ROBOTIQ

HOW TO AUTOMATE A SCREWDRIVING CELL

LEAN ROBOTICS

Content

Introduction **3**

Prepare **4**

Identify the right process 4

Master your process 6

Design **8**

Choose a robot 10

Screwdriving components 11

Fixtures 13

Infeed and outfeed 14

Layout 15

Safety 16

Information exchange 16

Integrate **17**

Identification 17

Error management 18

Information exchange 19

Test the cell 20

Operate **20**

Changeovers 21

Calibration 21

Conclusion **22**

Introduction

For a human operator, screwdriving is a dull, repetitive, and non-ergonomic task. But for a company, it's a crucial process. Fastenings make the difference between a good product and one that fails in the field.

As humans, we are prone to error, especially in repetitive tasks. To avoid defects due to tiredness or lack of concentration on the job—or to fill a position that may still be empty—automating this part of the manufacturing process can be worthwhile.

Taking the leap and investing in an automated solution will take time, might involve some assembly changes, and can be tough to sell to higher management. However, with the following tips and tricks, you will be able to successfully automate your screwdriving process.

"When you decide to deploy a robotic cell, it's usually because you're aiming to solve some issue in your company regarding productivity, quality, or product output capacity."

—Sam Bouchard, Robotiq CEO

Prepare

Identify the right process

If you're reading this, it's probably because you've already identified a problematic area in your process.

If you're simply aiming to shave time off your screwdriving process, automation is probably not the best way. When you consider the time for a single screw, the automated process will generally be slightly **slower** than a human operator.

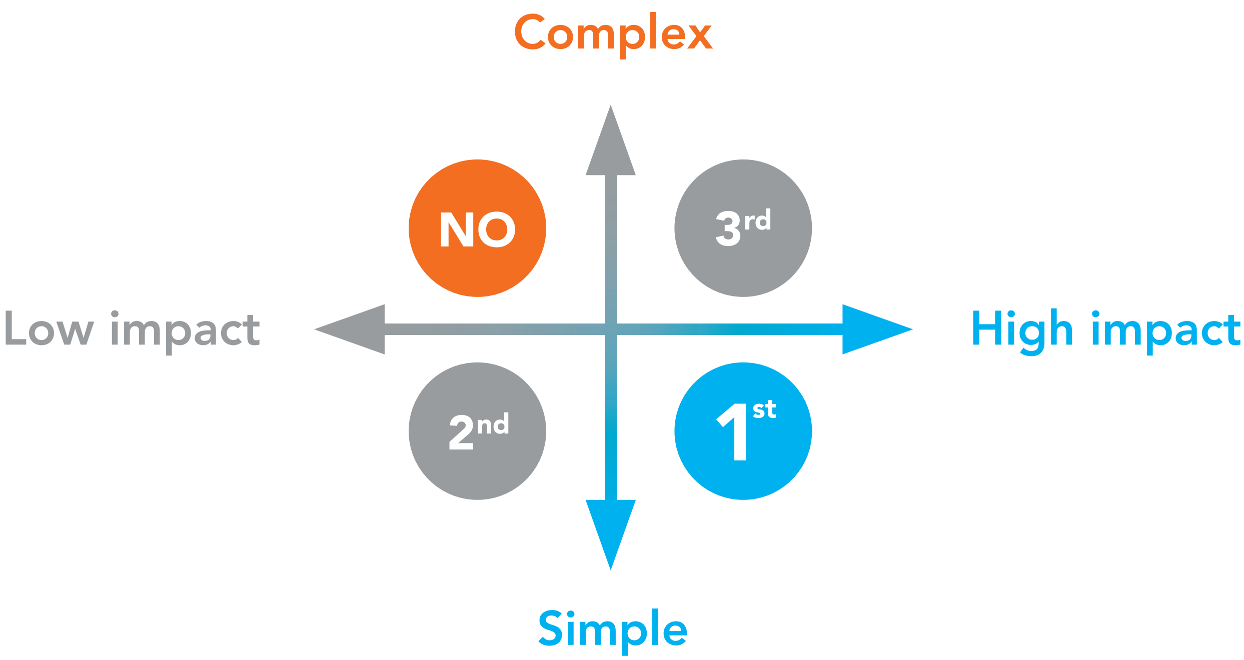
But consider the bigger picture. Automation may not improve your cycle time from the get-go, but at the end of the day, you will be able to increase cell output and produce more with fewer human resources.

Quality issues can also be a factor in your choice. Human operators are quicker than automated solutions, but they are also less consistent.

If you are at your first automated screwdriving project and you are at the point of analyzing which part of your process to automate, remember these words:

Start simple

This may seem obvious in theory, but it's easy to imagine a very complex cell that solves all the current process problems by doing a whole bunch of things at once. The reality is that these complex projects are more trouble than they're worth when you're just starting out. You need to start with an assembly project that is relatively easy to automate but still has a great impact on the production floor.



The best process to automate is simple and has a good return on investment. By contrast, the worst process to automate is a costly integration that does not provide any gain in throughput.

A **simple** assembly project with a **high impact** might be an operation where all the screws to assemble are on the same face and the part can be held with a simple fixture. The cell output and the overall product quality will go up with automation.

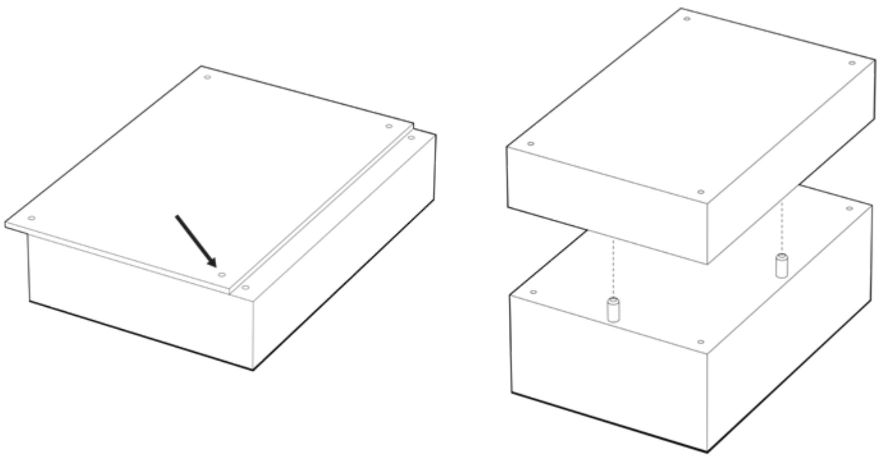
A **complex project** with a **low impact** could be a complex assembly with screws in a lot of different faces that requires a complex automated jig to turn the part in all different directions. The output of the cell won't increase much, and if the assembly fixtures are poorly designed, the output quality may even be worse.

Master your process

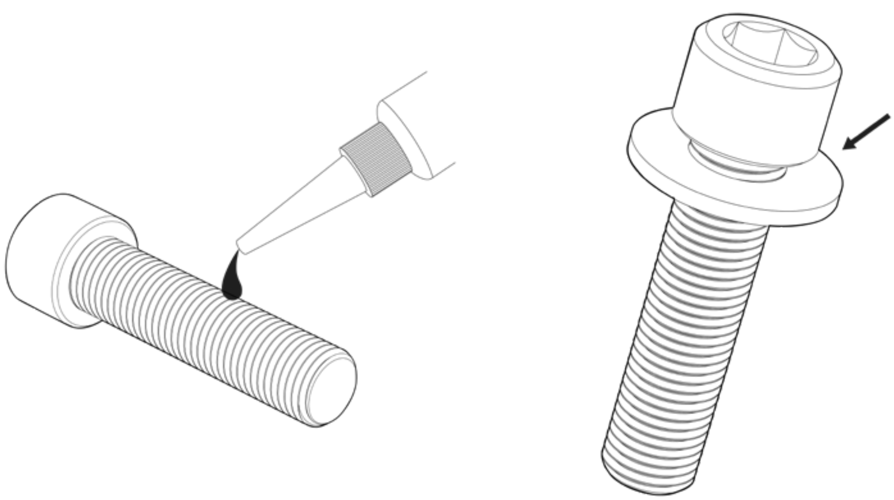
Aside from choosing a process that is simple, you also need to choose a process that has already been mastered. In fact, you need to consider a couple of things that make it easy for a robot to assemble.

Make sure:

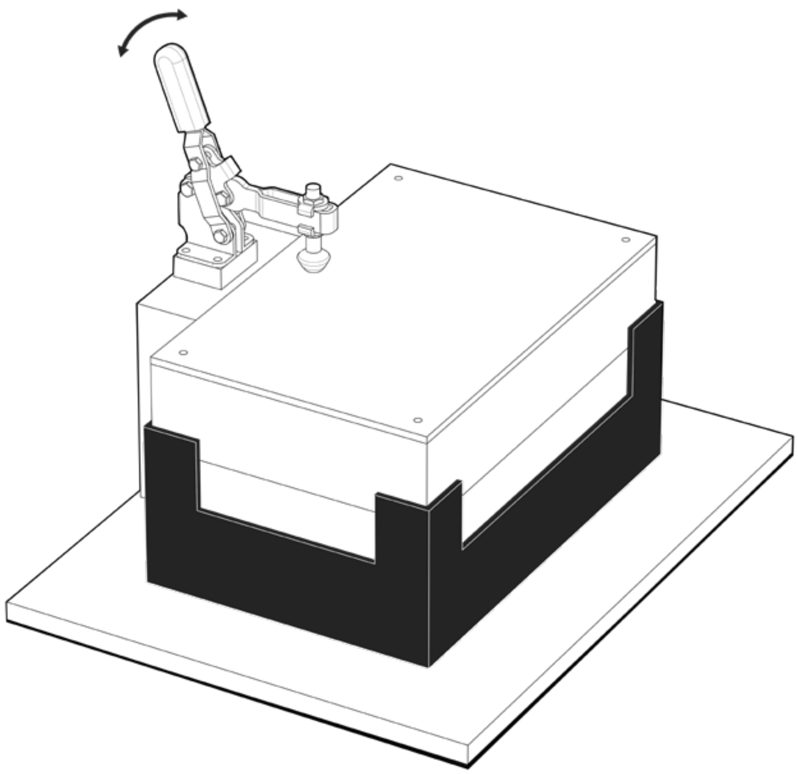
- **The assembled part can be aligned easily.** By aligning parts with pins or other mechanical features, you ensure a more stable fastening process.



- **The fastening process is simple.**
  <u>Don't</u> start with adding a locking compound or washer.



- **The assembly can be held by clamps.** You won't have a human operator using their hand to hold the parts together, so the parts need to be held with clamps or within an assembly jig.



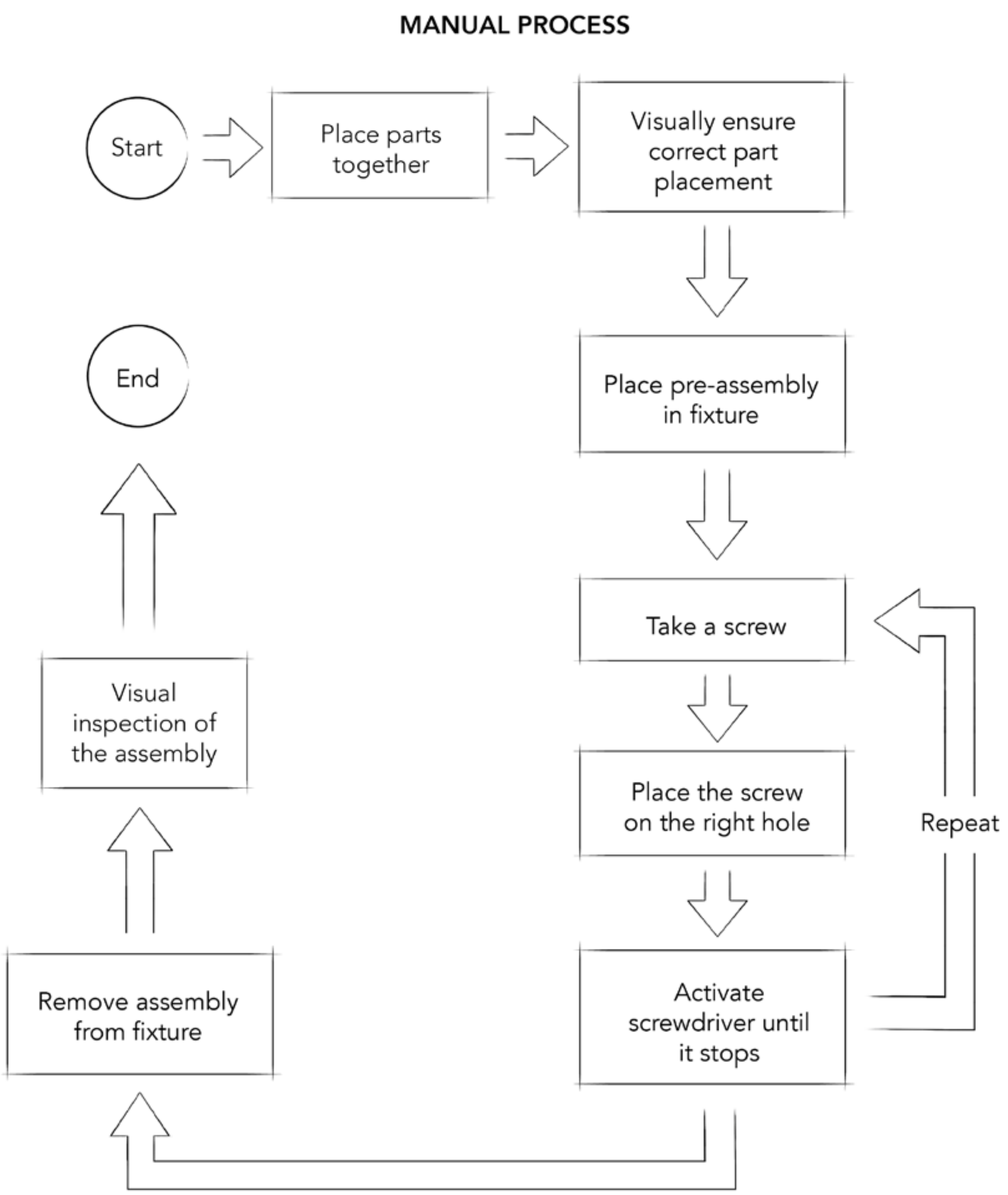
- **The process is stable.** If there is a lot of variability in part size or there are a lot of defects in production, the robot won't help. Make sure to master the production process before automating it.

Once all these details are out of the way, you should be able to choose which process you would like to automate first. If you have eliminated all the potential candidates at this point, you might have to go back to the drawing board and redesign your assembly to make it suitable for automation.

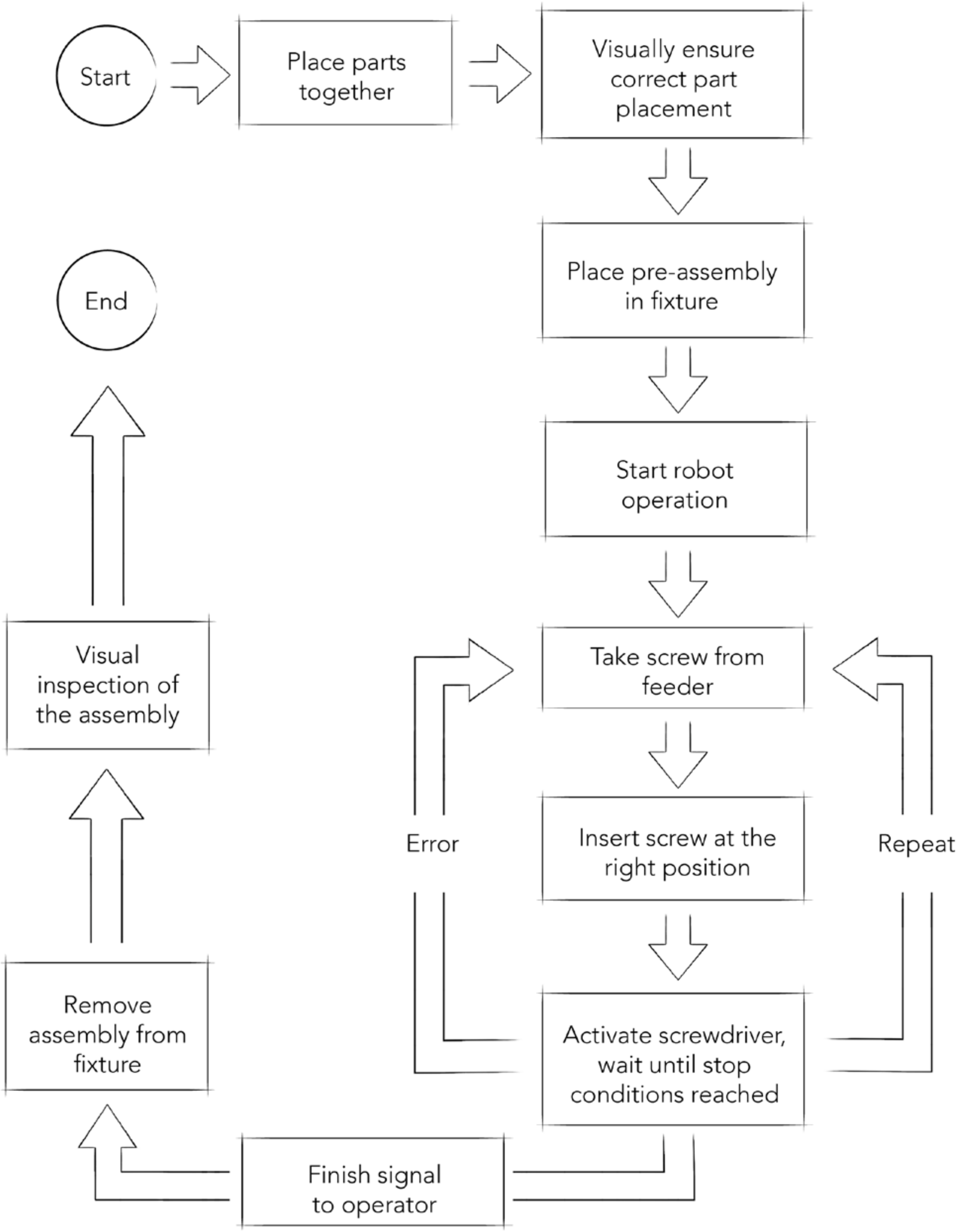
Design

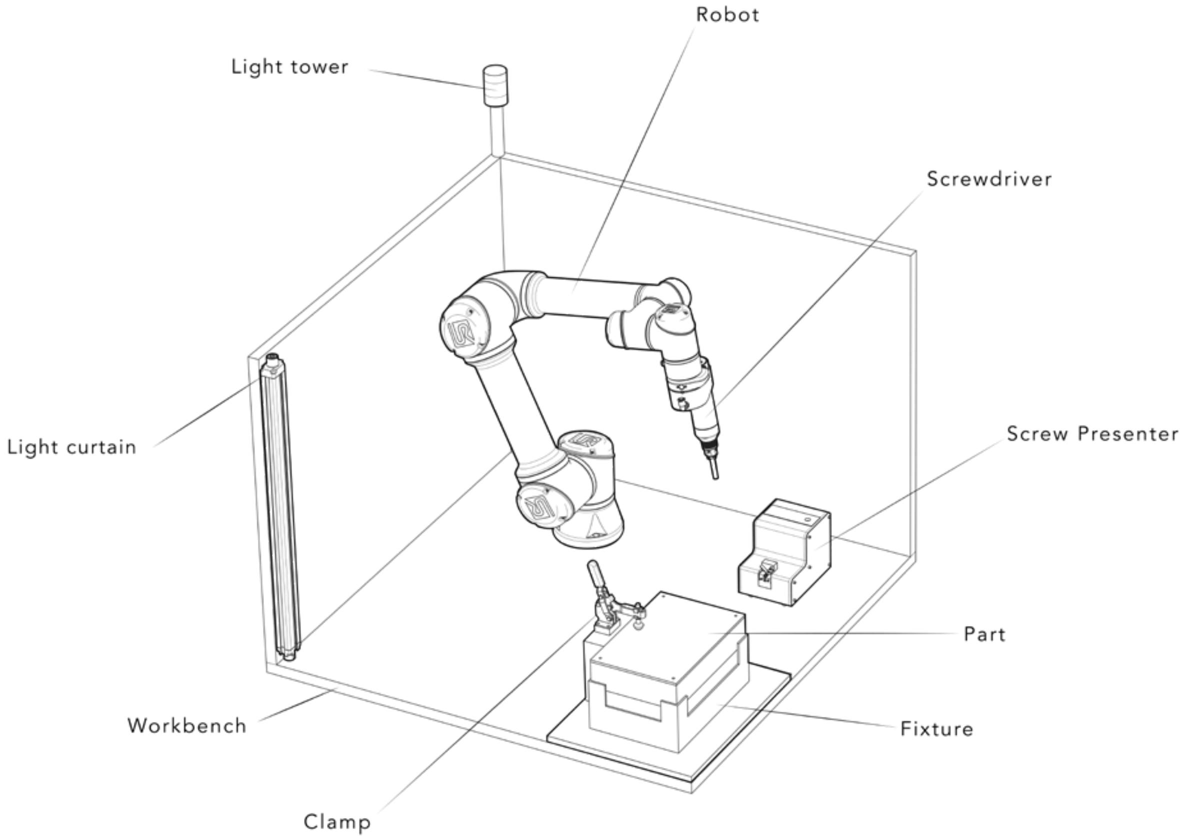
Every good engineer starts the process here, but make sure you've identified the right process to automate before starting to develop solutions.

The design phase is where you will start putting together all the parts of the puzzle. Since you have already identified the process, you will next need to choose the tooling and layout that suits this application.



AUTOMATED PROCESS





Choose a robot

The main component of the robotic cell is, of course, the robot. There are many different brands and types of robot that can do the job, but it really comes down to two categories: collaborative and industrial.

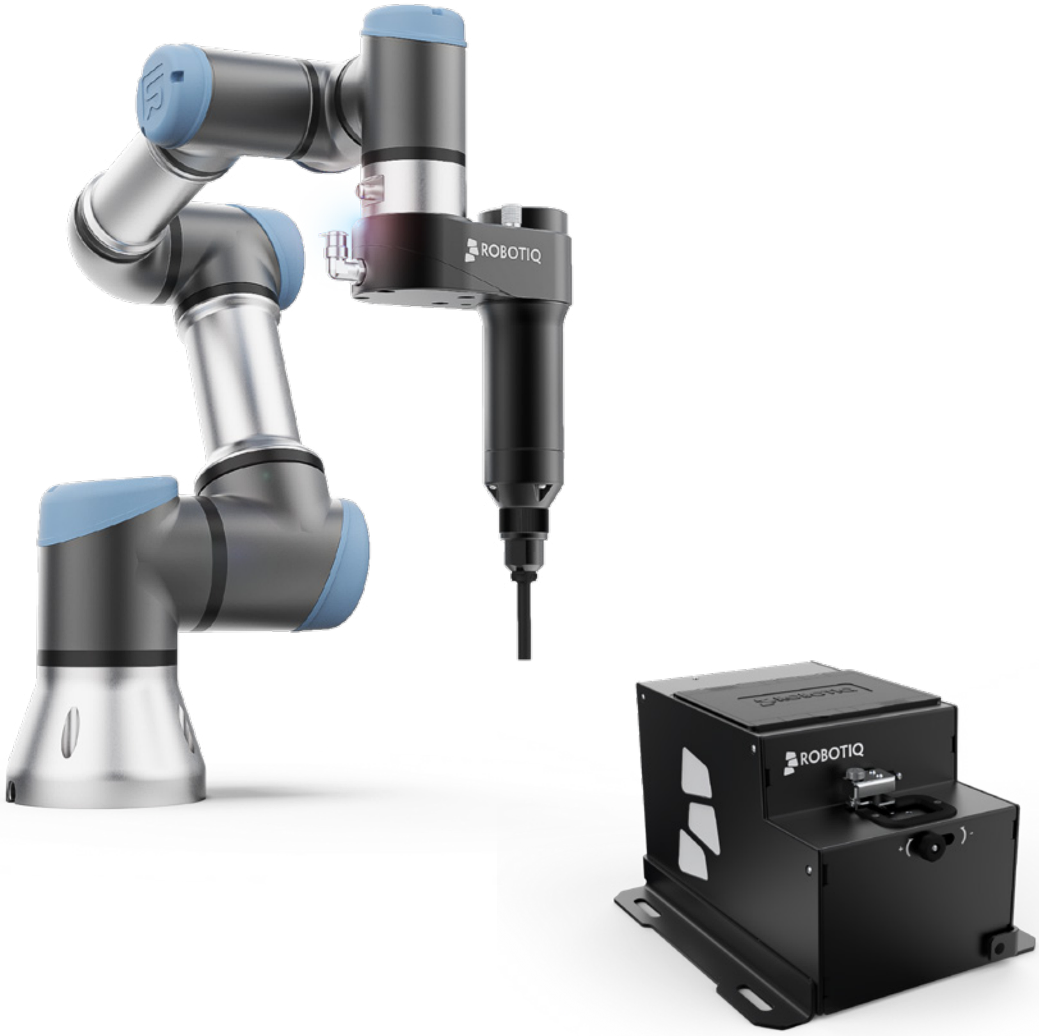
- **Collaborative robots.** These robots are designed to be used by almost anyone. They are easy to understand and program and can work alongside human operators. They are usually compatible with third-party hardware, which means you can get tooling with the same ease of use.

- **Industrial robots.** This category includes all robot shapes and sizes, from a small three-axis robot to a large car-carrying robot. The most important differentiator is that they are not safe to use alongside human operators, and they need more advanced knowledge to program them. Programming is usually done by a specialized robot technician, which can be costly when it's time to change parts in the assembly process. There is typically no third-party tooling available for these robots, so everything needs to be customized.

Screwdriving components

The heart of the screwdriving cell is the items that will perform the screwdriving. A screwdriver and a screw presenter are the bare minimum to automate a screwdriving process. There are a lot of screwdrivers on the market and probably as many screw presenters.

For maximum flexibility, try to choose components that are not too specific to your application. Doing so will reduce your risk of having a dedicated screwdriver that will sleep on the shelf if your project is canceled. Plus, it can also allow you to fasten different products on the same day or do changeovers without changing the robot hardware.



You will also need to think about how the screw will be carried from the screw presenter to the screw location. There are three main ways to do so:

- **Magnetic.** This solution uses a screwdriver bit that is magnetized and that can carry ferromagnetic (mainly steel alloy) screws. This process limits the type of screws that can be taken by the screwdriver, and there is no way for the robot to confirm screw presence. It also limits the position and angles at which the screwdriver can be fastened.

- **Pneumatic.** This system uses compressed air to generate a vacuum to carry the screw from the presenter to the final position. This system requires very little customization and has very small air consumption. It allows the robot to validate screw presence, to make sure the robot is traveling with a screw on its bit. The downside is that the venturi acts as a vacuum and will suck up all the small debris generated by the screwdriving operation.

- **Automatic feed system.** This solution carries the screw from the presenter to the screwdriver tip through air-compressed tubing. The robot does not need to travel back to the screw presenter each time it needs a screw; in fact, the screw is automatically sent to the screwdriver within a second. The downside of this solution is that it requires a lot of maintenance and customization, and it is usually more expensive.

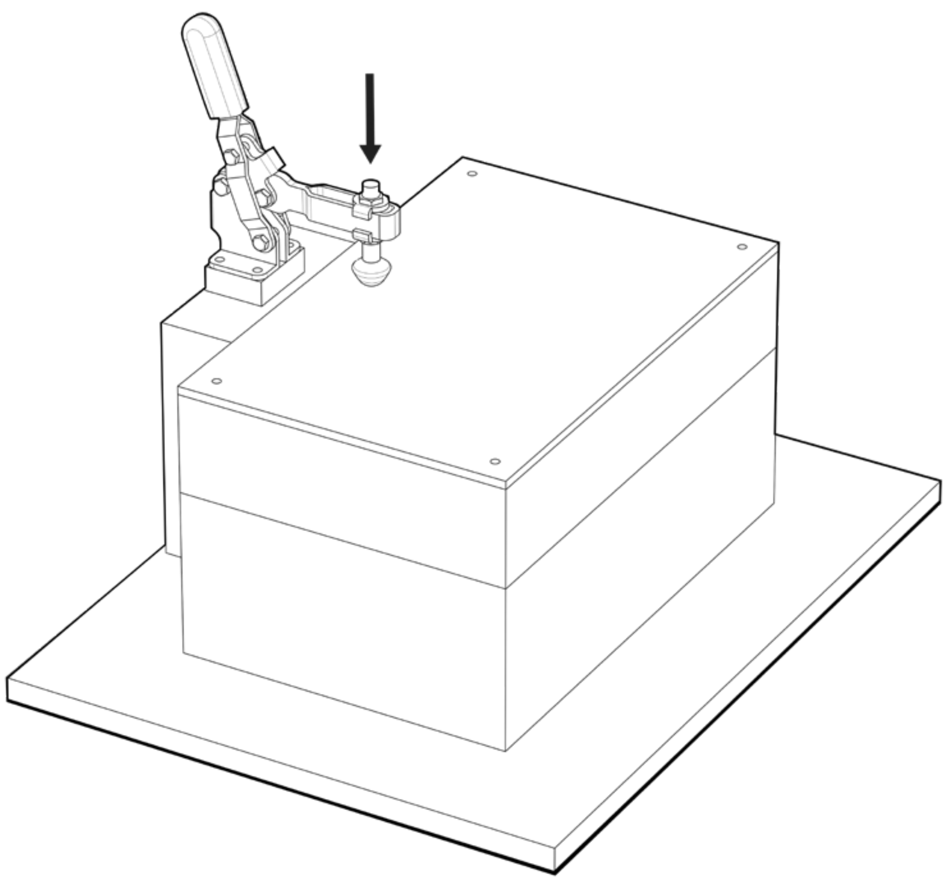
Fixtures

Since you are doing the process manually, it's unlikely that you are using precise fixtures to align parts or make sure they stay still during the screwdriving operation.

When going for an automated solution, however, it is mandatory to have fixtures that will not only hold the parts together but also locate the parts at a repeatable position. There is little room for error. Robots are excellent at being precise and repeatable, so you might as well make their environment the same.

For holding parts together, a manual clamp can do the job. A pneumatic toggle clamp can also be useful if the cell is more automated.

Consider whether you need to have a part confirmation process, such as via a sensor or limit switch. These sensing systems will make sure the robot won't start unless a part is present in the fixture.



Infeed and outfeed

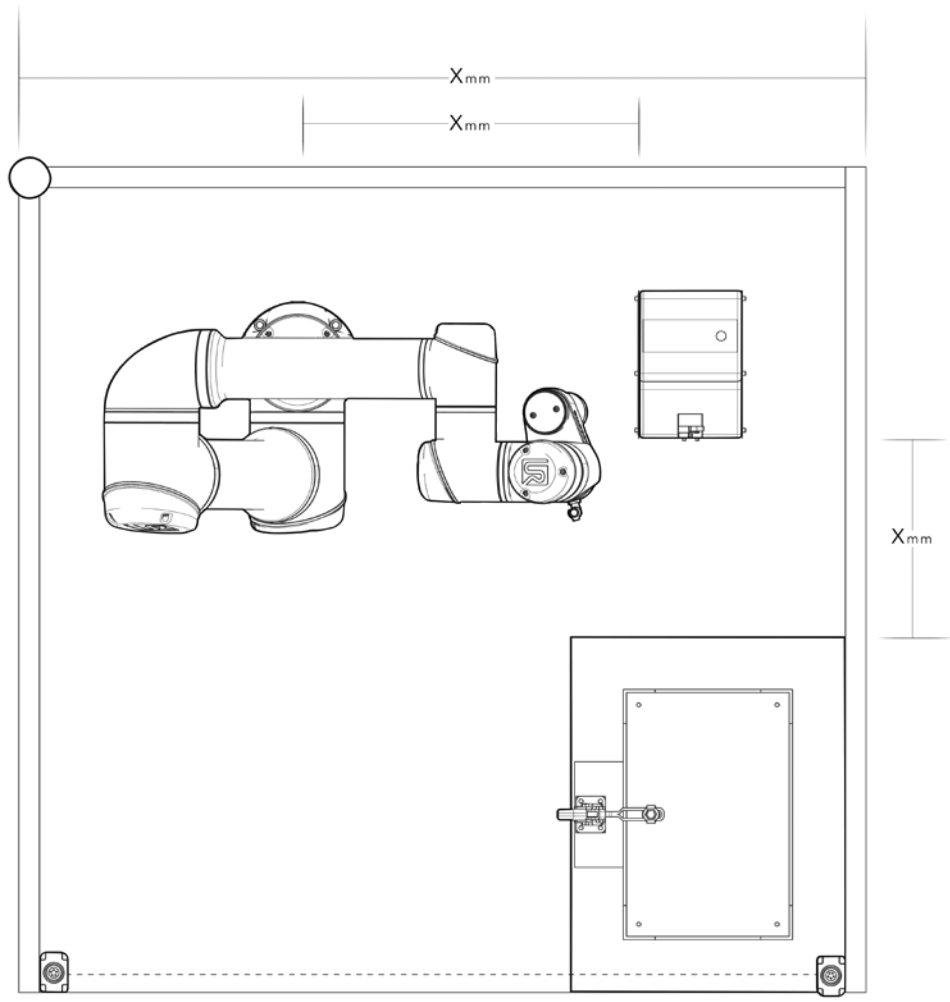
Your first screwdriving project needs to be simple. The best way to keep it simple is to have a dedicated screwdriving cell just for fastening parts together.

That means no other operation is done by the robot. Cell infeed and outfeed are done manually.

A human operator prepares the part to be fastened, installs the pre-assembly in the fixture, and lets the robot do the rest of the job. The outfeed is also handled by an operator.

It is possible to have the robot handle the parts, but you will need another end-effector to take the parts and assemble them together. You will also need a process to make sure all parts are aligned correctly. You might want to consider this—having the robot take parts from the infeed and place them in the outfeed by itself—for a second project, or after you have more experience with robot cells.

Layout



In most situations, the robot will be placed in between two workbenches, so the overall cell needs to be compact. Consider that the robot needs to move as little as possible to optimize the fastening cycle time.

In other words, if the robot has to carry the screw far from the screw feeder and travel the same distance back, it's a waste of time. Better to place the screw feeder and the part to be fastened close to each other.

For greater robot precision and repeatability, you need a rigid and stable stand. It is also important to have the parts, screw feeder, and robot base on the same workbench. This will avoid small deviations that could affect the robot program.

Safety

You can't talk about layout without considering safety. With most screwdriving applications, there will be a light curtain in front of the cell; the other three sides of the cell will have panels. This allows only front access to the process and virtually eliminates the risk of injury.

There are a few collaborative screwdrivers[1] on the market that can negate the need for such safety features. Using collaborative tooling, however, increases the screwdriver footprint and reduces the accessible area on the part to assemble. Because most of the screws are in constrained areas, it might be hard to fit a bulky screwdriver into these areas.

In other words, to reduce the screwdriver footprint because of the limited space, a slim screwdriver is a good choice. The downside is that you need to use a safety scanner to prevent injuries to nearby humans.

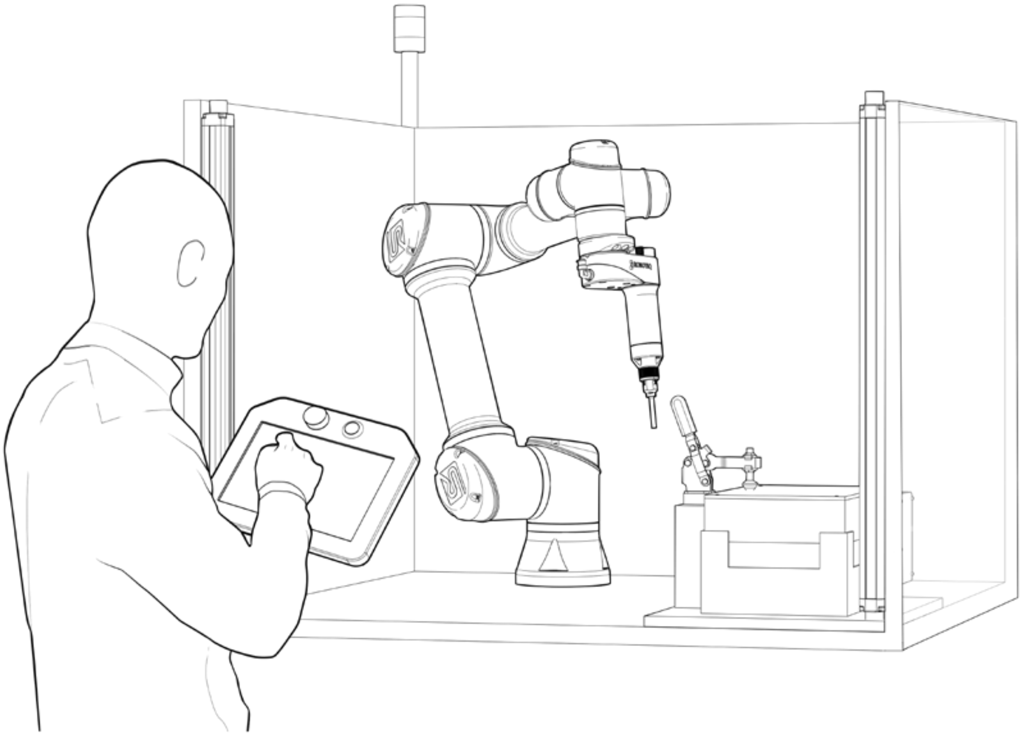
Information exchange

Finally, think about how information will be exchanged between the cell and the user. Some good ways to convey the cell state to the user include tower lights, flashing lights, and a dedicated human-machine interface.

Once the design is all set and approved, it is time to purchase or manufacture all that is needed to build your cell. Since this may take a few weeks or months depending on your choice, it is better to do it sooner rather than later.

1 Collaborative screwdriver: Screwdriver with safety function that eliminates the risk of injuries according to ISO/TS 15066 or equivalent safety standards.

Integrate



The integrate phase is where all the dots are connected. By following your design plan, you should be able to build the screwdriving cell and start your preliminary testing. There are just a few more things to consider before going into production.

Identification

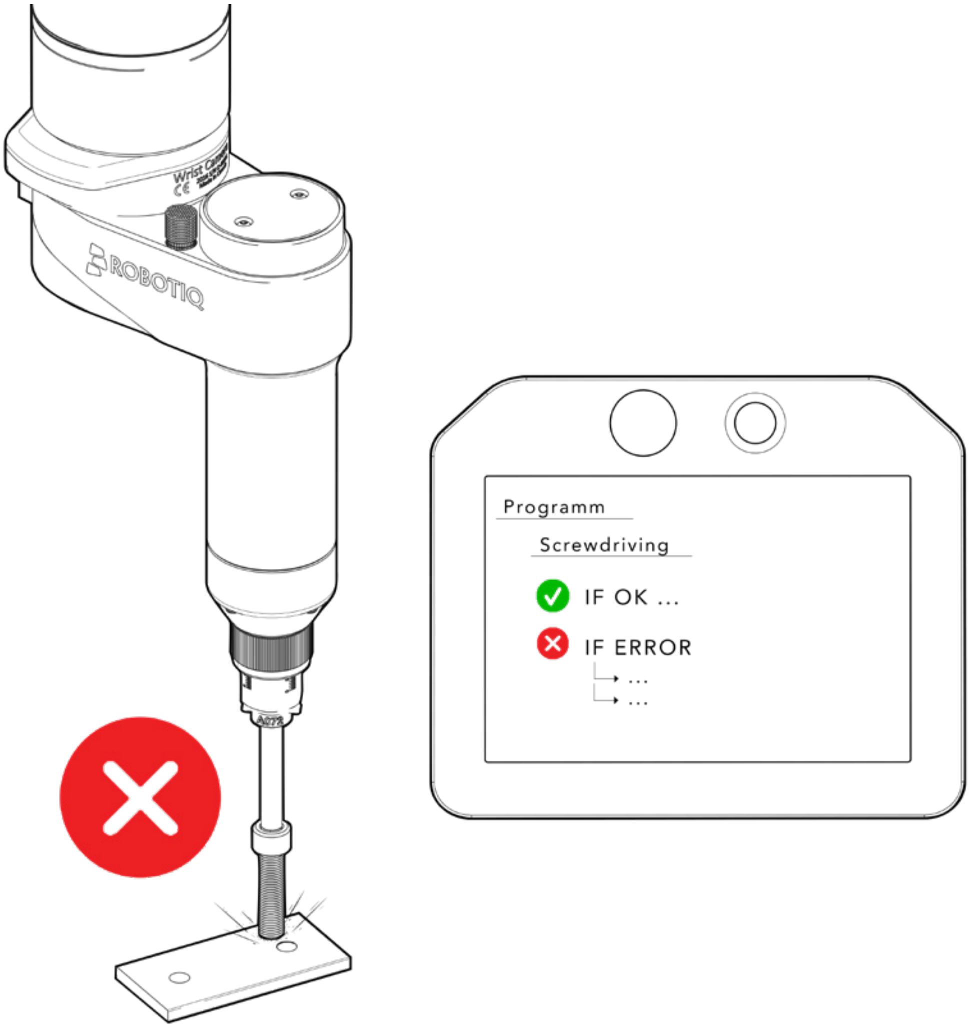
Even if you're not a programmer, you can easily program collaborative robots for a fastening operation. Keep in mind that the program needs to be readable for anyone else who might be debugging the cell later, so name your program sections wisely.

The same goes for any type of connection that goes in the robot controller. This part of the cell quickly becomes a cluster, so label all wires and cables to simplify debugging.

Error management

Even the best screwdriving cell may produce faulty fastening, because it is dependent on the quality of its input. The quality of the screws, bit wear, parts molded out of tolerance, misshapen threads, misaligned parts, and missing parts can all result in errors.

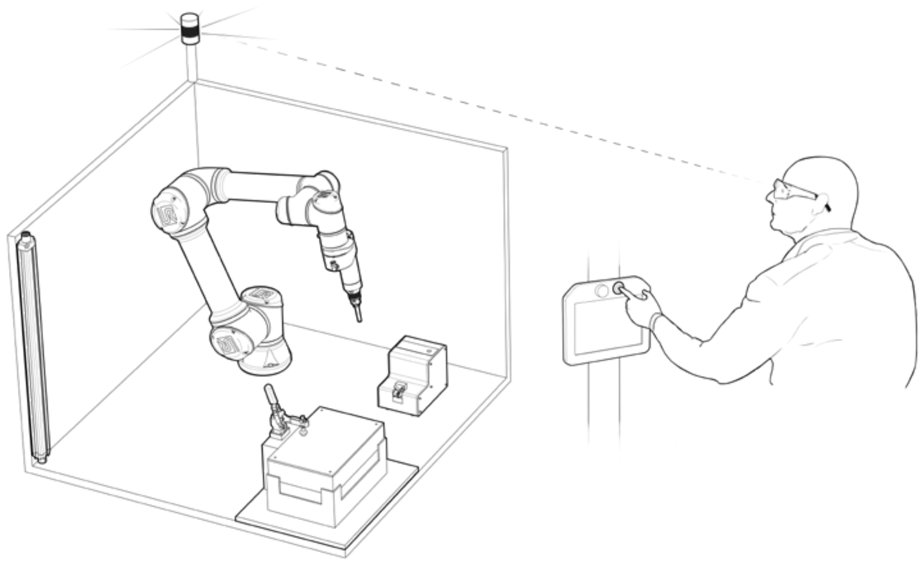
Part tolerance, screw characteristics, and screwdriver precision will all be variables that can create a faulty fastening operation. To mitigate these errors, the robot program should be built in such a way that it can generate a recovery sequence or signal to an operator it has encountered a problem.



For example, if torque is reached too early due to a cross thread, the robot should be precise enough to recognize its fault and trigger an "if" condition, since the distance has not been reached. The screwdriver can then unscrew the fastener and discard it.

That is just one example. You need to list all the errors that might happen and come up with ways to mitigate them. By doing so, you will ensure better reliability of your process and reduce the number and duration of cell stops.

Information exchange



Avoid any type of confusion in the information exchange that could paralyze the cell if operators are not trained to deal with the information. Restarting the cell should be a simple task. With collaborative robots in particular, there is usually a way an operator can restart the cell without having to call for assistance.

Example: One of the very first cells we installed made a loud sound to signal an error. The operator disconnected the speaker after the first error occurred because the sound was bothering them.

In any case, make sure that this information is easy to understand, visible from far away on the cell floor, and that operators are comfortable dealing with it.

Test the cell

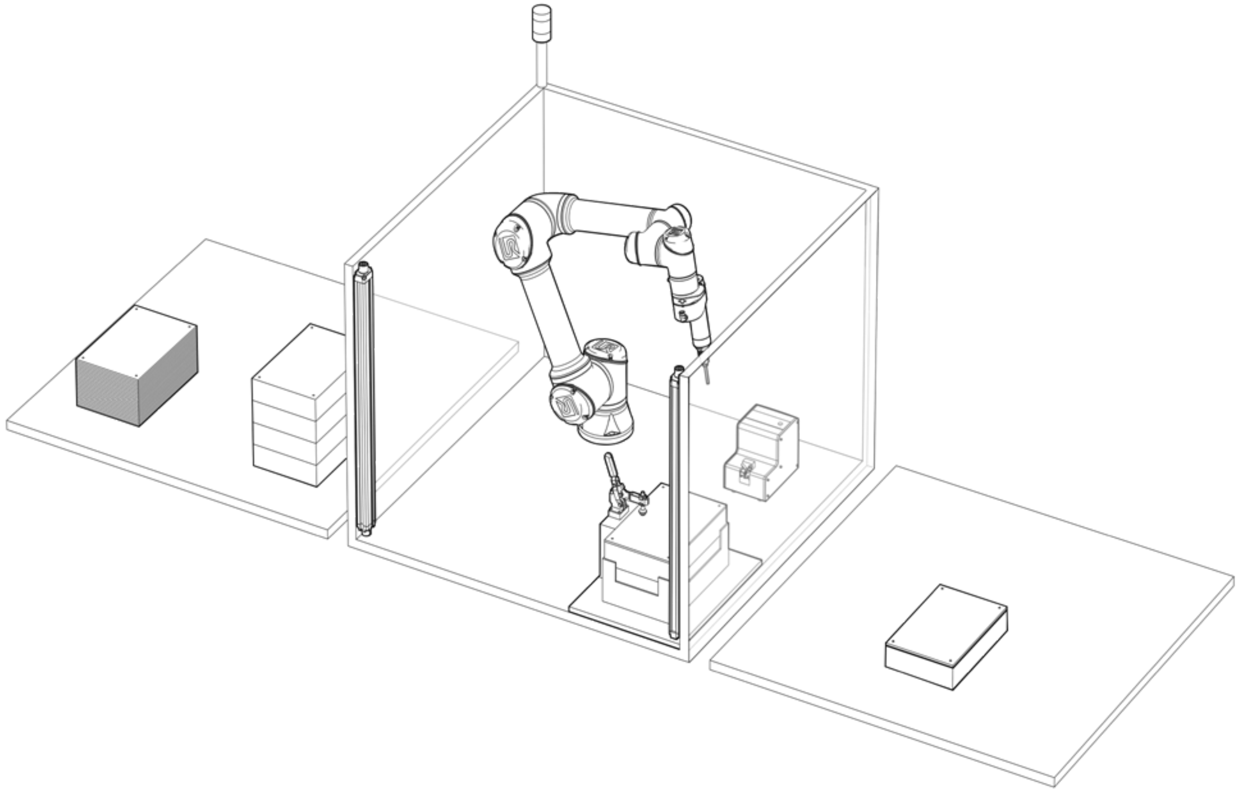
Before going all-out in production mode, make sure to test the cell and your robot program. Start by running the program at low speed and then increase the robot pace iteratively.

Next, do some advanced testing with potential users. Make sure to do extensive tests to ensure great cell repeatability and reliability. Train key personnel to debug, restart, or deal with other potential cell issues (like power cuts and emergency stops).

When people rush their cell into production too fast, they often have to remove the robot after a few days because it does not reach their expectations. Make sure every situation has been tested and that the cell can manage them all without error.

Operate

Cell operation should go smoothly if the design and integration have been done correctly.



Changeovers

Since the screwdriving cell has been proven to work robustly, it is now time to think about changeovers. The good news is you have already done most of the work. The cell is complete; you just need to make a few minor adjustments now in order to make it work for a different product.

Since you will basically be running the same program, you might as well just copy-paste your robot program so you don't have to start from scratch. The robot home position, and the different sensor IDs, for example, can all remain the same from one program to another.

In terms of hardware, if you have chosen a flexible solution for your screwdriver and screw feeder, only a minor adjustment is needed to switch from one screw to another. The bit and screw holder are probably the only things you need to change on the screwdriver side.

For the screw feeder, you will have to make various adjustments to fit a new screw. Make sure to dial everything in correctly before heading into production.

To make your changeover as fast and efficient as possible, think about having a second (or third) feeder ready and adjusted so you can just replace the current one. This eliminates the need to empty the current feeder and reduces downtime.

Make sure to revisit your fixture and layout according to the part you will now be fastening.

Calibration

Depending on your business's internal procedure, screwdrivers will have to be calibrated before production and recalibrated at a certain interval (typically yearly). Make sure you have the right tool to do this operation.

The vast majority of screwdrivers for robots have to be calibrated while attached to the robot. If calibrations are usually done in a separate building or room, have a plan to perform the calibration without removing the tooling from the robot.

Conclusion

Screwdriving can seem like a complex process to automate, but by following the lean robotic methodology (LINK) and using these tips you can be sure everything will run smoothly.

Above all, remember:

Start simple!

If you still feel hesitant about introducing a robot in your production line, a good way to mitigate the risk is to take a look at the Lean Robotics methodology. It contains a lot of useful information and you can use this book as a guide to your first integration.

Introducing a robot to do boring and tedious tasks is the best way to keep your operator safe, happy, and engaged in their job.

ROBOTIQ

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

DoF

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

Ask your question

robotiq.com | **leanrobotics.org**