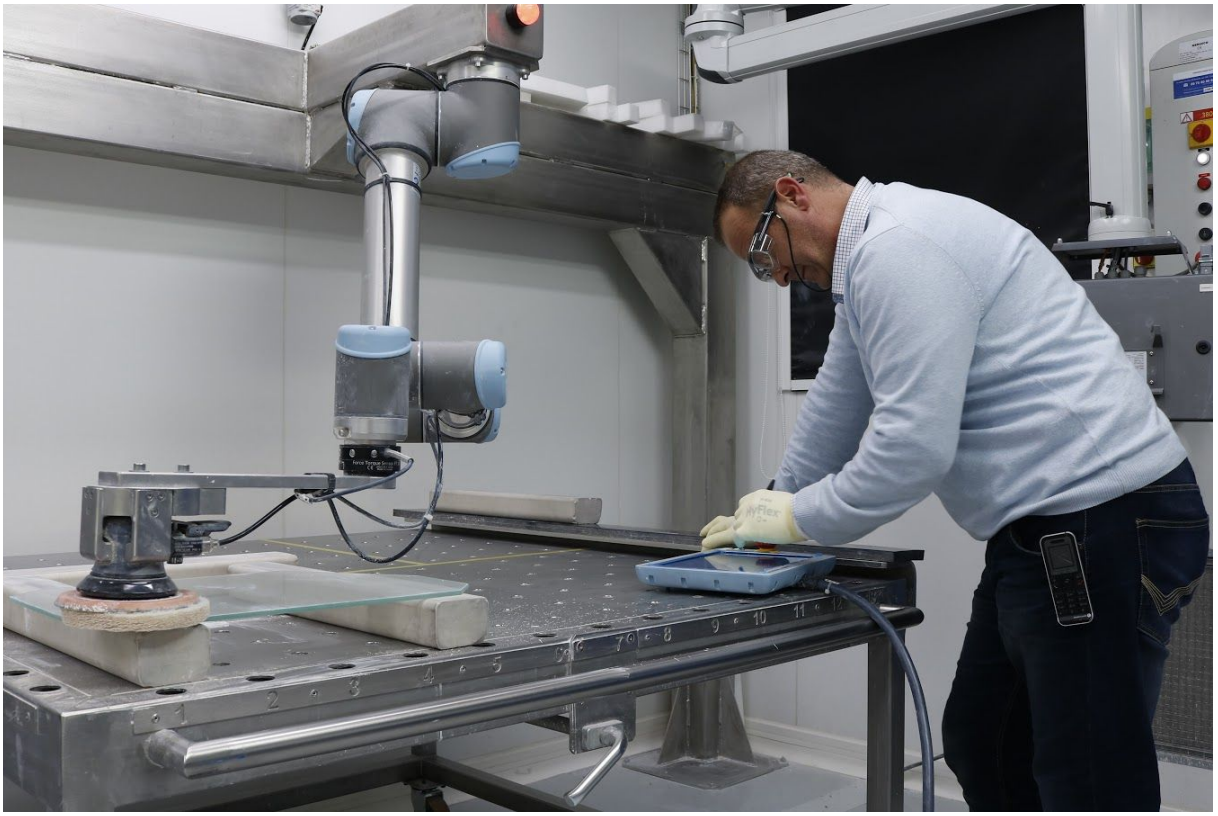


Table of contents

Lean robotics: simplify robotic cell deployments

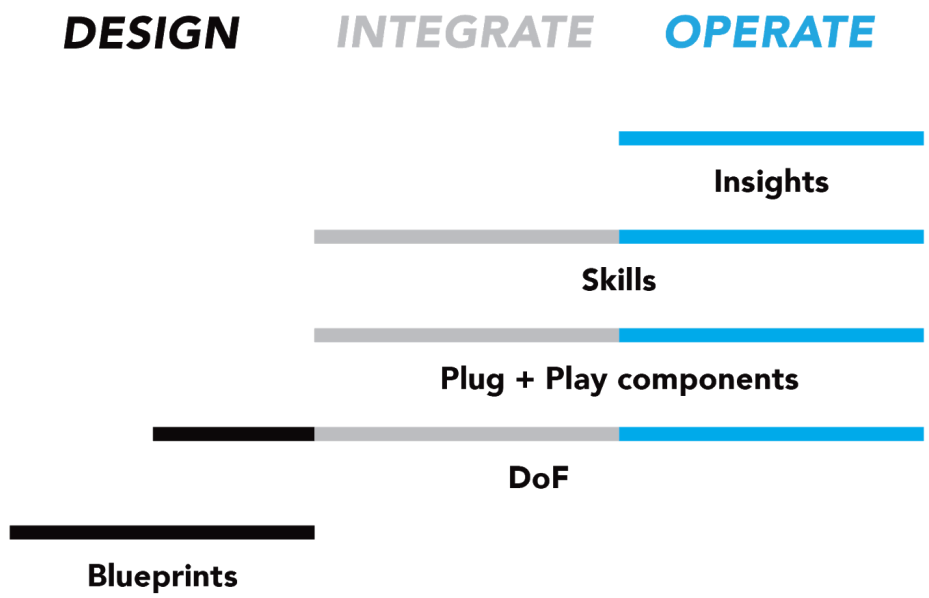
When you ask whether robots could work in your factory, the answer you'll get is probably a hesitant "It depends." It depends on your factory, your team, which robot you choose, what you want it to do... and a whole lot more.

If you're a first-time robot user, how can you get started? How do you get from your initial idea to a productive, working robot? And if you've already got a few robotic deployments under your belt, how can you scale up your robotics efforts throughout your factory—or across multiple factories?

The answers can be found in **lean robotics: a methodology for simplifying robotic cell deployments**.

Lean robotics is a systematic way to complete the robotic cell deployment cycle, from design to integration and operation. It will empower your team to deploy robots quicker and more efficiently than ever before.

Lean robotics divides robotic cell deployments into three phases: design, integrate, and operate.



Robotiq's library of eBooks covers each phase of robotic cell deployment, giving you access to advice from robotics experts each step of the way.

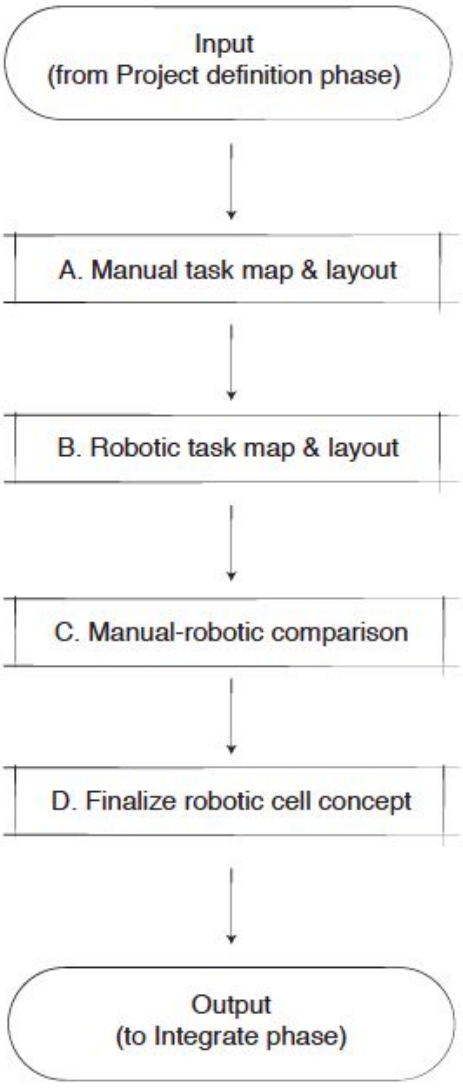
Learn more about lean robotics at **leanrobotics.org**.

This eBook covers the design phase

DESIGN

The design phase includes all the tasks needed to go from a manual process to having the plan and materials for the robotic cell.

This step involves mapping the robotic equivalent of your process and comparing your current manual cell task map to your planned robotic cell map. By the end of this step, you'll have a validated concept and plans for your robotic cell.



Most robots lack the sense of touch

Whether it's for machine tending, assembly, packaging, finishing, or any other application, robots are wonderful tools for reducing costs while increasing productivity and quality.

Of course, there are different kinds of robots and setups. On one hand, there are robots that require fences and cages around them so humans can't get too close. On the other hand, there are robots that can operate in the same workspace as human workers. These are known as collaborative robots, or cobots for short.

I actually don't like using the term "collaborative" because to me, it's somewhat misleading. You can't just buy a robot that's built for collaborative applications, and then declare that you have a cobot so nothing more needs to be done. Instead, the entire robotic cell has to be considered – and indeed, "collaborative robotic cell" sounds much better, in my opinion.

That being said, no matter what type of robotic cell you have implemented (or are about to implement), all robots have limitations. Since robotic arms lack a sense of touch, the object they need to pick up or assemble must always be in exactly the same place.

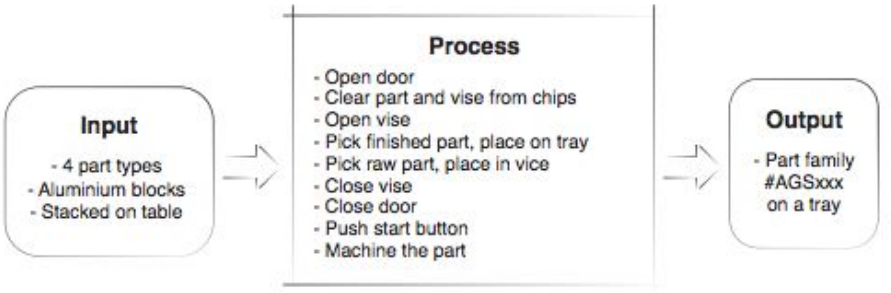
Most robots, out of the box, cannot even pick up a part, since they don't come with grippers. That's why there are so many end-of-arm tooling and accessory options available, like grippers and suction cups to grasp and handle parts, and 2D or 3D cameras to locate parts and perform inspections.

Do you know what else is an added accessory? Force torque sensors. These devices sense forces and torques applied to the robot's tool. They're useful for applications like assembly, placing parts in a jig, machine tending, probing to locate a part, indexing parts together, and checking if a package is full or empty. There are also many possibilities for what a sensor can do when mounted on a robot arm, but we'll get to that later.

If you haven't deployed robots in your factory yet, a force torque sensor may not seem useful at first. You already know you can design jigs and program a robot to move to those exact positions in order to pick parts, so why would you need sensors?

The eBook will show you why.

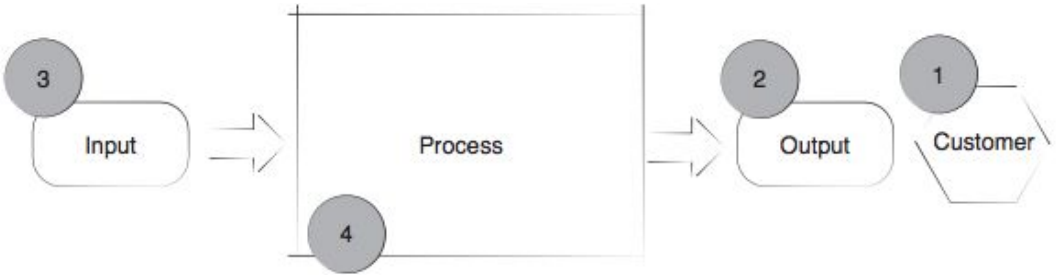
Before we get to that, it's important to understand exactly how a human operator performs the task you want to automate. This is your "manual task map," and it will guide your automation process.



What is the operator doing, aside from handling parts while performing the task? Are they quickly glancing at each part before handing them off to the next station? Are they using their sense of touch to insert a part, put it in a jig, or assemble parts properly?

Another thing that can help with completing the manual task map is to close your eyes and try to perform the task – or one step of it – with one hand.

Do you need to to use any feedback from your sense of touch in order to pick, manipulate or place the part? Do you need to open your eyes? Answering these questions will help you decide whether you'll need any sensors for your robotic cell.



This eBook focuses on how to choose a force torque sensor for your particular application.

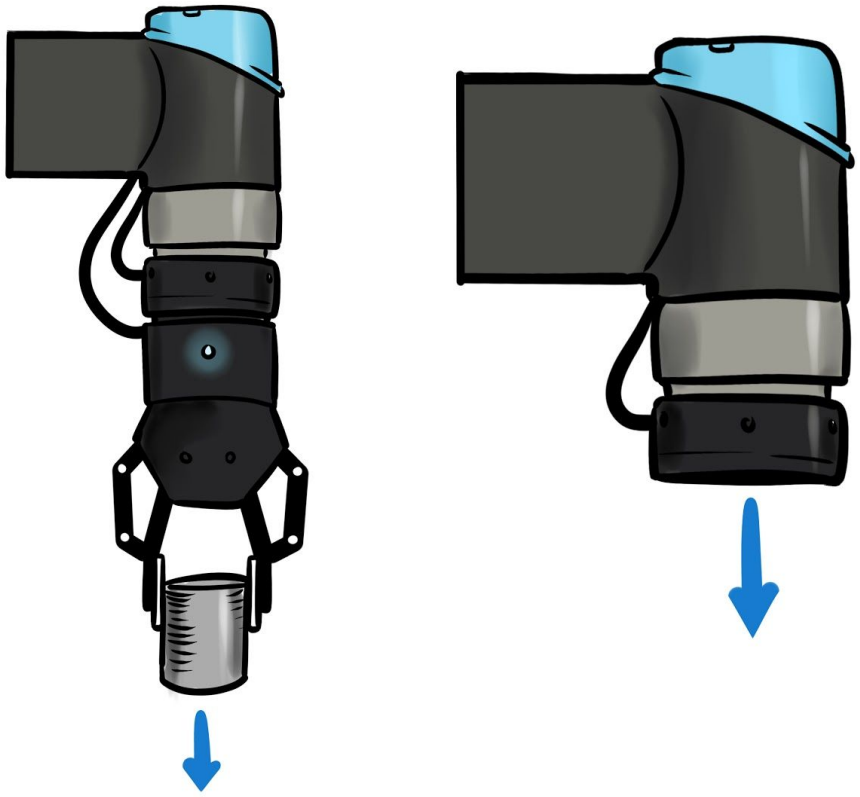
As the *Lean Robotics* methodology suggests, you should always start with the application when designing your robotic cell. This extends to the integration as well as the choice of components.

With this in mind, we'll explore which information from your application should affect your choice of force torque sensor. We'll cover tips and tricks as we go along, and finish up with some programming best practices for force torque sensors. Finally, if you're still not convinced you need a sensor, we'll show you numerous examples of all the things you can do with it. Ready? Let's design your robotic cell!

How sensors work

Before we get into applications and how to choose a sensor, let's get up to speed on the basics. The following information is aimed at people who've never worked with sensors before (so it may seem obvious to you if you have an engineering background).

In all robotic end-of-arm force torque sensors, there is a sensing element and a fixed element. This is true for all sensing technologies: the mobile casing or plate moves relative to the fixed one, and this displacement is translated into forces and torques.



When a force sensor is mounted on an end-effector, all forces applied to this gripper or tool will be sensed by the sensor.

There are many types of sensing technologies, including strain gauges, capacitive sensors, resistive sensors, and optics-based sensors. But although these are interesting devices, we won't go into much detail about them here.

The point of this eBook is to help you choose the right sensor – which is simply a matter of comparing your needs with what each sensor has to offer in terms of technical specifications, lifespan, and performance. Don't worry, we'll go over each of these points together!

Start with the application

First up, we'll look at some applications and tasks that sensors enable robotic cells to perform.

We'll then go through the specifications of sensors, so you'll know what to look for when choosing one. We'll also see some features that are not always part of the technical specifications, but that can make-or-break your sensor decision. Throughout all this, we'll offer tips and tricks on how to use your sensor.

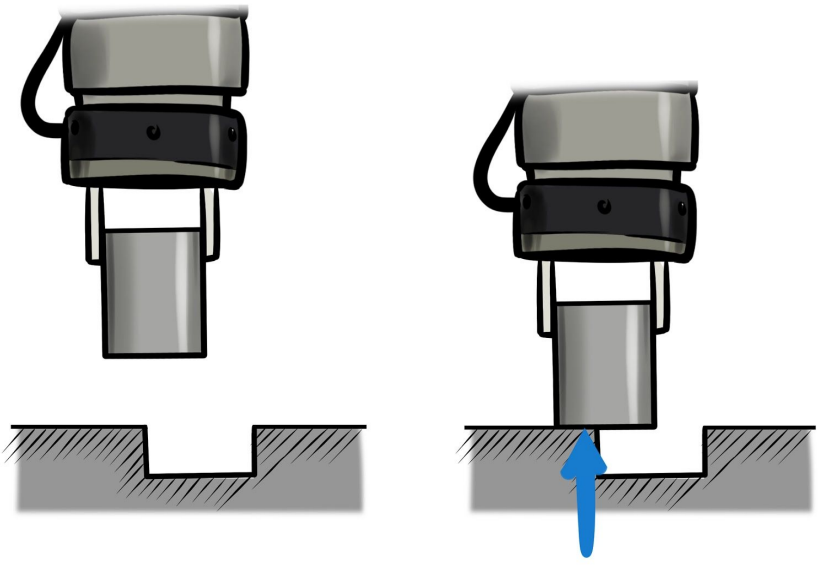
What sensors can do

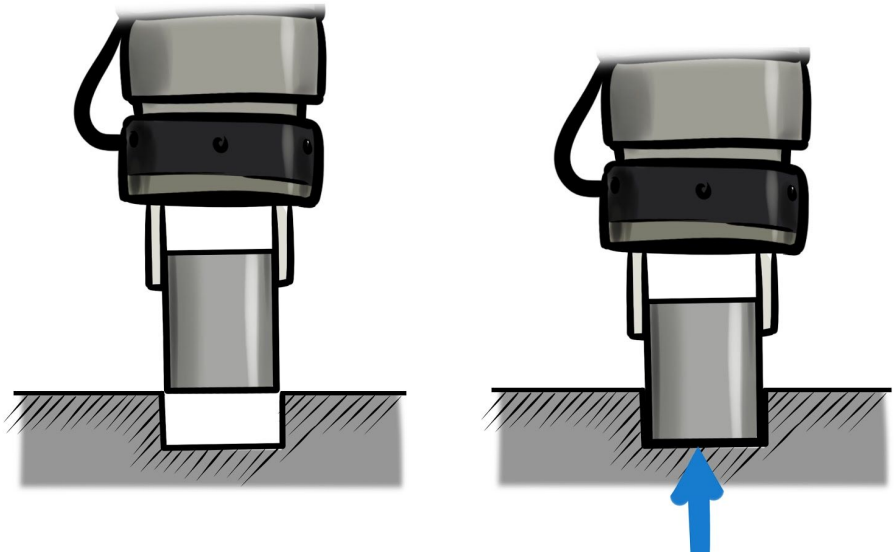
So many applications can benefit from a force torque sensor, because the essential function of a sensor is to enable tasks to be performed.

Example 1: Inserting a part

The figure below depicts one example of a sensor in action.

It's an insertion application, and the arrows indicate which forces are felt by the sensor. The figure also shows some strategies that can be used to incorporate force feedback when inserting the part.





The first step of a part-insertion application is programming the robot to move downwards until it makes contact with a surface. If it hasn't reached the bottom of the insertion area, the robot should adjust its position laterally until it senses a drop in contact force.

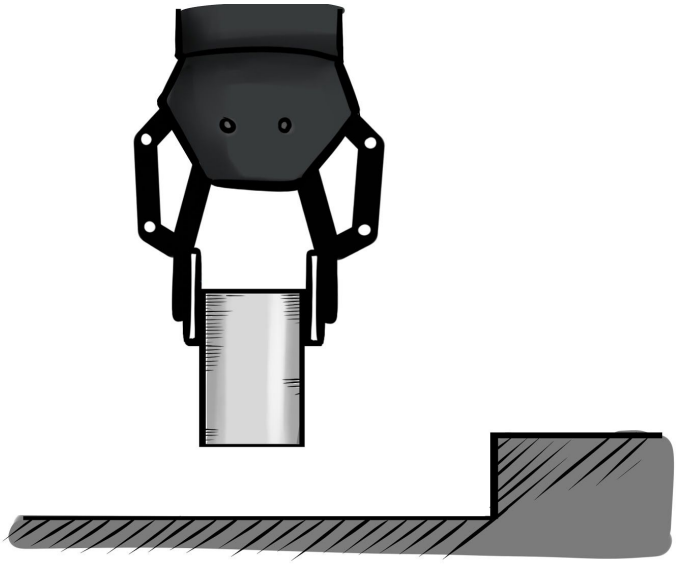
In other words, the robot should keep moving across the surface until it finds the hole. It should then move downwards to insert the part, and stop when it detects a high contact force (from reaching the bottom of the hole).

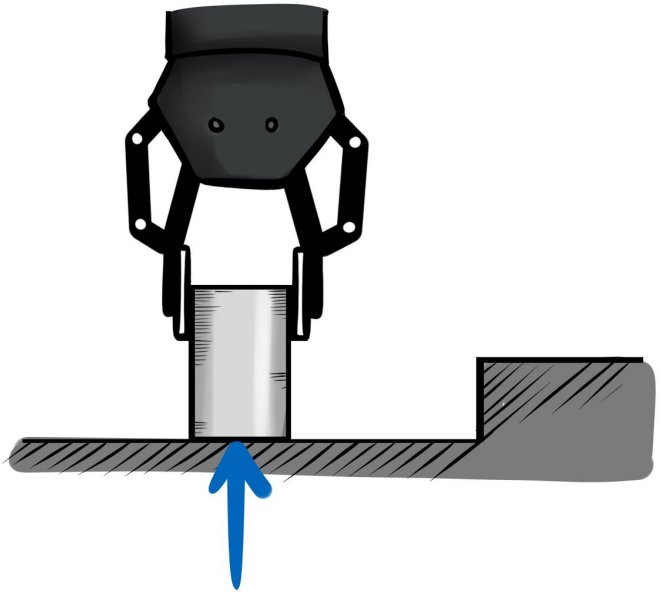
As you can see, a sensor alone is not enough to perform a complete assembly application. In addition to the robot and sensor, you need a gripper to hold the part, and possibly a camera (which is technically another type of sensor) to detect where the part is located.

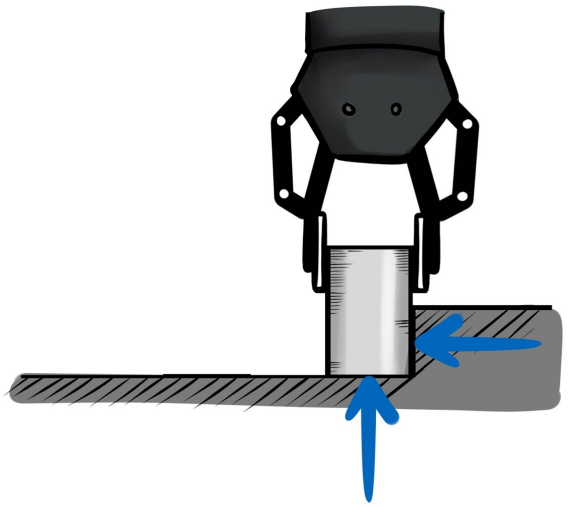
Example 2: Placing a part in a chuck

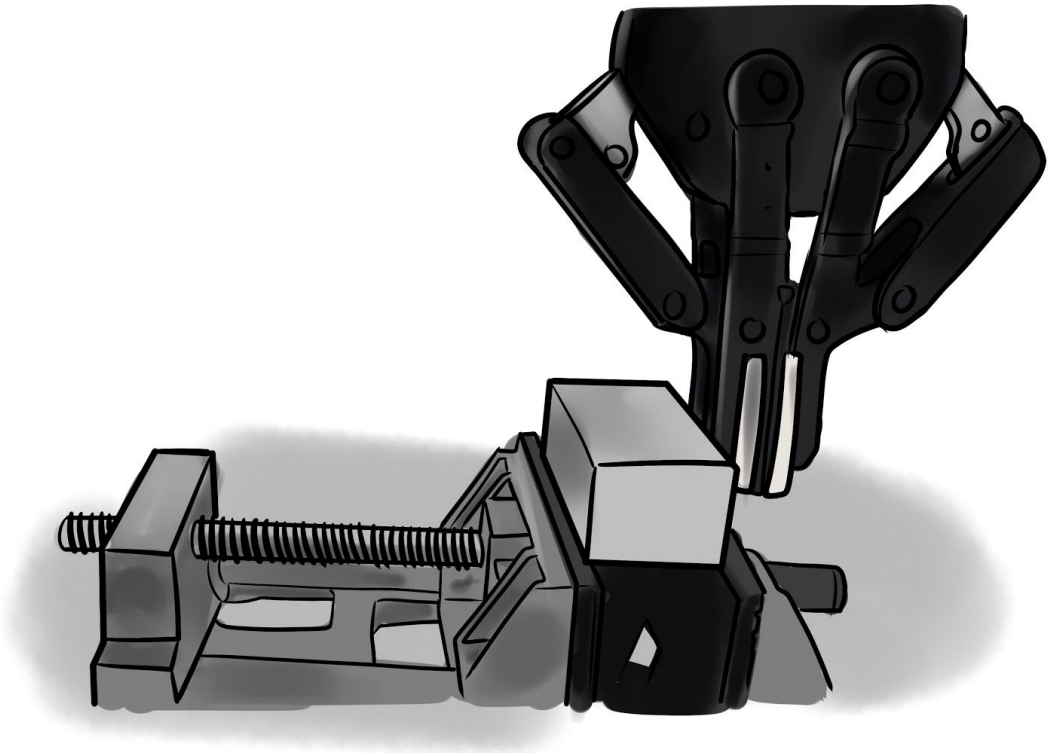
Here's another example of a common task. This one can be seen in many applications, such as machine tending, and it's likely to happen often in the lifespan of a robotic cell.

With this task, you need to move the robot in a known direction – where the surface is – until it reaches a certain force threshold. When that threshold is reached, it means your part is in contact with your jig or chuck.



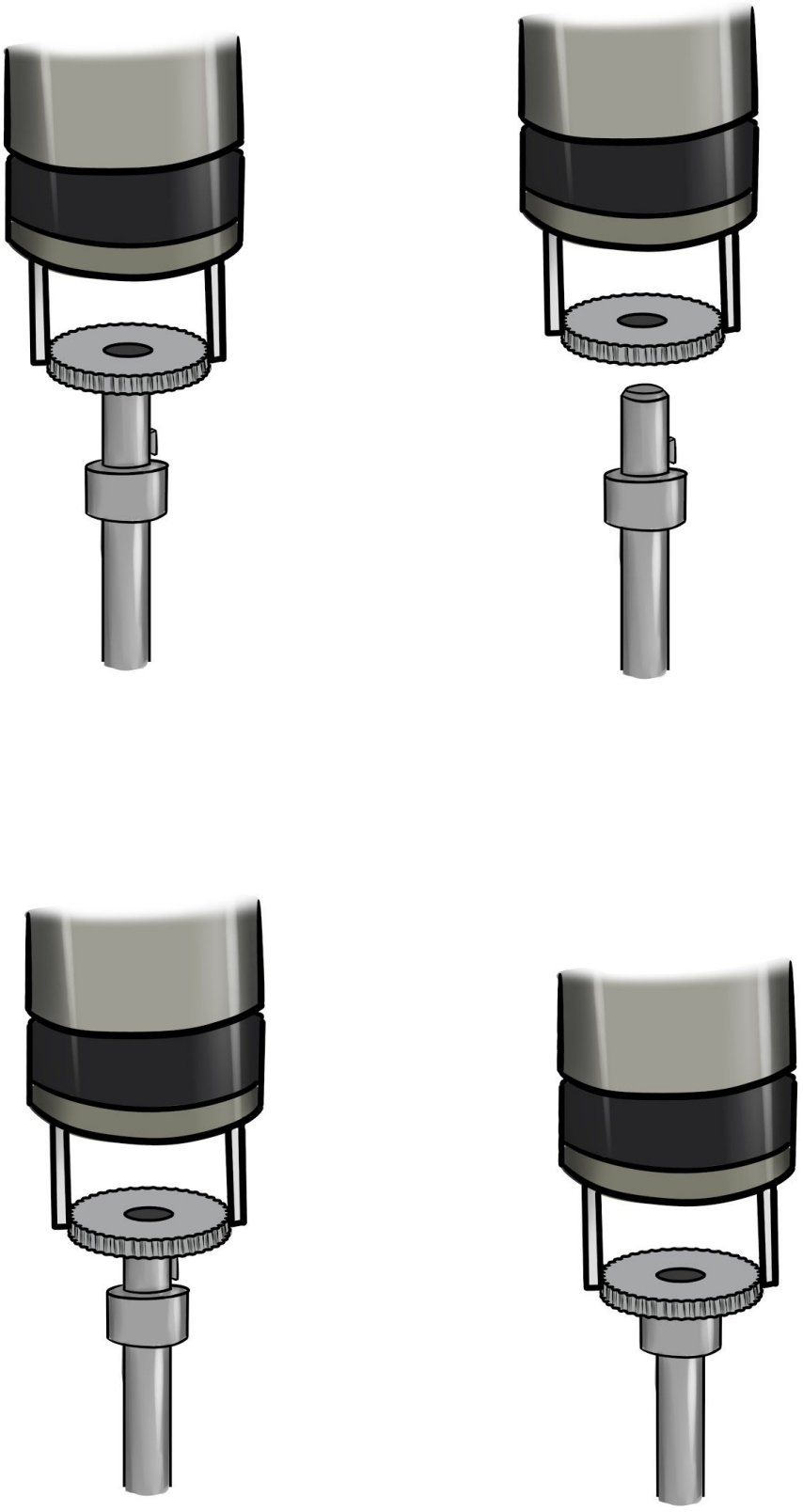






Example 3: Inserting with a positioning key

Let’s look at another example: inserting a gear into a shaft with a positioning key.



First, with the gear held securely between the gripper's fingers, the robot should move towards the shaft until it reaches it.

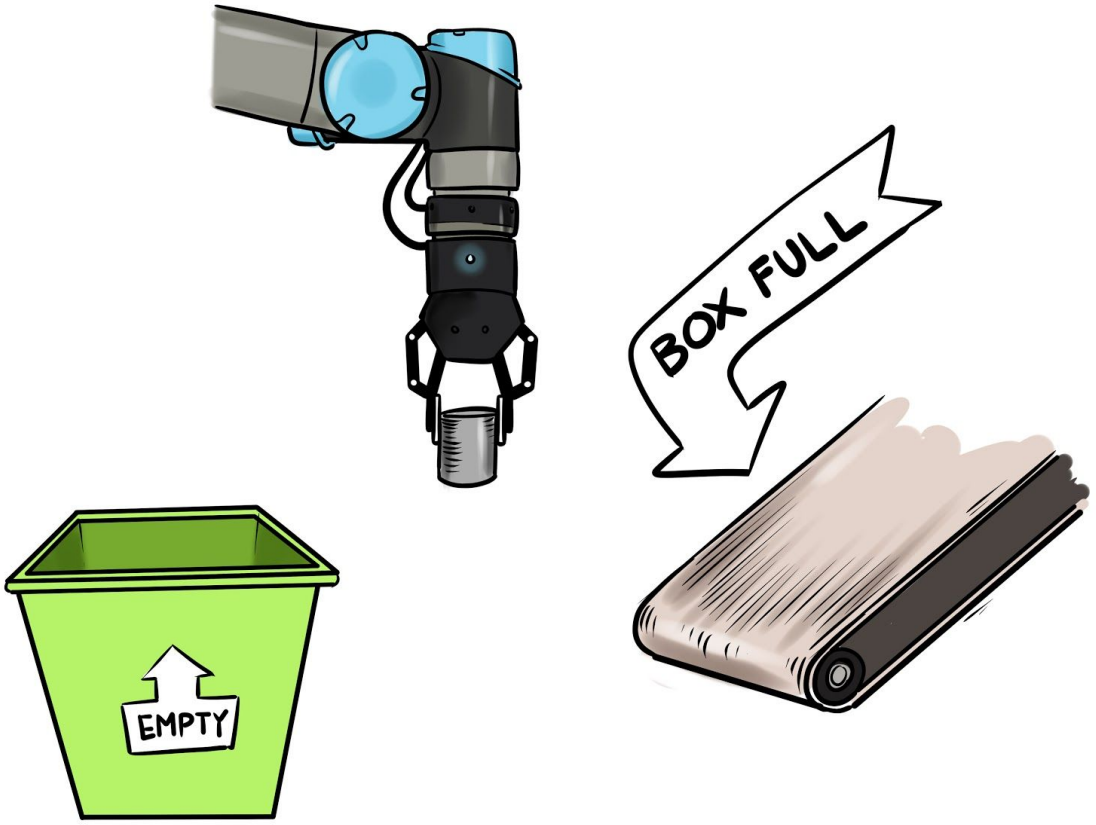
If the gear and shaft are not perfectly aligned, the sensor will output a high contact force. This tells the robot to perform a series of motions in order to reposition the gear.

Once the contact force drops – indicating that the gear and shaft are aligned – the robot can insert the gear until it reaches the positioning key, once again indicated by a rise in the contact force.

Finally, the robot can rotate the gear until the key and keyhole align. All of these precise and intricate tasks can only be performed by a robot with a force torque sensor.

Example 4: Quality testing

When using the robot to sort full from empty boxes, for example, you simply need to compare the force readings from a known-empty box and a known-full box.



You can do this by comparing both measurements and choosing the right threshold for your application.

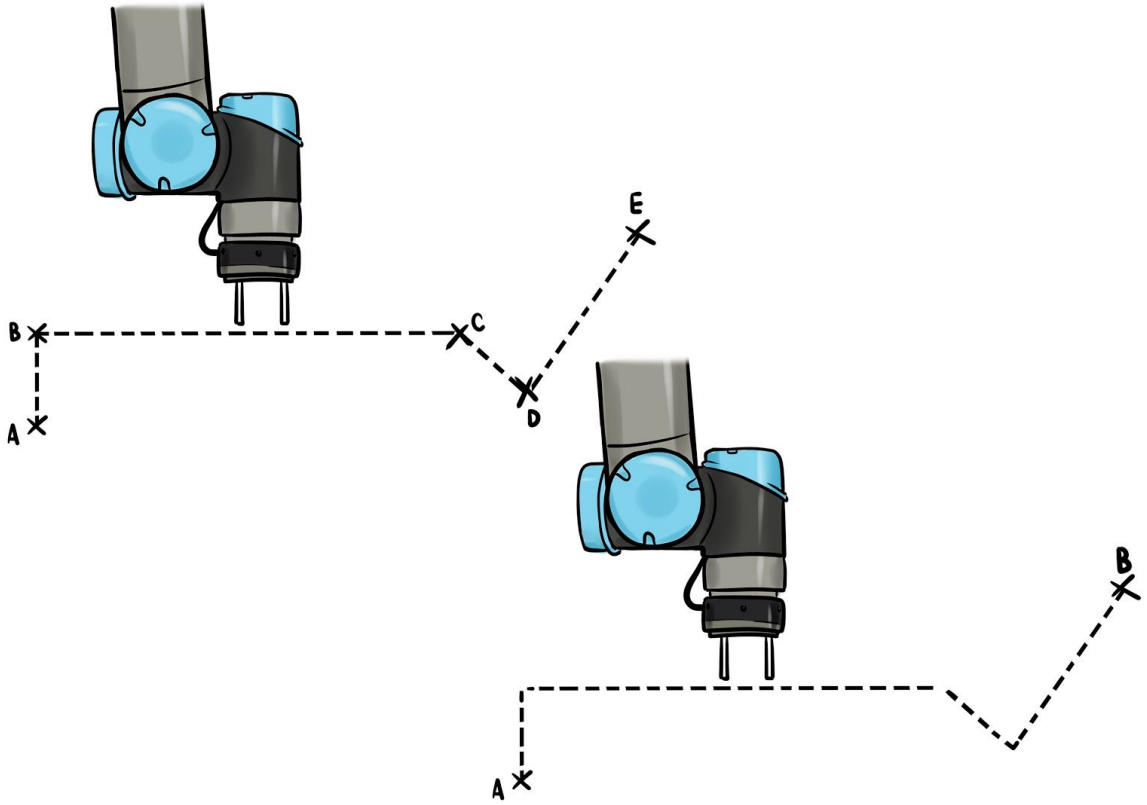
Bonus example: Trajectory teaching

Another feature that is sometimes provided with force torque sensors destined for cobots is the ability to guide and move the robot by applying force on the sensor.

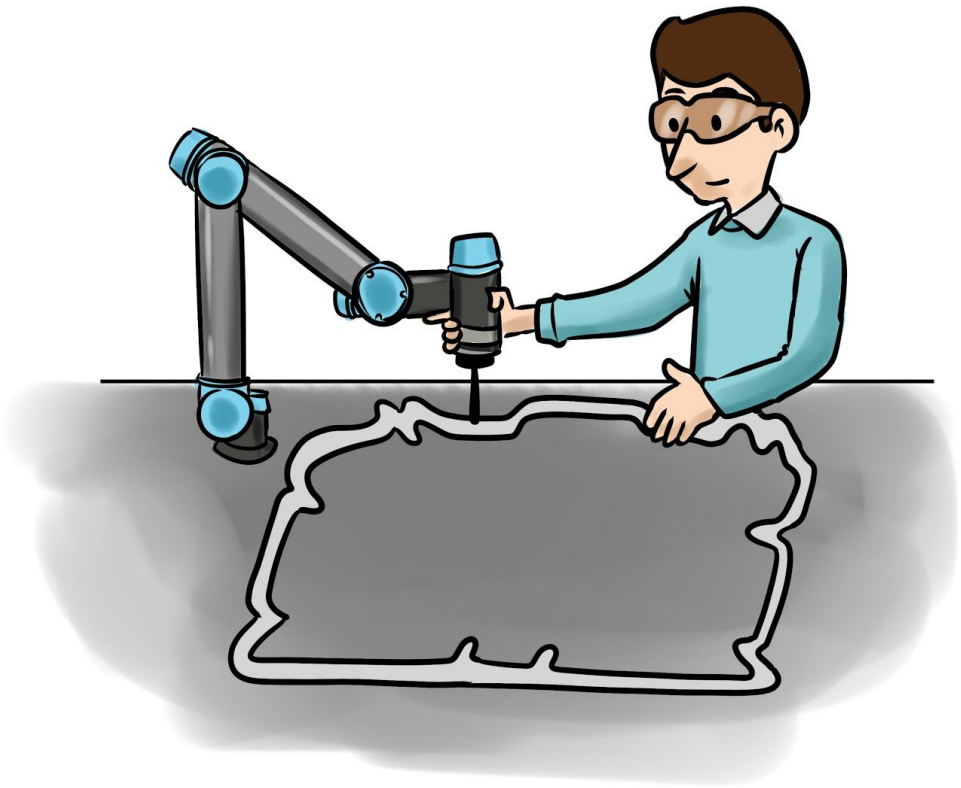
In this mode, the sensor will translate all forces and torques applied to it into robot motions. You can grab the end-effector and guide the robot easily by applying forces.

Some manufacturers even make it possible to record a trajectory and replay it in a program. If you're thinking "That's cool, but what can I do with it?," here's how you can use this feature:

- Teach trajectories instead of multiple motions.

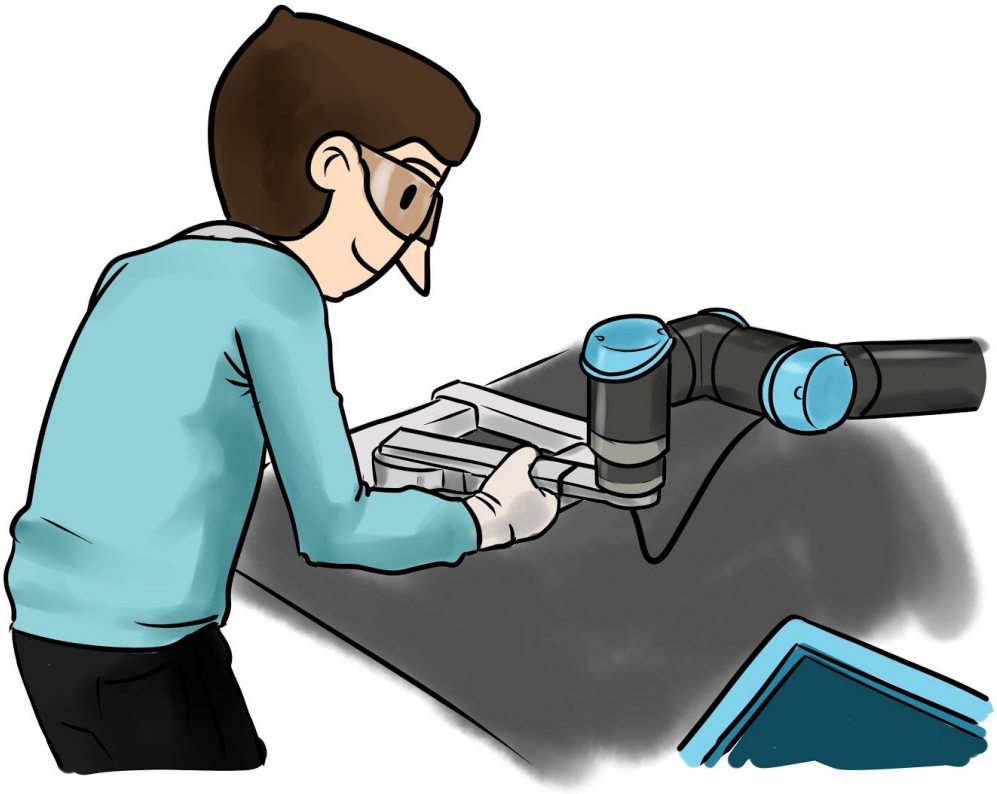


- Use the sensor to teach complex trajectories for dispensing on complex shapes. This feature saves a lot of programming time to get you to production faster.



- Teach motions that are intuitive to show the robot, but difficult to program traditionally.

- Teach motions for polishing or finishing a surface.

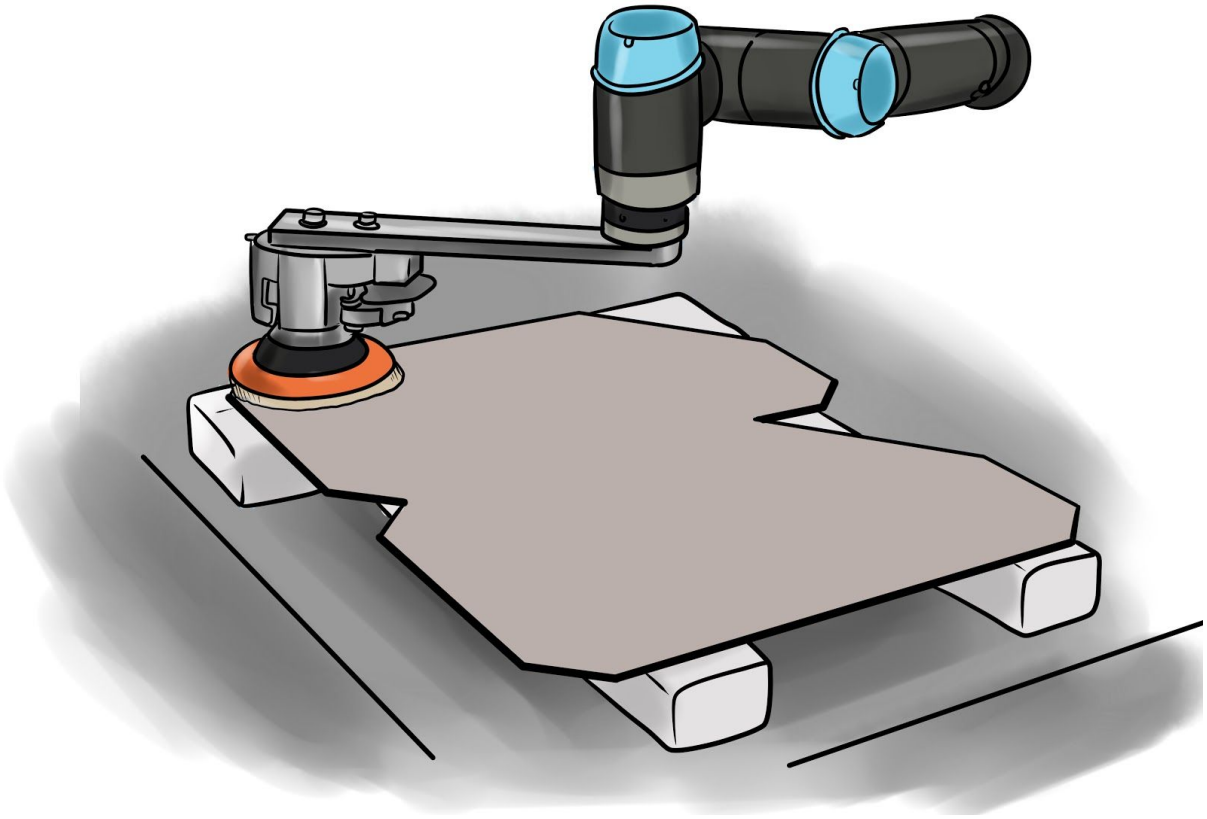


The “record trajectory” function makes it easy to teach motions for finishing or polishing. You can even use the sensor’s data to make sure a constant force is applied onto the surface while the motion is performed. That way, you’ll maintain the right amount of contact between the tool and the surface. Just make sure you confirm that your sensor includes this feature before you start!

Case study: Saint Gobain

At Saint Gobain, a glass production facility in France, they used a robot and Force Torque Sensor to:

- Reduce programming time
- Get to production faster
- Earn a great return on investment (ROI) on their robotic cell



A UR10 (robotic arm) combined with an FT 300 (Robotiq force torque sensor) was able to take over part of the operator's work. Now the two of them – human and robot – work together with greater productivity.

According to the integrator, "Without the FT 300, this operation would have been quite complex, since the programming of a robot movement that must follow a volume in space is a complicated thing to do."

He goes on to say that "With the FT 300's path recording functionality, the operator can simply grab the device and perform the movement; the Universal Robots UR10 then records and reproduces the operator's motion."

The sense of touch

At this point, you might be actively looking for the right sensor, but still have no idea how to choose one. Maybe you're still investigating what a robot could bring to your facility, or maybe you already have a robot and are just wondering what more you could do with a sensor. Well, you can do a lot more than you think.

No matter where you are in the process of getting a robot and/or sensors, begin with the application, and ask:

- What do you need the robot to do? Is it an assembly application?
- Do you need to place a part precisely in a jig or a chuck? Assemble gears? Polish a surface?

Once these questions have been answered, you'll know what sort of forces are at play.

Next, get a rough estimate of the forces and/or torques that your robot needs to apply or that your sensor needs to read. If possible, get an estimate of the precision required for your process. By defining these things, you'll have a general idea of what you want from a sensor.

Torques and forces

Let's take a step back here.

When I mentioned the "sense of touch" earlier, I was referring to forces and torques you can feel. You can feel these by the pressure applied on your skin, or by the amount of strength you need to lift, tighten, or assemble various objects. When you use your hand to tighten a screw, for example, you actually feel when it is tight enough by the amount of torque you apply with your arm and hand.

Forces are calculated in Newtons or pounds, whereas torques are calculated in newton-meters or foot-pounds. In robotics and automation, forces translate into masses that are either static or in motion.

Masses have inertia. You may remember – or know too well – this equation:

$$F = m \times a,$$

which tells us that a force is a mass times its acceleration.

A 1 kg part will apply a certain amount of force to a robot when it's picked up (approximately 9.8 N when static). Now, imagine if – with that part held securely by your gripper – you moved your robot at a certain speed and then suddenly stopped it.

This would create a great deceleration, resulting in the generation of high forces. That 1 kg part will apply a greater force on your robot and accessories under high acceleration. Thus, your sensor will feel it, even though no apparent external forces are applied.

That's why you might want to make sure your robot is at a stop before taking a precise measurement. You could also consider keeping your accelerations and decelerations low when performing a force-sensitive task, because otherwise it might affect the sensor's readings. One last small piece of advice: if you handle heavy parts with your robot, keep accelerations low.

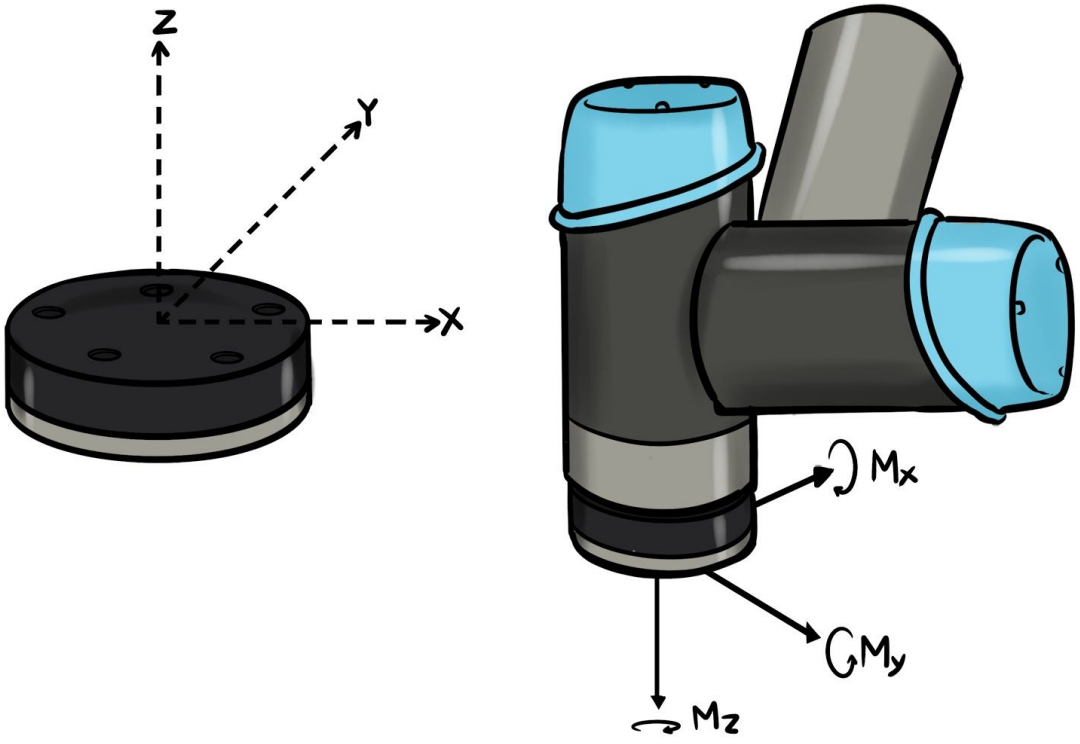
Now back to your application: what amount of force or torque should your robot detect or apply? It might be hard to estimate, especially if the task is usually performed by a human worker. What about how much force is required to insert part A in part B? Or the smallest amount of torque needed to detect whether a valve passes or fails quality control?

These things may not be easy to define, but if you can get a rough idea of them, it will save you valuable time when shopping for a sensor. You can also use tools like digital hanging scales,

torque wrenches, etc. Although we don't always know precisely how much force or torque we feel, with time and experience, you can get an estimate of it.

Sensing range

When looking at sensors for your robot, you'll probably focus on two main specifications: sensing axis and range.



Sensing range tells you the range of forces and torques that the sensor is able to read. (Some sensors can only read forces, whereas others can read both forces and torques; forces are given in Newtons or pounds, and torques in Newton-meters or foot-pounds.)

Make sure that the forces your application generates do not exceed this specification. If they're close, double check that they won't exceed the maximum load limit. You always want to have a margin of safety between your largest estimated load and the sensor's sensing limit.

Overload capacity

The maximum load limit, or overload capacity, is the load at which the sensor suffers damage. Make sure you have a good margin between the expected load and the sensor's overload capacity.

Also consider the robot's capacity itself: how much force is it able to apply? Larger robots are often chosen for their reach, but the ability to apply high forces may be unnecessary. Verify that the robot on which the sensor will be installed will not apply a force that could damage the

sensor. If it could cause damage, and there's no other workaround, you should limit the forces and torques that the robot is able to apply.

Sampling rate

Another important yet straightforward specification is the sampling rate. This is the rate at which measurements are taken by the sensor and sent to the PC, PLC, or other controller.

You might think the higher the sampling rate, the better the sensor, but take a minute to look at your controller's sampling rate and control rate. There's no need to get a sensor with a higher sampling rate than what your robot controller (or your cell's PC/PLC) can handle.

Precision

When looking at sensors, you may encounter a lot of technical specifications. The main one – sensing range – will always be present. Other than that, you'll want to seek out different specifications, some of which may not appear in all the specification sheets.

First, let's define what all these specifications are:

- **Accuracy**—How precise your sensor is, in absolute measurements. When your sensor sends you a 10 N reading, is it *really* 10 N? Use the accuracy specification to know for sure.
- **Resolution**—The smallest change in force that the sensor can measure.
- **Repeatability**—The sensor's ability to always return the same value for the same force. For example, with a low-repeatability sensor, you might have your robot pick up a part, and the sensor tells you it's a 7 N force. But later, when the robot picks up the same part (with the exact same routine), you get 8 N. Use the repeatability specification to avoid this scenario!

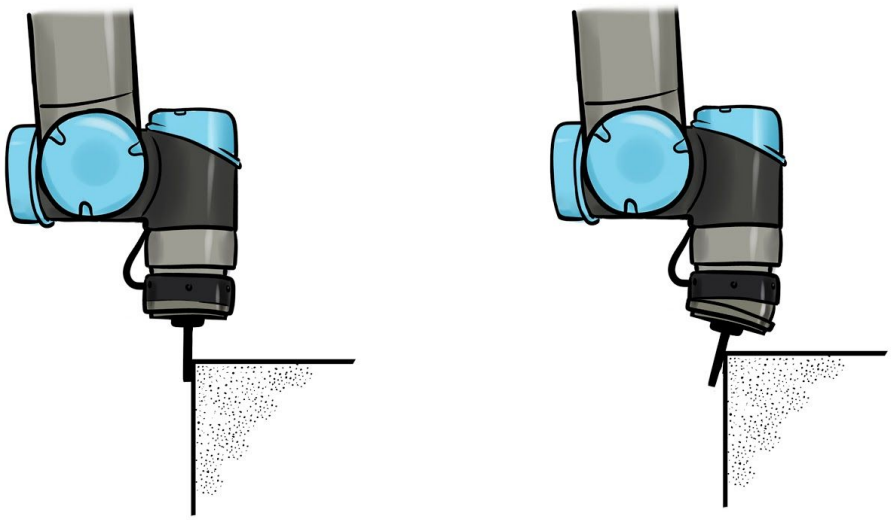
With these in mind, try to figure out what precision you need regarding accuracy, resolution, and repeatability. Do you need precision of +/-1 N, or +/-0.01 N?

Once again, a rough estimate is better than nothing, because this will greatly affect your choice of sensor.

Stiffness

One specification that's often overlooked is stiffness. This is usually described as a force (N) over a distance (mm). It basically lets you know how much the sensor will deform when under a certain amount of force.

In some cases, a "soft" (less stiff) sensor may be a good fit, but keep in mind that softness will affect the repeatability and precision of your robot and system.



Depending on the sensor's stiffness, certain forces can cause the part to shift its orientation (see the figure above). This shift might make the task harder to accomplish. Do yourself a favor and verify that the stiffness of the sensor you're considering will not lead to such undesirable situations.

Stiffness, along with sensing range and overload capacity, are specifications that let you know how robust a sensor is. Will it deform under certain loads? Will it get damaged under your operating conditions?

Sensors should unlock possibilities, not be a limiting factor in your robotic cell.

Mechanical fit

It's important to check that the sensor is a good mechanical fit with the rest of your robotic cell. To do so, ask:

- How will the sensor be mounted on your robot arm?
- Does the bolt pattern fit? What about the dimensions?
- Is there an adapter plate available for your robot model?

If the answers are "I don't know" or "No," you might need to design a custom mounting plate. This can be quick and easy, or it can add extra design and manufacturing costs, so make sure you know what you're getting into.

Communication protocols

Communication protocols are what I like to call the "dark magic" of automation. I call them that because, frankly, the only way to become familiar with things like Modbus, Ethernet, EtherCAT, etc., is to work with them regularly. But don't worry, this dark magic can be tamed!

As collaborative robots continue to rise in popularity, sensor manufacturers will try to make their tools simpler to integrate with robots. One of the main selling points of the new collaborative robots is that they're easier to program than ever, so it makes sense to design accessories that are complex to use.

Start with your robot controller or PC/PLC: what communication protocols do you have available? And which electrical interfaces do they use?

Ideally, your PC/PLC or robot controller will support the same communication protocol as the sensor you're looking for.

But you also need to check whether this communication protocol can be used with accessories. For instance, just because a robot can be controlled via Ethernet/IP, doesn't mean it can control and read values via Ethernet/IP.

Depending on the protocol, you need to ensure your controller has the right master/slave or client/server card. To take Modbus RTU as an example, your robot needs to be the master if your sensor is a slave.

You should also inspect the electrical interface. Is it an RJ45 connector, an M8 5-Pin connector, a USB, or something else?

If the sensor of your dreams doesn't have the right communication protocol or connector, you could look into converters or gateways. Converters allow you to keep the same communication protocol but convert the electrical interface. For instance, you can go from RS485 to RS232 with a simple converter.

If you need to completely change the communication protocol – say from Modbus RTU to Ethernet/IP – you need a gateway. With gateways, the right registers must be sent through and read by your controller at the end. Make sure to properly define the master/slave or client/server.

Although gateways and converters can be good workarounds, they will add to the bill and the integration time. You need to know how to configure your registers in order for the sensor to communicate with your robot.

Now, do you have to configure complex communication protocols? If so, do you have the in-house skills and time to do it? Or will the manufacturer provide you with configuration files or a software package that installs on your robot? If they do provide configuration files, great – these can save you a lot of time. As for software, that brings us to the next section.

Software

Software is a major aspect of your sensor decision, because it affects how you'll program the device. Is there an API – a software package that fits on your robot? Do you have the necessary in-house programming skills?

The complexity of the program depends on your application. For instance, do you need the robot to hold a part and determine whether it's heavy enough? Or let's say you need the robot to polish a surface: does it have to apply a constant force while following a curved surface? Those two applications will require very different programming skills.

You may also run into blockers that will make programming very complex, and which require time, testing, and possibly modifications to your setup. I've seen cases where the person programming the robot had to call an external integrator for help. This situation adds extra costs to a project that you might have thought was close to being finished.

When looking at sensors, inquire about if manufacturers provide software packages or programming help that might get you to production a lot faster.

Sensors are one of the most complex robot accessories to program. Once mastered, they are a powerful tool that can unlock so many applications for your robot: assembly, inserting parts in jigs or chucks, testing products, polishing or finishing a surface, applying glue or grease, performing complex trajectories, etc. Plus, when sensors allow you to hand-guide the robot, they make other robot-programming tasks much easier!

When integrating

In this section, we'll go over some best practices and things to keep in mind when you're integrating your sensor with your robotic cell.

Your system

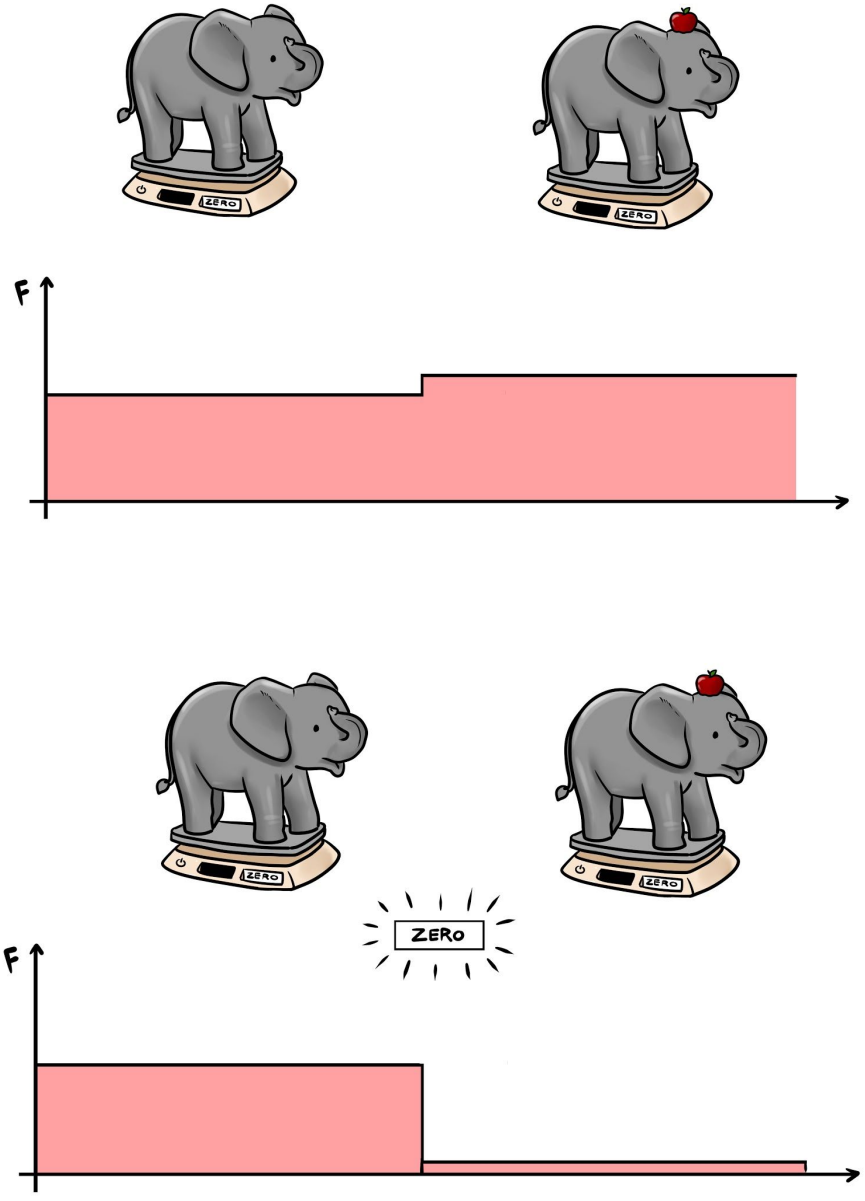
Sensors are great, but they have some limitations. Often times, the limitation does not lie within the sensor itself – instead, it stems from the robot arm vibrations, the end-effector tool vibrations, or the end-effector cables that can pull and create unwanted forces.

There's a well-known saying in engineering: garbage in, garbage out. If you try to get a precise force control in your application, but your robot arm shakes a lot, you can only expect your sensor to send back data containing noise. If your gripper's cable pulls on it, you'll get weird force readings. This is also true if you have a tool that causes your robot to shake, such as a grinding tool.

Remember: the sensor feels everything that is applied to its sensing element, but it can't feel anything beneath it or elsewhere. It's not aware of its environment. It won't know if you mount a polishing tool on it, or if the table on which the robot is mounted is shaking. You must ensure that all these factors are taken into account, and use your sensor wisely within its limits.

Relative vs. absolute calculations

There are two types of measurements you can take with a force torque sensor: relative and absolute.



A relative measurement is made relative to a known value that is set at zero. An absolute measurement is not compared with anything else: for instance, 20 N is 20 N no matter how you measure with an absolute measurement.

Some sensors have good accuracy; others don't. Does your application require exact measurements – such as measuring the exact weight of a finished product? If you want to use your sensor for this, make sure its accuracy is high enough.

In some cases, you'll care more about having good repeatability, and it will make sense to go with relative measurements. You can then zero your values before taking the measurement. I use this technique pretty much all the time with insertion or assembly applications, because I want to have good repeatability throughout all the cycles.

Changing conditions

Depending on your sensing technology, it may be sensitive to temperature. This is usually well documented in the product's user manual. Some technologies have to compensate for temperature change, so verify that your facility's temperature is within the acceptable range.

Programming

Sensors are amazing tools. I'll always put them on any robot I use, since they unlock so many possibilities. Give yourself time to get used to programming with a sensor's data. If you don't, you may end up feeling frustrated with your sensor. One of my teachers used to remind us that every sensor, and the programming that goes with it, is merely a tool. It's up to us to master it.

Once you have your sensor and you're ready to start programming with it, there are a few things to keep in mind.

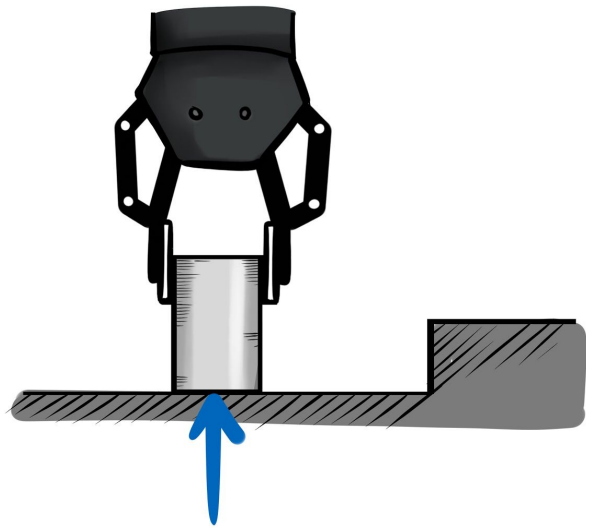
As explained earlier, your system limits your precision. If you have a shaky robot or a wobbly table, your whole setup will shake. Those vibrations will be felt by the sensor and will add noise to its signal. Whether you accept this higher noise or fix your setup is up to you, but don't blame your sensor.

In most applications, you'll be more interested in relative measurements than absolute ones. You'll want to have good repeatability. Let's say you want to place a part correctly in a jig. You need to have the robot press the part on two sides of the jig. It doesn't really matter if it applies 10 N or 20 N, but it does matter if it applies 150 N rather than 10 N. So you have to adjust the force at which you want to place your part onto the jig. Take some time to test it out and decide on the force you want for your setup.

Here, for instance, the way to go is to zero your values *before* placing the part using force feedback. The same rationale applies if you want to program an insertion application. You can zero the values right before you need to make measurements or verify a contact force, for instance.

You'll also need to test out which force thresholds work for your setup, which includes your robot, sensor, and parts. Don't forget to zero your sensor before those force feedback motions.

Here's a simple programming example of moving a part in a direction towards a surface. In the language of force torque sensors, this means moving the robot arm in a direction until the sensor reads a force that exceeds a defined threshold:



```
While Fz < 10N
   Move robot downward in increments
end
```

You can go even further by adding break conditions. These are conditions that, if met, will make the program go out of the force sensing loop. If your robot is ever totally out of place, these conditions ensure it won't move into unwanted poses. Let's take the short program example we just saw. For example, I've have added a break condition that stops the robot from doing X if Y condition is met:

```
While Fz < 10N
   Move robot downward in increments
   if displacement > 100mm
      break
   end
end
```

Of course, this is a really simple example. Programming advanced applications with an FT sensor can be intimidating. The reality used to be that you could either learn to master programming applications with FT sensors, or hire someone who already has this knowledge. However, some manufacturers are making this task easier by offering pre-built functions, lines of codes, or packaged tasks for your sensors.

In these cases, you simply need to edit a few parameters (such as force threshold, displacement limit, and direction of motion) and the pre-built function will take care of the rest.

Make sure you check out these possibilities (such as Robotiq's Skills) before you decide on a sensor. These functions and pre-built tasks make programming much easier, save you a lot of time, and allow your robotic cell to get to production faster. Your integration process will be simpler and more effective, and you'll enjoy the full potential of having a force torque sensor on your robot arm.

Takeaways

We've covered a lot of ground in this eBook. Hopefully by now you're able to survey all the available sensors and know exactly what to look for.

It's also important not to get intimidated by the thought of integrating the sensor with your robotic cell. I want to make sure you're confident about taking on the challenge!

With that in mind, here are the key takeaways from this eBook.

Tasks and applications

In case you're not yet convinced, here's a reminder of what FT sensors can do:

1. **Insertion**—With force and torque feedback, your robot can perform precise assemblies, insert a shaft in a hole, plug connectors, etc.
2. **Part placement**—Sensors enable robots to place parts in jigs and chucks, so you can be sure they're in the right position for the next task.
3. **Quality control**—This application can mean many different things. With a sensor, your robot can know if a box is empty or full, if a valve has enough torque to rotate, if a cap is well tightened, etc.
4. **Polishing, finishing**—These require some adjustment to get the right contact force and go over the entire surface properly. But the effort is often worth it, since automating these tasks can mean freeing human hands from some of the most grueling tasks.
5. **Trajectory teaching**—This is where sensors can save you a lot of programming time. For instance, you can even teach trajectories when you need to dispense grease on objects with complex shapes .

Quite simply, sensors unlock many possibilities and get you to production faster.

Technical specifications

Here's a checklist of the specifications and information you need to look for when you want to get a sensor:

1. **Sensing range**—This is the range of forces and/or torques you can read with a sensor. It's usually provided for each axis. Make sure it's high enough for the forces that will occur in your application.
2. **Overload capacity**—If you have forces higher than your sensing range, you need to be absolutely certain they won't exceed the sensor's overload capacity (the load at which mechanical sensor damage may occur). Verify if your robot itself can apply such a force. If so, try to limit it if possible, or look for a sensor that has a higher overload capacity.
3. **Sampling rate**—This the the frequency at which the sensor sends out force readings. Is it fast enough for your robot? Keep in mind that there's no need to get a sensor that has a higher sampling rate than your robot controller can handle.
4. **Precision**—This is one of the most important specifications in a sensor. Make sure you fully understand the difference between accuracy, resolution, and repeatability, and

whether you need relative or absolute measurement. Confirm your needs before you start comparing sensors based on precision.

5. **Stiffness**—While this specification may not seem important, it will be if you have a sensor that is not stiff and you want to do more precise insertion, for instance. In this case, the part may shift its orientation during insertion or force application. Make sure you compare sensors' stiffnesses with your needs before buying one.
6. **Mechanical fit**—Check that the sensor can be mounted directly on your robot's wrist. Otherwise, are there adapter plates or couplings available that would fit? If not, you'll need to design and fabricate one. Do you have the skills, time and budget for that?
7. **Communication protocol**—It's important to verify how the sensor communicates. If it fits with one of your controller's possible inputs, your job is much easier. If not, the manufacturer might have options, or you might have to use a gateway. A gateway will likely result in more integration time. In any case, does the manufacturer provide any configuration files or drivers? These will save a tremendous amount of time.
8. **Software**—Some sensor manufacturers will provide software packages that make integration much easier. Some will even go one step further by providing pre-built functions that make your robot perform tasks using the sensor's feedback. This is a huge time-saver, so make sure you investigate whether it's available.

Integration and programming

Here are a few things to consider when programming a robot application that includes a force torque sensor:

1. **Your setup**—No matter how precise your sensor is, if you setup has vibrations or unwanted forces applied to your end-effector, it will affect your force readings. Thus, when considering your setup, be consistent with what you expect from your sensor.
2. **Relative vs. absolute**—You can either measure an absolute force, the actual force, or a relative one (which is relative to a defined zero).
3. **Programming sequence**—Using a sensor in a program often requires loops and logic operations. While this may seem complicated if you've never done it before, you can try performing some simple applications first to get the hang of it.
4. **Pre-built functions**—Some sensor manufacturers will provide you with pre-built functions to simplify the programming of complex tasks with your sensor. All you need to do is change a few parameters. Check out some examples with these Skills.

Now that you understand a bit more about sensors, get out there and start automating force-sensitive tasks today.

About Robotiq

Robotiq's Lean Robotics methodology and products enable manufacturers to deploy productive robot cells across their factory.

They leverage the Lean Robotics methodology for faster time to production and increased productivity from their robots. Production engineers standardize on Robotiq's Plug + Play Components for their ease of programming, built-in integration, and adaptability to many processes. They rely on Flow's software suite to accelerate robot projects and optimize robot performance once in production.

Robotiq is the humans behind the robots: an employee-owned business with a passionate team and an international partner network.

Let’s keep in touch

For any questions concerning robotic and automated handling or if you want to learn more about the advantages of using flexible electric handling tools, contact us.

Join us on social media:

Workfloor: Robotiq’s Blog

Twitter

Linkedin

Facebook

Youtube

Google+

DoF

Robotiq’s community where industrial **automation Pros** share their **know-how** and **get answers**

Ask Your Question

LEARN MORE

robotiq.com | leanrobotics.org